Should the authentication process fail, the call controller circuit **100** is directed to an error handling block **134** which causes messages to be displayed at the telephone **12** to indicate that there was an authentication error. If the authentication process is successful, block **131** directs the call controller circuit **100** to determine whether or not the contents of the caller identifier field **60** of the SIP Invite message is a validly formatted IP address. If it is a valid IP address, then block **133** directs the call controller circuit **100** to associate a type code with the call to indicate that the call type is a third party invite.

If at block **131** the caller identifier field **60** contents do not identify an IP address, then block **135** directs the call controller circuit **100** to associate a type code with the call to indicate the call type is a regular SIP Invite message. Then, block **136** directs the call controller circuit **100** to establish a call ID by assigning the call ID provided in the call identifier field **65** of the SIP Invite message from the telephone **12**, and at block **138** the call controller circuit is directed to produce an RC Request message of the type shown in Figure **6** that includes that call ID. Referring back to Figure **5**, block **139** then directs the call controller circuit **100** to send the RC Request message to the routing controller **16**.

Referring to Figure **6**, an RC Request message is shown generally at **150** and includes a caller identifier field **152**, a callee identifier field **154**, a digest field **156**, a call ID field **158** and a type field **160**. The caller, callee, digest, and call identifier fields **152**, **154**, **156** and **158** contain copies of the caller, callee, digest parameters and call ID fields **60**, **62**, **64** and **65** of the SIP Invite message **59** shown in Figure **3**. The type field **160** contains the type code established at block **133** or **135** of Figure **5** to indicate whether the call is from a third party or system subscriber, respectively. The callee identifier field **154** may include a PSTN number or a system subscriber username as shown, for example.

Routing Controller

Referring to Figure **7**, the routing controller **16** is shown in greater detail and includes a routing controller processor circuit shown generally at **200**. The RC processor circuit **200** includes a microprocessor **202**, program memory **204**, a table memory **206** and an I/O interface **208**, all in communication with the processor. There may be a plurality of processor circuits (**202**), memories (**204**), etc.

The I/O interface **208** includes a database output port **210** through which a request to the database **18** (Figure **1**) can be made and includes a database response port **212** for receiving a reply from the database. The I/O interface **208** further includes an RC Request message input **214** for receiving the RC Request message from the call controller **14** and includes a routing message output **216** for sending a routing message back to the call controller **14**.

The program memory **204** includes blocks of codes for directing the RC processor circuit **200** to carry out various functions of the routing controller **16**. One of these blocks implements an RC Request message handler process **250** which directs the RC to produce a routing message in response to a received RC Request message of the type shown at **150** in Figure **6**. Referring back to Figure **7**, the program memory **204** further includes a Law Enforcement Authority (LEA) request message handler **1400** and an in-call intercept shut down route **1500**.

The RC Request message handler process **250** is shown in greater detail in Figures **8**A through **8**D.

RC Request Message Handler

Referring to Figure **8**A, the RC Request message handler process **250** begins with a first block **252** that directs the RC processor circuit **200** (Figure **7**) to store the contents of the RC Request message **150** (Figure **6**) in buffers. Block **254** then directs the RC processor circuit **200** to use the contents of the

caller identifier field **152** in the RC Request message shown in Figure **6**, to locate and retrieve a dialing profile for the caller from the database **18**.

The routing controller maintains, in the database, a dialing profile for each subscriber to the system. Referring to Figure **9**, an exemplary dialing profile is shown generally at **256** and includes system fields including a username field **258**, a domain field **260**, a national dialing digits (NDD) field **262**, an IDDs (IDD) field **264**, a country code field **266**, a local area codes field **267**, a caller minimum local length field **268**, a caller maximum local length field **270** and a reseller field **273**.

The exemplary dialing profile further includes lawful intercept related fields including a lawful intercept (LI) flag field **702**, at least one mediation device field **704**, at least one warrant ID field **706**, and intercept period start and stop date/time fields **708** and **710**. The LI flag field **702**, the warrant ID filed **706** and the LI start/stop fields **708** and **710** may be regarded as determination information fields for determining whether to intercept a communication involving the subscriber and the MD**1** address field **704** may be regarded as a destination information field for identifying a device to which intercepted communications involving the subscriber are to be sent.

The system fields (**258**, **260**, **262**, **264**, **266**, **267**, **268**, **270**, **273**) are assigned values by a system operator or are assigned automatically according to pre-defined algorithms (not shown) when a user registers with the system to become a subscriber. The lawful intercept fields (**702, 704, 706, 708, 710**) are assigned values in response to communications with one or more authorized devices and may be populated at any time regardless of whether or not communications involving the subscriber are in progress.

For example, referring back to Figure **1** the mediation device **31** may be regarded as an authorized device operated by a law enforcement authority **293**. A communications channel between the call controller **14** and the

mediation device **31** may be established to permit the mediation device to communicate with the call controller to cause the call controller to communicate with the routing controller **16** to find a subscriber record in the database **18** which is associated with a subscriber for which a warrant for lawful intercept has been obtained. For example, once a warrant identifying a user and permitting lawful intercept of that user's communications has been received by the law enforcement authority **293**, that authority can use its own computers to communicate with the mediation device **31** to cause the mediation device to communicate with the call controller **14** to cause the call controller to interact with the routing controller **16** to access a dialing profile (Figure **9**) for the user specified in the warrant and load the lawful intercept fields (**702, 704, 706, 708, 710**) with data that sets the lawful intercept flag field **702** to "on", stores an IP address of the mediation device **31** in the MD1 address field **704**, loads the warrant ID field **706** with an identifier of the warrant and loads the start and stop fields **708** and **710** with start and stop dates and times to specify a period during which lawful intercept of communications of the identified user may occur according to the warrant. Thus, intercept information is associated with the dialing profile by the routing controller, in response to information it receives from the call controller.

A plurality of groups of lawful intercept fields of the type shown may be added, each group being added by a different authorized device, for example, if several different law enforcement agencies operating the same or different mediation devices have warrants to monitor communications of a user. Alternatively the authorized device may include a handover interface operable to communicate with the call controller or routing controller to access the database to load the lawful intercept fields associated with a subscriber of interest.

An exemplary dialing profile for the Vancouver subscriber is shown generally at **276** in Figure **10** and indicates that the username field includes the

username **2001 1050 8667** which is the same as the contents of the username field **50** in the Vancouver telephone **12** shown in Figure **2**.

Referring back to Figure **10**, the domain field **260** includes a domain name as shown at **282**, including a supernode type identifier **284**, a location code identifier **286**, a system provider identifier **288** and a top level domain identifier **290**, identifying a domain or supernode associated with the user identified by the contents of the username field **258**.

In this embodiment, the supernode type identifier **284** includes the code "sp" identifying a supernode and the location code identifier **286** identifies the supernode as being in Vancouver (YVR). The system provider identifier **288** identifies the company supplying the service and the top level domain identifier **290** identifies the "com" domain.

The national dialing digit (NDD) field **262** in this embodiment includes the digit "**1**" and, in general, includes a digit specified by the International Telecommunications Union – Telecommunications Standardization Sector (ITU-T) E.**164** Recommendation which assigns national dialing digits to certain countries. Herein numbering sequences compliant with this standard will be regarded as "E.**164**" numbers.

The International Dialing Digit (IDD) field **264** includes the code **011** and in general includes a code assigned by the ITU-T according to the country or geographical location of the user.

The country code field **266** includes the digit "**1**" and in general includes a number assigned by the ITU-T to represent the country in which the user is located.

The local area codes field **267** includes the numbers **604** and **778** and generally includes a list of area codes that have been assigned by the ITU-T

to the geographical area in which the subscriber is located. The caller minimum and maximum local number length fields **268** and **270** hold the number **10** representing minimum and maximum local number lengths permitted in the area code(s) specified by the contents of the local area codes field **267**. The reseller field **273** holds a code identifying a retailer of the telephone services, and in the embodiment shown, the retailer is "Klondike".

Initially, the lawful intercept fields shown in Figure **9** might not be included in the dialing profile and may be added as described above, by the mediation device **31**, in the event a warrant is obtained to intercept the user's calls. Alternatively, the lawful intercept fields may be included, but populated with null values until modified by a mediation device **31**.

A dialing profile of the type shown at **256** in Figure **9** is produced whenever a user registers with the system or agrees to become a subscriber to the system. Thus, for example, a user wishing to subscribe to the system may contact an office maintained by a system operator and personnel in the office may ask the user certain questions about his location and service preferences, whereupon tables can be used to provide office personnel with appropriate information to be entered into the username, domain, NDD, IDD, country code, local area codes and caller minimum and maximum local length fields **258**, **260**, **262**, **264**, **266**, **267**, **268**, **270** to establish a dialing profile for the user.

Referring to Figures **11** and **12**, dialing profiles for subscribers in Calgary and London, respectively for example, are shown.

In addition to creating dialing profiles, optionally when a user registers with the system, a direct inward dialing (DID) record of the type shown at **268** in Figure **13** is added to a direct inward dialing table in the database **18** to associate the username with a host name of the supernode with which the user is associated and with an E.**164** number on the PSTN network.

In this embodiment, the DID bank table records include a username field **281**, a user domain field **272** and a DID field **274**, for holding the username, hostname of the supernode, and an E.**164** number respectively.

A DID bank table record for the London subscriber is shown generally at **291** in Figure **14**.

In addition to creating dialing profiles and DID records when a user registers with the system, call blocking records of the type shown in Figure **26**, call forwarding records of the type shown in Figure **28** and voicemail records of the type shown in Figure **30** may be stored in the database **18** when a new subscriber is added to the system.

Referring back to Figure **8**A, after being directed at block **254** to retrieve a dialing profile for the caller, a dialing profile such as shown at **276** in Figure **10** is retrieved and the RC processor circuit **200** is directed to perform certain checks on the callee identifier provided by the contents of the callee identifier field **154** of the RC Request message shown in Figure **6**. These checks are shown in greater detail in Figure **8**B.

Referring to Figure **8**B, the RC processor circuit **200** is directed to a first block **257** that causes it to determine whether a digit pattern of the callee identifier **154** provided in the RC Request message includes a pattern that matches the contents of the IDD field **264** in the caller dialing profile **276** shown in Figure **10**. If so, then block **259** directs the RC processor circuit **200** to set a call type code identifier (not shown) to indicate that the call is a long distance call, e.g., from the Vancouver subscriber to the London subscriber, and block **261** directs the RC processor circuit **200** to produce a reformatted callee identifier by reformatting the callee identifier into a predetermined target format. In this embodiment, this is done by removing the pattern of digits matching the IDD field contents **264** of the caller dialing profile **276** to effectively shorten the

number. Then, block **263** directs the RC processor circuit **200** to determine whether or not the reformatted callee identifier meets criteria establishing it as a number compliant with the E.**164** Recommendation set by the ITU-T and if the length does not meet this criteria, block **265** directs the RC processor circuit **200** to send back to the call controller **14** a message indicating that the length of the call identifier is not correct. The process **250** is then ended. At the call controller **14**, routines may respond to the incorrect length message by transmitting a message back to the telephone **12** to indicate that an invalid number has been dialed.

Still referring to Figure **8**B, if the length of the reformatted callee identifier meets the criteria set forth at block **263**, block **269** directs the RC processor circuit **200** to determine whether or not the reformatted callee identifier is associated with a direct inward dialing (DID) bank table record such as shown at **268** in Figure **13**.

An exemplary DID bank table record entry for the London callee is shown generally at **291** in Figure **14**. The username field **281** and user domain field **272** are as specified in the username and user domain fields **258** and **260** of the dialing profile **276** shown in Figure **12**. The contents of the DID field **274** include an E.**164** telephone number including a country code **283**, an area code **285**, an exchange code **287** and a number **289**. If the user has multiple telephone numbers, then multiple records of the type shown at **291** would be included in the DID bank table in the database **18**, each having the same username and user domain, but different DID field **274** contents reflecting the different telephone numbers associated with that user.

Referring back to Figure **8**B, at block **269**, if the RC processor circuit **200** finds that the reformatted callee identifier produced at block **261** is found in a record in the DID bank table, then the callee is a subscriber to the system and block **279** directs the RC processor circuit **200** to copy the contents of the corresponding username field **270** into a callee ID buffer (not shown). Thus,

the RC processor circuit **200** locates a subscriber username associated with the reformatted callee identifier. The processor is then directed to block **275** at point B in Figure **8**A.

Subscriber to Subscriber Calls Between Different Nodes

Referring back to Figure **8**A, block **275** then directs the RC processor circuit **200** to determine whether or not the subscriber username is associated with the same supernode as the caller. To do this, the RC processor circuit **200** determines whether or not the continent code (**61**) of the username stored in the callee ID buffer is the same as the continent code (**61**) of the username of the caller specified by the caller identifier field **152** of the RC Request message shown in Figure **6**. If they are not the same, block **277** directs the RC processor circuit **200** to set a call type flag (not shown) to indicate that the call is a cross-domain call. Then, block **350** directs the RC processor circuit **200** to produce a routing message identifying the supernode in the system with which the callee is associated and to set a TTL for the call to the maximum value of **99999**. The supernode in the system, with which the callee is associated, is determined by using the callee username stored in the callee ID buffer to address a supernode table having records of the type as shown at **370** in Figure **17**.

Referring to Figure **17**, each prefix to supernode table record **370** has a prefix field **372** and a supernode address field **374**. The prefix field **372** includes the first n digits of the callee identifier. In this case n=**1**. The supernode address field **374** holds a code representing the IP address or a fully qualified domain name of the supernode associated with the code stored in the prefix field **372**. Referring to Figure **18**, for example, if the prefix is **4**, the supernode address associated with that prefix is sp.lhr.digifonica.com, identifying the London supernode **21**, for example.

Referring to Figure **15**, a generic routing message is shown generally at **352** and includes a supplier prefix field **354**, a delimiter field **356**, a callee field **358**,

at least one route field **360**, a time-to-live (TTL) field **362** and other fields **364**. The supplier prefix field **354** holds a code for identifying supplier traffic. The delimiter field holds a symbol that delimits the supplier prefix code from the callee field **358** and in this embodiment, the symbol is a number sign (#). The route field **360** holds a domain name or an IP address of a gateway or supernode that is to carry the call and the TTL field **362** holds a value representing the number of seconds the call is permitted to be active, based on subscriber available minutes and other billing parameters, for example.

Referring to Figure **8A** and Figure **16**, in this example the routing message produced by the RC processor circuit **200** at block **350** is shown generally at **366** and includes only a callee field **358**, a route field **360** and a TTL field **362**.

The callee field **358** holds the full username of the callee and the route field **360**, shown in Figure **15**, contains the identification of the domain with which the callee is associated, i.e., sp.lhr.digifonica.com.

Having produced the routing message **366** as shown in Figure **16**A, referring back to Figure **8A**, block **351** then directs the RC processor circuit **200** to check the caller dialing profile (see Figure **9**) to determine whether or not it contains lawful intercept fields (**702, 704, 706, 708, 710**) and if so, to determine whether or not the determination information contained therein meets intercept criteria. The intercept criteria may be that the lawful intercept flag field **702** (Figure **9**) contains a flag indicating lawful intercept is enabled and whether the current date and time is within the period specified by the LI start date/time field contents **708** and the LI stop date/time field contents **710,** for example. If the intercept criteria are met, block **353** directs the RC processor circuit **200** to append the contents of the lawful intercept fields **702, 704, 706, 708, 710** to the routing message produced at block **350** to produce a routing message as shown in Figure **16**A. Generally, the determination of whether or not the destination information meets intercept criteria is done prior to producing the routing message so that when the intercept criteria are met,

at least some of the intercept information, in this embodiment all of it, can be included in the routing message.

If at block **351** in Figure **8A**, it is determined there are no lawful intercept fields associated with the caller dialing profile or that the intercept criteria are not met, the processor does not append any lawful intercept fields to the routing message produced at block **350** in Figure **8A** and the routing message shown in Figure **16** is sent to the call controller **14** as shown at block **380**. If the lawful intercept fields have been appended, block **380** directs the RC processor circuit **200** to send the routing message shown in Figure **16A** to the call controller **14** (Figure **1**).

Referring back to Figure **8B**, if at block **257**, the callee identifier specified by the contents of the callee field **154** of the RC Request message shown in Figure **6** does not begin with an IDD, block **381** directs the RC processor circuit **200** to determine whether or not the callee identifier begins with the same national dial digit code as assigned to the caller. To do this, the processor is directed to refer to the caller dialing profile shown in Figure **10**. In the embodiment shown, the NDD code **262** is the digit **1**. Thus, if the callee identifier begins with the digit **1**, the RC processor circuit **200** is directed to block **382** in Figure **8B**.

Block **382** directs the RC processor circuit **200** to examine the callee identifier to determine whether or not digits following the NDD code identify an area code that is the same as any of the area codes identified in the local area codes field **267** of the caller dialing profile **276** shown in Figure **10**. If not, block **384** directs the RC processor circuit **200** to set a call type variable (not shown) to a code indicating the call is a national code. If the digits identify an area code that is the same as a local area code associated with the caller, block **386** directs the RC processor circuit **200** to set the call type variable to indicate that the call type is a local call, national style. After executing blocks **384** or **386**, block **388** directs the RC processor circuit **200** to format the

number dialed by removing the national dial digit (NDD) and prepending a caller country code identified by the country code field **266** of the caller dialing profile shown in Figure **10**. The RC processor circuit **200** is then directed to block **263** to perform the processes described above beginning at block **263**.

If at block **381**, the callee identifier does not begin with an NDD code, block **390** directs the RC processor circuit **200** to determine whether the callee identifier begins with digits that identify the same area code as the caller. Again, the reference for this is the caller profile shown in Figure **10** and the RC processor circuit **200** determines whether or not the first few digits in the callee identifier identify an area code identified by the local area code field **267** of the caller profile. If so, then block **392** directs the RC processor circuit **200** to set the call type to a code indicating the call is a local call and block **394** directs the RC processor circuit **200** to prepend the caller country code to the callee identifier, the caller country code being determined from the country code field **266** in the caller profile shown in Figure **10**. The RC processor circuit **200** is then directed to block **263** for processing as described above beginning at block **263**.

If at block **390**, the callee identifier does not have the same area code as the caller, block **396** directs the RC processor circuit **200** to determine whether the callee identifier has the same number of digits as the number of digits indicated in either the caller minimum local number length field **268** or the caller maximum local number length field **270** of the caller profile shown in Figure **10**. If so, then block **398** directs the RC processor circuit **200** to set the call type to local and block **400** directs the processor to prepend to the callee identifier the caller country code as indicated by the country code field **266** of the caller profile shown in Figure **10** followed by the caller area code as indicated by the local area code field **267** of the caller profile shown in Figure **10**. The RC processor circuit **200** is then directed to block **263** for further processing as described above beginning at block **263**.

If at block **396**, the callee identifier has a length that does not match the length specified by the contents of the caller minimum local number length field **268** or the caller maximum local number length field **270**, block **402** directs the RC processor circuit **200** to determine whether or not the callee identifier identifies a valid username. To do this, the RC processor circuit **200** searches through the database of dialing profiles to find a dialing profile having username field contents **258** that match the callee identifier. If no match is found, block **404** directs the RC processor circuit **200** to send an error message back to the call controller (**14**). If at block **402**, a dialing profile having a username field **258** that matches the callee identifier is found, block **406** directs the RC processor circuit **200** to set the call type to a code indicating the call is a network call and the processor is directed to block **275** of Figure **8**A, to continue processing the RC message handler process **250**.

From Figure **8**B, it will be appreciated that there are certain groups of blocks of codes that direct the RC processor circuit **200** to determine whether the callee identifier has certain features such as an IDD code, a NDD code, an area code and a length that meet certain criteria and to reformat the callee identifier as necessary into a predetermined target format including only a country code, area code, and a normal telephone number, for example, to cause the callee identifier to be compatible with the E.**164** number plan standard, in this embodiment. This enables the RC processor circuit **200** directed by block **279** to have a consistent format of callee identifiers for use in searching through the DID bank table records of the type shown in Figure **13** to determine how to route calls for subscriber to subscriber calls on the same system.

Subscriber to Non-Subscriber Calls

Not all calls will be subscriber-to-subscriber calls and this will be detected by the RC processor circuit **200** when it executes block **269** of Figure **8**B, and does not find a record that is associated with the callee in the DID bank table. When this occurs, the RC processor circuit **200** is directed to block **408** which

causes it to set the callee identifier equal to the reformatted callee identifier, i.e., the number compatible with the E.**164** standard. Then, block **410** directs the RC processor circuit **200** to address a master list having records of the type shown in Figure **19**.

Each master list record includes a master list ID field **500**, a dialing code field **502**, a country code field **504**, a national sign number field **506**, a minimum length field **508**, a maximum length field **510**, a NDD field **512**, an IDD field **514** and a buffer rate field **516**.

The master list ID field **500** holds a unique code such as **1019**, for example, identifying a route identification (route ID). The dialing code field **502** holds a predetermined number pattern which the RC processor circuit **200** uses at block **410** in Figure **8**B to find the master list record having a dialing code matching the first few digits of the reformatted callee identifier. The country code field **504** holds a number representing the country code associated with the record and the national sign number field **506** holds a number representing the area code associated with the record. (It will be observed that the dialing code is a combination of the contents of the country code field **504** and the national sign number field **506**.) The minimum length field **508** holds a number representing the minimum number of digits that can be associated with the record and the maximum length field **51** holds a number representing the maximum number of digits in a number with which the record may be compared. The NDD field **512** holds a number representing an access code used to make a call within the country specified by the contents of the country code field **504** and the IDD field **514** holds a number representing the international prefix needed to dial a call from the country indicated by the country code.

Thus, for example, a master list record may have a format as shown in Figure **20** with exemplary field contents as shown.

Referring back to Figure **8B**, using the country code and area code portions of the reformatted callee identifier that has been formatted for compatibility with the E.**164** standard, block **410** directs the RC processor circuit **200** to find a master list record such as the one shown in Figure **20** having a dialing code that matches the country code and area code of the callee identifier. Thus, in this example, the RC processor circuit **200** would find a master list record having an ID field with the number **1019**. This number may be also referred to as a route ID. Thus, a route ID number is found in the master list record associated with a predetermined number pattern in the reformatted callee identifier.

After execution of block **410** in Figure **8B**, the process **250** continues as shown in Figure **8D**. Referring to Figure **8D**, block **412** directs the RC processor circuit **200** to use the route ID number to locate at least one supplier record identifying a supplier operable to supply a communications link for this route. To do this, block **412** directs the RC processor circuit **200** to search a supplier ID table having records of the type shown in Figure **21**.

Referring to Figure **21**, the supplier list records include a supplier ID field **540**, a route ID field **542**, an optional prefix field **544**, a route identifier field **546**, a NDD/IDD rewrite field **548** and a rate field **550**. The supplier ID field **540** holds a code identifying the name of the supplier and the route ID field **542** holds a code for associating the supplier record with a route, and hence with a master list record. The prefix field **544** holds a string used to identify the supplier traffic and the route identifier field **546** holds an IP address of a gateway operated by the supplier indicated by the supplier ID field **540**. The NDD/IDD rewrite field **548** holds a code and the rate field **550** holds a code indicating the cost per second to the system operator to use the route provided by the gateway specified by the contents of the route identifier field **546**. Exemplary supplier records are shown in Figures **22**, **23** and **24** for the suppliers shown in Figure **1** which may include Telus, Shaw and Sprint, respectively, for example.

Referring back to Figure 8D, at block **412** the RC processor circuit **200** finds all supplier records that identify the route ID found at block **410** of Figure **8B**.

Referring back to Figure **8**D, block **560** directs the RC processor circuit **200** to begin to produce routing messages of the type shown in Figure **16**. To do this, the RC processor circuit **200** loads a routing message buffer as shown in Figure **25** with a supplier prefix of the least costly supplier where the least costly supplier is determined from the rate fields **550** of the records associated with respective suppliers.

Referring to Figures **22-24**, in the embodiment shown, the supplier "Telus" has the lowest number in the rate field **550** and therefore the prefix **4973** associated with that supplier is loaded into the routing message buffer shown in Figure **25** first. The prefix **4973** is then delimited by the number sign and the reformatted callee identifier is next loaded into the routing message buffer. Then, the contents of the route identifier field **546** of the record associated with the supplier Telus are added to the message after an @ sign delimiter and then block **564** in Figure **8**D directs the RC processor circuit **200** to get a TTL value, which in this embodiment may be **3600** seconds, for example. Block **566** then directs the RC processor circuit **200** to load this TTL value in the routing message buffer shown in Figure **25**. Accordingly, the first part of the routing message is shown generally at **570** in Figure **25**.

Referring back to Figure **8**D, block **568** directs the RC processor circuit **200** back to block **560** and causes it to repeat blocks **560**, **562**, **564** and **566** for each successive supplier until the routing message buffer is loaded with information pertaining to each supplier. Thus, the second portion of the routing message is shown at **572** in Figure **25** and this second portion relates to the second supplier identified by the record shown in Figure **23** and referring back to Figure **25**, the third portion of the routing message is shown at **574** which is associated with a third supplier as indicated by the supplier

record shown in Figure **24**. Consequently, referring to Figure **25**, the routing message buffer holds a routing message identifying a plurality of different suppliers able to provide gateways to establish a communication link to permit the caller to contact the callee. Each of the suppliers is identified, in ascending order according the rates contained in the rate fields **550** of the supplier list records shown in Figures **22-24**, in this embodiment. Other criteria for determining the order in which suppliers are listed in the routing message may include preferred supplier priorities which may be established based on service agreements, for example. In this case additional fields may be provided in respective supplier records to hold values representing supplier priority.

After the routing message buffer has been loaded as shown in Figure **25**, block **567** directs the RC processor circuit **200** to check the caller dialing profile shown in Figure **10** to determine whether or not it contains lawful intercept fields as shown in Figure **9**, and if so, to determine whether or not the intercept criteria are met by checking whether the lawful intercept flag field **702** contains a flag indicating that lawful intercept is enabled and checking whether the current date and time are within the period specified by the LI start date/time field contents **708** and the LI stop date/time field contents **710**. If the intercept criteria are met, block **569** directs the RC processor circuit **200** to append the contents of the lawful intercept fields **702**, **704**, **706**, **708**, **710** to the routing message stored in the routing message buffer, as shown in Figure **25**A. Again, the determination of whether or not the destination information meets intercept criteria is done prior to producing the routing message so that when the intercept criteria are met, at least some of the intercept information, in this embodiment all of it, can be included in the routing message.

If at block **567**, it is determined there are no lawful intercept fields associated with the caller dialing profile shown in Figure **10** or that the intercept criteria are not met, the RC processor circuit **200** does not append any lawful

intercept fields to the routing message stored in the routing message buffer shown in Figure **25**.

Block **568** then directs the RC processor circuit **200** to send the contents of the routing message buffer, i.e. the routing message shown in Figure **25** or **25A**, to the call controller **14** in Figure **1**.

Subscriber to Subscriber Calls Within the Same Node

Referring back to Figure **8A**, if at block **275**, the callee identifier stored in the callee ID buffer has a prefix that identifies the same supernode as that associated with the caller, block **600** directs the RC processor circuit **200** to use the callee identifier to locate and retrieve a dialing profile for the callee identified by the callee identifier. The dialing profile is of the type shown in Figure **9**, and may contain data as shown in Figure **11**, for example. Block **602** of Figure **8A** directs the RC processor circuit **200** to get call block, call forward and voicemail tables from the database **18** based on the username identified in the callee profile retrieved by the RC processor circuit at block **600**. Call block, call forward and voicemail tables have records as shown in Figures **26**, **28** and **30** for example.

Referring to Figure **26**, the call block records include a username field **604** and a block pattern field **606**. The username field holds a username matching the username in the username field **258** of the dialing profile associated with the callee and the block pattern field **606** holds one or more E.**164**-compatible numbers or usernames identifying PSTN numbers or system subscribers from whom the subscriber identified by the contents of the username field **604** does not wish to receive calls.

Referring back to Figure 8A and referring to Figure **27**, block **608** directs the RC processor circuit **200** to determine whether or not the caller identifier matches a block pattern stored in the block pattern field **606** of the call block record associated with the callee identified by the contents of the username

field **604** in Figure **26**. If the caller identifier matches a block pattern stored in the block pattern field **606**, block **610** directs the RC processor circuit **200** to send a drop call or non-completion message to the call controller (**14**) and the process is ended. If the caller identifier does not match a block pattern associated with the callee, block **612** directs the RC processor circuit **200** to determine whether or not call forwarding is required.

Referring to Figure **28**, records in the call forwarding table include a username field **614**, a destination number field **616**, a destination number field **616** and a sequence number field **618**. The username field **614** stores a code representing a subscriber with which the record is associated. The destination number field **616** holds a username or number representing a number to which the current call should be forwarded and the sequence number field **618** holds an integer number indicating the order in which the username associated with the corresponding destination number field **616** should be attempted for call forwarding. The call forwarding table may have a plurality of records for a given user. The RC processor circuit **200** uses the contents of the sequence number field **618** to consider the records for a given subscriber in order. As will be appreciated below, this enables the call forwarding numbers to be tried in a ordered sequence.

Referring back to Figure **8**A and referring to Figure **28**, if at block **612** in Figure **8**A, the call forwarding record for the callee identified by the callee identifier contains no contents in the destination number field **616** and accordingly no contents in the sequence number field **618**, there are no call forwarding entries and the RC processor circuit **200** is directed to load the routing message buffer shown in Figure **32** with the callee username and domain, as shown at **650** in Figure **32**. The processor is then directed to block **620** in Figure **8**C.

If there are contents in the destination number field of the call forwarding record as shown in Figure **29**, block **622** shown in Figure **8**A directs the RC

processor circuit **200** to search the dialing profile table to find a dialing profile record of the type shown in Figure **9**, for the user identified in the destination number field **616** in the call forwarding table record of Figure **29** and to store the contents of the destination number field in the routing message buffer shown in Figure **32**. The RC processor circuit **200** is then directed to load the contents of the domain field **260** shown in Figure **9** associated with the username specified by the contents of the destination number field **616** of Figure **29** into the routing message buffer as shown at **652** in Figure **32**. This process is repeated for each call forwarding record associated with the callee identified by the callee identifier to add to the routing message buffer all call forwarding usernames and domains associated with the callee.

Referring to Figure **8**C, at block **620** the processor is directed to determine whether or not the user identified by the callee identifier has paid for voicemail service and this is done by checking to see whether or not a flag is set in a voicemail record of the type shown in Figure **30** in a voicemail table stored in the database **18** in Figure **1**.

Referring to Figure **30**, voicemail table records include a username field **624**, a voicemail server field **626**, a seconds-to-voicemail field **628** and an enable field **630**. The username field **624** stores the username of the subscriber who purchased the service. The voicemail server field **626** holds a code identifying an IP address or a fully qualified domain name (FQDN) of a voicemail server associated with the subscriber identified by the username field **624**. The seconds-to-voicemail field **628** holds a code identifying the time to wait before engaging voicemail and the enable field **630** holds a code representing whether or not voicemail is enabled for the user identified by the contents of the username field **624**. Therefore, referring back to Figure **8**C, at block **620** the processor searches for a voicemail record as shown in Figure **31** having username field **624** contents matching the callee identifier and looks at the contents of the enabled field **630** to determine whether or not voicemail is enabled. If voicemail is enabled, then block **640** in Figure **8C** directs the

processor to store the contents of the voicemail server field **626** of Figure **31** and the contents of the seconds to voicemail field **628** of Figure **31** in the routing message buffer as shown at **654** in Figure **32**. Referring back to Figure **8**C, block **642** then directs the processor to get time to live (TTL) values for each route specified by the routing message according to any of a plurality of criteria such as, for example, the cost of routing and the user's account balance. These TTL values are then appended to corresponding routes already stored in the routing message buffer.

Block **644** of Figure **8**C then directs the RC processor circuit **200** to store the IP address of the current supernode in the routing message buffer as shown at **656** in Figure **32**. An exemplary routing message is shown in the routing message buffer shown in Figure **32**.

Block **645** of Figure **8**C then directs the processor to check the caller dialing profile shown in Figure **10** to determine whether or not it contains lawful intercept fields of the type shown in Figure **9** and if so, to determine whether or not the intercept criteria are met. In this embodiment, this includes determining whether the lawful intercept flag field **702** contains a flag indicating that lawful intercept is enabled and checking whether the current date and time is within the period specified by the LI start date/time field contents **708** and the LI stop date/time field contents **710**. If the intercept criteria are met, block **647** directs the RC processor circuit **200** to append the contents of the lawful intercept fields **702**, **704**, **706**, **708**, **710** to the routing message shown in Figure **32**A to produce a routing message with lawful intercept field contents, as shown in Figure **32**A. Again, the determination of whether or not the destination information meets intercept criteria is done prior to producing the routing message so that when the intercept criteria are met, at least some of the intercept information, in this embodiment all of it, can be included in the routing message.

Referring back to Figure **8**C, if at block **645**, it is determined there are no lawful intercept fields associated with the caller dialing profile of Figure **10** or that the intercept criteria are not met after producing the routing message shown in Figure **32**A the processor is directed to block **649** which causes the processor to check the callee dialing profile shown in Figure **11** to determine whether or not it contains lawful intercept fields of the type shown in Figure **9** and if so, to determine whether or not the intercept criteria are met by checking whether the current date and time is within the period specified by the LI start date/time field contents **708** and the LI stop date/time field contents **710** of the callee dialing profile. If the intercept criteria are met, block **651** directs the RC processor circuit **200** to append the contents of the lawful intercept fields **702**, **704**, **706**, **708**, **710** associated with the callee dialing profile to the routing message shown in Figure **32**A to produce a routing message. If at block **649** of Figure **8**C, it is determined there are no lawful intercept fields associated with the callee dialing profile or that the intercept criteria are not met, no lawful intercept fields associated with the callee are appended to the routing message shown in Figure **32** or **32**A. Referring back to Figure **8**C, block **646** then directs the RC processor circuit **200** to send the routing message to the call controller **14**.

Response to Routing Message

Referring back to Figure **1**, the routing message, whether of the type shown in Figures **16**, **16**A, **25**, **25**A, **32**, **32**A or **32**B, is received at the call controller **14**. Referring to Figure **33**, when a routing message is received at the call controller, the routing message handler **122** is invoked at the call controller. The routing message handler is shown in detail in Figure **33**.

Referring to Figure **33**, the routing message handler begins with a first block **1200** that directs the processor circuit to determine whether the routing message includes lawful intercept fields. If not, the processor is directed to block **1206** which causes it to invoke a call handling routine shown in Figure **34**. Referring to Figure **34**, as a first step in the call handling routine, a

message **1100** is sent from the call controller **14** to the media relay **17**, the message including the caller telephone IP address and UDP port as determined from the caller IP address field **67** and caller UDP port field **69** in the SIP Invite message shown in Figure **3.**

The specific media relay **17** to which the message **1100** is sent may be selected from a pool of available media relays and such media relays may be at any geographical location. The purpose of the message **1100** is to advise the media relay that a call is desired to be set up to communicate with the IP address and UDP number of the caller telephone.

A media relay selected from media relays located at a geographical location that facilitates communication at a desired quality of service between the media relay **17** and the caller telephone **12** and callee telephone **15** may provide the best service. Alternatively, media relays may be pre-assigned or pre-associated with users by including and populating media relay fields of the dialing profiles of users, such as shown at **1150** in Figure **9**, identifying one or more media relays through which calls associated with the associated user are to be directed. In this case, the identifications of possible media relays obtained from the media relay fields **1150** may be sent to the call controller in additional fields in the routing message. These media relay fields are shown at **1152** in Figures **16**, **16**A, **25**, **25**A, **32**, **32**A and **32**B. In essence, the media relay through which communications involving the communications involving the subscriber will be conducted is identified in response to the routing message.

Referring back to Figure **34**, in this case, the message **1100** may be sent in a polling fashion to all media relays identified by the media relay fields **1150**, until one responds. Alternatively, the message **1100** may be sent simultaneously to all of the media relays.

In response, in the case where the media relay is known or is involved in polling as described above, the media relay **17** to which the message **1100** is sent sends a media relay status message **1102** back to the call controller **14**, the message including a media relay IP address and UDP port number at which the media relay will establish a UDP connection to the callee telephone **15**. Audio data to/from the callee telephone **15** will be transmitted over this connection. In the case where the message **1100** is sent to a plurality of media relays, the first one to respond with a media relay status message is the one through which the call will be carried. Media relay status messages from the remaining media relays can be ignored.

After the media relay status message **1102** is received at the call controller, the call controller **14** then sends a SIP Invite message **1104** of the type shown in Figure **3** to the callee telephone **15**, including the contents of the caller and callee identifier fields (**60** and **62**), the call identifier field (**65**) and the media relay IP address and the media relay UDP port number assigned to the audio path connection with the callee telephone **15**, to invite the callee telephone to establish a connection with the media relay **17**.

The purpose of the SIP Invite message **1104**, is to advise the callee telephone of the caller and call ID and of the IP address and UDP port number of the media relay through which the callee telephone should send and receive audio data.

The callee telephone **15** stores the media relay IP address and assigned UDP port number in the audio path IP address buffer **47** shown in Figure **2** and configures itself to create a socket between the media relay IP/UDP address and the callee telephone IP address and a UDP port number that the callee telephone **15** desires to use as an audio path to the caller telephone. Instead of being sent or received directly to or from the caller telephone, the callee telephone **15** will send and receive audio data from the media relay. To indicate this, the callee telephone **15** sends a SIP OK message **1106** back to

the call controller **14**, the message including the callee IP address and UDP port number from its IP address field (**53** in Figure **3**) at which the callee telephone **15** will establish an audio path connection with the media relay **17**. The purpose of this SIP OK message **1106** is to advise the call controller of the IP address and UDP port number through which the media relay should send and receive audio data to and from the callee telephone.

The call controller **14** then sends a message **1108** to the media relay **17** including the IP address and UDP port number that the callee telephone **15** will use for the audio path connection with the media relay. The purpose of the message **1108** is to advise the media relay of the IP address and UDP port number through which it should send and receive audio data to and from the callee telephone.

The media relay **17** then determines a UDP port through which it will carry audio data to and from the caller telephone **12** and sends a message **1110** to the call controller (**14**), the message including the media relay IP address and the media relay UDP port number the media relay will use to carry audio to and from the caller telephone **12**. The purpose of this message **1110** is to advise the call controller **14** of the IP address and UDP port number through which it expects to transfer audio data to and from the caller telephone.

The call controller **14** then sends a SIP OK message **1112** to the caller telephone **12** to indicate that the call may now proceed. The SIP OK message includes the caller and callee usernames, the call ID and the media relay **17** IP address and the UDP port number assigned to the audio connection with the caller telephone **12**. The purpose of this SIP OK message **1112** is to advise the caller telephone **12** of the IP address and UDP port number through which it should exchange audio data with the media relay **17**.

If the routing message is of the type shown in Figure **25** where there are a plurality of suppliers available, the call handling routine proceeds as described

above with the exception that instead of communicating with the callee telephone directly, the call controller **14** communicates with a gateway provided by a supplier. If a SIP OK message is not received back from the first gateway, the processor is directed to send the SIP Invite message **1104** to a gateway of the next indicated supplier. For example, the call controller **14** sends the SIP Invite message **1104** to the first supplier, in this case Telus, to determine whether or not Telus is able to handle the call. If Telus does not send back a SIP OK message **1106** within a specified time or sends a message indicating that it is not able to handle the call, the call controller proceeds to send a SIP Invite message **1104** to the next supplier, in this case Shaw. The process is repeated until one of the suppliers responds with a SIP OK message **1106** indicating that it is available to carry the call and the process proceeds as shown in connection with messages **1108**, **1110** and **1112**. For example, the supplier "Telus" sends back a SIP OK message and thus provides a gateway to the PSTN at IP address **72.64.39.58** as provided by the routing message from the contents of the route identifier field **546** of the corresponding supplier record shown in Figure **22**.

Referring back to Figure **1**, if the call controller **14** receives a message of the type shown in Figure **32**, i.e., a type that has one call forwarding number and/or a voicemail number, the call controller attempts to establish a call (using SIP Invite message **1104**) to the callee telephone **15** and if no call is established (i.e., message **1106** is not received) within a pre-determined time, the call controller **14** attempts to establish a call with the next user identified in the call routing message, by sending a SIP invite message like message **1104** to the next user. This process is repeated until all call forwarding possibilities have been exhausted, in which case an audio path is established with the voicemail server **19** identified in the routing message. The voicemail server **19** sends the SIP OK message **1106** in response to receipt of the SIP invite message **1104** and functions as described above in connection with the callee telephone **15** to permit an outgoing audio message provided by the voicemail

server to be heard by the caller and to permit the caller to record an audio message on the voicemail server.

When audio paths are established, a call timer (not shown) maintained by the call controller logs the start date and time of the call and logs the call ID and adds an active call record of the type shown in Figure **35** to an active call list, maintained by the call controller.

In this embodiment, the call controller active call record shown in Figure **35** includes a call ID field **1300**, a caller IP address field **1302**, a caller port field **1304**, a callee IP address field **1306**, a callee port field **1308**, a media relay ID field **1310**, a media relay caller port field **1312** and a media relay callee port field **1314**. The contents of the call ID field **1300** are established at block **136** in Figure **5**. The contents of the caller IP address field **1302** are established from the contents of the caller IP address field **67** of the SIP invite message shown in Figure **3**. The contents of the caller port field **1304** are established from the caller UDP port field **69** of the SIP invite message shown in Figure **3**. The contents of the callee IP address field **1306** and callee port field **1308** are established from the SIP OK message **1106** shown in Figure **34**.

The media relay ID field **1310** is populated with an identification of the media relay handling the call. In the example shown, the media relay is number **42**. The contents of the media relay caller port field are obtained from the message **1110** shown in Figure **34** and the contents in the media relay callee port field **1314** are obtained from the media relay status message **1102** shown in Figure **34**. Each time a call is established, an active call record of the type shown in Figure **35** is added to an active call log maintained by the call controller.

The routing controller also maintains an active call log containing active call records however the active call records maintained by the routing controller are different from the active call records held by the call controller. For

example, referring to Figure **36**, an active call record held by the routing controller includes a call ID field **1316**, a caller field **1318**, a callee field **1320** and a call controller ID field **1322**. Information for populating these fields may be received in a message (not shown) transmitted from the call controller to the routing controller after an active call record has been entered into the active call log of the call controller.

The message from the call controller **14** to the routing controller **16**, indicating that an active call has been established may include the contents of the call ID field **1300** shown in Figure **35** and a call controller unique ID number held by the call controller. The routing controller **16** matches the call ID with the caller and callee user names contained in the original call routing message (Fig **16**, **16**A, **25**, **25**A, **32**, **32**A, **32**B) that caused the call controller **14** to route the call, to populate the caller and callee fields **1318** and **1320** shown in Figure **36**, respectively. It will be appreciated that a plurality of call controllers may be associated with a single routing controller, in which case the call controller ID allows the routing controller to uniquely identify the call controller associated with the call ID indicated by the contents of the call ID field **1316**. In the example shown, the call controller is number **61**.

The active call records facilitate intercepting a call already in progress, as will be described below.

Referring back to Figure **33**, if at block **1200** it is determined that the routing message has lawful intercept fields, block **1202** directs the call controller circuit **100** (Figure **4**) to send a SIP Invite message as shown in Figure **37** to a mediation device identified by the mediation device IP address in the routing message as obtained from the user dialing profile MD**1** address field **704** as shown at **256** in Figure **9**. Referring to Figure **37**, the SIP Invite message includes caller and callee identifier fields **1020**, **1022**, a call ID field **1024**, a warrant ID field **1026** and other intercept related information fields **1028**, if desired. The caller, callee and call ID field contents **1020**, **1022**, and **1024** are

obtained from the original SIP Invite message shown in Figure **6**. The contents of the warrant ID field **1026** and intercept related info fields **1028** are obtained from the routing message which would be of the type shown in Figures **16**A, **25**A, **32**A or **32**B.

Referring back to Figure **33**, block **1204** then directs the call controller **14** to receive a reply message, as shown in Figure **38**, from the mediation device **31**. The reply message is a SIP OK message that includes caller, callee, and call ID fields **1040**, **1042**, **1044** as described above and further includes a mediation device IP address field **1046** and a mediation device UDP caller port number field **1048** and a UDP callee port number field **1050** identifying UDP ports at the mediation device IP address to which the media relay is to send copies of audio data streams received from the caller and callee telephones respectively. Block **1206** then directs the call controller to execute the call handling routine shown in Figure **34** with the exception that the message **1100** additionally includes the contents of the mediation device IP address field **1046**, the mediation device UDP caller port number field **1048** and the UDP callee port number field **1050** of the SIP OK message shown in Figure **38**.

All other messages are the same as described above in connection with the call handling routine as shown in Figure **34**, but in response to receiving the additional information in the message **1100**, the media relay automatically configures itself to provide for copying the audio data received from both the caller telephone and the callee telephone to the mediation device IP address and the UDP caller port number and the UDP callee port number respectively.

Referring back to Figure **1**, as audio data originating at the caller telephone **12** and callee telephone **15** passes through the media relay **17**, this data is copied to the mediation device UDP port for the caller and the mediation device UDP port for the callee, as indicated by the SIP invite message **1100**. This enables law enforcement agencies to monitor audio communications

between the caller and callee and/or to record such communications at the mediation device.

Thus, when the determination information in the dialing profile meets intercept criteria, the call controller communicates with the media relay through which communications involving the subscriber whose communications are to be monitored will be handled to cause the media relay to send a copy of such communications to a mediation device specified by the destination information included in the intercept information associated with the dialing profile associated with the subscriber whose communications are to be monitored.

Terminating the Call

In the event that either the caller or the callee terminates a call, the telephone of the terminating party sends a SIP Bye message to the call controller **14**. An exemplary SIP Bye message is shown at **900** in Figure **39** and includes a caller field **902**, a callee field **904** and a call ID field **906**. The caller field **902** holds the caller username, the callee field **904** holds a PSTN compatible number or username, and the call ID field **906** holds a unique call identifier field of the type shown in the call identifier field **65** of the SIP Invite message shown in Figure **3**.

Thus, for example, referring to Figure **40**, a SIP Bye message for the Calgary callee is shown generally at **908** and the caller field **902** holds a username identifying the Vancouver caller, in this case **2001 1050 8667**, the callee field **904** holds a username identifying the Calgary callee, in this case **2001 1050 2222**, and the call ID field **906** holds the code **FA10@192.168.0.20**, which is the call ID for the call.

The SIP Bye message shown in Figure **40** is received at the call controller **14** and the call controller executes a process as shown generally at **910** in Figure **41**. The process includes a first block **912** that directs the call controller circuit (**100**) to copy the caller, callee and call ID field contents from the SIP Bye

message **900** shown in Figure **39** received from the terminating party to corresponding fields of an RC stop message buffer (not shown). Block **914** then directs the call controller circuit **100** to copy the call start time from the call timer and to obtain a Call Stop time from the call timer. Block **916** then directs the call controller to calculate a communication session time by determining the difference in time between the call start time and the Call Stop time. This communication session time is then stored in a corresponding field of the RC Call Stop message buffer. Block **918** then directs the call controller circuit **100** to populate the route field with the IP address of the gateway supplier, if any. An RC Call Stop message produced as described above is shown generally at **1000** in Figure **42**. An RC Call Stop message specifically associated with the call made to the Calgary callee is shown generally at **1021** in Figure **43**.

Referring to Figure **42**, the RC call stop message **1000** includes a caller field **1002**, callee field **1004**, a call ID field **1006**, an account start time field **1008**, an account stop time field **1010**, a communication session time field **1012** and a route field **1014**. The caller field **1002** holds a username, the callee field **1004** holds a PSTN-compatible number or system number, the call ID field **1006** holds the unique call identifier received from the SIP Invite message shown in Figure **3**, the account start time field **1008** holds the date and start time of the call, the account stop time field **1010** holds the date and time the call ended, the communication session time field **1012** holds a value representing the difference between the start time and the stop time, in seconds, and the route field **1014** holds the IP address for a gateway, if a gateway is used to establish the call.

Referring to Figure **43**, an exemplary RC call stop message for the Calgary callee is shown generally at **1021**. In this example the caller field **1002** holds the username **2001 1050 8667** identifying the Vancouver caller and the callee field **1004** holds the username **2001 1050 2222** identifying the Calgary callee. The contents of the call ID field **1006** are FA**10@192.168.0.20**. The contents

of the account start time field **1008** are **2006-12-30 12:12:12** and the contents of the account stop time field **1010** are **2006-12-30 12:12:14**. The contents of the communication session time field **1012** are **2** to indicate **2** seconds call duration and the contents of the route field are blank but would be **72.64.39.58** if the "Telus" gateway were used, for example.

Referring back to Figure **41**, after having produced an RC Call Stop message, block **920** directs the call controller circuit **100** to send the RC stop message contained in the RC Call Stop message buffer to the routing controller (**16**).

The RC (**16**) receives the Call Stop message and an routing controller Call Stop message process (not shown) is invoked at the routing controller to deal with charges and billing for the call.

Block **922** directs the call controller circuit **100** to send a Bye message to the party that did not terminate the call i.e. to the non-terminating party.

Block **924** then directs the call controller circuit **100** to send a SIP Bye message of the type shown in Figure **39** to the media relay **17** to cause the media relay to disconnect the audio path sockets associated with the caller telephone IP/UDP address and the callee telephone IP/UDP address. In disconnecting these communication sockets, the media relay **17** deletes associations between the caller telephone IP/UDP address media relay caller IP/UDP address and between the caller telephone IP/UDP address and media relay callee IP/UDP address.

If the media relay (**17**) was configured for lawful intercept, block **926** of Figure **41** then directs the call controller circuit **100** to send a SIP Bye message of the type shown in Figure **39** to the mediation device **31** to inform the mediation device that the call has ended and to disconnect communication sockets between the media relay caller and callee IP/UDP port addresses and

the IP/UDP port address to which the audio data received at the caller and callee IP/UDP port addresses were being copied.

It will be appreciated that in the foregoing description, the components described cooperate to detect a requirement for intercept at the time a call is set up. In the following description an explanation is provided to describe how to intercept a call while the call is in progress.

Intercepting a Call in Progress

Referring back to Figure **1**, to intercept a call while the call is in progress, the law enforcement authority **293** may communicate with a mediation device, or may communicate with the call controller or may communicate with the routing controller or may communicate with a handover interface that communicates with any of the foregoing components to cause the routing controller to receive a law enforcement authority (LEA) intercept request message including intercept information. Such as that which would be associated with fields **702-710** in Figure **9**, for example..

In response to receipt of a, LEA intercept request message, the routing controller LEA request message handler shown at **1400** in Figure **44** is invoked.

The LEA request message handler **1400** begins with a first block **1402** that directs the routing controller processor circuit to communicate with the database **18** in which dialing profile records of the type shown in Figure **9** are stored to find a dialing profile associated with the user whose calls are to be monitored.

If the username is not known, but a DID number (i.e. a PSTN number) is known, the routing controller may cause a search through the DID bank table records of the type shown in Figure **13**, for example to find a username associated with a DID number. If the username is not known but a name and

address is known, other records such as billing records (not shown) associating names and addresses with usernames may be searched to find a username associated with a given name and/or address of a person whose calls are to be intercepted. Regardless of the information available, to facilitate call interception any way of finding the unique dialing profile associated with the user whose calls are to be intercepted is a first step to facilitating call interception, in this embodiment.

Once the dialing profile is located, block **1404** directs the routing controller processor circuit to associate the intercept information with the dialing profile by appending and/or populating the lawful intercept fields of the dialing profile with such information as provided in the LEA intercept request message..

Block **1406** then directs the routing controller processor circuit to determine whether the intercept criteria are met by the intercept information now included in the dialing profile. This is done by determining whether the LI flag (**702**) is on, and the current date and time is within the LI start stop date/time ranges. If the intercept criteria are not met, the process is ended. Otherwise the processor is directed to block **1408**.

Block **1408** directs the routing controller processor circuit to use the username of the dialing profile found at block **1402** to search caller and callee fields of routing controller active call records shown in Figure **36** that have contents matching the username associated with the dialing profile. If no such record is found, the user is not currently engaged in a call and the process is ended. If the user is engaged in a call, the routing controller active call record will be found. Block **1410** then directs the routing controller processor circuit to find the call controller id and call id of the associated call, from the routing controller active call record shown in Figure **36**.

Block **1412** then directs the routing controller processor circuit to transmit an in-call intercept message to the call controller identified by the contents of the

call controller id field **1322** of the routing controller active call record. The in-call intercept message includes the call id as determined from the routing controller active call record and the IP address of the mediation device associated with the law enforcement authority interested in intercepting the call. The IP address of the mediation device may be obtained from the law enforcement authority request message, or the dialing profile, for example.

Block **1414** then directs the routing controller processor circuit to wait a specified time to receive a call controller intercept status message back from the call controller indicating whether or not the intercept function has been activated.

Referring to Figure **45**, upon receipt of an in-call intercept message at the call controller (**14**) the call controller executes an in-call intercept message handler shown generally at **1450**. The in-call intercept message handler **1450** begins with a first block **1452** that directs the call controller processor circuit to send a SIP invite message to the mediation device associated with the IP address of the mediation device, received in the in-call intercept message.

Block **1454** then directs the call controller processor circuit to receive an IP address and callee and caller UDP port numbers from the mediation device, where this IP address and UDP port numbers are network locations at which the mediation device will expect to receive audio data streams from the media relay through which the call is carried.

Block **1456** then directs the call controller processor circuit to identify a media relay through which communications to be monitored are being conducted by using the username of the subscriber whose communications are to be monitored to locate an active call record in the call controller active call list to locate a media relay identifier such as the IP address of the media relay indicated by the contents of the media relay ID field **1310** of the call controller active call record shown in Figure **35.** The call controller processor circuit is

then directed to send an intercept request message to the media relay (**17**) that is handling the call. The intercept request message includes the mediation device IP address and caller and callee UDP port numbers to identify to the media relay (**17**) the mediation device IP address and UDP port number(s) at which it expects to receive a copy of the audio data stream from the caller and callee respectively.

In response, the media relay establishes internal connections between the caller and callee IP addresses and UDP ports and callee IP address and UDP port of the mediation device. Then, the media relay sends a media relay status message back to the call controller indicating whether or not internal connections have been established and that call intercept has been initiated.

As seen at block **1458**, the call controller processor circuit is directed to receive the media relay status message and block **1460** directs the call controller processor circuit to send a call controller intercept status message back to the routing controller to indicate that the call intercept function has been established. The routing controller may communicate this status back to the law enforcement authority that issued the law enforcement authority request message. In the meantime, communications involving the caller or callee whose communications are to be monitored, which travel through the media relay, are copied and sent to the mediation device.

Thus, after associating intercept information with the dialing profile of the subscriber whose communications are to be monitored, when the determination information included in the intercept information meets intercept criteria, the call controller communicates with the media relay through which the communications of the subscriber whose communications are to be monitored to cause such media relay to send a copy of such communications to a mediation device specified by the destination information included in the intercept information.

When the call is ended, the call is shut down in the same way as described above.

Should the law enforcement authority desire to cease interception of the call during the call, an LEA request message requesting that the intercept function be stopped is sent to the routing controller from the law enforcement authority through any of the paths described above. This invokes the LEA request message handler such as shown in Figure **44** which causes the routing controller processor circuit to execute blocks **1402**, **1404**. At block **1404**, the routing controller processor circuit is directed to change the contents of the lawful intercept fields to at least set the lawful intercept flag (**702** in Figure **9**) inactive.

Then, at block **1406**, the intercept criteria are not met and the processor is directed to block **1416**, which causes the routing controller processor circuit to determine whether or not an interception function is in progress. This can be determined, for example, by maintaining evidence of the receipt of the confirmation message from the call controller, received at block **1414** of the LEA request message handler **1400**.

If an intercept is not in progress, the LEA request message handler **1400** is ended.

If an intercept if in progress, block **1418** directs the routing controller processor circuit to execute an in-call intercept shut down routine as shown at **1500** in Figure **46**. The in-call intercept shut down routine begins with a first block **1502** which directs the routing controller processor circuit to locate the routing controller active call record having caller or callee field contents equal to the username indicated in the dialing profile found at bock **1402** of the LEA request message handler **1400** shown in Figure **44**. Having found the active call record, block **1504** directs the routing controller processor circuit to find, in the routing controller active call record shown in Figure **36**, the call

controller id (**1322**) and the call id (**1316**) associated with the call. Block **1506** then directs the routing controller processor circuit to send a cease intercept message (not shown) to the call controller identified by the call controller id determined at block **1504**. This cease intercept message includes the call id determined at block **1504** and an identification of the mediation device, the identification being obtained from the MD1 address field (**704** in Figure **9**) of the dialing profile for the user whose calls are currently being intercepted. Block **1508** then directs the routing controller processor circuit to wait a specified time to receive a confirmation message from the call controller to indicate that the intercept function has been shut down.

Referring to Figure **47**, upon receipt of the cease intercept message at the call controller (**14**), a cease intercept message handler **1520** is invoked at the call controller. The cease intercept message handler **1520** begins with a first block **1522** that directs the call controller processor circuit to send a SIP stop message to the mediation device identified in the cease intercept message received from the routing controller. In response to the SIP stop message, the mediation device stops receiving audio data and sends a confirmation message back to the call controller.

Block **1524** directs the call controller processor circuit to receive the confirmation message back from the mediation device.

Block **1526** then directs the call controller processor circuit to send a stop intercept message to the media relay **17** identified by the contents of the media relay ID field **1310** of the active call record shown in Figure **35**. The stop intercept message includes the contents of the media relay caller port ID field **1312** and media relay callee port field **1314** included in the active call record and identifies to the media relay which ports to shut down. In response to the stop intercept message, the media relay **17** disconnects the connections between the media relay caller port and the mediation device port that was receiving the audio data from the caller and the connection between

the media relay callee port and the mediation device port that was receiving audio data from the callee. The media relay then sends an MR stop status message to the call controller.

Block **1528** directs the call controller processor circuit to receive the MR stop status message and block **1530** directs the call controller to send a stop status message to the routing controller **16**.

In an alternative embodiment, the routing controller does not maintain active call records but each call controller does. In such an embodiment, blocks **1408** and **1410** of Figure **44** are replaced with a single block **1600** that directs the routing controller processor circuit to poll each call controller to determine whether or not its active call list contains an entry having caller or callee field contents equal to the username determined from the dialing profile located at block **1402**.

If any of the polled call controllers has such a record, that call controller transmits a response message back to the routing controller, the response message including a call controller ID identifying that call controller. More than one call controller may have an active call record having caller or callee field contents equal to the username determined from the user profile. Such would be the case in a conference call, for example.

The routing controller processor circuit then executes blocks **1412** and **1414** as described above or the process is ended if none of the polled call controllers contains a call record with caller and callee field contents matching the username determined from the dialing profile located at block **1402**.

In effect therefore, block **1600** provides an alternate way of finding call controllers that are currently carrying a call associated with the user of interest.

In another embodiment, an interface to the routing controller and/or the call controller may be provided to enable law enforcement authorities to have direct access or a copy of the active call list maintained by the call controller and/or routing controller.

From the foregoing, it will be appreciated that indications of whether or not communications of a subscriber to the system are to be monitored are provided by law enforcement agencies directly into a subscriber dialing profile shown in Figure **9**. This dialing profile is used to route a call involving the subscriber and is checked for lawful intercept requirements to determine whether or not the media relay should copy audio data associated with the call to a mediation device for lawful monitoring and/or recording purposes.

While the system has been described in connection with the monitoring of audio streams, it may similarly be used for monitoring any other data streams such as pure data and/or video or multimedia data, for example, between subscribers to the system or between a subscriber and a non-subscriber to the system.

While specific embodiments of the invention have been described and illustrated, such embodiments should be considered illustrative of the invention only and not as limiting the invention as construed in accordance with the accompanying claims.

What is claimed is:

1. A method for intercepting communications in an Internet Protocol (IP) network, the method comprising:

maintaining dialing profiles for respective subscribers to the IP network, each said dialing profile including a username associated with the corresponding subscriber;

associating intercept information with the dialing profile of a subscriber whose communications are to be monitored, said intercept information including determination information for determining whether to intercept a communication involving said subscriber, and destination information identifying a device to which intercepted communications involving said subscriber are to be sent; and

when said determination information meets intercept criteria, communicating with a media relay through which said communications involving said subscriber will be conducted or are being conducted to cause said media relay to send a copy of said communications to a mediation device specified by said destination information.

2. The method of clam 1 wherein associating intercept information comprises associating said intercept information with said dialing profile when communications involving said subscriber are not in progress.

3. The method of clam 1 wherein associating intercept information comprises associating said intercept information when communications involving said subscriber are in progress.

4. The method of claim **2** or **3** wherein associating said intercept information comprises populating intercept information fields in said dialing profile of the subscriber whose communications are to be monitored.

5. The method of claim **1** further comprising producing a routing message for routing communications involving the subscriber through components of the IP network and determining whether said determination information meets said intercept criteria prior to producing said routing message and including at least some of said intercept information in said routing message when said determination information meets said intercept criteria.

6. The method of claim **5** wherein determining whether said determination information meets said intercept criteria comprises determining whether a current date and time is within a range specified by said determination information.

7. The method of claim **6** further comprising identifying a media relay through which communications involving said subscriber will be conducted in response to said routing message.

8. The method of claim **7** further comprising pre-associating at least one media relay with said dialing profile of the subscriber whose communications are to be monitored and wherein identifying said media relay comprises identifying the media relay pre-associated with said subscriber whose communications are to be monitored.

9. The method of claim **8** wherein pre-associating comprises populating media relay fields in said dialing profile with an identification of at least one media relay.

10. The method of claim **1** wherein said intercept information is associated with said dialing profile of the subscriber whose communications are to be monitored, in response to receipt of an intercept request message, wherein said intercept request message comprises said intercept information.

11. The method of claim **10** further comprising invoking an intercept request message handler to:

a) find a dialing profile associated with the subscriber whose communications are to be monitored;

b) perform the step of associating said intercept information with said dialing profile;

c) determine whether said intercept criteria are met; and

d) identify a media relay through which said communications are being conducted.

12. The method of claim **11** further comprising maintaining a active call records for communications in progress, said active call records comprising a username identifier and a media relay identifier identifying the media relay through which said communications are being conducted and wherein identifying a media relay through which said communications are being conducted comprises locating an active call record associated with communications of the subscriber whose communication are to be monitored to find the media relay associated with said communications.

13. The method of claim **12** further comprising maintaining direct inward dialing (DID) records associating PST telephone numbers with usernames of users subscribing to said IP network, and wherein finding a dialing profile associated with the subscriber whose communications are to be monitored comprises finding a username in a DID record

bearing a PSTN number associated with the subscriber whose communications are to be monitored and using said username to locate a dialing profile associated with said username.

14. An apparatus for intercepting communications in an Internet Protocol (IP) network, the apparatus comprising:

means for maintaining dialing profiles for respective subscribers to the IP network, each said dialing profile including a username associated with the corresponding subscriber;

means for associating intercept information with the dialing profile of a subscriber whose communications are to be monitored, said intercept information including determination information for determining whether to intercept a communication involving said subscriber, and destination information identifying a device to which intercepted communications involving said subscriber are to be sent; and

means for communicating with a media relay through which said communications involving said subscriber will be conducted or are being conducted to cause said media relay to send a copy of said communications to a mediation device specified by said destination information, when said determination information meets intercept criteria.

15. The apparatus of clam 14 wherein said means for associating intercept information is operably configured to associate said intercept information with said dialing profile when communications involving said subscriber are not in progress.

16. The apparatus of clam **14** wherein said means for associating intercept information is operably configured to associate said intercept information when communications involving said subscriber are in progress.

17. The apparatus of claim **15** or **16** wherein said means for associating said intercept information is operably configured to populate intercept information fields in said dialing profile of the subscriber whose communications are to be monitored.

18. The apparatus of claim **14** further comprising means for producing a routing message for routing communications involving the subscriber through components of the IP network and means for determining whether said determination information meets said intercept criteria prior to producing said routing message and wherein said means for producing said routing message is operably configured to include at least some of said intercept information in said routing message when said determination information meets said intercept criteria.

19. The apparatus of claim **18** wherein said means for determining whether said determination information meets said intercept criteria is operably configured to determine whether a current date and time is within a range specified by said determination information.

20. The apparatus of claim **19** further comprising means for identifying a media relay through which communications involving said subscriber will be conducted in response to said routing message.

21. The apparatus of claim **20** further comprising means for pre-associating at least one media relay with said dialing profile of the subscriber whose communications are to be monitored and wherein said routing means is operably configured to identify from said dilaling

profile the media relay pre-associated with said subscriber whose communications are to be monitored.

22. The apparatus of claim **21** wherein said means for pre-associating is operably configured to populate media relay fields in said dialing profile with an identification of at least one media relay.

23. The apparatus of claim **14** wherein means for associating said intercept information is operably configured to associate said intercept information associated with said dialing profile of the subscriber whose communications are to be monitored, in response to receipt of an intercept request message, wherein said intercept request message comprises said intercept information.

24. The apparatus of claim **23** further comprising means for handling an intercept request message, said means for handling an intercept request message comprising:

a) means for find a dialing profile associated with the subscriber whose communications are to be monitored, said means for finding a dialing profile cooperating with said means for associating said intercept information with said dialing profile to cause said intercept information to be associated with said dialing profile;

b) means for determining whether said intercept criteria are met; and

c) means for identifying a media relay through which said communications are being conducted.

25. The apparatus of claim **24** further comprising means for maintaining active call records for communications in progress, said active call records comprising a username identifier and a media relay identifier identifying the media relay through which said communications are

being conducted and wherein said means for identifying a media relay through which said communications are being conducted is operably configured to locate an active call record associated with communications of the subscriber whose communication are to be monitored to find the media relay associated with said communications.

26. The apparatus of claim **25** further comprising means for maintaining direct inward dialing (DID) records associating PST telephone numbers with usernames of users subscribing to said IP network, and wherein said means for finding a dialing profile associated with the subscriber whose communications are to be monitored is operably configured to find a username in a DID record bearing a PSTN number associated with the subscriber whose communications are to be monitored and use said username to locate a dialing profile associated with said username.

AMENDED CLAIMS received by the International Bureau on 14 May 2008 (14/05/08)

What is claimed is:

1. A method for intercepting communications in an internet Protocol (IP) network, the method comprising:

maintaining dialing profiles for respective subscribers to the IP network, each said dialing profile including a username associated with the corresponding subscriber;

associating intercept information with a dialing profile of a subscriber whose communications are to be monitored, said intercept information including determination information for determining whether to intercept a communication involving said subscriber, and destination information identifying a mediation device to which intercepted communications involving said subscriber are to be sent; and

selecting a media relay through which communications involving said subscriber and a callee or caller of said subscriber will be conducted, by selecting a media relay from a pool of media relays at any of a plurality of geographical locations to identify a selected media relay;

when said determination information meets intercept criteria, communicating with said selected media relay through which said communications involving said subscriber will be conducted or are being conducted to cause said selected media relay to send a copy of said communications to a mediation device specified by said destination information.

2. The method of claim 1 wherein associating intercept information comprises associating said intercept information with said dialing

profile when communications involving said subscriber are not in progress.

3. The method of claim 1 wherein associating intercept information comprises associating said intercept information when communications involving said subscriber are in progress.

4. The method of claim 2 or 3 wherein associating said intercept information comprises populating intercept information fields in said dialing profile of the subscriber whose communications are to be monitored.

5. The method of claim 1 further comprising producing a routing message for routing communications involving the subscriber through components of the IP network and determining whether said determination information meets said intercept criteria prior to producing said routing message and including at least some of said intercept information in said routing message when said determination information meets said intercept criteria.

6. The method of claim 5 wherein determining whether said determination information meets said intercept criteria comprises determining whether a current date and time is within a range specified by said determination information.

7. The method of claim 6 wherein producing a routing message comprises identifying a media relay through which communications involving said subscriber will be conducted and including an identification of said media relay in said routing message.

8. The method of claim 7 further comprising pre-associating at least one media relay with said dialing profile of the subscriber whose

communications are to be monitored and wherein identifying said media relay comprises identifying the media relay pre-associated with said subscriber whose communications are to be monitored.

9. The method of claim 8 wherein pre-associating comprises populating media relay fields in said dialing profile with an identification of at least one media relay.

10. The method of claim 1 wherein said intercept information is associated with said dialing profile of the subscriber whose communications are to be monitored, in response to receipt of an intercept request message, wherein said intercept request message comprises said intercept information.

11. The method of claim 10 further comprising invoking an intercept request message handler to:

a) find a dialing profile associated with the subscriber whose communications are to be monitored;

b) perform the step of associating said intercept information with said dialing profile;

c) determine whether said intercept criteria are met; and

d) identify a media relay through which said communications are being conducted.

12. The method of claim 11 further comprising maintaining active call records for communications in progress, said active call records comprising a username identifier and a media relay identifier identifying the media relay through which said communications are being conducted and wherein identifying a media relay through which said communications are being conducted comprises locating an active call record associated with communications of the subscriber whose

communication are to be monitored to find the media relay associated with said communications.

13. The method of claim 12 further comprising maintaining direct-in-dial (DID) records associating PST telephone numbers with usernames of users subscribing to said IP network, and wherein finding a dialing profile associated with the subscriber whose communications are to be monitored comprises finding a username in a DID record bearing a PSTN number associated with the subscriber whose communications are to be monitored and using said username to locate a dialing profile associated with said username.

14. An apparatus for intercepting communications in an Internet Protocol (IP) network, the apparatus comprising:

means for maintaining dialing profiles for respective subscribers to the IP network, each said dialing profile including a username associated with the corresponding subscriber;

means for associating intercept information with a dialing profile of a subscriber whose communications are to be monitored, said intercept information including determination information for determining whether to intercept a communication involving said subscriber, and destination information identifying a mediation device to which intercepted communications involving said subscriber are to be sent; and

means for selecting a media relay through which communications involving said subscriber and a callee or caller of said subscriber will be conducted, by selecting a media relay

T-Mobile Exhibit 1002
Part 3 of 6

from a pool of media relays at any of a plurality of geographical locations to identify a selected media relay;

means for communicating with said selected media relay through which said communications involving said subscriber will be conducted or are being conducted to cause said selected media relay to send a copy of said communications to a mediation device specified by said destination information, when said determination information meets intercept criteria.

15. The apparatus of claim 14 wherein said means for associating intercept information is operably configured to associate said intercept information with said dialing profile when communications involving said subscriber are not in progress.

16. The apparatus of claim 14 wherein said means for associating intercept information is operably configured to associate said intercept information when communications involving said subscriber are in progress.

17. The apparatus of claim 15 or 16 wherein said means for associating said intercept information is operably configured to populate intercept information fields in said dialing profile of the subscriber whose communications are to be monitored.

18. The apparatus of claim 14 further comprising means for producing a routing message for routing communications involving the subscriber through components of the IP network and means for determining whether said determination information meets said intercept criteria prior to producing said routing message and wherein said means for producing said routing message is operably configured to include at

least some of said intercept information in said routing message when said determination information meets said intercept criteria.

19. The apparatus of claim 18 wherein said means for determining whether said determination information meets said intercept criteria is operably configured to determine whether a current date and time is within a range specified by said determination information.

20. The apparatus of claim 19 wherein said means for producing said routing message is operably configured to identify a media relay through which communications involving said subscriber will be conducted and to include ng an identification of said media relay in said routing message.

21. The apparatus of claim 20 further comprising means for pre-associating at least one media relay with said dialing profile of the subscriber whose communications are to be monitored and wherein said routing means is operably configured to identify from said dilaling profile the media relay pre-associated with said subscriber whose communications are to be monitored.

22. The apparatus of claim 21 wherein said means for pre-associating is operably configured to populate media relay fields in said dialing profile with an identification of at least one media relay.

23. The apparatus of claim 14 wherein means for associating said intercept information is operably configured to associate said intercept information associated with said dialing profile of the subscriber whose communications are to be monitored, in response to receipt of an intercept request message, wherein said intercept request message comprises said intercept information.

24. The apparatus of claim 23 further comprising means for handling an intercept request message, said means for handling an intercept request message comprising:

a) means for finding a dialing profile associated with the subscriber whose communications are to be monitored, said means for finding a dialing profile cooperating with said means for associating said intercept information with said dialing profile to cause said intercept information to be associated with said dialing profile;

b) means for determining whether said intercept criteria are met; and

c) means for identifying a media relay through which said communications are being conducted.

25. The apparatus of claim 24 further comprising means for maintaining active call records for communications in progress, said active call records comprising a username identifier and a media relay identifier identifying the media relay through which said communications are being conducted and wherein said means for identifying a media relay through which said communications are being conducted is operably configured to locate an active call record associated with communications of the subscriber whose communication are to be monitored to find the media relay associated with said communications.

26. The apparatus of claim 25 further comprising means for maintaining direct-in-dial (DID) records associating PST telephone numbers with usernames of users subscribing to said IP network, and wherein said means for finding a dialing profile associated with the subscriber whose communications are to be monitored is operably configured to find a username in a DID record bearing a PSTN number associated with the subscriber whose communications are to be monitored and use said username to locate a dialing profile associated with said username.

T-Mobile Exhibit 1002
Part 3 of 6

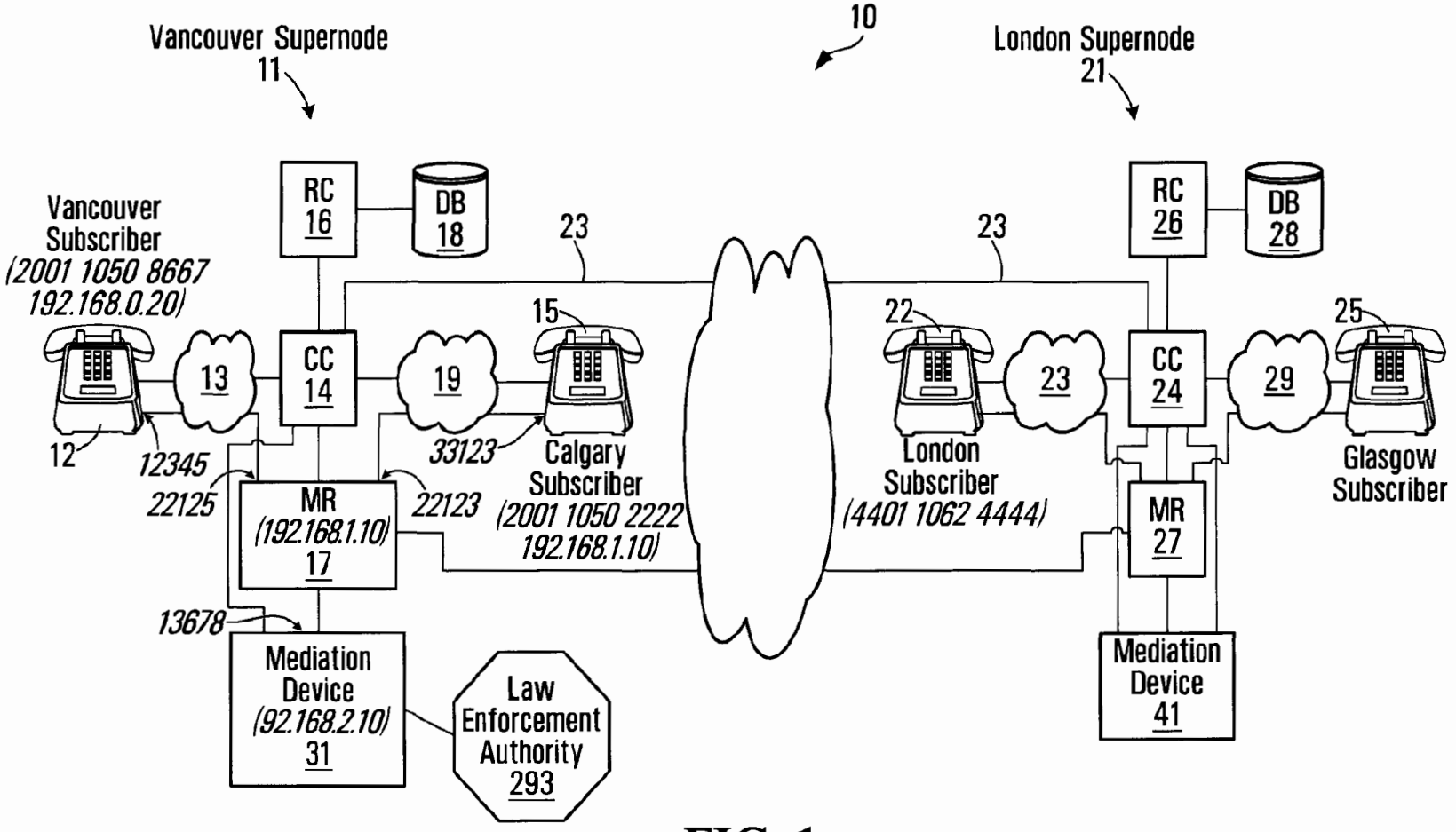

10
Vancouver Supernode
11
London Supernode
21
RC
16
DB
18
RC
26
DB
28
Vancouver
Subscriber
(2001 1050 8667
192.168.0.20)
23
23
15
22
25
13
CC
14
19
23
CC
24
29
12
12345
22125
33123
Calgary
Subscriber
(2001 1050 2222
192.168.1.10)
London
Subscriber
(4401 1062 4444)
Glasgow
Subscriber
MR
(192.168.1.10)
17
22123
MR
27
13678
Mediation
Device
(92.168.2.10)
31
Law
Enforcement
Authority
293
Mediation
Device
41

FIG. 1

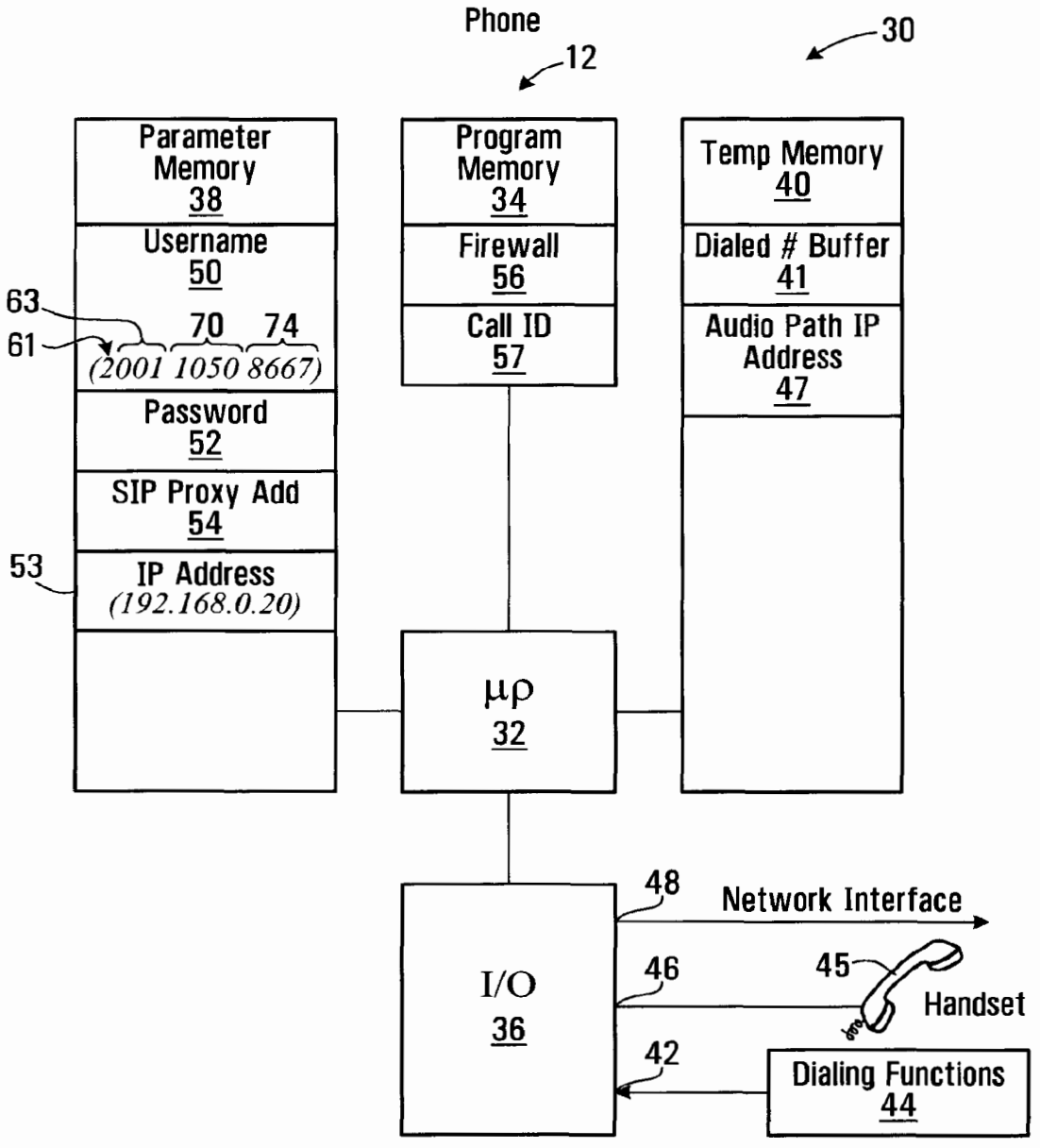

Phone
12
30
Parameter Memory 38
Username 50
63
61
70
74
(2001 1050 8667)
Password 52
SIP Proxy Add 54
53
IP Address (192.168.0.20)
Program Memory 34
Firewall 56
Call ID 57
Temp Memory 40
Dialed # Buffer 41
Audio Path IP Address 47
μρ 32
I/O 36
48
Network Interface
46
45
Handset
42
Dialing Functions 44

FIG. 2

SIP Invite Message

60	Caller	2001 1050 8667
62	Callee	2001 1050 2222
64	Digest Parameters	XXXXXXX
65	Call ID	FF10@ 192.168.0.20
67	Caller IP Address	192.168.0.20
69	Caller UDP port	12345

FIG. 3

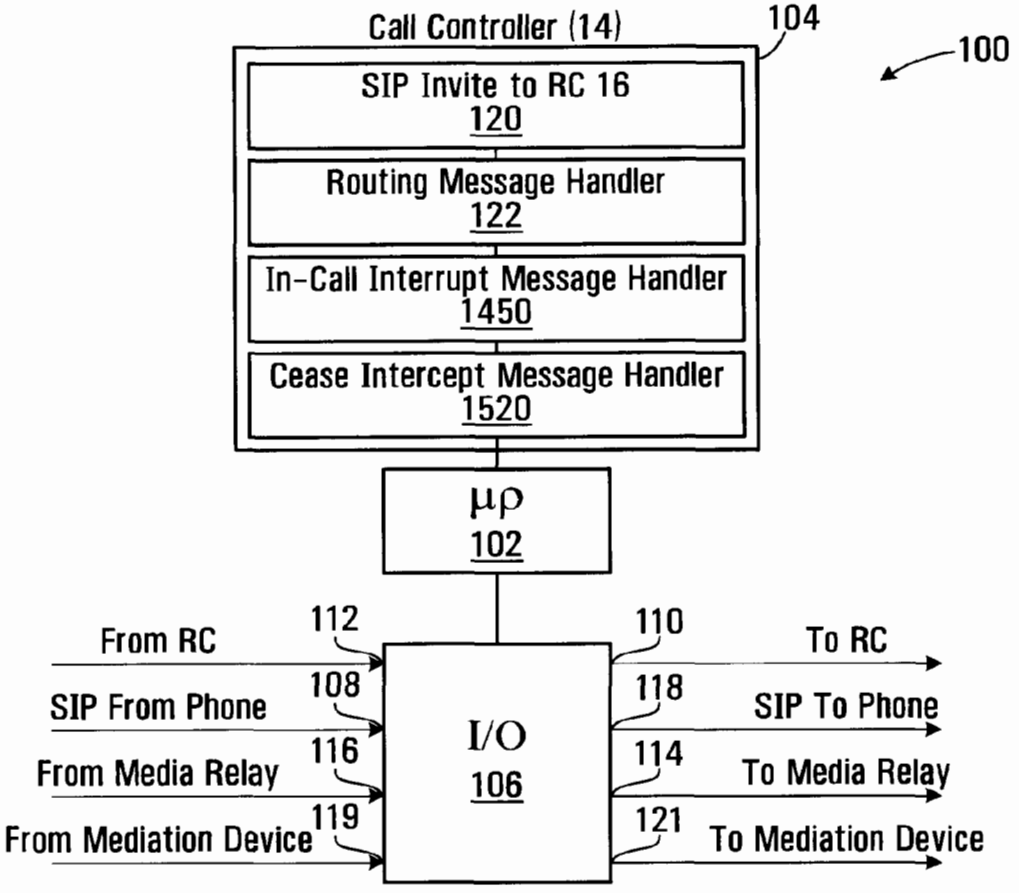


FIG. 4

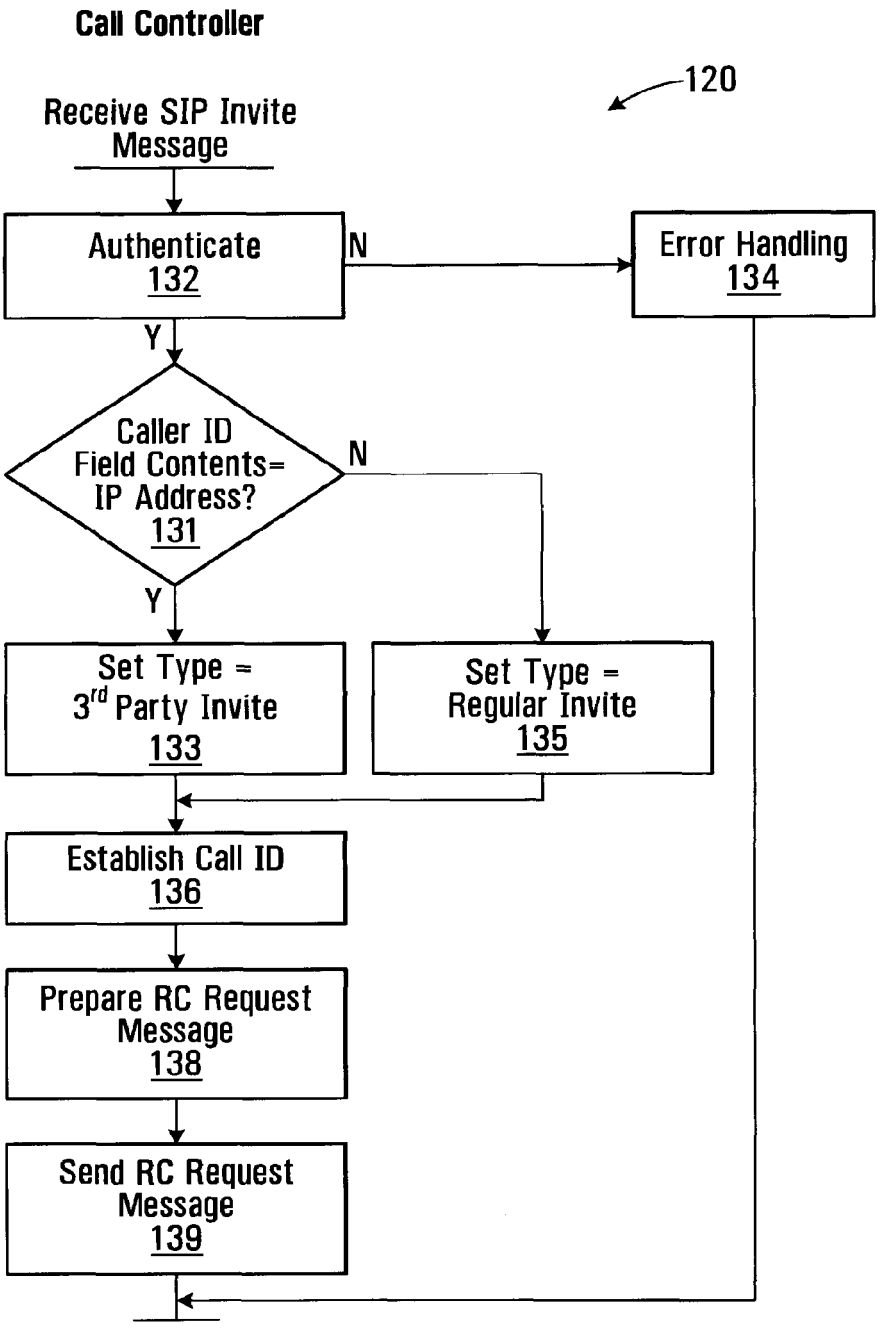

Call Controller
120
Receive SIP Invite Message
Authenticate 132
N
Error Handling 134
Y
Caller ID Field Contents= IP Address? 131
N
Y
Set Type = 3rd Party Invite 133
Set Type = Regular Invite 135
Establish Call ID 136
Prepare RC Request Message 138
Send RC Request Message 139

