



US 20030208621A1

(19) **United States**(12) **Patent Application Publication**
Bowman(10) **Pub. No.: US 2003/0208621 A1**(43) **Pub. Date: Nov. 6, 2003**(54) **PATH OPTIMIZER FOR PEER TO PEER NETWORKS**(22) Filed: **May 6, 2002****Publication Classification**(75) Inventor: **Don Bowman, Waterloo (CA)**

Correspondence Address:
BERESKIN AND PARR
SCOTIA PLAZA
40 KING STREET WEST-SUITE 4000 BOX
401
TORONTO, ON M5H 3Y2 (CA)

(73) Assignee: **Sandvine Incorporated, Waterloo (CA)**(21) Appl. No.: **10/138,336**(51) **Int. Cl.⁷ G06F 15/173; G06F 15/16**(52) **U.S. Cl. 709/242; 709/202**(57) **ABSTRACT**

The invention disclosed is a path optimizer for Peer to Peer (P2P) networks. Large amounts of P2P file exchanges increase the costs to maintain a network. Network service providers typically pay for the amount of data they exchange with other network service providers. The present invention examines P2P messages between and within networks and provides a cost efficient path to handle each P2P message.

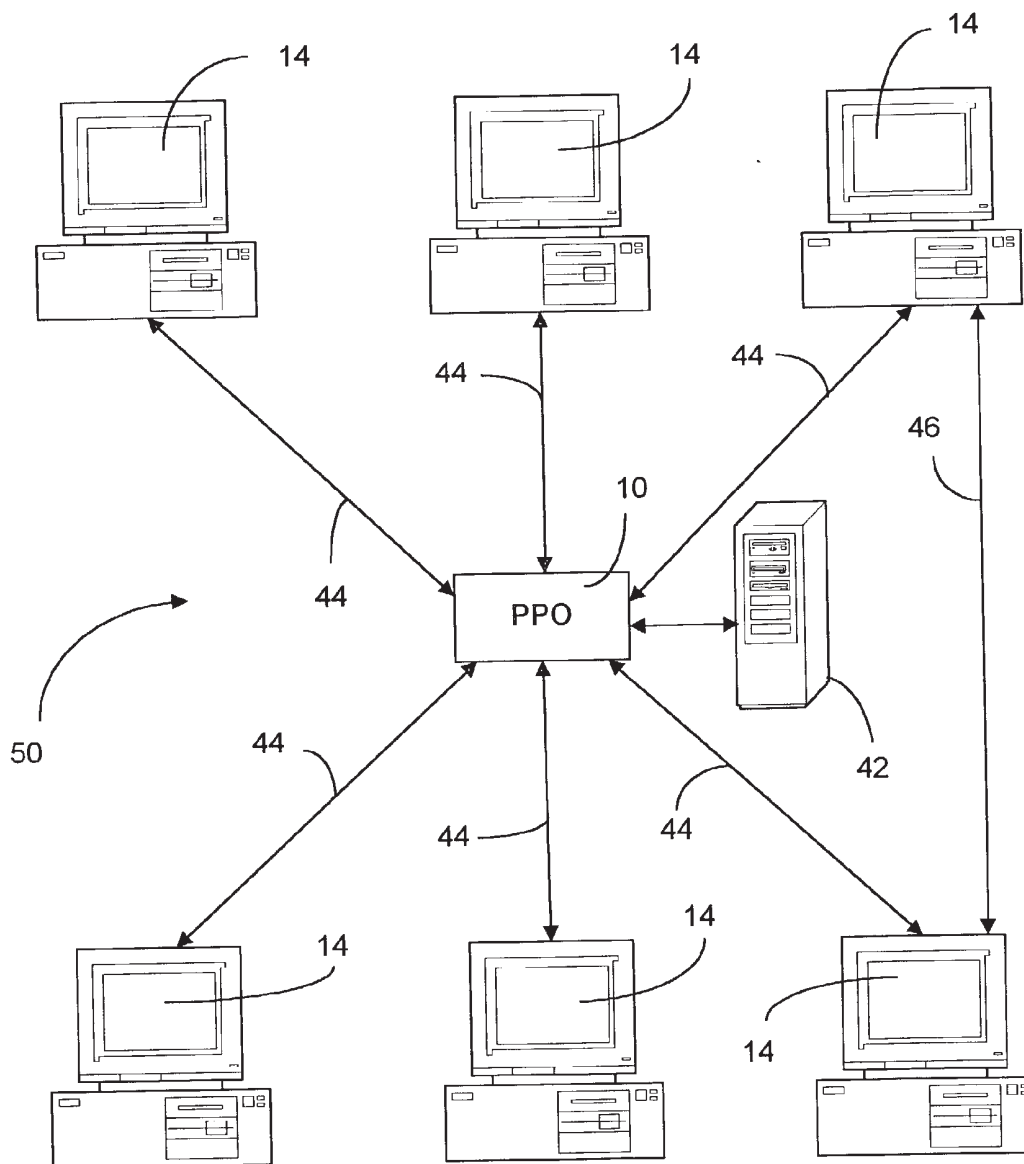


FIG. 1

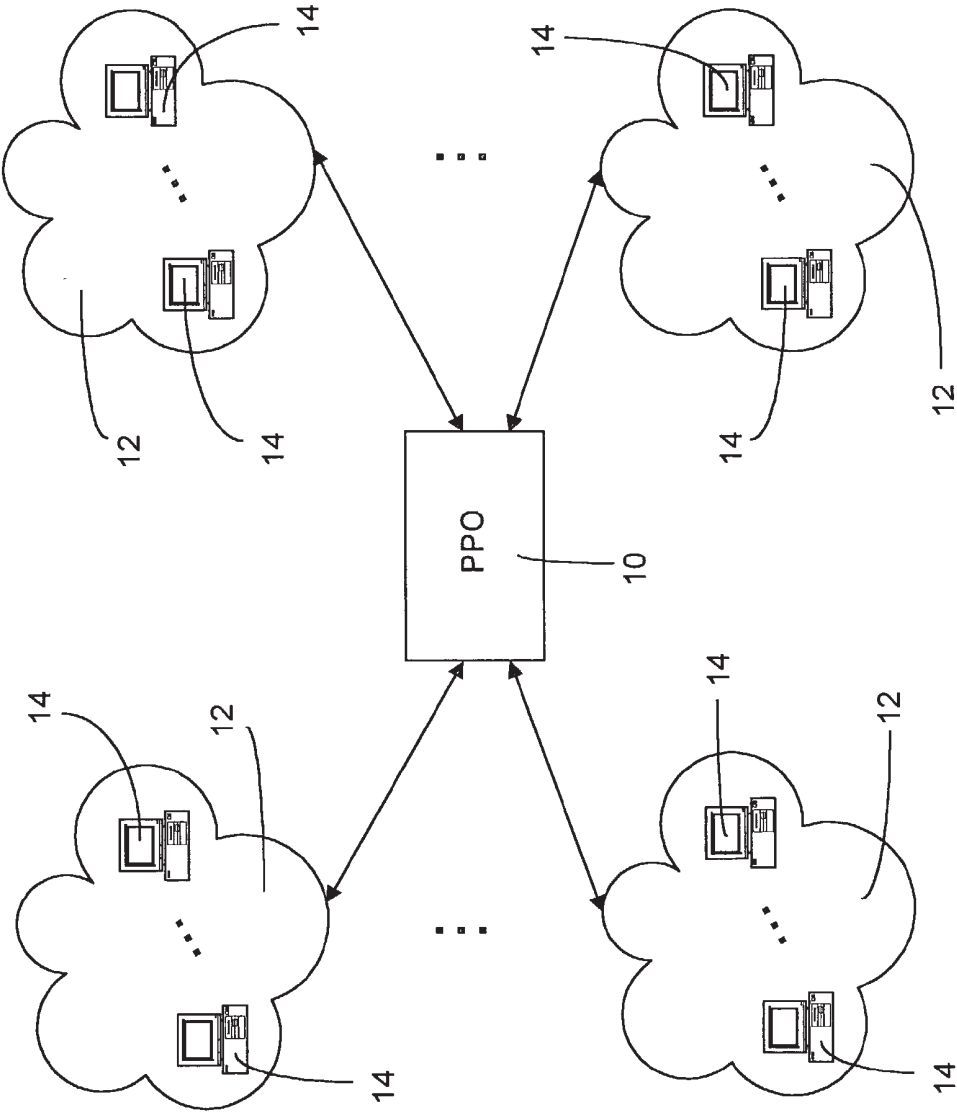


FIG. 2

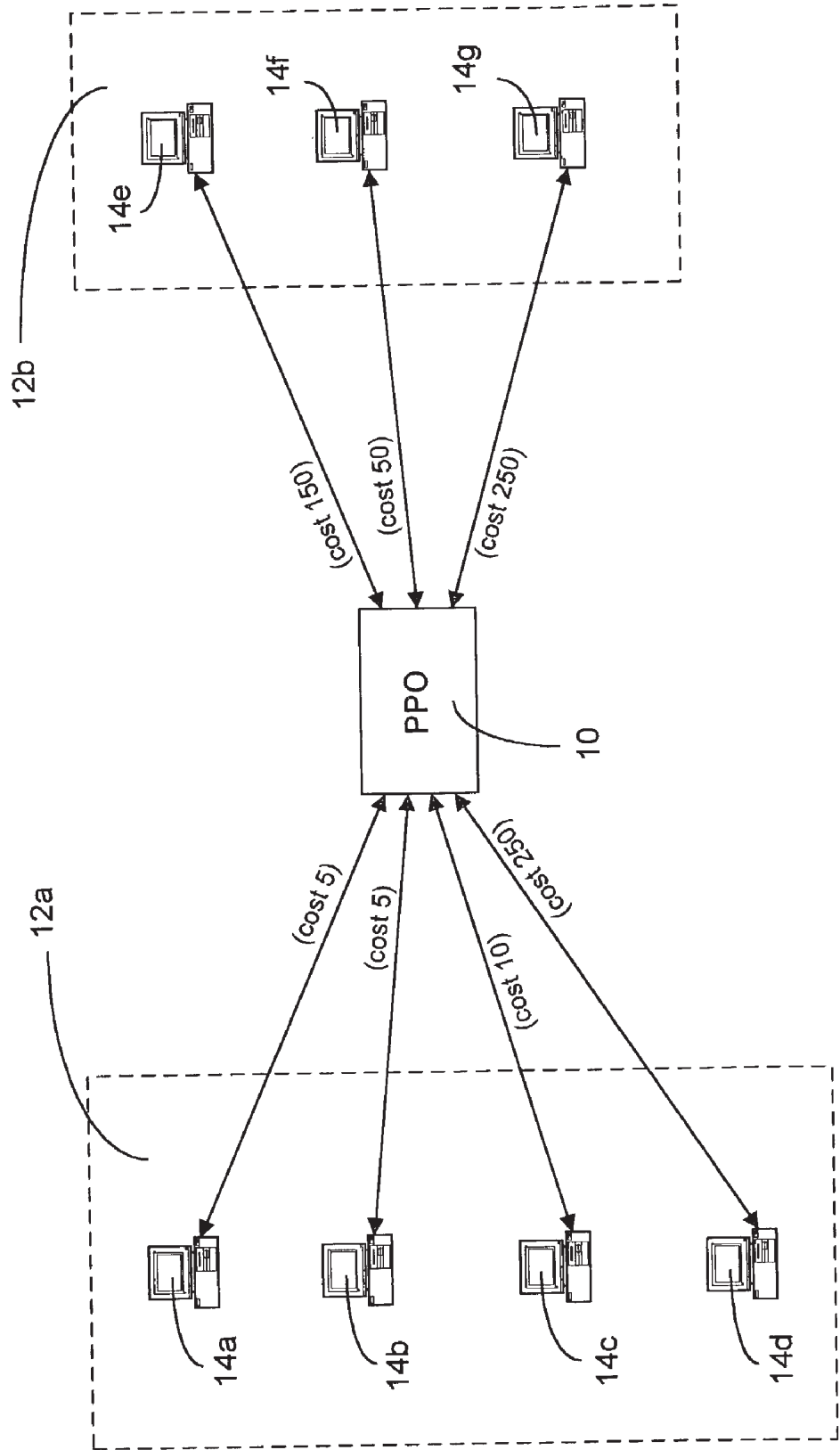
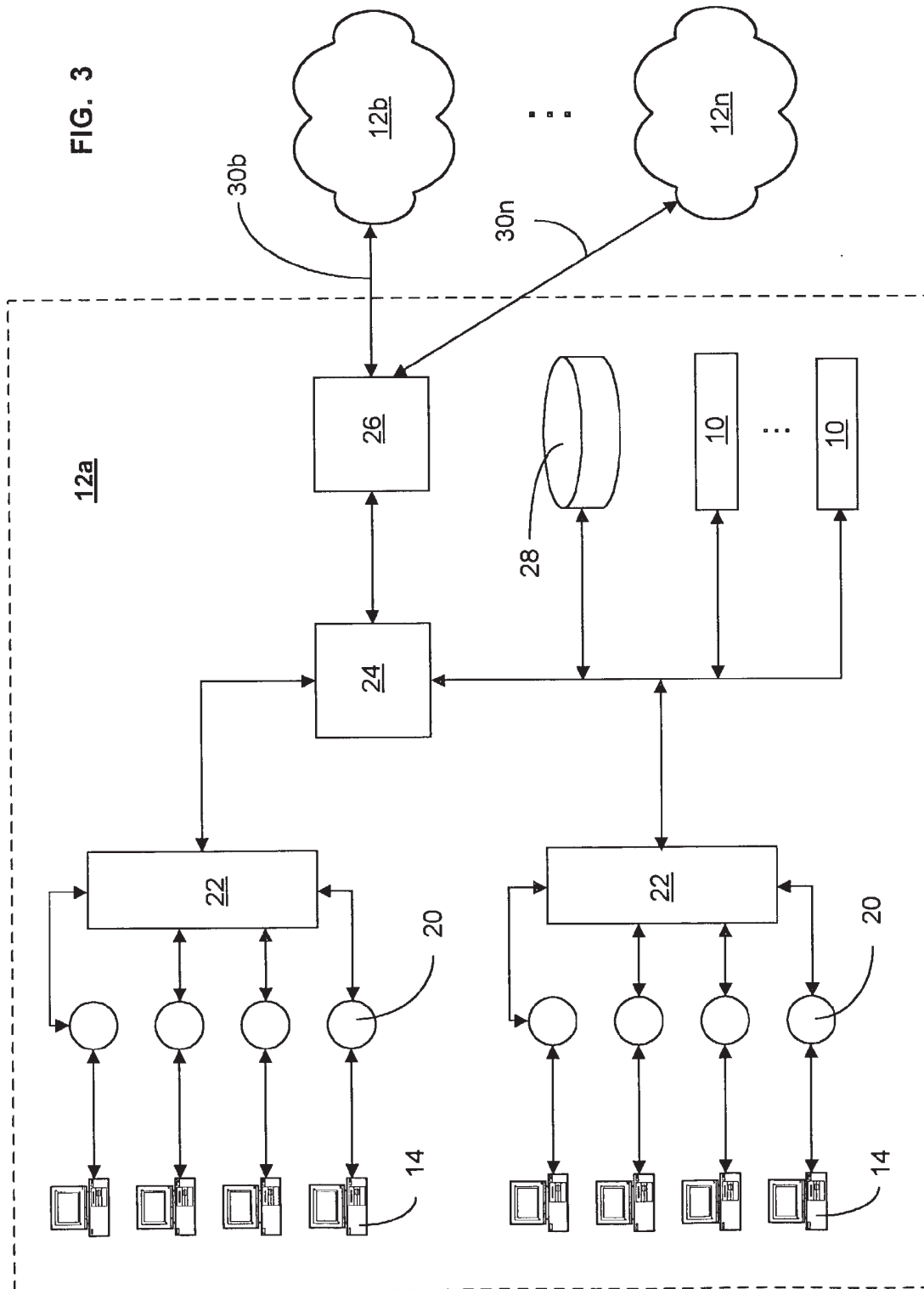


