

• Bracketed numbers in the Figures (e.g. (1)) designate explanatory notes which contain textual descriptions pertaining to various signals. The notes are given in the text relating to the relevant Figure. As it is believed that the Figures are self-explanatory, only brief comments will be made regarding the figures.

Joining of New Members

An "empty" synchronization group is one in which no members 18 are active. When the first member 18 joins, it is designated as the distinguished member by default. Figure 6 depicts a member 18 joining an empty group. The sequence of events is as depicted in Figure 6, to which attention is directed. The abbreviations used in the Figures are as follows: APPL means an application task; SNYC AGT means the synchronization agent task; MEM AGT means the member agent task; SYNC MST means the synchronization master task 20; GRP MST means the Group Master Task 25; MEM AGT (DM) means the distinguished member agent task; APPL (DM) means the application task which corresponds to the distinguished member agent.

The following notes refer to the bracketed numbers in Figure 6.

Notes:

(1) The Sync Agent will create a member agent task only if it had not existed previously, otherwise it will REINIT a previously created task instance. The START-AGT signal which follows initialization is used to pass initial data to the member agent 11.

(2) The AGT-MST-MSG signal to the sync master 13 contains the complete information about all member agents 11 on this processing element 12, including the newly-created member 18. (This ensures state convergence even in the presence of design faults.) The MST-REPLY message is used for positive acknowledgement so that the Sync Agent can send the next message to the Sync Master if it has one. (Only one outstanding message is allowed between the Sync Master and a Sync Agent.)

(3) If this group has not previously existed, a new Group Master is created. In that case a STARTUP signal follows to pass initial data to the Group Master Task, and an ACTIVATE signal is used to force it into an operational mode. If the group had existed previously (but had lost all its members) the existing Group Master Task is used.

(4) Once the Group Master has been activated, a GRP-EVENT signal is sent by the Sync Master informing it of the joining of the first member.

(5) Upon receipt of the GRP-EVENT signal, the Group Master selects the newly-created member agent 11 as the distinguished member and sends it a GRP-STATE signal. This signal establishes a connection between the Group Master and the Distinguished Member. All subsequent GRP-STATE signals are sequenced to ensure proper event ordering as well as to guard against communication failures. The GRP-REPLY signal is used to acknowledge one or more GRP-STATE signals and provides reliable communication.

The GRP-STATE signal contains the complete new state of the group rather than just information about the changes. This ensures that the system will converge to the true state even in the presence of design faults.

(6) The CHECK-FAIL signal is used to poll the application task to detect unexpected failures of the application. The application task never receives this signal; however, should the task fail, the member agent task will be notified by the underlying operating system kernel.

(7) The SYNC-REPLY signal contains a reply code of SYNC-DONE.

Figure 7 depicts the sequence for joining an existing synchronization group; the Group Master Task already exists, and the distinguished member is used to notify (broadcast) the other members 18 of the presence of a new member 18.

Notes:

(8) If the application has so requested, a GROUP-CHANGE signal is sent by each member agent 11 to the application whenever it detects a change in status of the group. In the case of the distinguished member the status change is gleaned from the GRP-STATE signal.

(9) When the distinguished member receives a GRP-STATE signal which indicates a group change (not all do), it broadcasts the new state to all other group members 18 using a MEM-MSG signal. Each member 18 acknowledges such messages with an ACK signal to provide reliable communication.

(10) When the newly-joining member 18 receives its MEM-MSG from the distinguished member it will send a SYNC-REPLY signal to the application (instead of a GROUP-CHANGE signal).

The control flow for the departure of a member 18 is shown in Figure 8. Note that the case of a processing element 12 failure is not shown here but is instead treated separately.

Notes:

(11) The sequence shown here corresponds to a voluntary departure initiated by the application program issuing an UNSYNC directive. This results in the member agent task terminating which, in turn, sends a COMPLETE signal to the parent task, the Sync Agent. The sequence is similar in situ-

ations where the departure is not voluntary:

-When the application task fails, the member agent 11 is notified (through the failure of the CHECK-FAIL message) which results in the termination of the member agent task (and consequently, raising of the COMPLETE signal).

-If the member agent task itself fails, the Sync Agent is notified by the operating system kernel with a COMPLETE signal.

(12) Indicates a <GROUP-CHANGE> signal to an application task not shown in the Figure.

Figures 9a and 9b together depict the recovery of the synchronization master control 13 (i.e. Sync Master). The most general case is considered, i.e. the case of a running synchronization service 10 with active synchronization groups. This includes, as a subset, the case of a "cold" start.

Notes:

(13) Upon activation, Sync Master sends an ACTIVATE signal to each configured Sync Agent. The Sync Agent, whether they are already active or not, will respond with an AGT-REPLY message which includes a list of all member agents supported on that processing element. (The MST-REPLY signal is used for acknowledgement: refer to Figure 6.)

(14) Following activation of the Sync Agent, the Polling Control subcomponent 22 of Sync Master Control sends a POLL-AGT message to which the Sync Agent responds with an AGT-REPLY message. This exchange is repeated periodically to detect outages of the processing element.

(15) A Group Master is initiated only the first time a group is encountered. Refer to Figure 6 for further details on Group Master initiation.

(16) Before activating a Group Master, it is provided with data regarding the status of its members through GRP-DATA signals. Each signal contains the information for one member agent of one group. (This info is obtained from the AGT-REPLY messages.) In this way, the Group Master reconstructs the status of its group.

(17) After all Sync Agents have responded, the reconstruction is complete and an ACTIVATE signal is sent to all Group Masters. The Group Masters respond by sending a GRP-STATE signal to all distinguished members. Since this signal contains the complete group state, any group changes that might have occurred while the Sync Master was down are detected.

Figure 10 depicts the procedure for handling the failure of a processing element 12 which contains a synchronization agent 14 (Sync Agent).

Notes:

(18) The Sync Master detects a failure of a processing element when a TIME_OUT event is received. This means that a Sync Agent has not responded to a poll.

(19) For each group affected by the processing element failure, the Sync Master will send a GRP-EVENT signal to the respective Group Master.

5 Figure 11 depicts the procedure for the recovery of a processing element 12 which is part of synchronization service 10. Note that the recovery of processing element 12 does not extend to recovering member agent 11 tasks. It is assumed that these will be recovered when the application tasks (i.e. programs) which use them are restarted. Thus, the only action to recover a process element 12 is to integrate the sync agent 14 with the rest of synchronization service 10.

10 Notes:

(20) When a previously failed Sync Agent finally responds to a POLL_AGT signal, the Sync Master initiates the recovery procedure.

15 (21) The Sync Master registers the new processing element 12 and sends an ACTIVATE signal to the Sync Agent on that processing element 12 (Refer to Figure 9a for a more detailed description of the activation sequence).

20 Figure 12 depicts the procedure for member 18 to member 18 messages. This procedure (protocol) is used both for group broadcasts and point-to-point messages between members 18.

25 Notes:

(22) In case of a broadcast, a copy of the message is sent to each member 18. If the member 18 is not active, the message is not sent.

(23) Upon receiving a MEM-MSG signal which indicates an application-level message the message is relayed to the application task responsible for receiving asynchronous messages.

30 (24) A SYNC-REPLY (codes: GRP-ACK or MSG-ACK) signal is sent back to the originator.

Figure 1 depicts the procedure employed in the processing of all directives which require distinguished member intervention (rights handling directives).

35 Notes:

(25) If the request is made on the distinguished member site, then no message is sent.

40 (26) A reply signal (MEM-MSG followed by a SYNC-REPLY to the application).

In one implementation made by the inventors, the code for synchronization service 10 was contained in ten files which were distributed into six units, the usage dependencies (and compilation order) of which are shown in Figure 14.

45 Notes:

SYNCTRL contains the stub of the Sync Master unit and directives to include three files (SYNCMST, SYNCGMST, and SYNCPLL) which implement the Sync Master function. It also contains the definitions required for the master database 16. SYNCMST is an "include" file which

contains the code for the Sync Master Task.

SYNCGMST is an "include" file which contains the code for the Group Master Task.

SYNCPOLL contains the Polling Control 22.

SYNCLOCL contains the stub of the Sync Agent unit as well as a definition of data and procedure objects shared by the Sync Agent Task and the Member Agent Tasks. It also contains directives to include two files (SYNCAGT, SYNCMAGT).

SYNCRESX contains the definition of the SYNCHRONIZE primitive. This unit must be loaded with the application code which uses the synchronization service.

SYNCDEFI contains a set of internal compile-object definitions for the synchronization service. This includes the definition of all entries used for communication between synchronization service components which are hidden from user programs. Since this file contains only compile objects it is not loaded.

SYNCDEFX contains a definition of all compile-objects which are exported by the synchronization service to its users. Since this file contains only compile objects it is not loaded.

Application tasks which use the synchronization service need to include SYNCRESX and SYNCDEFX in their usage lists.

Note that SYNCCTRI implements the function of synchronization master control 13 (Fig. 1), and that SYNCLOCL and SYNCRESX together implement the functions of member agent 11 and synchronization agent 14 (Fig. 1). The files SYNCDEFI and SYNCDEFX are not resident in the service; they can be thought of as tools used in the construction of the synchronization service but they are not themselves a part of it.

Simplified pseudocode listings for the main constituents of the invention follow as appendix I. They are believed to be self-explanatory. Any elaboration of the material is accomplished through the use of appended notes, to which attention is directed.

As a further aid to the understanding and to the use of the present invention the following (a copy of a "User's Reference" to the synchronization service of the present invention, as prepared by one of the inventors) is included as Appendix II. It will expand on the use of the present invention.

Claims

1. A synchronization service [10] for use with a computer having a distributed operating system, to allow the construction of a customized synchronization scheme, for synchronizing the constituent portions of a distributed program, said service comprising:

a general set of application-independent synchronization primitives, whereby the construction of said customized synchronization scheme is achieved by the selective implementation of said application-independent synchronization primitives.

2. The synchronization service of claim 1 wherein said application-independent primitives comprise the following functions: synchronize; synchronize done; and unsynchronize.

3. The synchronization service of claim 2 wherein said primitives further comprise the following functions: request right; right granted; right refused; release right; group broadcast; and group acknowledge.

4. The synchronization service of claim 3 wherein said primitives further comprise the following functions: unsynchronize done; send to member; and message acknowledge.

5. A synchronization service [10] for use with a computer having an operating system [15] distributed over a plurality of processing elements [12], to allow the construction of a customized synchronization scheme, for synchronizing the constituent portions [18] of a distributed program, said service comprising:
a common synchronization master control means [13];

a synchronization agent means [14] for each processing element;
a plurality of application program components [18], each component located on a different processing element, each said component having associated therewith a member agent [11], said member agent being a program for interfacing with said synchronization agent means, and said synchronization agent means interfacing between said master control means and said member agent, whereby a customized synchronization scheme can be constructed based upon a general set of application-independent synchronization primitives contained in both said synchronization agent means [14] and said member agent [11] and accessed via said synchronization agent means.

6. The synchronization service of claim 5 wherein said application-independent primitives comprise the following functions: synchronize; synchronize done; and unsynchronize.

7. The synchronization service of claim 6 wherein said application-independent primitives further comprise the following functions: request right; right granted; right refused; release right; group broadcast; and group acknowledge.

8. A synchronization service [10] for use with a computer having an operating system [15] distributed over a plurality of processing elements, to allow the construction of a customized synchroniza-

tion scheme, for synchronizing the constituent component of a distributed program, said service [10] comprising the steps of:

a) joining a program component [18] on a first processing element [12] to a group of existing program components [18] on at least a second processing element [12] so that each of the existing components is aware of the presence and location of the joining components;

b) informing each member of a group of physically distributed program components when one or more components which are members of said group, depart from it;

c) selecting, as a distinguished member, one program component from a group of distributed program components such that, within said group, there is never more than one said distinguished member; and

d) providing mutually exclusive rights to said group of distributed program components such that no more than one said component can appropriate a given right at any time.

9. The synchronization service of claim 8 further including the step of providing reliable point-to-point communication between said distributed program components on the basis of their internal group identifiers.

10. The synchronization service of claim 9 further including the step of providing a broadcast mechanism from any one program component to all other program components which are currently declared as being in the same group as the broadcasting component.

11. The synchronization service of claim 10 wherein said program components are components of an application program.

12. The synchronization service of claim 10 wherein said program components are components of an operating system program.

13. The synchronization service of claim 8 wherein said physical processing elements are logically distributed entities at one physical location.

14. A synchronization service [10], for use with a computer having an operating system [15] distributed over a plurality of processing elements [12], to allow the construction of customized synchronization schemes for synchronizing the constituent components [18] of a distributed program, said service comprising, as required, the steps of:

a) establishing a synchronization group for said distributed program, said group comprising at least one distributed program component [18];

b) joining a program component [18] to said group of existing program components so that each of the components is aware of the presence and the location of all the other components in said group;

c) informing each member of said group of distributed program components when one or more components which are members of said group, depart from it;

d) selecting, as a distinguished member for said group, one program component from said group of distributed program components such that, within said group, there is never more than one said distinguished member; and

e) providing mutually exclusive rights to said group of distributed program components such that no more than one said component can appropriate a given right at any time.

15. The synchronization service of claim 14 further including the step of providing full connectivity between all said distributed program components of said group.

16. The synchronization service of claim 15 wherein said distributed program is an application program.

17. The synchronization service of claim 15 wherein said distributed program is an operating system program.

18. The synchronization service of claim 15 wherein each said program component [18] is on a different processing element [12].

19. A synchronization service [10], for use with a computer having an operating system [15] distributed over a plurality of processing elements [12], to allow the construction of customized synchronization schemes for synchronizing the constituent components [18] of a distributed program, said service including a synchronization master control [13] comprising:

master control means [21] for activating said synchronization service;

polling means [22] for polling the processing elements [12] associated with said components of said distributed program so as to monitor the status of said processing element;

control means [24] for joining new members [18] to said group, and for handling departures of members [18] from said group; and

a database means [16] containing information representative of the current state of said synchronization service at a given point in time.

20. The synchronization service of claim 19 further including, at each said processing element, a synchronization agent [14] comprising:

means for accepting synchronization directives and for creating corresponding member agents; and means for monitoring the status of all active member agents on said processing element and reporting same to said synchronization master control [13].

21. The synchronization service of claim 20 further including at each said processing element [12], a member agent [11] each synchronization

group, comprising:

communications means [33] for providing a reliable communications service between program components [18];

storage means [32] for maintaining a local version of the current state of all other program components [18];

handler means [31] for providing the interface between user tasks and said member agent [11]; and distinguished member means [30] for implementing the distinguished member function on only one program component [18] at any given time.

15

20

25

30

35

40

45

50

55

19

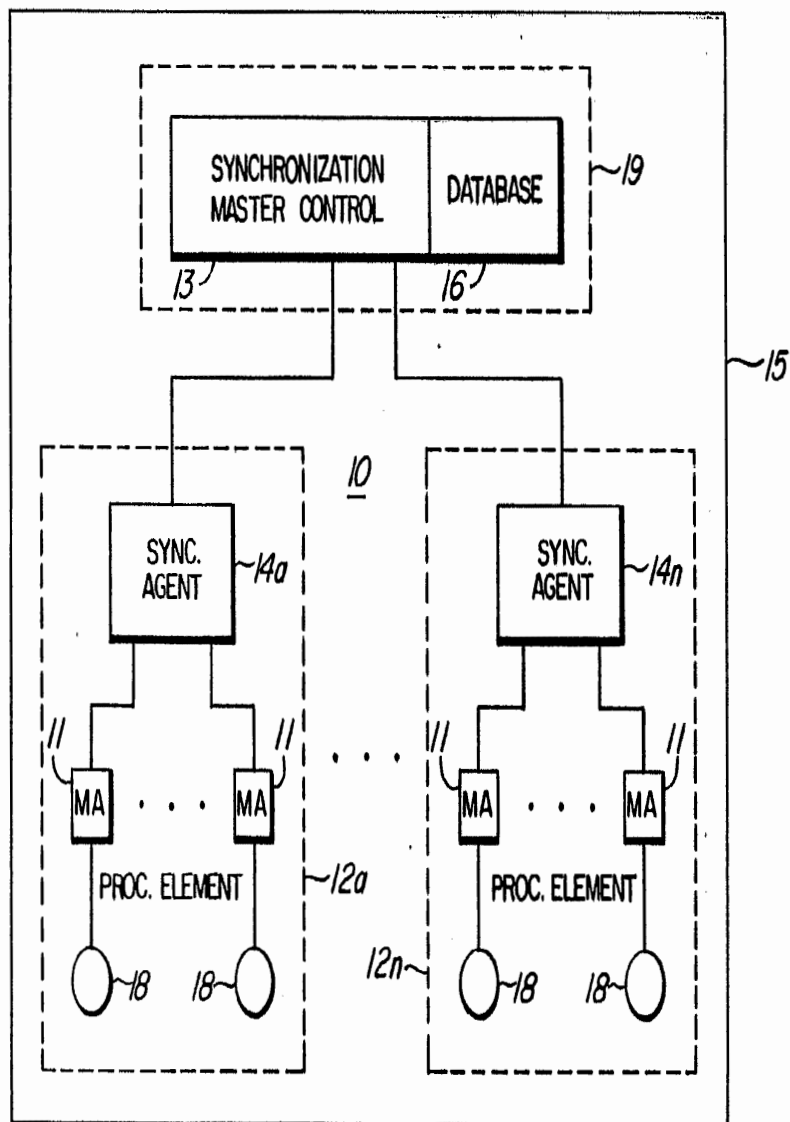


FIG. 1

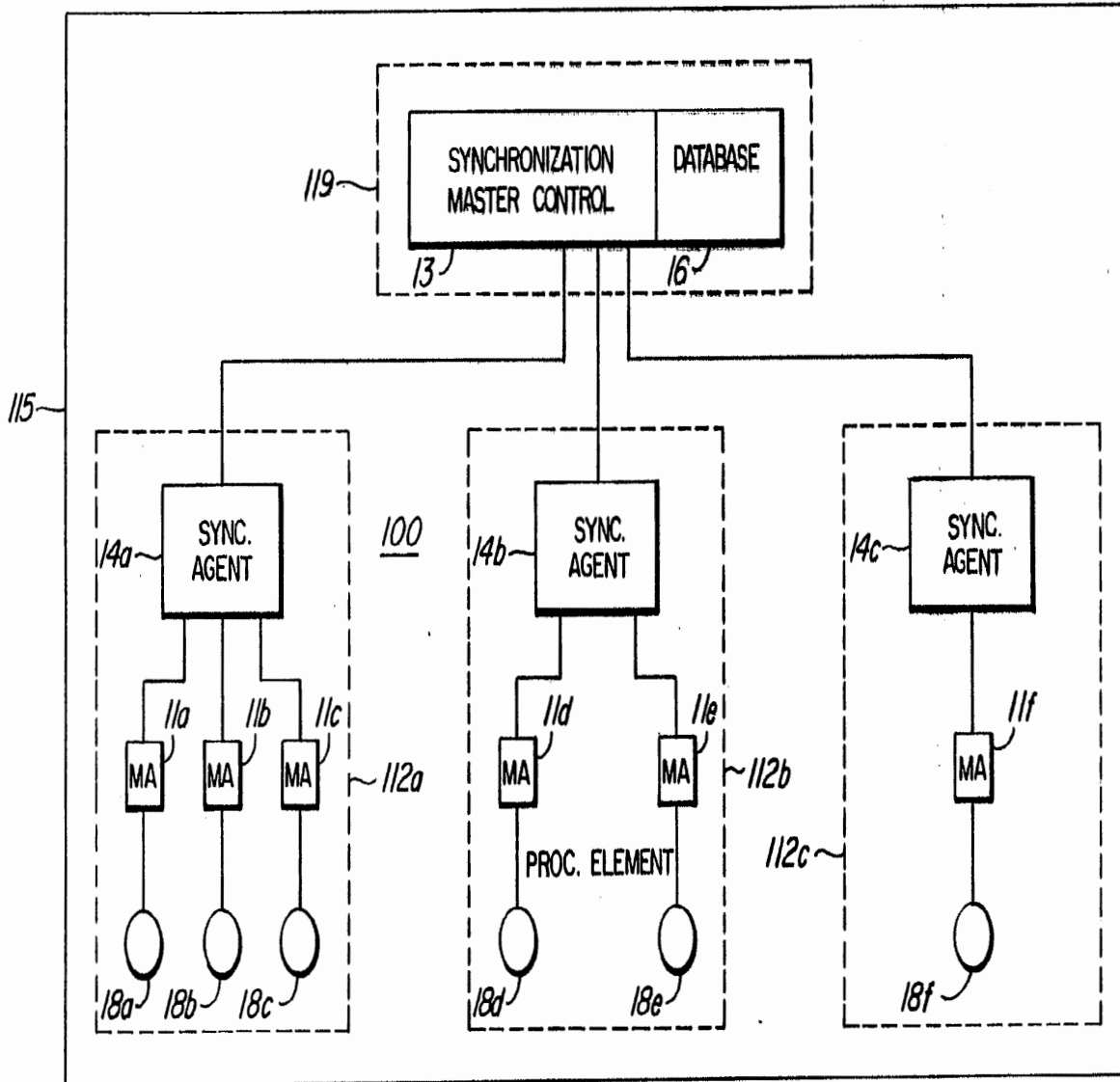


FIG. 2a

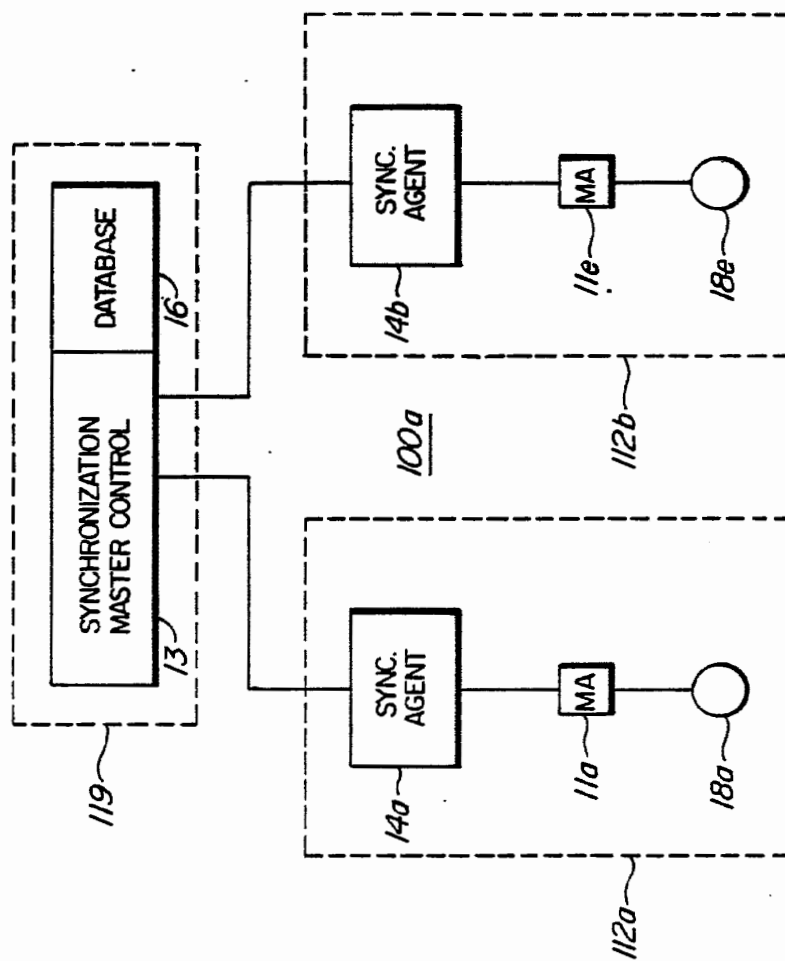


FIG. 2b

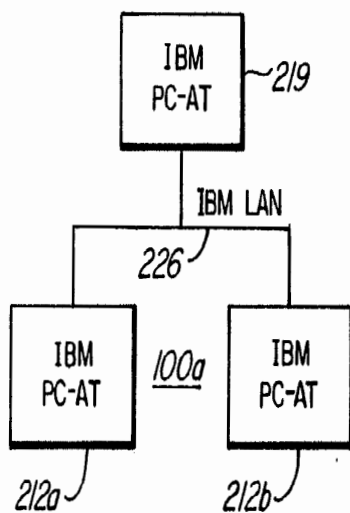


FIG. 2c

SYNCHRONOUS PRIMITIVES

REQUEST	REPLY
SYNCHRONIZE (group-id)	SYNC_DONE (group-status)
UNSYNC	UNSYNC_DONE
REQ_RIGHT (right-id, Immed)	R_GRANTED
	R_REFUSED
REQ_RIGHT (right-id, Queued)	R_GRANTED
REL_RIGHT(right-id)	R_RELEASED
QRY_RIGHTS	R_STATUS (rights-status)
GRP_BRDCST (message)	GRP_ACK
SND_TO_MEM (message)	MSG_ACK

ASYNCHRONOUS NOTIFICATIONS

GROUP-CHANGE (group-status)
GROUP-MSG (message)

FIG. 3a

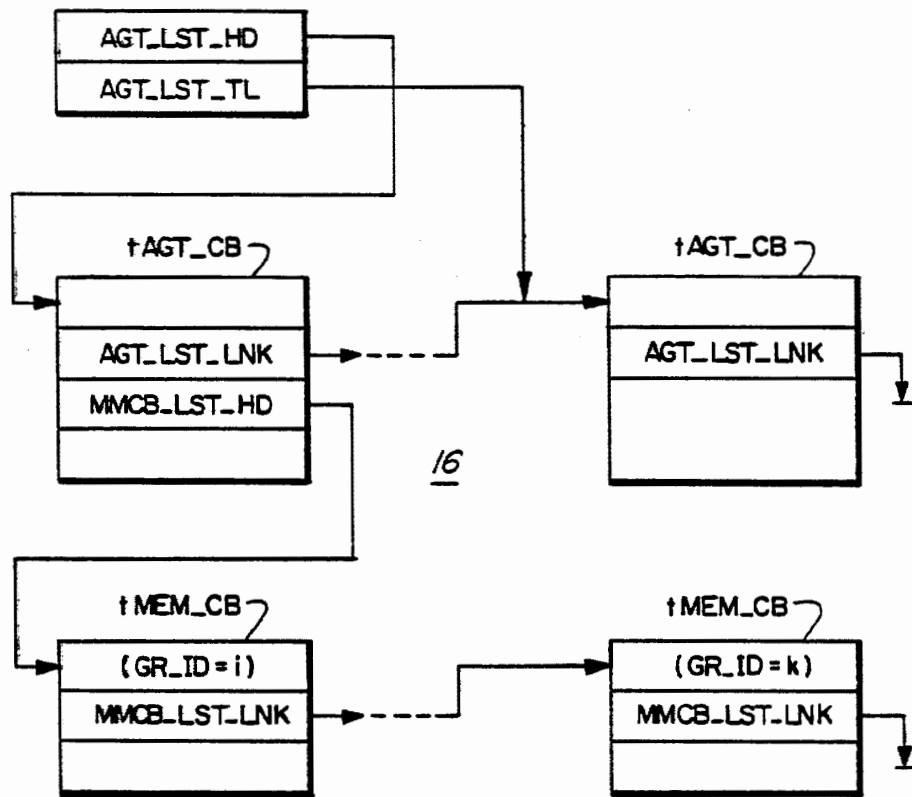


FIG. 4

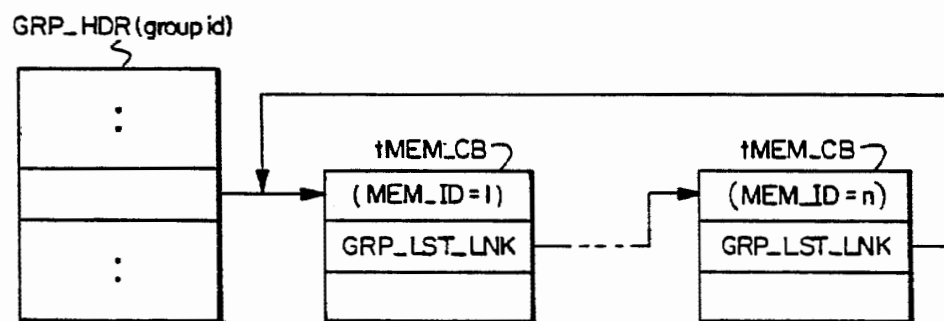


FIG. 5

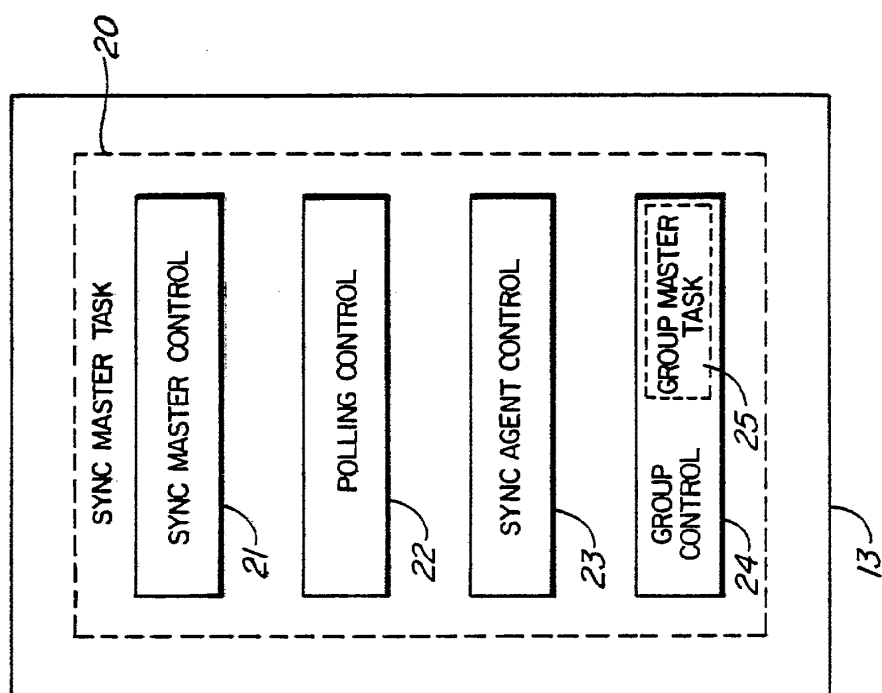


FIG. 3b

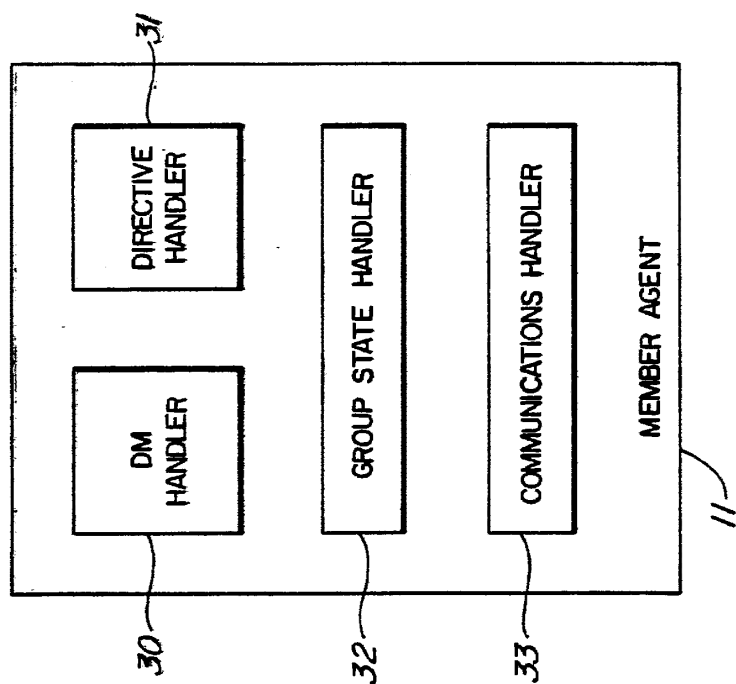
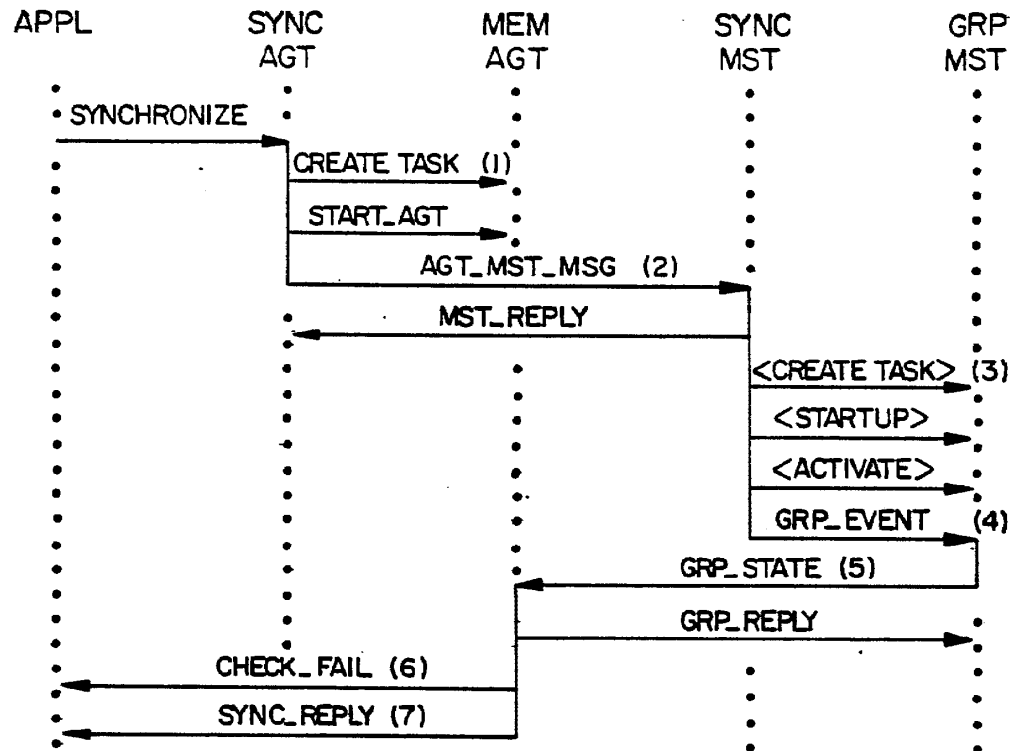


FIG. 3c



LEGEND
APPL = MEMBER 18
SYNC AGT = SYNCHRONIZATION AGENT 14
MEM AGT = MEMBER AGENT 11
SYNC MST = SYNCHRONIZATION MASTER 13
GRP MST = GROUP MASTER 25

FIG. 6

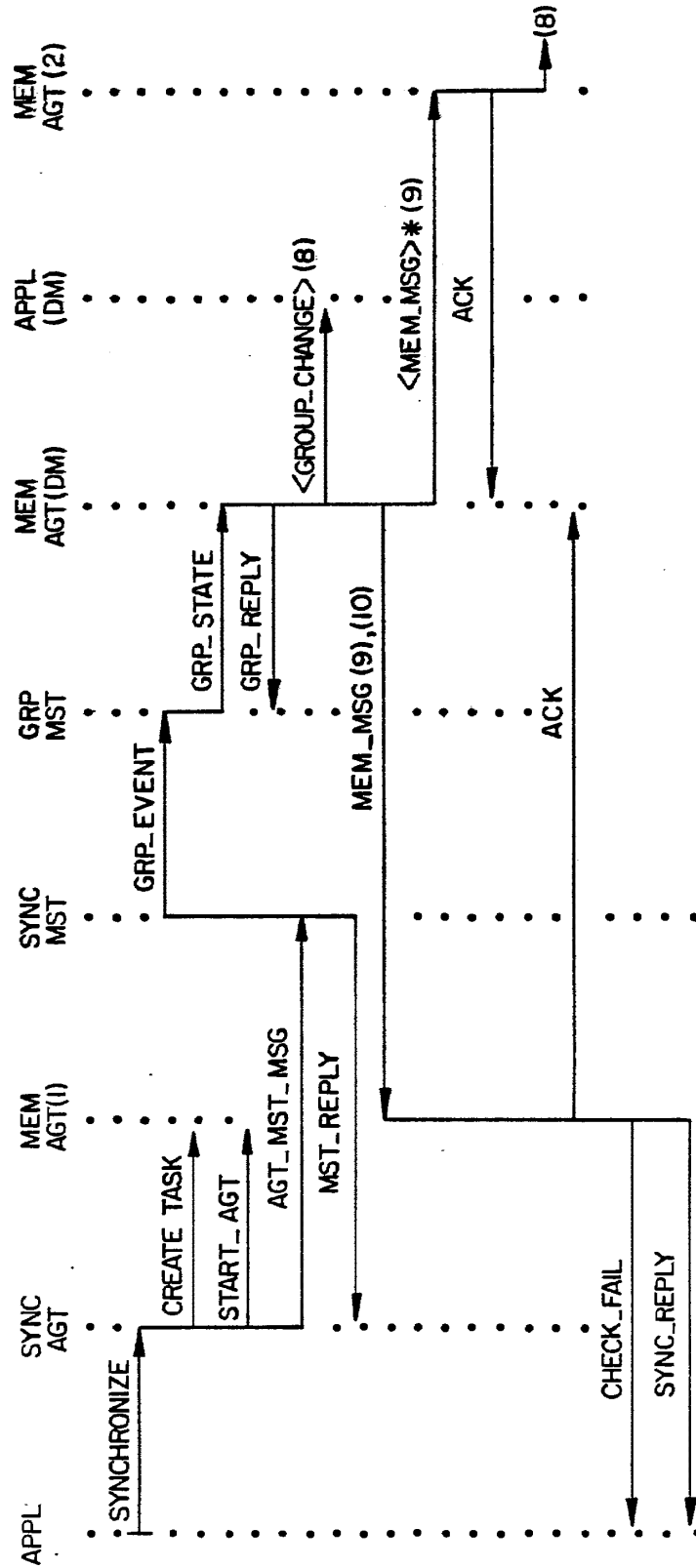


FIG. 7

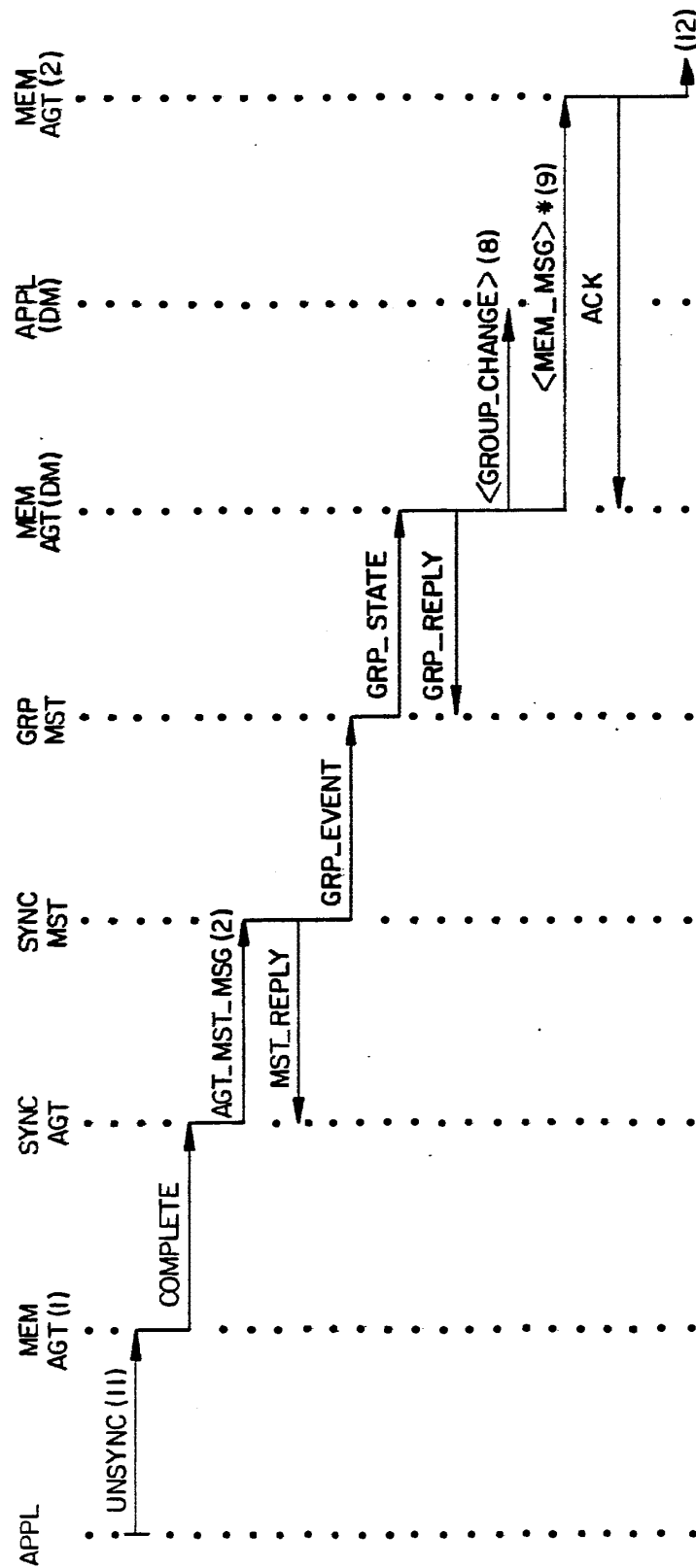


FIG. 8

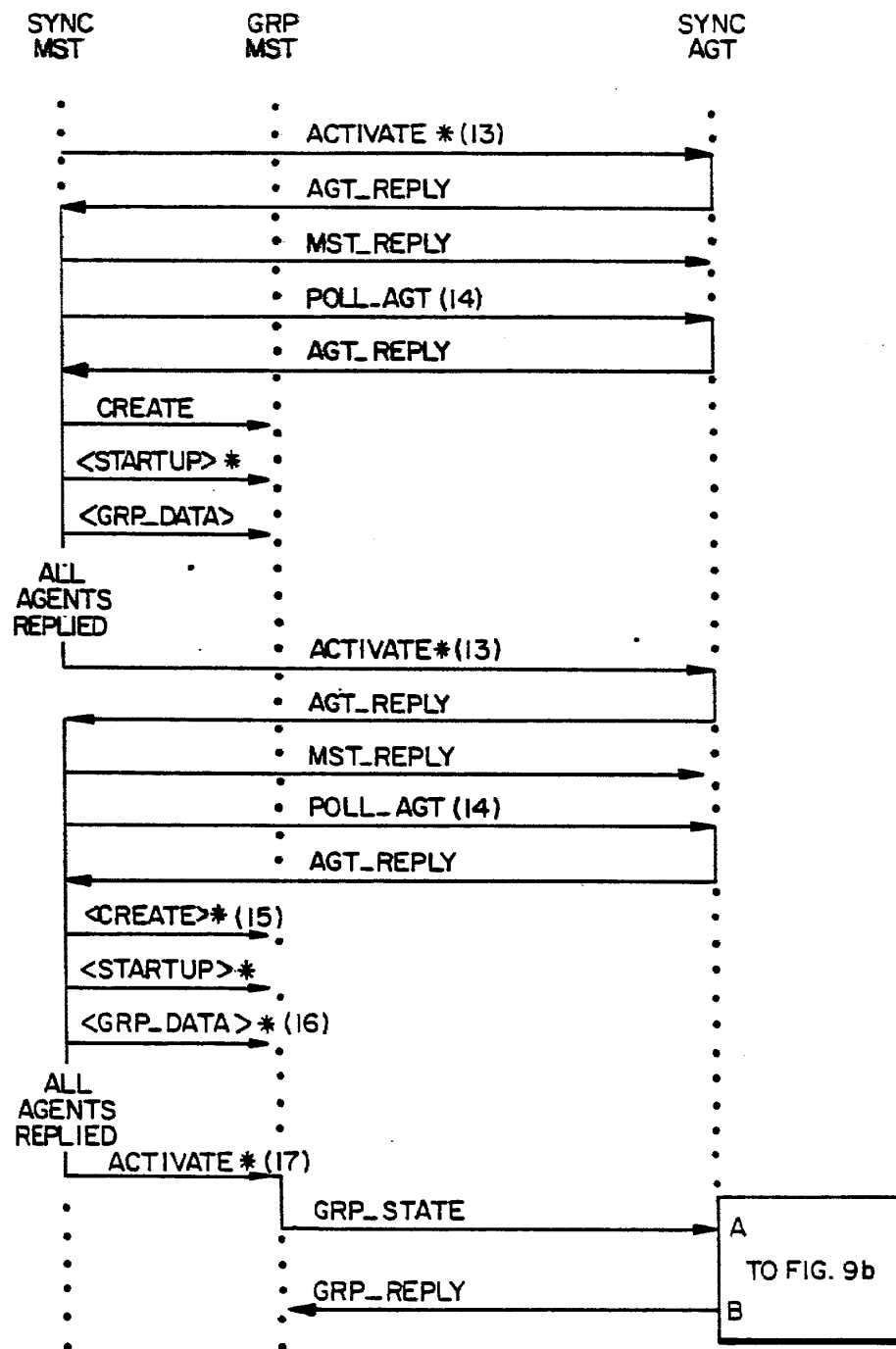


FIG. 9a

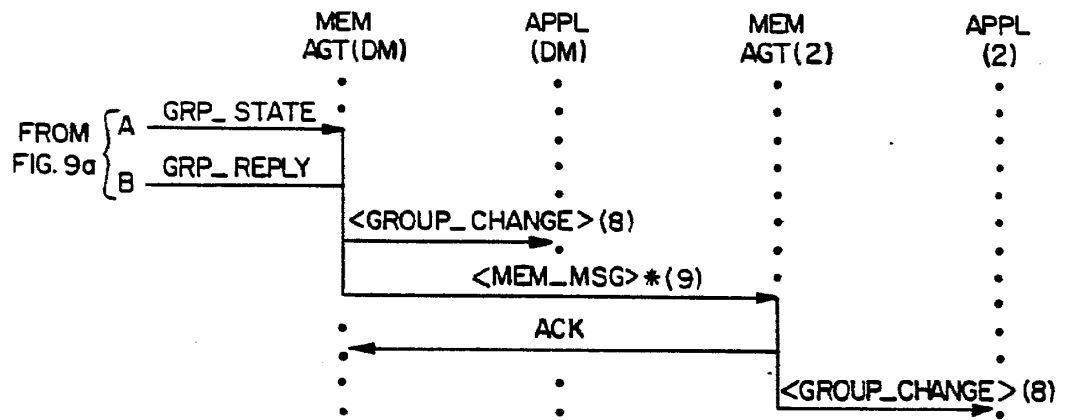


FIG. 9b

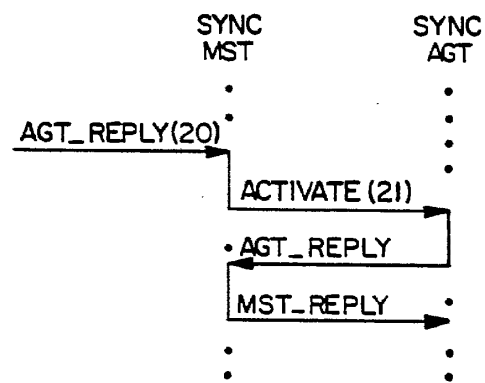


FIG. 11

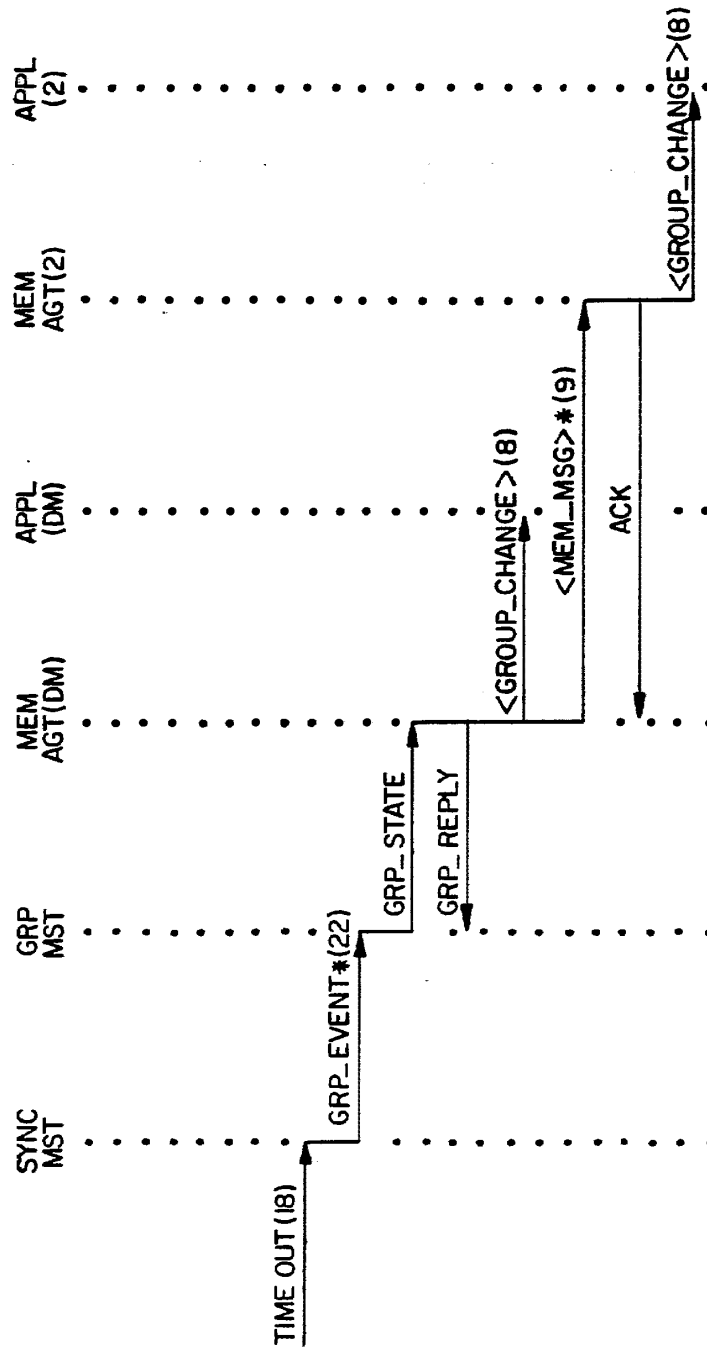


FIG.10

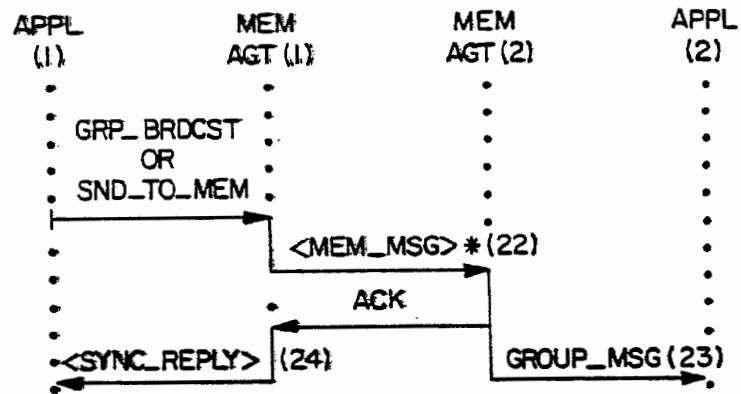


FIG. 12

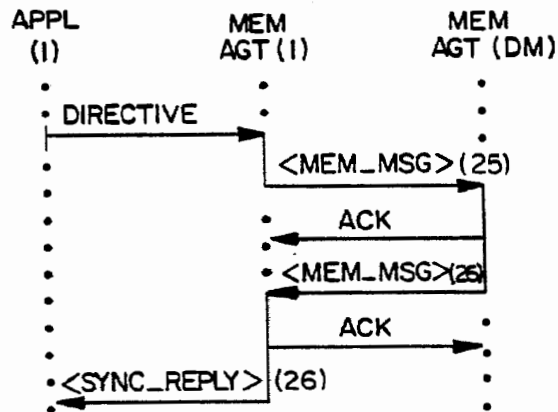
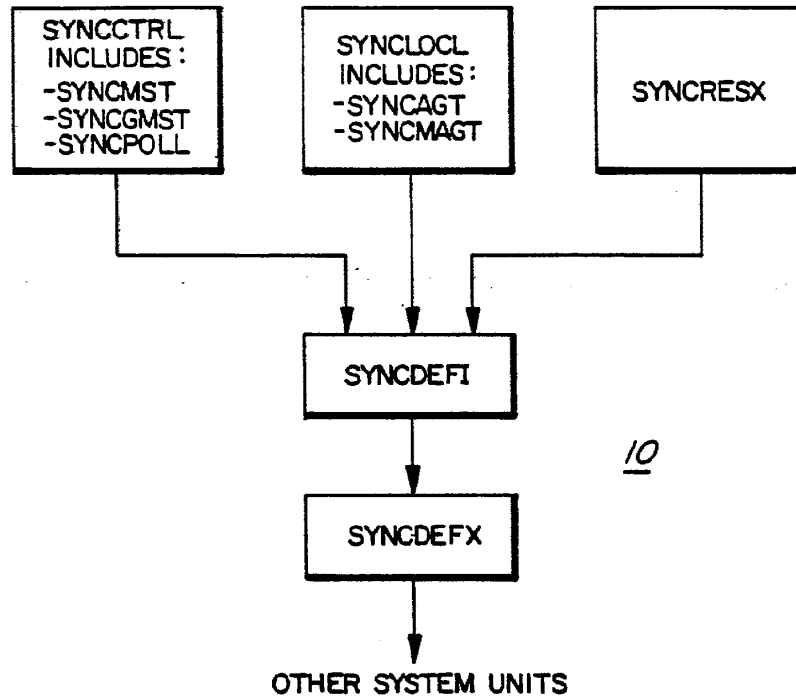


FIG. 13

**FIG. 14**



⑪ Publication number : **0 672 985 A1**

12

EUROPEAN PATENT APPLICATION

②① Application number : 95300753.1

⑤ Int. Cl.⁶: **G06F 11/14**, **G06F 11/20**

②② Date of filing : 07.02.95

③① Priority : 22.02.94 US 199444

④③ Date of publication of application :
20.09.95 Bulletin 95/38

⑧ Designated Contracting States :
DE FR GB

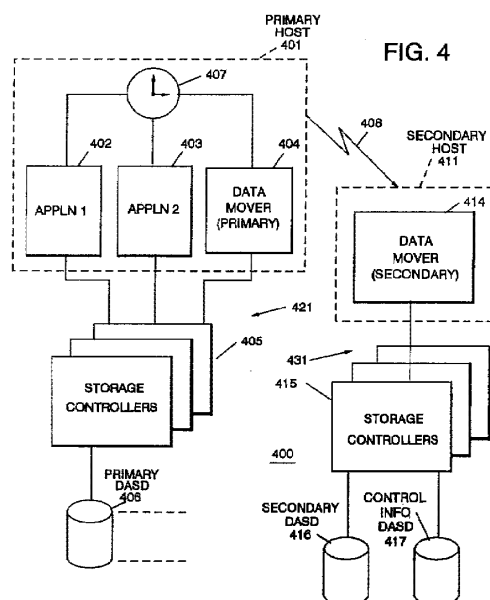
71 Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504 (US)

(72) Inventor : Kern, Robert Frederic
8338 East Calexico Street
Tucson, AZ 85730 (US)
Inventor : Kern, Ronald Maynard
761 North Colette Place
Tucson, AZ 85748 (US)
Inventor : McBride, Gregory Edward
8622 East Fairmount Place
Tucson, AZ 85715 (US)
Inventor : Shackelford, David Michael
1348 West Thurber Place Drive
Tucson, AZ 85705 (US)

74 Representative : Litherland, David Peter
IBM United Kingdom Limited
Intellectual Property Department
Hursley Park
Winchester, Hampshire SO21 2JN (GB)

(54) Asynchronous remote data duplexing.

(57) A remote data shadowing system provides storage based, real time disaster recovery capability. Record updates at a primary site cause write I/O operations in a storage subsystem therein. The write I/O operations are time stamped and the time, sequence, and physical locations of the record updates are collected in a primary data mover. The primary data mover groups sets of the record updates and associated control information based upon a predetermined time interval, the primary data mover appending a prefix header to the record updates thereby forming self describing record sets. The self describing record sets are transmitted to a remote secondary site wherein consistency groups are formed such that the record updates are ordered so that the record updates can be shadowed in an order consistent with the order the record updates cause write I/O operations at the primary site.



EP 0 672 985 A1

The present invention relates generally to disaster recovery techniques in data processing systems, and more particularly, to a system for real-time remote copying of data.

Data processing systems, in conjunction with processing data, typically are required to store large amounts of data (or records), which data can be efficiently accessed, modified, and re-stored. Data storage is typically separated into several different levels, or hierarchically, in order to provide efficient and cost effective data storage. A first, or highest level of data storage involves electronic memory, usually dynamic or static random access memory (DRAM or SRAM). Electronic memories take the form of semiconductor integrated circuits wherein millions of bytes of data can be stored on each circuit, with access to such bytes of data measured in nano-seconds. The electronic memory provides the fastest access to data since access is entirely electronic.

A second level of data storage usually involves direct access storage devices (DASD). DASD storage, for example, can comprise magnetic and/or optical disks, which store bits of data as micrometer sized magnetic or optical altered spots on a disk surface for representing the "ones" and "zeros" that make up those bits of the data. Magnetic DASD, includes one or more disks that are coated with remnant magnetic material. The disks are rotatably mounted within a protected environment. Each disk is divided into many concentric tracks, or closely spaced circles. The data is stored serially, bit by bit, along each track. An access mechanism, known as a head disk assembly (HDA), typically includes one or more read/write heads, and is provided in each DASD for moving across the tracks to transfer the data to and from the surface of the disks as the disks are rotated past the read/write heads. DASDs can store giga-bytes of data with the access to such data typically measured in milli-seconds (orders of magnitudes slower than electronic memory). Access to data stored on DASD is slower due to the need to physically position the disk and HDA to the desired data storage location.

A third or lower level of data storage includes tape and/or tape and DASD libraries. At this storage level, access to data is much slower in a library since a robot is necessary to select and load the needed data storage medium. The advantage is reduced cost for very large data storage capabilities, for example, tera-bytes of data storage. Tape storage is often used for back-up purposes, that is, data stored at the second level of the hierarchy is reproduced for safe keeping on magnetic tape. Access to data stored on tape and/or in a library is presently of the order of seconds.

Having a back-up data copy is mandatory for many businesses as data loss could be catastrophic to the business. The time required to recover data lost at the primary storage level is also an important recovery consideration. An improvement in speed over

tape or library back-up, includes dual copy. An example of dual copy involves providing additional DASD's so that data is written to the additional DASD's (sometimes referred to as mirroring). Then if the primary DASD's fail, the secondary DASD's can be depended upon for data. A drawback to this approach is that the number of required DASD's is doubled.

Another data back-up alternative that overcomes the need to provide double the storage devices involves writing data to a redundant array of inexpensive devices (RAID) configuration. In this instance, the data is written such that the data is apportioned amongst many DASD's. If a single DASD fails, then the lost data can be recovered by using the remaining data and error correction procedures. Currently there are several different RAID configurations available.

The aforementioned back-up solutions are generally sufficient to recover data in the event that a storage device or medium fails. These back-up methods are useful only for device failures since the secondary data is a mirror of the primary data, that is, the secondary data has the same volume serial numbers (VOLSERs) and DASD addresses as the primary data. System failure recovery, on the other hand, is not available using mirrored secondary data. Hence still further protection is required for recovering data if a disaster occurs destroying the entire system or even the site, for example, earthquakes, fires, explosions, hurricanes, etc. Disaster recovery requires that the secondary copy of data be stored at a location remote from the primary data. A known method of providing disaster protection is to back-up data to tape, on a daily or weekly basis, etc. The tape is then picked up by a vehicle and taken to a secure storage area usually some kilometers away from the primary data location. A problem is presented in this back-up plan in that it could take days to retrieve the back-up data, and meanwhile several hours or even days of data could be lost, or worst, the storage location could be destroyed by the same disaster. A somewhat improved back-up method would be to transmit data to a back-up location each night. This allows the data to be stored at a more remote location. Again, some data may be lost between back-ups since back-up does not occur continuously, as in the dual copy solution. Hence, a substantial data amount could be lost which may be unacceptable to some users.

More recently introduced data disaster recovery solutions include remote dual copy wherein data is backed-up not only remotely, but also continuously. In order to communicate duplexed data from one host processor to another host processor, or from one storage controller to another storage controller, or some combination thereof, a substantial amount of control data is required for realizing the process. A high overhead, however, can interfere with a secondary site's ability to keep up with a primary site's processing, thus threatening the ability of the secondary site to be

able to recover the primary in the event a disaster occurs.

Accordingly it is desired to provide a method and apparatus for providing a real time update of data consistent with the data at a primary processing location using minimal control data, wherein the method and apparatus operates independently of a particular application data being recovered, that is, generic storage media based rather than specific application data based.

An aim of the present invention is to provide an improved system and method for shadowing DASD data to a secondary site for disaster recovery.

According to a first aspect of the present invention, a method for forming consistency groups provides for disaster recovery capability from a remote site. Data updates generated by one or more applications running in a primary processor are received by a primary storage subsystem, wherein the primary storage subsystem causes I/O write operations to write each data update therein. The primary storage subsystem is synchronized by a common timer, and a secondary system, remote from the primary processor, shadows the data updates in sequence consistent order such that the secondary site is available for disaster recovery purposes. The method comprising steps of: (a) time stamping each write I/O operation occurring in the primary storage subsystem; (b) capturing write I/O operation record set information from the primary storage subsystem for each data update; (c) generating self describing record sets from the data updates and the respective record set information, such that the self describing record sets are sufficient to re-create a sequence of the write I/O operations; (d) grouping the self describing record sets into interval groups based upon a predetermined interval threshold; and (e) selecting a first consistency group as that interval group of self describing record sets having an earliest operational time stamp, the individual data updates being ordered within the first consistency group based upon time sequences of the I/O write operations in the primary storage subsystem.

In another aspect of the present invention, a primary system has a primary processor running one or more applications, wherein the applications generating record updates, and the primary processor generating self describing record sets therefrom. Each self describing record set is sent to a secondary system remote from the primary system, wherein the secondary system shadows the record updates in sequence consistent order based upon the self describing record sets for real time disaster recovery purposes. The primary processor is coupled to a primary storage subsystem wherein the primary storage subsystem receives the record updates and causes I/O write operations for storing each record update therein. The primary processor comprises a sysplex timer

for providing a common time source to the applications and to the primary storage subsystem for synchronization purposes, and a primary data mover, synchronized by the sysplex timer, prompts the primary storage subsystem for providing record set information to the primary data mover for each record update. The primary data mover groups a plurality of record updates and each corresponding record set information into time interval groups, and inserts a prefix header thereto. Each time interval group forms the self describing record sets.

A preferred embodiment of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

FIG. 1 is a block diagram of a disaster recovery system having synchronous remote data shadowing capabilities.

FIG. 2 is a flow diagram of a method for providing synchronous remote copy according to the disaster recovery system of FIG. 1.

FIG. 3 is a flow diagram of a method of an I/O error recovery program (I/O ERP) operation.

FIG. 4 is a block diagram of a disaster recovery system having asynchronous remote data shadowing capabilities.

FIG. 5 is a data format diagram showing a prefix header that prefixes read record sets from the primary site of FIG. 4.

FIG. 6 is a data format diagram describing fields making up a read record set.

FIG. 7 is a state table identifying volume configuration information.

FIG. 8 is a master journal as used by the secondary site of FIG. 4.

FIG. 9 is an example sequence for forming a consistency group.

FIG. 10 is a flow diagram showing a method of collecting information and read record sets for forming consistency groups.

FIG. 11 is a flow diagram showing a method of forming consistency groups.

FIG. 12 is a table indicating full consistency group recovery rules application for an ECKD architecture device for a given sequence of I/O operations to a DASD track.

FIG. 13 is a description of the rules to be used in the table of FIG. 12.

FIG. 14 is a flow diagram of a method of writing read record set copies to a secondary site with full consistency group recovery capability.

A typical data processing system may take the form of a host processor, such as an IBM System/360 or IBM System/370 processor for computing and manipulating data, and running, for example, data facility storage management subsystem/multiple virtual systems (DFSMS/MVS) software, having at least one IBM 3990 storage controller attached thereto, the storage controller comprising a memory controller

and one or more cache memory types incorporated therein. The storage controller is further connected to a group of direct access storage devices (DASDs) such as IBM 3380 or 3390 DASDs. While the host processor provides substantial computing power, the storage controller provides the necessary functions to efficiently transfer, stage/dstage, convert and generally access large databases.

Disaster recovery protection for the typical data processing system requires that primary data stored on primary DASDs be backed-up at a secondary or remote location. The distance separating the primary and secondary locations depends upon the level of risk acceptable to the user, and can vary from several kilometers to thousands of kilometers. The secondary or remote location, in addition to providing a back-up data copy, must also have enough system information to take over processing for the primary system should the primary system become disabled. This is due in part because a single storage controller does not write data to both primary and secondary DASD strings at the primary and secondary sites. Instead, the primary data is stored on a primary DASD string attached to a primary storage controller while the secondary data is stored on a secondary DASD string attached to a secondary storage controller.

The secondary site must not only be sufficiently remote from the primary site, but must also be able to back-up primary data in real time. The secondary site needs to back-up primary data as the primary data is updated with some minimal delay. Additionally, the secondary site has to back-up the primary data regardless of the application program (e.g., IMS, DB2) running at the primary site and generating the data and/or updates. A difficult task required of the secondary site is that the secondary data must be order consistent, that is, secondary data is copied in the same sequential order as the primary data (sequential consistency) which requires substantial systems considerations. Sequential consistency is complicated by the existence of multiple storage controllers each controlling multiple DASDs in a data processing system. Without sequential consistency, secondary data inconsistent with primary data would result, thus corrupting disaster recovery.

Remote data duplexing falls into two general categories, synchronous and asynchronous. Synchronous remote copy involves sending primary data to the secondary location and confirming the reception of such data before ending a primary DASD input/output (I/O) operation (providing a channel end (CE) and device end (DE) to the primary host). Synchronous copy, therefore, slows the primary DASD I/O response time while waiting for secondary confirmation. Primary I/O response delay is increased proportionately with the distance between the primary and secondary systems - a factor that limits the remote distance to tens of kilometers. Synchronous copy,

however, provides sequentially consistent data at the secondary site with relatively little system overhead.

Asynchronous remote copy provides better primary application system performance because the primary DASD I/O operation is completed (providing a channel end (CE) and device end (DE) to the primary host) before data is confirmed at the secondary site. Therefore, the primary DASD I/O response time is not dependent upon the distance to the secondary site and the secondary site could be thousands of kilometers remote from the primary site. A greater amount of system overhead is required, however, for ensuring data sequence consistency since data received at the secondary site will often not be in order of the primary updates. A failure at the primary site could result in some data being lost that was in transit between the primary and secondary locations.

Synchronous real time remote copy for disaster recovery requires that copied DASD volumes form a set. Forming such a set further requires that a sufficient amount of system information be provided to the secondary site for identifying those volumes (VOLSERs) comprising each set and the primary site equivalents. Importantly, the secondary site forms a "duplex pair" with the primary site and the secondary site must recognize when one or more volumes are out of sync with the set, that is, "failed duplex" has occurred. Connect failures are more visible in synchronous remote copy than in asynchronous remote copy because the primary DASD I/O is delayed while alternate paths are retried. The primary site can abort or suspend copy to allow the primary site to continue while updates for the secondary site are queued, the primary site marking such updates to show the secondary site is out of sync. Recognizing exception conditions that may cause the secondary site to fall out of sync with the primary site is needed in order that the secondary site be available at any time for disaster recovery. Error conditions and recovery actions must not make the secondary site inconsistent with the primary site.

Maintaining a connection between the secondary site and the primary site with secondary DASD present and accessible, however, does not ensure content synchronism. The secondary site may lose synchronism with the primary site for a number of reasons. The secondary site is initially out of sync when the duplex pair is being formed and reaches sync when an initial data copy is completed. The primary site may break the duplex pair if the primary site is unable to write updated data to the secondary site in which case the primary site writes updates to the primary DASD under suspended duplex pair conditions so that the updating application can continue. The primary site is thus running exposed, that is, without current disaster protection copy until the duplex pair is restored. Upon restoring the duplex pair, the secondary site is not immediately in sync. After applying now pending up-

dates, the secondary site returns to sync. The primary site can also cause the secondary site to lose sync by issuing a suspend command for that volume to the primary DASD. The secondary site re-syncs with the primary site after the suspend command is ended, duplex pair is re-established, and pending updates are copied. On-line maintenance can also cause synchronization to be lost.

When a secondary volume is out of sync with a primary volume, the secondary volume is not useable for secondary system recovery and resumption of primary applications. An out-of-sync volume at the secondary site must be identified as such and secondary site recovery-takeover procedures need to identify the out-of-sync volumes for denying application access (forcing the volumes off-line or changing their VOLSERs). The secondary site may be called upon to recover the primary site at any instant wherein the primary site host is inaccessible - thus the secondary site requires all pertinent information about a sync state of all volumes. The secondary storage subsystem, that is the secondary storage controllers and DASD, is unable to determine all conditions causing the primary site to break synchronism due to primary site-encountered exceptions. For example, the primary site may break a duplex pair if the primary site is unable to access the secondary peer due to a primary I/O path or link failure that the secondary site is unaware of. In this case the secondary site shows in-sync state while the primary site indicates the duplex pair is broken.

External communication may notify the secondary site that an out-of-sync duplex pair volume exists. This is realizable by employing a user systems management function. Primary I/O operations end with channel end/device end/unit check (CE/DE/UC) status and sense data indicates the nature of the error. With this form of I/O configuration an error recovery program (ERP) processes the error and send an appropriate message to the secondary processor before posting the primary application that I/O is complete. The user is then responsible to recognize the ERP suspend duplex pair message and secure that information at the secondary location. When the secondary site is depended upon to become operational in place of the primary site, a start-up procedure brings the secondary DASD on-line to the secondary host wherein sync status stored in the secondary DASD subsystem is retrieved for ensuring that out-of-sync volumes are not brought on-line for application allocation. This sync status merged with all ERP suspend duplex pair messages gives a complete picture of the secondary out-of-sync volumes.

Referring now to FIG. 1, a disaster recovery system 10 is shown having a primary site 14 and a secondary site 15, wherein the secondary site 15 is located, for example, 20 kilo-meters remote from the primary site 14. The primary site 14 includes a host

processor or primary processor 1 having an application and system I/O and Error Recovery Program 2 running therein (hereinafter referred to as I/O ERP 2). The primary processor 1 could be, for example, an IBM Enterprise Systems/9000 (ES/9000) processor running DFSMS/MVS operating software and further may have several application programs running thereon. A primary storage controller 3, for example, an IBM 3990 Model 6 storage controller, is connected to the primary processor 1 via a channel 12. As is known in the art, several such primary storage controllers 3 can be connected to the primary processor 1, or alternately, several primary processors 1 can be attached to the primary storage controllers 3. A primary DASD 4, for example, an IBM 3390 DASD, is connected to the primary storage controller 3. Several primary DASDs 4 can be connected to the primary storage controller 3. The primary storage controller 3 and attached primary DASD 4 form a primary sub-storage system. Further, the primary storage controller 3 and the primary DASD 4 could be single integral units.

The secondary site 15 includes a secondary processor 5, for example, an IBM ES/9000, connected to a secondary storage controller 6, for example an IBM 3990 Model 3, via a channel 13. A DASD 7 is further connected to the secondary storage controller 6. The primary processor 1 is connected to the secondary processor 5 by at least one host-to-host communication link 11, for example, channel links or telephone T1/T3 line links, etc. The primary processor 1 may also have direct connectivity with the secondary storage controller 6 by, for example, multiple Enterprise Systems Connection (ESCON) links 9. As a result, the I/O ERP 2 can communicate, if required, with the secondary storage controller 6. The primary storage controller 3 communicates with the secondary storage controller 6 via multiple peer-to-peer links 8, for example, multiple ESCON links.

When a write I/O operation is executed by an application program running in the primary processor 1, a hardware status channel end/device end (CE/DE) is provided indicating the I/O operation completed successfully. Primary processor 1 operating system software marks the application write I/O successful upon successful completion of the I/O operation, thus permitting the application program to continue to a next write I/O operation which may be dependent upon the first or previous write I/O operation having successfully completed. On the other hand, if the write I/O operation was unsuccessful, the I/O status of channel end/device end/unit check (hereinafter referred to as CE/DE/UC) is presented to the primary processor 1 operating system software. Having presented unit check, the I/O ERP 2 takes control obtaining specific sense information from the primary storage controller 3 regarding the nature of the failed write I/O operation. If a unique error to a volume oc-

curs then a unique status related to that error is provided to the I/O ERP 2. The I/O ERP 2 can thereafter perform new peer-to-peer synchronization error recovery for maintaining data integrity between the primary storage controller 3 and the secondary storage controller 6, or in the worst case, between the primary processor 1 and the secondary processor 5.

Referring to FIGs 2 and 3, the error recovery procedure is set forth. In FIG. 2, a step 201 includes an application program running in the primary processor 1 sending a data update to the primary storage controller 3. At step 203 the data update is written to the primary DASD 4, and the data update is shadowed to the secondary storage controller 6. At step 205 the duplex pair status is checked to determine whether the primary and secondary sites are synchronized. If the duplex pair status is in a synchronized state, then the data update is written to the secondary DASD 7 at step 207 while processing then continues at the primary processor 1 via application programs running thereat.

In the case that the duplex pair is in a "failed" state, then at step 209 the primary storage controller 3 notifies the primary processor 1 that duplex pair has suspended or failed. The duplex pair can become "failed" due to communication failure between the primary storage controller 3 and the secondary storage controller 6 via communication links 8. Alternatively, duplex pair can become "failed" due to errors in either the primary or secondary subsystem. If the failure is in the communication links 8, then the primary storage controller 3 is unable to communicate the failure directly to the secondary storage controller 6. At step 211 the primary storage controller 3 returns I/O status CE/DE/UC to the primary processor 1. The I/O ERP 2 quiesces the application programs hence taking control of the primary processor 1 at step 213 for error recovery and data integrity before returning control to the application requesting the write I/O operation.

FIG. 3 represents steps performed by the I/O ERP 2. The I/O ERP 2 issues a sense I/O to the primary storage controller 3 at step 221. The sense I/O operation returns information describing the cause of the I/O error, that is, the data description information is unique to the storage controllers or duplex pair operation regarding specific errors. In the event that the data description information indicates that the peer-to-peer communication links 8 have failed between the primary storage controller 1 and the secondary storage controller 6, then at step 223 the I/O ERP 2 issues a storage controller level I/O operation against the primary storage controller 3 and the secondary storage controller 6 indicating that the affected volume is to be placed in failed synchronous remote copy state. This secondary storage controller 6 is able to receive the state of the affected volume from the I/O ERP 2 via the - multiple ESCON links 9 or the host-

to-host communication link 11. Consequently, the current status of the duplex pair operation is maintained at both the primary processor 1 and the secondary processor 5 in conjunction with applications running in the primary processor 1. Consoles 18 and 19 are provided for communicating information from the primary processor 1 and secondary processor 4, respectively, wherein the I/O ERP posts status information to both consoles 18 and 19.

Data integrity has been maintained at step 225 upon successful completion of the failed synchronous remote copy I/O operation to the primary storage controller 3 and the secondary storage controller 6. Therefore, if a recovery is attempted at the secondary site 15 the secondary storage controller 6 identifies the volume marked "failed synchronous remote copy" as not being useable until data on that volume are synchronized with other data in that synchronization group by data recovery means (conventional data base logs and/or journals for determining the state of that data on the volume).

Step 227 tests to determine whether the I/O ERP 2 received successful completion of the I/O operations at the primary storage controller 3 and the secondary storage controller 6 on the failed synchronous remote copy status update. Upon successful completion, the I/O ERP 2 returns control to the Primary processor 1 at step 229. Otherwise step 231 performs a next level recovery notification which involves notifying an operator, via the console 18, of the failed volume and that a status of that volume at either the primary storage controller 3 or the secondary storage controller 6 may not be correct. The notification is shadowed to the secondary site 15, via the console 19 or a shared DASD data set, for indicating the specific volume status there.

An error log recording data set is updated at step 233. This update is written to either the primary DASD 4 or some other storage location and is shadowed to the secondary site 15. Having completed the error recovery actions, the I/O ERP 2, at step 235, posts to the primary application write I/O operation a "permanent error" for causing the primary application an error normal "permanent error" recovery for the failed write I/O operation. Once the error is corrected, the volume states can be recovered, first to pending (re-copy changed data) and then back to full duplex. The data may later be re-applied to the secondary DASD 7 once duplex pair is re-established.

When establishing a duplex pair a volume can be identified as CRITICAL according to a customer's needs. For a CRITICAL volume, when an operation results in failing a duplex pair, a permanent error failure of the primary volume is reported irrespective of the actual error's location. With CRIT=Y, all subsequent attempts to write to the primary DASD 405 of the failed pair will receive a permanent error, ensuring that no data is written to that primary volume that can-

not also be shadowed to the paired secondary volume. This permits complete synchronization with the primary application actions and the I/O data operations when required.

Consequently, the disaster recovery system 10 described herein, introduces outboard synchronous remote copy such that a primary host process error recovery procedure having an I/O order (channel command word (CCW)) may change a status of a primary and secondary synchronous remote copy volume from duplex pair to failed duplex thereby maintaining data integrity for several types of primary and secondary subsystem errors. Storage based back-up, rather than application based back-up, wherein data updates are duplicated in real time has been provided. The disaster recovery system 10 also attempts several levels of primary/secondary status updates, including: (1) primary and secondary storage controller volume status updates; (2) primary and secondary host processor notification on specific volume update status via operator messages or error log recording common data sets; and (3) CRITICAL volume indication, future updates to the primary volume can be prevented if the volume pair goes failed duplex. Hence, real time, full error disaster recovery is accomplished.

Asynchronous remote data shadowing is used when it is necessary to further increase a distance between primary and secondary sites for reducing the probability that a single disaster will corrupt both primary and secondary sites, or when primary application performance impact needs to be minimized. While the distance between primary and secondary sites can now stretch across the earth or beyond, the synchronization of write updates across multiple DASD volumes behind multiple primary subsystems to multiple secondary subsystems is substantially more complicated. Record write updates can be shipped from a primary storage controller via a primary data mover to a secondary data mover for shadowing on a secondary storage subsystem, but the amount of control data passed therebetween must be minimized while still being able to re-construct an exact order of the record write updates on the secondary system across several storage controllers as occurred on the primary system across multiple DASD volumes behind several storage controllers.