FIG. 5

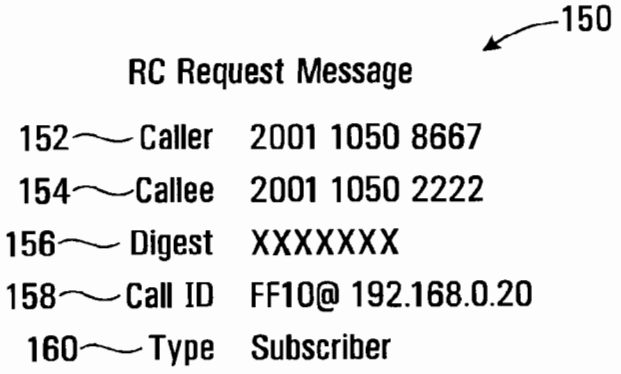

150
RC Request Message
152 Caller 2001 1050 8667
154 Callee 2001 1050 2222
156 Digest XXXXXXX
158 Call ID FF10@ 192.168.0.20
160 Type Subscriber

FIG. 6

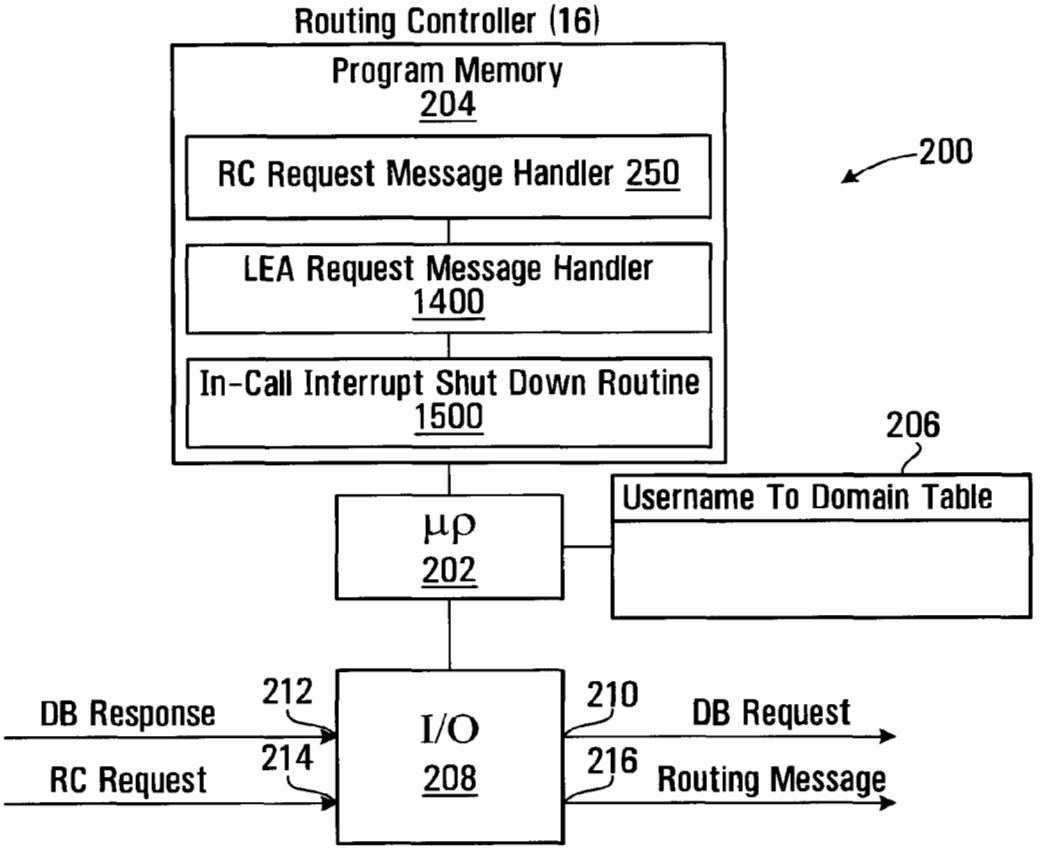

Routing Controller (16)
Program Memory
204
RC Request Message Handler 250
LEA Request Message Handler
1400
In-Call Interrupt Shut Down Routine
1500
200
206
Username To Domain Table
μρ
202
DB Response
212
RC Request
214
I/O
208
210
DB Request
216
Routing Message

FIG. 7

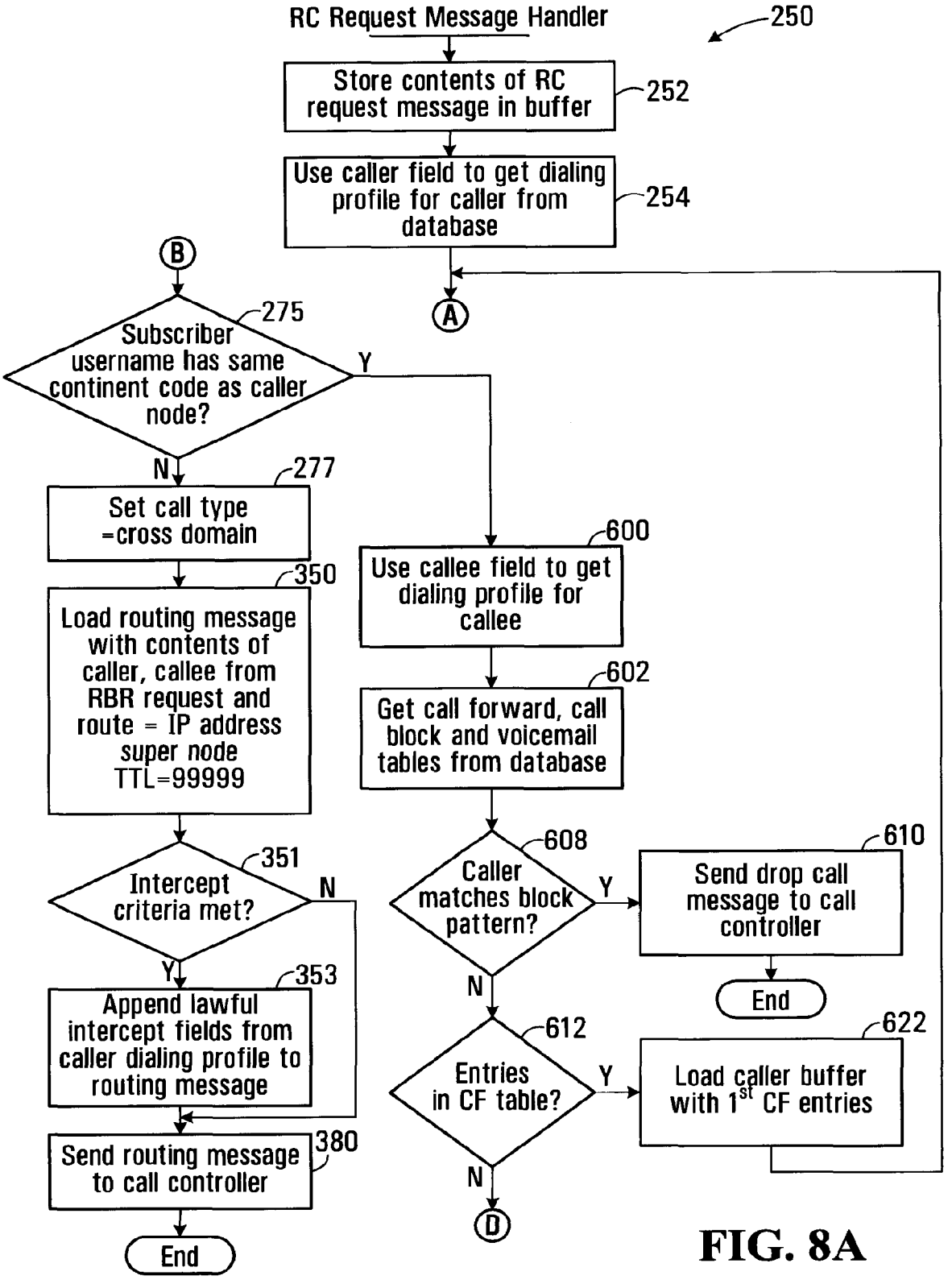

RC Request Message Handler
250
Store contents of RC request message in buffer
252
Use caller field to get dialing profile for caller from database
254
B
A
Subscriber username has same continent code as caller node?
275
Y
N
Set call type =cross domain
277
Load routing message with contents of caller, callee from RBR request and route = IP address super node TTL=99999
350
Intercept criteria met?
351
N
Y
Append lawful intercept fields from caller dialing profile to routing message
353
Send routing message to call controller
380
End
Use callee field to get dialing profile for callee
600
Get call forward, call block and voicemail tables from database
602
Caller matches block pattern?
608
Y
N
Send drop call message to call controller
610
End
Entries in CF table?
612
Y
N
Load caller buffer with 1st CF entries
622
D

FIG. 8A

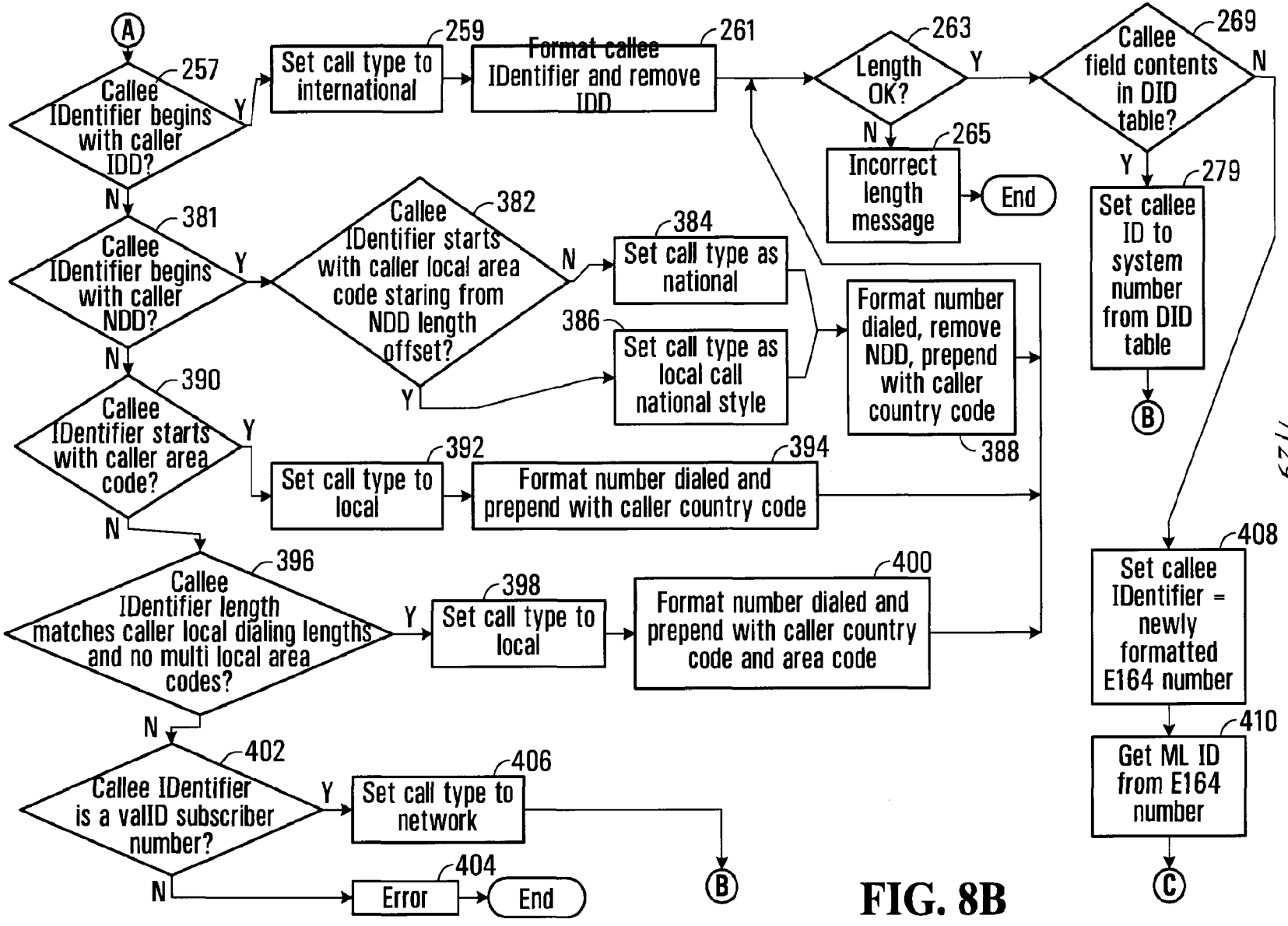

A
257
Callee IDentifier begins with caller IDD?
Y
N
259
Set call type to international
261
Format callee IDentifier and remove IDD
263
Length OK?
Y
N
265
Incorrect length message
End
269
Callee field contents in DID table?
N
Y
279
Set callee ID to system number from DID table
B
381
Callee IDentifier begins with caller NDD?
Y
N
382
Callee IDentifier starts with caller local area code staring from NDD length offset?
N
Y
384
Set call type as national
386
Set call type as local call national style
388
Format number dialed, remove NDD, prepend with caller country code
390
Callee IDentifier starts with caller area code?
Y
N
392
Set call type to local
394
Format number dialed and prepend with caller country code
396
Callee IDentifier length matches caller local dialing lengths and no multi local area codes?
Y
N
398
Set call type to local
400
Format number dialed and prepend with caller country code and area code
402
Callee IDentifier is a valID subscriber number?
Y
N
406
Set call type to network
404
Error
End
B
408
Set callee IDentifier = newly formatted E164 number
410
Get ML ID from E164 number
C

FIG. 8B

7/29

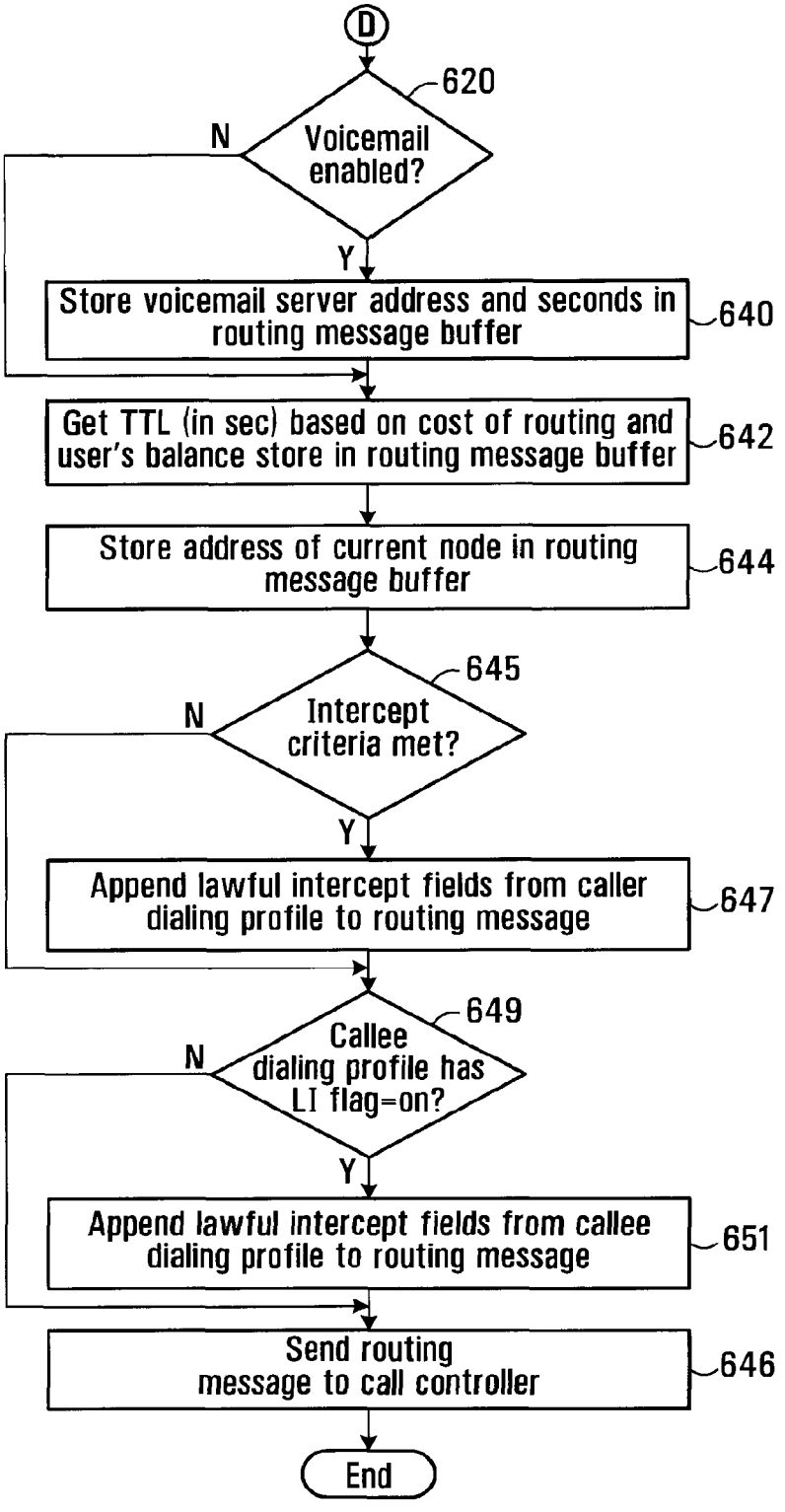

D
620
Voicemail enabled?
N
Y
Store voicemail server address and seconds in routing message buffer
640
Get TTL (in sec) based on cost of routing and user's balance store in routing message buffer
642
Store address of current node in routing message buffer
644
645
Intercept criteria met?
N
Y
Append lawful intercept fields from caller dialing profile to routing message
647
649
Callee dialing profile has LI flag=on?
N
Y
Append lawful intercept fields from callee dialing profile to routing message
651
Send routing message to call controller
646
End

FIG. 8C

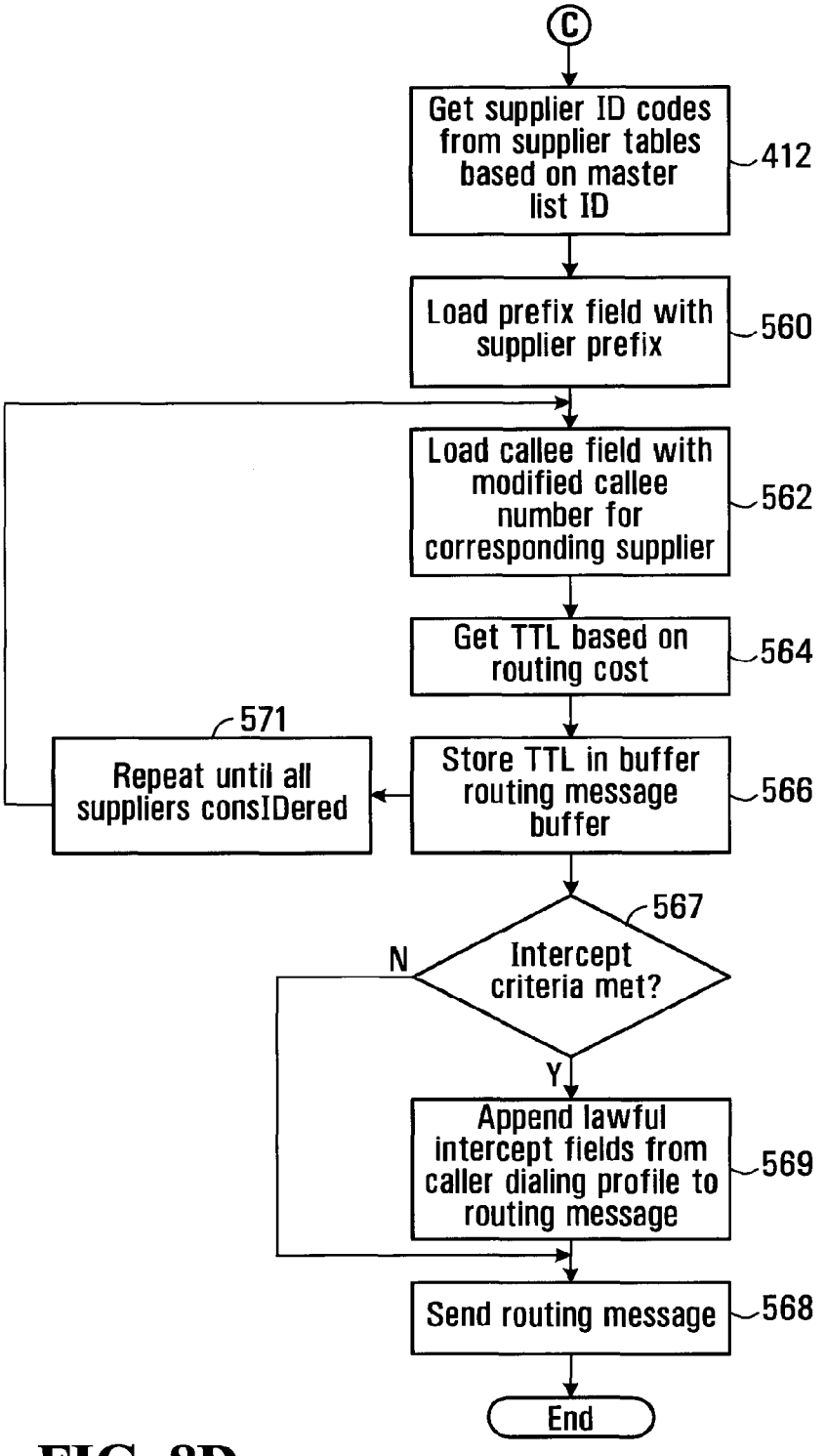

C
Get supplier ID codes from supplier tables based on master list ID
412
Load prefix field with supplier prefix
560
Load callee field with modified callee number for corresponding supplier
562
Get TTL based on routing cost
564
571
Repeat until all suppliers consIDered
Store TTL in buffer routing message buffer
566
567
N
Intercept criteria met?
Y
Append lawful intercept fields from caller dialing profile to routing message
569
Send routing message
568
End

FIG. 8D

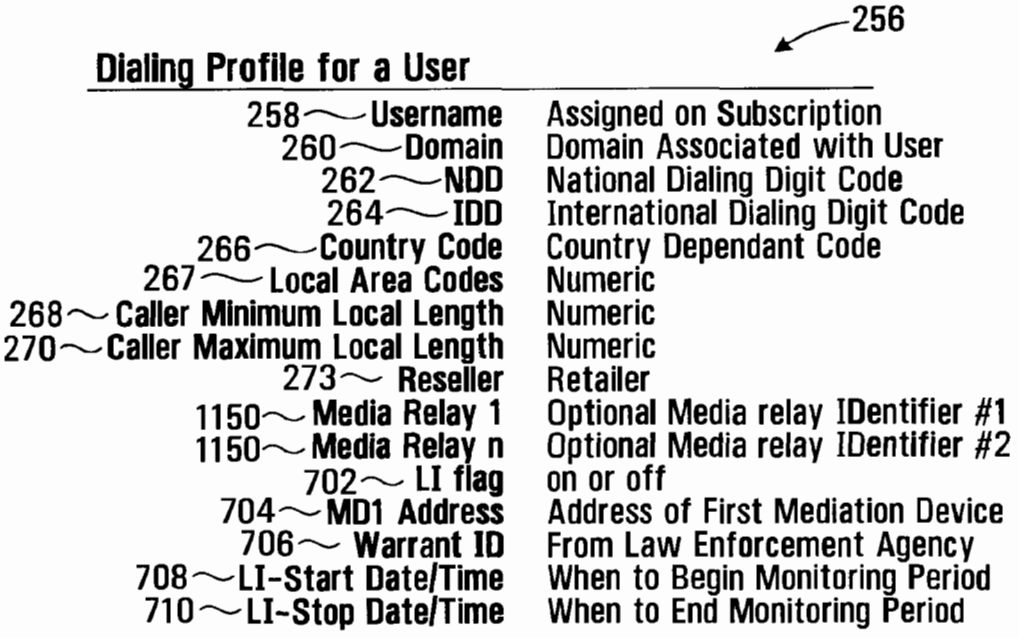

256
Dialing Profile for a User
258 Username
Assigned on Subscription
260 Domain
Domain Associated with User
262 NDD
National Dialing Digit Code
264 IDD
International Dialing Digit Code
266 Country Code
Country Dependant Code
267 Local Area Codes
Numeric
268 Caller Minimum Local Length
Numeric
270 Caller Maximum Local Length
Numeric
273 Reseller
Retailer
1150 Media Relay 1
Optional Media relay IDentifier #1
1150 Media Relay n
Optional Media relay IDentifier #2
702 LI flag
on or off
704 MD1 Address
Address of First Mediation Device
706 Warrant ID
From Law Enforcement Agency
708 LI-Start Date/Time
When to Begin Monitoring Period
710 LI-Stop Date/Time
When to End Monitoring Period

FIG. 9

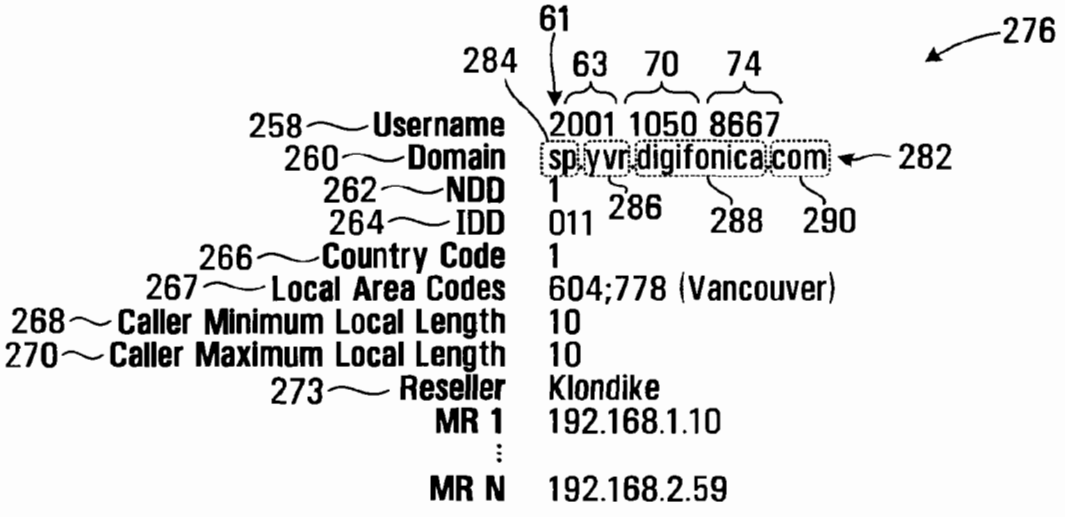

Dialing Profile for Vancouver Subscriber
276
61
284
63
70
74
258 Username
2001 1050 8667
260 Domain
sp yvr digifonica com
282
262 NDD
1
264 IDD
011
286
288
290
266 Country Code
1
267 Local Area Codes
604;778 (Vancouver)
268 Caller Minimum Local Length
10
270 Caller Maximum Local Length
10
273 Reseller
Klondike
MR 1
192.168.1.10
MR N
192.168.2.59

FIG. 10

Dialing Profile for Calgary Subscriber

Username	2001 1050 2222
Domain	sp.yvr.digifonica.com
NDD	1
IDD	011
Country Code	1
Local Area Codes	403 (Calgary)
Caller Minimum Local Length	7
Caller Maximum Local Length	10
Reseller	ABC
MR1	192.168.3.60
⋮	
MRn	192.168.4.69

FIG. 11

Dialing Profile for London Subscriber

Username	4401 1062 4444
Domain	sp.lhr.digifonica.com
NDD	0
IDD	00
Country Code	44
Local Area Codes	20 (London)
Caller Minimum Local Length	10
Caller Maximum Local Length	11
Reseller	DEF
MR1	192.168.5.70
⋮	
MRn	192.168.6.79

FIG. 12

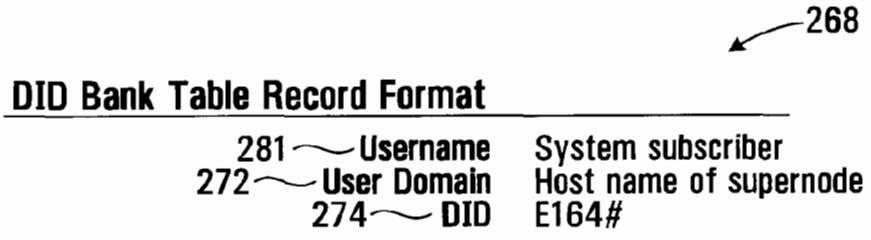

268
DID Bank Table Record Format
281 Username
System subscriber
272 User Domain
Host name of supernode
274 DID
E164#

FIG. 13

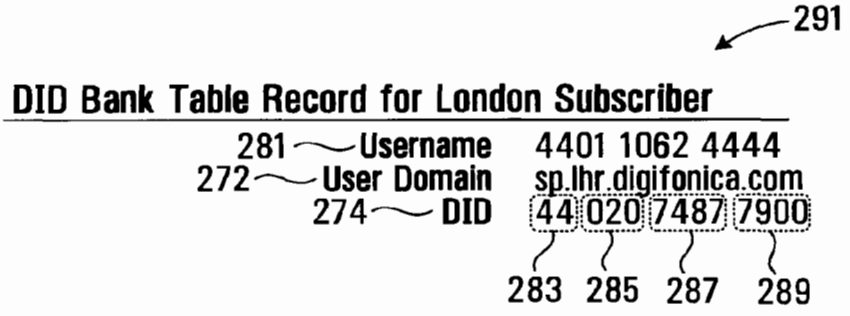

291
DID Bank Table Record for London Subscriber
281 Username
4401 1062 4444
272 User Domain
sp.lhr.digifonica.com
274 DID
44 020 7487 7900
283 285 287 289

FIG. 14

352

Routing Message Format

354	Supplier Prefix (optional)	Code IDentifying supplier traffic
356	Delimiter	Symbol separating fields
358	Callee	PSTN compatible number or Digifonica number
360	Route	Domain name and IP address
362	Time to Live(TTL)	In seconds
364	Other	TBD

FIG. 15

366

Routing Message - Different Node

4401106244444@sp.lhr.digifonica.com;ttl=9999

358 360 362

1152 Media Relays (optional)

FIG. 16

Routing Message - Different Node with lawful intercept fields

4401106244444@sp.lhr.digifonica.com;ttl=999;LIflag=on;MDaddress=192.168.1.10;
WarrantID=20060515142;
LIstart=2006 05 16 00:00:00
LIstop=2006 12 31 23:59:59;
1152 Media Relays (optional)

FIG. 16A

370

Prefix to Supernode Table Record Format

372	Prefix	First n digits of callee IDentifier
374	Supernode Address	IP address or fully qualified domain name

FIG. 17

Prefix to Supernode Table Record for London Subscriber

Prefix	4
Supernode Address	sp.lhr.digifonica.com

FIG. 18

Master List Record Format

500	ml_ID	1019
502	Dialing code	1604
504	Country code	The country code is the national prefix to be used when dialing TO a particular country FROM another country.
506	Nat Sign #(Area Code)	Numeric
508	Min Length	Numeric
510	Max Length	Numeric
512	NDD	The NDD prefix is the access code used to make a call **WITHIN** that country from on city to another (when calling another city in the same vicinity, this may not be necessary).
514	IDD	The **IDD prefix** is the international prefix needed to dial a call **FROM** the country listed TO another country.
516	Buffer rate	Safe charge rate above the highest rate charged by suppliers

FIG. 19

Example: Master List Record with Populated Fields

Route_ID	1019
Dialing code	1604
Country code	1
Nat Sign #(Area Code)	604
Min Length	7
Max Length	7
NDD	1
IDD	011
Buffer rate	$0.009/min

FIG. 20

Suppliers List Record Format

540	Sup_ID	Name code
542	Route_ID	Numeric code
544	Prefix (optional)	String IDentifying supplier's traffic #
546	Route	IP address
548	NDD/IDD rewrite	
550	Rate	Cost per second to Digifonica to use this route

FIG. 21

Telus Supplier Record

	Sup_ID	2010 (Telus)
	Route_ID	1019
	Prefix (optional)	4973#
546	Route	72.64.39.58
	NDD/IDD rewrite	011
550	Rate	$0.02/min

FIG. 22

Shaw Supplier Record

	Sup_ID	2011 (Shaw)
	Route_ID	1019
	Prefix (optional)	4974#
	Route	73.65.40.59
	NDD/IDD rewrite	011
550	Rate	$0.025/min

FIG. 23

Sprint Supplier Record

	Sup_ID	2012 (Sprint)
	Route_ID	1019
	Prefix (optional)	4975#
	Route	74.66.41.60
	NDD/IDD rewrite	011
550	Rate	$0.03/min

FIG. 24

Routing Message Buffer for Gateway Call

4973#0116048675309@72.64.39.58;ttl=3600 ~570
4974#0116048675309@73.65.40.59;ttl=3600 ~572
4975#0116048675309@74.66.41.60;ttl=3600 ~574
Media Relays (optional) ~1152

FIG. 25

Routing Message Buffer for Gateway Call with Lawful Intercept Fields

4973#0116048675309@72.64.39.58;ttl=3600
4974#0116048675309@73.65.40.59;ttl=3600
4975#0116048675309@74.66.41.60;ttl=3600
LIflag=on;MDaddress=192.168.1.10;WarrandID=20060515142;
LIstart=2006051600:00:00;LIstop=2006123123:59:59
Media Relays (optional) ~1152

FIG. 25A

Call Block Record Format

604~	Username	Digifonica #
606~	Block Pattern	PSTN compatible or Digifonica #

FIG. 26

Call Block Record for Calgary Callee

604~	Username of Callee	2001 1050 2222
606~	Block Pattern	2001 1050 8664

FIG. 27

Call Forwarding Record Format for Callee

614~	Username of Callee	Digifonica #
616~	Destination Number	Digifonica #
618~	Sequence Number	Integer indicating order to try this

FIG. 28

Call Forwarding Table Record for Calgary Callee

614	Username of Callee	2001 1050 2222
616	Destination Number	2001 1055 2223
618	Sequence Number	1

FIG. 29

Voicemail Table Record Format

624	Username of Callee	Digifonica #
626	Vm Server	domain name
628	Seconds to Voicemail	time to wait before engaging voicemail
630	Enabled	yes/no

FIG. 30

Voicemail Table Record for Calgary Callee

624	Username of Callee	2001 1050 2222
626	Vm Server	vm.yvr.digifonica.com
628	Seconds to Voicemail	20
630	Enabled	1

FIG. 31

Routing Message Buffer for CF/VM Routing Message

650 ~ 200110502222@sp.yvr.digifonica.com;ttl=3600
652 ~ 200110552223@sp.yvr.digifonica.com;ttl=3600
654 ~ vm.yvr.digifonica.com;20;ttl=60
656 ~ sp.yvr.digifonica.com
1152 ~ Media Relays (optional)

FIG. 32

Routing Message Buffer for CF/VM Routing Message with Caller Lawful Interrupt Fields

200110502222@sp.yvr.digifonica.com;ttl=3600
200110552223@sp.yvr.digifonica.com;ttl=3600
vm.yvr.digifonica.com;20;ttl=60
sp.yvr.digifonica.com
LIflag=on;MDaddress=192.168.1.10;WarrantID=20060615142;
LIstart=2006061500:00:00;LIstop=2006123123:59:59
Media Relays (optional) ~ 1152

FIG. 32A

Routing Message Buffer for CF/VM Routing Message with Caller and Callee Lawful Interrupt Fields

200110502222@sp.yvr.digifonica.com;ttl=3600
200110552223@sp.yvr.digifonica.com;ttl=3600
vm.yvr.digifonica.com;20;ttl=60
sp.yvr.digifonica.com
LI1flag=on;Mdaddress=192.168.1.10;WarrantID=20060515142;
LI1start=2006051600:00:00;LI1stop=2006123123:59:59
LI2flag=0;MD2address=192.168.1.20;WarrantID=20060615142;
LI2start=2006061500:00:00;LI2stop=2006123123:59:59
Media Relays (optional) ~ 1152

FIG. 32B

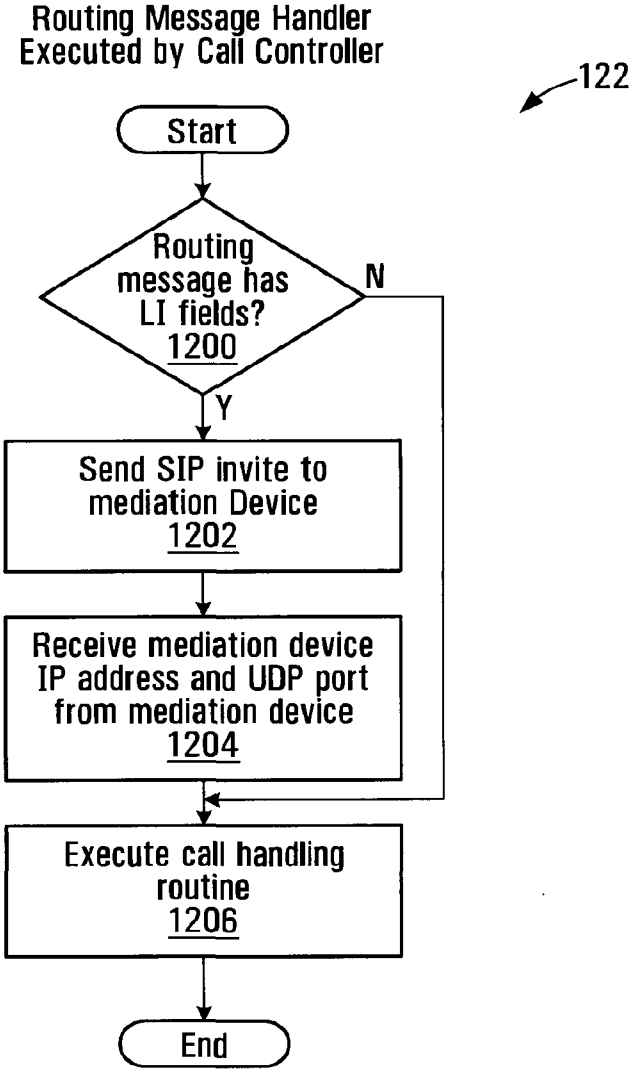

Routing Message Handler
Executed by Call Controller
122
Start
Routing message has LI fields? 1200
N
Y
Send SIP invite to mediation Device 1202
Receive mediation device IP address and UDP port from mediation device 1204
Execute call handling routine 1206
End

FIG. 33

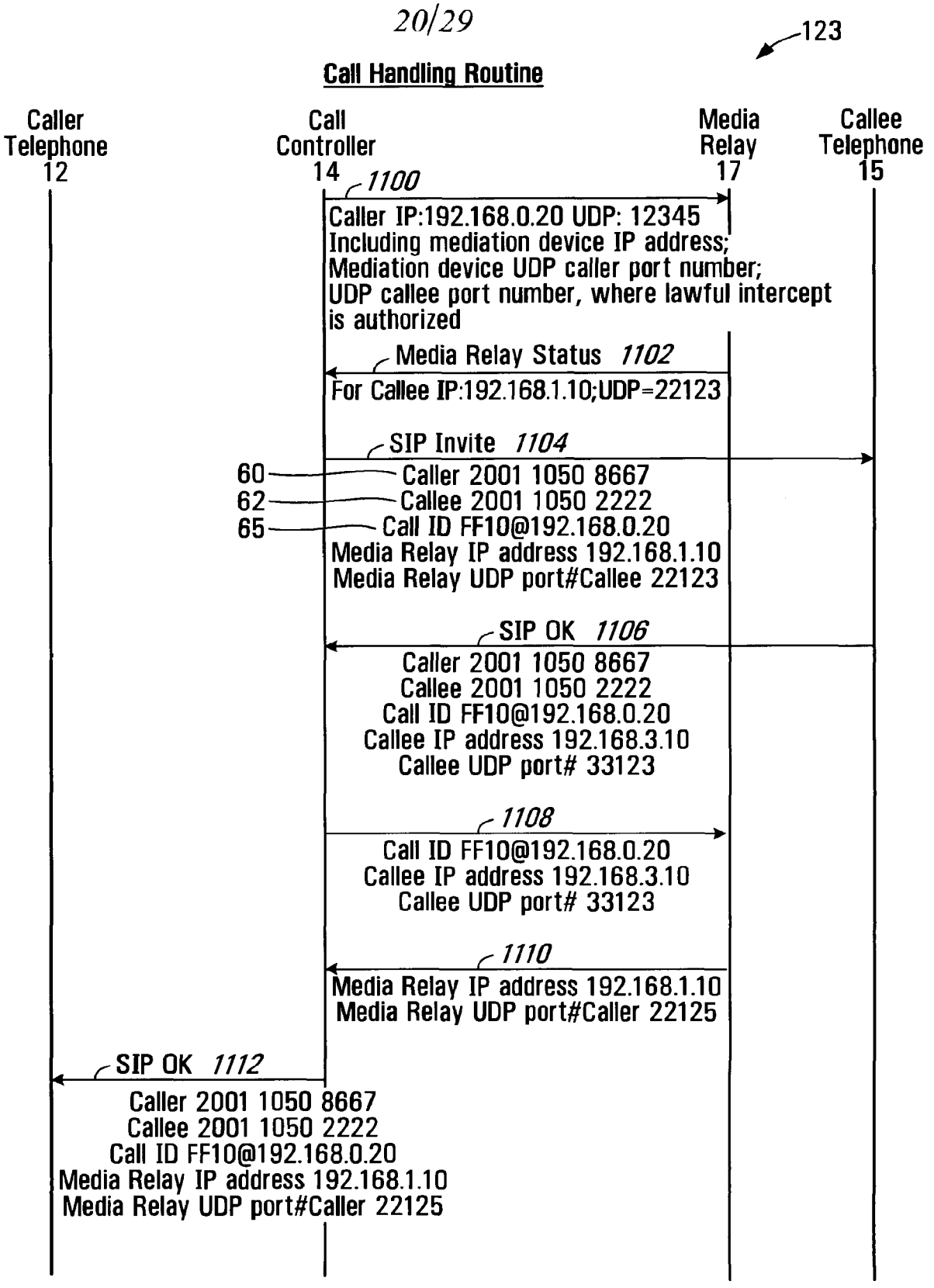

123
Call Handling Routine
Caller Telephone 12
Call Controller 14
Media Relay 17
Callee Telephone 15
1100
Caller IP:192.168.0.20 UDP: 12345
Including mediation device IP address;
Mediation device UDP caller port number;
UDP callee port number, where lawful intercept is authorized
Media Relay Status 1102
For Callee IP:192.168.1.10;UDP=22123
SIP Invite 1104
60
Caller 2001 1050 8667
62
Callee 2001 1050 2222
65
Call ID FF10@192.168.0.20
Media Relay IP address 192.168.1.10
Media Relay UDP port#Callee 22123
SIP OK 1106
Caller 2001 1050 8667
Callee 2001 1050 2222
Call ID FF10@192.168.0.20
Callee IP address 192.168.3.10
Callee UDP port# 33123
1108
Call ID FF10@192.168.0.20
Callee IP address 192.168.3.10
Callee UDP port# 33123
1110
Media Relay IP address 192.168.1.10
Media Relay UDP port#Caller 22125
SIP OK 1112
Caller 2001 1050 8667
Callee 2001 1050 2222
Call ID FF10@192.168.0.20
Media Relay IP address 192.168.1.10
Media Relay UDP port#Caller 22125

FIG. 34

Call Controller Active Call Record

1300	Call ID	FF10@192.168.0.20
1302	Caller IP Address	192.168.0.20
1304	Caller Port	12345
1306	Callee IP Address	192.168.3.10
1308	Callee Port	33123
1310	Media Relay ID	42
1312	Media Relay Caller Port	22125
1314	Media Relay Callee Port	22123

FIG. 35

Routing Controller Active Call Record

1316	Call ID	FF10@192.168.0.20
1318	Caller	2001 1050 8667
1320	Callee	2001 1050 2222
1322	Call Controller ID	61

FIG. 36

Message from Call Controller to Mediation Device - SIP Invite

1020	Caller	2001 1050 8667
1022	Callee	2001 1050 2222
1024	Call ID	FF10@192.168.0.20
1026	Warrant ID	12345678
1028	Intercept Related Info	XXXXXXXX

FIG. 37

Reply Message from Mediation Device - SIP Ok

1040	Caller	2001 1050 8667
1042	Callee	2001 1050 2222
1044	Call ID	FF10@192.168.0.20
1046	Mediation Device IP Address	192.138.2.10
1048	Mediation Device UDP Port # Caller	13678
1050	Mediation Device UDP Port # Callee	13679

FIG. 38

900

SIP Bye Message

902	Caller	Username
904	Callee	PSTN compatible # or Username
906	Call ID	unique call IDentifier (hexadecimal string@IP))

FIG. 39

908

SIP Bye Message

902	Caller	2001 1050 8667
904	Callee	2001 1050 2222
906	Call ID	FA10@192.168.0.20

FIG. 40

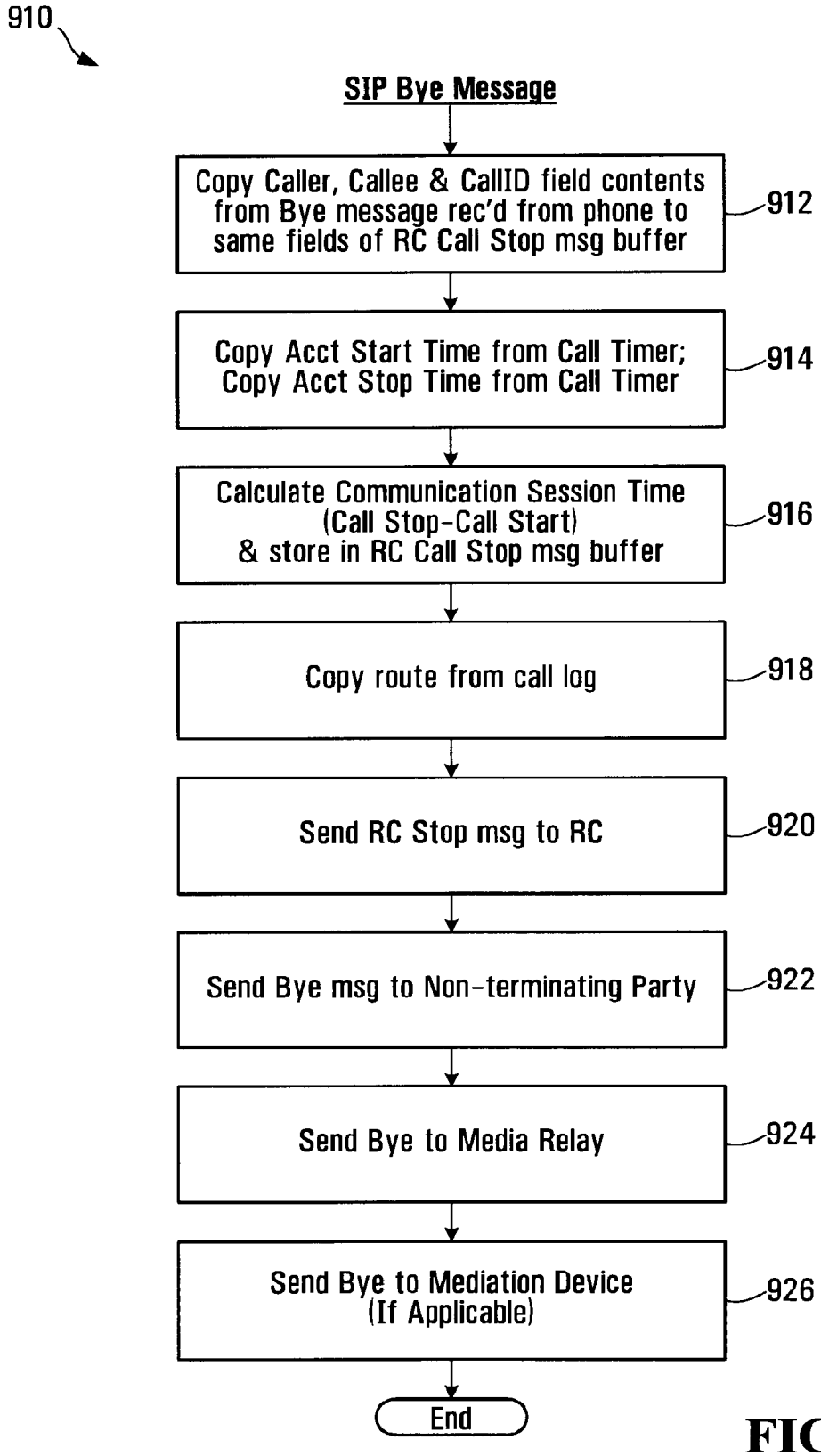

910
SIP Bye Message
Copy Caller, Callee & CallID field contents from Bye message rec'd from phone to same fields of RC Call Stop msg buffer
912
Copy Acct Start Time from Call Timer; Copy Acct Stop Time from Call Timer
914
Calculate Communication Session Time (Call Stop-Call Start) & store in RC Call Stop msg buffer
916
Copy route from call log
918
Send RC Stop msg to RC
920
Send Bye msg to Non-terminating Party
922
Send Bye to Media Relay
924
Send Bye to Mediation Device (If Applicable)
926
End

FIG. 41

1000

RC Call Stop Message

1002	**Caller**	Username
1004	**Callee**	PSTN compatible # or Username
1006	**Call ID**	unique call IDentifier (hexadecimal string@IP)
1008	**Acct Start Time**	start time of call
1010	**Acct Stop Time**	time the call ended
1012	**Acct Session Time**	start time-stop time (in seconds)
1014	**Route**	IP address for gateway, where a gateway is used

FIG. 42

1021

RC Call Stop Message for Calgary Callee

1002	**Caller**	2001 1050 8667
1004	**Callee**	2001 1050 2222
1006	**Call ID**	FA10@192.168.0.20
1008	**Acct Start Time**	2006-12-30 12:12:12
1010	**Acct Stop Time**	2006-12-30 12:12:14
1012	**Acct Session Time**	2
1014	**Route**	(72.64.39.58 if Telus gateway is used)

FIG. 43

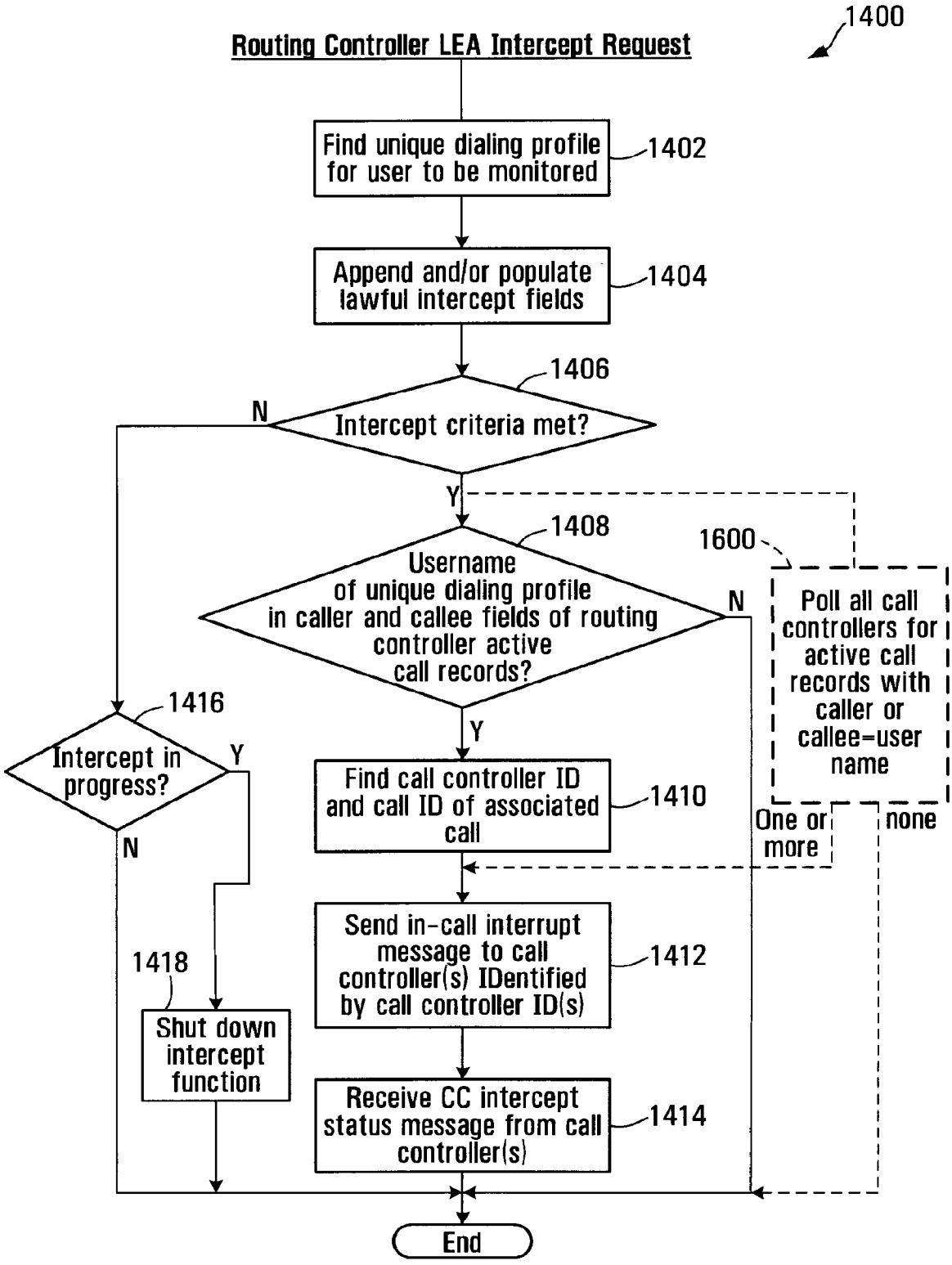

Routing Controller LEA Intercept Request
1400
Find unique dialing profile for user to be monitored
1402
Append and/or populate lawful intercept fields
1404
1406
Intercept criteria met?
N
Y
1408
Username of unique dialing profile in caller and callee fields of routing controller active call records?
N
Y
1600
Poll all call controllers for active call records with caller or callee=user name
One or more
none
1416
Intercept in progress?
Y
N
Find call controller ID and call ID of associated call
1410
Send in-call interrupt message to call controller(s) IDentified by call controller ID(s)
1412
1418
Shut down intercept function
Receive CC intercept status message from call controller(s)
1414
End

FIG. 44

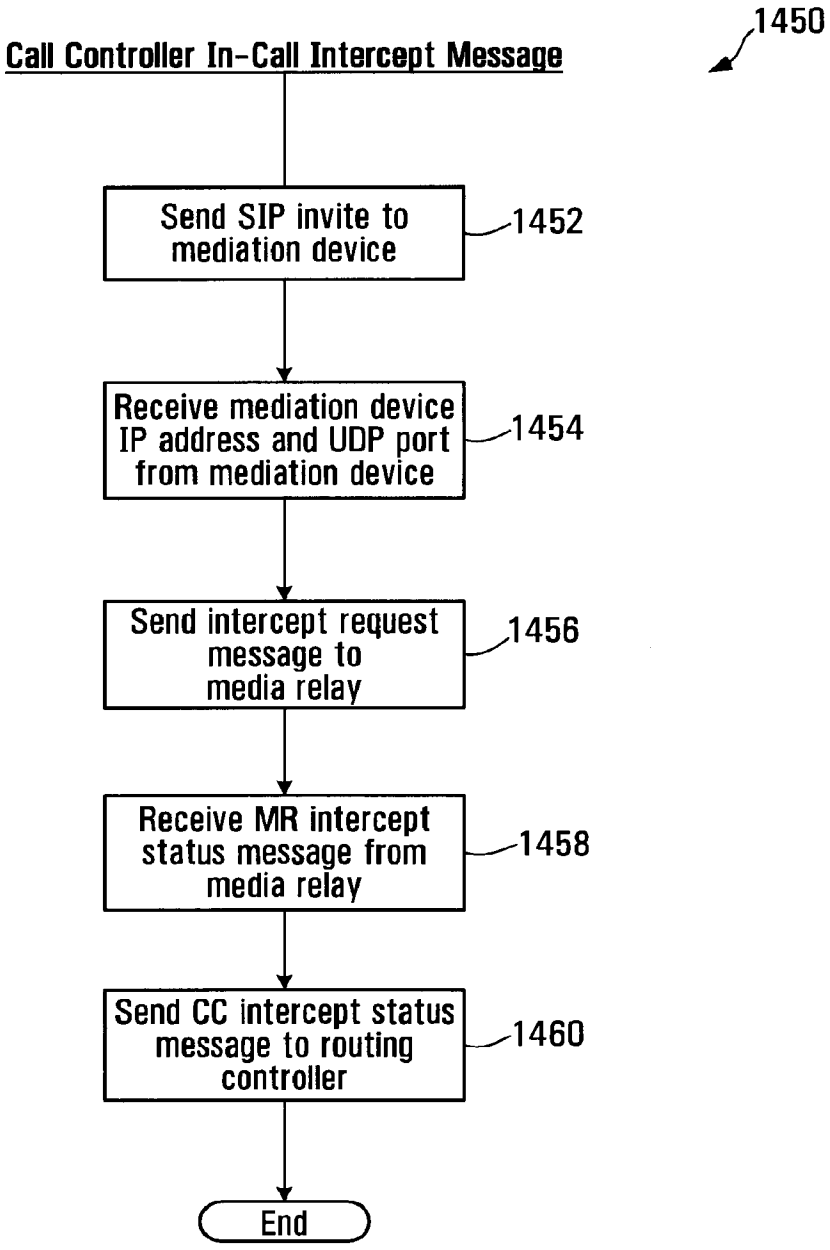

Call Controller In-Call Intercept Message
1450
Send SIP invite to mediation device
1452
Receive mediation device IP address and UDP port from mediation device
1454
Send intercept request message to media relay
1456
Receive MR intercept status message from media relay
1458
Send CC intercept status message to routing controller
1460
End

FIG. 45

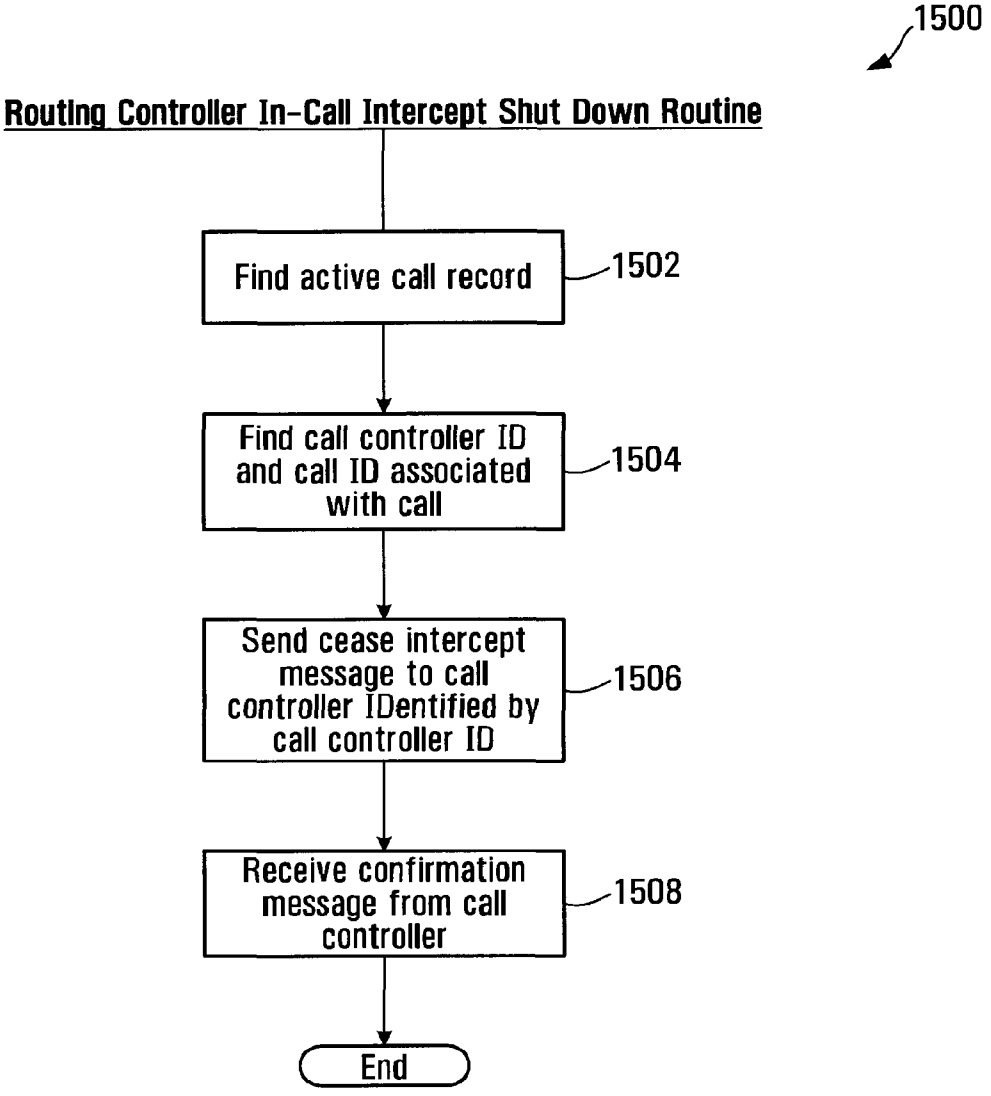

1500
Routing Controller In-Call Intercept Shut Down Routine
Find active call record
1502
Find call controller ID and call ID associated with call
1504
Send cease intercept message to call controller IDentified by call controller ID
1506
Receive confirmation message from call controller
1508
End

FIG. 46

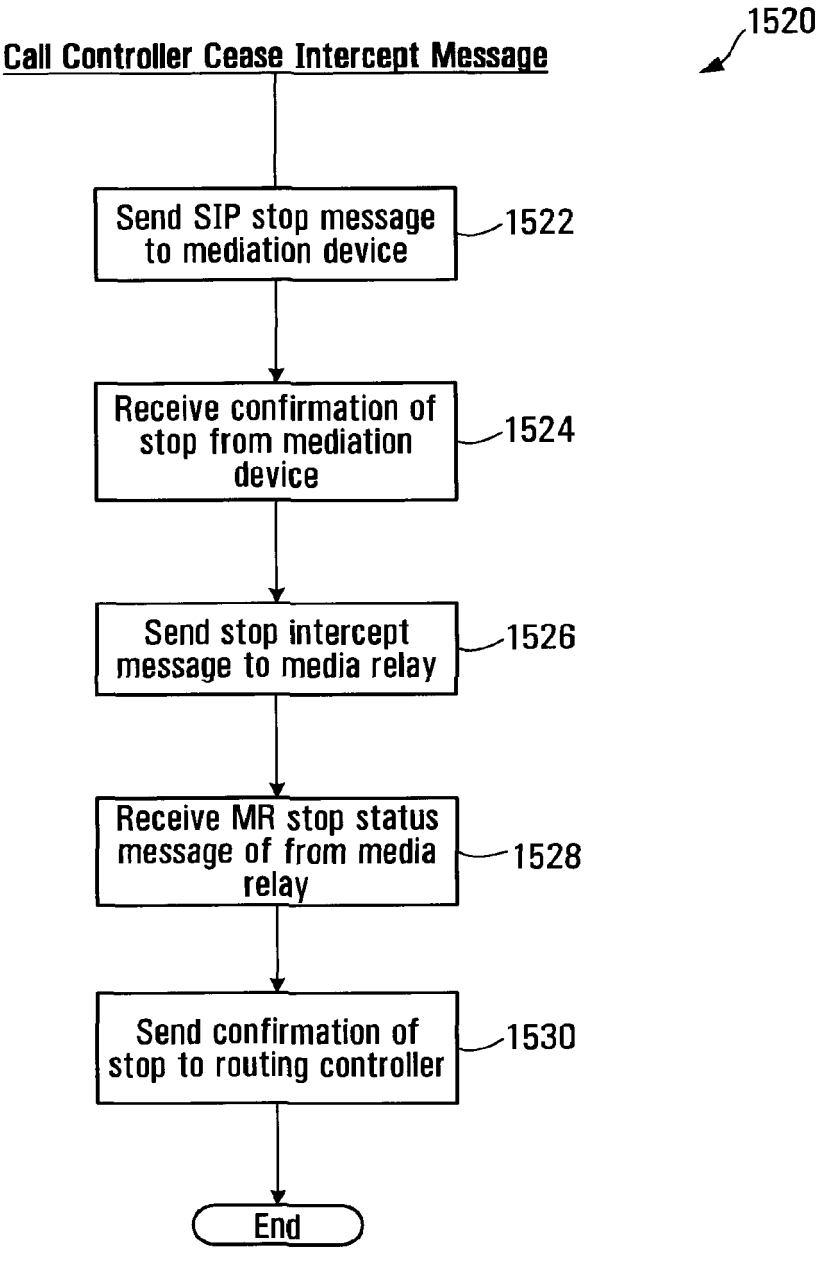

Call Controller Cease Intercept Message
1520
Send SIP stop message to mediation device
1522
Receive confirmation of stop from mediation device
1524
Send stop intercept message to media relay
1526
Receive MR stop status message of from media relay
1528
Send confirmation of stop to routing controller
1530
End

FIG. 47

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2007/002150

A. CLASSIFICATION OF SUBJECT MATTER
IPC: ***H04L 12/26*** (2006.01) , ***H04L 12/66*** (2006.01) , ***H04M 11/06*** (2006.01) , ***H04M 3/22*** (2006.01)
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC: ***H04L 12/26*** (2006.01) , ***H04L 12/66*** (2006.01) , ***H04M 11/06*** (2006.01) , ***H04M 3/22*** (2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)
West, Delphion, Canadian Patents Database, IEEEXplore, Google
Keywords: lawful intercept, (monitor* OR record* or intercept*) near (communicat* OR voip OR phone call* OR audio OR video), electronic surveillance, intercept* near device*, intercept* same IP network*, record* same IP network*, intercept* same voip, record*

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	US 2004/0181599 A1 (Kreusch et al.) 16 September 2004 (16-09-2004) *paragraphs [0011]-[0015], [0019]-[0022], [0028], [0034]-[0036], [0048]-[0053], [0055]-[0061], [0067], [0072]-[0074], [0078]-[0083]; Figs. 1, 2a-2b; claims 1-3, 7-8, 25-26*	1-2, 4-5, 10-15, 17-18, 23-26 3, 6-9, 16, 19-22
X	US 2003/0219103 A1 (Rao et al.) 27 November 2003 (27-11-2003) *Abstract; paragraphs [0005], [0026]-[0037], [0051]-[0062], [0071]-[0079], [0086]-[0090]; Figs. 1-9*	1, 14
Y	US 2002/0051518 A1 (Bondy et al.) 2 May 2002 (02-05-2002) *Abstract; paragraphs [0030]-[0032], [0036]-[0037], [0039], [0044]-[0052], [0055]-[0057], [0060]; Figs 1, 3, 5; claims 1-5*	3, 6-9, 16, 19-22
A	EP 1 389 862 B1 (Shen et al.) 3 November 2004 (03-11-2004) *paragraphs [0007]-[0014], [0051]-[0060]; Fig. 2; claim 1*	1-3, 14-16
A	US 2004/0165709 A1 (Pence et al.) 26 August 2004 (26-08-2004) *whole document*	1-26

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents :
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
3 March 2008 (03-03-2008)	14 March 2008 (14-03-2008)
Name and mailing address of the ISA/CA Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476	Authorized officer Daniela Savin 819- 934-4890

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2004/0255126 A1 (Reith) 16 December 2004 (16-12-2004) *paragraphs [0010]-[0014], [0020]-[0031], [0041]-[0046]; Figs. 1-2, 4-6; claims 1, 6-10*	1-26
A	US 2004/0157629 A1 (Kallio et al.) 12 August 2004 (12-08-2004) *paragraphs [0006]-[0021], [0050]-[0057], [0080]-[0109]; Figs. 1-12; claims 1, 7-23, 29-43*	1-26
A	US 2005/0174937 A1 (Scoggins et al.) 11 August 2005 (11-08-2005) *paragraphs [0068]-[0089], [0112]-[0138], [0153]-[0156], [0173]-[0176], [0184]-[0193]; Figs. 1-11; claims 1-2*	1-26

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US2004181599	16-09-2004	BR0208272 A	09-03-2004
		CN1274114 C	06-09-2006
		DE50211291 D1	10-01-2008
		EP1244250 A1	25-09-2002
		EP1371173 A1	17-12-2003
		RU2280331 C2	20-07-2006
		WO02082728 A1	17-10-2002
US2003219103	27-11-2003	NONE	
US2002051518	02-05-2002	US7006508 B2	28-02-2006
		US2001052081 A1	13-12-2001
		US2002009973 A1	24-01-2002
EP1389862	03-11-2004	AT281734 T	15-11-2004
		DE60201827 D1	09-12-2004
		DE60201827 T2	10-11-2005
		ES2229073 T3	16-04-2005
		US2004202295 A1	14-10-2004
US2004165709	26-08-2004	NONE	
US2004255126	16-12-2004	DE602004003518 D1	18-01-2007
		EP1484892 A2	08-12-2004
US2004157629	12-08-2004	EP1396113 A1	10-03-2004
		WO02093838 A1	21-11-2002
US2005174937	11-08-2005	US2006212933 A1	21-09-2006

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 October 2008 (02.10.2008)

PCT

(10) International Publication Number
WO 2008/116296 A1

(51) International Patent Classification:
H04L 12/66 (2006.01) *H04Q 3/00* (2006.01)
H04M 11/06 (2006.01) *H04Q 3/64* (2006.01)
H04M 3/42 (2006.01)

(21) International Application Number:
PCT/CA2008/000545

(22) International Filing Date: 20 March 2008 (20.03.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/907,224 26 March 2007 (26.03.2007) US

(71) Applicant *(for all designated States except US)*: **DIGIFONICA (INTERNATIONAL) LIMITED** [CA/CA]; Suite 1401, 4710 Kingsway, Burnaby, British Columbia V5H 4M2 (CA).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **BJORSELL, Johan, Emil, Viktor** [SE/CA]; 203 - 2288 Broadway West, Vancouver, British Columbia V6K 0B3 (CA). **SOBOLYEV, Maksym** [RU/CA]; 307 - 265 Tenth Street, New Westminster, British Columbia V3M 3Y1 (CA).

(74) Agents: **KNOX, John, W.** et al.; Smart & Biggar, 2200 - 650 West Georgia StreetBox 11560, Vancouver Centre, Vancouver, British Columbia V6B 4N8 (CA).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(54) **Title:** EMERGENCY ASSISTANCE CALLING FOR VOICE OVER IP COMMUNICATIONS SYSTEMS

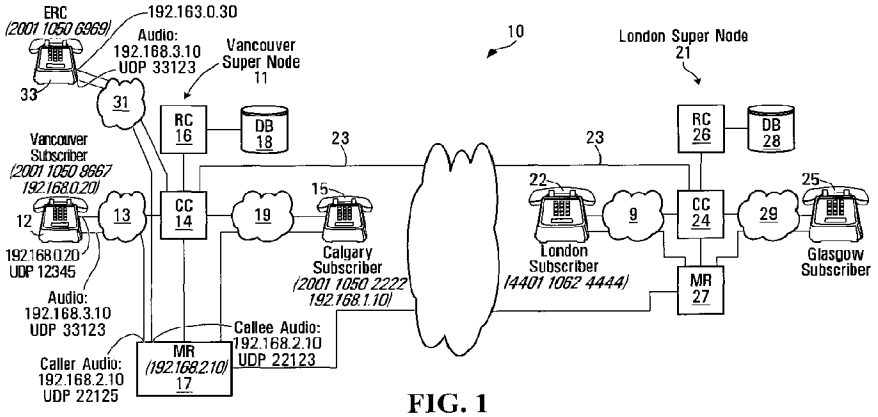


FIG. 1

(57) **Abstract:** In accordance with one aspect of the invention there is provided a process for handling emergency calls from a caller in a voice over IP system. The process involves receiving a routing request message including a caller identifier and a callee identifier. The process also involves setting an emergency call flag active in response to the callee identifier matching an emergency call identifier pre-associated with the caller. The process further involves producing an emergency response center identifier in response to the emergency call identifier. The process also involves determining whether the caller identifier is associated with a pre-associated direct inward dialing (DID) identifier. The process further involves producing a direct inward dialing (DID) identifier for the caller by associating a temporary DID identifier with the caller identifier when the emergency call flag is active and it is determined that the caller has no pre-associated DID. The process also involves producing a routing message including the emergency response center identifier and the temporary DID identifier for receipt by a routing controller operable to cause a route to be established between the caller and the emergency response center.

WO 2008/116296 A1

EMERGENCY ASSISTANCE CALLING FOR VOICE OVER IP COMMUNICATIONS SYSTEMS

BACKGROUND OF THE INVENTION

Field of Invention

This invention relates to emergency assistance calling, voice over internet protocol communications and methods and apparatus for emergency assistance calling for voice over IP data communications.

An essential feature of traditional telephone systems (PSTN) is the ability of its subscribers to dial a universal emergency number (911 in North America) to access a host of emergency services such as fire, police and ambulance. Because of the hierarchical nature of telephone networks and numbering schemes, a call coming from a specific telephone number on the PSTN network is automatically routed to a nearest Emergency Response Center (ERC) based on the area code and exchange code contained in the specific telephone number. Normally, the specific telephone number will be compliant with the E.**164** standard set by the International Telecommunication Union. When the call comes into the ERC, call information presended the ERC operator includes the phone number, and where available, the address associated with this phone number.

Since the late **1990**s, an enhanced emergency service (E**911**) was mandated for PSTN and cellular carriers in North America and elsewhere. In particular, with this enhanced service the information automatically provided to the ERC includes the physical location of the person calling, even where the caller is using a cellular telephone. Moreover, a callback functionality is integrated into E**911**-compliant systems allowing an ERC operator to call back the person who placed the emergency call even if the original phone call was disconnected or if the calling line became busy.

T-Mobile Exhibit 1002
Part 3 of 6

In the realm of VoIP networks, implementation of **911** and E**911** services often presents significant problems.

Even to provide basic **911** services, VoIP systems present a number of problems because they do not employ hierarchical numbering schemes, and the phone numbers assigned to VoIP system subscribers, while still in the E.**164** format, do not actually reflect the subscribers physical location via area code and exchange codes. As a result, a VoIP provider is not able to automatically route an emergency call to an ERC nearest to the subscriber. Because VoIP subscriber phone numbers are assigned from a bulk of phone numbers that VoIP providers purchase from wireline PSTN carriers, a VoIP **911** emergency services call coming into the ERC is not associated with a subscriber address that can be accessed by the ERC operator.

In addition, because VoIP systems are not based on the Signaling System **7** (SS**7**) protocol, they do not natively support special short phone numbers such as **911**. In particular, they do not natively support variable length phone number dialing, or dynamic translation of dialed universal phone numbers into actual destination phone numbers based on user attributes such as location or service type.

VoIP systems are also typically not able to comply with E**911** service requirements, for the same reasons they are not able to comply with regular **911** services.

In accordance with one aspect of the invention, there is provided a process for handling emergency calls from a caller in a voice over IP system. The method involves receiving a routing request message including a caller identifier and a callee identifier. The method also involves setting an emergency call flag active in response to the callee identifier matching an emergency call identifier pre-associated with the caller. The method further involves producing an emergency response center identifier in response to the emergency call

identifier. The method also involves determining whether the caller identifier is associated with a pre-associated direct inward dialing (DID) identifier. The method further involves producing a direct inward dialing (DID) identifier for the caller by associating a temporary DID identifier with the caller identifier when the emergency call flag is active and it is determined that the caller has no pre-associated DID identifier. The method also involves producing a routing message including the emergency response center identifier and the temporary DID identifier for receipt by a routing controller operable to cause a route to be established between the caller and the emergency response center.

Setting the emergency call flag active may involve retrieving a dialing profile associated with the caller and setting the emergency call flag active when the contents of an emergency call identifier field of the dialing profile match the callee identifier.

Determining whether the caller identifier is associated with a pre-associated DID identifier may involve searching a database for a DID record associating a DID identifier with the caller and determining that the caller identifier is associated with a pre-associated DID identifier when the record associating a DID identifier with the caller is found.

Associating a pre-assigned DID identifier with the caller identifier may involve copying the pre-associated DID identifier from the DID record to a DID identifier buffer.

Producing the routing message may involve causing the contents of the DID identifier buffer to define the DID identifier in the routing message.

Determining whether the caller identifier is associated with a pre-associated DID identifier may involve searching a database for a DID record associating a DID identifier with the caller and determining that the caller identifier is not

associated with a pre-associated DID identifier when a record associating a DID identifier with the caller is not found.

Associating a temporary DID identifier with the caller identifier may involve associating with the caller identifier a DID identifier from a pool of pre-determined DID identifiers.

Associating the DID identifier from the pool may involve associating a temporary DID record with the caller, the temporary DID record having a DID identifier field populated with the DID identifier from the pool.

Associating the DID identifier from the pool may involve copying the DID identifier from the temporary DID record to a DID identifier buffer.

The method may involve canceling the temporary DID record after a pre-defined period of time.

Producing the emergency response center identifier may involve obtaining an emergency response center identifier from an emergency response center field of the dialing profile associated with the caller.

Obtaining may involve copying an emergency response center identifier from the dialing profile associated with the caller to a routing message buffer such that the emergency response center identifier is included in the routing message.

Producing the routing message may involve causing the routing message to specify a maximum call time for the emergency call, the maximum call time exceeding a duration of an average non-emergency telephone call.

In accordance with another aspect of the invention, there is provided an apparatus for handling emergency calls from a caller in a voice over IP

system. The apparatus includes provisions for receiving a routing request message including a caller identifier and a callee identifier. The apparatus also includes setting provisions for setting an emergency call flag active in response to the callee identifier matching an emergency call identifier pre-associated with the caller. The apparatus further includes provisions for producing an emergency response center identifier in response to the emergency call identifier. The apparatus also includes provisions for determining whether the caller identifier is associated with a pre-associated direct inward dialing (DID) identifier. The apparatus further includes provisions for producing a direct inward dialing (DID) identifier for the caller including provisions for associating a temporary DID identifier with the caller identifier in response to the emergency call flag being active and the caller identifier not being pre-associated with direct inward dialing identifier. The provisions for producing a direct inward dialing (DID) identifier for the caller further include provisions for associating a pre-assigned DID identifier with the caller identifier when the caller identifier has no pre-associated direct inward dialing identifier. The apparatus also includes provisions for producing a routing message including the emergency response center identifier and the temporary DID identifier for receipt by a routing controller operable to cause a route to be established between the caller and the emergency response center.

The apparatus may further include provisions for accessing a database of dialing profiles associated with respective subscribers to the system, each of the dialing profiles including an emergency call identifier field and an emergency call center field and the setting provisions may comprise provisions for retrieving a dialing profile associated with the caller and for setting the emergency call flag active when the contents of the emergency call identifier field of the dialing profile match the callee identifier.

The apparatus may further include database accessing provisions for accessing a database including direct inward dialing (DID) records associated

with at least some subscribers to the system, each of the direct inward dialing records comprising a system username and a direct inward dialing number, and wherein the determining provisions comprise searching provisions for searching a database for a DID record associating a DID identifier with the caller. The determining provisions may be operably configured to determine that the caller identifier is associated with a pre-associated DID identifier when a record associating a DID identifier with the caller is found.

The apparatus may further include a DID identifier buffer and the provisions for associating a pre-assigned DID identifier with the caller identifier may comprise provisions for copying the pre-associated DID identifier from the DID record to the DID identifier buffer.

The provisions for producing the routing message may include provisions for causing the contents of the DID identifier buffer to define the DID identifier in the routing message.

The apparatus may further include database accessing provisions for accessing a database including direct inward dialing records associated with at least some subscribers to the system, each of the direct inward dialing records comprising a system username and a direct inward dialing number and the determining provisions may comprise searching provisions for searching a database for a DID record associating a DID identifier with the caller and wherein the determining provisions may be operably configured to determine that the caller identifier is not associated with a pre-associated DID identifier when a record associating a DID identifier with the caller is not found.

The apparatus may further include provisions for accessing a pool of pre-determined DID identifiers and the provisions for associating a temporary DID identifier with the caller identifier may comprise provisions for associating a

DID identifier from the pool of pre-determined DID identifiers with the caller identifier.

The provisions for associating the DID identifier from the pool may include provisions for associating a temporary DID record with the caller, the temporary DID record having a DID identifier field populated with the DID identifier from the pool.

The provisions for associating the DID identifier may include provisions for copying the DID identifier from the temporary DID record to a DID identifier buffer.

The apparatus may further include provisions for canceling the temporary DID record after a period of time.

The provisions for producing the emergency response center identifier may include provisions for obtaining an emergency response center identifier from an emergency response center field of the dialing profile associated with the caller.

The apparatus may include a routing message buffer and the provisions for obtaining may include provisions for copying the contents of the emergency response center field of the dialing profile associated with the caller to the routing message buffer such that the contents of the emergency response center field are included in the routing message.

The provisions for producing the routing message may include provisions for causing the routing message to include a maximum call time for the emergency call, the maximum call time exceeding a duration of an average non-emergency telephone call.

In accordance with another aspect of the invention, there is provided an apparatus for handling emergency calls from a caller in a voice over IP system. The apparatus includes an processor circuit operably configured to receive a routing request message including a caller identifier and a callee identifier. The processor circuit is also operably configured to set an emergency call flag active in response to the callee identifier matching an emergency call identifier pre-associated with the caller. The processor circuit is further operably configured to produce an emergency response center identifier in response to the emergency call identifier and to determine whether the caller identifier is associated with a pre-associated direct inward dialing (DID) identifier. The processor circuit is also operably configured to produce a direct inward dialing (DID) identifier for the caller by associating a temporary DID identifier with the caller identifier when the emergency call flag is active and it is determined that the caller identifier has no pre-associated DID identifier. The processor circuit is further operably configured to produce a routing message including the emergency response center identifier and the temporary DID identifier for receipt by a routing controller operable to cause a route to be established between the caller and the emergency response center.

The processor circuit may be operably configured to retrieve a dialing profile associated with the caller and to set the emergency call flag active when the contents of an emergency call identifier field of the dialing profile match the callee identifier.

The processor circuit may be operably configured to search a database for a DID record associating a DID identifier with the caller and to determine that the caller identifier is associated with a pre-associated DID identifier when the record associating a DID identifier with the caller is found.

The processor circuit may be operably configured to copy the pre-associated DID identifier from the DID record to a DID identifier buffer.

The processor circuit may be operably configured to cause the contents of the DID identifier buffer to define the DID identifier in the routing message.

The processor circuit may be operably configured to search a database for a DID record associating a DID identifier with the caller and to determine that the caller identifier is not associated with a pre-associated DID identifier when a record associating a DID identifier with the caller is not found.

The processor circuit may be operably configured to associate with the caller identifier a DID identifier from a pool of pre-determined DID identifiers.

The processor circuit may be operably configured to associate a temporary DID record with the caller, the temporary DID record having a DID identifier field populated with the DID identifier from the pool.

The processor circuit may be operably configured to copy the DID identifier from the temporary DID record to a DID buffer.

The processor circuit may be operably configured to cancel the temporary DID record after a period of time.

The processor circuit may be operably configured to obtain an emergency response center identifier from an emergency response center field of the dialing profile associated with the caller.

The apparatus may further a routing message buffer and the processor circuit may be operably configured to copy an emergency response center identifier from the dialing profile associated with the caller to the routing message buffer such that the emergency response center identifier is included in the routing message.

The processor circuit may be operably configured to cause the routing message to include a maximum call time for the emergency call, the maximum call time exceeding a duration of an average non-emergency telephone call.

In accordance with another aspect of the invention, there is provided a computer readable medium encoded with codes for directing a processor circuit to handle emergency calls from callers in a voice over IP system. The codes direct the processor circuit to receive a routing request message including a caller identifier and a callee identifier. The codes also direct the processor circuit to set an emergency call flag active in response to the callee identifier matching an emergency call identifier pre-associated with the caller. The codes further direct the processor circuit to produce an emergency response center identifier in response to the emergency call identifier. The codes also direct the processor circuit to determine whether the caller identifier is associated with a pre-associated direct inward dialing (DID) identifier. The codes further direct the processor circuit to produce a direct inward dialing (DID) identifier for the caller by associating a temporary DID identifier with the caller identifier when the emergency call flag is active and it is determined that the caller identifier has no pre-associated DID identifier. The codes also direct the processor circuit to produce a routing message including the emergency response center identifier and the temporary DID identifier for receipt by a routing controller operable to cause a route to be established between the caller and the emergency response center.