FIG. 3



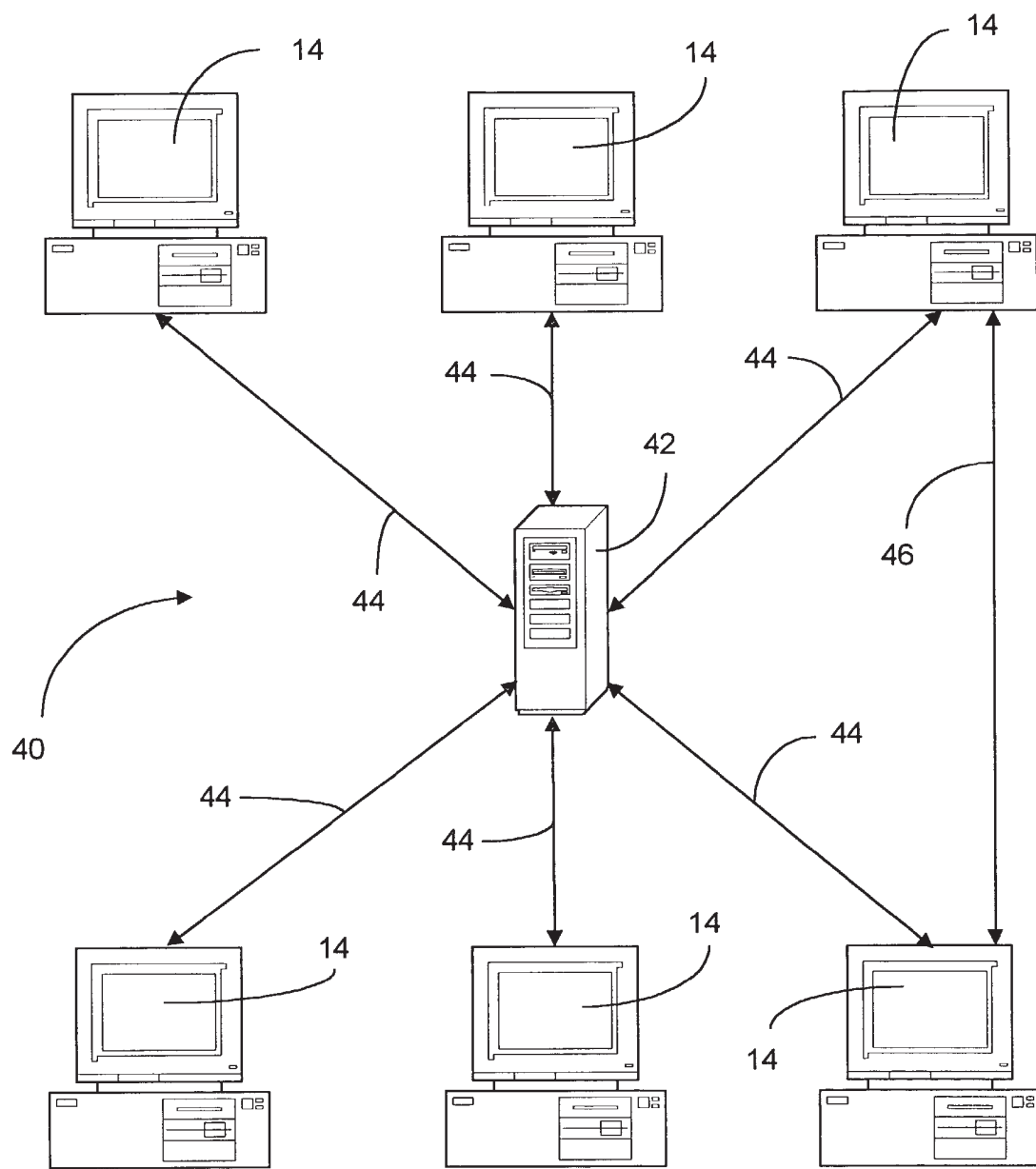


FIG. 4