FIG. 4 depicts an asynchronous disaster recovery system 400 including a primary site 421 and a remote or secondary site 431. The primary site 421 includes a primary processor 401, for example, an IBM ES/9000 running DFSMS/MVS host software. The primary processor 401 further includes application programs 402 and 403, for example, IMS and DB2 applications, and a primary data mover (PDM) 404. A common sysplex clock 407 is included in the primary processor 401 for providing a common reference to all applications (402, 403) running therein, wherein all

system clocks or time sources (not shown) synchronize to the sysplex clock 407 ensuring all time dependent processes are properly timed relative to one another. The primary storage controllers 406, for example, synchronize to a resolution appropriate to ensure differentiation between record write update times, such that no two consecutive write I/O operations to a single primary storage controller 404 can exhibit the same time stamp value. The resolution, and not the accuracy, of the sysplex timer 407 is critical. The PDM 404, though shown connected to the sysplex timer 407, is not required to synchronize to the sysplex timer 407 since write I/O operations are not generated therein. A sysplex timer 407 is not required if the primary processor 401 has a single time reference (for example, a single multi-processor ES/9000 system).

A plurality of primary storage controllers 405, for example, IBM 3990 Model 6 storage controllers, are connected to the primary processor 401 via a plurality of channels, for example, fiber optic channels. Connected to each primary storage controller 405 is at least one string of primary DASDs 406, for example, IBM 3390 DASDs. The primary storage controllers 405 and the primary DASDs 406 form a primary storage subsystem. Each storage controller 405 and primary DASD 406 need not be separate units, but may be combined into a single drawer.

The secondary site 431, located for example, some thousands of kilometers remote from the primary site 421, similar to the primary site 421, includes a secondary processor 411 having a secondary data mover (SDM) 414 operating therein. Alternatively, the primary and secondary sites can be the same location, and further, the primary and secondary data movers can reside on a single host processor (secondary DASDs may be just over a fire-wall). A plurality of secondary storage controllers 415 are connected to the secondary processor 411 via channels, for example, fiber optic channels, as is known in the art. Connected to the storage controllers 415 are a plurality of secondary DASDs 416 and a control information DASD(s) 417. The storage controllers 415 and DASDs 416 and 417 comprise a secondary storage subsystem.

The primary site 421 communicates with the secondary site 431 via a communication link 408. More specifically, the primary processor 401 transfers data and control information to the secondary processor 411 by a communications protocol, for example, a virtual telecommunications access method (VTAM) communication link 408. The communication link 408 can be realized by several suitable communication methods, including telephone (T1, T3 lines), radio, radio/telephone, microwave, satellite, etc.

The asynchronous data shadowing system 400 encompasses collecting control data from the primary storage controllers 405 so that an order of all

data writes to the primary DASDs 406 is preserved and applied to the secondary DASDs 416 (preserving the data write order across all primary storage subsystems). The data and control information transmitted to the secondary site 431, must be sufficient such that the presence of the primary site 421 is no longer required to preserve data integrity.

The applications 402, 403 generate data or record updates, which record updates are collected by the primary storage controllers 405 and read by the PDM 404. The primary storage controllers 405 each grouped its respective record updates for an asynchronous remote data shadowing session and provides those record updates to the PDM 404 via non-specific primary DASD 406 READ requests. Transferring record updates from the primary storage controllers 405 to the PDM 404 is controlled and optimized by the PDM 404 for minimizing a number of START I/O operations and time delay between each read, yet maximizing an amount of data transferred between each primary storage controller 405 and the primary processor 401. The PDM 404 can vary a time interval between non-specific READs to control this primary storage controller-host optimization as well as a currency of the record updates for the secondary DASDs 416.

Collecting record updates by the PDM 404, and transmitting those record updates to the SDM 414, while maintaining data integrity, requires the record updates to be transmitted for specific time intervals and in appropriate multiple time intervals with enough control data to reconstruct the primary DASDs 406 record WRITE sequence across all primary storage subsystems to the secondary DASDs 416. Re-constructing the primary DASDs 406 record WRITE sequences is accomplished by passing self describing records from the PDM 404 to the SDM 414. The SDM 414 inspects the self describing records for determining whether any records for a given time interval have been lost or are incomplete.

FIGs 5 and 6 show a journal record format created by the PDM 404 for each self describing record, including a prefix header 500 (FIG. 5), and a record set information 600 (FIG. 6) as generated by the primary storage controller 405. Each self describing record is further journaled by the SDM 414 for each time interval so that each self describing record can be applied in time sequence for each time interval to the secondary DASDs 416.

Referring now to FIG. 5, the prefix header 500, which is inserted at the front of each record set, includes a total data length 501 for describing the total length of the prefix header 500 and actual primary record set information 600 that is transmitted to the SDM 414 for each record set. An operational time stamp 502 is a time stamp indicating a start time for the operational set that the PDM 404 is currently processing. The operational time stamp 502 is generated

by the PDM 404 (according to the sysplex timer 407) when performing a READ RECORD SET function to a set of the primary storage controllers 405. An I/O time 610 (FIG. 6) of the primary DASDs 406 write is unique for each primary storage controller 405 READ RECORD SET. The operational time stamp 502 is common across all storage controllers.

A READ RECORD SET command is issued by the PDM 404 and can be predicated upon one of the following conditions:

- (1) Primary storage controller 405 attention interrupt based upon that primary storage controller predetermined threshold;
- (2) Primary processor 401 timer interrupt based upon a predetermined time interval; or
- (3) Record set information indicates additional information on outstanding record sets available but not yet read.

Condition (2) uses a timer interval to control how far behind the secondary system 431 executes during periods of low activity. Condition (3) occurs when the PDM 404 fails to drain all record sets during a processing interval which drives further activity for ensuring that the PDM 404 keeps up with primary storage controller 405 activity.

A time interval group number 503 is supplied by the PDM 404 to identify a time interval (bounded by operational time stamp 502 and a records read time 507) for which the current record sets belong (sets of records across all primary storage controllers 405 for a given time interval group form consistency groups). A sequence number within group 504 is derived based upon a hardware provided identification (to the PDM 404) of a write sequence order of application WRITE I/Os for primary storage controller 405 for each record set within a given time interval group 503. A primary SSID (substorage identification) 505 uniquely identifies the specific primary storage controller of the primary storage controllers 405 for each record set. A secondary target volume 506 is assigned by either the PDM 404 or the SDM 414 depending upon performance considerations. A records read time 507 supplies an operational time stamp that is common to all primary storage controllers 405 indicating an end time for the PDM 404 read record set process current interval.

The operational time stamp 502 and the records read time 507 are used by the PDM 404 to group sets of read record sets from each of the primary storage controllers 405. Time synchronization for grouping sets of read record sets is key only to the PDM 404 and as such, the PDM 404 could be synchronized to a central processing unit (CPU) clock running only the PDM 404 not attached to the sysplex timer 407. The PDM 404 does not write record updates, but the record updates, as stated previously, must be synchronized to a common time source.

Referring now to FIG. 6, the record set informa-

tion 600 is generated by the primary storage controllers 405 and collected by the PDM 404. Update Specific Information 601-610, includes a primary device unit address 601 of each record indicating the actual primary DASD 406 that the record update occurred on. A cylinder number/head number (CCHH) 602 indicates a location on primary DASD 406 for each record update. Primary SSID 603, the primary storage controller session identifier is the same as primary SSID 505. Status flags 604 provide status information regarding whether specific data records 620 follow. Sequence numbers 605 and 630 assign a number to each record for indicating whether the entire record set has been read (all data transferred to the PDM 404). Primary DASD write I/O type 606 is an operation indicator identifying the type of write operation performed on each record, the operation indicators including: update write; format write; partial track records follow; full track data follows; erase command performed; or write any performed. Search argument 607 indicates initial positioning information for the first read record set data record 620. A sector number 608 identifies that sector that the record was updated at. Count field 609 describes a number of specific record data fields 620 that follow. A host application time when the primary DASD 406 write update occurred is recorded in time of updates 610. Specific record data 620 provides a count/key/data (CKD) field of each record update. Lastly, the sequence number 630 is compared to the sequence number 605 for indicating whether the entire read record set was transferred to the PDM 404.

The update records are handled in software groups called consistency groups so that the SDM 414 can copy the record updates in the same order they were written at the primary DASDs 406. The information used for creating the consistency groups (across all record sets collected from all storage controllers 405) includes the: operational time stamp 502; time interval group number 503; sequence number within group 504; primary controller SSID 505; records read time 507; primary device address 601; the primary SSID 603; and the status flags 604. The information used for determining whether all records for a time interval group have been received for each storage controller 405 at the SDM 414 includes the: time interval group number 503; sequence number within group 504; physical controller ID 505; the primary SSID 603; and a total number of read record sets returned from each primary storage controller 405 for each operational time interval. The information necessary to place record updates on the secondary DASDs 416 equivalently to the primary DASDs 406 record updates with full recover possible includes the: secondary target volume 506; CCHH 602; primary DASD write I/O type 606; search argument 607; sector number 608; count 609; time of updates 610; and the specific record data 620.

FIGs 7 and 8 show a state table 700 and a master journal 800, respectively, for describing a current journal contents, which simplifies recovery time and journal transfer time. The state table 700 provides configuration information, collected by and common to the PDM 404 and SDM 414, and includes primary storage controller session identifiers (SSID numbers) and the volumes therein, and the corresponding secondary storage controller session identifiers and the corresponding volumes. Thus the configuration information tracks which primary volumes 710 or primary DASD extents map to secondary volumes 711 or secondary DASD extents. With a simple extension to the state table 700 indicating partial volume extents 712 (CCHH to CCHH), partial volume remote copy can be accomplished using the same asynchronous remote copy methods described herein, but for a finer granularity (track or extent) than full volume.

The master journal 800 includes: consistency group number; location on journal volumes; and operational time stamp. The master journal 800 further maintains specific record updates as grouped in consistency groups. The state table 700 and master journal 800 support disaster recovery, and hence must be able to operate in a stand-alone environment wherein the primary system 401 no longer exists.

A time stamp control is placed at the front and back of each master journal 800 to ensure that the entire control entry was successfully written. The time stamp control is further written to the secondary DASDs 417. The control elements include dual entries (1) and (2), wherein one entry is always a current entry, for example:

- (1) Timestamp control | Control Info | Time-stamp Control
- (2) Timestamp Control | Control Info | Time-stamp Control.

At any point in time either entry (1) or (2) is the current or valid entry, wherein a valid entry is that entry with equal timestamp controls at the front and back. Disaster recovery uses the valid entry with the latest timestamp to obtain control information. This control information, along with state information (environmental information regarding storage controllers, devices, and applied consistency groups), is used for determining what record updates have been applied to the secondary storage controllers 415.

After all read record sets across all primary storage controllers 405 for a predetermined time interval are received at the secondary site 431, the SDM 414 interprets the received control information and applies the received read record sets to the secondary DASDs 416 in groups of record updates such that the record updates are applied in the same sequence that those record updates were originally written on the primary DASDs 406. Thus, all primary application order (data integrity) consistency is maintained at the secondary site 431. This process is hereinafter refer-

red to as forming consistency groups. Forming consistency groups is based on the following assumptions: (A) application writes that are independent can be performed in any order if they do not violate controller sequence order; (B) application writes that are dependent must be performed in timestamp order, hence an application cannot perform a dependent write number two before receiving control unit end, device end from write number one; and (C) a second write will always be either (1) in a same record set consistency group as a first write with a later timestamp or (2) in a subsequent record set consistency group.

Referring to FIG. 9, an example of forming a consistency group (the consistency group could be formed at either the primary site 421 or secondary site 431), for example, for storage controllers SSID 1, SSID 2, and SSID 3 is shown (any number of storage controllers can be included but three are used in this example for clarity). Time intervals T1, T2 and T3 are assumed to occur in ascending order. An operational time stamp 502 of time interval T1 is established for storage controllers SSID 1, SSID 2 and SSID 3. The PDM 404 obtains record set data from storage controller SSIDs 1, 2, and 3 for time interval T1-T3. The record sets for SSIDs 1, 2, and 3 for time interval T1 are assigned to time interval group 1, G1 (time interval group number 503). The sequence number within group 504 is shown for each SSID 1, 2, and 3, wherein SSID 1 has three updates at 11:59, 12:00, and 12:01, SSID 2 has two updates at 12:00 and 12:02, and SSID 3 has three updates at 11:58, 11:59, and 12:02. Record sets of time intervals T2 and T3 are listed but example times of updates are not given for simplicity.

Consistency group N can now be formed based upon the control information and record updates received at the secondary site 431. In order to ensure that no record update in time interval group number one is later than any record update of time interval group number two, a min-time is established which is equal to a the earliest read record set time of the last record updates for each storage controller SSID 1, 2, and 3. In this example then, min-time is equal to 12:01. Any record updates having a read record set time greater than or equal to min-time is included in the consistency group N + 1. If two record update times to a same volume were equal, though unlikely given sufficient resolution of the sysplex timer 407, the record update having the earlier sequence number within the time interval group N is kept with that group for consistency group N. The record updates are now ordered based upon read record set times. Record updates having equal times will cause the record update having the lower sequence number to be placed before the later sequence numbered record update. Alternatively, record updates having equal time stamps, but to differing volumes, may be ordered arbitrarily as long as they are kept in the same consistency group.

tency group.

If a primary storage controller 405 fails to complete a response to a read record set during a specified time interval, then a consistency group cannot be formed until that primary storage controller 405 completes. In the event that the primary storage controller 405 fails to complete its operation, then a missing interrupt results causing a system missing interrupt handler to receive control and the operation will be terminated. On the other hand, if the primary storage controller 405 timely completes the operation then the I/O will be driven to completion and normal operation will continue. Consistency group formation expects that write operations against the primary storage controllers 405 will have time stamps. Some programs, however, will cause writes to be generated without time stamps, in which case the primary storage controller 405 will return zeros for the time stamp. Consistency group formation can bound those records without time stamps based upon the timestamp that the data was read. If too many record updates without time stamps occur over a time interval such that the record updates are not easily bounded by consistency group times, then an error that the duplex volumes are out of synchronization may result.

FIGs 10 and 11 are flow diagrams presenting the method of forming consistency groups. Referring to FIG. 10, the process starts at step 1000 with the primary site 421 establishing remote data shadowing to occur. At step 1010 all application I/O operations are time stamped using the sysplex timer 407 as a synchronization clock (FIG. 4). The PDM 404 starts a remote data shadowing session with each primary storage controller 405 at step 1020 which includes identifying those primary volumes that will have data or records shadowed. Record set information 600 is trapped by the primary storage controllers 405 for each application WRITE I/O operation (see FIG. 6) by step 1030.

Step 1040 involves the PDM 404 reading the captured record set information 600 from each primary storage controller 405 according to a prompt including an attention message, a predetermined timing interval, or a notification of more records to read as described earlier. When the PDM 404 begins reading record sets, at step 1050, the PDM 404 prefixes each record set with a prefix header 500 (see FIG. 5) for creating specific journal records (a journal record includes the prefix header 500 and the record set information 600). The journal records contain the control information (and records) necessary for forming consistency groups at the secondary site 431 (or at the primary site 421).

At step 1060 the PDM 404 transmits the generated journal records to the SDM 414 via the communications link 408 (or within the same data mover system if the consistency groups are formed therein). The SDM 414 uses the state table 700 at step 1070

to gather the received record updates by group and sequence numbers for each time interval group and primary storage controller 405 established for the data shadowing session. The SDM 414 inspects the journal records at step 1080 to determine whether all record information has been received for each time interval group. If the journal records are incomplete, then step 1085 causes the SDM 414 to notify the PDM 404 to resend the required record sets. If the PDM 404 is unable to correctly resend, then the duplex volume pair is failed. If the journal records are complete, then step 1090 is performed which encompasses the SDM 414 forming the consistency groups.

Referring to FIG. 11, steps 1100-1160 representing step 1090 (FIG. 10) for forming consistency groups is shown. Consistency group formation starts at step 1100 wherein each software consistency group is written to an SDM 414 journal log ("hardened") on the secondary DASD 417 (FIG. 4). Step 1110 performs a test for determining whether the time interval groups are complete, that is, each primary storage controller 405 must have either presented at least one read record set buffer or have confirmation from the PDM 404 that no such record updates were placed in the record set buffer, and all read record set buffers with data (or null) must have been received by the SDM 414. If a time interval group is incomplete, then step 1110 retries reading the record sets from the primary storage controller 405 until the required data is received. If errors occur, a specific duplex volume pair or pairs may be failed. Having received complete time interval groups, step 1120 determines a first consistency group journal record. The first (or current) consistency group journal record is that record which contains the earliest operational time stamp 502 and the earliest time of update 610 of all records having equal operational time stamps 502.

Step 1130 inspects the records contained in the current consistency group journal record to determine which record will be the last record to be included therein (some records will be dropped and included in the next consistency group journal record). The last record in the current consistency group journal record is determined as a minimum update time (min-time) of the maximum update times for each primary storage controller 405 (that is, the last update of each primary storage controller 405 is compared and only the earliest of these remains in the current consistency group journal record).

Those remaining record updates in the current consistency group journal record are ordered according to time of update 610 and sequence number within group 504 by step 1140. A primary storage controller 405 that had no record updates does not participate in the consistency group. At step 1150, the remaining record updates of the current consistency group (having update times later than min-time) are passed to the next consistency group. Each sequence number

within a group 504 should end with a null buffer indicating that all read record sets have been read for that operational time interval. If the null buffer is absent, then the step 1120 of defining the last record in the current software consistency group, coupled with the records read time 507 and time of update 610 can be used to determine the proper order of the application WRITE I/O operations across the primary storage controllers 405.

Step 1160 represents a back-end of the remote data shadowing process wherein specific write updates are applied to secondary DASDs 416 under full disaster recovery constraint. If when writing the updates to the secondary DASDs 416 an I/O error occurs, or the entire secondary site 431 goes down and is re-initialized, then the entire consistency group that was in the process of being written can be re-applied from the start. This permits the remote shadowing to occur without having to track which secondary DASDs 416 I/Os have occurred, which I/Os have not occurred, and which I/Os were in process, etc.

A key component of step 1160 is that the PDM 414 causes the records to be written efficiently to the secondary DASDs 416 so that the secondary site 431 can keep pace with the primary site 421. The requisite efficiency is accomplished, in part, by concurrently executing multiple I/O operations to different secondary DASDs 416. Serially writing one secondary DASD 416 at a time would cause the secondary site 431 to fall too far behind the primary site 421. Yet more efficiency is gained at the secondary site 431 by writing the records for each consistency group destined for a single secondary device via a single channel command word (CCW) chain. Within each single CCW chain, the I/O operations therein can be further optimized as long as those I/O operations to each secondary DASD 416 data track are maintained in the order of occurrence on the primary volumes.

Optimizing secondary I/O operations for specific consistency groups and within single CCW chains is based in part upon the pattern of primary write I/O operations, and in part upon the physical characteristics of the secondary DASDs 416. Optimization may vary somewhat depending upon whether secondary DASD 415 is count/key/data (CKD), extended count/key/data (ECKD), fixed block architecture (FBA), etc. Consequently, a number of WRITE I/Os (m) to a primary DASD 406 volume during a given time interval can be reduced to a single START I/O operation to a secondary DASD 416 volume. This optimization of the number of START I/Os to the secondary storage controllers 415 of m:1 can allow the secondary DASDs 416 to catch up with and thereby closer shadow the record updates at the primary site 421.

A key to successful remote data shadowing, and hence secondary I/O optimization, is minimizing unrecoverable errors in any of the concurrent multiple I/O operations to secondary DASDs 416 so that con-

sistent copies are available for recovery. A failure in a given secondary write could permit a later dependent write to be recorded without the conditioning write (e.g., a log entry indicating that a data base record has been updated when in reality the actual update write for the data base had failed violates the sequence integrity of the secondary DASD 416 copy).

A failed secondary 416 copy is unusable for application recovery until that failure to update has been recovered. The failed update could be corrected by having the SDM 414 request a current copy from the PDM 404. In the mean time the secondary data copy is inconsistent and hence unusable until the PDM 404 responds with the current update and all other previous updates are processed by the PDM 414. The time required to recover the failed update typically presents an unacceptably long window of non-recovery for adequate disaster recovery protection.

Effective secondary site 431 I/O optimization is realized by inspecting the data record sets to be written for a given consistency group and building chains based upon rules of the particular secondary DASD 416 architecture, for example, an ECKD architecture. The optimization technique disclosed herein simplifies recovery from I/O errors such that when applying a consistency group, if an I/O error occurs, the CCW chain can be re-executed, or in the event of a secondary initial program load (IPL) recovery, the entire consistency group can be re-applied without data loss.

FIG. 12 summarizes full consistency group recovery (FCGR) rules for building CCW chains for all WRITE I/O combinations for an ECKD architecture, wherein CCHHR record format is used (cylinder number, head number, record number). FIG. 12 is created by inspecting each possible combination of WRITE I/O operations to a DASD track within a consistency group's scope. The FCGR rules of FIG. 12, described in FIG.s 13A and 13B, are then followed to govern data placement (secondary DASD 416 I/O write CCW chains) for yielding full recovery for an error in applying a consistency group. The FCGR rules depicted in FIG. 12 would be extended appropriately as new WRITE I/O operations are added. These rules can exist in hardware or software at the secondary site 431. The FCGR rules advantageously reduces READ record set to a same DASD track analysis to an inspection of the primary DASD 406 WRITE I/O type, search argument, and count and key fields.

If a DASD track is written without inspecting the consistency group write operations as shown in FIG. 12, then previously written data records potentially cannot be re-written. For example, assume that a chain includes:

WRITE UPDATE to record five; and
FORMAT WRITE to record one,

wherein record one and record five occur on the same DASD track with record one preceding record five. Record five is updated by an UPDATE WRITE CCW

and a FORMAT WRITE I/O CCW updates record one erasing a remainder of the track thus deleting record five. If this chain had to be re-executed, a LOCATE RECORD CCW that will position to the beginning of record five will no longer have a positioning point (record five no longer exists), and the chain is not fully recoverable from the beginning. Since the write operations have already been successful at the primary site 421, always being able to apply an entire consistency group on the secondary DASD 416 is required to maintain data consistency and integrity.

FIG. 14, steps 1410 through 1470, provides more detail as to the process represented by step 1160 of FIG. 11, while using the FCGR rules defined in FIG. 12. At step 1410 the SDM 414 divides the records of the current consistency group into two categories. A first category includes I/O orders directed to a same secondary DASD volume, and a second category includes I/O orders of those records in the first category that are directed to a same CCHH (i.e., records being updated to a same DASD track).

Having categorized the records of the current consistency group, step 1420 conforms application WRITE I/Os and SDM 414 WRITE I/Os to the architecture of the secondary DASDs 416, for example, to ECKD architecture FCGR rules (see FIG. 12) for identifying data placement on a track and track/record addressing. The SDM 414 groups secondary DASD WRITE I/O operations to the same volumes into single I/O CCW chains at step 1430. Step 1440 involves moving the head disk assembly (HDA) of each secondary DASD 416 according to search arguments and specific record data (CKD fields) for the actual secondary DASD 416 writes.

Step 1450 compares READ SET BUFFERS one and two for those records making up the second categories (there typically will be a plurality of second categories, one for each track receiving records), using the FCGR rules of FIG. 12 for determining whether a subsequent write operation invalidates a previous write operation or DASD search argument (positioning at a record that is now erased, etc.). The READ SET BUFFERS one and two contain adjacent read record sets. Following the FCGR rules ensures that the SDM 414 can re-write an entire consistency group, in the event of an error, without re-receiving record updates from the primary site 421. After the SDM 414 applies the current consistency group to the secondary DASD 416, step 1460 updates the state table (FIG. 7) and the master journal (FIG. 8).

The remote copy process continues in real time as step 1470 gets a next consistency group (which becomes the current consistency group) and returns processing to step 1410. The remote copy process will stop if the primary site 421 to secondary site 431 communication terminates. The communication may terminate if volume pairs are deleted from the process by the PDM 404, the primary site is destroyed

(disaster occurs), an orderly shutdown is performed, or a specific takeover action occurs at the secondary site 431. Consistency groups journaled on the secondary site 431 can be applied to the secondary DASD 416 during a takeover operation. The only data lost is that data captured by the primary site 421 that has not been completely received by the SDM 414.

In summary, synchronous and asynchronous remote data duplexing systems have been described. The asynchronous remote data duplexing system provides storage based, real time data shadowing. A primary site runs applications generating record updates, and a secondary site, remote from the primary site, shadows the record updates and provides disaster recovery for the primary site. The asynchronous remote data duplexing system comprises a sysplex timer for synchronizing time dependent processes in the primary site, and a primary processor at the primary site for running the applications, the primary processor having a primary data mover therein. A plurality of primary storage controllers are coupled to the primary processor for issuing write I/O operations for each record update, each primary storage controller DASD write I/O operation being synchronized to the sysplex timer. A plurality of primary storage devices receive the write I/O operations and store the record updates therein accordingly. The primary data mover collects record set information from the plurality of primary storage controllers for each record update and appends a prefix header to a predetermined group of record set informations. The prefix header and predetermined group of record set informations form the self describing record sets. Each record set information includes a primary device address, a cylinder number and head number (CCHH), a record update sequence number, a write I/O type, a search argument, a sector number, and a record update time. The prefix header includes a total data length, an operational time stamp, a time interval group number, and a records read time. A secondary processor at the secondary site has a secondary data mover, the secondary data mover receiving the self describing record sets from the primary site. A plurality of secondary storage controllers are coupled to the secondary processor, and a plurality of secondary storage devices are coupled to the secondary storage controllers for storing the record updates copies. The secondary data mover determines whether the transmitted self describing record sets are complete and forms consistency groups from the self describing record sets and provides the record updates from each consistency group to the plurality of secondary storage controllers for writing to the plurality of secondary storage devices in an order consistent with a sequence that the record updates were written to the plurality of primary storage devices.

While the invention has been particularly shown and described with reference to preferred embodi-

ments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention as defined in the appended claims. For example, the consistency groups have been described as being formed by the secondary data mover based upon received self describing record sets, however, the consistency groups could be formed at the primary site based upon write record sets or elsewhere in the secondary site. The formats the storage devices at the primary and secondary sites need not be identical. For example, CKD records could be converted to fixed block architecture (FBA) type records, etc. Nor are the storage devices meant to be limited to DASD devices.

Claims

1. A method of providing asynchronous data duplexing, wherein data updates generated by one or more applications running in a primary processor are received by a primary storage subsystem, the primary storage subsystem causing I/O write operations to write each data update therein, each write I/O operation being time stamped, the time stamps synchronized by a common timer, and wherein a secondary system, whether local to or remote from the primary processor, shadows the data updates in sequence consistent order such that the secondary site is available for disaster recovery purposes, the method comprising the steps of:
 - (a) time stamping each write I/O operation occurring in the primary storage subsystem;
 - (b) capturing write I/O operation record set information from the primary storage subsystem for each data update;
 - (c) generating self describing record sets from the data updates and respective record set informations, the self describing record sets containing sufficient control information to enable recreation of a sequence of the write I/O operations solely by the secondary system;
 - (d) grouping the self describing record sets into interval groups, each interval group being measured from an operational time stamp start time and continuing for a predetermined interval threshold; and
 - (e) selecting a current consistency group as that interval group of self describing record sets having an earliest operational time stamp, the individual data updates being ordered within the current consistency group based upon time sequences of the I/O write operations in the primary storage subsystem.

2. The method as claimed in claim 1 wherein the step (b) further includes initiating sessions with the primary storage subsystem based upon the operational time stamps for identifying a starting time for each interval group, each interval group being bounded by consecutive operational time stamps. 5
 3. The method as claimed in claim 1 or claim 2 wherein the step (d) includes adding a prefix header describing each interval group. 10
 4. The method as claimed in any preceding claim further comprising a step (f) transmitting the interval groups of self describing record sets to the secondary site . 15
 5. The method as claimed in claim 4 further comprising a step (g) determining at the secondary site whether each received self describing record set is complete. 20
 6. The method as claimed in claim 5 wherein the step (g) further includes the secondary site requesting the primary site to re-transmit any missing data updates if the secondary site determined a self describing record set is incomplete. 25
 7. The method as claimed in claim 6 further comprising a step (h) determining at the secondary site whether each time interval group is complete. 30
 8. The method as claimed in claim 7 wherein the step (h) further includes the secondary site requesting the primary site to re-send a missing record set if the secondary site determined that an interval group was incomplete. 35
 9. The method as claimed in claim 8 further comprising a step (i) writing the received data updates at the secondary site to the secondary storage subsystem according to the sequence of the corresponding write I/O operations at the primary site as ordered in the consistency groups. 40 45
 10. A method for providing remote data shadowing for disaster recovery purposes in a data processing system including a primary site having a primary processor running a primary data mover and applications generating record updates, the primary processor coupled to a primary storage subsystem having storage devices for storing the record updates according to write I/O operations issued by the primary processor to the primary storage subsystem, the primary site further including a common system timer for synchronizing time dependent operations in the primary site, 50 55
- the system further including a secondary site having a secondary processor communicating with the primary processor, and a secondary storage subsystem for storing copies of the record updates in sequence consistent order, the method comprising the steps of:
- (a) time stamping each write I/O operation in the primary storage subsystem;
 - (b) establishing a session with each storage device in the primary storage subsystem;
 - (c) capturing record set information from each storage device in the primary storage subsystem;
 - (d) reading record sets and respective record set information in the primary data mover;
 - (e) prefixing each record set with a header and creating self describing record sets therefrom;
 - (f) transmitting the self describing record sets to the secondary processor in time interval groups according to predetermined time intervals;
 - (g) forming consistency groups from the self describing record sets; and
 - (h) shadowing the record updates of each consistency group to the secondary storage subsystem in a sequence consistent order.
11. The method as claimed in claim 10 wherein the record sets are transmitted to the secondary processor asynchronously.
 12. The method as claimed in claim 10 or claim 11 wherein the step (g) is performed at the secondary site.
 13. The method as claimed in any of claims 10 to 12 wherein the step (f) further includes determining at the secondary site whether each received self describing record set is complete.
 14. The method as claimed in claim 13 wherein the step (f) further includes requesting the primary site to re-transmit any missing record updates if the primary site determined a received self describing record set is incomplete.
 15. The method as claimed in any of claims 10 to 14 further comprising a step (i) determining at the secondary site whether each time interval group is complete.
 16. The method as claimed in claim 15 wherein the step (i) further includes requesting the primary site to re-send a missing record set if the secondary site determined that a interval group was incomplete.

17. The method as claimed in any of claims 10 to 16 wherein the step (c) identifies, in the record set information, a physical location on the primary storage devices where each record update is stored. 5
18. The method as claimed in claim 17 wherein the step (c) identifies, in the record set information, a sequence and time of update of each record update stored to the primary storage devices within the session. 10
19. The method as claimed in any of claims 10 to 18 wherein the step (e) identifies, in the prefix header, an interval group number for the session and sequence within group for each record update referred to therein. 15
20. A primary data processing system having a primary processor running one or more applications, the one or more applications generating record updates, the primary processor generating self describing record sets therefrom, the self describing record sets being sent to a secondary system, the secondary system shadowing the record updates in sequence consistent order based upon the self describing record sets for real time disaster recovery purposes, the primary processor being coupled to a primary storage subsystem wherein the primary storage subsystem receives the record updates and executes write I/O operations for storing each record update therein, the primary processor comprising: 20
a timer for providing a common time source to the applications and primary storage subsystem for synchronization purposes; and 25
primary data mover means prompting the primary storage subsystem for providing record set information to the primary data mover means for each record update, the primary data mover means grouping a plurality of record updates and each corresponding record set information into time interval groups, and inserting a prefix header thereto, each time interval group the self describing record sets. interval groups, and inserting a prefix header thereto, each time interval group the self describing record sets. 30
35
40
45
21. The primary system as claimed in claim 20 wherein the primary storage subsystem comprises: 50
a plurality of primary storage controllers, the plurality of primary storage controllers issuing the write I/O operations; and
a plurality of primary storage devices coupled to the plurality of primary storage controllers. 55
22. The primary system as claimed in claim 20 or claim 21 wherein the primary data mover means collects record set information for each write I/O operation for each primary storage controller of the plurality of primary storage controllers participating with each time interval group.
23. The primary system as claimed in any of claims 20 to 22 wherein each write I/O operation is time-stamped in the primary processor relative to the sysplex timer, each write I/O operation being issued to a primary storage controller of the plurality of primary storage controllers, each primary storage controller preserving the time-stamp and returning that time-stamp in the corresponding read record set to the primary data mover means.
24. The primary system as claimed in claim 23 wherein each record set information identifies a corresponding record update's physical location on a primary storage device of the plurality of primary storage devices.
25. The primary system as claimed in claim 23 or claim 24 wherein each record set information identifies a corresponding record update's primary subsystem identification, primary device address, cylinder number, and head number.
26. The primary system as claimed in claim 23 wherein the primary data mover means identifies a relative sequence of each write I/O update across all primary storage controllers participating in a time interval group.
27. The primary system as claimed in claim 26 wherein the primary data mover means creates a state table for journaling record updates and cross referencing a storage location of each record update on the primary system and the secondary system.
28. The primary system as claimed in claim 26 wherein the primary data mover means communicates the state table to the secondary system.
29. A remote data shadowing system including a primary site and a secondary site, the secondary site asynchronously shadowing record updates of the primary site in real time for disaster recovery purposes, the record updates generated by applications running at the primary site, the primary site comprising: 15
a sysplex timer;
a primary processor running the applications generating the record updates and issuing a corresponding write I/O operation for each record update, the primary processor having a pri-

mary data mover therein;

a plurality of primary storage controllers directed to store the record updates, the plurality of primary storage controllers executing the issued write I/O operation for each record update; 5
and

a plurality of primary storage devices receiving and storing the record updates therein according to the corresponding write I/O operations, wherein the primary processor and each write I/O are time-stamped by the primary processor, as synchronized by the sysplex timer, such that write I/O operations are accurately sequence ordered relative to each other, the primary data mover collecting sets of record updates and combining each record set information as provided by each one of the plurality of primary storage controllers with the corresponding record update, each record set information including a relative sequence and time of each corresponding write I/O operation, the primary data mover collecting record updates into time interval groups and inserting a prefix header to each time interval group, wherein the prefix header includes information identifying the record updates included in each time interval group, each record set information and prefix header combined for creating self describing record sets, the self describing record sets being transmitted to the secondary site, wherein the self describing record sets provide information adequate for the secondary site to shadow the record updates therein in sequence consistent order without further communications from the primary site. 35

35

40

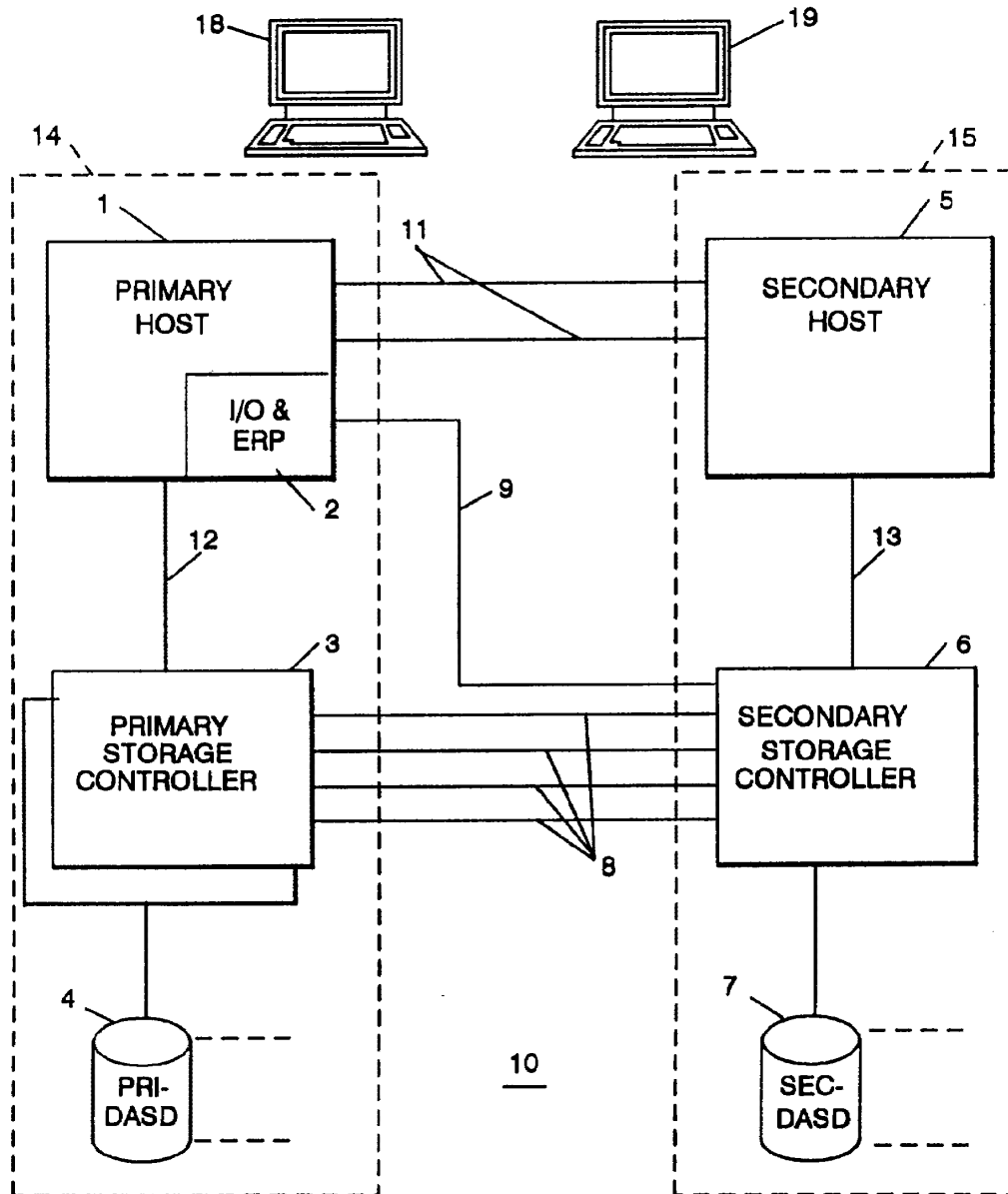
45

50

55

16

FIG. 1



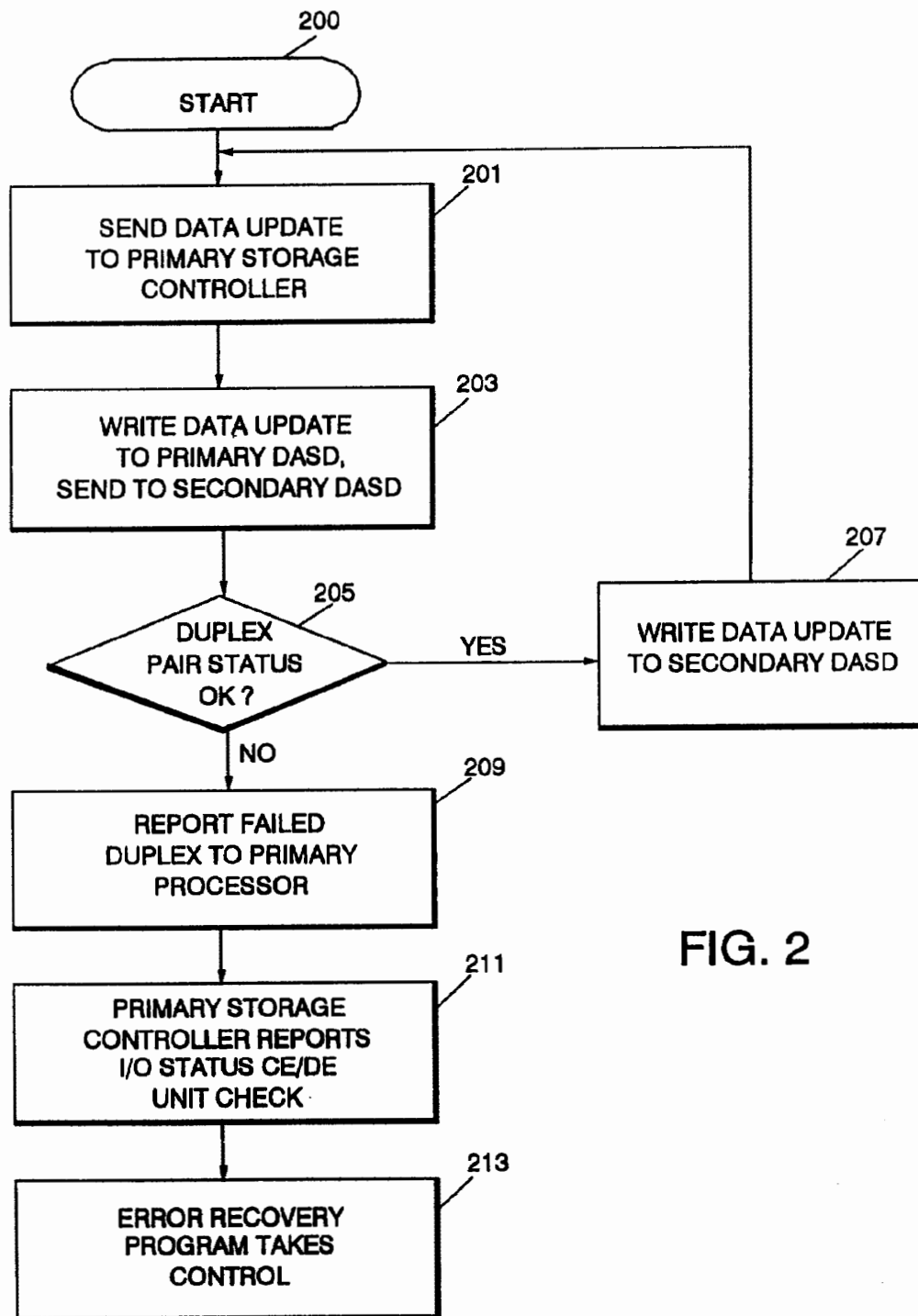
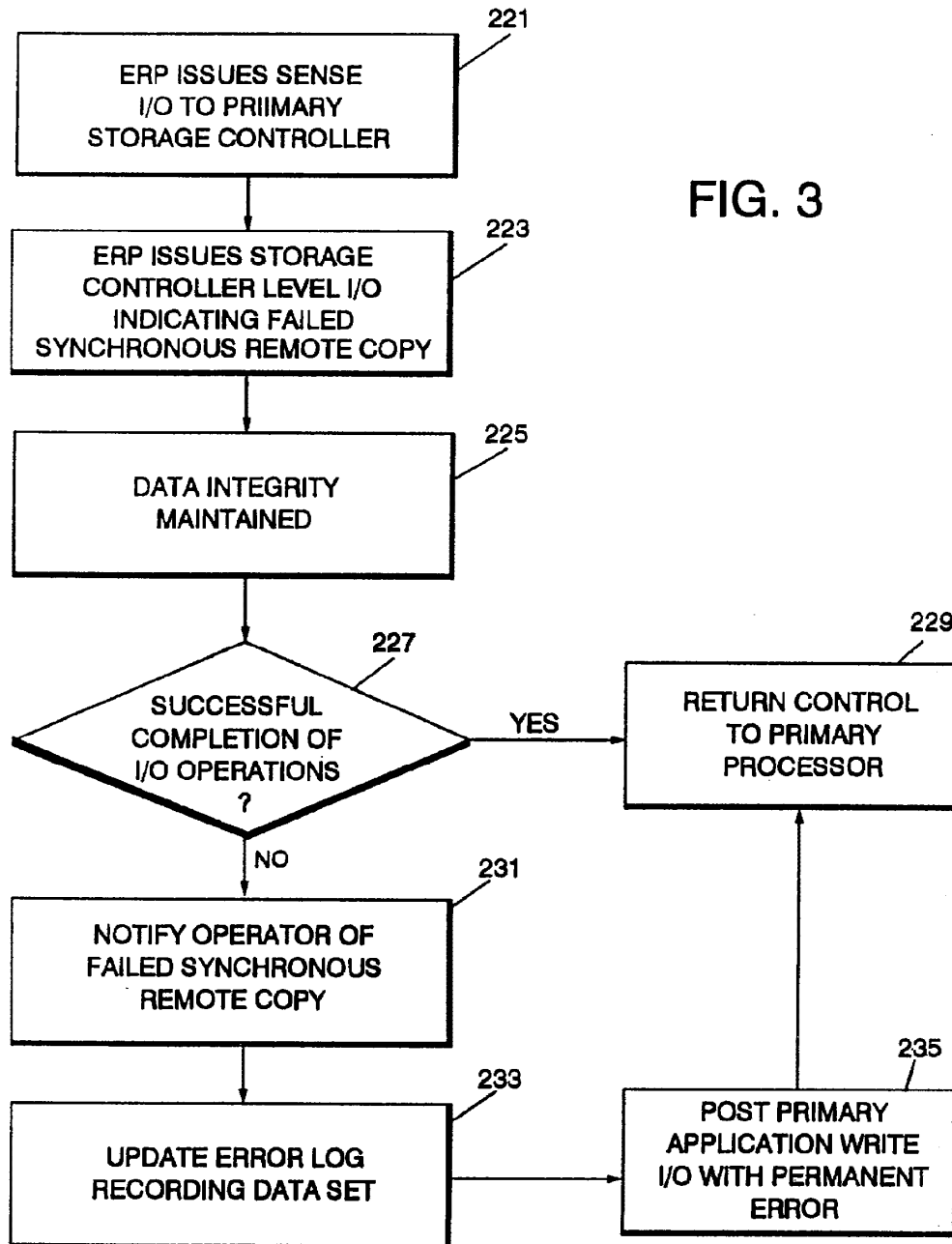


FIG. 2



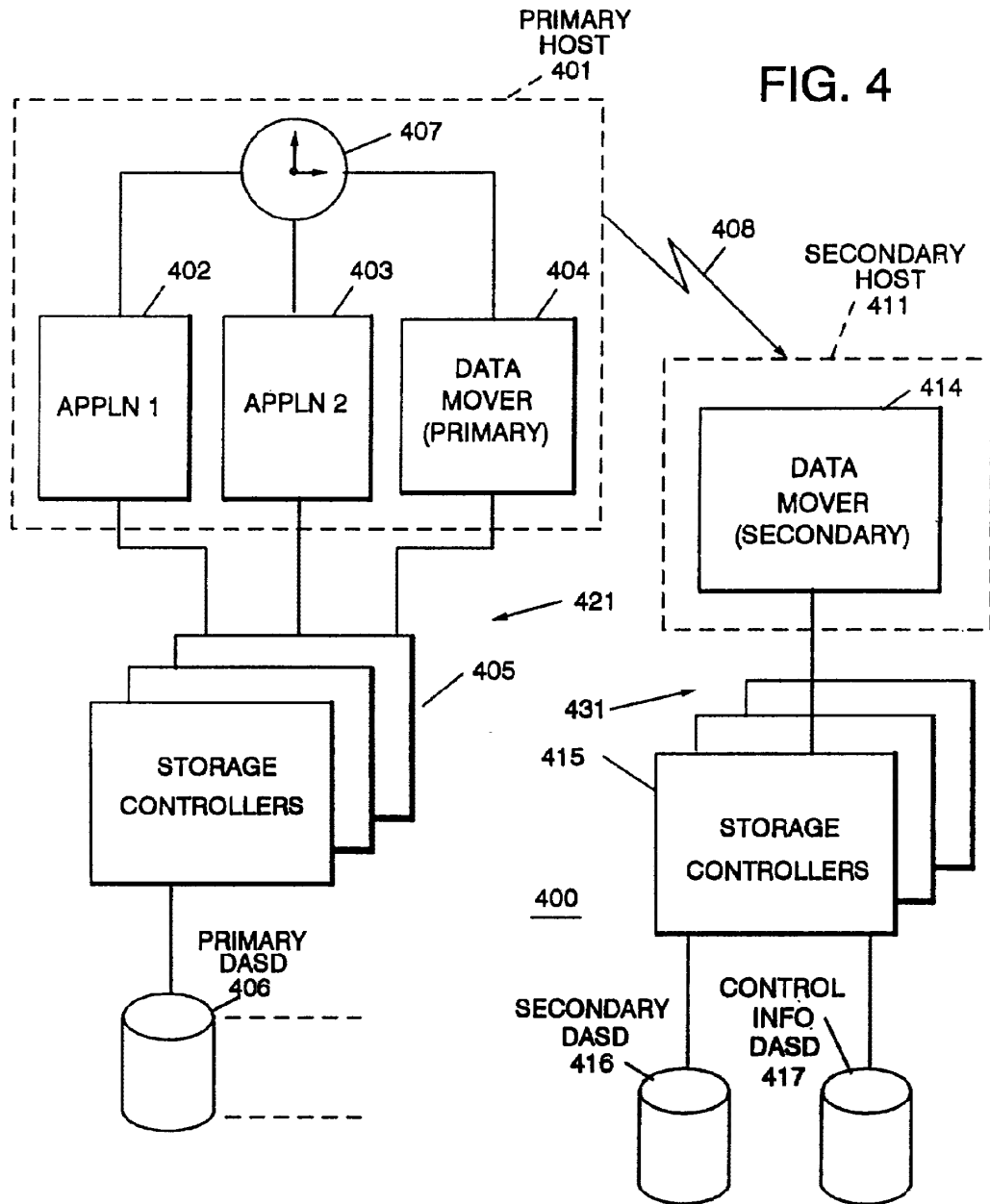
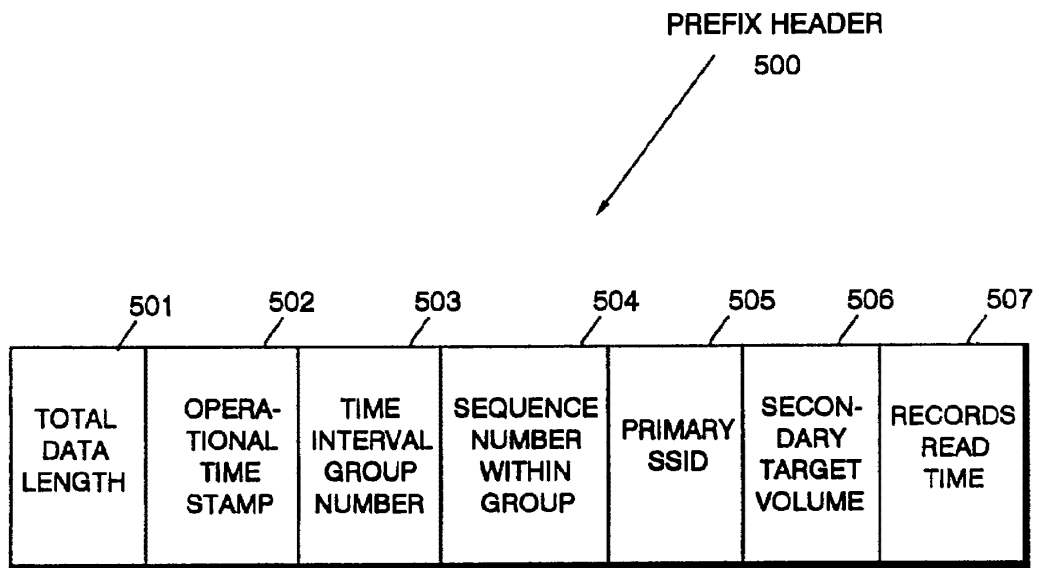


FIG. 5



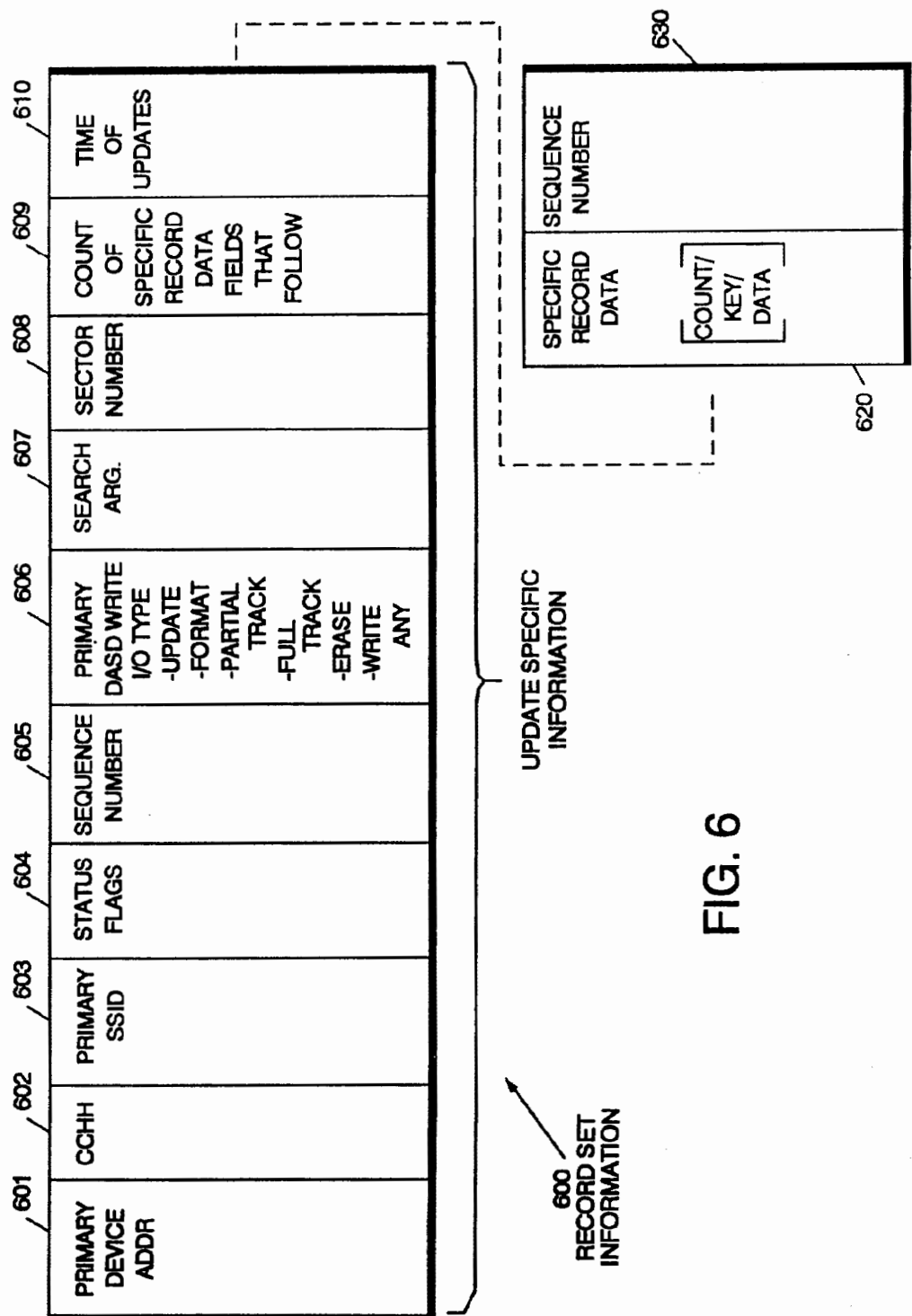


FIG. 6

FIG. 7

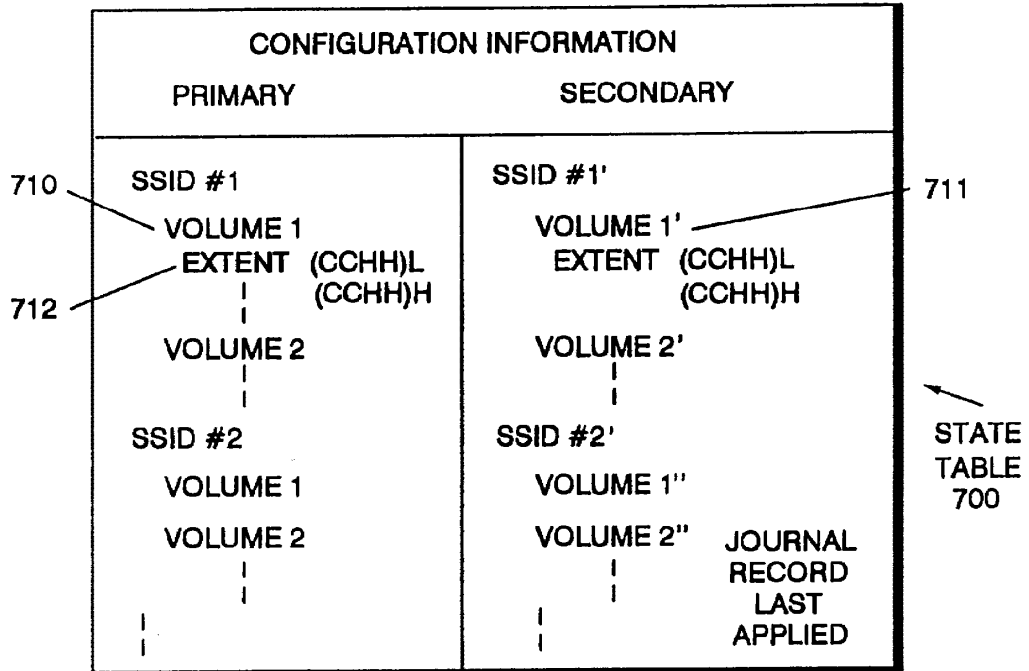
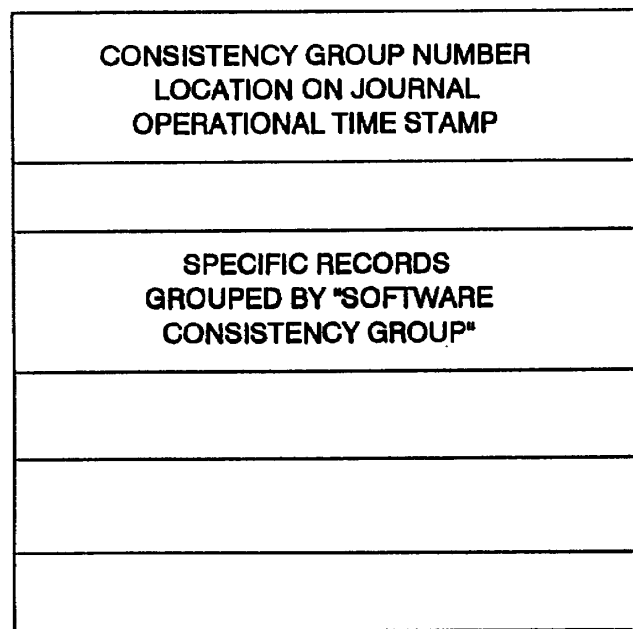


FIG. 8

800
MASTER
JOURNAL

FIG. 9

PHYSICAL CONTROL- LER ID	OPER- ATIONAL TIME STAMP	TIME INT GROUP #	READ RECORD SET TIME OF UPDATE / CONTROLLER		
			SEQ. 1 OF 3	SEQ. 2 OF 3	SEQ. 3 OF 3
SSID1	T ₁	G ₁	11:59 (2)	12:00 (5) *	12:01 (6)
SSID2	T ₁	G ₁	12:00 (4)	12:02 (7) *	—
SSID3	T ₁	G ₁	11:58 (1)	11:59 (3)	12:02 (8) *
SSID1	T ₂	G ₂			
SSID2	T ₂	G ₂			
SSID3	T ₂	G ₂			
	T ₃	G ₃			

CONSISTENCY GROUP #1

(1) 11:58 (2) 11:59 (3) 11:59 (4) 12:00 (5) 12:00 (6) 12:01	EARLIEST OPERATIONAL TIME T₁, EARLIEST TIME OF UPDATE ACROSS SSID, ORDERED IN READ SEQUENCE MIN TIME OF THE MAX TIMES OF UPDATE ACROSS ALL SSIDS
--	--

CONSISTENCY GROUP #2

(7) ----- (8) -----	
----------------------------	--

FIG. 10

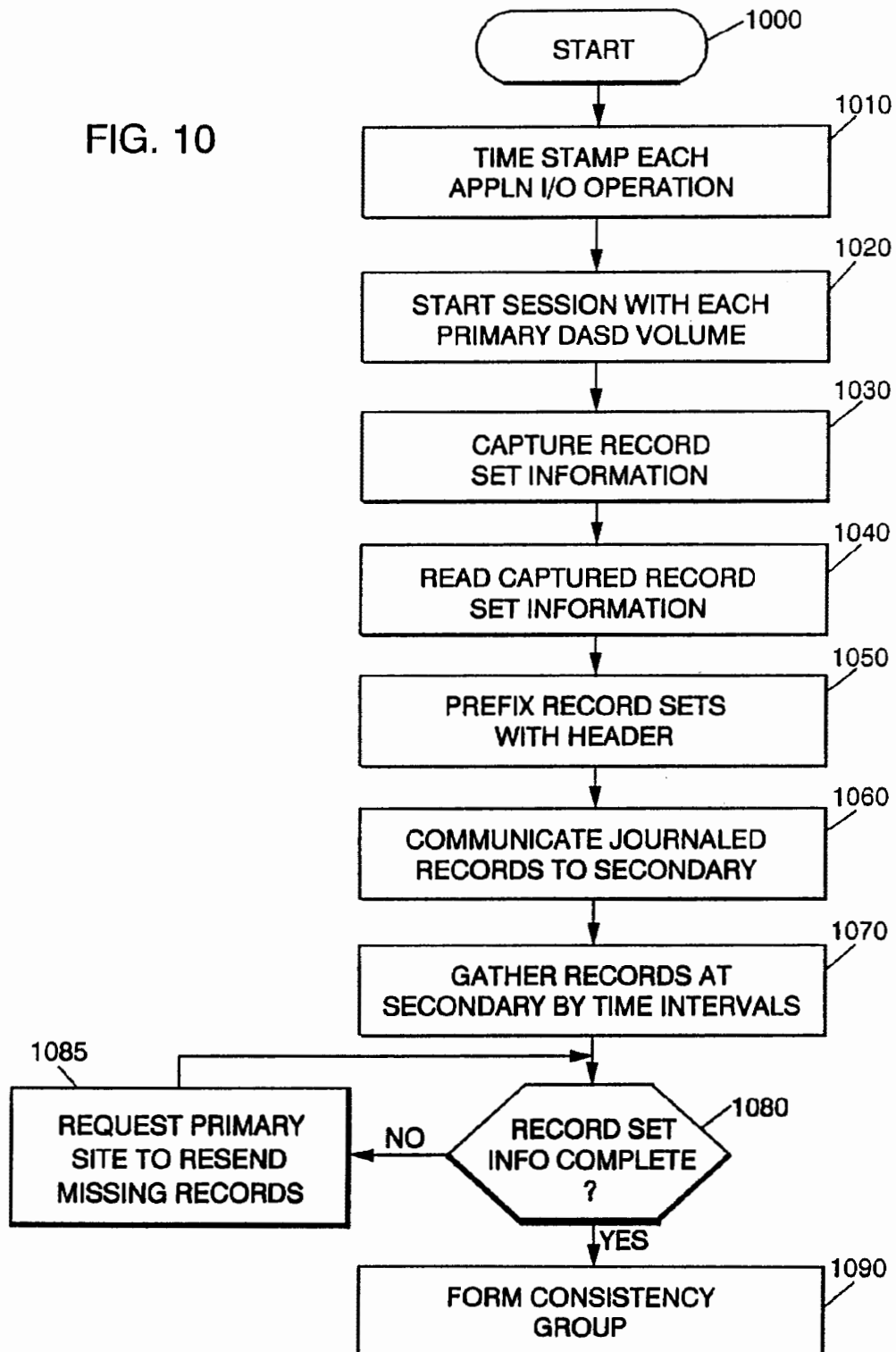


FIG. 11

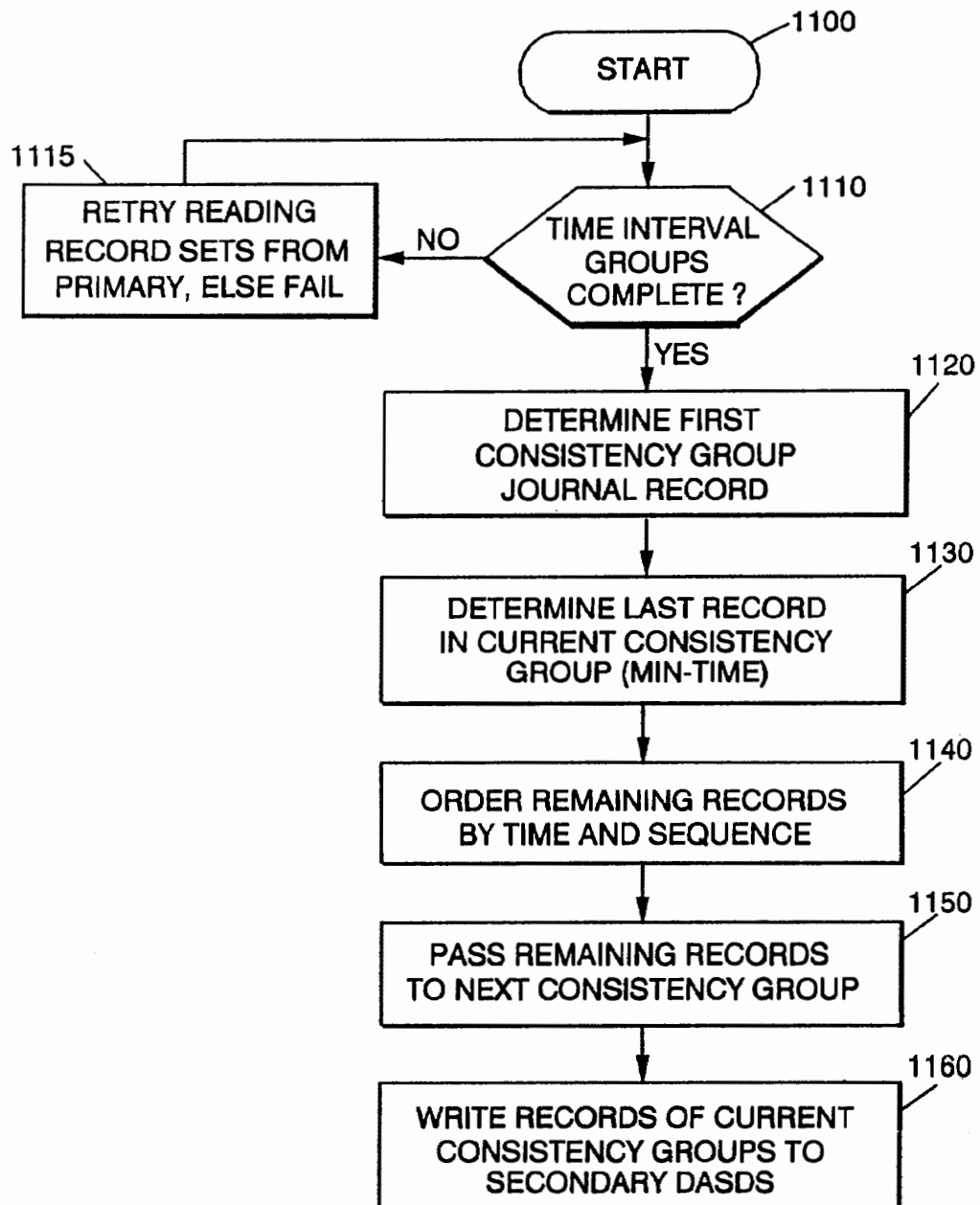


FIG. 12
FULL CONSISTENCY GROUP RECOVERY RULES

READ RECORD SET BUFFER #2		READ RECORD SET BUFFER #1									
TYPE I/O WRITE OPERATION											
	UPDATE WRITE KL = 0	UPDATE WRITE KL ≠ 0	FORMAT WRITE FULL	FORMAT WRITE PARTIAL	ERASE FULL	ERASE PARTIAL	WRITE ANY KL = 0	WRITE ANY KL ≠ 0			
UPDATE WRITE KL = 0	W*	E*	N	J	D	K	W	E*			
UPDATE WRITE KL ≠ 0	E*	W*	N	J	D	K	E*	W			
FORMAT WRITE FULL	T	T	R	T	R	T	R	R			
FORMAT WRITE PARTIAL	C	C	N	H	F	L	W	W			
ERASE FULL	T	T	R	R	R	T	T	T			
ERASE PARTIAL	B	B	N	M	E	G	W	W			
WRITE ANY KL = 0	W	E*	W	W	E	W	W	E*			
WRITE ANY KL ≠ 0	E*	W	W	W	E	W	E*	W			

FIG. 13

FIG.
13A

FIG.
13B

FIG. 13A

- B - IF #1'S SEARCH ARG IS HIGHER THAN THE SEARCH ARG FOR #2,
THEN THROW #1, ELSE DO BOTH.
- C - IF #1'S RECORD IS EQUAL TO OR HIGHER THAN THE FIRST RECORD
IN #2, THEN THROW #1, ELSE DO BOTH.
- D - IF #2 IS UPDATING R0, THEN DO BOTH, ELSE ERROR.
- E - ERROR (SHOULD NEVER HAPPEN).
- E*- ERROR IF #1 AND #2 ARE THE SAME RECORD. (SHOULD
NEVER HAPPEN WITHOUT A FORMAT WRITE IN BETWEEN).
- F - IF FIRST RECORD IN #2 IS R1, THEN WRITE BOTH, ELSE ERROR.
- G - IF #1'S SEARCH ARG IS EQUAL TO OR HIGHER THAN THE
SEARCH ARG FOR #2, THEN THROW #1, ELSE ERROR.
- H - IF #1'S SEARCH ARG IS HIGHER THAN THE LAST RECORD
FOR #2. THEN THROW #1. ELSE IF #2'S SEARCH ARG IS
HIGHER THAN THE LAST RECORD IN #1, THEN ERROR, ELSE
WRITE BOTH.

TO OPTIMIZE FURTHER, CAN DO THE FOLLOWING. INSTEAD:

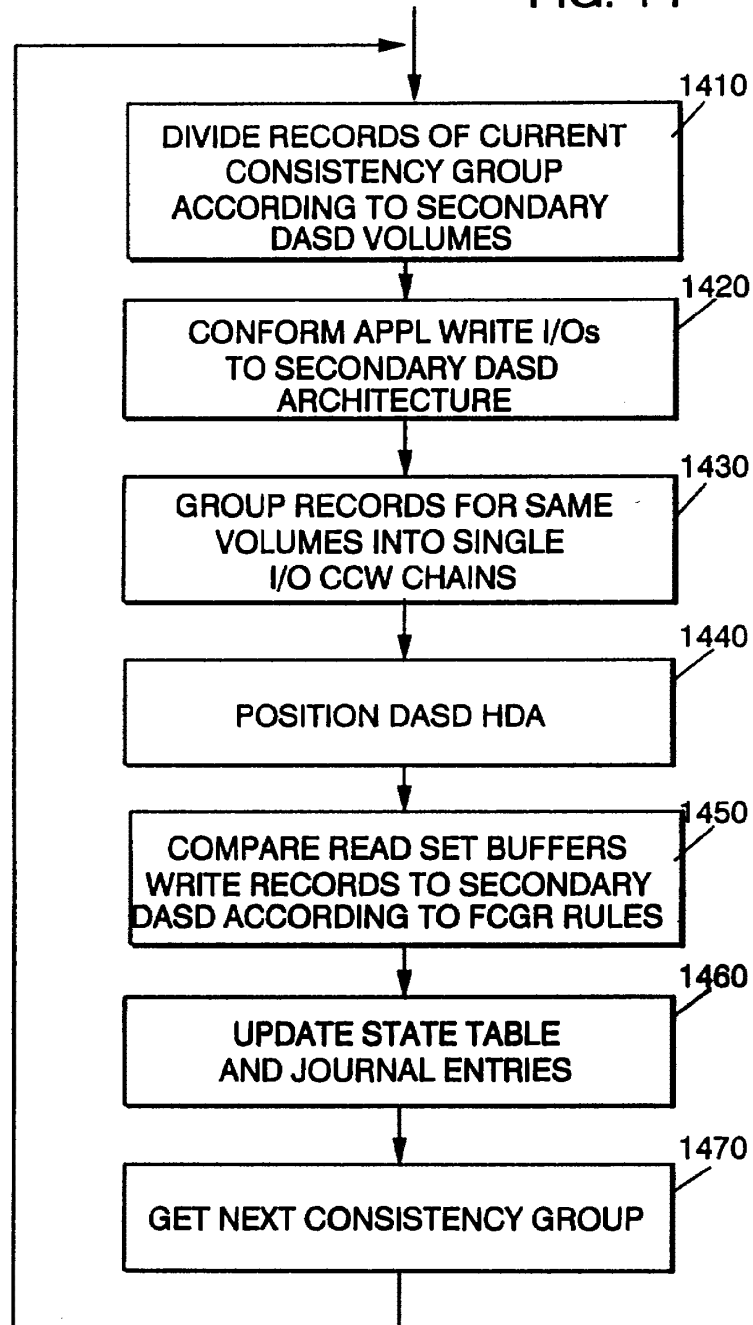
IF (1'S SEARCH IS EQUAL TO OR HIGHER THAN THE LAST RECORD
IN #2) OR (THE LAST RECORD IN #1 IS EQUAL TO OR
HIGHER THAN THE LAST RECORD IN #2) AND (2'S SEARCH
IS LESS THAN OR EQUAL TO THE LAST RECORD IN #1)
THEN THROW 1
ELSE IF (#2'S SEARCH ARG IS HIGHER THAN THE LAST RECORD
IN #1)
THEN ERROR
ELSE WRITE BOTH



- J - IF #2'S RECORD (OR SEARCH) IS HIGHER THAN THE LAST RECORD IN #1, THEN ERROR, ELSE WRITE BOTH.
- K -IF #2'S RECORD (OR SEARCH) IS HIGHER THAN #1'S SEARCH, THEN ERROR, ELSE WRITE BOTH.
- L -IF #1'S SEARCH ARG IS EQUAL TO OR HIGHER THAN #2'S SEARCH ARG THEN EITHER WRITE BOTH OR OK TO THROW #1, ELSE ERROR.
- M -IF (#1'S SEARCH ARG IS EQUAL TO OR HIGHER THAN #2'S SEARCH ARG) THEN THROW 1
ELSE IF (#2'S SEARCH ARG IS HIGHER THAN THE LAST RECORD IN #1)
THEN ERROR
ELSE WRITE BOTH.
- N -IF #2'S SEARCH ARG IS HIGHER THAN THE LAST RECORD IN #1,
THEN ERROR, ELSE WRITE BOTH.
- R -OK TO THROW #1.
- T -MUST THROW 1.
- W -WRITE BOTH.
- W*-IF #1 AND #2 HAVE THE SAME RECORDS, THEN THROW #1, ELSE
DO BOTH OR MERGE RECORDS AND DO ONE WRITE.

FIG. 13B

FIG. 14





European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 30 0753

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	WO-A-91 20034 (STORAGE TECHNOLOGY CORP) 26 December 1991 * abstract * * page 13, line 6 - page 14, line 20 * * page 18, line 34 - page 19, line 6 * * page 20, line 1 - line 35 * * figure 3 *	1, 10, 20, 21, 29	G06F11/14 G06F11/20
A	PROCEEDINGS OF THE ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMEN OF DATA, SAN DIEGO, JUNE 2 - 5, 1992, vol. 21, STONEBRAKER M, pages 236-245, XP 000393592 ANUPAM BHIDE ET AL 'AN EFFICIENT SCHEME FOR PROVIDING HIGH AVAILABILITY' * abstract * * page 237, left column, line 36 - right column, line 52 *	1, 11, 20, 29	
			TECHNICAL FIELDS SEARCHED (Int.Cl.6)
			G06F
The present search report has been drawn up for all claims			
Place of search BERLIN		Date of completion of the search 27 June 1995	Examiner Masche, C
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document</p>			

EPF FORM 1503 (11/97) (PUBLISHED)



(19)

Europäisches Patentamt
European Patent Office
Office européen des brevets



(11)

EP 1 111 527 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
22.03.2006 Bulletin 2006/12

(51) Int Cl.:
G06Q 90/00^(2006.01) H04L 29/06^(2006.01)
G06F 17/30^(2006.01)

(21) Application number: 00310244.9

(22) Date of filing: 17.11.2000

(54) **Communication method for data synchronization processing and electronic device therefor**
Kommunikationsverfahren für Datensynchronisierung und elektronische Vorrichtung dafür
Procédé de communication pour la synchronisation de données et dispositif électronique à cet effet

(84) Designated Contracting States:
DE FR GB

(30) Priority: 19.11.1999 JP 33023599

(43) Date of publication of application:
27.06.2001 Bulletin 2001/26

(73) Proprietor: KABUSHIKI KAISHA TOSHIBA
Kawasaki-shi,
Kanagawa-ken 210-8572 (JP)

(72) Inventors:
• Kizu, Sachio,
Kabushiki Kaisha Toshiba
Tokyo 105-8001 (JP)
• Natsubori, Shigeyasu,
Kabushiki Kaisha Toshiba
Tokyo 105-8001 (JP)
• Imai, Toru,
Kabushiki Kaisha Toshiba
Tokyo 105-8001 (JP)

- Amano, Shinya,
Kabushiki Kaisha Toshiba
Tokyo 105-8001 (JP)
- Muranaga, Tetsuro,
Kabushiki Kaisha Toshiba
Tokyo 105-8001 (JP)
- Takabatake, Yoshiaki,
Kabushiki Kaisha Toshiba
Tokyo 105-8001 (JP)

(74) Representative: Midgley, Jonathan Lee
Marks & Clerk
90 Long Acre
London WC2E 9RA (GB)

(56) References cited:
EP-A- 0 926 608

- LI GONG: "ENCLAVES: ENABLING SECURE COLLABORATION OVER THE INTERNET" IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE INC. NEW YORK, US, vol. 15, no. 3, 1 April 1997 (1997-04-01), pages 567-575, XP000683948 ISSN: 0733-8716

EP 1 111 527 B1

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] The present invention relates to a communication method for synchronously processing data between a plurality of devices and an electronic device therefor. In particular, the present invention relates to a communication method and an electronic device that have been modified so as to synchronize data in a wireless or wired network such that a network configuration is dynamically changed.