BRIEF DESCRIPTION OF THE DRAWINGS

In drawings which illustrate embodiments of the invention,

Figure **1** is a block diagram of a system according to a first embodiment of the invention;

Figure **2** is a block diagram of a caller VoIP telephone according to the first embodiment of the invention;

Figure **3** is a schematic representation of a SIP Invite message transmitted between the caller telephone and a call controller (CC) shown in Figure **1**;

Figure **4** is a block diagram of the call controller shown in Figure **1**;

Figure **5** is a flowchart of a process executed by the call controller shown in Figure **1**;

Figure **6** is a schematic representation of a routing controller (RC) Request message produced by the call controller shown in Figure **1**;

Figure **7** is a block diagram of a routing controller (RC) processor circuit of the routing controller shown in Figure **1**;

Figures **8**A-**8**D are flowcharts of a RC Request message handler executed by the RC processor circuit shown in Figure **7**;

Figure **9** is a tabular representation of a dialling profile stored in a database accessible by the RC shown in Figure **1**;

Figure **10** is a tabular representation of a dialling profile for a Vancouver caller using the caller telephone shown in Figure **1**;

Figure **10**A is a tabular representation of a dialling profile for the Emergency Response Center subscriber shown in Figure **1**;

Figure **11** is a tabular representation of a dialing profile for the Calgary subscriber shown in Figure **1**;

Figure **12** is a tabular representation of a dialing profile for the London subscriber shown in Figure **1**;

Figure **13** is a tabular representation of a DID bank table record stored in the database shown in Figure **1**;

Figure **13A** is a tabular representation of an exemplary DID bank table record for the Vancouver subscriber;

Figure **13B** is a tabular representation of an exemplary DID bank table record for the Calgary subscriber;

Figure **14** is a tabular representation of an exemplary DID bank table record for the London subscriber;

Figure **15** is a tabular representation of a routing message buffer for holding a routing message to be transmitted from the RC to the call controller shown in Figure **1**;

Figure **16** is a tabular representation of a routing message for routing a call to the Emergency Response Center;

Figure **16A** is a tabular representation of a routing message for routing a call to the London subscriber;

Figure **17** is a tabular representation of a prefix to supernode table record stored in the database shown in Figure **1**;

Figure **18** is a tabular representation of a prefix to supernode table record that would be used for the London subscriber;

Figure **19** is a tabular representation of a master list record stored in a master list table in the database shown in Figure **1**;

Figure **20** is a tabular representation of an exemplary populated master list record;

Figure **21** is a tabular representation of a suppliers list record stored in the database shown in Figure **1**;

Figure **22** is a tabular representation of a specific supplier list record for a first supplier;

Figure **23** is a tabular representation of a specific supplier list record for a second supplier;

Figure **24** is a tabular representation of a specific supplier list record for a third supplier;

Figure **25** is a tabular representation of a routing message buffer for holding a routing message identifying a plurality of possible suppliers that may carry the call;

Figure **26** is a tabular representation of a call block table record;

Figure **27** is a tabular representation of a call block table record for the Calgary subscriber;

Figure **28** is a tabular representation of a call forwarding table record;

Figure **29** is a tabular representation of an exemplary call forwarding table record specific to the Calgary subscriber;

Figure **30** is a tabular representation of a voicemail table record specifying voicemail parameters to enable the caller to leave a voicemail message for the callee;

Figure **31** is a tabular representation of an exemplary voicemail table record for the Calgary subscriber;

Figure **32** is a tabular representation of an exemplary routing message, held in a routing message buffer, indicating call forwarding numbers and a voicemail server identifier;

Figure **33** is a tabular representation of a SIP Bye message transmitted from any of the telephones to the call controller;

Figure **34** is a tabular representation of a SIP Bye message sent to the call controller from the callee or caller gateway;

Figure **35** is a flowchart of a process executed by the call controller for producing a RC Call Stop message in response to receipt of a SIP Bye message;

Figure **36** is a tabular representation of an exemplary RC Call Stop message;

Figure **37** is a tabular representation of an exemplary RC Call Stop message for the Calgary subscriber;

Figure **38** is a schematic representation of messages exchanged during a process for establishing audio paths between telephones and a media relay.

DETAILED DESCRIPTION

Referring to Figure **1**, a system for making voice over IP telephone calls including emergency calls is shown generally at **10**. The system includes a first supernode shown generally at **11** and a second supernode shown generally at **21**. The first supernode **11** is located in a geographical area, such as Vancouver B.C., for example and the second supernode **21** is located in London, England, for example. Different supernodes may be located in different geographical regions throughout the world to provide telephone service to subscribers in respective regions. These supernodes may be in communication with each other through high speed / high data throughput links including optical fiber, satellite and/or cable links, for example, forming a system backbone. These supernodes may alternatively or in addition be in communication with each other through conventional Internet services. In the embodiment shown, data communication media for providing for data communications between the first and second supernodes **11** and **21** are shown generally at **23** and may include very high speed data links, for example.

In the embodiment shown, the Vancouver supernode **11** provides telephone service to a geographical region comprising Western Canadian customers from Vancouver Island to Ontario and includes a Vancouver subscriber, a Calgary subscriber and an emergency response center (ERC) that is also a subscriber. The second supernode **21** may be located in London, England, for example, to service London and Glasgow subscribers, **22** and **25**, for example through their own service providers **9** and **29**. As will be seen below however, the emergency response center need not be a subscriber.

Other supernodes similar to the type shown may also be employed within the geographical area serviced by a supernode, to provide for call load sharing, for example within a region of the geographical area serviced by the supernode. However, in general, all supernodes are similar and have the properties described below in connection with the Vancouver supernode **11**.

In this embodiment, the Vancouver supernode includes a call controller (CC) **14**, a routing controller (RC) **16**, a database **18** and a media relay (MR) **17**. Subscribers such as the Vancouver subscriber, the Calgary subscriber and the Emergency Response Center subscriber communicate with the Vancouver supernode **11** using their own Internet Service Providers (ISPs) **13**, **19** and **31** respectively which route Internet Protocol (IP) traffic from these subscribers to the Vancouver Supernode over the Internet. To these subscribers the Vancouver supernode **11** is accessible through their ISP at a pre-determined IP address or a fully qualified domain name (FQDN). The subscriber in the city of Vancouver uses a telephone **12** that is capable of communicating with the Vancouver supernode **11** using Session Initiation Protocol (SIP) messages, and the Calgary and Emergency Response Center subscribers use similar telephones **15** and **33** respectively, to communicate with the Vancouver supernode from their locations. The London supernode **21** also has a call controller **24**, a routing controller **26** and a database **28** and functions in a manner similar to the Vancouver supernode **11**.

It should be noted that throughout the description of the embodiments of this invention, the IP/UDP addresses of all elements such as the caller and callee telephones, call controller, media relay, and any others, will be assumed to be valid IP/UDP addresses directly accessible via the Internet or a private IP network, for example, depending on the specific implementation of the system. As such, it will be assumed, for example, that the caller and callee telephones will have IP/UDP addresses directly accessible by the call controllers and the media relays on their respective supernodes, and those addresses will not be obscured by Network Address Translation (NAT) or similar mechanisms. In other words, the IP/UDP information contained in SIP messages (for example the SIP Invite message or the RC Request message which will be described below) will match the IP/UDP addresses of the IP packets carrying these SIP messages.

It will be appreciated that in many situations, the IP addresses assigned to various elements of the system may be in a private IP address space, and thus not directly accessible from other elements. Furthermore, it will also be appreciated that NAT is commonly used to share a "public" IP address between multiple devices, for example between home PCs and IP telephones sharing a single Internet connection. For example, a home PC may be assigned an IP address such as **192.168.0.101** and a Voice over IP telephone may be assigned an IP address of **192.168.0.103**. These addresses are located in so called "non-routable" (IP) address space and cannot be accessed directly from the Internet. In order for these devices to communicate with other computers located on the Internet, these IP addresses have to be converted into a "public" IP address, for example **24.10.10.123** assigned by the Internet Service Provider to the subscriber, by a device performing NAT, typically a home router. In addition to translating the IP addresses, NAT typically also translates UDP port numbers, for example an audio path originating at a VoIP telephone and using a UDP port **12378** at its private IP address, may have been translated to UDP port **23465** associated with the public IP address of the NAT device. In other words, when a packet originating from the above VoIP telephone arrives at an Internet-based supernode, the source IP/UDP address contained in the IP packet header will be **24.10.10.123:23465**, whereas the source IP/UDP address information contained in the SIP message inside this IP packet will be **192.168.0.103:12378**. The mismatch in the IP/UDP addresses may cause a problem for SIP-based VoIP systems because, for example, a supernode will attempt to send messages to a private address of a telephone – the messages will never get there.

It will be appreciated that a number of methods are available to overcome this problem. For example, the SIP NATHelper open source software module may run on the supernode to correlate public IP/UDP address contained in the headers of the IP packets arriving from SIP devices with private IP/UDP addresses in the SIP messages contained in these packets. Therefore, the

embodiments of the invention described below will function whether or not any of the elements of the system are located behind NAT devices that obscure their real IP/UDP addresses.

Referring to Figure **1**, in an attempt to make a regular call by the Vancouver telephone **12** to the London telephone **22**, for example, the Vancouver telephone sends a SIP Invite message to the Vancouver supernode **11** and in response, the call controller **14** sends an RC Request message to the routing controller **16** which makes various enquiries of the database **18** to produce a routing message which is sent to the call controller. The call controller **14** then causes a communications link, including audio paths, to be established through the media relay **17** which may include the same Vancouver supernode **11**, a different supernode or a communications supplier gateway, for example, to carry voice traffic to and from the call recipient or callee.

In an attempt to make an emergency call, generally the call is made by dialling a short number such as **911** and the call is routed to an emergency response center (ERC) associated with the caller such as the emergency response center associated with the telephone **33**. However, as will be appreciated from the description below, this system will permit emergency calls originating from subscribers associated with one supernode to be received by emergency response centers associated with a different supernode, if necessary.

Subscriber Telephone

Referring to Figure **2**, in this embodiment, the telephone **12** includes a processor circuit shown generally at **30** comprising a microprocessor **32**, program memory **34**, an input/output (I/O) interface **36**, parameter memory **38** and temporary memory **40**. The program memory **34**, I/O interface **36**, parameter memory **38** and temporary memory **40** are all in communication with the microprocessor **32**. The I/O interface **36** has a dial input **42** for receiving a dialed telephone number from a keypad, for example, or from a

voice recognition unit or from pre-stored telephone numbers stored in the parameter memory **38**, for example. For simplicity, a box labelled dialling functions **44** represents any device capable of informing the microprocessor **32** of a callee identifier, e.g., a callee telephone number.

The processor **32** stores the callee identifier in a dialed number buffer **41**. Where the callee is the London subscriber, the callee identifier may be **4401 1062 4444**, for example, identifying the London subscriber or the callee identifier may be a standard telephone number, or where the callee is the Emergency Response Center, the callee identifier may be **911**, for example.

The I/O interface **36** also has a handset interface **46** for receiving and producing signals from and to a handset that receives user's speech to produce audio signals and produces sound in response to received audio signals. The handset interface **46** may include a BLUETOOTH™ wireless interface, a wired interface or speakerphone, for example. The handset **45** acts as a termination point for an audio path (not shown) which will be appreciated later.

The I/O interface **36** also has a network interface **48** to an IP network, and is operable, for example, to connect the telephone to an ISP via a high speed Internet connection. The network interface **48** also acts as a part of the audio path, as will be appreciated later.

The parameter memory **38** has a username field **50**, a password field **52**, an IP address field **53** and a SIP proxy address field **54**. The username field **50** is operable to hold a username associated with the telephone **12**, which in this case is **2001 1050 8667**. The username is assigned upon subscription or registration into the system and, in this embodiment includes a twelve digit number having a prefix **61**, a country code **63**, a dealer code **70** and a unique number code **74**. The prefix **61** is comprised of the first or left-most digit of the username in this embodiment. The prefix may act as a continent code in

some embodiments, for example. The country code **63** is comprised of the next three digits. The dealer code **70** is comprised of the next four digits and the unique number code **74** is comprised of the last four digits. The password field **52** holds a password of up to **512** characters, in this example. The IP address field **53** stores an IP address of the telephone **30**, which for this explanation is **192.168.0.20**. The SIP proxy address field **54** stores an IP address of a SIP proxy which may be provided to the telephone **12** through the network interface **48** as part of a registration procedure, for example.

The program memory **34** stores blocks of codes for directing the microprocessor **32** to carry out the functions of the telephone **12**, one of which includes a firewall block **56** which provides firewall functions to the telephone, to prevent unauthorized access through the network interface **48** to the microprocessor **32** and memories **34, 38** and **40**. The program memory **34** also stores codes **57** for establishing a call ID. The call ID codes **57** direct the microprocessor **32** to produce call identifiers, that may, for example have the format of a hexadecimal string and an IP address of the telephone stored in IP address field **53**. Thus, an exemplary call identifier for a call might be FF**10** @ **192.168.0.20**.

Generally, in response to activating the handset **45** and using the dialling function **44**, the microprocessor **32** produces and sends a SIP Invite message **59** as shown in Figure **3**, to the routing controller (RC) **14** shown in Figure **1**.

Referring to Figure **3**, the SIP Invite message includes a caller identifier field **60**, a callee identifier field **62**, a digest parameters field **64**, a call ID field **65**, a caller IP address field **67** and a caller UDP port field **69**. In this embodiment, the caller identifier field **60** includes the username **2001 1050 8667**, which is the username stored in the username field **50** of the parameter memory **38** in the Vancouver telephone **12** shown in Figure **2**. In addition, as an example, referring back to Figure **3**, where the call is a normal, non-emergency call to the London subscriber the callee identifier field **62** includes the username

4401 1062 4444 which is the dialed number of the London subscriber stored in the dialed number buffer **41** shown in Figure **2**. The digest parameters field **64** includes digest parameters and the call ID field **65** includes a code comprising a generated prefix code (FF**10**, for example) and a suffix which is the IP address of the telephone **12** stored in the IP address field **53**. The IP address field 67 and UDP port field 69 define a socket for audio communications. The IP address field **67** holds the IP address assigned to the telephone, in this embodiment **192.168.0.20,** and the caller UDP port field **69** includes a UDP port identifier identifying a UDP port at which the audio path will be terminated at the caller's telephone.

Call Controller

Referring to Figure **4**, a call controller circuit of the call controller **14** (Figure **1**) is shown in greater detail at **100**. The call controller circuit **100** includes a microprocessor **102**, program memory **104**, random access memory **105** and an I/O interface **106**. The call controller circuit **100** may include a plurality of microprocessors, a plurality of program memories and a plurality of I/O interfaces to be able to handle a large volume of calls. However, for simplicity, the call controller circuit **100** will be described as having only one microprocessor, program memory and I/O interface, it being understood that there may be more.

Generally, the I/O interface **106** includes an input **108** for receiving messages, such as the SIP Invite message shown in Figure **3**, from the telephone **12** shown in Figure **2**. The I/O interface **106** also has an RC Request message output **110** for transmitting an RC Request message to the routing controller **16** in Figure **1**, an RC message input **112** for receiving routing messages from the RC **16**, a MR output **114** for transmitting messages to the media relay **17** (Figure **1**) to advise the media relay to establish an audio path, and a MR input **116** for receiving messages from the media relay to which a message has been sent to attempt to establish the audio path. The I/O interface **106** further includes a SIP output **118** for transmitting SIP messages to the

telephone **12** (Figure **2**) to advise the telephone of the IP address of the media relay **17** (Figure **1**) which will establish the audio path.

While certain inputs and outputs have been shown as separate, it will be appreciated that some may be associated with a single IP address and TCP or UDP port. For example, the messages sent and received from the RC **16** may be transmitted and received at the same single IP address and TCP or UDP port.

The program memory **104** of the call controller circuit **100** includes blocks of code for directing the microprocessor **102** to carry out various functions of the call controller **14**. For example, these blocks of code include a first block **120** for causing the call controller circuit **100** to execute a SIP Invite to RC request process to produce a RC Request message in response to a received SIP Invite message. In addition, there is a Routing Message to Media Relay message block **122** which causes the call controller circuit **100** to produce an MR Query message in response to a received routing message from the routing controller **16**.

Referring to Figure **5**, the SIP Invite-to-RC Request process is shown in more detail at **120**. On receipt of a SIP Invite message of the type shown in Figure **3**, block **132** of Figure **5** directs the call controller circuit **100** of Figure **4** to authenticate the user operating the telephone from which the SIP Invite message originated. This may be done, for example, by prompting the user for a password by sending a message back to the caller telephone **12** in Figure **1**, which is interpreted at the telephone as a request for password entry or the password may automatically be sent to the call controller **14** from the telephone, in response to the message. The call controller **14** may then make enquiries of the database **18** to determine whether or not the user's password matches a password stored in the database. Various functions may be used to pass encryption keys or hash codes back and forth to ensure the secure

transmission of passwords. Authentication may be bypassed when the call is to the ERC.

Should the authentication process fail, the call controller circuit **100** is directed to an error handling block **134** which causes messages to be displayed at the caller telephone **12** to indicate that there was an authentication error. If the authentication process is successful, block **131** directs the call controller circuit **100** of Figure **4** to determine whether or not the contents of the caller identifier field **60** of the SIP Invite message shown in Figure **3** is a validly formatted IP address. If it is a valid IP address, then block **133** of Figure **5** directs the call controller circuit **100** of Figure **4** to associate a type code with the call to indicate that the call type is a third party invite.

If at block **131** the caller identifier field **60** contents do not identify an IP address (for example, they may identify a PSTN number or Emergency Calling short number such as **911**), then block **135** directs the call controller circuit **100** to associate a type code with the call to indicate the call type is a regular invite. Then, block **136** directs the call controller circuit **100** to establish a call ID by reading the call ID provided in the call ID field **65** of the SIP Invite message from the telephone **12**, and at block **138** the call controller circuit is directed to produce a routing request message of the type shown in Figure **6** that includes that call ID. Block **139** of Figure **5** then directs the call controller circuit **100** of Figure **4** to send the RC Request message to the routing controller **16** of Figure **1**.

Referring to Figure **6**, a routing request message is shown generally at **150** and includes a caller identifier field **152**, a callee identifier field **154**, a digest field **156**, a call ID field **158** and a type field **160**. The caller, callee, digest, and call ID fields **152**, **154**, **156** and **158** contain copies of the caller, callee, digest parameters and call ID fields **60**, **62**, **64** and **65** of the SIP Invite message shown in Figure **3**. The type field **160** contains the type code established at blocks **133** or **135** of Figure **5** to indicate whether the call is from a third party

or system subscriber, respectively. For a normal non-emergency call the callee identifier field **154** may include a PSTN number or a system subscriber username as shown, for example. For an emergency call, the callee identifier field **154** includes the Emergency short number **911**, in this embodiment.

Routing Controller

Referring to Figure **7**, the routing controller **16** is shown in greater detail and includes an RC processor circuit shown generally at **200**. The RC processor circuit **200** includes a processor **202**, program memory **204**, a table memory **206**, a DID identifier buffer **203**, a caller ID buffer **205**, a callee ID buffer **209**, an emergency call flag **211**, a DID identifier buffer **203**, a and an I/O interface **208**, all in communication with the processor. (As earlier indicated, there may be a plurality of processors (**202**), memories (**204**), etc.) Separate caller ID buffers **205**, callee id buffers **209** and emergency call flags **211** are instantiated for each call and are associated with respective call IDs.

The I/O interface **208** includes a database output port **210** through which a request to the database **18** (Figure **1**) can be made and includes a database response port **212** for receiving a reply from the database. The I/O interface **208** further includes an RC Request message input **214** for receiving the routing request message from the call controller **14**. Thus, the routing controller receives a routing request message including a caller identifier and a callee identifier. The I/O interface **208** further includes a routing message output **216** for sending a routing message back to the call controller **14**.

The program memory **204** includes blocks of codes for directing the RC processor circuit **200** to carry out various functions of the routing controller **16**. One of these blocks includes an RC Request message handler process **250** which directs the RC processor circuit to produce a routing message in response to a received routing request message of the type shown at **150** in Figure **6**. The RC Request message handler process is shown in greater detail at **250** in Figures **8**A through **8**D.

RC Request Message Handler

Referring to Figure **8**A, the routing request message handler **250** begins with a first block **252** that directs the RC processor circuit **200** (Figure **7**) to store the contents of the RC Request message **150** (Figure **6**) in the callee ID buffer **209** and the caller buffer **205** buffers for separately storing the contents of the callee field (**154** in Figure **6**) and the caller field (**152** in Figure **6**) respectively of the RC Request message. Block **254** then directs the RC processor circuit **200** to use the contents of the caller field (**152** in Figure **6**)in the RC Request message **150**, to search the database **18** shown in Figure **1** and retrieve a dialling profile associated with the caller.

Referring to Figure **9**, a dialling profile is shown generally at **256** and includes system fields including a username field **258**, a domain field **260**, a national dialling digits (NDD) field **262**, an International dialing digits (IDD) field **264**, a country code field **266**, a local area codes field **267**, a caller minimum local length field **268**, a caller maximum local length field **270**, a reseller field **273**, a user address field **275**, an emergency call identifier field **277** and an emergency response center (ERC) field **279**.

An exemplary dialling profile for the Vancouver subscriber is shown generally at **276** in Figure **10** and indicates that the username field **258** includes the username **2001 1050 8667** which is the same as the contents of the username field **50** in the Vancouver telephone **12** shown in Figure **2**.

Referring back to Figure **10**, the domain field **260** includes a domain name as shown at **282**, including a supernode type identifier **284**, a location code identifier **286**, a system provider identifier **288** and a top level domain identifier **290**, identifying a domain or supernode associated with the user identified by the contents of the username field **258**.

In this embodiment, the supernode type identifier **284** includes the code "sp" identifying a supernode and the location code identifier **286** identifies the supernode as being in Vancouver (yvr). The system provider identifier **288** identifies the company supplying the service and the top level domain identifier **290** identifies the "com" domain.

The NDD field **262** in this embodiment includes the digit "**1**" and in general includes a digit specified by the International Telecommunications Union – Telecommunications Standardization Sector (ITU-T) E.**164** Recommendation which assigns national dialling digits to certain countries.

The IDD field **264** includes the code **011** and, in general, includes a code assigned by the ITU-T according to the country or geographical location of the subscriber.

The country code field **266** includes the digit "**1**" and, in general, includes a number assigned by the ITU-T to represent the country in which the subscriber is located.

The local area codes field **267** includes the numbers **604** and **778** and generally includes a list of area codes that have been assigned by the ITU-T to the geographical area in which the subscriber is located. The caller minimum and maximum local number length fields **268** and **270** each hold the number **10** representing minimum and maximum local number lengths permitted in the area code(s) specified by the contents of the local area codes field **267**. The reseller field **273** holds a code identifying a retailer of the telephone services, and in the embodiment shown, the retailer is "Klondike".

The address field **275** holds an address at which the subscriber telephone is normally located. The emergency short number field **277** holds the short emergency number such as "**911**" that the user is expected to dial in the event of an emergency. The ERC number field **279** holds a full PSTN number

associated with an emergency response center that would desireably be geographically nearest to the address specified in the address field **275**.

A dialling profile of the type shown at **256** in Figure **9** is produced whenever a user registers with the system or agrees to become a subscriber to the system. An ERC may register as a user, but need not do so since, as will be appreciated below, provisions are made for making VoIP to PSTN calls which may include calls to an ERC only available via the PSTN. Of importance here is that the contents of the emergency short number field **277** and the contents of the ERC number field **279** are assigned when the user registers with the system and thus it may be said that these numbers are "pre-assigned" to the user before the user makes any calls.

A user wishing to subscribe to the system may contact an office maintained by a system operator. Personnel in the office may ask the user certain questions about his location and service preferences, whereupon tables can be used to provide office personnel with appropriate information to be entered into the username, domain, NDD, IDD, country code, local area codes and caller minimum and maximum local length fields, emergency short number field and ERC number field **258**, **260**, **262**, **264**, **266**, **267**, **268**, **270**, **277**, **279** to establish a dialling profile for the user.

Referring to Figures **10**A, **11**, and **12**, dialling profiles for the ERC subscriber, Calgary subscriber, and the London subscriber, respectively for example, are shown.

In addition to creating dialling profiles when a user registers with the system, a direct-in-dial (DID) record of the type shown at **268** in Figure **13** may optionally be added to a direct-in-dial table in the database **18** to associate the username and a host name of the supernode, with which the user is associated, with an E.**164** number on the PSTN network. If the user does not

have such an E.**164** number, no DID record need be created at this time for that user.

In this embodiment, the DID bank table records include a username field **291**, a user domain field **272** and DID identifier field **274**, for holding the username, hostname of the supernode and E.**164** number respectively. Thus a DID bank table record pre-associates a DID identifier with a user (e.g. caller).

A DID bank table record may also include a creation time field and an expiration time field for use when the DID bank table record is a temporary record as will be explained below.

DID bank table records for the Vancouver, Calgary and London subscribers are shown in Figures **13**A, **13**B, and **14**, respectively

In addition to creating dialling profiles and DID records when a user registers with the system, call blocking records of the type shown in Figure **26**, call forwarding records of the type shown in Figure **28** and voicemail records of the type shown in Figure **30** may be added to the database **18** when a new subscriber is added to the system.

Referring back to Figure **8**A, after being directed at block **254** to retrieve a dialling profile associated with the caller, such as shown at **276** in Figure **10**, the RC processor circuit (**200**) is directed to block **255** which causes it to determine whether the contents of the callee ID buffer **209** shown in Figure **7** are equal to the contents of the emergency call identifier field **277** of the dialling profile **276** for the caller, shown in Figure **10**. If the contents of the callee ID buffer **209** are not equal to the contents of the emergency short number field **277**, the call is deemed not to be an emergency call and the RC processor circuit **200** is directed to location A in Figure **8**B to carry out further processing on the basis that the call is to be a normal, non-emergency call.

If the contents of the callee ID buffer **209** match the contents of the emergency call identifier field (**277** in Figure **10**), the call is deemed to be an emergency call and block **157** directs the RC processor circuit **200** to set a time to live (TTL) value to a high number such as **9999** to indicate that the call may have a long duration of **9999** seconds, for example. In addition block **157** directs the RC processor circuit **200** to set active the emergency call flag **211** in Figure **7**, to indicate that the call is an emergency call. Then, block **159** directs the RC processor circuit **200** to replace the contents of the callee ID buffer **209** with the contents of the ERC # field **279** of the caller dialling profile **276** (Figure **10**). Thus, the RC processor circuit produces an emergency response center identifier in response to the emergency call identifier by copying the emergency response center identifier from the ERC field **279** of the dialing profile **276** (Figure **10**) associated with the caller to the callee ID buffer **209** shown in Figure **7** so that effectively, the contents of the callee ID buffer are replaced with the Emergency Response Center number. The RC processor circuit **200** is then directed to location A in Figure **8**B.

In this embodiment, for regular and emergency call processing, beginning at location A in Figure **8**B, the RC processor circuit **200** is directed to perform certain checks on the callee identifier provided by the contents of the callee identifier buffer **209** shown in Figure **7**. Most of these checks are shown in greater detail in Figure **8**B and are used for regular non-emergency call handling. Emergency calls in which the ERC number has been substituted for the short emergency calling number (i.e., **911**) will pass all of the checks. Subjecting both emergency and non-emergency calls to these checks enables all calls, whether emergency or non-emergency, to be passed through the same process and, simplifies the introduction of emergency call handling processes into regular call processing routines depicted in Figures **8**A to **8**D. Alternatively, the RC processor circuit may be directed directly from block **159** to block **269** in Figure **8**B whenever the emergency call flag is set, as shown in broken outline in Figure **8**B.

Figure 8B

IDD Testing

Referring to Figure **8B**, to start the first of the checks, the RC processor circuit **200** is directed to a first block **257** that causes it to determine whether a digit pattern of the callee identifier provided in the callee ID buffer **209** includes a pattern that matches the contents of the IDD field **264** in the caller dialling profile **276** shown in Figure **10**. If so, then block **259** directs the RC processor circuit **200** to set a call type identifier code (not shown) to indicate that the call is a long distance call, e.g., from the Vancouver subscriber to the London subscriber, and block **261** directs the RC processor circuit **200** to produce a reformatted callee identifier by reformatting the current callee identifier into a predetermined target format. In this embodiment, this is done by removing the pattern of digits matching the IDD field contents **264** of the caller dialling profile **276** to effectively shorten the number. Then, block **263** directs the RC processor circuit **200** to determine whether or not the reformatted callee identifier meets criteria establishing it as an E.**164** compliant number and if the length does not meet this criteria, block **265** directs the RC processor circuit **200** to send back to the call controller **14** a message indicating that the length of the call identifier is not correct. The process **250** is then ended. At the call controller **14**, routines may respond to the incorrect length message by transmitting a message back to the telephone **12** to indicate that an invalid number has been dialed, for example. Thus at the conclusion of block **263** a callee identifier having a pre-defined format should be available.

NDD Testing

Referring back to Figure **8B**, if at block **257**, the callee identifier specified by the contents of the callee buffer **209** Figure **7** does not begin with an IDD, block **381** directs the RC processor circuit **200** to determine whether or not the callee identifier begins with the same NDD code as assigned to the caller. To do this, the RC processor circuit is directed to refer to the caller dialling profile **276** shown in Figure **10**. In the embodiment shown, the NDD code stored in

an NDD field **262** is the digit **1**. Thus, if the callee identifier begins with the digit **1**, the RC processor circuit **200** is directed to block **382** in Figure **8**B.

Block **382** directs the RC processor circuit **200** to examine the callee identifier to determine whether or not digits following the NDD code identify an area code that is the same as any of the area codes identified in the local area codes field **267** of the caller dialling profile **276** shown in Figure **10**. If not, block **384** directs the RC processor circuit **200** to set a call type variable (not shown) to a code indicating the call is a national call. If the digits identify an area code that is the same as a local area code associated with the caller, block **386** directs the RC processor circuit **200** to set the call type variable to indicate that the call type is as a local call, national style. After executing blocks **384** or **386**, block **388** directs the RC processor circuit **200** to reformat the callee identifier by removing the national dial digit and prepending a caller country code identified by the country code field **266** of the caller dialling profile **276** shown in Figure **10**. The RC processor circuit **200** is then directed to block **263** to perform the processes described above beginning at block **263**. Again, at the conclusion of block **263** a callee identifier having a pre-defined format should be available.

Area Code Testing

If at block **381** the callee identifier does not begin with an NDD code, block **390** directs the RC processor circuit **200** to determine whether the callee identifier in the callee ID buffer **209** begins with digits that identify the same area code as the caller. Again, the reference for this is the caller profile **276** shown in Figure **10** and the RC processor circuit **200** determines whether or not the first few digits in the callee identifier identify an area code identified by the local area code field **267** of the caller profile **276**. If so, then block **392** directs the RC processor circuit **200** to set the call type to a code indicating the call is a local call and block **394** directs the RC processor circuit **200** to prepend the caller country code to the callee identifier, the caller country code being determined from the country code field **266** in the caller profile **276**. The

RC processor circuit **200** is then directed to block **263** for processing as described above beginning at block **263**. Emergency calls are likely to follow this path since the Emergency Response Center number that supplants the short emergency number (**911**) will normally be formatted to include an area code, but no IDD or NDD. Again at the conclusion of block **263** a callee identifier having a pre-defined length should be available.

Callee ID Length Testing

If at block **390**, the callee identifier does not have the same area code as the caller, as may be the case with non-emergency calls, block **396** directs the RC processor circuit **200** to determine whether the callee identifier in the callee ID buffer **209** has the same number of digits as the number of digits indicated in either the caller minimum local number length field **268** or the caller maximum local number length field **270** of the caller profile **276** shown in Figure **10**. If so, then block **398** directs the RC processor circuit **200** to set the call type to local and block **400** directs the processor to prepend to the callee identifier the caller country code as indicated by the country code field **266** of the caller profile **276** followed by the caller area code as indicated by the local area code field **267** of the caller profile shown in Figure **10**. The RC processor circuit **200** is then directed to block **263** for further processing as described above beginning at block **263**. Again at the conclusion of block **263** a callee identifier having a pre-defined length should be available.

Valid Subscriber Testing

If at block **396**, the callee identifier in the callee ID buffer **209** has a length that does not match the length specified by the contents of the caller minimum local number length field **268** or the caller maximum local number length field **270** of the caller profile **276**, block **402** directs the RC processor circuit **200** to determine whether or not the callee identifier identifies a valid username. To do this, the RC processor circuit **200** searches through the database **18** of dialling profiles to find a dialling profile having a username field **258** that matches the callee identifier. If no match is found, block **404** directs the RC

processor circuit **200** to send an error message back to the call controller (**14**). If at block **402**, a dialling profile having a username field **258** that matches the callee identifier is found, block **406** directs the RC processor circuit **200** to set the call type to a code indicating the call is a network call and the processor is directed to block **275** of Figure **8**A, to continue executing the RC message handler process **250**.

From Figure **8**B, it will be appreciated that there are certain groups of blocks of codes that direct the RC processor circuit **200** to determine whether the callee identifier in the callee ID buffer **209** has certain features such as an IDD code, a NDD code, an area code and a length that meet certain criteria and to reformat the callee identifier, as necessary, into a predetermined target format including only a country code, area code, and a normal telephone number, for example, to cause the callee identifier to be compatible with the E.**164** standard, in this embodiment. This enables the RC processor circuit **200** to have a consistent format of callee identifiers for use at block **269** in searching through the DID bank table records of the type **268** shown in Figure **13** to determine how to route calls for subscriber to subscriber calls on the same system. Recall that the ERC may be a subscriber.

Still referring to Figure **8**B, if the length of the reformatted callee identifier meets the length criteria set forth at block **263**, block **269** directs the RC processor circuit **200** to determine whether or not the reformatted callee identifier is associated with a direct-in-dial bank (DID) record of the type shown at **268** in Figure **13**.

Exemplary DID records for the Vancouver, Calgary and London subscribers are shown in Figures **13**A, **13**B and **14**. The username field **291** and user domain field **272** are as specified in the username and user domain fields **258** and **260** of the corresponding dialing profiles shown in Figures **10**, **11** and **12** respectively. Referring to Figure **13**A the contents of the DID field **274** include an E.**164** telephone number including a country code **293**, an area code **295**,

T-Mobile Exhibit 1002
Part 3 of 6

an exchange code **297** and a number **299**. If the user has multiple telephone numbers, then multiple records of the type shown at **276** would be included in the DID bank table in the database **18**, each having the same username and user domain, but different DID field **274** contents reflecting the different E.**164** telephone numbers associated with that user.

Referring back to Figure **8**B, at block **269**, if the RC processor circuit **200** determines that the current, (e.g.reformatted callee identifier produced at block **261**) can be found in a record in the DID bank table, then the callee is a subscriber to the system and block **279** directs the RC processor circuit **200** to copy the contents of the corresponding username field **291** from the DID bank table record into the callee ID buffer **209** shown in Figure **7**. Thus, the RC processor circuit **200** locates a subscriber username associated with the reformatted callee identifier. If the call is being made to the Emergency Response Center and the Emergency Response Center (ERC) is a subscriber to the system, a DID record would be found in the DID bank table, otherwise a DID record for the ERC would not be found. Assuming the Emergency Response Center is a subscriber to the system, the RC processor circuit **200** is directed to block **275** at point B in Figure **8**A for further processing now that it is known that the call is essentially a subscriber to subscriber call.

Subscriber to Subscriber Calls Between Different Nodes

Referring back to Figure **8**A, block **275** directs the RC processor circuit **200** to determine whether or not the username stored in the callee ID buffer **209** (in Figure **7**) is associated with the same supernode as the caller. To do this, the RC processor circuit **200** determines whether or not the prefix (i.e. the leftmost digit) of the username stored in the callee ID buffer **209** is the same as the prefix of the username of the caller specified by the caller identifier field **152** of the RC. Request message **150** shown in Figure **6**. If they are not the same, block **277** of Figure **8**A directs the RC processor circuit (**200**) to set a call type flag (not shown) to indicate that the call is a cross-domain call. Then, block **281** directs the RC processor circuit (**200**) to determine whether

the emergency call flag **211** shown in Figure **7** has been set and if so, block **283** of Figure **8**A directs the RC processor to determine whether the caller identifier is associated with a pre-associated direct inward dialing (DID) identifier. This is done by searching the DID bank table to attempt to locate a DID record having DID field **(274)** contents matching the contents of the caller identifier stored in the caller ID buffer **(205)**. If such a DID record is found, the processor circuit **200** has effectively determined that the caller has a pre-associated DID identifier.

If no such DID record is found, the RC processor circuit **200** has effectively determined that the caller has no pre-associated DID identifier. In this case, block **285** then directs the RC processor circuit **200** to produce a DID identifier for the caller by associating a temporary DID identifier with the caller identifier by associating with the caller identifier a DID identifier from a pool of pre-determined DID identifiers. This is done by creating and associating with the caller a temporary DID record of the type shown in Figure **13**. The temporary DID record has a DID identifier field **274** populated with the DID identifier from the pool. The DID identifier from the pool may be **1 604 867 5309**, for example. The pool may be provided by causing the RC processor circuit **200** to maintain a list of pre-defined DID identifiers and pointers identifying a current read point in the list and a current write point in the list. The current read pointer may be incremented each time the pool is addressed to obtain a temporary DID identifier.

A temporary DID record may be canceled after a pre-defined period of time. For example, the temporary DID identifier records are desirably as shown in Figure **13** and may further include a creation time field and an expiry time field for holding a creation time value and an expiry time value respectively. The expiry time may be **2** hours after the creation time, for example, such that the temporary DID record is deleted two hours after it is created. A separate process, not shown, may continuously or periodically scan the DID records to determine whether any DID records have expiry times that have been

exceeded and if so, cause such temporary DID records to be cancelled or deleted. Thus, the RC processor produces a direct inward dialing identifier for the caller by associating a temporary DID identifier with the caller identifier when the emergency call flag is active and it is determined that the caller has no pre-associated DID identifier, or by associating a DID identifier pre-assigned to the caller identifier.

After a temporary DID record has been created and stored in the DID bank table in the database **18** shown in Figure **1**, or if the caller already had a DID record, block **287** of Figure **8**A directs the RC processor circuit to load the DID identifier buffer **203** with the contents of the field of DID temporary or pre-associated DID record. Then the RC processor circuit loads a routing message buffer with the contents of the DID identifier buffer **203** acting as the caller identifier and the contents of the callee ID buffer **209** as the callee identifier. This will provide for a PSTN call back number to be provided to the emergency response center.

Thus, where the caller identifier has no pre-assigned DID identifier, the RC processor produces a routing message including the emergency response center identifier and the temporary DID identifier for receipt by the routing controller to cause the routing controller to establish a route between the caller and the emergency response center.

Referring to Figure **15**, a routing message buffer is shown generally at **352** and includes a supplier prefix field **354**, a delimiter field **356**, a callee field **358**, at least one route field **360**, a time-to-live (TTL) field **362** and a caller ID field **364**. The supplier prefix field **354** holds a code for identifying supplier traffic. The delimiter field **356** holds a symbol that delimits the supplier prefix code from the callee field **358** and in this embodiment, the symbol is a number sign (#) as illustrated in Figure **25**. Referring back to Figure **15**, the callee field **358** holds a copy of the contents of the callee ID buffer **209** of Figure **7**. The route field **360** holds a domain name or an IP address of a gateway or supernode

that is to carry the call and the TTL field **362** holds a value representing the number of seconds the call is permitted to be active, based on subscriber available minutes and other billing parameters, for example.

Desirably, the time to live field holds a number indicating a maximum call time for the call and where the call is an emergency call, desirably the maximum call time exceeds a duration of an average non-emergency telephone call. The caller ID field **364** holds a caller identifier which in this case, is the temporary or pre-associated DID number from the DID record associated with the caller.

Referring to Figure 8A and Figure **16**, a routing message produced by the RC processor circuit **200** at block **287** is shown generally at **366** and includes only the callee field **358**, route field **360**, TTL field **362** and caller ID field **364**.

The callee field **358** holds the full username of the callee, and where the call is an emergency call as shown, the full username of the callee is the username of the emergency response center. The route field **360** contains the identification of the domain with which the emergency response center is associated, i.e., sp.yvr.digifonica.com. The TTL field holds the value **9999** set at block **157** in Figure **8**A and the caller ID field **364** holds the DID identifier associated with the caller. Block **380** then directs the RC processor circuit to send the routing message shown in Figure **16** to the call controller **14** (Figure **1**).

Referring to Figure **8**A, if at block **281**, the emergency call flag is not set, the call is not an emergency call, and the RC processor is directed to block **350** which causes it to direct the RC processor circuit **200** to load the routing message buffer with information identifying the supernode in the system with which the callee is associated and to set a time to live for the call to a high value such as **9999**. The supernode, with which the callee is associated, is determined by using the callee username stored in the callee ID buffer **209** to

address a supernode table having records of the type as shown at **370** in Figure **17**.

Referring to Figure **17**, each prefix to a supernode table record **370** has a prefix field **372** and a supernode address field **374**. The prefix field **372** includes the first n digits of the callee identifier. In this case n=**1**. The supernode address field **374** holds a code representing the IP address or a fully qualified domain name (FQDN) of the supernode associated with the code stored in the prefix field **372**. Referring to Figure **18**, for example, if the prefix is **4**, the supernode address associated with that prefix is sp.lhr.digifonica.com, identifying the London supernode (**21** in Figure **1**), for example. After the routing message buffer has been loaded with identification of the supernode, block **380** of Figure **8**A directs the RC processor circuit to send the routing message shown in Figure **16**A to the call controller **14** (Figure **1**).

Subscriber to Subscriber Calls Within the Same Node

Referring back to Figure **8**A, if at block **275**, the callee identifier stored in the callee ID buffer **209** (Figure **7**) has a prefix that identifies the same supernode as that associated with the caller, block **559** directs the RC processor circuit **200** to determine whether or not the emergency call flag **211** of Figure **7** has been set. If at block **559**, the RC processor circuit **200** determines that the emergency call flag **211** is set, the RC processor circuit **200** is directed to resume processing at block **283** to scan the DID bank table to determine whether the caller has a DID record and to assign a temporary DID number if necessary, as described above and then to send a routing message of the type shown in Figure **16** to the call controller.

If at block **559** the emergency call flag has not been set, regular non-emergency call processing ensues beginning with block **600** which directs the RC processor circuit **200** to use the callee identifier to locate and retrieve a dialling profile for the callee identified by the callee identifier stored in the

callee ID buffer **209**. The dialling profile is of the type shown in Figure **9**, and may contain data as shown in Figure **11**, for example. In this case the same-node subscriber is the Calgary subscriber. Block **602** of Figure **8**A directs the RC processor circuit **200** to get call block, call forward and voicemail tables from the database **18** based on the username identified in the callee dialing profile retrieved by the RC processor circuit at block **600**. Call block, call forward and voicemail tables have records as shown in Figures **26**, **28** and **30** for example.

Referring to Figure **26**, the call block records include a username field **604** and a block pattern field **606**. The username field **604** holds a username matching the username in the username field **258** of the dialing profile (Figure **9**) associated with the callee, and the block pattern field **606** holds one or more E.**164**-compatible numbers or usernames identifying PSTN telephone numbers or system subscribers from whom the subscriber identified by the contents of the username field **604** does not wish to receive calls.

Referring back to Figure **8**A and referring to Figure **27**, block **608** directs the RC processor circuit **200** to determine whether or not the caller identifier matches a block pattern stored in the block pattern field **606** of the call block record associated with the callee identified by the contents of the username field **604** in Figure **26**. If the caller identifier matches a block pattern stored in the field **606**, block **610** directs the RC processor circuit **200** to send a drop call or non-completion message to the call controller **14** and the process **250** is ended. If the caller identifier does not match a block pattern associated with the callee, block **612** directs the RC processor circuit **200** to determine whether or not call forwarding is required.

Referring to Figure **28**, records in the call forwarding table include a username field **614**, a destination number field **616** and a sequence number field **618**. The username field **614** stores a code representing a username of a subscriber with whom the call forwarding record is associated. The destination

number field **616** holds a username or E.**164** number representing a number to which the current call should be forwarded, and the sequence number field **618** holds an integer number indicating the order in which the username associated with the corresponding destination number field should be attempted for call forwarding. The call forwarding table may have a plurality of records for a given subscriber. The RC processor circuit **200** uses the contents of the sequence number field **618** to place the records for a given subscriber in order. As will be appreciated below, this enables the call forwarding numbers to be tried in an ordered sequence.

Referring back to Figure **8**A and referring to Figure **28**, if at block **612**, the call forwarding record for the callee identified by the callee identifier contains no contents in the destination number field **616** and accordingly no contents in the sequence number field **618**, then there are no call forwarding entries and the RC processor circuit **200** is directed to load the routing message buffer shown in Figure **32** with the callee username, domain and time to live as shown at **650**. The RC processor circuit **200** is then directed to block **620** in Figure **8**C. However, if there are contents in the call forwarding record as shown in Figure **29**, block **622** shown in Figure **8**A directs the RC processor circuit **200** to search the dialling profile table in the database **18** to find a dialling profile record of the type shown in Figure **9**, for the callee identified in the destination number field **616** of the first call forwarding record and to store the contents in the routing message buffer. The RC processor circuit **200** is then directed to load the contents of the domain field **260** associated with the dialing profile specified by the contents of the destination number field **616** of the first call forwarding record into the routing message buffer as shown at **652** in Figure **32**. This process is repeated for each call forwarding record associated with the callee identified by the callee identifier to add to the routing message buffer all call forwarding usernames and domains associated with the callee.

Referring to Figure **8**C, at block **620** the RC processor circuit **200** is directed to determine whether or not the user identified by the callee identifier has paid for voicemail service and this is done by checking to see whether or not a flag **630** is set in a voicemail record of the type shown in Figure **30** in a voicemail table stored in the database **18** in Figure **1**.

Referring to Figure **30**, voicemail table records include a username field **624**, a *voicemail server* field **626**, *a seconds-to-voicemail field* **628** and an enabled field **630**. The username field **624** stores the username of the subscriber who purchased the service. The voicemail server field **626** holds a code identifying an IP address or a fully qualified domain name (FQDN) of a voicemail server associated with the subscriber identified by the username field **624**. The seconds-to-voicemail field **628** holds a code identifying the time to wait before engaging voicemail and the enable field **630** holds a code representing whether or not voicemail is enabled for the user identified by the contents of the username field **624**. Therefore, referring back to Figure **8**C, at block **620** the RC processor circuit **200** finds a voicemail record as shown in Figure **31** having username field **624** contents matching the callee identifier and examines the contents of the enabled field **630** to determine whether or not voicemail is enabled. If voicemail is enabled, then block **640** in Figure **8**C directs the RC processor circuit **200** to store the contents of the voicemail server field **626** of Figure **31**, and the contents of the seconds to voicemail field **628** of Figure **31** in the routing message buffer as shown at **654** in Figure **32**.

Referring back to Figure **8**C, block **642** then directs the processor to get time to live (TTL) values for each route specified by the routing message according to any of a plurality of criteria such as, for example, the cost of routing and the user's account balance. These TTL values are then appended to corresponding routes already stored in the routing message buffer. Block **643** then directs the RC processor circuit **200** to store the TTL value determined at

block **642** in the routing message buffer. In the routing message shown in Figure **32**, the time to live value is set at **60** seconds, for example.

Block **644** of Figure **8**C then directs the RC processor circuit **200** to store the IP address or FQDM of the current supernode in the routing message buffer as shown at **656** in Figure **32**. An exemplary routing message for a subscriber to subscriber call on the same node is thus shown in the routing message buffer shown in Figure **32**.

Subscriber to Non-Subscriber Calls

Not all calls will be subscriber-to-subscriber calls and this will be detected by the RC processor circuit **200** when it executes block **269** of Figure **8**B and does not find a DID bank table record associated with the callee in the DID bank table. This may be the case, for example, where the Emergency Response Center (ERC) is not a subscriber to the system. When this occurs, the RC processor circuit **200** is directed to block **408** in Figure **8B** which causes it to set the contents of the callee identifier buffer **209** equal to the reformatted callee identifier, i.e., the E.**164** compatible number produced prior to block **263** in Figure **8B**. Block **409** then directs the RC processor circuit **200** to determine whether the emergency call flag **211** in Figure **7** has been set. If the emergency call flag is set, block **411** in Figure **8**D directs the RC processor to search the DID bank table to attempt to locate a DID record having DID field (**274**, Figure **13**) contents matching the contents of the caller identifier stored in the caller ID buffer (**205** in Figure **7**).

If no such DID record is found, the RC processor circuit **200** has effectively determined that the caller identifier is not associated with a pre-associated DID identifier. In this case, block **413** then directs the RC processor circuit **200** to associate a temporary DID identifier with the caller identifier by associating with the caller identifier a DID identifier from the pool of pre-determined DID identifiers. Again, this is done by creating and associating with the caller a temporary DID record of the type shown in Figure **13**.

After a temporary DID record has been created or if the caller already has a DID record, block **415** directs the RC processor circuit to store the DID number (**274** in Figure **13**) in the caller ID buffer **209** in Figure **7**.

After having loaded the caller ID buffer **209** with the temporary or pre-associated DID number, or after having determined that the emergency call flag is not set, block **410** (Figure **8**B) directs the RC processor circuit **200** to initiate a process for identifying gateways to the PSTN through which the call will be established. This process begins with block **410** which directs the RC processor circuit **200** to address a master list having records of the type shown in Figure **19**.

Each master list record includes a master list ID field **500**, a dialling code field **502**, a country code field **504**, a national sign number field **506**, a minimum length field **508**, a maximum length field **510**, a NDD field **512**, an IDD field **514** and a buffer rate field **516**.

The master list ID field **500** holds a unique code such as **1019**, for example, identifying the record. The dialling code field **502** holds a predetermined number pattern that the RC processor circuit **200** uses at block **410** in Figure **8**B to find the master list record having a dialling code matching the first few digits of the reformatted callee identifier. The country code field **504** holds a number representing the country code associated with the record and the national sign number field **506** holds a number representing the area code associated with the record. (It will be observed that the dialling code **502** is a combination of the contents of the country code field **504** and the national sign number field **506**.) The minimum length field **508** holds a number representing the minimum number of digits that can be associated with the record and the maximum length field **510** holds a number representing the maximum number of digits in a number with which the record may be compared. The NDD field **512** holds a number representing an access code used to make a call within

the country specified by the country code **504** and IDD field **514** holds a number representing the international prefix needed to dial a call from the country indicated by the country code.

Thus, for example, a master list record may have a format as shown in Figure **20** with exemplary field contents as shown.

Referring back to Figure **8**B, using the country code and area code portions of the reformatted callee identifier that has been formatted for compatibility with the E.**164** standard, block **410** directs the RC processor circuit **200** to find a master list record such as the one shown in Figure **20** having a dialling code that matches the country code and area code of the reformatted callee identifier held in the callee identifier buffer **209**. Thus, in this example, the RC processor circuit **200** might find a master list record having an ID field with the number **1019**. This number may be also referred to as a route ID number. Thus, a route ID number is found in the master list record associated with a predetermined number pattern in the reformatted callee identifier.

After execution of block **410** in Figure 8B, the process **250** continues as shown in Figure **8**D. Referring to Figure **8**D, block **412** directs the RC processor circuit **200** to use the route ID number determined at block **410** to locate at least one supplier record identifying a supplier operable to supply a communications link for this route. To do this, block **412** directs the RC processor circuit **200** to search a supplier ID table having records of the type shown in Figure **21**.

Referring to Figure **21**, supplier list records include a supplier ID field **540**, a master list ID field **542**, an optional prefix field **544**, a route identifier field **546**, a NDD/IDD rewrite field **548** and a rate field **550**. The supplier ID field **540** holds a code identifying the name of the supplier and the master list ID field **542** holds a code for associating the supplier record with the master list record. The prefix field **544** optionally holds a string used to identify the

supplier traffic and the route identifier field **546** holds an IP address of a gateway operated by the supplier indicated by the supplier ID field **540**. The NDD/IDD rewrite field **548** holds a code and the rate field **550** holds a code indicating the cost per second to the system operator to use the route through the gateway specified by the contents of the route identifier field **546**. Exemplary supplier records are shown in Figures **22**, **23** and **24** for Telus, Shaw and Sprint, respectively, for example.

Referring back to Figure **8**D, at block **412** the RC processor circuit **200** finds all supplier records that contain the master list ID found at block **410** of Figure **8**B.

Block **560** of Figure **8**D directs the RC processor circuit **200** to begin to produce routing messages. To do this, the RC processor circuit **200** loads a routing message buffer as shown in Figure **25** with a supplier prefix of the least costly supplier where the least costly supplier is determined from the rate fields **550** of the records associated with respective suppliers.

Referring to Figures **22**-**24**, in the embodiment shown, the supplier "Telus" has the lowest number in the rate field **550** and therefore the prefix **4973** associated with that supplier is loaded into the routing message buffer shown in Figure **25** first. At block **562**, the prefix **4973** is then delimited by the number sign (as defined by the contents of the delimiter field **356** in the routing message format **352** in Figure **15**) and the reformatted callee identifier is next loaded into the routing message buffer after the delimiter . Then, the contents of the route identifier field **546** of the record associated with the supplier Telus are added to the message after an @ sign delimiter and then block **564** in Figure **8**D directs the RC processor circuit **200** to get a TTL value (algorithm not shown), which in this embodiment may be **3600** seconds, for example. Block **566** of Figure **8**D then directs the RC processor circuit **200** to append this TTL value to the contents already in the routing message buffer shown in Figure **25**. Block **567** of Figure **8**D then directs the processor circuit to append

the contents of the caller ID buffer **205** of Figure **7** to the contents already in the routing message buffer shown in Figure **25**. Accordingly, the first part of the routing message is shown generally at **570** in Figure **25**.

Referring back to Figure 8D, block **571** directs the RC processor circuit **200** back to block **560** and causes it to repeat blocks **560**, **562**, **564**, **566** and **567** for each successive supplier until the routing message buffer is loaded with information pertaining to each supplier. Thus, the second portion of the routing message is shown at **572** in Figure **25** and this second portion relates to the second supplier identified by the record shown in Figure **23** and referring back to Figure **25**, the third portion of the routing message is shown at **574** which is associated with a third supplier as indicated by the supplier record shown in Figure **24**. Consequently, referring to Figure **25**, the routing message buffer holds a routing message identifying a plurality of different suppliers able to provide gateways to establish a communication link to permit the caller to contact the callee. Each of the suppliers is identified, in succession, according to rate contained in the rate field **550** of the supplier list record shown in Figure **21**, in this embodiment. Other criteria for determining the order in which suppliers are listed in the routing message may include preferred supplier priorities which may be established based on service agreements, for example.

Response to Routing Message

Referring back to Figure **1**, the routing message of the type shown in Figures **16**, **16**A, **25** or **32**, is received at the call controller **14**. It will be recalled that the call controller **14** already has the original SIP invite message shown in Figure **3**. Referring to Figure **4**, the program memory **104** of the call controller **14** includes a routing-to-media relay routine depicted generally at **122**.

Referring to Figure **38**, the routing to media relay routine **122** directs the processor to participate in a process for establishing audio paths. Assume the call is directed to the ERC.

As a first step in the process for establishing audio paths, a message **1100** is sent from the call controller **14** to the media relay **17**, the message including the call ID, the caller telephone IP address and UDP port as determined from the caller IP address field **67** and caller UDP port field **69** in the SIP Invite message **59** shown in broken outline.

In response, the media relay (MR) **17** sends a confirmation message **1102** back to the call controller **14**, the message including a media relay IP address (**192.168.2.10**) and UDP port number (**22123**) defining a callee socket that the media relay will use to establish an audio path to the ERC telephone or a PSTN gateway to the ERC, where the Emergency Response Center is only available through the PSTN

The call controller **14** then sends a SIP Invite message **1104** of type shown in Figure **3** to the callee telephone **15** (or PSTN gateway), to advise the calllee that telephone of the socket the media relay expects to use for audio communication with the caller telephone. The SIP invite message includes the caller and callee identifiers (**60** and **62**), the call ID (**65**) and the media relay **17** IP address (**192.168.2.10**) and the media relay UDP port number (**22123**) assigned to the callee socket as received from the confirmation message **1102**. The caller identifier may be that which was associated with the caller at blocks **413** in Figure **8**D or block **285** in Figure **8**A, for example, or may be the DID associated with the caller as determined from a DID record already associated with the caller. Such caller identifier, as obtained from the routing message, may be used as calling line identification (CLID) information and may be caused to appear on a display of the callee telephone, which is particularly advantageous where the callee telephone is one at an ERC. Such CLID information provides an ERC operator with callback information, enabling the operator to call back the caller who made the emergency call. Since the temporarily assigned DID records persist for some time after the emergency call has taken place, the ERC operator can call back the person

who made the emergency call during a period of time after the emergency call is terminated. In this embodiment, assume the callee telephone identifies its socket as IP address **192.168.3.10** and UDP port **33123**.

The callee (ERC) telephone **33** of Figure **1** (or PSTN gateway) stores the media relay **17** IP address (**192.168.2.10**) and assigned UDP port number (**22123**) and configures itself to create a socket for an audio path between the media relay. Referring to Figures **1** and **38** the callee telephone **15** (or PSTN gateway) then sends a SIP OK message **1106** back to the call controller **14**, the message including the CALL ID, the callee IP address (**192.168.3.10**) and UDP port number (**33123**) to advise the call controller of the socket at which it expects to use for audio communications with the media relay **17**.

The call controller **14** then sends a message **1108** to the media relay **17** including the IP address (**192.168.3.10**) and UDP port number (**33123**) identifying the socket at that the callee telephone **15** (or PSTN gateway) that is to be used for audio communications with the media relay. The media relay **17** then creates a caller socket identified by IP address **192.168.2.10** and UDP port number **22125** and creates an internal bridge for relaying audio traffic between the caller socket (**192.168.2.10**: **22125**) and the callee socket (**192.168.2.10**: **22123**).

The media relay **17** then sends a message **1110** including the call ID and the IP address (**192.168.2.10**) and UDP port number (**22125**) identifying the caller socket that the media relay assigned to the caller telephone **12**, back to the call controller **14** to indicate that the caller and callee sockets have been established and that the call can proceed.

The call controller **14** then sends a SIP OK message **1112** to the caller telephone **12** to indicate that the call may now proceed. The SIP OK message includes the caller and callee usernames, the call ID and the IP address

(**192.168.2.10**) and UDP port number (**22125**) identiying the caller socket at the media relay **17**.

Alternatively, referring back to Figure **1**, if the routing message is of a type that identifies a domain associated with another supernode in the system, the call controller **14** may communicate with a different media relay (for example **27**) adapted to establish the above-mentioned links between separate media relays associated with respective supernodes, where the IP network links are provided by the communications medium **23**.

In the case of an emergency call, the routing message is unlikely to identify a domain other than that of the caller.

In the case of a regular, non-emergency call, if the routing message is of the type shown in Figure **25** where there are a plurality of suppliers available, the process proceeds as described above with the exception that instead of communicating with the callee telephone directly, the call controller **14** communicates with a gateway provided by a supplier. If a SIP OK message is not received back from the first gateway, the processor is directed to send the SIP Invite message **1104** to a gateway of the next indicated supplier. For example, the call controller **14** sends the SIP Invite message **1104** to the first supplier, in this case Telus, to determine whether or not Telus is able to handle the call. If Telus does not send back an OK message **1106** or sends a message indicating that it is not able to handle the call, the call controller proceeds to send a SIP Invite message **1104** to the next supplier, in this case Shaw. The process is repeated until one of the suppliers responds with a SIP OK message **1106** indicating that it is available to carry the call and the process proceeds as shown in connection with messages **1108**, **1110** and **1112**.

Referring to Figure **2**, in response to receiving the SIP OK message **1112** at the network interface **48**, the microprocessor **32** of the caller telephone **12**

stores the media relay IP address (**192.168.2.10**) and UDP port number (**22125**) identifying the caller socket at the media relay in an audio path IP address buffer **47** in the temporary memory **40**. The microprocessor **32** is now ready to transfer audio signals and from the handset and the media relay **17** using the sockets created above.

Referring back to Figure **1**, if the call is a regular, non-emergency call, and the call controller **14** receives a message of the type shown in Figure **32**, i.e., a type which has one call forwarding number and/or a voicemail number, the call controller attempts to establish a call (using message **1104** in Figure **38**) to the callee telephone **15** and if no call is established (i.e., message **1106** in Figure **38** is not received) within the associated TTL (**3600** seconds), the call controller **14** attempts to establish a call with the next user identified in the call routing message. This process is repeated until all call forwarding possibilities have been exhausted after respective times to live, in which case an audio path is established with the voicemail server **19** identified in the routing message. The voicemail server **19** sends message **1106** in response to receipt of message **1104** and functions as described above in connection with the callee telephone **15** to permit an outgoing audio message provided by the voicemail server to be heard by the caller and to permit the caller to record an audio message on the voicemail server.

When audio paths are established, a call timer (not shown) maintained by the call controller logs the start date and time of the call and logs the call ID and an identification of the route (i.e., audio path IP address) for later use in billing, for example.

Terminating the Call

In the event that either the caller or the callee (or callee via the PSTN) terminates a call, the telephone of the terminating party (or gateway associated with the terminating party) sends a SIP Bye message to the call controller **14**. An exemplary SIP Bye message is shown at **900** in Figure **33**

and includes a caller field **902**, a callee field **904** and a call ID field **906**. The caller field **902** holds the caller username, the callee field **904** holds a PSTN compatible number or username, and the call ID field **906** holds a unique call identifier field of the type shown in the caller ID field **65** of the SIP Invite message shown in Figure **3**.

Thus, when terminating a regular non-emergency call, such as initiated by the Vancouver subscriber to the Calgary subscriber for example, referring to Figure **34**, a SIP Bye message is produced as shown generally at **908** and the caller field **902** holds a username identifying the Vancouver caller, in this case **2001 1050 8667**, the callee field **904** holds a username identifying the Calgary callee, in this case **2001 1050 2222**, and the callee ID field **906** holds the code FA**10** @ **192.168.0.20**, which is the call ID for the call.

The SIP Bye message shown in Figure **34** is received at the call controller **14** and the call controller executes a process as shown generally at **910** in Figure **35**. The process includes a first block **912** that directs the call controller circuit **100** to copy the caller, callee and call ID field contents from the SIP Bye message **900** shown in Figure **33** received from the terminating party to corresponding fields of an RC Call Stop message buffer (not shown). Block **914** then directs the call controller circuit **100** to copy the call start time from the call timer and to obtain a Call Stop time from the call timer. Block **916** then directs the call controller to calculate a communication session time by determining the difference in time between the call start time and the call stop time. This communication session time is then stored in a corresponding field of the RC Call Stop message buffer. Block **918** then directs the call controller circuit **100** to copy the route identifier from the call log. An RC Call Stop message produced as described above is shown generally at **1000** in Figure **36**. An RC Call Stop message specifically associated with the call made to the Calgary callee is shown generally at **1020** in Figure **37**.

Referring to Figure **36**, the RC Call Stop message includes a caller field **1002**, callee field **1004**, a call ID field **1006**, an account start time field **1008**, an account stop time field **1010**, a communication session time **1012** and a route field **1014**. The caller field **1002** holds a username, the callee field **1004** holds a PSTN-compatible number or system number, the call ID field **1006** holds the unique call identifier received from the SIP Invite message shown in Figure **3**, the account start time field **1008** holds the date and start time of the call, the account stop time field **1010** holds the date and time the call ended, the account session time field **1012** holds a value representing the difference between the start time and the stop time, in seconds, and the route field **1014** holds the IP address for the communications link that was established.

Referring to Figure **37**, an exemplary RC stop call message for the Calgary callee is shown generally at **1020**. In this example the caller field **1002** holds the username **2001 1050 8667** identifying the Vancouver caller and the callee field **1004** holds the username **2001 1050 2222** identifying the Calgary callee. The contents of the call ID field **1006** are FA**10** @ **192.168.0.20**. The contents of the accounting start time field **1008** are **2006-12-30 12:12:12** and the contents of the accounting stop time field are **2006-12-30 12:12:14**. The contents of the communication session time field **1012** are **2** to indicate **2** seconds call duration and the contents of the route field are **72.64.39.58**.

Referring back to Figure **35**, after having produced an RC Call Stop message, block **920** directs the call controller circuit **100** to send the RC stop message contained in the RC Call Stop message buffer to the routing controller **16**.

The routing controller **16** receives the Call Stop message and an RC Call Stop message process is invoked at the RC to deal with charges and billing for the call.

Block **922** directs the call controller circuit **100** to send a Bye message back to the party that did not terminate the call.

Block **924** then directs the call controller circuit **100** to send a "Bye" message of the type shown in Figure **33** to the media relay **17** to cause the media relay to delete the caller and callee sockets it established for the call and to delete the bridge between the sockets.

While specific embodiments of the invention have been described and illustrated, such embodiments should be considered illustrative of the invention only and not as limiting the invention as construed in accordance with the accompanying claims.

What is claimed is:

1. A process for handling emergency calls from a caller in a voice over IP system, the method comprising:

receiving a routing request message including a caller identifier and a callee identifier;

setting an emergency call flag active in response to said callee identifier matching an emergency call identifier pre-associated with the caller;

producing an emergency response center identifier in response to said emergency call identifier;

determining whether said caller identifier is associated with a pre-associated direct inward dialing (DID) identifier;

producing a direct inward dialing (DID) identifier for said caller by associating a temporary DID identifier with said caller identifier when said emergency call flag is active and it is determined that said caller has no pre-associated DID identifier;

producing a routing message including said emergency response center identifier and said temporary DID identifier for receipt by a routing controller operable to cause a route to be established between the caller and said emergency response center.

2. The process of claim 1 wherein setting said emergency call flag active comprises retrieving a dialing profile associated with the caller and setting said emergency call flag active when the contents of an

emergency call identifier field of said dialing profile match said callee identifier.

3. The process of claim **2** wherein determining whether said caller identifier is associated with a pre-associated DID identifier comprises searching a database for a DID record associating a DID identifier with said caller and determining that said caller identifier is associated with a pre-associated DID identifier when said record associating a DID identifier with said caller is found.

4. The process of claim **3** wherein associating a pre-assigned DID identifier with said caller identifier comprises copying said pre-associated DID identifier from said DID record to a DID identifier buffer.

5. The process of claim **4** wherein producing said routing message comprises causing the contents of said DID identifier buffer to define said DID identifier in said routing message.

6. The process of claim **2** wherein determining whether said caller identifier is associated with a pre-associated DID identifier comprises searching a database for a DID record associating a DID identifier with said caller and determining that said caller identifier is not associated with a pre-associated DID identifier when a record associating a DID identifier with said caller is not found.

7. The process of claim **6** wherein associating a temporary DID identifier with said caller identifier comprises associating with said caller identifier a DID identifier from a pool of pre-determined DID identifiers.

8. The process of claim **7** wherein associating said DID identifier from said pool comprises associating a temporary DID record with said

caller, said temporary DID record having a DID identifier field populated with said DID identifier from said pool.

9. The process of claim **8** wherein associating said DID identifier from said pool comprises copying said DID identifier from said temporary DID record to a DID identifier buffer.

10. The process of claim **9** wherein producing said routing message comprises causing the contents of said DID identifier buffer to define said DID identifier in said routing message.

11. The process of claim **8** further comprising canceling said temporary DID record after a pre-defined period of time.

12. The process of claim **2** wherein producing said emergency response center identifier comprises obtaining an emergency response center identifier from an emergency response center field of said dialing profile associated with said caller.

13. The process of claim **12** wherein obtaining comprises copying an emergency response center identifier from said dialing profile associated with said caller to a routing message buffer such that the emergency response center identifier is included in the routing message.

14. The process of claim **1** wherein producing said routing message comprises causing said routing message to specify a maximum call time for said emergency call, said maximum call time exceeding a duration of an average non-emergency telephone call.

15. An apparatus for handling emergency calls from a caller in a voice over IP system, the apparatus comprising:

means for receiving a routing request message including a caller identifier and a callee identifier;

setting means for setting an emergency call flag active in response to said callee identifier matching an emergency call identifier pre-associated with the caller;

means for producing an emergency response center identifier in response to said emergency call identifier;

means for determining whether said caller identifier is associated with a pre-associated direct inward dialing (DID) identifier; and

means for producing a direct inward dialing (DID) identifier for said caller including:

means for associating a temporary DID identifier with said caller identifier in response to said emergency call flag being active and said caller not being pre-associated with a direct inward dialing identifier; and

means for producing a routing message including said emergency response center identifier and said temporary DID identifier or said pre-assigned DID identifier for receipt by a routing controller operable to cause a route to be established between the caller and said emergency response center.

16. The apparatus of claim **15** further comprising means for accessing a database of dialing profiles associated with respective subscribers to

said system, each of said dialing profiles including an emergency call identifier field and an emergency call center field and wherein said setting means comprises means for retrieving a dialing profile associated with the caller and for setting said emergency call flag active when the contents of said emergency call identifier field of said dialing profile match said callee identifier.

17. The apparatus of claim **16** further comprising database accessing means for accessing a database including direct inward dialing (DID) records associated with at least some subscribers to said system, each of said direct inward dialing records comprising a system username and a direct inward dialing number, and wherein said determining means comprises searching means for a database for a DID record associating a DID identifier with said caller and wherein said determining means is operably configured to determine that said caller identifier is associated with a pre-associated DID identifier when said record associating a DID identifier with said caller is found.

18. The apparatus of claim **17** further comprising a DID identifier buffer and wherein said means for associating a pre-assigned DID identifier with said caller identifier comprises means for copying said pre-associated DID identifier from said DID record to said DID identifier buffer.

19. The apparatus of claim **18** wherein said means for producing said routing message comprises means for causing the contents of said DID identifier buffer to define said DID identifier in said routing message.

20. The apparatus of claim **16** further comprising database accessing means for accessing a database including direct inward dialing records associated with at least some subscribers to said system, each of said direct inward dialing records comprising a system username and a

direct inward dialing number and wherein said determining means comprises searching means for searching a database for a DID record associating a DID identifier with said caller and wherein said determining means is operably configured to determine that said caller identifier is not associated with a pre-associated DID identifier when a record associating a DID identifier with said caller is not found.

21. The apparatus of claim **18** further comprising means for accessing a pool of pre-determined DID identifiers and wherein said means for associating a temporary DID identifier with said caller identifier comprises means for associating a DID identifier from said pool of pre-determined DID identifiers with said caller identifier.

22. The apparatus of claim **20** wherein said means for associating said DID identifier from said pool comprises means for associating a temporary DID record with said caller, said temporary DID record having a DID identifier field populated with said DID identifier from said pool.

23. The apparatus of claim **22** wherein said means for associating comprises means for copying said DID identifier from said temporary DID record to a DID identifier buffer.

24. The apparatus of claim **22** wherein said means for producing said routing message comprises means for causing the contents of said DID identifier buffer to define said DID identifier in said routing message.

25. The apparatus of claim **22** further comprising means for canceling said temporary DID record after a period of time.

26. The apparatus of claim **16** wherein said means for producing said emergency response center identifier comprises means for obtaining

an emergency response center identifier from an emergency response center field of said dialing profile associated with said caller.

27. The apparatus of claim **26** further comprising a routing message buffer and wherein said means for obtaining comprises means for copying the contents of said emergency response center field of said dialing profile associated with said caller to the routing message bufer such that said contents of said emergency response center field are included in said routing message.

28. The apparatus of claim **27** wherein said means for producing said routing message comprises means for causing said routing message to include a maximum call time for said emergency call, said maximum call time exceeding a duration of an average non-emergency telephone call.

29. An apparatus for handling emergency calls from a caller in a voice over IP system, the apparatus comprising a processor circuit operably configured to:

receive a routing request message including a caller identifier and a callee identifier;

set an emergency call flag active in response to said callee identifier matching an emergency call identifier pre-associated with the caller;

produce an emergency response center identifier in response to said emergency call identifier;

determine whether said caller identifier is associated with a pre-associated direct inward dialing (DID) identifier;

produce a direct inward dialing (DID) identifier for said caller by associating a temporary DID identifier with said caller identifier when said emergency call flag is active and it is determined that said caller has no pre-associated DID identifier; and

produce a routing message including said emergency response center identifier and said temporary DID identifier for receipt by a routing controller operable to cause a route to be established between the caller and said emergency response center.

30. The apparatus of claim **29** wherein said processor circuit is operably configured to retrieve a dialing profile associated with the caller and set said emergency call flag active when the contents of an emergency call identifier field of said dialing profile match said callee identifier.

31. The apparatus of claim **30** wherein said processor circuit is operably configured to search a database for a DID record associating a DID identifier with said caller and determine that said caller identifier is associated with a pre-associated DID identifier when said record associating a DID identifier with said caller is found.

32. The apparatus of claim **31** wherein said processor circuit is operably configured to copy said pre-associated DID identifier from said DID record to a DID identifier buffer.

33. The apparatus of claim **32** wherein said processor circuit is operably configured to cause the contents of said DID identifier buffer to define said DID identifier in said routing message.

34. The apparatus of claim **30** wherein said processor circuit is operably configured to search a database for a DID record associating a DID

identifier with said caller and determine that said caller identifier is not associated with a pre-associated DID identifier when a record associating a DID identifier with said caller is not found.

35. The apparatus of claim **32** wherein said processor circuit is operably configured to associate with said caller identifier a DID identifier from a pool of pre-determined DID identifiers.

36. The apparatus of claim **35** wherein said processor circuit is operably configured to associate a temporary DID record with the caller, said temporary DID record having a DID identifier field populated with said DID identifier from said pool.

37. The apparatus of claim **36** wherein said processor circuit is operably configured to copy said DID identifier from said temporary DID record to a DID buffer.

38. The apparatus of claim **35** wherein said processor circuit is operably configured to cause the contents of said DID identifier buffer to define said DID identifier in said routing message.

39. The apparatus of claim **36** wherein said processor circuit is operably configured to cancel said temporary DID record after a period of time.

40. The apparatus of claim **30** wherein said processor circuit is operably configured to obtain an emergency response center identifier from an emergency response center field of said dialing profile associated with said caller.

41. The apparatus of claim **40** further comprising a routing message buffer and wherein said processor circuit is operably configured to copy an emergency response center identifier from said dialing profile

associated with said caller to said routing message buffer such that said emergency response center identifier is included in said routing message.

42. The apparatus of claim **29** wherein said processor circuit is operably configured to cause said routing message to include a maximum call time for said emergency call, said maximum call time exceeding a duration of an average non-emergency telephone call.

43. A computer readable medium encoded with codes for directing a processor circuit to handle emergency calls from callers in a voice over IP system, said codes directing said processor circuit to:

receive a routing request message including a caller identifier and a callee identifier;

set an emergency call flag active in response to said callee identifier matching an emergency call identifier pre-associated with the caller;

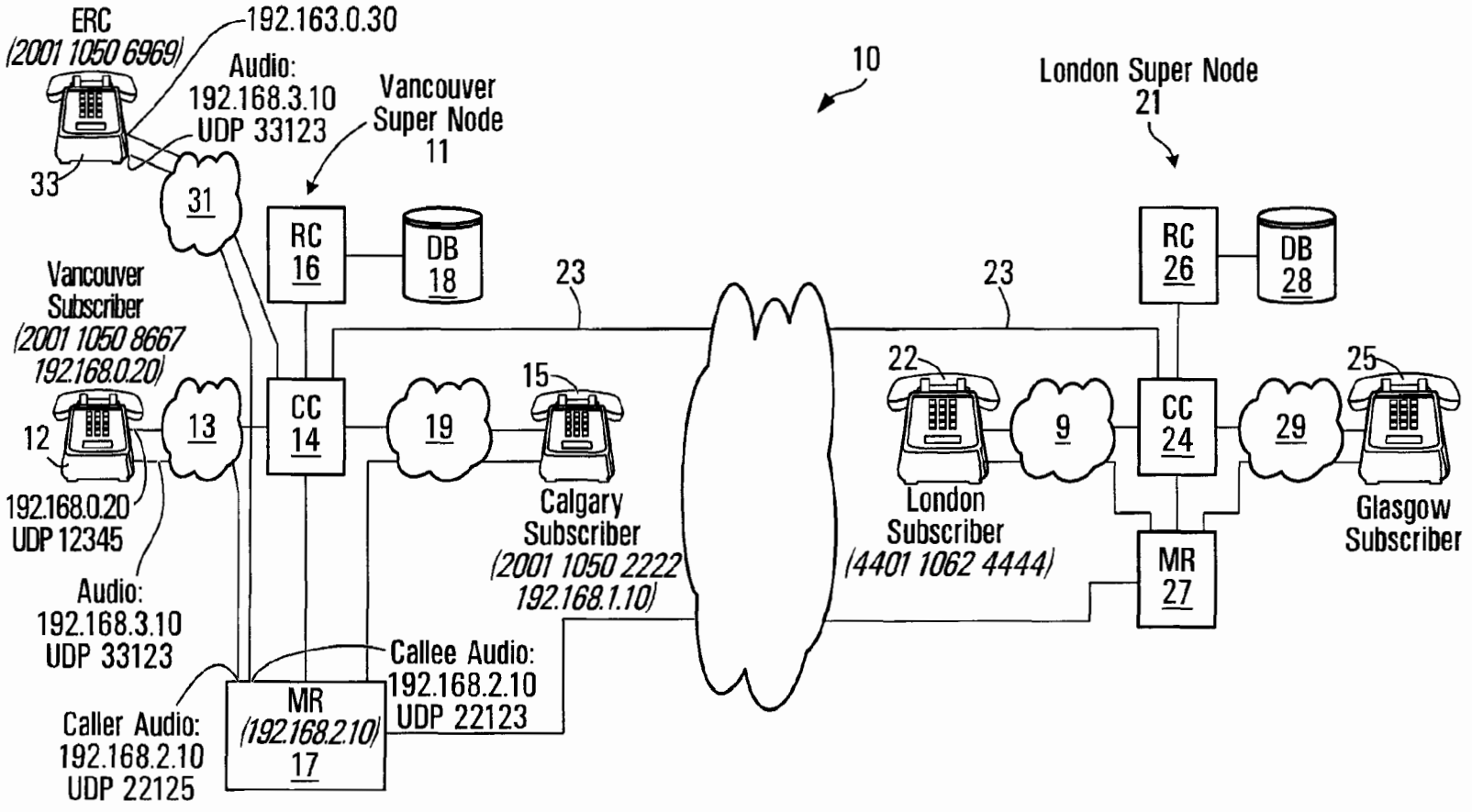
produce an emergency response center identifier in response to said emergency call identifier;

determine whether said caller identifier is associated with a pre-associated direct inward dialing (DID) identifier;

produce a direct inward dialing (DID) identifier for said caller by associating a temporary DID identifier with said caller identifier when said emergency call flag is active and it is determined that said caller has no pre-associated DID identifier; and

produce a routing message including said emergency response center identifier and said temporary DID identifier for receipt by a routing controller operable to cause a route to be established between the caller and said emergency response center.