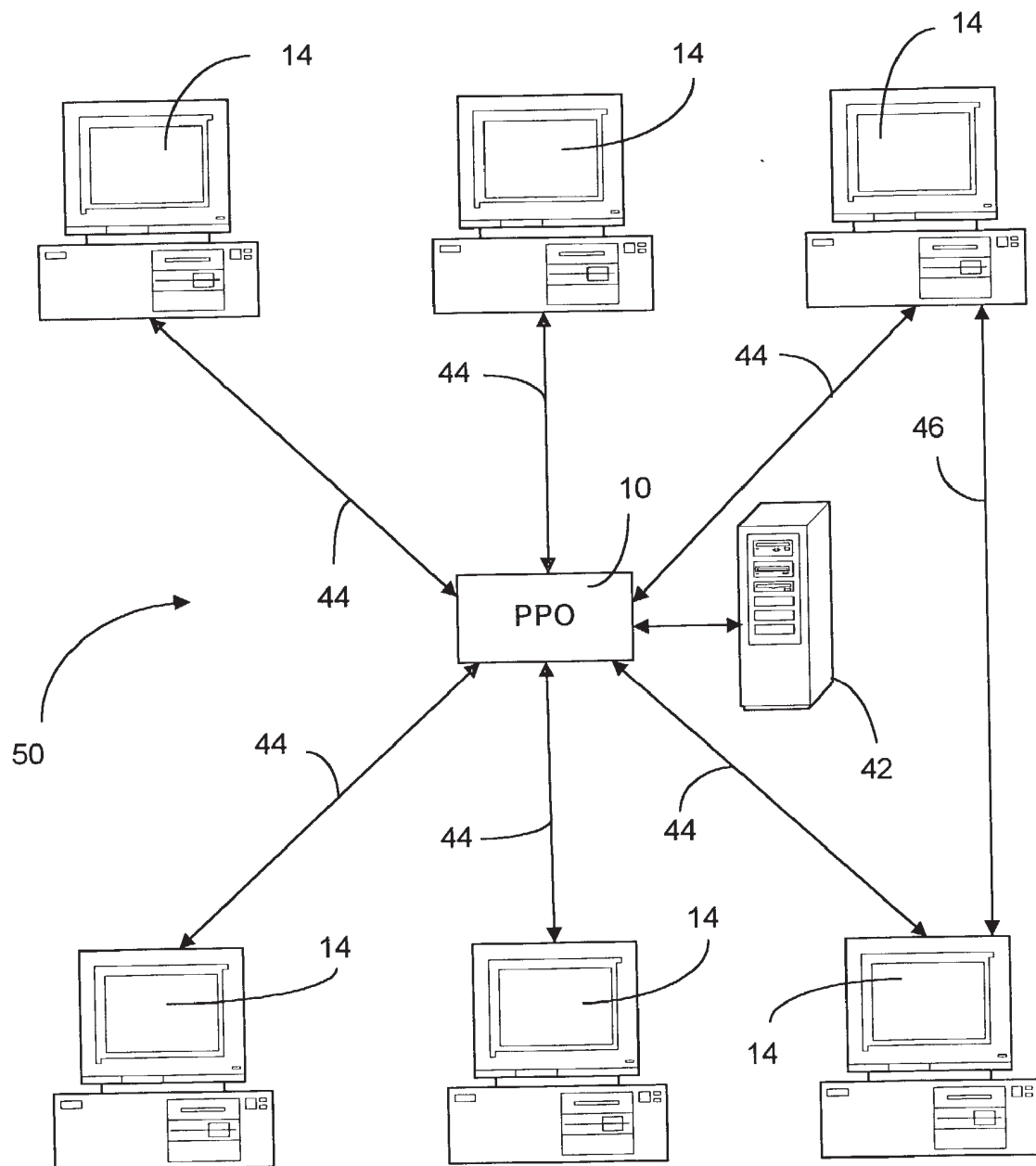


FIG. 5