[0002] In recent years, apart from a personal computer, in a portable information processing terminal such as a portable cellular phone, a palm top computer or a wrist watch, personal data such as a telephone number (address) directory, schedule data, a task list, an E-mail can be managed and stored. Using this function, data is shared by using duplicate data in order to use the same data at home or outside as well as in office between electronic devices such as a personal computer or a portable information processing terminal. As a technique for keeping coincidence of a plurality of shared data, a data synchronization processing is known. For example, in the case where an attempt is made to share personal data such as an address directory or a schedule between a personal computer and a portable information processing terminal, duplication of data is created, and original data is held in the host and the duplicate data is held in each device other than the host. In the case where a change occurs with these items of data, the changed data is exchanged between these devices, whereby data consistency is maintained. This method is referred to as data synchronization.

[0003] As a typical computer program for data synchronization, Intelli Sync (Intelli Sync User's Guide Rev 3.1 Puma TECHNOLOGY 1999) is known. This enables data synchronization between personal computers or between a personal computer and a portable information processing terminal. This program enables data synchronization between two personal computers, one personal computer and one portable information processing terminal, or one personal computer and two portable information processing terminals.

[0004] In the case of this system, however, a device serving as a server must be participated in data synchronization processing. In the case where three or more devices share data, there applies a restriction that data synchronization cannot be performed between two devices other than server. Therefore, for example, in the case where a personal computer connected to a network is used as a server, thereby causing two portable information processing terminals to include the duplicate data, even if an attempt is made to use duplicate data after updated in one portable information processing terminal by the other portable information processing terminal outside, data synchronization processing cannot be performed by these portable information processing terminals other than server. Stated another way, the conventional data synchronization processing is performed be-

tween the server and the client in a one-to-one manner. It is not possible to perform the data synchronization processing in a one-to-plural manner.

[0005] Furthermore, in general, it is required to provide communication settings for a data synchronization party, settings regarding type of synchronization data, or setting of a method of solving a problem during data conflict. These settings are conventionally provided by a user operating an input device of a personal computer or portable information processing terminal through the setting screen displayed on a display device of the personal computer or portable information processing terminal and inputting data. This means that at least the display device and input device of the portable terminal device are required. Therefore, in the conventional method, because of a restriction specific to a device such as a wrist watch including device size, for example, it is very difficult to provide settings of data synchronization in a device having only a limited input device or display device.

[0006] Accordingly, it is an object of the present invention to provide a communication method and electronic device that enables data synchronization between arbitrary devices without requiring a server.

[0007] EP-A-0926608 discloses a method of maintaining mutual consistency between copies of data stored at nodes in a network, the nodes comprising a server node, and a number of client nodes at remote locations, such as laptop computers linked by cellular telephone connections, the nodes being connected intermittently.

[0008] Following a reconnection between the remote client node and another of the nodes, timestamp information on objects cached at the other node, is sent to the remote client node. At the remote client node, the timestamp information is compared with corresponding information relating to the local copy, to determine if each of the cached objects is more up to date than the local copy. The cached objects are updated based on the local copy, according to the result of the comparison. This peer to peer resynchronization means that the server need not monitor and control the process, so the system is more easily scaled up to handle more clients.

[0009] The document "Enclaves enabling secure collaboration over the internet", Li Gong, IEEE journal of selected areas in communications, IEEE Inc. New York, vol. 15, n. 3, 1 April 1997, discloses a group of collaborators which can dynamically form a secure virtual subnet wherein priorities are assigned to the group members for secure co-edition of files.

[0010] The invention provides a data synchronization method as defined in Claims 1 and 22, and an electronic device and a slave device as defined in Claims 12 and 13 respectively.

[0011] Advantageous embodiments thereof are defined in the respective dependent claims.

[0012] According to the present invention, there is provided a communication method for synchronising data between a plurality of arbitrary devices using a wireless or wired network, the communication method comprising

a member registration step and a data synchronisation step. In the member registration step, a device specified as a master by a user transmits to a device specified as a slave by the user a certificate indicating that the slave belongs to a data synchronisation group that performs a data synchronisation processing and a priority to be used when the slave performs the data synchronisation processing, and then, the slave is registered as a member of the data synchronisation group. In the data synchronisation step, it is determined using the certificate whether or not one slave and one or more other slaves capable of making wireless or wired communication belong to the same data synchronisation group, and the data synchronisation processing is performed through wireless or wired communication in accordance with the priority between the slaves which are determined to belong to the same data synchronization group.

[0013] In this communication method, apart from the data synchronization step of actually processing data synchronization, the member registration step is provided for the purpose of specifying a target device for data synchronization. At this member registration step, the certificate and priority are transmitted from a device specified as the master to a device specified as the slave, whereby the slave can be registered as a member of the data synchronization group. The certificate and priority are provided to each member, whereby data synchronization can be easily performed between slaves even if the master is absent, as long as these devices are registered in the same data synchronization group.

[0014] In this case, as to whether or not the slaves capable of making wireless or wired communication belong to the same group, it is determined by exchanging the certificate between the slaves. Thus, even if a master does not exist, data synchronization can be performed reliably and safely between the slaves. In the case where a conflict occurs during data synchronization processing between slaves, such conflict can be solved by the priority assigned to each slave at the member registration step. Therefore, each slave may hold only the certificate and priority assigned to each slave itself, making it unnecessary to set data or the like concerning a communication party for data synchronization. Hence, a device having only its limited input device and display device can be used as a device targeted for synchronization. Further, as in a wired or wireless network such that a network configuration is dynamically changed, even in a case of use under environment that makes it difficult to provide data for statically specifying a communication party for data synchronization, a device that is such data synchronization party can be reliably specified.

[0015] The data synchronization step can be performed between one master and one or more arbitrary slaves having a certificate indicating that these slaves belong to the same group as well as between slaves.

[0016] In addition, there is no need to specify a master and a slave by devices, and an arbitrary device can be specified as a master or slave of the data synchronization

group by data targeted for data synchronization. One device may function as a slave of the data synchronization group for one set of data or may function as a master of the data synchronization group for another set of data. Alternatively, one device may function as a slave of two data synchronization groups corresponding to two sets of different data.

[0017] In the case where a master and a slave are thus specified individually by data targeted for synchronization, the member registration step is also executed between a device specified as a master of a data synchronization group for the data targeted for data synchronization processing and a device specified as a slave of the master. Thus, for transmission of the certificate and priority as well, a master that is a transmission source and a slave that is a transmission destination are set by data, and the certificate and priority are set by data accordingly.

[0018] In addition to the aforementioned member registration step and data synchronization step, a master privilege transfer step may be provided, whereby only the master's function can be transferred to another device while the function and configuration of the current data synchronization group are maintained intact. Hence, a newly purchased device can be used as a master, and a slave can be temporarily used as a master outside.

[0019] In addition, it is preferable that a priority exchange step of exchanging a priority between slaves be further provided. In this manner, the priority registered in the member registration step can be changed later, and data synchronization processing can be achieved more flexibly between slaves.

[0020] The master privilege transfer step and priority exchange step as well are preferably performed after it has been determined that there does not exist any device other than two devices set in a mode for executing the above steps.

[0021] In order to easily change a member of the data synchronization group or the like, it is preferable to further provide a member registration releasing step of releasing a master or a slave from the data synchronization group. The slave releasing step may be performed by that slave device solely. However, the master releasing step is desirably executed together with the member registration releasing step of a slave communicable with the master. In this manner, a plurality of devices including a master can be released from the data synchronization group with one operation.

[0022] In addition, a conventional data synchronizing method assumes one-to-one cable connection, one-to-one connection with the server by modem, or one-to-one connection with the server by infrared-ray radiation. This is because a communication party for data synchronization can be uniquely specified.

[0023] However, if the data synchronization is performed in a bus type wired network or a wireless network free of directivity, unlike a case of one-to-one connection,

it is required to uniquely identify a party for data synchronization from the devices in a network by any method. In addition, in the case of a wireless or wired network such that a network configuration is dynamically changed, the network address of each device is not fixed, and a network address is often assigned during network connection or the like. Thus, the network address of a communication party cannot be used as static identification information for specifying a communication party for data synchronization. Therefore, in a wire or wireless network such that a configuration is dynamically changed as well, it is required to introduce a new technology for making it possible to uniquely identify a party for data synchronization.

[0024] Although it is not presumed in one-to-one connection, in a bus type wired network or a wireless network free of directivity, there is a possibility that a device other than device for data synchronization, for example, a malicious third party exists in the same network, especially in a wireless network. Thus, it is required to ensure data security, and at the same time, to prevent data leak due to computer hacking or the like. In particular, in a wireless network, there is high possibility that a device is merely disposed within the reach of wireless communication, whereby the device can be easily participated in the network. Thus, data security is important. In addition, without maliciousness, there is a danger that data contained in a device other than that targeted for synchronization, the device existing within the reach of wireless communication, is mistakenly or unintentionally changed by data synchronization processing between devices via wireless communication.

[0025] According to the present invention, such an undesired data synchronization is prevented from being occurred since the party for data synchronization can be easily identified by exchanging the certificate between the devices.

[0026] This summary of the invention does not necessarily describe all necessary features so that the invention may also be a sub-combination of these described features.

[0027] The invention can be more fully understood from the following detailed description when taken in conjunction with the accompanying drawings, in which:

FIG. 1 is a view illustrating a member registration step and a data synchronization step that are employed in a communication method according to one embodiment of the present invention;

FIG. 2 is a view showing an example of data set of each device employed according to the present embodiment;

FIG. 3 is a block diagram depicting the configuration of each device employed according to the present embodiment;

FIG. 4 is a view showing an example when functions provided to each device employed according to the present embodiment are classified by communication mode;

FIG. 5 is a view showing master and slave data exchange during data synchronization member registration according to the present embodiment;

FIG. 6 is a flow chart showing procedures for master processing during data synchronization member registration according to the present embodiment;

FIG. 7 is a flow chart showing procedures for slave processing during data synchronization member registration according to the present embodiment;

FIG. 8 is a view showing an example of communication mode during certificate data exchange and synchronizing process in the present embodiment;

FIG. 9 is a view showing another example of communication mode during certificate data exchange and synchronizing process in the present embodiment;

FIG. 10 is a view showing an example of processing at the time of data conflict in the present embodiment;

FIG. 11 is a flow chart showing procedures for master declaration processing in the present embodiment;

FIG. 12 is a flow chart showing procedures for slave processing in data synchronization member release processing according to the present embodiment;

FIG. 13 is a flow chart showing procedures for master processing in data synchronization member release processing according to the present embodiment;

FIG. 14 is a view showing master and slave data exchange in master privilege transfer processing according to the present embodiment;

FIG. 15 is a flow chart showing procedures for master processing during master privilege transfer processing according to the present embodiment;

FIG. 16 is a flow chart showing procedures for slave processing during master privilege transfer processing according to the present embodiment;

FIG. 17 is a view showing data exchange between devices during priority exchange in the present embodiment;

FIG. 18 is a flow chart showing procedures for device processing in priority exchange processing according to the present embodiment;

FIG. 19 is a timing chart showing set mode and execute mode applied to the present embodiment;

FIG. 20 is a view showing an example of an interface of a wrist watch to which the communication method according to the present embodiment is applied; and

FIG. 21 is a view showing an example of an interface of a portable information terminal to which the communication method according to the present embodiment is applied.

[0028] A preferred embodiment of a communication method for data synchronization processing according to the present invention will now be described with reference to the accompanying drawings.

First Embodiment

[0029] Hereinafter, an embodiment of a method and apparatus for synchronously processing data according to the present invention will be described with reference to the accompanying drawings.

Data Synchronization Group

[0030] Data synchronization denotes processing for, in the case where part of duplicated data is asynchronously updated independently relevant to plural sets of the duplicated data, entirely synchronizing the plural sets of these duplicated data. For example, assume that a set of data (data set) that a device A has is duplicated in a device B. Synchronization processing corresponds to processing for synchronizing data sets of the devices A and B each other. If a partial data field of the data set that the device A has is changed, a partial data field of the data set that the device B has is updated during next synchronization processing.

[0031] If the same data field of the data sets of the devices A and B is changed during and after the previous synchronization processing, in order to synchronize these data fields, it is required to select which of the data field values has a higher priority. When the data in the same data field is changed with a plurality of devices during or after synchronization processing, it is referred to as data conflict.

[0032] In the present invention, unlike a conventional method, data synchronization processing between information devices is not performed by one-to-one processing between a server and a client. This processing is performed based on a data synchronization group constructed exclusively for synchronization processing. Of the information devices capable of making data communication wirelessly or using a wire, a group configured between devices that performs data synchronization is referred to as a data synchronization group (or network). Namely, a data synchronization group composed of devices that should perform data synchronization processing corresponds to a data synchronization group. In this case, such data synchronization group may not be always configured by devices, and an arbitrary data synchronization group can be configured by each data set that enables data synchronization. Namely, each information device can configure the data synchronization group by the number of data sets that can be managed.

[0033] In addition, information devices may be a device employed for limited use such as wrist watch or may be a device having a plurality of functions such as personal computer. These information devices are not restricted by their sizes or the types or functions of input/output devices or display devices and the like. The data synchronization group is configured by a plurality of information devices. A device configuring one data synchronization group is referred to as a member of the data synchronization group.

[0034] A data synchronization group is a permanent concept. Data synchronization processing is performed by a member of the data synchronization group that is configured in advance, unlike a case in which a data synchronization group is configured every time data synchronization processing is performed.

Data Synchronization Group Communication

[0035] A communication method in a data synchronization group comprises a member registration step and a data synchronization step. FIG. 1 shows an example.

[0036] The member registration step is a step to be performed when a member is added to a data synchronization group. At this member registration step, a certificate and a priority are transmitted from a device specified as a master by a user to a device that the user want to newly register as a slave. The certificate indicates that a device receiving it belongs to the same data synchronization group (data synchronization network) as the master. That is, the certificate corresponds to a group ID or the like. The priority is used when a device receiving it performs data synchronization processing, and is employed to solve data conflict.

[0037] The data synchronization step is a step that performs data synchronization processing between a plurality of arbitrary members configuring a data synchronous group.

[0038] In FIG. 1, M denotes a master of a data synchronization group. The master M is a core of the data synchronization group. Only one master exist in one data synchronization group, and has a privilege of adding a member to the data synchronization group. S₁ to S₄ denote slaves of the data synchronization group. One or more slaves exist in one data synchronization group. Data synchronization member registration is performed between the slave and the master M in the data synchronization group, whereby the slave becomes a member of the data synchronization group.

[0039] A master and a slave may not be always specified by devices, and an arbitrary device can be specified as a master or slave of the data synchronization group corresponding to the data sets. Thus, one device may function as a slave of the data synchronization group relevant to one set of data or may function as a master of the data synchronization group relevant to another set of data. Alternatively, one device may function as each of the slaves of the two data synchronization groups corresponding to two different types of data.

Member Registration Step

[0040] The member registration step is executed by making one-to-one communication between a device that is a master of a data synchronization group and a slave that newly participates in the data synchronization group. Two members configure the data synchronization group at a stage at which a first slave S₁ is registered in

the master M. Subsequently, the member registration step is sequentially performed similarly between the master M and each of the slaves S_2 to S_4 to be newly registered, whereby the number of members configuring the data synchronization group increases one by one. The member registration step is not performed every data synchronization. Once a device is registered as the member, such registered device is continuously a member of the data synchronization group unless the device has been released explicitly. Data synchronization processing can be executed any time between the members in the same data synchronization group.

Data Synchronization Step

[0041] At the data synchronization step, data is synchronized between the members that belong to a data synchronization group. Here, all the members may perform data synchronization processing at one time or arbitrary members may perform data synchronization processing. In addition, the master M may be or may not be participated in data synchronization processing. According to the present invention, data synchronization processing can be performed between slaves. A plurality of synchronizing processes configured by other members can be performed independently at an arbitrary time, and a plurality of these synchronizing processes may be performed simultaneously in another place.

[0042] For example, as shown in FIG. 1, in the case where only the master M and the slave S_1 configure a data synchronization group, the data synchronization step is executed between these two devices, namely between the master M and the slave S_1 . Alternatively, in a state in which a slave S_2 is newly registered, the data synchronization step can be executed between the master M and slave S_1 , between the master M and slave S_2 , between the slave S_1 and the slave S_2 , or between the master M and the slaves S_1 and S_2 . Hereinafter, the synchronization step is executed similarly between arbitrary two devices irrespective of the master or slave.

Member Configuration

[0043] Now, the data and functional structures set to each of the master and slave devices will be described with reference to FIG. 2 and FIG. 3.

[0044] Set data for setting the data synchronization group that each device stores, includes master/slave specification data, identification data, certificate data, priority data, and a counter value.

[0045] The master/slave specification data is like a flag, and its initial value is blanked. When a device is registered as a master or slave, the corresponding flag is set.

[0046] The identification data denotes data for uniquely specifying a member in one data synchronization group, and is used as a key for generating certificate data. The synchronization group is defined by data type,

and thus, the identification data differs depending on type of data set in one device. The identification data is generated in each device when one device is registered as a slave of a data synchronization group or when a device is operated as a master by a master declaration described later.

[0047] The certificate data corresponds to the aforementioned certificate, and is employed to certify that one device belongs to one data synchronization group.

[0048] For example, during data synchronization assume that it is authenticated by a public key encryption system as to whether or not devices belong to the same data synchronization group and encryption of synchronization data after authenticated. The identification data and the certificate data are employed as follows in an authentication and encryption system.

[0049] First, each of the master and slave generates a secret key and a public key. A pair of these secret key and public key corresponds to the identification data of each device. When the identification data is transmitted to another device, only the public key that is part of the identification data is transmitted. During data synchronization group registration, when a slave passes a public key as the identification data to a master, the master returns this public key of the slave as a certificate after the public key has been encrypted by own secret key. In this manner, the slave has own secret key and public key; and a certificate certified by the master and the public key of the master. Here, this certificate and the public key of the master are referred to as certificate data.

[0050] During actual authentication and encryption, when a device performing authentication receives a certificate from the counterpart, this certificate is decrypted by own public key of the master. If the certificate is correctly decrypted, it means that the device of the counterpart belongs to the same data synchronization group. Then, the public key of the counterpart is extracted from the decrypted data, and the synchronization data is encrypted by the extracted public key of the counterpart, and is transmitted, whereby synchronization data can be transmitted safely.

[0051] The certificate data may certify reliably that each device belongs to a data synchronization group. The data may be generated by other well known various methods without being limited to the aforementioned method.

[0052] The priority data is used to solve conflict of data synchronization as described previously. The data is used to determine a precedence (namely, priority) for solving such data conflict.

[0053] The counter value is data used to count the number of slaves registered by the master and determine the priority of that slave. The highest priority of the master is "1". The counter value is counted up every time a slave is registered, and the priorities "2", "3", ... are assigned to the slaves in order of registration. This counter is used by the master only.

[0054] As described above, the data synchronization

group is configured by each data set. Thus, the master/slave specification data, certificate data, identification data, priority data, and counter value are provided and managed by each data set. That is, in a device having a plurality of data sets, the certificate data, identification data, priority data, and counter value are not managed in one-dimensional manner, and are managed in two-dimensional manner, namely, by each data set, as shown in FIG. 2. FIG. 2 shows an example when a device belongs to three data synchronization groups corresponding to three types of data sets, i.e., "address directory", "time", and "E-mail", respectively. In this case, in a device having these three types of data sets, whether the device operates as a master or slave by each data set is managed. The counter value is valid only for a data set of the data synchronization group that operates a master. In addition, the aforementioned member registration step is performed by data synchronization group, the certificate data, identification data, priority data, and counter value are independent of each other by each data set, respectively. The secret key and public key configuring identification data are generated by the member registration step by employing random numbers, for example. The value of the identification data is continuously maintained until it has been removed from the data synchronization group in which a device is registered as a member at the member registration step.

[0055] In this manner, in the case where set data on a data synchronization group is managed by each data set, when one performs each of the functions including data synchronization member registration, data synchronization processing, master declaration, data synchronization member releasing, master privilege transfer, and priority exchange, a step is executed such that a user can select a target data set from among the data sets in the row direction shown in FIG. 2 in order to specify what is a data set targeted for processing.

[0056] FIG. 3 shows the functional configuration of each device. This network configuration is provided as an example when a device can be configured as a master and a slave.

[0057] As shown in FIG. 3, each device includes a communication device 11, a communication controller 12, a device input/output device 13, a device input/output controller 14, an authentication unit 15, a data synchronization group setting and management unit 16, a data synchronization processing unit 17, and a synchronous data storage device 18.

[0058] The communication device 11 makes communication with another device via a wireless or wired communication. This device 11 transmits data to another device, and receives data from such another device. As a wireless communication, there can be used an electric wave communication, free of directivity such as "Bluetooth", for example. In addition, a bus type LAN such as "Ethernet" can be used as a wired network. In either of the wireless and wired communication, the network configuration may not be statically determined. The present

invention is applicable to a network such that the network configuration is dynamically changed.

[0059] The communication controller 12 controls data reception and/or transmission. The controller 12 includes a function for checking data and a request received by the communication device 11; a function for transmitting and receiving a processing request and response from the device input/output controller 14; a function for making a data transmission and reception request for the communication device 11; and a function for transmitting and receiving a request and a response among the authentication unit 15, data synchronization group setting and management unit 16, and data synchronization processing unit 17, respectively.

[0060] The device input/output device 13 is a user interface, and is an input device or a display device of own device.

[0061] The device input/output controller 14 controls data request and transmission and/or reception relevant to the device input/output device 13. The controller 14 includes a function for checking data and a request received by the device input/output device 13; a function for making a processing request for the communication controller 12; a function for receiving the processing request from the communication controller 12; a management function for writing, reading, deleting data relevant to the synchronization data storage device 18; and a function making a request for and a response to the data synchronization group setting and management unit 16.

[0062] The authentication unit 15 authenticates certificate data received from another device in response to an authentication request from the communication controller 12 by correlating the data with certificate data of the own device, the data being owned by a certificate data management controller 161 through the data synchronization group setting and management unit 16. The unit 15 determines whether or not another device belongs to its data synchronization group, i.e., whether or not both of the own and other devices belong to the same data synchronization group.

[0063] The data synchronization group setting and management unit 16 generally manages settings of the data synchronization group. The unit 16 includes a certificate data management controller 161 and a priority management controller 162. The data synchronization group setting and management unit 16 manages own settings by using the certificate data management controller 161 and the priority management controller 162. The unit 16 transfers data or a processing request to the communication controller 12, authentication unit 15, device input/output controller 14, data synchronization processing unit 17, priority management controller 162, and certificate data management controller 161.

[0064] The certificate data management controller 161 has a region for storing certificate data certified by a master of the own device and a region for storing identification data that is a source of the certificate data. The controller 161 manages certificate data and identification data for

generating such certificate data. The controller 161 includes functions for generating, deleting, and setting the certificate data and the identification that is a source of the certificate data according to a request from the data synchronization group setting and management unit 16.

[0065] The priority management controller 162 has a region for storing priority data on own device; a region for storing a counter value changed when a priority is issued in a master; and a counter device. The controller 162 includes a function for processing requests for generating, setting, deleting, and transferring priority data from the data synchronization group setting and management unit 16; and a function for resetting and incrementing the counter device.

[0066] The data synchronization controller 17 performs synchronization processing for a data set targeted for synchronization (synchronization data). The controller 17 includes a function for reading, writing, and deleting data relevant to the synchronization data storage device 18; a function for solving data conflict in view of the priority of own device and that of another synchronization device; and a function for processing a request from the communication controller 12 and delivering data.

[0067] The synchronization data storage device 18 stores synchronization data, and has a region for storing data.

[0068] The configuration of this device is provided as an example when a device can be configured as a master and a slave. This configuration may include a device (member) that can be configured as a slave only or a device (member) that can be configured as a master only. In this case, there can be provided a configuration in which functions or configurations required for a slave and a master are eliminated. For example, a certificate data producing function and a counter device and its control function and the like may be provided to only a device that can operate as a master, and is not required for a device that functions as a slave only.

[0069] Now, data synchronization group management and operation will be described here.

[0070] First, all functions owned by each device in a data synchronization group will be described with reference to FIG. 4.

[0071] A device that can configure a data synchronization group includes a total of six functions. The device includes a function to be processed without making communication with another device; a function requiring one-to-one communication between one device and another device; and a function requiring communication between one device to a plurality of devices (actually, made of a plurality of one-to-one communications). The "master declaration" function is a stand alone function that is processed without making communication with another device. The "data synchronization member registration", "master privilege transfer", and "priority exchange" functions are processed through one-to-one communication. The "data synchronization" function is performed between two or more devices. The "data synchronization

member releasing" function is a stand alone function in a slave, and a function that is processed through communication with one or more slaves in a master. Hereinafter, these functions will be described in detail. These functions can be performed by data synchronization group. When a function is read, it is required to specify an identification code of the data synchronization group.

(1) Data Synchronization Member Registration

[0072] In the case where a new member is added to a data synchronization group, "data synchronization member registration" processing is performed. This processing is adopted to achieve the aforementioned member registration step. The processing is performed by making data communication between a master of a data synchronization group and a member to be newly added as a slave. The master and newly registered member must be communicable with each other via a wireless or wired communication channel while power is supplied. The user calls the "data synchronization member registration" function each in the master and the member to be registered. Subsequently, each of the master and the member to be registered is placed in a member registration mode. In order to enter the member registration mode, the user interaction and exclusive control for another device are used such that the user holds down the registration button in each of the master and the new member, for example, and registration is permitted only when only two devices in the member registration mode exist, and security authentication during registration is provided. It is possible to determine whether or not only two devices in the member registration mode exist by checking a response to a broadcast from the master, for example. This prevents accidental registration of a device owned by a malicious third party within the reach of electric waves.

[0073] The effective communication range means that a device is within the reach of wireless communication in a wireless network. In a wired network, this range means a limited network area such as sub-network, for example, in which broadcasting is effective. Namely, in the range which has a possibility that a third party receives the communication, under the condition that there exists any device other than only two devices, i.e., a device targeted for member registration and a master, which are set in the member registration mode, data synchronization member registration is started.

[0074] Next, a function for registering a data synchronization member will be described in detail. In FIG. 5, there is shown master and slave communication processing to be performed together with synchronization member registration.

[0075] First, both of the master M and the slave S transmit to the counterpart a request for registering a data synchronization member, and then, the master notifies its identity to the slave. In this state, the master has all set data, but the slave has only identification data. When the slave receives the notification, it transmits the iden-

tification data to the master. The master transmits to the slave the certificate data for this slave generated using the identification data and the priority data, and communication processing is completed.

[0076] Now, the processing step in each device will be described in detail. In FIG. 6, there is shown the step of master processing when a master device registers a slave device as a member of the data synchronization group. A description is given with reference to FIG. 3.

[0077] First, when a user call of the "data synchronization member registration" function is transmitted to the device input/output controller 14 (step S11), the current setting of the data synchronization group (whether the own device is a master or slave) is determined at the data synchronization group setting and management unit 16 (step S12). If the own device is set as a master by a master declaration function described later, the user is prompted for a request for executing registration processing. When it is set as a slave, processing is terminated. In the case where nothing is set (i.e., blank data is specified), the device operates as a newly registered member described later (FIG. 7).

[0078] In the case where the own device is set as a master, when the device input/output device 13 receives a request for executing registration processing from the user (step S13), the communication device 11 transmits a request for registering a data synchronization member in a broadcasting manner (step S14). Then, the communication device 11 waits for a registration request from a new member. Although not shown, in the case where a registration request is not made for a predetermined time, processing terminates with time out. When a registration request from a new member is received (step S15), the registered device determines whether or not only two devices requires the registration, i.e., the own device and a new member (step S16). Namely, after a request for registering a data synchronization member has been transmitted between the own device and the counterpart, if only one counterpart transmits such request for registering a data synchronization member, it is determined that only two device, i.e., the own device and new member participate the registration.

[0079] In the case where any device other than only two devices, i.e., the own device and new member participate the registration (NO at the step S16), processing is terminated. In the case where only two devices participate the registration (YES at step S16), the communication device 11 notifies its identity to the new member (step S17). Then, when identification data that is a source of certificate data from the new member is received by the communication device 11 (step S18), the data synchronization group setting and management unit 16 generates the certificate data and the priority data to be passed to the new member, based on this data (steps S19 and S20). Further, the value of the counter that counts the number of registered devices is incremented by one (step S21).

[0080] Last, the communication device 11 transmits

the generated certificate data and the priority data to the new member (steps S22 and S23). Although the counter value is used as the priority data, the priority may be determined in any system as long as a precedence can be defined by generating different values every issue, thereby making it possible to uniquely judge the priority among these generated values.

[0081] Now, in FIG. 7, there is shown the step of processing a new member when a desired device (member) newly participating in a data synchronization group is registered as a data synchronization group member by a master device. A description will be described with reference to FIG. 3.

[0082] When the device input/output device 13 receives the user request for "data synchronization member registration", as in the case with FIG. 6, the current data synchronization group setting is determined by the data synchronization group setting and management unit 16. In the case where the own device is set as a master, the aforementioned "data synchronization member registration" (FIG. 6) is performed. In the case where the own device is set as a slave, processing terminates. If a data synchronization group is not set (blank data is specified as a master or slave), the user is prompted for request for executing registration processing. When the user makes a request for executing registration processing to the device input/output device 13 (step S31), the communication device 11 receives a registration request from the master (step S32), and transmits such registration request to the master (step S33). Although not shown, in the case where there exists no response from the master within a predetermined period of time, processing terminates with time out. After the registration request has been transmitted to the master, the registered device determines whether or not only two devices, i.e., the own device and master participate the registration (step S34). In the case where only the two devices participate the registration, the own device waits for the counterpart master notification. In this manner, it is determined that the counterpart is a master (step S35), the data synchronization setting and management unit 16 generates identification data that is a source of the certificate data (step S36), and transmits the data to the master (step S37).

[0083] When the communication device 11 waits for the certificate data and priority data from the master, and receives them, the certificate data and priority data are set through the data synchronization group setting and management unit 16, and registration processing completes (steps S38 and S39).

[0084] In registration processing of a data synchronization member, it is required to share a data set of the same type between the master issuing the certificate data and priority and the new member to be registered. In the case where an attempt is made to register data sets of their different types, it is required to disable registration. This collation step can be performed by checking whether or not identifiers 01, 02, and 03 are assigned in advance

to an address directory, a time, and schedule, for example, during "transmission of a request for registering a data synchronization member" (step S14) and during "transmission of a request for registering a data synchronization member" (step S33), transmitting these identifiers between the master and the new member, and checking whether or not the identifiers are identical to each other.

[0085] Once it is guaranteed that a master and a slave that belong to a data synchronization group and a slave have a data set of the same type, it is authenticated as to whether or not the master and the slave belong to the same data synchronization group when another function is used. Thus, there is no possibility that data sets of the different types coexist. Even in a device having a plurality of data sets, the data sets in that device configure, respectively, other completely independent data synchronization groups, the data sets of different types and processing do not coexist.

(2) Data synchronization processing

[0086] Now, data synchronization processing will be described here.

[0087] When a master in a data synchronization group transmits certificate data and priority data to each member by data synchronization member registration processing, actual data synchronization processing can be performed between arbitrary members (even without intervening a master) by using at least these two sets of data.

[0088] During synchronization processing, the members belonging to the same data synchronization group first authenticate their identities each other, and then, perform data synchronization. Sets of data that do not conflict with each other are merged, respectively. In the case where a conflict occurs with some sets of data, the data with the highest priority is employed as data for solving a conflict.

[0089] As shown in FIG. 8, assume that a data synchronization group is configured by five members A, B, C, D, and E, wherein the member A is a master.

[0090] In the case where data synchronization processing is performed when all the members are present, one member transmits own certificate data to another member, receives certificate data from the counterpart, and authenticates their identity with each other. In transmission and authentication of certificate data, there is provided a method for all members transmitting/receiving certificate data to/from all the counterparts, thereby individual members using their received certificate data to authenticate their identities. Namely, among the members A to E, certificate data transmission/reception and authentication may be performed according to the number of combinations of different two members.

[0091] Alternatively, as shown in FIG. 9, certificate data transmission/reception and authentication may be performed only between a member with the highest priority

(the member A here) and another member. Namely, there is provided a method in which the pairs of members such as A and B, A and C, and A and D, and A and E (the member A serves as a server), thereby performing authentication. After each of the members A to E has been authenticated to be a member of the same data synchronization group, in actual data synchronization processing, a member with the highest priority (the member A here) transmits/receives synchronization data in a data field to/from all other members, checks a conflict, solves such conflict based on the priority, and changes data so that the member can sequentially process the synchronization data in the next data field.

[0092] A conflict during data synchronization processing occurs, as shown in FIG. 10, in the case where the members A, C, and E have not been changed relevant to a data field 'x' but data owned by the members B and D is updated since previous synchronization processing. Means for solving such data conflict is to employ data owned by the member B with the highest priority. This is because, although the member A has the highest priority of the five, data is changed in the members B and D only. Even if data is not changed, a conflict during synchronization processing may be solved using the data owned by the member A with the highest priority.

[0093] Next, assume that only the members A, B, and C in this data synchronization group perform data synchronization. Such situation can occur when the members D and E are not turned ON or when only the members A, B, and C are carried to the outside, and data synchronization is performed outside. Authentication and data synchronization are performed in the same manner as described above. Then, assume that data owned by the members B and C conflict with each other (that is, assume that the data in the same data field owned by the members B and C has been updated during and after the previous synchronization). Such conflict is solved by using the data of a member having the highest priority of the members B and C. As a result, the data owned by the members A, B, and C are synchronized with each other. Assume that the remaining members D and E perform data synchronization in another place. As is the case with the members A, B, and C, if a conflict occurs, such conflict is solved by using the data with the highest priority of the members D and E, and the members D and E undergoes data synchronization.

[0094] As described above, data synchronization processing can be performed between slaves as well as between one master and one or more arbitrary slaves, each of which has a certificate indicating that they belong to the same data synchronization group that performs data synchronization processing. In this case as well, certificate data is transmitted between devices capable of making wireless or wired communication, whereby it is determined that these devices belong to the same data synchronization group. Thus, data synchronization can be performed safely by using only the certificate data without having specific data for uniquely specifying the

counterpart.

[0095] Data synchronization processing can be performed as follows, for example.

- (a) A device set in a data synchronization processing mode transmits a data synchronization processing request in a broadcasting manner to other devices that exist within a communicable range;
- (b) A certificate is transmitted between these devices, and it is authenticated as to whether or not they belong to the same data synchronization group each other; and
- (c) When the above devices belong to the same data synchronization group, data synchronization of the corresponding data sets is executed in accordance with each priority.

[0096] In the case of data synchronization processing, unlike registration processing of a data synchronization member as described in FIG. 6 and FIG. 7, there does not apply a restriction that only two devices, i.e., a master and a member targeted for new registration. Data synchronization processing can be performed between two or more arbitrary devices that exist within a communicable range provided they belong to the same data synchronization group. All the devices to be synchronized may be set in a data synchronization processing mode; only one device may be set in a data synchronization processing mode by the user, and another device receiving a data synchronization processing request from that device may automatically enter the data synchronization processing mode according to that request.

[0097] In addition, in actual data synchronization processing, the user may specify a data set targeted for data synchronization processing in the data sets for which data synchronization groups are set. Alternatively, it is determined as to whether devices belong to the same data synchronization group relevant to all the data sets for which the data synchronization groups are set, whereby these synchronizing processes are executed simultaneously relevant to the data sets which belong to the same data synchronization group. Authentication data exists by each data set for which the data synchronization group is set. Thus, all the authentication data is exchanged each other, for example, thereby making it possible to check a data synchronization group to which both of the counterpart device and the own device belong. Of course, the identifier assigned to each data set is transmitted, whereby a data synchronization group to be authenticated may be specified.

(3) Master Declaration

[0098] Now, a "master declaration" function for setting the own device as a master of a data synchronization group will be described here.

[0099] In the case where a data synchronization group is newly constructed, it is required to determine a member

as a master of the network. The "master declaration" function is provided to achieve this. The "master declaration" can provide setting such that only a member having a data set, which does not belong to any data synchronization group, can be a master of a data synchronization group that the member will construct. That is, a device that has already been set as a slave cannot be changed to a master. This setting is provided by a stand alone, and communication with another member is not performed.

[0100] In the existing data synchronization group, the "master declaration" cannot be performed by changing to a master a member that does not belong to the data synchronization group or a member that has been set as a slave or temporarily executing "data synchronization member releasing" for a slave to remove the slave from the data synchronization group, thereafter adding it as a master again. This is required to ensure security. In the latter case, there is defined another new data synchronization group in which a member performing "master declaration" is set as a master. The "master declaration" is used any time only when a new data synchronization group is constructed.

[0101] In FIG. 11, there is shown the processing step of "master declaration" in which a new member is set as a master of a new data synchronization group by the "master declaration".

[0102] When the user first makes an operation for calling the "master declaration" function for a data set through the device input/output device 13 (step S41), the data synchronization group setting and management unit 16 checks the current setting of the own device in the data synchronization group (step S42). In the case where the device has been set as a master or slave, processing is terminated. If no data synchronization group is defined (i.e., if blank data is specified as a master or slave), the data synchronization group setting and management unit 16 generates identification data on a data synchronization group (step S43), and generates authentication data based on the generated identification data (step S44).

[0103] Further, in the data synchronization group setting and management unit 16, a priority counter is reset, and the initial value of the counter is set as priority data indicative of own priority (steps S45 and S46). Then, the counter is incremented, and processing is terminated.

(4) Data Synchronization Member Releasing

[0104] "Data synchronization member releasing" is performed when a registered device is released from a data synchronization group. This operation differs depending on a master and a slave, and thus, a description will be given separately.

[0105] First, a description will be given with respect to a case in which a slave is released from a data synchronization group. This function is used when a slave participating in a data synchronization group is used while the slave is excluded from the member of the synchro-

nization group or when a slave device is discarded. The "data synchronization member releasing" is a stand-alone operation, and communication with a master or another device is not required. The existing data synchronization group is not affected by this function. The priority data owned by the released slave will be vacant because it is not reused in "data synchronization member registration". However, this does not mean that consistency of the data synchronization group is not destroyed. [0106] In FIG. 12, there is shown the processing step when "data synchronization member releasing" is performed in a slave.

[0107] First, when a user make a request for "data synchronization member releasing" for the device input/output device 13 relevant to a data set (step S51), the data synchronization group setting and management unit 16 determines the current setting of the own device in the data synchronization group (step S52). When the own device is a slave, the data synchronization group setting and management unit 16 deletes the corresponding priority data, certificate data, and identification data (steps S53, S54, and S55). This releasing is performed by each synchronization group, and thus, network identification data is also specified when the releasing function is called.

[0108] In the case where a master is released from a data synchronization group, if a slave currently communicable with the master exists when "data synchronization member releasing" is performed, this slave is also released from the data synchronization group. This is because the master generally dismembers a slave without transferring its master privilege when the master desired to release the slave from the data synchronization group. In addition, only a communicable slave is released from the data synchronization group simultaneously because it is not assumed that all members in the data synchronization group are always present by the fact that the master cannot grasp all the members in the data synchronization group and the fact the slave is placed in another place due to purchase of a new device, discarding, faults, theft or the like. In this case, a slave that does not exist on the spot still participates in the previous data synchronization group, and data synchronization can be performed between the slaves that do not exist on the spot. However, in this case, since a master does not exist any longer, a data synchronization group member cannot be increased by newly performing data synchronization member registration.

[0109] In FIG. 13, there is shown the processing step when "data synchronization member releasing" is performed in a master. When a user makes a request for "data synchronization member releasing" relevant to a data set to the device input/output device 13, as described in FIG. 12, the data synchronization group setting and management unit 16 determines the current data synchronization group. In the case where the own device is a master, this device searches for a currently communicable slave (step S61). This may be determined ac-

cording to a response from the slave relevant to data transmitted in a broadcasting manner. Then, the own device transmits a request for "data synchronization member releasing" and the master certificate data to a slave within the currently communicable range (steps S62 and S63).

[0110] The slave receiving this request performs authentication by using the master certificate data, and then, resets own setting as in "data synchronization member releasing" in the slave.

[0111] The master is then deletes the priority data, certificate data, identification data, and counter values of the own device, the counter is reset, and processing is terminated (steps S64 to S67).

(5) Master Privilege Transfer

[0112] The "master privilege transfer" function allows a data synchronization group to continuously exist intact in the existing data synchronization group, and allows the current master to transfer only the master's privilege to another member (slave). This function is used when one attempt to transfer a master privilege to another device while maintaining the functions and configuration of the data synchronization group, including cases in which where one attempts to specify a newly purchased device as a master of the existing data synchronization group; a master device is discharged; one attempts to change the current master device to a new device; and, when a desk top personal computer that is not mobile is specified as a master, one attempt to temporarily perform "data synchronization member registration" outside. This means that the slave and master privileges are exchanged in the existing data synchronization group. Alternatively, the master privilege may be transferred to a member that is not included in a data synchronization group by automatically exchange the privileges in combination with "data synchronization member registration".

[0113] In this process, the "master privilege transfer" function is called by a master and each slave device that transfers the master privilege. In a method similar to the "data synchronization member registration", the set data held by the master is transferred to a slave, and the set data held by the slave is transferred to the master. The data held by each of the master and slave is exchanged intact. Individual members do not have data concerning another device each other, and thus, these individual members do not affect those in another data synchronization group. Thus, processing can be performed safely and simply as in the "data synchronization member registration".

[0114] Now, the "master privilege transfer" function will be described in detail,

[0115] Here, master and slave communication processing to be performed together with the "master privilege transfer" processing will be described with reference to FIG. 14.

[0116] First, both of the master and the slave transmit a master privilege transfer request to each other. When both of them receive this request, they transmit certificate data for device authentication. After authentication processing, the master transmits the identification data, priority data, and counter value to the slave. Alternatively, the slave transmits the priority data and identification data, and communication processing is completed.

[0117] Now, the processing steps in each device will be described in detail. FIG. 15 shows the processing steps in a master device having the "master privilege transfer" function. A description will be given with reference to FIG. 3.

[0118] When the device input/output device 13 receives the "master privilege transfer" request from the user (step S71), the data synchronization group setting and management unit 16 judges the current setting of the own device in the data synchronization group (step S72). When the device is a master, the device input/output device 13 continuously waits for a request for executing processing from the user. When the device input/output device 13 receives a request for executing processing (step S73), the communication device 11 transmits a request for master privilege transfer processing in a broadcasting manner (step S74). Then, the communication device 11 waits a request for such transfer processing from the counterpart device. When such request from the counterpart device arrives (step S75), the communication device 11 and the communication controller 12 determine whether or not only the own device and the counterpart device are executable the master privilege transfer processing (step S76). In the case of affirmative determination (YES at the step S76), the communication device 11 transmits certificate data to the counterpart device (step S77). The sequence of the steps of transferring certificate data and determining the number of executable devices may be reversed. When the communication device 11 receives certificate data from the counterpart device (step S78), the authentication unit 15 and data synchronization group setting and management unit 16 authenticate the counterpart device, and determines that the counterpart device exists in the same data synchronization group as the master (step S79). If authentication fails, processing terminates (NO at the step S79). When authentication is successful (YES at the step S79), the data synchronization group setting and management unit 16, communication controller 12, and communication device 11 transmit the identification data, priority data, and counter value owned by the own device to the slave (steps S80, S81, and S82). Then, the communication device 11 receives the priority data and identification data from the slave (steps S83 and S84). The data synchronization group setting and management unit 16 sets the priority data, certificate data, and identification data received from the slave as own priority data, certificate data, and identification data (step S85), and resets the counter. Then, processing terminates (step S86). In this manner, a device that has been

a master is specified as a slave device.

[0119] The timings of transmitting the priority data, counter value, certificate data, and identification data owned by the own device (steps S77, S80, S81, and S82) and those of receiving the priority data, certificate data, and identification data from a slave (steps S78, S83, and S84) may be in accordance with an arbitrary sequence.

[0120] Slave operations are generally similar to master operation. The processing steps are shown in FIG. 16.

[0121] When the device input/output device 13 receives a "master privilege transfer" request from the user, as described in FIG. 15, the data synchronization group setting and management unit 16 judges the current setting of the own device in the data synchronization group. In the case where this setting is not provided, processing terminates. When the device is a master, the aforementioned master operation is performed. In the case where it is determined as a slave, the device input/output device 13 waits for a request for executing master privilege transfer processing from the user. When the user executes master privilege transfer processing for the device input/output device 13 (step S91), the communication device 11 receives a master privilege transfer request from a master (step S92), and transmits the master privilege transfer request to the master (step S93). Then, at the step S94, the communication device 11 and the communication controller 12 check the number of devices that are executable for master privilege transfer processing. In the case where only one device other than own device is executable (YES at the step S94), it goes to the following steps.

[0122] First, the certificate data is exchanged with the master (steps S95 and S96). Then, the authentication unit 15 and the data synchronization group setting and management unit 16 authenticates the counterpart (step S97). After authentication, in the case where the communication counterpart is not a master of the data synchronization group to which the own device belong (NO at the step S97), processing terminates. In the case where the counterpart is the master (YES at the step S97), the priority data, counter value, and identification data owned by the master is received from the communication device 11 (steps S99, S100, and S101). Then, after the priority data and identification data owned by the own device have been transferred from the communication device 11 (steps S101 and S102), the data synchronization setting and management unit 16 sets the received master's identification data, certificate data, priority data, and counter value as own data (steps S103 and S104), and processing terminates.

[0123] The data synchronization group master can be set by each data set. In the case where a master device has a plurality of data sets, the master privilege transfer processing may be performed for only set data concerning a specific data set specified by the user. Of course, if a slave device specified as a transfer destination has a plurality of data sets, the master privileges concerning all the data sets owned by the master device can be trans-

ferred to one slave device in all.

(6) Priority Exchange

[0124] A "priority exchange" function exchanges the priorities owned by two slaves with each other. If a conflict occurs during data synchronization, such conflict is solved by using data on a member with its higher priority. The priority is assigned from the master in registration order. Thus, there can occur a case in which the user wants to cause the data on a slave registered later to precede the data on a slave registered previously. Although the priority may be selected during data synchronization member registration, the master or slave requires a display interface for selecting the priority. Here, devices are numbered in registration order for the purpose of simplification of processing so that priorities can be exchanged. At the time of "priority exchange" as well, as is the case with "data synchronization member registration" or "master privilege transfer", processing is performed only when only two devices are set.

[0125] Now, the communication processing step when the priorities are exchanged between slaves will be described with reference to FIG. 17.

[0126] First, slaves that want to exchange priorities transmit a priority exchange request to each other. When one slave receives such request, both of them transmit certificate data to each other. After it has been determined that the slaves belong to the same data synchronization group, both exchange priority data, and communication processing terminates.

[0127] In FIG. 18, there is shown the processing step of "priority exchange" in each slave device that exchanges the priority.

[0128] When the device input/output device 13 receives a "priority exchange" request from the user (step S101), the data synchronization group setting and management unit 16 judges the current setting of the own device in the data synchronization group. In the case where the setting is not provided or in the case of a master, processing terminates. In the case of a slave, the device input/output device 13 waits for a request for executing processing from the user. When the device input/output device 13 receives a request for executing processing (step S103), the communication device 11 transmits a request for processing "priority exchange" in a broadcasting manner, and waits for such request for processing "priority exchange" from the counterpart that exchange the priority. Then, when the communication device 11 receives a request for processing "priority exchange" from the counterpart (step S105), the communication controller 12 determines the number of devices that are currently executable for "priority exchange" (step S105) by using the data from the communication device 11 (step S106). In the case where only one device other than the own device is executable for "priority exchange" (YES at the step S106), the data synchronization group setting and management unit 16, communication con-

troller 12, and communication device 11 receive and/or transmit certificate data from and/or to each other (steps S107 and S108). If three or more devices are executable within a communicable range (NO at the step S106), processing is terminated.

[0129] After certificate data has been transmitted and/or received, the authentication unit 15 and the data synchronization group setting and management unit 16 authenticate the communication counterpart based on the authentication data received at the communication device 11 (step S109). When it is authenticated that the counterpart device is a member of the same synchronization group (YES at the step S109), the own device transmits own priority data, and receives the counterpart's priority data (steps S110 and S111). Then, the data synchronization group setting and management unit 16 rewrites its own priority with the received priority, and terminates processing (step S112).

[0130] In the case where a device functions as a slave of the data synchronization group, the device having a plurality of data sets, the processing shown in FIG. 18 may be performed for a data set in which priority exchange is specified by the user.

(7) Method for Setting a Device

[0131] During each of the "data synchronization member registration", "priority exchange", and "master privilege transfer" processes, it is required to ensure safety of data communication to be performed together with setting. Therefore, prior to exchange actually set data, two devices targeted for setting are moved to be requested for executing processing. Then, after it is determined as to whether or not only two devices are requested for executing processing in a communicable range, it is required to determine whether or not such setting is continued. A series of these processes will be described in more detail.

[0132] In connection to these processes, each device has a set mode and an execute mode. The set mode serves to specify call functions including "data synchronization member registration", "priority change", and "master privilege transfer". The execute mode serves to start actual processing of a function called by the set mode, and make data communication with the set counterpart as required. Therefore, this execute mode means a request for executing processing.

[0133] In the case where a user attempts to perform "data synchronization member registration", "priority change", or "master privilege transfer", as shown in FIG. 19, the user moves two devices targeted for setting (devices 1 and 2) to the set mode. Then, each device determines which function is called, and determines whether or not the called function is valid in current setting of the own device. For example, in the case where the user attempts to perform "data synchronization member registration", the user determines that the called function is the "data synchronization member registration", and de-

termines whether or not the own device is a master, slave or none. When the called function is valid, it can go to the execute mode. User's operation is required for the current mode to go to the execute mode.

[0134] The user simultaneously moves two devices to the execute mode relevant to the same function (strictly, an error Δt of about 1 second is permitted). At a stage before communication with the counterpart is established in the execute mode, each device waits for data indicating the execute mode from another device. After each device has received the data, each device transmits it to a transmission channel. During this data transmission, a data broadcasting transmission may be repeated with predetermined time intervals, for example. When each device receives data indicating the execute mode of the counterpart, after it is determined that there exists no other device that enters this mode and the devices 1 and 2 communicate with each other, communication of set data is started. In communication of the set data, data is encrypted to ensure security by using an encryption system such as the aforementioned public key encryption system. In the case where there exists any other device entering the execute mode, communication regarding the set data is not performed, and processing is interrupted and terminated.

[0135] A method of configuring a data synchronization group and the functions of each device has been described hereinbefore. Now, a specific example of providing devices configuring a data synchronization group will be described here.

[0136] FIG. 20 shows an example when a system for this communication method is applied to a wrist watch, and FIG. 21 shows an example when this system is applied to a portable information processing terminal.

[0137] In the wrist watch shown in FIG. 20, sections associated with data synchronization are two buttons that exist at the left of a character board and four display sections at the upper part of a clock display section. The two buttons include an upper left button employed to instruct a request for executing each function for setting a data synchronization group and for instructing the start of data synchronization processing; and a lower left button for selecting each function for setting a data synchronization group. The display sections include a mode display section for displaying each function for setting a data synchronization group; a display section for displaying a character M or S that represents a master or slave; a display section indicating a numeric value of priority data; and a display section indicating whether or not a data synchronization group is currently set. For a description of use, procedures for executing two typical functions, "data synchronization member registration" and "data synchronization processing" will be discussed here.

[0138] First, the "data synchronization member registration" will be described here. The lower left button is used to call a "data synchronization member registration" mode. The mode display section displays that the "data synchronization member registration" function has been

called. When this call is valid, the user holds down the upper left button, and executes a request for "data synchronization member registration." In this example, if this wrist watch is registered as a member by a given master, the similar procedures are performed in the master as well. When the aforementioned "data synchronization member registration" processing is successful while both devices execute the request, registration processing completes. In the case of "data synchronization processing", the upper left button is first pressed, whereby a function for starting synchronization processing is executed. If one or more devices belong to the same synchronization group that exists in a communicable range, and that can execute synchronization processing, data synchronization processing is actually started. The start of data synchronization processing may be executed by any one device calling the corresponding function or may be executed by all the devices calling the data synchronization processing start functions simultaneously, all of these devices being actually targeted for data synchronization processing at this time.

[0139] A data set that the wrist watch has is generally limited to time data (clock data). Thus, a device targeted for data synchronization processing with the wrist watch is a device that belongs to a data synchronization group concerning time data (clock data) in the devices that exist in a range communicable with the wrist watch.

[0140] In FIG. 21, there is shown an example of the screen for setting data synchronization of a portable information processing terminal. Shown here is an example of the screen for setting data synchronization concerning time data. This screen is composed of a "set menu" for calling each function provided at the upper left part; a display section that is set to display the current settings; and an execute button for starting data synchronization processing, the button being provided at the lower left. In the case where an attempt is made to perform data synchronization with a device that exists in the range communicable with the portable information processing terminal, belongs to the same data synchronization group capable of executing synchronization processing, a synchronization button at the lower right is pressed, whereby the data synchronization processing is called. In addition, the "data synchronization member registration" is selected from the set menu, whereby the data synchronization member registration screen is displayed. The execute button provided on this screen is held down, whereby the function for data synchronization member registration can be executed in the same manner as in the foregoing example of the wrist watch.

[0141] As has been described above, according to the communication method of the present embodiment, there is no need to provide set data concerning a communication counterpart for data synchronization. Basically, data synchronization processing can be performed safely and reliably by a certificate (certificate data) and a priority. Thus, data synchronization processing can be performed between arbitrary slaves without requiring a

master. In particular, the present invention is preferable to application to a device that makes wireless communication outside. Moreover, synchronization processing of a data set that belongs to the same data synchronization group is automatically executed with another device that exists in a communicable range by instructing the start of synchronization processing. Therefore, in a device having only a limited user interface such as wrist watch or portable cellular phone, the device being difficult in setting concerning a communication counterpart as well, data synchronization can be easily performed by data synchronization group such that a group member is dynamically changed.

[0142] For example, although each embodiment has been described singly, a plurality of embodiments may be combined with each other as required. The communication method according to the present embodiment can be achieved by a computer program. Thus, the computer program is recorded in a computer readable recording medium such as CD-ROM, and then, is introduced into the computer or the computer program is recorded into the recording medium such as ROM, and then, is merely incorporated into an electronic device, making it possible to achieve an effect similar to the present embodiment using a general electronic device.

[0143] As has been described above, according to the present invention, it is possible to provide a communication method and electronic device that enables data synchronization between arbitrary devices without requiring a server. It is possible to perform data synchronization with a device that does not have a limited input device and a display device. It is possible to make communication for data synchronization safely between arbitrary devices in a wired or wireless network such that a network configuration dynamically changes. In exchanging set data between devices, only specific two devices perform setting operations simultaneously after the presence of two set devices has been guaranteed, whereby setting can be done safely.

Claims

1. A data synchronisation method comprising the steps of:

transmitting from a master device to a slave device a certificate indicating that the slave device belongs to a data synchronisation group to which the master device belongs and a priority to be used for solving conflict of data during data synchronisation, thereby registering the slave device as a member of the data synchronisation group to which the master device belongs; determining whether or not a first slave device and a second slave device which is capable of communicating with the first slave device belong to the same data synchronisation group by using

the certificate;

and performing data synchronisation between the first slave device and the second slave device based on the priority if the first slave device and the second slave device belong to the same data synchronisation group.

2. The method according to Claim 1, wherein said registering is performed after it is confirmed that there is no device other than said master device and said slave device capable of communicating with said master device and set in a registration mode.
3. The method according to Claim 1, wherein said data synchronisation is performed between one master device and one or more slave devices which belong to the same synchronisation group or between at least two slave devices which belong to the same synchronisation group.
4. The method according to Claim 1, wherein said master device and slave device store plural types of data and belong to plural data synchronisation groups defined for each of said plural types of data.
5. The method according to Claim 4, wherein said registering is performed by transmitting said certificate and said priority from the master device belonging to a given synchronisation group defined for a given type of data to the slave device belonging to the given synchronisation group, said certificate and said priority being set depending on the given type of data.
6. The method according to Claim 1, further comprising transmitting data required to operate as the master device from said master device to a slave device which is targeted to transfer a master privilege, thereby transferring the master privilege to the slave device.
7. The method according to Claim 6, wherein said master privilege transferring is performed after it is confirmed that there is no slave device other than said master device and said slave device capable of communicating with said master device and set in a master privilege transfer mode.
8. The method according to Claim 1, further comprising exchanging the priority between slave devices that belong to the same data synchronisation group.
9. The method according to Claim 8, wherein said priority exchanging is performed after it is confirmed that there is no slave device other than said master device and said slave device capable of communicating with said master device and set in a priority exchanging mode.

10. The method according to Claim 1, further comprising releasing said master device or said slave device from the data synchronisation group, wherein, if said master device is released, said master device transmits a releasing instruction to a slave device which is capable of communicating with said master device to release said slave device as well as said master device.
11. A method according to claim 4, wherein said registering is performed by transmitting said certificate and said priority from the master device set by a user for a given type of data to the slave device set by the user for the given type of data, said certificate and said priority being set depending on the type of data.
12. An electronic device comprising:
- a master unit, when an own device is set by a user as a master, configured to transmit to a slave device a certificate indicating that the slave device belongs to a data synchronization group to which the master device belongs and a priority to be used for solving conflict of data during data synchronisation, thereby registering the slave device as a member of the data synchronization group to which the master device belongs;
 - a slave unit, when the own device is set by the user as the slave, configured to receive from the master unit the certificate and the priority; and
 - a data synchronization unit (16, 17) configured to determine whether or not the own device and another device which is capable of communicating with the own device belong to the same data synchronization group by using the certificate, and perform data synchronization between the own device and the other device based on the priority if the own device and the other device belong to the same data synchronization group.
13. A slave device comprising:
- a receiver (11, 12) configured to receive, from a master device for a data synchronization group, a certificate indicating that the slave device belongs to the data synchronization group to which the master device belongs and a priority to be used for solving conflict of data during data synchronisation;
 - a memory (18) configured to store the received certificate and the priority;
 - a certification unit (15) configured to determine, when a synchronization request from another device is received, whether or not the other device and the slave device belong to the same data synchronization group by using the certificate stored in the memory; and
 - a synchronization unit (17) configured to perform data synchronization between the other device and the slave device based on the priority stored in the memory when the certification unit determines that the other device and the slave device belong to the same data synchronization group.
14. The device according to claim 13, wherein said memory (18) is configured to store the received certificate and the priority after it is confirmed that there is no device other than said master device and said slave device, capable of communicating with said master device and set in a registration mode.
15. The device according to claim 13, wherein said other device comprises a master device or one or more slave devices.
16. The device according to claim 13, wherein said master device and slave device are adapted to store plural types of data and belong to plural data synchronization groups defined for each of said plural types of data.
17. The device according to claim 16, wherein said memory (18) is configured to store said certificate and said priority from the master device belonging to a given synchronization group defined for a given type of data, said certificate and said priority being set depending on the given type of data.
18. The device according to claim 13, wherein the receiver (11, 12) is configured to receive data required to operate as the master device from said master device, thereby transferring a master privilege to the slave device.
19. The device according to claim 18, wherein said master privilege transferring is performed after it is confirmed that there is no slave device other than said master device and said slave device capable of communicating with said master device and set in a master privilege transfer mode.
20. The device according to claim 13, further comprising an exchanging unit configured to exchange the priority between another slave device that belongs to the same data synchronization group.
21. The device according to claim 20, wherein said priority exchanging is performed after it is confirmed that there is no slave device other than said master device and said slave device capable of communicating with said master device and set in a priority exchanging mode.
22. A data synchronization method for a slave device comprising the steps of:

receiving from a master device for a data synchronization group, a certificate indicating that the slave device belongs to the data synchronization group to which the master device belongs and a priority to be used for solving conflict of data during data synchronization;
 storing the received certificate and the priority;
 determining, when a synchronization request from another device is received, whether or not the other device and the slave device belong to the same data synchronization group by using the stored certificate; and
 performing data synchronization between the other device and the slave device based on the stored priority when it is determined that the other device and the slave device belong to the same data synchronization group.

23. The method according to claim 22, comprising the step of storing the received certificate and the priority after it is confirmed that there is no device other than said master device and said slave device capable of communicating with said master device and set in a registration mode.
24. The method according to claim 22, wherein said other device comprises: a master device or one or more slave devices.
25. The method according to claim 22, wherein said master device and said slave device belong to plural data synchronization groups and comprising storing, in said master device and said slave device, plural types of data defined for each of said plural data synchronization groups.
26. The method according to claim 25, comprising storing said certificate and said priority from the master device belonging to a given synchronization group defined for a given type of data, said certificate and said priority being set depending on the given type of data.
27. The method according to claim 22, comprising receiving data required for a slave device to operate as the master device from said master device, for transferring a master privilege to the slave device.
28. The method according to claim 27, wherein transferring of said master privilege is performed after it is confirmed that there is no slave device other than said master device and said slave device capable of communicating with said master device and set in a master privilege transfer mode.
29. The method according to claim 22, further comprising exchanging the priority between another slave device that belongs to the same data synchroniza-

tion group.

30. The method according to claim 29, wherein exchanging of priority is performed after it is confirmed that there is no slave device other than said master device and said slave device capable of communicating with said master device and set in a priority exchanging mode.
31. A carrier medium carrying computer readable instructions for controlling a computer to carry out a method of any one of claims 1 to 11 and 22 to 30.

Revendications

1. Procédé de synchronisation de données, comprenant les étapes consistant à :

transmettre d'un dispositif maître à un dispositif esclave un certificat indiquant que le dispositif esclave appartient à un groupe de synchronisation de données auquel appartient le dispositif maître et une priorité à utiliser pour résoudre un conflit de données pendant la synchronisation de données, enregistrant ainsi le dispositif esclave comme un membre du groupe de synchronisation de données auquel appartient le dispositif maître ;
 déterminer si oui ou non un premier dispositif esclave et un deuxième dispositif esclave, qui est capable de communiquer avec le premier dispositif esclave, appartiennent au même groupe de synchronisation de données en utilisant le certificat ; et
 effectuer la synchronisation de données entre le premier dispositif esclave et le deuxième dispositif esclave sur la base de la priorité si le premier dispositif esclave et le deuxième dispositif esclave appartiennent au même groupe de synchronisation de données.

2. Procédé selon la revendication 1, dans lequel ledit enregistrement est effectué après qu'il a été confirmé qu'il n'existe aucun dispositif autre que ledit dispositif maître et ledit dispositif esclave capable de communiquer avec ledit dispositif maître et paramétré en mode d'enregistrement.
3. Procédé selon la revendication 1, dans lequel ladite synchronisation de données est effectuée entre un seul dispositif maître et un ou plusieurs dispositifs esclaves qui appartiennent au même groupe de synchronisation ou entre au moins deux dispositifs esclaves qui appartiennent au même groupe de synchronisation.
4. Procédé selon la revendication 1, dans lequel ledit

dispositif maître et ledit dispositif esclave stockent plusieurs types de données et appartiennent à plusieurs groupes de synchronisation de données définis pour chacun desdits plusieurs types de données.

5. Procédé selon la revendication 4, dans lequel ledit enregistrement est effectué en transmettant ledit certificat et ladite priorité du dispositif maître appartenant à un groupe de synchronisation donné défini pour un type donné de données au dispositif esclave appartenant au groupe de synchronisation donné, ledit certificat et ladite priorité étant paramétrés en fonction du type donné de données.

6. Procédé selon la revendication 1, comprenant en outre la transmission de données nécessaires pour fonctionner comme le dispositif maître, dudit dispositif maître à un dispositif esclave qui est ciblé pour transférer un privilège de maître, transférant ainsi le privilège de maître au dispositif esclave.

7. Procédé selon la revendication 6, dans lequel ledit transfert de privilège de maître est effectué après qu'il a été confirmé qu'il n'existe aucun dispositif esclave autre que ledit dispositif maître et ledit dispositif esclave capable de communiquer avec ledit dispositif maître et paramétré en mode de transfert de privilège de maître.

8. Procédé selon la revendication 1, comprenant en outre l'échange de la priorité entre des dispositifs esclaves qui appartiennent au même groupe de synchronisation de données.

9. Procédé selon la revendication 8, dans lequel ledit échange de priorité est effectué après qu'il a été confirmé qu'il n'existe aucun dispositif esclave autre que ledit dispositif maître et que ledit dispositif esclave, capable de communiquer avec ledit dispositif maître et paramétré en mode d'échange de priorité.

10. Procédé selon la revendication 1, comprenant en outre la libération dudit dispositif maître ou dudit dispositif esclave du groupe de synchronisation de données, dans lequel, si ledit dispositif maître est libéré, ledit dispositif maître transmet une instruction de libération à un dispositif esclave qui est capable de communiquer avec ledit dispositif maître pour libérer ledit dispositif esclave ainsi que ledit dispositif maître.

11. Procédé selon la revendication 4, dans lequel ledit enregistrement est effectué en transmettant ledit certificat et ladite priorité du dispositif maître paramétré par un utilisateur pour un type donné de données au dispositif esclave paramétré par l'utilisateur pour le type donné de données, ledit certificat et la-

dite priorité étant paramétrés en fonction du type de données.

12. Dispositif électronique, comprenant :

une unité de maître, quand un dispositif lui-même est paramétré par un utilisateur comme maître, configurée pour transmettre à un dispositif esclave un certificat indiquant que le dispositif esclave appartient à un groupe de synchronisation de données auquel appartient le dispositif maître et une priorité à utiliser pour résoudre un conflit de données pendant la synchronisation de données, enregistrant ainsi le dispositif esclave comme un membre du groupe de synchronisation de données auquel appartient le dispositif maître ;
une unité d'esclave, quand le dispositif lui-même est paramétré par l'utilisateur comme esclave, configurée pour recevoir de l'unité de maître le certificat et la priorité ; et
une unité de synchronisation de données (16, 17) configurée pour déterminer si oui ou non le dispositif lui-même et un autre dispositif, qui est capable de communiquer avec le dispositif lui-même, appartiennent au même groupe de synchronisation de données en utilisant le certificat, et pour effectuer une synchronisation de données entre le dispositif lui-même et l'autre dispositif sur la base de la priorité si le dispositif lui-même et l'autre dispositif appartiennent au même groupe de synchronisation de données.

13. Dispositif esclave, comprenant :

un récepteur (11, 12) configuré pour recevoir, d'un dispositif maître pour un groupe de synchronisation de données, un certificat indiquant que le dispositif esclave appartient au groupe de synchronisation de données auquel appartient le dispositif maître et une priorité à utiliser pour résoudre un conflit de données pendant la synchronisation de données ;
une mémoire (18) configurée pour stocker le certificat reçu et la priorité ;
une unité de certification (15) configurée pour déterminer, quand une demande de synchronisation d'un autre dispositif est reçue, si oui ou non l'autre dispositif et le dispositif esclave appartiennent au même groupe de synchronisation de données en utilisant le certificat stocké dans la mémoire ; et
une unité de synchronisation (17) configurée pour effectuer la synchronisation de données entre l'autre dispositif et le dispositif esclave sur la base de la priorité stockée dans la mémoire quand l'unité de certification détermine que l'autre dispositif et le dispositif esclave appar-

tiennent au même groupe de synchronisation de données.

14. Dispositif selon la revendication 13, dans lequel ladite mémoire (18) est configurée pour stocker le certificat reçu et la priorité après qu'il a été confirmé qu'il n'existe aucun dispositif autre que ledit dispositif maître et ledit dispositif esclave, capable de communiquer avec ledit dispositif maître et paramétré en mode d'enregistrement.

15. Dispositif selon la revendication 13, dans lequel ledit autre dispositif comprend un dispositif maître ou un ou plusieurs dispositifs esclaves.

16. Dispositif selon la revendication 13, dans lequel lesdits dispositifs maître et esclave sont adaptés pour stocker plusieurs types de données et appartiennent à plusieurs groupes de synchronisation de données définis pour chacun desdits plusieurs types de données.

17. Dispositif selon la revendication 16, dans lequel ladite mémoire (18) est configurée pour stocker ledit certificat et ladite priorité provenant du dispositif maître appartenant à un groupe de synchronisation donné défini pour un type donné de données, ledit certificat et ladite priorité étant paramétrés en fonction du type donné de données.

18. Dispositif selon la revendication 13, dans lequel le récepteur (11, 12) est configuré pour recevoir des données nécessaires pour fonctionner comme le dispositif maître dudit dispositif maître, transférant ainsi un privilège de maître au dispositif esclave.

19. Dispositif selon la revendication 18, dans lequel ledit transfert de privilège de maître est effectué après qu'il a été confirmé qu'il n'existe aucun dispositif esclave autre que ledit dispositif maître et ledit dispositif esclave capable de communiquer avec ledit dispositif maître et paramétré en mode de transfert de privilège de maître.

20. Dispositif selon la revendication 13, comprenant en outre une unité d'échange configurée pour échanger la priorité entre un autre dispositif esclave qui appartient au même groupe de synchronisation de données.

21. Dispositif selon la revendication 20, dans lequel ledit échange de priorité est effectué après qu'il a été confirmé qu'il n'existe aucun dispositif esclave autre que ledit dispositif maître et ledit dispositif esclave capable de communiquer avec ledit dispositif maître et paramétré en mode d'échange de priorité.

22. Procédé de synchronisation de données pour un dis-

positif esclave comprenant les étapes consistant à :

recevoir d'un dispositif maître pour un groupe de synchronisation de données, un certificat indiquant que le dispositif esclave appartient au groupe de synchronisation de données auquel appartient le dispositif maître et une priorité à utiliser pour résoudre un conflit de données pendant la synchronisation de données ;
stocker le certificat reçu et la priorité ;
déterminer, quand une demande de synchronisation d'un autre dispositif est reçue, si oui ou non l'autre dispositif et le dispositif esclave appartiennent au même groupe de synchronisation de données en utilisant le certificat stocké ;
et
effectuer la synchronisation de données entre l'autre dispositif et le dispositif esclave sur la base de la priorité stockée quand il est déterminé que l'autre dispositif et le dispositif esclave appartiennent au même groupe de synchronisation de données.

23. Procédé selon la revendication 22, comprenant l'étape consistant à stocker le certificat reçu et la priorité après qu'il a été confirmé qu'il n'existe aucun dispositif autre que ledit dispositif maître et que ledit dispositif esclave, capable de communiquer avec ledit dispositif maître et paramétré en mode d'enregistrement.

24. Procédé selon la revendication 22, dans lequel ledit autre dispositif comprend : un dispositif maître ou un ou plusieurs dispositifs esclaves.

25. Procédé selon la revendication 22, dans lequel ledit dispositif maître et ledit dispositif esclave appartiennent à plusieurs groupes de synchronisation de données et comprenant le stockage, dans ledit dispositif maître et ledit dispositif esclave, de plusieurs types de données définis pour chacun desdits plusieurs groupes de synchronisation de données.

26. Procédé selon la revendication 25, comprenant le stockage dudit certificat et de ladite priorité dudit dispositif maître appartenant à un groupe de synchronisation donné défini pour un type donné de données, ledit certificat et ladite priorité étant paramétrés en fonction du type donné de données.

27. Procédé selon la revendication 22, comprenant la réception de données nécessaires pour qu'un dispositif esclave fonctionne comme le dispositif maître dudit dispositif maître, pour transférer un privilège de maître au dispositif esclave.

28. Procédé selon la revendication 27, dans lequel le transfert dudit privilège de maître est effectué après

qu'il a été confirmé qu'il n'existe aucun dispositif esclave autre que ledit dispositif maître et que ledit dispositif esclave capable de communiquer avec ledit dispositif maître et paramétré en mode de transfert de privilège de maître.

29. Procédé selon la revendication 22, comprenant en outre l'échange de la priorité entre un autre dispositif esclave qui appartient au même groupe de synchronisation de données.

30. Procédé selon la revendication 29, dans lequel l'échange de priorité est effectué après qu'il a été confirmé qu'il n'existe aucun dispositif esclave autre que ledit dispositif maître et que ledit dispositif esclave capable de communiquer avec ledit dispositif maître et paramétré en mode d'échange de priorité.

31. Moyen de support supportant des instructions lisibles par ordinateur pour commander un ordinateur pour qu'il exécute un procédé selon l'une quelconque des revendications 1 à 11 et 22 à 30.

Patentansprüche

1. Ein Datensynchronisationsverfahren umfassend die Schritte:

Übertragen eines Zertifikats von einem Haupt- bzw. Master-Gerät zu einem Folge- bzw. Slave-Gerät, wobei das Zertifikat andeutet, dass das Slave-Gerät zu einer Datensynchronisationsgruppe gehört, zu der das Master-Gerät gehört, sowie einer Priorität, die während der Datensynchronisation zum Lösen von Datenkonflikten einzusetzen ist, wodurch das Slave-Gerät als ein Mitglied der Datensynchronisationsgruppe, zu der das Master-Gerät gehört, registriert wird;

Bestimmen ob oder nicht ein erstes Slave-Gerät und ein zweites Slave-Gerät, das in der Lage ist, mit dem zu derselben Datensynchronisationsgruppe gehörenden, ersten Slave-Gerät unter Verwendung des Zertifikats zu kommunizieren;

und Ausführen einer Datensynchronisation zwischen dem ersten Slave-Gerät und dem zweiten Slave-Gerät gestützt auf die Priorität, falls das erste Slave-Gerät und das zweite Slave-Gerät zu derselben Datensynchronisationsgruppe gehören.

2. Das Verfahren nach Anspruch 1, wobei das Registrieren ausgeführt wird, nachdem bestätigt worden ist, dass es kein anderes Gerät als das erste Master-Gerät und das Slave-Gerät gibt, das in der Lage ist, mit dem Master-Gerät zu kommunizieren, und in

einen Registrierungsmodus eingestellt ist.

3. Das Verfahren nach Anspruch 1, wobei die Datensynchronisation zwischen einem Master-Gerät und einem oder mehreren Slave-Geräten, die zu derselben Synchronisationsgruppe gehören, oder zwischen wenigstens zwei Slave-Geräten, die zu derselben Synchronisationsgruppe gehören, ausgeführt wird.

4. Das Verfahren nach Anspruch 1, wobei das Master-Gerät und Slave-Geräte vielfache Datentypen speichern und zu mehreren Datensynchronisationsgruppen, die jeweils für die vielfachen Datentypen definiert sind, gehören.

5. Das Verfahren nach Anspruch 4, wobei das Registrieren ausgeführt wird, indem das Zertifikat und die Priorität von dem Master-Gerät, das zu einer gegebenen Synchronisationsgruppe gehört, die für einen gegebenen Datentyp definiert ist, zu dem Slave-Gerät, das zu der gegebenen Synchronisationsgruppe gehört, wobei das Zertifikat und die Priorität in Abhängigkeit vom gegebenen Datentyp eingestellt werden.

6. Das Verfahren nach Anspruch 1, ferner umfassend das Übertragen von Daten, die zum Operieren als das Master-Gerät des Master-Geräts erforderlich sind, zu einem Slave-Gerät, das darauf gerichtet ist, ein Master-Privileg zu übertragen, wodurch das Master-Privileg auf das Slave-Gerät übertragen wird.

7. Das Verfahren nach Anspruch 6, wobei das Übertragen des Master-Privilegs ausgeführt wird, nachdem bestätigt worden ist, dass es kein anderes Slave-Gerät verschieden von dem Master-Gerät und dem Slave-Gerät gibt, das in der Lage ist, mit dem Master-Gerät zu kommunizieren, und in einen Master-Privileg-Übertragungsmodus eingestellt ist.

8. Das Verfahren nach Anspruch 1, ferner umfassend Austauschen der Priorität zwischen zu der gleichen Datensynchronisationsgruppe gehörenden Slave-Geräten.

9. Das Verfahren nach Anspruch 8, wobei das Austauschen der Priorität ausgeführt wird, nachdem bestätigt worden ist, dass es kein Slave-Gerät verschieden von dem Master-Gerät und dem Slave-Gerät gibt, das in der Lage ist, mit dem Master-Gerät zu kommunizieren, und in einen Prioritätsaustauschmodus eingestellt ist.

10. Das Verfahren nach Anspruch 1, ferner umfassend Freigeben des Master-Geräts oder Slave-Geräts aus der Datensynchronisationsgruppe, wobei, wenn das Master-Gerät freigegeben wird, das Master-Ge-

rät eine Freigabeanweisung an ein Slave-Gerät überträgt, welches in der Lage ist, mit dem Master-Gerät zu kommunizieren, um sowohl das Slave-Gerät als auch das Master-Gerät freizugeben.

11. Ein Verfahren nach Anspruch 4, wobei das Registrieren ausgeführt wird, indem das Zertifikat und die Priorität von dem durch einen Benutzer für einen gegebenen Datentyp eingestellten Master-Gerät übertragen wird, an das von dem Benutzer für einen gegebenen Datentyp eingestellte Slave-Gerät, wobei das Zertifikat und die Priorität in Abhängigkeit von dem Datentyp eingestellt werden.

12. Ein elektronisches Gerät umfassend:

eine Haupt- bzw. Master-Einheit, wenn ein eigenes Gerät durch einen Benutzer als ein Master eingestellt ist, konfiguriert zum Übertragen eines Zertifikats zu einem Slave-Gerät, wobei das Zertifikat andeutet, dass das Slave-Gerät zu einer Datengruppe gehört, zu der das Master-Gerät gehört, sowie einer Priorität, die zum Lösen von Datenkonflikten während der Datensynchronisation einzusetzen ist, wodurch das Slave-Gerät als ein Mitglied der Datensynchronisationsgruppe, zu der das Master-Gerät gehört, registriert wird;
eine Slave-Einheit, wenn das eigene Gerät durch den Benutzer als der Slave eingestellt wird, konfiguriert, zum Empfangen des Zertifikats und der Priorität von der Master-Einheit; und
eine Datensynchronisationseinheit (16, 17), konfiguriert zum Bestimmen unter Verwendung des Zertifikats, ob oder nicht das eigene Gerät und ein anderes Gerät, das in der Lage ist, mit dem eigenen Gerät zu kommunizieren, zu der gleichen Datensynchronisationsgruppe gehört, und Ausführen der Datensynchronisation zwischen dem eigenen Gerät und dem anderen Gerät gestützt auf die Priorität, falls das eigene Gerät und das andere Gerät zu der gleichen Datensynchronisationsgruppe gehören.