5


ERC
(2001 1050 6969)
192.163.0.30
Audio:
192.168.3.10
UDP 33123
Vancouver
Super Node
11
10
London Super Node
21
33
31
RC
16
DB
18
23
23
RC
26
DB
28
Vancouver
Subscriber
(2001 1050 8667
192.168.0.20)
12
13
CC
14
19
15
22
9
CC
24
29
25
192.168.0.20
UDP 12345
Calgary
Subscriber
(2001 1050 2222
192.168.1.10)
London
Subscriber
(4401 1062 4444)
Glasgow
Subscriber
MR
27
Audio:
192.168.3.10
UDP 33123
Callee Audio:
192.168.2.10
UDP 22123
Caller Audio:
192.168.2.10
UDP 22125
MR
(192.168.2.10)
17

1/22

FIG. 1

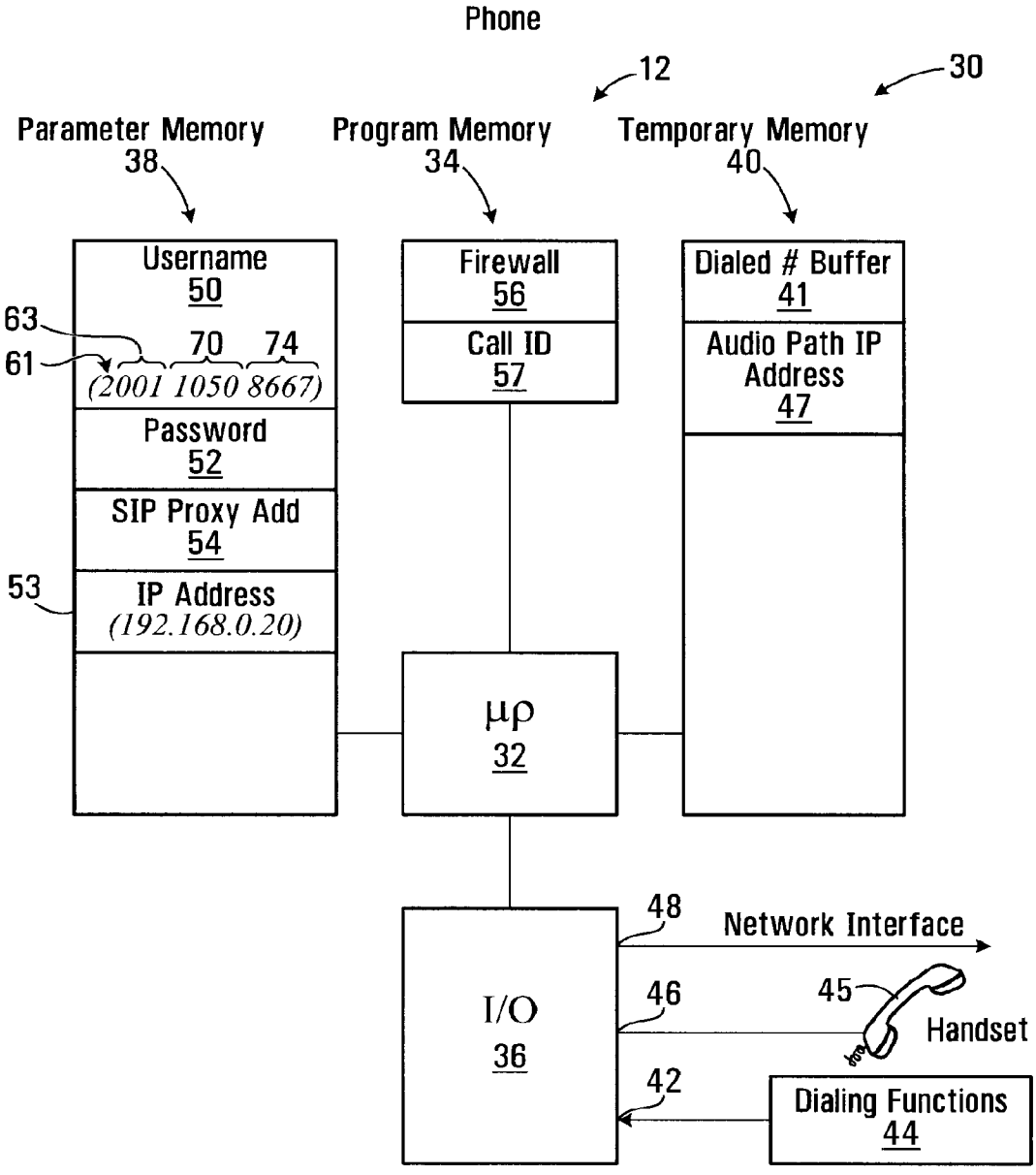

Phone
12
30
Parameter Memory
38
Program Memory
34
Temporary Memory
40
Username
50
63
61
70
74
(2001 1050 8667)
Password
52
SIP Proxy Add
54
53
IP Address
(192.168.0.20)
Firewall
56
Call ID
57
Dialed # Buffer
41
Audio Path IP Address
47
μρ
32
48
Network Interface
I/O
36
46
45
Handset
42
Dialing Functions
44

FIG. 2

SIP Invite Message 59

60	Caller	2001 1050 8667
62	Callee	4401 1062 4444
64	Digest Parameters	XXXXXXX
65	Call ID	FF10@ 192.168.0.20
67	Caller IP Address	192.168.0.20
69	Caller UDP port	12345

FIG. 3

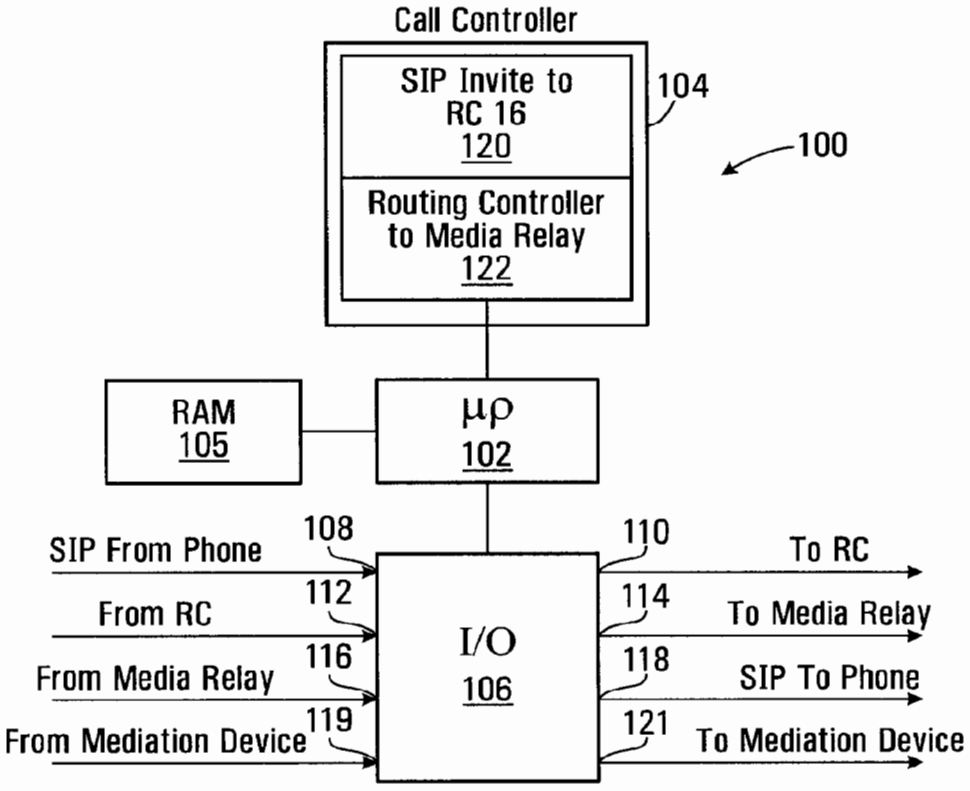


FIG. 4

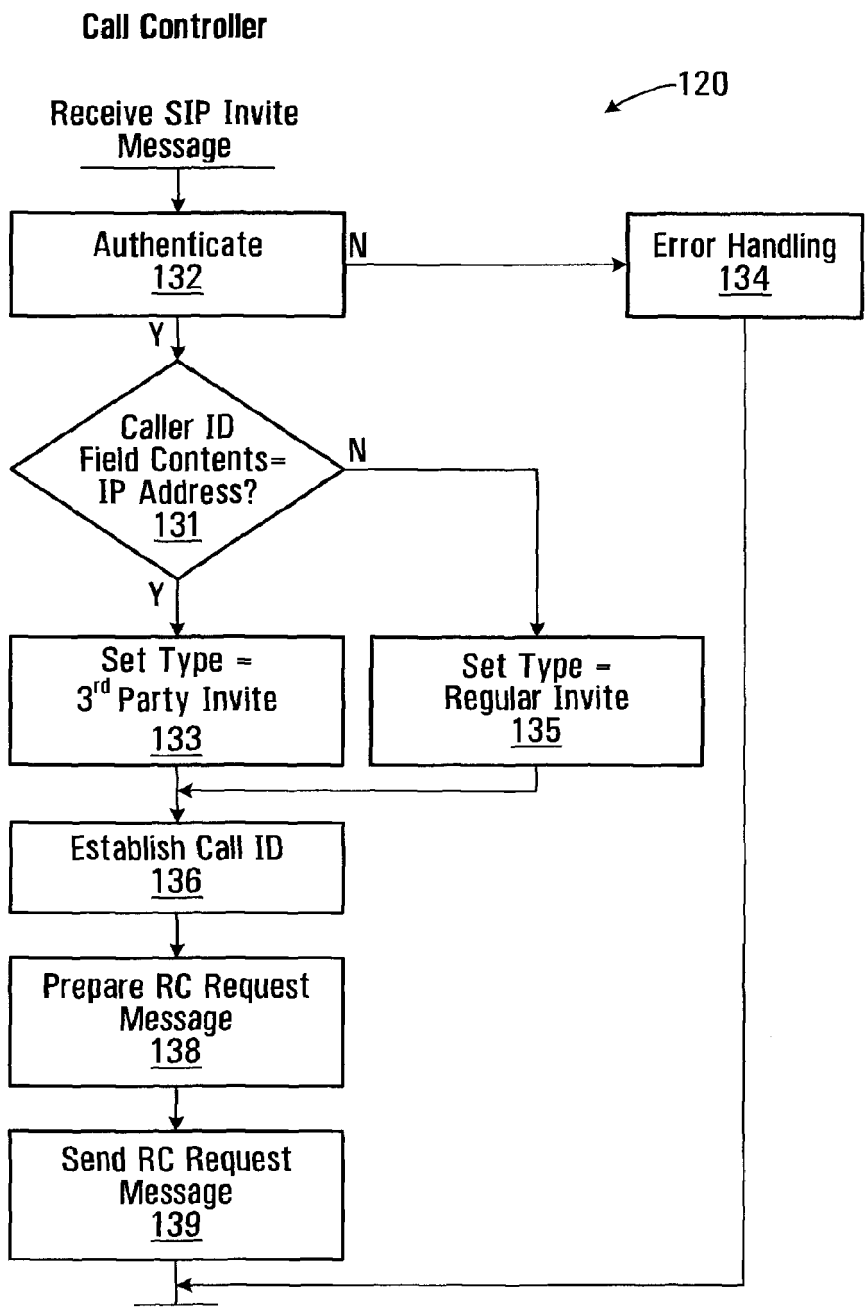

Call Controller
120
Receive SIP Invite Message
Authenticate 132
N
Error Handling 134
Y
Caller ID Field Contents= IP Address? 131
N
Y
Set Type = 3rd Party Invite 133
Set Type = Regular Invite 135
Establish Call ID 136
Prepare RC Request Message 138
Send RC Request Message 139

FIG. 5

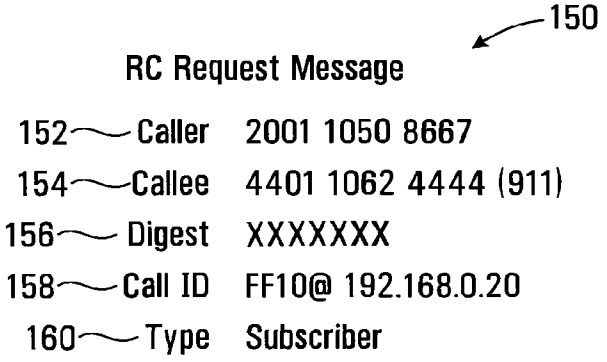

150
RC Request Message
152 Caller 2001 1050 8667
154 Callee 4401 1062 4444 (911)
156 Digest XXXXXXX
158 Call ID FF10@ 192.168.0.20
160 Type Subscriber

FIG. 6

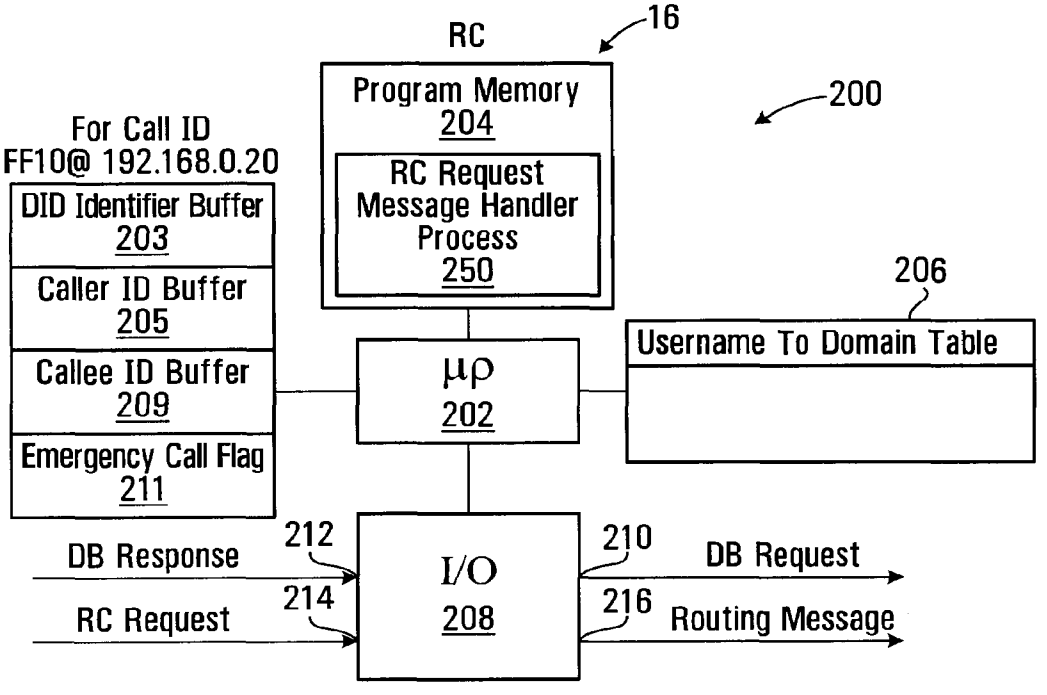

16
RC
Program Memory
204
RC Request
Message Handler
Process
250
200
For Call ID
FF10@ 192.168.0.20
DID Identifier Buffer
203
Caller ID Buffer
205
Callee ID Buffer
209
Emergency Call Flag
211
206
Username To Domain Table
µp
202
212
DB Response
214
RC Request
I/O
208
210
DB Request
216
Routing Message

FIG. 7

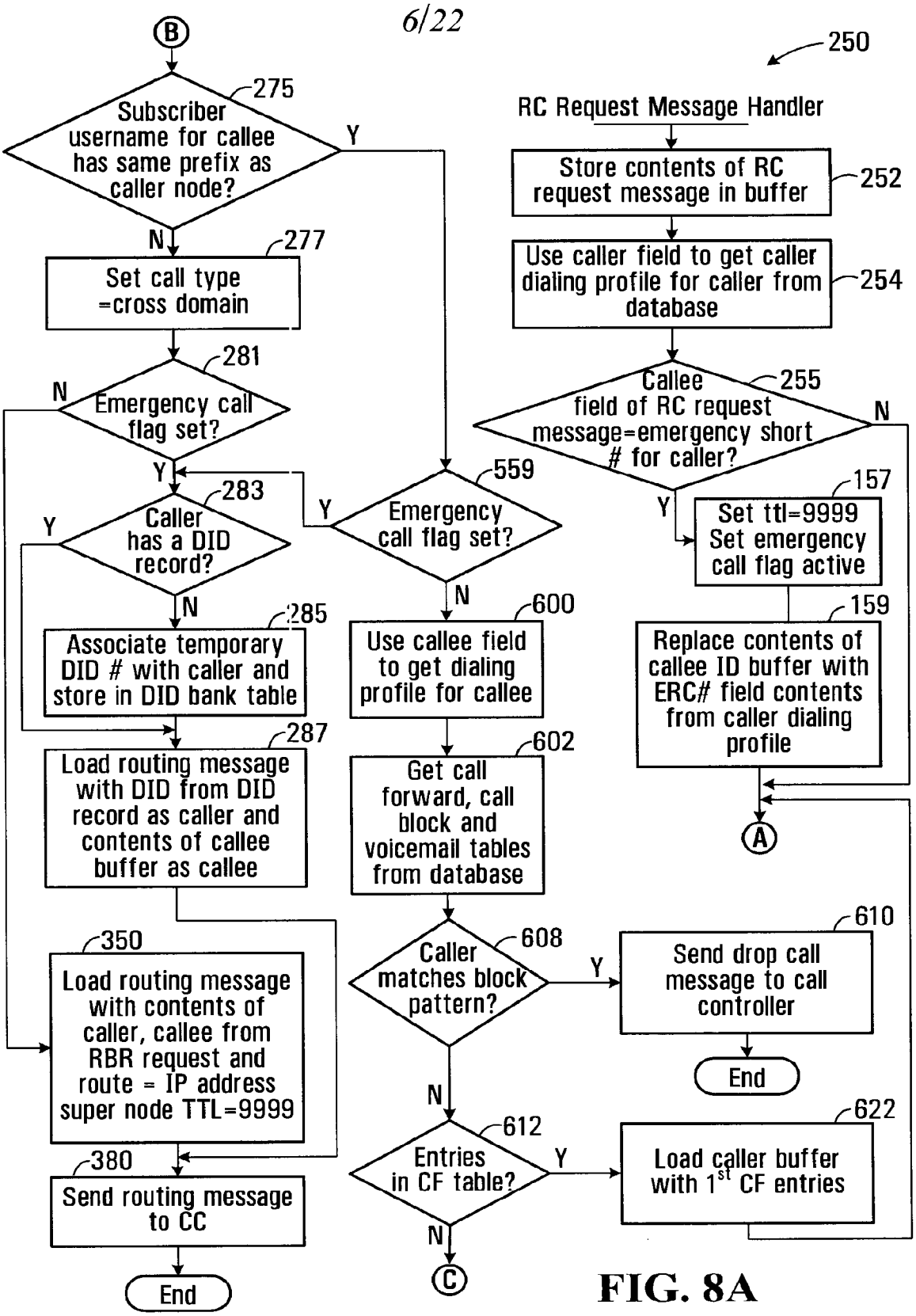

250
RC Request Message Handler
Store contents of RC request message in buffer
252
Use caller field to get caller dialing profile for caller from database
254
Callee field of RC request message=emergency short # for caller?
255
Y
N
Set ttl=9999 Set emergency call flag active
157
Replace contents of callee ID buffer with ERC# field contents from caller dialing profile
159
A
B
Subscriber username for callee has same prefix as caller node?
275
Y
N
Set call type =cross domain
277
Emergency call flag set?
281
Caller has a DID record?
283
Associate temporary DID # with caller and store in DID bank table
285
Load routing message with DID from DID record as caller and contents of callee buffer as callee
287
Load routing message with contents of caller, callee from RBR request and route = IP address super node TTL=9999
350
Send routing message to CC
380
End
Emergency call flag set?
559
Use callee field to get dialing profile for callee
600
Get call forward, call block and voicemail tables from database
602
Caller matches block pattern?
608
Send drop call message to call controller
610
End
Entries in CF table?
612
Load caller buffer with 1st CF entries
622
C

FIG. 8A

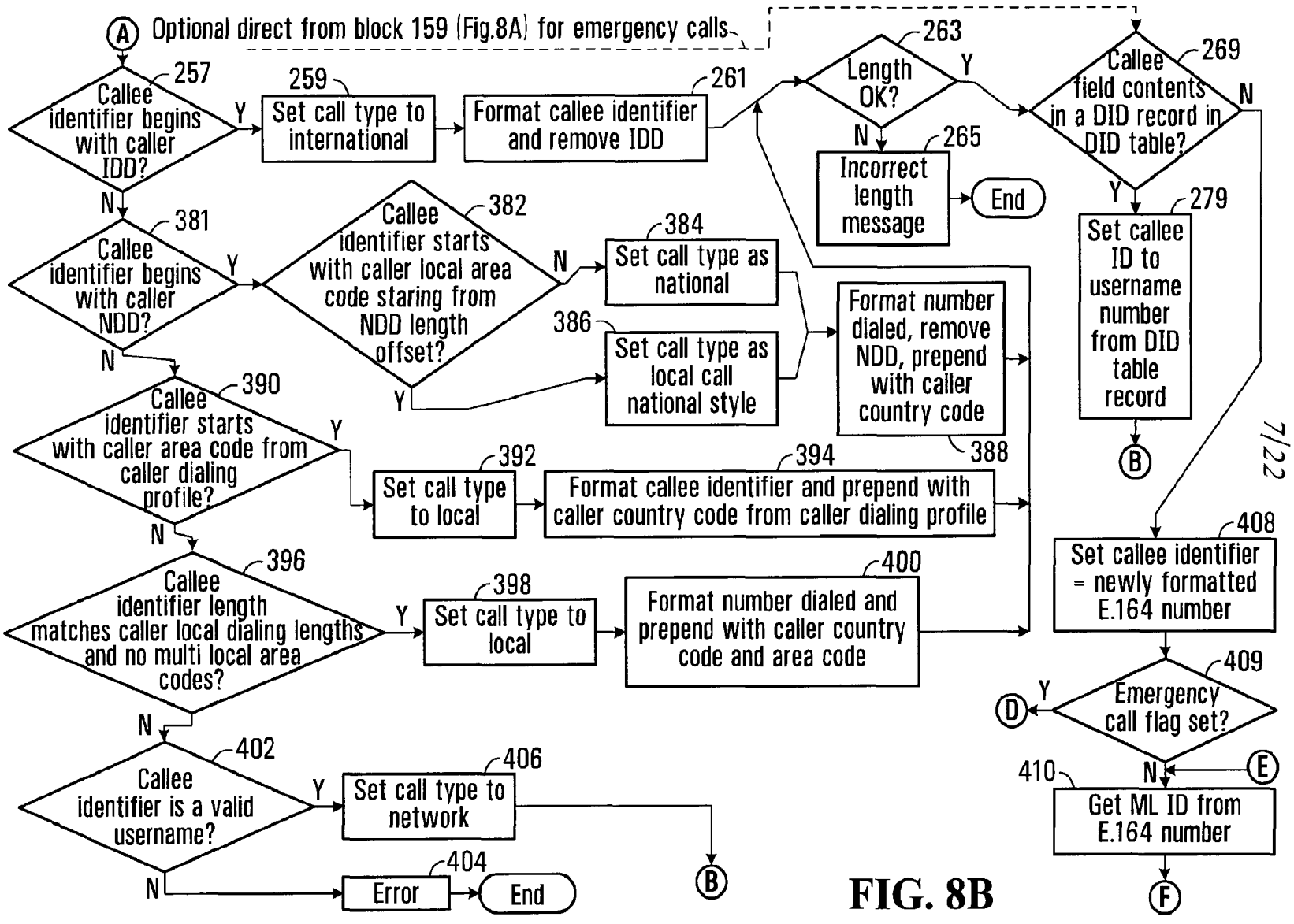

A
Optional direct from block 159 (Fig.8A) for emergency calls
257
Callee identifier begins with caller IDD?
Y
259
Set call type to international
261
Format callee identifier and remove IDD
263
Length OK?
Y
N
265
Incorrect length message
End
269
Callee field contents in a DID record in DID table?
N
Y
279
Set callee ID to username number from DID table record
B
N
381
Callee identifier begins with caller NDD?
Y
382
Callee identifier starts with caller local area code staring from NDD length offset?
N
384
Set call type as national
386
Set call type as local call national style
Y
Format number dialed, remove NDD, prepend with caller country code
388
N
390
Callee identifier starts with caller area code from caller dialing profile?
Y
392
Set call type to local
394
Format callee identifier and prepend with caller country code from caller dialing profile
408
Set callee identifier = newly formatted E.164 number
N
396
Callee identifier length matches caller local dialing lengths and no multi local area codes?
Y
398
Set call type to local
400
Format number dialed and prepend with caller country code and area code
409
Emergency call flag set?
Y
D
N
E
410
Get ML ID from E.164 number
F
N
402
Callee identifier is a valid username?
Y
406
Set call type to network
B
N
404
Error
End

FIG. 8B

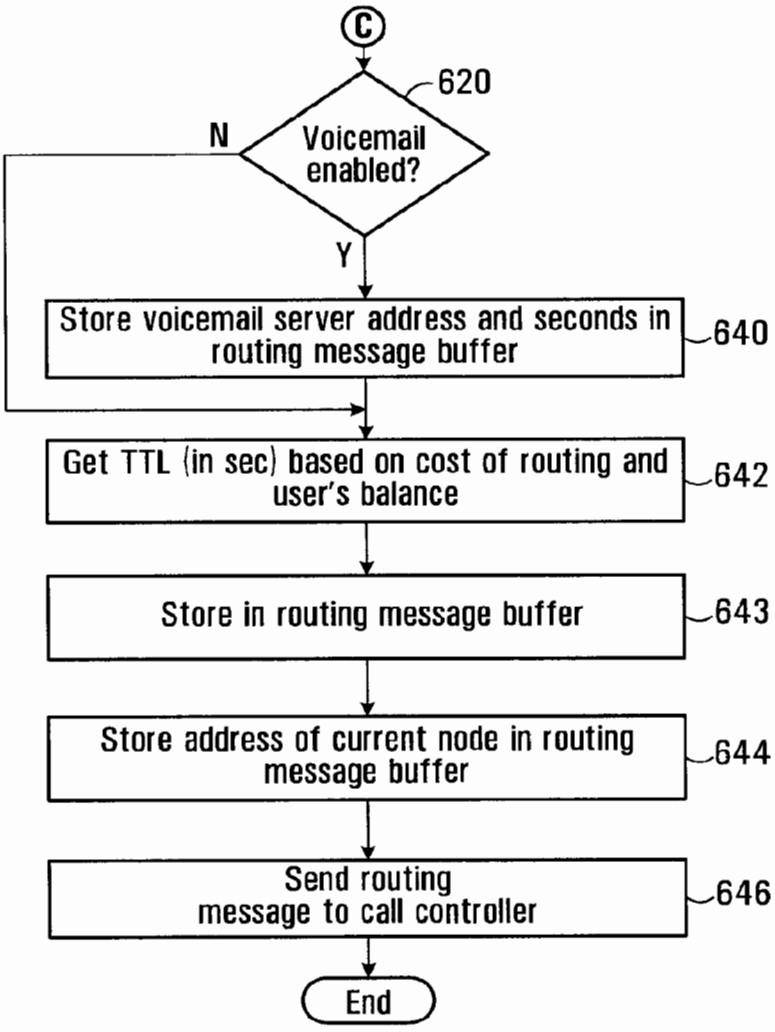

C
620
Voicemail enabled?
N
Y
Store voicemail server address and seconds in routing message buffer
640
Get TTL (in sec) based on cost of routing and user's balance
642
Store in routing message buffer
643
Store address of current node in routing message buffer
644
Send routing message to call controller
646
End

FIG. 8C

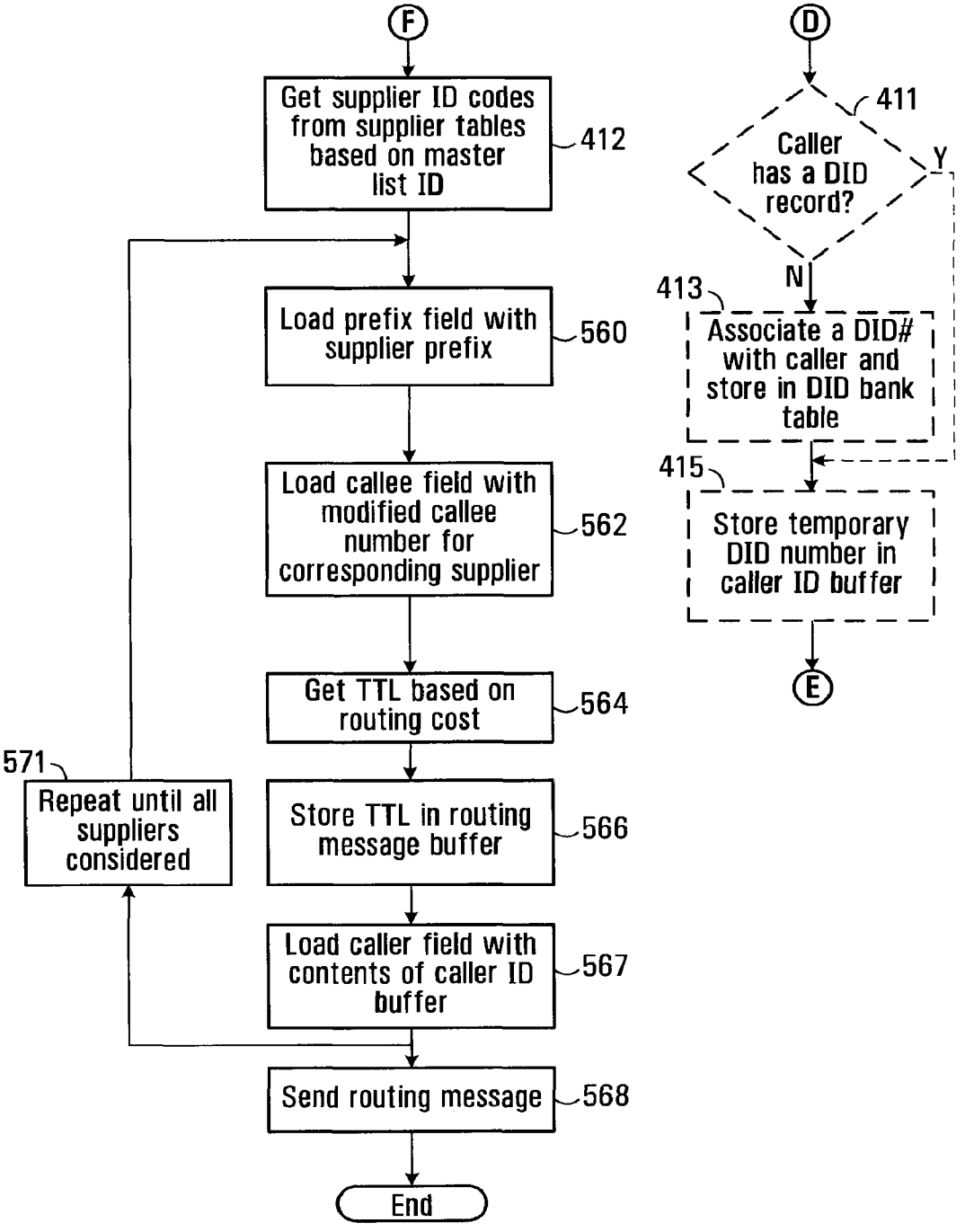

F
Get supplier ID codes from supplier tables based on master list ID
412
Load prefix field with supplier prefix
560
Load callee field with modified callee number for corresponding supplier
562
Get TTL based on routing cost
564
Store TTL in routing message buffer
566
Load caller field with contents of caller ID buffer
567
Send routing message
568
End
571
Repeat until all suppliers considered
D
411
Caller has a DID record?
Y
N
413
Associate a DID# with caller and store in DID bank table
415
Store temporary DID number in caller ID buffer
E

FIG. 8D

256

Dialing Profile for a User

258	**Username**	Assigned on Subscription
260	**Domain**	Domain Associated with User
262	**NDD**	1
264	**IDD**	011
266	**Country Code**	1
267	**Local Area Codes**	604;778
268	**Caller Minimum Local Length**	10
270	**Caller Maximum Local Length**	10
273	**Reseller**	Retailer
275	**Subscriber Address**	The Users Street Address
277	**Emergency Short#**	Local Emergency No. (e.g. 911)
279	**ERC#**	Emergency Response Center Number

FIG. 9

Dialing Profile for Vancouver Subscriber

276

61

284 63 70 74

258	**Username**	2001 1050 8667
260	**Domain**	sp.yvr.digifonica.com ← 282
262	**NDD**	1 (286)
264	**IDD**	011 (288) (290)
266	**Country Code**	1
267	**Local Area Codes**	604;778 (Vancouver)
268	**Caller Minimum Local Length**	10
270	**Caller Maximum Local Length**	10
273	**Reseller**	Klondike
275	**User Address**	2200 650 Georgia St. Van. B.C. Canada V4M 3G8
277	**Emergency Call Identifier#**	911
279	**ERC#**	604 666 3655

FIG. 10

Dialing Profile for ERC Subscriber

Username	2001 1050 6969
Domain	sp.yvr.digifonica.com
NDD	1
IDD	011
Country Code	1
Local Area Codes	604
Caller Minimum Local Length	7
Caller Maximum Local Length	10
Reseller	Klondike
User Address	#Epson downs, Tripson Falls B.C. Canada VON 2N3
Emergency Call Identifier#	911
ERC#	604 666 3655

FIG. 10A

Dialing Profile for Calgary Subscriber	
Username	2001 1050 2222
Domain	sp.yvr.digifonica.com
NDD	1
IDD	011
Country Code	1
Local Area Codes	403
Caller Minimum Local Length	7
Caller Maximum Local Length	10
User Address	1210 Deerfoot Trail SE, Calgary Alberta Canada H0H 0H0
Emergency Short#	911
ERC#	403 123 4567

FIG. 11

Dialing Profile for London Subscriber	
Username	4401 1062 4444
Domain	sp.lhr.digifonica.com
NDD	0
IDD	00
Country Code	44
Local Area Codes	20 (London)
Caller Minimum Local Length	10
Caller Maximum Local Length	11
User Address	21 Basil Rd. Faulty Towers, London NW1 4NS
Emergency Short#	911
ERC#	7487-7973

FIG. 12

268

DID Bank Table Record Format

291	Username	System subscriber
272	User Domain	Host name of supernode
274	DID	E164#
For temporary DID records only	Creation Time	Current time when record is created
	Expiration Time	Time at which record is to be deleted

FIG. 13

276

DID Bank Table Record for Vancouver Subscriber

291	Username	2001 1050 8667
272	User Domain	sp.yvr.digifonica.com
274	DID	1 604 321 3353

293 295 297 299

FIG. 13A

276

DID Bank Table Record for Calgary Subscriber

291	Username	2001 1050 2222
272	User Domain	sp.yvr.digifonica.com
274	DID	1 403 516 0744

293 295 297 299

FIG. 13B

276

DID Bank Table Record for London Subscriber

291	Username	4401 1062 4444
272	User Domain	sp.lhr.digifonica.com
274	DID	44 020 7487 7900

293 295 297 299

FIG. 14

352

Routing Message Buffer

354	Supplier Prefix (optional)	Code identifying supplier traffic
356	Delimiter	Symbol separating fields
358	Callee	PSTN compatible number or Digifonica number
360	Route	Domain name and IP address
362	Time to Live(TTL)	In seconds
364	Caller ID	DID number or username (contents of callee ID buffer)

FIG. 15

366

Routing Message for Emergency Call

200110506969@sp.yvr.digifonica.com;ttl=9999;CLI=16043213353

358 360 362 364

FIG. 16

366

Routing Message Different Node

440110624444@sph.lhr.digifonica.com;ttl=9999;CLI=200110508667

358 360 362 364

FIG. 16A

370

Prefix to Supernode Table Record Format

372	Prefix	First n digits of callee identifier
374	Supernode Address	IP address or fully qualified domain name

FIG. 17

Prefix to Supernode Table Record for London Subscriber

Prefix	4
Supernode Address	sp.lhr.digifonica.com

FIG. 18

Master List Record Format

500	ml_id	1019
502	Dialing code	1604
504	Country code	The country code is the national prefix to be used when dialing TO a particular country FROM another country.
506	Nat Sign #(Area Code)	604
508	Min Length	7
510	Max Length	7
512	NDD	The NDD prefix is the access code used to make a call **WITHIN** that country from on city to another (when calling another city in the same vicinity, this may not be necessary).
514	IDD	The **IDD prefix** is the international prefix needed to dial a call **FROM** the country listed **TO** another country.
516	Buffer rate	Safe charge rate above the highest rate charged by suppliers

FIG. 19

Example: Master List Record with Populated Fields

ml_id	1019
Dialing code	1604
Country code	1
Nat Sign #(Area Code)	604
Min Length	7
Max Length	7
NDD	1
IDD	011
Buffer rate	$0.009/min

FIG. 20

Suppliers List Record Format

540	Sup_id	Name code
542	MI_id	Numeric code
544	Prefix (optional)	String identifying supplier's traffic #
546	Route	IP address
548	NDD/IDD rewrite	
550	Rate	Cost per second to Digifonica to use this route

FIG. 21

(Telus) Supplier Record

Sup_id	2010 (Telus)
MI_id	1019
Prefix (optional)	4973#
Route	72.64.39.58
NDD/IDD rewrite	011
Rate	$0.02/min

FIG. 22

(Shaw) Supplier Record

Sup_id	2011 (Shaw)
MI_id	1019
Prefix (optional)	4974#
Route	73.65.40.59
NDD/IDD rewrite	011
Rate	$0.025/min

FIG. 23

(Sprint) Supplier Record

Sup_id	2012 (Sprint)
MI_id	1019
Prefix (optional)	4975#
Route	74.66.41.60
NDD/IDD rewrite	011
Rate	$0.03/min

FIG. 24

356

Routing Message Buffer for Gateway Call

570 ~ 4973#0116046663655@72.64.39.58;ttl=3600;16046827780
572 ~ 4974#0116046663655@73.65.40.59;ttl=3600;16046827780
574 ~ 4975#0116046663655@74.66.41.60;ttl=3600;16046827780

FIG. 25

Call Block Table Record Format

604 ~	**Username**	Digifonica #
606 ~	**Block Pattern**	PSTN compatible or Digifonica #

FIG. 26

Call Block Table Record for Calgary Subscriber

604 ~	**Username of Callee**	2001 1050 2222
606 ~	**Block Pattern**	2001 1050 8664

FIG. 27

Call Forwarding Table Record Format for Callee

614 ~	**Username of Callee**	Digifonica #
616 ~	**Destination Number**	Digifonica #
618 ~	**Sequence Number**	Integer indicating order to try this

FIG. 28

Call Forwarding Table Record for Calgary Subscriber

614	Username of Callee	2001 1050 2222
616	Destination Number	2001 1055 2223
618	Sequence Number	1

FIG. 29

Voicemail Table Record Format

624	Username of Callee	Digifonica #
626	Vm Server	domain name
628	Seconds to Voicemail	time to wait before engaging voicemail
630	Enabled	yes/no

FIG. 30

Voicemail Table Record for Calgary Subscriber

624	Username of Callee	2001 1050 2222
626	Vm Server	vm.yvr.digifonica.com
628	Seconds to Voicemail	20
630	Enabled	1

FIG. 31

Routing Message Buffer for CF/VM Routing Message

650 ~ 200110502222@sp.yvr.digifonica.com;ttl=3600
652 ~ 200110552223@sp.yvr.digifonica.com;ttl=3600
654 ~ vm.yvr.digifonica.com;20;ttl=60
656 ~ sp.yvr.digifonica.com

FIG. 32

900

Bye Message

902	**Caller**	Username
904	**Callee**	PSTN compatible # or Username
906	**Call ID**	unique call identifier (hexadecimal string@IP))

FIG. 33

908

Bye Message

902	**Caller**	2001 1050 8667
904	**Callee**	2001 1050 2222
906	**Call ID**	FA10@192.168.0.20

FIG. 34

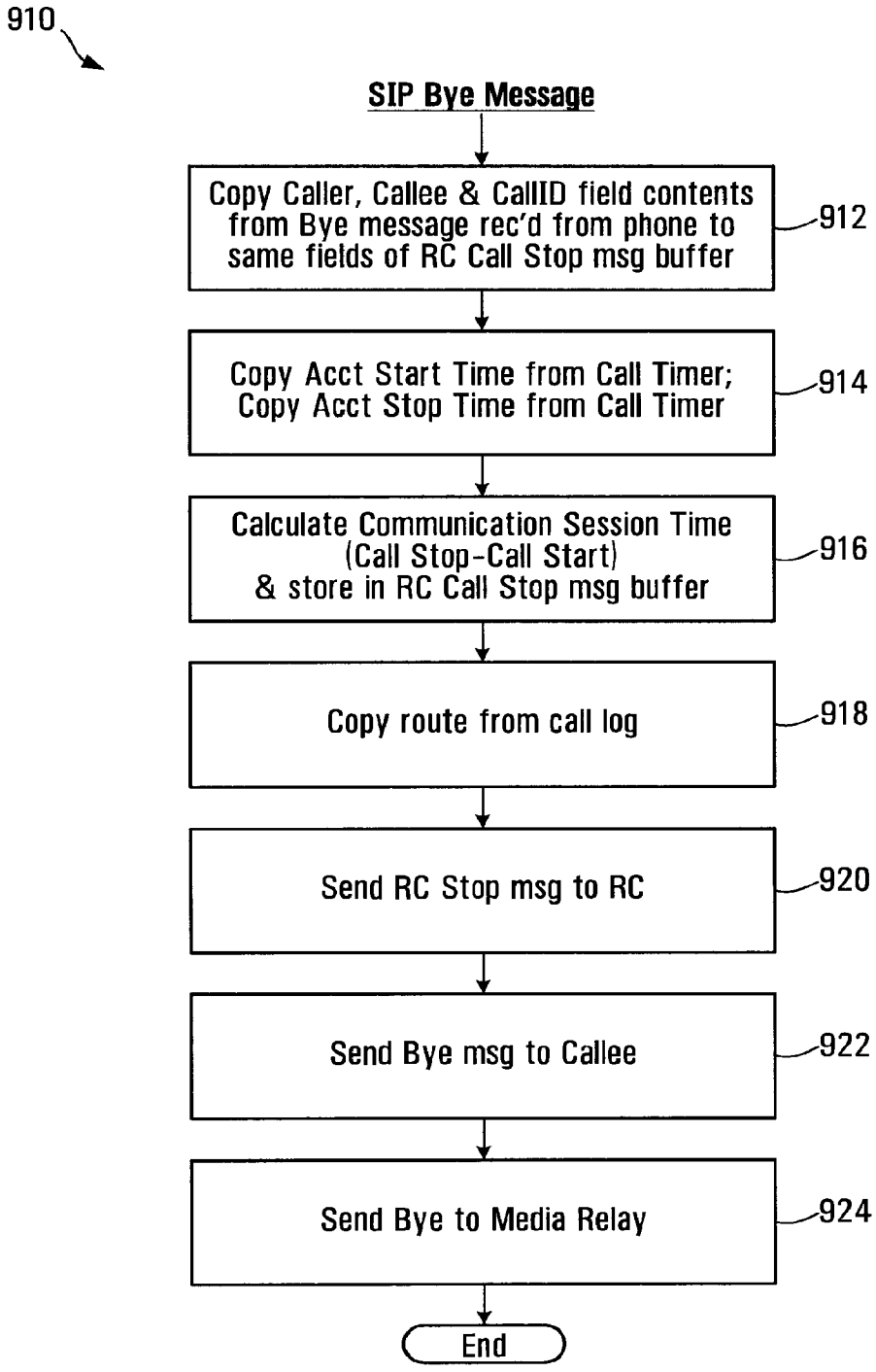

910
SIP Bye Message
Copy Caller, Callee & CallID field contents from Bye message rec'd from phone to same fields of RC Call Stop msg buffer
912
Copy Acct Start Time from Call Timer; Copy Acct Stop Time from Call Timer
914
Calculate Communication Session Time (Call Stop-Call Start) & store in RC Call Stop msg buffer
916
Copy route from call log
918
Send RC Stop msg to RC
920
Send Bye msg to Callee
922
Send Bye to Media Relay
924
End

FIG. 35

1000

RC Call Stop Message

1002	**Caller**	Username
1004	**Callee**	PSTN compatible # or Username
1006	**Call ID**	unique call identifier (hexadecimal string@IP)
1008	**Acct Start Time**	start time of call
1010	**Acct Stop Time**	time the call ended
1012	**Acct Session Time**	start time-stop time (in seconds)
1014	**Route**	IP address for the communications link that was established

FIG. 36

1020

RC Call Stop Message for Calgary Subscriber

1002	**Caller**	2001 1050 8667
1004	**Callee**	2001 1050 2222
1006	**Call ID**	FA10@192.168.0.20
1008	**Acct Start Time**	2006-12-30 12:12:12
1010	**Acct Stop Time**	2006-12-30 12:12:14
1012	**Acct Session Time**	2
1014	**Route**	72.64.39.58

FIG. 37

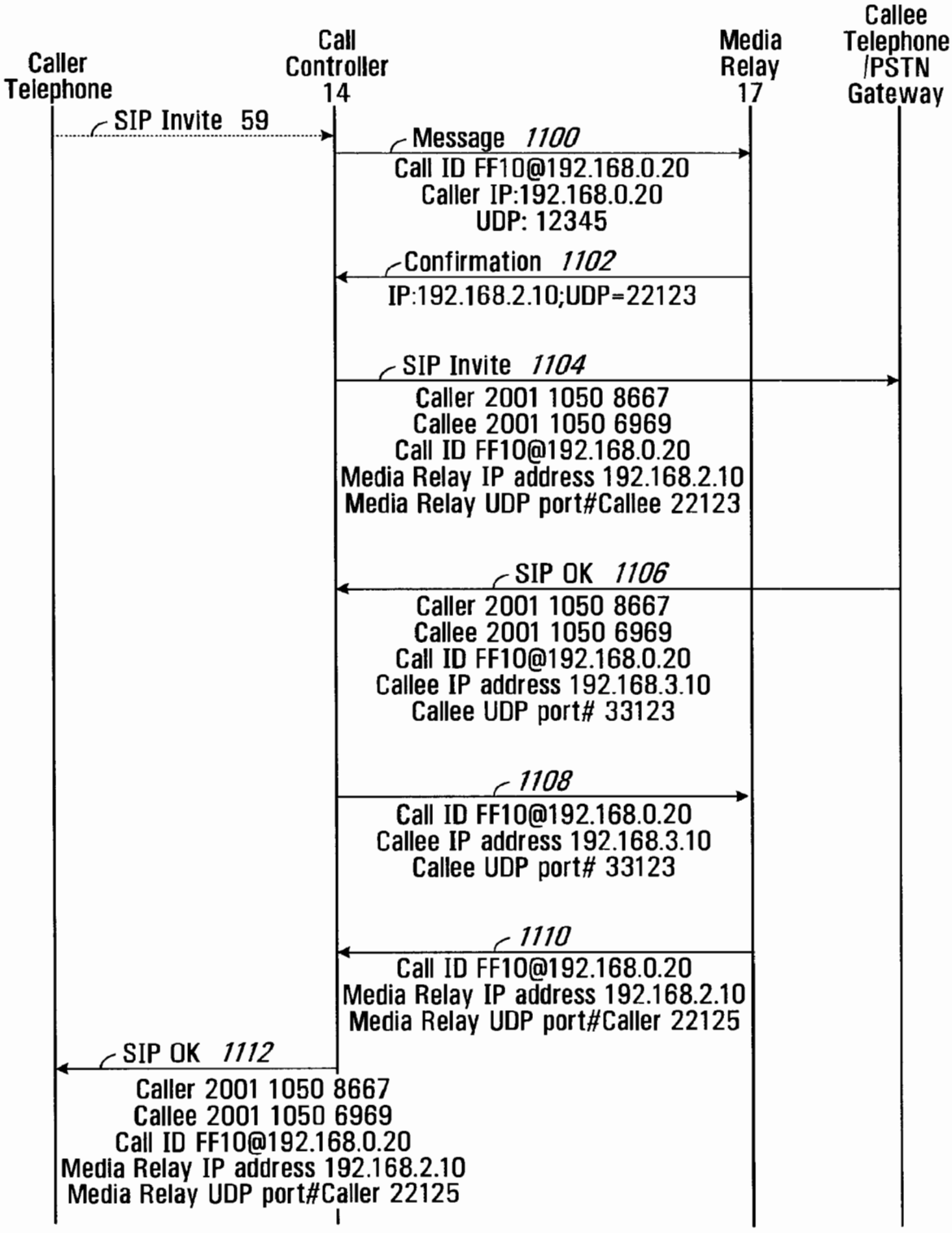

Caller Telephone
Call Controller 14
Media Relay 17
Callee Telephone /PSTN Gateway
SIP Invite 59
Message 1100
Call ID FF10@192.168.0.20
Caller IP:192.168.0.20
UDP: 12345
Confirmation 1102
IP:192.168.2.10;UDP=22123
SIP Invite 1104
Caller 2001 1050 8667
Callee 2001 1050 6969
Call ID FF10@192.168.0.20
Media Relay IP address 192.168.2.10
Media Relay UDP port#Callee 22123
SIP OK 1106
Caller 2001 1050 8667
Callee 2001 1050 6969
Call ID FF10@192.168.0.20
Callee IP address 192.168.3.10
Callee UDP port# 33123
1108
Call ID FF10@192.168.0.20
Callee IP address 192.168.3.10
Callee UDP port# 33123
1110
Call ID FF10@192.168.0.20
Media Relay IP address 192.168.2.10
Media Relay UDP port#Caller 22125
SIP OK 1112
Caller 2001 1050 8667
Callee 2001 1050 6969
Call ID FF10@192.168.0.20
Media Relay IP address 192.168.2.10
Media Relay UDP port#Caller 22125

FIG. 38

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2008/000545

A. CLASSIFICATION OF SUBJECT MATTER
IPC: ***H04L 12/66*** (2006.01) , ***H04M 11/06*** (2006.01) , ***H04M 3/42*** (2006.01) , ***H04Q 3/00*** (2006.01) , ***H04Q 3/64*** (2006.01)
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC: H04L (2006.01), H04M (2006.01), H04Q (2006.01); US classes: 379/45, 370 in combination with keywords

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)
Canadian Patent Database, USPTO West, Delphion. Keywords: pbx, inward dialing, dial-in, voip, sip, psap, erc, ali, flag and variations thereof.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, P	US 2008/0063153 A1 (Krivorot et al.) 13 March 2008 (13-03-2008) * Para. [0004] - [0024], [0042] - [0082]; Fig. 1 *	1, 15, 29, 43
A, P		2-14, 16-28, 30-42
A	US 2005/0083911 A1 (Grabelsky et al.) 21 April 2005 (21-04-2005) * Para. [0041] - [0055], [0060] - [0138]; Fig. 1, 2 *	1-43
A	US 2005/0169248 A1 (Truesdale et al.) 4 August 2005 (04-08-2005) * Para. [0013] - [0050] *	1-43
A, P	US 2008/0037715 A1 (Prozeniuk et al.) 14 February 2008 (14-02-2008) * Entire document *	1-43
A, P	WO 2007/044454 A2 (Croy et al.) 19 April 2007 (19-04-2007) * Entire document *	1-43

[ ] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents :
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
6 June 2008 (06-06-2008)	20 June 2008 (20-06-2008)

Name and mailing address of the ISA/CA	Authorized officer
Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476	Arthur Smith 819-953-1360

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CA2008/000545

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US2008063153	13-03-2008	CA2598200 A1	21-02-2008
US2005083911	21-04-2005	EP1526697 A2	27-04-2005
US2005169248	04-08-2005	NONE	
US2008037715	14-02-2008	NONE	
WO2007044454	19-04-2007	US2007091831 A1	26-04-2007
		US2007091906 A1	26-04-2007
		US2007092070 A1	26-04-2007
		WO2007044455 A2	19-04-2007
		WO2007055971 A2	18-05-2007

Lawful Intercept requirements summarized by CISCO

From http://www.faqs.org/rfcs/rfc3924.html :

Note that there are a variety of requirements that have been defined for lawfully authorized intercept throughout the world. Some of these have been defined by standards bodies (e.g., [13]), while others are country specific. The following itemized list is a distillation of some of these, although a given item may or may not apply to a specific country:

* Lawful Intercept (LI) should be undetectable by the intercept subject.

* Mechanisms should be in place to limit unauthorized personnel from performing or knowing about lawfully authorized intercepts.

* There is often a requirement (especially for telecommunications services) to provide intercept related information (IRI) separately from the actual Internet Protocol (IP) traffic (or content) of interest (Note: some authorizations may be restricted to IRI).

* If IRI is delivered separately from content, there should be some means to correlate the IRI and the content with each other.

* If the information being intercepted is encrypted by the service provider and the service provider has access to the keys, then the information should be decrypted before delivery to the Law Enforcement Agency (LEA) or the encryption keys should be passed to the Law Enforcement Agency to allow them to decrypt the information.

* If the information being intercepted is encrypted by the intercept subject and its associate and the service provider has access to the keys, then the service provider may deliver the keys to the LEA.

* There is often a requirement for a service provider to be able to do multiple simultaneous intercepts on a single subject. The fact that there are multiple intercepts should be transparent to the LEAs.

* There is often a requirement that the service provider should not deliver any unauthorized information to the LEA.

Reference Model

This section describes a generic reference model (Figure 1) for lawful intercept.

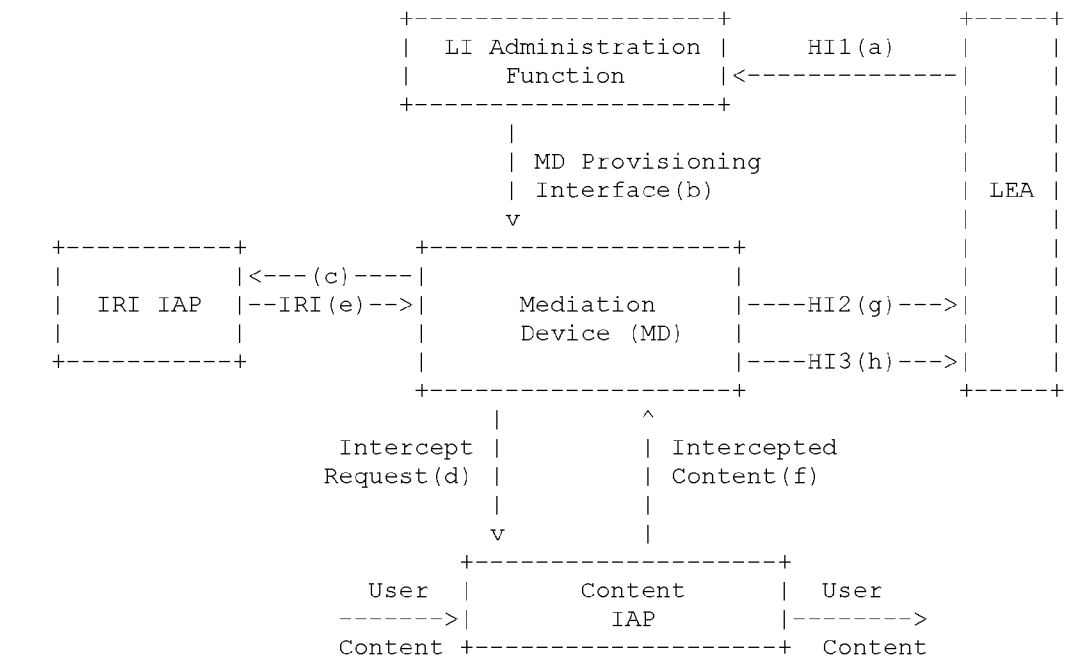


Figure 1: Intercept Architecture

A brief description of the interfaces is included in table 1 below. For a more detailed description of the interfaces refer to section 3.

For a description of the components refer to section 2.1.

Table 1 LI Interfaces

Interface	Description
(a) HI1	Handover Interface 1 - Administration Interface: The LEA provides intercept information to the service provider administration function.
(b) MD Provisioning	Mediation Device provisioning interface. Parameters include: target identifier, duration of intercept, type of intercept, etc.
(c) IRI IAP Provisioning	Specifies Target identifier, duration, etc. for provisioning of delivery of Intercept Related Information (IRI).
(d) Content Intercept Provisioning	Provisioning of the Content IAP.

T-Mobile Exhibit 1002
Part 3 of 6

(e) IRI to MD	Internal interface between IRI Intercept Access Point (IAP) and Mediation device (MD) for delivery of IRI.
(f) Content to MD	Internal interface between content IAP and MD for delivery of Content.
(g) HI2	Handover Interface 2: Interface between the MD and LEA for delivering IRI. This interface may vary from country to country.
(h) HI3	Handover Interface 3: Interface between the MD and LEA for delivering Content. This interface may vary from country to country.

T-Mobile Exhibit 1002
Part 3 of 6

ETSI TS 102 232-5 V2.3.1 (2008-04)

Technical Specification

Lawful Interception (LI);
Handover Interface and
Service-Specific Details (SSD) for IP delivery;
Part 5: Service-specific details for IP Multimedia Services



T-Mobile Exhibit 1002
Part 3 of 6

Reference
RTS/LI-00051-5

Keywords
IMS, IP, lawful interception, security

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2008.
All rights reserved.

DECT™, **PLUGTESTS**™, **UMTS**™, **TIPHON**™, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

T-Mobile Exhibit 1002
Part 3 of 6

Contents

Intellectual Property Rights 5
Foreword 5
Introduction 5
1 Scope 6
2 References 6
2.1 Normative references 6
2.2 Informative references 7
3 Definitions and abbreviations 7
3.1 Definitions 7
3.2 Abbreviations 8
4 General 8
4.1 Reference Model for Lawful Interception 8
4.2 Reference system model 10
4.2.1 Network layer Interception 10
4.2.2 Service layer Interception 11
4.3 General Requirements 11
5 Interception of IP Multimedia services 12
5.1 Identification of target of interception 12
5.1.1 SIP Target Identification 12
5.1.2 H.323 Target Identification 12
5.2 Interception of signalling 13
5.2.1 Provisioning of the SIP IRI IIF 13
5.2.2 Provisioning of the H.323 IRI IIF 13
5.3 Assigning a value to the Communication Identity Number 13
5.3.1 Assigning a CIN value to SIP related IRI 13
5.3.2 Assigning a CIN value to H.323 related IRI 14
5.4 Events and IRI record types 14
5.5 Interception of Content of Communication 15
6 Handover Interface 15
6.1 Intercepted Related Information 15
6.2 Correlation of IRI and CC 15
7 ASN.1 specification for IRI and CC 16
Annex A (informative): Interception using H.248 18
A.1 Purpose of this annex 18
A.2 Notes on interception using H.248 18
A.2.1 Target identification (see also clause 5.1) 18
A.2.2 Provisioning of the H.248 IRI IIF (see also clause 5.2) 19
A.3 Problems in H.248 interception 19
A.3.1 Missing information in H.248 signalling 19
A.3.2 Missing call content 20
Annex B (normative): Minimum set of functional attributes to be provided 21
B.1 General requirements 21
B.2 Result of interception 22
B.3 Location information 22
B.4 Time constraints 22

B.5 Technical handover interfaces and format requirements.....23

Annex C (informative): Change request history.....24

History.....25

T-Mobile Exhibit 1002
Part 3 of 6

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Lawful Interception (LI).

The present document is part 5 of a multi-part deliverable. Full details of the entire series can be found in part 1 [2].

Introduction

The present document focuses on Lawful Interception of IP Multimedia Services. It is to be used in conjunction with TS 102 232-1 [2], in which the handling of the intercepted information is described.

T-Mobile Exhibit 1002
Part 3 of 6

1 Scope

The present document specifies interception of Internet Protocol (IP) Multimedia (MM) Services based on the Session Initiation Protocol (SIP) and Real Time Transport Protocol (RTP) and IP MM services as described by the ITU-T Recommendations H.323 [6] and H.248 [7].

The present document is consistent with the definition of the Handover Interface, as described in TS 102 232-1 [2].

The present document does not override or supersede any specifications or requirements in 3GPP TS 33.108 [9] and TS 101 671 [1].

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1] ETSI TS 101 671: "Lawful Interception (LI); Handover interface for the lawful interception of telecommunications traffic".

NOTE: Periodically TS 101 671 is published as ES 201 671. A reference to the latest version of the TS as above reflects the latest stable content from ETSI/TC LI.

[2] ETSI TS 102 232-1: "Lawful Interception (LI); Handover Interface and Service-Specific Details (SSD) for IP delivery; Part 1: Handover specification for IP delivery".

[3] ITU-T Recommendation X.680: "Information Technology - Abstract Notation One (ASN.1): Specification of basic notation".

[4] IETF RFC 3261: "SIP: Session Initiation Protocol".

[5] IETF RFC 3550: "RTP: A Transport Protocol for Real-Time Applications".

T-Mobile Exhibit 1002
Part 3 of 6

[6] ITU-T Recommendation H.323: "Packet-based multimedia communications systems".

[7] ITU-T Recommendation H.248.1: "Gateway control protocol".

[8] Void.

[9] ETSI TS 133 108: "Universal Mobile Telecommunications System (UMTS); 3G security; Handover interface for Lawful Interception (LI) (3GPP TS 33.108)".

[10] ETSI TS 101 331: "Lawful Interception (LI); Requirements of Law Enforcement Agencies".

[11] ATIS-1000678.2006: "Lawfully Authorized Electronic Surveillance (LAES) for Voice over Packet Technologies in Wireline Telecommunication Networks", Version 2 (revision of ANS T1.678-2004).

[12] ITU-T Recommendation H.225.0: "Call signalling protocols and media stream packetization for packet-based multimedia communication systems".

[13] ITU-T Recommendation H.245: "Control protocol for multimedia communication".

[14] IETF RFC 4566: "SDP: Session Description Protocol".

2.2 Informative references

The following referenced documents are not essential to the use of the present document but they assist the user with regard to a particular subject area. For non-specific references, the latest version of the referenced document (including any amendments) applies.

[15] ETSI TR 102 528: "Lawful Interception (LI); Interception domain Architecture for IP networks".

[16] ETSI TR 102 503: "Lawful Interception (LI); ASN.1 Object Identifiers in Lawful Interception Specifications".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 101 671 [1], TS 102 232-1 [2] and the following apply:

context: logical collection of H.248 terminations

IP MultiMedia service: MultiMedia service that utilizes the Internet Protocol (IP) for the transport of data

MultiMedia (MM): use of computers to present text, graphics, video, animation, and sound in an integrated way

MultiMedia service: communication service that offers Multimedia communication to end-users

termination: entity in H.248 that acts as a source or sink of media. Terminations may be physical, such as a given channel on a TDM line, or ephemeral, such as an IP endpoint

TSAP identifier: piece of information used to multiplex several transport connections of the same type on a single H.323 entity with all transport connections sharing the same Network Address (e.g. the port number in a TCP/UDP/IP environment)

NOTE: Transport layer Service Access Point (TSAP) identifiers may be (pre)assigned statically by some international authority or may be allocated dynamically during the setup of a call. Dynamically assigned TSAP identifiers are of transient nature, i.e. their values are only valid for the duration of a single call.

T-Mobile Exhibit 1002
Part 3 of 6

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AF	Administration Function
ASN.1	Abstract Syntax Notation One
CC	Content of Communication
CC IIF	CC Internal Interception Function
CCCI	Content of Communication Control Interface
CCTF	Content of Communication Trigger Function
CCTI	Content of Communication Trigger Interface
CID	Communication IDentifier
CIN	Communication Identity Number
CLI	Calling Line Identity
CSP	Communications Service Provider (covers all AP/NWO/SvP)
DTMF	Dual Tone Multi Frequency
GW	GateWay
HI1	Handover Interface 1 (for Administrative Information)
HI2	Handover Interface 2 (for Intercept Related Information)
HI3	Handover Interface 3 (for Content of Communication)
IAP	Internet Access Provider
IF	Interception Function
IIF	Internal Interception Function
INI	Internal Network Interface
IP	Internet Protocol
IRI IIF	IRI Internal Interception Function
IRI	Intercept Related Information
LEA	Law Enforcement Agency
LEMF	Law Enforcement Monitoring Facility
LI	Lawful Interception
LIID	Lawful Interception IDentifier
MF	Mediation Function
MG	Media Gateway
MGC	Media Gateway Controller
MM	Multi Media
PDU	Protocol Data Unit
RTCP	RTP Control Protocol
RTP	Realtime Transport Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SvP	Service Provider
TCP	Transmission Control Protocol
TSAP	Transport layer Service Access Point
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

4 General

4.1 Reference Model for Lawful Interception

The present document adopts the generic reference model for the interception domain from TR 102 528 [15], its internal intercept functions, Intercept Related Information Interception Function (IRI IIF), Content of Communication Trigger Function (CCTF), and Content of Communication Internal Interception Function (CC IIF), and the Internal Network Interfaces INI1, INI2, INI3, Content of Communication Trigger Interface (CCTI) and Content of Communication Control Interface (CCCI) as shown in figure 1.

T-Mobile Exhibit 1002
Part 3 of 6

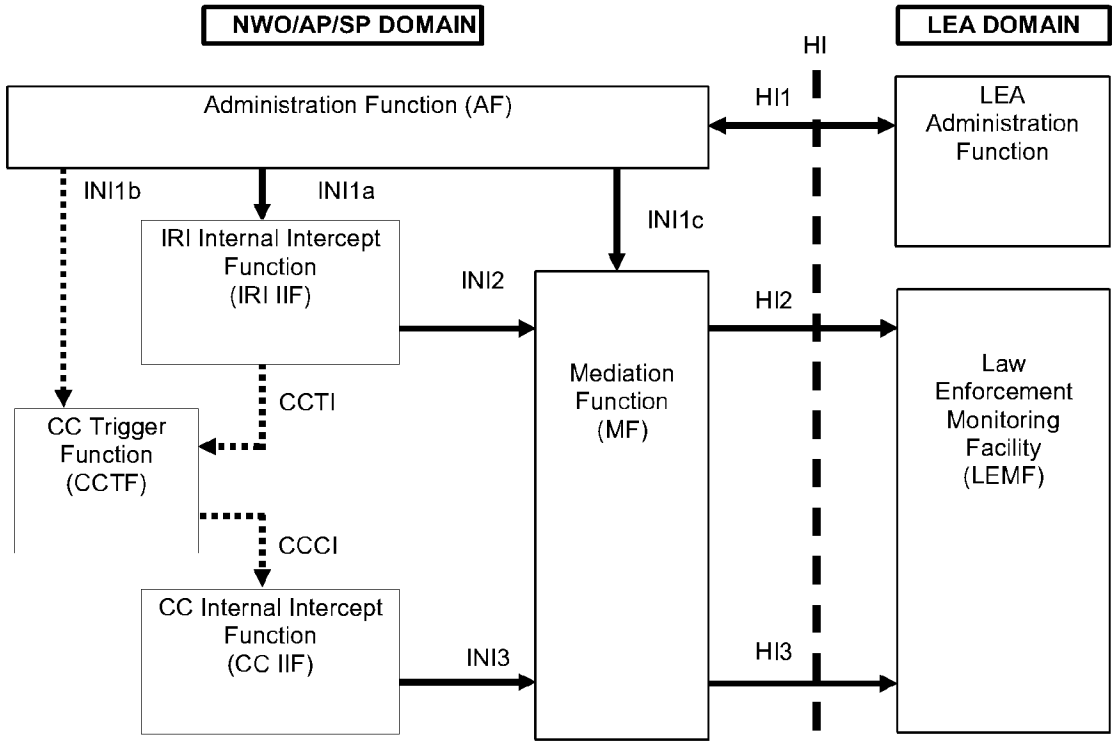


Figure 1: Reference Model for Lawful Interception

The reference model depicts the following functions and interfaces:

- INI1a provisions Intercept Related Information Internal Interception Function (IRI IIF).
- INI1b may (statically) provision Content of Communications Trigger Function (CCTF).
- INI1c provisions the Mediation Function (MF).
- Intercept Related Information Internal Intercept Function (IRI IIF) generates IRI.
- Content of Communication Internal Interception Function (CC IIF) generates CC.
- Content of Communication Trigger Function (CCTF) controls CC IIF.
- Content of Communication Control Interface (CCCI) provisions CC IIF.
- Content of Communication Trigger Interface (CCTI) may trigger CCTF for provisioning of the CC IIF.
- Content of Communication Control Interface (CCCI) may dynamically provision the CC IIF.
- Internal interface INI1 carries provisioning information from LIAF to the Internal Intercept Functions (IIF).
- Internal interface INI2 carries Intercept Related Information (IRI) from IRI IIF to the MF.
- Internal interface INI3 carries Content of Communication (CC) information from CC IIF to the MF.

For an in-depth explanation of the functions and interface, is referred to clause 4 of TR 102 528 [15].

Page 989
T-Mobile Exhibit 1002
Part 3 of 6

4.2 Reference system model

The reference system model applied in the present document, as depicted in figure 2, provides a simplified model of a technology independent, IP Multi Media (MM) service platform, access by multiple different access networks. The access networks may provide different forms of network access, using different technologies; they all have in common that they provide IP connectivity among end-users and between end-users and the IP MM services provided by the IP MM service platform.

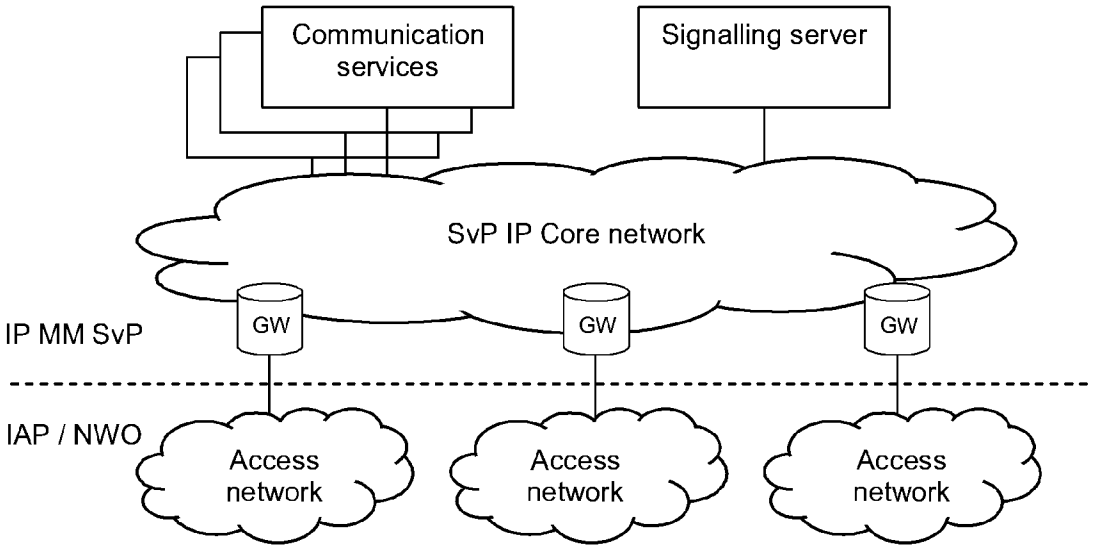


Figure 2: Reference System Model

Access from the access networks into the IP Core network of the IP MM service provider is assumed to be protected by some gateway device (e.g. a session border controller, border gateway controller or a firewall/router combination).

The IP MM Service platform contains a signalling server that provides session initiation functionality (e.g. a SIP call manager or an H.323 gatekeeper) among end-users and between end-users and communication services (e.g. unified messaging, audio or video conference servers).

4.2.1 Network layer Interception

Network Layer interception requires a copy of all signalling information as well as call content exchanged in the platform to be available at a central point in the infrastructure.

Page 990
T-Mobile Exhibit 1002
Part 3 of 6

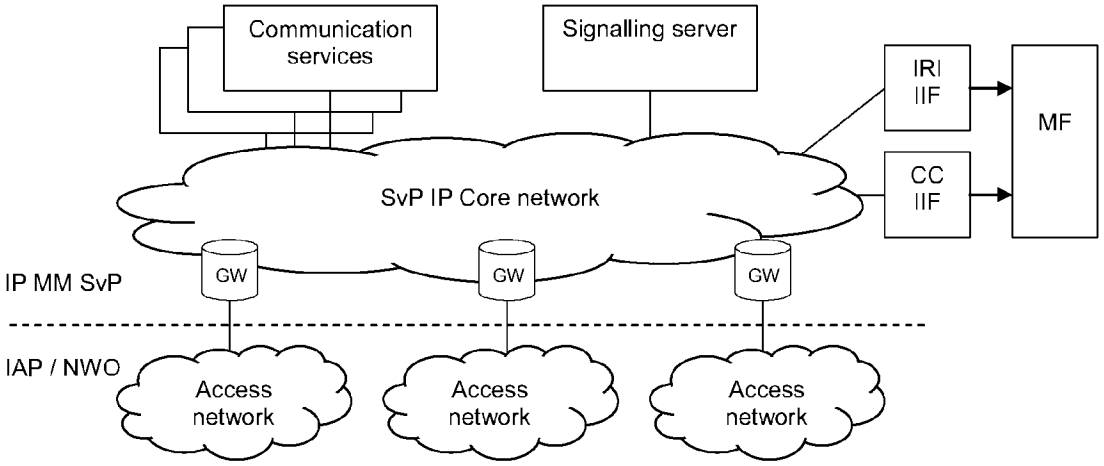


Figure 3: Network layer Interception Model

This can be achieved by means of span-ports in the layer 2 switching backbone or by means of passive splitters (either copper or fibre) at strategic points in the SvP's core network. Either way, due to the bandwidth of copied network traffic, some form of filtering will most likely be required (e.g. by means of a layer 3 switch).

4.2.2 Service layer Interception

Service Layer interception requires LI interfaces and functionality to be available in both the signalling server and the gateways.

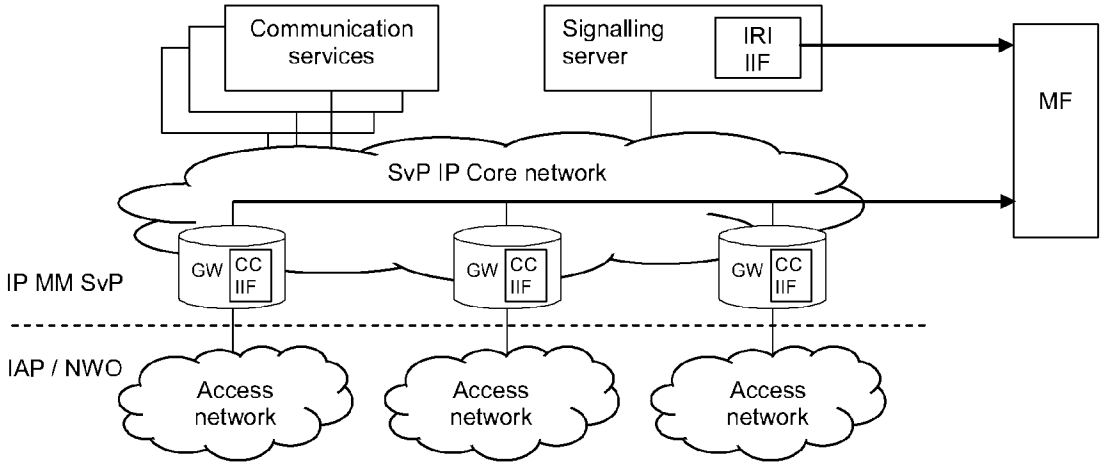


Figure 4: Service layer Interception Model

Typically the IRI IIF in the signalling server is provisioned with the target ID. At detection of a session setup for a target, the IRI IIF will provide the IRI for the intercepted session and may provide session information to be used for ad-hoc provisioning of the gateway devices. In some implementations, the gateway devices are also provisioned with the target ID beforehand and are capable of detecting sessions independent from the signalling server.

4.3 General Requirements

The following requirements regarding the interception of signalling shall apply:

1) Annex B provides the functional description of the minimal set of information that is to be provided to Law Enforcement for each intercepted communication.

T-Mobile Exhibit 1002
Part 3 of 6

2) The present document supports the interception of communication services defined in the following IETF/ITU-T standards and recommendations:

   - RFC 3261[4] (SIP);
   - RFC 3550 [5] (RTP);
   - RFC 4566 [14] (SDP);
   - ITU-T Recommendation H.323 [6];
   - ITU-T Recommendation H.225.0 [12];
   - ITU-T Recommendation H.245 [13].

3) Any deviation from the supported IETF and ITU-T specifications identified in item 2, e.g. vendor specific parameters, shall be agreed in advance between the Communications Service Provider (CSP) and Law Enforcement Agency (LEA).

4) The present document specifies the hand over of intercepted signalling containing all information required in TS 101 331 [10] by encapsulating that intercepted signalling.

5) IRI that is not part of intercepted signalling shall also be delivered. The format of such information on the handover interface shall be agreed in advance between the CSP and LEA.

6) As a national option, mapping of the IRI information onto specific messages at the handover interface may be mandated, e.g. according to the ANS T1.678 [11] specification.

5 Interception of IP Multimedia services

5.1 Identification of target of interception

5.1.1 SIP Target Identification

The target identity is not a network layer or transport layer address. The target identity shall be an address type that uniquely identifies the target in the CSP's network and by means of which sessions among users can be established, such as:

- TEL URI;
- SIP URI;
- E.164 Number.

5.1.2 H.323 Target Identification

The target identity is not a network layer or transport layer address. The target identity shall be an address type that uniquely identifies the target in the CSP's network and by means of which sessions among users can be established, such as:

- H.323 URL;
- H.323 ID;
- E.164 Number.

T-Mobile Exhibit 1002
Part 3 of 6

5.2 Interception of signalling

5.2.1 Provisioning of the SIP IRI IIF

SIP messaging RFC 3261 [4] is reported as Intercept Related Information (IRI) for the interception of multi-media service. All SIP messages executed on behalf of a target subscriber are subject to interception at the IRI Internal Interception Function (IIF). Based upon network configuration, the Administration Function (AF) shall provision IRI IIF with SIP Uniform Resource Identifier (URI) or TEL Uniform Resource Locator (URL) target identifiers. These resulting intercepted SIP messages shall be sent to the Mediation Function (MF) over the INI2 interface for mediation prior to transmittal across the HI2 interface.

5.2.2 Provisioning of the H.323 IRI IIF

H.323 call signalling, call control and subscriber controlled input messages are reported as Intercepted Related Information (IRI) for the interception of multi-media services. H.323 call signalling and control messages refer to the basic call signalling (H.225.0), call control (H.245) and those messages required for the signalling of supplementary services (i.e.: H.450.x). Subscriber controlled input messages refer to those messages generated as a result of user procedures for the control of Supplementary Services (activation/deactivation/interrogation).

All H.323 call signalling, call control and subscriber controlled input messages that are transmitted on behalf of the target subscriber are subject to intercept at the IRI IIF. Based upon the network configuration, the AF shall provision IRI IIF with either a H.323 Unique Resource Locator (H.323-URL), or a H.323 Identity (H.323-ID), or a public E.164 telephone number.

If available events related to the Registration, Administration and Status (i.e. H.323 RAS) of the target subscriber's terminal equipment are also subject to intercept at the IRI IIF.

5.3 Assigning a value to the Communication Identity Number

In order to produce useful IRI records from events, the IRI records of a communication session must be correlated with a single value for the Communication Identity Number (CIN) field.

5.3.1 Assigning a CIN value to SIP related IRI

All IRI events resulting from SIP messages in a single call signalling leg (i.e. with an identical value for the SIP field "Call-ID") will be assigned the same value for the CIN. A call may consist of two or more call signalling legs (e.g. when communicating via a SIP proxy). The various related call signalling legs are matched using the O field in the RFC 4566 [14] SDP part of the SIP messages; call legs with an identical O field will be assigned the CIN value of the active session from which they are spawned.

This is depicted in figure 5, where *all* resulting IRI-Event are assigned the same CIN.

T-Mobile Exhibit 1002
Part 3 of 6

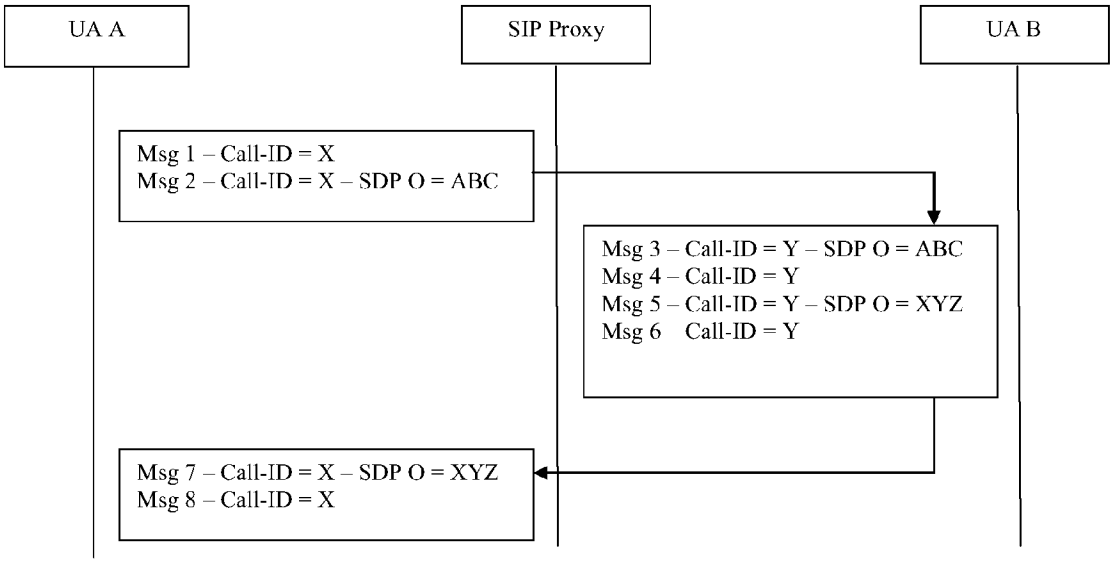


Figure 5: Relation between call signalling legs

5.3.2 Assigning a CIN value to H.323 related IRI

All IRI events resulting from messages within a single H.323 call will be assigned the same value for the CIN. Messages within a H.323 call can be identified as those messages containing, or associated with the same unique H.225.0 Call ID. The term H.323 Call is defined in the ITU-T Recommendation H.323 [6]. The term Call ID also referred to as H.225.0 Call ID and is described in the ITU-T Recommendations H.323 [6] and H.225.0 [12].

5.4 Events and IRI record types

The following requirements need to be met:

- All multimedia signalling and control messages to or from a targeted subscriber, and all multimedia signalling and control messages executed on behalf of or related to a targeted subscriber are intercepted by the IRI IIF and sent to the MF over the INI2 interface as IRI records.
- IRI records for all events related to a single communication session, which may consist of multiple media streams, that are being exchanged among the possibly more than two users, will be assigned the same CIN value.
- In addition, information on non-transmission related actions of a target constitute IRI is assigned CIN and is sent via HI2 (e.g. information on subscriber controlled input).

NOTE: This includes events related to the target service handled internally by network functions (e.g. communications among GWs, Signalling Servers and Communication Services). The format of such information on the handover interface should be agreed in advance between the CSP and LEA.

For each event, a record is sent to the LEMF. A new value for the CIN field is assigned if an event is detected that is not related to an already existing communication session. This event is reported as an IRI-REPORT or IRI-BEGIN record. Every subsequent record related to this communication session is reported as an IRI-REPORT or IRI-CONTINUE record. At the detection of the event that denotes the end of a session an IRI-REPORT or IRI-END record is sent.

In case of implementation issues, operational flaws or problems, events may need to be reported for a communication session for which the IRI-END record already has been sent. These events are reported as an IRI-REPORT with the proper value for the CIN.

Table 1 summarizes the mapping between event type and record type sent to the LEMF.

T-Mobile Exhibit 1002
Part 3 of 6

Table 1: Mapping between IP MM Events and HI2 Records Type

Event	IRI Record Type
At assignment of a new CIN value	BEGIN/REPORT
All intermediate signalling, other than the last event	CONTINUE
The last event related to a communication session	END
Delayed IRI events related to an already ended session	REPORT
Events that are not mapped	REPORT
NOTE: Not mapped events could for example be encapsulated SIP messages.	

5.5 Interception of Content of Communication

The Communication Content (CC) provided is a copy of the multimedia stream, i.e. every RFC 3550 [5] RTP and RTCP packet sent through the SvP's network that is addressed to, sent from or related to the targeted subscriber as described in the intercepted call signalling and call control information. The CC shall also contain the RTP header, UDP header and IP header, except by agreement between CSP and LEA (for example these headers may not be available at the point of interception).

The frameType field indicates which headers are present in a given CC stream.

In the case where the RTP header is unavailable, one may be inserted by the mediation function, subject to agreement between LEA and CSP. The addition of an inserted RTP header may aid processing the audio stream at the receiver.

IP and UDP headers shall not be inserted to the intercepted material by the mediation function if they are unavailable.

In addition to IRI messages, all RTP and RTCP packets identified to be associated with a specific intercepted call or session shall be assigned the same CIN.

In case where some headers are not reported within the CC (because not available at the point of interception) there may be a need to identify the media streams inside the CC, typically in situations where the SIP communication consists in multiple media streams of same type. The streamIdentifier field identifies the media stream unambiguously (it may contain the corresponding SDP 'c=' and 'm=' lines for instance).

NOTE 1: The ASN.1 definition for CC is presented as the iPMultimediaPDU in clause 7.

NOTE 2: The CC also contains the target media streams that are not transferred via RTP or RTCP (for example, those streams handled by a GW and not passing the core network). The format of such information on the handover interface should be agreed in advance between the CSP and LEA.

6 Handover Interface

6.1 Intercepted Related Information

The Communication Identity Number (CIN) is used to uniquely identify a communication session (as described in TS 102 232-1 [2] and TS 101 671 [1]). Applied to IP MM Services, a communication session refers to a single self contained transaction or a series of protocol data units that together form a self-contained communication such as a SIP or H.323 Session.

6.2 Correlation of IRI and CC

To assure correlation between the independently transmitted Content of Communication and the Intercepted Related Information (IRI) of an intercepted call the following parameters are used:

- Lawful Interception Identifier (LIID);
- Communication Identifier (CID).

T-Mobile Exhibit 1002
Part 3 of 6

NOTE: The target identifier may not necessarily be found in both communication content and interception related information.

In situations where the SIP communication consists of multiple media streams the streamIdentifier field of CC may be used to correlate each media stream of CC with the corresponding SDP media description of IRI.

7 ASN.1 specification for IRI and CC

The ASN.1 (ITU-T Recommendation X.680 [3]) module that represents the information in the present document and meets all stated requirements is shown below. TR 102 503 [16] gives an overview of the relevant Object IDentifiers (OID) used in ASN.1 modules of the Lawful Intercept specifications and points to the specification where the modules can be found.

```
-- ====================================
-- Description of the IP Multimedia PDU
-- ====================================

IPMultimediaPDU
{itu-t(0) identified-organization(4) etsi(0) securityDomain(2) lawfulIntercept(2) li-ps(5)
iPMultimedia(5) version3(3)}

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

IMPORTS
    -- from TS 101 671 [1]
    IPAddress
        FROM HI2Operations
        {itu-t(0) identified-organization(4) etsi(0) securityDomain(2) lawfulIntercept(2) hi2(1)
version10(10)};

-- Object Identifier Definition

iPMMIRIObjId RELATIVE-OID          ::= {li-ps(5) iPMultimedia(5) version3(3) iRI(1)}
iPMMCCObjId RELATIVE-OID           ::= {li-ps(5) iPMultimedia(5) version3(3) cC(2)}
    -- both definitions relative to:
    -- {itu-t(0) identified-organization(4) etsi(0) securityDomain(2) lawfulIntercept(2)}

-- ------------------------------------
-- IP Multimedia Communications Contents
-- ====================================

IPMMCC      ::= SEQUENCE
{
    iPMMCCObjId                 [0] RELATIVE-OID,
    rTPCCContents               [1] OCTET STRING,
        -- Copy of the multimediastream, i.e. all related RTP and RTCP packets
    ...,
    frameType                   [2] FrameType OPTIONAL,
        -- The availability of header information shall be signalled with the frameType parameter
        -- The module is defined as OPTIONAL because of backwards compatibility reasons
        -- For new implementations the module is MANDATORY to be used as defined in clause 5.5
    streamIdentifier            [3] OCTET STRING OPTIONAL
        -- Used to identify the media stream within the current CIN, typically in case of
        -- multiple media streams communications
        -- May be used to correlate each media stream with the relevant SDP media description of IRI
        -- May contain c= and m= lines extracts for instance
}

FrameType   ::= ENUMERATED
{
    ipFrame(0),
        -- All headers are present
    udpFrame(1),
        -- IP header is missing
    rtpFrame(2),
        -- UDP and IP headers are missing
    audioFrame(3),
        -- All headers are missing
    ...
}
```


T-Mobile Exhibit 1002
Part 3 of 6

```
-- ========================================================
-- Intercept-related information for IP Multimedia sessions
-- ========================================================

IPMMIRI         ::= SEQUENCE
{
    iPMMIRIObjId                [0] RELATIVE-OID,
    iPMMIRIContents             [1] IPIRIContents,
    ...
}

IPIRIContents   ::= CHOICE
{
    originalIPMMMessage         [0] OCTET STRING,
        -- Copy of the IP MM signalling packet including the original IP and UDP/TCP headers
    sIPMessage                  [1] SIPMessage,
        -- Copy of the SIP content and the source and destination IP address
    h323Message                 [2] H323Message,
        -- Copy of the H.323 content and the source and destination IP address
    ...

}

SIPMessage      ::= SEQUENCE
{
    iPSourceAddress             [0] IPAddress,
    iPDestinationAddress        [1] IPAddress,
    sIPContent                  [2] OCTET STRING,
    ...
}

H323Message     ::= SEQUENCE
{
    iPSourceAddress             [0] IPAddress,
    iPDestinationAddress        [1] IPAddress,
    h323Content                 [2] H323MessageContent,
    ...
}

H323MessageContent      ::= CHOICE
{
    h225CSMessageContent        [0] OCTET STRING,
    h225RASMessageContent       [1] OCTET STRING,
    h245MessageContent          [2] OCTET STRING,
    genericMessageContent       [3] OCTET STRING,
    ...
}

END -- end of IP Multimedia PDU
```


T-Mobile Exhibit 1002
Part 3 of 6

Annex A (informative): Interception using H.248

A.1 Purpose of this annex

There are a number of problems with extracting useful IRI from H.248 signalling. These problems are largely due to the fact that H.248 is a gateway control protocol, not an end-to-end call initiation protocol like SIP or H.323. Often, useful information may be more easily extracted from other signalling links such as SIP, or from on-switch interception solutions. However, in certain circumstances it may be necessary to use the H.248 link. For example, for security or architecture reasons, it may be that the simpler options are not possible or acceptable. This annex provides some notes on how LI could be performed on a H.248 link (clause A.2) and describes some of the problems and issues that arise (clause A.3). It is not intended to be a complete or comprehensive description and solutions to the problems and issues are not presented.

A.2 Notes on interception using H.248

A.2.1 Target identification (see also clause 5.1)

H.248 messages typically do not contain authoritative target identifiers, such as E.164 phone numbers. Instead, they will typically contain termination and context identifiers. Hence, a probe provisioned with a list of target identifiers, and sniffing an H.248 link, will not be able to discriminate between target and non-target H.248 messages.

A possible solution would be to provide a mapping between authoritative identifiers such as E.164 numbers, and H.248 identifiers such as termination IDs. The MGC would provide the IF or MF with a list of the termination IDs in each MG it controls, and the corresponding authoritative identifier for each termination. This list of correspondences, along with the target list held by the IF or MF, would allow the IF to target H.248 transactions based on termination ID.

Information such as the IP endpoint for the RTP stream will be sent to a different termination. However, it will be in the same context as the target termination. Hence, it will be necessary for the probe to extract the relevant Context ID from any messages it intercepts by Termination ID, and intercept subsequent messages by this Context ID.

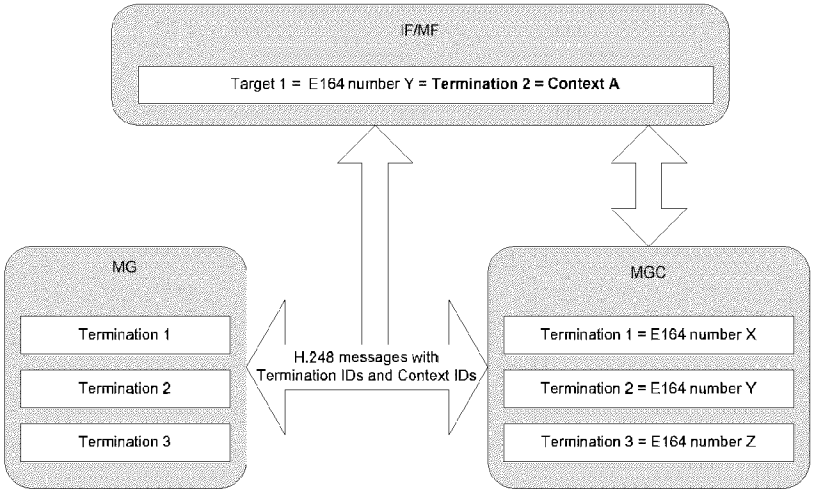


Figure A.1: Mapping of Authoritative Identifiers to Termination and Context Identifiers

T-Mobile Exhibit 1002
Part 3 of 6

There is still a problem with this method. There is no guarantee that the first termination to be added to a context will be the target physical termination. For example, the MGC may choose to add an ephemeral IP termination first, followed by the target termination. The IF will only intercept signalling after the target termination is added. Information contained in the first Add message is missed. This message may contain the IP endpoints and codec for the media stream, and so missing it means that the call content cannot be intercepted.

It is worth noting that typically, there is no advantage to adding the ephemeral termination first, and so H.248 implementations tend to add the physical termination at the same time as, or before, the ephemeral one.

The recommendation is therefore that:

- The IF will intercept H.248 signalling using a physical termination ID as an identifier.
- The IF will subsequently intercept any H.248 signalling to the same context as the physical termination.
- The MGC provides a mapping of authoritative identifiers to physical termination IDs.