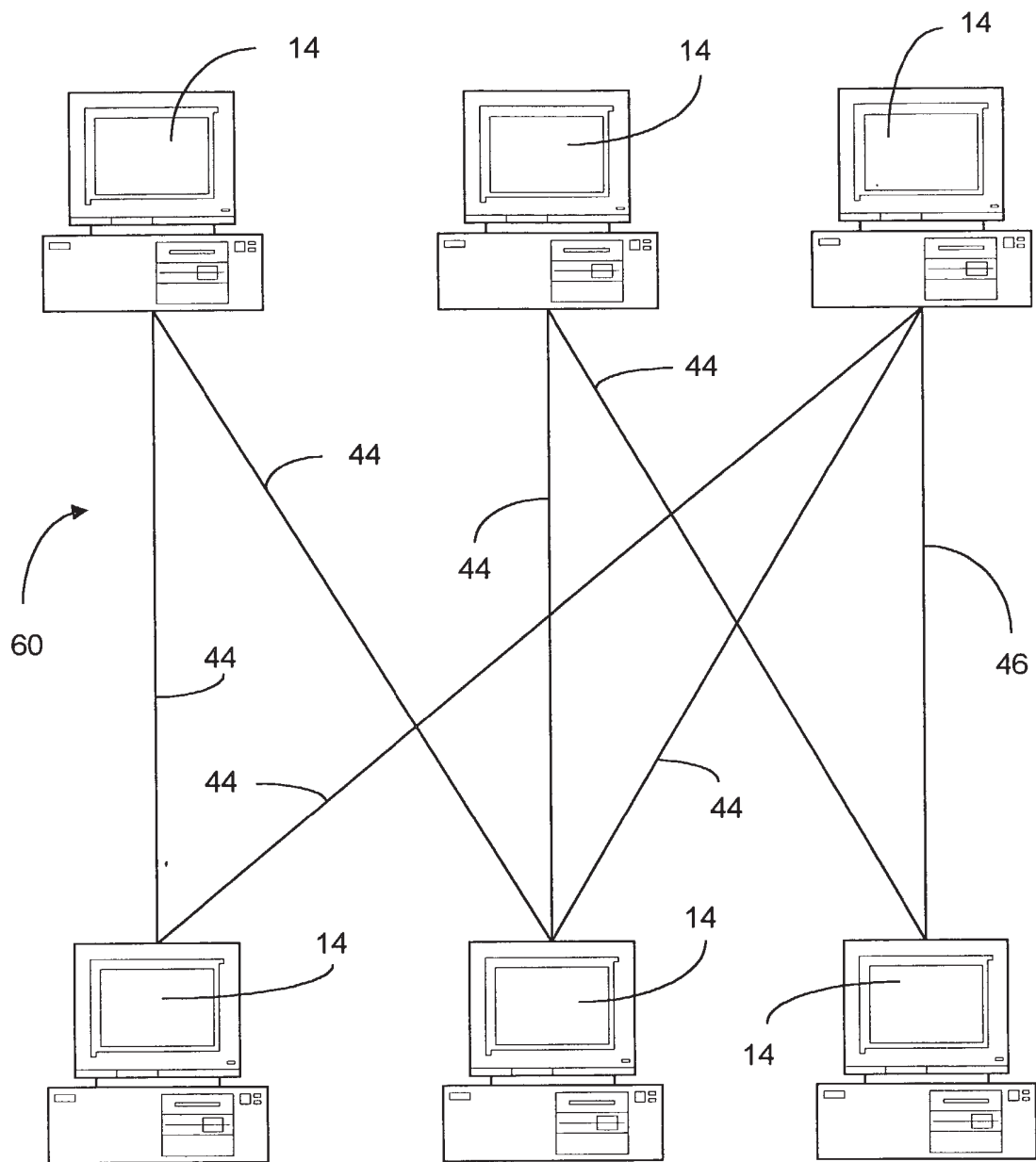


FIG. 6

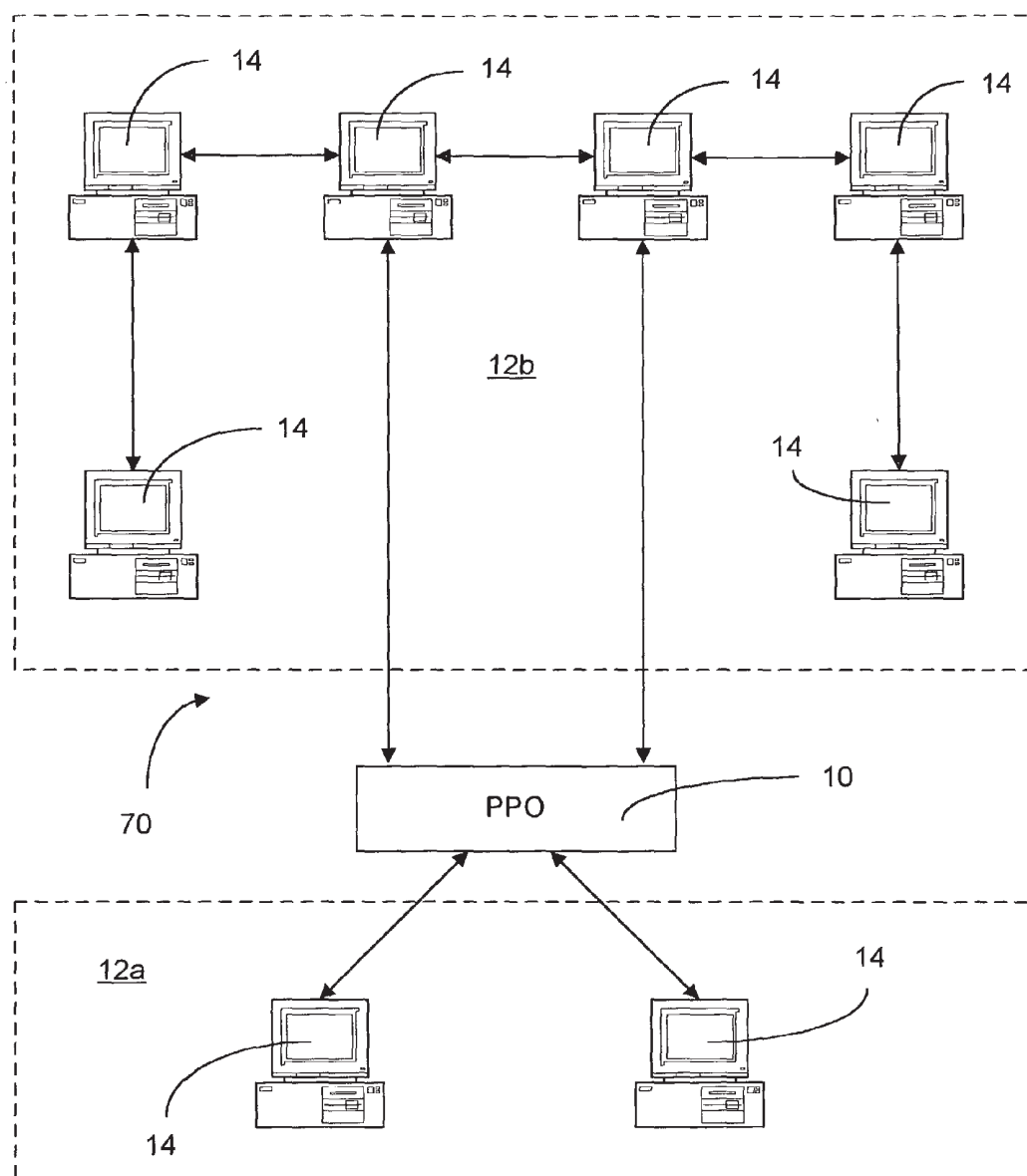