13. Ein Slave-Gerät umfassend:

einen Empfänger (11, 12), von einem Master-Gerät für eine Datensynchronisationsgruppe konfiguriert zum Empfangen eines Zertifikats, das andeutet, dass das Slave-Gerät zu der Datensynchronisationsgruppe gehört, zu der das Master-Gerät gehört, sowie einer Priorität, die zum Lösen von Datenkonflikten während der Datensynchronisation einzusetzen ist;
einen Speicher (18), konfiguriert zum Speichern des empfangenen Zertifikats und der Priorität; eine Zertifizierungseinheit (15), konfiguriert zum

Bestimmen, unter Verwendung des in dem Speicher gespeicherten Zertifikats, wenn eine Synchronisationsanfrage von einem anderen Gerät empfangen wird, ob oder nicht das andere Gerät und das Slave-Gerät zu derselben Datensynchronisationsgruppe gehören; und eine Synchronisationseinheit (17), konfiguriert zum Ausführen von Datensynchronisation, gestützt auf der in dem Speicher gespeicherten Priorität, zwischen dem anderen Gerät und dem Slave-Gerät, wenn die Zertifizierungseinheit feststellt, dass das andere Gerät und das Slave-Gerät zu der gleichen Datensynchronisationsgruppe gehören.

14. Das Gerät nach Anspruch 13, wobei der Speicher (18) konfiguriert ist zum Speichern des empfangenen Zertifikats und der Priorität, nachdem bestätigt worden ist, dass es kein Gerät verschieden von dem Master-Gerät und dem Slave-Gerät gibt, das in der Lage ist, mit dem Master-Gerät zu kommunizieren, und in einen Registrierungsmodus eingestellt ist.

15. Das Gerät nach Anspruch 13, wobei das andere Gerät ein Master-Gerät oder eine oder mehrere Slave-Geräte umfasst.

16. Das Gerät nach Anspruch 13, wobei das Master-Gerät und das Slave-Gerät dazu angepasst sind, vielfältige Datentypen zu speichern und zu vielfältigen, jeweils für die vielfältigen Datentypen definierten Datensynchronisationsgruppen zu gehören.

17. Das Verfahren nach Anspruch 16, wobei der Speicher (18) konfiguriert ist zum Speichern des Zertifikats und der Priorität von dem Master-Gerät, das zu einer gegebenen, für einen gegebenen Datentypen definierten Synchronisationsgruppe gehört, wobei das Zertifikat und die Priorität in Abhängigkeit von dem gegebenen Datentyp eingestellt werden.

18. Das Gerät nach Anspruch 13, wobei der Empfänger (11, 12) konfiguriert ist zum Empfangen von Daten, die zum Operieren als das Master-Gerät des Master-Geräts benötigt werden, wodurch ein Master-Privileg auf das Slave-Gerät übertragen wird.

19. Das Gerät nach Anspruch 18, wobei das Übertragen des Master-Privilegs ausgeführt wird, nachdem bestätigt worden ist, dass es kein Slave-Gerät verschieden von dem Master-Gerät gibt, und dass das Slave-Gerät in der Lage ist, mit dem Master-Gerät zu kommunizieren, und in einen Master-Privileg-Übertragungsmodus eingestellt ist.

20. Das Gerät nach Anspruch 13, ferner umfassend eine Austauschseinheit, konfiguriert zum Austauschen der Priorität zwischen einem anderen, zu der gleichen

Datensynchronisationsgruppe gehörenden Slave-Gerät.

21. Das Gerät nach Anspruch 20, wobei das Austauschen der Priorität ausgeführt wird, nachdem bestätigt worden ist, dass es kein Slave-Gerät verschieden von dem Master-Gerät und dem Slave-Gerät gibt, das in der Lage ist, mit dem Master-Gerät zu kommunizieren, und in einen Prioritätsaustauschmodus eingestellt ist.

22. Ein Datensynchronisationsverfahren für ein Slave-Gerät, umfassend die Schritte:

Empfangen eines Zertifikats von einem Haupt- bzw. Master-Gerät für eine Datensynchronisationsgruppe, wobei das Zertifikat andeutet, dass das Slave-Gerät zu der Datensynchronisationsgruppe gehört, zu der das Master-Gerät gehört, sowie einer Priorität, die zum Lösen von Datenkonflikten während der Datensynchronisation einzusetzen ist;
Abspeichern des empfangenen Zertifikats und der Priorität;
Bestimmen, unter Verwendung des gespeicherten Zertifikats, wann eine Synchronisationsanfrage von einem anderen Gerät empfangen wird, ob oder nicht das andere Gerät und das Slave-Gerät zu der gleichen Datensynchronisationsgruppe gehören; und Ausführen einer Datensynchronisation zwischen dem anderen Gerät und dem Slave-Gerät gestützt auf die gespeicherte Priorität, wenn bestimmt worden ist, dass das andere Gerät und das Slave-Gerät zu der gleichen Datensynchronisationsgruppe gehören.

23. Das Verfahren nach Anspruch 22, umfassend den Schritt des Abspeicherns des empfangenen Zertifikats und der Priorität, nachdem bestätigt worden ist, dass es kein Gerät anders als das Master-Gerät und das Slave-Gerät gibt, das dazu in der Lage ist, mit dem Master-Gerät zu kommunizieren, und in einen Registrierungsmodus eingestellt ist.

24. Das Verfahren nach Anspruch 22, wobei das andere Gerät ein Master-Gerät und eine oder mehrere Slave-Geräte umfasst.

25. Das Verfahren nach Anspruch 22, wobei das Master-Gerät und die Slave-Geräte zu mehreren Datensynchronisationsgruppen gehören, und das umfasst Speichern in dem Master-Gerät und dem Slave-Gerät von vielfältigen, für jedes der vielfältigen Datensynchronisationsgruppen definierten Typen von Daten.

26. Das Verfahren nach Anspruch 25, umfassend Spei-

chern des Zertifikats und der Priorität, die von dem zu einer gegebenen, für einen gegebenen Datentyp definierten Synchronisationsgruppe gehörenden Master-Gerät, wobei das Zertifikat und die Priorität in Abhängigkeit von dem gegebenen Datentyp eingestellt werden.

27. Das Verfahren nach Anspruch 22, umfassend Empfangen von Daten, die für ein Slave-Gerät benötigt werden zum Operieren als das Master-Gerät von dem Master-Gerät, zum Übertragen eines Master-Privilegs auf das Slave-Gerät.

28. Das Verfahren nach Anspruch 27, wobei das Übertragen des Master-Privilegs ausgeführt wird, nachdem bestätigt worden ist, dass es kein anderes Slave-Gerät verschieden von dem Master-Gerät und dem Slave-Gerät gibt, das in der Lage ist, mit dem Master-Gerät zu kommunizieren, und in einen Master-Privileg-Übertragungsmodus eingestellt ist.

29. Das Verfahren nach Anspruch 22, ferner umfassend das Austauschen der Priorität zwischen einem anderen, zu der gleichen Datensynchronisationsgruppe gehörenden Slave-Gerät.

30. Das Verfahren nach Anspruch 29, wobei das Austauschen der Priorität ausgeführt wird, nachdem bestätigt worden ist, dass es kein Slave-Gerät gibt verschieden von dem Master-Gerät und dem Slave-Gerät, das in der Lage ist, mit dem Master-Gerät zu kommunizieren, und in einen Prioritätsaustauschmodus eingestellt ist.

31. Ein Speichermedium, das computerlesbare Anweisungen trägt zum Steuern eines Computers, um ein Verfahren nach einem der Ansprüche 1 bis 11 und 22 bis 30 auszuführen.

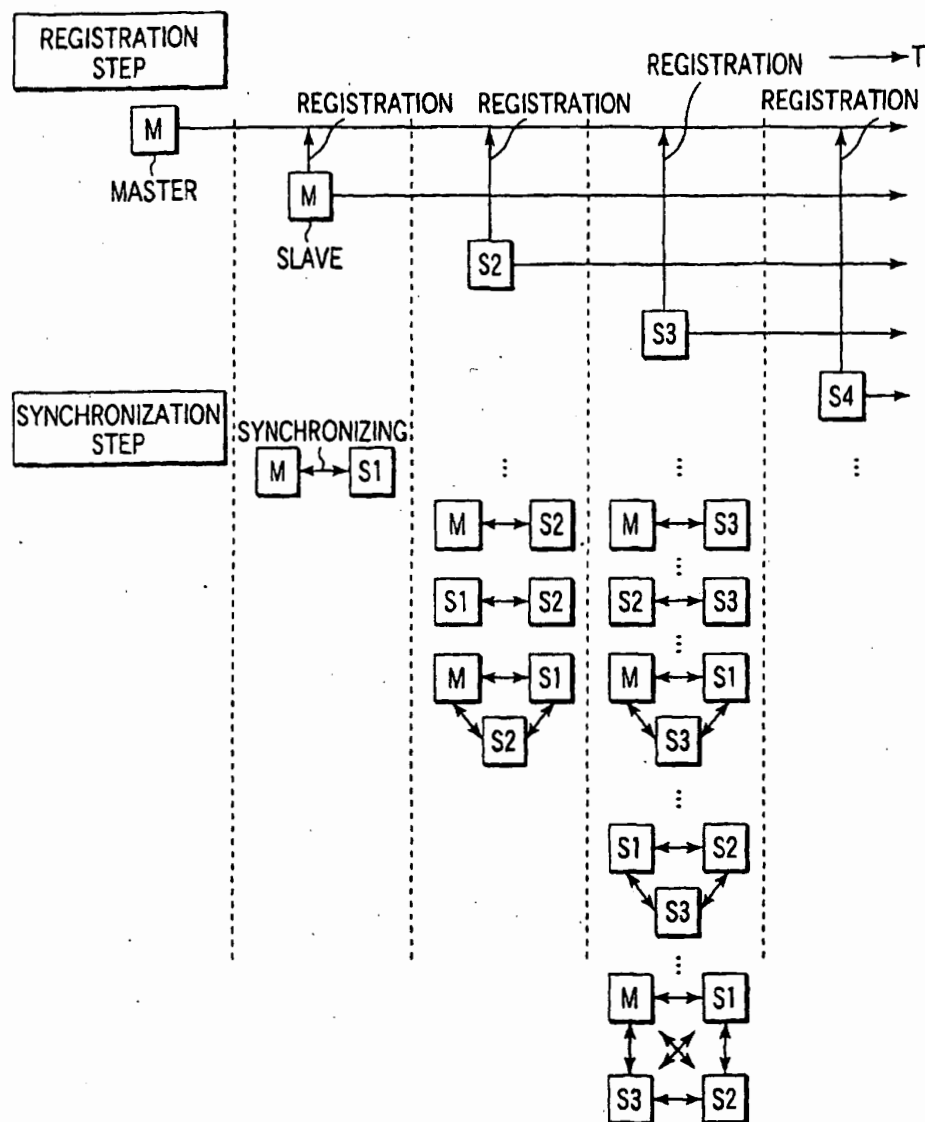


FIG. 1

TYPE OF DATA SET	MASTER/SLAVE	PRIORITY DATA	CERTIFICATE DATA	IDENTIFICATION DATA	COUNTER VALUE
ADDRESS DIRECTORY	MASTER	1	01101110	10101010	12
TIME	SLAVE	5	00111010	01010101	—
SCHEDULE	—	—	—	—	—
TASK LIST	—	—	—	—	—
E-MAIL	SLAVE	8	11010011	11110100	—
⋮	⋮	⋮	⋮	⋮	⋮

FIG. 2

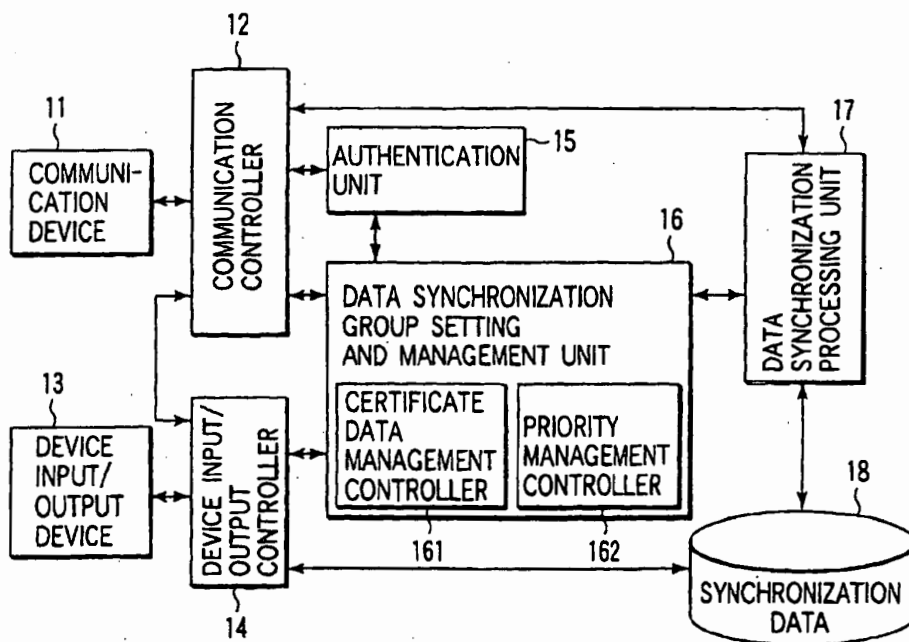


FIG. 3

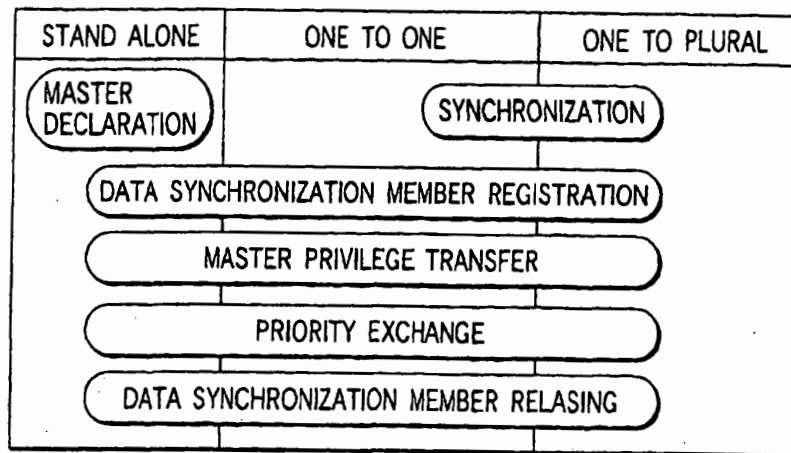


FIG. 4

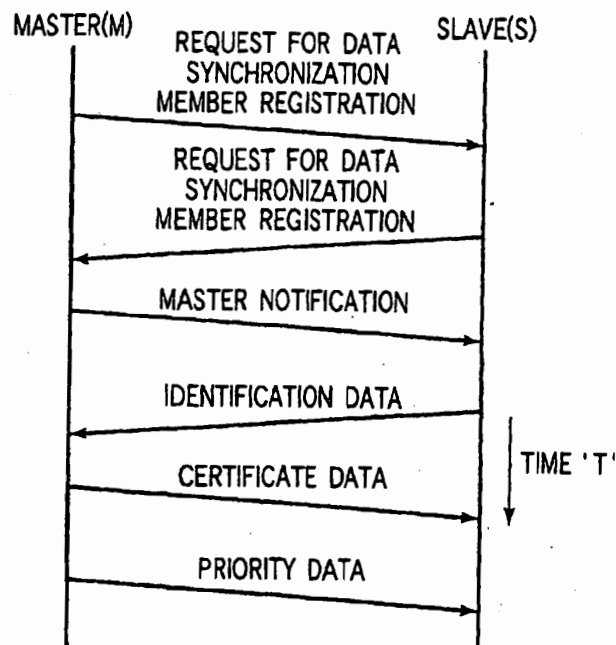


FIG. 5

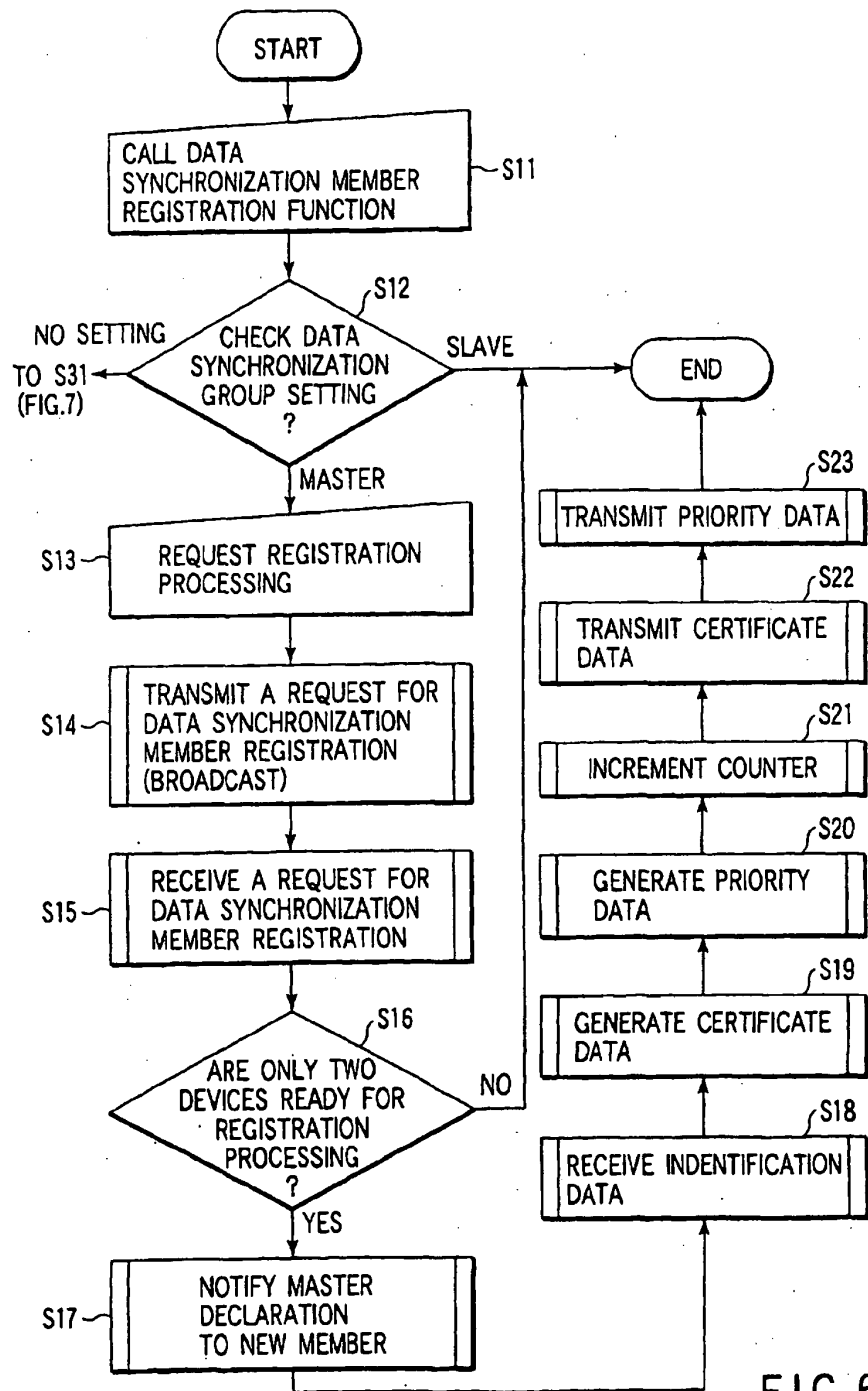


FIG. 6

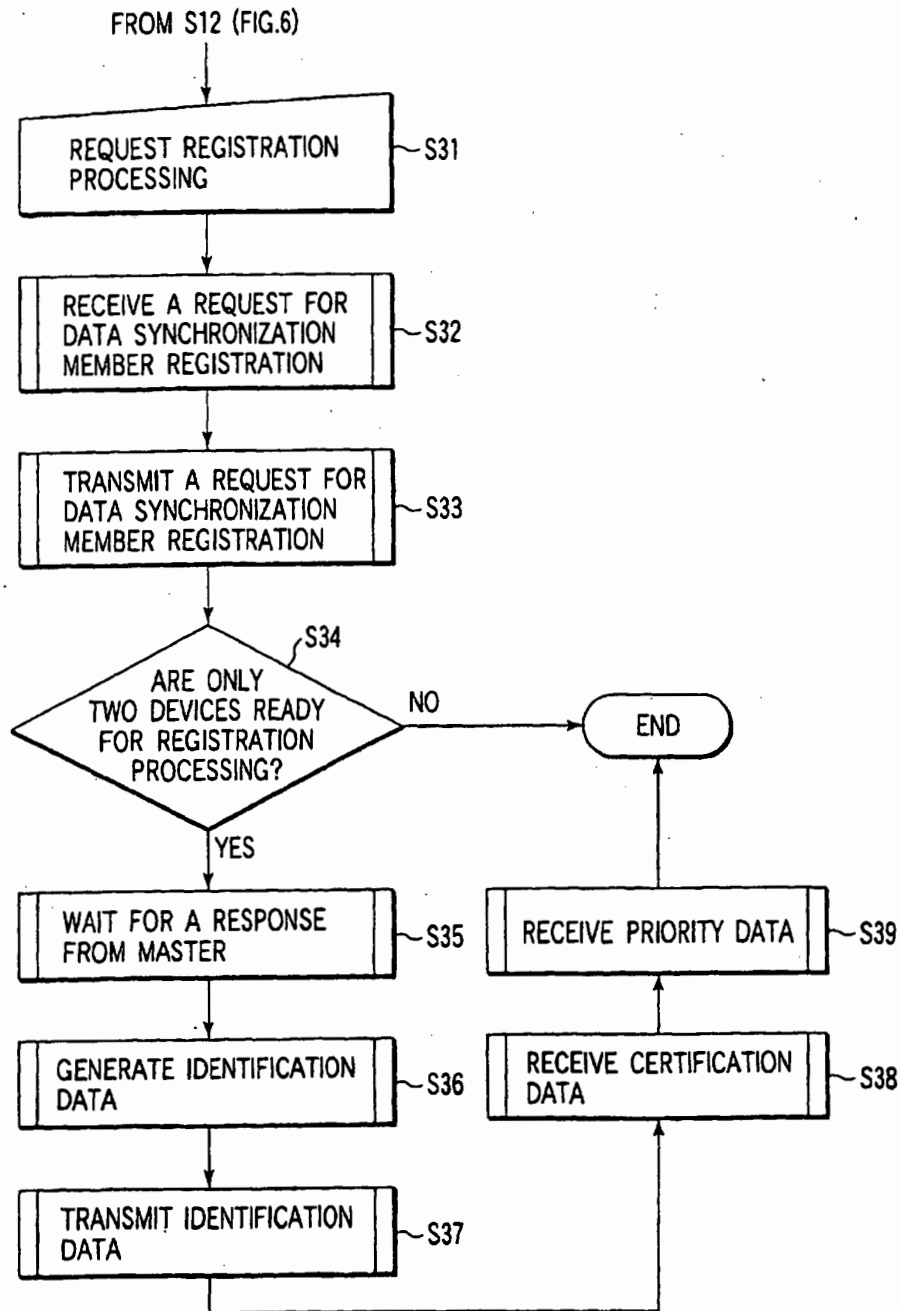


FIG. 7

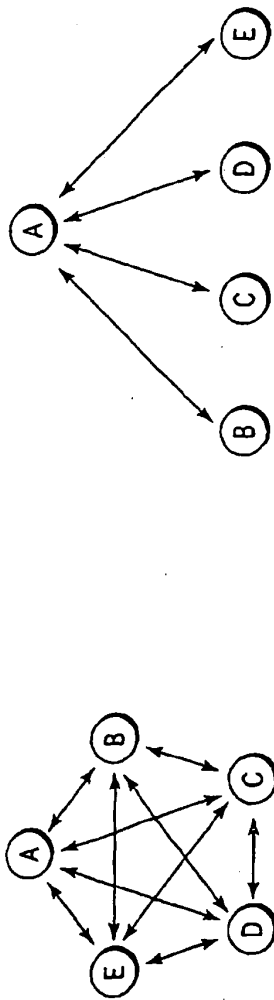


FIG. 9

FIG. 8

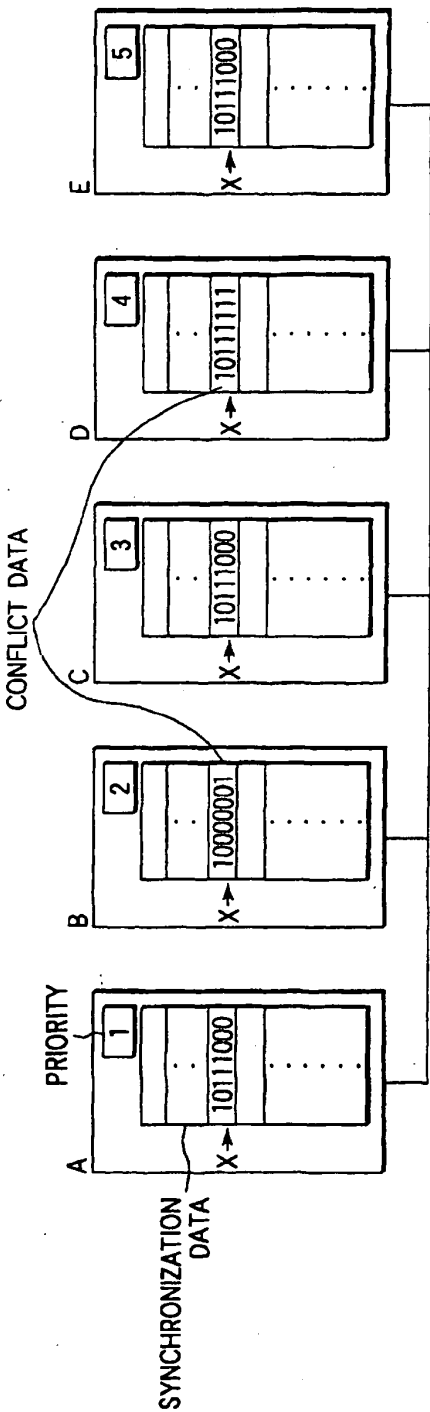


FIG. 10

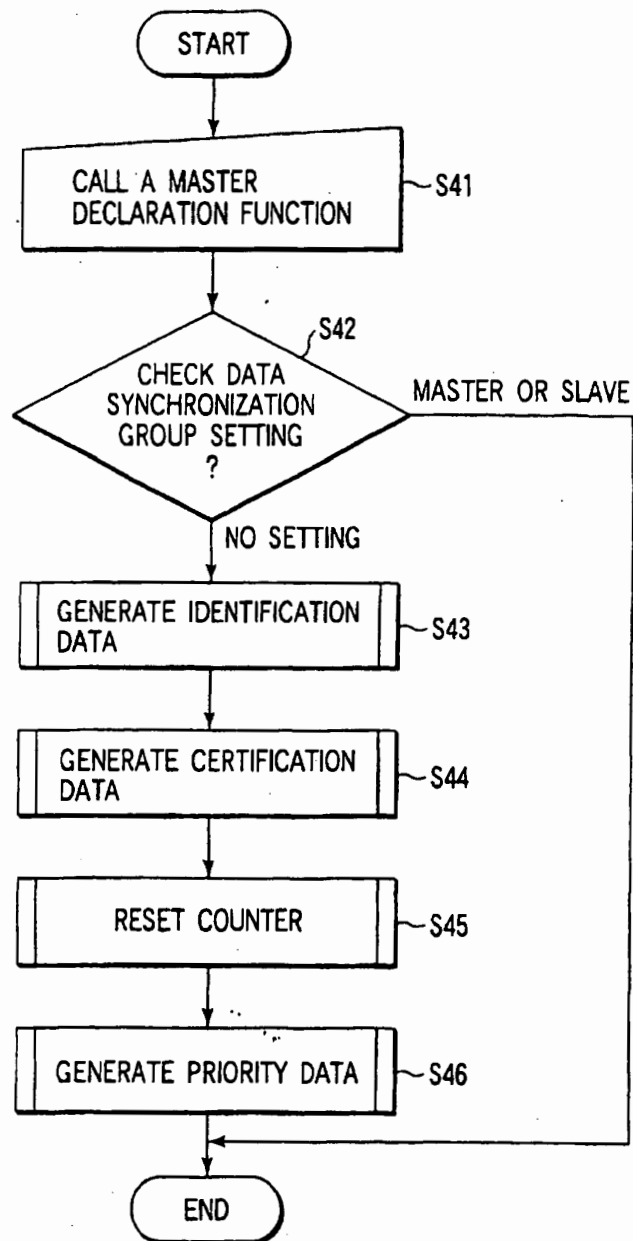


FIG. 11

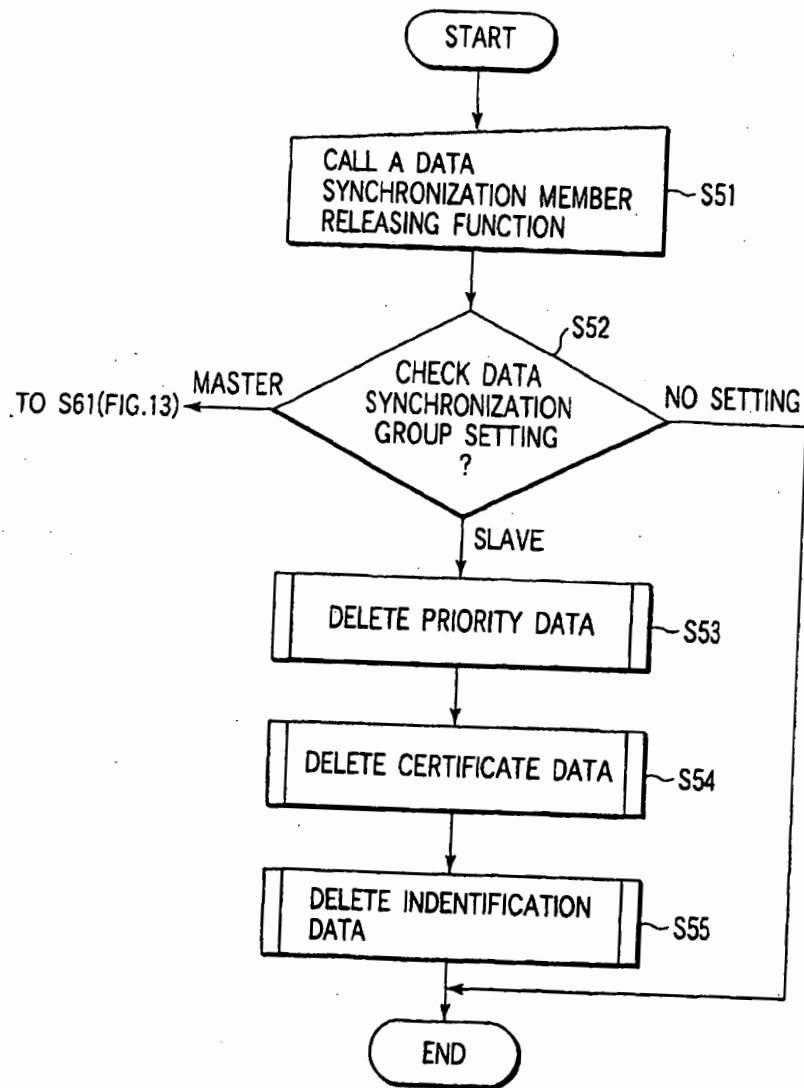


FIG. 12

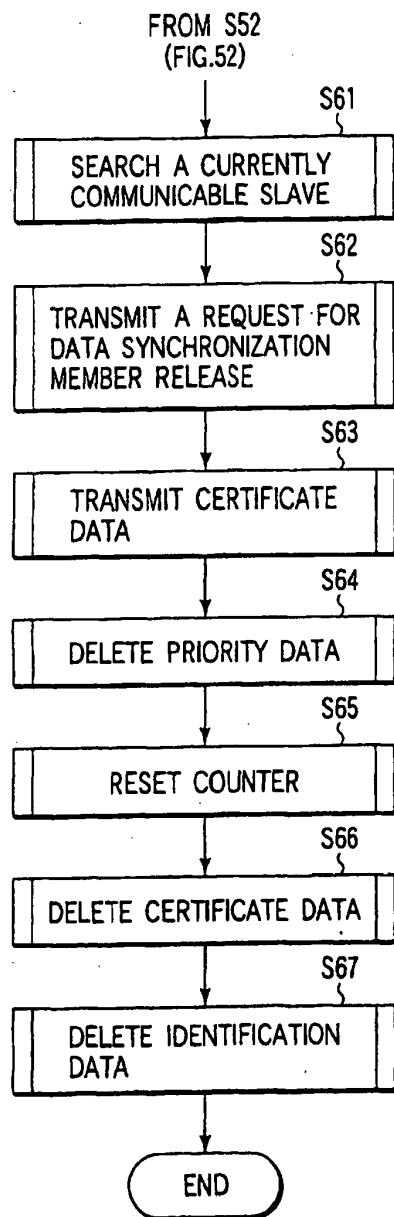


FIG. 13

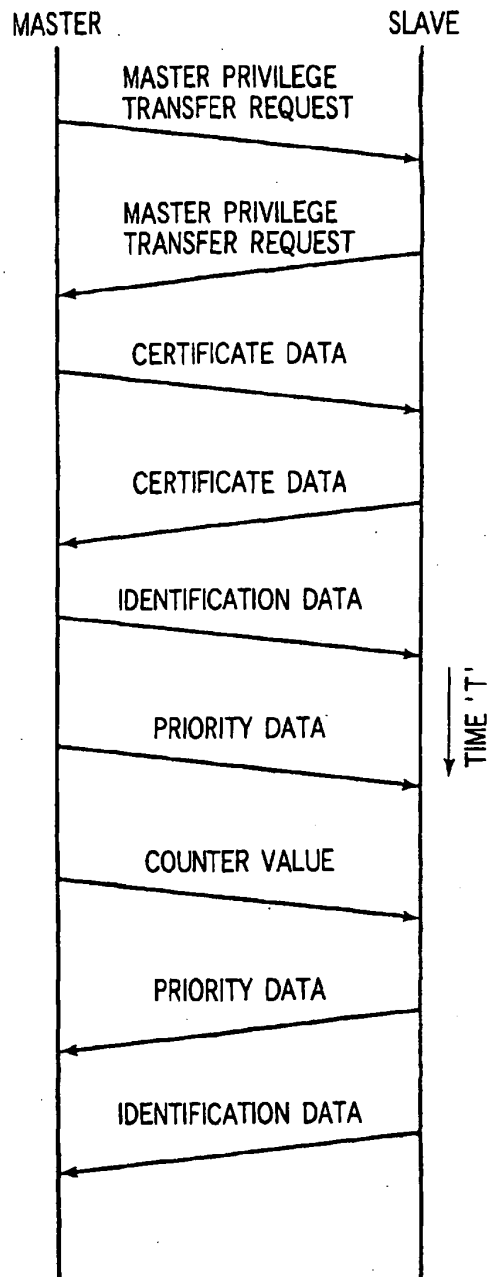
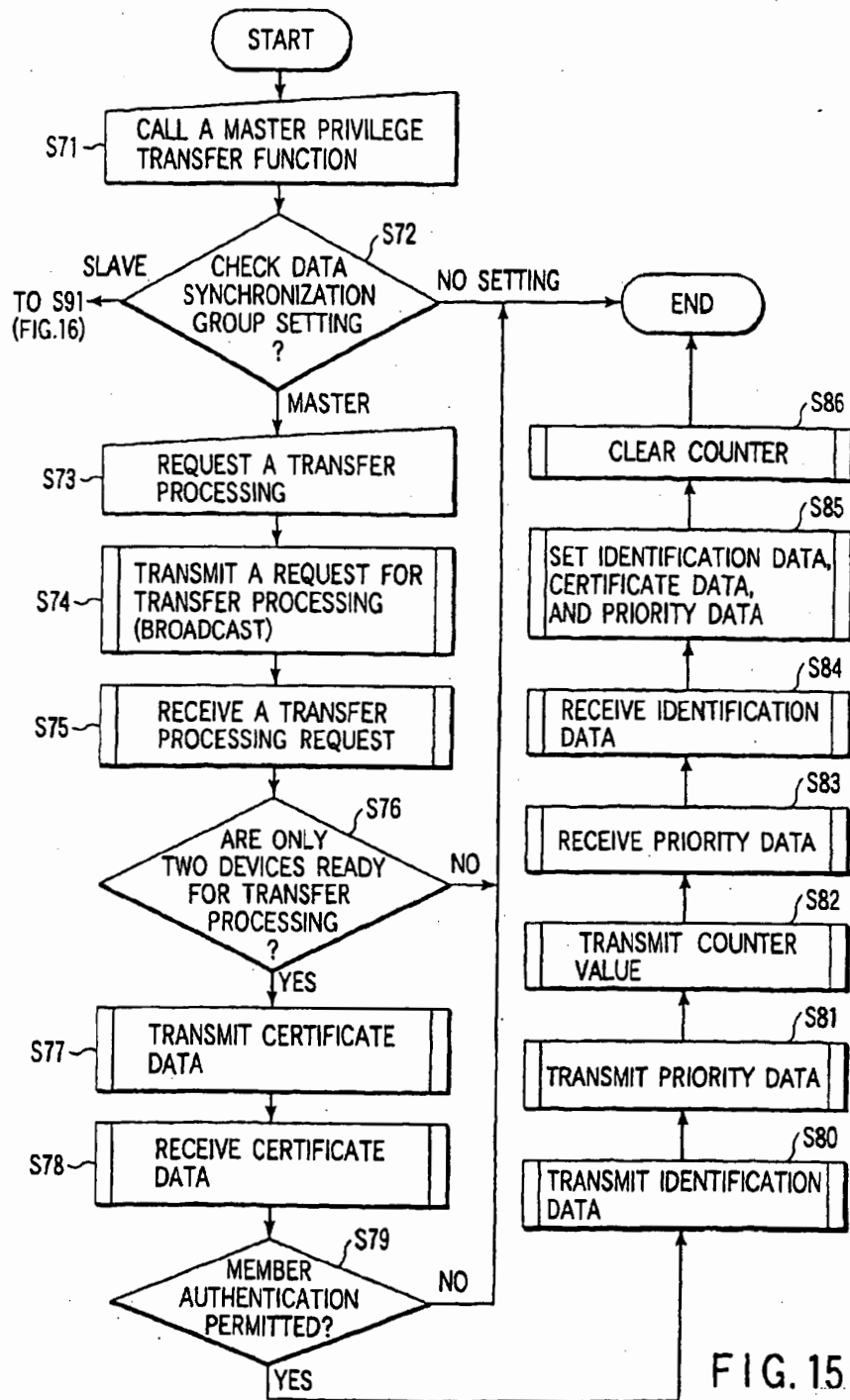
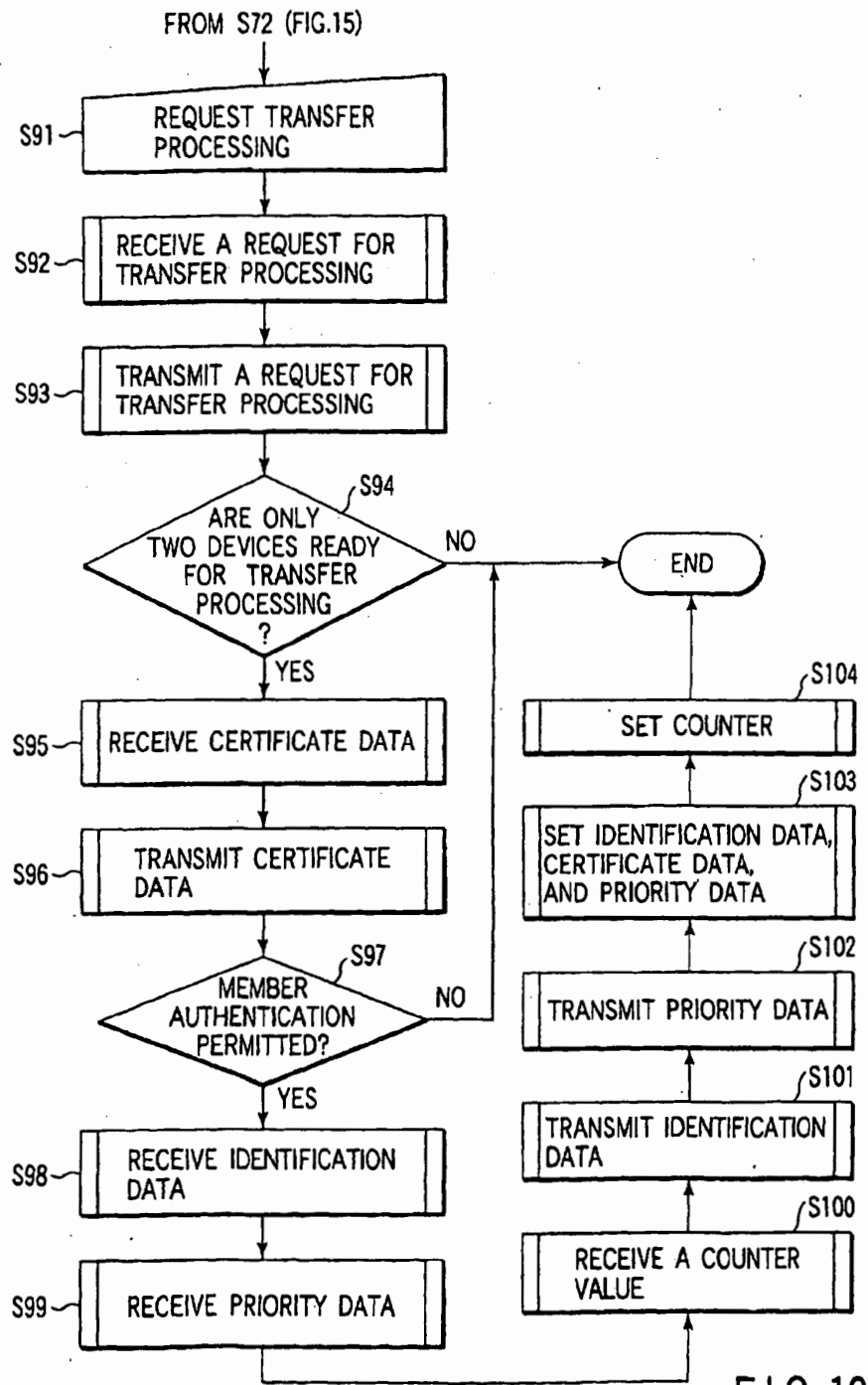


FIG. 14





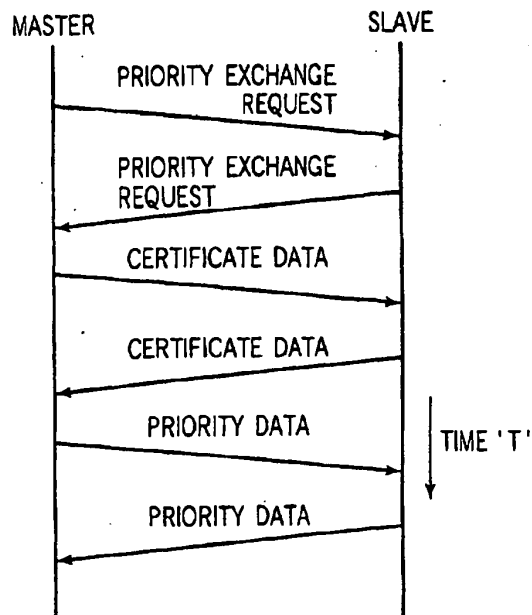


FIG. 17

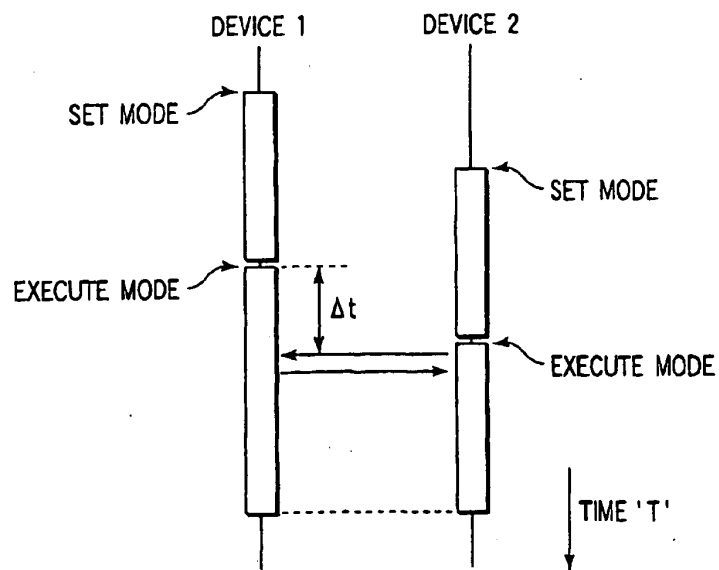
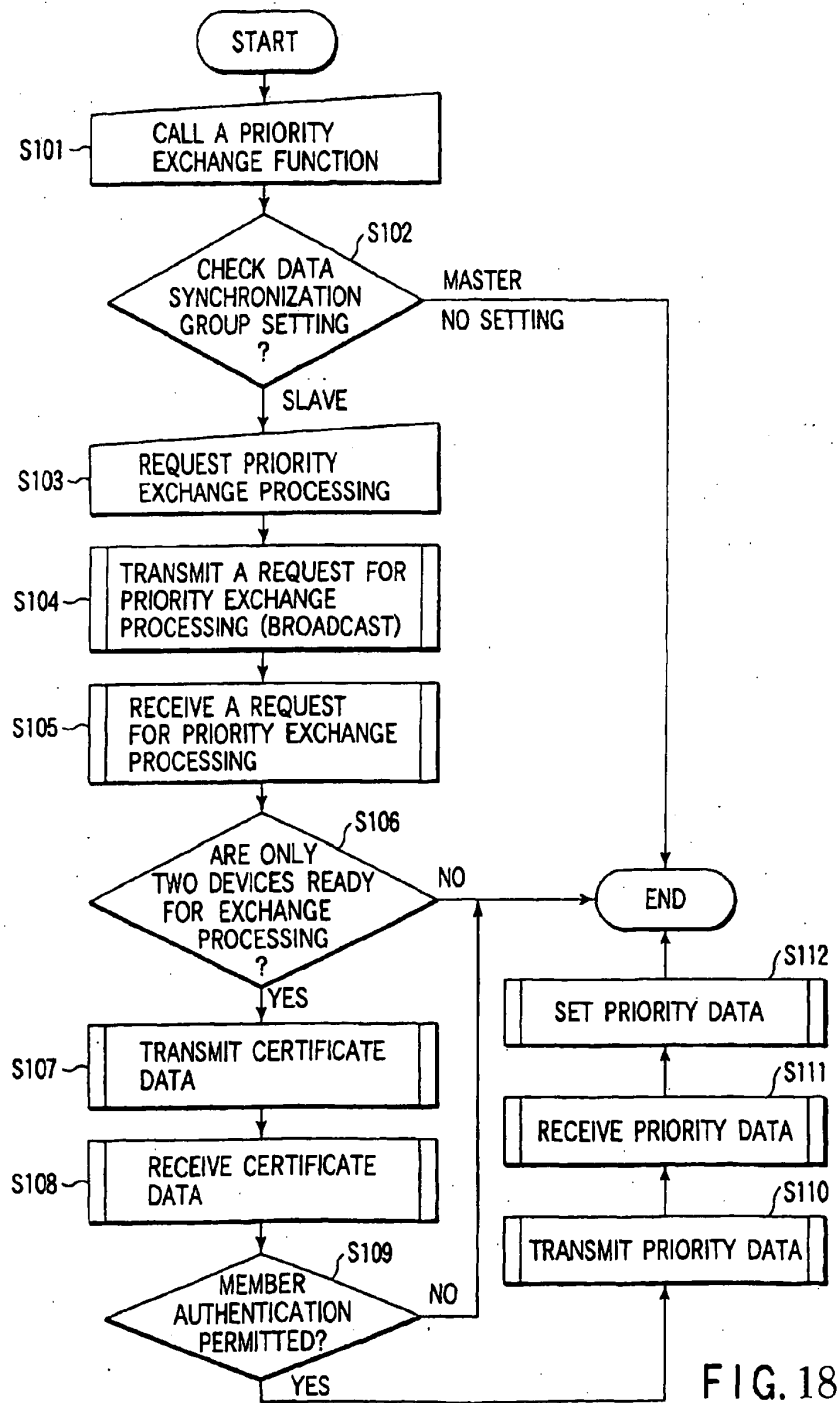
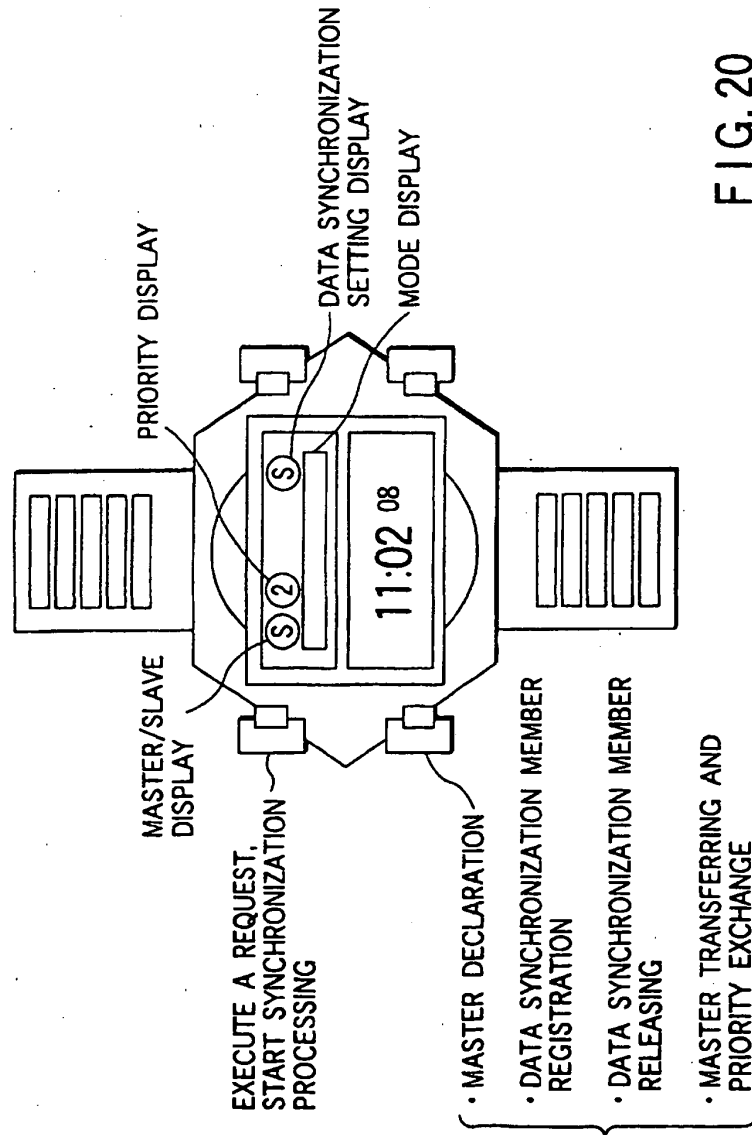


FIG. 19





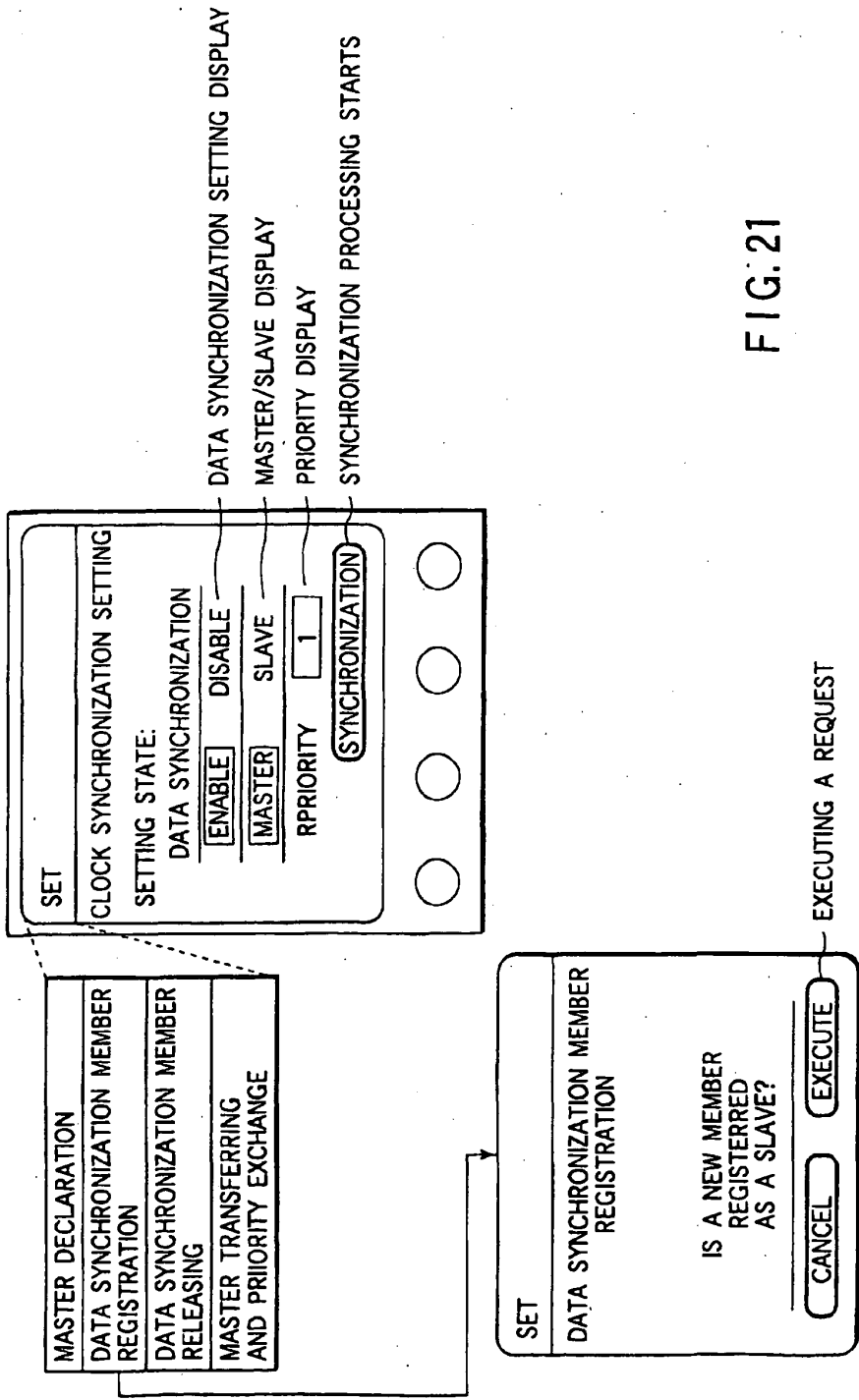


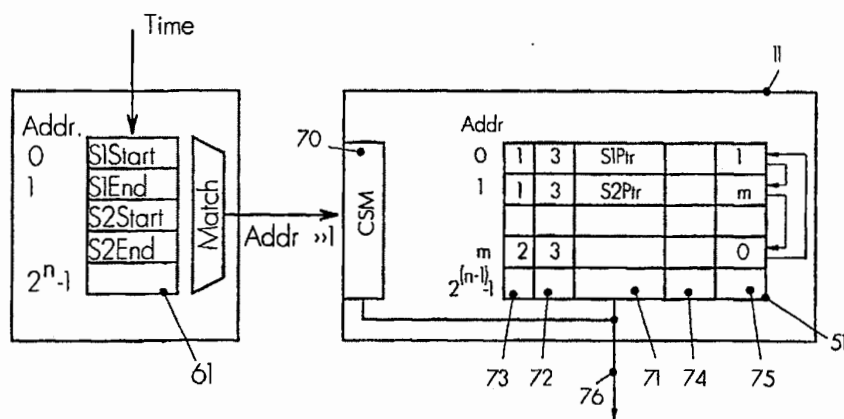
FIG. 21



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 17/30		(11) International Publication Number: WO 95/25313
A1		(43) International Publication Date: 21 September 1995 (21.09.95)
(21) International Application Number: PCT/EP94/00833		(81) Designated States: JP, US, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
(22) International Filing Date: 16 March 1994 (16.03.94)		
(71) Applicant (for all designated States except US): INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; Old Orchard Road, Armonk, NY 10504 (US).		
(72) Inventors; and (75) Inventors/Applicants (for US only): KAISERSWERTH, Matthias [DE/CH]; Leigrubenstrasse 2, CH-8805 Richterswil (CH). RÜTSCHÉ, Erich [CH/FR]; 1500, chemin des Adrets, F-06370 Mouans-Sartoux (FR).		
(74) Agent: BARTH, Carl, O.; IBM Research Laboratory, Intellectual Property Dept., Säumerstrasse 4, CH-8803 Rüschlikon (CH).		
Published With international search report.		

(54) Title: METHOD AND APPARATUS FOR SYNCHRONIZATION AND SCHEDULING OF MULTIPLE DATA STREAMS AND REAL TIME TASKS



(57) Abstract

The present application concerns a method and an apparatus for the synchronization and the scheduling of multiple data streams and real time tasks. In accordance with the present invention, pointers pointing to the information streams to be synchronized and processed are kept in the table (51) of a selector (11). The time marks (S1Start etc.) of said information streams are stored in a table (61) of a timer. The system time is checked against said time marks and if a time mark has been reached, the table (51) of said selector (11) is updated. If the status of an information stream is such that its trigger condition is fulfilled, the pointer to said information stream is made available at the output of the selector (11). This pointer is then used to retrieve the respective information stream and to forward it to a playout device.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgyzstan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LU	Luxembourg	TD	Chad
CS	Czechoslovakia	LV	Latvia	TG	Togo
CZ	Czech Republic	MC	Monaco	TJ	Tajikistan
DE	Germany	MD	Republic of Moldova	TT	Trinidad and Tobago
DK	Denmark	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	US	United States of America
FI	Finland	MN	Mongolia	UZ	Uzbekistan
FR	France			VN	Viet Nam
GA	Gabon				

1

DESCRIPTION

5

**Method and Apparatus for Synchronization and Scheduling
of Multiple Data Streams and Real Time Tasks****TECHNICAL FIELD**

10 The present invention concerns a method and apparatus for the synchronization and the scheduling of multiple data streams as well as for the scheduling of tasks in operating systems with hard real-time requirements.

15

BACKGROUND OF THE INVENTION

Multimedia information is composed of several objects that constitute a document. These objects could be of different data types, such as text, still images, video, and sound. In order for these objects to be meaningful to a user, there must be a logical and temporal structure of the objects constituting the documents' contents. To ensure this logical and temporal structure, e.g. when displaying a video sequence on a display device and providing a soundtrack for this video sequence via an adapter card to a speaker, the respective data streams have to be synchronized and scheduled before being sent to the output devices. The same principle applies to other related multimedia information which has to be synchronized in communicating with other devices or users (e.g. teleconferencing), before processing and presentation.

30 Synchronization and scheduling is a critical task for a wide spectrum of applications ranging from computer-integrated telephony, text with voice annotations, any kind of multi media systems, and interactive services (e.g. interactive television) to cooperative teleworking on documents including

1 foils and video, and distributed games. A further application, where
information streams have to be synchronized, is known as medical imaging,
for which high-resolution images must be distributed reliably and fast
throughout a hospital.

5

To date, the synchronization is typically performed in software. However, if
the scheduling intervals are too short, or if too many information streams
must be synchronized, a software solution, which is very CPU intensive, is
often the limiting factor as far as processing speed is concerned.

10

In real-time systems time-critical tasks have to be scheduled reliably and
fast. In the present context, any system in which the time at which an output
is produced is significant, is referred to as real-time system. The lag from
the input time of a real-time system to its output time, i.e. the delay, must be
15 sufficiently small for acceptable timeliness. With other words, a real-time
system has to respond to an externally-generated input stimuli within a
finite and specified period. In hard real-time systems, i.e. in those systems
where it is absolutely imperative that responses occur within a specified
deadline, scheduling of time-critical tasks is of particular interest. However,
20 the present invention is not limited to hard real-time systems. It might be
used in soft real-time systems, too. Such a soft real-time system would still
function correctly if deadlines are occasionally missed, and a hardware
solution in accordance with the present invention might be to costly under
certain circumstances.

25

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a method and apparatus
30 for off-loading the synchronization of multiple information streams from the
processor of the system handling these information streams.

1 It is an object of the present invention to provide a method and apparatus
for fast and reliable synchronization and scheduling of multiple information
streams.

5 It is another object of the present invention to provide a method and
apparatus for scheduling tasks or processes in real-time systems.

The above objects have been accomplished by handling pointers pointing to
the information streams to be synchronized and by maintaining a table with
10 status information of each information stream. The time marks of said
information streams are stored in a table and the system time is checked
against said time marks. If a time mark has been reached the status
information of the respective information stream is updated. If the status of
an information stream is such that its trigger condition is fulfilled, the
15 pointer to said information stream is made available for playout of this
information stream.

20

25

30

1 DESCRIPTION OF THE DRAWINGS
AND NOTATIONS USED

The invention is described in detail below with reference to the following
5 drawings:

FIG. 1 shows a system, in accordance with the present invention, for the

10 **FIG. 2** illustrates the synchronization of multiple data streams.

FIG. 3 illustrates the synchronization and play out of two data streams when the respective trigger condition is fulfilled.

15 **FIG. 4** illustrates how a priority value might be increased with advancing time.

FIG. 5 shows the two core components of an apparatus for synchronization and scheduling of data streams and real time tasks, in accordance with the present invention.

FIG. 6A shows a first exemplary embodiment of the timer.

FIG. 6B shows a second exemplary embodiment of the timer.

FIG. 7 shows an embodiment of the timer and selector, in accordance with the present invention.

FIG. 8 shows an example of an 8-bit timer where every second bit is used.

1 GENERAL DESCRIPTION

2 In multimedia systems multiple data streams must be synchronized and
3 scheduled for playout to, for example, a speaker 15 and a video display 14,
4 as illustrated in Figure 1. The synchronization of multiple data streams is
5 illustrated in Figure 2. The three streams are synchronized at so-called
6 synchronization points (SyncPoint) 22. A stream may be synchronized to
7 one or multiple other streams or to time stamps 23 defined by an external
8 clock. This synchronization is typically performed in software, as mentioned
9 in the introductory portion of the present description. In the following, a
10 mechanism is presented which performs synchronization and output
11 scheduling of multiple data streams in hardware without central processing
12 unit (CPU) interaction. The same mechanism can also be used for
13 scheduling tasks in real-time systems.

14
15 A data stream is a sequence of data units. In Figure 3, the synchronization
16 of two data units is illustrated. The SyncPoints are translated to time marks
17 relative to the system which processes the received data streams. The
18 SyncPoints might be translated to time marks by means of a clock or
19 counter of said system. In the example given in Figure 3, a data unit from
20 stream 1 cannot be played out before a time mark 30 (S1Start) and it must
21 be played out before a time mark 31 (S1End). The same applies to the
22 second data stream (stream 2). As illustrated in Figure 3, at least two
23 different states (WAIT, READY) can be associated with a data stream.
24 Before SxStart of a data stream x has been reached, the respective data unit
25 is in the WAIT state. After SxStart and before SxEnd it is in the READY state.
26 It is conceivable to assign further states, such as a DELAYED state, to these
27 data streams. The DELAYED state might be associated to a data unit after
28 SxEnd.

29
30 In order to be able to synchronize two data units for playout, trigger
31 conditions are defined in accordance with the present invention. A typical
32 trigger condition might be:

1

IF both data units are in the READY state
THEN they can be played out.

5

The trigger conditions are checked and as soon as the condition is fulfilled, the data units are played out. This concept is schematically illustrated in Figure 3, where means for checking the trigger condition 32 are shown.

10

In general, two or more data units can be related in time by trigger conditions, which can be an arbitrary logical combination of the different states of the respective data units. Fulfilling a given trigger condition then initiates the appropriate operation on the units.

15

The synchronization can be further enhanced with a priority value P to distinguish between data units (or tasks) which fulfill a trigger condition to select an older or more important stream (or task). This, for example, allows to prioritize streams that should not be delayed, whereas other data streams might be delayed. In Figure 4 an example is shown how the priority value P of a data unit (or task) can be increased in time. In this example, three different states (WAIT, READY, DELAYED) are associated with the data stream. The priority value P might also be increased or decreased by interaction of a processor or upon receipt of a feedback signal of a playout device.

20

25

A first embodiment of the present invention is schematically illustrated in Figure 1. According to this example, two data streams are received by means 18 for extraction of time marks. The first data stream originates from a storage disk 16, whereas the second data stream is sent via a network 17 to said means 18 for extraction of time marks. After extraction of the time marks, the data units of the two data streams are stored in buffers 19 and 20, and the respective time marks are provided to a timer 10. The buffer addresses are forwarded to a selector 11. In addition, a system clock or counter 13 is connected to said timer 10, and a processor, e.g. a CPU, might

30

1 be linked to the timer 10 and selector 11. Two playout devices, a display unit 14 and a loudspeaker 15, are connected via an interfacial playout device to the buffers 19 and 20.

5 The two core components of the present invention are the timer 10 and the selector 11. Their operation is described in connection with Figure 5.

The timer 10 holds a table 50 with entries concerning the time marks of each data unit received and stored in a buffer. These time marks are relative to a time signal either being generated by a system clock 13 or a counter. Each time mark reached is then output to the selector 11. The table 50 is written by an external processor. The selector comprises a table 51 where for each data unit in a buffer a pointer to said data unit, i.e. the respective buffer address, the state of said data unit, and the trigger condition are held. Data units to be synchronized are linked into a circular list via their table entries. For each time mark received from the timer 10, the selector 11 updates the status field of the concerned data unit and then checks the trigger condition of all data units to be synchronized with said data unit. Once the trigger condition is fulfilled, the pointer to the data unit (or units) is made available at the output of selector 11. The output of the pointer then triggers retrieval and playout of the data unit by means of the playout device 21. Table 51 of the selector 11 might be maintained and updated by an external processor, e.g. CPU 12.

25 The timer 10 reads the current system time and returns the address of a unit pointer in the selector 11. This function can be provided in several ways. First, a solution based on a content addressable memory (CAM) is presented which is suitable for a high resolution timer but relatively expensive. The second solution is less expensive and is suitable for lower resolution timers and for implementing timer priority as described below. The two solutions are shown in Figures 6A and 6B, respectively.

1 The CAM timer 60 is illustrated in Figure 6A. It comprises a CAM 61. The
start and end time mark for each data unit to be scheduled are written into
consecutive addresses of the CAM 61. The addresses then differ only in the
least significant bit. Thus, for a CAM of 2^n addresses, $n-1$ bits determine the
5 address of the scheduling unit pointer in the selector table 51. The width of
a CAM entry is determined by the resolution of the time signal received at
input 62. For a 32-bit time signal also a 32-bit wide CAM 61 is required.

For lower time resolution a timer 63 with counter 64, as shown in Figure 6B,
10 might be employed. In this case the difference between the current time
and the time mark is calculated and its two's complement is written to the
timer table 65. The counter 64 increments each entry for each time tick at
input 66. If the carry bit of an entry becomes set, its address is shifted right
by one bit and output to the selector 11. As the counter 64 continues to
15 increment the entries, the content of the entry (with its carry bit = 1) can
then be used to reflect the time passed since reaching the start and/or end
time. The counter value thus describes a time based priority. With this
counter value a priority value P might be generated. The width of a timer
entry can range from 8 to 32 bits depending on the timer resolution and the
20 required timer differences. A free flag 67 might be provided in order to
mark free table entries in table 65. This flag 67 is used by the counter to
determine which entries it must increment.

For both schemes, a FIFO can hold free timer table entries to reduce the
25 management load of the external processor. The processor gets a free entry
from the FIFO and writes delayed entries, or entries which are freed, to the
FIFO. The selector might directly write delayed entries to this FIFO.

It is obvious that the present invention is not limited to the above two
30 implementations of timers since other timers are conceivable which are well
suited, too.

- 1 The selector 11 comprises a table 51 and a comparator state machine (CSM)
 70. The table 51 has $2^{(n-1)}$ entries (rows or columns). The exemplary
 implementation of the selector 11, given in Figure 7, shows a table 51 with
 $2^{(n-1)}$ rows. Each row has a pointer field 71 holding the pointer, i.e. the
 5 buffer address S1Ptr, to the data unit to be scheduled, a p-bit wide priority
 field 72, a k-bit wide status field 73, a 2-bit wide trigger field 74, and an
 (n-1)-bit wide synchronization field 75. The width of said status field 73
 depends on the number of different states. In case of three different state
 (WAIT, READY, DELAYED) the status field 73 is 2-bits wide. The
 10 synchronization field 75 holds a pointer linking all those entries with which a
 data unit must be synchronized into a circular list. This circular list is
 schematically illustrated by means of arrows on the right hand side of table
 51. The CSM 70 receives the address of a unit pointer from the timer 10
 and updates the state of the status field 73 in this row. The CSM 70 then
 15 follows the linked list of synchronizations entries and computes whether the
 trigger condition is fulfilled by comparing the status fields 73 with the trigger
 fields 74. The trigger condition might also be checked by means of simple
 logical elements such as AND-gates, OR-gates etc.
- 20 In the following, it is assumed, without loss of generality, that the trigger
 condition is fulfilled when all entries in the respective trigger fields 74 are in
 the READY state. In this case, the trigger state fields are redundant and
 might be omitted.
- 25 Once all the entries linked by the synchronization field 75 are READY, the
 CSM 70 outputs the entries according to their priorities, if any. With $p=4$, 16
 priority classes can be distinguished. The width w of the scheduling unit
 pointer is determined by the memory (buffer) architecture of the system. The
 width s of the synchronization field 75 is determined by the size of the
 30 selector table 51, with $s=n-1$.

In Figure 7 the problem shown in 3 is elaborated. The time marks 30
 (S1Start) and 31 (S1End) of data unit S1 are written to the addresses 0 and 1

1 of the CAM 61. The address of the pointer to the data unit S1 is the CAM
address shifted right by one bit. An external processor 12 writes the time
marks to the timer table, i.e. the CAM 61 and the entries to the selector
table 51. For a data unit to be scheduled which requires priority, the priority
5 field 72 is set. In this example a 4-bit wide priority field 72 is used. For each
time mark reached, the state, reflected by the status field 73, is updated. If
the state is READY, the CSM 70 reads the synchronization field 75 and
follows the linked list to determine whether the state of all other entries
fulfills the trigger condition. If the trigger condition is reached, the unit
10 pointer with the highest priority is made available at output 76 first. If
multiple units have the same priority, or in case that no priority fields are
provided, the unit pointers might either be output in parallel or in a
consecutive manner, e.g. by order of their position in the selector table 51. It
is obvious that the priority fields 72 can be omitted, too.

15 In many systems, multimedia as well as real-time, the scheduling of actions,
data streams, tasks, processes, or the playout of data units is determined by
a priority which increases with time. If a counter timer, as illustrated in
Figure 6B, is used as the timer component 10, the entry in the timer table 65
20 also describes its priority. This priority can be used in the priority field 72 of
the selector 11 to achieve time dependent scheduling. The four most
significant bits of the timer entry are used to distinguish again 16 priorities.
A different set of bits might be used if the priority must increase
non-linearly. Figure 8 shows an example of an 8-bit timer where every
25 second bit is used to get a priority mechanism that maps the 256 timer
values to 16 non-linear priority values P.

According to the present invention, a synchronization point (SyncPoint) is
set up by writing the time marks to the free entries in the timer table and
30 setting up the associated entries in the selector table. For each stream this
takes one write operation to the timer and to the selector table, plus the
operations to set up the selector table entry. The actual scheduling takes
place in hardware without any further intervention of an external processor.

1 For a solution in software a similar number of setup operations would be
necessary but the search in a list of timers would take much longer.
Typically timers involve interrupt processing which is very time consuming
in particular in modern RISC processors with a large register file and
5 instruction pipeline. In the present invention these operations are avoided
and implemented in hardware. Additionally, the present invention can
handle timers with much finer granularity than any software solution.

The present invention, which has been described by means of a multimedia
10 related example, can be used in various multimedia communication
subsystems, e.g. for multimedia multiparty teleconferencing, or multimedia
protocol adapters. In ST-II and RTP protocol systems, for example, a playout
time stamp can be extracted from the protocol header and used in the
present scheduler (ST-II is the second version of an experimental Internet
15 stream protocol, and RTP is a transport protocol for real-time applications).
The present invention might also be used in multimedia bus systems which
provide a time signal. Data units to be triggered at a specified time are
written to the scheduler and the multimedia bus provides the timer input to
the timer unit. The invention can also be used in a multimedia adapter such
20 as the IBM ActionMedia II adapter (see IBM ActionMedia II Technical
Reference, IBM 04G5144, 2nd Edition, June 1992, Mechanicsburg), or the
MWave System (see "MWave Multimedia System", Technical Brief, Texas
Instruments, 1992) to synchronize multiple data streams. The term
ActionMedia is a trademark of the Intel Corporation and MWave is a
25 trademark of the IBM Corporation.

As already mentioned in the introduction, the present invention might also
be used in real-time systems for the scheduling of time-critical tasks or
processes. For these kind of application there is no synchronization field in
30 the table of the selector required. The CSM selects from the selector table
the entry with the highest priority. The scheduling unit pointer, i.e. a pointer
pointing to a task control block or to executable code, is then output. The
timer priority mechanism described above can be used to implement

1 scheduling policies with an explicit deadline, e.g. the earliest-deadline-first
or least-slack time scheduling algorithm as disclosed in the book "Real-Time
Systems and their Programming Languages", by A. Burns and A. Wellings,
Addison-Wesley Publishing Company, International Computer Science
5 Series, 1990. An external processor calculates the difference to the time
marks and writes it to the timer table. The entries are then used to derive
the priority of a job.

10

15

20

25

30

1

CLAIMS

1. Apparatus for synchronization of at least two information streams which comprises:
 - 5 a) a timer (10) for holding a first table (50) with time marks for each one of said information streams, and for making each time mark reached available at the timer output,
 - b) a selector (11) for holding a second table (51) with
 - 10 • pointer fields (71) carrying pointers to each of said information streams,
 - status fields (73) carrying information concerning the status of each of said information streams, the status field of an information stream being updated each time a time mark of said information stream is reached and made available at said
 - 15 timer output,
 - c) means for checking whether a trigger condition of an information stream is fulfilled by watching the corresponding status field, and
 - d) means for making the pointer to an information stream whose
 - 20 trigger condition is fulfilled available for playout of said information stream.
2. The apparatus of claim 1, wherein said second table (50) further comprises:
 - 25 • trigger fields (74) carrying information concerning the trigger conditions of each of said information streams.
3. The apparatus of claims 1 or 2, wherein said second table (51) further comprises:
 - 30 • synchronization fields (74) carrying, by means of a linked list, entries concerning information streams to be played out in a certain relationship to each other.

- 1 4. The apparatus of claims 1, 2, or 3, wherein said second table (51)
further comprises:
- priority fields (72) carrying priority values (P) such that the
information stream with highest priority is played out first if the
5 respective trigger condition has been fulfilled.
5. The apparatus of claim 4, further comprising means for increasing said
priority values (P) with advancing time.
- 10 6. The apparatus of any of the preceding claims, further comprising:
- means (18) for extracting said time marks from incoming
information streams,
 - buffer means (19, 20) for storing said information streams,
 - playout means (21) for retrieval of said information streams stored
15 in said buffer means (19, 20) and for playout to the respective
playout devices (14, 15).
7. A multimedia system comprising an apparatus in accordance with any
of the preceding claims.
- 20 8. Apparatus for scheduling at least two real time tasks or two processes
which comprises:
- a) a timer (10) for holding a first table (50) with time marks for each
one of said tasks or processes, and for making each time mark
25 reached available at the timer output,
 - b) a selector (11) for holding a second table (51) with
 - pointer fields (71) carrying pointers to task control blocks or
executable code,
 - means for making a pointer to a task control block or an
30 executable code available at an output of said selector each
time a time mark of said task or process is reached in said
timer.

- 1 9. The apparatus of claim 8, wherein the pointer to a task or process with highest priority is made available first.
- 5 10. Method for synchronization of at least two information streams, characterized by the following steps:
- a) checking whether a time mark of one of said information streams has been reached,
 - b) updating the status information of said information stream whose time mark has been reached,
 - 10 c) determining whether the trigger condition of said information stream is fulfilled by watching said status information, and
 - d) providing a pointer to said information stream for further processing if the trigger condition of said information stream is fulfilled.
- 15 11. The method of claim 10, wherein the playout of information streams, which have to be played out in a predefined relationship to each other, is scheduled by means of a linked list being organized such that each time when a status information of one of said information streams is updated, the status information of the other information streams are
- 20 checked, and said information streams played out when all trigger conditions are fulfilled.
- 25 12. The method of claim 10, wherein a priority value (P) is assigned to each of said information streams, the information streams with highest priority being played out first.
13. The method of claim 12, wherein the priority value (P) of an information stream is increased with advancing time.
- 30 14. The method of claim 10, further comprising the following steps:
- extracting and/or synchronizing the time marks of said information streams,
 - storing said information streams in buffers,

- 1 • making the addresses of said buffers available as pointers for later
retrieval of said information streams,

15. The method of claim 14, wherein said information stream is retrieved
5 from said buffer by means of said pointer, prior to being send to a
 playout device.