The problems with this approach are:

- Any signalling to a context sent before a target termination is added to that context may be missed. This may mean important information such as details of the call content media stream are missed.

Other methods of identifying target H.248 transactions (for example, examining some other signalling link) are also possible.

A.2.2 Provisioning of the H.248 IRI IIF (see also clause 5.2)

For H.248, the IRI IIF is provisioned with a mapping between H.248 target identifiers and physical termination IDs. All H.248 signalling to and from a targeted termination is subject to interception. In addition, to retrieve information about the subsequent media path, any signalling sent to or from the same context as a targeted termination is also subject to interception.

A.3 Problems in H.248 interception

A.3.1 Missing information in H.248 signalling

In some circumstances, target H.248 messages may not contain all of the information an end user might expect. The following list shows what information may be missing:

- If a non-target calls a target, the non-target's number may be unavailable. A CLI signal may be used to discover the caller's number, but if there is no CLI package in place, or the user withholds their number, the calling number is unavailable.
- A failed attempt to call the target (because, for example, the target was busy) may not generate any target signalling, as the MGC will determine that the call cannot be made.
- If a target is called, the H.248 signalling gives no indication whether the call was made directly, or if the call was, for example, diverted or transferred to the target. If the call was diverted, there is no way of obtaining the original number (unless it is provided in some kind of package).
- Similarly, when a call is torn down, unless the target hung up, there is no indication of why the call ended.
- In some implementations, features such as call-holding or conference calling may be dealt with by a separate dedicated server (not the MGC). For example, a user may flash-hook, and be automatically redirected to a dedicated conferencing server, which would subsequently listen for the DTMF digits of the person the user wants to conference with. In this case, the digits the user dials are carried over the media stream to the conferencing server, rather than over H.248 signalling to the MGC. This number would only be found by analysing the call content afterwards.

Page 999 T-Mobile Exhibit 1002
Part 3 of 6

A.3.2 Missing call content

Local call turnaround creates a problem in capturing the call media. If the call is turned around locally at a Media Gateway, then the RTP media stream is never presented to the IP network. Therefore no interception function in the IP network will be able to intercept the call media.

Page 1000 T-Mobile Exhibit 1002
Part 3 of 6

Annex B (normative): Minimum set of functional attributes to be provided

This annex provides the functional description of the minimal set of information that is to be provided to Law Enforcement for each intercepted communication.

The full law enforcement requirements are in TS 101 331 [10].

The present annex describes the requirements from a Law Enforcement Agency's (LEA's) point of view.

Not all requirements necessarily apply in one individual nation.

These requirements shall be used to derive specific network requirements and furthermore to standardize handover interfaces.

B.1 General requirements

a) The obligation of the CSP as to which telecommunications traffic shall be intercepted is subject to national laws.

b) In accordance with the relevant lawful authorization a CSP shall ensure that:

b.1) the entire content of communication associated with a target identity being intercepted can be intercepted during the entire period of the lawful authorization;

b.2) any content of communication associated with a target identity being intercepted which is routed to technical storage facilities or is retrieved from such storage facilities can be intercepted during the entire period of the lawful authorization;

b.3) the delivery of the intercept related information is reliable. If the intercept related information can not be delivered immediately to the relevant LEMF, then the intercept related information shall be buffered until they can be delivered;

b.4) the delivery of the content of communication is reliable. If the content of communication can not be delivered immediately to the relevant LEMF, then the content of communication shall be buffered if this is required by national laws;

b.5) the CSP shall not monitor or permanently record the results of interception.

c) The ability to intercept telecommunications shall be provided relating to the interception subjects operating permanently within a telecommunications system (e.g. a subscriber or account).

d) The ability to intercept telecommunications shall be provided relating to the interception subjects operating temporarily within a telecommunications system (e.g. a visiting mobile subscriber or a visiting subscriber using an access network to a home service).

e) The results of interception relating to a target service shall be provided by the CSP in such a way that any telecommunications that do not fall within the scope of the lawful authorization shall be excluded by the CSP.

NOTE 1: It is assumed that the intercepting system exercises best effort to exclude non-authorized interception patterns (e.g. transferred communication).

f) All results of interception provided at the handover interface shall be given a unique identification relating to lawful authorization.

NOTE 2: Information used for the IRI is expected to be part of standard network signalling procedures. No additional signalling is expected for the IRI.

Page 1001 T-Mobile Exhibit 1002 Part 3 of 6

B.2 Result of interception

The CSP shall, in relation to each target service:

a) Provide the content of communication.

b) Remove any service coding or encryption which has been applied to the content of communication (i.e. en clair) and the intercept related information at the instigation of the CSP.

c) Provide the LEA with any other decryption keys whose uses include encryption of the content of communication, where such keys are available for CSP.

d) Intercept related information shall be provided:

d.1) when communication is attempted;

d.2) when communication is established;

d.3) when no successful communication is established;

d.4) on change of status (e.g. in the access network);

d.5) on change of service or service parameter;

d.6) on change of location (this can be related or unrelated to the communication or at all times when the apparatus is switched on).

e) Intercept related information shall contain:

e.1) the identities that have attempted telecommunications with the target identity, successful or not;

e.2) identities used by or associated with the target identity;

e.3) details of services used and their associated parameters;

e.4) information relating to status;

e.5) time stamps.

f) The conditions mentioned above also apply to multi-party or multi-way telecommunication if and as long as the target identity participates.

B.3 Location information

An LEA may request location information relating to locations, in a number of forms: the current geographic, physical or logical location of the target identity. This information is expected to be made available from normal network operation.

B.4 Time constraints

a) A CSP shall make the necessary arrangements to fulfil his obligation to enable the interception and delivery of the result of interception from the point in time when the telecommunication installation commences commercial service.

b) The above requirement applies accordingly to the introduction of modifications to the telecommunication installation or to new operational features for existing telecommunications services to the extent of their impact on existing interception capabilities.

T-Mobile Exhibit 1002
Part 3 of 6

c) When a lawful authorization is presented a CSP provider shall co-operate immediately.

d) After a lawful authorization has been issued, provision of the results of interception of a target identity shall proceed on a real-time basis.

B.5 Technical handover interfaces and format requirements

a) The configuration of the handover interface shall ensure that the quality of service of the telecommunications traffic provided at the handover interface is not inferior to that offered to the target service for each particular call.

b) Each interception target shall be uniquely associated with a single instance of the handover interface. This could be achieved by separate channels or the use of identifiers.

c) The correlation between the content of communication and intercept related information shall be unique.

Page 1003

T-Mobile Exhibit 1002
Part 3 of 6

Annex C (informative): Change request history

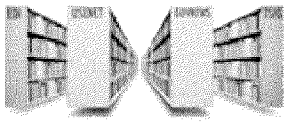
Status of present document: TS 102 232-5 Handover Interface and Service-Specific Details (SSD) for IP delivery; Part 5: Service-specific details for IP Multimedia Service; Handover specification for IP delivery		
Date	**Version**	**Remarks**
January 2007	2.1.1	First publication of the TS after approval by ETSI/TC LI#14 (30 January - 1 February 2007, Puerto de la Cruz). Version 2.1.1 prepared by Johan Bakker (KPN) (rapporteur V2.1.1)
October 2007	2.2.1	Included Change Requests: TS102232-05CR001 (cat D) IRI Record type TS102232-05CR002r2 (cat B) Clarification of use of RTP/UDP/IP headers These CRs were approved TC LI#16, 2-4 October 2007; Berlin version 2.2.1 prepared by Peter van der Arend (KPN) (chairman TC LI)
December 2007	2.2.2	Figure 5 was updated.
January 2008	2.3.1	Included Change Request: TS102232-05CR003r1 (cat B) on Media stream identification for correlation in case of multiple media streams communications This CR was approved TC LI#17, 22-24 January 2008, Como version 2.3.1 prepared by Peter van der Arend (KPN) (chairman TC LI)
Rapporteur of the present document is Mark Lastdrager (Pine Digital Security)		

Page 1004
T-Mobile Exhibit 1002
Part 3 of 6

History

Document history		
V2.1.1	February 2007	Publication
V2.2.1	November 2007	Publication
V2.2.2	December 2007	Publication
V2.3.1	April 2008	Publication

Page 1005 T-Mobile Exhibit 1002
Part 3 of 6



RFC 3924 (RFC3924)

Internet RFC/STD/FYI/BCP Archives

[RFC Index | RFC Search | Usenet FAQs | Web FAQs | Documents | Cities]

Alternate Formats: rfc3924.txt | rfc3924.txt.pdf

Comment on RFC 3924

RFC 3924 - Cisco Architecture for Lawful Intercept in IP Networks

---

Network Working Group F. Baker
Request for Comments: 3924 B. Foster
Category: Informational C. Sharp
Cisco Systems

October 2004

Cisco Architecture for Lawful Intercept in IP Networks

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2004).

IESG Note

This RFC is not a candidate for any level of Internet Standard. The IETF disclaims any knowledge of the fitness of this RFC for any purpose, and in particular notes that the decision to publish is not based on IETF review for such things as security, congestion control or inappropriate interaction with deployed protocols. The RFC Editor has chosen to publish this document at its discretion. Readers of this document should exercise caution in evaluating its value for implementation and deployment.

T-Mobile Exhibit 1002
Part 3 of 6

Abstract

For the purposes of this document, lawful intercept is the lawfully authorized interception and monitoring of communications. Service providers are being asked to meet legal and regulatory requirements for the interception of voice as well as data communications in IP networks in a variety of countries worldwide. Although requirements vary from country to country, some requirements remain common even though details such as delivery formats may differ. This document describes Cisco's Architecture for supporting lawful intercept in IP networks. It provides a general solution that has a minimum set of common interfaces. This document does not attempt to address any of the specific legal requirements or obligations that may exist in a particular country.

Table of Contents

1. Introduction. 2
   1.1. Requirements Motivating the Architecture 3
   1.2. Document Organization. 4
2. Reference Model 5
   2.1. Reference Model Components 6
   2.2. Operational Considerations 7
3. Interfaces. 9
   3.1. Content Intercept Request Interface. 9
   3.2. Intercept Content Interface (f). 10
4. Applying the Reference Model. 11
   4.1. Voice over IP networks 11
      4.1.1. Interception of Voice over IP Services. 11
      4.1.2. Local Voice Services. 12
   4.2. Data Services. 13
5. Security Considerations 13
   5.1. Content Request Interface (d) - SNMPv3 Control 14
6. Informative References. 14
7. Acronyms. 16
8. Authors' Addresses. 17

T-Mobile Exhibit 1002
Part 3 of 6

9. Full Copyright Statement. 18

1. Introduction

For the purposes of this document, lawful intercept is the lawfully authorized interception and monitoring of communications of an intercept subject. The term "intercept subject", "subject", "target subscriber" or "target" in this document refers to the subscriber of a telecommunications service whose communications and/or intercept related information (IRI) has been lawfully authorized to be intercepted and delivered to some agency. Note that although the term "Law Enforcement Agency" (LEA) is used throughout this document, this may refer to any agency that is able to request lawfully authorized interception.

By intercept related information (IRI) we mean information related to the IP traffic of interest. There is currently no standardized definition for IRI for IP traffic. IRI has been defined for a few services that might run over IP (e.g., Voice over IP) or that IP runs on top of (e.g., GPRS). For example, IRI for voice over IP could be the called and calling phone numbers. The definition of IRI from [14] is shown below:

> Intercept Related Information: collection of information or data associated with telecommunication services involving the target identity, specifically communication associated information or data (e.g., unsuccessful communication attempts), service associated information or data and location information.

Service providers are being asked to meet legal and regulatory requirements for the interception of voice as well as data communications in IP networks in a variety of countries worldwide. Although requirements vary from country to country, some requirements remain common even though details such as delivery formats may differ. This document describes Cisco's Architecture for supporting lawful intercept in IP networks. It provides a general solution that has a minimum set of common interfaces. This document does not deal with legal requirements or obligations.

This document describes one method for supporting lawful intercept. Other methods may be available.

The IESG wishes to draw the reader's attention to RFC 2804 [15] for a description of why architectures such as these are vendor-specific, rather than a topic of standardization for the IETF.

1.1. Requirements Motivating the Architecture

T-Mobile Exhibit 1002
Part 3 of 6

The purpose of the following list of requirements is to provide an understanding of the motivation behind the architecture and some of the requirements imposed on components and interfaces that are described in the later sections of the document. This does not imply any legal requirements on service providers or equipment vendors although such requirements may coincide.

Note that there are a variety of requirements that have been defined for lawfully authorized intercept throughout the world. Some of these have been defined by standards bodies (e.g., [13]), while others are country specific. The following itemized list is a distillation of some of these, although a given item may or may not apply to a specific country:

* Lawful Intercept (LI) should be undetectable by the intercept subject.

* Mechanisms should be in place to limit unauthorized personnel from performing or knowing about lawfully authorized intercepts.

* There is often a requirement (especially for telecommunications services) to provide intercept related information (IRI) separately from the actual Internet Protocol (IP) traffic (or content) of interest (Note: some authorizations may be restricted to IRI).

* If IRI is delivered separately from content, there should be some means to correlate the IRI and the content with each other.

* If the information being intercepted is encrypted by the service provider and the service provider has access to the keys, then the information should be decrypted before delivery to the Law Enforcement Agency (LEA) or the encryption keys should be passed to the Law Enforcement Agency to allow them to decrypt the information.

* If the information being intercepted is encrypted by the intercept subject and its associate and the service provider has access to the keys, then the service provider may deliver the keys to the LEA.

* There is often a requirement for a service provider to be able to do multiple simultaneous intercepts on a single subject. The fact that there are multiple intercepts should be transparent to the LEAs.

* There is often a requirement that the service provider should not deliver any unauthorized information to the LEA.

The architecture and interfaces described in this document attempts to address these requirements.

T-Mobile Exhibit 1002
Part 3 of 6

1.2. Document Organization

Section 1 of this document lists requirements motivating the architecture. Section 2 of this document describes a reference model along with some operation considerations. Section 3 provides more detailed requirements on the interfaces related to content interception. Section 4 applies the reference model to voice over IP and data intercepts and Section 5 examines security considerations.

2. Reference Model

This section describes a generic reference model (Figure 1) for lawful intercept.

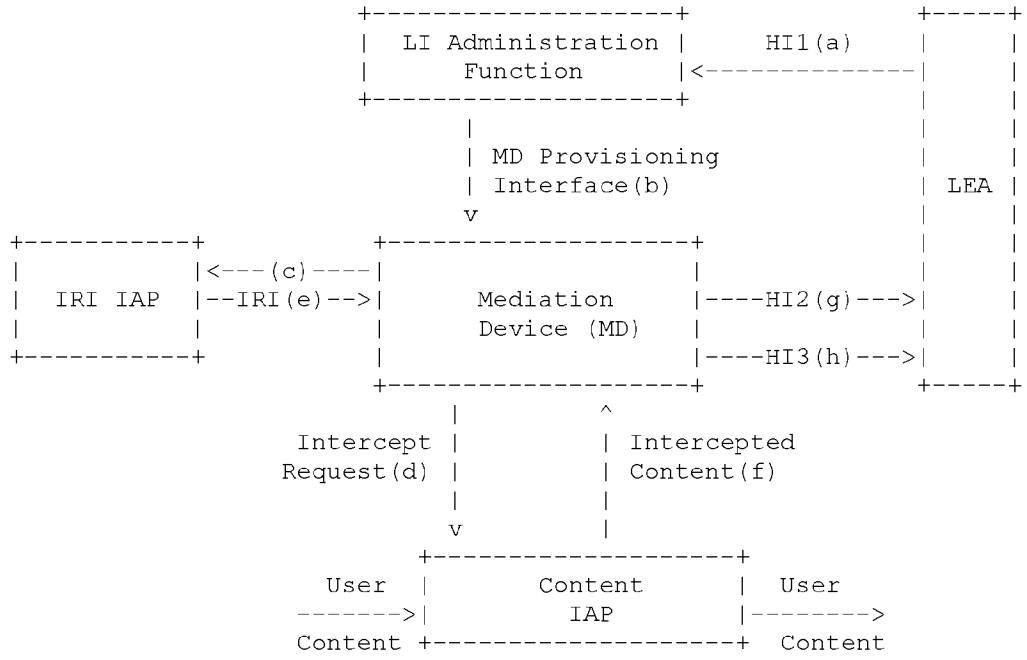


Figure 1: Intercept Architecture

A brief description of the interfaces is included in table 1 below. For a more detailed description of the interfaces refer to section 3.

For a description of the components refer to section 2.1.

Table 1 LI Interfaces

Interface	Description
(a) HI1	Handover Interface 1 - Administration Interface: The LEA provides intercept information to the service provider administration function.

T-Mobile Exhibit 1002
Part 3 of 6

(b)	MD Provisioning	Mediation Device provisioning interface. Parameters include: target identifier, duration of intercept, type of intercept, etc.
(c)	IRI IAP Provisioning	Specifies Target identifier, duration, etc. for provisioning of delivery of Intercept Related Information (IRI).
(d)	Content Intercept Provisioning	Provisioning of the Content IAP.
(e)	IRI to MD	Internal interface between IRI Intercept Access Point (IAP) and Mediation device (MD) for delivery of IRI.
(f)	Content to MD	Internal interface between content IAP and MD for delivery of Content.
(g)	HI2	Handover Interface 2: Interface between the MD and LEA for delivering IRI. This interface may vary from country to country.
(h)	HI3	Handover Interface 3: Interface between the MD and LEA for delivering Content. This interface may vary from country to country.

2.1. Reference Model Components

A brief description of the key components in the reference model is as follows:

Lawful Intercept (LI) Administration Function:
This function provides the (typically manual) provisioning interface for the intercept as a result of a court order or warrant delivered by the Law Enforcement Agency (LEA). It could involve separate provisioning interfaces for several components, but more typically is a single interface to the Mediation Device (MD), which then takes care of provisioning of other components in the network. Because of the requirement in some laws to limit accessibility to authorized personnel, the provisioning interface has to be strictly controlled. In many cases, the identity of the subject received from the LEA has to be translated into an identity that can be used by the network to enable the intercept.

Intercept Access Point (IAP):
An IAP is a device within the network that is used for intercepting lawfully authorized intercept information. It may be an existing device that has intercept capability or it could be a special device that is provided for that purpose. Two types of IAP's are discussed here: IAP's that provide content; and IAP's

T-Mobile Exhibit 1002
Part 3 of 6

that provide intercept related information (IRI).

Content IAP:
A content IAP is an IAP that is used to intercept the IP traffic of interest.

IRI IAP: This is an IAP that is used to provide intercept related information (IRI).

Law Enforcement Agency (LEA):
This is the agency that has requested the intercept and to which the service provider delivers the information.

Mediation Device (MD):
The MD requests intercepts from IAPs through interfaces (c) and (d) in Figure 1. The Mediation Device receives the data from the IAP, packages it in the correct format (which may vary from country to country) and delivers it to the LEA. In the case where multiple law enforcement agencies are intercepting the same subject, the mediation device may replicate the information multiple times. The assumption is that the service provider operates the MD (via specially authorized personnel) and that the LEA only has access to interfaces (a), (g) and (h) in Figure 1.

2.2. Operational Considerations

In a typical operation, a lawfully authorized surveillance request arrives for a specified intercept subject. Authorized personnel provision the intercept using interface (b) in Figure 1, which may be for content only, IRI only or both. Once the intercept is provisioned, the IAP's send the IRI and/or content to the MD, which formats the information into the appropriate format for delivery to the LEA. Some operational issues that need to be considered:

* Location and Address Information for Content Intercepts: In some cases where the location and/or addressing information for the intercept is not known until the subject registers (or makes a call in the case of voice), the IRI may provide needed information in order to do the content tap (e.g., the IP address and port for the content streams).

* Content Encryption: If the intercept content is encrypted and the service provider has access to the encryption keys (e.g., receives keys in Session Description Protocol for Voice over IP), then the keys can be sent via IRI. It is, however, possible for end-users to exchange keys by some other means without any knowledge of the service provider in which case the service provider will not be able to provide the keys. Content transformations could make decryption at the LEA impossible. This is why the original packets are provided on interface (f) rather than attempting to convert them to some other format.

T-Mobile Exhibit 1002
Part 3 of 6

* Detection by the Intercept Subject: One requirement is to ensure that the intercept subject is unable to detect that they are being intercepted. This document assumes a sophisticated subject:

  - Able to check IP addresses, use traceroute, etc.

  - Able to check if any unusual signaling is occurring on their customer premises equipment (CPE).

  - Able to detect degradation or interruptions in service.

  This is why the intercept mechanism described here does not involve special requests to the CPE, re-routing of packets or end-to-end changes in IP addresses. Instead, content intercept is done on a device along the normal content path (i.e., no re-routing has occurred) that is within the service provider's network. A convenient content IAP is a router or switch at the edge of the service provider's network to which the intercept subject connects. This is illustrated in Figure 2.

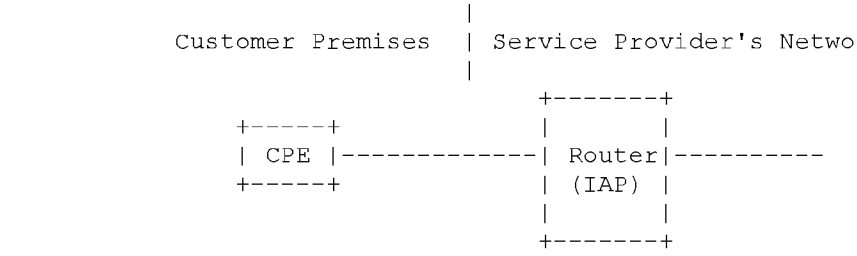


Figure 2 Content IAP - Router

  Another possibility of course is to provide a special device along the path to provide the content IAP capabilities.

  Note that in the case where there is multi-homing (two or more routers connected to provide access for the CPE), intercept taps may have to be installed on more than one access router. If the CPE is multi-homed to multiple service providers, then the intercept will have to be installed on each service provider separately and the LEA will have to correlate the data.

* Unauthorized Creation and Detection: Another concern is the prevention of unauthorized creation and detection of intercepts. This is particularly important when a network element such as a router is used as a content IAP. Those routers that have the capability should be carefully controlled with access to intercept capability and information only via authorized personnel. In one approach using the reference model in Figure 1, the MD is in a controlled environment and the MD does the intercept request to the content IAP over an encrypted link. Logging and auditing are used to detect unauthorized attempts to access the intercept capability.

T-Mobile Exhibit 1002
Part 3 of 6

* Capacity: Support for lawful intercept on a network element supporting customers consumes resources on that equipment. Therefore, support for lawful intercept requires capacity planning and engineering to ensure that revenue-producing services are not adversely affected.

3. Interfaces

This section provides a brief description of the interfaces in the reference model (Figure 1). A list of these interfaces is included in Table 1 in Section 2.

One of the objectives in defining these interfaces is to keep the internal interfaces (b to f) the same regardless of country-specific requirements. The MD then formats the IRI and the content to meet the country specific requirements for interfaces (g) and (h).

3.1. Content Intercept Request Interface

This section describes some of the requirements for the content intercept request interface (d) in Figure 1. It makes use of a common request protocol (SNMPv3) regardless of the type of application (e.g., voice, data) and suggests the usage of a TAP-MIB, which is defined in a separate document [1]. Some of the considerations that lead to the use of SNMPv3 and to the definition of the specific Management Information Base (MIB) defined in [1] are provided here.

In order to provide a generic interface for intercepting, replicating, encapsulating and transporting content packets to the MD, the content intercept interface ((d) in Figure 1) should specify:

* A Filter specification for classifying the packets to be intercepted.

* The destination address of the MD (where to send the packets).

* Encapsulation and Transport parameters.

In addition, a timeout value for the intercept should also be specified. This defines a limited lifetime for the intercept so that failures will not result in intercepts remaining beyond their authorized lifetime. If a failure of the MD occurs such that it is not able to supply the refresh to the timeout, then the intercept will cease to exist after the timeout expires. Similarly, if the IAP re-boots, then the intercept will not survive the re-boot unless the IAP is capable of ascertaining that the intercept lifetime requirements will continue to be met.

In order for this to work, it must be possible for the mediation device to realize that there is a failure in the IAP such that it

T-Mobile Exhibit 1002
Part 3 of 6

must re-establish the intercept. This may be in the form of an audit (from the MD to the IAP), or in the form of a heartbeat mechanism in the content stream, or both.

3.2. Intercept Content Interface (f)

The encapsulation method should retain all of the information in the original packets (source and destination addresses as well as payload) and provide an identifier for correlating the packets with the IRI. One encapsulation that meets those requirements is described in Section 4 of [2]. For non-voice intercepts, the "Intercepted Information" field in Table 1 of [2] contains the original intercepted IP packet.

Note, however, that the interface defined in [2] is based on UDP which is an unreliable and unordered transport protocol (i.e., provides neither retransmission on detection of errors nor ordering of data). If this transport is used, the underlying network (Layers 1 - - 3) should be engineered to meet the overall reliability requirements for delivery of content.

If a more reliable transport protocol is required, then a mechanism that provides timely delivery as well as limits the burden (both processing and buffering) on the Content IAP should be used. One mechanism that meets these requirements is a NACK-oriented retransmission scheme based on [12].

If [12] is used, the call content channel identifier may be placed in the SSRC field of the encapsulating RTP packet. The payload type may be used to identify the type of packet encapsulated in RTP (e.g., IP, PPP, Ethernet MAC). Note that usage of [12] is still under investigation and may need further specification. Usage of [12] in the content IAP places more processing burden on the content IAP than a UDP-based intercept and can affect the capacity of the content IAP.

4. Applying the Reference Model

This section applies the reference model to some example applications.

4.1. Voice over IP networks

This section will look at some of the issues surrounding interception of voice over IP calls, taking local voice services as a specific service example. The reference model from Figure 1 will be applied with the use of a common set of interfaces that are independent of the call signaling protocols in use.

4.1.1. Interception of Voice over IP Services

T-Mobile Exhibit 1002
Part 3 of 6

There are a variety of architectures in use for voice over IP (e.g., centralized versus distributed) as well as various protocols (SIP [6], H.323 [9], MGCP [7], H.248 [8]). There are also a variety of services that may be offered:

* Local Voice Services (i.e., service to a user that has an IP phone or a phone connected to a gateway)

* Transit services

* Long distance access services (e.g., calling/debit card).

This document does not address any obligations that a service provider might or might not have to support intercepts. It simply describes how intercept might be done using the reference model in Figure 1.

Note that in the case of services where the intercept subject accesses the network via a non-IP endpoint (e.g., TDM), the detectability issue is less acute (e.g., re-routing of packets to intercept them in a special device is a possible option), since the intercept subject does not have access to the IP addresses or to traceroute.

However, in the case of local services, this is a much more difficult problem. The intercept for a call originating and terminating on-net (i.e., a call that is voice over IP end-to-end) has to be intercepted along its normal route in order to be undetectable. In addition, the call-forwarding feature that is often provided as a local service feature makes interception even more difficult: If call forwarding is invoked, a call that was intended to terminate on the intercept subject may be forwarded anywhere in the network resulting in the media stream bypassing the original content IAP (since in voice over IP, the media stream goes directly from end-to-end). Also, since call forwarding can often be set up on a call-by-call basis, the location of the content IAP will often not be known until the call is set up.

4.1.2. Local Voice Services

This sub-section will look at the specific case in which the intercept subject under surveillance is being provided with a local voice service by the same provider that also provides the network access (e.g., controls the edge router or switch). This is an important assumption, since in VoIP the entity providing call control (e.g., SIP server) can be totally separate from the entity providing network access (e.g., operates edge routers).

T-Mobile Exhibit 1002
Part 3 of 6

Suppose that a subscriber that subscribes to a local (e.g., residential) voice service is a target for a lawfully authorized surveillance. Part of the system providing these services is a subscriber database that includes addressing information about the subscriber as well information on what features are in effect (e.g., call forwarding). Some call control entity (CCE) accesses that database in order to provide local services. For example, if the subject has call forwarding invoked, that fact (and where to forward the call) is indicated in the subscriber database. A call arriving at the CCE that "owns" that subscriber can then take the appropriate action (e.g., forward the call).

The CCE that "owns" the target subscriber (which could be an H.323 gatekeeper, a SIP proxy or a Media Gateway Controller) is provisioned with the intercept parameters (e.g., subject identification information such as the telephone number and where to deliver the IRI). The provisioning of this CCE could be through interface (c) in Figure 1. The CCE in question is the IRI IAP and once provisioned, it passes the IRI to the MD. In the scenario being discussed, the CCE typically remains in the signaling path throughout the call, even in the call-forwarding case. Part of the IRI it passes to the MD is the media signaling information (i.e., SDP [11] or H.245 [10]), which includes endpoint IP address and port information for the media (content) streams. Armed with this media address information, the MD can determine the content IAP (e.g., [5]) and make the request via interface (d). The request identifies the voice stream to be intercepted based on information received in the call signaling (i.e., IP addresses and UDP port numbers).

Note that the content IAP in the case of voice over IP could be an edge router or a PSTN gateway (e.g., a call from the PSTN forwarded to the PSTN). SIP, H.323, MGCP or H.248 call signaling protocols could be used. However, the protocol (SNMPv3 [1]) used for interface

(d), is not dependent on the type of call signaling protocol used; nor is the encapsulation format and transport protocol (interface "f"). The same reference model (Figure 1) with the same interfaces can be used for lawfully authorized surveillance, regardless of the signaling protocol and regardless of the type of service being provided (Note: even though a local voice service was used in this example, other voice services could use the same model and interfaces).

4.2. Data Services

The same model (Figure 1) can also be used for data services. In this case the IRI IAP could be a server that acts as registration, authentication and authorization point for the data service (e.g., a RADIUS server). If a potential IRI IAP does not have the available interfaces (c) and (e), the MD may have to do a content tap on registration signaling in order to obtain the IRI.

T-Mobile Exhibit 1002
Part 3 of 6

The IRI in the case of a data service could include:

* The time that the user registered or de-registered for the service.
* Addressing information (i.e., given the user identity, what IP address or other information is available that could be used in interface (d) to do the content tap).

Once suitable addressing information is available in order to do content tapping the MD can invoke the tap via interface (d).

Clearly the IRI interfaces (c, e, g) are different for data than they are for voice services. However, the content IAP is typically the same (an edge router). Interfaces (d, f, and h) may also be the same.

5. Security Considerations

Given the sensitive nature of lawful intercept (LI) -- both from the standpoint of the need to protect sensitive data, as well as conceal the identities of the intercept subjects, the LI solution should have the ability to provide stringent security measures to combat threats such as impersonation of MD's, privacy and confidentiality breaches, as well as message forgery and replay attacks.

While this document doesn't discuss issues of physical security, operating system, or application hardening within the principals of the LI solution, they are clearly important. In particular, the MD server would be considered a prime target for attacks.

In general, all interfaces should have the capability of providing strong cryptographic authentication to establish the identity of the principals, and be able to correlate the identity of the principal with the action they are attempting to perform. All interfaces should be capable of performing some sort of cryptographic message integrity checking such as, for example, HMAC-MD5. Message integrity checking can also be used to counter replay attacks. Privacy and confidentiality considerations, may also require the use of encryption.

The content and IRI IAPs also should also provide protection of the identity of the intercept subject and the existence of an intercept.

5.1. Content Request Interface (d) - SNMPv3 Control

For interface (d,) native SNMPv3 security module mechanism is used. The additional requirement is that the IAP should support the ability to protect the TAP MIB's [1] from disclosure or control by unauthorized USM [3] users. VACM [4] provides the necessary tools to limit the views to particular USM users, but there are also special considerations:

T-Mobile Exhibit 1002
Part 3 of 6

* The ability to limit access to the appropriate TAP MIB's by only those SNMPv3 USM users which have keys established and the proper VACM views defined.

* Segregation of the TAP MIB such that only operators of sufficient privilege level can create VACM views that include the TAP MIB [1].

6. Informative References

[1] Baker, F., "Cisco Lawful Intercept Control MIB", Work in Progress, April 2004.

[2] PacketCable(TM) Electronic Surveillance Specification, PKT-SP-ESP-I04-040723, http://www.packetcable.com/specifications/

[3] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[4] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002.

[5] Warnicke, E., "A Suggested Scheme for DNS Resolution of Networks and Gateways", Work in Progress.

[6] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[7] Andreasen, F. and B. Foster, "Media Gateway Control Protocol (MGCP) Version 1.0", RFC 3435, January 2003.

[8] ITU-T Recommendation H.248.1, Gateway Control Protocol: Version 2, May 2002.

[9] ITU-T Recommendation H.323, Packet-based Multimedia Communications Systems, July 2003.

[10] ITU-T Recommendation H.245, Control Protocol for Multimedia Communications, July 2003.

[11] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[12] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenber, "RTP Retransmission Payload Format", Work in Progress.

[13] ETSI TS 101 331, Telecommunications security; Lawful Interception (LI); Requirements of law enforcement agencies.

[14] ETSI TS 33.108 v6.7.0, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Handover Interface for Lawful Interception (Release

6).

[15] IAB and IESG, "IETF Policy on Wiretapping", RFC 2804, May 2000.

7. Acronyms

CCE	Call Control Entity
CMTS	Cable Modem Termination System
CPE	Customer Premises Equipment
ETSI	European Telecommunications Standards Institute
GPRS	Generalized Packet Radio Service
HMAC-MD5	Hash-based Message Authentication Code - Message Digest 5
IAP	Intercept Access Point
IETF	Internet Engineering Task Force
IRI	Intercept Related Information
ITU-T	International Telecommunications Union - Telecommunications Sector
LEA	Law Enforcement Agency
LI	Lawful Intercept
MGCP	Media Gateway Control Protocol
MD	Mediation Device
MIB	Management Information Base
NACK	Negative Acknowledgement
PSTN	Public Switched Telecommunications Network
RFC	Request for Comment
RTP	Real-time Transport Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SSRC	Synchronization Source
TDM	Time Division Multiplex
UDP	User Datagram Protocol
USM	User Service Model
VACM	View-based Access Control Model
VoIP	Voice over IP

8. Authors' Addresses

Fred Baker
Cisco Systems
1121 Via Del Rey
Santa Barbara, CA 93117
US

Phone: +1-408-526-4257
Fax: +1-413-473-2403
EMail: fred@cisco.com

Bill Foster
Cisco Systems
Suite 2150
1050 West Pender St.
Vancouver, BC, V6E 3S7
Canada

Phone: +1-604-647-2315
EMail: bfoster@cisco.com

T-Mobile Exhibit 1002
Part 3 of 6

Chip Sharp
Cisco Systems
7025 Kit Creek Road
RTP, NC 27709 USA

Tel:+1.919.392.3121
EMail: chsharp@cisco.com

9. Full Copyright Statement

Copyright (C) The Internet Society (2004).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and at www.rfc-editor.org, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the ISOC's procedures with respect to rights in ISOC Documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

T-Mobile Exhibit 1002
Part 3 of 6

Funding for the RFC Editor function is currently provided by the Internet Society.

Comment on RFC 3924

Previous: RFC 3923 - End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP)

Next: RFC 3925 - Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol version 4 (DHCPv4)

[RFC Index | RFC Search | Usenet FAQs | Web FAQs | Documents | Cities]

Network Working Group
Request for Comments: 2543
Category: Standards Track

M. Handley
ACIRI
H. Schulzrinne
Columbia U.
E. Schooler
Cal Tech
J. Rosenberg
Bell Labs
March 1999

SIP: Session Initiation Protocol

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

IESG Note

The IESG intends to charter, in the near future, one or more working groups to produce standards for "name lookup", where such names would include electronic mail addresses and telephone numbers, and the result of such a lookup would be a list of attributes and characteristics of the user or terminal associated with the name. Groups which are in need of a "name lookup" protocol should follow the development of these new working groups rather than using SIP for this function. In addition it is anticipated that SIP will migrate towards using such protocols, and SIP implementors are advised to monitor these efforts.

Abstract

The Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying and terminating sessions with one or more participants. These sessions include Internet multimedia conferences, Internet telephone calls and multimedia distribution. Members in a session can communicate via multicast or via a mesh of unicast relations, or a combination of these.

SIP invitations used to create sessions carry session descriptions which allow participants to agree on a set of compatible media types. SIP supports user mobility by proxying and redirecting requests to the user's current location. Users can register their current location. SIP is not tied to any particular conference control protocol. SIP is designed to be independent of the lower-layer transport protocol and can be extended with additional capabilities.

Table of Contents

1	Introduction	7
1.1	Overview of SIP Functionality	7
1.2	Terminology	8
1.3	Definitions	9
1.4	Overview of SIP Operation	12
1.4.1	SIP Addressing	12
1.4.2	Locating a SIP Server	13
1.4.3	SIP Transaction	14
1.4.4	SIP Invitation	15
1.4.5	Locating a User	17
1.4.6	Changing an Existing Session	18
1.4.7	Registration Services	18
1.5	Protocol Properties	18
1.5.1	Minimal State	18
1.5.2	Lower-Layer-Protocol Neutral	18
1.5.3	Text-Based	20
2	SIP Uniform Resource Locators	20
3	SIP Message Overview	24
4	Request	26
4.1	Request-Line	26
4.2	Methods	27
4.2.1	INVITE	28
4.2.2	ACK	29
4.2.3	OPTIONS	29
4.2.4	BYE	30
4.2.5	CANCEL	30
4.2.6	REGISTER	31
4.3	Request-URI	34
4.3.1	SIP Version	35
4.4	Option Tags	35
4.4.1	Registering New Option Tags with IANA	35
5	Response	36
5.1	Status-Line	36
5.1.1	Status Codes and Reason Phrases	37
6	Header Field Definitions	39
6.1	General Header Fields	41
6.2	Entity Header Fields	42
6.3	Request Header Fields	43

6.4	Response Header Fields	43
6.5	End-to-end and Hop-by-hop Headers	43
6.6	Header Field Format	43
6.7	Accept	44

6.8 Accept-Encoding 44
6.9 Accept-Language 45
6.10 Allow 45
6.11 Authorization 45
6.12 Call-ID 46
6.13 Contact 47
6.14 Content-Encoding 50
6.15 Content-Length 51
6.16 Content-Type 51
6.17 CSeq 52
6.18 Date 53
6.19 Encryption 54
6.20 Expires 55
6.21 From 56
6.22 Hide 57
6.23 Max-Forwards 59
6.24 Organization 59
6.25 Priority 60
6.26 Proxy-Authenticate 60
6.27 Proxy-Authorization 61
6.28 Proxy-Require 61
6.29 Record-Route 62
6.30 Require 63
6.31 Response-Key 63
6.32 Retry-After 64
6.33 Route 65
6.34 Server 65
6.35 Subject 65
6.36 Timestamp 66
6.37 To 66
6.38 Unsupported 68
6.39 User-Agent 68
6.40 Via 68
6.40.1 Requests 68
6.40.2 Receiver-tagged Via Header Fields 69
6.40.3 Responses 70
6.40.4 User Agent and Redirect Servers 70
6.40.5 Syntax 71
6.41 Warning 72
6.42 WWW-Authenticate 74
7 Status Code Definitions 75
7.1 Informational 1xx 75
7.1.1 100 Trying 75
7.1.2 180 Ringing 75

7.1.3 181 Call Is Being Forwarded 75
7.1.4 182 Queued 76
7.2 Successful 2xx 76
7.2.1 200 OK 76
7.3 Redirection 3xx 76
7.3.1 300 Multiple Choices 77
7.3.2 301 Moved Permanently 77
7.3.3 302 Moved Temporarily 77

7.3.4 305 Use Proxy 77
7.3.5 380 Alternative Service 78
7.4 Request Failure 4xx 78
7.4.1 400 Bad Request 78
7.4.2 401 Unauthorized 78
7.4.3 402 Payment Required 78
7.4.4 403 Forbidden 78
7.4.5 404 Not Found 78
7.4.6 405 Method Not Allowed 78
7.4.7 406 Not Acceptable 79
7.4.8 407 Proxy Authentication Required 79
7.4.9 408 Request Timeout 79
7.4.10 409 Conflict 79
7.4.11 410 Gone 79
7.4.12 411 Length Required 79
7.4.13 413 Request Entity Too Large 80
7.4.14 414 Request-URI Too Long 80
7.4.15 415 Unsupported Media Type 80
7.4.16 420 Bad Extension 80
7.4.17 480 Temporarily Unavailable 80
7.4.18 481 Call Leg/Transaction Does Not Exist 81
7.4.19 482 Loop Detected 81
7.4.20 483 Too Many Hops 81
7.4.21 484 Address Incomplete 81
7.4.22 485 Ambiguous 81
7.4.23 486 Busy Here 82
7.5 Server Failure 5xx 82
7.5.1 500 Server Internal Error 82
7.5.2 501 Not Implemented 82
7.5.3 502 Bad Gateway 82
7.5.4 503 Service Unavailable 83
7.5.5 504 Gateway Time-out 83
7.5.6 505 Version Not Supported 83
7.6 Global Failures 6xx 83
7.6.1 600 Busy Everywhere 83
7.6.2 603 Decline 84
7.6.3 604 Does Not Exist Anywhere 84
7.6.4 606 Not Acceptable 84
8 SIP Message Body 84
8.1 Body Inclusion 84

8.2 Message Body Type 85
8.3 Message Body Length 85
9 Compact Form 85
10 Behavior of SIP Clients and Servers 86
10.1 General Remarks 86
10.1.1 Requests 86
10.1.2 Responses 87
10.2 Source Addresses, Destination Addresses and Connections 88
10.2.1 Unicast UDP 88
10.2.2 Multicast UDP 88
10.3 TCP 89

10.4 Reliability for BYE, CANCEL, OPTIONS, REGISTER Requests 90
10.4.1 UDP 90
10.4.2 TCP 91
10.5 Reliability for INVITE Requests 91
10.5.1 UDP 92
10.5.2 TCP 95
10.6 Reliability for ACK Requests 95
10.7 ICMP Handling 95
11 Behavior of SIP User Agents 95
11.1 Caller Issues Initial INVITE Request 96
11.2 Callee Issues Response 96
11.3 Caller Receives Response to Initial Request 96
11.4 Caller or Callee Generate Subsequent Requests 97
11.5 Receiving Subsequent Requests 97
12 Behavior of SIP Proxy and Redirect Servers 97
12.1 Redirect Server 97
12.2 User Agent Server 98
12.3 Proxy Server 98
12.3.1 Proxying Requests 98
12.3.2 Proxying Responses 99
12.3.3 Stateless Proxy: Proxying Responses 99
12.3.4 Stateful Proxy: Receiving Requests 99
12.3.5 Stateful Proxy: Receiving ACKs 99
12.3.6 Stateful Proxy: Receiving Responses 100
12.3.7 Stateless, Non-Forking Proxy 100
12.4 Forking Proxy 100
13 Security Considerations 104
13.1 Confidentiality and Privacy: Encryption 104
13.1.1 End-to-End Encryption 104
13.1.2 Privacy of SIP Responses 107
13.1.3 Encryption by Proxies 108
13.1.4 Hop-by-Hop Encryption 108
13.1.5 Via field encryption 108
13.2 Message Integrity and Access Control: Authentication 109

13.2.1 Trusting responses 112
13.3 Callee Privacy 113
13.4 Known Security Problems 113
14 SIP Authentication using HTTP Basic and Digest Schemes 113
14.1 Framework 113
14.2 Basic Authentication 114
14.3 Digest Authentication 114
14.4 Proxy-Authentication 115
15 SIP Security Using PGP 115
15.1 PGP Authentication Scheme 115
15.1.1 The WWW-Authenticate Response Header 116
15.1.2 The Authorization Request Header 117
15.2 PGP Encryption Scheme 118
15.3 Response-Key Header Field for PGP 119
16 Examples 119

16.1 Registration 119
16.2 Invitation to a Multicast Conference 121
16.2.1 Request 121
16.2.2 Response 122
16.3 Two-party Call 123
16.4 Terminating a Call 125
16.5 Forking Proxy 126
16.6 Redirects 130
16.7 Negotiation 131
16.8 OPTIONS Request 132
A Minimal Implementation 134
A.1 Client 134
A.2 Server 135
A.3 Header Processing 135
B Usage of the Session Description Protocol (SDP)..... 136
B.1 Configuring Media Streams 136
B.2 Setting SDP Values for Unicast 138
B.3 Multicast Operation 139
B.4 Delayed Media Streams 139
B.5 Putting Media Streams on Hold 139
B.6 Subject and SDP "s=" Line 140
B.7 The SDP "o=" Line 140
C Summary of Augmented BNF 141
C.1 Basic Rules 143
D Using SRV DNS Records 146
E IANA Considerations 148
F Acknowledgments 149
G Authors' Addresses 149
H Bibliography 150
I Full Copyright Statement 153

1 Introduction

1.1 Overview of SIP Functionality

The Session Initiation Protocol (SIP) is an application-layer control protocol that can establish, modify and terminate multimedia sessions or calls. These multimedia sessions include multimedia conferences, distance learning, Internet telephony and similar applications. SIP can invite both persons and "robots", such as a media storage service. SIP can invite parties to both unicast and multicast sessions; the initiator does not necessarily have to be a member of the session to which it is inviting. Media and participants can be added to an existing session.

SIP can be used to initiate sessions as well as invite members to sessions that have been advertised and established by other means. Sessions can be advertised using multicast protocols such as SAP, electronic mail, news groups, web pages or directories (LDAP), among others.

SIP transparently supports name mapping and redirection services, allowing the implementation of ISDN and Intelligent Network telephony subscriber services. These facilities also enable personal mobility. In the parlance of telecommunications intelligent network services, this is defined as: "Personal mobility is the ability of end users to originate and receive calls and access subscribed telecommunication services on any terminal in any location, and the ability of the network to identify end users as they move. Personal mobility is based on the use of a unique personal identity (i.e., personal number)." [1]. Personal mobility complements terminal mobility, i.e., the ability to maintain communications when moving a single end system from one subnet to another.

SIP supports five facets of establishing and terminating multimedia communications:

User location: determination of the end system to be used for communication;

User capabilities: determination of the media and media parameters to be used;

User availability: determination of the willingness of the called party to engage in communications;

Call setup: "ringing", establishment of call parameters at both called and calling party;

Call handling: including transfer and termination of calls.

SIP can also initiate multi-party calls using a multipoint control unit (MCU) or fully-meshed interconnection instead of multicast. Internet telephony gateways that connect Public Switched Telephone Network (PSTN) parties can also use SIP to set up calls between them.

SIP is designed as part of the overall IETF multimedia data and control architecture currently incorporating protocols such as RSVP (RFC 2205 [2]) for reserving network resources, the real-time transport protocol (RTP) (RFC 1889 [3]) for transporting real-time data and providing QOS feedback, the real-time streaming protocol (RTSP) (RFC 2326 [4]) for controlling delivery of streaming media, the session announcement protocol (SAP) [5] for advertising multimedia sessions via multicast and the session description protocol (SDP) (RFC 2327 [6]) for describing multimedia sessions. However, the functionality and operation of SIP does not depend on any of these protocols.

SIP can also be used in conjunction with other call setup and signaling protocols. In that mode, an end system uses SIP exchanges to determine the appropriate end system address and protocol from a given address that is protocol-independent. For example, SIP could be used to determine that the party can be reached via H.323 [7], obtain

the H.245 [8] gateway and user address and then use H.225.0 [9] to establish the call.

In another example, SIP might be used to determine that the callee is reachable via the PSTN and indicate the phone number to be called, possibly suggesting an Internet-to-PSTN gateway to be used.

SIP does not offer conference control services such as floor control or voting and does not prescribe how a conference is to be managed, but SIP can be used to introduce conference control protocols. SIP does not allocate multicast addresses.

SIP can invite users to sessions with and without resource reservation. SIP does not reserve resources, but can convey to the invited system the information necessary to do this.

1.2 Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [10] and indicate requirement levels for compliant SIP implementations.

1.3 Definitions

This specification uses a number of terms to refer to the roles played by participants in SIP communications. The definitions of client, server and proxy are similar to those used by the Hypertext Transport Protocol (HTTP) (RFC 2068 [11]). The terms and generic syntax of URI and URL are defined in RFC 2396 [12]. The following terms have special significance for SIP.

Call: A call consists of all participants in a conference invited by a common source. A SIP call is identified by a globally unique call-id (Section 6.12). Thus, if a user is, for example, invited to the same multicast session by several people, each of these invitations will be a unique call. A point-to-point Internet telephony conversation maps into a single SIP call. In a multiparty conference unit (MCU) based call-in conference, each participant uses a separate call to invite himself to the MCU.

Call leg: A call leg is identified by the combination of Call-ID, To and From.

Client: An application program that sends SIP requests. Clients may or may not interact directly with a human user. User agents and proxies contain clients (and servers).

Conference: A multimedia session (see below), identified by a common session description. A conference can have zero or more members and includes the cases of a multicast conference, a full-mesh

conference and a two-party "telephone call", as well as combinations of these. Any number of calls can be used to create a conference.

Downstream: Requests sent in the direction from the caller to the callee (i.e., user agent client to user agent server).

Final response: A response that terminates a SIP transaction, as opposed to a provisional response that does not. All 2xx, 3xx, 4xx, 5xx and 6xx responses are final.

Initiator, calling party, caller: The party initiating a conference invitation. Note that the calling party does not have to be the same as the one creating the conference.

Invitation: A request sent to a user (or service) requesting participation in a session. A successful SIP invitation consists of two transactions: an INVITE request followed by an ACK request.

Invitee, invited user, called party, callee: The person or service that the calling party is trying to invite to a conference.

Isomorphic request or response: Two requests or responses are defined to be isomorphic for the purposes of this document if they have the same values for the Call-ID, To, From and CSeq header fields. In addition, isomorphic requests have to have the same Request-URI.

Location server: See location service.

Location service: A location service is used by a SIP redirect or proxy server to obtain information about a callee's possible location(s). Location services are offered by location servers. Location servers MAY be co-located with a SIP server, but the manner in which a SIP server requests location services is beyond the scope of this document.

Parallel search: In a parallel search, a proxy issues several requests to possible user locations upon receiving an incoming request. Rather than issuing one request and then waiting for the final response before issuing the next request as in a sequential search , a parallel search issues requests without waiting for the result of previous requests.

Provisional response: A response used by the server to indicate progress, but that does not terminate a SIP transaction. 1xx responses are provisional, other responses are considered final.

Proxy, proxy server: An intermediary program that acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are serviced internally or by passing

them on, possibly after translation, to other servers. A proxy interprets, and, if necessary, rewrites a request message before forwarding it.

Redirect server: A redirect server is a server that accepts a SIP request, maps the address into zero or more new addresses and returns these addresses to the client. Unlike a proxy server , it does not initiate its own SIP request. Unlike a user agent server , it does not accept calls.

Registrar: A registrar is a server that accepts REGISTER requests. A registrar is typically co-located with a proxy or redirect server and MAY offer location services.

Ringback: Ringback is the signaling tone produced by the calling client's application indicating that a called party is being alerted (ringing).

Server: A server is an application program that accepts requests in order to service requests and sends back responses to those requests. Servers are either proxy, redirect or user agent servers or registrars.

Session: From the SDP specification: "A multimedia session is a set of multimedia senders and receivers and the data streams flowing from senders to receivers. A multimedia conference is an example of a multimedia session." (RFC 2327 [6]) (A session as defined for SDP can comprise one or more RTP sessions.) As defined, a callee can be invited several times, by different calls, to the same session. If SDP is used, a session is defined by the concatenation of the user name , session id , network type , address type and address elements in the origin field.

(SIP) transaction: A SIP transaction occurs between a client and a server and comprises all messages from the first request sent from the client to the server up to a final (non-1xx) response sent from the server to the client. A transaction is identified by the CSeq sequence number (Section 6.17) within a single call leg. The ACK request has the same CSeq number as the corresponding INVITE request, but comprises a transaction of its own.

Upstream: Responses sent in the direction from the user agent server to the user agent client.

URL-encoded: A character string encoded according to RFC 1738, Section 2.2 [13].

User agent client (UAC), calling user agent: A user agent client is a client application that initiates the SIP request.

User agent server (UAS), called user agent: A user agent server is a server application that contacts the user when a SIP request is received and that returns a response on behalf of the user. The response accepts, rejects or redirects the request.

User agent (UA): An application which contains both a user agent client and user agent server.

An application program MAY be capable of acting both as a client and a server. For example, a typical multimedia conference control application would act as a user agent client to initiate calls or to

invite others to conferences and as a user agent server to accept invitations. The properties of the different SIP server types are summarized in Table 1.

property	redirect server	proxy server	user agent server	registrar
also acts as a SIP client	no	yes	no	no
returns 1xx status	yes	yes	yes	yes
returns 2xx status	no	yes	yes	yes
returns 3xx status	yes	yes	yes	yes
returns 4xx status	yes	yes	yes	yes
returns 5xx status	yes	yes	yes	yes
returns 6xx status	no	yes	yes	yes
inserts Via header	no	yes	no	no
accepts ACK	yes	yes	yes	no

Table 1: Properties of the different SIP server types

1.4 Overview of SIP Operation

This section explains the basic protocol functionality and operation. Callers and callees are identified by SIP addresses, described in Section 1.4.1. When making a SIP call, a caller first locates the appropriate server (Section 1.4.2) and then sends a SIP request (Section 1.4.3). The most common SIP operation is the invitation (Section 1.4.4). Instead of directly reaching the intended callee, a SIP request may be redirected or may trigger a chain of new SIP requests by proxies (Section 1.4.5). Users can register their location(s) with SIP servers (Section 4.2.6).

1.4.1 SIP Addressing

The "objects" addressed by SIP are users at hosts, identified by a SIP URL. The SIP URL takes a form similar to a mailto or telnet URL, i.e., user@host. The user part is a user name or a telephone number. The host part is either a domain name or a numeric network address.

See section 2 for a detailed discussion of SIP URL's.

A user's SIP address can be obtained out-of-band, can be learned via existing media agents, can be included in some mailers' message headers, or can be recorded during previous invitation interactions. In many cases, a user's SIP URL can be guessed from their email address.

A SIP URL address can designate an individual (possibly located at one of several end systems), the first available person from a group of individuals or a whole group. The form of the address, for example, sip:sales@example.com , is not sufficient, in general, to determine the intent of the caller.

If a user or service chooses to be reachable at an address that is guessable from the person's name and organizational affiliation, the traditional method of ensuring privacy by having an unlisted "phone" number is compromised. However, unlike traditional telephony, SIP offers authentication and access control mechanisms and can avail itself of lower-layer security mechanisms, so that client software can reject unauthorized or undesired call attempts.

1.4.2 Locating a SIP Server

When a client wishes to send a request, the client either sends it to a locally configured SIP proxy server (as in HTTP), independent of the Request-URI, or sends it to the IP address and port corresponding to the Request-URI.

For the latter case, the client must determine the protocol, port and IP address of a server to which to send the request. A client SHOULD follow the steps below to obtain this information, but MAY follow the alternative, optional procedure defined in Appendix D. At each step, unless stated otherwise, the client SHOULD try to contact a server at the port number listed in the Request-URI. If no port number is present in the Request-URI, the client uses port 5060. If the Request-URI specifies a protocol (TCP or UDP), the client contacts the server using that protocol. If no protocol is specified, the client tries UDP (if UDP is supported). If the attempt fails, or if the client doesn't support UDP but supports TCP, it then tries TCP.

A client SHOULD be able to interpret explicit network notifications (such as ICMP messages) which indicate that a server is not reachable, rather than relying solely on timeouts. (For socket-based programs: For TCP, connect() returns ECONNREFUSED if the client could not connect to a server at that address. For UDP, the socket needs to be bound to the destination address using connect() rather than sendto() or similar so that a second write() fails with ECONNREFUSED if there is no server listening) If the client finds the server is not reachable at a particular address, it SHOULD behave as if it had received a 400-class error response to that request.

The client tries to find one or more addresses for the SIP server by querying DNS. The procedure is as follows:

1. If the host portion of the Request-URI is an IP address, the client contacts the server at the given address. Otherwise, the client proceeds to the next step.

2. The client queries the DNS server for address records for the host portion of the Request-URI. If the DNS server returns no address records, the client stops, as it has been unable to locate a server. By address record, we mean A RR's, AAAA RR's, or other similar address records, chosen according to the client's network protocol capabilities.

There are no mandatory rules on how to select a host name for a SIP server. Users are encouraged to name their SIP servers using the sip.domainname (i.e., sip.example.com) convention, as specified in RFC 2219 [16]. Users may only know an email address instead of a full SIP URL for a callee, however. In that case, implementations may be able to increase the likelihood of reaching a SIP server for that domain by constructing a SIP URL from that email address by prefixing the host name with "sip.". In the future, this mechanism is likely to become unnecessary as better DNS techniques, such as the one in Appendix D, become widely available.

A client MAY cache a successful DNS query result. A successful query is one which contained records in the answer, and a server was contacted at one of the addresses from the answer. When the client wishes to send a request to the same host, it MUST start the search as if it had just received this answer from the name server. The client MUST follow the procedures in RFC1035 [15] regarding DNS cache invalidation when the DNS time-to-live expires.

1.4.3 SIP Transaction

Once the host part has been resolved to a SIP server, the client sends one or more SIP requests to that server and receives one or more responses from the server. A request (and its retransmissions) together with the responses triggered by that request make up a SIP transaction. All responses to a request contain the same values in the Call-ID, CSeq, To, and From fields (with the possible addition of a tag in the To field (section 6.37)). This allows responses to be matched with requests. The ACK request following an INVITE is not part of the transaction since it may traverse a different set of hosts.

If TCP is used, request and responses within a single SIP transaction are carried over the same TCP connection (see Section 10). Several SIP requests from the same client to the same server MAY use the same TCP connection or MAY use a new connection for each request.

If the client sent the request via unicast UDP, the response is sent to the address contained in the next Via header field (Section 6.40) of the response. If the request is sent via multicast UDP, the response is directed to the same multicast address and destination port. For UDP, reliability is achieved using retransmission (Section 10).

The SIP message format and operation is independent of the transport protocol.

1.4.4 SIP Invitation

A successful SIP invitation consists of two requests, INVITE followed by ACK. The INVITE (Section 4.2.1) request asks the callee to join a particular conference or establish a two-party conversation. After the callee has agreed to participate in the call, the caller confirms that it has received that response by sending an ACK (Section 4.2.2) request. If the caller no longer wants to participate in the call, it sends a BYE request instead of an ACK.

The INVITE request typically contains a session description, for example written in SDP (RFC 2327 [6]) format, that provides the called party with enough information to join the session. For multicast sessions, the session description enumerates the media types and formats that are allowed to be distributed to that session. For a unicast session, the session description enumerates the media types and formats that the caller is willing to use and where it wishes the media data to be sent. In either case, if the callee wishes to accept the call, it responds to the invitation by returning a similar description listing the media it wishes to use. For a multicast session, the callee SHOULD only return a session description if it is unable to receive the media indicated in the caller's description or wants to receive data via unicast.

The protocol exchanges for the INVITE method are shown in Fig. 1 for a proxy server and in Fig. 2 for a redirect server. (Note that the messages shown in the figures have been abbreviated slightly.) In Fig. 1, the proxy server accepts the INVITE request (step 1), contacts the location service with all or parts of the address (step 2) and obtains a more precise location (step 3). The proxy server then issues a SIP INVITE request to the address(es) returned by the location service (step 4). The user agent server alerts the user (step 5) and returns a success indication to the proxy server (step

6). The proxy server then returns the success result to the original caller (step 7). The receipt of this message is confirmed by the caller using an ACK request, which is forwarded to the callee (steps 8 and 9). Note that an ACK can also be sent directly to the callee, bypassing the proxy. All requests and responses have the same Call-ID.

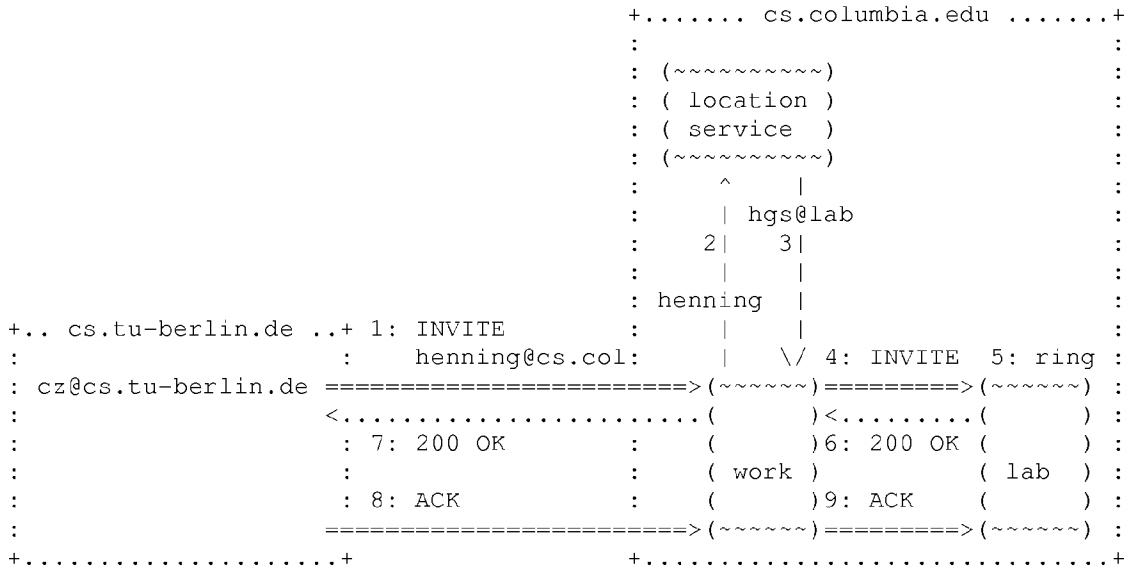


Figure 1: Example of SIP proxy server

The redirect server shown in Fig. 2 accepts the INVITE request (step 1), contacts the location service as before (steps 2 and 3) and, instead of contacting the newly found address itself, returns the address to the caller (step 4), which is then acknowledged via an ACK

request (step 5). The caller issues a new request, with the same call-ID but a higher CSeq, to the address returned by the first server (step 6). In the example, the call succeeds (step 7). The caller and callee complete the handshake with an ACK (step 8).

The next section discusses what happens if the location service returns more than one possible alternative.

1.4.5 Locating a User

A callee may move between a number of different end systems over time. These locations can be dynamically registered with the SIP server (Sections 1.4.7, 4.2.6). A location server MAY also use one or more other protocols, such as finger (RFC 1288 [17]), rwhois (RFC 2167 [18]), LDAP (RFC 1777 [19]), multicast-based protocols [20] or operating-system dependent mechanisms to actively determine the end system where a user might be reachable. A location server MAY return several locations because the user is logged in at several hosts simultaneously or because the location server has (temporarily) inaccurate information. The SIP server combines the results to yield a list of a zero or more locations.

The action taken on receiving a list of locations varies with the type of SIP server. A SIP redirect server returns the list to the client as Contact headers (Section 6.13). A SIP proxy server can sequentially or in parallel try the addresses until the call is successful (2xx response) or the callee has declined the call (6xx response). With sequential attempts, a proxy server can implement an "anycast" service.

If a proxy server forwards a SIP request, it MUST add itself to the beginning of the list of forwarders noted in the Via (Section 6.40) headers. The Via trace ensures that replies can take the same path back, ensuring correct operation through compliant firewalls and avoiding request loops. On the response path, each host MUST remove its Via, so that routing internal information is hidden from the callee and outside networks. A proxy server MUST check that it does not generate a request to a host listed in the Via sent-by, via-received or via-maddr parameters (Section 6.40). (Note: If a host has several names or network addresses, this does not always work. Thus, each host also checks if it is part of the Via list.)

A SIP invitation may traverse more than one SIP proxy server. If one of these "forks" the request, i.e., issues more than one request in response to receiving the invitation request, it is possible that a client is reached, independently, by more than one copy of the

invitation request. Each of these copies bears the same Call-ID. The user agent MUST return the same status response returned in the first response. Duplicate requests are not an error.

1.4.6 Changing an Existing Session

In some circumstances, it is desirable to change the parameters of an existing session. This is done by re-issuing the INVITE, using the same Call-ID, but a new or different body or header fields to convey the new information. This re INVITE MUST have a higher CSeq than any previous request from the client to the server.

For example, two parties may have been conversing and then want to add a third party, switching to multicast for efficiency. One of the participants invites the third party with the new multicast address and simultaneously sends an INVITE to the second party, with the new multicast session description, but with the old call identifier.

1.4.7 Registration Services

The REGISTER request allows a client to let a proxy or redirect server know at which address(es) it can be reached. A client MAY also use it to install call handling features at the server.

1.5 Protocol Properties

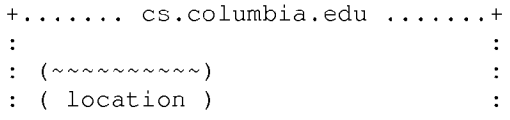
1.5.1 Minimal State

A single conference session or call involves one or more SIP request-response transactions. Proxy servers do not have to keep state for a particular call, however, they MAY maintain state for a single SIP transaction, as discussed in Section 12. For efficiency, a server MAY cache the results of location service requests.

1.5.2 Lower-Layer-Protocol Neutral

SIP makes minimal assumptions about the underlying transport and network-layer protocols. The lower-layer can provide either a packet or a byte stream service, with reliable or unreliable service.

In an Internet context, SIP is able to utilize both UDP and TCP as transport protocols, among others. UDP allows the application to more carefully control the timing of messages and their retransmission, to perform parallel searches without requiring TCP connection state for each outstanding request, and to use multicast. Routers can more readily snoop SIP UDP packets. TCP allows easier passage through existing firewalls.



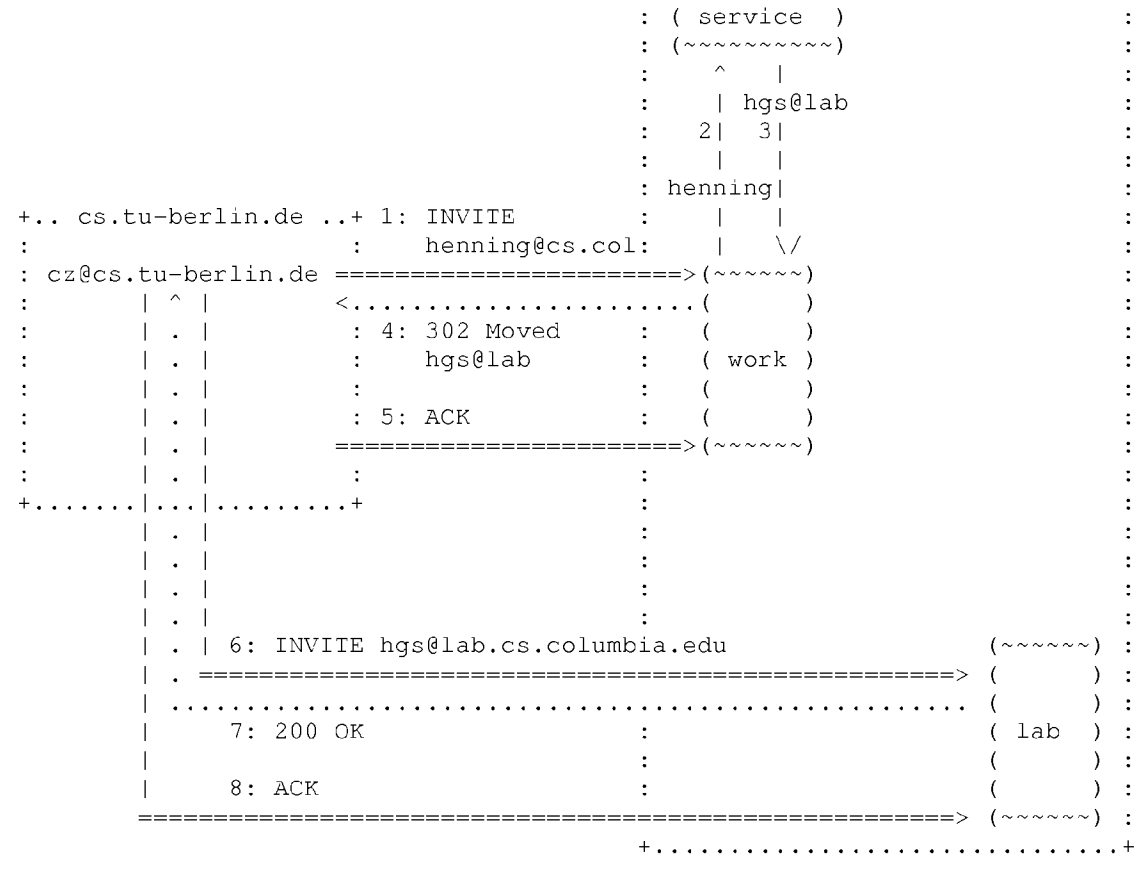


Figure 2: Example of SIP redirect server

When TCP is used, SIP can use one or more connections to attempt to contact a user or to modify parameters of an existing conference. Different SIP requests for the same SIP call MAY use different TCP connections or a single persistent connection, as appropriate.

For concreteness, this document will only refer to Internet protocols. However, SIP MAY also be used directly with protocols such as ATM AAL5, IPX, frame relay or X.25. The necessary naming conventions are beyond the scope of this document. User agents SHOULD implement both UDP and TCP transport. Proxy, registrar, and redirect servers MUST implement both UDP and TCP transport.

1.5.3 Text-Based

SIP is text-based, using ISO 10646 in UTF-8 encoding throughout. This allows easy implementation in languages such as Java, Tcl and Perl, allows easy debugging, and most importantly, makes SIP flexible and extensible. As SIP is used for initiating multimedia conferences rather than delivering media data, it is believed that the additional overhead of using a text-based protocol is not significant.

2 SIP Uniform Resource Locators

SIP URLs are used within SIP messages to indicate the originator (From), current destination (Request-URI) and final recipient (To) of a SIP request, and to specify redirection addresses (Contact). A SIP URL can also be embedded in web pages or other hyperlinks to indicate that a particular user or service can be called via SIP. When used as a hyperlink, the SIP URL indicates the use of the INVITE method.

The SIP URL scheme is defined to allow setting SIP request-header fields and the SIP message-body.

> This corresponds to the use of mailto: URLs. It makes it possible, for example, to specify the subject, urgency or media types of calls initiated through a web page or as part of an email message.

A SIP URL follows the guidelines of RFC 2396 [12] and has the syntax shown in Fig. 3. The syntax is described using Augmented Backus-Naur Form (See Section C). Note that reserved characters have to be escaped and that the "set of characters reserved within any given URI component is defined by that component. In general, a character is reserved if the semantics of the URI changes if the character is replaced with its escaped US-ASCII encoding" [12].


```
SIP-URL          = "sip:" [ userinfo "@" ] hostport
                   url-parameters [ headers ]
userinfo         = user [ ":" password ]
user             = *( unreserved | escaped
                 | "&" | "=" | "+" | "$" | "," )
password         = *( unreserved | escaped
                 | "&" | "=" | "+" | "$" | "," )
hostport         = host [ ":" port ]
host             = hostname | IPv4address
hostname         = *( domainlabel "." ) toplabel [ "." ]
domainlabel      = alphanum | alphanum *( alphanum | "-" ) alphanum
toplabel         = alpha | alpha *( alphanum | "-" ) alphanum
IPv4address      = 1*digit "." 1*digit "." 1*digit "." 1*digit
port             = *digit
```

```
  url-parameters  = *( ";" url-parameter )
  url-parameter   = transport-param | user-param | method-param
                  | ttl-param | maddr-param | other-param
  transport-param = "transport=" ( "udp" | "tcp" )
  ttl-param       = "ttl=" ttl
  ttl             = 1*3DIGIT       ; 0 to 255
  maddr-param     = "maddr=" host
  user-param      = "user=" ( "phone" | "ip" )
  method-param    = "method=" Method
  tag-param       = "tag=" UUID
  UUID            = 1*( hex | "-" )
  other-param     = ( token | ( token "=" ( token | quoted-string )))
  headers         = "?" header *( "&" header )
  header          = hname "=" hvalue
  hname           = 1*uric
  hvalue          = *uric
  uric            = reserved | unreserved | escaped
  reserved        = ";" | "/" | "?" | ":" | "@" | "&" | "=" | "+" |
                    "$" | ","
  digits          = 1*DIGIT
```

Figure 3: SIP URL syntax

The URI character classes referenced above are described in Appendix C.

The components of the SIP URI have the following meanings.


```
telephone-subscriber  = global-phone-number | local-phone-number
   global-phone-number   = "+" 1*phonedigit [isdn-subaddress]
                           [post-dial]
   local-phone-number    = 1*(phonedigit | dtmf-digit |
                           pause-character) [isdn-subaddress]
                           [post-dial]
   isdn-subaddress       = ";isub=" 1*phonedigit
   post-dial             = ";postd=" 1*(phonedigit | dtmf-digit
                         |  pause-character)
   phonedigit            = DIGIT | visual-separator
   visual-separator      = "-" | "."
   pause-character       = one-second-pause | wait-for-dial-tone
   one-second-pause      = "p"
   wait-for-dial-tone    = "w"
   dtmf-digit            = "*" | "#" | "A" | "B" | "C" | "D"
```

Figure 4: SIP URL syntax; telephone subscriber

user: If the host is an Internet telephony gateway, the user field MAY also encode a telephone number using the notation of telephone-subscriber (Fig. 4). The telephone number is a special case of a user name and cannot be distinguished by a BNF. Thus, a URL parameter, user, is added to distinguish telephone numbers from user names. The phone identifier is to be used when connecting to a telephony gateway. Even without this parameter, recipients of SIP URLs MAY interpret the pre-@ part as a phone number if local restrictions on the name space for user name allow it.

password: The SIP scheme MAY use the format "user:password" in the userinfo field. The use of passwords in the userinfo is NOT RECOMMENDED, because the passing of authentication information in clear text (such as URIs) has proven to be a security risk in almost every case where it has been used.

host: The mailto: URL and RFC 822 email addresses require that numeric host addresses ("host numbers") are enclosed in square brackets (presumably, since host names might be numeric), while host numbers without brackets are used for all other URLs. The SIP URL requires the latter form, without brackets.

The issue of IPv6 literal addresses in URLs is being looked at elsewhere in the IETF. SIP implementers are advised to keep up to date on that activity.

port: The port number to send a request to. If not present, the procedures outlined in Section 1.4.2 are used to determine the port number to send a request to.

URL parameters: SIP URLs can define specific parameters of the request. URL parameters are added after the host component and are separated by semi-colons. The transport parameter determines the the transport mechanism (UDP or TCP). UDP is to be assumed when no explicit transport parameter is included. The maddr parameter provides the server address to be contacted for this user, overriding the address supplied in the host field. This address is typically a multicast address, but could also be the address of a backup server. The ttl parameter determines the time-to-live value of the UDP multicast packet and MUST only be used if maddr is a multicast address and the transport protocol is UDP. The user parameter was described above. For example, to specify to call j.doe@big.com using multicast to 239.255.255.1 with a ttl of 15, the following URL would be used:

```
sip:j.doe@big.com;maddr=239.255.255.1;ttl=15
```

The transport, maddr, and ttl parameters MUST NOT be used in the From

and To header fields and the Request-URI; they are ignored if present.

Headers: Headers of the SIP request can be defined with the "?" mechanism within a SIP URL. The special hname "body" indicates that the associated hvalue is the message-body of the SIP INVITE request. Headers MUST NOT be used in the From and To header fields and the Request-URI; they are ignored if present. hname and hvalue are encodings of a SIP header name and value, respectively. All URL reserved characters in the header names and values MUST be escaped.

Method: The method of the SIP request can be specified with the method parameter. This parameter MUST NOT be used in the From and To header fields and the Request-URI; they are ignored if present.

Table 2 summarizes where the components of the SIP URL can be used and what default values they assume if not present.

Examples of SIP URLs are:

	default	Req.-URI	To	From	Contact	external
user	--	x	x	x	x	x
password	--	x	x		x	x
host	mandatory	x	x	x	x	x
port	5060	x	x	x	x	x
user-param	ip	x	x	x	x	x
method	INVITE				x	x
maddr-param	--				x	x
ttl-param	1				x	x
transp.-param	--				x	x
headers	--				x	x

Table 2: Use and default values of URL components for SIP headers, Request-URI and references

```
sip:j.doe@big.com
sip:j.doe:secret@big.com;transport=tcp
sip:j.doe@big.com?subject=project
sip:+1-212-555-1212:1234@gateway.com;user=phone
sip:1212@gateway.com
sip:alice@10.1.2.3
sip:alice@example.com
sip:alice%40example.com@gateway.com
sip:alice@registrar.com;method=REGISTER
```

Within a SIP message, URLs are used to indicate the source and intended destination of a request, redirection addresses and the current destination of a request. Normally all these fields will contain SIP URLs.

SIP URLs are case-insensitive, so that for example the two URLs sip:j.doe@example.com and SIP:J.Doe@Example.com are equivalent. All URL parameters are included when comparing SIP URLs for equality.

SIP header fields MAY contain non-SIP URLs. As an example, if a call from a telephone is relayed to the Internet via SIP, the SIP From header field might contain a phone URL.

3 SIP Message Overview

SIP is a text-based protocol and uses the ISO 10646 character set in UTF-8 encoding (RFC 2279 [21]). Senders MUST terminate lines with a CRLF, but receivers MUST also interpret CR and LF by themselves as line terminators.

Except for the above difference in character sets, much of the message syntax is and header fields are identical to HTTP/1.1; rather than repeating the syntax and semantics here we use [HX.Y] to refer to Section X.Y of the current HTTP/1.1 specification (RFC 2068 [11]). In addition, we describe SIP in both prose and an augmented Backus-Naur form (ABNF). See section C for an overview of ABNF.

Note, however, that SIP is not an extension of HTTP.

Unlike HTTP, SIP MAY use UDP. When sent over TCP or UDP, multiple SIP transactions can be carried in a single TCP connection or UDP datagram. UDP datagrams, including all headers, SHOULD NOT be larger than the path maximum transmission unit (MTU) if the MTU is known, or 1500 bytes if the MTU is unknown.

> The 1500 bytes accommodates encapsulation within the "typical" ethernet MTU without IP fragmentation. Recent studies [22] indicate that an MTU of 1500 bytes is a reasonable assumption. The next lower common MTU values are 1006 bytes for SLIP and 296 for low-delay PPP (RFC 1191 [23]). Thus, another reasonable value would be a message size of 950 bytes, to accommodate packet headers within the SLIP MTU without fragmentation.

A SIP message is either a request from a client to a server, or a response from a server to a client.

```
SIP-message  =  Request | Response
```

Both Request (section 4) and Response (section 5) messages use the generic-message format of RFC 822 [24] for transferring entities (the body of the message). Both types of messages consist of a start-line, one or more header fields (also known as "headers"), an empty line (i.e., a line with nothing preceding the carriage-return line-feed (CRLF)) indicating the end of the header fields, and an optional message-body. To avoid confusion with similar-named headers in HTTP, we refer to the headers describing the message body as entity headers. These components are described in detail in the upcoming sections.

```
generic-message  =  start-line
                    *message-header
```



```
                    CRLF
                    [ message-body ]

start-line       =  Request-Line |     ;Section 4.1
                    Status-Line        ;Section 5.1


message-header   =  ( general-header
                    | request-header
                    | response-header
                    | entity-header )
```

In the interest of robustness, any leading empty line(s) MUST be ignored. In other words, if the Request or Response message begins with one or more CRLF, CR, or LFs, these characters MUST be ignored.

4 Request

The Request message format is shown below:

```
Request  =  Request-Line       ;  Section 4.1
            *( general-header
            | request-header
            | entity-header )
            CRLF
            [ message-body ]   ;  Section 8
```

4.1 Request-Line

The Request-Line begins with a method token, followed by the

Request-URI and the protocol version, and ending with CRLF. The elements are separated by SP characters. No CR or LF are allowed except in the final CRLF sequence.

```
Request-Line  =  Method SP Request-URI SP SIP-Version CRLF
```



```
general-header   =  Accept               ; Section 6.7
                 |  Accept-Encoding      ; Section 6.8
                 |  Accept-Language      ; Section 6.9
                 |  Call-ID              ; Section 6.12
                 |  Contact              ; Section 6.13
                 |  CSeq                 ; Section 6.17
                 |  Date                 ; Section 6.18
                 |  Encryption           ; Section 6.19
                 |  Expires              ; Section 6.20
                 |  From                 ; Section 6.21
                 |  Record-Route         ; Section 6.29
                 |  Timestamp            ; Section 6.36
                 |  To                   ; Section 6.37
                 |  Via                  ; Section 6.40
entity-header    =  Content-Encoding     ; Section 6.14
                 |  Content-Length       ; Section 6.15
                 |  Content-Type         ; Section 6.16
request-header   =  Authorization        ; Section 6.11
                 |  Contact              ; Section 6.13
                 |  Hide                 ; Section 6.22
                 |  Max-Forwards         ; Section 6.23
                 |  Organization         ; Section 6.24
                 |  Priority             ; Section 6.25
                 |  Proxy-Authorization  ; Section 6.27
                 |  Proxy-Require        ; Section 6.28
                 |  Route                ; Section 6.33
                 |  Require              ; Section 6.30
                 |  Response-Key         ; Section 6.31
                 |  Subject              ; Section 6.35
                 |  User-Agent           ; Section 6.39
response-header  =  Allow                ; Section 6.10
                 |  Proxy-Authenticate   ; Section 6.26
                 |  Retry-After          ; Section 6.32
                 |  Server               ; Section 6.34
                 |  Unsupported          ; Section 6.38
                 |  Warning              ; Section 6.41
                 |  WWW-Authenticate     ; Section 6.42
```

Table 3: SIP headers

4.2 Methods

The methods are defined below. Methods that are not supported by a proxy or redirect server are treated by that server as if they were an OPTIONS method and forwarded accordingly. Methods that are not

supported by a user agent server or registrar cause a 501 (Not Implemented) response to be returned (Section 7). As in HTTP, the Method token is case-sensitive.

```
Method  =  "INVITE" | "ACK" | "OPTIONS" | "BYE"
           | "CANCEL" | "REGISTER"
```

4.2.1 INVITE

The INVITE method indicates that the user or service is being invited to participate in a session. The message body contains a description of the session to which the callee is being invited. For two-party calls, the caller indicates the type of media it is able to receive and possibly the media it is willing to send as well as their parameters such as network destination. A success response MUST indicate in its message body which media the callee wishes to receive and MAY indicate the media the callee is going to send.

> Not all session description formats have the ability to indicate sending media.

A server MAY automatically respond to an invitation for a conference the user is already participating in, identified either by the SIP Call-ID or a globally unique identifier within the session description, with a 200 (OK) response.

If a user agent receives an INVITE request for an existing call leg with a higher CSeq sequence number than any previous INVITE for the same Call-ID, it MUST check any version identifiers in the session description or, if there are no version identifiers, the content of the session description to see if it has changed. It MUST also inspect any other header fields for changes. If there is a change, the user agent MUST update any internal state or information generated as a result of that header. If the session description has changed, the user agent server MUST adjust the session parameters accordingly, possibly after asking the user for confirmation. (Versioning of the session description can be used to accommodate the capabilities of new arrivals to a conference, add or delete media or change from a unicast to a multicast conference.)

This method MUST be supported by SIP proxy, redirect and user agent

servers as well as clients.

4.2.2 ACK

The ACK request confirms that the client has received a final response to an INVITE request. (ACK is used only with INVITE requests.) 2xx responses are acknowledged by client user agents, all other final responses by the first proxy or client user agent to receive the response. The Via is always initialized to the host that originates the ACK request, i.e., the client user agent after a 2xx response or the first proxy to receive a non-2xx final response. The ACK request is forwarded as the corresponding INVITE request, based on its Request-URI. See Section 10 for details.

The ACK request MAY contain a message body with the final session description to be used by the callee. If the ACK message body is empty, the callee uses the session description in the INVITE request.

A proxy server receiving an ACK request after having sent a 3xx, 4xx, 5xx, or 6xx response must make a determination about whether the ACK is for it, or for some user agent or proxy server further downstream. This determination is made by examining the tag in the To field. If the tag in the ACK To header field matches the tag in the To header field of the response, and the From, CSeq and Call-ID header fields in the response match those in the ACK, the ACK is meant for the proxy server. Otherwise, the ACK SHOULD be proxied downstream as any other request.

> It is possible for a user agent client or proxy server to receive multiple 3xx, 4xx, 5xx, and 6xx responses to a request along a single branch. This can happen under various error conditions, typically when a forking proxy transitions from stateful to stateless before receiving all responses. The various responses will all be identical, except for the tag in the To field, which is different for each one. It can therefore be used as a means to disambiguate them.

This method MUST be supported by SIP proxy, redirect and user agent servers as well as clients.

4.2.3 OPTIONS

The server is being queried as to its capabilities. A server that believes it can contact the user, such as a user agent where the user is logged in and has been recently active, MAY respond to this request with a capability set. A called user agent MAY return a status reflecting how it would have responded to an invitation, e.g.,

600 (Busy). Such a server SHOULD return an Allow header field indicating the methods that it supports. Proxy and redirect servers simply forward the request without indicating their capabilities.

This method MUST be supported by SIP proxy, redirect and user agent servers, registrars and clients.

4.2.4 BYE

The user agent client uses BYE to indicate to the server that it wishes to release the call. A BYE request is forwarded like an INVITE request and MAY be issued by either caller or callee. A party to a call SHOULD issue a BYE request before releasing a call ("hanging up"). A party receiving a BYE request MUST cease transmitting media streams specifically directed at the party issuing the BYE request.

If the INVITE request contained a Contact header, the callee SHOULD send a BYE request to that address rather than the From address.

This method MUST be supported by proxy servers and SHOULD be supported by redirect and user agent SIP servers.

4.2.5 CANCEL

The CANCEL request cancels a pending request with the same Call-ID, To, From and CSeq (sequence number only) header field values, but does not affect a completed request. (A request is considered completed if the server has returned a final status response.)

A user agent client or proxy client MAY issue a CANCEL request at any time. A proxy, in particular, MAY choose to send a CANCEL to destinations that have not yet returned a final response after it has received a 2xx or 6xx response for one or more of the parallel-search requests. A proxy that receives a CANCEL request forwards the request to all destinations with pending requests.

The Call-ID, To, the numeric part of CSeq and From headers in the CANCEL request are identical to those in the original request. This allows a CANCEL request to be matched with the request it cancels. However, to allow the client to distinguish responses to the CANCEL from those to the original request, the CSeq Method component is set to CANCEL. The Via header field is initialized to the proxy issuing the CANCEL request. (Thus, responses to this CANCEL request only reach the issuing proxy.)

Once a user agent server has received a CANCEL, it MUST NOT issue a 2xx response for the cancelled original request.

A redirect or user agent server receiving a CANCEL request responds with a status of 200 (OK) if the transaction exists and a status of 481 (Transaction Does Not Exist) if not, but takes no further action. In particular, any existing call is unaffected.

> The BYE request cannot be used to cancel branches of a parallel search, since several branches may, through intermediate proxies, find the same user agent server and then terminate the call. To terminate a call instead of just pending searches, the UAC must use BYE instead of or in addition to CANCEL. While CANCEL can terminate any pending request other than ACK or CANCEL, it is typically useful only for INVITE. 200 responses to INVITE and 200 responses to CANCEL are distinguished by the method in the Cseq header field, so there is no ambiguity.

This method MUST be supported by proxy servers and SHOULD be supported by all other SIP server types.

4.2.6 REGISTER

A client uses the REGISTER method to register the address listed in the To header field with a SIP server.

A user agent MAY register with a local server on startup by sending a REGISTER request to the well-known "all SIP servers" multicast address "sip.mcast.net" (224.0.1.75). This request SHOULD be scoped to ensure it is not forwarded beyond the boundaries of the administrative system. This MAY be done with either TTL or administrative scopes [25], depending on what is implemented in the network. SIP user agents MAY listen to that address and use it to become aware of the location of other local users [20]; however, they do not respond to the request. A user agent MAY also be configured with the address of a registrar server to which it sends a REGISTER request upon startup.

Requests are processed in the order received. Clients SHOULD avoid sending a new registration (as opposed to a retransmission) until they have received the response from the server for the previous one.

> Clients may register from different locations, by necessity using different Call-ID values. Thus, the CSeq value cannot be used to enforce ordering. Since registrations are additive, ordering is less of a problem than if each REGISTER request completely replaced all earlier ones.

The meaning of the REGISTER request-header fields is defined as

follows. We define "address-of-record" as the SIP address that the registry knows the registrand, typically of the form "user@domain" rather than "user@host". In third-party registration, the entity issuing the request is different from the entity being registered.

To: The To header field contains the address-of-record whose registration is to be created or updated.

From: The From header field contains the address-of-record of the person responsible for the registration. For first-party registration, it is identical to the To header field value.

Request-URI: The Request-URI names the destination of the registration request, i.e., the domain of the registrar. The user name MUST be empty. Generally, the domains in the Request-URI and the To header field have the same value; however, it is possible to register as a "visitor", while maintaining one's name. For example, a traveler sip:alice@acme.com (To) might register under the Request-URI sip:atlanta.hiayh.org , with the former as the To header field and the latter as the Request-URI. The REGISTER request is no longer forwarded once it has reached the server whose authoritative domain is the one listed in the Request-URI.

Call-ID: All registrations from a client SHOULD use the same Call-ID header value, at least within the same reboot cycle.

Cseq: Registrations with the same Call-ID MUST have increasing CSeq header values. However, the server does not reject out-of-order requests.

Contact: The request MAY contain a Contact header field; future non-REGISTER requests for the URI given in the To header field SHOULD be directed to the address(es) given in the Contact header.

If the request does not contain a Contact header, the registration remains unchanged.

This is useful to obtain the current list of registrations in the response. Registrations using SIP URIs that differ in one or more of host, port, transport-param or maddr-param (see Figure 3) from an existing registration are added to the list of registrations. Other URI types are compared according to the standard URI equivalency rules for the URI schema. If the URIs are equivalent to that of an existing registration, the new registration replaces the

old one if it has a higher q value or, for the same value of q, if the ttl value is higher. All current registrations MUST share the same action value. Registrations that have a different action than current registrations for the same user MUST be rejected with status of 409 (Conflict).

A proxy server ignores the q parameter when processing non-REGISTER requests, while a redirect server simply returns that parameter in its Contact response header field.

> Having the proxy server interpret the q parameter is not sufficient to guide proxy behavior, as it is not clear, for example, how long it is supposed to wait between trying addresses.

If the registration is changed while a user agent or proxy server processes an invitation, the new information SHOULD be used.

> This allows a service known as "directed pick-up". In the telephone network, directed pickup permits a user at a remote station who hears his own phone ringing to pick up at that station, dial an access code, and be connected to the calling user as if he had answered his own phone.

A server MAY choose any duration for the registration lifetime. Registrations not refreshed after this amount of time SHOULD be silently discarded. Responses to a registration SHOULD include an Expires header (Section 6.20) or expires Contact parameters (Section 6.13), indicating the time at which the server will drop the registration. If none is present, one hour is assumed. Clients MAY request a registration lifetime by indicating the time in an Expires header in the request. A server SHOULD NOT use a higher lifetime than the one requested, but MAY use a lower one. A single address (if host-independent) MAY be registered from several different clients.

A client cancels an existing registration by sending a REGISTER request with an expiration time (Expires) of zero seconds for a particular Contact or the wildcard Contact designated by a "*" for all registrations. Registrations are matched based on the user, host, port and maddr parameters.

The server SHOULD return the current list of registrations in the 200 response as Contact header fields.

It is particularly important that REGISTER requests are authenticated since they allow to redirect future requests (see Section 13.2).

> Beyond its use as a simple location service, this method is needed if there are several SIP servers on a single host. In that case, only one of the servers can use the default port number.

Support of this method is RECOMMENDED.

4.3 Request-URI

The Request-URI is a SIP URL as described in Section 2 or a general URI. It indicates the user or service to which this request is being addressed. Unlike the To field, the Request-URI MAY be re-written by proxies.

When used as a Request-URI, a SIP-URL MUST NOT contain the transport-param, maddr-param, ttl-param, or headers elements. A server that receives a SIP-URL with these elements removes them before further processing.

> Typically, the UAC sets the Request-URI and To to the same SIP URL, presumed to remain unchanged over long time periods. However, if the UAC has cached a more direct path to the callee, e.g., from the Contact header field of a response to a previous request, the To would still contain the long-term, "public" address, while the Request-URI would be set to the cached address.

Proxy and redirect servers MAY use the information in the Request-URI and request header fields to handle the request and possibly rewrite the Request-URI. For example, a request addressed to the generic address sip:sales@acme.com is proxied to the particular person, e.g., sip:bob@ny.acme.com , with the To field remaining as sip:sales@acme.com. At ny.acme.com , Bob then designates Alice as the temporary substitute.

The host part of the Request-URI typically agrees with one of the host names of the receiving server. If it does not, the server SHOULD proxy the request to the address indicated or return a 404 (Not Found) response if it is unwilling or unable to do so. For example, the Request-URI and server host name can disagree in the case of a firewall proxy that handles outgoing calls. This mode of operation is similar to that of HTTP proxies.

If a SIP server receives a request with a URI indicating a scheme other than SIP which that server does not understand, the server MUST return a 400 (Bad Request) response. It MUST do this even if the To

header field contains a scheme it does understand. This is because proxies are responsible for processing the Request-URI; the To field is of end-to-end significance.

4.3.1 SIP Version

Both request and response messages include the version of SIP in use, and follow [H3.1] (with HTTP replaced by SIP, and HTTP/1.1 replaced by SIP/2.0) regarding version ordering, compliance requirements, and upgrading of version numbers. To be compliant with this specification, applications sending SIP messages MUST include a SIP-Version of "SIP/2.0".

4.4 Option Tags

Option tags are unique identifiers used to designate new options in SIP. These tags are used in Require (Section 6.30) and Unsupported (Section 6.38) fields.

Syntax:

```
option-tag  =  token
```

See Section C for a definition of token. The creator of a new SIP option MUST either prefix the option with their reverse domain name or register the new option with the Internet Assigned Numbers Authority (IANA). For example, "com.foo.mynewfeature" is an apt name for a feature whose inventor can be reached at "foo.com". Individual organizations are then responsible for ensuring that option names don't collide. Options registered with IANA have the prefix "org.iana.sip.", options described in RFCs have the prefix "org.ietf.rfc.N", where N is the RFC number. Option tags are case-insensitive.

4.4.1 Registering New Option Tags with IANA

When registering a new SIP option, the following information MUST be provided:

- Name and description of option. The name MAY be of any length, but SHOULD be no more than twenty characters long. The name MUST consist of alphanum (See Figure 3) characters only;

- Indication of who has change control over the option (for example, IETF, ISO, ITU-T, other international standardization bodies, a consortium or a particular company or group of companies);

- A reference to a further description, if available, for example (in order of preference) an RFC, a published paper, a patent filing, a technical report, documented source code or a computer manual;

- Contact information (postal and email address);

Registrations should be sent to iana@iana.org

This procedure has been borrowed from RTSP [4] and the RTP AVP [26].

5 Response

After receiving and interpreting a request message, the recipient responds with a SIP response message. The response message format is shown below:

```
Response  =  Status-Line        ;  Section 5.1
             *( general-header
             | response-header
             | entity-header )
             CRLF
             [ message-body ]   ;  Section 8
```

SIP's structure of responses is similar to [H6], but is defined explicitly here.

5.1 Status-Line

The first line of a Response message is the Status-Line, consisting of the protocol version (Section 4.3.1) followed by a numeric Status-Code and its associated textual phrase, with each element separated by SP characters. No CR or LF is allowed except in the final CRLF sequence.

```
Status-Line  =  SIP-version SP Status-Code SP Reason-Phrase CRLF
```


5.1.1 Status Codes and Reason Phrases

The Status-Code is a 3-digit integer result code that indicates the outcome of the attempt to understand and satisfy the request. The Reason-Phrase is intended to give a short textual description of the Status-Code. The Status-Code is intended for use by automata, whereas the Reason-Phrase is intended for the human user. The client is not required to examine or display the Reason-Phrase.

```
Status-Code     =  Informational                    ;Fig. 5
                |  Success                          ;Fig. 5
                |  Redirection                      ;Fig. 6
                |  Client-Error                     ;Fig. 7
                |  Server-Error                     ;Fig. 8
                |  Global-Failure                   ;Fig. 9
                |  extension-code
extension-code  =  3DIGIT
Reason-Phrase   =  *<TEXT-UTF8,  excluding CR, LF>
```

We provide an overview of the Status-Code below, and provide full definitions in Section 7. The first digit of the Status-Code defines the class of response. The last two digits do not have any categorization role. SIP/2.0 allows 6 values for the first digit:

1xx: Informational -- request received, continuing to process the request;

2xx: Success -- the action was successfully received, understood, and accepted;

3xx: Redirection -- further action needs to be taken in order to complete the request;

4xx: Client Error -- the request contains bad syntax or cannot be fulfilled at this server;

5xx: Server Error -- the server failed to fulfill an apparently valid request;

6xx: Global Failure -- the request cannot be fulfilled at any server.

Figures 5 through 9 present the individual values of the numeric response codes, and an example set of corresponding reason phrases for SIP/2.0. These reason phrases are only recommended; they may be replaced by local equivalents without affecting the protocol. Note

that SIP adopts many HTTP/1.1 response codes. SIP/2.0 adds response codes in the range starting at x80 to avoid conflicts with newly defined HTTP response codes, and adds a new class, 6xx, of response codes.

SIP response codes are extensible. SIP applications are not required to understand the meaning of all registered response codes, though such understanding is obviously desirable. However, applications MUST understand the class of any response code, as indicated by the first digit, and treat any unrecognized response as being equivalent to the x00 response code of that class, with the exception that an unrecognized response MUST NOT be cached. For example, if a client receives an unrecognized response code of 431, it can safely assume that there was something wrong with its request and treat the response as if it had received a 400 (Bad Request) response code. In such cases, user agents SHOULD present to the user the message body returned with the response, since that message body is likely to include human-readable information which will explain the unusual status.

```
Informational  =  "100"  ;  Trying
                |  "180"  ;  Ringing
                |  "181"  ;  Call Is Being Forwarded
```

```
                  |   "182"  ;  Queued
      Success      =  "200"  ;  OK
```

Figure 5: Informational and success status codes

```
      Redirection  =  "300"  ;  Multiple Choices
                  |   "301"  ;  Moved Permanently
                  |   "302"  ;  Moved Temporarily
                  |   "303"  ;  See Other
                  |   "305"  ;  Use Proxy
                  |   "380"  ;  Alternative Service
```

Figure 6: Redirection status codes


```
      Client-Error  =  "400"  ;  Bad Request
                   |   "401"  ;  Unauthorized
                   |   "402"  ;  Payment Required
                   |   "403"  ;  Forbidden
                   |   "404"  ;  Not Found
                   |   "405"  ;  Method Not Allowed
                   |   "406"  ;  Not Acceptable
                   |   "407"  ;  Proxy Authentication Required
                   |   "408"  ;  Request Timeout
                   |   "409"  ;  Conflict
                   |   "410"  ;  Gone
                   |   "411"  ;  Length Required
                   |   "413"  ;  Request Entity Too Large
                   |   "414"  ;  Request-URI Too Large
                   |   "415"  ;  Unsupported Media Type
                   |   "420"  ;  Bad Extension
                   |   "480"  ;  Temporarily not available
                   |   "481"  ;  Call Leg/Transaction Does Not Exist
                   |   "482"  ;  Loop Detected
                   |   "483"  ;  Too Many Hops
                   |   "484"  ;  Address Incomplete
                   |   "485"  ;  Ambiguous
                   |   "486"  ;  Busy Here
```

Figure 7: Client error status codes

```
Server-Error  =  "500"  ;  Internal Server Error
              |  "501"  ;  Not Implemented
              |  "502"  ;  Bad Gateway
              |  "503"  ;  Service Unavailable
              |  "504"  ;  Gateway Time-out
              |  "505"  ;  SIP Version not supported
```

Figure 8: Server error status codes

6 Header Field Definitions

SIP header fields are similar to HTTP header fields in both syntax and semantics. In particular, SIP header fields follow the syntax for message-header as described in [H4.2]. The rules for extending header fields over multiple lines, and use of multiple message-header fields with the same field-name, described in [H4.2] also apply to SIP. The


```
Global-Failure |  "600"  ;  Busy Everywhere
               |  "603"  ;  Decline
               |  "604"  ;  Does not exist anywhere
               |  "606"  ;  Not Acceptable
```

Figure 9: Global failure status codes

rules in [H4.2] regarding ordering of header fields apply to SIP, with the exception of Via fields, see below, whose order matters. Additionally, header fields which are hop-by-hop MUST appear before any header fields which are end-to-end. Proxies SHOULD NOT reorder header fields. Proxies add Via header fields and MAY add other hop-by-hop header fields. They can modify certain header fields, such as Max-Forwards (Section 6.23) and "fix up" the Via header fields with "received" parameters as described in Section 6.40.1. Proxies MUST NOT alter any fields that are authenticated (see Section 13.2).

The header fields required, optional and not applicable for each method are listed in Table 4 and Table 5. The table uses "o" to indicate optional, "m" mandatory and "-" for not applicable. A "*" indicates that the header fields are needed only if message body is not empty. See sections 6.15, 6.16 and 8 for details.

The "where" column describes the request and response types with which the header field can be used. "R" refers to header fields that can be used in requests (that is, request and general header fields). "r" designates a response or general-header field as applicable to all responses, while a list of numeric values indicates the status codes with which the header field can be used. "g" and "e" designate

general (Section 6.1) and entity header (Section 6.2) fields, respectively. If a header field is marked "c", it is copied from the request to the response.

The "enc." column describes whether this message header field MAY be encrypted end-to-end. A "n" designates fields that MUST NOT be encrypted, while "c" designates fields that SHOULD be encrypted if encryption is used.

The "e-e" column has a value of "e" for end-to-end and a value of "h" for hop-by-hop header fields.

	where	enc.	e-e	ACK	BYE	CAN	INV	OPT	REG
Accept	R		e	-	-	-	o	o	o
Accept	415		e	-	-	-	o	o	o
Accept-Encoding	R		e	-	-	-	o	o	o
Accept-Encoding	415		e	-	-	-	o	o	o
Accept-Language	R		e	-	o	o	o	o	o
Accept-Language	415		e	-	o	o	o	o	o
Allow	200		e	-	-	-	-	m	-
Allow	405		e	o	o	o	o	o	o
Authorization	R		e	o	o	o	o	o	o
Call-ID	gc	n	e	m	m	m	m	m	m
Contact	R		e	o	-	-	o	o	o
Contact	1xx		e	-	-	-	o	o	-
Contact	2xx		e	-	-	-	o	o	o
Contact	3xx		e	-	o	-	o	o	o
Contact	485		e	-	o	-	o	o	o
Content-Encoding	e		e	o	-	-	o	o	o
Content-Length	e		e	o	-	-	o	o	o
Content-Type	e		e	*	-	-	*	*	*
CSeq	gc	n	e	m	m	m	m	m	m
Date	g		e	o	o	o	o	o	o
Encryption	g	n	e	o	o	o	o	o	o
Expires	g		e	-	-	-	o	-	o
From	gc	n	e	m	m	m	m	m	m
Hide	R	n	h	o	o	o	o	o	o
Max-Forwards	R	n	e	o	o	o	o	o	o
Organization	g	c	h	-	-	-	o	o	o

Table 4: Summary of header fields, A--O

Other header fields can be added as required; a server MUST ignore header fields not defined in this specification that it does not understand. A proxy MUST NOT remove or modify header fields not defined in this specification that it does not understand. A compact

form of these header fields is also defined in Section 9 for use over UDP when the request has to fit into a single packet and size is an issue.

Table 6 in Appendix A lists those header fields that different client and server types MUST be able to parse.

6.1 General Header Fields

General header fields apply to both request and response messages. The "general-header" field names can be extended reliably only in combination with a change in the protocol version. However, new or

	where	enc.	e-e	ACK	BYE	CAN	INV	OPT	REG
Proxy-Authenticate	407	n	h	o	o	o	o	o	o
Proxy-Authorization	R	n	h	o	o	o	o	o	o
Proxy-Require	R	n	h	o	o	o	o	o	o
Priority	R	c	e	-	-	-	o	-	-
Require	R		e	o	o	o	o	o	o
Retry-After	R	c	e	-	-	-	-	-	o
Retry-After	404,480,486	c	e	o	o	o	o	o	o
	503	c	e	o	o	o	o	o	o
	600,603	c	e	o	o	o	o	o	o
Response-Key	R	c	e	-	o	o	o	o	o
Record-Route	R		h	o	o	o	o	o	o
Record-Route	2xx		h	o	o	o	o	o	o
Route	R		h	o	o	o	o	o	o
Server	r	c	e	o	o	o	o	o	o
Subject	R	c	e	-	-	-	o	-	-
Timestamp	g		e	o	o	o	o	o	o
To	gc(1)	n	e	m	m	m	m	m	m
Unsupported	420		e	o	o	o	o	o	o
User-Agent	g	c	e	o	o	o	o	o	o
Via	gc(2)	n	e	m	m	m	m	m	m
Warning	r		e	o	o	o	o	o	o
WWW-Authenticate	401	c	e	o	o	o	o	o	o

Table 5: Summary of header fields, P--Z; (1): copied with possible addition of tag; (2): UAS removes first Via header field

experimental header fields MAY be given the semantics of general header fields if all parties in the communication recognize them to be "general-header" fields. Unrecognized header fields are treated as "entity-header" fields.

6.2 Entity Header Fields

The "entity-header" fields define meta-information about the message-body or, if no body is present, about the resource identified by the request. The term "entity header" is an HTTP 1.1 term where the response body can contain a transformed version of the message

body. The original message body is referred to as the "entity". We retain the same terminology for header fields but usually refer to the "message body" rather then the entity as the two are the same in SIP.

6.3 Request Header Fields

The "request-header" fields allow the client to pass additional information about the request, and about the client itself, to the server. These fields act as request modifiers, with semantics equivalent to the parameters of a programming language method invocation.

The "request-header" field names can be extended reliably only in combination with a change in the protocol version. However, new or experimental header fields MAY be given the semantics of "request-header" fields if all parties in the communication recognize them to be request-header fields. Unrecognized header fields are treated as "entity-header" fields.

6.4 Response Header Fields

The "response-header" fields allow the server to pass additional information about the response which cannot be placed in the Status-Line. These header fields give information about the server and about further access to the resource identified by the Request-URI.

Response-header field names can be extended reliably only in combination with a change in the protocol version. However, new or experimental header fields MAY be given the semantics of "response-header" fields if all parties in the communication recognize them to be "response-header" fields. Unrecognized header fields are treated as "entity-header" fields.

6.5 End-to-end and Hop-by-hop Headers

End-to-end headers MUST be transmitted unmodified across all proxies, while hop-by-hop headers MAY be modified or added by proxies.

6.6 Header Field Format

Header fields ("general-header", "request-header", "response-header", and "entity-header") follow the same generic header format as that given in Section 3.1 of RFC 822 [24]. Each header field consists of a name followed by a colon (":") and the field value. Field names are case-insensitive. The field value MAY be preceded by any amount of leading white space (LWS), though a single space (SP) is preferred. Header fields can be extended over multiple lines by preceding each extra line with at least one SP or horizontal tab (HT). Applications MUST follow HTTP "common form" when generating these constructs,

since there might exist some implementations that fail to accept anything beyond the common forms.


```
message-header  =  field-name ":" [ field-value ] CRLF
field-name      =  token
field-value     =  *( field-content | LWS )
field-content   =  < the OCTETs  making up the field-value
                     and consisting of either *TEXT-UTF8
                     or combinations of token,
                     separators, and quoted-string>
```

The relative order of header fields with different field names is not significant. Multiple header fields with the same field-name may be present in a message if and only if the entire field-value for that header field is defined as a comma-separated list (i.e., #(values)). It MUST be possible to combine the multiple header fields into one "field-name: field-value" pair, without changing the semantics of the message, by appending each subsequent field-value to the first, each separated by a comma. The order in which header fields with the same field-name are received is therefore significant to the interpretation of the combined field value, and thus a proxy MUST NOT change the order of these field values when a message is forwarded.

Field names are not case-sensitive, although their values may be.

6.7 Accept

The Accept header follows the syntax defined in [H14.1]. The semantics are also identical, with the exception that if no Accept header is present, the server SHOULD assume a default value of application/sdp.

This request-header field is used only with the INVITE, OPTIONS and REGISTER request methods to indicate what media types are acceptable in the response.

Example:

```
Accept: application/sdp;level=1, application/x-private, text/html
```

6.8 Accept-Encoding

The Accept-Encoding request-header field is similar to Accept, but restricts the content-codings [H3.4.1] that are acceptable in the response. See [H14.3]. The syntax of this header is defined in [H14.3]. The semantics in SIP are identical to those defined in [H14.3].

6.9 Accept-Language

The Accept-Language header follows the syntax defined in [H14.4]. The rules for ordering the languages based on the q parameter apply to SIP as well. When used in SIP, the Accept-Language request-header field can be used to allow the client to indicate to the server in which language it would prefer to receive reason phrases, session descriptions or status responses carried as message bodies. A proxy MAY use this field to help select the destination for the call, for example, a human operator conversant in a language spoken by the caller.

Example:

```
Accept-Language: da, en-gb;q=0.8, en;q=0.7
```

6.10 Allow

The Allow entity-header field lists the set of methods supported by the resource identified by the Request-URI. The purpose of this field is strictly to inform the recipient of valid methods associated with the resource. An Allow header field MUST be present in a 405 (Method Not Allowed) response and SHOULD be present in an OPTIONS response.

```
Allow = "Allow" ":" 1#Method
```

6.11 Authorization

A user agent that wishes to authenticate itself with a server -- usually, but not necessarily, after receiving a 401 response -- MAY do so by including an Authorization request-header field with the request. The Authorization field value consists of credentials containing the authentication information of the user agent for the realm of the resource being requested.

Section 13.2 overviews the use of the Authorization header, and section 15 describes the syntax and semantics when used with PGP based authentication.

6.12 Call-ID

The Call-ID general-header field uniquely identifies a particular invitation or all registrations of a particular client. Note that a single multimedia conference can give rise to several calls with different Call-IDs, e.g., if a user invites a single individual several times to the same (long-running) conference.

For an INVITE request, a callee user agent server SHOULD NOT alert the user if the user has responded previously to the Call-ID in the INVITE request. If the user is already a member of the conference and the conference parameters contained in the session description have not changed, a callee user agent server MAY silently accept the call, regardless of the Call-ID. An invitation for an existing Call-ID or session can change the parameters of the conference. A client application MAY decide to simply indicate to the user that the conference parameters have been changed and accept the invitation automatically or it MAY require user confirmation.

A user may be invited to the same conference or call using several different Call-IDs. If desired, the client MAY use identifiers within the session description to detect this duplication. For example, SDP contains a session id and version number in the origin (o) field.

The REGISTER and OPTIONS methods use the Call-ID value to unambiguously match requests and responses. All REGISTER requests issued by a single client SHOULD use the same Call-ID, at least within the same boot cycle.

> Since the Call-ID is generated by and for SIP, there is no reason to deal with the complexity of URL-encoding and case-ignoring string comparison.

```
Call-ID   =  ( "Call-ID" | "i" ) ":" local-id "@" host
local-id  =  1*uric
```

"host" SHOULD be either a fully qualified domain name or a globally routable IP address. If this is the case, the "local-id" SHOULD be an identifier consisting of URI characters that is unique within "host". Use of cryptographically random identifiers [27] is RECOMMENDED. If, however, host is not an FQDN or globally routable IP address (such as a net 10 address), the local-id MUST be globally unique, as opposed

to unique within host. These rules guarantee overall global

uniqueness of the Call-ID. The value for Call-ID MUST NOT be reused for a different call. Call-IDs are case-sensitive.

Using cryptographically random identifiers provides some protection against session hijacking. Call-ID, To and From are needed to identify a call leg. The distinction between call and call leg matters in calls with third-party control.

For systems which have tight bandwidth constraints, many of the mandatory SIP headers have a compact form, as discussed in Section 9. These are alternate names for the headers which occupy less space in the message. In the case of Call-ID, the compact form is i.

For example, both of the following are valid:

Call-ID: f81d4fae-7dec-11d0-a765-00a0c91e6bf6@foo.bar.com

or

i:f81d4fae-7dec-11d0-a765-00a0c91e6bf6@foo.bar.com

6.13 Contact

The Contact general-header field can appear in INVITE, ACK, and REGISTER requests, and in 1xx, 2xx, 3xx, and 485 responses. In general, it provides a URL where the user can be reached for further communications.

INVITE and ACK requests: INVITE and ACK requests MAY contain Contact headers indicating from which location the request is originating.

This allows the callee to send future requests, such as BYE, directly to the caller instead of through a series of proxies. The Via header is not sufficient since the desired address may be that of a proxy.

INVITE 2xx responses: A user agent server sending a definitive, positive response (2xx) MAY insert a Contact response header field indicating the SIP address under which it is reachable most directly for future SIP requests, such as ACK, within the

same Call-ID. The Contact header field contains the address of the server itself or that of a proxy, e.g., if the host is behind a firewall. The value of this Contact header is copied into the Request-URI of subsequent requests for this call if the response did not also contain a Record-Route header. If the

response also contains a Record-Route header field, the address in the Contact header field is added as the last item in the Route header field. See Section 6.29 for details.

The Contact value SHOULD NOT be cached across calls, as it may not represent the most desirable location for a particular destination address.

INVITE lxx responses: A UAS sending a provisional response (1xx) MAY insert a Contact response header. It has the same semantics in a 1xx response as a 2xx INVITE response. Note that CANCEL requests MUST NOT be sent to that address, but rather follow the same path as the original request.

REGISTER requests: REGISTER requests MAY contain a Contact header field indicating at which locations the user is reachable. The REGISTER request defines a wildcard Contact field, "*", which MUST only be used with Expires: 0 to remove all registrations for a particular user. An optional "expires" parameter indicates the desired expiration time of the registration. If a Contact entry does not have an "expires" parameter, the Expires header field is used as the default value. If neither of these mechanisms is used, SIP URIs are assumed to expire after one hour. Other URI schemes have no expiration times.

REGISTER 2xx responses: A REGISTER response MAY return all locations at which the user is currently reachable. An optional "expires" parameter indicates the expiration time of the registration. If a Contact entry does not have an "expires" parameter, the value of the Expires header field indicates the expiration time. If neither mechanism is used, the expiration time specified in the request, explicitly or by default, is used.

3xx and 485 responses: The Contact response-header field can be used with a 3xx or 485 (Ambiguous) response codes to indicate one or more alternate addresses to try. It can appear in responses to BYE, INVITE and OPTIONS methods. The Contact header field contains URIs giving the new locations or user names to try, or may simply specify additional transport parameters. A 300 (Multiple Choices), 301 (Moved Permanently), 302 (Moved Temporarily) or 485 (Ambiguous) response SHOULD contain a Contact field containing URIs of new addresses to be tried. A

301 or 302 response may also give the same location and username that was being tried but specify additional transport parameters such as a different server or multicast address to try or a change of SIP transport from UDP to TCP or vice versa. The client copies the "user", "password", "host", "port" and "user-param" elements of the Contact URI into the Request-URI of the redirected request and directs the request to the address specified by the "maddr" and "port" parameters, using the transport protocol given in the "transport" parameter. If

"maddr" is a multicast address, the value of "ttl" is used as the time-to-live value.

Note that the Contact header field MAY also refer to a different entity than the one originally called. For example, a SIP call connected to GSTN gateway may need to deliver a special information announcement such as "The number you have dialed has been changed."

A Contact response header field can contain any suitable URI indicating where the called party can be reached, not limited to SIP URLs. For example, it could contain URL's for phones, fax, or irc (if they were defined) or a mailto: (RFC 2368, [28]) URL.

The following parameters are defined. Additional parameters may be defined in other specifications.

q: The "qvalue" indicates the relative preference among the locations given. "qvalue" values are decimal numbers from 0 to 1, with higher values indicating higher preference.

action: The "action" parameter is used only when registering with the REGISTER request. It indicates whether the client wishes that the server proxy or redirect future requests intended for the client. If this parameter is not specified the action taken depends on server configuration. In its response, the registrar SHOULD indicate the mode used. This parameter is ignored for other requests.

expires: The "expires" parameter indicates how long the URI is valid. The parameter is either a number indicating seconds or a quoted string containing a SIP-date. If this parameter is not provided, the value of the Expires header field determines how long the URI is valid. Implementations MAY treat values larger than 2**32-1 (4294967295 seconds or 136 years) as equivalent to 2**32-1.

```
Contact = ( "Contact" | "m" ) ":"
          ("*" | (1# (( name-addr | addr-spec )
          [ *( ";" contact-params ) ] [ comment ] )))

name-addr      = [ display-name ] "<" addr-spec ">"
addr-spec      = SIP-URL | URI
display-name   = *token | quoted-string

contact-params = "q"       "=" qvalue
               | "action"  "=" "proxy" | "redirect"
               | "expires" "=" delta-seconds | <"> SIP-date <">
               | extension-attribute

extension-attribute = extension-name [ "=" extension-value ]
```

only allows one address, unquoted. Since URIs can contain commas and semicolons as reserved characters, they can be mistaken for header or parameter delimiters, respectively. The current syntax corresponds to that for the To and From header, which also allows the use of display names.

Example:

```
Contact: "Mr. Watson" <sip:watson@worcester.bell-telephone.com>
   ;q=0.7; expires=3600,
   "Mr. Watson" <mailto:watson@bell-telephone.com> ;q=0.1
```

6.14 Content-Encoding

```
Content-Encoding  =  ( "Content-Encoding" | "e" ) ":"
                     1#content-coding
```

The Content-Encoding entity-header field is used as a modifier to the "media-type". When present, its value indicates what additional content codings have been applied to the entity-body, and thus what decoding mechanisms MUST be applied in order to obtain the media-type referenced by the Content-Type header field. Content-Encoding is primarily used to allow a body to be compressed without losing the identity of its underlying media type.

If multiple encodings have been applied to an entity, the content codings MUST be listed in the order in which they were applied.

All content-coding values are case-insensitive. The Internet Assigned Numbers Authority (IANA) acts as a registry for content-coding value tokens. See [3.5] for a definition of the syntax for content-coding.

Clients MAY apply content encodings to the body in requests. If the server is not capable of decoding the body, or does not recognize any of the content-coding values, it MUST send a 415 "Unsupported Media Type" response, listing acceptable encodings in the Accept-Encoding

header. A server MAY apply content encodings to the bodies in responses. The server MUST only use encodings listed in the Accept-Encoding header in the request.

6.15 Content-Length

The Content-Length entity-header field indicates the size of the message-body, in decimal number of octets, sent to the recipient.

```
Content-Length  =  ( "Content-Length" | "l" ) ":" 1*DIGIT
```

An example is

```
Content-Length: 3495
```

Applications SHOULD use this field to indicate the size of the message-body to be transferred, regardless of the media type of the entity. Any Content-Length greater than or equal to zero is a valid value. If no body is present in a message, then the Content-Length header field MUST be set to zero. If a server receives a UDP request without Content-Length, it MUST assume that the request encompasses the remainder of the packet. If a server receives a UDP request with a Content-Length, but the value is larger than the size of the body sent in the request, the client SHOULD generate a 400 class response. If there is additional data in the UDP packet after the last byte of the body has been read, the server MUST treat the remaining data as a separate message. This allows several messages to be placed in a single UDP packet.

If a response does not contain a Content-Length, the client assumes that it encompasses the remainder of the UDP packet or the data until the TCP connection is closed, as applicable. Section 8 describes how to determine the length of the message body.

6.16 Content-Type

The Content-Type entity-header field indicates the media type of the message-body sent to the recipient. The "media-type" element is defined in [H3.7].

```
Content-Type  =  ( "Content-Type" | "c" ) ":" media-type
```


Examples of this header field are

```
Content-Type: application/sdp
Content-Type: text/html; charset=ISO-8859-4
```

6.17 CSeq

Clients MUST add the CSeq (command sequence) general-header field to every request. A CSeq header field in a request contains the request method and a single decimal sequence number chosen by the requesting client, unique within a single value of Call-ID. The sequence number MUST be expressible as a 32-bit unsigned integer. The initial value of the sequence number is arbitrary, but MUST be less than 2**31. Consecutive requests that differ in request method, headers or body, but have the same Call-ID MUST contain strictly monotonically increasing and contiguous sequence numbers; sequence numbers do not wrap around. Retransmissions of the same request carry the same sequence number, but an INVITE with a different message body or different header fields (a "re-invitation") acquires a new, higher

sequence number. A server MUST echo the CSeq value from the request in its response. If the Method value is missing in the received CSeq header field, the server fills it in appropriately.

The ACK and CANCEL requests MUST contain the same CSeq value as the INVITE request that it refers to, while a BYE request cancelling an invitation MUST have a higher sequence number. A BYE request with a CSeq that is not higher should cause a 400 response to be generated.

A user agent server MUST remember the highest sequence number for any INVITE request with the same Call-ID value. The server MUST respond to, and then discard, any INVITE request with a lower sequence number.

All requests spawned in a parallel search have the same CSeq value as the request triggering the parallel search.

CSeq = "CSeq" ":" 1*DIGIT Method

Strictly speaking, CSeq header fields are needed for any SIP request that can be cancelled by a BYE or CANCEL request or where a client can issue several requests for the same Call-ID in close succession. Without a sequence

number, the response to an INVITE could be mistaken for the response to the cancellation (BYE or CANCEL). Also, if the network duplicates packets or if an ACK is delayed until the server has sent an additional response, the client could interpret an old response as the response to a re-invitation issued shortly thereafter. Using CSeq also makes it easy for the server to distinguish different versions of an invitation, without comparing the message body.

The Method value allows the client to distinguish the response to an INVITE request from that of a CANCEL response. CANCEL requests can be generated by proxies; if they were to increase the sequence number, it might conflict with a later request issued by the user agent for the same call.

With a length of 32 bits, a server could generate, within a single call, one request a second for about 136 years before needing to wrap around. The initial value of the sequence number is chosen so that subsequent requests within the same call will not wrap around. A non-zero initial value allows to use a time-based initial sequence number, if the client desires. A client could, for example, choose the 31 most significant bits of a 32-bit second clock as an initial sequence number.

Forked requests MUST have the same CSeq as there would be ambiguity

otherwise between these forked requests and later BYE issued by the client user agent.

Example:

```
CSeq: 4711 INVITE
```

6.18 Date

Date is a general-header field. Its syntax is:

```
SIP-date  =  rfc1123-date
```

See [H14.19] for a definition of rfc1123-date. Note that unlike HTTP/1.1, SIP only supports the most recent RFC1123 [29] formatting for dates.

The Date header field reflects the time when the request or response is first sent. Thus, retransmissions have the same Date header field value as the original.

> The Date header field can be used by simple end systems without a battery-backed clock to acquire a notion of current time.

6.19 Encryption

The Encryption general-header field specifies that the content has been encrypted. Section 13 describes the overall SIP security architecture and algorithms. This header field is intended for end-to-end encryption of requests and responses. Requests are encrypted based on the public key belonging to the entity named in the To header field. Responses are encrypted based on the public key conveyed in the Response-Key header field. Note that the public keys themselves may not be used for the encryption. This depends on the particular algorithms used.

For any encrypted message, at least the message body and possibly other message header fields are encrypted. An application receiving a request or response containing an Encryption header field decrypts the body and then concatenates the plaintext to the request line and headers of the original message. Message headers in the decrypted part completely replace those with the same field name in the plaintext part. (Note: If only the body of the message is to be encrypted, the body has to be prefixed with CRLF to allow proper

concatenation.) Note that the request method and Request-URI cannot be encrypted.

> Encryption only provides privacy; the recipient has no guarantee that the request or response came from the party listed in the From message header, only that the sender used the recipient's public key. However, proxies will not be able to modify the request or response.

```
Encryption         =  "Encryption" ":" encryption-scheme 1*SP
                      #encryption-params
encryption-scheme  =  token
encryption-params  =  token "=" ( token | quoted-string )
```

> The token indicates the form of encryption used; it is described in section 13.

The example in Figure 10 shows a message encrypted with ASCII-armored PGP that was generated by applying "pgp -ea" to the payload to be encrypted.

```
INVITE sip:watson@boston.bell-telephone.com SIP/2.0
Via: SIP/2.0/UDP 169.130.12.5
From: <sip:a.g.bell@bell-telephone.com>
To: T. A. Watson <sip:watson@bell-telephone.com>
Call-ID: 187602141351@worcester.bell-telephone.com
Content-Length: 885
Encryption: PGP version=2.6.2,encoding=ascii

hQEMAxkp5GPd+j5xAQf/ZDIfGD/PDOM1wayvwdQAKgGgjmZWe+MTy9NEX8O25Red
h0/pyrd/+DV5C2BYs7yzSOSXajlC/tTK/4do6rtjhP8QA3vbDdVdaFciwEVAcuXs
ODxlNAVqyDi1RqFC28BJIvQ5KfEkPuACKTK7WlRSBc7vNPEA3nyqZGBTwhxRSbIR
RuFEsHSVojdCam4htcqxGnFwD9sksqs6LIyCFaiTAhWtwcCaN437G7mUYzy2KLcA
zPVGq1VQg83b99zPzIxRdlZ+K7+bAnu8Rtu+ohOCMLV3TPXbyp+err1YiThCZHIu
X9dOVj3CMjCP66RSHa/ea0wYTRRNYA/G+kdP8DSUcqYAAAE/hZPX6nFIqk7AVnf6
IpWHUPTelNUJpzUp5Ou+q/5P7ZAsn+cSAuF2YWtVjCf+SQmBR13p2EYYWHoxlA2/
GgKADYe4M3JSwOtqwU8zUJF3FIfk7vsxmSqtUQrRQaiIhqNyG7KxJt4YjWnEjF5E
WUIPhvyGFMJaeQXIyGRYZAYvKKklyAJcm29zLACxU5alX4M25lHQd9FR9Zmq6Jed
wbWvia6cAIfsvlZ9JGocmQYF7pcuz5pnczqP+/yvRqFJtDGD/v3s++G2R+ViVYJO
z/lxGUZaM4IWBCf+4DUjNanZM0oxAE28NjaIZ0rrldDQmO8V9FtPKdHxkqA5iJP+
6vGOFti1Ak4kmEz0vM/Nsv7kkubTFhRl05OiJIGr9S1UhenlZv9l6RuXsOY/EwH2
z8X9N4MhMyXEVuC9rt8/AUhmVQ==
=bOW+
```

Figure 10: PGP Encryption Example

T-Mobile Exhibit 1002
Part 3 of 6

Since proxies can base their forwarding decision on any combination of SIP header fields, there is no guarantee that an encrypted request "hiding" header fields will reach the same destination as an otherwise identical un-encrypted request.

6.20 Expires

The Expires entity-header field gives the date and time after which the message content expires.

This header field is currently defined only for the REGISTER and INVITE methods. For REGISTER, it is a request and response-header field. In a REGISTER request, the client indicates how long it wishes the registration to be valid. In the response, the server indicates

the earliest expiration time of all registrations. The server MAY choose a shorter time interval than that requested by the client, but SHOULD NOT choose a longer one.

For INVITE requests, it is a request and response-header field. In a request, the caller can limit the validity of an invitation, for example, if a client wants to limit the time duration of a search or a conference invitation. A user interface MAY take this as a hint to leave the invitation window on the screen even if the user is not currently at the workstation. This also limits the duration of a search. If the request expires before the search completes, the proxy returns a 408 (Request Timeout) status. In a 302 (Moved Temporarily) response, a server can advise the client of the maximal duration of the redirection.

The value of this field can be either a SIP-date or an integer number of seconds (in decimal), measured from the receipt of the request. The latter approach is preferable for short durations, as it does not depend on clients and servers sharing a synchronized clock. Implementations MAY treat values larger than 2**32-1 (4294967295 or 136 years) as equivalent to 2**32-1.

```
Expires  =  "Expires" ":" ( SIP-date | delta-seconds )
```

Two examples of its use are

```
Expires: Thu, 01 Dec 1994 16:00:00 GMT
Expires: 5
```

6.21 From

Requests and responses MUST contain a From general-header field,

indicating the initiator of the request. The From field MAY contain the "tag" parameter. The server copies the From header field from the request to the response. The optional "display-name" is meant to be rendered by a human-user interface. A system SHOULD use the display name "Anonymous" if the identity of the client is to remain hidden.

The SIP-URL MUST NOT contain the "transport-param", "maddr-param", "ttl-param", or "headers" elements. A server that receives a SIP-URL with these elements removes them before further processing.

Even if the "display-name" is empty, the "name-addr" form MUST be used if the "addr-spec" contains a comma, question mark, or semicolon.

```
From         =  ( "From" | "f" ) ":" ( name-addr | addr-spec )
                *( ";" addr-params )
addr-params  =  tag-param
tag-param    =  "tag=" UUID
UUID         =  1*( hex | "-" )
```

Examples:

```
From: "A. G. Bell" <sip:agb@bell-telephone.com>
From: sip:+12125551212@server.phone2net.com
From: Anonymous <sip:c8oqz84zk7z@privacy.org>
```

The "tag" MAY appear in the From field of a request. It MUST be present when it is possible that two instances of a user sharing a SIP address can make call invitations with the same Call-ID.

The "tag" value MUST be globally unique and cryptographically random with at least 32 bits of randomness. A single user maintains the same tag throughout the call identified by the Call-ID.

Call-ID, To and From are needed to identify a call leg. The distinction between call and call leg matters in calls with multiple responses to a forked request. The format is similar to the equivalent RFC 822 [24] header, but with a URI instead of just an email address.

6.22 Hide

A client uses the Hide request header field to indicate that it wants the path comprised of the Via header fields (Section 6.40) to be

hidden from subsequent proxies and user agents. It can take two forms: Hide: route and Hide: hop. Hide header fields are typically added by the client user agent, but MAY be added by any proxy along the path.

If a request contains the "Hide: route" header field, all following proxies SHOULD hide their previous hop. If a request contains the "Hide: hop" header field, only the next proxy SHOULD hide the previous hop and then remove the Hide option unless it also wants to remain anonymous.

A server hides the previous hop by encrypting the "host" and "port" parts of the top-most Via header field with an algorithm of its choice. Servers SHOULD add additional "salt" to the "host" and "port" information prior to encryption to prevent malicious downstream proxies from guessing earlier parts of the path based on seeing identical encrypted Via headers. Hidden Via fields are marked with the "hidden" Via option, as described in Section 6.40.

A server that is capable of hiding Via headers MUST attempt to decrypt all Via headers marked as "hidden" to perform loop detection. Servers that are not capable of hiding can ignore hidden Via fields in their loop detection algorithm.

> If hidden headers were not marked, a proxy would have to decrypt all headers to detect loops, just in case one was encrypted, as the Hide: Hop option may have been removed along the way.

A host MUST NOT add such a "Hide: hop" header field unless it can guarantee it will only send a request for this destination to the same next hop. The reason for this is that it is possible that the request will loop back through this same hop from a downstream proxy. The loop will be detected by the next hop if the choice of next hop is fixed, but could loop an arbitrary number of times otherwise.

A client requesting "Hide: route" can only rely on keeping the request path private if it sends the request to a trusted proxy. Hiding the route of a SIP request is of limited value if the request results in data packets being exchanged directly between the calling and called user agent.

The use of Hide header fields is discouraged unless path privacy is truly needed; Hide fields impose extra processing costs and restrictions for proxies and can cause requests to generate 482 (Loop Detected) responses that could otherwise be avoided.

The encryption of Via header fields is described in more detail in Section 13.

The Hide header field has the following syntax:


```
Hide  =  "Hide" ":" ( "route" | "hop" )
```

6.23 Max-Forwards

The Max-Forwards request-header field may be used with any SIP method to limit the number of proxies or gateways that can forward the request to the next downstream server. This can also be useful when the client is attempting to trace a request chain which appears to be failing or looping in mid-chain.

```
Max-Forwards  =  "Max-Forwards" ":" 1*DIGIT
```

The Max-Forwards value is a decimal integer indicating the remaining number of times this request message is allowed to be forwarded.

Each proxy or gateway recipient of a request containing a Max-Forwards header field MUST check and update its value prior to forwarding the request. If the received value is zero (0), the recipient MUST NOT forward the request. Instead, for the OPTIONS and REGISTER methods, it MUST respond as the final recipient. For all other methods, the server returns 483 (Too many hops).

If the received Max-Forwards value is greater than zero, then the forwarded message MUST contain an updated Max-Forwards field with a value decremented by one (1).

Example:

```
Max-Forwards: 6
```

6.24 Organization

The Organization general-header field conveys the name of the organization to which the entity issuing the request or response belongs. It MAY also be inserted by proxies at the boundary of an organization.

The field MAY be used by client software to filter calls.

```
Organization  =  "Organization" ":" *TEXT-UTF8
```

6.25 Priority

The Priority request-header field indicates the urgency of the request as perceived by the client.

```
Priority        =  "Priority" ":" priority-value
priority-value  =  "emergency" | "urgent" | "normal"
                |  "non-urgent"
```

It is RECOMMENDED that the value of "emergency" only be used when life, limb or property are in imminent danger.

Examples:

```
Subject: A tornado is heading our way!
Priority: emergency

Subject: Weekend plans
Priority: non-urgent
```

These are the values of RFC 2076 [30], with the addition of "emergency".

6.26 Proxy-Authenticate

The Proxy-Authenticate response-header field MUST be included as part of a 407 (Proxy Authentication Required) response. The field value consists of a challenge that indicates the authentication scheme and parameters applicable to the proxy for this Request-URI.

Unlike its usage within HTTP, the Proxy-Authenticate header MUST be passed upstream in the response to the UAC. In SIP, only UAC's can authenticate themselves to proxies.

The syntax for this header is defined in [H14.33]. See 14 for further details on its usage.

A client SHOULD cache the credentials used for a particular proxy server and realm for the next request to that server. Credentials are, in general, valid for a specific value of the Request-URI at a particular proxy server. If a client contacts a proxy server that has required authentication in the past, but the client does not have credentials for the particular Request-URI, it MAY attempt to use the most-recently used credential. The server responds with 401 (Unauthorized) if the client guessed wrong.

> This suggested caching behavior is motivated by proxies restricting phone calls to authenticated users. It seems likely that in most cases, all destinations require the same password. Note that end-to-end authentication is likely to be destination-specific.

6.27 Proxy-Authorization

The Proxy-Authorization request-header field allows the client to identify itself (or its user) to a proxy which requires authentication. The Proxy-Authorization field value consists of credentials containing the authentication information of the user agent for the proxy and/or realm of the resource being requested.

Unlike Authorization, the Proxy-Authorization header field applies only to the next outbound proxy that demanded authentication using the Proxy- Authenticate field. When multiple proxies are used in a chain, the Proxy-Authorization header field is consumed by the first outbound proxy that was expecting to receive credentials. A proxy MAY relay the credentials from the client request to the next proxy if that is the mechanism by which the proxies cooperatively authenticate a given request.

See [H14.34] for a definition of the syntax, and section 14 for a discussion of its usage.

6.28 Proxy-Require

The Proxy-Require header field is used to indicate proxy-sensitive features that MUST be supported by the proxy. Any Proxy-Require header field features that are not supported by the proxy MUST be negatively acknowledged by the proxy to the client if not supported. Proxy servers treat this field identically to the Require field.

See Section 6.30 for more details on the mechanics of this message and a usage example.

6.29 Record-Route

The Record-Route request and response header field is added to a request by any proxy that insists on being in the path of subsequent requests for the same call leg. It contains a globally reachable Request-URI that identifies the proxy server. Each proxy server adds its Request-URI to the beginning of the list.

The server copies the Record-Route header field unchanged into the response. (Record-Route is only relevant for 2xx responses.)

The calling user agent client copies the Record-Route header into a Route header field of subsequent requests within the same call leg, reversing the order of requests, so that the first entry is closest to the user agent client. If the response contained a Contact header field, the calling user agent adds its content as the last Route header. Unless this would cause a loop, any client MUST send any subsequent requests for this call leg to the first Request-URI in the Route request header field and remove that entry.

The calling user agent MUST NOT use the Record-Route header field in requests that contain Route header fields.

> Some proxies, such as those controlling firewalls or in an automatic call distribution (ACD) system, need to maintain call state and thus need to receive any BYE and ACK packets for the call.

The Record-Route header field has the following syntax:

```
Record-Route  =  "Record-Route" ":" 1# name-addr
```

Proxy servers SHOULD use the "maddr" URL parameter containing their address to ensure that subsequent requests are guaranteed to reach exactly the same server.

Example for a request that has traversed the hosts ieee.org and bell-telephone.com , in that order:

```
Record-Route: <sip:a.g.bell@bell-telephone.com>,
  <sip:a.bell@ieee.org>
```


6.30 Require

The Require request-header field is used by clients to tell user agent servers about options that the client expects the server to support in order to properly process the request. If a server does

not understand the option, it MUST respond by returning status code 420 (Bad Extension) and list those options it does not understand in the Unsupported header.

```
Require  =  "Require" ":" 1#option-tag
```

Example:

```
C->S:   INVITE sip:watson@bell-telephone.com SIP/2.0
        Require: com.example.billing
        Payment: sheep_skins, conch_shells

S->C:   SIP/2.0 420 Bad Extension
        Unsupported: com.example.billing
```

This is to make sure that the client-server interaction will proceed without delay when all options are understood by both sides, and only slow down if options are not understood (as in the example above). For a well-matched client-server pair, the interaction proceeds quickly, saving a round-trip often required by negotiation mechanisms. In addition, it also removes ambiguity when the client requires features that the server does not understand. Some features, such as call handling fields, are only of interest to end systems.

Proxy and redirect servers MUST ignore features that are not understood. If a particular extension requires that intermediate devices support it, the extension MUST be tagged in the Proxy-Require field as well (see Section 6.28).

6.31 Response-Key

The Response-Key request-header field can be used by a client to request the key that the called user agent SHOULD use to encrypt the response with. The syntax is:


```
Response-Key  =  "Response-Key" ":" key-scheme 1*SP #key-param
key-scheme    =  token
key-param     =  token "=" ( token | quoted-string )
```

The "key-scheme" gives the type of encryption to be used for the response. Section 13 describes security schemes.

If the client insists that the server return an encrypted response,

it includes a

```
Require: org.ietf.sip.encrypt-response
```

header field in its request. If the server cannot encrypt for whatever reason, it MUST follow normal Require header field procedures and return a 420 (Bad Extension) response. If this Require header field is not present, a server SHOULD still encrypt if it can.

6.32 Retry-After

The Retry-After general-header field can be used with a 503 (Service Unavailable) response to indicate how long the service is expected to be unavailable to the requesting client and with a 404 (Not Found), 600 (Busy), or 603 (Decline) response to indicate when the called party anticipates being available again. The value of this field can be either an SIP-date or an integer number of seconds (in decimal) after the time of the response.

A REGISTER request MAY include this header field when deleting registrations with "Contact: * ;expires: 0". The Retry-After value then indicates when the user might again be reachable. The registrar MAY then include this information in responses to future calls.

An optional comment can be used to indicate additional information about the time of callback. An optional "duration" parameter indicates how long the called party will be reachable starting at the initial time of availability. If no duration parameter is given, the service is assumed to be available indefinitely.

```
Retry-After  =  "Retry-After" ":" ( SIP-date | delta-seconds )
                [ comment ] [ ";" "duration" "=" delta-seconds ]
```

Examples of its use are

```
Retry-After: Mon, 21 Jul 1997 18:48:34 GMT (I'm in a meeting)
```



```
Retry-After: Mon, 01 Jan 9999 00:00:00 GMT
  (Dear John: Don't call me back, ever)
Retry-After: Fri, 26 Sep 1997 21:00:00 GMT;duration=3600
Retry-After: 120
```

In the third example, the callee is reachable for one hour starting at 21:00 GMT. In the last example, the delay is 2 minutes.

6.33 Route

The Route request-header field determines the route taken by a

request. Each host removes the first entry and then proxies the request to the host listed in that entry, also using it as the Request-URI. The operation is further described in Section 6.29.

The Route header field has the following syntax:

```
Route  =  "Route" ":" 1# name-addr
```

6.34 Server

The Server response-header field contains information about the software used by the user agent server to handle the request. The syntax for this field is defined in [H14.39].

6.35 Subject

This is intended to provide a summary, or to indicate the nature, of the call, allowing call filtering without having to parse the session description. (Also, the session description does not have to use the same subject indication as the invitation.)

```
Subject  =  ( "Subject" | "s" ) ":" *TEXT-UTF8
```

Example:

```
Subject: Tune in - they are talking about your work!
```


6.36 Timestamp

The timestamp general-header field describes when the client sent the request to the server. The value of the timestamp is of significance only to the client and it MAY use any timescale. The server MUST echo the exact same value and MAY, if it has accurate information about this, add a floating point number indicating the number of seconds that have elapsed since it has received the request. The timestamp is used by the client to compute the round-trip time to the server so that it can adjust the timeout value for retransmissions.

```
Timestamp  =  "Timestamp" ":" *(DIGIT) [ "." *(DIGIT) ] [ delay ]
delay      =  *(DIGIT) [ "." *(DIGIT) ]
```

Note that there MUST NOT be any LWS between a DIGIT and the decimal point.

6.37 To

The To general-header field specifies recipient of the request, with the same SIP URL syntax as the From field.

```
To  =  ( "To" | "t" ) ":" ( name-addr | addr-spec )
       *( ";" addr-params )
```

Requests and responses MUST contain a To general-header field, indicating the desired recipient of the request. The optional "display-name" is meant to be rendered by a human-user interface. The UAS or redirect server copies the To header field into its response, and MUST add a "tag" parameter if the request contained more than one Via header field.

> If there was more than one Via header field, the request was handled by at least one proxy server. Since the receiver cannot know whether any of the proxy servers forked the request, it is safest to assume that they might have.

The SIP-URL MUST NOT contain the "transport-param", "maddr-param", "ttl-param", or "headers" elements. A server that receives a SIP-URL with these elements removes them before further processing.

The "tag" parameter serves as a general mechanism to distinguish multiple instances of a user identified by a single SIP URL. As proxies can fork requests, the same request can reach multiple instances of a user (mobile and home phones, for example). As each can respond, there needs to be a means to distinguish the responses from each at the caller. The situation also arises with multicast requests. The tag in the To header field serves to distinguish responses at the UAC. It MUST be placed in the To field of the response by each instance when there is a possibility that the request was forked at an intermediate proxy. The "tag" MUST be added by UAS, registrars and redirect servers, but MUST NOT be inserted into responses forwarded upstream by proxies. The "tag" is added for all definitive responses for all methods, and MAY be added for informational responses from a UAS or redirect server. All subsequent transactions between two entities MUST include the "tag" parameter, as described in Section 11.

See Section 6.21 for details of the "tag" parameter.

The "tag" parameter in To headers is ignored when matching responses to requests that did not contain a "tag" in their To header.

A SIP server returns a 400 (Bad Request) response if it receives a request with a To header field containing a URI with a scheme it does not recognize.

Even if the "display-name" is empty, the "name-addr" form MUST be used if the "addr-spec" contains a comma, question mark, or semicolon.

The following are examples of valid To headers:

```
To: The Operator <sip:operator@cs.columbia.edu>;tag=287447
To: sip:+12125551212@server.phone2net.com
```

Call-ID, To and From are needed to identify a call leg. The distinction between call and call leg matters in calls with multiple responses from a forked request. The "tag" is added to the To header field in the response to allow forking of future requests for the same call by proxies, while addressing only one of the possibly several responding user agent servers. It also allows several instances of the callee to send requests that can be distinguished.

6.38 Unsupported

The Unsupported response-header field lists the features not supported by the server. See Section 6.30 for a usage example and motivation.

Syntax:

```
Unsupported  =  "Unsupported" ":" 1#option-tag
```

6.39 User-Agent

The User-Agent general-header field contains information about the client user agent originating the request. The syntax and semantics are defined in [H14.42].

6.40 Via

The Via field indicates the path taken by the request so far. This prevents request looping and ensures replies take the same path as the requests, which assists in firewall traversal and other unusual routing situations.

6.40.1 Requests

The client originating the request MUST insert into the request a Via field containing its host name or network address and, if not the default port number, the port number at which it wishes to receive responses. (Note that this port number can differ from the UDP source port number of the request.) A fully-qualified domain name is RECOMMENDED. Each subsequent proxy server that sends the request onwards MUST add its own additional Via field before any existing Via fields. A proxy that receives a redirection (3xx) response and then searches recursively, MUST use the same Via headers as on the original proxied request.

A proxy SHOULD check the top-most Via header field to ensure that it contains the sender's correct network address, as seen from that proxy. If the sender's address is incorrect, the proxy MUST add an additional "received" attribute, as described 6.40.2.

> A host behind a network address translator (NAT) or firewall may not be able to insert a network address into the Via header that can be reached by the next hop beyond

> the NAT. Use of the received attribute allows SIP requests to traverse NAT's which only modify the source IP address. NAT's which modify port numbers, called Network Address Port Translator's (NAPT) will not properly pass SIP when transported on UDP, in which case an application layer gateway is required. When run over TCP, SIP stands a better chance of traversing NAT's, since its behavior is similar to HTTP in this case (but of course on different ports).

A proxy sending a request to a multicast address MUST add the "maddr" parameter to its Via header field, and SHOULD add the "ttl" parameter. If a server receives a request which contained an "maddr" parameter in the topmost Via field, it SHOULD send the response to the multicast address listed in the "maddr" parameter.

If a proxy server receives a request which contains its own address in the Via header value, it MUST respond with a 482 (Loop Detected) status code.

A proxy server MUST NOT forward a request to a multicast group which already appears in any of the Via headers.

> This prevents a malfunctioning proxy server from causing loops. Also, it cannot be guaranteed that a proxy server can always detect that the address returned by a location service refers to a host listed in the Via list, as a single host may have aliases or several network interfaces.

6.40.2 Receiver-tagged Via Header Fields

Normally, every host that sends or forwards a SIP message adds a Via field indicating the path traversed. However, it is possible that Network Address Translators (NATs) changes the source address and port of the request (e.g., from net-10 to a globally routable address), in which case the Via header field cannot be relied on to route replies. To prevent this, a proxy SHOULD check the top-most Via header field to ensure that it contains the sender's correct network address, as seen from that proxy. If the sender's address is incorrect, the proxy MUST add a "received" parameter to the Via header field inserted by the previous hop. Such a modified Via header field is known as a receiver-tagged Via header field. An example is:

```
Via: SIP/2.0/UDP erlang.bell-telephone.com:5060
Via: SIP/2.0/UDP 10.0.0.1:5060 ;received=199.172.136.3
```


In this example, the message originated from 10.0.0.1 and traversed a NAT with the external address border.ieee.org (199.172.136.3) to reach erlang.bell-telephone.com. The latter noticed the mismatch, and added a parameter to the previous hop's Via header field, containing the address that the packet actually came from. (Note that the NAT border.ieee.org is not a SIP server.)

6.40.3 Responses

Via header fields in responses are processed by a proxy or UAC according to the following rules:

1. The first Via header field should indicate the proxy or client processing this response. If it does not, discard the message. Otherwise, remove this Via field.

2. If there is no second Via header field, this response is destined for this client. Otherwise, the processing depends on whether the Via field contains a "maddr" parameter or is a receiver-tagged field:

   - If the second Via header field contains a "maddr" parameter, send the response to the multicast address listed there, using the port indicated in "sent-by", or port 5060 if none is present. The response SHOULD be sent using the TTL indicated in the "ttl" parameter, or with a TTL of 1 if that parameter is not present. For robustness, responses MUST be sent to the address indicated in the "maddr" parameter even if it is not a multicast address.

   - If the second Via header field does not contain a "maddr" parameter and is a receiver-tagged field (Section

6.40.2), send the message to the address in the "received" parameter, using the port indicated in the "sent-by" value, or using port 5060 if none is present.

- If neither of the previous cases apply, send the message to the address indicated by the "sent-by" value in the second Via header field.

6.40.4 User Agent and Redirect Servers

A UAS or redirect server sends a response based on one of the following rules:

o If the first Via header field in the request contains a "maddr" parameter, send the response to the multicast address

listed there, using the port indicated in "sent-by", or port 5060 if none is present. The response SHOULD be sent using the TTL indicated in the "ttl" parameter, or with a TTL of 1 if that parameter is not present. For robustness, responses MUST be sent to the address indicated in the "maddr" parameter even if it is not a multicast address.

o If the address in the "sent-by" value of the first Via field differs from the source address of the packet, send the response to the actual packet source address, similar to the treatment for receiver-tagged Via header fields (Section 6.40.2).

o If neither of these conditions is true, send the response to the address contained in the "sent-by" value. If the request was sent using TCP, use the existing TCP connection if available.

6.40.5 Syntax

The format for a Via header field is shown in Fig. 11. The defaults for "protocol-name" and "transport" are "SIP" and "UDP", respectively. The "maddr" parameter, designating the multicast address, and the "ttl" parameter, designating the time-to-live (TTL) value, are included only if the request was sent via multicast. The "received" parameter is added only for receiver-added Via fields (Section 6.40.2). For reasons of privacy, a client or proxy may wish to hide its Via information by encrypting it (see Section 6.22). The "hidden" parameter is included if this header field was hidden by the upstream proxy (see 6.22). Note that privacy of the proxy relies on the cooperation of the next hop, as the next-hop proxy will, by necessity, know the IP address and port number of the source host.

The "branch" parameter is included by every forking proxy. The token MUST be unique for each distinct request generated when a proxy forks. CANCEL requests MUST have the same branch value as the

corresponding forked request. When a response arrives at the proxy it can use the branch value to figure out which branch the response corresponds to. A proxy which generates a single request (non-forking) MAY also insert the "branch" parameter. The identifier has to be unique only within a set of isomorphic requests.

```
Via: SIP/2.0/UDP first.example.com:4000;ttl=16
  ;maddr=224.2.0.1 ;branch=a7c6a8dlze (Example)
Via: SIP/2.0/UDP adk8
```



```
Via               = ( "Via" | "v") ":" 1#( sent-protocol sent-by
                    *( ";" via-params ) [ comment ] )
via-params        = via-hidden | via-ttl | via-maddr
                  | via-received | via-branch
via-hidden        = "hidden"
via-ttl           = "ttl" "=" ttl
via-maddr         = "maddr" "=" maddr
via-received      = "received" "=" host
via-branch        = "branch" "=" token
sent-protocol     = protocol-name "/" protocol-version "/" transport
protocol-name     = "SIP" | token
protocol-version  = token
transport         = "UDP" | "TCP" | token
sent-by           = ( host [ ":" port ] ) | ( concealed-host )
concealed-host    = token
ttl               = 1*3DIGIT     ; 0 to 255
```

Figure 11: Syntax of Via header field

6.41 Warning

The Warning response-header field is used to carry additional information about the status of a response. Warning headers are sent with responses and have the following format:

```
Warning        =  "Warning" ":" 1#warning-value
warning-value  =  warn-code SP warn-agent SP warn-text
warn-code      =  3DIGIT
warn-agent     =  ( host [ ":" port ] ) | pseudonym
                  ;  the name or pseudonym of the server adding
                  ;  the Warning header, for use in debugging
warn-text      =  quoted-string
```

A response MAY carry more than one Warning header.

The "warn-text" should be in a natural language that is most likely to be intelligible to the human user receiving the response. This decision can be based on any available knowledge, such as the location of the cache or user, the Accept-Language field in a request, or the Content-Language field in a response. The default language is i-default [31].

Any server MAY add Warning headers to a response. Proxy servers MUST place additional Warning headers before any Authorization headers. Within that constraint, Warning headers MUST be added after any existing Warning headers not covered by a signature. A proxy server MUST NOT delete any Warning header field that it received with a response.

When multiple Warning headers are attached to a response, the user agent SHOULD display as many of them as possible, in the order that they appear in the response. If it is not possible to display all of the warnings, the user agent first displays warnings that appear early in the response.

The warn-code consists of three digits. A first digit of "3" indicates warnings specific to SIP.

This is a list of the currently-defined "warn-code"s, each with a recommended warn-text in English, and a description of its meaning. Note that these warnings describe failures induced by the session description.

Warnings 300 through 329 are reserved for indicating problems with keywords in the session description, 330 through 339 are warnings related to basic network services requested in the session description, 370 through 379 are warnings related to quantitative QoS parameters requested in the session description, and 390 through 399 are miscellaneous warnings that do not fall into one of the above categories.

300 Incompatible network protocol: One or more network protocols contained in the session description are not available.

301 Incompatible network address formats: One or more network address formats contained in the session description are not available.

302 Incompatible transport protocol: One or more transport protocols described in the session description are not available.

303 Incompatible bandwidth units: One or more bandwidth measurement units contained in the session description were not understood.

304 Media type not available: One or more media types contained in the session description are not available.

305 Incompatible media format: One or more media formats contained in

the session description are not available.

306 Attribute not understood: One or more of the media attributes in the session description are not supported.

307 Session description parameter not understood: A parameter other than those listed above was not understood.

330 Multicast not available: The site where the user is located does not support multicast.

331 Unicast not available: The site where the user is located does not support unicast communication (usually due to the presence of a firewall).

370 Insufficient bandwidth: The bandwidth specified in the session description or defined by the media exceeds that known to be available.

399 Miscellaneous warning: The warning text can include arbitrary information to be presented to a human user, or logged. A system receiving this warning MUST NOT take any automated action.

1xx and 2xx have been taken by HTTP/1.1.

Additional "warn-code"s, as in the example below, can be defined through IANA.

Examples:

```
Warning: 307 isi.edu "Session parameter 'foo' not understood"
Warning: 301 isi.edu "Incompatible network address type 'E.164'"
```

6.42 WWW-Authenticate

The WWW-Authenticate response-header field MUST be included in 401 (Unauthorized) response messages. The field value consists of at least one challenge that indicates the authentication scheme(s) and parameters applicable to the Request-URI. See [H14.46] for a definition of the syntax, and section 14 for an overview of usage.

The content of the "realm" parameter SHOULD be displayed to the user. A user agent SHOULD cache the authorization credentials for a given value of the destination (To header) and "realm" and attempt to re-use these values on the next request for that destination.

In addition to the "basic" and "digest" authentication schemes defined in the specifications cited above, SIP defines a new scheme, PGP (RFC 2015, [32]), Section 15. Other schemes, such as S/MIME, are for further study.

7 Status Code Definitions

The response codes are consistent with, and extend, HTTP/1.1 response codes. Not all HTTP/1.1 response codes are appropriate, and only those that are appropriate are given here. Other HTTP/1.1 response codes SHOULD NOT be used. Response codes not defined by HTTP/1.1 have codes x80 upwards to avoid clashes with future HTTP response codes. Also, SIP defines a new class, 6xx. The default behavior for unknown response codes is given for each category of codes.

7.1 Informational 1xx

Informational responses indicate that the server or proxy contacted is performing some further action and does not yet have a definitive response. The client SHOULD wait for a further response from the server, and the server SHOULD send such a response without further prompting. A server SHOULD send a 1xx response if it expects to take more than 200 ms to obtain a final response. A server MAY issue zero or more 1xx responses, with no restriction on their ordering or uniqueness. Note that 1xx responses are not transmitted reliably, that is, they do not cause the client to send an ACK. Servers are free to retransmit informational responses and clients can inquire about the current state of call processing by re-sending the request.

7.1.1 100 Trying

Some unspecified action is being taken on behalf of this call (e.g., a database is being consulted), but the user has not yet been located.

7.1.2 180 Ringing

The called user agent has located a possible location where the user has registered recently and is trying to alert the user.

7.1.3 181 Call Is Being Forwarded

A proxy server MAY use this status code to indicate that the call is being forwarded to a different set of destinations.

7.1.4 182 Queued

The called party is temporarily unavailable, but the callee has decided to queue the call rather than reject it. When the callee becomes available, it will return the appropriate final status response. The reason phrase MAY give further details about the status of the call, e.g., "5 calls queued; expected waiting time is 15 minutes". The server MAY issue several 182 responses to update the caller about the status of the queued call.

7.2 Successful 2xx

The request was successful and MUST terminate a search.

7.2.1 200 OK

The request has succeeded. The information returned with the response depends on the method used in the request, for example:

BYE: The call has been terminated. The message body is empty.

CANCEL: The search has been cancelled. The message body is empty.

INVITE: The callee has agreed to participate; the message body indicates the callee's capabilities.

OPTIONS: The callee has agreed to share its capabilities, included in the message body.

REGISTER: The registration has succeeded. The client treats the message body according to its Content-Type.

7.3 Redirection 3xx

3xx responses give information about the user's new location, or about alternative services that might be able to satisfy the call. They SHOULD terminate an existing search, and MAY cause the initiator to begin a new search if appropriate.

Any redirection (3xx) response MUST NOT suggest any of the addresses in the Via (Section 6.40) path of the request in the Contact header field. (Addresses match if their host and port number match.)

To avoid forwarding loops, a user agent client or proxy MUST check whether the address returned by a redirect server equals an address tried earlier.

7.3.1 300 Multiple Choices

The address in the request resolved to several choices, each with its own specific location, and the user (or user agent) can select a preferred communication end point and redirect its request to that location.

The response SHOULD include an entity containing a list of resource characteristics and location(s) from which the user or user agent can choose the one most appropriate, if allowed by the Accept request header. The entity format is specified by the media type given in the Content-Type header field. The choices SHOULD also be listed as Contact fields (Section 6.13). Unlike HTTP, the SIP response MAY contain several Contact fields or a list of addresses in a Contact field. User agents MAY use the Contact header field value for automatic redirection or MAY ask the user to confirm a choice. However, this specification does not define any standard for such automatic selection.

This status response is appropriate if the callee can be reached at several different locations and the server cannot or prefers not to proxy the request.

7.3.2 301 Moved Permanently

The user can no longer be found at the address in the Request-URI and the requesting client SHOULD retry at the new address given by the Contact header field (Section 6.13). The caller SHOULD update any local directories, address books and user location caches with this new value and redirect future requests to the address(es) listed.

7.3.3 302 Moved Temporarily

The requesting client SHOULD retry the request at the new address(es) given by the Contact header field (Section 6.13). The duration of the redirection can be indicated through an Expires (Section 6.20) header. If there is no explicit expiration time, the address is only valid for this call and MUST NOT be cached for future calls.

7.3.4 305 Use Proxy

The requested resource MUST be accessed through the proxy given by the Contact field. The Contact field gives the URI of the proxy. The recipient is expected to repeat this single request via the proxy. 305 responses MUST only be generated by user agent servers.

7.3.5 380 Alternative Service

The call was not successful, but alternative services are possible. The alternative services are described in the message body of the response. Formats for such bodies are not defined here, and may be

the subject of future standardization.

7.4 Request Failure 4xx

4xx responses are definite failure responses from a particular server. The client SHOULD NOT retry the same request without modification (e.g., adding appropriate authorization). However, the same request to a different server might be successful.

7.4.1 400 Bad Request

The request could not be understood due to malformed syntax.

7.4.2 401 Unauthorized

The request requires user authentication.

7.4.3 402 Payment Required

Reserved for future use.

7.4.4 403 Forbidden

The server understood the request, but is refusing to fulfill it. Authorization will not help, and the request SHOULD NOT be repeated.

7.4.5 404 Not Found

The server has definitive information that the user does not exist at the domain specified in the Request-URI. This status is also returned if the domain in the Request-URI does not match any of the domains handled by the recipient of the request.

7.4.6 405 Method Not Allowed

The method specified in the Request-Line is not allowed for the address identified by the Request-URI. The response MUST include an Allow header field containing a list of valid methods for the indicated address.

7.4.7 406 Not Acceptable

The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.

7.4.8 407 Proxy Authentication Required

This code is similar to 401 (Unauthorized), but indicates that the

client MUST first authenticate itself with the proxy. The proxy MUST return a Proxy-Authenticate header field (section 6.26) containing a challenge applicable to the proxy for the requested resource. The client MAY repeat the request with a suitable Proxy-Authorization header field (section 6.27). SIP access authentication is explained in section 13.2 and 14.

This status code is used for applications where access to the communication channel (e.g., a telephony gateway) rather than the callee requires authentication.

7.4.9 408 Request Timeout

The server could not produce a response, e.g., a user location, within the time indicated in the Expires request-header field. The client MAY repeat the request without modifications at any later time.

7.4.10 409 Conflict

The request could not be completed due to a conflict with the current state of the resource. This response is returned if the action parameter in a REGISTER request conflicts with existing registrations.

7.4.11 410 Gone

The requested resource is no longer available at the server and no forwarding address is known. This condition is expected to be considered permanent. If the server does not know, or has no facility to determine, whether or not the condition is permanent, the status code 404 (Not Found) SHOULD be used instead.

7.4.12 411 Length Required

The server refuses to accept the request without a defined Content-Length. The client MAY repeat the request if it adds a valid Content-Length header field containing the length of the message-body in the request message.

7.4.13 413 Request Entity Too Large

The server is refusing to process a request because the request entity is larger than the server is willing or able to process. The server MAY close the connection to prevent the client from continuing the request.

If the condition is temporary, the server SHOULD include a Retry-After header field to indicate that it is temporary and after what time the client MAY try again.

7.4.14 414 Request-URI Too Long

The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.

7.4.15 415 Unsupported Media Type

The server is refusing to service the request because the message body of the request is in a format not supported by the requested resource for the requested method. The server SHOULD return a list of acceptable formats using the Accept, Accept-Encoding and Accept-Language header fields.

7.4.16 420 Bad Extension

The server did not understand the protocol extension specified in a Require (Section 6.30) header field.

7.4.17 480 Temporarily Unavailable

The callee's end system was contacted successfully but the callee is currently unavailable (e.g., not logged in or logged in in such a manner as to preclude communication with the callee). The response MAY indicate a better time to call in the Retry-After header. The user could also be available elsewhere (unbeknownst to this host), thus, this response does not terminate any searches. The reason phrase SHOULD indicate a more precise cause as to why the callee is unavailable. This value SHOULD be setable by the user agent. Status 486 (Busy Here) MAY be used to more precisely indicate a particular reason for the call failure.

This status is also returned by a redirect server that recognizes the user identified by the Request-URI, but does not currently have a valid forwarding location for that user.

7.4.18 481 Call Leg/Transaction Does Not Exist

This status is returned under two conditions: The server received a BYE request that does not match any existing call leg or the server received a CANCEL request that does not match any existing transaction. (A server simply discards an ACK referring to an unknown transaction.)

7.4.19 482 Loop Detected

The server received a request with a Via (Section 6.40) path containing itself.

7.4.20 483 Too Many Hops

The server received a request that contains more Via entries (hops) (Section 6.40) than allowed by the Max-Forwards (Section 6.23) header

field.

7.4.21 484 Address Incomplete

The server received a request with a To (Section 6.37) address or Request-URI that was incomplete. Additional information SHOULD be provided.

This status code allows overlapped dialing. With overlapped dialing, the client does not know the length of the dialing string. It sends strings of increasing lengths, prompting the user for more input, until it no longer receives a 484 status response.

7.4.22 485 Ambiguous

The callee address provided in the request was ambiguous. The response MAY contain a listing of possible unambiguous addresses in Contact headers.

Revealing alternatives can infringe on privacy concerns of the user or the organization. It MUST be possible to configure a server to respond with status 404 (Not Found) or to suppress the listing of possible choices if the request address was ambiguous.

Example response to a request with the URL lee@example.com :

```
485 Ambiguous SIP/2.0
Contact: Carol Lee <sip:carol.lee@example.com>
```



```
Contact: Ping Lee <sip:p.lee@example.com>
Contact: Lee M. Foote <sip:lee.foote@example.com>
```

Some email and voice mail systems provide this functionality. A status code separate from 3xx is used since the semantics are different: for 300, it is assumed that the same person or service will be reached by the choices provided. While an automated choice or sequential search makes sense for a 3xx response, user intervention is required for a 485 response.

7.4.23 486 Busy Here

The callee's end system was contacted successfully but the callee is currently not willing or able to take additional calls. The response MAY indicate a better time to call in the Retry-After header. The user could also be available elsewhere, such as through a voice mail service, thus, this response does not terminate any searches. Status 600 (Busy Everywhere) SHOULD be used if the client knows that no

other end system will be able to accept this call.

7.5 Server Failure 5xx

5xx responses are failure responses given when a server itself has erred. They are not definitive failures, and MUST NOT terminate a search if other possible locations remain untried.

7.5.1 500 Server Internal Error

The server encountered an unexpected condition that prevented it from fulfilling the request. The client MAY display the specific error condition, and MAY retry the request after several seconds.

7.5.2 501 Not Implemented

The server does not support the functionality required to fulfill the request. This is the appropriate response when the server does not recognize the request method and is not capable of supporting it for any user.

7.5.3 502 Bad Gateway

The server, while acting as a gateway or proxy, received an invalid response from the downstream server it accessed in attempting to fulfill the request.

7.5.4 503 Service Unavailable

The server is currently unable to handle the request due to a temporary overloading or maintenance of the server. The implication is that this is a temporary condition which will be alleviated after some delay. If known, the length of the delay MAY be indicated in a Retry-After header. If no Retry-After is given, the client MUST handle the response as it would for a 500 response.

Note: The existence of the 503 status code does not imply that a server has to use it when becoming overloaded. Some servers MAY wish to simply refuse the connection.

7.5.5 504 Gateway Time-out

The server, while acting as a gateway, did not receive a timely response from the server (e.g., a location server) it accessed in attempting to complete the request.

7.5.6 505 Version Not Supported

The server does not support, or refuses to support, the SIP protocol version that was used in the request message. The server is indicating that it is unable or unwilling to complete the request using the same major version as the client, other than with this

error message. The response MAY contain an entity describing why that version is not supported and what other protocols are supported by that server. The format for such an entity is not defined here and may be the subject of future standardization.

7.6 Global Failures 6xx

6xx responses indicate that a server has definitive information about a particular user, not just the particular instance indicated in the Request-URI. All further searches for this user are doomed to failure and pending searches SHOULD be terminated.

7.6.1 600 Busy Everywhere

The callee's end system was contacted successfully but the callee is busy and does not wish to take the call at this time. The response MAY indicate a better time to call in the Retry-After header. If the callee does not wish to reveal the reason for declining the call, the callee uses status code 603 (Decline) instead. This status response is returned only if the client knows that no other end point (such as a voice mail system) will answer the request. Otherwise, 486 (Busy Here) should be returned.

7.6.2 603 Decline

The callee's machine was successfully contacted but the user explicitly does not wish to or cannot participate. The response MAY indicate a better time to call in the Retry-After header.

7.6.3 604 Does Not Exist Anywhere

The server has authoritative information that the user indicated in the To request field does not exist anywhere. Searching for the user elsewhere will not yield any results.

7.6.4 606 Not Acceptable

The user's agent was contacted successfully but some aspects of the session description such as the requested media, bandwidth, or addressing style were not acceptable.

A 606 (Not Acceptable) response means that the user wishes to communicate, but cannot adequately support the session described. The 606 (Not Acceptable) response MAY contain a list of reasons in a Warning header field describing why the session described cannot be supported. Reasons are listed in Section 6.41. It is hoped that negotiation will not frequently be needed, and when a new user is being invited to join an already existing conference, negotiation may not be possible. It is up to the invitation initiator to decide whether or not to act on a 606 (Not Acceptable) response.

8 SIP Message Body

8.1 Body Inclusion

Requests MAY contain message bodies unless otherwise noted. Within this specification, the BYE request MUST NOT contain a message body. For ACK, INVITE and OPTIONS, the message body is always a session description. The use of message bodies for REGISTER requests is for further study.

For response messages, the request method and the response status code determine the type and interpretation of any message body. All responses MAY include a body. Message bodies for 1xx responses contain advisory information about the progress of the request. 2xx responses to INVITE requests contain session descriptions. In 3xx responses, the message body MAY contain the description of alternative destinations or services, as described in Section 7.3. For responses with status 400 or greater, the message body MAY

contain additional, human-readable information about the reasons for failure. It is RECOMMENDED that information in 1xx and 300 and greater responses be of type text/plain or text/html

8.2 Message Body Type

The Internet media type of the message body MUST be given by the Content-Type header field. If the body has undergone any encoding (such as compression) then this MUST be indicated by the Content-Encoding header field, otherwise Content-Encoding MUST be omitted. If applicable, the character set of the message body is indicated as part of the Content-Type header-field value.

8.3 Message Body Length

The body length in bytes SHOULD be given by the Content-Length header field. Section 6.15 describes the behavior in detail.

The "chunked" transfer encoding of HTTP/1.1 MUST NOT be used for SIP. (Note: The chunked encoding modifies the body of a message in order to transfer it as a series of chunks, each with its own size indicator.)

9 Compact Form

When SIP is carried over UDP with authentication and a complex session description, it may be possible that the size of a request or response is larger than the MTU. To address this problem, a more compact form of SIP is also defined by using abbreviations for the common header fields listed below:

short field name long field name note

```
c                 Content-Type
e                 Content-Encoding
f                 From
i                 Call-ID
m                 Contact           from "moved"
l                 Content-Length
s                 Subject
t                 To
v                 Via
```

Thus, the message in section 16.2 could also be written:


```
INVITE sip:schooler@vlsi.caltech.edu SIP/2.0
v:SIP/2.0/UDP 131.215.131.131;maddr=239.128.16.254;ttl=16
v:SIP/2.0/UDP 128.16.64.19
f:sip:mjh@isi.edu
t:sip:schooler@cs.caltech.edu
i:62729-27@128.16.64.19
c:application/sdp
CSeq: 4711 INVITE
l:187

v=0
o=user1 53655765 2353687637 IN IP4 128.3.4.5
s=Mbone Audio
i=Discussion of Mbone Engineering Issues
e=mbone@somewhere.com
c=IN IP4 224.2.0.1/127
t=0 0
m=audio 3456 RTP/AVP 0
```

Clients MAY mix short field names and long field names within the same request. Servers MUST accept both short and long field names for requests. Proxies MAY change header fields between their long and short forms, but this MUST NOT be done to fields following an Authorization header.

10 Behavior of SIP Clients and Servers

10.1 General Remarks

SIP is defined so it can use either UDP (unicast or multicast) or TCP as a transport protocol; it provides its own reliability mechanism.

10.1.1 Requests

Servers discard isomorphic requests, but first retransmit the

appropriate response. (SIP requests are said to be idempotent , i.e., receiving more than one copy of a request does not change the server state.)

After receiving a CANCEL request from an upstream client, a stateful proxy server MAY send a CANCEL on all branches where it has not yet received a final response.

When a user agent receives a request, it checks the Call-ID against those of in-progress calls. If the Call-ID was found, it compares the tag value of To with the user's tag and rejects the request if the

two do not match. If the From header, including any tag value, matches the value for an existing call leg, the server compares the CSeq header field value. If less than or equal to the current sequence number, the request is a retransmission. Otherwise, it is a new request. If the From header does not match an existing call leg, a new call leg is created.

If the Call-ID was not found, a new call leg is created, with entries for the To, From and Call-ID headers. In this case, the To header field should not have contained a tag. The server returns a response containing the same To value, but with a unique tag added. The tag MAY be omitted if the request contained only one Via header field.

10.1.2 Responses

A server MAY issue one or more provisional responses at any time before sending a final response. If a stateful proxy, user agent server, redirect server or registrar cannot respond to a request with a final response within 200 ms, it SHOULD issue a provisional (1xx) response as soon as possible. Stateless proxies MUST NOT issue provisional responses on their own.

Responses are mapped to requests by the matching To, From, Call-ID, CSeq headers and the branch parameter of the first Via header. Responses terminate request retransmissions even if they have Via headers that cause them to be delivered to an upstream client.

A stateful proxy may receive a response that it does not have state for, that is, where it has no a record of an associated request. If the Via header field indicates that the upstream server used TCP, the proxy actively opens a TCP connection to that address. Thus, proxies have to be prepared to receive responses on the incoming side of passive TCP connections, even though most responses will arrive on the incoming side of an active connection. (An active connection is a TCP connection initiated by the proxy, a passive connection is one accepted by the proxy, but initiated by another entity.)

100 responses SHOULD NOT be forwarded, other 1xx responses MAY be forwarded, possibly after the server eliminates responses with status codes that had already been sent earlier. 2xx responses are forwarded according to the Via header. Once a stateful proxy has received a 2xx

response, it MUST NOT forward non-2xx final responses. Responses with status 300 and higher are retransmitted by each stateful proxy until the next upstream proxy sends an ACK (see below for timing details) or CANCEL.

A stateful proxy SHOULD maintain state for at least 32 seconds after the receipt of the first definitive non-200 response, in order to handle retransmissions of the response.

> The 32 second window is given by the maximum retransmission duration of 200-class responses using the default timers, in case the ACK is lost somewhere on the way to the called user agent or the next stateful proxy.

10.2 Source Addresses, Destination Addresses and Connections

10.2.1 Unicast UDP

Responses are returned to the address listed in the Via header field (Section 6.40), not the source address of the request.

> Recall that responses are not generated by the next-hop stateless server, but generated by either a proxy server or the user agent server. Thus, the stateless proxy can only use the Via header field to forward the response.

10.2.2 Multicast UDP

Requests MAY be multicast; multicast requests likely feature a host-independent Request-URI. This request SHOULD be scoped to ensure it is not forwarded beyond the boundaries of the administrative system. This MAY be done with either TTL or administrative scopes[25], depending on what is implemented in the network.