10

15

20

25

30

1/4

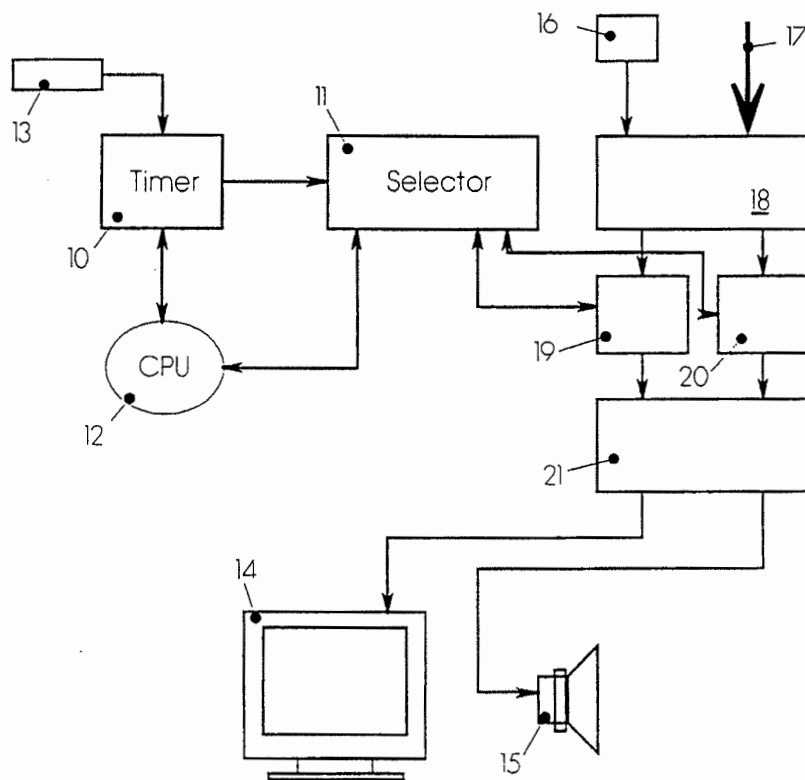


FIG. 1

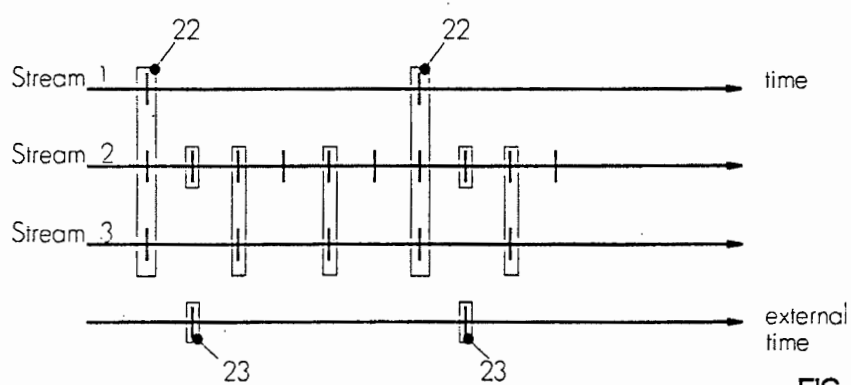


FIG. 2

2/4

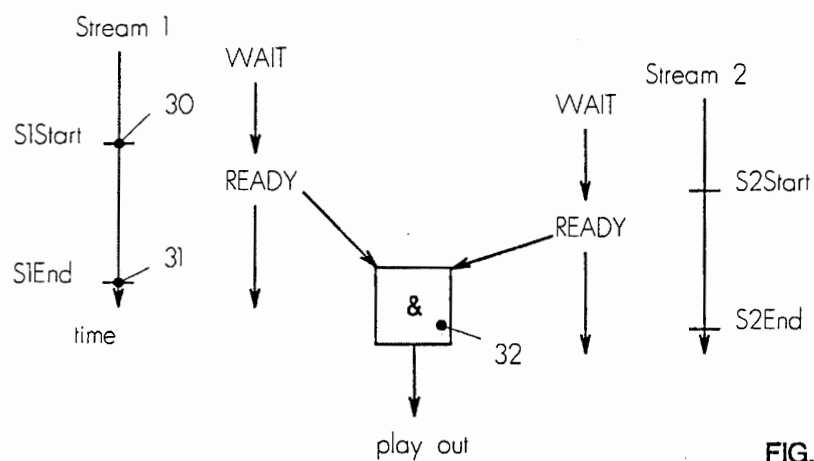


FIG. 3

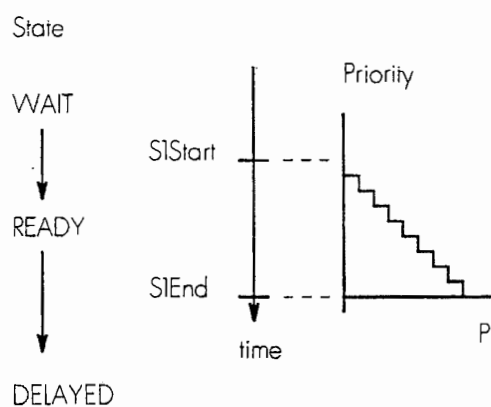


FIG. 4

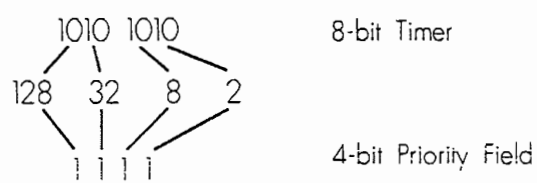


FIG. 8

3/4

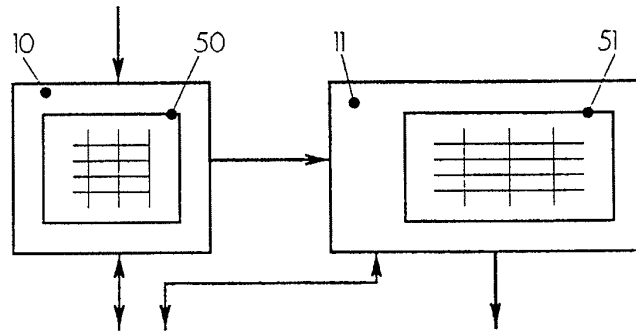


FIG. 5

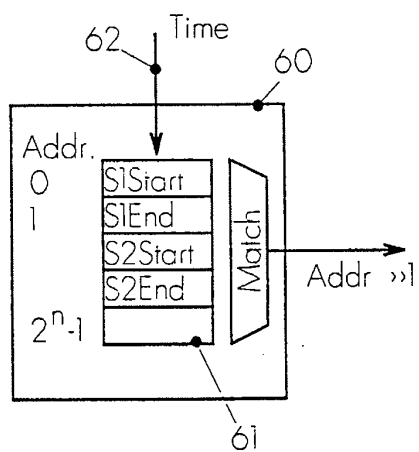


FIG. 6A

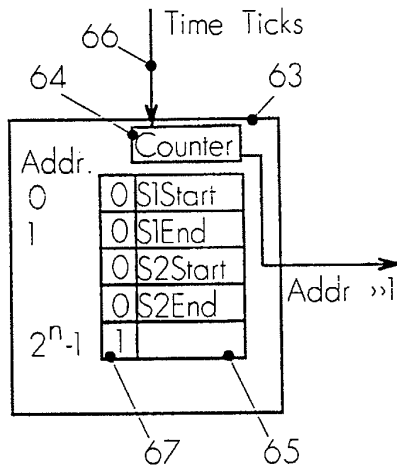


FIG. 6B

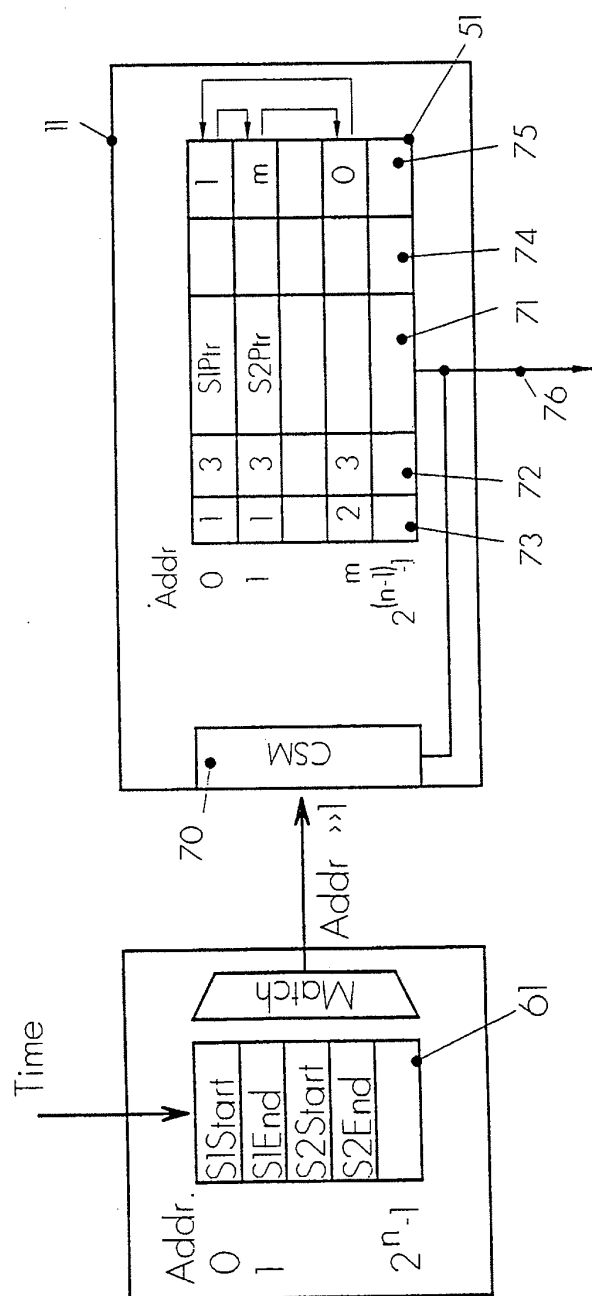


FIG. 7

INTERNATIONAL SEARCH REPORT

International Application No
PCT/EP 94/00833

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/30

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 6 G06F

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP,A,0 239 884 (I.B.M. CORPORATION) 7 October 1987 see the whole document ---	1,8,10
A	EP,A,0 550 197 (I.B.M. CORPORATION) 7 July 1993 see the whole document ---	1,8,10
A	COMPUTER COMMUNICATIONS, vol.16, no.3, March 1993, GUILDFORD GB pages 168 - 176 P.V. RANGAN ET AL. 'Techniques for multimedia synchronization in network file systems' see the whole document --- -/--	1,8,10

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in 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 document 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

10 November 1994

Date of mailing of the international search report

01.12.94

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl,
Fax (+ 31-70) 340-3016

Authorized officer

Katerbau, R

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inter. Appl. No.
PCT/EP 94/00833

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	COMPUTER, vol.24, no.10, October 1991, LONG BEACH US pages 51 - 57 D.P. ANDERSON ET AL. 'A continuous media I/O server and its synchronization mechanism' see the whole document ---	1,8,10
A	BYTE, vol.18, no.2, February 1993, LONDON, GB pages 225 - 230 A.W. DAVIS ET AL. 'The Mac goes to the movies' see the whole document -----	1,8,10

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

page 2 of 2

INTERNATIONAL SEARCH REPORT

Information on patent family members

Intern. Application No

PCT/EP 94/00833

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A-0239884	07-10-87	US-A- 4893256	09-01-90
		DE-D- 3787553	04-11-93
		DE-T- 3787553	11-05-94
		JP-B- 6054469	20-07-94
		JP-A- 62239233	20-10-87

EP-A-0550197	07-07-93	US-A- 5333299	26-07-94
		CA-A- 2084574	01-07-93
		JP-A- 5265778	15-10-93

Form PCT/ISA/210 (patent family annex) (July 1992)

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

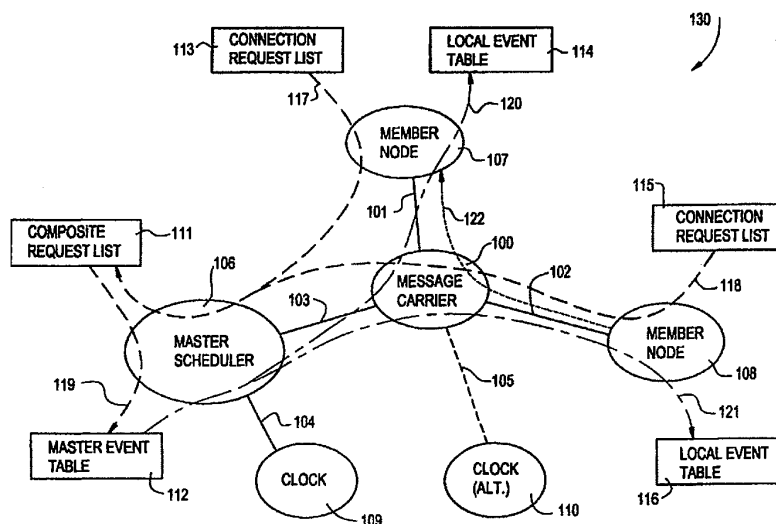
(51) International Patent Classification ⁶ : G06F 9/46		A1	(11) International Publication Number: WO 99/61985
			(43) International Publication Date: 2 December 1999 (02.12.99)
(21) International Application Number: PCT/US99/06634		(81) Designated States: CA, CN, JP, MX, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).	
(22) International Filing Date: 26 March 1999 (26.03.99)		Published With international search report.	
(30) Priority Data: 60/086,737 26 May 1998 (26.05.98) US			
(71) Applicant: THE DOW CHEMICAL COMPANY [US/US]; 2030 Dow Center, Midland, MI 48674 (US).			
(72) Inventors: WOODS, Randy, D.; 3788 Old Pine, Midland, MI 48642 (US). JACHIM, David, M.; Apartment L, 3063 Signature Boulevard, Ann Arbor, MI 48103 (US). DUPREE, Wayne, P.; 5301 Stony Creek, Midland, MI 48640 (US). VERNIERS, Gerrit, H.; Heirweg 88, B-9180 Moerbeke-Waas (BE). CHURCHILL, Stephen, G.; 2588 Waskevich, Midland, MI 48642 (US). FERNANDEZ, George, P.; 449 North 5 Mile, Midland, MI 48640 (US).			
(74) Agent: SCHULTZ, Dale, H.; Patent Dept., P.O. Box 1967, Midland, MI 48641-1967 (US).			

(54) Title: DISTRIBUTED COMPUTING ENVIRONMENT USING REAL-TIME SCHEDULING LOGIC AND TIME DETERMINISTIC ARCHITECTURE

(57) Abstract

A system is presented for a plurality of computer nodes in mutual data communication to execute logic in a coordinated manner. The approach features aligning the definition of the time in the nodes so that each node has essentially the same definition of time, defining a schedule of communications for each computer node, with the schedule having future moments defined for initiating each communication, communicating each schedule to its respective node, and initiating each communication at the time defined in the schedule. The computer network has

a message carrier supporting multiple communication paths; a plurality of computers coupled to the message carrier, a clock system for maintaining time synchronism among the computers; and a scheduler coupled to the message carrier for determining the sequence of communication between the computers. The message carrier establishes communication between selected computers based on the time-deterministic tasks being performed by the computers. A computer node in the plurality of computer nodes executes logic in coordination with the other nodes by defining a communication request list of each communication needed by the computer node with any other computer node respective to a future time period, sending the communication request list to the master scheduler, receiving back a communication schedule back from the master scheduler, defining a local task schedule from the communication schedule, and then executing the communication at the scheduled time.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

DISTRIBUTED COMPUTING ENVIRONMENT USING REAL-TIME SCHEDULING LOGIC
AND TIME DETERMINISTIC ARCHITECTURE

This invention is directed to the field of data processing control systems, particularly to
5 integrated systems where a plurality of computers execute their logic in a coordinated
manner within a network. The primary application of the system described in this
specification is for computerized control of machines deployed on a large physical scale (for
example a chemical manufacturing facility), although benefits to endeavors related to plural
computer processors, parallel computer processors, and networked computer processors
10 are also enabled.

When computational demands or geographical constraints require the physical distribution of
computing facilities (for example such as for process control in a large, geographically
dispersed chemical plant), implementation factors (such as product quality) may benefit from
15 centralized coordination of the computing tasks. This invention describes a system and
method for enabling a plurality of geographically dispersed computers to be networked for
enabling time coordinated inter-activity. This resultant network enables task coordination
and facilitates the implementation of other qualities beneficial to the task being performed by
the system - qualities such as redundancy, security, and high throughput.

20 One embodiment of a system showing such substantive real-time networking to a set of
physically distributed remote field units for use in a process control system is described the
United States Patent 5,428,769 issued on Jun. 27, 1995 and entitled "Process control
interface system having triply redundant remote field units" to Glaser, Robert S.; Hoy, Robert
25 S.; Fernandez, G. Paul; and Grai, Timothy J. While the system is adequate for many
process control situations, the networking methodology effectively addresses each system in
each time period on the presumption that there is a need to do so; however, many of these
addressing cycles are unnecessary if all variables for exchange are stable -- a more elegant
scenario from the standpoint of network efficiency would provide for communications only
30 when the status of either the inputs or the outputs require change.

Physical distribution of computing facilities combined with centralized coordination of the
computing tasks is also a driving concept in high performance processors characterized by
parallelism. Vector processors, as discussed in sections 7.1 and 7.2 ("Vector Processors")
35 of Computer Architecture A Quantitative Approach by John L. Hennessy and David A.

Patterson (Morgan Kaufmann; San Mateo, California; 1990; pages 351-363) are one manifestation of such systems, and processor-memory pair approaches such as the Cm* design discussed from pages 11 to 18 of "Parallel Processing - The Cm* Experience" by E. F. Gehringer, D. P. Siewiorek, and Z. Segall (Digital Press via Editorial Inc.; Rockport, Massachusetts; 1987) provide a second example of these types of approaches.

Centralized coordination of computing tasks in a scheduled context for the future also draws on concepts from traditional project scheduling such as is discussed from pages 569 to 584 of the text Productions and Operations Management by J. William Gavett (Harcort, Brace, and World; New York; 1968).

Process control systems development is characterized by an ever-increasing incorporation and management of issues at large in the system being controlled, requiring that a comparably larger amount of logic on a comparably larger amount of data be executed in a time frame which is not permitted to expand. This can be achieved to some degree by hardware which processes logic more quickly and holds more data in an addressed physical store, but there are substantial costs in pursuing this solution. What is needed to resolve the ability of future systems to effectively expand their comprehensive treatment of the control situation is to truly provide a solution path for enabling a very large number of distributed computers to execute massive amounts of logic in a mutual manner which is time-synchronous across an entire network. The present invention provides such a method and system.

The invention provides a computer implemented method for enabling a plurality of computer nodes in mutual data communication to execute logic in a coordinated manner by

- (a) aligning the definition of time between the computer nodes so that each computer node has essentially the same "understanding" of time;
- (b) defining a schedule of communications for each computer node, where the schedule has a different time in the future defined for initiating each respective communication;
- (c) communicating each schedule to its respective computer node; and

(d) initiating each communication from each node at the time defined in its schedule.

- 5 The described embodiment provides a computer network architecture, having a message carrier with a plurality of computers coupled to it; a clock system (accessible by all computers in the network) for maintaining time synchronism among the computers; and a scheduler for determining the sequence of communication between the computers by causing the message carrier to establish communication between selected computers based
10 on time-deterministic tasks being performed by the computers.

The described embodiment further provides a global time system for aligning the sense of time in the set of computers to a globally defined reference.

- 15 The described embodiment further provides a computer implemented method for enabling a plurality of member nodes to execute logic in a coordinated manner.

The described embodiment further provides for concurrency in communications between computers via the message carrier.

20

The described embodiment further provides provides a computer implemented method for scheduling tasks for resource objects in a plurality of resource objects functioning as a collective group.

- 25 The described embodiment further provides a method by which different computers can align their definition of time by adjusting their oscillator's rate of oscillation.

- The computer network architecture is enabled with a message carrier supporting multiple communication paths; a plurality of computers coupled to the message carrier, with each
30 computer being programmed to perform time-deterministic tasks by cycling through a plurality of different operating states, including a communication state; a clock system accessible by all the computers for maintaining time synchronism among the computers; and a scheduler coupled to the message carrier for determining the sequence of communication between the computers by enabling the message carrier to establish communication

between selected computers based on the time-deterministic tasks being performed by the computers.

- 5 A computer node (element for computing with an independent central processing unit) in the plurality of computer nodes can execute logic in coordination with the other nodes by being connected to a message carrier along with the other computer nodes where there is also a provision for (a) master scheduling capability in one computer node to establish a master scheduler; (b) defining, in each computer node, a communication request list of each communication needed by the computer node with any other computer node respective to a
- 10 future time period; (c) sending of the communication request list from each computer node to the master scheduler via the message carrier; (d) receiving of, in each computer node, a communication schedule from the master scheduler via the message carrier; and (e) defining, in each computer node, a local task schedule from the communication schedule.
- 15 The above approach enables a plurality of member nodes in a system to execute a large amount of logic in a coordinated manner on a group of distributed platforms.

In performing the above, the step of defining a reference moment time value is performed in one embodiment by the master scheduler.

20

In performing the above, the step of defining a communication request list in each member node in one embodiment limits the communications to a maximum number in the future time period and each communication has an duration time attribute and the value of the duration time attribute is less than a fixed maximum value in order to achieve efficient operation.

25

Use of a global time system in one embodiment enables a universal understanding of time across a large number of different computer nodes (member nodes) in an extended system.

- 30 The method for scheduling involves a process of scheduling tasks for resource (member) objects in a plurality of resource objects functioning as a collective group; and this computer implemented process (called an "apparent critical resource" process) involves the steps of:

defining a set of future tasks for the collective group;

defining each resource object needed for each future task in the set of future tasks;

5 defining each interaction needed between any two resource objects
respective to the set of future tasks;

defining a use instance for each resource object in achieving each interaction where the use instance has an use duration attribute value;

10 defining an accessing resource object and an accessed resource object
respective to each use instance;

15 determining a summed accessing time value for each accessing resource
object as a sum of all use duration attribute values for all use instances
respective to that resource object when it is the accessing resource object;

determining a summed accessed time value for each accessed resource
object as a sum of all use duration attribute values for all interactions
between the accessed resource object and the accessing resource object;

20 defining a set of sequential future time increments within a scheduling period
in a schedule;

25 scheduling, into the earliest available time in the first future time increment, a
future resource interaction event based on the first available future task in the
set of future tasks between

(a) the accessing resource object having the highest summed
accessing value respective to the set of future tasks and

30 (b) the accessed resource object having the highest summed
accessed value respective to the set of future tasks (if a use instance
exists between the accessed resource object and the accessing
resource object -- the future resource interaction event in the schedule
35 record has an initiation time equal to the earliest available time in the

schedule, an accessing resource object identifier attribute respective to the accessing resource object, an accessed resource object identifier attribute respective to the accessed resource object, and the aforementioned use duration attribute value)

5

removing from scheduling candidacy the use instance respective to the accessed resource object in the scheduled future resource interaction event and the accessing member object identifier attribute in the scheduled future resource interaction event to redefine the set of future tasks to those future tasks for which interactions have not been scheduled;

10

repeating the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future communication task, and removing all use instances respective to scheduled resource objects until all use instances have been scheduled which can be scheduled into the first future time increments; and

15

recurrently executing the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future resource interaction event, removing all resource objects respective to scheduled resource objects, and repeating such steps until all resource interaction events have been scheduled which can be scheduled into each future time increment and until all resource interaction events have been scheduled into the scheduling period.

20

25

In one embodiment, the message carrier has a plurality of channels for permitting concurrency in the communication events, the step of scheduling being respective to the plurality of channels as a group wherein the earliest available time is respective to any channel in the message carrier.

30

In some embodiments, each member node has a variable speed oscillator for adjusting a local clock to align with a global clock on the network. Time alignment logic and oscillator rate adjustment logic for execution by the computer circuitry are also provided to achieve rapid time alignment of the particular member node (computer node).

The message carrier used in the described embodiment is enabled with a network topology which can be manifested (in the preferred embodiment) with a crossbar switch, or, in an alternative embodiment, with a radio frequency communications system.

5

Additional features and advantages of the present invention will become more fully apparent from a reading of the detailed description of the preferred embodiment and the accompanying drawings.

10 Figure 1 presents a diagram of the data flows between the various elements in a distributed network.

Figure 2 depicts a distributed network of computing elements which are resident in physically different locations throughout the world. This figure includes a network of computing
15 elements, a master scheduler, and a common source of time.

Figure 3 presents a diagram of a network in a bus structure.

Figure 4 shows a distributed computing environment where the network is implemented in a
20 ring topology.

Figure 5 presents a diagram of a network in a switched star topology which is a preferred network architecture embodiment.

25 Figure 6 presents a diagram of data structures communicated between the computing elements and a master scheduler.

Figure 7 depicts a flowchart of the process implemented in the master scheduler in the scheduling of communications between two separate computing elements.

30

Figure 8 presents a simplified diagram of the scheduling logic (software in execution by electrical circuitry) implemented in the master scheduler in the scheduling of communications between two separate computing elements.

Figure 9 presents a detailed flowchart of the scheduling logic which is followed at the master scheduler.

5 Figures 10 presents a bar chart depicting the first scheduled communications between two separate computing elements after the first connection is scheduled.

Figure 11 presents a bar chart depicting the scheduled communications between computing elements after all connections are scheduled which can be scheduled in the first step.

10 Figure 12 presents a bar chart depicting the scheduled communications between computing elements after all connections have been scheduled.

Figure 13 provides a detailed composition of the Message Carrier, the Master Scheduler, the Clock, and the Member Nodes.

15 Figure 14 shows the relationships between the technologies which have been combined in the preferred embodiment, with emphasis on those most critical to the application.

20 Figure 15 presents a flowchart of the logic which executes in each of the computing elements in order to facilitate communication between the member nodes.

Figure 16 depicts a diagram of the data flows between the various elements in a distributed network where communications are facilitated by radio frequency.

25 As an introduction to this part of the specification, the system and method of the present invention implement centralized scheduling of cooperatively performed tasks, time deterministic event operations, and effective real-time synchronization and alignment of oscillators respective to the plurality of computers (also referenced as "member nodes") on the network. In this regard, the time deterministic network involves communicating between
30 separate computing elements based on time rather than on some event (examples of events are the reception of a token or the detected availability of a particular bus). Centralized scheduling of cooperatively performed tasks, time deterministic event operations, and effective real-time synchronization and alignment of oscillators respective to the plurality of computers on the network requires that the computers share an essentially common view of
35 the datalogical representation of an actual moment in real-time. This means, for instance

and assuming for this example (without limitation to the scope of the present invention) a system wide-precision requirement of one millionth of a second, that two different computers in the system essentially perceive that, when the moment of time in real-time defined by an undisputed source respective to the network is 08:00:0000000 hours, that the first of the two computers has a datalogical representation of the moment in real-time between 07:59:9999995 and 08:00:0000005 and that the second of the two computers also has a datalogical representation of the moment in real-time time between 07:59:9999995 and 08:00:0000005; as can be appreciated, the difference in the datalogical representation of the moment in real-time between the two computers is no more than 1 millionth of a second.

When the computers in a distributed computing network share an essentially common view of the datalogical representation of an actual moment in real-time, they can then be programmed to perform specific tasks based on time. Tasks which involve multiple computers can then be scheduled centrally and a basis is established for time-dependent activities to be invoked on geographically dispersed computers in a deterministic manner. In this regard, determinism in real-time operations characterizes the occurrence of a predefined event at a predefined moment of real-time.

The use of such time coordination in communications between computers also facilitates the employment of active redundancy. In active redundancy, there are two or more computers which are each executing essentially identical logic, exchanging the results from these executions, optionally arbitrating the independently derived results to come to a commonly understood mutual conclusion for further use, and acting to concurrently effect the result. This process proceeds most effectively when data received by each computer is shared with the other computer(s) executing the same logic prior to arbitrating the results, so there must be some mechanism by which the time to begin the arbitration process is mutually determined. In the present invention, the mechanism for providing time synchronization between the redundant computers and to schedule the tasks related to a communication instance is (1) to provide an essentially identical datalogical representation of the moment in real-time time among the computers; (2) to use a scheduling approach to define in each computer the moment in time when the communication instance in that computer will occur; and (3) to then to implement the appropriate sending, receiving, and linkage tasks in each system involved in the communication instance at the time defined for the communication instance. In this way, the time at which the referenced arbitration task will be performed is

pre-determined after the communications tasks have been scheduled (as opposed to the approach of waiting to determine the start time of the arbitration task after the communication tasks have actually executed).

- 5 The scheduling approach is implemented by execution of scheduling logic. The concept of logic relates to software (a computer program) whose purpose is achieved in execution by an enabling system of electrical circuitry (usually denoted by the term "computer"). The scheduling logic used for the scheduling of communications is based on analysis of demands for the resources (for example specific computers, data space within each
- 10 computer, execution time in each computer, and specific circuits within each computer) which are being scheduled. This scheduling logic is used in the scheduling of network communications, but it has further applicability in the scheduling of any resources which require interaction among one another, and it is especially suited for real-time applications. The scheduling logic is further designed to be efficient respective to the resources it is
- 15 responsible for scheduling and to be predictable as to the amount of time required to generate its output, a schedule, from the inputs (that is a collection of connection requests between any two computers on this network). In achieving this task, the scheduling logic schedules the available resources with the highest demand for the time that the schedule is being created. Reflecting the nature of (1) evaluating the resources with the highest demand
- 20 at each point in the schedule and (2) giving precedence to the resources which have the highest demand, the particular scheduling logic method described in this specification is also referred to herein as Apparent Critical Resource, or ACR, scheduling logic. In this regard, in the preferred embodiment, the scheduling logic is not iterative in the sense that, once a particular task is scheduled, it is NOT removed from the schedule in order to pursue an
- 25 alternative to its placement in the schedule. It is this quality which enables the preferred ACR scheduling logic type of scheduling logic to be well-suited for real-time applications.

In the application of scheduling logic to the scheduling of network communications, there are transmission resources and reception resources which are to be considered. Each time a

30 network communication is scheduled, a transmission resource is associated with a reception resource.

In the scheduling of network communications, there are responsibilities which are handled by the master scheduler and those which are handled by the member nodes.

Figure 1 presents a data flow diagram 130 of the general elements and associated data flows between the various general elements in a distributed network and shows a message carrier 100, a first member node 107 and a second member node 108, a master scheduler 106 for coordinating communications between member nodes 107, 108 via data communication path 101 and data communication path 102, clock alternatives (as clock 109 and alternative clock 110) for achieving synchronization between member nodes 107, 108 (via data communication path 104, data communication path 105, data communication path 101, and data communication path 102), and the primary data tables and exchanges required to interact effectively in this distributed computing environment (via data communication path 118, data communication path 117, data communication path 119, data communication path 120, and data communication path 121).

The clock 109 is distributed to the member nodes through the message carrier from the master scheduler (alternatively, the message carrier 100 could distribute an alternative local clock 110 to the member nodes 107, 108 and to the master scheduler 106). This permits all nodes attached to the message carrier to share a common view of time. In this regard, clock 109 represents a global clock for use around the physical globe (for example in North America, in Europe, and in Asia) where the alternative clock 110 represents a clock which has been designated for the network referenced only by message carrier 100 and where a computer executing aspects of message carrier 100 arbitrarily defines time for its affiliated computers in the distributed network. Clock 109 or clock 110 periodically output a common clock signal via either linkage 104 or 105, respectively.

Each member node 107, 108 is responsible for receiving the common clock signal; effectively "synchronizing" (aligning the definition of time in the node to a close tolerance with the definition of time respective to the common clock signal to create a system time base operating in synchronism with an established clock frequency and anchor point) to that common clock signal; receiving an appropriate local event table 114, 116 from the master scheduler 106; formulating a respective task execution list based on the received local event table 116, 114 and any additional tasks which are not described in the local event table 114, 116; executing the task execution list; generating the request list of communications 113, 115; and transmitting the request list of communications to the master scheduler 106. Each member node 107, 108 generates a connection request list 113, 115 based on the tasks required of the node 107, 108, and forwards these requests at a prescribed time through the

message carrier 100 to the master scheduler 106. The master scheduler 106 combines the connection request lists 113, 115 from the member nodes into a composite request list 111, which it uses to generate a communications schedule represented in the master event table 112. This master event table 112 is then broken apart into sections applicable to each of the member nodes 107, 108 and sent through the message carrier 100 (also at prescribed times) to the member nodes 107, 108 for storage in local event tables 114, 116. The local event tables 114, 116 are used in the member nodes 107, 108 to control timing for applicable time dependent tasks, including communications with other member nodes.

Figure 2 depicts a distributed network 220 of computing elements which are resident in physically different locations throughout the world to show an instance of the situation abstracted in the data flow diagram 130 of Figure 1. This figure includes a network of computers 201-214 and two sources of time in clock 203 and clock 214 representing time sources respective to the data flow diagram clocks 109, 110; in this regard, Figure 2 shows clock 203 as the basis for time definition for all systems in North America and Eurasia while clock 206 defines a basis for time definition for the less globally dispersed domain of computers 206, 207, 208, 209, and 214 in Africa. It should be noted that clock 203, then, "maps" to clock 109 in the data flow diagram context of Figure 1 and that clock 214 "maps" to clock 110 in the data flow diagram context of Figure 1. Figure 2 thereby depicts two possible configurations for distributed computing environments (facilitating the associated message carriers 100, master schedulers 106, and clocks 104, 105). A distributed computing environment can be located in a geographically limited area or dispersed throughout the world. When spread across large geographical areas, the clock 203 is preferably a satellite timing distribution system such as the Global Positioning System.

Figure 3 presents a diagram of a network in a bus structure 330. Figure 3 shows the fundamental components of Figure 1 except that the message carrier 100 is a bus 300 connecting the member nodes 107, 108, to the master scheduler 106.

Figure 4 shows a distributed computing environment where the network is implemented in a ring topology 430. Figure 4 shows the fundamental components of Figure 1 except that the message carrier 100 is a ring enabled by ring segments 402, 403, 405, and 401 to effectively connect the member nodes 107, 108, to the master scheduler 106.

Figure 5 presents a diagram of a network in a switched star topology 530 which is a preferred network architecture embodiment. Figure 5 also shows the fundamental components of Figure 1 except that the message carrier 100 is explicitly a switched network implemented in a star topology which connects the member nodes 107, 108 and the master scheduler 106, and that this master scheduler 106 is embedded into the network switch 501.

5 This diagram shows the preferred embodiment as it could be used in a large manufacturing process (a manufacturing process referencing the effecting of goods manufacture in an apparatus whose movable components are modified in real-time using control signals from a computer control system executing logic reading measurements from the apparatus and

10 outputting the control signals) through a large number of input/output processing devices and computing platforms in a potentially geographically distributed arrangement. The star topology network 530 enables the reading of measurements and outputting of control signals respective to the apparatus for manufacture through I/O (input signal/output signal) interface 510, I/O interface 511, I/O interface 522, I/O interface 519, I/O interface 512, I/O interface

15 513, I/O interface 520, and I/O interface 521. I/O interface 510, I/O interface 511, I/O interface 512, and I/O interface 513 are all joined via ring breakout 508 and ring breakout 509 to control computer 504 and historian computer 505 through local network 515. In a similar manner, I/O interface 522, I/O interface 519, I/O interface 520, and I/O interface 521 are all joined via ring breakout 518 and ring breakout 517 to control computer 506 and

20 historian computer 507 through local network 514. I/O interface 510, I/O interface 511, I/O interface 512, I/O interface 513, ring breakout 508, ring breakout 509, control computer 504, historian computer 505, and local network 515 operate with local logical integrity and high real-time datalogical cohesion. In a similar manner, I/O interface 522, I/O interface 519, I/O interface 520, I/O interface 521, ring breakout 518, ring breakout 517, control computer 506,

25 historian computer 507, and local network 514 operate with local logical integrity and high real-time datalogical cohesion. Cohesion between

(a) I/O interface 510, I/O interface 511, I/O interface 512, I/O interface 513, ring breakout 508, ring breakout 509, control computer 504, historian computer 505, and

30 local network 515

and

(b) I/O interface 522, I/O interface 519, I/O interface 520, I/O interface 521, ring breakout 518, ring breakout 517, control computer 506, historian computer 507, and local network 514

5 via messaging is enabled by network switch 501 which implements (1) communications for provision of an essentially identical datalogical representation of the moment in real-time time among the computers; (2) master scheduler 106 processes to define for control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and network switch 501 each moment in time when
10 each communication instance between any two of computer 506, historian computer 507, control computer 504, historian computer 505, and user interface 503 VIA THE USE OF network switch 501 will occur; and (3) appropriate sending, receiving, and linkage tasks respective to components involved in the communication instance at the time defined for the communication instance. Network support 502 is also coordinated as a node into switched
15 star topology 530 as a single source of information for defining acceptable communication partners in switched star topology 530 and also for accumulating and providing access to diagnostic information in switched star topology 530. Network switch 501 references network support 502 for a listing of acceptable communication partners in switched star topology 530 and confirms communication instance requests against this listing in defining
20 sending, receiving, and linkage tasks.

In one embodiment, I/O interface 522, I/O interface 519, I/O interface 520, and I/O interface 521 are facilitated by a process control system such as is described in United States Patent 5,555,424 (24Sederlund, et. al.) issued on Sept. 10, 1996 and entitled "Extended Harvard
25 architecture computer memory system with programmable variable address increment" to Sederlund, Edward R.; Lindesmith, Robert J.; Root, Larry A.; Dupree, Wayne P.; and Thomas, Lowell V. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

30 In a more preferred embodiment, I/O interface 522, I/O interface 519, I/O interface 520, and I/O interface 521, are facilitated via a redundant process control computer system in a general process control system using two process control computers (such as the process control computer described in 24Sederlund, et.) as is described in United States Patent 5,583,757 (Baca, Jr., et. al.) issued on Dec. 10, 1996 and entitled "Method of input signal
35 resolution for actively redundant process control computers" to Baca, Jr., Eloy; Dupree,

Wayne P.; Grinwis, Donald J.; Kanse, Johannes C.; Pelletier, Douglas P.; and Schulze, Oscar E. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

5 An embodiment of a system for achieving data access for the process control computer described in 24Sederlund, et al. is described in United States Patent 5,568,615 (15Sederlund, et. al.) issued on Oct. 22, 1996 and entitled "Stealth interface for process control computers" to Sederlund, Edward R.; Thomas, Nadene T.; Lindesmith, Robert J.; and Cowles, Russell W. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

An embodiment of a system providing a remote field unit for use with the process control computer described in 24Sederlund, et al. is described the United States Patent 5,428,769 (69Glaser, et. al.) issued on Jun. 27, 1995 and entitled "Process control interface system
15 having triply redundant remote field units" to Glaser, Robert S.; Hoy, Robert S.; Fernandez, G. Paul; and Grai, Timothy J. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

An embodiment of a system providing an interface for reading electrical current in power
20 distribution systems for use with the field unit described in 69Glaser, et. al. is described in United States Patent 5,151,866 (66Glaser, et. al.) issued on Sept. 29, 1992 and entitled "High speed power analyzer " to Glaser, R. Steven and Bade, Jeffrey M. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

25 An embodiment of a system providing a high speed gateway for use with the redundant process control computer system described in Baca, et al. and the system for achieving data access for the process control computer described in 24Sederlund, et al. is described in

30 (a) United States Patent 5,519,603 (Allbery, Jr., et. al.) issued on May 21, 1996 and entitled "Intelligent process control communication system and method having capability to time align corresponding data sets" to Allbery, Jr., James D.; Troisi, Peter A.; Johnson, Susan J.; Cullen, James H.; Butler, Richard L.; Ferreira, James P.; Ellison, Joseph ; Patel, Chiman L.; Uban, James E.; and Schultz, Dale H.;

(b) United States Patent 5,428,745 (45de Bruijn, et. al.) issued on Jun. 27, 1995 and entitled "Secure communication system for re-establishing time limited communication between first and second computers before communication time period expiration using new random number" to de Bruijn, Ronny P.; Verboven, Marc L. K.; van Weele, Leonardus A.; Vermeire, Roger R.; Schulze, Oscar E.; Schultz, Dale H.; and Bell, Brian G.; and

(c) United States Patent 5,561,770 (70de Bruijn, et. al.) issued on Oct. 1, 1996 and entitled "System and method for determining whether to transmit command to control computer by checking status of enable indicator associated with variable identified in the command" to de Bruijn, Ronny P.; van Weele, Leonardus A.; Verboven, Marc L. K.; Vermeire, Roger R.; Schulze, Oscar E.; Bell, Brian G.; and Schultz, Dale H.

These patents are expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

An embodiment of a system providing human interfacing for use with the redundant process control computer system and interfaces described in the above patents is described in United States Patent 5,631,825 (van Weele, et. al.) issued on May 20, 1997 and entitled "Operator station for manufacturing process control system" to van Weele, Leonardus A.; de Bruijn, Ronny P.; Vermeire, Roger R.; Zemerling, Christo; and Lenting, Ben. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention. While this human interface system could be used as one user interface in each or any one of I/O interface 522, I/O interface 519, I/O interface 520, and I/O interface 521, it should be noted that user interface 503 is preferably designed for interface via network switch 501 to the entire set of systems in a particular instance of switched star topology 530.

Embodiments of systems providing real-time interpretation of application program code executing in the above systems are described (a) in United States Patent 5,491,625 (Pressnall, et. al.) issued on Feb. 13, 1996 and entitled "Information display system for actively redundant computerized process control" to Pressnall, Dana W.; Polishak, Jeffery T.; Felix, Bradley K.; Durisin, Michael J.; and Ellison, Joseph. and (b) in United States Patent 5,408,603 (Van de Lavoie, et. al.) issued on Apr. 18, 1995 and entitled "Global process

control information system and method" to Van de Lavoie, Ronny; Follon, Marinus (Neerpelt, BE); and Ravenscroft, Ian. These patents are expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

5 A large graphical overview system providing interface to humans is deployed in the preferred embodiment along with the Operator Station. This graphical overview system is described in United States Patent 5,726,668 (Clement) issued on Mar. 10, 1998 and entitled "Programmable graphics panel" to John L. Clement. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the
10 present invention.

In a conceptualized most preferred embodiment, I/O interface 510, I/O interface 511, I/O interface 522, I/O interface 519, I/O interface 512, I/O interface 513, I/O interface 520, I/O interface 521, ring breakout 508, ring breakout 509, control computer 504, historian
15 computer 505, ring breakout 518, ring breakout 517, control computer 506, historian computer 507, I/O interface 522, I/O interface 519, I/O interface 520, I/O interface 521, ring breakout 518, ring breakout 517, control computer 506, network support 502, network switch 501, and historian computer 507 are implemented using a dedicated context cycling computer as is described in US patent application number 08/797,967 which was filed on
20 February 12, 1997 by Wayne Dupree, Jeffery Lucas, Larry Root, Gerrit Verniers, and Stephen Churchill entitled "A Dedicated Context-Cycling Computer" with the system of 69Glaser, et. al. facilitating interface to field instrumentation. This application for patent is expressly incorporated herein by reference in the present application for showing the preferred contemplated manner of making and using the present invention.

25 The context cycling processor provides for a substantial number and variety of input and output circuits, with each of the specialized circuits having at least one dedicated register for retaining the process and configuration state of pertinent resources, attributes, or aspects associated with either the circuit or function while another context is active; with the
30 combination of specialized circuit and dedicated register(s) establishing the electrical circuitry base for a context. Contexts also include affiliated logic specific for the circuitry. A plurality of contexts share common assets within the CPU.

As needed, a co-processor for use with the context cycling computer is described in United
35 States Patent 5,655,133 (Dupree, et. al.) issued on Aug. 5, 1997 and entitled "Massively

multiplexed superscalar Harvard architecture computer" to Dupree, Wayne P.; Churchill, Stephen G.; Gallant, Jeffrey R.; Root, Larry A.; Bressette, William J.; Orr, III, Robert A.; Ramaswamy, Srikala; Lucas, Jeffrey A.; and Bleck, James. This patent is expressly incorporated herein by reference in the present application for showing the preferred
5 contemplated manner of making and using the present invention. Network switch 501 also preferably uses a cross-bar (Xbar) switch 516 capable of implementing at least 99 parallel and simultaneous communication linkages.

The master scheduler 106 is implemented in network switch 501 to provide a common clock
10 signal (as an access to either clock 109 or a manifestation of clock 110) and is further responsible for the distribution of the common clock signal; the collection of the requests from each of the member nodes 108, 107 (in this instance of network 530, control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and network switch 501 are all instances of member nodes 107
15 and 108 in the data flow context of Figure 1); the scheduling of the communication tasks for each of the member nodes 108, 107; the transmission of the schedule to each of the member nodes 108, 107; and the implementing of the connections between the member nodes 108, 107.

20 The purpose of distributing a common clock signal is so that each member node 108, 107 on the network 530 has the same understanding of time. This aspect of having the same understanding of time is crucial in this invention since the scheduling of tasks is done based on time. Thus, the common clock signal is used to effectively synchronize and time-align all member nodes 108, 107 to a common time. On a pre-defined periodic basis, the master
25 scheduler 106 in network switch 501 sends a message that is propagated to each of the member nodes 108, 107 to define the beginning of a time period.

The collection of the requests from each member node 108, 107 (any one of control computer 506, historian computer 507, control computer 504, historian computer 505, user
30 interface 503, network support 502, and network switch 501) occurs once each scheduling period. The master scheduler 106 in network switch 501 sets aside the time to receive requests from each member node 108, 107 (any one of control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and network switch 501) each scheduling period. Requests originate from the
35 member nodes 108, 107 which will need the communications requested. The master

scheduler 106 in network switch 501 receives each of these requests and considers them in its scheduling logic. Each request contains information on the source, destination, and length of the communication. In addition, each request may contain a priority indicator as well as information needed to deliver its message to a particular application which is
5 executing on the destination node. After the task of collecting the connection requests from all of the member nodes 108, 107 is completed, the result is collected into a composite request list 111. Note that it is also possible for entries in the composite request list 111 to originate inside of the master scheduler 106 in network switch 501 itself; it is also possible for particular connection requests to have a repetitive nature such that multiple connections
10 in a set of future time periods are invoked from a single connection request.

The master scheduler 106 in network switch 501 then schedules the communications which are to occur in the future. The method which is used in order to perform the scheduling of the communications is not critical in the functioning of the time deterministic network,
15 although the Apparent Critical Resource approach discussed in this specification is the preferred approach. Logic implements to assign times to the requested communications. The determination of when to stop the scheduling process during the scheduling period can occur in one of three ways: first, all of the communication requests which can be scheduled are scheduled and therefore, there is no more scheduling work remaining; second, there is
20 no more time remaining in which to perform the scheduling (the processor must stop the scheduling process in order to continue on with its next task); and third, there have been more connections requested than could be scheduled in the available scheduling window of time increment (an exhaustion of bandwidth), in which case the excess requests are retained as a queue for the next scheduling attempt.

25 The resulting schedule is stored in a master event table 112. This master event table 112 contains the start time 621 of the scheduled communication in addition to some of the same information similar to that in the composite request list 111. Note that it is also possible that certain tasks are pre-loaded into the master event table 112 in order to bypass the
30 scheduling process, such as is required to enforce certain timing requirements of a task.

The master scheduler 106 in network switch 501 then communicates the local event table 114, 116 (that portion of the schedule which pertains to a particular member node 107, 108) to each appropriate member node 107, 108. For each event appearing in the master event
35 table 112 which includes an appropriate member node 108, 107 as either the source or the

destination, an entry exists in the local event table 114, 116. Each member node 108, 107 (any one of control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and network switch 501) which is involved in any communication receives its own local event table 114, 116.

5

The master scheduler 106 in network switch 501 then implements the connections according to the times which are assigned and appear in the master event table 112. The processing of these connections when using the preferred embodiment of the non-blocking switch leads to the situation whereby there may be multiple connections which are implemented

10 concurrently.

The Composite Request List 111 residing in the master scheduler 106 in network switch 501 may contain both continuously repetitive connections to be implemented without contingencies, or dynamically requested connections (implemented when requested, usually initiated by one of the nodes on the network). This Connection Request List contains, at a minimum, a Source specifier, a Destination specifier, and a Duration specifier (see Figure 6 showing record details 600). Additional parameters pertinent to scheduling optionally include such specifiers as Priority 617.

Figure 6 shows the composition of the primary tables and messages 600 used in the scheduling and task coordination process. The connection request list 115 has Source 614, Destination 615, Length 616, and possibly Priority 617 fields 610 in both the member nodes 108, 107 and when combined into the composite request list 111 in the master scheduler 106 in network switch 501. The scheduler logic 611 in master scheduler 106 uses the data in these fields 610 to construct a schedule in the form of a master event table 112 with fields 612 for the Source 618, Destination 619, Length 620 and Start Time 621 for each of the connections. Note that the scheduler logic 611 takes the Source 614, Destination 615, and Length 616 from the connection request list 111, while the Start Time 621 is a product of the scheduling process itself.

30

The local event tables 114, 116 are simply subsets of the master event table 112, broken into those portions applicable to the respective member node 108, 107 (any one of control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and network switch 501) - no manipulation of the fields 613 is required. Virtual paths 601, 602, 603, 606, 607, and 608 show the flow of the

35

information from the connection request list fields 610 to the local event table fields 613. Virtual path 604 shows the input of priority 617 to the scheduler; and virtual path 609 shows the start time 621, 625 as communicated from scheduler logic 611.

- 5 This data is then used by the scheduler logic 611 to derive a schedule for servicing the requested connections. The outcome of the scheduling process of the scheduler logic 611 is recorded in the Master Event Table 112, which is in turn parsed up into respective Local Event Tables 116 for distribution to the network nodes (see Figure 1). The Event Tables contain Source 622, Destination 623, Duration 624, and Start Time 625 fields 613.

10

Within the network nodes, the Local Event Tables 114, 116 are merged together with the Local Task Lists and logic executes to ensure that all the necessary work can be completed and that all timing corresponds to (or doesn't conflict with) that specified by the scheduler logic 611.

15

- Figure 7 summarizes the general scheduling process 710. Connection requests are collected during a defined request acceptance period. After a defined period of time, the collection of connection requests must be processed. Additional connection requests are not prohibited, but a cutoff point must be established. Figure 7 shows the basic steps involved in the master scheduler 106 in network switch 501 node which are required in scheduling the connection requests. The first three steps 701-703 provide for aligned time definition in all nodes. In step 704, connection requests are received from all member nodes 107, 108 (the nodes sending their requests as a communication at predefined moments in their local task schedules from prior scheduling operations). In step 705, the connection requesting process ends so that a schedule can be defined for the future as the scheduling process applies to those requests which were collected from the member nodes 108, 107 during a particular window of time increment which is about to conclude. Schedule connections requests step 706 can begin once the requests have been collected (more detail in Figure 8). Once the schedule is complete, the applicable portions of the schedule are extracted from the master event table 112 in the creation of the local event tables 114, 116, in step 707 which are then conveyed to the nodes 107, 108 in step 708 for later implementation in step 709.
- 20
- 25
- 30

- For distributed computing environments in general, connections between nodes 107, 108 may be required at any time, meaning that connection requests may originate any time, and some service time must be tolerable in handling the connections. There are four processes,
- 35

then, that repetitively execute at a rate conformant to the desired response in the network 530: collecting connection requests, scheduling connection requests, distributing the resulting schedule to the network nodes, and communicating through the network 530 via the established requested connections (see Table 1).

5

Table 1 - Four Scheduling Steps

Request Connections
Schedule Connections
Convey Schedule
Communicate

10

Table 2 shows the timing relationship between these four processes which, as stated, are always in progress.

Table 2 - The Scheduling Pipeline

15

Request Connections			
Request Connections	Schedule Connections		
Request Connections	Schedule Connections	Convey Schedule A	
Request Connections	Schedule Connections	Convey Schedule B	Communicate A
Request Connections	Schedule Connections	Convey Schedule C	Communicate B
Request Connections	Schedule Connections	Convey Schedule D	Communicate C
Request Connections	Schedule Connections	Convey Schedule E	Communicate D
Request Connections	Schedule Connections	Convey Schedule F	Communicate E

20

Note in the rightmost column of Table 2 that communications are always taking place after an initial few iterations establish a basis for communications. For any particular communications in a window of time increment, the schedule for the communications being implemented was conveyed in a previous window of time increment, the requests were scheduled in a window of time increment prior to the conveyance of the schedule, and the requests were requested even prior to scheduling. Timing variations can exist in this pipeline, but this resembles the pipeline used in the preferred embodiment.

30 The logic used in the currently preferred embodiment has been designed with two objectives in mind: an efficient schedule must be the outcome, and the number of cycles required to generate that schedule must be bounded at a reasonable level. Efficient scheduling involves a simple process based on the premise that the efficiency of the resultant schedule is primarily a function of the efficiency to which the busiest resources have been utilized in the schedule. This amounts to making sure that, at all times in the scheduling process,

35

scheduling precedence is given to the resources with the highest remaining demands. In order to derive a schedule in an acceptable amount of time, the logic was designed to schedule each request once and only once (trial and error logics deemed unacceptable), and with a moderate handling requirement for each connection request.

5

The basic process in scheduling 806, as shown in Figure 8, is to

- 1) define a criteria for measuring resource demand;
- 2) sort the resources according to criticality;
- 10 3) service those connections which can be serviced in this step and in the order of resource criticality;
- 4) step forward to the completion of the previously serviced requests thereby freeing up the applicable resources;
- 5) repeat steps 2-4 until all requests have been serviced or until there's no
- 15 opportunity to service more requests.

Any requests which weren't able to be serviced in this scheduling pass can then be resubmitted during the next scheduling pass.

- 20 The flowchart of Figure 8 outlines the process used in the Apparent Critical Resource scheduling logic. The first step is to define the criteria by which resource usage is to be measured 800. For the preferred embodiment, transmit and receive channels are the resources which are scheduled, and the usage duration for each of these resources is the criteria chosen.

25

- Once the resources are evaluated for usage, they are sorted by usage 801. This allows the resources which are used most heavily to be scheduled in step 802 first. When all resources which can be scheduled have been scheduled (the existing requested connections may not require or permit the simultaneous use of all resources), the scheduling window of time
- 30 increment is advanced in step 803 until applicable resources become available. If there are more connection requests to be serviced and more time in the schedule to service these requests, this process is repeated after step 804 resolves the question respective to the need for another iteration through steps 801 to 803). The sub process 806 of the scheduler logic proceeds to step 805 to exit upon resolution by step 804 that the scheduling instance
- 35 has been completed.

Figure 9 presents a more detailed overview of the Apparent Critical Resource (ACR) preferred embodiment sub-process 950 of the scheduling logic. This diagram encompasses the steps mentioned in Figure 8 and some additional actions required for conditional operation and exception handling as the preferred implementation of the process shown in Figure 8. Step 901 represents the initiation point for the preferred ACR process. In step 902, the ACR table is constructed, and a determination is made in step 903 respecting the presence of unscheduled requests based on the values of the SAV in the ACR table (a rapid exit to step 918 is executed if the values indicates that no requests are in the ACR table). If requests exist, the ACR sub-process 950 proceeds to calculate window size in step 904, look for a specific request in step 905, and check on the existence of the specific request and parties for the communication in step 906. If the request and parties exist, the process proceeds to add the event to the event table in step 912 and modify the ACR path in step 913 before proceeding to the next request in step 914. If the high value is not zero in step 915, the process proceeds again to step 904. If the high value is zero in step 915, the definition of (future) time in the ACR is advanced in step 916, the bandwidth is checked in step 917 for sufficiency in time to enable further scheduling in this scheduling operation, and the process returns to step 914 for the next request. If in step 906, a receiver for the communication did not exist, an alternative receiver is defined in step 907 and evaluated in step 908. If step 908 does not define an alternative receiver, an alternative transmitter is selected in step 919 and evaluated in step 920; if acceptable (step 920), step 921 selects the receiver with the highest ACR value and returns the process to step 908; and, if not acceptable (step 920), the process advances to step 916. When an alternative is defined in step 908, the process proceeds to schedule via steps 909, 910 and 911 to step 912. The virtual datalogical process being executed by these steps can be best described by considering an example respective to the list of communication requests for a hypothetical network consisting of five nodes shown in Table 3:

30

Table 3: Initial Request List

Source	Destination	Length	Priority
1	2	15	1
1	4	23	1
1	5	2	1
2	1	24	1
3	1	16	1
3	1	12	1
3	2	20	1
3	3	8	1
3	4	19	1
3	4	16	1
3	5	4	1
3	5	9	1
4	1	13	1
4	5	9	1
5	1	21	1
5	1	8	1
5	2	8	1
5	3	1	1
5	4	18	1
Total		246	

- 5 Assume that the length is measured in some arbitrary units of time, perhaps in milliseconds, and that there are n different priority levels, 1 through n , where 1 is the most urgent and n is the least.

10 First, note that with a broadcast type network (where all transactions are required to share the bandwidth, such as a Carrier Sense Multiple Access with Collision Detection or CSMA/CD network), service time required to complete the above requests would require a theoretical minimum of 246 units of time.

The incorporation of a non-blocking switch permits the simultaneous operation of the 5 transmit resources and the 5 receive resources. In analyzing the benefit of implementing a non-blocking network, the scheduler logic of the example first assesses the demands on each of the resources by constructing a table such as Table 4:

Table 4: Initial Summed Access Values

X / R		1	2	3	4	5		SAV
1		-	15	-	23	2		40
2		24	-	-	-	-		24
3		28	20	8	35	13		104
4		13	-	-	-	9		22
5		29	8	1	18	-		56
SAV		94	43	9	76	24		246

The summed access values, or SAVs, are simply the row or column totals for the transmitter (X-in the rows of the table) and receiver (R-in the columns of the table) interaction times and reflect the total demands on the particular resource, as measured in the arbitrary units of length. Note now that the theoretical minimum time required to service the requests has dropped from 246 units of time to 104 by selecting a non-blocking switch solution. The minimum service time for these requests is probably constrained by transmitter 3 since transmitter 3 has the highest SAV and the other transmitters can typically be serviced while transmitter 3 is being serviced. While the theoretical lower limit may be 104 units of time, the actual minimum service time may be greater and may even be dependent upon the order in which the connections are serviced. The next objective, however, is determining the optimum scheduling method in order to maximize the capabilities of the existing network (minimizing the required service time).

Assume for this particular example that the resultant schedule, regardless of the number or composition of the requests, applies to a future window of time increment of time with a duration of 150 units of time. The example begin by looking at the connections to be made at

Time = 0 for that 150 unit schedule. The primary objective, judging by the SAVs, is to schedule some request involving Transmitter 3 and Receiver 1. Referring back to the initial Request List, the scheduler logic of the example finds a request involving Transmitter 3 and Receiver 1 which has a length of 16 units of time. Since the scheduler logic of the example is currently scheduling connections at Time = 0 and the scheduler logic of the example has 150 units of time, the scheduler logic of the example concludes that the 16 word request fits into the scheduling window of time increment , and records the transaction into the Event Table of Table 5.

Table 5: Intermediate Event Table

[illegible]

Pictorially, Table 5 may be shown as schedule chart 1001 in Figure 10. Transmitter scheduled element 1002 and receiver scheduled element 1003 denote the scheduled tasks.

- 5 The Request List must then somehow reflect the fact that this last request has been serviced. This modified Request List may now resemble that shown in Table 6.

Table 6: Intermediate Request List

Source	Destination	Length	Priority	Start Time
1	2	15	1	-
1	4	23	1	-
1	5	2	1	-
2	1	24	1	-
3	1	16	1	0
3	1	12	1	-
3	2	20	1	-
3	3	8	1	-
3	4	19	1	-
3	4	16	1	-
3	5	4	1	-
3	5	9	1	-
4	1	13	1	-
4	5	9	1	-
5	1	21	1	-
5	1	8	1	-
5	2	8	1	-
5	3	1	1	-
5	4	18	1	-
Total		246		

10

Since Transmitter 3 and Receiver 1 are now busy for Time = 0, their status must indicate this, so the updated SAV Table, following some reformatting, may now resemble Table 7:

Table 7: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	40	78
2	24	43
3	88	9
4	22	76
5	56	24

- 5 Note that the SAVs for Transmitter 3 and Receiver 1 have been adjusted to reflect the serviced request. This is because the scheduler logic continues to give scheduling precedence to the resources with the highest impact on the schedule. This happens at the expense of the other resource needs, and at some point the inefficiencies injected into the resources with low SAVs may cause those resources to have a higher impact on the
- 10 remaining portion of the schedule than those resources which had the highest impact at the start. For this reason, the protocol is referred to as Apparent Critical Resource scheduling (scheduling precedence is always given to resources which appear to be on the critical path at the time of consideration), or ACR scheduling.
- 15 It is now evident that the scheduler logic of the example should try to schedule a request from Transmitter 5 to Receiver 4, the next two concurrent resources in regards to scheduling impact which remain available for scheduling. There is such a request, so the scheduler logic of the example executes so that the tables now resemble Tables 8, 9, and 10.

20

25

Table 8: Intermediate Request List

Source	Destination	Length	Priority	Start Time
1	2	15	1	-
1	4	23	1	-
1	5	2	1	-
2	1	24	1	-
3	1	16	1	0
3	1	12	1	-
3	2	20	1	-
3	3	8	1	-
3	4	19	1	-
3	4	16	1	-
3	5	4	1	-
3	5	9	1	-
4	1	13	1	-
4	5	9	1	-
5	1	21	1	-
5	1	8	1	-
5	2	8	1	-
5	3	1	1	-
5	4	18	1	0
Total		246		

5

10

30

Table 9: Intermediate Event Table

[illegible]

Table 10: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	40	78
2	24	43
3	88	9
4	22	58
5	38	24

- 5 Note that the Event Table is effectively a (chronologically) sorted list of the scheduled requests from the Request List. For this reason, intermediate stages are not shown for each scheduling step.

Repeating the process as before, the scheduler logic of the example schedules a 1-to-2
10 communication instance request (that is a connection from Transmitter 1 to Receiver 2), and the tables are modified to Tables 11 and 12 as follows:

15

20

25

Table 11: Intermediate Request List

Source	Destination	Length	Priority	Start Time
1	2	15	1	0
1	4	23	1	-
1	5	2	1	-
2	1	24	1	-
3	1	16	1	0
3	1	12	1	-
3	2	20	1	-
3	3	8	1	-
3	4	19	1	-
3	4	16	1	-
3	5	4	1	-
3	5	9	1	-
4	1	13	1	-
4	5	9	1	-
5	1	21	1	-
5	1	8	1	-
5	2	8	1	-
5	3	1	1	-
5	4	18	1	0
Total		246		

5

10

33

Table 12: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	25	78
2	24	28
3	88	9
4	22	58
5	38	24

The scheduler logic of the example next looks for a 2-to-5 communication instance request,
 5 but such a request was not made. Several alternatives exist in determining the next
 combination to be attempted, and the optimum solution is chosen based on the desired
 tradeoffs between bandwidth efficiency and processing costs. (With minimal bandwidth
 impact, a simple rule set can be chosen, and for the preferred embodiment, an alternative
 receiver is selected for the existing transmitter until all receiver choices have been
 10 exhausted, after which that particular transmitter is ignored and the next transmitter is
 selected, and the highest impact receiver is re-considered - see Figure 9.)

Using this logic, a 2 to 3 communication instance request is searched for but not found, then
 a 4 to 5 communication instance request is searched for, found, and selected. The SAV
 15 Table is updated, and it is determined that no other connections can be made at Time = 0.
 The Event Table now is shown in Table 13:

20

25

Table 13: Intermediate Event Table

Source	Destination	Length	Priority	Start Time
3	1	16	1	0
5	4	18	1	0
1	2	15	1	0
4	5	9	1	0
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

- Again, these four connections can be represented pictorially in schedule chart 1101 in Figure 11. Transmitter scheduled element 1002 and receiver scheduled element 1003 from Figure 10 have been denoted as scheduled elements 1102 and 1103, respectively, and have been joined by scheduled elements 1104 and 1105 (the 5-to-4 communication), 1106 and 1107 (the 1-to-2 communication), and 1108 and 1109 (the 4-to-5 communication).
- The logic involved in Advancing Time amounts to advancing to the next chronological finish. In this case, the 4 to 5 communication instance connection is the next chronological finish,

so the example advances time to Time = 9, at which time Transmitter 4 and Receiver 5 are again enabled. The Request List and SAV Table are now as shown in Tables 14 and 15.

Table 14: Intermediate Request List

5

Source	Destination	Length	Priority	Start Time
1	2	15	1	0
1	4	23	1	-
1	5	2	1	-
2	1	24	1	-
3	1	16	1	0
3	1	12	1	-
3	2	20	1	-
3	3	8	1	-
3	4	19	1	-
3	4	16	1	-
3	5	4	1	-
3	5	9	1	-
4	1	13	1	-
4	5	9	1	0
5	1	21	1	-
5	1	8	1	-
5	2	8	1	-
5	3	1	1	-
5	4	18	1	0
Total		246		

10

Table 15: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	25	78
2	24	28
3	88	9
4	13	58
5	38	15

- 5 After searches for a 2-to-5 communication instance request, a 2-to-3 communication instance request, a 4-to-5 communication instance request and a 4-to-3 communication instance request, the scheduler logic of the example concludes that the example must again advance time - nothing could actually be scheduled at Time = 9. Now, the next finish is at Time = 15, at which time the 1-to-2 communication instance connection finishes and the
- 10 SAV Table is now Table 16.

Table 16: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	25	78
2	24	28
3	88	9
4	13	58
5	38	15

- 15 The scheduler logic of the example now looks for a 1-to-2 communication instance request, but the only 1-to-2 communication instance has already been scheduled, so the example look for a 1-to-5 communication instance request, which the example finds and schedules. The new SAV Table is then Table 17.

Table 17: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	23	78
2	24	28
3	88	9
4	13	58
5	38	13

The scheduler logic of the example then looks for a 2-to-2 communication instance request, a 2-to-3 communication instance request, a 4-to-2 communication instance request and a 4-to-3 communication instance request before determining that it's time again to advance time. The example advances time from 15 to 16, at which time the SAV Table looks like Table 18.

Table 18: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	23	78
2	24	28
3	88	9
4	13	58
5	38	13

The scheduler logic of the example then schedules a 3-to-1 communication instance request, which turns out to be the only request which can be scheduled at Time = 16. The tables, after the scheduler logic of the example advances time to Time = 17, look like Tables 19, 20, and 21.

Table 19: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	23	66
2	24	28
3	76	9
4	13	58
5	38	13

5

10

15

20

25

Table 20: Intermediate Request List

Source	Destination	Length	Priority	Start Time
1	2	15	1	0
1	4	23	1	-
1	5	2	1	15
2	1	24	1	-
3	1	16	1	0
3	1	12	1	16
3	2	20	1	-
3	3	8	1	-
3	4	19	1	-
3	4	16	1	-
3	5	4	1	-
3	5	9	1	-
4	1	13	1	-
4	5	9	1	0
5	1	21	1	-
5	1	8	1	-
5	2	8	1	-
5	3	1	1	-
5	4	18	1	0
Total		246		

Nothing can be scheduled at Time = 17, so the example advance to Time = 18 where the 5-
5 to-4 communication instance request finishes up.

Table 21: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	23	66
2	24	28
3	76	9
4	13	58
5	38	13

- 5 After unsuccessfully searching for a 5-to-4 communication instance request, a 5-to-2 communication instance request is found, a 1-to-4 communication instance request is found, and the tables now look like Tables 22 and 23:

Table 22: Intermediate Summed Access Value Table

10

Channel	Transmitter SAV	Receiver SAV
1	-	66
2	24	20
3	76	9
4	13	35
5	30	13

15

Table 23: Intermediate Request List

Source	Destination	Length	Priority	Start Time
1	2	15	1	0
1	4	23	1	18
1	5	2	1	15
2	1	24	1	-
3	1	16	1	0
3	1	12	1	16
3	2	20	1	-
3	3	8	1	-
3	4	19	1	-
3	4	16	1	-
3	5	4	1	-
3	5	9	1	-
4	1	13	1	-
4	5	9	1	0
5	1	21	1	-
5	1	8	1	-
5	2	8	1	18
5	3	1	1	-
5	4	18	1	0
Total		246		

The scheduler logic of the example now advances time to Time = 26 for which the scheduler
 5 logic of the example defines Table 24:

Table 24: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	66
2	24	20
3	76	9
4	13	35
5	30	13

The scheduler logic of the example now schedules a 5-to-3 communication instance, which
 5 is all that can be scheduled at this time. Time is then advanced to 27, then to 28, and the SAV Table looks like Table 25.

Table 25: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	66
2	24	20
3	76	8
4	13	35
5	29	13

10

Since there are no 3-to-1 communication instance requests, the scheduler logic of the
 example schedules a 3-to-2 communication instance request and a 5-to-1 communication
 instance request, and advances time to 31 and then to 48, at which time the SAV Table
 becomes Table 26.

15

Table 26: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	45
2	24	-
3	37	8
4	13	16
5	8	13

The scheduler logic of the example then schedules the 3-to-4 communication instance request and advances time to 49, giving Table 27 as a result.

Table 27: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	45
2	24	-
3	37	8
4	13	16
5	8	13

The scheduler logic of the example then books a 2-to-1 communication instance request and advances time to 67, giving Table 28.

Table 28: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	21
2	-	-
3	29	8
4	13	16
5	8	13

The scheduler logic of the example now books the 3-to-3 communication instance request
 5 and advances time to 73 as shown in Table 29.

Table 29: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	21
2	-	-
3	29	-
4	13	16
5	8	13

10 The scheduler logic of the example then books the 4-to-1 communication instance request,
 and advances time to 75, resulting in Table 30.

Table 30: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	8
2	-	-
3	29	-
4	-	16
5	8	13

This allows the remaining 3-to-4 communication instance request to be scheduled by the scheduler logic of the example, after which the scheduler logic of the example advances to Time = 86, resulting in Table 31 data.

Table 31: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	8
2	-	-
3	13	-
4	-	-
5	8	13

10

The scheduler logic of the example now schedules the remaining 5-to-1 communication instance request, advances time to 91, schedules the first 3-to-5 communication instance request, advances time to 94, then to 95, and schedules the final request, another 3-to-5 communication instance request. The resultant Event Table then looks like Table 32.

15

Table 32: Final Event Table

Source	Destination	Length	Priority	Start Time
3	1	16	1	0
5	4	18	1	0
1	2	15	1	0
4	5	9	1	0
1	5	2	1	15
3	1	12	1	16
5	2	8	1	18
1	4	23	1	18
5	3	1	1	26
3	2	20	1	28
5	1	21	1	28
3	4	19	1	48
2	1	24	1	49
3	3	8	1	67
4	1	13	1	73
3	4	16	1	75
5	1	8	1	86
3	5	4	1	91
3	5	9	1	95

Note that the entire list of request completes in 104 units of time - the theoretical minimum
 5 amount of time required to service these particular requests.

Pictorially, the final schedule is shown in as schedule chart 1201 in Figure 12. Transmitter
 scheduled element 1002 and receiver scheduled element 1003 from Figure 10 have been
 denoted as scheduled elements 1202 and 1203; scheduled elements 1104 and 1105, 1106
 10 and 1107, and 1108 and 1109 from Figure 11 are denoted as scheduled elements 1204,
 1205, 1206, 1207, 1208, and 1209; and new elements 1216-1217 for the 1-to-4
 communication instance, 1214-1215 for the 5-to-2 communication instance, 1212-1213 for
 the 3-to-1 communication instance, 1220-1221 for the 3-to-2 communication instance, 1222-
 1223 for the 5-to-1 communication instance, 1226-1227 for the 2-to-1 communication

instance, 1224-1225 for the 3-to-4 communication instance, 1228-1229 for the 3-to-3 communication instance, 1230-1231 for the 4-to-1 communication instance, 1232-1233 for the 3-to-4 communication instance, 1234-1235 for the 5-to-1 communication instance, and 1238-1239 for the 3-to-5 communication instance are shown. Three of the interactions are
5 shown without drawing element identifiers (the 1-to-5 communication having a duration of 2, the 5-to-3 communication having a duration of 1, and the 3-to-5 communication having a duration of 4.

Note that transmitter 3, the critical resource, is booked solid until Time = 104, a criteria which
10 must be met in order to reach the theoretical minimum scheduling time. Note also that receiver 1, the second most critical resource, was booked solid until Time = 94. While this logic doesn't always produce the theoretically optimum schedule, it is designed to be efficient with the resources having the highest demands.

15 If the scheduling process had applied to a 100 unit window of time increment of time rather than a 150 unit window of time increment, the very last request would have been precluded from being scheduled in this window of time increment, but would have been queued until the scheduling process was resumed for the next time window of time increment.

20 The above example showed a list of requests where all of the requests had priority levels of 1. In one embodiment, four different priority levels are supported. All of the highest level requests are scheduled according to the logic before any of the other priority requests are even considered for scheduling.

25 Once all of the guaranteed requests have been scheduled, connection requests from the other three priorities are scheduled around the guaranteed requests already in the schedule. When there are already connections in the schedule, the process of calculating the window of time increment size is a little more complex, but the logic is essentially the same. Though the ACR value (or SAV) is still the predominant factor in deciding what gets scheduled when,
30 connection requests between the same two concurrent resources are selected based on priority. This means, for example, that the ACR logic may elect to schedule a low priority 4-to-5 communication instance request before it schedules a medium priority 4-to-2 communication instance request, but never before it schedules a medium priority 4-to-5 communication instance request.

Since there are requirements for certain connections to happen at a particular time rather than simply in some window of time increment of time, accommodations can be made for having these requests bypass scheduling. These fixed time connections are entered into the Event Table before scheduling begins and the scheduled connections must be worked in around the fixed connections. These fixed time connections obviously have the highest priority of any of the requests.

In the above example, advancing time was simply a matter of finding the next chronological finish of an already scheduled connection. However, the process of advancing time consumes cycles which might be better spent performing other functions. The network scheduler in the preferred embodiment uses a minimum time increment of 256 units (with a scheduling window of time increment of over 25,000 units rather than the 150 unit window of time increment described above) to reduce the processing requirements. While this does have an obvious negative impact on scheduling efficiencies, there is also a less obvious positive impact. On the average, more resources are candidates for scheduling at each time increment as a result of having processed more finishes between increments. With more resources contending simultaneously, those resources with higher critical impact values are less likely to be preempted by some other resource which had simply finished earlier. The end result of this compromise is that compute cycles are saved, but bandwidth efficiencies aren't substantially impacted.

In the above example, all requests were scheduled in their entirety - there was no need to break up the requests. Had any one of the requests been of a length longer than the defined window of time increment (150 units of time in the above example), however, it would have been segmented into at least two portions to enable scheduling. If a long message cannot be avoided by design, by breaking up the message and using a low priority attribute in this situation, long lower priority messages are less likely to preclude higher priority messages contending for the same resources.

In the preferred embodiment, requests greater than 1023 units are broken into 1000 unit "full" blocks and an optional "partial" block up to 1023 units in length.

In the preferred embodiment, the Itineraries (the particular messages used to convey the schedule) are themselves scheduled messages due to their varying, and potentially large,

size. The Agendas (the messages used to collect the connection requests), however, are small and handled as fixed time connections (they are not literally scheduled). The scheduling pipeline used in the preferred embodiment involves collecting the agendas and scheduling the Itineraries in the first window of time increment, where the second, third, and fourth window of time increments are used as described (for scheduling the messages, conveying the schedule, and implementing the connections, respectively).

Turning now to Figure 13, a more detailed representation of components respective to a member node 107, 108 is depicted. While Figure 13 reprises a number of elements of Figures 1 and 5, details of logic types are shown enabling concurrent transmission and receipt of messages respective to the other nodes. In this regard, Transmit Logic 1301 and Receive Logic 1302 are shown for the Master Scheduler 106, with companion Transmit Logic 1319 and Receive Logic 1320 on member node 107 and further companion Transmit Logic 1321 and Receive Logic 1322 on member node 108. Pathways 1311, 1305, and 1306 show vectored message paths outbound from message carrier 100 (with crossbar switch 516). Such messaging approaches are achievable with a context cycling computer such as has previously been discussed, enabling a particular node 107, 108, 106 to be engaged in highly interleaved transmission and receipt of messages. Further detail showing the local scheduler 1307, oscillator 1308, CPU 1309, and VCXO (Voltage Controlled Crystal Oscillator) 1309 is shown for node 107 along with the previously discussed local task list 1303 and local task schedule 1304 which have also been discussed in general.

Turning now to Figure 14, focus is directed to three technologies which, when combined, illustrate a unique and valuable abstracted characterization of considerations in networking distributed, real-time computers to further put the nature of the present invention into full perspective. In this regard key technologies relate to Tight Synchronization between computers; Time Deterministic operation, including network communications; and Centralized Scheduling of distributed tasks, particularly those involving network communications. This diagram shows the interrelationships 1400 between the various technologies implemented in the preferred embodiment, with emphasis on those technologies considered critical to the implementation. The fundamental technologies which, when combined, provide an architecture amenable to real time computing in a distributed computing environment are (1) Time Deterministic Event Control 1422; (2) Tight Synchronization 1423 between systems; and (3) Centralized Scheduling 1421 of the tasks critical to the operation of the real time systems in the distributed environment.

The relationships between the technologies are shown using arrows, where the arrowheads point from the enabling technology to the enabled technology, and where the term "enabling" is loosely used here to describe an improvement in the utility of that technology, or in the simplicity of applying the technology. Note that many of the technologies are mutually beneficial, as depicted with a double headed arrow.

The following table explains the relationships between the implementation characteristics as virtual linkages of Figure 14 between the features and goals of tight synchronization 1423, prioritization 1424, time deterministic event control 1422, security 1426, limit checking 1425, centralized scheduling 1421, physical circuit switching 1434, non-blocking tree topology 1429, reliability 1428, active redundancy 1427, and low cost 1435 are interrelated in an abstracted characterization.