A client receiving a multicast query does not have to check whether the host part of the Request-URI matches its own host or domain name. If the request was received via multicast, the response is also returned via multicast. Responses to multicast requests are multicast with the same TTL as the request, where the TTL is derived from the ttl parameter in the Via header (Section 6.40).

To avoid response implosion, servers MUST NOT answer multicast requests with a status code other than 2xx or 6xx. The server delays its response by a random interval uniformly distributed between zero and one second. Servers MAY suppress responses if they hear a lower-numbered or 6xx response from another group member prior to sending. Servers do not respond to CANCEL requests received via multicast to avoid request implosion. A proxy or UAC SHOULD send a CANCEL on

receiving the first 2xx or 6xx response to a multicast request.

Server response suppression is a MAY since it requires a server to violate some basic message processing rules. Lets say A sends a multicast request, and it is received by B,C, and D. B sends a 200 response. The topmost Via field in the response will contain the address of A. C will also receive this response, and could use it to suppress its own response. However, C would normally not examine this response, as the topmost Via is not its own. Normally, a response received with an incorrect topmost Via MUST be dropped, but not in this case. To distinguish this packet from a misrouted or multicast looped packet is fairly complex, and for this reason the procedure is a MAY. The CANCEL, instead, provides a simpler and more standard way to perform response suppression. It is for this reason that the use of CANCEL here is a SHOULD

10.3 TCP

A single TCP connection can serve one or more SIP transactions. A transaction contains zero or more provisional responses followed by one or more final responses. (Typically, transactions contain exactly one final response, but there are exceptional circumstances, where, for example, multiple 200 responses can be generated.)

The client SHOULD keep the connection open at least until the first final response arrives. If the client closes or resets the TCP connection prior to receiving the first final response, the server treats this action as equivalent to a CANCEL request.

This behavior makes it less likely that malfunctioning clients cause a proxy server to keep connection state indefinitely.

The server SHOULD NOT close the TCP connection until it has sent its final response, at which point it MAY close the TCP connection if it wishes to. However, normally it is the client's responsibility to close the connection.

If the server leaves the connection open, and if the client so desires it MAY re-use the connection for further SIP requests or for requests from the same family of protocols (such as HTTP or stream control commands).

If a server needs to return a response to a client and no longer has a connection open to that client, it MAY open a connection to the address listed in the Via header. Thus, a proxy or user agent MUST be prepared to receive both requests and responses on a "passive" connection.

10.4 Reliability for BYE, CANCEL, OPTIONS, REGISTER Requests

10.4.1 UDP

A SIP client using UDP SHOULD retransmit a BYE, CANCEL, OPTIONS, or REGISTER request with an exponential backoff, starting at a T1 second interval, doubling the interval for each packet, and capping off at a T2 second interval. This means that after the first packet is sent, the second is sent T1 seconds later, the next 2*T1 seconds after that, the next 4*T1 seconds after that, and so on, until the interval hits T2. Subsequent retransmissions are spaced by T2 seconds. If the client receives a provisional response, it continues to retransmit the request, but with an interval of T2 seconds. Retransmissions cease when the client has sent a total of eleven packets, or receives a definitive response. Default values for T1 and T2 are 500 ms and 4 s, respectively. Clients MAY use larger values, but SHOULD NOT use smaller ones. Servers retransmit the response upon receipt of a request retransmission. After the server sends a final response, it cannot be sure the client has received the response, and thus SHOULD cache the results for at least 10*T2 seconds to avoid having to, for example, contact the user or location server again upon receiving a request retransmission.

> Use of the exponential backoff is for congestion control purposes. However, the back-off must cap off, since request retransmissions are used to trigger response retransmissions at the server. Without a cap, the loss of a single response could significantly increase transaction latencies.

The value of the initial retransmission timer is smaller than that that for TCP since it is expected that network paths suitable for interactive communications have round-trip times smaller than 500 ms. For congestion control purposes, the retransmission count has to be bounded. Given that most transactions are expected to consist of one request and a few responses, round-trip time estimation is not likely to be very useful. If RTT estimation is desired to more quickly discover a missing final response, each request retransmission needs to be labeled with its own Timestamp (Section 6.36), returned in the response. The server caches the result until it can be sure that the client will not retransmit the same request again.

Each server in a proxy chain generates its own final response to a CANCEL request. The server responds immediately upon receipt of the CANCEL request rather than waiting until it has received final responses from the CANCEL requests it generates.

BYE and OPTIONS final responses are generated by redirect and user agent servers; REGISTER final responses are generated by registrars. Note that in contrast to the reliability mechanism described in Section 10.5, responses to these requests are not retransmitted periodically and not acknowledged via ACK.

10.4.2 TCP

Clients using TCP do not need to retransmit requests.

10.5 Reliability for INVITE Requests

Special considerations apply for the INVITE method.

1. After receiving an invitation, considerable time can elapse before the server can determine the outcome. For example, if the called party is "rung" or extensive searches are performed, delays between the request and a definitive response can reach several tens of seconds. If either caller or callee are automated servers not directly controlled by a human being, a call attempt could be unbounded in time.

2. If a telephony user interface is modeled or if we need to interface to the PSTN, the caller's user interface will provide "ringback", a signal that the callee is being alerted. (The status response 180 (Ringing) MAY be used to initiate ringback.) Once the callee picks up, the caller needs to know so that it can enable the voice path and stop ringback. The callee's response to the invitation could get lost. Unless the response is transmitted reliably, the caller will continue to hear ringback while the callee assumes that the call exists.

3. The client has to be able to terminate an on-going request, e.g., because it is no longer willing to wait for the connection or search to succeed. The server will have to wait several retransmission intervals to interpret the lack of request retransmissions as the end of a call. If the call succeeds shortly after the caller has given up, the callee will "pick up the phone" and not be "connected".

10.5.1 UDP

For UDP, A SIP client SHOULD retransmit a SIP INVITE request with an interval that starts at T1 seconds, and doubles after each packet transmission. The client ceases retransmissions if it receives a provisional or definitive response, or once it has sent a total of 7 request packets.

A server which transmits a provisional response should retransmit it upon reception of a duplicate request. A server which transmits a final response should retransmit it with an interval that starts at T1 seconds, and doubles for each subsequent packet. Response retransmissions cease when any one of the following occurs:

1. An ACK request for the same transaction is received;

2. a BYE request for the same call leg is received;

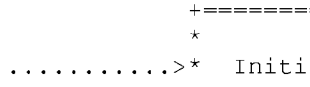
3. a CANCEL request for the same call leg is received and the final response status was equal or greater to 300;

4. the response has been transmitted 7 times.

Only the user agent client generates an ACK for 2xx final responses, If the response contained a Contact header field, the ACK MAY be sent to the address listed in that Contact header field. If the response did not contain a Contact header, the client uses the same To header field and Request-URI as for the INVITE request and sends the ACK to the same destination as the original INVITE request. ACKs for final responses other than 2xx are sent to the same server that the original request was sent to, using the same Request-URI as the original request. Note, however, that the To header field in the ACK is copied from the response being acknowledged, not the request, and thus MAY additionally contain the tag parameter. Also note than unlike 2xx final responses, a proxy generates an ACK for non-2xx final responses.

The ACK request MUST NOT be acknowledged to prevent a response-ACK feedback loop. Fig. 12 and 13 show the client and server state diagram for invitations.

The mechanism in Sec. 10.4 would not work well for INVITE because of the long delays between INVITE and a final response. If the 200 response were to get lost, the callee would believe the call to exist, but the voice path would



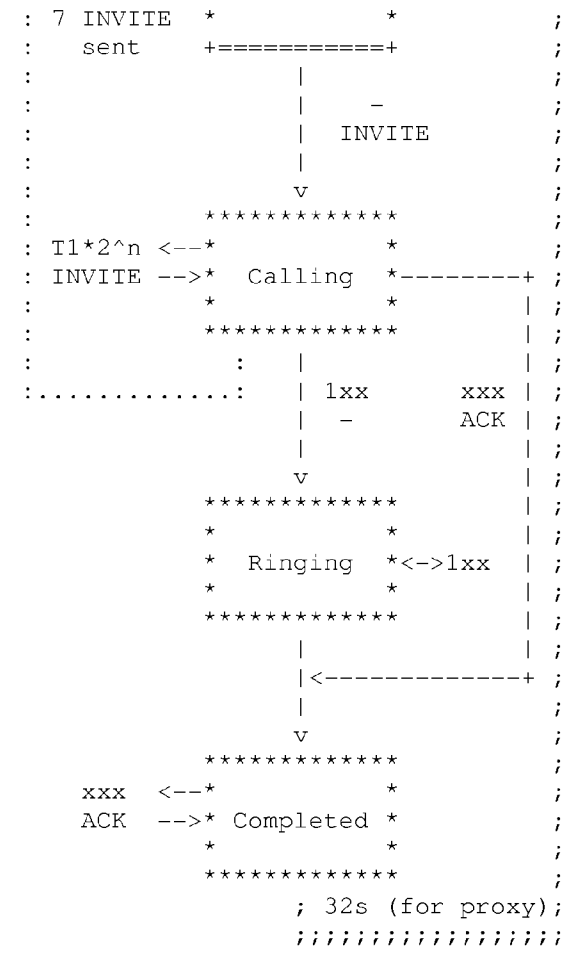
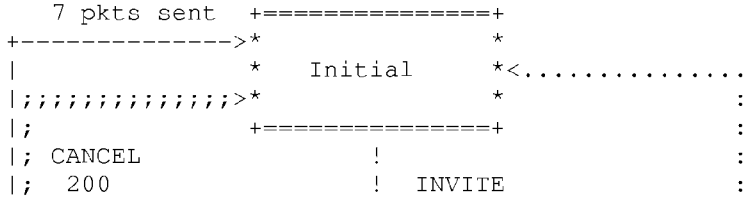


Figure 12: State transition diagram of client for INVITE method



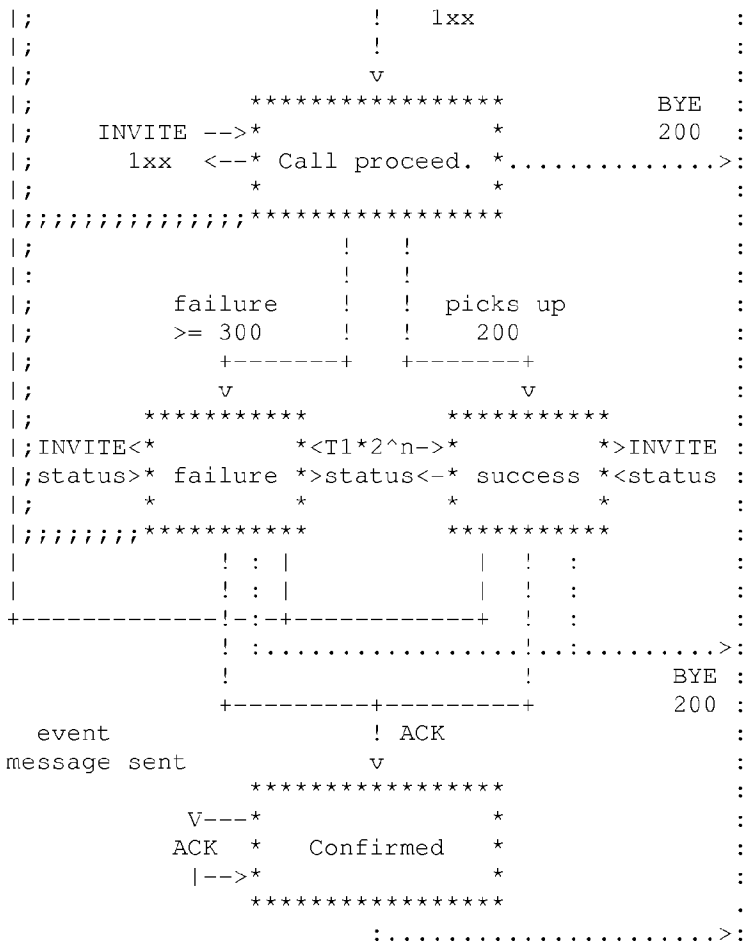


Figure 13: State transition diagram of server for INVITE method

be dead since the caller does not know that the callee has picked up. Thus, the INVITE retransmission interval would have to be on the order of a second or two to limit the duration of this state confusion. Retransmitting the response with an exponential back-off helps ensure that the response is received, without placing an undue burden on the network.

10.5.2 TCP

A user agent using TCP MUST NOT retransmit requests, but uses the same algorithm as for UDP (Section 10.5.1) to retransmit responses until it receives an ACK.

It is necessary to retransmit 2xx responses as their reliability is assured end-to-end only. If the chain of proxies has a UDP link in the middle, it could lose the

response, with no possibility of recovery. For simplicity, we also retransmit non-2xx responses, although that is not strictly necessary.

10.6 Reliability for ACK Requests

The ACK request does not generate responses. It is only generated when a response to an INVITE request arrives (see Section 10.5). This behavior is independent of the transport protocol. Note that the ACK request MAY take a different path than the original INVITE request, and MAY even cause a new TCP connection to be opened in order to send it.

10.7 ICMP Handling

Handling of ICMP messages in the case of UDP messages is straightforward. For requests, a host, network, port, or protocol unreachable error SHOULD be treated as if a 400-class response was received. For responses, these errors SHOULD cause the server to cease retransmitting the response.

Source quench ICMP messages SHOULD be ignored. TTL exceeded errors SHOULD be ignored. Parameter problem errors SHOULD be treated as if a 400-class response was received.

11 Behavior of SIP User Agents

This section describes the rules for user agent client and servers for generating and processing requests and responses.

11.1 Caller Issues Initial INVITE Request

When a user agent client desires to initiate a call, it formulates an INVITE request. The To field in the request contains the address of the callee. The Request-URI contains the same address. The From field contains the address of the caller. If the From address can appear in requests generated by other user agent clients for the same call, the caller MUST insert the tag parameter in the From field. A UAC MAY optionally add a Contact header containing an address where it would like to be contacted for transactions from the callee back to the caller.

11.2 Callee Issues Response

When the initial INVITE request is received at the callee, the callee can accept, redirect, or reject the call. In all of these cases, it formulates a response. The response MUST copy the To, From, Call-ID, CSeq and Via fields from the request. Additionally, the responding UAS MUST add the tag parameter to the To field in the response if the request contained more than one Via header field. Since a request from a UAC may fork and arrive at multiple hosts, the tag parameter serves to distinguish, at the UAC, multiple responses from different

UAS's. The UAS MAY add a Contact header field in the response. It contains an address where the callee would like to be contacted for subsequent transactions, including the ACK for the current INVITE. The UAS stores the values of the To and From field, including any tags. These become the local and remote addresses of the call leg, respectively.

11.3 Caller Receives Response to Initial Request

Multiple responses may arrive at the UAC for a single INVITE request, due to a forking proxy. Each response is distinguished by the "tag" parameter in the To header field, and each represents a distinct call leg. The caller MAY choose to acknowledge or terminate the call with each responding UAS. To acknowledge, it sends an ACK request, and to terminate it sends a BYE request. The To header field in the ACK or BYE MUST be the same as the To field in the 200 response, including any tag. The From header field MUST be the same as the From header field in the 200 (OK) response, including any tag. The Request-URI of the ACK or BYE request MAY be set to whatever address was found in the Contact header field in the 200 (OK) response, if present. Alternately, a UAC may copy the address from the To header field into the Request-URI. The UAC also notes the value of the To and From header fields in each response. For each call leg, the To header field becomes the remote address, and the From header field becomes the local address.

11.4 Caller or Callee Generate Subsequent Requests

Once the call has been established, either the caller or callee MAY generate INVITE or BYE requests to change or terminate the call. Regardless of whether the caller or callee is generating the new request, the header fields in the request are set as follows. For the desired call leg, the To header field is set to the remote address, and the From header field is set to the local address (both including any tags). The Contact header field MAY be different than the Contact header field sent in a previous response or request. The Request-URI MAY be set to the value of the Contact header field received in a previous request or response from the remote party, or to the value of the remote address.

11.5 Receiving Subsequent Requests

When a request is received subsequently, the following checks are made:

1. If the Call-ID is new, the request is for a new call, regardless of the values of the To and From header fields.

2. If the Call-ID exists, the request is for an existing call. If the To, From, Call-ID, and CSeq values exactly match (including tags) those of any requests received previously, the request is a retransmission.

3. If there was no match to the previous step, the To and From fields are compared against existing call leg local and remote addresses. If there is a match, and the CSeq in the request is higher than the last CSeq received on that leg, the request is a new transaction for an existing call leg.

12 Behavior of SIP Proxy and Redirect Servers

This section describes behavior of SIP redirect and proxy servers in detail. Proxy servers can "fork" connections, i.e., a single incoming request spawns several outgoing (client) requests.

12.1 Redirect Server

A redirect server does not issue any SIP requests of its own. After receiving a request other than CANCEL, the server gathers the list of alternative locations and returns a final response of class 3xx or it refuses the request. For well-formed CANCEL requests, it SHOULD return a 2xx response. This response ends the SIP transaction. The

redirect server maintains transaction state for the whole SIP transaction. It is up to the client to detect forwarding loops between redirect servers.

12.2 User Agent Server

User agent servers behave similarly to redirect servers, except that they also accept requests and can return a response of class 2xx.

12.3 Proxy Server

This section outlines processing rules for proxy servers. A proxy server can either be stateful or stateless. When stateful, a proxy remembers the incoming request which generated outgoing requests, and the outgoing requests. A stateless proxy forgets all information once an outgoing request is generated. A forking proxy SHOULD be stateful. Proxies that accept TCP connections MUST be stateful.

Otherwise, if the proxy were to lose a request, the TCP client would never retransmit it.

A stateful proxy SHOULD NOT become stateless until after it sends a definitive response upstream, and at least 32 seconds after it received a definitive response.

A stateful proxy acts as a virtual UAS/UAC. It implements the server state machine when receiving requests, and the client state machine for generating outgoing requests, with the exception of receiving a 2xx response to an INVITE. Instead of generating an ACK, the 2xx

response is always forwarded upstream towards the caller. Furthermore, ACK's for 200 responses to INVITE's are always proxied downstream towards the UAS, as they would be for a stateless proxy.

A stateless proxy does not act as a virtual UAS/UAC (as this would require state). Rather, a stateless proxy forwards every request it receives downstream, and every response it receives upstream.

12.3.1 Proxying Requests

To prevent loops, a server MUST check if its own address is already contained in the Via header field of the incoming request.

The To, From, Call-ID, and Contact tags are copied exactly from the original request. The proxy SHOULD change the Request-URI to indicate the server where it intends to send the request.

A proxy server always inserts a Via header field containing its own address into those requests that are caused by an incoming request. Each proxy MUST insert a "branch" parameter (Section 6.40).

12.3.2 Proxying Responses

A proxy only processes a response if the topmost Via field matches one of its addresses. A response with a non-matching top Via field MUST be dropped.

12.3.3 Stateless Proxy: Proxying Responses

A stateless proxy removes its own Via field, and checks the address in the next Via field. In the case of UDP, the response is sent to the address listed in the "maddr" tag if present, otherwise to the "received" tag if present, and finally to the address in the "sent-by" field. A proxy MUST remain stateful when handling requests received via TCP.

A stateless proxy MUST NOT generate its own provisional responses.

12.3.4 Stateful Proxy: Receiving Requests

When a stateful proxy receives a request, it checks the To, From (including tags), Call-ID and CSeq against existing request records. If the tuple exists, the request is a retransmission. The provisional or final response sent previously is retransmitted, as per the server state machine. If the tuple does not exist, the request corresponds to a new transaction, and the request should be proxied.

A stateful proxy server MAY generate its own provisional (1xx) responses.

12.3.5 Stateful Proxy: Receiving ACKs

When an ACK request is received, it is either processed locally or proxied. To make this determination, the To, From, CSeq and Call-ID fields are compared against those in previous requests. If there is no match, the ACK request is proxied as if it were an INVITE request. If there is a match, and if the server had ever sent a 200 response upstream, the ACK is proxied. If the server had never sent any responses upstream, the ACK is also proxied. If the server had sent a 3xx, 4xx, 5xx or 6xx response, but no 2xx response, the ACK is processed locally if the tag in the To field of the ACK matches the tag sent by the proxy in the response.

12.3.6 Stateful Proxy: Receiving Responses

When a proxy server receives a response that has passed the Via checks, the proxy server checks the To (without the tag), From (including the tag), Call-ID and CSeq against values seen in previous requests. If there is no match, the response is forwarded upstream to the address listed in the Via field. If there is a match, the "branch" tag in the Via field is examined. If it matches a known branch identifier, the response is for the given branch, and processed by the virtual client for the given branch. Otherwise, the response is dropped.

A stateful proxy should obey the rules in Section 12.4 to determine if the response should be proxied upstream. If it is to be proxied, the same rules for stateless proxies above are followed, with the following addition for TCP. If a request was received via TCP (indicated by the protocol in the top Via header), the proxy checks to see if it has a connection currently open to that address. If so, the response is sent on that connection. Otherwise, a new TCP connection is opened to the address and port in the Via field, and the response is sent there. Note that this implies that a UAC or proxy MUST be prepared to receive responses on the incoming side of a TCP connection. Definitive non 200-class responses MUST be retransmitted by the proxy, even over a TCP connection.

12.3.7 Stateless, Non-Forking Proxy

Proxies in this category issue at most a single unicast request for each incoming SIP request, that is, they do not "fork" requests. However, servers MAY choose to always operate in a mode that allows issuing of several requests, as described in Section 12.4.

The server can forward the request and any responses. It does not have to maintain any state for the SIP transaction. Reliability is assured by the next redirect or stateful proxy server in the server chain.

A proxy server SHOULD cache the result of any address translations

and the response to speed forwarding of retransmissions. After the cache entry has been expired, the server cannot tell whether an incoming request is actually a retransmission of an older request. The server will treat it as a new request and commence another search.

12.4 Forking Proxy

The server MUST respond to the request immediately with a 100 (Trying) response.

Successful responses to an INVITE request MAY contain a Contact header field so that the following ACK or BYE bypasses the proxy search mechanism. If the proxy requires future requests to be routed through it, it adds a Record-Route header to the request (Section 6.29).

The following C-code describes the behavior of a proxy server issuing several requests in response to an incoming INVITE request. The function request(r, a, b) sends a SIP request of type r to address a, with branch id b. await_response() waits until a response is received and returns the response. close(a) closes the TCP connection to client with address a. response(r) sends a response to the client. ismulticast() returns 1 if the location is a multicast address and zero otherwise. The variable timeleft indicates the amount of time left until the maximum response time has expired. The variable recurse indicates whether the server will recursively try addresses returned through a 3xx response. A server MAY decide to recursively try only certain addresses, e.g., those which are within the same domain as the proxy server. Thus, an initial multicast request can trigger additional unicast requests.

```
/* request type */
typedef enum {INVITE, ACK, BYE, OPTIONS, CANCEL, REGISTER} Method;

process_request(Method R, int N, address_t address[])
{
  struct {
    int branch;          /* branch id */
    int done;            /* has responded */
  } outgoing[];
  int done[];            /* address has responded */
  char *location[];      /* list of locations */
  int heard = 0;         /* number of sites heard from */
  int class;             /* class of status code */
  int timeleft = 120;    /* sample timeout value */
  int loc = 0;           /* number of locations */
  struct {               /* response */
    int status;          /* response: CANCEL=-1 */
    int locations;       /* number of redirect locations */
    char *location[];    /* redirect locations */
    address_t a;         /* address of respondent */
```

```
  int branch;          /* branch identifier */
} r, best;             /* response, best response */
int i;

best.status = 1000;
for (i = 0; i < N; i++) {
```



```
  request(R, address[i], i);
  outgoing[i].done = 0;
  outgoing[i].branch = i;
}

while (timeleft > 0 && heard < N) {
  r = await_response();
  class = r.status / 100;

  /* If final response, mark branch as done. */
  if (class >= 2) {
    heard++;
    for (i = 0; i < N; i++) {
      if (r.branch == outgoing[i].branch) {
        outgoing[i].done = 1;
        break;
      }
    }
  }
  /* CANCEL: respond, fork and wait for responses */
  else if (class < 0) {
    best.status = 200;
    response(best);
    for (i = 0; i < N; i++) {
      if (!outgoing[i].done)
        request(CANCEL, address[i], outgoing[i].branch);
    }
    best.status = -1;
  }

  /* Send an ACK */

  if (class != 2) {
    if (R == INVITE) request(ACK, r.a, r.branch);
  }


  if (class == 2) {
    if (r.status < best.status) best = r;
    break;
  }
  else if (class == 3) {
    /* A server MAY optionally recurse.  The server MUST check
     * whether it has tried this location before and whether
     * the location is part of the Via path of the incoming
     * request.  This check is omitted here for brevity.
```

```
           * Multicast locations MUST NOT be returned to the client if
           * the server is not recursing.
```



```
           */
          if (recurse) {
            multicast = 0;
            N += r.locations;
            for (i = 0; i < r.locations; i++) {
              request(R, r.location[i]);
            }
          } else if (!ismulticast(r.location)) {
            best = r;
          }
        }
        else if (class == 4) {
          if (best.status >= 400) best = r;
        }
        else if (class == 5) {
          if (best.status >= 500) best = r;
        }
        else if (class == 6) {
          best = r;
          break;
        }
      }

      /* We haven't heard anything useful from anybody. */
      if (best.status == 1000) {
        best.status = 404;
      }
      if (best.status/100 != 3) loc = 0;
      response(best);
    }
```

Responses are processed as follows. The process completes (and state can be freed) when all requests have been answered by final status responses (for unicast) or 60 seconds have elapsed (for multicast). A proxy MAY send a CANCEL to all branches and return a 408 (Timeout) to the client after 60 seconds or more.

1xx: The proxy MAY forward the response upstream towards the client.

2xx: The proxy MUST forward the response upstream towards the client, without sending an ACK downstream. After receiving a 2xx, the server MAY terminate all other pending requests by sending a CANCEL request and closing the TCP connection, if applicable. (Terminating pending requests is advisable as searches consume resources. Also, INVITE requests could "ring" on a number of workstations if the callee is currently logged in more than once.)

3xx: The proxy MUST send an ACK and MAY recurse on the listed Contact addresses. Otherwise, the lowest-numbered response is returned if there were no 2xx responses.

Location lists are not merged as that would prevent forwarding of authenticated responses. Also, responses can have message bodies, so that merging is not feasible.

4xx, 5xx: The proxy MUST send an ACK and remember the response if it has a lower status code than any previous 4xx and 5xx responses. On completion, the lowest-numbered response is returned if there were no 2xx or 3xx responses.

6xx: The proxy MUST forward the response to the client and send an ACK. Other pending requests MAY be terminated with CANCEL as described for 2xx responses.

A proxy server forwards any response for Call-IDs for which it does not have a pending transaction according to the response's Via header. User agent servers respond to BYE requests for unknown call legs with status code 481 (Transaction Does Not Exist); they drop ACK requests with unknown call legs silently.

Special considerations apply for choosing forwarding destinations for ACK and BYE requests. In most cases, these requests will bypass proxies and reach the desired party directly, keeping proxies from having to make forwarding decisions.

A proxy MAY maintain call state for a period of its choosing. If a proxy still has list of destinations that it forwarded the last INVITE to, it SHOULD direct ACK requests only to those downstream servers.

13 Security Considerations

13.1 Confidentiality and Privacy: Encryption

13.1.1 End-to-End Encryption

SIP requests and responses can contain sensitive information about the communication patterns and communication content of individuals. The SIP message body MAY also contain encryption keys for the session itself. SIP supports three complementary forms of encryption to protect privacy:

- End-to-end encryption of the SIP message body and certain sensitive header fields;

- hop-by-hop encryption to prevent eavesdropping that tracks who is calling whom;

- hop-by-hop encryption of Via fields to hide the route a request has taken.

Not all of the SIP request or response can be encrypted end-to-end because header fields such as To and Via need to be visible to proxies so that the SIP request can be routed correctly. Hop-by-hop encryption encrypts the entire SIP request or response on the wire so that packet sniffers or other eavesdroppers cannot see who is calling whom. Hop-by-hop encryption can also encrypt requests and responses that have been end-to-end encrypted. Note that proxies can still see who is calling whom, and this information is also deducible by performing a network traffic analysis, so this provides a very limited but still worthwhile degree of protection.

SIP Via fields are used to route a response back along the path taken by the request and to prevent infinite request loops. However, the information given by them can also provide useful information to an attacker. Section 6.22 describes how a sender can request that Via fields be encrypted by cooperating proxies without compromising the purpose of the Via field.

End-to-end encryption relies on keys shared by the two user agents involved in the request. Typically, the message is sent encrypted with the public key of the recipient, so that only that recipient can read the message. All implementations SHOULD support PGP-based encryption [33] and MAY implement other schemes.

A SIP request (or response) is end-to-end encrypted by splitting the message to be sent into a part to be encrypted and a short header that will remain in the clear. Some parts of the SIP message, namely the request line, the response line and certain header fields marked with "n" in the "enc." column in Table 4 and 5 need to be read and returned by proxies and thus MUST NOT be encrypted end-to-end. Possibly sensitive information that needs to be made available as plaintext include destination address (To) and the forwarding path (Via) of the call. The Authorization header field MUST remain in the clear if it contains a digital signature as the signature is generated after encryption, but MAY be encrypted if it contains "basic" or "digest" authentication. The From header field SHOULD normally remain in the clear, but MAY be encrypted if required, in which case some proxies MAY return a 401 (Unauthorized) status if they require a From field.

Other header fields MAY be encrypted or MAY travel in the clear as desired by the sender. The Subject, Allow and Content-Type header

fields will typically be encrypted. The Accept, Accept-Language, Date, Expires, Priority, Require, Call-ID, Cseq, and Timestamp header fields will remain in the clear.

All fields that will remain in the clear MUST precede those that will be encrypted. The message is encrypted starting with the first character of the first header field that will be encrypted and continuing through to the end of the message body. If no header fields are to be encrypted, encrypting starts with the second CRLF pair after the last header field, as shown below. Carriage return and line feed characters have been made visible as "$", and the encrypted part of the message is outlined.

```
  INVITE sip:watson@boston.bell-telephone.com SIP/2.0$
  Via: SIP/2.0/UDP 169.130.12.5$
  To: T. A. Watson <sip:watson@bell-telephone.com>$
  From: A. Bell <sip:a.g.bell@bell-telephone.com>$
  Encryption: PGP version=5.0$
  Content-Length: 224$
  Call-ID: 187602141351@worcester.bell-telephone.com$
  CSeq: 488$
  $
**********************************************************
* Subject: Mr. Watson, come here.$                       *
* Content-Type: application/sdp$                         *
* $                                                      *
* v=0$                                                   *
* o=bell 53655765 2353687637 IN IP4 128.3.4.5$           *
* c=IN IP4 135.180.144.94$                               *
* m=audio 3456 RTP/AVP 0 3 4 5$                          *
**********************************************************
```

An Encryption header field MUST be added to indicate the encryption mechanism used. A Content-Length field is added that indicates the length of the encrypted body. The encrypted body is preceded by a blank line as a normal SIP message body would be.

Upon receipt by the called user agent possessing the correct decryption key, the message body as indicated by the Content-Length field is decrypted, and the now-decrypted body is appended to the clear-text header fields. There is no need for an additional Content-Length header field within the encrypted body because the length of the actual message body is unambiguous after decryption.

Had no SIP header fields required encryption, the message would have been as below. Note that the encrypted body MUST then include a blank line (start with CRLF) to disambiguate between any possible SIP header fields that might have been present and the SIP message body.

```
  INVITE sip:watson@boston.bell-telephone.com SIP/2.0$
  Via: SIP/2.0/UDP 169.130.12.5$
  To: T. A. Watson <sip:watson@bell-telephone.com>$
  From: A. Bell <a.g.bell@bell-telephone.com>$
  Encryption: PGP version=5.0$
  Content-Type: application/sdp$
  Content-Length: 107$
  $
**********************************************
* $                                          *
* v=0$                                       *
* o=bell 53655765 2353687637 IN IP4 128.3.4.5$  *
* c=IN IP4 135.180.144.94$                   *
* m=audio 3456 RTP/AVP 0 3 4 5$              *
**********************************************
```

13.1.2 Privacy of SIP Responses

SIP requests can be sent securely using end-to-end encryption and authentication to a called user agent that sends an insecure response. This is allowed by the SIP security model, but is not a good idea. However, unless the correct behavior is explicit, it would not always be possible for the called user agent to infer what a reasonable behavior was. Thus when end-to-end encryption is used by the request originator, the encryption key to be used for the response SHOULD be specified in the request. If this were not done, it might be possible for the called user agent to incorrectly infer an appropriate key to use in the response. Thus, to prevent key-guessing becoming an acceptable strategy, we specify that a called user agent receiving a request that does not specify a key to be used for the response SHOULD send that response unencrypted.

Any SIP header fields that were encrypted in a request SHOULD also be encrypted in an encrypted response. Contact response fields MAY be encrypted if the information they contain is sensitive, or MAY be left in the clear to permit proxies more scope for localized searches.

13.1.3 Encryption by Proxies

Normally, proxies are not allowed to alter end-to-end header fields and message bodies. Proxies MAY, however, encrypt an unsigned request or response with the key of the call recipient.

> Proxies need to encrypt a SIP request if the end system cannot perform encryption or to enforce organizational security policies.

13.1.4 Hop-by-Hop Encryption

SIP requests and responses MAY also be protected by security mechanisms at the transport or network layer. No particular mechanism is defined or recommended here. Two possibilities are IPSEC [34] or TLS [35]. The use of a particular mechanism will generally need to be specified out of band, through manual configuration, for example.

13.1.5 Via field encryption

When Via header fields are to be hidden, a proxy that receives a request containing an appropriate "Hide: hop" header field (as specified in section 6.22) SHOULD encrypt the header field. As only the proxy that encrypts the field will decrypt it, the algorithm chosen is entirely up to the proxy implementor. Two methods satisfy these requirements:

- The server keeps a cache of Via header fields and the associated To header field, and replaces the Via header field with an index into the cache. On the reverse path, take the Via header field from the cache rather than the message.

  This is insufficient to prevent message looping, and so an additional ID MUST be added so that the proxy can detect loops. This SHOULD NOT normally be the address of the proxy as the goal is to hide the route, so instead a sufficiently large random number SHOULD be used by the proxy and maintained in the cache.

  It is possible for replies to get directed to the wrong originator if the cache entry gets reused, so great care needs to be taken to ensure this does not happen.

- The server MAY use a secret key to encrypt the Via field, a timestamp and an appropriate checksum in any such message with the same secret key. The checksum is needed to detect whether successful decoding has occurred, and the timestamp is

required to prevent possible replay attacks and to ensure that no two requests from the same previous hop have the same encrypted Via field. This is the preferred solution.

13.2 Message Integrity and Access Control: Authentication

Protective measures need to be taken to prevent an active attacker from modifying and replaying SIP requests and responses. The same cryptographic measures that are used to ensure the authenticity of the SIP message also serve to authenticate the originator of the message. However, the "basic" and "digest" authentication mechanism offer authentication only, without message integrity.

Transport-layer or network-layer authentication MAY be used for hop-

by-hop authentication. SIP also extends the HTTP WWW-Authenticate (Section 6.42) and Authorization (Section 6.11) header field and their Proxy counterparts to include cryptographically strong signatures. SIP also supports the HTTP "basic" and "digest" schemes (see Section 14) and other HTTP authentication schemes to be defined that offer a rudimentary mechanism of ascertaining the identity of the caller.

> Since SIP requests are often sent to parties with which no prior communication relationship has existed, we do not specify authentication based on shared secrets.

SIP requests MAY be authenticated using the Authorization header field to include a digital signature of certain header fields, the request method and version number and the payload, none of which are modified between client and called user agent. The Authorization header field is used in requests to authenticate the request originator end-to-end to proxies and the called user agent, and in responses to authenticate the called user agent or proxies returning their own failure codes. If required, hop-by-hop authentication can be provided, for example, by the IPSEC Authentication Header.

SIP does not dictate which digital signature scheme is used for authentication, but does define how to provide authentication using PGP in Section 15. As indicated above, SIP implementations MAY also use "basic" and "digest" authentication and other authentication mechanisms defined for HTTP. Note that "basic" authentication has severe security limitations. The following does not apply to these schemes.

To cryptographically sign a SIP request, the order of the SIP header fields is important. When an Authorization header field is present, it indicates that all header fields following the Authorization

header field have been included in the signature. Therefore, hop-by-hop header fields which MUST or SHOULD be modified by proxies MUST precede the Authorization header field as they will generally be modified or added-to by proxy servers. Hop-by-hop header fields which MAY be modified by a proxy MAY appear before or after the Authorization header. When they appear before, they MAY be modified by a proxy. When they appear after, they MUST NOT be modified by a proxy. To sign a request, a client constructs a message from the request method (in upper case) followed, without LWS, by the SIP version number, followed, again without LWS, by the request headers to be signed and the message body. The message thus constructed is then signed.

For example, if the SIP request is to be:

```
INVITE sip:watson@boston.bell-telephone.com SIP/2.0
Via: SIP/2.0/UDP 169.130.12.5
Authorization: PGP version=5.0, signature=...
```

```
From: A. Bell <sip:a.g.bell@bell-telephone.com>
To: T. A. Watson <sip:watson@bell-telephone.com>
Call-ID: 187602141351@worcester.bell-telephone.com
Subject: Mr. Watson, come here.
Content-Type: application/sdp
Content-Length: ...

v=0
o=bell 53655765 2353687637 IN IP4 128.3.4.5
c=IN IP4 135.180.144.94
m=audio 3456 RTP/AVP 0 3 4 5
```

Then the data block that is signed is:

```
INVITESIP/2.0From: A. Bell <sip:a.g.bell@bell-telephone.com>
To: T. A. Watson <sip:watson@bell-telephone.com>
Call-ID: 187602141351@worcester.bell-telephone.com
Subject: Mr. Watson, come here.
Content-Type: application/sdp
Content-Length: ...

v=0
o=bell 53655765 2353687637 IN IP4 128.3.4.5
c=IN IP4 135.180.144.94
m=audio 3456 RTP/AVP 0 3 4 5
```


Clients wishing to authenticate requests MUST construct the portion of the message below the Authorization header using a canonical form. This allows a proxy to parse the message, take it apart, and reconstruct it, without causing an authentication failure due to extra white space, for example. Canonical form consists of the following rules:

- No short form header fields
- Header field names are capitalized as shown in this document
- No white space between the header name and the colon
- A single space after the colon
- Line termination with a CRLF
- No line folding
- No comma separated lists of header values; each must appear as a separate header

- Only a single SP between tokens, between tokens and quoted strings, and between quoted strings; no SP after last token or quoted string

- No LWS between tokens and separators, except as described above for after the colon in header fields

Note that if a message is encrypted and authenticated using a digital signature, when the message is generated encryption is performed before the digital signature is generated. On receipt, the digital signature is checked before decryption.

A client MAY require that a server sign its response by including a Require: org.ietf.sip.signed-response request header field. The client indicates the desired authentication method via the WWW-Authenticate header.

The correct behavior in handling unauthenticated responses to a request that requires authenticated responses is described in section 13.2.1.

13.2.1 Trusting responses

There is the possibility that an eavesdropper listens to requests and then injects unauthenticated responses that terminate, redirect or otherwise interfere with a call. (Even encrypted requests contain enough information to fake a response.)

Clients need to be particularly careful with 3xx redirection responses. Thus a client receiving, for example, a 301 (Moved Permanently) which was not authenticated when the public key of the called user agent is known to the client, and authentication was requested in the request SHOULD be treated as suspicious. The correct behavior in such a case would be for the called-user to form a dated response containing the Contact field to be used, to sign it, and give this signed stub response to the proxy that will provide the redirection. Thus the response can be authenticated correctly. A client SHOULD NOT automatically redirect such a request to the new location without alerting the user to the authentication failure before doing so.

Another problem might be responses such as 6xx failure responses which would simply terminate a search, or "4xx" and "5xx" response failures.

If TCP is being used, a proxy SHOULD treat 4xx and 5xx responses as valid, as they will not terminate a search. However, fake 6xx

responses from a rogue proxy terminate a search incorrectly. 6xx responses SHOULD be authenticated if requested by the client, and failure to do so SHOULD cause such a client to ignore the 6xx response and continue a search.

With UDP, the same problem with 6xx responses exists, but also an active eavesdropper can generate 4xx and 5xx responses that might cause a proxy or client to believe a failure occurred when in fact it did not. Typically 4xx and 5xx responses will not be signed by the called user agent, and so there is no simple way to detect these rogue responses. This problem is best prevented by using hop-by-hop encryption of the SIP request, which removes any additional problems that UDP might have over TCP.

These attacks are prevented by having the client require response authentication and dropping unauthenticated responses. A server user agent that cannot perform response authentication responds using the normal Require response of 420 (Bad Extension).

13.3 Callee Privacy

User location and SIP-initiated calls can violate a callee's privacy. An implementation SHOULD be able to restrict, on a per-user basis, what kind of location and availability information is given out to certain classes of callers.

13.4 Known Security Problems

With either TCP or UDP, a denial of service attack exists by a rogue proxy sending 6xx responses. Although a client SHOULD choose to ignore such responses if it requested authentication, a proxy cannot do so. It is obliged to forward the 6xx response back to the client. The client can then ignore the response, but if it repeats the request it will probably reach the same rogue proxy again, and the process will repeat.

14 SIP Authentication using HTTP Basic and Digest Schemes

SIP implementations MAY use HTTP's basic and digest authentication mechanisms to provide a rudimentary form of security. This section overviews usage of these mechanisms in SIP. The basic operation is almost completely identical to that for HTTP [36]. This section outlines this operation, pointing to [36] for details, and noting the differences when used in SIP.

14.1 Framework

The framework for SIP authentication parallels that for HTTP [36]. In particular, the BNF for auth-scheme, auth-param, challenge, realm,

realm-value, and credentials is identical. The 401 response is used by user agent servers in SIP to challenge the authorization of a user agent client. Additionally, registrars and redirect servers MAY make use of 401 responses for authorization, but proxies MUST NOT, and instead MAY use the 407 response. The requirements for inclusion of the Proxy-Authenticate, Proxy-Authorization, WWW-Authenticate, and Authorization in the various messages is identical to [36].

Since SIP does not have the concept of a canonical root URL, the notion of protections spaces are interpreted differently for SIP. The realm is a protection domain for all SIP URIs with the same value for the userinfo, host and port part of the SIP Request-URI. For example:

```
INVITE sip:alice.wonderland@example.com SIP/2.0
WWW-Authenticate:  Basic realm="business"
```


and

```
INVITE sip:aw@example.com SIP/2.0
WWW-Authenticate: Basic realm="business"
```

define different protection realms according to this rule.

When a UAC resubmits a request with its credentials after receiving a 401 or 407 response, it MUST increment the CSeq header field as it would normally do when sending an updated request.

14.2 Basic Authentication

The rules for basic authentication follow those defined in [36], but with the words "origin server" replaced with "user agent server, redirect server , or registrar".

Since SIP URIs are not hierarchical, the paragraph in [36] that states that "all paths at or deeper than the depth of the last symbolic element in the path field of the Request-URI also are within the protection space specified by the Basic realm value of the current challenge" does not apply for SIP. SIP clients MAY preemptively send the corresponding Authorization header with requests for SIP URIs within the same protection realm (as defined above) without receipt of another challenge from the server.

14.3 Digest Authentication

The rules for digest authentication follow those defined in [36], with "HTTP 1.1" replaced by "SIP/2.0" in addition to the following differences:

1. The URI included in the challenge has the following BNF:

   URI = SIP-URL

2. The BNF for digest-uri-value is:

   digest-uri-value = Request-URI ; a defined in Section 4.3

3. The example procedure for choosing a nonce based on Etag does not work for SIP.

4. The Authentication-Info and Proxy-Authentication-Info fields are not used in SIP.

5. The text in [36] regarding cache operation does not apply to SIP.

6. [36] requires that a server check that the URI in the request line, and the URI included in the Authorization header, point to the same resource. In a SIP context, these two URI's may actually refer to different users, due to forwarding at some proxy. Therefore, in SIP, a server MAY check that the request-uri in the Authorization header corresponds to a user that the server is willing to accept forwarded or direct calls for.

14.4 Proxy-Authentication

The use of the Proxy-Authentication and Proxy-Authorization parallel that as described in [36], with one difference. Proxies MUST NOT add the Proxy-Authorization header. 407 responses MUST be forwarded upstream towards the client following the procedures for any other response. It is the client's responsibility to add the Proxy-Authorization header containing credentials for the proxy which has asked for authentication.

If a proxy were to resubmit a request with a Proxy-Authorization header field, it would need to increment the CSeq in the new request. However, this would mean that the UAC which submitted the original request would discard a response from the UAS, as the CSeq value would be different.

See sections 6.26 and 6.27 for additional information on usage of these fields as they apply to SIP.

15 SIP Security Using PGP

15.1 PGP Authentication Scheme

The "pgp" authentication scheme is based on the model that the client authenticates itself with a request signed with the client's private key. The server can then ascertain the origin of the request if it has access to the public key, preferably signed by a trusted third party.

15.1.1 The WWW-Authenticate Response Header

```
WWW-Authenticate =  "WWW-Authenticate" ":" "pgp" pgp-challenge
pgp-challenge    =  * (";" pgp-params )
pgp-params       =  realm | pgp-version | pgp-algorithm | nonce
realm            =  "realm" "=" realm-value
realm-value      =  quoted-string
pgp-version      =  "version" "="
                     <"> digit *( "." digit ) *letter <">
pgp-algorithm    =  "algorithm" "=" ( "md5" | "sha1" | token )
nonce            =  "nonce" "=" nonce-value
nonce-value      =  quoted-string
```

The meanings of the values of the parameters used above are as follows:

realm: A string to be displayed to users so they know which identity to use. This string SHOULD contain at least the name of the host performing the authentication and MAY additionally indicate the collection of users who might have access. An example might be " Users with call-out privileges ".

pgp-algorithm: The value of this parameter indicates the PGP message integrity check (MIC) to be used to produce the signature. If this not present it is assumed to be "md5". The currently defined values are "md5" for the MD5 checksum, and "sha1" for the SHA.1 algorithm.

pgp-version: The version of PGP that the client MUST use. Common values are "2.6.2" and "5.0". The default is 5.0.

nonce: A server-specified data string which should be uniquely generated each time a 401 response is made. It is RECOMMENDED that this string be base64 or hexadecimal data. Specifically, since the string is passed in the header lines as a quoted string, the double-quote character is not allowed. The contents of the nonce are implementation dependent. The quality of the implementation depends on a good choice. Since the nonce is used

only to prevent replay attacks and is signed, a time stamp in units convenient to the server is sufficient.

Replay attacks within the duration of the call setup are of limited interest, so that timestamps with a resolution of a few seconds are often should be sufficient. In that case, the server does not have to keep a record of the nonces.

Example:

```
WWW-Authenticate: pgp ;version="5.0"
  ;realm="Your Startrek identity, please" ;algorithm=md5
  ;nonce="913082051"
```

15.1.2 The Authorization Request Header

The client is expected to retry the request, passing an Authorization header line, which is defined as follows.

```
Authorization  =  "Authorization" ":" "pgp" *( ";" pgp-response )
pgp-response   =  realm | pgp-version | pgp-signature
                  | signed-by | nonce
pgp-signature  =  "signature" "=" quoted-string
signed-by      =  "signed-by" "=" <"> URI <">
```

The client MUST increment the CSeq header before resubmitting the request. The signature MUST correspond to the From header of the request unless the signed-by parameter is provided.

pgp-signature: The PGP ASCII-armored signature [33], as it appears between the "BEGIN PGP MESSAGE" and "END PGP MESSAGE" delimiters, without the version indication. The signature is included without any linebreaks.

The signature is computed across the nonce (if present), request method, request version and header fields following the Authorization header and the message body, in the same order as they appear in the message. The request method and version are prepended to the header fields without any white space. The signature is computed across the headers as sent, and the terminating CRLF. The CRLF following the Authorization header is NOT included in the signature.

A server MAY be configured not to generate nonces only if replay attacks are not a concern.

Not generating nonces avoids the additional set of request, 401 response and possibly ACK messages and reduces delay by one round-trip time.

Using the ASCII-armored version is about 25% less space-efficient than including the binary signature, but it is significantly easier for the receiver to piece together. Versions of the PGP program always include the full (compressed) signed text in their output unless ASCII-armored mode (-sta) is specified. Typical signatures are about 200 bytes long. -- The PGP signature mechanism allows the client to simply pass the request to an external PGP program. This relies on the requirement that proxy servers are not allowed to reorder or change header fields.

realm: The realm is copied from the corresponding WWW-Authenticate header field parameter.

signed-by: If and only if the request was not signed by the entity listed in the From header, the signed-by header indicates the name of the signing entity, expressed as a URI.

Receivers of signed SIP messages SHOULD discard any end-to-end header fields above the Authorization header, as they may have been maliciously added en route by a proxy.

Example:

```
Authorization: pgp version="5.0"
  ;realm="Your Startrek identity, please"
  ;nonce="913082051"
  ;signature="iQB1AwUBNNJiUaYBnHmiiQhlAQFYsgL/Wt3dk6TWK81/b0gcNDf
  VAUGU4rhEBW972IPxFSOZ94LlqhCLInTPaqhHFw1cb3lB0lrA0RhpV4t5yCdUt
  SRYBSkOK29o5elKlFeW23EzYPVUm2TlDAhbcjbMdfC+KLFX
  =aIrx"
```

15.2 PGP Encryption Scheme

The PGP encryption scheme uses the following syntax:

```
Encryption    =  "Encryption" ":" "pgp" pgp-eparams
pgp-eparams   =  1# ( pgp-version | pgp-encoding )
pgp-encoding  =  "encoding" "=" "ascii" | token
```

encoding: Describes the encoding or "armor" used by PGP. The value "ascii" refers to the standard PGP ASCII armor, without the lines containing "BEGIN PGP MESSAGE" and "END PGP MESSAGE" and without the version identifier. By default, the encrypted part is included as binary.

Example:

```
Encryption: pgp version="2.6.2", encoding="ascii"
```

15.3 Response-Key Header Field for PGP

```
Response-Key  =  "Response-Key" ":" "pgp" pgp-eparams
pgp-eparams   =  1# ( pgp-version | pgp-encoding | pgp-key)
pgp-key       =  "key" "=" quoted-string
```

If ASCII encoding has been requested via the encoding parameter, the key parameter contains the user's public key as extracted from the pgp key ring with the "pgp -kxa user ".

Example:

```
Response-Key: pgp version="2.6.2", encoding="ascii",
  key="mQBtAzNWHNYAAAEDAL7QvAdK2utY05wuUG+ItYK5tCF8HNJM60sU4rLaV+eUnkMk
  mOmJWtc2wXcZx1XaXb2lkydTQOesrUR75IwNXBuZXPEIMThEa5WLsT7VLme7njnx
  sE86SgWmAZx5ookIdQAFEbQxSGVubmluZyBTY2hlbHpyaW5uZSA8c2NodWx6cmlu
  bmVAY3MuY29sdW1iaWEuZWR1Pg==
  =+y19"
```

16 Examples

In the following examples, we often omit the message body and the corresponding Content-Length and Content-Type headers for brevity.

16.1 Registration

A user at host saturn.bell-tel.com registers on start-up, via multicast, with the local SIP server named bell-tel.com. In the example, the user agent on saturn expects to receive SIP requests on UDP port 3890.

```
C->S: REGISTER sip:bell-tel.com SIP/2.0
      Via: SIP/2.0/UDP saturn.bell-tel.com
      From: sip:watson@bell-tel.com
      To: sip:watson@bell-tel.com
      Call-ID: 70710@saturn.bell-tel.com
      CSeq: 1 REGISTER
      Contact: <sip:watson@saturn.bell-tel.com:3890;transport=udp>
      Expires: 7200
```

The registration expires after two hours. Any future invitations for watson@bell-tel.com arriving at sip.bell-tel.com will now be redirected to watson@saturn.bell-tel.com, UDP port 3890.

If Watson wants to be reached elsewhere, say, an on-line service he uses while traveling, he updates his reservation after first cancelling any existing locations:

```
C->S: REGISTER sip:bell-tel.com SIP/2.0
      Via: SIP/2.0/UDP saturn.bell-tel.com
      From: sip:watson@bell-tel.com
      To: sip:watson@bell-tel.com
      Call-ID: 70710@saturn.bell-tel.com
      CSeq: 2 REGISTER
      Contact: *
      Expires: 0

C->S: REGISTER sip:bell-tel.com SIP/2.0
      Via: SIP/2.0/UDP saturn.bell-tel.com
      From: sip:watson@bell-tel.com
      To: sip:watson@bell-tel.com
      Call-ID: 70710@saturn.bell-tel.com
      CSeq: 3 REGISTER
      Contact: sip:tawatson@example.com
```

Now, the server will forward any request for Watson to the server at example.com, using the Request-URI tawatson@example.com. For the server at example.com to reach Watson, he will need to send a REGISTER there, or inform the server of his current location through some other means.

It is possible to use third-party registration. Here, the secretary jon.diligent registers his boss, T. Watson:


```
C->S: REGISTER sip:bell-tel.com SIP/2.0
      Via: SIP/2.0/UDP pluto.bell-tel.com
```

```
From: sip:jon.diligent@bell-tel.com
To: sip:watson@bell-tel.com
Call-ID: 17320@pluto.bell-tel.com
CSeq: 1 REGISTER
Contact: sip:tawatson@example.com
```

The request could be sent to either the registrar at bell-tel.com or the server at example.com. In the latter case, the server at example.com would proxy the request to the address indicated in the Request-URI. Then, Max-Forwards header could be used to restrict the registration to that server.

16.2 Invitation to a Multicast Conference

The first example invites schooler@vlsi.cs.caltech.edu to a multicast session. All examples use the Session Description Protocol (SDP) (RFC 2327 [6]) as the session description format.

16.2.1 Request

```
C->S: INVITE sip:schooler@cs.caltech.edu SIP/2.0
      Via: SIP/2.0/UDP csvax.cs.caltech.edu;branch=8348
        ;maddr=239.128.16.254;ttl=16
      Via: SIP/2.0/UDP north.east.isi.edu
      From: Mark Handley <sip:mjh@isi.edu>
      To: Eve Schooler <sip:schooler@caltech.edu>
      Call-ID: 2963313058@north.east.isi.edu
      CSeq: 1 INVITE
      Subject: SIP will be discussed, too
      Content-Type: application/sdp
      Content-Length: 187

      v=0
      o=user1 53655765 2353687637 IN IP4 128.3.4.5
      s=Mbone Audio
      i=Discussion of Mbone Engineering Issues
      e=mbone@somewhere.com
      c=IN IP4 224.2.0.1/127
      t=0 0
      m=audio 3456 RTP/AVP 0
```


The From request header above states that the request was initiated by mjh@isi.edu and addressed to schooler@caltech.edu (From header fields). The Via fields list the hosts along the path from invitation initiator (the last element of the list) towards the callee. In the example above, the message was last multicast to the administratively scoped group 239.128.16.254 with a ttl of 16 from the host

csvax.cs.caltech.edu. The second Via header field indicates that it was originally sent from the host north.east.isi.edu. The Request-URI indicates that the request is currently being being addressed to schooler@cs.caltech.edu, the local address that csvax looked up for the callee.

In this case, the session description is using the Session Description Protocol (SDP), as stated in the Content-Type header.

The header is terminated by an empty line and is followed by a message body containing the session description.

16.2.2 Response

The called user agent, directly or indirectly through proxy servers, indicates that it is alerting ("ringing") the called party:

```
S->C: SIP/2.0 180 Ringing
      Via: SIP/2.0/UDP csvax.cs.caltech.edu;branch=8348
        ;maddr=239.128.16.254;ttl=16
      Via: SIP/2.0/UDP north.east.isi.edu
      From: Mark Handley <sip:mjh@isi.edu>
      To: Eve Schooler <sip:schooler@caltech.edu> ;tag=9883472
      Call-ID: 2963313058@north.east.isi.edu
      CSeq: 1 INVITE
```

A sample response to the invitation is given below. The first line of the response states the SIP version number, that it is a 200 (OK) response, which means the request was successful. The Via headers are taken from the request, and entries are removed hop by hop as the response retraces the path of the request. A new authentication field MAY be added by the invited user's agent if required. The Call-ID is taken directly from the original request, along with the remaining fields of the request message. The original sense of From field is preserved (i.e., it is the session initiator).

In addition, the Contact header gives details of the host where the user was located, or alternatively the relevant proxy contact point which should be reachable from the caller's host.


```
S->C: SIP/2.0 200 OK
      Via: SIP/2.0/UDP csvax.cs.caltech.edu;branch=8348
        ;maddr=239.128.16.254;ttl=16
      Via: SIP/2.0/UDP north.east.isi.edu
      From: Mark Handley <sip:mjh@isi.edu>
      To: Eve Schooler <sip:schooler@caltech.edu> ;tag=9883472
      Call-ID: 2963313058@north.east.isi.edu
      CSeq: 1 INVITE
      Contact: sip:es@jove.cs.caltech.edu
```

The caller confirms the invitation by sending an ACK request to the location named in the Contact header:

```
C->S: ACK sip:es@jove.cs.caltech.edu SIP/2.0
      Via: SIP/2.0/UDP north.east.isi.edu
      From: Mark Handley <sip:mjh@isi.edu>
      To: Eve Schooler <sip:schooler@caltech.edu> ;tag=9883472
      Call-ID: 2963313058@north.east.isi.edu
      CSeq: 1 ACK
```

16.3 Two-party Call

For two-party Internet phone calls, the response must contain a description of where to send the data. In the example below, Bell calls Watson. Bell indicates that he can receive RTP audio codings 0 (PCMU), 3 (GSM), 4 (G.723) and 5 (DVI4).

```
C->S: INVITE sip:watson@boston.bell-tel.com SIP/2.0
      Via: SIP/2.0/UDP kton.bell-tel.com
      From: A. Bell <sip:a.g.bell@bell-tel.com>
      To: T. Watson <sip:watson@bell-tel.com>
      Call-ID: 3298420296@kton.bell-tel.com
      CSeq: 1 INVITE
      Subject: Mr. Watson, come here.
      Content-Type: application/sdp
      Content-Length: ...

      v=0
      o=bell 53655765 2353687637 IN IP4 128.3.4.5
      s=Mr. Watson, come here.
      c=IN IP4 kton.bell-tel.com
      m=audio 3456 RTP/AVP 0 3 4 5
```



```
S->C: SIP/2.0 100 Trying
      Via: SIP/2.0/UDP kton.bell-tel.com
      From: A. Bell <sip:a.g.bell@bell-tel.com>
      To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311
      Call-ID: 3298420296@kton.bell-tel.com
      CSeq: 1 INVITE
      Content-Length: 0

S->C: SIP/2.0 180 Ringing
      Via: SIP/2.0/UDP kton.bell-tel.com
      From: A. Bell <sip:a.g.bell@bell-tel.com>
      To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311
      Call-ID: 3298420296@kton.bell-tel.com
      CSeq: 1 INVITE
```

```
      Content-Length: 0

S->C: SIP/2.0 182 Queued, 2 callers ahead
      Via: SIP/2.0/UDP kton.bell-tel.com
      From: A. Bell <sip:a.g.bell@bell-tel.com>
      To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311
      Call-ID: 3298420296@kton.bell-tel.com
      CSeq: 1 INVITE
      Content-Length: 0

S->C: SIP/2.0 182 Queued, 1 caller ahead
      Via: SIP/2.0/UDP kton.bell-tel.com
      From: A. Bell <sip:a.g.bell@bell-tel.com>
      To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311
      Call-ID: 3298420296@kton.bell-tel.com
      CSeq: 1 INVITE
      Content-Length: 0

S->C: SIP/2.0 200 OK
      Via: SIP/2.0/UDP kton.bell-tel.com
      From: A. Bell <sip:a.g.bell@bell-tel.com>
      To: <sip:watson@bell-tel.com> ;tag=37462311
      Call-ID: 3298420296@kton.bell-tel.com
      CSeq: 1 INVITE
      Contact: sip:watson@boston.bell-tel.com
      Content-Type: application/sdp
      Content-Length: ...

      v=0
      o=watson 4858949 4858949 IN IP4 192.1.2.3
      s=I'm on my way
      c=IN IP4 boston.bell-tel.com
      m=audio 5004 RTP/AVP 0 3
```


The example illustrates the use of informational status responses. Here, the reception of the call is confirmed immediately (100), then, possibly after some database mapping delay, the call rings (180) and is then queued, with periodic status updates.

Watson can only receive PCMU and GSM. Note that Watson's list of codecs may or may not be a subset of the one offered by Bell, as each party indicates the data types it is willing to receive. Watson will send audio data to port 3456 at c.bell-tel.com, Bell will send to port 5004 at boston.bell-tel.com.

By default, the media session is one RTP session. Watson will receive RTCP packets on port 5005, while Bell will receive them on port 3457.

Since the two sides have agreed on the set of media, Bell confirms the call without enclosing another session description:

```
C->S: ACK sip:watson@boston.bell-tel.com SIP/2.0
      Via: SIP/2.0/UDP kton.bell-tel.com
      From: A. Bell <sip:a.g.bell@bell-tel.com>
      To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311
      Call-ID: 3298420296@kton.bell-tel.com
      CSeq: 1 ACK
```

16.4 Terminating a Call

To terminate a call, caller or callee can send a BYE request:

```
C->S: BYE sip:watson@boston.bell-tel.com SIP/2.0
      Via: SIP/2.0/UDP kton.bell-tel.com
      From: A. Bell <sip:a.g.bell@bell-tel.com>
      To: T. A. Watson <sip:watson@bell-tel.com> ;tag=37462311
      Call-ID: 3298420296@kton.bell-tel.com
      CSeq: 2 BYE
```

If the callee wants to abort the call, it simply reverses the To and From fields. Note that it is unlikely that a BYE from the callee will traverse the same proxies as the original INVITE.

16.5 Forking Proxy

In this example, Bell (a.g.bell@bell-tel.com) (C), currently seated at host c.bell-tel.com wants to call Watson (t.watson@ieee.org). At the time of the call, Watson is logged in at two workstations, t.watson@x.bell-tel.com (X) and watson@y.bell-tel.com (Y), and has registered with the IEEE proxy server (P) called sip.ieee.org. The IEEE server also has a registration for the home machine of Watson, at watson@h.bell-tel.com (H), as well as a permanent registration at watson@acm.org (A). For brevity, the examples omit the session description and Via header fields.

Bell's user agent sends the invitation to the SIP server for the ieee.org domain:

```
C->P: INVITE sip:t.watson@ieee.org SIP/2.0
      Via:     SIP/2.0/UDP c.bell-tel.com
      From:    A. Bell <sip:a.g.bell@bell-tel.com>
      To:      T. Watson <sip:t.watson@ieee.org>
      Call-ID: 31415@c.bell-tel.com
      CSeq:    1 INVITE
```

The SIP server at ieee.org tries the four addresses in parallel. It sends the following message to the home machine:

```
P->H: INVITE sip:watson@h.bell-tel.com SIP/2.0
      Via:     SIP/2.0/UDP sip.ieee.org ;branch=1
      Via:     SIP/2.0/UDP c.bell-tel.com
      From:    A. Bell <sip:a.g.bell@bell-tel.com>
      To:      T. Watson <sip:t.watson@ieee.org>
      Call-ID: 31415@c.bell-tel.com
      CSeq:    1 INVITE
```

This request immediately yields a 404 (Not Found) response, since Watson is not currently logged in at home:

```
H->P: SIP/2.0 404 Not Found
      Via:     SIP/2.0/UDP sip.ieee.org ;branch=1
      Via:     SIP/2.0/UDP c.bell-tel.com
      From:    A. Bell <sip:a.g.bell@bell-tel.com>
      To:      T. Watson <sip:t.watson@ieee.org>;tag=87454273
```



```
      Call-ID: 31415@c.bell-tel.com
      CSeq:    1 INVITE
```

The proxy ACKs the response so that host H can stop retransmitting it:

```
P->H: ACK sip:watson@h.bell-tel.com SIP/2.0
      Via:     SIP/2.0/UDP sip.ieee.org ;branch=1
      From:    A. Bell <sip:a.g.bell@bell-tel.com>
      To:      T. Watson <sip:t.watson@ieee.org>;tag=87454273
      Call-ID: 31415@c.bell-tel.com
      CSeq:    1 ACK
```

Also, P attempts to reach Watson through the ACM server:

```
P->A: INVITE sip:watson@acm.org SIP/2.0
      Via:     SIP/2.0/UDP sip.ieee.org ;branch=2
      Via:     SIP/2.0/UDP c.bell-tel.com
      From:    A. Bell <sip:a.g.bell@bell-tel.com>
      To:      T. Watson <sip:t.watson@ieee.org>
      Call-ID: 31415@c.bell-tel.com
      CSeq:    1 INVITE
```

In parallel, the next attempt proceeds, with an INVITE to X and Y:

```
P->X: INVITE sip:t.watson@x.bell-tel.com SIP/2.0
      Via:     SIP/2.0/UDP sip.ieee.org ;branch=3
      Via:     SIP/2.0/UDP c.bell-tel.com
      From:    A. Bell <sip:a.g.bell@bell-tel.com>
      To:      T. Watson <sip:t.watson@ieee.org>
      Call-ID: 31415@c.bell-tel.com
      CSeq:    1 INVITE

P->Y: INVITE sip:watson@y.bell-tel.com SIP/2.0
      Via:     SIP/2.0/UDP sip.ieee.org ;branch=4
      Via:     SIP/2.0/UDP c.bell-tel.com
      From:    A. Bell <sip:a.g.bell@bell-tel.com>
      To:      T. Watson <sip:t.watson@ieee.org>
      Call-ID: 31415@c.bell-tel.com
      CSeq:    1 INVITE
```


As it happens, both Watson at X and a colleague in the other lab at host Y hear the phones ringing and pick up. Both X and Y return 200s via the proxy to Bell.

```
X->P: SIP/2.0 200 OK
      Via:      SIP/2.0/UDP sip.ieee.org ;branch=3
      Via:      SIP/2.0/UDP c.bell-tel.com
      From:     A. Bell <sip:a.g.bell@bell-tel.com>
      To:       T. Watson <sip:t.watson@ieee.org> ;tag=192137601
      Call-ID:  31415@c.bell-tel.com
      CSeq:     1 INVITE
      Contact:  sip:t.watson@x.bell-tel.com

Y->P: SIP/2.0 200 OK
      Via:      SIP/2.0/UDP sip.ieee.org ;branch=4
      Via:      SIP/2.0/UDP c.bell-tel.com
      Contact:  sip:t.watson@y.bell-tel.com
      From:     A. Bell <sip:a.g.bell@bell-tel.com>
      To:       T. Watson <sip:t.watson@ieee.org> ;tag=35253448
      Call-ID:  31415@c.bell-tel.com
      CSeq:     1 INVITE
```

Both responses are forwarded to Bell, using the Via information. At this point, the ACM server is still searching its database. P can now cancel this attempt:

```
   P->A: CANCEL sip:watson@acm.org SIP/2.0
         Via:     SIP/2.0/UDP sip.ieee.org ;branch=2
         From:    A. Bell <sip:a.g.bell@bell-tel.com>
         To:      T. Watson <sip:t.watson@ieee.org>
         Call-ID: 31415@c.bell-tel.com
         CSeq:    1 CANCEL
```

The ACM server gladly stops its neural-network database search and responds with a 200. The 200 will not travel any further, since P is the last Via stop.

```
   A->P: SIP/2.0 200 OK
         Via:     SIP/2.0/UDP sip.ieee.org ;branch=2
         From:    A. Bell <sip:a.g.bell@bell-tel.com>
         To:      T. Watson <sip:t.watson@ieee.org>
```



```
         Call-ID: 31415@c.bell-tel.com
         CSeq:    1 CANCEL
```

Bell gets the two 200 responses from X and Y in short order. Bell's reaction now depends on his software. He can either send an ACK to both if human intelligence is needed to determine who he wants to talk to or he can automatically reject one of the two calls. Here, he acknowledges both, separately and directly to the final destination:

```
   C->X: ACK sip:t.watson@x.bell-tel.com SIP/2.0
         Via:      SIP/2.0/UDP c.bell-tel.com
         From:     A. Bell <sip:a.g.bell@bell-tel.com>
         To:       T. Watson <sip:t.watson@ieee.org>;tag=192137601
         Call-ID:  31415@c.bell-tel.com
         CSeq:     1 ACK

   C->Y: ACK sip:watson@y.bell-tel.com SIP/2.0
         Via:      SIP/2.0/UDP c.bell-tel.com
         From:     A. Bell <sip:a.g.bell@bell-tel.com>
         To:       T. Watson <sip:t.watson@ieee.org>;tag=35253448
         Call-ID:  31415@c.bell-tel.com
         CSeq:     1 ACK
```

After a brief discussion between Bell with X and Y, it becomes clear that Watson is at X. (Note that this is not a three-way call; only Bell can talk to X and Y, but X and Y cannot talk to each other.) Thus, Bell sends a BYE to Y, which is replied to:

```
   C->Y: BYE sip:watson@y.bell-tel.com SIP/2.0
         Via:      SIP/2.0/UDP c.bell-tel.com
         From:     A. Bell <sip:a.g.bell@bell-tel.com>
         To:       T. Watson <sip:t.watson@ieee.org>;tag=35253448
         Call-ID:  31415@c.bell-tel.com
         CSeq:     2 BYE

   Y->C: SIP/2.0 200 OK
         Via:      SIP/2.0/UDP c.bell-tel.com
         From:     A. Bell <sip:a.g.bell@bell-tel.com>
         To:       T. Watson <sip:t.watson@ieee.org>;tag=35253448
         Call-ID:  31415@c.bell-tel.com
         CSeq:     2 BYE
```


16.6 Redirects

Replies with status codes 301 (Moved Permanently) or 302 (Moved Temporarily) specify another location using the Contact field. Continuing our earlier example, the server P at ieee.org decides to redirect rather than proxy the request:

```
   P->C: SIP/2.0 302 Moved temporarily
         Via:     SIP/2.0/UDP c.bell-tel.com
         From:    A. Bell <sip:a.g.bell@bell-tel.com>
         To:      T. Watson <sip:t.watson@ieee.org>;tag=72538263
         Call-ID: 31415@c.bell-tel.com
         CSeq:    1 INVITE
         Contact: sip:watson@h.bell-tel.com,
                   sip:watson@acm.org, sip:t.watson@x.bell-tel.com,
                   sip:watson@y.bell-tel.com
         CSeq: 1 INVITE
```

As another example, assume Alice (A) wants to delegate her calls to Bob (B) while she is on vacation until July 29th, 1998. Any calls meant for her will reach Bob with Alice's To field, indicating to him what role he is to play. Charlie (C) calls Alice (A), whose server returns:

```
   A->C: SIP/2.0 302 Moved temporarily
         From: Charlie <sip:charlie@caller.com>
         To: Alice <sip:alice@anywhere.com> ;tag=2332462
         Call-ID: 27182@caller.com
         Contact: sip:bob@anywhere.com
         Expires: Wed, 29 Jul 1998 9:00:00 GMT
         CSeq: 1 INVITE
```

Charlie then sends the following request to the SIP server of the anywhere.com domain. Note that the server at anywhere.com forwards the request to Bob based on the Request-URI.

```
C->B: INVITE sip:bob@anywhere.com SIP/2.0
      From: sip:charlie@caller.com
      To: sip:alice@anywhere.com
      Call-ID: 27182@caller.com
      CSeq: 2 INVITE
```


In the third redirection example, we assume that all outgoing requests are directed through a local firewall F at caller.com, with Charlie again inviting Alice:

```
C->F: INVITE sip:alice@anywhere.com SIP/2.0
      From: sip:charlie@caller.com
      To: Alice <sip:alice@anywhere.com>
      Call-ID: 27182@caller.com
      CSeq: 1 INVITE
```

The local firewall at caller.com happens to be overloaded and thus redirects the call from Charlie to a secondary server S:

```
F->C: SIP/2.0 302 Moved temporarily
      From: sip:charlie@caller.com
      To: Alice <sip:alice@anywhere.com>
      Call-ID: 27182@caller.com
      CSeq: 1 INVITE
      Contact: <sip:alice@anywhere.com:5080;maddr=spare.caller.com>
```

Based on this response, Charlie directs the same invitation to the secondary server spare.caller.com at port 5080, but maintains the same Request-URI as before:

```
C->S: INVITE sip:alice@anywhere.com SIP/2.0
      From: sip:charlie@caller.com
      To: Alice <sip:alice@anywhere.com>
      Call-ID: 27182@caller.com
      CSeq: 2 INVITE
```

16.7 Negotiation

An example of a 606 (Not Acceptable) response is:

```
   S->C: SIP/2.0 606 Not Acceptable
         From: sip:mjh@isi.edu
         To: <sip:schooler@cs.caltech.edu> ;tag=7434264
         Call-ID: 14142@north.east.isi.edu
```



```
         CSeq: 1 INVITE
         Contact: sip:mjh@north.east.isi.edu
         Warning: 370 "Insufficient bandwidth (only have ISDN)",
           305 "Incompatible media format",
           330 "Multicast not available"
         Content-Type: application/sdp
         Content-Length: 50

         v=0
         s=Let's talk
         b=CT:128
         c=IN IP4 north.east.isi.edu
         m=audio 3456 RTP/AVP 5 0 7
         m=video 2232 RTP/AVP 31
```

In this example, the original request specified a bandwidth that was higher than the access link could support, requested multicast, and requested a set of media encodings. The response states that only 128 kb/s is available and that (only) DVI, PCM or LPC audio could be supported in order of preference.

The response also states that multicast is not available. In such a case, it might be appropriate to set up a transcoding gateway and re-invite the user.

16.8 OPTIONS Request

A caller Alice can use an OPTIONS request to find out the capabilities of a potential callee Bob, without "ringing" the designated address. Bob returns a description indicating that he is capable of receiving audio encodings PCM Ulaw (payload type 0), 1016 (payload type 1), GSM (payload type 3), and SX7300/8000 (dynamic payload type 99), and video encodings H.261 (payload type 31) and H.263 (payload type 34).

```
   C->S: OPTIONS sip:bob@example.com SIP/2.0
         From: Alice <sip:alice@anywhere.org>
         To: Bob <sip:bob@example.com>
         Call-ID: 6378@host.anywhere.org
         CSeq: 1 OPTIONS
         Accept: application/sdp

   S->C: SIP/2.0 200 OK
```

```
         From: Alice <sip:alice@anywhere.org>
         To: Bob <sip:bob@example.com> ;tag=376364382


Handley, et al.             Standards Track                    [Page 132]

RFC 2543            SIP: Session Initiation Protocol          March 1999


         Call-ID: 6378@host.anywhere.org
         Content-Length: 81
         Content-Type: application/sdp

         v=0
         m=audio 0 RTP/AVP 0 1 3 99
         m=video 0 RTP/AVP 31 34
         a=rtpmap:99 SX7300/8000
```

A Minimal Implementation

A.1 Client

All clients MUST be able to generate the INVITE and ACK requests. Clients MUST generate and parse the Call-ID, Content-Length, Content-Type, CSeq, From and To headers. Clients MUST also parse the Require header. A minimal implementation MUST understand SDP (RFC 2327, [6]). It MUST be able to recognize the status code classes 1 through 6 and act accordingly.

The following capability sets build on top of the minimal implementation described in the previous paragraph. In general, each capability listed below builds on the ones above it:

Basic: A basic implementation adds support for the BYE method to allow the interruption of a pending call attempt. It includes a User-Agent header in its requests and indicates its preferred language in the Accept-Language header.

Redirection: To support call forwarding, a client needs to be able to understand the Contact header, but only the SIP-URL part, not the parameters.

Firewall-friendly: A firewall-friendly client understands the Route and Record-Route header fields and can be configured to use a local proxy for all outgoing requests.

Negotiation: A client MUST be able to request the OPTIONS method and understand the 380 (Alternative Service) status and the Contact parameters to participate in terminal and media negotiation. It SHOULD be able to parse the Warning response header to provide useful feedback to the caller.

Authentication: If a client wishes to invite callees that require caller authentication, it MUST be able to recognize the 401 (Unauthorized) status code, MUST be able to generate the Authorization request header and MUST understand the WWW-Authenticate response header.

If a client wishes to use proxies that require caller authentication, it MUST be able to recognize the 407 (Proxy Authentication Required) status code, MUST be able to generate the Proxy-Authorization request header and understand the Proxy-Authenticate response header.

A.2 Server

A minimally compliant server implementation MUST understand the INVITE, ACK, OPTIONS and BYE requests. A proxy server MUST also understand CANCEL. It MUST parse and generate, as appropriate, the Call-ID, Content-Length, Content-Type, CSeq, Expires, From, Max-Forwards, Require, To and Via headers. It MUST echo the CSeq and Timestamp headers in the response. It SHOULD include the Server header in its responses.

A.3 Header Processing

Table 6 lists the headers that different implementations support. UAC refers to a user-agent client (calling user agent), UAS to a user-agent server (called user-agent).

The fields in the table have the following meaning. Type is as in Table 4 and 5. "-" indicates the field is not meaningful to this system (although it might be generated by it). "m" indicates the field MUST be understood. "b" indicates the field SHOULD be understood by a Basic implementation. "r" indicates the field SHOULD be understood if the system claims to understand redirection. "a" indicates the field SHOULD be understood if the system claims to support authentication. "e" indicates the field SHOULD be understood if the system claims to support encryption. "o" indicates support of the field is purely optional. Headers whose support is optional for all implementations are not shown.

	type	UAC	proxy	UAS	registrar
Accept	R	-	o	m	m
Accept-Encoding	R	-	-	m	m
Accept-Language	R	-	b	b	b
Allow	405	o	-	-	-
Authorization	R	a	o	a	a
Call-ID	g	m	m	m	m
Content-Encoding	g	m	-	m	m
Content-Length	g	m	m	m	m
Content-Type	g	m	-	m	m
CSeq	g	m	m	m	m
Encryption	g	e	-	e	e
Expires	g	-	o	o	m
From	g	m	o	m	m
Hide	R	-	m	-	-
Contact	R	-	-	-	m
Contact	r	r	r	-	-
Max-Forwards	R	-	b	-	-
Proxy-Authenticate	407	a	-	-	-
Proxy-Authorization	R	-	a	-	-
Proxy-Require	R	-	m	-	-
Require	R	m	-	m	m
Response-Key	R	-	-	e	e
Route	R	-	m	-	-
Timestamp	g	o	o	m	m
To	g	m	m	m	m
Unsupported	r	b	b	-	-
User-Agent	g	b	-	b	-
Via	g	m	m	m	m
WWW-Authenticate	401	a	-	-	-

Table 6: Header Field Processing Requirements

B Usage of the Session Description Protocol (SDP)

This section describes the use of the Session Description Protocol (SDP) (RFC 2327 [6]).

B.1 Configuring Media Streams

The caller and callee align their media descriptions so that the nth media stream ("m=" line) in the caller's session description corresponds to the nth media stream in the callee's description.

All media descriptions SHOULD contain "a=rtpmap" mappings from RTP payload types to encodings.

This allows easier migration away from static payload types.

If the callee wants to neither send nor receive a stream offered by the caller, the callee sets the port number of that stream to zero in its media description.

> There currently is no other way than port zero for the callee to refuse a bidirectional stream offered by the caller. Both caller and callee need to be aware what media tools are to be started.

For example, assume that the caller Alice has included the following description in her INVITE request. It includes an audio stream and two bidirectional video streams, using H.261 (payload type 31) and MPEG (payload type 32).

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.anywhere.com
c=IN IP4 host.anywhere.com
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 51372 RTP/AVP 31
a=rtpmap:31 H261/90000
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
```

The callee, Bob, does not want to receive or send the first video stream, so it returns the media description below:

```
v=0
o=bob 2890844730 2890844730 IN IP4 host.example.com
c=IN IP4 host.example.com
m=audio 47920 RTP/AVP 0 1
a=rtpmap:0 PCMU/8000
a=rtpmap:1 1016/8000
m=video 0 RTP/AVP 31
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
```


B.2 Setting SDP Values for Unicast

If a session description from a caller contains a media stream which is listed as send (receive) only, it means that the caller is only willing to send (receive) this stream, not receive (send). The same is true for the callee.

For receive-only and send-or-receive streams, the port number and address in the session description indicate where the media stream should be sent to by the recipient of the session description, either

caller or callee. For send-only streams, the address and port number have no significance and SHOULD be set to zero.

The list of payload types for each media stream conveys two pieces of information, namely the set of codecs that the caller or callee is capable of sending or receiving, and the RTP payload type numbers used to identify those codecs. For receive-only or send-and-receive media streams, a caller SHOULD list all of the codecs it is capable of supporting in the session description in an INVITE or ACK. For send-only streams, the caller SHOULD indicate only those it wishes to send for this session. For receive-only streams, the payload type numbers indicate the value of the payload type field in RTP packets the caller is expecting to receive for that codec type. For send-only streams, the payload type numbers indicate the value of the payload type field in RTP packets the caller is planning to send for that codec type. For send-and-receive streams, the payload type numbers indicate the value of the payload type field the caller expects to both send and receive.

If a media stream is listed as receive-only by the caller, the callee lists, in the response, those codecs it intends to use from among the ones listed in the request. If a media stream is listed as send-only by the caller, the callee lists, in the response, those codecs it is willing to receive among the ones listed in the the request. If the media stream is listed as both send and receive, the callee lists those codecs it is capable of sending or receiving among the ones listed by the caller in the INVITE. The actual payload type numbers in the callee's session description corresponding to a particular codec MUST be the same as the caller's session description.

If caller and callee have no media formats in common for a particular stream, the callee MUST return a session description containing the particular "m=" line, but with the port number set to zero, and no payload types listed.

If there are no media formats in common for all streams, the callee SHOULD return a 400 response, with a 304 Warning header field.

B.3 Multicast Operation

The interpretation of send-only and receive-only for multicast media sessions differs from that for unicast sessions. For multicast, send-only means that the recipient of the session description (caller or callee) SHOULD only send media streams to the address and port indicated. Receive-only means that the recipient of the session description SHOULD only receive media on the address and port indicated.

For multicast, receive and send multicast addresses are the same and all parties use the same port numbers to receive media data. If the session description provided by the caller is acceptable to the callee, the callee can choose not to include a session description or

MAY echo the description in the response.

A callee MAY, in the response, return a session description with some of the payload types removed, or port numbers set to zero (but no other value). This indicates to the caller that the callee does not support the given stream or media types which were removed. A callee MUST NOT change whether a given stream is send-only, receive-only, or send-and-receive.

If a callee does not support multicast at all, it SHOULD return a 400 status response and include a 330 Warning.

B.4 Delayed Media Streams

In some cases, a caller may not know the set of media formats which it can support at the time it would like to issue an invitation. This is the case when the caller is actually a gateway to another protocol which performs media format negotiation after call setup. When this occurs, a caller MAY issue an INVITE with a session description that contains no media lines. The callee SHOULD interpret this to mean that the caller wishes to participate in a multimedia session described by the session description, but that the media streams are not yet known. The callee SHOULD return a session description indicating the streams and media formats it is willing to support, however. The caller MAY update the session description either in the ACK request or in a re-INVITE at a later time, once the streams are known.

B.5 Putting Media Streams on Hold

If a party in a call wants to put the other party "on hold", i.e., request that it temporarily stops sending one or more media streams, a party re-invites the other by sending an INVITE request with a modified session description. The session description is the same as

in the original invitation (or response), but the "c" destination addresses for the media streams to be put on hold are set to zero (0.0.0.0).

B.6 Subject and SDP "s=" Line

The SDP "s=" line and the SIP Subject header field have different meanings when inviting to a multicast session. The session description line describes the subject of the multicast session, while the SIP Subject header field describes the reason for the invitation. The example in Section 16.2 illustrates this point. For invitations to two-party sessions, the SDP "s=" line MAY be left empty.

B.7 The SDP "o=" Line

The "o=" line is not strictly necessary for two-party sessions, but MUST be present to allow re-use of SDP-based tools.

C Summary of Augmented BNF

All of the mechanisms specified in this document are described in both prose and an augmented Backus-Naur Form (BNF) similar to that used by RFC 822 [9]. Implementors will need to be familiar with the notation in order to understand this specification. The augmented BNF includes the following constructs:

```
name  =  definition
```

The name of a rule is simply the name itself (without any enclosing "<" and ">") and is separated from its definition by the equal "=" character. White space is only significant in that indentation of continuation lines is used to indicate a rule definition that spans more than one line. Certain basic rules are in uppercase, such as SP, LWS, HT, CRLF, DIGIT, ALPHA, etc. Angle brackets are used within definitions whenever their presence will facilitate discerning the use of rule names.

"literal"

Quotation marks surround literal text. Unless stated otherwise, the text is case-insensitive.

rule1 | rule2

Elements separated by a bar ("|") are alternatives, e.g., "yes | no" will accept yes or no.

(rule1 rule2)

Elements enclosed in parentheses are treated as a single element. Thus, "(elem (foo | bar) elem)" allows the token sequences "elem foo elem" and "elem bar elem".

*rule

The character "*" preceding an element indicates repetition. The full form is "<n>*<m>element" indicating at least <n> and at most <m> occurrences of element. Default values are 0 and infinity so that "*(element)" allows any number, including zero; "1*element" requires at least one; and "1*2element" allows one or two.

[rule]

Square brackets enclose optional elements; "[foo bar]" is equivalent to "*1(foo bar)".

N rule

Specific repetition: "<n>(element)" is equivalent to "<n>*<n>(element)"; that is, exactly <n> occurrences of (element). Thus 2DIGIT is a 2-digit number, and 3ALPHA is a string of three alphabetic characters.

#rule

A construct "#" is defined, similar to "*", for defining lists of elements. The full form is "<n>#<m> element" indicating at least <n> and at most <m> elements, each separated by one or more commas (",") and OPTIONAL linear white space (LWS). This makes the usual form of lists very easy; a rule such as

```
( *LWS element *( *LWS "," *LWS element ))
```

can be shown as 1# element. Wherever this construct is used, null elements are allowed, but do not contribute to the count of elements present. That is, "(element), , (element)" is permitted, but counts as only two elements. Therefore, where at least one element is required, at least one non-null element MUST be present. Default values are 0 and infinity so that "#element" allows any number, including zero; "1#element" requires at least one; and "1#2element" allows one or two.

; comment

A semi-colon, set off some distance to the right of rule text, starts a comment that continues to the end of line. This is a simple way of including useful notes in parallel with the specifications.

implied *LWS

The grammar described by this specification is word-based. Except where noted otherwise, linear white space (LWS) can be included between any two adjacent words (token or quoted-string), and between adjacent tokens and separators, without changing the interpretation of a field. At least one delimiter (LWS and/or separators) MUST exist between any two tokens (for the definition of "token" below), since they would otherwise be interpreted as a single token.

C.1 Basic Rules

The following rules are used throughout this specification to describe basic parsing constructs. The US-ASCII coded character set is defined by ANSI X3.4-1986.