Table 33 Interactions of key technical features

Connection In drawing	Quality
1401	Centralized scheduling assures that all network interactions occur at prescribed times. The first requirement for Time Determinism, by this description, is having the nodes on the network in tight synchronization. None of the network nodes can know the needs of the entire network, so in the absence of the central scheduler, the alternatives are to either remove all dynamics or to implement a solution tolerant of the variability. By incorporating centralized connection scheduling, prioritization, and limit checking, the (high priority) time critical interconnections can be time deterministic while the (low priority) high volume interconnections can exist through variability tolerant designs.
1402	In a distributed computing environment, deterministic networking can be achieved either with tightly synchronized systems, or by providing enough excess bandwidth to realize timely delivery in a worst case timing and bandwidth demand situation. Because such an analysis of worst case situation is difficult and because providing the excess bandwidth can be costly, synchronization is an enabling technology for Time

	<p>Determinism. Additionally, the difficulty in achieving a desired level of synchronization is usually proportional to the extent that the communications are indeterminate.</p>
1403	<p>Active Redundancy enables Time Determinism primarily in regards to failure recovery. It is difficult to manage the possible failure scenarios in the absence of redundancy, yet it is also difficult to manage failure recovery in a Time Deterministic system with passive redundancy. Time Determinism facilitates the use of Active Redundancy in that one of the biggest challenges to active redundancy is in assuring that all sides operate identically. Furthermore, with a Time Deterministic environment, a redundant element of a system can be diagnosed as dysfunctional simply by noting that the appropriate actions aren't occurring at the prescribed times.</p> <p>The time deterministic network involves communicating based on time between separate computing elements. The fact that the communications are scheduled based on time facilitates the employment of active redundancy. In active redundancy, there are two or more computers which are each executing identical logics via its own sequence of program instructions, exchanging the results of this execution, and then arbitrating to come to a result. Due to the fact that the data received by each computer must be shared with the other computer(s) executing the same logic prior to arbitrating this data, there must be some mechanism by which the time to begin the arbitration process is determined. The mechanism which has been chosen is to provide time synchronization between the redundant computers and schedule the tasks related to the communications based on time. In this way, there is known the time at which the arbitration task can begin.</p>
1404	<p>One of the most difficult challenges in implementing active redundancy is in handling the variation between sides. All sides of a redundant system must be expected to match at defined checkpoints, and discrepancies must be resolved in order to retain the value in redundancy. All sides must know when a checkpoint has been reached, and rules must be defined to allow differences to be resolved. With centralized scheduling,</p>

	<p>variations in connection requests, either in content or in timing, can be reduced or even eliminated, thereby increasing the probability that all sides can maintain operational equivalence.</p> <p>On the other hand, centralized scheduling places a tremendous dependency on the availability of that scheduling element. Centralized scheduling becomes much more viable in the presence of redundancy.</p>
1405	Centralized scheduling requires that all connection requests be assembled together for analysis. This permits limit checking to verify that (1) certain undesirable node-to-node combinations are prohibited and (2) traffic levels (possibly at a particular priority) between certain combinations of nodes doesn't exceed predefined levels.
1406	Limit checking enhances the security of the distributed computing system by assuring that certain restrictions aren't violated. These limit checks help against malicious attacks where certain systems shouldn't be allowed to interact with others, and against accidental complications such as overloading a system with legitimate low priority data or unintentional overloading of high priority data.
1407	Prioritization of the messages enhances the utility of the limit checking. Not only can the highest priority data be made time deterministic without excessively restricting the lower priority messages, but security measures can be put in place which take message priority into account.
1408	Limit checking improves the overall reliability of the system by preventing certain security risks and by retaining the time determinism. In the absence of limit checking, certain network faults could cripple the network, such as when a node were to accidentally request an excessive amount of traffic at the highest priority level.
1409	In the implementation of Limit Checking, the highest priority traffic is testing against throughput thresholds to assure that no errors on one channel preclude the highest priority messaging to take place between any nodes not including the node with errors.
1410	Non-blocking network architectures allow the network to realize throughputs higher than the throughput available to any particular node on the network. This translates to higher bandwidth at any particular

	price when compared to blocking architectures.
1411	Security can often be implemented more easily when a centralized security agent can be entrusted with the administration of the security policies. Specifically in the preferred embodiment, the central scheduler is given the security tables and the interconnecting systems can delegate the security task to the central authority, allowing the administration to be done in one place rather than in two. Furthermore, the same security policies don't have to be re-implemented in as many environments as can exist on the star.
1412	By introducing a centralized scheduler, the scheduling logic can be implemented so as to enforce prioritization on the received connection requests.
1413	Prioritization simplifies the task of implementing Time Determinism. This is because large but infrequent tasks such as program loading can be set to operate at a low priority, eliminating the analysis of this high impact operation would have on the simpler yet more critical task of peer to peer communications.
1415	On networks employing a broadcast protocol, all network nodes have the opportunity to view the data going to all other nodes (though encryption can make this viewing rather futile). Blocking protocols aren't always implemented in broadcast protocols, but non-blocking protocols preclude the possibility of broadcasting, thereby avoiding the obvious security risks.
1416	One of the obvious ways of achieving reliability is in the introduction of redundancy.
1417	High throughput, the primary objective of high bandwidth, is positively impacted in two ways through the introduction of prioritization. First, there is the perception of high throughput in that the data that has the most value is delivered in a more timely fashion by invoking prioritization. Secondly, higher volumes of data can be permitted knowing that the high priority data, which often must be delivered in a timely fashion, won't be sacrificed by permitting large volumes of low priority traffic to be requested.
1418	Active Redundancy requires that a failed element be capable of

	recovering from downtime. This often involves the exchange of historical data from the elements which did not fail. In order to minimize the impact of this exchange, prioritization can be introduced assuring that the recovering element can recover and that it can do so without a substantial detrimental impact on the healthy elements.
1419	Certain architectures and topologies are inherently at risk of network outages in the presence of system outages. Use of a non-blocking architecture and physical circuit switching, while not the only means to achieve a higher reliability, offers a level of isolation from network node faults.
1420	Centralized scheduling is an enabling technology for effectively using the bandwidth available to non-blocking networking architectures such as those implementing certain crossbar switches. In the absence of scheduling, fixed connection patterns must be implemented, thereby reducing the practical throughput available to the nodes.
1431	Physical Circuit Switching technology can be difficult to implement for several reasons, including challenges associated with determining what circuits should be put in place at any particular time. By knowing the needs ahead of time, the connection patterns for the circuit switch can be determined a priori and implemented as needed.
1432	Circuit switching of systems configured in a tree topology can be implemented with a separate circuit, or pair of circuits, for each system on the network. This allows each channel to have its own path for communications, permitting a non-blocking architecture.
1433	Because physical circuit switching can reduce the requirement for active circuitry, high line speeds can often lead to higher network bandwidths.
1436	The simplicity of the circuitry associated with physical circuit switching often permits a given set of capabilities to be implemented at a lower cost.

Turning now to Figure 15, the general process 1500 executed respective to scheduling in each member node 108, 107 (any one of control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and
5 network switch 501) is discussed. Each member node 108, 107 is responsible for receiving

the common clock signal; synchronizing to that common clock signal; receiving the local event table 114, 116 from the master scheduler 106 in network switch 501; formulating the task execution list based on the received local event table 114, 116 and merging in any additional tasks which are not described in the local event table 114, 116; executing the task execution list; generating the request list of communications; and transmitting the request list of communications to the master scheduler 106 in network switch 501.

Steps 1501 and 1502 - Since the network is so highly dependent on a global time base, the first step 1501 at a member node 108, 107 is to receive the common clock signal. The clock signal is communicated as a message from the master scheduler 106 in network switch 501 to each of the member nodes 108, 107. The member node 108, 107 receives this clock signal in step 1501, stamps the actual time of reception, and validates the integrity of this message. The validated message is used as an input to the process of synchronization and, in step 1502, the difference between the expected and actual times of time message receipt is optionally determined (optionally if the nature of the time message is a pulse instead of a data bearing message).

Steps 1503 and 1504 - Each member node 108, 107 must now synchronize to the common clock signal. The process of performing this synchronization can be broken up into two separate tasks: first, in step 1503, the adjustment of a voltage controlled oscillator (1308) to a frequency which matches the source of the common clock signal and second, in step 1504, the alignment of the time such that the start of each time period which is defined by the common clock signal occurs at the same time in the member nodes 108, 107 as it does in the master scheduler 106 in network switch 501.

Step 1505 - Once the synchronization of a member node 108, 107 to the master scheduler 106 in network switch 501 is complete, the member node 108, 107 must receive the local event table 114, 116 from the master scheduler 106 in network switch 501. The local event table 114, 116 contains the schedule for the communications which are involving this member node. Note that the reception of the local event table 114, 116 itself is scheduled by the master scheduler 106 in network switch 501. Therefore, the first time that the node is to receive the local event table 114, 116 it must be available to receive that message at any possible time that the local event table 114, 116 might be received.

Step 1506 - The next step for the member node 108, 107 is to formulate the task execution list. The task execution list is formulated from the local event table 114, 116 and the additional tasks which the member node 108, 107 must do which are not described in the local event table 114, 116. The local event table 114, 116 which is communicated from the
5 master scheduler 106 in network switch 501 provides the timing associated with the transmission and reception tasks. In order to formulate the task execution list, the following sub-steps must be taken:

Substep 1506A: Adjust the start time received from the master scheduler 106 in network
10 switch 501 to account for the propagation delay. The timing which is received from the master scheduler 106 in network switch 501 is relative to the master scheduler 106 in network switch 501. Although the member node 108, 107 is synchronized with the master scheduler 106 in network switch 501, there is still an inherent delay in the propagation through the media being communicated. Either this delay must be accounted for by the
15 master scheduler 106 in network switch 501, the member node, or the communication window of time increment must be sufficiently large that the inherent delay is included as overhead in the time allotted to the connection. In the preferred embodiment, this delay is accounted for in the member node.

20 Substep 1506B: Schedule the tasks which are dependent on the communication tasks. There are tasks which must be scheduled prior to a transmission (for example, the transfer of the data to be transmitted from memory); there are also tasks which must be scheduled after a reception (for example, the arbitration of the data between the multiple, different sources in a redundant computing environment and the transfer of data to memory). The
25 member node 108, 107 is responsible for the scheduling of these tasks.

Substep 1506C: Schedule the tasks which are neither described by the local event table 114, 116 nor are dependent on the communications. There are tasks which the member node 108, 107 must accomplish in addition to communication related tasks. These tasks
30 may be for the operation of other I/O, localized display, etc.

Step 1507 - The next step for the member node 108, 107 is to execute the task execution list. Each task in the list contains the task which is to be executed, the start time for the task, and any parameters which are necessary in order to successfully complete the task.

Step 1508 - The member node 108, 107 can now formulate the communication request list, which is a list of the member nodes 108, 107 to which this member node 108, 107 is desiring to communicate. The information which is included in this communication request list is the
5 source node identifier, destination node identifier, and the total length of the communication.

Step 1509 - This communication request list is communicated to the master scheduler 106 in network switch 501 for consideration in its scheduling. At this point, the member node 108, 107 has performed all of the necessary functions in a particular time period. The next step
10 1510 is to determine whether or not it is time to receive the next reference moment message. If it is, then the entire process repeats itself starting with the very first step 1501 of receiving the reference moment message. If it is not time to receive the next reference moment message, then the logic of the member node may forward to step 1505.

15 As one contemplated embodiment, Figure 16 presents the general data flow diagram 130 of Figure 1 as a data flow diagram 1600 where communications are effected via radio frequency means and transmissions 1601, 1602, and 1603 are effected via radio frequency transmitters and receivers.

20 The present invention is enabled through use of machines which are considered as a special purpose computers created in one embodiment by combining a computer with computer program code (alternatively termed, for example, software or logic, where logic denotes software in and for execution by an enabling system of electrical circuitry) directed to the particular function of process control so that, when connected to an electrical power source,
25 powered by electricity, and activated to execute its computer program code (software), the special purpose computer executes the particular function related to the present invention pursuant to the instructions from the computer program code. The computer has electrically activated components denoted as a central processing unit, at least one physical memory connected to the central processing unit, an optional input keyboard connected to the central
30 processing unit, an optional output display (which can include, without limitation, a printer, printing terminal where output is printed on paper, cathode ray tube monitor, and/or flat panel monitor) connected to the central processing unit, a computer clock pulse generator, and a connection to electrical power for energizing all of the electrically activated components of the computer. The central processor further has a control bus and specific
35 computer circuits for either temporarily holding data (for example, a register or an

accumulator) or for executing a fundamental data processing operation (for example, an addition circuit); the specific computer circuits are connected in communication to the control bus through latching circuits which can individually be in either a conducting (communicating) or non-conducting (non-communicating) configuration; the collective configurations of all the latching circuits at a particular moment define a particular state of the central processor; and the state of the central processor is respectively modified and changed as the individual instructions of the computer program code are sequentially accessed by the central processing unit and output from the clock pulse generator is directed to the latches. The computer optionally also has computer program code in the form of a commercially available operating system which functions as a computer program code platform for enabling the execution of the special purpose computer program code directed to the particular function of the present invention. In overview, the present invention in this embodiment is implemented in the preferred embodiment by and through a plurality of computers which have been programmed to contribute to technology by performing the specifically useful purpose and practical application of the described system; each computer functions as a medium for realizing the functionality provided by the computer program code's functionality. In some embodiments, some communication linkages within an extended form of the special purpose computer may be implemented by electrical, optical, or radio-frequency signal communication interfaces and communication media which collectively use electrically-powered transmitter and receiver components which are directly or indirectly linked to at least one central processing unit.

The computer program code resident within a computer provides at least one component part of that machine. The computer program code may be in the form of, without limitation, inter-linked modules, routines and subroutines, program objects, and/or installed interactive processes. In some emergent computer designs, enabled by modern application specific integrated circuitry (ASIC) technology, computer program code may be at least partially expressed in hardwired circuitry reminiscent of traditional hardwired gate arrangements which execute simplified logical scenarios without need for a clock pulse. The Figures of this description show a number structural relationships between key modules and components within the computer program code and other aspects of the invention which enable the functionality of the machine of the present invention to be realized; in this regard, the computing elements in the system are constructed out of a combination of hardware and software.

The present invention has been described in an illustrative manner. In this regard, it is evident that those skilled in the art, once given the benefit of the foregoing disclosure, may now make modifications to the specific embodiment described herein without departing from the spirit of the present invention. Such modifications are to be considered within the scope of the present invention which is limited solely by the scope and spirit of the appended claims.

We claim:

1. A computer implemented method for enabling a plurality of computer nodes in mutual
5 data communication to execute logic in a coordinated manner, comprising:
 - aligning real-time time between said plurality of computer nodes so that each
computer node has essentially the same real-time time;
 - 10 defining a schedule of communications for each computer node, said
schedule having a different time in the future defined for initiating each
respective communication;
 - communicating each schedule to its respective computer node; and
 - 15 initiating each communication from each node at the time defined in its
schedule.
2. A computer network architecture, comprising:
20
 - a message carrier supporting multiple communication paths;
 - a plurality of computers coupled to said message carrier, said computers
being programmed to perform time-deterministic tasks by cycling through a
25 plurality of different operating states, including a communication state;
 - a clock system accessible by all said computers for maintaining time
synchronism among said computers; and
 - 30 a scheduler coupled to said message carrier for determining the sequence of
communication between said computers by causing said message carrier to
establish communication between selected computers based on said time-
deterministic tasks being performed by said computers.

3. A computer implemented method for enabling a computer node in a plurality of computer nodes to execute logic in coordination with the other nodes in said plurality of computer nodes, comprising:

- 5 connecting a message carrier to each computer node in said plurality of computer nodes;
- providing master scheduling capability in one computer node to establish a master scheduler;
- 10 defining, in each computer node, a communication request list of each communication needed by said computer node with any other computer node respective to a future time period;
- 15 sending said communication request list from each computer node to said master scheduler via said message carrier;
- receiving, in each computer node, a communication schedule from said master scheduler via said message carrier;
- 20 defining, in each computer node, a local task schedule from said communication schedule.

4. A computer implemented method for enabling a plurality of member nodes to execute logic in a coordinated manner, comprising:

- 25 connecting a message carrier to each member node in said plurality of member nodes;
- 30 providing a master scheduler in data communication with each said member node via said message carrier;
- defining a reference moment time value for said plurality of member nodes;

communicating said reference moment time value to each member node and said master scheduler via said message carrier;

5 defining a communication request list in each said member node showing each communication needed by said member node with any other member node in a future time period;

communicating said communication request list from each member node to said master scheduler via said message carrier;

10 defining, in said master scheduler, scheduled communications between member nodes at communication event times respective to said reference moment time value and also respective to each said communication listed in each said received communication request list;

15 defining, in said master scheduler, a communication schedule for each member node respective to said scheduled communications ;

20 sending each said communication schedule from said master scheduler to each respective member node via said message carrier;

25 defining, in each member node, the difference in time between said reference moment time value and the time when said reference moment time value is communicated into said member node;

30 defining, in each member node, a local task schedule from said communication schedule and from said difference in time for the member node so that the member node will communicate to other member nodes via said message carrier at said scheduled communication event times;

executing said scheduled communications between said member nodes respective to each scheduled communication via said message carrier according to said respective local task schedules.

5. The method of Claim 4 wherein said step of defining a reference moment time value is performed by said master scheduler.

6. The method of Claim 4 wherein said step of defining a communication request list in each said member node limits said communications to a maximum number in said future time period.

7. The method of Claim 4 wherein each communication has a duration time attribute and the value of said duration time attribute is less than a fixed maximum value.

10

8. The method of Claim 4 wherein the step defining a reference moment time value for said plurality of member nodes is done by a global time system.

9. A computer implemented method for enabling a plurality of member nodes to execute logic in a coordinated manner, comprising:

15

connecting a message carrier to each member node in said plurality of member nodes, each node having a central processing unit;

20

providing a master scheduler in data communication with each said member node via said message carrier;

defining a reference moment time value as a moment in time for said plurality of member nodes;

25

communicating said reference moment time value to each member node and said master scheduler via said message carrier;

30

defining into a communication request list in each said member node, each access to any other member node needed by said member node in achieving a future task increment respective to a future time period;

communicating said communication request list from each said member node to said master scheduler via said message carrier;

5 defining, in said master scheduler, a communication event data object
respective to each said access to any other member node needed by said
member node in achieving a future task increment, said communication event
data object having an accessing member identification attribute value, an
accessed member identification attribute value, and an access duration
attribute value;

10 collecting, in said master scheduler, all defined communication event data
objects into a communication event set;

15 determining in said master scheduler a summed accessed value for each
member node in said communication event set as a sum of all access
duration attribute values in all communication event data objects having a
accessed member identification attribute value respective to said member
node;

20 determining in said master scheduler a summed accessing value for each
member node in said communication event set as a sum of all access
duration attribute values in all communication event data objects having an
accessing member identification attribute value respective to said member
node;

25 scheduling in said master scheduler and into the earliest available time in a
schedule, a future communication task respective to said reference moment
time value between the accessing member node having the highest said
summed accessing value respective to said communication event set and the
accessed member node respective to the accessed member identification
attribute value in the communication event data object having the highest
30 affiliated access duration attribute value for all communication event data
objects respective to said accessing member node, said future
communication task having a task initiation time equal to the earliest available
time in the schedule, an accessing member node identifier attribute respective
to the accessing member, and an accessed member node identifier attribute
35 respective to the accessed member node;

5 removing, in said master scheduler, the communication event data object
respective to the accessed member node identifier attribute in the scheduled
future communication task and the accessed member node identifier attribute
in the scheduled future communication task to redefine the communication
event set to those communication event data objects which have not been
scheduled; and

10 repeating, in said master scheduler, respective to the redefined
communication event set, the steps of determining a summed accessed
value, determining a summed accessing value, scheduling a future
communication task, and removing all communication event data objects
respective to scheduled member nodes until all communication event data
objects have been scheduled into a communication schedule for each
15 member node;

sending each said communication schedule from said master scheduler to
each respective member node via said message carrier;

20 defining, in each member node, the difference in time between said reference
moment time value and the time when said reference moment time value is
communicated into said member node;

25 defining, in each member node, a local task schedule from said
communication schedule and from said difference in time for the member
node so that the member node will communicate to other member nodes via
said message carrier at said scheduled communication event times; and

30 executing said scheduled communications between member nodes respective
to each scheduled communication event via said message carrier according
to said respective local task schedules.

10. A computer implemented method for scheduling communications between member
objects in a plurality of member objects functioning as a collective group, comprising the
35 steps of:

providing a computer;

5 defining in said computer, for each member object, a communication event data object respective to each access to any other member object needed by said member object in achieving a future task increment, said communication event data object having an accessing member identification attribute value, an accessed member identification attribute value, and an access duration attribute value;

10 collecting, in said computer, all defined communication event data objects into a communication event set;

15 determining, in said computer, a summed accessed value for each member object in said communication event set as a sum of all access duration attribute values in all communication event data objects having a accessed member identification attribute value respective to said member object;

20 determining, in said computer, a summed accessing value for each member object in said communication event set as a sum of all access duration attribute values in all communication event data objects having an accessing member identification attribute value respective to said member object;

25 defining a set of sequential future time increments within a scheduling period respective to a schedule;

30 scheduling, in said computer, into the earliest available time in the first of said future time increments, a future communication task based on the first available communication event data object in said communication event set between

the accessing member object having the highest said summed accessing value respective to said communication event set and

the accessed member object having the highest said summed accessed value respective to said communication event set if a communication event data object exists between said accessed member object and said accessing member object,

5

said future communication task having a task initiation time equal to the earliest available time in the schedule, an accessing member object identifier attribute respective to the accessing member object, an accessed member object identifier attribute respective to the accessed member object, and said access duration attribute value;

10

removing from scheduling candidacy, in said computer, the communication event data object respective to the accessed member object identifier attribute in the scheduled future communication task and the accessing member object identifier attribute in the scheduled future communication task to redefine the communication event set to those communication event data objects which have not been scheduled;

15

repeating, in said computer respective to the redefined communication event set, the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future communication task, and removing all communication event data objects respective to scheduled member objects until all communication event data objects have been scheduled which can be scheduled into said first future time increment; and

20

25

recurrently executing, in said computer respective to the redefined communication event set and sequentially respective to each of the remaining of said plurality of future time increments, the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future communication task, and removing all communication event data objects respective to scheduled member objects, and repeating such steps until all communication event data objects have been scheduled which can be scheduled into each said future time increment and until all communication event data objects have been scheduled into said scheduling period.

30

11. The method according to 10, wherein said defining is confined to future task increments which are needed within a predefined time period and said scheduling is respective to said predefined time period.
- 5
12. The method according to 11, further comprising the step of connecting a message carrier to each member object in said plurality of member objects, said message carrier having a plurality of channels for permitting concurrency in said communication events, said step of scheduling being respective to said plurality of channels as a group wherein said
- 10 earliest available time is respective to any channel in said message carrier.
13. The method according to Claim 12 wherein each said member object is a member node being in data communication with said message carrier, each member node having a central processing unit.
- 15
14. A computer implemented method for scheduling tasks for resource objects in a plurality of resource objects functioning as a collective group, comprising the steps of:
- providing a computer;
- 20
- defining, in said computer, a set of future tasks for said collective group;
- defining, in said computer, each resource object needed for each future task in said set of future tasks;
- 25
- defining, in said computer, each interaction needed between any two resource objects respective to said set of future tasks;
- defining, in said computer, a use instance for each said resource object in achieving each said interaction, said use instance having an use duration attribute value;
- 30
- defining, in said computer, an accessing resource object and an accessed resource object respective to each said use instance;

5 firstly determining, in said computer, a summed accessing time value for each accessing resource object as a sum of all use duration attribute values for all said use instances respective to that resource object when it is the accessing resource object;

10 secondly determining, in said computer, a summed accessed time value for each accessed resource object as a sum of all use duration attribute values for all said interactions between said accessed resource object and said accessing resource object;

defining, in said computer, a set of sequential future time increments within a scheduling period respective to a schedule;

15 scheduling, in said computer, into the earliest available time in the first of said future time increments, a future resource interaction event based on the first available future task in said set of future tasks between

20 the accessing resource object having the highest said summed accessing value respective to said set of future tasks and

25 the accessed resource object having the highest said summed accessed value respective to said set of future tasks if a use instance exists between said accessed resource object and said accessing resource object,

30 said future resource interaction event having an initiation time equal to the earliest available time in the schedule, an accessing resource object identifier attribute respective to the accessing resource object, an accessed resource object identifier attribute respective to the accessed resource object, and said use duration attribute value;

35 removing from scheduling candidacy, in said computer, the use instance respective to the accessed resource object in the scheduled future resource interaction event and the accessing member object identifier attribute in the

scheduled future resource interaction event to redefine the set of future tasks to those future tasks for which interactions have not been scheduled;

repeating, in said computer, respective to the redefined future task set, the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future communication task, and removing all use instances respective to scheduled resource objects until all use instances have been scheduled which can be scheduled into said first future time increment; and

recurrently executing, in said computer, respective to the redefined future task set and sequentially respective to each of the remaining of said plurality of future time increments, the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future resource interaction event, removing all resource objects respective to scheduled resource objects, and repeating such steps until all resource interaction events have been scheduled which can be scheduled into each future time increment and until all resource interaction events have been scheduled into said scheduling period.

15. A computer implemented apparatus for enabling a plurality of computer nodes in mutual data communication to execute logic in a coordinated manner, comprising:

means for aligning real-time time between said plurality of computer nodes so that each computer node has essentially the same real-time time;

means for defining a schedule of communications for each computer node, said schedule having a different time in the future defined for initiating each respective communication;

means for communicating each schedule to its respective computer node;

means for initiating each communication from each node at the time defined in its schedule.

16. A computer-implemented apparatus having a message carrier, comprising:

- 5 a plurality of member nodes, each member node being in data communication with said message carrier, each member node having
- a central processing unit,
- 10 time phase logic for receiving from said message carrier a reference moment message having a reference moment time value and for defining the difference in time between said reference moment time value and the time of receiving said reference moment message,
- 15 communication definition logic for identifying each internodal communication needed by said member node in a future time period and for establishing a communication request list respective to all said internodal communications,
- 20 communication request sending logic for communicating said communication request list via said message carrier to a master scheduler,
- 25 communication task receiving logic for receiving a communication schedule via said message carrier from said master scheduler,
- 30 task integration logic for defining a local task schedule from said communication schedule and said difference in time, and
- communication implementation logic for communicating via said message carrier with other member nodes according to said local task schedule; and
- a master scheduler in data communication with said plurality of member nodes via said message carrier, said master scheduler

having scheduler logic executed by a central processing unit, said scheduler logic having

5 reference moment definition logic for defining said reference moment time value as a moment in time,
reference moment communication logic for creating a reference moment message having said reference moment time value and for communicating said reference moment message via said message carrier to each member node,

10 communication request receiving logic for receiving each said communication request list from each said member node via said message carrier,

15 communication schedule definition logic for scheduling communications between member nodes at communication event times respective to said reference moment time value and also respective to each said internodal communication listed in each said received communication request list and for further defining a
20 communication schedule for each member node respective to said scheduled communications, and

member node communication schedule response logic for sending each said communication schedule to each respective member node
25 via said message carrier.

17. The apparatus of Claim 16, wherein each member node further comprises a variable speed oscillator.

30 18. The apparatus of Claim 16, wherein each member node further comprises time alignment logic and oscillator rate adjustment logic.

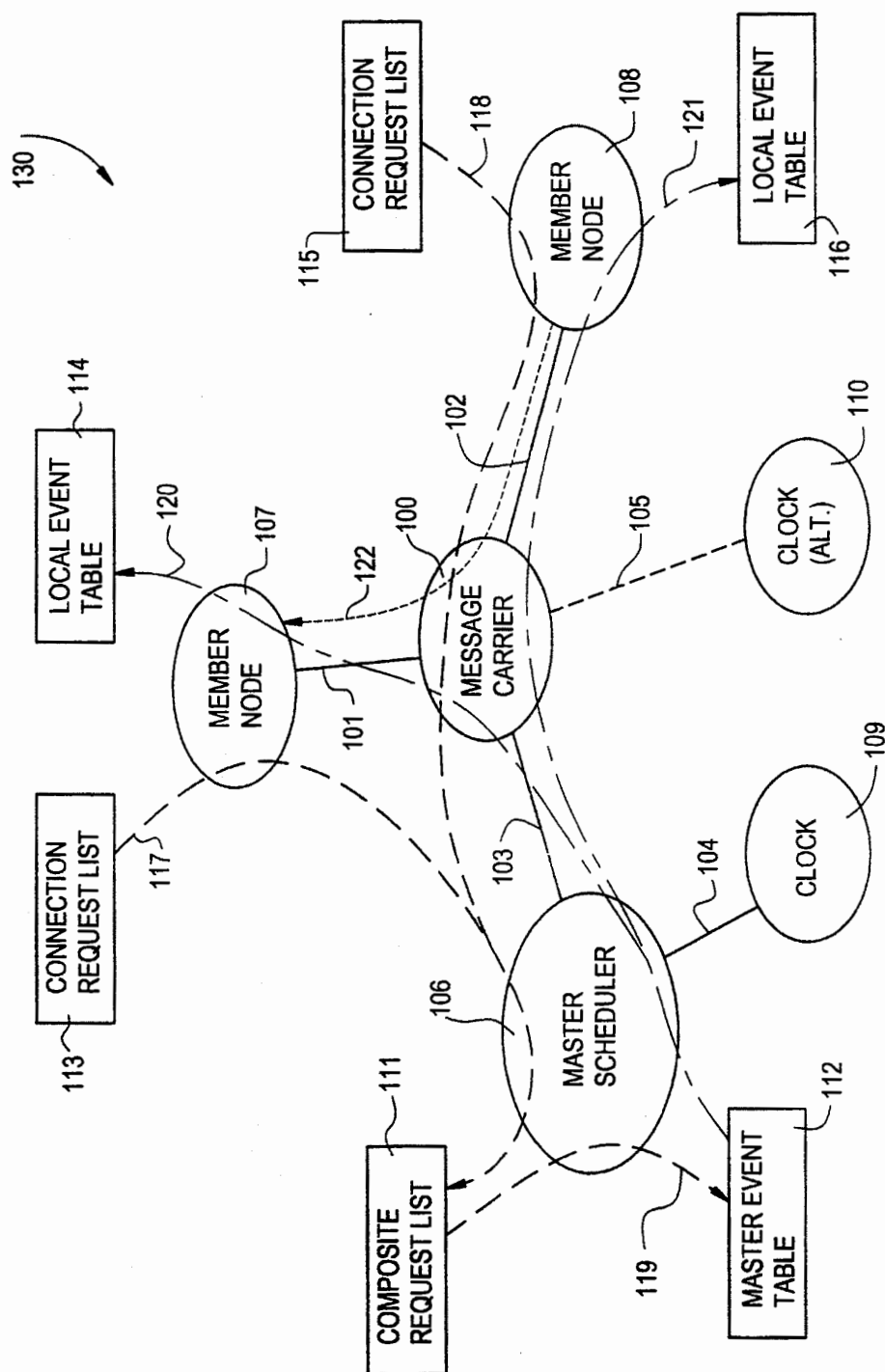
19. The apparatus of Claim 16, wherein said message carrier comprises a crossbar switch.

20. The apparatus of Claim 16, wherein said message carrier comprises a network topology.

21. The apparatus of Claim 16, wherein said message carrier comprises a radio frequency
5 communications system.

1 / 16

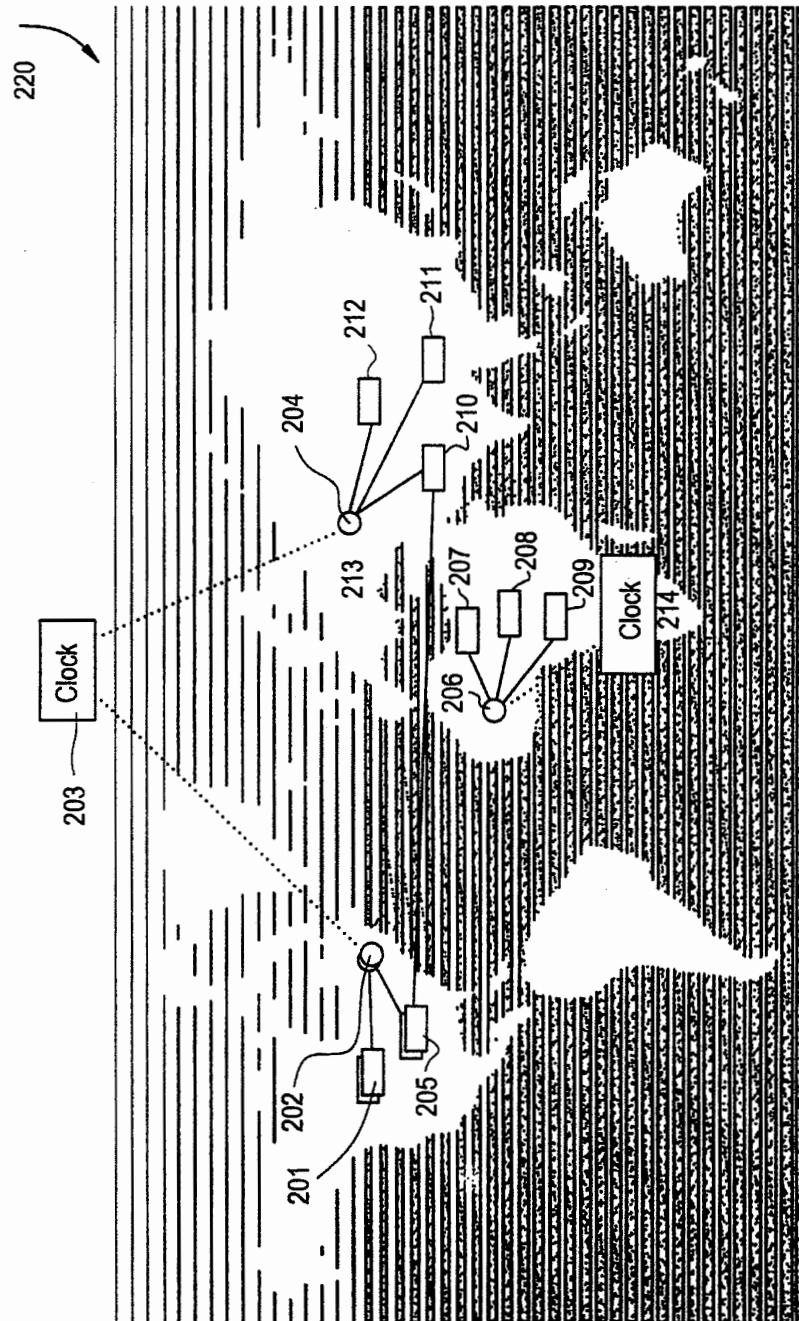
FIG. 1



SUBSTITUTE SHEET (RULE 26)

2 / 16

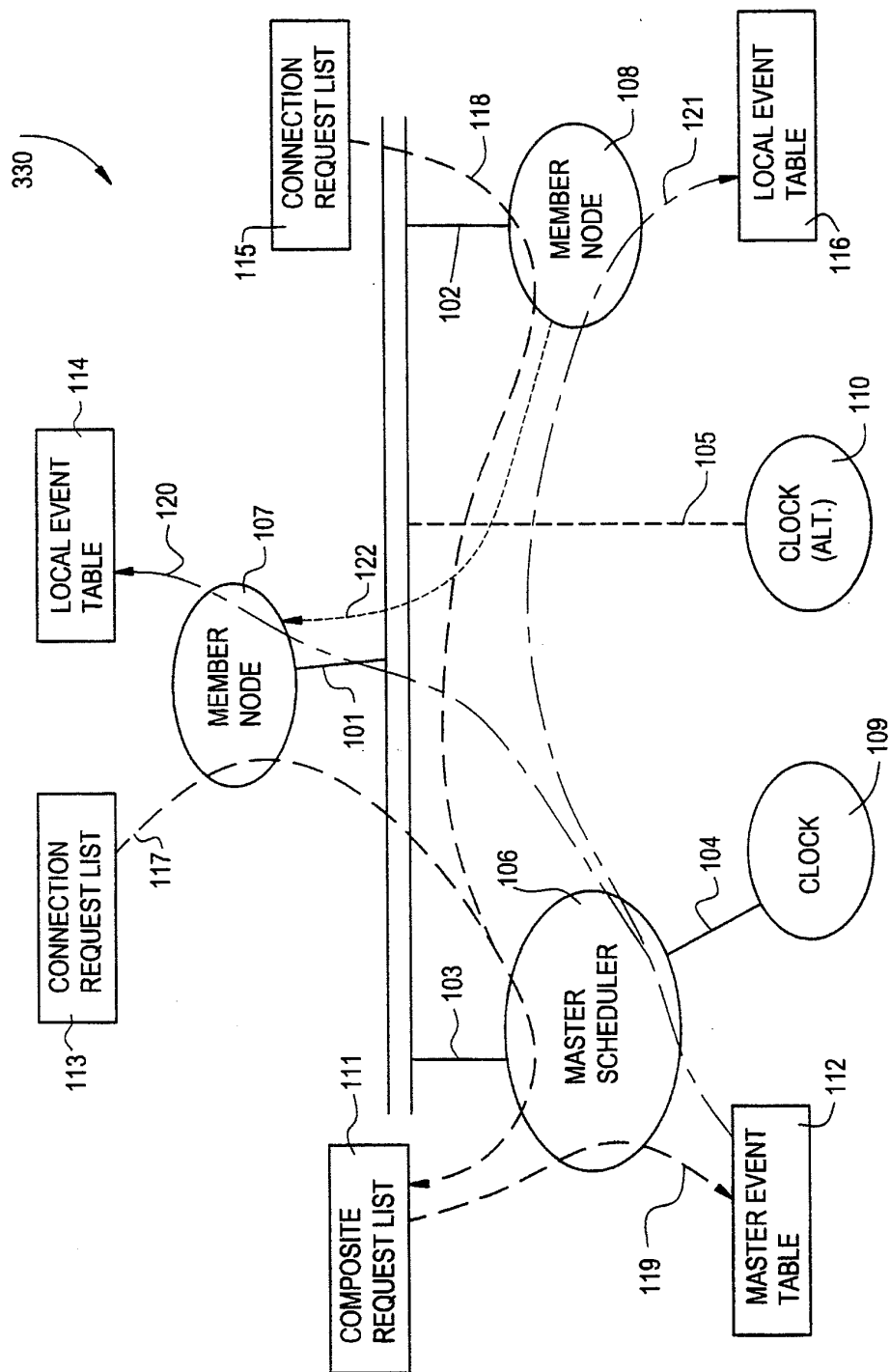
FIG. 2



SUBSTITUTE SHEET (RULE 26)

3 / 1 6

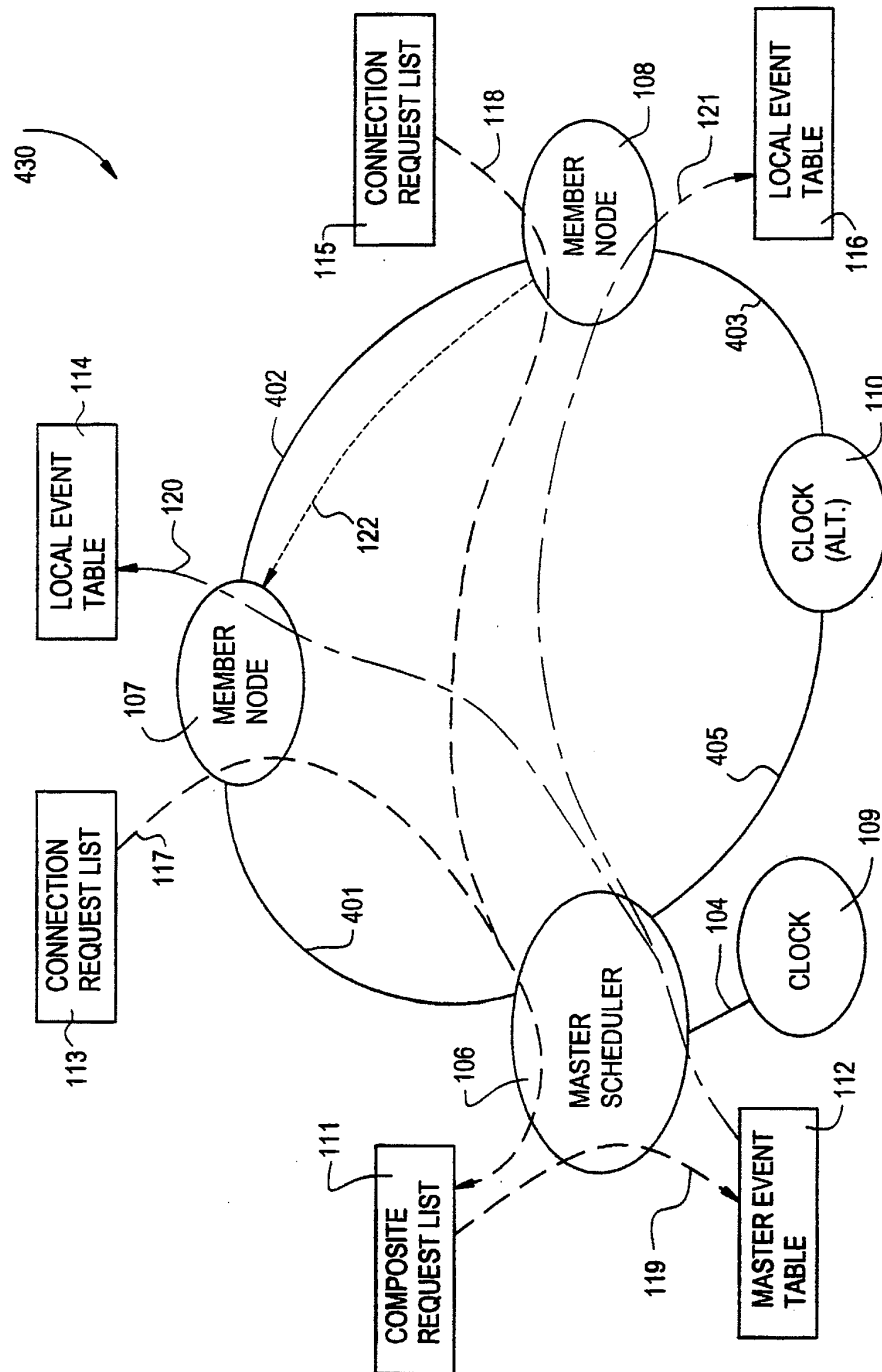
FIG. 3



SUBSTITUTE SHEET (RULE 26)

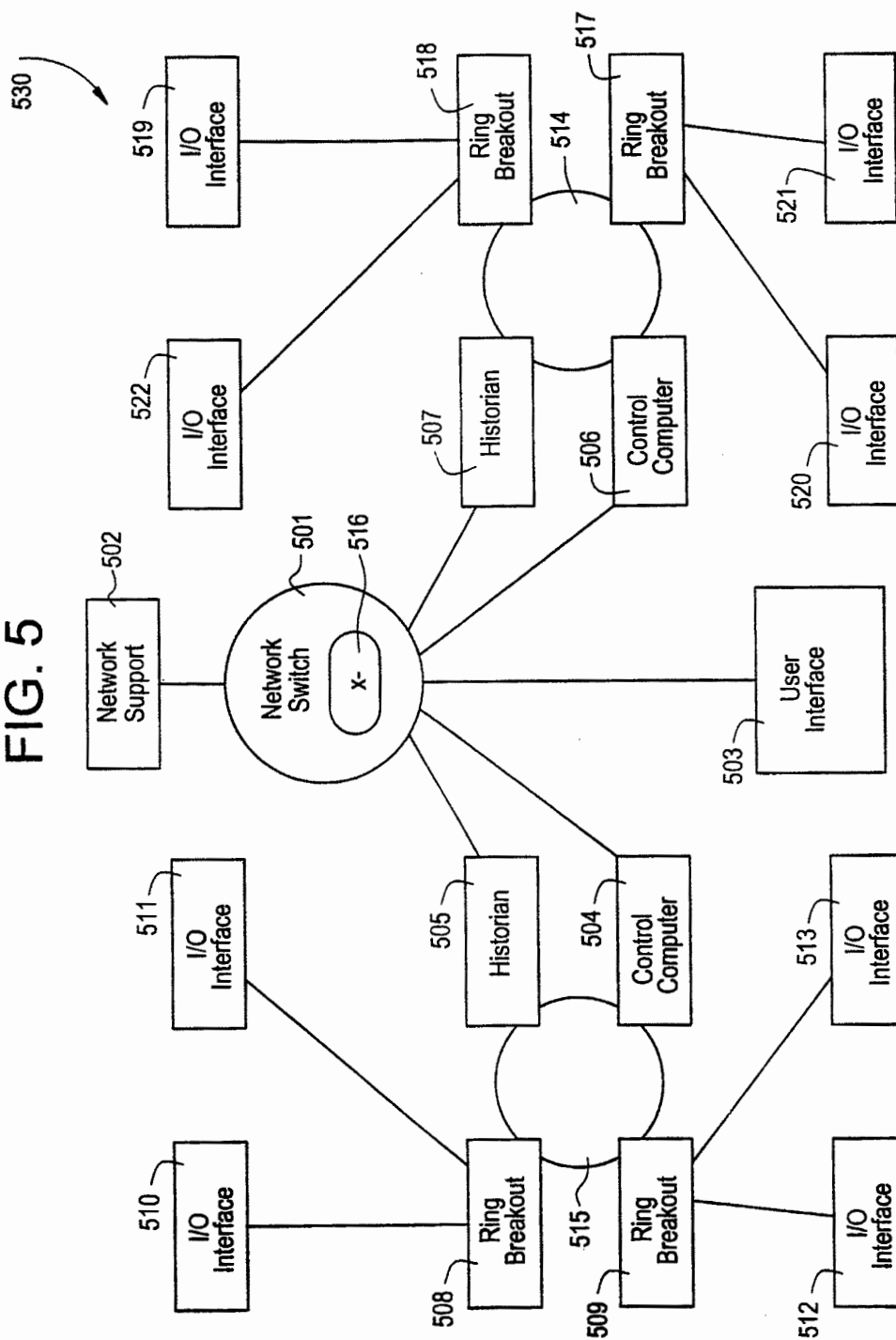
4 / 16

FIG. 4

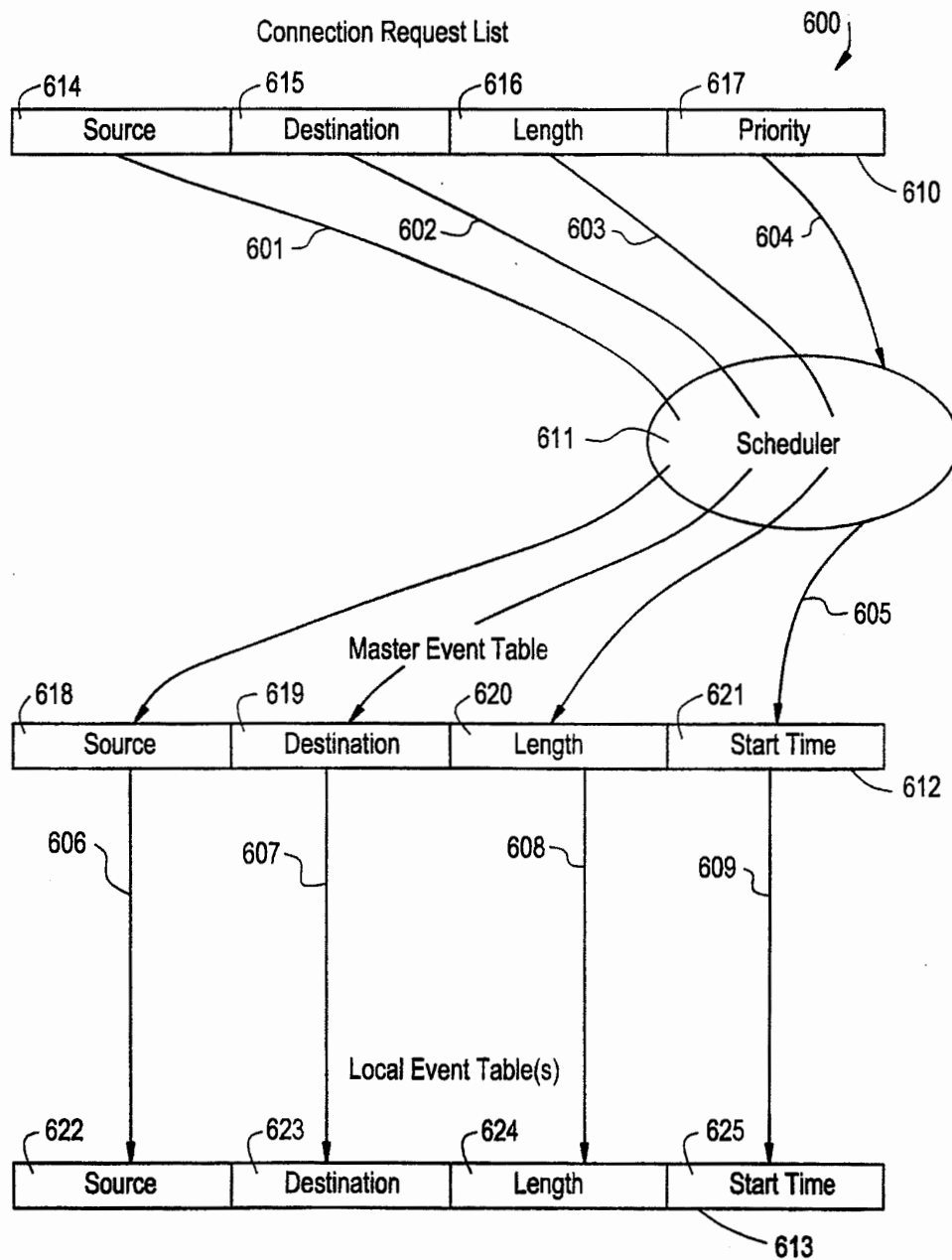


SUBSTITUTE SHEET (RULE 26)

FIG. 5



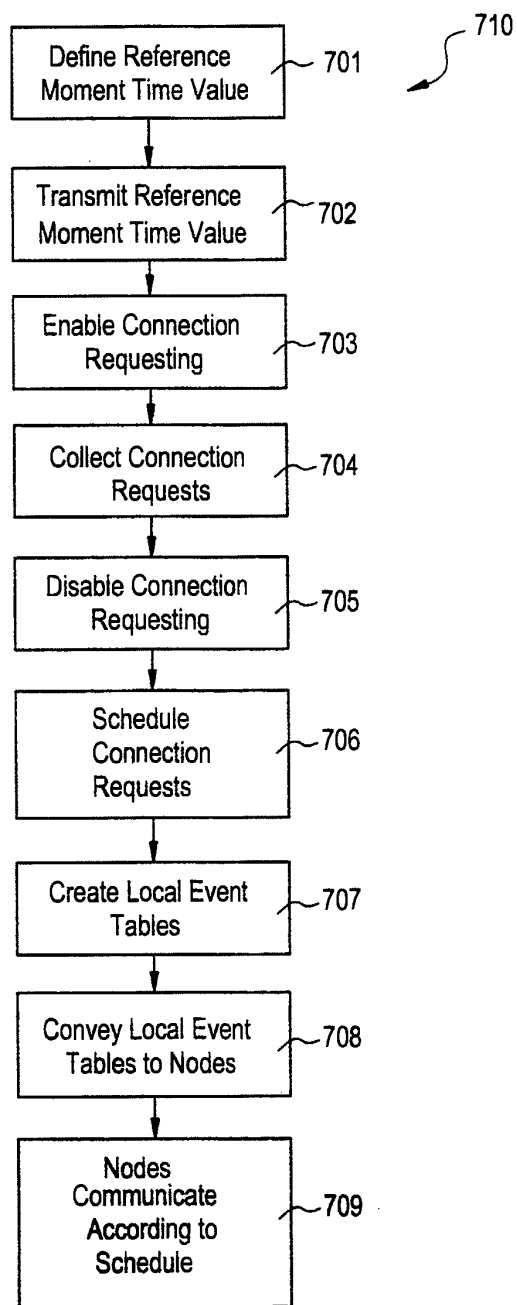
6 / 1 6
FIG. 6



SUBSTITUTE SHEET (RULE 26)

7 / 1 6

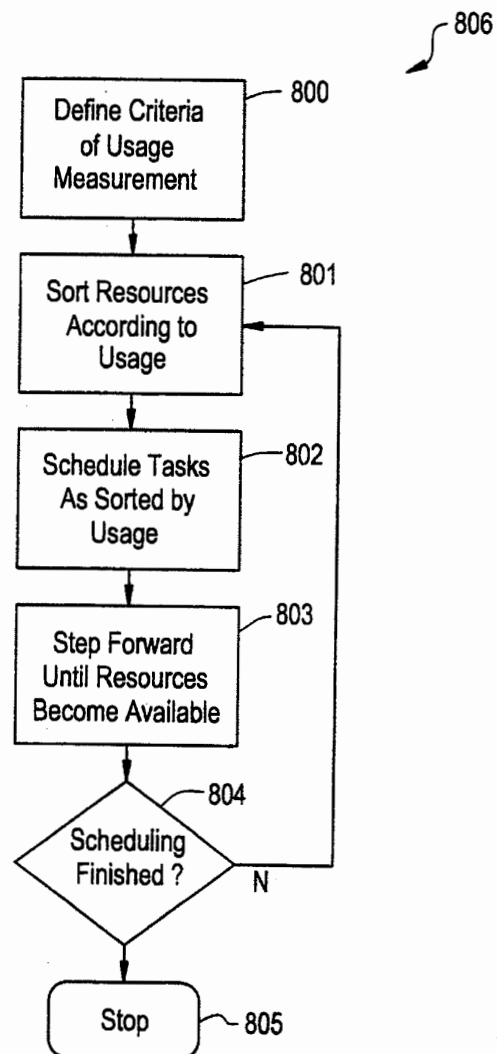
FIG. 7



SUBSTITUTE SHEET (RULE 26)

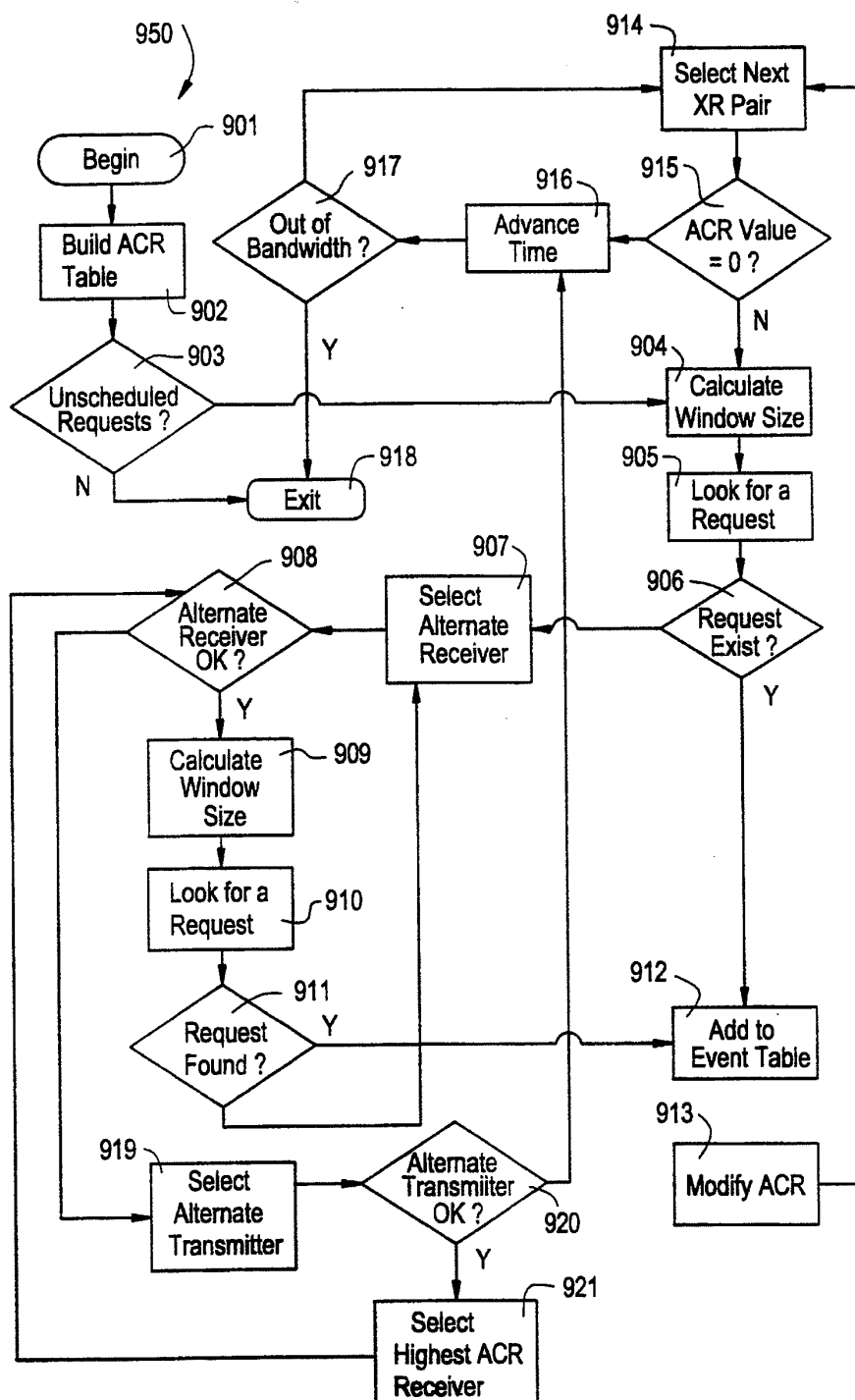
8 / 16

FIG. 8



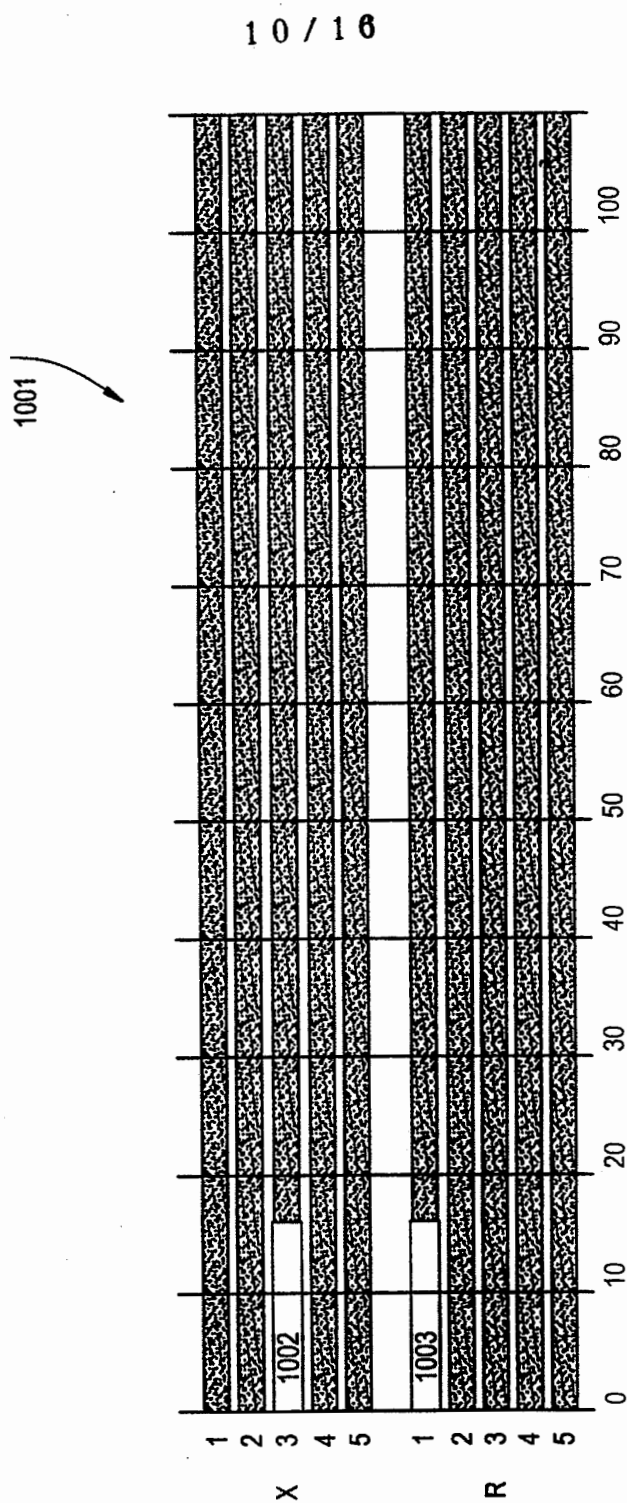
SUBSTITUTE SHEET (RULE 26)

FIG. 9



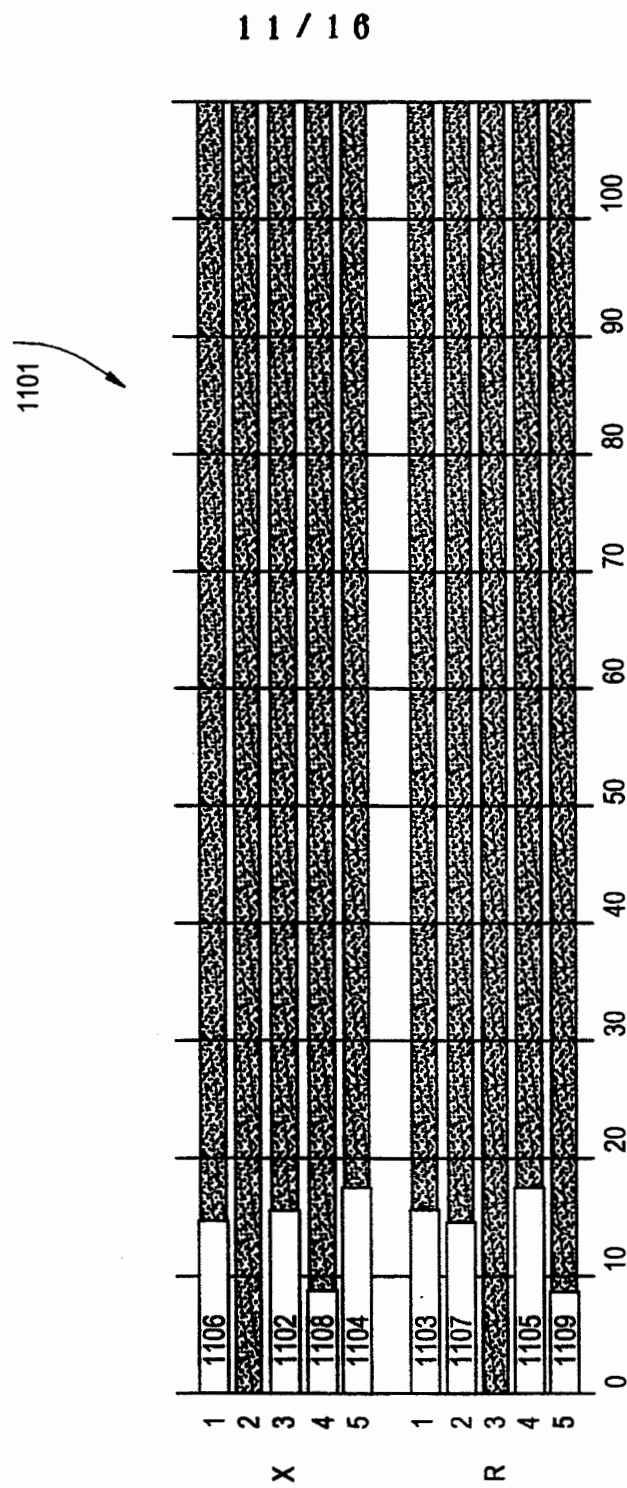
SUBSTITUTE SHEET (RULE 26)

FIG. 10



SUBSTITUTE SHEET (RULE 26)

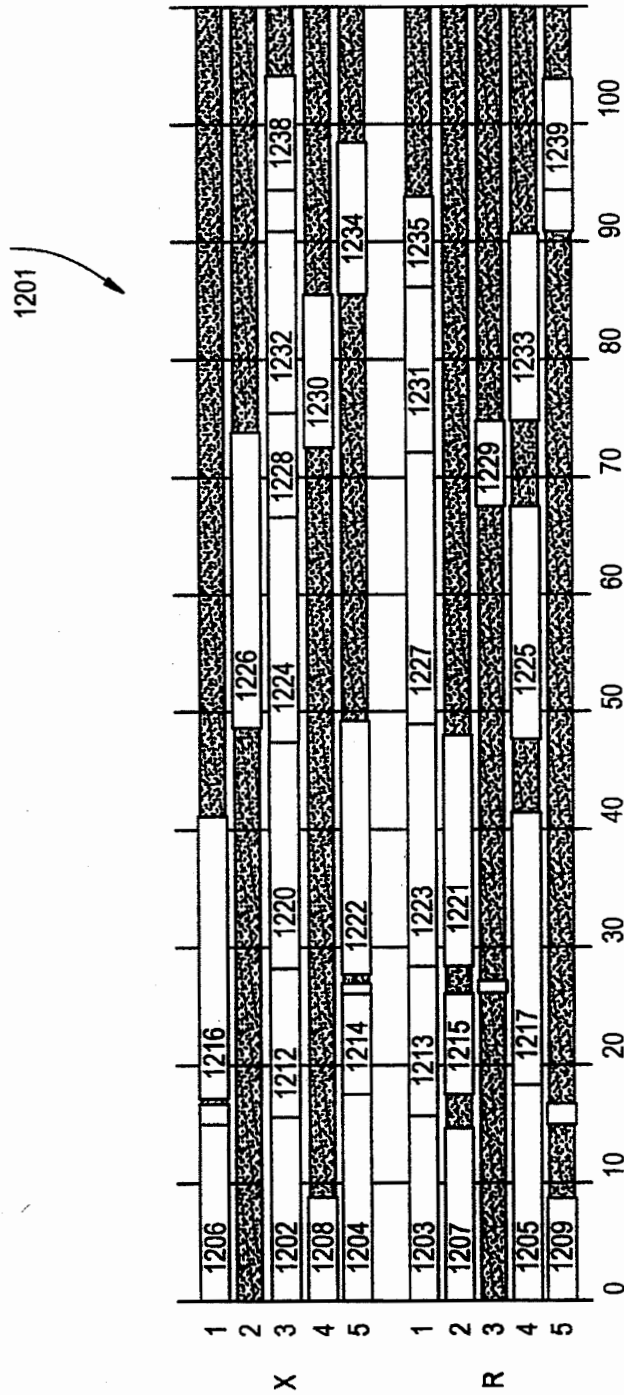
FIG. 11



SUBSTITUTE SHEET (RULE 26)

12 / 16

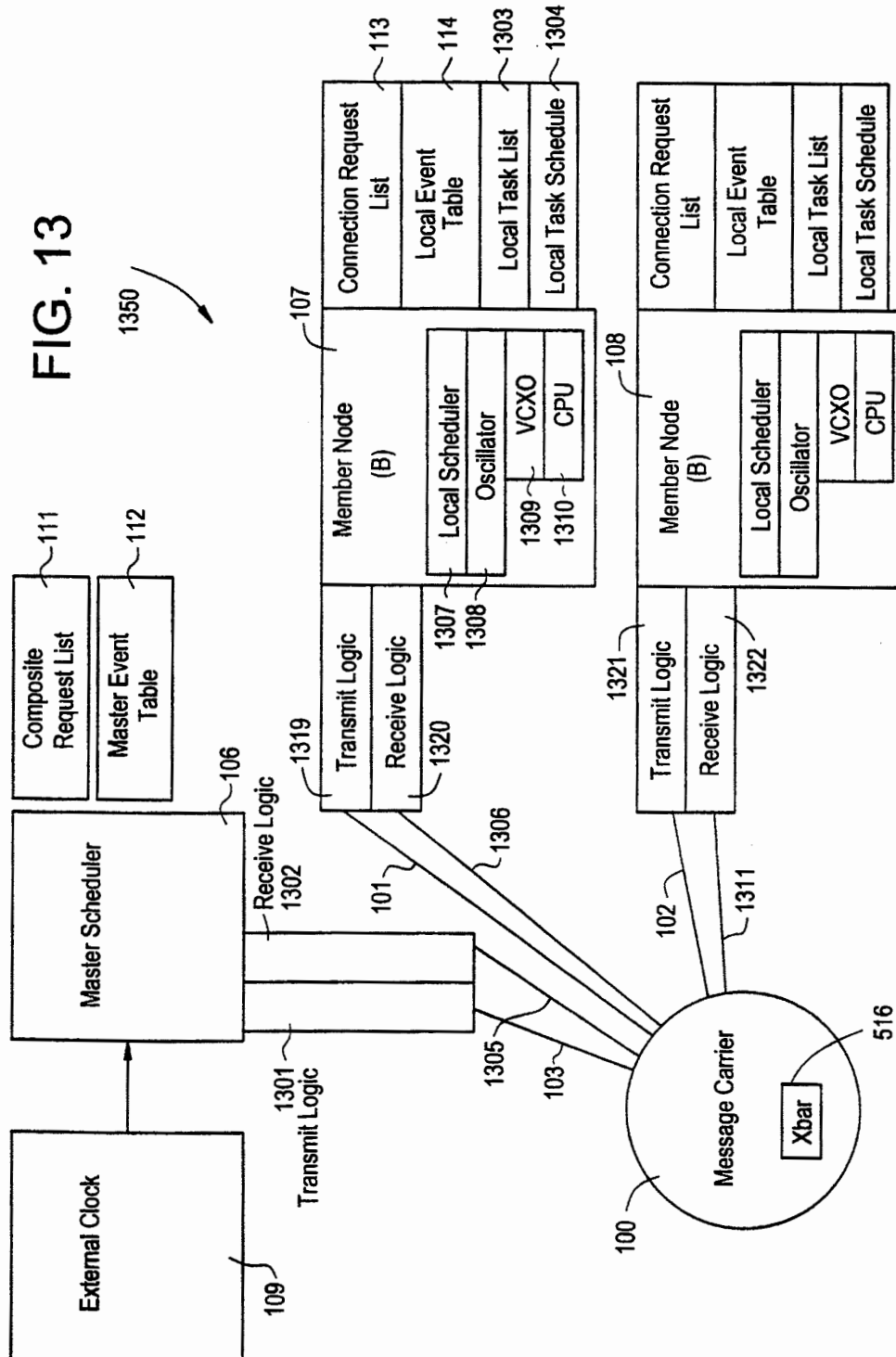
FIG. 12



SUBSTITUTE SHEET (RULE 26)

13 / 16

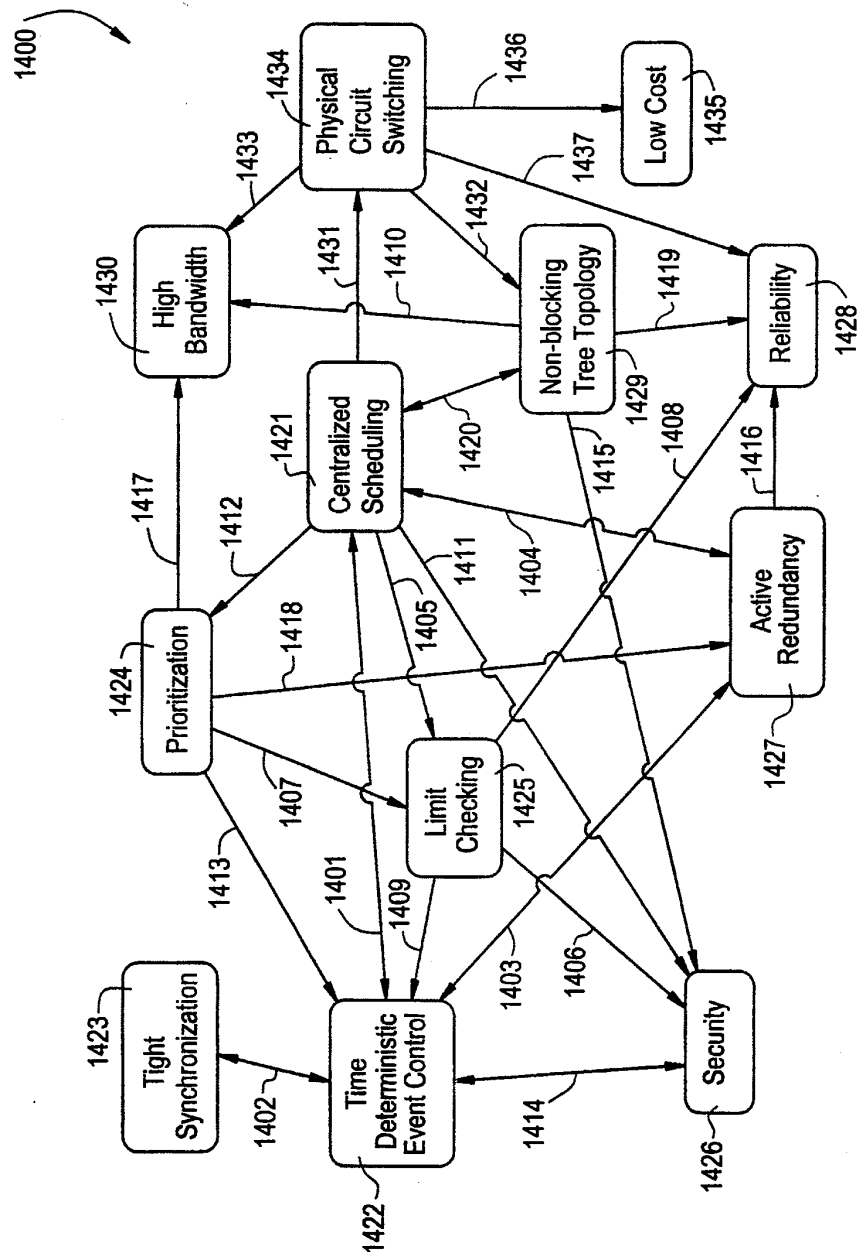
FIG. 13



SUBSTITUTE SHEET (RULE 26)

14 / 16

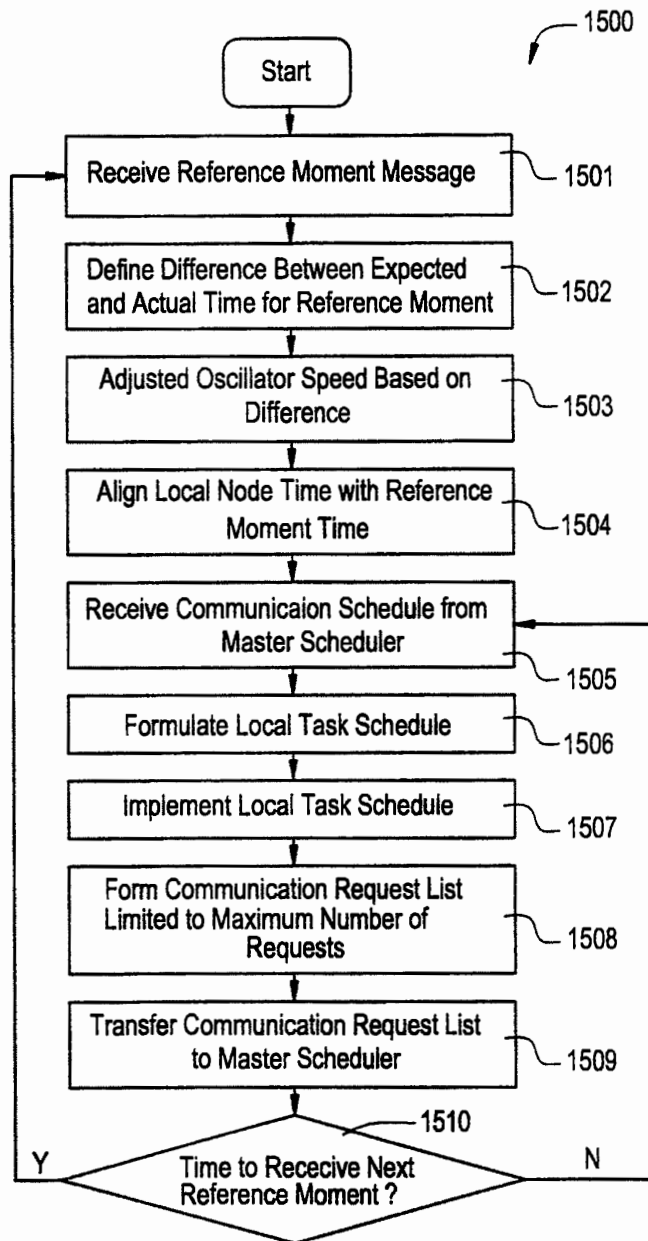
FIG. 14



SUBSTITUTE SHEET (RULE 26)

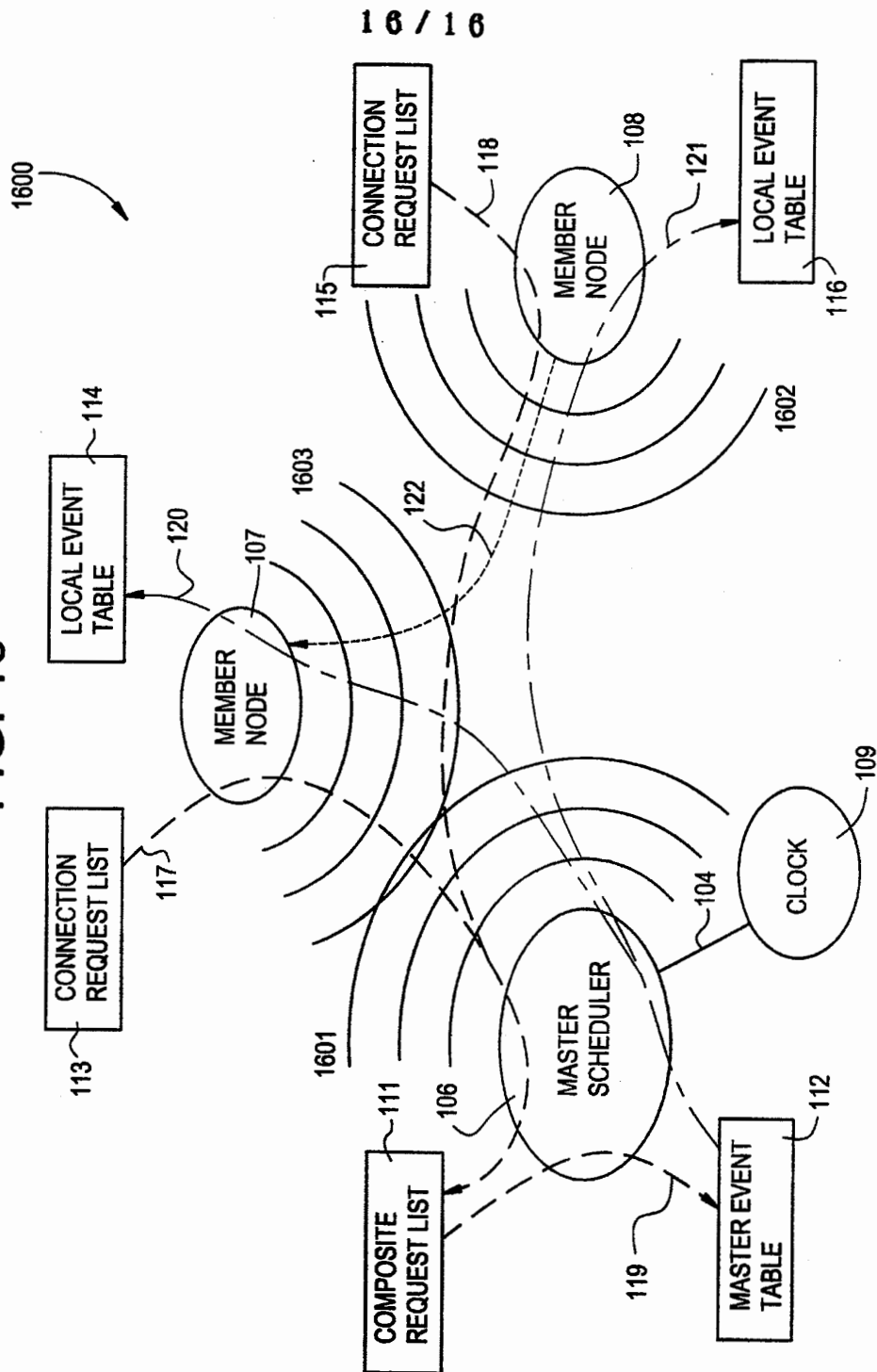
15 / 16

FIG. 15



SUBSTITUTE SHEET (RULE 26)

FIG. 16



SUBSTITUTE SHEET (RULE 26)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 99/06634

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F9/46

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 6 G06F

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SHIN K G ET AL: "REAL-TIME COMPUTING: A NEW DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING" PROCEEDINGS OF THE IEEE, vol. 82, no. 1, 1 January 1994 (1994-01-01), pages 6-23, XP000435878 ISSN: 0018-9219 page 8, paragraph II - page 17, paragraph VII	1-4, 10, 11, 14-16
A	EP 0 183 877 A (INMOS LTD) 11 June 1986 (1986-06-11) the whole document	1-4, 10, 11, 14-16
A	EP 0 250 011 A (INMOS LTD) 23 December 1987 (1987-12-23) the whole document	1-4, 10, 11, 14-16
-/--		

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in 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 document 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

27 July 1999

Date of mailing of the international search report

03/08/1999

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Michel, T

Form PCT/ISA/210 (second sheet) (July 1992)

page 1 of 2

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 99/06634

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>RAMAKRISHNAN K V: "AN EVENT SCHEDULER FOR CONTROL APPLICATIONS"</p> <p>EDN ELECTRICAL DESIGN NEWS,</p> <p>vol. 41, no. 12, 6 June 1996 (1996-06-06),</p> <p>pages 155-158, 160, 16, XP000622016</p> <p>ISSN: 0012-7515</p> <p>the whole document</p> <p>-----</p>	9-11

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

page 2 of 2

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 99/06634

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0183877 A	11-06-1986	NONE	
EP 0250011 A	23-12-1987	DE 3375844 A	07-04-1988
		DE 3376894 A	07-07-1988
		DE 3378083 A	27-10-1988
		DE 3382708 A	16-09-1993
		DE 3382708 T	10-02-1994
		EP 0111399 A	20-06-1984
		EP 0113178 A	11-07-1984
		EP 0110642 A	13-06-1984
		EP 0113516 A	18-07-1984
		EP 0110643 A	13-06-1984
		JP 1028967 B	07-06-1989
		JP 1547176 C	28-02-1990
		JP 59149543 A	27-08-1984
		JP 1789597 C	29-09-1993
		JP 4070652 B	11-11-1992
		JP 59133647 A	01-08-1984
		JP 1982715 C	25-10-1995
		JP 6101042 B	12-12-1994
		JP 59194256 A	05-11-1984
		JP 2111325 C	21-11-1996
		JP 8007739 B	29-01-1996
		JP 59133662 A	01-08-1984
		JP 2502960 B	29-05-1996
		JP 59133650 A	01-08-1984
		JP 2509453 B	19-06-1996
		JP 6342378 A	13-12-1994
		JP 2688320 B	10-12-1997
		JP 6260551 A	16-09-1994
		US 5452467 A	19-09-1995
		US 5506437 A	09-04-1996
		US 5491359 A	13-02-1996
		US 5031092 A	09-07-1991
		US 4680698 A	14-07-1987
		US 4704678 A	03-11-1987
		US 4724517 A	09-02-1988
		US 5243698 A	07-09-1993
		US 4819151 A	04-04-1989
		US 4967326 A	30-10-1990

Form PCT/ISA/210 (patent family annex) (July 1992)

Electronic Acknowledgement Receipt	
EFS ID:	15721089
Application Number:	13864249
International Application Number:	
Confirmation Number:	9806
Title of Invention:	SYSTEM AND METHOD FOR SYNCHRONIZING OPERATIONS AMONG A PLURALITY OF INDEPENDENTLY CLOCKED DIGITAL DATA PROCESSING DEVICES
First Named Inventor/Applicant Name:	Nicholas A. J. Millington
Customer Number:	109473
Filer:	Robert Johnson Irvine
Filer Authorized By:	
Attorney Docket Number:	SNS-71301.US.11 (13-0402)
Receipt Date:	08-MAY-2013
Filing Date:	
Time Stamp:	18:46:16
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment		no			
File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Form (SB08)	71301-US11-IDS.pdf	712396	no	19
			020473d426ab9ea8eb2a492ad4f95e7d50959b3f		
Warnings:					
Information:					

2	Foreign Reference	EP0251584A2-NorthernTelecom.pdf	1871427	no	33
			e4060935063351b4642542a8f21ee0f379f46cdc		
Warnings:					
Information:					
3	Foreign Reference	EP0672985A1-IBM.pdf	1715502	no	31
			82829d8bae9a69fe31d924894be437476c07d679		
Warnings:					
Information:					
4	Foreign Reference	EP1111527B1.pdf	2175253	no	38
			54fb66a34a8c2836746d8102e61cd137b0659bdf		
Warnings:					
Information:					
5	Foreign Reference	WO9525313.pdf	707368	no	25
			11f89c485fe088f060c29d4ea21245e5a843160e		
Warnings:					
Information:					
6	Foreign Reference	WO9961985.pdf	3653814	no	96
			6c32d809f0d15c1a00bce9a3f87c4a0b7b334fd1		
Warnings:					
Information:					
7	Non Patent Literature	AKYILDIZ-MultimediaGroup.pdf	2498395	no	12
			69554162fc4e51427d9c319e879a433693ddbfc		
Warnings:					
Information:					
8	Non Patent Literature	Benslimane-MultimediaSynchronization.pdf	1209082	no	8
			2ff64e8ed450d8cb5de0108808864985ed162ae5		
Warnings:					
Information:					
9	Non Patent Literature	Intra-InterStreamSync-Biersack.pdf	731857	no	10
			35794d83dd743d327e07825f217b68829a97479f		
Warnings:					
Information:					
10	Non Patent Literature	MPEG2Tutorial-Bretl.pdf	1294244	no	23
			911b6ab9efcf6e86f994de6b6a821bd9bd1a117		
Warnings:					
Information:					

11	Non Patent Literature	Huang-SyncInfrastructure.pdf	2480357	no	11
			b781148c0e2b6051001ce7913f481fbb079ed66d		
Warnings:					
Information:					
12	Non Patent Literature	PCTUS04-023102-IntlSearchReport.pdf	239507	no	6
			63ef304c9af50597a27e0d476723a348a893e612		
Warnings:					
Information:					
13	Non Patent Literature	Ishibashi-GroupSync-LiveMedia.pdf	1692122	no	7
			e5e93fef756c319896baa9cf95b6dffa1338616a		
Warnings:					
Information:					
14	Non Patent Literature	Ishibashi-GroupSync-StoredMedia.pdf	1672500	no	9
			08c674c017aee8409b6fd6327e7e0e3d23780544		
Warnings:					
Information:					
15	Non Patent Literature	Joreference-synchronizedonetomany.pdf	542871	no	11
			f6969e60ce4a5f4cd6d00b0627023766db5eafe		
Warnings:					
Information:					
16	Non Patent Literature	Mills-NetworkTimeProtocol.pdf	6081281	no	96
			f6207bf3fd09206c3d5571514be05927e7a684ba		
Warnings:					
Information:					
17	Non Patent Literature	Mills-PrecisionClockSync.pdf	1366650	no	16
			53928c660ae7df7414888151a0eabdd85db75ca9		
Warnings:					
Information:					
18	Non Patent Literature	Motorola2001SimplefiManual.pdf	945125	no	111
			163341eb0c6d16fff34c3ee075d3e8b5afc30a61		
Warnings:					
Information:					
19	Non Patent Literature	Park-GroupSyncMechanism.pdf	354862	no	5
			0b1939edff956eb25c657d70adb9bb60df2b2ce5		
Warnings:					
Information:					

20	Non Patent Literature	Prismiq-UserGuide.pdf	647554	no	44
			ac49597896459778a52c6b4a843a0d3bfb81405b		
Warnings:					
Information:					
21	Non Patent Literature	Rothermel-AdaptiveStreamSync.pdf	1009707	no	22
			7fa20df4279b93306158108cc4100a36a685568b		
Warnings:					
Information:					
22	Non Patent Literature	Schulzrinne-RTP-TransportProtocol.pdf	4486868	no	89
			045bf0b66a252ccb18f334231c9ebe42f165c80b		
Warnings:					
Information:					
23	Non Patent Literature	MPEG2Transport.pdf	317893	no	5
			fdd15d79a37834e2d53ee663cae10b090f507ce		
Warnings:					
Information:					
24	Non Patent Literature	UPnP-reprint.pdf	5649097	no	54
			c30d43df6be46fbed6e7f0168f57fdae1e66ed40		
Warnings:					
Information:					
25	Non Patent Literature	AudioTronQuickStartGuide2.pdf	2102621	no	24
			4b12d08d4da7e9d34220dd6edfb49f5bb632eac17		
Warnings:					
Information:					
26	Non Patent Literature	AudiotronReferenceManual-version3-70p.pdf	6675703	no	70
			8996836378ac6cc677323bd9a88105b716386ff5		
Warnings:					
Information:					
27	Non Patent Literature	Audiotronsetupguide-version1.pdf	4645843	no	38
			070ec3b852e417efee9fc22b8842cf8a2e47d24c		
Warnings:					
Information:					
28	Non Patent Literature	10816217-finalOA-13JUL2009.pdf	494964	no	16
			fdb38a5dd85ff71a57798e8efe5c72ab2152327d		
Warnings:					
Information:					

29	Non Patent Literature	10816217-finalOA-21OCT2011.pdf	640324	no	19
			07f58dff07fb2e273c0f176593c25b69252c8d91		
Warnings:					
Information:					
30	Non Patent Literature	10816217-finalOA-28JAN2011.pdf	672280	no	21
			e87a3a10b4b4db76617f43bf151de7f860d9c66a		
Warnings:					
Information:					
31	Non Patent Literature	10816217-finalOA-30JUN2008.pdf	838638	no	30
			dd0e8f7fd8add62c9cb2db62a82bc8439c9a8fc		
Warnings:					
Information:					
32	Non Patent Literature	10816217-nonfinalOA-18JAN2008.pdf	1135142	no	38
			59ce0684e9c6d36094203bf1817b88cc48971613		
Warnings:					
Information:					
33	Non Patent Literature	10816217-nonfinalOA-21JUN2011.pdf	419738	no	13
			c1ee6116de2b310276450c61abe974cfe7a54aee		
Warnings:					
Information:					
34	Non Patent Literature	10816217-nonfinalOA-22JAN2009.pdf	577911	no	18
			949b7a488f2fd167d23950d541cfd319f52d73d3		
Warnings:					
Information:					
35	Non Patent Literature	10816217-nonfinalOA-25JUN2010.pdf	539926	no	17
			6b0333e9cc2573392dad5b0110c331f45697660c		
Warnings:					
Information:					
36	Non Patent Literature	10816217-NOA-27DEC2011.pdf	691357	no	16
			16a70ecdb799f369f8830e28a005b379434d801f		
Warnings:					
Information:					
37	Non Patent Literature	13297000-nonfinalOA-29FEB2012.pdf	308664	no	10
			a11567d38b79990e922273a4a24768f28ad12e7a		
Warnings:					
Information:					

38	Non Patent Literature	13297000-FinalOA-13SEP2012.pdf	515124 304e3353e85741d7c9ed90b908213e6fc2166895	no	16
Warnings:					
Information:					
Total Files Size (in bytes):				64313967	
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

PATENTS

Attorney Docket No. RIN-001

DECLARATION AND POWER OF ATTORNEY

As a below-named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled "System and Method For Synchronizing Operations Among A Plurality Of Independently Clocked Digital Data Processing Devices," the specification of which is filed herewith and identified by Attorney Docket No. RIN-001.

I hereby state that I have reviewed and understand the contents of the above-identified application specification, including the claims.

I acknowledge the duty to disclose information that is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, section 1.56(a).

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment or both under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

I hereby appoint Richard A. Jordan, Reg. No. 27,807 my attorney, with full power of substitution, delegation and revocation, to prosecute this application, to make alterations and amendments therein, to receive the patent and to transact all business in the Patent and Trademark Office connected therewith. Please direct all telephone calls to Richard A. Jordan at (781) 431-1357. Please address all correspondence to Richard A. Jordan, at P. O. Box 81363, Wellesley Hills, MA 02481-0004.


Nicholas A. J. Millington

3/31/04
Date

Residence: 1221 Salsipuedes Street
Santa Barbara, CA 93103

Attorney Docket No. RIN-001

Citizenship U. S.

Post Office Address: 1221 Salsipuedes Street
Santa Barbara, CA 93103

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Nicholas A.J. Millington

Title: SYSTEM AND METHOD FOR
SYNCHRONIZING OPERATIONS
AMONG A PLURALITY OF
INDEPENDENTLY CLOCKED DIGITAL
DATA PROCESSING DEVICES

Appl. No.: TBD

Filing Date: 04/16/2013

Examiner: TBD

Art Unit: TBD

Confirmation
Number:

PRELIMINARY AMENDMENT

Sir:

Please enter the following amendments prior to examination of the application.

Amendments to the Specification are presented on page 2;

Amendments to the Abstract are presented on page 3;

Amendments to the Claims are reflected in the listing of claims that begins on page 4 of this document.

Remarks/Arguments begin on page 8 of this document.

Amendments to the Specification:

On page 1 of the specification, please replace the first paragraph after the section heading “Incorporation By Reference” with the new paragraph as follows:

This application claims priority under 35 USC §120 of US Application No. 13/297,000, filed November 15, 2011, entitled “System And Method For Synchronizing Operations Among A Plurality Of Independently Clocked Digital Data Processing Devices”, which is a continuation of Patent Application Serial No. 10/816,217, now US Patent 8,234,395, filed on April 1, 2004, entitled "System and Method for Synchronizing Channel Handoff as Among a Plurality of Independently Clocked Digital Devices", which is a non-provisional application claiming priority under 35 USC §119 to Provisional Patent Application Serial No. 60/490,768, filed on July 28, 2003, entitled "Method for Synchronizing Audio Playback Between Multiple Networked Devices," now expired, all of which are assigned to the assignee of the present application and are incorporated herein by reference.

Please amend the application as follows:

Amendments to the Abstract:

Please replace the Abstract with the new abstract as follows:

In a network comprising a plurality of synchrony groups, including at least a first synchrony group and a second synchrony group, wherein a first zone player is a member of the first synchrony group and a second zone player is a member of the second synchrony group, a method comprising: transmitting, by the first zone player while the first zone player is playing a first audio information, a second audio information to the second zone player, wherein the second audio information is different from the first audio information; and playing, by the second synchrony group, the second audio information while the first synchrony group is playing the first audio information.

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-20 (cancelled)

21. (new) In a network comprising a plurality of synchrony groups, including at least a first synchrony group and a second synchrony group, wherein a first zone player is a member of the first synchrony group and a second zone player is a member of the second synchrony group, a method comprising:

transmitting, by the first zone player while the first zone player is playing a first audio information, a second audio information to the second zone player, wherein the second audio information is different from the first audio information; and

playing, by the second synchrony group, the second audio information while the first synchrony group is playing the first audio information.

22. (new) The method of claim 21 further comprising:

transmitting, by the first zone player, the second audio information to a third zone player, wherein the third zone player is a member of the second synchrony group.

23. (new) The method of claim 22, further comprising:

transmitting, by the first zone player, playback timing information to the second zone player and the third zone player.

24. (new) The method of claim 23, wherein playing, by the second synchrony group, the second audio information comprises:

utilizing, by the second zone player and the third zone player, the playback timing information to synchronously playback the second audio information.

25. (new) The method of claim 21, further comprising:

receiving, at the first zone player, the first audio information and playback timing information from a fourth zone player;

wherein the fourth zone player is a member of the first synchrony group.

26. (new) The method of claim 25 further comprising:
utilizing, at the first zone player, the playback timing information received from the fourth zone player to synchronously playback the first audio information with the fourth zone player.
27. (new) The method of claim 21, further comprising:
receiving, at the first zone player, the second audio information from an audio information source, wherein the audio information source is directly connected to the first zone player.
28. (new) The method of claim 27, wherein the second audio information is received at the first zone player as analog audio, and wherein the first zone player converts the second audio information from analog to digital prior to transmission to the second zone player.
29. (new) A system comprising a first synchrony group and a second synchrony group, wherein:
a first zone player is a member of the first synchrony group;
a second zone player is a member of the second synchrony group;
the first synchrony group is arranged to play a first audio information;
the first zone player is arranged to transmit a second audio information to the second zone player, wherein the second audio information is different from the first audio information; and
the second synchrony is arranged to play the second audio information while the first synchrony group is playing the first audio information.
30. (new) The system of claim 29 wherein the first zone player is further arranged to transmit the second audio information to a third zone player, wherein the third zone player is a member of the second synchrony group.

31. (new) The system of claim 30, wherein the first zone player is further arranged to transmit playback timing information to the second zone player and the third zone player.
32. (new) The system of claim 31, wherein the second zone player and the third zone player are arranged to utilize the playback timing information to synchronously playback the second audio information.
33. (new) The system of claim 9, further comprising a fourth zone player, wherein the fourth zone player is a member of the first synchrony group, and wherein the first zone player is further arranged to receive the first audio information and playback timing information from the fourth zone player.
34. (new) The method of claim 25 wherein the first zone player is arranged to utilize the playback timing information received from the fourth zone player to synchronously playback the first audio information with the fourth zone player.
35. (new) The system of claim 29, wherein the first zone player is further arranged to receive the second audio information from an audio information source, wherein the audio information source is directly connected to the first zone player.
36. (new) The system of claim 35, wherein the first zone player is arranged to receive the second audio information as analog audio, and wherein the first zone player is arranged to convert the second audio information from analog to digital prior to transmission to the second zone player.
37. (new) A tangible computer-readable memory having instructions stored thereon that when executed cause a first zone player to:
- join a first synchrony group with a second zone player;
 - receive a first audio information from the second zone player;
 - play the first audio information in synchrony with the second zone player; and

transmit a second audio information to a third zone player while playing the first audio information, wherein the second audio information is different from the first audio information; and,

wherein the third zone player is a member of a second synchrony group.

38. (new) The computer readable medium of claim 37, further comprising instructions that when executed cause the first zone player to:

transmit playback timing information to the third zone player.

39. (new) The computer readable medium of claim 37, further comprising instructions that when executed cause the first zone player to:

receive the second audio information from an audio information source, wherein the audio information source is directly connected to the first zone player.

40. (new) The computer readable medium of claim 37, further comprising instructions that when executed cause the first zone player to:

convert the second audio information from analog to digital prior to transmitting the second audio information to the third zone player.

REMARKS

Applicants request that the above-presented amendments are entered prior to examination. The Examiner is invited to contact the undersigned if it would be helpful to advance the prosecution of the application.

Respectfully submitted,

Date: April 16, 2013

By: /Robert J. Irvine III/

TechLaw LLP
Telephone: (312) 834-4334

Robert J. Irvine III
Attorney for Applicant
Registration No. 41,865

Electronic Patent Application Fee Transmittal				
Application Number:				
Filing Date:				
Title of Invention:		SYSTEM AND METHOD FOR SYNCHRONIZING OPERATIONS AMONG A PLURALITY OF INDEPENDENTLY CLOCKED DIGITAL DATA PROCESSING DEVICES		
First Named Inventor/Applicant Name:		Nicholas A.J. Millington		
Filer:		Robert Johnson Irvine		
Attorney Docket Number:		SNS-71301.US.11 / 13-0402		
Filed as Large Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Utility application filing	1011	1	280	280
Utility Search Fee	1111	1	600	600
Utility Examination Fee	1311	1	720	720
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				1600

Electronic Acknowledgement Receipt	
EFS ID:	15536234
Application Number:	13864249
International Application Number:	
Confirmation Number:	9806
Title of Invention:	SYSTEM AND METHOD FOR SYNCHRONIZING OPERATIONS AMONG A PLURALITY OF INDEPENDENTLY CLOCKED DIGITAL DATA PROCESSING DEVICES
First Named Inventor/Applicant Name:	Nicholas A.J. Millington
Customer Number:	109473
Filer:	Robert Johnson Irvine
Filer Authorized By:	
Attorney Docket Number:	SNS-71301.US.11 / 13-0402
Receipt Date:	17-APR-2013
Filing Date:	
Time Stamp:	02:04:23
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	\$ 1600
RAM confirmation Number	6995
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	----------------------------------	------------------	------------------

1	Application Data Sheet	ADS-71301_US_11.pdf	1064872	no	6
			ab1bbba016366df1c207287137620ce9e6748514		
Warnings:					
Information:					
2		71301_US-Specification.pdf	3285365	yes	67
			f36f9701ab8cd3989b81f0a21d17040f52bc7bca		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Specification		1	58	
	Claims		59	61	
	Abstract		62	62	
	Drawings-only black and white line drawings		63	67	
Warnings:					
Information:					
3	Oath or Declaration filed	13297000-Dec.pdf	52966	no	2
			b421f9358bd38188761d3667976f972dcdfda526		
Warnings:					
Information:					
4		71301US11_PA.pdf	90052	yes	8
			39bd991e5b3e1ce007891dcae394b4d46e38d0a		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Preliminary Amendment		1	1	
	Specification		2	2	
	Abstract		3	3	
	Claims		4	7	
Applicant Arguments/Remarks Made in an Amendment		8	8		
Warnings:					
Information:					

5	Fee Worksheet (SB06)	fee-info.pdf	33254 6032ac5c85f07e833d924083c952b561e2c851aa	no	2
Warnings:					
Information:					
Total Files Size (in bytes):				4526509	
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	SNS-71301.US.11
		Application Number	
Title of Invention	SYSTEM AND METHOD FOR SYNCHRONIZING OPERATIONS AMONG A PLURALITY OF INDEPENDENTLY CLOCKED DIGITAL DATA PROCESSING DEVICES		
The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76. This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.			

Secrecy Order 37 CFR 5.2

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)
--

Inventor Information:

Inventor 1					Remove		
Legal Name							
Prefix	Given Name		Middle Name		Family Name		Suffix
Mr.	Nicholas		A. J.		Millington		
Residence Information (Select One) ☒ US Residency ☐ Non US Residency ☐ Active US Military Service							
City	Santa Barbara		State/Province	CA	Country of Residence i	US	
Mailing Address of Inventor:							
Address 1		223 E. De La Guerra					
Address 2							
City	Santa Barbara			State/Province	CA		
Postal Code	93101			Country i	US		
All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the Add button.							
Add							

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below. For further information see 37 CFR 1.33(a).			
☐ An Address is being provided for the correspondence information of this application.			
Customer Number	109473		
Email Address	bob@techlawllp.com		<input type="button" value="Add Email"/> <input type="button" value="Remove Email"/>

Application Information:

Title of the Invention	SYSTEM AND METHOD FOR SYNCHRONIZING OPERATIONS AMONG A PLURALITY OF INDEPENDENTLY CLOCKED DIGITAL DATA PROCESSING DEVICES		
Attorney Docket Number	SNS-71301.US.11	Small Entity Status Claimed ☐	
Application Type	Nonprovisional		
Subject Matter	Utility		
Suggested Class (if any)		Sub Class (if any)	
Suggested Technology Center (if any)			
Total Number of Drawing Sheets (if any)	5	Suggested Figure for Publication (if any)	

EFS Web 2.2.3

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	SNS-71301.US.11
		Application Number	
Title of Invention	SYSTEM AND METHOD FOR SYNCHRONIZING OPERATIONS AMONG A PLURALITY OF INDEPENDENTLY CLOCKED DIGITAL DATA PROCESSING DEVICES		

Publication Information:

☐	Request Early Publication (Fee required at time of Request 37 CFR 1.219)
☐	Request Not to Publish. I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application has not and will not be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.			
Please Select One:			
☒	Customer Number	☐	US Patent Practitioner
☐	Limited Recognition (37 CFR 11.9)		
Customer Number	109473		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.					
Prior Application Status	Pending		Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)		
	Continuation of	13297000	2011-11-15		
Prior Application Status	Pending		Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)		
13297000	Continuation of	10816217	2004-04-01		
Prior Application Status	Patented		Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
10816217	non provisional of	60490768	2003-07-28	8234395	2012-07-31
Prior Application Status	Expired		Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)		
60490768					
Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the Add button.					

Foreign Priority Information:

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	SNS-71301.US.11
		Application Number	
Title of Invention	SYSTEM AND METHOD FOR SYNCHRONIZING OPERATIONS AMONG A PLURALITY OF INDEPENDENTLY CLOCKED DIGITAL DATA PROCESSING DEVICES		
<p>This section allows for the applicant to claim benefit of foreign priority and to identify any prior foreign application for which priority is not claimed. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(a).</p>			
			<input type="button" value="Remove"/>
Application Number	Country i	Filing Date (YYYY-MM-DD)	Priority Claimed
			☐ Yes ☒ No
Additional Foreign Priority Data may be generated within this form by selecting the Add button.			<input type="button" value="Add"/>

Authorization to Permit Access:

☐ Authorization to Permit Access to the Instant Application by the Participating Offices
<p>If checked, the undersigned hereby grants the USPTO authority to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the World Intellectual Property Office (WIPO), and any other intellectual property offices in which a foreign application claiming priority to the instant patent application is filed access to the instant patent application. See 37 CFR 1.14(c) and (h). This box should not be checked if the applicant does not wish the EPO, JPO, KIPO, WIPO, or other intellectual property office in which a foreign application claiming priority to the instant patent application is filed to have access to the instant patent application.</p> <p>In accordance with 37 CFR 1.14(h)(3), access will be provided to a copy of the instant patent application with respect to: 1) the instant patent application-as-filed; 2) any foreign application to which the instant patent application claims priority under 35 U.S.C. 119(a)-(d) if a copy of the foreign application that satisfies the certified copy requirement of 37 CFR 1.55 has been filed in the instant patent application; and 3) any U.S. application-as-filed from which benefit is sought in the instant patent application.</p> <p>In accordance with 37 CFR 1.14(c), access may be provided to information concerning the date of filing this Authorization.</p>

Applicant Information:

<p>Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.</p>
--

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	SNS-71301.US.11
		Application Number	
Title of Invention	SYSTEM AND METHOD FOR SYNCHRONIZING OPERATIONS AMONG A PLURALITY OF INDEPENDENTLY CLOCKED DIGITAL DATA PROCESSING DEVICES		

Applicant 1

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Remove

☒ Assignee	☐ Legal Representative under 35 U.S.C. 117
☐ Person to whom the inventor is obligated to assign.	☐ Person who shows sufficient proprietary interest

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

--	--

Name of the Deceased or Legally Incapacitated Inventor : If the Assignee is an Organization check here. ☒

Organization Name	SONOS, INC.
-------------------	-------------

Mailing Address Information:

Address 1	223 E De La Guerra St.		
Address 2			
City	Santa Barbara	State/Province	CA
Country ⁱ	US	Postal Code	93101
Phone Number		Fax Number	
Email Address			

Additional Applicant Data may be generated within this form by selecting the Add button.

Add

Signature:

Remove

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications

Signature	/Robert J. Irvine III/		Date (YYYY-MM-DD)	2013-04-16
First Name	Robert	Last Name	Irvine	Registration Number
			41865	

Additional Signature may be generated within this form by selecting the Add button.

Add

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	SNS-71301.US.11
		Application Number	
Title of Invention	SYSTEM AND METHOD FOR SYNCHRONIZING OPERATIONS AMONG A PLURALITY OF INDEPENDENTLY CLOCKED DIGITAL DATA PROCESSING DEVICES		

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ATTORNEY'S DOCKET NO. 11-0901

PATENTS

UNITED STATES PATENT APPLICATION

OF

NICHOLAS A. J. MILLINGTON

FOR

SYSTEM AND METHOD FOR SYNCHRONIZING OPERATIONS AMONG A PLURALITY OF

INDEPENDENTLY CLOCKED DIGITAL DATA PROCESSING DEVICES

INCORPORATION BY REFERENCE

This application claims the benefit of Provisional Patent Application Serial No. 60/490,768, filed on July 28, 2003, entitled "Method for Synchronizing Audio Playback Between Multiple Networked Devices," and Non-provisional Patent Application Serial No. 10/816,217, filed on April 1, 2004, entitled "System and Method for Synchronizing Channel Handoff as Among a Plurality of Independently Clocked Digital Devices," both of which are assigned to the assignee of the present application and are incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates generally to the field of digital data processing devices, and more particularly to systems and methods for synchronizing operations among a plurality of independently-clocked digital data processing devices. The invention is embodied in a system for synchronizing operations among a plurality of devices, in relation to information that is provided by a common source. One embodiment of the invention enables synchronizing of audio playback as among two or more audio playback devices that receive audio information from a common information source, or channel.

More generally, the invention relates to the field of arrangements that synchronize output generated by a number of output generators, including audio output, video output, combinations of audio and video, as well as other types of output as will be appreciated by those skilled in the art, provided by a common channel. Generally, the invention will find utility in connection with any type of information for which synchrony among independently-clocked devices is desired.

BACKGROUND OF THE INVENTION

There are a number of circumstances under which it is desirable to maintain synchrony of operations among a plurality of independently-clocked digital data processing devices in relation to, for example, information that is provided thereto by a common source. For example, systems are

1 being developed in which one audio information source can distribute audio information in digital
2 form to a number of audio playback devices for playback. The audio playback devices receive the
3 digital information and convert it to analog form for playback. The audio playback devices may be
4 located in the same room or they may be distributed in different rooms in a residence such as a house
5 or an apartment, in different offices in an office building, or the like. For example, in a system
6 installed in a residence, one audio playback device may be located in a living room, while another
7 audio playback device is be located in a kitchen, and yet other audio playback devices may be
8 located in various bedrooms of a house. In such an arrangement, the audio information that is
9 distributed to various audio playback devices may relate to the same audio program, or the
10 information may relate to different audio programs. If the audio information source provides audio
11 information relating to the same audio program to two or more audio playback devices at the same
12 time, the audio playback devices will generally contemporaneously play the same program. For
13 example, if the audio information source provides audio information to audio playback devices
14 located in the living room and kitchen in a house at the same time, they will generally
15 contemporaneously play the same program.

16 One problem that can arise is to ensure that, if two or more audio playback devices are
17 contemporaneously attempting to play back the same audio program, they do so simultaneously.
18 Small differences in the audio playback devices' start times and/or playback speeds can be perceived
19 by a listener as an echo effect, and larger differences can be very annoying. Differences can arise
20 because for a number of reasons, including delays in the transfer of audio information over the
21 network. Such delays can differ as among the various audio playback devices for a variety of
22 reasons, including where they are connected into the network, message traffic and other reasons as
23 will be apparent to those skilled in the art.

24 Another problem arises from the following. When an audio playback device converts the
25 digital audio information from digital to analog form, it does so using a clock that provides timing
26 information. Generally, the audio playback devices that are being developed have independent
27 clocks, and, if they are not clocking at precisely the same rate, the audio playback provided by the
28 various devices can get out of synchronization.

SUMMARY OF THE INVENTION

The invention provides a new and improved system and method for synchronizing operations among a number of digital data processing devices that are regulated by independent clocking devices. Generally, the invention will find utility in connection with any type of information for which synchrony among devices connected to a network is desired. The invention is described in connection with a plurality of audio playback devices that receive digital audio information that is to be played back in synchrony, but it will be appreciated that the invention can find usefulness in connection with any kind of information for which coordination among devices that have independent clocking devices would find utility.

In brief summary, the invention provides, in one aspect, a system for maintaining synchrony of operations among a plurality of devices that have independent clocking arrangements. The system includes a task distribution device that distributes tasks to a synchrony group comprising a plurality of devices that are to perform the tasks distributed by the task distribution device in synchrony. The task distribution device distributes each task to the members of the synchrony group over a network. Each task is associated with a time stamp that indicates a time, relative to a clock maintained by the task distribution device, at which the members of the synchrony group are to execute the task. Each member of the synchrony group periodically obtains from the task distribution device an indication of the current time indicated by its clock, determines a time differential between the task distribution device's clock and its respective clock and determines therefrom a time at which, according to its respective clock, the time stamp indicates that it is to execute the task.

In one embodiment, the tasks that are distributed include audio information for an audio track that is to be played by all of the devices comprising the synchrony group synchronously. The audio track is divided into a series of frames, each of which is associated with a time stamp indicating the time, relative to the clock maintained by an audio information channel device, which, in that embodiment, serves as the task distribution device, at which the members of the synchrony group are to play the respective frame. Each member of the synchrony group, using a very accurate

1 protocol, periodically obtains the time indicated by the audio information channel device, and
2 determines a differential between the time as indicated by its local clock and the audio information
3 channel device's clock. The member uses the differential and the time as indicated by the time stamp
4 to determine the time, relative to its local clock, at which it is to play the respective frame. The
5 members of the synchrony group do this for all of the frames, and accordingly are able to play the
6 frames in synchrony.

7 **BRIEF DESCRIPTION OF THE DRAWINGS**

8 This invention is pointed out with particularity in the appended claims. The above and
9 further advantages of this invention may be better understood by referring to the following
10 description taken in conjunction with the accompanying drawings, in which:

11 FIG. 1 schematically depicts an illustrative networked audio system, constructed in
12 accordance with the invention;

13 FIG. 2 schematically depicts a functional block diagram of a synchrony group utilizing a
14 plurality of zone players formed within the networked audio system depicted in FIG. 1;

15 FIG. 2A schematically depicts two synchrony groups, illustrating how a member of one
16 synchrony group can provide audio information to the members of another synchrony group;

17 FIG. 3 depicts an functional block diagram of a zone player for use in the networked audio
18 system depicted in FIG. 1; and

19 FIG. 4 is useful in understanding an digital audio information framing methodology useful
20 in the network audio system depicted in FIG. 1.

DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

FIG. 1 depicts an illustrative network audio system 10 constructed in accordance with the invention. With reference to FIG. 1, the network audio system 10 includes a plurality of zone players 11(1) through 11(N) (generally identified by reference numeral 11(n)) interconnected by a local network 12, all of which operate under control of one or more user interface modules generally identified by reference numeral 13. One or more of the zone players 11(n) may also be connected to one or more audio information sources, which will generally be identified herein by reference numeral 14(n)(s), and/or one or more audio reproduction devices, which will generally be identified by reference numeral 15(n)(r). In the reference numeral 14(n)(s), index "n" refers to the index "n" of the zone player 11(n) to which the audio information source is connected, and the index "s" ($s=1, \dots, S_n$) refers to the "s-th" audio information source connected to that "n-th" zone player 11(n). Thus, if, for example, a zone player 11(n) is connected to four audio information sources 14(n)(1) through 14(n)(4), the audio information sources may be generally identified by reference numeral 14(n)(s), with $S_n=4$. It will be appreciated that the number of audio information sources S_n may vary as among the various zone players 11(n), and some zone players may not have any audio information sources connected thereto. Similarly, in the reference numeral 15(n)(r), index "n" refers to the index "n" of the zone player 11(n) to which the audio reproduction device is connected, and the index "r" ($r=1, \dots, R_n$) refers to the "r-th" audio information source connected to that "n-th" zone player 11(n). In addition to the audio information sources 14(n)(s), the network audio system 10 may include one or more audio information sources 16(1) through 16(M) connected through appropriate network interface devices (not separately shown) to the local network 12. Furthermore, the local network may include one or more network interface devices (also not separately shown) that are configured to connect the local network 12 to other networks, including a wide area network such as the Internet, the public switched telephony network (PSTN) or other networks as will be apparent to those skilled in the art, over which connections to audio information sources may be established.

1 The zone players 11(n) associated with system 10 may be distributed throughout an
2 establishment such as residence, an office complex, a hotel, a conference hall, an amphitheater or
3 auditorium, or other types of establishments as will be apparent to those skilled in the art or the like.
4 For example, if the zone players 11(n) and their associated audio information source(s) and/or audio
5 reproduction device(s) are distributed throughout a residence, one, such as zone player 11(1) and its
6 associated audio information source(s) and audio reproduction device(s) may be located in a living
7 room, another may be located in a kitchen, another may be located in a dining room, and yet others
8 may be located in respective bedrooms, to selectively provide entertainment in those rooms. On the
9 other hand, if the zone players 11(n) and their associated audio information source(s) and/or audio
10 reproduction device(s) are distributed throughout an office complex, one may, for example, be
11 provided in each office to selectively provide entertainment to the employees in the respective
12 offices. Similarly, if the zone players 11(n) and associated audio information source(s) and/or audio
13 reproduction device(s) are used in a hotel, they may be distributed throughout the rooms to provide
14 entertainment to the guests. Similar arrangements may be used with zone players 11(n) and
15 associated audio information source(s) and/or audio reproduction device(s) used in an amphitheater
16 or auditorium. Other arrangements in other types of environments will be apparent to those skilled
17 in the art. In each case, the zone players 11(n) can be used to selectively provide entertainment in
18 the respective locations, as will be described below.

19 The audio information sources 14(n)(s) and 16(m) may be any of a number of types of
20 conventional sources of audio information, including, for example, compact disc ("CD") players,
21 AM and/or FM radio receivers, analog or digital tape cassette players, analog record turntables and
22 the like. In addition, the audio information sources 14(n)(s) and 16(m) may comprise digital audio
23 files stored locally on, for example, personal computers (PCs), personal digital assistants (PDAs),
24 or similar devices capable of storing digital information in volatile or non-volatile form. As noted
25 above, the local network 12 may also have an interface (not shown) to a wide area network, over
26 which the network audio system 10 can obtain audio information. Moreover, one or more of the
27 audio information sources 14(n)(s) may also comprise an interface to a wide area network such as
28 the Internet, the public switched telephony network (PSTN) or any other source of audio

1 information. In addition, one or more of the audio information sources 14(n)(s) and 16(m) may
 2 comprise interfaces to radio services delivered over, for example, satellite. Audio information
 3 obtained over the wide area network may comprise, for example, streaming digital audio information
 4 such as Internet radio, digital audio files stored on servers, and other types of audio information and
 5 sources as will be appreciated by those skilled in the art. Other arrangements and other types of
 6 audio information sources will be apparent to those skilled in the art.

7 Generally, the audio information sources 14(n)(s) and 16(m) provide audio information
 8 associated with audio programs to the zone players for playback. A zone player that receives audio
 9 information from an audio information source 14(n)(s) that is connected thereto can provide
 10 playback and/or forward the audio information, along with playback timing information, over the
 11 local network 12 to other zone players for playback. Similarly, each audio information source
 12 16(m) that is not directly connected to a zone player can transmit audio information over the network
 13 12 to any zone player 11(n) for playback. In addition, as will be explained in detail below, the
 14 respective zone player 11(n) can transmit the audio information that it receives either from an audio
 15 information source 14(n)(s) connected thereto, or from an audio information source 16(m), to
 16 selected ones of the other zone players 11(n'), 11(n''),... (n not equal to n', n'',...) for playback by
 17 those other zone players. The other zone players 11(n'), 11(n''),... to which the zone player 11(n)
 18 transmits the audio information for playback may be selected by a user using the user interface
 19 module 13. In that operation, the zone player 11(n) will transmit the audio information to the
 20 selected zone players 11(n'), 11(n''),... over the network 12. As will be described below in greater
 21 detail, the zone players 11(n), 11(n'), 11(n''),... operate such that the zone players 11(n'), 11(n''),...
 22 synchronize their playback of the audio program with the playback by the zone player 11(n), so that
 23 the zone players 11(n), 11(n'), 11(n'') provide the same audio program at the same time.

24 Users, using user interface module 13, may also enable different groupings or sets of zone
 25 players to provide audio playback of different audio programs synchronously. For example, a user,
 26 using a user interface module 13, may enable zone players 11(1) and 11(2) to play one audio
 27 program, audio information for which may be provided by, for example, one audio information
 28 source 14(1)(1). The same or a different user may, using the same or a different user interface

1 module 13, enable zone players 11(4) and 11(5) to contemporaneously play another audio program,
2 audio information for which may be provided by a second audio information source, such as audio
3 information source 14(5)(2). Further, a user may enable zone player 11(3) to contemporaneously
4 play yet another audio program, audio information for which may be provided by yet another audio
5 information source, such as audio information source 16(1). As yet another possibility, a user may
6 contemporaneously enable zone player 11(1) to provide audio information from an audio
7 information source connected thereto, such as audio information source 14(1)(2), to another zone
8 player, such as zone player 11(6) for playback.

9 In the following, the term "synchrony group" will be used to refer to a set of one or more
10 zone players that are to play the same audio program synchronously. Thus, in the above example,
11 zone players 11(1) and 11(2) comprise one synchrony group, zone player 11(3) comprises a second
12 synchrony group, zone players 11(4) and 11(5) comprise a third synchrony group, and zone player
13 11(6) comprises yet a fourth synchrony group. Thus, while zone players 11(1) and 11(2) are playing
14 the same audio program, they will play the audio program synchronously. Similarly, while zone
15 players 11(4) and 11(5) are playing the same audio program, they will play the audio program
16 synchronously. On the other hand, zone players that are playing different audio programs may do
17 so with unrelated timings. That is, for example, the timing with which zone players 11(1) and 11(2)
18 play their audio program may have no relationship to the timing with which zone player 11(3), zone
19 players 11(4) and 11(5), and zone player 11(6) play their audio programs. It will be appreciated that,
20 since "synchrony group" is used to refer to sets of zone players that are playing the same audio
21 program synchronously, zone player 11(1) will not be part of zone player 11(6)'s synchrony group,
22 even though zone player 11(1) is providing the audio information for the audio program to zone
23 player 11(6).

24 In the network audio system 10, the synchrony groups are not fixed. Users can enable them
25 to be established and modified dynamically. Continuing with the above example, a user may enable
26 the zone player 11(1) to begin providing playback of the audio program provided thereto by audio
27 information source 14(1)(1), and subsequently enable zone player 11(2) to join the synchrony group.
28 Similarly, a user may enable the zone player 11(5) to begin providing playback of the audio program

1 provided thereto by audio information source 14(5)(2), and subsequently enable zone player 11(4)
2 to join that synchrony group. In addition, a user may enable a zone player to leave a synchrony
3 group and possibly join another synchrony group. For example, a user may enable the zone player
4 11(2) to leave the synchrony group with zone player 11(1), and join the synchrony group with zone
5 player 11(6). As another possibility, the user may enable the zone player 11(1) to leave the
6 synchrony group with zone player 11(2) and join the synchrony group with zone player 11(6). In
7 connection with the last possibility, the zone player 11(1) can continue providing audio information
8 from the audio information source 14(1)(1) to the zone player 11(2) for playback thereby.

9 A user, using the user interface module 13, can enable a zone player 11(n) that is currently
10 not a member of a synchrony group to join a synchrony group, after which it will be enabled to play
11 the audio program that is currently being played by that synchrony group. Similarly, a user, also
12 using the user interface module 13, can enable a zone player 11(n) that is currently a member of one
13 synchrony group, to disengage from that synchrony group and join another synchrony group, after
14 which that zone player will be playing the audio program associated with the other synchrony group.
15 For example, if a zone player 11(6) is currently not a member of any synchrony group, it, under
16 control of the user interface module 13, can become a member of a synchrony group, after which it
17 will play the audio program being played by the other members of the synchrony group, in
18 synchrony with the other members of the synchrony group. In becoming a member of the synchrony
19 group, zone player 11(6) can notify the zone player that is the master device for the synchrony group
20 that it wishes to become a member of its synchrony group, after which that zone player will also
21 transmit audio information associated with the audio program, as well as timing information, to the
22 zone player 11(6). As the zone player 11(6) receives the audio information and the timing
23 information from the master device, it will play the audio information with the timing indicated by
24 the timing information, which will enable the zone player 11(6) to play the audio program in
25 synchrony with the other zone player(s) in the synchrony group.

26 Similarly, if a user, using the user interface module 13, enables a zone player 11(n)
27 associated with a synchrony group to disengage from that synchrony group, and if the zone player
28 11(n) is not the master device of the synchrony group, the zone player 11(n) can notify the master

1 device, after which the master device can terminate transmission of the audio information and timing
 2 information to the zone player 11(n). If the user also enables the zone player 11(n) to begin playing
 3 another audio program using audio information from an audio information source 14(n)(s) connected
 4 thereto, it will acquire the audio information from the audio information source 14(n)(s) and initiate
 5 playback thereof. If the user enables another zone player 11(n') to join the synchrony group
 6 associated with zone player 11(n), operations in connection therewith can proceed as described
 7 immediately above.

8 As yet another possibility, if a user, using the user interface module 13, enables a zone player
 9 11(n) associated with a synchrony group to disengage from that synchrony group and join another
 10 synchrony group, and if the zone player is not the master device of the synchrony group from which
 11 it is disengaging, the zone player 11(n) can notify the master device of the synchrony group from
 12 which it is disengaging, after which that zone player will terminate transmission of audio
 13 information and timing information to the zone player 11(n) that is disengaging.
 14 Contemporaneously, the zone player 11(n) can notify the master device of the synchrony group that
 15 it (that is, zone player 11(n)) is joining, after which the master device can begin transmission of
 16 audio information and timing information to that zone player 11(n). The zone player 11(n) can
 17 thereafter begin playback of the audio program defined by the audio information, in accordance with
 18 the timing information so that the zone player 11(n) will play the audio program in synchrony with
 19 the master device.

20 As yet another possibility, a user, using the user interface module 13, may enable a zone
 21 player 11(n) that is not associated with a synchrony group, to begin playing an audio program using
 22 audio information provided to it by an audio information source 14(n)(s) connected thereto. In that
 23 case, the user, also using the user interface module 13 or a user interface device that is specific to
 24 the audio information source 14(n)(s), can enable the audio information source 14(n)(s) to provide
 25 audio information to the zone player 11(n). After the zone player 11(n) has begun playback, or
 26 contemporaneously therewith, the user, using the user interface module 13, can enable other zone
 27 players 11(n'), 11(n''),... to join zone player 11(n)'s synchrony group and enable that zone player

1 11(n) to transmit audio information and timing information thereto as described above, to facilitate
2 synchronous playback of the audio program by the other zone players 11(n'), 11(n'')....

3 A user can use the user interface module 13 to control other aspects of the network audio
4 system 10, including but not limited to the selection of the audio information source 14(n)(s) that
5 a particular zone player 11(n) is to utilize, the volume of the audio playback, and so forth. In
6 addition, a user may use the user interface module 13 to turn audio information source(s) 14(n)(s)
7 on and off and to enable them to provide audio information to the respective zone players 11(n).

8 Operations performed by the various devices associated with a synchrony group will be
9 described in connection with FIG. 2, which schematically depicts a functional block diagram of a
10 synchrony group in the network audio system 10 described above in connection with FIG. 1. With
11 reference to FIG. 2, a synchrony group 20 includes a master device 21 and zero or more slave
12 devices 22(1) through 22 (G) (generally identified by reference numeral 22(g)), all of which
13 synchronously play an audio program provided by an audio information channel device 23. Each
14 of the master device 21, slave devices 22(g) and audio information channel device 23 utilizes a zone
15 player 11(n) depicted in FIG. 1, although it will be clear from the description below that a zone
16 player may be utilized both for the audio information channel device for the synchrony group 20,
17 and the master device 21 or a slave device 22(g) of the synchrony group 20. As will be described
18 below in more detail, the audio information channel device 23 obtains the audio information for the
19 audio program from an audio information source, adds playback timing information, and transmits
20 the combined audio and playback timing information to the master device 21 and slave devices 22(g)
21 over the network 12 for playback. The playback timing information that is provided with the audio
22 information, together with clock timing information provided by the audio information channel
23 device 23 to the various devices 21 and 22(g) as will be described below, enables the master device
24 21 and slave devices 22(g) of the synchrony group 20 to play the audio information simultaneously.

25 The master device 21 and the slave devices 22(g) receive the audio and playback timing
26 information, as well as the clock timing information, that are provided by the audio information
27 channel device 23, and play back the audio program defined by the audio information. The master

1 device 21 is also the member of the synchrony group 20 that communicates with the user interface
2 module 13 and that controls the operations of the slave devices 22(g) in the synchrony group 20.
3 In addition, the master device 21 controls the operations of the audio information channel device 23
4 that provides the audio and playback timing information for the synchrony group 20. Generally, the
5 initial master device 21 for the synchrony group will be the first zone player 11(n) that a user wishes
6 to play an audio program. However, as will be described below, the zone player 11(n) that operates
7 as the master device 21 can be migrated from one zone player 11(n) to another zone player 11(n'),
8 which preferably will be a zone player that is currently operating as a slave device 22(g) in the
9 synchrony group.

10 In addition, under certain circumstances, as will be described below, the zone player 11(n)
11 that operates as the audio information channel device 23 can be migrated from one zone player to
12 another zone player, which also will preferably will be a zone player that is currently operating as
13 a member of the synchrony group 20. It will be appreciated that the zone player that operates as the
14 master device 21 can be migrated to another zone player independently of the migration of the audio
15 information channel device 23. For example, if one zone player 11(n) is operating as both the master
16 device 21 and the audio information channel device 23 for a synchrony group 20, the master device
17 21 can be migrated to another zone player 11(n') while the zone player 11(n) is still operating as the
18 audio information channel device 23. Similarly, if one zone player 11(n) is operating as both the
19 master device 21 and the audio information channel device 23 for a synchrony group 20, the audio
20 information channel device 23 can be migrated to another zone player 11(n') while the zone player
21 11(n) is still operating as the master device 21. In addition, if one zone player 11(n) is operating as
22 both the master device 21 and the audio information channel device 23 for a synchrony group 20,
23 the master device 21 can be migrated to another zone player 11(n') and the audio information
24 channel device can be migrated to a third zone player 11(n").

25 The master device 21 receives control information from the user interface module 13 for
26 controlling the synchrony group 20 and provides status information indicating the operational status
27 of the synchrony group to the user interface module 13. Generally, the control information from the
28 user interface module 13 enables the master device 21 to, in turn, enable the audio information

1 channel device 23 to provide audio and playback timing information to the synchrony group to
2 enable the devices 21 and 22(g) that are members of the synchrony group 20 to play the audio
3 program synchronously. In addition, the control information from the user interface module 13
4 enables the master device 21 to, in turn, enable other zone players to join the synchrony group as
5 slave devices 22(g) and to enable slave devices 22(g) to disengage from the synchrony group.
6 Control information from the user interface module 13 can also enable the zone player 11(n) that is
7 currently operating as the master device 21 to disengage from the synchrony group, but prior to
8 doing so that zone player will enable the master device 21 to transfer from that zone player 11(n) to
9 another zone player 11(n'), preferably to a zone player 11(n') that is currently a slave device 22(g) in
10 the synchrony group 20. The control information from the user interface module 13 can also enable
11 the master device 21 to adjust its playback volume and to enable individual ones of the various slave
12 devices 22(g) to adjust their playback volumes. In addition, the control information from the user
13 interface module 13 can enable the synchrony group 20 to terminate playing of a current track of the
14 audio program and skip to the next track, and to re-order tracks in a play list of tracks defining the
15 audio program that is to be played by the synchrony group 20.

16 The status information that the master device 21 may provide to the user interface module
17 13 can include such information as a name or other identifier for the track of the audio work that is
18 currently being played, the names or other identifiers for upcoming tracks, the identifier of the zone
19 player 11(n) that is currently operating as the master device 21, and identifiers of the zone players
20 that are currently operating as slave devices 22(g). In one embodiment, the user interface module
21 13 includes a display (not separately shown) that can display the status information to the user.

22 It will be appreciated that the zone player 11(n) that is operating as the audio information
23 channel device 23 for one synchrony group may also comprise the master device 21 or any of the
24 slave devices 22(g) in another synchrony group. This may occur if, for example, the audio
25 information source that is to provide the audio information that is to be played by the one synchrony
26 group is connected to a zone player also being utilized as the master device or a slave device for the
27 other synchrony group. This will be schematically illustrated below in connection with FIG. 2A.
28 Since, as noted above, the zone player 11(n) that is operating as the audio information channel

1 device 23 for the synchrony group 20 may also be operating as a master device 21 or slave device
2 22(g) for another synchrony group, it can also be connected to one or more audio reproduction
3 devices 15(n)(r), although that is not depicted in FIG. 2. Since the master device 21 and slave
4 devices 22(g) are all to provide playback of the audio program, they will be connected to respective
5 audio reproduction devices 15(n)(r). Furthermore, it will be appreciated that one or more of the
6 zone players 11(n) that operate as the master device 21 and slave devices 22(g) in synchrony group
7 20 may also operate as an audio information channel device for that synchrony group or for another
8 synchrony group and so they may be connected to one or more audio information sources 14(n)(s),
9 although that is also not depicted in FIG. 2. In addition, it will be appreciated that a zone player
10 11(n) can also operate as a audio information channel device 23 for multiple synchrony groups.

11 If the audio information channel device 23 does not utilize the same zone player as the master
12 device 21, the master device 21 controls the audio information channel device by exchanging control
13 information over the network 12 with the audio information channel device 23. The control
14 information is represented in FIG. 2 by the arrow labeled CHAN_DEV_CTRL_INFO. The control
15 information that the master device 21 provides to the audio information channel device 23 will
16 generally depend on the nature of the audio information source that is to provide the audio
17 information for the audio program that is to be played and the operation to be enabled by the control
18 information. If, for example, the audio information source is a conventional compact disc, tape, or
19 record player, broadcast radio receiver, or the like, which is connected to a zone player 11(n), the
20 master device 21 may merely enable the zone player serving as the audio information channel device
21 23 to receive the audio information for the program from the audio information source. It will be
22 appreciated that, if the audio information is not in digital form, the audio information channel device
23 23 will convert it to digital form and provide the digitized audio information, along with the
24 playback timing information, to the master device 21 and slave devices 22(g).

25 On the other hand, if the audio information source is, for example, a digital data storage
26 device, such as may be on a personal computer or similar device, the master device 21 can provide
27 a play list to the audio information channel device 23 that identifies one or more files containing the
28 audio information for the audio program. In that case, the audio information channel device 23 can

1 retrieve the files from the digital data storage device and provide them, along with the playback
2 timing information, to the master device 21 and the slave devices 22(g). It will be appreciated that,
3 in this case, the audio information source may be directly connected to the audio information channel
4 device 23, as, for example, an audio information source 14(n)(s), or it may comprise an audio
5 information source 16(m) connected to the network 12. As a further alternative, if the audio
6 information source is a source available over the wide area network, the master device 21 can
7 provide a play list comprising a list of web addresses identifying the files containing the audio
8 information for the audio program that is to be played, and in that connection the audio information
9 channel device 23 can initiate a retrieval of the files over the wide area network. As yet another
10 alternative, if the audio information source is a source of streaming audio received over the wide area
11 network, the master device 21 can provide a network address from which the streaming audio can
12 be received. Other arrangements by which the master device 21 can control the audio information
13 channel device 23 will be apparent to those skilled in the art.

14 The master device 21 can also provide control information to the synchrony group's audio
15 information channel device 23 to enable a migration from one zone player 11(n) to another zone
16 player 11(n'). This may occur if, for example, the audio information source is one of audio
17 information sources 16 or a source accessible over the wide area network via the network 12. The
18 master device 21 can enable migration of the audio information channel device 23 for several
19 reasons, including, for example, to reduce the loading of the zone player 11(n), to improve latency
20 of message transmission in the network 12, and other reasons as will be appreciated by those skilled
21 in the art.

22 As noted above, the audio information channel device 23 provides audio and playback timing
23 information for the synchrony group to enable the master device 21 and slave devices 22(g) to play
24 the audio program synchronously. Details of the audio and playback timing information will be
25 described in detail below in connection with FIGS. 3 and 4, but, in brief, the audio information
26 channel device 23 transmits the audio and playback timing information in messages over the network
27 12 using a multi-cast message transmission methodology. In that methodology, the audio
28 information channel device 23 will transmit the audio and playback timing information in a series

1 of messages, with each message being received by all of the zone players 11(n) comprising the
 2 synchrony group 20, that is, by the master device 21 and the slave devices 22(g). Each of the
 3 messages includes a multi-cast address, which the master device 21 and slave devices 22(g) will
 4 monitor and, when they detect a message with that address, they will receive and use the contents
 5 of the message. The audio and playback timing information is represented in FIG. 2 by the arrow
 6 labeled "AUD+PBTIME_INF0," which has a single tail, representing a source for the information
 7 at the audio information channel device 23, and multiple arrowheads representing the destinations
 8 of the information, with one arrowhead extending to the master device 21 and other arrowheads
 9 extending to each of the slave devices 22(g) in the synchrony group 20. The audio information
 10 channel device 23 may make use of any convenient multi-cast message transmission methodology
 11 in transmitting the audio and playback timing information to the synchrony group 20. As will be
 12 described in detail in connection with FIG. 4, the audio and playback timing information is in the
 13 form of a series of frames, with each frame having a time stamp. The time stamp indicates a time,
 14 relative to the time indicated by a clock maintained by the audio information channel device 23, at
 15 which the frame is to be played. Depending on the size or sizes of the messages used in the selected
 16 multi-cast message transmission methodology and the size or sizes of the frames, a message may
 17 contain one frame, or multiple frames, or, alternatively, a frame may extend across several messages.

18 The audio information channel device 23 also provides clock time information to the master
 19 device 21 and each of the slave devices 22(g) individually over network 12 using a highly accurate
 20 clock time information transmission methodology. The distribution of the clock time information
 21 is represented in FIG. 2 by the arrows labeled "AICD_CLK_INF (M)" (in the case of the clock time
 22 information provided to the master device 21) and "AICD_CLK_INF (S₁)" through
 23 "AICD_CLK_INF (S_G)" (in the case of audio information channel device clock information provided
 24 to the slave devices 22(g)). In one embodiment, the master device 21 and slave devices 22(g) make
 25 use of the well-known SNTP (Simple Network Time Protocol) to obtain current clock time
 26 information from the audio information channel device 23. The SNTP makes use of a unicast
 27 message transfer methodology, in which one device, such as the audio information channel device
 28 23, provides clock time information to a specific other device, such as the master device 21 or a

1 slave device 22(g), using the other device's network, or unicast, address. Each of the master device
 2 21 and slave devices 22(g) will periodically initiate SNTP transactions with the audio information
 3 channel device 23 to obtain the clock time information from the audio information channel device
 4 23. As will be described below in more detail, the master device 21 and each slave device 22(g)
 5 make use of the clock time information to determine the time differential between the time indicated
 6 by the audio information channel device's clock and the time indicated by its respective clock, and
 7 use that time differential value, along with the playback time information associated with the audio
 8 information and the respective device's local time as indicated by its clock to determine when the
 9 various frames are to be played. This enables the master device 21 and the slave devices 22(g) in
 10 the synchrony group 20 to play the respective frames simultaneously.

11 As noted above, the control information provided by the user to the master device 21 through
 12 the user interface module 13 can also enable the master device 21 to, in turn, enable another zone
 13 player 11(n') to join the synchrony group as a new slave device 22(g). In that operation, the user
 14 interface module 13 will provide control information, including the identification of the zone player
 15 11(n') that is to join the synchrony group to the master device 21. After it receives the identification
 16 of the zone player 11(n') that is to join the synchrony group, the master device 21 will exchange
 17 control information, which is represented in FIG. 2 by the arrows labeled SLV_DEV_CTRL_INF
 18 (S_i) through SLV_DEV_CTRL_INF (S_G) group slave control information, over the network 12 with
 19 the zone player 11(n') that is identified in the control information from the user interface module 13.
 20 The control information that the master device 21 provides to the new zone player 11(n') includes
 21 the network address of the zone player 11(n) that is operating as the audio information channel
 22 device 23 for the synchrony group, as well as the multi-cast address that the audio information
 23 channel device 23 is using to broadcast the audio and playback timing information over the network.
 24 The zone player that is to operate as the new slave device 22(g') uses the multi-cast address to begin
 25 receiving the multi-cast messages that contain the audio information for the audio program being
 26 played by the synchrony group.

27 It will be appreciated that, if the zone player 11(n) that is operating as the master device 21
 28 for the synchrony group 20 is also operating the audio information channel device 23, and if there

1 are no slave devices 22(g) in the synchrony group 20, the audio information channel device 23 may
2 not be transmitting audio and playback timing information over the network. In that case, if the new
3 slave device 22(g') is the first slave device in the synchrony group, the zone player 11(n) that is
4 operating as both the master device 21 and audio information channel device 23, can begin
5 transmitting the audio and playback timing information over the network 12 when the slave device
6 22(g') is added to the synchrony group 20. The zone player 11(n) can maintain a count of the
7 number of slave devices 22(g) in the synchrony group 20 as they join and disengage, and, if the
8 number drops to zero, it can stop transmitting the audio and playback timing information over the
9 network 12 to reduce the message traffic over the network 12.

10 The new slave device 22(g') added to the synchrony group 20 uses the network address of
11 the audio information channel device 23 for several purposes. In particular, the new slave device
12 22(g') will, like the master device 21 (assuming the zone player 11(n) operating as the master device
13 21 is not also the audio information channel device 23), engage in SNTP transactions with the audio
14 information channel device 23 to obtain the clock timing information from the audio information
15 channel device 23. In addition, the new slave device 22(g') can notify the audio information channel
16 device 23 that it is a new slave device 22(g') for the synchrony group 20 and provide the audio
17 information channel device 23 with its network address. As will be described below, in one
18 embodiment, particularly in connection with audio information obtained from a source, such as a
19 digital data storage device, which can provide audio information at a rate that is faster than the rate
20 at which it will be played, the audio information channel device 23 will buffer audio and timing
21 information and broadcast it over the network 12 to the synchrony group 20 generally at a rate at
22 which it is provided by the source. Accordingly, when a new slave device 22(g') joins the synchrony
23 group 20, the playback timing information may indicate that the audio information that is currently
24 being broadcast by the audio information channel device 23 using the multi-cast methodology is to
25 be played back some time in the future. To reduce the delay with which the new slave device 22(g')
26 will begin playback, the audio information channel device 23 can also retransmit previously
27 transmitted audio and timing information that it had buffered to the new slave device 22(g') using
28 the unicast network address of the slave device 22(g').

1 The master device 21 can also use the slave device control information exchanged with the
2 slave devices 22(g) for other purposes. For example, the master device 21 can use the slave device
3 control information to initiate a migration of the master from its zone player 11(n) to another zone
4 player 11(n'). This may occur for any of a number of reasons, including, for example, that the master
5 device 21 is terminating playback by it of the audio program and is leaving the synchrony group 20,
6 but one or more of the other devices in the synchrony group is to continue playing the audio
7 program. The master device 21 may also want to initiate a migration if it is overloaded, which can
8 occur if, for example, the zone player 11(n) that is the master device 21 for its synchrony group is
9 also operating as an audio information channel device 23 for another synchrony group.

10 The user can also use the user interface module 13 to adjust playback volume by the
11 individual zone players 11(n) comprising the synchrony group. In that operation, the user interface
12 module 13 provides information identifying the particular device whose volume is to be adjusted,
13 and the level at which the volume is to be set to the master device 21. If the device whose volume
14 is to be adjusted is the master device 21, the master device 21 can adjust its volume according to the
15 information that it receives from the user interface module 13. On the other hand, if the device
16 whose volume is to be adjusted is a slave device 22(g), the master device 21 can provide group slave
17 control information to the respective slave device 22(g), to enable it to adjust its volume.

18 The user can also use the user interface module 13 to enable a synchrony group 20 to cancel
19 playing of the track in an audio program that is currently being played, and to proceed immediately
20 to the next track. This may occur, for example, if the tracks for the program is in the form of a series
21 of digital audio information files, and the user wishes to cancel playback of the track that is defined
22 by one of the files. In that case, when the master device 21 receives the command to cancel playback
23 of the current track, it will provide channel device control information to the audio information
24 channel device 23 so indicating. In response, the audio information channel device 23 inserts control
25 information into the audio and playback timing information, which will be referred to as a
26 "resynchronize" command. In addition, the audio information channel device 23 will begin
27 transmitting audio information for the next track, with timing information to enable it to be played

1 immediately. The resynchronize command can also enable playback of a track to be cancelled
2 before it has been played. Details of these operations will be described below.

3 As noted above, there may be multiple synchrony groups in the network audio system 10,
4 and further that, for example, a zone player 11(n) may operate both as a master device 21 or a slave
5 device 22(g) in one synchrony group, and as the audio information channel device 23 providing
6 audio and playback timing information and clock timing information for another synchrony group.
7 An illustrative arrangement of this will be described in connection with FIG. 2A. With reference
8 to FIG. 2A, that FIG. depicts elements of two synchrony groups, identified by reference numerals
9 20(1) and 20(2), respectively. For clarity, FIG. 2A does not show a number of elements, the
10 presence of which would be evident from FIGS. 1 and 2 as described above. For example, FIG. 2A
11 does not depict the audio information sources from which audio information is obtained for the
12 synchrony groups or the audio reproduction devices that are used to produce sound for the master
13 and slave devices, which are depicted in both FIGS. 1 and 2. In addition, FIG. 2A does not depict
14 arrows that represent control information provided by the respective master devices to the slave
15 devices in the respective synchrony groups, or to the audio information channel devices that provide
16 audio and timing information for the respective synchrony groups, which are depicted in FIG. 2. In
17 addition, FIG. 2A does not depict the arrows that represent the clock timing information provided
18 by the audio information channel devices to the respective members of the respective synchrony
19 groups, which are also depicted in FIG. 2. As will be noted below, however, FIG. 2A does depict
20 arrows representing the audio and playback timing information provided by the respective audio
21 information channel devices for the respective synchrony groups 20(1), 20(2), to the master and
22 slave devices comprising the respective synchrony groups 20(1), 20(2).

23 Each synchrony group 20(1), 20(2) comprises elements of a number of zone players. A
24 functional block diagram of a zone player will be described below in connection with FIG. 3.
25 Synchrony group 20(1) includes a master device 21(1) and "K" slave devices 22(1)(1) through
26 22(K)(1) (the index "1" in reference numeral 21(1) and the last index in reference numeral 22(1)(1)
27 through 21(K)(1) corresponds to the index of the synchrony group 20(1) to which they belong)
28 utilize zone players 11(1) and 11(K+1) respectively. Similarly, synchrony group 20(2) includes a

1 master device 21(2) and "L" slave devices 22(1)(2) through 22(L)(2) that utilize zone players
 2 11(K+2) through 11(K+L+2). In the illustrative arrangement depicted in FIG. 2A, both synchrony
 3 groups 20(1) and 20(2) are controlled by the user interface module 13, which can provide control
 4 information to, and receive status information from, the master devices 21(1) and 21(2)
 5 independently. It will be appreciated that separate user interface modules may be provided to
 6 provide control information to, and receive status information from, the respective master devices
 7 21(1), 21(2).

8 As noted above, the slave device 22(1)(2) in synchrony group 20(2) utilizes zone player
 9 11(K+3). In the illustrative arrangement depicted in FIG. 2A, the audio information channel device
 10 23(1) that provides audio and playback timing information to the master and slave devices 21(1),
 11 22(1)(1),..., 22(K)(1) of synchrony group 20(1) also utilizes zone player 11(K+3). As noted above,
 12 this may occur if, for example, the audio information source that is to provide audio information to
 13 be played by the synchrony group 20(1) is connected to the zone player 11(K+3). Thus, when the
 14 master device 21(1) of synchrony group 20(1) exchanges channel device control information with
 15 the audio information channel device 23(1), it is effectively exchanging channel device control
 16 information with the zone player 11(K+3). Similarly, when the master and slave devices 21(1),
 17 22(1)(1),..., 22(K)(1) of synchrony group 20(1) receive audio and playback timing information, as
 18 well as clock timing information, from the audio information channel device 23(1), they are
 19 effectively receiving the information from the zone player 11(K+3). FIG. 2A depicts a multi-headed
 20 arrow representing audio and playback timing information transmitted by the zone player 11(K+3),
 21 as audio information channel device 23(1), to the master and slave devices 21(1),
 22 22(1)(1),..., 11(K)(1) comprising synchrony group 20(1).

23 On the other hand, in the illustrative arrangement depicted in FIG. 2A, the synchrony group
 24 20(2) utilizes a zone player 11(K+L+3) as its audio information channel device 23(2). As with
 25 synchrony group 20(1), when the master device 21(2) of synchrony group 20(2) exchanges channel
 26 device control information with the audio information channel device 23(2), it is effectively
 27 exchanging channel device control information with the zone player 11(K+L+3). Similarly, when
 28 the master and slave devices 21(2), 22(1)(2),..., 22(L)(2) of synchrony group 20(2) receive audio and

1 playback timing information, as well as clock timing information, from the audio information
2 channel device 23(2), they are effectively receiving the information from the zone player
3 11(K+L+3). FIG. 2A depicts a multi-headed arrow representing audio and playback timing
4 information transmitted by the zone player 11(K+3) as audio information channel device 23(2) to
5 the master and slave devices 21(2), 22(1)(2),...,22(L)(2) comprising synchrony group 20(2).

6 In the illustrative arrangement depicted in FIG. 2A, zone player 11(K+L+3), which is the
7 audio information channel device 23(2) for synchrony group 20(2), is not shown as being either a
8 master or a slave device in another synchrony group. However, it will be appreciated that zone
9 player 11(K+L+3) could also be utilized as the master device or a slave device for another synchrony
10 group. Indeed, it will be appreciated that the zone player that is utilized as the audio information
11 channel device for synchrony group 20(2) may also be a zone player that is utilized as the master
12 device 21(1) or a slave device 22(1)(1),..., 22(K)(1) in the synchrony group 20(1).

13 A zone player 11(n) that is utilized as a member of one synchrony group may also be utilized
14 as the audio information channel device for another synchrony group if the audio information source
15 that is to supply the audio information that is to be played by the other synchrony group is connected
16 to that zone player 11(n). A zone player 11(n) may also be utilized as the audio information channel
17 device for the other synchrony group if, for example, the audio information source is an audio
18 information source 16(m) (FIG. 1) that is connected to the network 12 or an audio information
19 source that is available over a wide area network such as the Internet. The latter may occur if, for
20 example, the zone player 11(n) has sufficient processing power to operate as the audio information
21 channel device and it is in an optimal location in the network 12, relative to the zone players
22 comprising the other synchrony group (that is the synchrony group for which it is operating as audio
23 information channel device) for providing the audio and playback timing information to the
24 members of the other synchrony group. Other circumstances under which the zone player 11(n) that
25 is utilized as a member of one synchrony group may also be utilized as the audio information
26 channel device for another synchrony group will be apparent to those skilled in the art.

As was noted above, the master device 21 for a synchrony group 20 may be migrated from one zone player 11(n) to another zone player 11(n'). As was further noted above, the audio information channel device 23 for a synchrony group 20 may be migrated from one zone player 11(n) to another zone player 11(n'). It will be appreciated that the latter may occur if, for example, the audio information source that provides the audio program for the synchrony group is not connected to the zone player 11(n) that is operating as the audio information channel device 23, but instead is one of the audio information sources 16(m) connected to the network 12 or a source available over a wide area network such as the Internet. Operations performed during a migration of an audio information channel device 23 from one zone player 11(n) to another zone player 11(n') will generally depend on the nature of the audio information that is being channeled by the audio information channel device 23. For example, if the audio information source provides streaming audio, the zone player 11(n) that is currently operating as the audio information channel device 23 for the synchrony group 20, can provide the following information to the other zone player 11(n') that is to become the audio information channel device 23 for the synchrony group 20:

(a) the identification of the source of the streaming audio information,

(b) the time stamp associated with the frame that the zone player 11(n) is currently forming,

and

(c) the identifications of the zone players that are operating as the master device 21 and slave devices 22(g) comprising the synchrony group 20.

After the zone player 11(n') receives the information from the zone player 11(n), it will begin receiving the streaming audio from the streaming audio information source identified by the zone player 11(n), assemble the streaming audio information into frames, associate each frame with a time stamp, and transmit the resulting audio and playback timing information over the network 12. The zone player 11(n') will perform these operations in the same manner as described above, except that, instead of using the time indicated by its digital to analog converter clock 34 directly in generating the time stamps for the frames, the initial time stamp will be related to the value of the time stamp that is provided by the zone player 11(n) (reference item (b) above), with the rate at which the time

1 stamps are incremented corresponding to the rate at which its (that is, the zone player 11(n)'s) clock
 2 increments. In addition, the zone player 11(n') will notify the zone players that are operating as the
 3 master device 21 and slave devices 22(g) of the synchrony group 20 that it is the new audio
 4 information channel device 23 for the synchrony group 20, and provide the multi-cast address that
 5 it will be using to multi-cast the audio and playback timing information, as well as its unicast
 6 network address. After the members of the synchrony group 20 receive the notification from the
 7 zone player 11(n') indicating that it is the new audio information channel device 23 for the synchrony
 8 group 20, they will receive the audio and playback timing information from the zone player 11(n')
 9 instead of the zone player 11(n), using the multi-cast address provided by the zone player 11(n').
 10 In addition, they can utilize the zone player 11(n)'s unicast network address to obtain current time
 11 information therefrom. It will be appreciated that the zone player 11(n') will determine its current
 12 time in relation to the time stamp that is provided by the zone player 11(n) (reference item (b) above)
 13 or the current time information that it received from the zone player 11(n) using the SNTP protocol
 14 as described above.

15 Generally similar operations can be performed in connection with migrating the audio
 16 information channel device from one zone player 11(n) to another zone player 11(n') if the audio
 17 information is from one or more audio information files, such as may be the case if the audio
 18 information comprises MP3 or WAV files that are available from sources such as sources 16(m)
 19 connected to the network 12 or over from sources available over a wide area network such as the
 20 Internet, except for differences to accommodate the fact that the audio information is in files. In that
 21 case, the zone player 11(n) that is currently operating as the audio information channel device 23 for
 22 the synchrony group 20 can provide the following information to the zone player 11(n') that is to
 23 become the audio information channel device 23 for the synchrony group 20:

24 (d) a list of the audio information files containing the audio information that is to be played;

25 (e) the identification of the file for which the zone player 11(n) is currently providing audio
 26 and playback timing information, along with the offset into the file for which the current item of

1 audio and playback timing information is being generated and the time stamp that the zone player
2 11(n) is associating with that frame, and

3 (f) the identifications of the zone players that comprise the master device 21 and slave
4 devices 22(g) comprising the synchrony group 20.

5 After the zone player 11(n') receives the information from the zone player 11(n), it will begin
6 retrieving audio information from the file identified in item (e), starting at the identified offset. In
7 addition, the zone player 11(n') can assemble the retrieved audio information into frames, associate
8 each frame with a time stamp and transmit the resulting audio and playback timing information over
9 the network 12. The zone player 11(n') will perform these operations in the same manner as
10 described above, except that, instead of using the time indicated by its digital to analog converter
11 clock 34 directly in generating the time stamps for the frames, the value of the initial time stamp will
12 be related to the time stamp that is provided by the zone player 11(n) (reference item (e) above), with
13 the rate at which the time stamps are incremented corresponding to the rate at which its (that is, the
14 zone player 11(n')s) clock increments. In addition, the zone player 11(n') will notify the zone
15 players that are operating as the master device 21 and slave devices 22(g) of the synchrony group
16 20 that it is the new audio information channel device 23 for the synchrony group 20, and provide
17 the multi-cast address that it will be using to multi-cast the audio and playback timing information,
18 as well as its unicast network address. After the members of the synchrony group 20 receive the
19 notification from the zone player 11(n') indicating that it is the new audio information channel device
20 23 for the synchrony group 20, they will receive the audio and playback timing information from
21 the zone player 11(n') instead of the zone player 11(n), using the multi-cast address provided by the
22 zone player 11(n'). In addition, they can utilize the zone player 11(n')s unicast network address to
23 obtain current time information therefrom. It will be appreciated that the zone player 11(n') will
24 determine its current time in relation to the time stamp that is provided by the zone player 11(n)
25 (reference item (b) above) or the current time information that it received from the zone player 11(n)
26 using the SNTP protocol as described above. The zone player 11(n') will process successive audio
27 information files in the list that it receives from the zone player 11(n) (reference item (d)).

1 Operations performed by the zone players 11(n) and 11(n') in connection with migration of
2 the audio information channel device 23 for other types of audio information will be apparent to
3 those skilled in the art. In any case, preferably, the zone player 11(n) will continue operating as an
4 audio information channel device 23 for the synchrony group 20 for at least a brief time after it
5 notifies the zone player 11(n') that it is to become audio information channel device for the
6 synchrony group, so that the zone player 11(n') will have time to notify the zone players in the
7 synchrony group 20 that it is the new audio information channel device 23 for the synchrony group.

8 Before proceeding further in describing operations performed by the network audio system
9 10, it would be helpful to provide a detailed description of a zone player 11(n) constructed in
10 accordance with the invention. FIG. 3 depicts a functional block diagram of a zone player 11(n)
11 constructed in accordance with the invention. All of the zone players in the network audio system
12 10 may have similar construction. With reference to FIG. 3, the zone player 11(n) includes an audio
13 information source interface 30, an audio information buffer 31, a playback scheduler 32, a digital
14 to analog converter 33, an audio amplifier 35, an audio reproduction device interface 36, a network
15 communications manager 40, and a network interface 41, all of which operate under the control of
16 a control module 42. The zone player 11(n) also has a device clock 43 that provides timing signals
17 that control the general operations of the zone player 11(n). In addition, the zone player 11(n)
18 includes a user interface module interface 44 that can receive control signals from the user interface
19 module 13 (FIGS. 1 and 2) for controlling operations of the zone player 11(n), and provide status
20 information to the user interface module 13.

21 Generally, the audio information buffer 31 buffers audio information, in digital form, along
22 with playback timing information. If the zone player 11(n) is operating as the audio information
23 channel device 23 (FIG. 2) for a synchrony group 20, the information that is buffered in the audio
24 information buffer 31 will include the audio and playback timing information that will be provided
25 to the devices 21 and 22(g) in the synchrony group 20. If the zone player 11(n) is operating as the
26 master device 21 or a slave device 22(g) for a synchrony group, the information that is buffered in
27 the audio information buffer 31 will include the audio and playback timing information that the zone
28 player 11(n) is to play.

1 The audio information buffer 31 can receive audio and playback timing information from two
 2 sources, namely, the audio information source interface 30 and the network communications
 3 manager 40. In particular, if the zone player 11(n) is operating as the audio information channel
 4 device 23 for a synchrony group 20, and if the audio information source is a source 14(n)(s)
 5 connected to the zone player 11(n), the audio information buffer 31 will receive and buffer audio and
 6 playback timing information from the audio information source interface 30. On the other hand, if
 7 the zone player 11(n) is operating as the audio information channel device 23 for a synchrony group
 8 20, and if the audio information source is a source 16(m) connected to the network 12, or a source
 9 available over the wide area network, the audio information buffer 31 will receive and buffer audio
 10 and playback timing information from the network communications manager 40. It will be
 11 appreciated that, if the zone player 11(n) is not a member of the synchrony group, the zone player
 12 11(n) will not play this buffered audio and playback timing information.