```
OCTET     =  <any 8-bit sequence of data>
CHAR      =  <any US-ASCII character (octets 0 - 127)>
upalpha   =  "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" |
             "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" |
```

```
                "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"
lowalpha  =   "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" |
                "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" |
                "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
alpha     =   lowalpha | upalpha
digit     =   "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
                "8" | "9"
alphanum  =   alpha | digit
CTL       =   <any US-ASCII control character
                (octets 0 -- 31) and DEL (127)>
CR        =   %d13 ; US-ASCII CR, carriage return character
LF        =   %d10 ; US-ASCII LF, line feed character
SP        =   %d32 ; US-ASCII SP, space character
HT        =   %d09 ; US-ASCII HT, horizontal tab character
CRLF      =   CR LF ; typically the end of a line
```

The following are defined in RFC 2396 [12] for the SIP URI:


```
unreserved  =  alphanum | mark
mark        =  "-" | "_" | "." | "!" | "~" | "*" | "'"
            |  "(" | ")"
escaped     =  "%" hex hex
```

SIP header field values can be folded onto multiple lines if the continuation line begins with a space or horizontal tab. All linear white space, including folding, has the same semantics as SP. A recipient MAY replace any linear white space with a single SP before interpreting the field value or forwarding the message downstream.

```
LWS  =  [CRLF] 1*( SP | HT ) ; linear whitespace
```

The TEXT-UTF8 rule is only used for descriptive field contents and values that are not intended to be interpreted by the message parser. Words of *TEXT-UTF8 contain characters from the UTF-8 character set (RFC 2279 [21]). In this regard, SIP differs from HTTP, which uses the ISO 8859-1 character set.

```
TEXT-UTF8  =  <any UTF-8 character encoding, except CTLs,
              but including LWS>
```

A CRLF is allowed in the definition of TEXT-UTF8 only as part of a header field continuation. It is expected that the folding LWS will be replaced with a single SP before interpretation of the TEXT-UTF8 value.

Hexadecimal numeric characters are used in several protocol elements.

```
hex  =  "A" | "B" | "C" | "D" | "E" | "F"
        | "a" | "b" | "c" | "d" | "e" | "f" | digit
```

Many SIP header field values consist of words separated by LWS or special characters. These special characters MUST be in a quoted string to be used within a parameter value.


```
token       =  1*< any CHAR  except CTL's  or separators>
separators  =  "(" | ")" | "<" | ">" | "@" |
               "," | ";" | ":" | "\" | <"> |
               "/" | "[" | "]" | "?" | "=" |
               "{" | "}" | SP | HT
```

Comments can be included in some SIP header fields by surrounding the comment text with parentheses. Comments are only allowed in fields containing "comment" as part of their field value definition. In all other fields, parentheses are considered part of the field value.

```
comment  =  "(" *(ctext | quoted-pair | comment) ")"
ctext    =  < any TEXT-UTF8  excluding "("  and ")">
```

A string of text is parsed as a single word if it is quoted using double-quote marks.

```
quoted-string  =  ( <"> *(qdtext | quoted-pair ) <"> )
qdtext         =  <any TEXT-UTF8 except <">>
```

The backslash character ("\") MAY be used as a single-character quoting mechanism only within quoted-string and comment constructs.

```
quoted-pair  =  " \ " CHAR
```

D Using SRV DNS Records

The following procedure is experimental and relies on DNS SRV records (RFC 2052 [14]). The steps listed below are used in place of the two steps in section 1.4.2.

If a step elicits no addresses, the client continues to the next step. However if a step elicits one or more addresses, but no SIP server at any of those addresses responds, then the client concludes the server is down and doesn't continue on to the next step.

When SRV records are to be used, the protocol to use when querying for the SRV record is "sip". SRV records contain port numbers for servers, in addition to IP addresses; the client always uses this port number when contacting the SIP server. Otherwise, the port number in the SIP URI is used, if present. If there is no port number in the URI, the default port, 5060, is used.

1. If the host portion of the Request-URI is an IP address, the client contacts the server at the given address. If the host portion of the Request-URI is not an IP address, the client proceeds to the next step.

2. The Request-URI is examined. If it contains an explicit port number, the next two steps are skipped.

3. The Request-URI is examined. If it does not specify a protocol (TCP or UDP), the client queries the name server for SRV records for both UDP (if supported by the client) and TCP (if supported by the client) SIP servers. The format of these queries is defined in RFC 2052 [14]. The results of the query or queries are merged together and ordered based on priority. Then, the searching technique outlined in RFC 2052 [14] is used to select servers in order. If DNS doesn't return any records, the user goes to the last step. Otherwise, the user attempts to contact each server in the order listed. If no server is contacted, the user gives up.

4. If the Request-URI specifies a protocol (TCP or UDP) that is supported by the client, the client queries the name server for SRV records for SIP servers of that protocol

type only. If the client does not support the protocol specified in the Request-URI, it gives up. The searching technique outlined in RFC 2052 [14] is used to select servers from the DNS response in order. If DNS doesn't

return any records, the user goes to the last step. Otherwise, the user attempts to contact each server in the order listed. If no server is contacted, the user gives up.

5. The client queries the name server for address records for the host portion of the Request-URI. If there were no address records, the client stops, as it has been unable to locate a server. By address record, we mean A RR's, AAAA RR's, or their most modern equivalent.

A client MAY cache a successful DNS query result. A successful query is one which contained records in the answer, and a server was contacted at one of the addresses from the answer. When the client wishes to send a request to the same host, it starts the search as if it had just received this answer from the name server. The server uses the procedures specified in RFC1035 [15] regarding cache invalidation when the time-to-live of the DNS result expires. If the client does not find a SIP server among the addresses listed in the cached answer, it starts the search at the beginning of the sequence described above.

For example, consider a client that wishes to send a SIP request. The Request-URI for the destination is sip:user@company.com. The client only supports UDP. It would follow these steps:

1. The host portion is not an IP address, so the client goes to step 2 above.

2. The client does a DNS query of QNAME="sip.udp.company.com", QCLASS=IN, QTYPE=SRV. Since it doesn't support TCP, it omits the TCP query. There were no addresses in the DNS response, so the client goes to the next step.

3. The client does a DNS query for A records for "company.com". An address is found, so that client attempts to contact a server at that address at port 5060.

E IANA Considerations

Section 4.4 describes a name space and mechanism for registering SIP options.

Section 6.41 describes the name space for registering SIP warn-codes.

F Acknowledgments

We wish to thank the members of the IETF MMUSIC WG for their comments and suggestions. Detailed comments were provided by Anders Kristensen, Jim Buller, Dave Devanathan, Yaron Goland, Christian Huitema, Gadi Karmi, Jonathan Lennox, Keith Moore, Vern Paxson, Moshe J. Sambol, and Eric Tremblay.

This work is based, inter alia, on [37,38].

G Authors' Addresses

Mark Handley
AT&T Center for Internet Research at ISCI (ACIRI)
1947 Center St., Suite 600
Berkeley, CA 94704-119
USA
Email: mjh@aciri.org

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
USA
Email: schulzrinne@cs.columbia.edu

Eve Schooler
Computer Science Department 256-80
California Institute of Technology
Pasadena, CA 91125
USA
Email: schooler@cs.caltech.edu

Jonathan Rosenberg
Lucent Technologies, Bell Laboratories
Rm. 4C-526
101 Crawfords Corner Road
Holmdel, NJ 07733
USA
Email: jdrosen@bell-labs.com

H Bibliography

[1] Pandya, R., "Emerging mobile and personal communication systems," IEEE Communications Magazine , vol. 33, pp. 44--52, June 1995.

[2] Braden, B., Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReSerVation protocol (RSVP) -- version 1 functional specification", RFC 2205, October 1997.

[3] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: a transport protocol for real-time applications", RFC 1889, Internet Engineering Task Force, Jan. 1996.

[4] Schulzrinne, H., Lanphier, R. and A. Rao, "Real time streaming protocol (RTSP)", RFC 2326, April 1998.

[5] Handley, M., "SAP: Session announcement protocol," Internet Draft, Internet Engineering Task Force, Nov. 1996. Work in progress.

[6] Handley, M. and V. Jacobson, "SDP: session description protocol", RFC 2327, April 1998.

[7] International Telecommunication Union, "Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service," Recommendation H.323, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, May 1996.

[8] International Telecommunication Union, "Control protocol for multimedia communication," Recommendation H.245, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.

[9] International Telecommunication Union, "Media stream packetization and synchronization on non-guaranteed quality of service LANs," Recommendation H.225.0, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Nov. 1996.

[10] Bradner, S., "Key words for use in RFCs to indicate requirement levels", BCP 14, RFC 2119, Mardch 1997.

[11] Fielding, R., Gettys, J., Mogul, J., Nielsen, H. and T. Berners-Lee, "Hypertext transfer protocol -- HTTP/1.1", RFC 2068, January 1997.

[12] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform resource identifiers (URI): generic syntax", RFC 2396, August 1998.

[13] Berners-Lee, T., Masinter, L. and M. McCahill, "Uniform resource locators (URL)", RFC 1738, December 1994.

[14] Gulbrandsen, A. and P. Vixie, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2052, October 1996.

[15] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, Noveberm 1997.

[16] Hamilton, M. and R. Wright, "Use of DNS aliases for network services", RFC 2219, October 1997.

[17] Zimmerman, D., "The finger user information protocol", RFC 1288, December 1991.

[18] Williamson, S., Kosters, M., Blacka, D., Singh, J. and K. Zeilstra, "Referral whois (rwhois) protocol V1.5", RFC 2167, June 1997.

[19] Yeong, W., Howes, T. and S. Kille, "Lightweight directory access protocol", RFC 1777, March 1995.

[20] Schooler, E., "A multicast user directory service for synchronous rendezvous," Master's Thesis CS-TR-96-18, Department of Computer Science, California Institute of Technology, Pasadena, California, Aug. 1996.

[21] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998.

[22] Stevens, W., TCP/IP illustrated: the protocols , vol. 1. Reading, Massachusetts: Addison-Wesley, 1994.

[23] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[24] Crocker, D., "Standard for the format of ARPA internet text messages", RFC STD 11, RFC 822, August 1982.

[25] Meyer, D., "Administratively scoped IP multicast", RFC 2365, July 1998.

[26] Schulzrinne, H., "RTP profile for audio and video conferences with minimal control", RFC 1890, January 1996

[27] Eastlake, D., Crocker, S. and J. Schiller, "Randomness recommendations for security", RFC 1750, December 1994.

[28] Hoffman, P., Masinter, L. and J. Zawinski, "The mailto URL scheme", RFC 2368, July 1998.

[29] Braden, B., "Requirements for internet hosts - application and support", STD 3, RFC 1123, October 1989.

[30] Palme, J., "Common internet message headers", RFC 2076, February 1997.

[31] Alvestrand, H., "IETF policy on character sets and languages", RFC 2277, January 1998.

[32] Elkins, M., "MIME security with pretty good privacy (PGP)", RFC 2015, October 1996.

[33] Atkins, D., Stallings, W. and P. Zimmermann, "PGP message exchange formats", RFC 1991, August 1996.

[34] Atkinson, R., "Security architecture for the internet protocol", RFC 2401, November 1998.

[35] Allen, C. and T. Dierks, "The TLS protocol version 1.0," RFC 2246, January 1999.

[36] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A. and L. Stewart, "HTTP authentication: Basic and digest access authentication," Internet Draft, Internet Engineering Task Force, Sept. 1998. Work in progress.

[37] Schooler, E., "Case study: multimedia conference control in a packet-switched teleconferencing system," Journal of Internetworking: Research and Experience , vol. 4, pp. 99--120, June 1993. ISI reprint series ISI/RS-93-359.

[38] Schulzrinne, H., "Personal mobility for multimedia services in the Internet," in European Workshop on Interactive Distributed Multimedia Systems and Services (IDMS) , (Berlin, Germany), Mar. 1996.

Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this

document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

About Sippy RTPproxy

The Sippy RTPproxy is a high-performance software proxy server for RTP streams that can work together with SIP Express Router (SER), OpenSIPS, Kamailio, Sippy B2BUA or reSIProcate B2BUA.

The main purpose of RTPproxy is to make the communication between SIP user agents behind NAT(s) (Network Address Translator) possible. Several cases exists when direct end-to-end communication is not possible and RTP streams have to be relayed through another host. The RTPproxy can be used to setup such a relaying host.

23.03.09 New stable version 1.2.0 of RTPproxy has been released. The new release includes several new features and numerous fixes and improvements.

12.01.09 New beta snapshot of 1.2 version of RTPproxy has been released. This snapshot includes Linux build fix as well as some improvements in re-packetization feature.

24.12.08 New snapshot of development 1.2 version of RTPproxy has been released. This snapshot includes Sun Solaris fixes (now it should compile and run clearly) and improved disconnect notification feature.

04.11.08 New snapshot of development 1.2 version of RTPproxy has been released. This release includes many bug fixes and new features, including Solaris compatibility fixes, support for the new rtpproxy_stream2xxx() command in SER, improved support for re-INVITE, port randomization support and syslog support.

Older Entries..

Originally created for handling NAT scenarious it can also act as a generic media relay as well as gateway RTP sessions between IPv4 and IPv6 networks. It can also perform number of additional functions, including call recording, playing pre-encoded announcements, real-time stream copying and RTP payload reframing.

The RTPproxy supports some advanced features, such as remote control mode, allowing building scalable distributed SIP VoIP networks. The nathelper module included into the SIP Express Router (SER), OpenSIPS or Kamailio as well Sippy B2BUA allow using multiple RTPproxy instances running on remote machines for fault-tolerance and load-balancing purposes.

RTPproxy was developed by Maxim Sobolev and now is being actively maintained by the Sippy Software, Inc.

Getting RTPproxy

Downloads

Current Versions

- Stable version 1.2.0 [Release Notes]

Archive Versions

- Stable version 1.1 [Release Notes]
- Stable version 1.0.2

Source code via anonymous CVS

To get the latest source code from the CVS use the following commands:

```
cvs -d:pserver:anonymous@cvs.ser.berlios.de:/cvsroot/ser login
```

press enter when prompted for a password

```
cvs -z3 -d:pserver:anonymous@cvs.ser.berlios.de:/cvsroot/ser co rtpproxy
```

then do the usual: ./configure; make; make install

You can also browse the repository here.

License

The Sippy RTPproxy is distributed under BSD-style license. You can find the full text of the license here. For a license to use the software under conditions other than those described here, or to purchase support for this software, please contact Sippy Software, Inc.

Getting Help

You can obtain free best-effort help from the RTPproxy community. To participate on the mailing list, subscribe at the following web address: http://lists.rtpproxy.org/mailman/listinfo/users and stick to problem reporting

guidelines. Discussion of development, new features and RTPproxy status as on CVS takes place at the following mailing list: http://lists.rtpproxy.org/mailman/listinfo/devel.

Please make sure to check FAQ page before requesting help from the mailing list.

Links

- RTPproxy client library for Java
- ErlRTPproxy - RTPproxy clone written in Erlang

PATENT COOPERATION TREATY

From the INTERNATIONAL BUREAU

PCT

NOTIFICATION CONCERNING TRANSMITTAL OF COPY OF INTERNATIONAL PRELIMINARY REPORT ON PATENTABILITY (CHAPTER I OF THE PATENT COOPERATION TREATY)
(PCT Rule 44bis.1(c))

RECEIVED
2011 FEB 14 A 11:
PEL
VANCOUVER, BC

To:

KNOX, John, W.
SMART & BIGGAR
2200-650 West Georgia Street
Vancouver, British Columbia V6B 4N8
CANADA

Date of mailing *(day/month/year)*
10 February 2011 (10.02.2011)

Applicant's or agent's file reference
83636-55

IMPORTANT NOTICE

International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/CA2009/001062	28 July 2009 (28.07.2009)	28 July 2008 (28.07.2008)

Applicant
DIGIFONICA (INTERNATIONAL) LIMITED et al

The International Bureau transmits herewith a copy of the international preliminary report on patentability (Chapter I of the Patent Cooperation Treaty)

The International Bureau of WIPO
34, chemin des Colombettes
1211 Geneva 20, Switzerland

Facsimile No. +41 22 338 82 70

Authorized officer
Athina Nickitas-Etienne

e-mail: pt04.pct@wipo.int

Form PCT/IB/326 (January 2004)

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY REPORT ON PATENTABILITY
(Chapter I of the Patent Cooperation Treaty)

(PCT Rule 44*bis*)

RECEIVED
2011 FEB 24 A 11: 38

Applicant's or agent's file reference 83636-55	**FOR FURTHER ACTION**	See item 4 below.
International application No. PCT/CA2009/001062	International filing date *(day/month/year)* 28 July 2009 (28.07.2009)	Priority date *(day/month/year)* 28 July 2008 (28.07.2008)
International Patent Classification (8th edition unless older edition indicated) See relevant information in Form PCT/ISA/237		
Applicant DIGIFONICA (INTERNATIONAL) LIMITED		

1. This international preliminary report on patentability (Chapter I) is issued by the International Bureau on behalf of the International Searching Authority under Rule 44 *bis*.1(a).

2. This REPORT consists of a total of 6 sheets, including this cover sheet.

   In the attached sheets, any reference to the written opinion of the International Searching Authority should be read as a reference to the international preliminary report on patentability (Chapter I) instead.

3. This report contains indications relating to the following items:

☒	Box No. I	Basis of the report
☐	Box No. II	Priority
☐	Box No. III	Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
☐	Box No. IV	Lack of unity of invention
☒	Box No. V	Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
☐	Box No. VI	Certain documents cited
☐	Box No. VII	Certain defects in the international application
☒	Box No. VIII	Certain observations on the international application

4. The International Bureau will communicate this report to designated Offices in accordance with Rules 44*bis*.3(c) and 93*bis*.1 but not, except where the applicant makes an express request under Article 23(2), before the expiration of 30 months from the priority date (Rule 44*bis* .2).

	Date of issuance of this report 01 February 2011 (01.02.2011)
The International Bureau of WIPO 34, chemin des Colombettes 1211 Geneva 20, Switzerland Facsimile No. +41 22 338 82 70	Authorized officer Athina Nickitas-Etienne e-mail: pt04.pct@wipo.int

Form PCT/IB/373 (January 2004)

PATENT COOPERATION TREATY

From the
INTERNATIONAL SEARCHING AUTHORITY

To:
SMART & BIGGAR
Box 11560 Vancouver Centre
2200 - 650 W. Georgia Street
VANCOUVER, British Columbia
Canada, V6B 4N8

PCT

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

(PCT Rule 43*bis*.1)

Date of mailing *(day/month/year)* 12 March 2010 (12-03-2010)

Applicant's or agent's file reference
83636-55

FOR FURTHER ACTION
See paragraph 2 below

International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/CA2009/001062	28 July 2009 (28-07-2009)	28 July 2008 (28-07-2008)

International Patent Classification (IPC) or both national classification and IPC
IPC: ***H04W 76/02*** (2009.01), ***H04W 88/06*** (2009.01), ***H04W 88/16*** (2009.01)

Applicant
DIGIFONICA (INTERNATIONAL) LIMITED ET AL

1. This opinion contains indications relating to the following items :

[X]	Box No. I	Basis of the opinion
[ ]	Box No. II	Priority
[ ]	Box No. III	Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
[ ]	Box No. IV	Lack of unity of invention
[X]	Box No. V	Reasoned statement under Rule 43*bis*.1(a)(I) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
[ ]	Box No. VI	Certain documents cited
[ ]	Box No. VII	Certain defects in the international application
[X]	Box No. VIII	Certain observations on the international application

2. **FURTHER ACTION**
If a demand for international preliminary examination is made, this opinion will be considered to be a written opinion of the International Preliminary Examining Authority ("IPEA") except that this does not apply where the applicant chooses an Authority other than this one to be the IPEA and the chosen IPEA has notified the International Bureau under Rule 66.1*bis*(b) that written opinions of this International Searching Authority will not be so considered.

If this opinion is, as provided above, considered to be a written opinion of the IPEA, the applicant is invited to submit to the IPEA a written reply together, where appropriate, with amendments, before the expiration of 3 months from the date of mailing of Form PCT/ISA/220 or before the expiration of 22 months from the priority date, whichever expires later.

For further options, see Form PCT/ISA/220.

3. For further details, see notes to Form PCT/ISA/220.

Name and mailing address of the ISA/CA	Date of completion of this opinion	Authorized officer
Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476	12 March 2010 (12-03-2010)	Elena Mikhailova (819) 953-5206

Box No. I **Basis of this opinion**

1. With regard to the **language**, this opinion has been established on the basis of:

   [X] the international application in the language in which it was filed

   [ ] a translation of the international application into , which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).

2. [ ] This opinion has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43*bis*.1(a))

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, this opinion has been established on the basis of a sequence listing filed or furnished:

   a. (means)

      [ ] on paper

      [ ] in electronic form

   b. (time)

      [ ] the international application as filed.

      [ ] together with the international application in electronic form

      [ ] subsequently to this Authority for the purposes of search

4. [ ] In addition, in the case that more than one version or copy of a sequence listing has been filed or furnished, the required statements that the information in the subsequent or additional copies is identical to that in the application as filed or does not go beyond the application as filed, as appropriate, were furnished.

5. Additional comments :

Box No. V **Reasoned statement under Rule 43*bis*.1(a)(I) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement**

1. Statement

Novelty (N)	Claims	1 to 90	YES
	Claims	None	NO
Inventive step (IS)	Claims	None	YES
	Claims	1 to 90	NO
Industrial applicability (IA)	Claims	1 to 90	YES
	Claims	None	NO

2. Citations and explanations :

Reference is made to the following documents:

D1: US 20070112964 A1 (Guedalia et al.) 17 May 2007 (17-05-2007)
D2: US 20040022237 A1 (Elliot et al.) 05 February 2004 (05-02-2004)

The present application relates to initiating a long distance call using a mobile telephone. The application is concerned with providing a technique (other than using a "calling card") for avoiding long distance charges when the mobile telephone is roaming in another mobile telephone service provider's network. To achieve the above object the following technique is suggested: the user of the mobile telephone provides a callee identifier and requests an access code from an access server. Upon receiving the access code, which is different from the callee identifier, a call to the callee is initiated using the access code. Claim 1, which defines a method of initiating a call to a callee using a mobile telephone, is directed to the above described technique. D1 discloses systems and methods for establishing and maintaining communications between two or more devices coupled to different communication networks. In particular, D1 teaches a method of initiating a call to a callee *using a mobile telephone* (abstract, claims 15, 53, par. 0014, 0115) including the steps of receiving from a user of the mobile phone a callee identifier associated with the callee (claim 1, par. 0068), transmitting an access code request message including the callee identifier (claim 1, par. 0011, 0014, 0068, 0142), receiving an access code reply message (par. 0135, 0142), and initiating a call with the mobile telephone using the access code to identify the callee (par. 0137). Although D1 does not explicitly mention that the access code reply message includes an access code different from the callee identifier and associated with the callee identifier, the mentioned technical feature is well known in the art, and disclosed in D2. D2 describes a toll-free Service Control Point (SCP) which can translate a toll free number to a final routing destination based on a flexible set of options selected by a subscriber which include NPA or NPA-XXX of a calling party. The SCP can have several possible responses to a routing request. One of the responses includes returning *a circuit identifier* associated with a called party to connect the call (par. 0710 and 0711). Therefore, D1 in combination with D2 discloses the subject matter of claim 1. As independent claims 12, 21, 22, 31, 32, 33, 52, 71 and 90 are all directed to the same subject matter as claim 1 the above arguments also apply to claims 12, 21, 22, 31, 32, 33, 52, 71 and 90.

It appears that dependent claims 2 to 11, 13 to 20, 23 to 30, 34 to 51, 53 to 70 and 72 to 89 fail to introduce any inventive limitation.

Article 33(2) PCT - Novelty

As none of the cited references taken alone discloses the technical features of claims 1 to 90, the subject matter of the claimed invention is considered to be novel thus meeting the requirements of **Article 33(2) PCT.**

Article 33(3) PCT - Inventive step

For the reason indicated in the above discussion the subject matter of claims 1 to 90 is not considered is not considered to involve inventive step thus failing to fulfill the requirements of **Article 33(3) PCT.**

Box No. VIII Certain observations on the international application

The following observations on the clarity of the claims, description, and drawings or on the question whether the claims are fully supported by the description, are made :

Article 6

In claims 52, 71 and 90 the term *said routing controller* lacks the antecedent.

Supplemental Box

In case **the space in any of the preceding boxes is not sufficient.**

Continuation of: Box V on page 3

Article 33(4) PCT - Industrial Applicability

As the present application relates to a well established field of wireless communication, the subject matter of claims 1 to 90 is considered to be industrially applicable under **Article 33(4) PCT.**

PATENT COOPERATION TREATY

From the
INTERNATIONAL PRELIMINARY EXAMINING AUTHORITY

To:
SMART & BIGGAR
Box 11560 Vancouver Centre
2200 - 650 W. Georgia Street
VANCOUVER, British Columbia
Canada, V6B 4N8

RECEIVED
2009 FEB 19 A 9:35
650 WEST GEORGIA ST.
VANCOUVER, B.C.

PCT

NOTIFICATION OF TRANSMITTAL OF INTERNATIONAL PRELIMINARY REPORT ON PATENTABILITY
(Chapter II of the Patent Cooperation Treaty)

(PCT Rule 71.1)

Date of mailing *(day/month/year)*: 13 February 2009 (13-02-2009)

Applicant's or agent's file reference
83636-13

IMPORTANT NOTIFICATION

International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/CA2007/002150	29 November 2007 (29-11-2007)	29 November 2006 (29-11-2006)

Applicant
DIGIFONICA (INTERNATIONAL) LIMITED ET AL

1. The applicant is hereby notified that this International Preliminary Examining Authority transmits herewith the international preliminary report on patentability and its annexes, if any, established on the international application.

2. A copy of the report and its annexes, if any, is being transmitted to the International Bureau for communication to all the elected Offices.

3. Where required by any of the elected Offices, the International Bureau will prepare an English translation of the report (but not of any annexes) and will transmit such translation to those Offices.

4. **REMINDER**

The applicant must enter the national phase before each elected Office by performing certain acts (filing translations and paying national fees) within 30 months from the priority date (or later in some Offices) (Article 39(1)) (see also the reminder sent by the International Bureau with Form PCT/IB/301).

Where a translation of the international application must be furnished to an elected Office, that translation must contain a translation of any annexes to the international preliminary report on patentability. It is the applicant's responsibility to prepare and furnish such translation directly to each elected Office concerned.

For further details on the applicable time limits and requirements of the elected Offices, see Volume II of the *PCT Applicant's Guide*.

The applicant's attention is drawn to Article 33(5), which provides that the criteria of novelty, inventive step and industrial applicability described in Article 33(2) to (4) merely serve the purposes of international preliminary examination and that "any Contracting State may apply additional or different criteria for the purposes of deciding whether, in that State, the claimed invention is patentable or not" (see also Article 27(5)). Such additional criteria may relate, for example, to exemptions from patentability, requirements for enabling disclosure, clarity and support for the claims.

Name and mailing address of the IPEA/CA	Authorized officer
Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476	Maureen Matheson 819- 953-1495

Form PCT/IPEA/416 (January 2004)

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY REPORT ON PATENTABILITY
(Chapter II of the Patent Cooperation Treaty)

(PCT Article 36 and Rule 70)

RECEIVED
2009 FEB 19 A 9:35
2200-650 WEST GEORGIA ST.
VANCOUVER, B.C

Applicant's or agent's file reference 83636-13	**FOR FURTHER ACTION**	See Form PCT/IPEA/416
International application No. PCT/CA2007/002150	International filing date *(day/month/year)* 29 November 2007 (29-11-2007)	Priority date *(day/month/year)* 29 November 2006 (29-11-2006)

International Patent Classification (IPC) or national classification and IPC
IPC: ***H04L 12/26*** (2006.01) , ***H04L 12/66*** (2006.01) , ***H04M 11/06*** (2006.01) , ***H04M 3/22*** (2006.01)

Applicant
DIGIFONICA (INTERNATIONAL) LIMITED ET AL

1. This report is the international preliminary examination report, established by this International Preliminary Examining Authority under Article 35 and transmitted to the applicant according to Article 36.
2. This REPORT consists of a total of 3 sheets, including this cover sheet.
3. This report is also accompanied by ANNEXES, comprising:
   a. [X] *(sent to the applicant and to the International Bureau)* a total of 23 sheets, as follows:
      [X] sheets of the description, claims and/or drawings which have been amended and are the basis of this report and/or sheets containing rectifications authorized by this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions).
      [ ] sheets which supersede earlier sheets, but which this Authority considers contain an amendment that goes beyond the disclosure in the international application as filed, as indicated in item 4 of Box No. 1 and the Supplemental Box.
   b. [ ] *(sent to the International Bureau only)* a total of (indicate type and number of electronic carrier(s)) _____, containing a sequence listing and/or tables related thereto, in electronic form only, as indicated in the Supplemental Box Relating to Sequence Listing (see Section 802 of the Administrative Instructions).
4. This report contains indications relating to the following items:
   - [X] Box No. I Basis of the report
   - [ ] Box No. II Priority
   - [ ] Box No. III Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
   - [ ] Box No. IV Lack of unity of invention
   - [X] Box No. V Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
   - [ ] Box No. VI Certain documents cited
   - [ ] Box No. VII Certain defects in the international application
   - [ ] Box No. VIII Certain observations on the international application

Date of submission of the demand	Date of completion of this report
13 May 2008 (13-05-2008)	13 February 2009 (13-02-2009)
Name and mailing address of the IPEA/CA Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476	Authorized officer Daniela Savin 819- 934-4890

INTERNATIONAL PRELIMINARY REPORT ON PATENTABILITY

International application No.
PCT/CA2007/002150

Box No. I Basis of the report

1. With regard to the **language**, this report is based on:

[X] the international application in the language in which it was filed

[ ] a translation of the international application into _____, which is the language of a translation furnished for the purposes of:

[ ] international search (Rules 12.3(a) and 23.1(b))

[ ] publication of the international application (Rule 12.4(a))

[ ] international preliminary examination (Rules 55.2(a) and/or 55.3(a))

2. With regard to the **elements** of the international application, this report is based on *(replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to this report):*

[ ] the international application as originally filed/furnished

[X] the description:

pages 2-23, 25-54, 56-62 as originally filed/furnished

pages* 1 received by this Authority on 13 May 2008 (13-05-2008)

pages* 24, 55 received by this Authority on 22 January 2009 (22-01-2009)

[X] the claims:

pages _____ as originally filed/furnished

pages* _____ as amended (together with any statement) under Article 19

pages* 63-70 received by this Authority on 05 February 2009 (05-02-2009)

pages* _____ received by this Authority on _____

[X] the drawings:

pages 1/29-5/29, 10/29-12/29, 14/29, 17/29, 19/29-22/29, 24/29, 27/29, 29/29 as originally filed/furnished

pages* 6/29-9/29, 13/29, 15/29-16/29 received by this Authority on 22 January 2009 (22-01-2009)

pages* 18/29, 23/29, 25/29-26/29, 28/29 received by this Authority on 22 January 2009 (22-01-2009)

[ ] a sequence listing and/or any related table(s) - see Supplemental Box Relating to Sequence Listing.

3. [ ] The amendments have resulted in the cancellation of:

[ ] the description, pages _____

[ ] the claims, Nos. _____

[ ] the drawings, sheets/figs _____

[ ] the sequence listing *(specify):* _____

[ ] any table(s) related to sequence listing *(specify):* _____

4. [ ] This report has been established as if (some of) the amendments annexed to this report and listed below had not been made, since they have been considered to go beyond the disclosure as filed, as indicated in the Supplemental Box (Rule 70.2(c)).

[ ] the description, pages _____

[ ] the claims, Nos. _____

[ ] the drawings, sheets/figs _____

[ ] the sequence listing *(specify):* _____

[ ] any table(s) related to sequence listing *(specify):* _____

5. [ ] This opinion has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 66.1(d-*bis*))

6. [ ] Supplementary international search report(s) from Authority(ies) _____ have been received and taken into account in drawing up this report (Rule45*bis*.8(b) and (c)).

**If item 4 applies, some or all of those sheets may be marked "superseded."*

T-Mobile Exhibit 1002
Part 3 of 6

Box No. V Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. Statement

Novelty (N)	Claims	1-26	YES
	Claims	None	NO
Inventive step (IS)	Claims	1-26	YES
	Claims	None	NO
Industrial applicability (IA)	Claims	1-26	YES
	Claims	None	NO

2. Citations and explanations (Rule 70.7)

Reference is made to the following document cited in the International Search Report:
D1: US 2004/0181599 A1 (Kreusch et al.) 16 September 2004 (16-09-2004)

The following is a review of how the subject matter described by the claims relates to the prior art of record.

Claims 1-26 are directed to a method and apparatus for intercepting communications in an IP network, in which communications between a subscriber and another party occur through a media relay. The aforementioned claims recite a methodology for intercepting IP communications comprising the following: determining whether determination information associated with a subscriber dialing profile meets intercept criteria; when said determination information meets said intercept criteria, causing the same media relay through which communications between said subscriber and said another party are relayed to produce a copy of said communications between said subscriber and said another party, while said same media relay relays communications between said subscriber and said another party; and causing said same media relay to send said copy to a mediation device identified by destination information associated with said subscriber dialing profile. Associating intercept information with a subscriber dialing profile happens when communications involving the subscriber are not in progress, as well as when the communications are in progress.

D1 is considered to be the prior art closest to claims 1-26. D1 describes a method for monitoring an IP data flow between at least two telecommunications terminals, which are connected to a data network via at least one access server. When monitoring takes place, the data flow is rerouted from the access server to a monitoring server, which makes a copy of the data flow and further transmits the copy to an evaluation unit (*see D1: abstract; paragraphs [0011]-[0015], [0019]-[0022], [0028], [0034]-[0036], [0048]-[0053], [0055]-[0061], [0067], [0072]-[0074], [0078]-[0083]; Figs. 1, 2a-2b; claims 1-3, 7-8, 25-26*).

However, D1 does not describe the same features as are found in independent claims 1 and 14. Specifically, the prior art of reference does not show that communications between a subscriber of an IP network and another party occur through a media relay only, and not through a gateway, an access server and eventually a monitoring server. Moreover, the same media relay through which communications between the subscriber and the other party are relayed, produces a copy of the communications when determination information meets intercept criteria, while continuing to relay the communications. This is in contrast with the intercept method disclosed in prior art, which diverts the communications to a monitoring server in order to make a copy of the data flow, since the monitoring server is not part of the normal communications flow, and it is only used when the communications need to be intercepted. Thus, the present application describes a methodology for intercepting IP communications directly at layer 3 of the OSI model, using only a media relay that relays communications between two telecommunications terminals and also makes copies of the communications, without the need of a dedicated monitoring server.

2.1 Novelty
Claims 1-26 are novel under **Article 33(2)** of the **PCT**, as the features of these claims are not explicitly shown in the prior art.

2.2 Inventive Step
Claims 1-26 involve an inventive step over the prior art, and therefore they **comply** with **PCT Article 33(3)**.

2.3 Industrial Applicability
Claims 1-26 are considered to be industrially applicable as per **PCT Article 33(4)**.

INTERCEPTING VOICE OVER IP COMMUNICATIONS AND OTHER DATA COMMUNICATIONS

BACKGROUND OF THE INVENTION

1. Field of Invention

This invention relates to data communications and methods and apparatus for intercepting data communications, particularly voice over IP data communications, in an IP network.

2. Description of Related Art

The term "lawful intercept" is used to describe a procedure which allows law enforcement agencies to perform electronic surveillance of telecommunications. Lawful intercept of telecommunications, particularly phone calls, is premised on a notion that a law enforcement agency has identified a person of interest, obtained a legal authorization for the surveillance (for example, a judicial or administrative warrant), and then contacted the person's telecommunications service provider that will be required to provide the law enforcement agency with a real-time copy of the person's communications. This real-time copy can then be used by the law enforcement agency to monitor or record the person's communications. Within the framework of traditional telecommunications networks, such as, for example, the Public Switched Telephone Network (PSTN) or cellular networks, lawful intercept generally presents a purely economic problem for the service providers that have to ensure that sufficient interception equipment and dedicated links to the law enforcement agencies have been deployed to satisfy lawful intercept requirements mandated by law. However, in the context of Voice over Internet Protocol (VoIP) communications, in addition to the economic problems mentioned above, lawful intercept presents

AMENDED SHEET

T-Mobile Exhibit 1002
Part 3 of 6

Routing Controller

Referring to Figure **7**, the routing controller **16** is shown in greater detail and includes a routing controller processor circuit shown generally at **200**. The RC processor circuit **200** includes a microprocessor **202**, program memory **204**, a table memory **206** and an I/O interface **208**, all in communication with the processor. There may be a plurality of processor circuits (**202**), memories (**204**), etc.

The I/O interface **208** includes a database output port **210** through which a request to the database **18** (Figure **1**) can be made and includes a database response port **212** for receiving a reply from the database. The I/O interface **208** further includes an RC Request message input **214** for receiving the RC Request message from the call controller **14** and includes a routing message output **216** for sending a routing message back to the call controller **14**.

The program memory **204** includes blocks of codes for directing the RC processor circuit **200** to carry out various functions of the routing controller **16**. One of these blocks implements an RC Request message handler process **250** which directs the RC to produce a routing message in response to a received RC Request message of the type shown at **150** in Figure **6**. Referring back to Figure **7**, the program memory **204** further includes a Law Enforcement Authority (LEA) request message handler **1400** and an in-call intercept shut down routine **1500**.

The RC Request message handler process **250** is shown in greater detail in Figures **8**A through **8**D.

RC Request Message Handler

Referring to Figure **8**A, the RC Request message handler process **250** begins with a first block **252** that directs the RC processor circuit **200** (Figure **7**) to store the contents of the RC Request message **150** (Figure **6**) in buffers. Block **254** then directs the RC processor circuit **200** to use the contents of the

AMENDED SHEET

T-Mobile Exhibit 1002
Part 3 of 6

the IP/UDP port address to which the audio data received at the caller and callee IP/UDP port addresses were being copied.

It will be appreciated that in the foregoing description, the components described cooperate to detect a requirement for intercept at the time a call is set up. In the following description an explanation is provided to describe how to intercept a call while the call is in progress.

Intercepting a Call in Progress

Referring back to Figure **1**, to intercept a call while the call is in progress, the law enforcement authority **293** may communicate with a mediation device, or may communicate with the call controller or may communicate with the routing controller or may communicate with a handover interface that communicates with any of the foregoing components to cause the routing controller to receive a law enforcement authority (LEA) intercept request message including intercept information, such as that which would be associated with fields **702-710** in Figure **9**, for example.

In response to receipt of a LEA intercept request message, the routing controller LEA request message handler shown at **1400** in Figure **44** is invoked.

The LEA request message handler **1400** begins with a first block **1402** that directs the routing controller processor circuit to communicate with the database **18** in which dialing profile records of the type shown in Figure **9** are stored to find a dialing profile associated with the user whose calls are to be monitored.

If the username is not known, but a DID number (i.e. a PSTN number) is known, the routing controller may cause a search through the DID bank table records of the type shown in Figure **13**, for example to find a username associated with a DID number. If the username is not known but a name and

AMENDED SHEET

What is claimed is:

1. A method for intercepting communications in an Internet Protocol (IP) network system in which communications between a subscriber of said system and another party occur through a media relay to which said subscriber and said another party address their communications destined for each other and which relays said communications between said subscriber and said another party, the method comprising:

determining whether determination information associated with a subscriber dialing profile associated with said subscriber meets intercept criteria;

when said determination information meets said intercept criteria, causing the same media relay through which communications between said subscriber and said another party are relayed to produce a copy of said communications between said subscriber and said another party, while said same media relay relays communications between said subscriber and said another party; and

causing said same media relay to send said copy to a mediation device identified by destination information associated with said subscriber dialing profile.

2. The method of claim 1 further comprising associating said determination information and said destination information with said dialing profile when communications involving said subscriber are not in progress.

AMENDED SHEET

T-Mobile Exhibit 1002
Part 3 of 6

3. The method of claim **1** further comprising associating said determination information and said destination information with said subscriber dialing profile when communications involving said subscriber are in progress.

4. The method of claim **2** or **3** wherein associating said determination information and said destination information comprises populating intercept information fields in said dialing profile of a subscriber whose communications are to be monitored.

5. The method of claim **1** further comprising producing a routing message for routing communications involving the subscriber through components of the IP network and determining whether said determination information meets said intercept criteria prior to producing said routing message and including at least some of said determination information and said destination information in said routing message when said determination information meets said intercept criteria.

6. The method of claim **5** wherein determining whether said determination information meets said intercept criteria comprises determining whether a current date and time is within a range specified by said determination information.

7. The method of claim **6** wherein producing a routing message comprises identifying a media relay through which communications involving said subscriber will be conducted and including an identification of said media relay in said routing message such that said media relay acts as said same media relay through which communications between said subscriber and said another party are relayed.

AMENDED SHEET

8. The method of claim **7** further comprising pre-associating at least one media relay with said dialing profile associated with the subscriber whose communications are to be monitored and wherein identifying said media relay comprises identifying the media relay pre-associated with said subscriber whose communications are to be monitored.

9. The method of claim **8** wherein pre-associating comprises populating media relay fields in said dialing profile with an identification of said at least one media relay.

10. The method of claim **3** wherein associating said determination information and said destination information comprises associating said determination information and said destination information with said dialing profile of the subscriber whose communications are to be monitored, in response to receipt of an intercept request message, wherein said intercept request message comprises said determination information and said destination information.

11. The method of claim **10** further comprising invoking an intercept request message handler to:

a) find a dialing profile associated with the subscriber whose communications are to be monitored;

b) perform the step of associating said determination information and said destination information with said dialing profile;

c) determine whether said intercept criteria are met; and

d) identify a media relay through which said communications are being conducted such that said media relay can be caused to send said copy to said mediation device.

AMENDED SHEET

12. The method of claim **11** wherein said dialing profile includes a username identifier and further comprising maintaining active call records for communications in progress, said active call records comprising a username identifier and a media relay identifier identifying the media relay through which said communications are being conducted and wherein identifying the media relay comprises locating an active call record associated with communications of the subscriber whose communication are to be monitored to identify the media relay associated with said communications.

13. The method of claim **12** further comprising maintaining direct-in-dial (DID) records associating PST telephone numbers with usernames of users subscribing to said IP network, and wherein finding a dialing profile associated with the subscriber whose communications are to be monitored comprises finding a username in a DID record bearing a PSTN number associated with the subscriber whose communications are to be monitored and using said username to locate a dialing profile associated with said username.

14. An apparatus for intercepting communications in an Internet Protocol (IP) network, the apparatus comprising:

means for accessing dialing profiles associated with respective subscribers of the IP network, at least one of said dialing profiles being associated with a subscriber whose communications are to be monitored, the dialing profile of the subscriber whose communications are to be monitored including intercept information including determination information for determining whether to intercept a communication involving said subscriber, and destination information identifying a mediation device to which intercepted communications involving said subscriber are to be sent;

AMENDED SHEET

means for determining whether said determination information meets intercept criteria;

means for causing the same media relay through which communications between said subscriber and said another party are relayed to produce a copy of said communications between said subscriber and said another party, while said media relay relays communications between said subscriber and said another party;

means for communicating with said same media relay to cause said same media relay to send said copy of said communications to a mediation device specified by said destination information, when said determination information meets said intercept criteria.

15. The apparatus of claim **14** further comprising means for associating said intercept information with said dialing profile when communications involving said subscriber are not in progress.

16. The apparatus of claim **14** further comprising means for associating said intercept information with said dialing profile when communications involving said subscriber are in progress.

17. The apparatus of claim **15** or **16** wherein said means for associating said intercept information is operably configured to populate intercept information fields in said dialing profile of the subscriber whose communications are to be monitored.

18. The apparatus of claim **14** further comprising means for producing a routing message for routing communications involving the subscriber through components of the IP network and means for determining

AMENDED SHEET

whether said determination information meets said intercept criteria prior to producing said routing message and wherein said means for producing said routing message is operably configured to include at least some of said intercept information in said routing message when said determination information meets said intercept criteria.

19. The apparatus of claim **18** wherein said means for determining whether said determination information meets said intercept criteria is operably configured to determine whether a current date and time is within a range specified by said determination information.

20. The apparatus of claim **19** wherein said means for producing said routing message is operably configured to identify a media relay through which communications involving said subscriber will be conducted and to include an identification of said media relay in said routing message such that said media relay acts as said same media relay through which communications between said subscriber and said another party are relayed.

21. The apparatus of claim **20** further comprising means for pre-associating at least one media relay with said dialing profile of the subscriber whose communications are to be monitored and wherein said routing means is operably configured to identify from said dialing profile the media relay pre-associated with said subscriber whose communications are to be monitored.

22. The apparatus of claim **21** wherein said means for pre-associating is operably configured to populate media relay fields in said dialing profile with an identification of at said least one media relay.

23. The apparatus of claim **16** wherein said means for associating said intercept information is operably configured to associate said intercept information associated with said dialing profile of the subscriber whose

AMENDED SHEET

communications are to be monitored, in response to receipt of an intercept request message, wherein said intercept request message comprises said intercept information.

24. The apparatus of claim **23** further comprising means for handling an intercept request message, said means for handling said intercept request message comprising:

a) means for finding a dialing profile associated with the subscriber whose communications are to be monitored, said means for finding a dialing profile cooperating with said means for associating said intercept information with said dialing profile to cause said intercept information to be associated with said dialing profile;

b) means for determining whether said intercept criteria are met; and

c) means for identifying a media relay through which said communications are being conducted such that said media relay can be caused to send said copy to said mediation device.

25. The apparatus of claim **24** wherein said dialing profile includes a username identifier and further comprising means for maintaining active call records for communications in progress, said active call records comprising a username identifier and a media relay identifier identifying a media relay through which said communications are being conducted and wherein said means for identifying the media relay is operably configured to locate an active call record associated with communications of the subscriber whose communications are to be monitored to identify the media relay associated with said communications.

AMENDED SHEET

26. The apparatus of claim **25** further comprising means for maintaining direct-in-dial (DID) records associating PST telephone numbers with usernames of users subscribing to said IP network, and wherein said means for finding a dialing profile associated with the subscriber whose communications are to be monitored is operably configured to find a username in a DID record bearing a PSTN number associated with the subscriber whose communications are to be monitored and use said username to locate a dialing profile associated with said username.

AMENDED SHEET

T-Mobile Exhibit 1002
Part 3 of 6

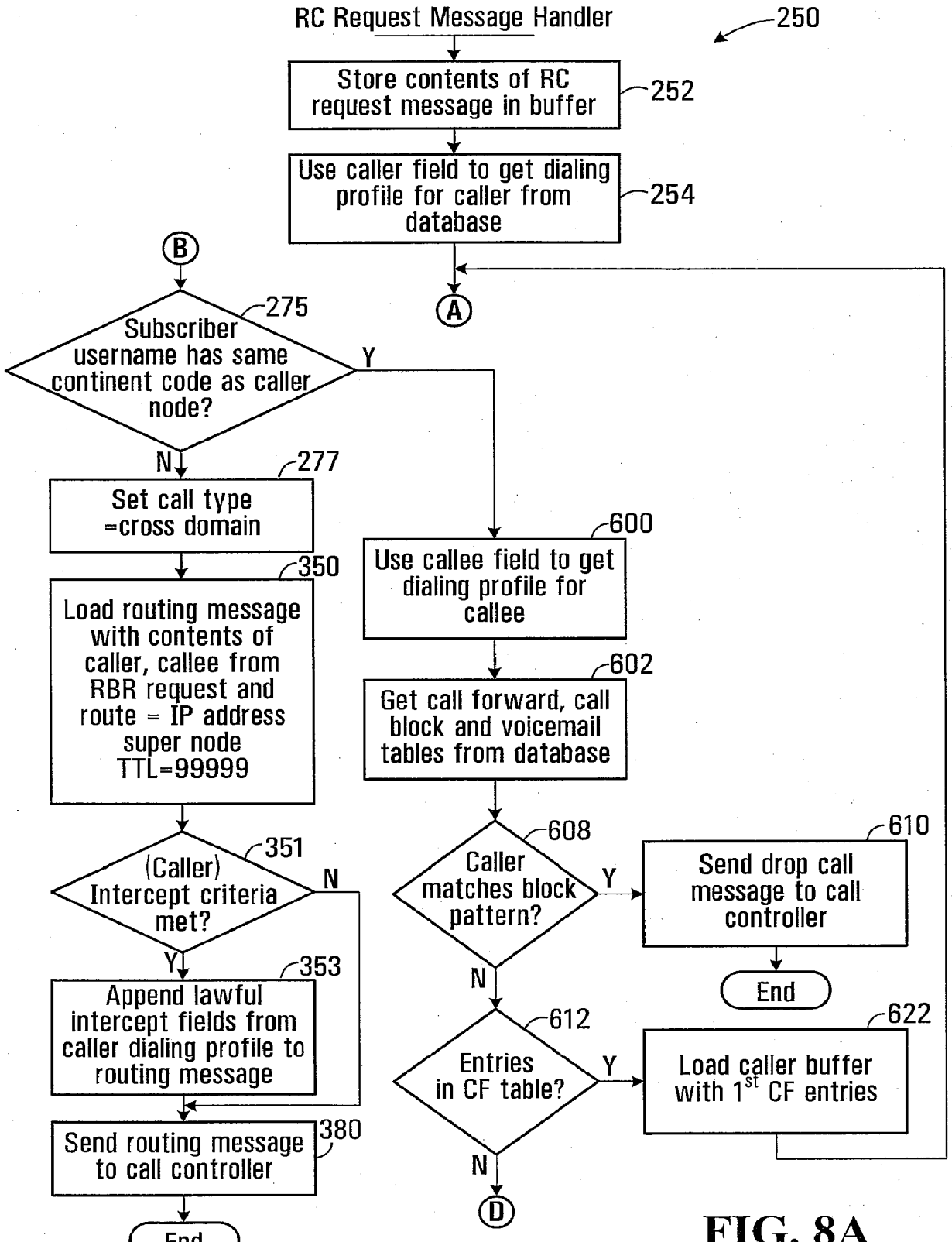


FIG. 8A

AMENDED SHEET

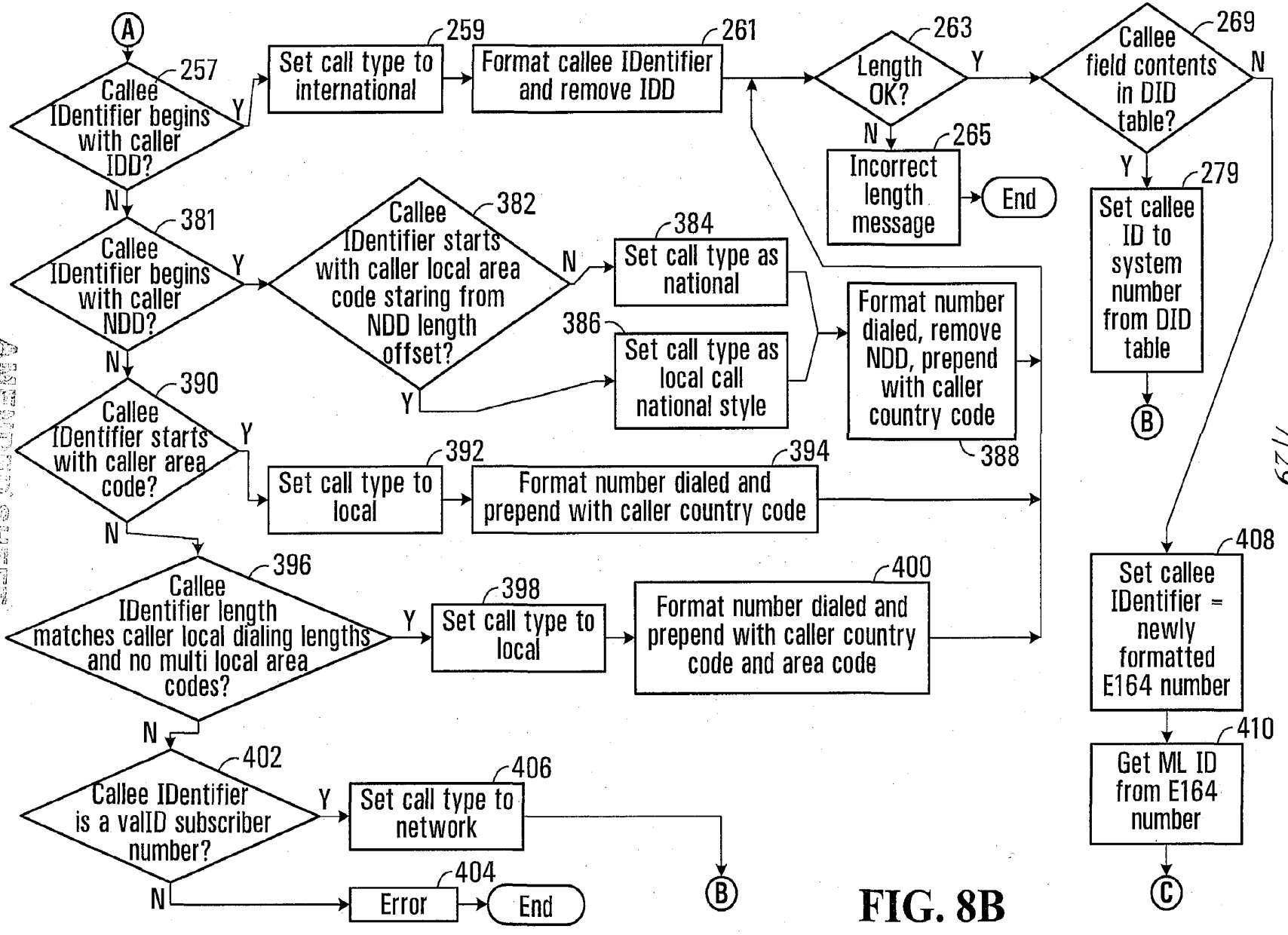


FIG. 8B

AMENDED SHEET

7/29

PCT/CA 2007/002150
22 JANUARY 2009 22.01.09

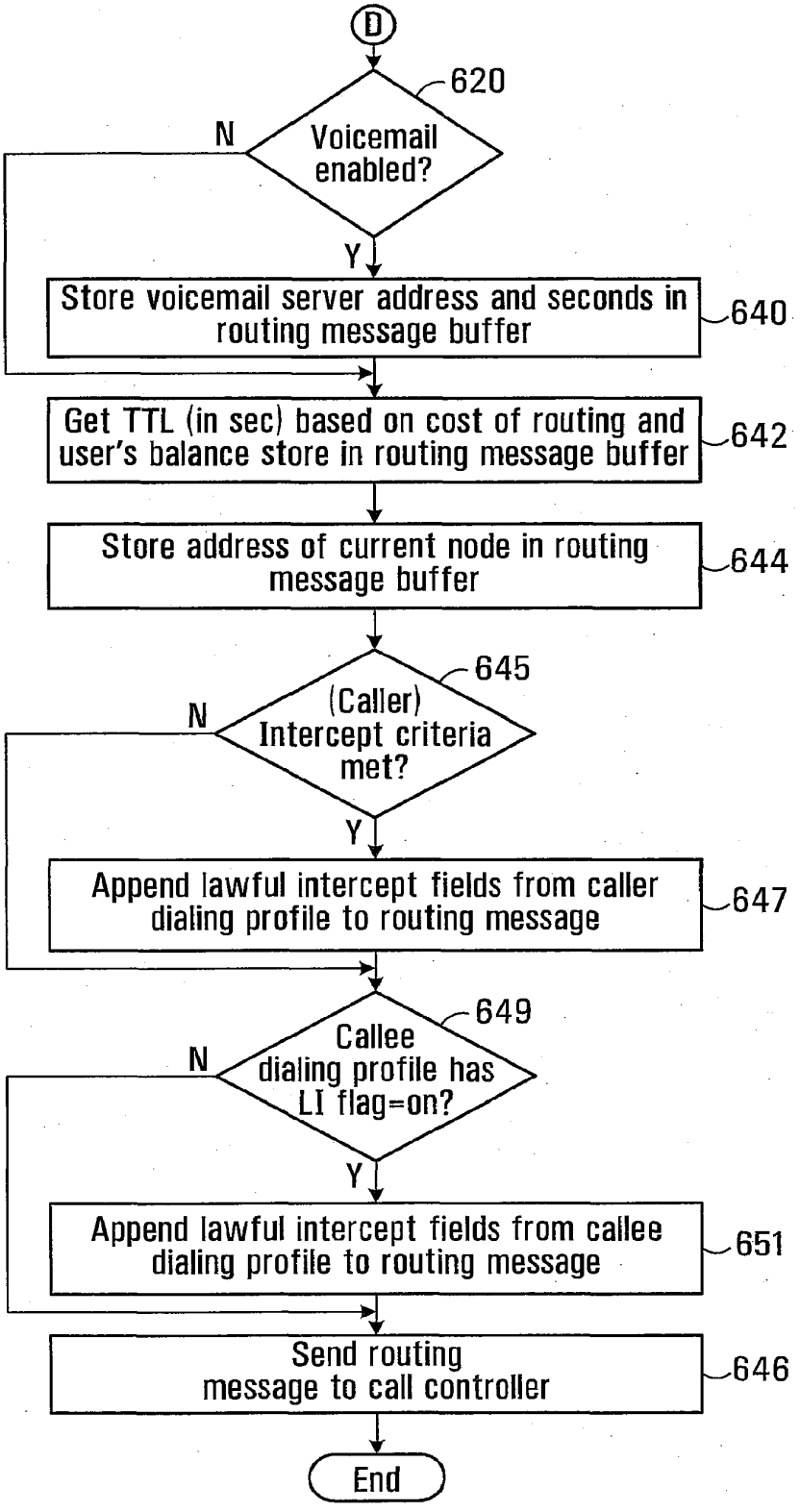


FIG. 8C

AMENDED SHEET

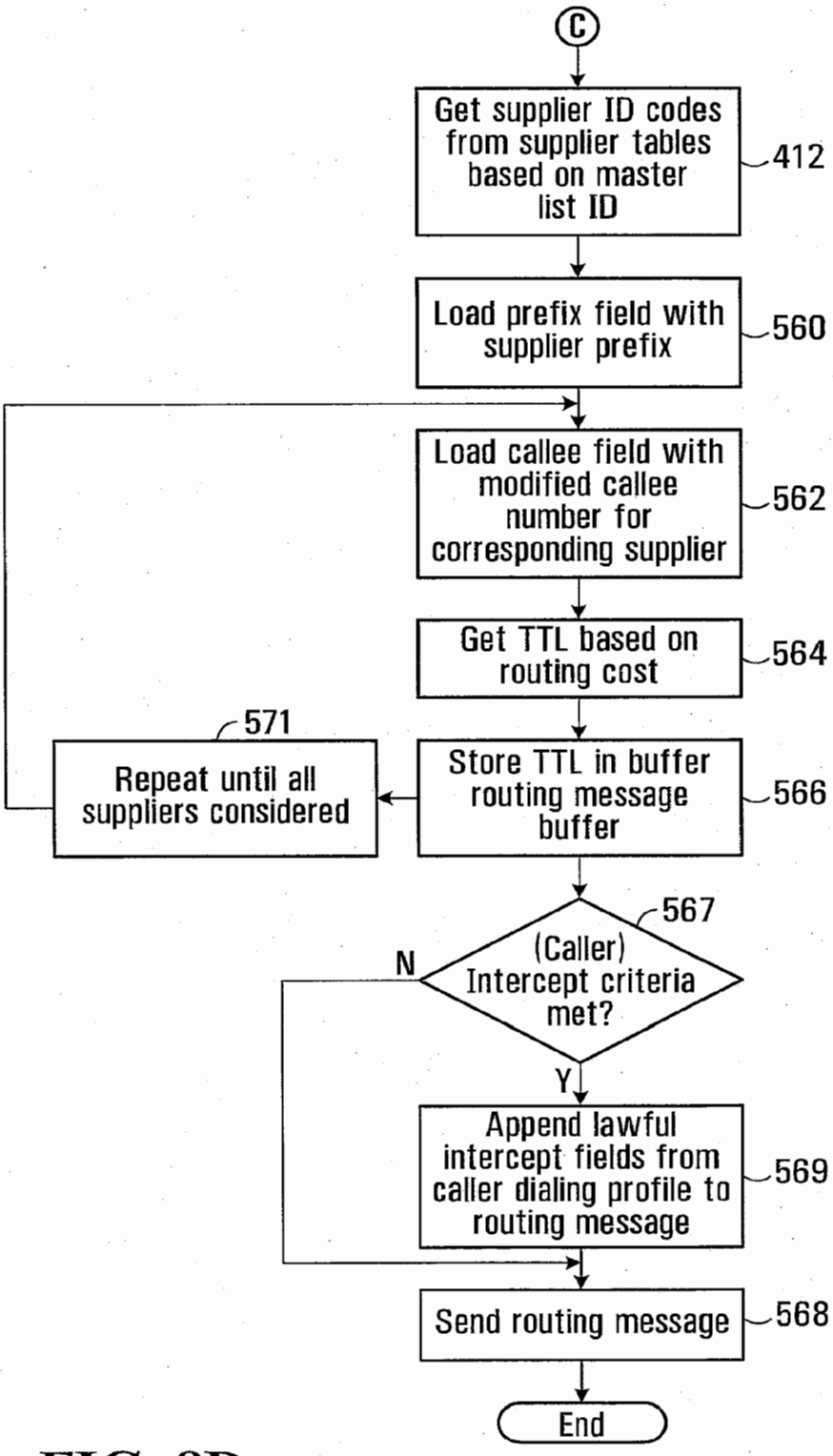


FIG. 8D

AMENDED SHEET

352

Routing Message Format

354	Supplier Prefix (optional)	Code identifying supplier traffic
356	Delimiter	Symbol separating fields
358	Callee	PSTN compatible number or Digifonica number
360	Route	Domain name and IP address
362	Time to Live(TTL)	In seconds
364	Other	TBD

FIG. 15

366

Routing Message - Different Node

440110624444@sp.lhr.digifonica.com;ttl=9999

358 360 362

1152 Media Relays (optional)

FIG. 16

Routing Message - Different Node with lawful intercept fields

440110624444@sp.lhr.digifonica.com;ttl=999;LIflag=on;MDaddress=192.168.1.10;
WarrantID=20060515142;
LIstart=2006 05 16 00:00:00
LIstop=2006 12 31 23:59:59;
1152 Media Relays (optional)

FIG. 16A

370

Prefix to Supernode Table Record Format

372	Prefix	First n digits of callee identifier
374	Supernode Address	IP address or fully qualified domain name

FIG. 17

Prefix to Supernode Table Record for London Subscriber

Prefix	4
Supernode Address	sp.lhr.digifonica.com

FIG. 18

AMENDED SHEET

Suppliers List Record Format

540	Sup_ID	Name code
542	Route_ID	Numeric code
544	Prefix (optional)	String identifying supplier's traffic #
546	Route	IP address
548	NDD/IDD rewrite	
550	Rate	Cost per second to Digifonica to use this route

FIG. 21

Telus Supplier Record

	Sup_ID	2010 (Telus)
	Route_ID	1019
	Prefix (optional)	4973#
546	Route	72.64.39.58
	NDD/IDD rewrite	011
550	Rate	$0.02/min

FIG. 22

Shaw Supplier Record

	Sup_ID	2011 (Shaw)
	Route_ID	1019
	Prefix (optional)	4974#
	Route	73.65.40.59
	NDD/IDD rewrite	011
550	Rate	$0.025/min

FIG. 23

Sprint Supplier Record

	Sup_ID	2012 (Sprint)
	Route_ID	1019
	Prefix (optional)	4975#
	Route	74.66.41.60
	NDD/IDD rewrite	011
550	Rate	$0.03/min

FIG. 24

AMENDED SHEET

Routing Message Buffer for Gateway Call

4973#0116048675309@72.64.39.58;ttl=3600 ~ 570
4974#0116048675309@73.65.40.59;ttl=3600 ~ 572
4975#0116048675309@74.66.41.60;ttl=3600 ~ 574
Media Relays (optional) ~ 1152

FIG. 25

Routing Message Buffer for Gateway Call with Lawful Intercept Fields

4973#0116048675309@72.64.39.58;ttl=3600
4974#0116048675309@73.65.40.59;ttl=3600
4975#0116048675309@74.66.41.60;ttl=3600
LIflag=on;MDaddress=192.168.1.10;WarrantID=20060515142;
LIstart=2006051600:00:00;LIstop=2006123123:59:59
Media Relays (optional) ~ 1152

FIG. 25A

Call Block Record Format

604 ~	**Username**	Digifonica #
606 ~	**Block Pattern**	PSTN compatible or Digifonica #

FIG. 26

Call Block Record for Calgary Callee

604 ~	**Username of Callee**	2001 1050 2222
606 ~	**Block Pattern**	2001 1050 8664

FIG. 27

Call Forwarding Record Format for Callee

614 ~	**Username of Callee**	Digifonica #
616 ~	**Destination Number**	Digifonica #
618 ~	**Sequence Number**	Integer indicating order to try this

FIG. 28

AMENDED SHEET

Routing Message Buffer for CF/VM Routing Message

650 ~ 200110502222@sp.yvr.digifonica.com;ttl=3600
652 ~ 200110552223@sp.yvr.digifonica.com;ttl=3600
654 ~ vm.yvr.digifonica.com;20;ttl=60
656 ~ sp.yvr.digifonica.com
1152 ~ Media Relays (optional)

FIG. 32

Routing Message Buffer for CF/VM Routing Message with Caller Lawful Intercept Fields

200110502222@sp.yvr.digifonica.com;ttl=3600
200110552223@sp.yvr.digifonica.com;ttl=3600
vm.yvr.digifonica.com;20;ttl=60
sp.yvr.digifonica.com
LIflag=on;MDaddress=192.168.1.10;WarrantID=20060615142;
LIstart=2006061500:00:00;LIstop=2006123123:59:59
Media Relays (optional) ~ 1152

FIG. 32A

Routing Message Buffer for CF/VM Routing Message with Caller and Callee Lawful Intercept Fields

200110502222@sp.yvr.digifonica.com;ttl=3600
200110552223@sp.yvr.digifonica.com;ttl=3600
vm.yvr.digifonica.com;20;ttl=60
sp.yvr.digifonica.com
LI1flag=on;Mdaddress=192.168.1.10;WarrantID=20060515142;
LI1start=2006051600:00:00;LI1stop=2006123123:59:59
LI2flag=0;MD2address=192.168.1.20;WarrantID=20060615142;
LI2start=2006061500:00:00;LI2stop=2006123123:59:59
Media Relays (optional) ~ 1152

FIG. 32B

AMENDED SHEET

900

SIP Bye Message

902	**Caller**	Username
904	**Callee**	PSTN compatible # or Username
906	**Call ID**	unique call identifier (hexadecimal string@IP))

FIG. 39

908

SIP Bye Message

902	**Caller**	2001 1050 8667
904	**Callee**	2001 1050 2222
906	**Call ID**	FA10@192.168.0.20

FIG. 40

AMENDED SHEET

1000

RC Call Stop Message

1002	**Caller**	Username
1004	**Callee**	PSTN compatible # or Username
1006	**Call ID**	unique call identifier (hexadecimal string@IP)
1008	**Acct Start Time**	start time of call
1010	**Acct Stop Time**	time the call ended
1012	**Acct Session Time**	start time-stop time (in seconds)
1014	**Route**	IP address for gateway, where a gateway is used

FIG. 42

1021

RC Call Stop Message for Calgary Callee

1002	**Caller**	2001 1050 8667
1004	**Callee**	2001 1050 2222
1006	**Call ID**	FA10@192.168.0.20
1008	**Acct Start Time**	2006-12-30 12:12:12
1010	**Acct Stop Time**	2006-12-30 12:12:14
1012	**Acct Session Time**	2
1014	**Route**	(72.64.39.58 if Telus gateway is used)

FIG. 43

AMENDED SHEET

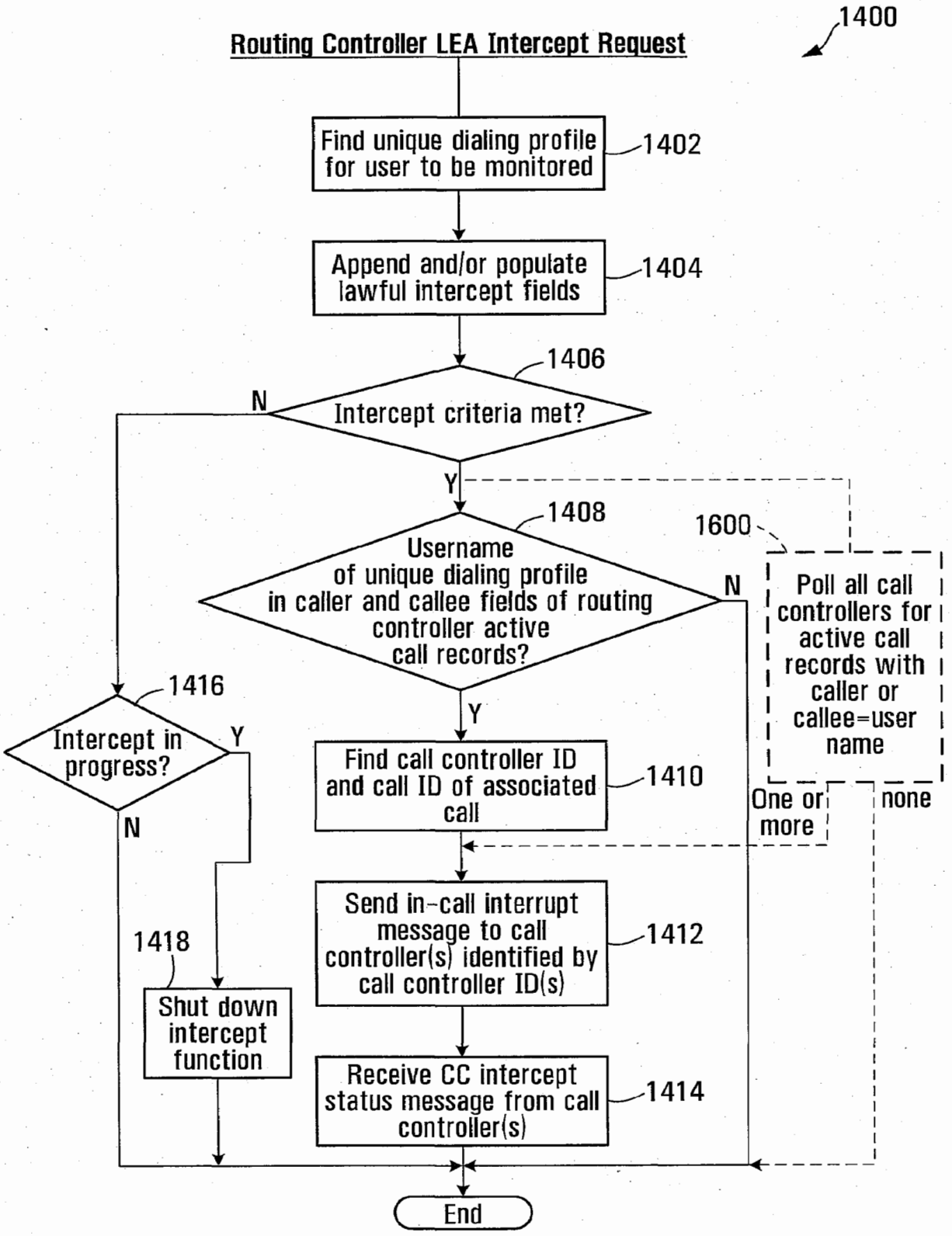


FIG. 44

AMENDED SHEET

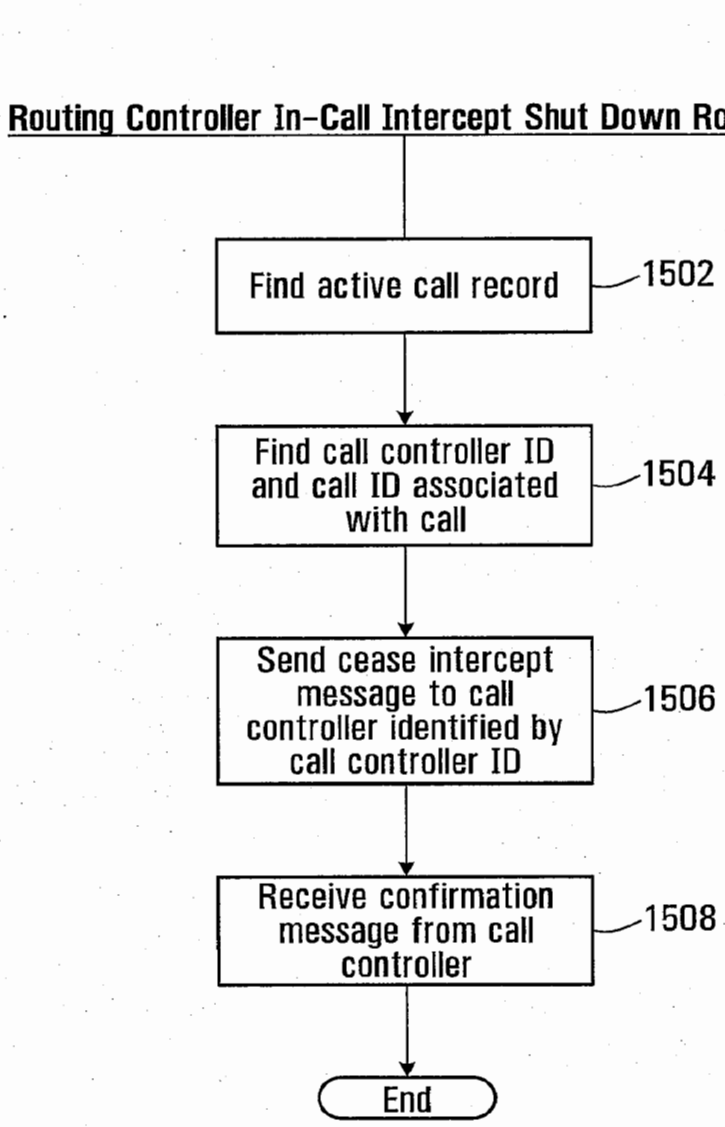


FIG. 46

AMENDED SHEET

PATENT COOPERATION TREATY

From the INTERNATIONAL BUREAU

PCT

NOTIFICATION CONCERNING TRANSMITTAL OF COPY OF INTERNATIONAL PRELIMINARY REPORT ON PATENTABILITY (CHAPTER I OF THE PATENT COOPERATION TREATY)
(PCT Rule 44bis.1(c))

To:

RECEIVED
2009 MAY 26 A 10:00
[illegible] WEST GEORGIA ST.
VANCOUVER, B.C

KNOX, John, W.
SMART & BIGGAR
Box 11560, Vancouver Centre
650 West Georgia Street
Suite 2200
Vancouver, British Columbia V6B 4N8
CANADA

Date of mailing *(day/month/year)*	
14 May 2009 (14.05.2009)	
Applicant's or agent's file reference	**IMPORTANT NOTICE**
83636-16	

International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/CA2007/001956	01 November 2007 (01.11.2007)	02 November 2006 (02.11.2006)

Applicant
DIGIFONICA (INTERNATIONAL) LIMITED et al

The International Bureau transmits herewith a copy of the international preliminary report on patentability (Chapter I of the Patent Cooperation Treaty)

The International Bureau of WIPO 34, chemin des Colombettes 1211 Geneva 20, Switzerland	Authorized officer Athina Nickitas-Etienne
Facsimile No. +41 22 338 82 70	e-mail: pt04.pct@wipo.int

Form PCT/IB/326 (January 2004)

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY REPORT ON PATENTABILITY

(Chapter I of the Patent Cooperation Treaty)

(PCT Rule 44*bis*)

RECEIVED
2009 MAY 26 A 10:00
VANCOUVER, B.C

Applicant's or agent's file reference 83636-16	FOR FURTHER ACTION	See item 4 below
International application No. PCT/CA2007/001956	International filing date *(day/month/year)* 01 November 2007 (01.11.2007)	Priority date *(day/month/year)* 02 November 2006 (02.11.2006)
International Patent Classification (8th edition unless older edition indicated) See relevant information in Form PCT/ISA/237		
Applicant DIGIFONICA (INTERNATIONAL) LIMITED		

1. This international preliminary report on patentability (Chapter I) is issued by the International Bureau on behalf of the International Searching Authority under Rule 44 *bis*.1(a).

2. This REPORT consists of a total of 8 sheets, including this cover sheet.

   In the attached sheets, any reference to the written opinion of the International Searching Authority should be read as a reference to the international preliminary report on patentability (Chapter I) instead.

3. This report contains indications relating to the following items:

☒	Box No. I	Basis of the report
☐	Box No. II	Priority
☐	Box No. III	Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
☒	Box No. IV	Lack of unity of invention
☒	Box No. V	Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
☐	Box No. VI	Certain documents cited
☐	Box No. VII	Certain defects in the international application
☒	Box No. VIII	Certain observations on the international application

4. The International Bureau will communicate this report to designated Offices in accordance with Rules 44*bis*.3(c) and 93*bis*.1 but not, except where the applicant makes an express request under Article 23(2), before the expiration of 30 months from the priority date (Rule 44*bis* .2).

	Date of issuance of this report 05 May 2009 (05.05.2009)
The International Bureau of WIPO 34, chemin des Colombettes 1211 Geneva 20, Switzerland Facsimile No. +41 22 338 82 70	Authorized officer Athina Nickitas-Etienne e-mail: pt04.pct@wipo.int

Form PCT/IB/373 (January 2004)

T-Mobile Exhibit 1002
Part 3 of 6

PATENT COOPERATION TREATY

From the
INTERNATIONAL SEARCHING AUTHORITY

To:
SMART & BIGGAR
Box 11560 Vancouver Centre
2200 - 650 W. Georgia Street
VANCOUVER, British Columbia
Canada, V6B 4N8

RECEIVED
2008 MAY 26 A 10:00
2200-650 VANCOUVER, B.C

PCT

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY
(PCT Rule 43*bis*.1)

Date of mailing *(day/month/year)*	20 February 2008 (20-02-2008)
Applicant's or agent's file reference 83636-16	**FOR FURTHER ACTION** See paragraph 2 below

International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/CA2007/001956	01 November 2007 (01-11-2007)	02 November 2006 (02-11-2006)

International Patent Classification (IPC) or both national classification and IPC
IPC: ***H04L 12/66*** (2006.01), ***H04L 12/14*** (2006.01), ***H04M 11/06*** (2006.01), ***H04M 15/00*** (2006.01), ***H04Q 3/64*** (2006.01)

Applicant
DIGIFONICA (INTERNATIONAL) LIMITED ET AL

1. This opinion contains indications relating to the following items :

[X]	Box No. I	Basis of the opinion
[ ]	Box No. II	Priority
[ ]	Box No. III	Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
[X]	Box No. IV	Lack of unity of invention
[X]	Box No. V	Reasoned statement under Rule 43*bis*.1(a)(i) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
[ ]	Box No. VI	Certain documents cited
[ ]	Box No. VII	Certain defects in the international application
[X]	Box No. VIII	Certain observations on the international application

2. **FURTHER ACTION**
If a demand for international preliminary examination is made, this opinion will be considered to be a written opinion of the International Preliminary Examining Authority ("IPEA") except that this does not apply where the applicant chooses an Authority other than this one to be the IPEA and the chosen IPEA has notified the International Bureau under Rule 66.1*bis*(b) that written opinions of this International Searching Authority will not be so considered.

If this opinion is, as provided above, considered to be a written opinion of the IPEA, the applicant is invited to submit to the IPEA a written reply together, where appropriate, with amendments, before the expiration of 3 months from the date of mailing of Form PCT/ISA/220 or before the expiration of 22 months from the priority date, whichever expires later.

For further options, see Form PCT/ISA/220.

3. For further details, see notes to Form PCT/ISA/220.

Name and mailing address of the ISA/CA	Date of completion of this opinion	Authorized officer
Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476	7 February 2008 (07-02-2008)	Arthur Smith 819-953-1360

Box No. I **Basis of this opinion**

1. With regard to the **language**, this opinion has been established on the basis of:

   [X] the international application in the language in which it was filed

   [ ] a translation of the international application into , which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).

2. [ ] This opinion has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43*bis*.1(a))

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application and necessary to the claimed invention, this opinion has been established on the basis of :

   a. type of material

      [ ] a sequence listing

      [ ] table(s) related to the sequence listing

   b. format of material

      [ ] on paper

      [ ] in electronic form

   c. time of filing/furnishing

      [ ] contained in the international application as filed.

      [ ] filed together with the international application in electronic form

      [ ] furnished subsequently to this Authority for the purposes of search.

4. [ ] In addition, in the case that more than one version or copy of a sequence listing and/or table(s) relating thereto has been filed or furnished, the required statements that the information in the subsequent or additional copies is identical to that in the application as filed or does not go beyond the application as filed, as appropriate, were furnished.

5. Additional comments :

Box No. IV **Lack of unity of invention**

1. [X] In response to the invitation (Form PCT/ISA/206) to pay additional fees the applicant has, within the applicable time limit :

   [X] paid additional fees

   [ ] paid additional fees under protest and, where applicable, the protest fee

   [ ] paid additional fees under protest but the applicable protest fee was not paid

   [ ] not paid additional fees

2. [ ] This Authority found that the requirement of unity of invention is not complied with and chose not to invite the applicant to pay additional fees.

3. This Authority considers that the requirement of unity of invention in accordance with Rules 13.1, 13.2 and 13.3 is

   [ ] complied with

   [X] not complied with for the following reasons :

   This International Searching Authority considers that there are four inventions claimed in the international application covered by the claims indicated below:

   I Claims 1-59
   II Claims 60, 61
   III Claims 62-84
   IV Claims 85-107

   The claims of Group I have in common a call routing controller for facilitating communications between callers and callees in a communications system comprising a plurality of nodes in which, in response to initiation of a call, uses call classification criteria to classify the call as a public network call or a private network call, and produces accordingly a routing message.

   The claims of Group II have in common a data structure for access by an apparatus for producing a routing message for use by a call routing controller in a communications system.

   The claims of Group III have in common determining a time to permit a communication session to be conducted, the determination based on calculating a cost per unit time, a participant's billing pattern, and the quotient of a funds balance held by the participant.

   The claims of Group IV have in common attributing charges for communications services by determining chargeable times and changing account balances of both user and communications services reseller.

   Groups I and II have in common the call routing controller; however, call routing controllers are well known in the art so the claims of Groups I and II lack unity *a posteriori*.

   Because the remainder of the claims of Groups I, II, III, and IV have no elements in common and would require separate searches by the examiner, these groups lack unity *a priori*.

4. Consequently, this opinion has been established in respect of the following parts of the international application :

   [X] all parts

   [ ] the parts relating to claim Nos.

Box No. V **Reasoned statement under Rule 43*bis*.1(a)(i) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement**

1. Statement

Novelty (N)	Claims	1-107	YES
	Claims	None	NO
Inventive step (IS)	Claims	1-61, 64-72, 76-107	YES
	Claims	62, 63, 73-75	NO
Industrial applicability (IA)	Claims	1-107	YES
	Claims	None	NO

2. Citations and explanations :

Group I (Claims 1-59)
The following document is referred to in this communication:
D1 CA 2249668

D1 is considered to form the closest prior art. D1 discloses routing information in an integrated global communications network in which a central routing processor collects routing capabilities of network nodes for which it has responsibility. The routing processor evaluates the routing requirements of a routing query signal transmitted by a source router, determines which routers and communication paths within the network are capable and available to route the information, evaluates the statistical availability of such routers, and selects an optimal routing path to a destination router.

Novelty
D1 fails to individually disclose all the elements of claims 1-59; therefore, claims 1-59 **are considered to be novel** in accordance with Article 33(2) PCT.

Inventive Step
Independent claims 1, 30, and 31 each claim operating a call routing controller to facilitate communication between callers and callees in a system or network comprising a plurality of nodes in which call classification criteria associated with a caller identifier is used to classify the call as a public network call or a private network call, and producing a routing message in accordance with the classification. D1 teaches facilitation of communication between callers and callees within a private network, including producing a routing message for a private network call. However, D1 fails to teach classification of a call as a public network call, and fails to teach producing an appropriate routing message for a public network call.

Claims 2-29 and 32-59 depend on independent claims 1 and 31, respectively.

Therefore, claims 1-59 **are considered to have an inventive step** in accordance with Article 33(3) PCT.

Industrial Applicability
Routing IP phone calls through a communication network including both private and public aspects finds use in telecommunications, and, thus, claims 1-59 **are industrially applicable** in accordance with Article 33(4) PCT.

Group II (Claims 60, 61)
The following documents are referred to in this communication:
D2 US7,068,772
D3 US2006/0209768

D2 and D3 are considered to form the closest prior art. D2 discloses a call processing system and method for providing one-number telecommunication services, wherein a data structure of a subscriber record for access by an apparatus for producing a routing message, the data structure comprising the subscriber's various profiles and the associated 1-800 number or address that the subscriber may be contacted.

Box No. VIII Certain observations on the international application

The following observations on the clarity of the claims, description, and drawings or on the question whether the claims are fully supported by the description, are made :

Claim-Related Objections

Claim 60 is unclear and does not comply with Article 6 of the PCT. The following terms lack a proper antecedent basis:
"the subscriber" (claim 60, page 83, line 16)
"subscriber name" (claim 60, page 83, line 17)

Claim 60 is unclear and does not comply with Article 6 of the PCT. The double inclusion of any element renders the claims indefinite. The following expressions have already been defined previously in the claims and should therefore be referred to using a definite article:
"a user domain" (claim 60, page 83, lines 14, 17)
"a direct-in-dial number" (claim 60, page 83, line 18)

Claim 60 is indefinite and does not comply with Article 6 of the PCT. The terms "a subscriber user name" (claim 60, page 82, line 32) and "subscriber name" (claim 60, page 83, lines 14, 15, 16-17, 17) cause ambiguity. It is not clear whether they are the same or different.

Claim 61 is indefinite and does not comply with Article 6 of the PCT. The term "master list records" (page 83, line 27) causes ambiguity. It should read "said master list records".

Claim 61 is indefinite and does not comply with Article 6 of the PCT. The term "aid" (page 83, line 28) causes ambiguity. It should read "said aid".

Claim 61 is indefinite and does not comply with Article 6 of the PCT. The term "dialing codes" (page 84, line 8) causes ambiguity. It should read "said dialing codes".