13 On yet another hand, if the zone player 11(n) is operating as the master device 21 or a slave
 14 device 22(g) in a synchrony group, and if the zone player 11(n) is not also the audio information
 15 channel device 23 providing audio and playback timing information for the synchrony group 20, the
 16 audio information buffer 31 will receive and buffer audio and playback timing information from the
 17 network communications manager 40.

18 The audio information source interface 30 connects to the audio information source(s)
 19 14(n)(s) associated with the zone player 11(n). While the zone player 11(n) is operating as audio
 20 information channel device 23 for a synchrony group 20, and if the audio information is to be
 21 provided by a source 14(n)(s) connected to the zone player 11(n), the audio information source
 22 interface 30 will selectively receive audio information from one of the audio information source(s)
 23 14(n)(s) to which the zone player is connected and store the audio information in the audio
 24 information buffer 21. If the audio information from the selected audio information source 14(n)(s)
 25 is in analog form, the audio information source interface 30 will convert it to digital form. The
 26 selection of the audio information source 14(n)(s) from which the audio information source interface
 27 30 receives audio information is under control of the control module 42, which, in turn, receives
 28 control information from the user interface module through the user interface module interface 44.

1 The audio information source interface 30 adds playback timing information to the digital audio
2 information and buffers the combined audio and playback timing information in the audio
3 information buffer 21.

4 More specifically, as noted above, the audio information source interface 30 receives audio
5 information from an audio information source 14(n)(s), converts it to digital form if necessary, and
6 buffers it along with playback timing information in the audio information buffer 21. In addition,
7 the audio information source interface 30 will also provide formatting and scheduling information
8 for the digital audio information, whether as received from the selected audio information source
9 14(n)(s) or as converted from an analog audio information source. As will be made clear below, the
10 formatting and scheduling information will control not only playback by the zone player 11(n) itself,
11 but will also enable other zone players 11(n'), 11(n''),... that may be in a synchrony group for which
12 the zone player 11(n) is the master device, to play the audio program associated with the audio
13 information in synchrony with the zone player 11(n).

14 In one particular embodiment, the audio information source interface 30 divides the audio
15 information associated with an audio work into a series of frames, with each frame comprising
16 digital audio information for a predetermined period of time. As used herein, an audio track may
17 comprise any unit of audio information that is to be played without interruption. On the other hand,
18 an audio program may comprise a series of one or more audio tracks that are to be played in
19 succession. It will be appreciated that the tracks comprising the audio program may also be played
20 without interruption, or alternatively playback between tracks may be interrupted by a selected time
21 interval. FIG. 4 schematically depicts an illustrative framing strategy used in connection with one
22 embodiment of the invention for a digital audio stream comprising an audio work. More
23 specifically, FIG. 4 depicts a framed digital audio stream 50 comprising a sequence of frames 51(1)
24 through 51(F) (generally identified by reference numeral 51(f)). Each frame 51(f), in turn, comprises
25 a series of audio samples 52(f)(1) through 52(f)(S) (generally identified by reference numeral
26 52(f)(s)) of the audio track. Preferably all of the frames will have the same number "S" of audio
27 samples, although it will be appreciated from the following that that is primarily for convenience.
28 On the other hand, it will be appreciated that, the number of audio samples may differ from "S"; this

1 may particularly be the case if the frame 51(f) contains the last audio samples for the digital audio
 2 stream for a particular audio work. In that case, the last frame 51(F) will preferably contain samples
 3 52(F)(1) through 52(F)(x), where "x" is less than "S." Generally, it is desirable that the number of
 4 samples be consistent among all frames 51(f), and in that case padding, which will not be played,
 5 can be added to the last frame 51(F).

6 Associated with each frame 51(f) is a header 55(f) that includes a number of fields for storing
 7 other information that is useful in controlling playback of the audio samples in the respective frame
 8 51(f). In particular, the header 55(f) associated with a frame 51(f) includes a frame sequence number
 9 field 56, an encoding type field 57, a sampling rate information field 58, a time stamp field 60, an
 10 end of track flag 61, and a length flag field 62. The header 55(f) may also include fields (not shown)
 11 for storing other information that is useful in controlling playback. Generally, the frame sequence
 12 number field 56 receives a sequence number "f" that identifies the relative position of the frame 51(f)
 13 in the sequence of frames 51(1)..51(f)...51(F) containing the digital audio stream 50. The encoding
 14 type field 57 receives a value that identifies the type of encoding and/or compression that has been
 15 used in generating the digital audio stream. Conventional encoding or compression schemes include,
 16 for example, the well-known MP3 and WAV encoding and/or compression schemes, although it will
 17 be appreciated that other schemes may be provided for as well. The sampling rate information field
 18 58 receives sampling rate information that indicates the sampling rate for the audio samples 52(f)(s).
 19 As will be apparent to those skilled in the art, the sampling rate determines the rate at which the zone
 20 player 11(n) is to play the audio samples 52(f)(s) in the frame, and, as will be described below,
 21 determines the period of the digital to analog converter clock 34.

22 The condition of the end of work flag 61 indicates whether the frame 51(f) contains the last
 23 digital audio samples for the audio track associated with the framed digital audio work 50. If the
 24 frame 51(f) does not contain the audio samples that are associated with the end of the digital audio
 25 stream 50 for a respective audio work, the end of work flag will be clear. On the other hand, if the
 26 frame 51(f) does contain the audio samples that are associated with the end of the digital audio
 27 stream 50 for a respective audio work, the end of work flag 61 will be set. In addition, since the
 28 number of valid audio samples 52(F)(s) in the frame 51(F), that is, the samples that are not padding,

1 may be less than "S," the default number of audio samples in a frame 51(f), the length flag field 62
 2 will contain a value that identifies the number of audio samples in 52(F)(s) in the last frame 51(F)
 3 of the audio work 50. If, as noted above, the frames have a consistent number "S" of samples, the
 4 samples 52(F)(x+1) through 52(F)(S) will contain padding, which will not be played.

5 The time stamp field 60 stores a time stamp that identifies the time at which the zone player
 6 11(n) is to play the respective frame. More specifically, for each frame of a framed digital audio
 7 stream 50 that is buffered in the audio information buffer 21, the audio information source interface
 8 30, using timing information from the digital to analog converter clock 34, will determine a time at
 9 which the zone player 11(n) is to play the respective frame, and stores a time stamp identifying the
 10 playback time in the time stamp field 60. The time stamp associated with each frame will later be
 11 used by the playback scheduler 32 to determine when the portion of the digital audio stream stored
 12 in the frame is to be coupled to the digital to analog converter 33 to initiate play back. It will be
 13 appreciated that the time stamps that are associated with frames in sequential frames 51(1),
 14 51(2),...,51(F), will be such that they will be played back in order, and without an interruption
 15 between the sequential frames comprising the digital audio stream 50. It will further be appreciated
 16 that, after a time stamp has been determined for the first frame, stored in frame 51(1), of a digital
 17 audio stream 50, the audio information source interface 30 can determine time stamps for the
 18 subsequent frame 51(2), 51(3),...,51(F) in relation to the number of samples "S" in the respective
 19 frames and the sample rate. The time stamps will also preferably be such that frames will be played
 20 back after some slight time delay after they have been buffered in the audio information buffer 21;
 21 the purpose for the time delay will be made clear below.

22 Returning to FIG. 3, in addition to dividing the digital audio information into frames, the
 23 audio information source interface 30 also aggregates and/or divides the frames 51(f) as necessary
 24 into packets, each of which will be of a length that would fit into a message for transmission over
 25 the network, and associates each packet with a packet sequence number. For example, if a packet
 26 will accommodate multiple frames 51(f), 51(f+1),... 51(f+y-1), it will aggregate them into a packet
 27 and associate them with a packet number, for example p(x). If the entire frames 51(f) and 51(f+y-1)
 28 was accommodated in packet p(x), where "x" is the sequence number, which will occur if the size

1 of a packet is an exact multiple of the frame size, the next packet, $p(x+1)$ will begin with frame
 2 $51(f+y)$ and will include frames $51(f+y), \dots, 51(f+2y-1)$. Subsequent packets $p(x+2), \dots$ will be
 3 formed in a similar manner. On the other hand, if the packet length will not accommodate an exact
 4 multiple of the frame size, the last frame in the packet will be continued at the beginning of the next
 5 packet.

6 If the audio information source interface 30 is aware of track boundaries, which may be the
 7 case if the tracks are divided into files, the packets will reflect the track boundaries, that is, the
 8 packets will not contain frames from two tracks. Thus, if the last frames associated with a track are
 9 insufficient to fill a packet, the packet will contain padding from the last frame associated with the
 10 track to the end of the packet, and the next packet will begin with the first frames associated with
 11 the next track.

12 In one embodiment, the audio information source interface 30 stores the packets in the audio
 13 information buffer 31 in a ring buffer. As is conventional, a ring buffer includes a series of storage
 14 locations in the buffer. Each entry will be sufficient to store one packet. Four pointers are used in
 15 connection with the ring buffer, a first pointer pointing to the beginning of the ring buffer, a second
 16 pointer pointing to the end of the ring buffer, a third "write" pointer pointing to the entry into which
 17 a packet will be written and a fourth "read" pointer pointing to the entry from which packet will be
 18 read for use in playback. When a packet is read from the ring buffer for playback, it will be read
 19 from the entry pointed to by the read pointer. After the packet has been read, the read pointer will
 20 be advanced. If the read pointer points beyond the end of the ring buffer, as indicated by the end
 21 pointer, it will be reset to point to the entry pointed to by the beginning pointer, and the operations
 22 can be repeated.

23 On the other hand, when the audio information source interface 30 stores a packet in the ring
 24 buffer, first determine whether the entry pointed to by the write pointer points to the same entry as
 25 the entry pointed to by the read pointer. If the write pointer points to the same entry as the entry
 26 pointed to by the read pointer, the entry contains at least a portion of a packet that has not yet been
 27 read for playback, and the audio information source interface 30 will delay storage of the packet until

1 the entire packet has been read and the read pointer advanced. After the read pointer has been
2 advanced, the audio information source interface 30 can store the packet in the entry pointed to by
3 the write pointer. After the packet has been stored, the audio information source interface 30 will
4 advance the write pointer. If the write pointer points beyond the end of the ring buffer, as indicated
5 by the end pointer, it will be reset to point to the entry pointed to by the beginning pointer, and the
6 operations can be repeated.

7 As noted above, the zone player 11(n) can operate both as an audio information channel
8 device 23 and a member of the synchrony group 20 of which it is a member. In that case, the audio
9 information buffer 31 can contain one ring buffer. On the other hand, the zone player 11(n) can
10 operate as an audio information channel device 23 for one synchrony group 20(1) (FIG. 2A) and a
11 member of another synchrony group 20(2). In that case, the audio information buffer 31 would
12 maintain two ring buffers, one for the audio and timing information associated with synchrony group
13 20(1), and the other for the audio and timing information associated with synchrony group 20(2).
14 It will be appreciated that, in the latter case, the zone player 11(n) will only use the audio and timing
15 information that is associated with synchrony group 20(2) for playback.

16 The playback scheduler 32 schedules playback of the audio information that is buffered in
17 the audio information buffer 31 that is to be played by the zone player 11(n). Accordingly, under
18 control of the playback scheduler 32, the digital audio information that is buffered in the audio
19 information buffer 21 that is to be played by the zone player 11(n) is transferred to the digital to
20 analog converter 33 for playback. As noted above, if the zone player 11(n) is operating as an audio
21 information channel device 23 for a synchrony group 20 for which it is not a member, the playback
22 scheduler 32 will not schedule the digital audio information that is to be played by that synchrony
23 group 20 for playback. The playback scheduler 32 only schedules the digital audio information, if
24 any, that is buffered in the audio information buffer 31 that is associated with a synchrony group for
25 which the zone player 11(n) is a member, whether as master device 21 or a slave device 22(g).

26 Essentially, the playback scheduler 32 makes use of the read pointer associated with the
27 circular buffer that contains the audio and playback timing information that is to be played by the

1 zone player 11(n). The playback scheduler 32 retrieves the packet information from the entry of the
 2 ring buffer pointed to by the read pointer, and then advances the ring pointer as described above. The
 3 playback scheduler 32 determines the boundaries of the frames in the packet and uses the time
 4 stamps in the time stamp fields 60 associated with the respective frame 51(f), along with timing
 5 information provided by the zone player 11(n)'s digital to analog converter clock 34, to determine
 6 when the respective frame is to be transferred to the digital to analog converter 33. Generally, when
 7 the time stamp associated with a buffered digital audio information frame corresponds to the current
 8 time as indicated by the digital to analog converter clock 34, the playback scheduler 32 will enable
 9 the respective frame to be transferred to the digital to analog converter 33.

10 The digital to analog converter 33, also under control of the digital to analog converter clock
 11 34, converts the buffered digital audio information to analog form, and provides the analog audio
 12 information to the audio amplifier 35 for amplification. The amplified analog information, in turn,
 13 is provided to the audio reproduction devices 15(n)(r) through the audio reproduction device
 14 interface 36. The audio reproduction devices 15(n)(r) transform the analog audio information signal
 15 to sound thereby to provide the audio program to a listener. The amount by which the audio
 16 amplifier 35 amplifies the analog signal is controlled by the control module 42, in response to
 17 volume control information provided by the user through the user interface module 13.

18 The network communications manager 40 controls network communications over the
 19 network 12, and the network interface 41 transmits and receives message packets over the network
 20 12. The network communications manager 40 generates and receives messages to facilitate the
 21 transfer of the various types of information described above in connection with FIG. 2, including the
 22 channel device control information, slave device control information, audio and playback timing
 23 information and the audio information channel device's clock timing information. In connection
 24 with the channel device control information and the slave device control information, the network
 25 communications manager 40 will generate messages for transfer over the network 12 in response to
 26 control information from the control module 42. Similarly, when the network communications
 27 manager 40 receives messages containing channel device control information and slave device

1 control information, the network communications manager will provide the information to the
2 control module 42 for processing.

3 With regards to the audio information channel device's clock timing information, as noted
4 above, the master device 21 and slave devices 22(g) of the synchrony group 20 obtain the clock
5 timing information from the audio information channel device 23 using the well-known SNTP. If
6 the zone player 11(n) is operating as the audio information channel device 23 for a synchrony group,
7 during the SNTP operation, it will provide its current time, particularly a current time as indicated
8 by its digital to analog converter clock 34. On the other hand, if the zone player 11(n) is operating
9 as the master device 21 or slave device 22(g) of a synchrony group 20, it will receive the clock
10 timing information from the audio information channel device 23. After the respective device 21,
11 22(g) has obtained the audio information channel device's clock timing information, it will generate
12 a differential time value ΔT representing the difference between the time T indicated by its digital
13 to analog converter clock 34 and the current time information from the audio information channel
14 device 23. The differential time value will be used to update the time stamps for the frames of the
15 digital audio stream 50 (FIG. 4) that are received from the audio information channel device.

16 With regards to the audio and playback timing information, operations performed by the
17 network communications manager 40 will depend on whether

18 (i) the audio and playback timing information has been buffered in the audio information
19 buffer 31 for transmission, as audio information channel device 23, over the network 12 to the
20 master device 21 and/or slave devices 22(g) of a synchrony group, or

21 (ii) the audio and playback timing information has been received from the network 12 to be
22 played by the zone player 11(n) as either the master device 21 for a synchrony group or a slave
23 device in a synchrony group.

24 It will be appreciated that the network communications manager 40 may be engaged in both (i) and
25 (ii) contemporaneously, since the zone player 11(n) may operate both as the audio information
26 channel device 23(1) for a synchrony group 20(1) (reference FIG. 2A) of which it is not a member,

1 and a member of another synchrony group 20(2) for which another zone player 11(n') is the audio
2 information channel device 23(2). With reference to item (i) above, after a packet that is to be
3 transmitted has been buffered in the respective ring buffer, the network communications manager
4 40 retrieves the packet, packages it into a message and enables the network interface 41 to transmit
5 the message over the network 12. If the control module 42 receives control information from the
6 user interface module 13 (if the master device 21 is also the audio information channel device 23 for
7 the synchrony group 20) or from the master device (if the master device 21 is not the audio
8 information channel device 23 for the synchrony group 20) that would require the transmission of
9 a "resynchronize" command as described above, the control module 42 of the audio information
10 channel device 23 enables the network communications manager 40 to insert the command into a
11 message containing the audio and playback timing information. Details of the operations performed
12 in connection with the "resynchronize" command will be described below. As noted above, the
13 "resynchronize" command is used if the user enables a synchrony group to terminate the playback
14 of a track that is currently being played, or cancel playback of a track whose playback has not begun.

15 On the other hand, with reference to item (ii) above, if network interface 41 receives a
16 message containing a packet containing frames of audio and playback timing information that the
17 zone player 11(n) is to play either as a master device 21 or a slave device for a synchrony group 20,
18 the network interface 41 provides the audio and playback timing information to the network
19 communications manager 40. The network communications manager 40 will determine whether the
20 packet contains a resynchronize command and, if so, notify the control module 42, which will enable
21 operations to be performed as described below. In any case, the network communications manager
22 40 will normally buffer the various frames comprising the audio and playback timing information
23 in the audio information buffer 31, and in that operation will generally operate as described above
24 in connection with the audio information source interface 30. Before buffering them, however, the
25 network communications manager 40 will update their time stamps using the time differential value
26 described above. It will be appreciated that the network communications manager 40 will perform
27 similar operations whether the messages that contain the packets were multi-cast messages or unicast
28 messages as described above

1 The updating of the time stamps by the master device 21 and the slave devices 22(g) in the
 2 synchrony group 20 will ensure that they all play the audio information synchronously. In particular,
 3 after the network communications manager 40 has received a frame 51(f) from the network interface
 4 41, it will also obtain, from the digital to analog converter clock 34, the zone player 11(n)'s current
 5 time as indicated by its digital to analog converter clock 34. The network communications manager
 6 40 will determine a time differential value that is the difference between the slave device's current
 7 clock time, as indicated by its digital to analog converter 34, and the audio information channel
 8 device's time as indicated by the audio information channel device's clock timing information.
 9 Accordingly, if the master or slave device's current time has a value T_S and the audio information
 10 channel device's current time, as indicated by the clock timing information, has a value T_C , the time
 11 differential value $\Delta T = T_S - T_C$. If the current time of the master or slave device in the synchrony group
 12 20, as indicated by its digital to analog converter clock 34, is ahead of the audio information channel
 13 device's clock time as indicated by the clock timing information received during the SNTP operation,
 14 the time differential value will have a positive value. On the other hand, if the master or slave
 15 device's current time is behind the audio information channel device's clock time, the time
 16 differential value ΔT will have a negative value. If the zone player 11(n) obtains clock timing
 17 information from the audio information channel device 23 periodically while it is a member of the
 18 synchrony group 20, the network communications manager 40 can generate an updated value for the
 19 time differential value ΔT when it receives the clock timing information from the audio information
 20 channel device 23, and will subsequently use the updated time differential value.

21 The network communications manager 40 uses the time differential value ΔT that it generates
 22 from the audio information channel device timing information and zone player 11(n)'s current time
 23 to update the time stamps that will be associated with the digital audio information frames that the
 24 zone player 11(n) receives from the audio information channel device. For each digital audio
 25 information frame that is received from the audio information channel device, instead of storing the
 26 time stamp that is associated with the frame as received in the message in the audio information
 27 buffer 21, the network communications manager 40 will store the updated time stamp with the
 28 digital audio information frame. The updated time stamp is generated in a manner so that, when the

1 zone player 11(n), as a member of the synchrony group plays back the digital audio information
2 frame, it will do so in synchrony with other devices in the synchrony group.

3 More specifically, after the zone player 11(n)'s network interface 41 receives a message
4 containing a packet that, in turn, contains one or more frames 51(f), it will provide the packet to the
5 network communications manager 40. For each frame 51(f) in the packet that the network
6 communications manager 40 receives from the network interface 41, the network communications
7 manager 40 will add the time differential value ΔT to the frame's time stamp, to generate the updated
8 time stamp for the frame 51(f), and store the frame 51(f), along with the header 55(f) with the
9 updated time stamp in the audio information buffer 31. Thus, for example, if a frame's time stamp
10 has a time value T_F , the network communications manager 40 will generate an updated time stamp
11 T_F^U having a time value $T_F^U = T_F + \Delta T$. Since time value T_F^U according to the slave device's digital
12 to analog converter clock 34 is simultaneous to the time value T_F according to the audio information
13 channel device's digital to analog converter clock 34, the zone player 11(n) device will play the
14 digital audio information frame at the time determined by the audio information channel device 23.
15 Since all of the members of the synchrony group 20 will perform the same operations, generating
16 the updated time stamps T_F^U for the various frames 51(f) in relation to their respective differential
17 time values, all of the zone players 11(n) in the synchrony group 20 will play them synchronously.
18 The network communications manager 40 will generate updated time stamps T_F^U for all of the time
19 stamps 60 in the packet, and then store the packet in the audio information buffer 31.

20 It will be appreciated that, before storing a packet in the audio information buffer 21, the
21 network communications manager 40 can compare the updated time stamps T_F^U associated with the
22 frames in the packet to the slave device's current time as indicated by its digital to analog converter
23 clock 34. If the network communications manager 40 determines the time indicated by the updated
24 time stamps of frames 51(f) in the packet are earlier than the zone player's current time, it can discard
25 the packet instead of storing it in the audio information buffer 21, since the zone player 11(n) will
26 not play them. That is, if the updated time stamp has a time value T_F^U that identifies a time that is
27 earlier than the zone player's current time T_S as indicated by the zone player's digital to analog
28 converter clock 34, the network communications manager 40 can discard the packet.

1 If the zone player 11(n) is operating as the master device 21 of a synchrony group 20, when
 2 the user, through the user interface module 13, notifies the zone player 11(n) that another zone player
 3 11(n') is to join the synchrony group 20 as a slave device 22(g), the control module 42 of the zone
 4 player 11(n) enables the network communications manager 40 to engage in an exchange of
 5 messages, described above in connection with FIG. 2, to enable the other zone player 11(n') to join
 6 the synchrony group 20 as a slave device. As noted above, during the message exchange, the
 7 messages generated by the network communications manager 40 of the zone player 11(n) will
 8 provide the network communications manager of the zone player 11(n') that is to join the synchrony
 9 group 20 with information such as the multi-cast address being used by the audio information
 10 channel device 23 that is providing the audio program to the synchrony group 20, as well as a unicast
 11 network address for the audio information channel device 23. After receiving that information, the
 12 network communications manager and network interface of the zone player 11(n') that is to join the
 13 synchrony group 20 can begin receiving the multi-cast messages containing the audio program for
 14 the synchrony group, engage in SNTP transactions with the audio information channel device 23 to
 15 obtain the latter's current time, and also enable the audio information channel device 23 to send the
 16 zone player 11(n') frames 51(f) that it had previously broadcast using the unicast message
 17 transmission methodology as described above.

18 On the other hand, if the network communications manager 40 and network interface 41 of
 19 the zone player 11(n) receive a message over the network 12 indicating that it is to become a slave
 20 device 22(g) of a synchrony group for which another zone player 11(n') is the master device, the
 21 network communications manager 40 for zone player 11(n) will provide a notification to the control
 22 module 42 of zone player 11(n). Thereafter, the control module 42 of zone player 11(n) can enable
 23 the network communications manager 40 of zone player 11(n) to perform the operations described
 24 above to enable it to join the synchrony group 20.

25 As noted above, the user, using user interface module 13, can enable the synchrony group
 26 to terminate playback of a track of an audio program that is currently being played. After playback
 27 of a track that is currently being played has been terminated, playback will continue in a
 28 conventional manner with the next track that has been buffered in the audio information buffer 31.

1 It will be appreciated that that could be the next track that is on the original play list, or a previous
2 track. In addition, the user can enable the synchrony group 20 to cancel playback of a track that it
3 has not yet begun to play, but for which buffering of packets has begun in the synchrony group 20.
4 Both of these operations make use of the "resynchronize" command that the master device 21 of the
5 synchrony group 20 can enable the audio information channel device 23 to include in the multi-cast
6 message stream that it transmits to the synchrony group 20. Generally, in response to receipt of the
7 resynchronize command, the members of the synchrony group 20 flush the ring buffer containing
8 the packets that they are to play in the future. In addition, if the members of the synchrony group
9 provide separate buffers for their respective digital to analog converters 33, the members will also
10 flush those buffers as well. After the audio information channel device transmits a packet containing
11 the resynchronize command:

12 (i) in the case of the use of the resynchronize command to terminate playing of a track that
13 is currently being played, the audio information channel device 23 will begin multi-casting packets
14 for the next track, to begin play immediately, and will continue through the play list in the manner
15 described above; and

16 (ii) in the case of the use of the resynchronize command to cancel play of a track for which
17 buffering has begun, but which is to be played in the future, the audio information channel device
18 23 will begin multi-casting packets for the track after the track that has been cancelled, to be played
19 beginning at the time the cancelled track was to begin play, and will also continue through the play
20 list in the manner described above.

21 It will be appreciated that,

22 (a) in the first case (item (i) directly above), the resynchronize command can enable the read
23 pointer to be set to point to the entry in the circular buffer into which the first packet for the next
24 track will be written, which will correspond to the entry to which the write pointer points, but

(b) in the second case (item (ii) directly above), the resynchronize command can enable the write pointer for the circular buffer to be set to point to the entry that contains the first packet for the track whose play is being cancelled.

It will further be appreciated that, if a track is cancelled for which buffering has not begun, the resynchronize command will generally not be necessary, since the audio information channel device 23 for the synchrony group 20 need only delete that track from the play list.

Operations performed in connection with use of the resynchronize command to cancel playback of a track that is currently being played will be described in connection with Packet Sequence A below, and operations performed in connection with the use of the resynchronize command to cancel playback of a track that is has not yet begun to play, but for which buffering of packets has begun, will be described in connection with Packet Sequence B below.

Packet Sequence A

(A1.0) [packet 57]

(A1.1) [continuation of frame 99]

(A1.2) [frame 100, time = 0:00:01, type = mp3 audio]

(A1.3) [frame 101, time = 0:00:02, type = mp3 audio]

(A1.4) [frame 102, time = 0:00:03, type = mp3 audio]

(A2.0) [packet 58]

(A2.1) [continuation of frame 102]

(A2.2) [frame 103, time = 0:00:04, type = mp3 audio]

(A2.3) [frame 104, time = 0:00:05, type = mp3 audio]

(A2.4) [frame 105, time = 0:00:06, type = mp3 audio]

(A3.0) [packet 59]

(A3.1) [continuation of frame 105]

(A3.2) [frame 106, time = 0:00:07, type = mp3 audio]

- 1 (A3.3) [frame 107, time = 0:00:08, type = mp3 audio]
- 2 (A3.4) [frame 108, time = 0:00:09, type = mp3 audio]
- 3
- 4 (A4.0) [packet 60]
- 5 (A4.1) [continuation of frame 108]
- 6 (A4.2) [frame 109, time = 0:00:10, type = mp3 audio]
- 7 (A4.3) [Resynchronize command]
- 8 (A4.4) [Padding, if necessary]
- 9
- 10 (A5.0) [packet 61]
- 11 (A5.1) [frame 1, time = 0:00:07, type = mp3 audio]
- 12 (A5.2) [frame 2, time = 0:00:08, type = mp3 audio]
- 13 (A5.3) [frame 3, time = 0:00:09, type = mp3 audio]
- 14 (A5.4) [frame 4, time = 0:00:10, type = mp3 audio]
- 15
- 16 (A6.0) [packet 62]
- 17 (A6.1) [continuation of frame 4]
- 18 (A6.2) [frame 5, time = 0:00:11, type = mp3 audio]
- 19 (A6.3) [frame 6, time = 0:00:12, type = mp3 audio]
- 20 (A6.4) [frame 7, time = 0:00:13, type = mp3 audio]

21 Packet Sequence A comprises a sequence of six packets, identified by packet 57 through
 22 packet 62, that the audio information channel device 23 multi-casts in respective messages to the
 23 members of a synchrony group 20. It will be appreciated that the series of messages that the audio
 24 information channel device 23 may multi-cast to the synchrony group 20 may include a messages
 25 prior to the packet 57, and may also include messages after packet 62. Each packet comprises a
 26 packet header, which is symbolized by lines (A1.0), (A2.0),...(A6.0) in Packet Sequence A, and will
 27 generally also include information associated with at least a portion of a frame. In the packets
 28 represented in Packet Sequence A, each packet includes information associated with a plurality of

frames. Depending on the lengths of the packets, each packet may contain information associated with a portion of a frame, an entire frame, or multiple frames. In the illustration represented by Packet Sequence A, it is assumed that each packet may contain information associated with multiple frames. In addition, it is assumed that a packet does not necessarily contain information associated with an integral number of frames; in that case, a packet may contain information associated with a portion of a frame, and the next packet will contain the information associated with the rest of a frame.

The frames and associated header playback timing information contained in the various packets are symbolized by lines (A1.1), (A1.2),..., (A1.4), (A2.1),..., (A6.4) of Packet Sequence A. Thus, for example, line (A1.2) of packet 57 represents the one-hundredth frame, that is, frame 51(100) (reference FIG. 4), of the track whose audio information is being transmitted in the sequence of packets that includes packet 57. The frame 51(100) is to be played at an illustrative time, according to the audio information channel device's digital to analog converter clock, of "time = 0:00:01," and the frame is encoded and/or compressed using the well-known MP3 encoding and compression methodology. In that case, the legend "time = 0:00:01" represents the time stamp that would be included in field 60 (FIG. 4) of the header associated with the frame 50(100) as multi-cast by the audio information channel device for the synchrony group. It will be appreciated that the playback time and encoding/compression methodology will be referred in the header 55(100) that is associated with the frame 51(100). It will also be appreciated that the header may also contain additional information as described above.

Similarly, line (A1.3) of packet 57 represents the one-hundred and first frame, that is, frame 51(101), of the track whose audio information is being transmitted in the sequence of packets that includes packet 57. The frame 51(101) is to be played at an illustrative time, according to the audio information channel device's digital to analog converter clock, of "0:00:02," and the frame is also encoded and/or compressed using the MP3 encoding and compression methodology. Line (A1.4) of packet 57 represents similar information, although it will be appreciated that, depending on the length of packet 57, the line may not represent the information for an entire frame 51(102) and/or its associated header. If the length of packet 57 is not sufficient to accommodate the information for

1 the entire frame 51(102) and/or associated header, the information will continue in packet 58, as
 2 represented by line (A2.1) in Packet Sequence A. Similarly, if the length of packet 56 was not
 3 sufficient to contain the information for an entire frame 51(99) preceding frame 51(100), packet 57
 4 (lines (A1.0) through 1.4) may contain any information from frame 51(99) that packet 56 was unable
 5 to accommodate.

6 As noted above, when the master device 21 or a slave device 22(g) in the synchrony group
 7 20 receives the packet 57, its respective network communications manager 40 will update the time
 8 stamps associated with the various frames 51(f) as described above before buffering the respective
 9 frames in the respective audio information buffer 31.

10 Packets 58 and 59 contain information that is organized along the lines described above in
 11 connection with packet 57.

12 Packet 60 also contains, as represented by lines (A4.1) and (A4.2), information that is
 13 organized along the lines of the information represented by lines (Ax.1) and (Ax.2) ("x" equals an
 14 integer) described above in connection with packets 57 through 59. On the other hand, packet 60
 15 contains a resynchronize command, as represented by line (A4.3). Packet 60 also may contain
 16 padding, as represented by line 4.4, following the resynchronize command. As noted above, the
 17 master device 21 of a synchrony group 20 will enable the audio information channel device 23 that
 18 is providing audio information to the synchrony group 20 to multi-cast a message containing the
 19 resynchronize command when it receives notification from the user interface module 13 that the user
 20 wishes to cancel playback of a track that is currently being played. In the example depicted in
 21 Packet Sequence A, as will be described below, the audio information channel device 23 receives
 22 notification from the master device 21 that the user wishes to cancel playback of a track at a time
 23 corresponding to "time = 0:00:07" according to its digital to analog converter clock 34, and, in line
 24 (A4.3) of packet 60 it will provide the resynchronize command, followed by padding, if necessary.

25 As will be apparent from examining lines (A3.1) through (A3.4) of packet 59 and lines
 26 (A4.1) and (A4.2) of packet 60, although the audio information channel device 23 has received the
 27 notification from the synchrony group's master device 21 to multi-cast the resynchronize command

1 at a time corresponding to "time = 0:00:07" according to the clock time indicated by its digital to
2 analog converter clock 34, it (that is, the audio information channel device 23) has already multi-cast
3 messages containing frames that are to be played at that time and subsequently. That is, the audio
4 information channel device 23 has, multi-cast in packet 59, frames 51(106) through 51(108) that
5 contain time stamps "time = 0:00:07," "time = 0:00:08" and "time = 0:00:09," respectively, and, in
6 packet 60, in addition to the continuation of frame 51(108), frame 51(109) that contains time stamp
7 "time = 0:00:10." (It will be appreciated that the times indicated by the illustrative time stamps are
8 for illustration purposes only, and that in an actual embodiment the time stamps may have different
9 values and differentials.)

10 As noted above, the audio information channel device 23 multi-casts a message containing
11 a packet that, in turn, contains the resynchronize command when it receives the notification from
12 the master device 21 to do so. In the example depicted in Packet Sequence A, the packet will be
13 multi-cast when the audio information channel device's digital to analog converter clock time
14 corresponds to "0:00:07." Subsequently, two things happen. In one, aspect, when the master device
15 21 and slave devices 22(g) receive the packet that contains the resynchronize command, they will
16 stop playing the audio program that they are playing.

17 In addition, the audio information channel device 23 will begin transmitting frames
18 containing audio information for the next track, including therewith time stamps immediately
19 following the digital to analog converter clock time at which the packet including the resynchronize
20 command was transmitted. Accordingly, and with further reference to Packet Sequence A, the audio
21 information channel device 23 will multi-cast a message containing packet 61. As indicated above,
22 packet 61 contains, as represented in lines (A5.1) through (A5.3), frames 51(1) through 51(3), which
23 are the first three frames of the next track of the audio program that is to be played. It is also
24 compressed and encoded using the MP3 encoding and compression scheme, and it is accompanied
25 by time stamps "time = 0:00:07," "time = 0:00:08" and "time = 0:00:10." As noted above, the time
26 stamp "time = 0:00:07" corresponds to the clock time at which the audio information channel device
27 23 multi-casts the resynchronize command, and, when the master device 21 and slave devices 22(g)
28 receive these frames, they would be expected to begin playing them very shortly, if not immediately

after the audio information channel device 23 multi-casts the message containing the packet that, in turn, contains the resynchronize command. Packet 61 also includes at least a portion of the next frame, that is, frame 51(4), for that track. In addition, Packet Sequence A depicted above further includes a subsequent packet, namely, packet 62, that contains any necessary continuation of frame 51(4), as well as three subsequent frames. If any additional packets are required for the track, as well as for subsequent tracks, they can be multi-cast in a similar manner.

As further noted above, the resynchronize command can also be used to cancel playing of one or more tracks for which playback has begun. This will be illustrated in connection with Packet Sequence B:

Packet Sequence B

(B1.0) [packet 157]

(B1.1) [continuation of frame 99]

(B1.2) [frame 100, time = 0:00:01, type = mp3 audio]

(B1.3) [frame 101, time = 0:00:02, type = mp3 audio]

(B1.4) [frame 102, time = 0:00:03, type = mp3 audio]

(B2.0) [packet 158]

(B2.1) [continuation of frame 102]

(B2.2) [frame 103, time = 0:00:04, type = mp3 audio]

(B2.3) [frame 104, time = 0:00:05, type = mp3 audio]

(B2.4) [frame 105, time = 0:00:06, type = mp3 audio]

(B3.0) [packet 159]

(B3.1) [continuation of frame 105]

(B3.2) [frame 106, time = 0:00:07, type = mp3 audio]

(B3.3) [track boundary notification]

(B3.4) [Padding, if necessary]

- 1
- 2 (B4.0) [packet 160]
- 3 (B4.1) [frame 1, time = 0:00:08, type = mp3 audio]
- 4 (B4.2) [frame 2, time = 0:00:09, type = mp3 audio]
- 5 (B4.3) [frame 3, time = 0:00:10, type = mp3 audio]
- 6
- 7 (B5.0) [packet 161]
- 8 (B5.1) [continuation of frame 3]
- 9 (B5.2) [frame 4, time = 0:00:11, type = mp3 audio]
- 10 (B5.3) [Resynchronize, after packet 159]
- 11 (B5.4) [Padding, if necessary]
- 12
- 13 (B6.0) [packet 162]
- 14 (B6.1) [frame 1, time = 0:00:08, type = mp3 audio]
- 15 (B6.2) [frame 2, time = 0:00:09, type = mp3 audio]
- 16 (B6.3) [frame 3, time = 0:00:10, type = mp3 audio]
- 17 (B6.4) [frame 4, time = 0:00:11, type = mp3 audio]
- 18
- 19 (B7.0) [packet 163]
- 20 (B7.1) [continuation of frame 4]
- 21 (B7.2) [frame 5, time = 0:00:12, type = mp3 audio]
- 22 (B7.3) [frame 6, time = 0:00:13, type = mp3 audio]
- 23 (B7.4) [frame 7, time = 0:00:14, type = mp3 audio]

24 Packet Sequence B comprises a series of seven packets, identified by packet 157 through 163
 25 , that the audio information channel device 23 multi-casts to the members of a synchrony group 20.
 26 As with Packet Sequence A, it will be appreciated that the series of packets that the audio
 27 information channel device 23 may multi-cast to the synchrony group 20 may include packets prior
 28 to the packet 157, and may also include packets after packet 162. Each packet comprises a packet

header, which is symbolized by lines (B1.0), (B2.0),...(B7.0) in Packet Sequence B. As in Packet Sequence A, each packet will also generally include information associated with at least a portion of a frame 51(f) along with its associated frame 55(f). As in the packets represented in Packet Sequence A, each packet includes information associated with a plurality of frames. Depending on the lengths of the packets, each packet may contain information associated with a portion of a frame, an entire frame, or multiple frames. Further, as with Packet Sequence A, it is assumed that each packet may contain information associated with multiple frames. In addition, it is assumed that a packet does not necessarily contain information associated with an integral number of frames; in that case, a packet may contain information associated with a portion of a frame, and the next packet will contain the information associated with the rest of a frame.

The structures of the packets represented by Packet Sequence B are similar to those described above in connection with Packet Sequence A, and will not be repeated here. Generally, Packet Sequence B illustratively contains a sequence of packets that represent at least portions of three tracks that may have been selected from, for example, a play list. In particular, packets 157 through 159 represent frames from a portion of one track, packets 160 and 161 represent frames from a second track and packets 162 and 163 represent frames from a third track. The play list indicated that the first, second and third tracks were to be played in that order. With particular reference to Packet Sequence B, it should be noted that line (B3.3) indicates that packet 159 includes an indication that that packet contains the last frame for the track, and line (B3.4) provides for padding to the end of the packet. The first frame of the next track begins in packet 160.

In connection with the use of the resynchronize command to cancel playback of a track, at least a portion of which the audio information channel device 23 has multi-cast to the members of the synchrony group, packet 161, in line (B5.3) represents a resynchronize command that indicates that resynchronization is to occur after packet 159, that is, immediately after the packet that contains the last frame of the first of the three tracks represented by the packets in Packet Sequence B. It should be noted that the resynchronize command is in the packet 161, while the resynchronization is to occur at packet 160, that is, the synchrony group is to not play the track starting with packet 160, but instead is to begin playing the track frames for which begin with the next packet, that is,

1 packet 162. As with Packet Sequence A, in Packet Sequence B the audio information channel device
2 23, in packet 162 and 163, multi-casts frames whose time stamps indicate that they are to be played
3 when the frames that were multi-cast in packets 160 and 161 were to be played. By use of the
4 resynchronize command and specifying a packet in this manner, the audio information channel
5 device can cancel playback of a track for which playback has not yet begun.

6 It will be appreciated that the resynchronize command is generally not necessary for
7 cancelling play back of a track that the audio information channel device 23 has not started multi-
8 casting to the synchrony group 20, since the audio information channel device 23 itself can re-order
9 the play list to accommodate the cancellation.

10 The invention provides a number of advantages. In particular, the invention provides a
11 network audio system in which a number of devices share information can reproduce audio
12 information synchronously, notwithstanding the fact that packets, which may contain digital audio
13 information, transmitted over the network to the various zone players connected thereto may have
14 differing delays and the zone players operate with independent clocks. Moreover, although the
15 invention has been described in connection with audio information, it will be appreciated that the
16 invention will find utility in connection with any type of isochronous information for which
17 synchrony among devices is desired. The system is such that synchrony groups are created and
18 destroyed dynamically, and in such a manner as to avoid requiring a dedicated device as the master
19 device.

20 It will be appreciated that a number of changes and modifications may be made to the
21 network audio system 10 as described above. For example, although the invention has been
22 described as providing that the audio information channel device 23 provides digital audio
23 information to the members synchrony group 20 that has been encoded using particular types of
24 encoding and compression methodologies, it will be appreciated that the audio information channel
25 device 23 can provide digital audio information to various members of the synchrony group 20 that
26 have been encoded and compressed using different types of encoding and compression
27 methodologies, and, moreover, for which different sampling rates have been used. For example, the

1 audio information channel device 23 may provide digital audio information to the master device 21
 2 and slave devices 22(1) through 22(g_1) using the MP3 methodology at a specified sampling rate, the
 3 digital audio information for the same program to slave devices 22(g_1+1) through 22(g_2) using the
 4 WAV methodology at one specified sampling rate, and to slave devices 22(g_2+1) through 22(G)
 5 using the WAV methodology at another specified sampling rate. In that case, the audio information
 6 channel device 23 can specify the particular encoding and compression methodology that has been
 7 used in the encoding type field 57 associated with each frame and the sampling rate in the sampling
 8 rate field 58. Moreover, since the encoding and compression type and sampling rate are specified
 9 for each frame, the encoding and compression type and sampling rate can be changed from frame
 10 to frame. The audio information channel device 23 may use different multi-cast addresses for the
 11 different encoding and compression types and sampling rates, but it will be appreciated that that
 12 would not be required.

13 It will be appreciated that two advantages of providing that the encoding and compression
 14 methodology and the sampling rate is provided on a frame-by-frame basis, instead of on, for
 15 example, a track-by-track basis, is that that would facilitate a slave device joining the synchrony
 16 group 20 at a frame mid-track, without requiring, for example, the master device 21 or the audio
 17 information channel device 23 to notify it of the encoding and compression methodology or the
 18 sampling rate.

19 Another modification is that, instead of the network communications manager 40 of a
 20 member of a synchrony group 20 generating the updated time stamp T_F^u for a digital audio
 21 information frame by adding the time differential value ΔT to the time stamp T_F associated with a
 22 frame, the network communications manager 40 may instead generate the updated time stamp T_F^u
 23 by subtracting the differential time value ΔT from the member's current time T_s as indicated by the
 24 member's digital to analog converter clock 34 at the time at which the digital audio information is
 25 received. It will be appreciated, however, that there may be variable time delays in processing of
 26 messages by the slave device's network communications manager 40, and so it may be preferable
 27 to generate the time differential value ΔT using the time stamp T_F provided by the audio information
 28 channel device 23.

1 In addition, instead of the network communications manager 40 of a member of a synchrony
 2 group generating an updated time stamp to reflect the difference between the times indicated by the
 3 member's digital to analog converter clock and the audio information channel device's digital to
 4 analog converter clock, the network communications manager 40 can generate the time differential
 5 value ΔT and provide it to the member's playback scheduler 32. In that case, the member's network
 6 communications manager 40 can store each digital audio information frame along with the time
 7 stamp T_F as received from the master device in the audio information buffer 21. The playback
 8 scheduler 32 can utilize the time differential value ΔT , and the time stamps T_F associated with the
 9 digital audio information frames, to determine when the respective digital audio information frames
 10 are to be played. In determining when a digital audio information frame is to be played, the
 11 playback scheduler can add the time differential value to the time stamp T_F associated with the
 12 digital audio frame, and enable the digital audio frame to be coupled to the digital to analog
 13 converter 33 when the time indicated by the sum corresponds to the current time as indicated by the
 14 slave device's digital to analog converter clock 34. Alternatively, when the member's digital to
 15 analog converter clock 34 updates its current time T_s , the playback scheduler can generate an updated
 16 current time T'_s by subtracting the differential time value ΔT from the current time T_s , and using the
 17 updated current time T'_s to determine when to play a digital audio information frame.

18 As described above, the members of a synchrony group 20 periodically obtain the audio
 19 information channel device's current time value and uses the current time value that it receives from
 20 the audio information channel device to periodically update the time differential value ΔT that it uses
 21 in updating the time stamps associated with the various frames. It will be appreciated that, if the
 22 digital to analog converter clock(s) associated with the member(s) of a synchrony group 20 are
 23 ensured to have the same rate as the digital to analog converter clock, a member need only obtain
 24 the current time value from the audio information channel device once, at the beginning of playback.

25 As another alternative, if the zone players are provided with digital to analog converter clock
 26 34 whose time and rate can be set by an element such as the network communications manager 40,
 27 when a zone player 11(n) is operating as a member of a synchrony group 20, its network
 28 communications manager 40 can use the various types of timing information that it receives from

the audio information channel device 23, including the current time information and the playback timing information indicated by the time stamps that are associated with the various frames 51(f) comprising the audio and playback timing information that it receives, to adjust the synchrony group member's digital to analog converter clock's time value and/or the clock rate that it uses for playback. If the clock's time value is to be adjusted, when the synchrony group member's network communications manager 40 initially receives the current time information from the audio information channel device 23 for the synchrony group 20, the network communications manager 40 can set the synchrony group member's digital to analog converter clock 34 to the current time value as indicated by the audio information channel device's current time information. The network communications manager 40 can set the clock 34 to the current time value indicated by the audio information channel device's current time information once, or periodically as it receives the current time information.

Alternatively or in addition, the synchrony group member's network communications manager 40 can use one or both of the current time information and/or the playback timing information in the time stamps associated with the respective frames 51(f) to adjust the clock rate of the clock 34 that it uses for playback. For example, when the synchrony group member's network communications manager 40 receives a frame 51(f_x) having a time stamp having a time value T_{f_x} , it can generate the updated time value $T_{f_x}^U = T_{f_x} + \Delta T$ as described above, and store the frame with the time stamp with the updated time value in the audio information buffer 30. In addition, since both the number of samples in a frame and the sampling rate, which determines the rate at which the frame is to be played, are known to the network communications manager 40, it can use that information, along with the updated time value $T_{f_x}^U$ that is to be used for frame 51(f_x) to generate an expected updated time value $T_{f_{x+1}}^E$ that is expected for the updated time stamp of the next frame 51(f_{x+1}). After the synchrony group member's network communications manager 40 receives the next frame 51(f_{x+1}), it can generate the updated time value $T_{f_{x+1}}^U$ and compare that

1 value to the expected updated time value $T^E_{f_{x+1}}$. If the two time values do not correspond, or if
 2 the difference between them is above a selected threshold level, the clock that is used by the audio
 3 information channel device 23 to generate the time stamps is advancing at a different rate than the
 4 synchrony group member's digital to analog converter clock 34, and so the network communications
 5 manager 40 can adjust the rate of the digital to analog converter clock 34 to approach that of the
 6 clock used by the audio information channel device 23 so that the differential time value ΔT is
 7 constant. On the other hand, if the two time values do correspond, then the time differential value
 8 ΔT is constant, or the difference is below a threshold level, and the network communications
 9 manager 40 need not change the clock rate of the digital to analog converter clock 34. It will be
 10 appreciated that, if the clock rate is to be adjusted, the rate adjustment can be fixed, or it can vary
 11 based on, for example, the difference between the updated time value $T^U_{f_{x+1}}$ and the expected
 12 updated time value $T^E_{f_{x+1}}$.

13 It will also be appreciated that, if no rate adjustment is performed for one frame $51(f_{x+1})$, the
 14 synchrony group member's network communications manager 40 can generate an expected updated
 15 time value $T^E_{f_{x+2}}$ that is expected for the updated time stamp of the next frame $51(f_{x+2})$ using the
 16 updated time value $T^U_{f_x}$ determined for frame $51(f_x)$, along with the number of samples in a frame
 17 and the sampling rate, and compare the expected updated time value $T^E_{f_{x+2}}$ to the updated time
 18 value $T^U_{f_{x+2}}$ that it generates when it receives frame $51(f_{x+2})$. At that point, if the network
 19 communications manager 41 determines that two time values do not correspond, or if the difference
 20 between them is above a selected threshold level, it can adjust the rate of the digital to analog
 21 converter clock 34. Similar operations can be performed if no rate adjustment is performed for
 22 several successive frames $51(f_{x+1})$, $51(f_{x+2})$,.... This will accommodate the possibility that the rate
 23 differential between the clock 34 and the clock used by the audio information channel device 23 in

1 generating the time stamps have rates that differ by an amount sufficiently small that it cannot be
2 detected using time stamps of two or more successive frames.

3 Instead or in addition to adjusting the clock rate as described above, the synchrony group
4 member's network communications manager 40 can perform similar operations in connection with
5 adjusting the clock rate in connection with the current time information that it receives from the
6 audio information channel device 23.

7 Furthermore, although the network audio system 10 has been described such that the master
8 device 21 of a synchrony group 20 can, in response to control information provided thereto by a user
9 through the user interface module 13, provide a notification to a zone player 11(n) that it is to
10 become a member of its synchrony group 20 as a slave device 22(g), it will be appreciated that the
11 user interface module 13 can provide the notification directly to the zone player 11(n) that is to
12 become a member of the synchrony group 20. In that case, the zone player 11(n) can notify the
13 master device 21 that it is to become a slave device 22(g) in the synchrony group 20, after which the
14 master device 21 can provide information regarding the synchrony group 20, including the multi-cast
15 and unicast addresses of the audio information channel device and other information as described
16 above.

17 Similarly, although the network audio system 10 has been described such that the master
18 device 21 of a synchrony group 20 can, in response to control information provided thereto by a user
19 through the user interface module 13, provide a command to a slave device 22(g) to enable the slave
20 device 22(g) to adjust its volume, it will be appreciated that the user interface module 13 can provide
21 control information directly to the slave device 22(g) to enable the slave device 22(g) to adjust its
22 volume.

23 In addition, although the network audio system 10 has been described such that each frames
24 51(f) is associated with a frame sequence number (reference field 56, FIG. 4), it will be appreciated
25 that, if the packets described above in connection with Packet Sequence A and Packet Sequence B
26 are provided with packet sequence numbers, the frame sequence numbers need not be provided,
27 since the packet sequence numbers can suffice for defining the frame sequencing.

1 Furthermore, although the network audio system 10 has been described such that the zone
2 players 11(n) are provided with an audio amplifier 35 for amplifying the analog signal provided by
3 the respective digital to analog converters 33, it will be appreciated that a zone player may be
4 provided that does not itself include an audio amplifier. In that case, the analog signal may be
5 coupled to an external amplifier for amplification as necessary before being provided to the audio
6 reproduction device(s) 15(n)(r). It will be appreciated that a single zone player 11(n) may be
7 provided with multiple audio amplifiers and audio reproduction device interfaces, and, if necessary,
8 multiple digital to analog converters 33, to provide audio programs for corresponding numbers of
9 synchrony groups.

10 Similarly, although the zone players 11(n) have been described such that they may be
11 connected to one or more audio information sources, it will be appreciated that an audio information
12 source may form part of and be integrated into a zone player 11(n). For example, a zone player may
13 include a compact disk player, cassette tape player, broadcast radio receiver, or the like, that has been
14 integrated into it. In addition, as noted above, an individual zone player 11(n) may be connected to
15 multiple audio information sources and may contemporaneously operate as the audio information
16 channel device 23 for multiple synchrony groups .

17 In addition, although FIG. 1 shows the network audio system 10 as including one user
18 interface module 13, it will be appreciated that the system 10 may include a plurality of user
19 interface modules. Each user interface module be useful for controlling all of the zone players as
20 described above, or alternatively one or more of the user interface modules may be useful for
21 controlling selected subsets of the zone players.

22 Moreover, it will be appreciated that, although the invention has been described in connection
23 with audio information, it will be appreciated that the invention will find utility in connection with
24 any type of information for which synchrony among devices connected to a network is desired.

25 As noted above, while a zone player 11(n) is operating as audio information channel device
26 23 for a synchrony group 20, when the zone player 11(n)'s audio information source interface 30 or
27 network communications manager 40 stores digital audio information frames based on audio

1 information from an audio information source 14(n)(s) in the audio information buffer 31, it will
2 provide time stamps for the respective frames to schedule them for playback after some time delay
3 after they have been buffered in the audio information buffer 31. The delay is provided so that, for
4 other zone players 11(n'), 11(n''),... that are operating as members of a synchrony group, there will
5 be sufficient time for the audio and playback timing information to be transferred over the network
6 12 to those other zone players 11(n'), 11(n''),... so that it can be processed and played by them at the
7 appropriate time as described above. The time period that is selected for the time delay may be fixed
8 or variable, and in either case may be based on a number of factors. If the time period selected for
9 the time delay is fixed, it may be based on, for example, factors such as an estimate of the maximum
10 latency in the network 12, the estimated maximum loading of the various components comprising
11 the zone players 11(n), and other estimates as will be appreciated by those skilled in the art.

12 The time delay may be the same for audio information from all types of audio information
13 sources, and may be constant over the entire period that the synchrony group 20 is playing an audio
14 work. Alternatively, different time delays may be utilized based on various criteria. For example,
15 if the audio information is to be played independently of information associated with other types of
16 media, the time delay may be selected to be relatively long, on the order of a significant fraction of
17 a second, or longer. On the other hand, if the audio information is to be played contemporaneously
18 with, for example, video information, which may be supplied by, for example, a video disk, video
19 tape cassette, over cable, satellite, or broadcast television, which may not be buffered or which may
20 be displayed independently of the network audio system 10, it may be undesirable to provide for
21 such a lengthy delay, since the time delay of the audio playback, in relation to the video display, may
22 be noticeable. In that case, the zone player 11(n) may provide for a much shorter time delay. In one
23 embodiment, the time delay provided for audio information to be played concurrently with video
24 information is selected to be generally on the order of fifty milliseconds, which would barely, if at
25 all, be perceptible to someone viewing the video. Other desirable time delays for information from
26 other types of sources will be apparent to those skilled in the art.

27 As yet a further possibility, the zone player 11(n), when operating as an audio information
28 channel device 23 for a synchrony group 20, can dynamically determine the time delay based on a

1 number of conditions in the network audio system 10, including, for example, the message transfer
2 latency in network 12, the loading of microprocessors or other components that are used in the
3 various zone players 11(n'), 11(n''),... that may comprise a synchrony group 20, as well as other
4 factors. For example, if the audio information channel device 23 determines that the latency in the
5 network 12 has increased beyond a selected threshold, the audio information channel device 23 can
6 adjust the delay to increase the likelihood that the members of the synchrony group 20 will be able
7 to receive the packets and process the frames so that they will be able to play them at the appropriate
8 times. Similarly, if the audio information channel device 23 is notified that a member of the
9 synchrony group 20 to which it provides audio information requires additional time to receive and
10 process the frames that it transmits, the audio information channel device 23 can adjust the delay
11 accordingly. It will be appreciated that, to reduce or minimize possible discontinuities in the audio
12 playback by the members of the synchrony group, the audio information channel device 23 can,
13 instead of adjusting the time delay during a particular audio track, adjust the time delay between
14 tracks, during silent periods of a track or otherwise as will be appreciated by those skilled in the art.
15 In addition, the audio information channel device 23 can use conventional audio compression
16 methodologies to facilitate a speeding up and/or slowing down of playback of an audio track while
17 it is in the process of providing additional time delay. Generally, the members of the synchrony
18 group 20 can provide notifications to the audio information channel device 23 if they determine that
19 they will need an additional time delay, and the audio information channel device 23 can adjust the
20 time delay in accordance with the notifications from the members of the synchrony group 20.

21 It will be appreciated that a system in accordance with the invention can be constructed in
22 whole or in part from special purpose hardware or a general purpose computer system, or any
23 combination thereof, any portion of which may be controlled by a suitable program. Any program
24 may in whole or in part comprise part of or be stored on the system in a conventional manner, or it
25 may in whole or in part be provided in to the system over a network or other mechanism for
26 transferring information in a conventional manner. In addition, it will be appreciated that the system
27 may be operated and/or otherwise controlled by means of information provided by an operator using
28 operator input elements (not shown) which may be connected directly to the system or which may

1 transfer the information to the system over a network or other mechanism for transferring
2 information in a conventional manner.

3 The foregoing description has been limited to a specific embodiment of this invention. It will
4 be apparent, however, that various variations and modifications may be made to the invention, with
5 the attainment of some or all of the advantages of the invention. It is the object of the appended
6 claims to cover these and such other variations and modifications as come within the true spirit and
7 scope of the invention.

8 What is claimed as new and desired to be secured by Letters Patent of the United States is:

1. A method for providing synchronized audio play back amongst a plurality of audio devices, the method comprising:

retrieving, by a computing device, audio information from any of a locally stored digital audio file, a remotely stored digital audio file via a wide area network, and a streaming digital audio service;

transmitting a plurality of frames to a plurality of audio devices over a local network from the computing device, each frame of the plurality of frames to include the retrieved audio information and timing information inserted at the computing device, wherein the timing information is to be used to play the audio information by the plurality of audio devices in synchrony.

2. The method of claim 1, wherein the computing device comprises a personal computer.

3. The method of claim 2, further comprising transmitting the plurality of frames to the plurality of audio devices responsive to receiving a user command via a user interface of the personal computer.

4. The method of claim 1, wherein the computing device comprises a mobile computing device.

5. The method of claim 4, further comprising transmitting the plurality of frames to the plurality of audio devices responsive to receiving a user command via a user interface of the mobile computing device.

6. The method of claim 1, wherein the computing device is configured to play the audio information in synchrony with the plurality of audio devices.

7. The method of claim 1, wherein the computing device is configured to play different audio information than the plurality of audio devices and at the same time as the plurality of audio devices play the audio information.

8. The method of claim 1, wherein the locally stored digital audio file is stored on a personal computer or a mobile computing device configured to store digital information.

9. The method of claim 1, further comprising wirelessly transmitting the plurality of frames to the plurality of audio devices.

10. The method of claim 1, wherein the computing device comprises a user interface to receive a user command to transmit the plurality of frames.

11. The method of claim 10, wherein the user interface is to enable a user to at least one of select an audio track, display status information, and display an upcoming audio playback selection.

12. The method of claim 10, wherein the user interface is to enable a user to select one or more of the plurality of audio devices for audio playback.

13. A non-transitory computer readable medium having stored therein instructions executable by a processor to perform a method comprising:

retrieving, by a computing device, audio information from any of a locally stored digital audio file, a remotely stored digital audio file via a wide area network, and a streaming digital audio service;

transmitting a plurality of frames to a plurality of audio devices over a local network from the computing device, each frame of the plurality of frames to include the retrieved audio information and timing information inserted at the computing device, wherein the timing information is to be used to play the audio information by the plurality of audio devices in synchrony.

14. A method for providing synchronized audio amongst a plurality of audio devices, the method comprising:

receiving, by an audio device, a plurality of frames over a local network from a computing device, each frame of the plurality of frames to include audio information and timing information, wherein the timing information is to be used to play the audio information by the audio device in synchrony with another receiving audio device; and

converting, by the audio device, the audio information from a digital form into an analog signal to be output based on the timing information via an amplifier and a speaker driver.

15. The method of claim 15, wherein the computing device comprises a mobile computing device.

16. The method of claim 15, wherein the computing device comprises a personal computer.

17. The method of claim 15, wherein the audio information is from any of a locally stored digital audio file, a remotely stored digital audio file via a wide area network, and a streaming digital audio service.

18. The method of claim 15, wherein the local network comprises a wired network.

19. The method of claim 15, wherein the local network comprises a wireless network.

20. A non-transitory computer readable medium having stored therein instructions executable by a processor to perform a method comprising:

receiving, by an audio device, a plurality of frames over a local network from a computing device, each frame of the plurality of frames to include audio information and timing information, wherein the timing information is to be used to play the audio information by the audio device in synchrony with another receiving audio device; and

converting, by the audio device, the audio information from a digital form into an analog signal to be output based on the timing information via an amplifier and a speaker driver.

ABSTRACT OF THE DISCLOSURE

A system is described for maintaining synchrony of operations among a plurality of devices that have independent clocking arrangements. The system includes a task distribution device that distributes tasks to a synchrony group comprising a plurality of devices that are to perform the tasks distributed by the task distribution device in synchrony. The task distribution device distributes each task to the members of the synchrony group over a network. Each task is associated with a time stamp that indicates a time, relative to a clock maintained by the task distribution device, at which the members of the synchrony group are to execute the task. Each member of the synchrony group periodically obtains from the task distribution device an indication of the current time indicated by its clock, determines a time differential between the task distribution device's clock and its respective clock and determines therefrom a time at which, according to its respective clock, the time stamp indicates that it is to execute the task.

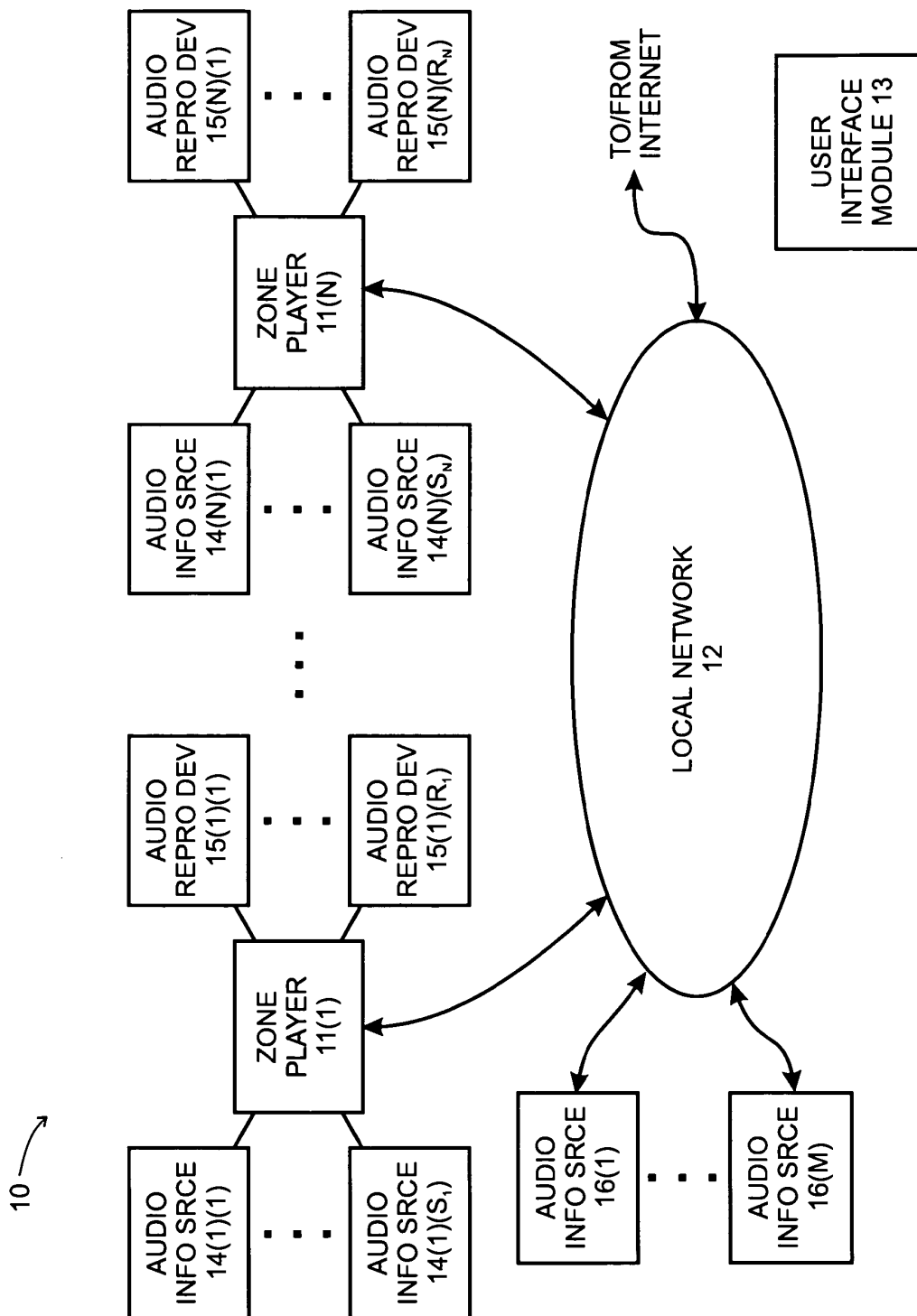


FIG. 1

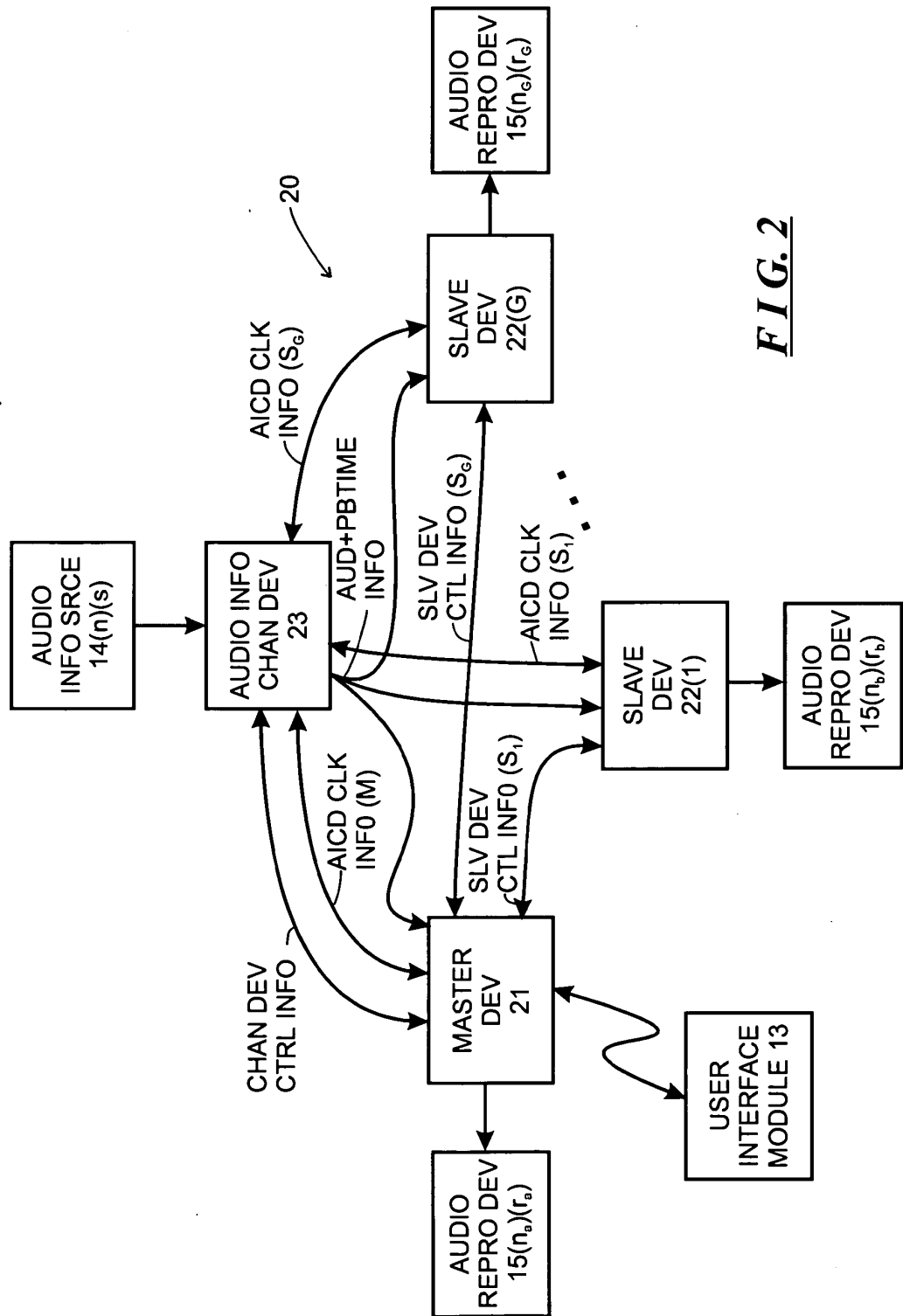


FIG. 2

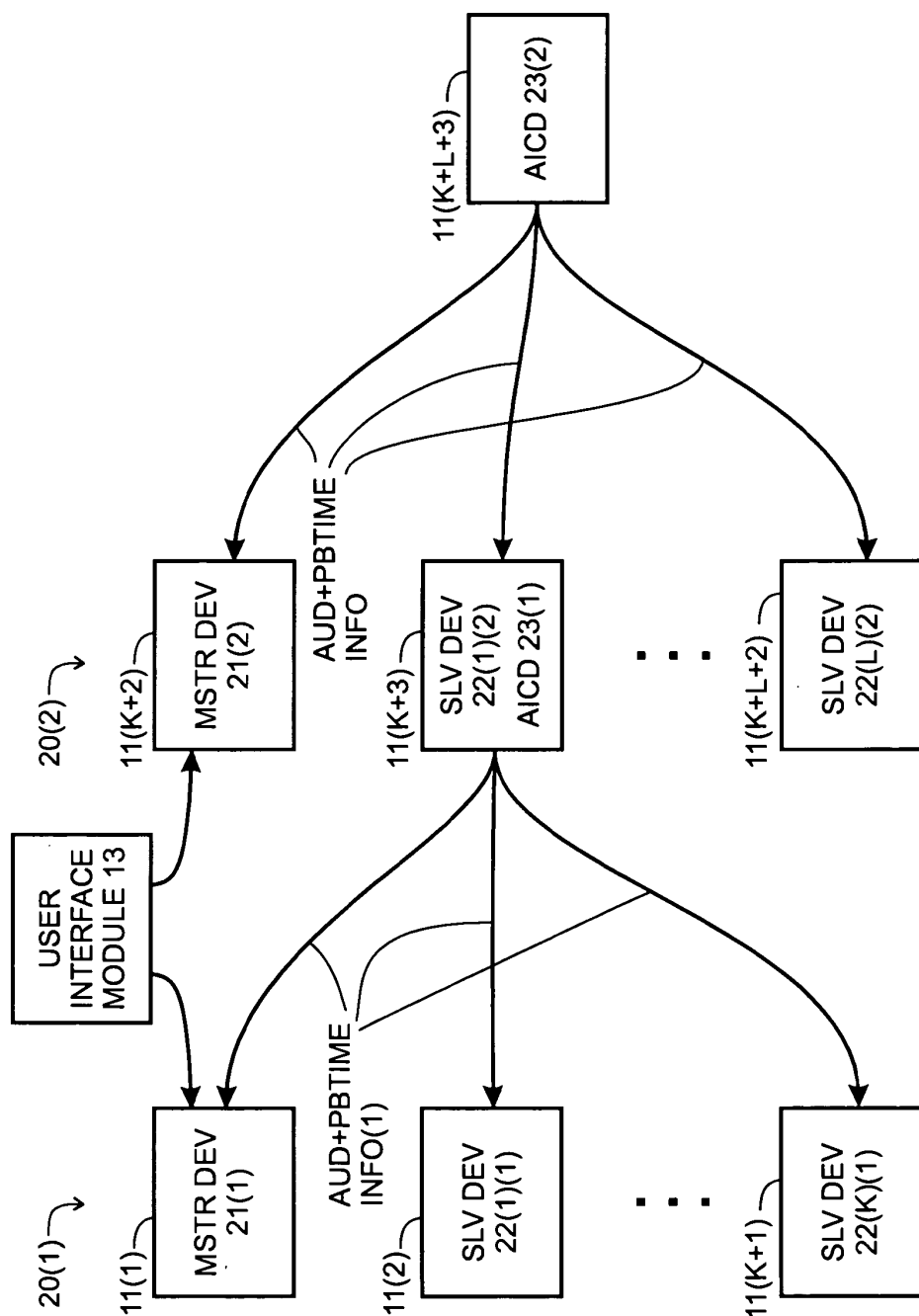


FIG. 2A

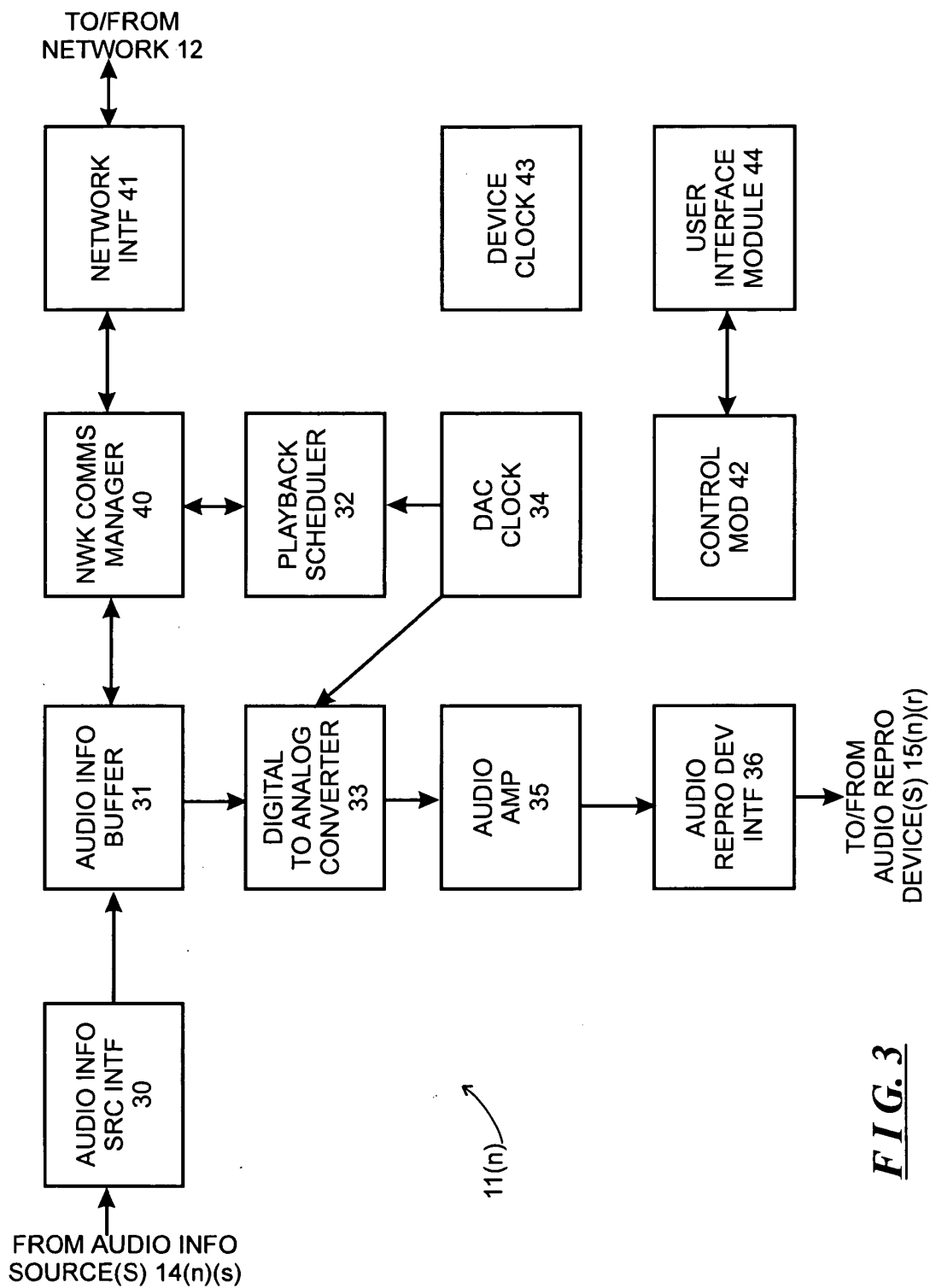


FIG. 3

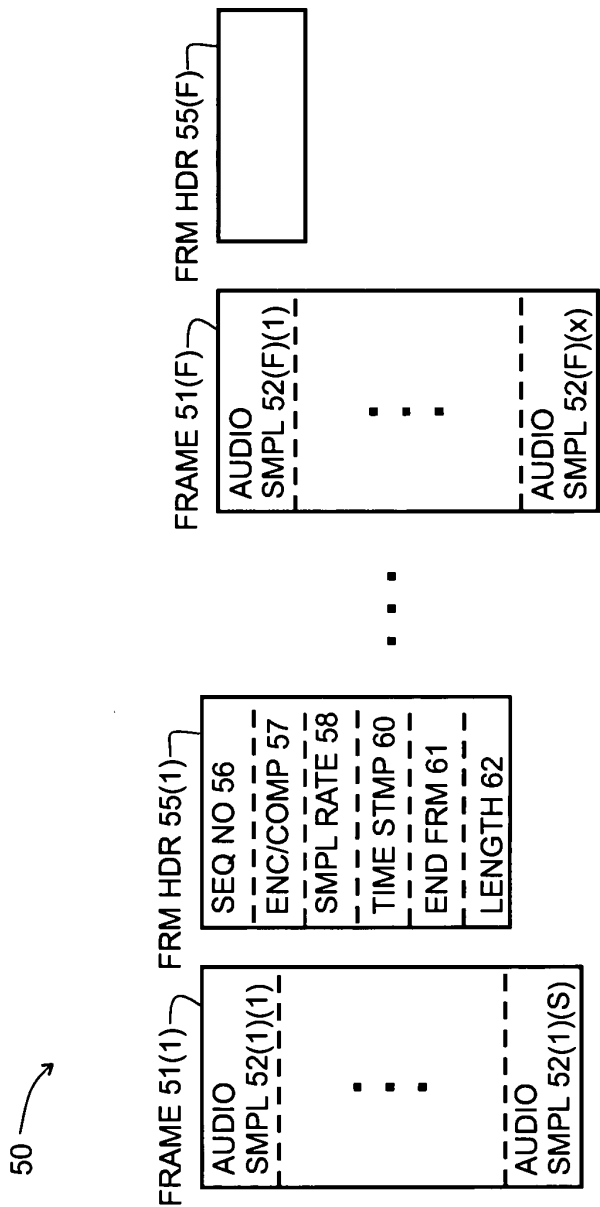


FIG. 4