FIG. 7

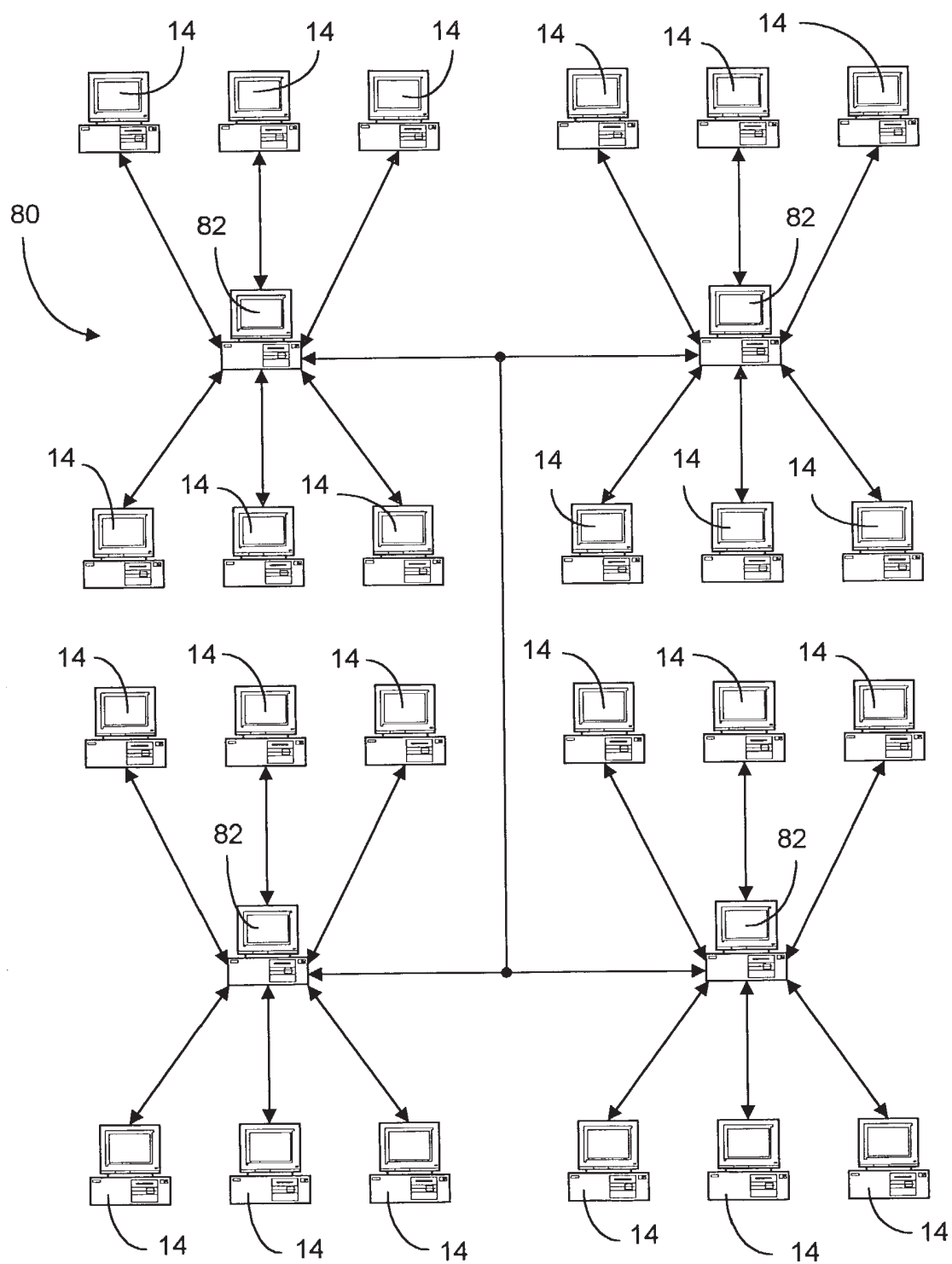


FIG. 8

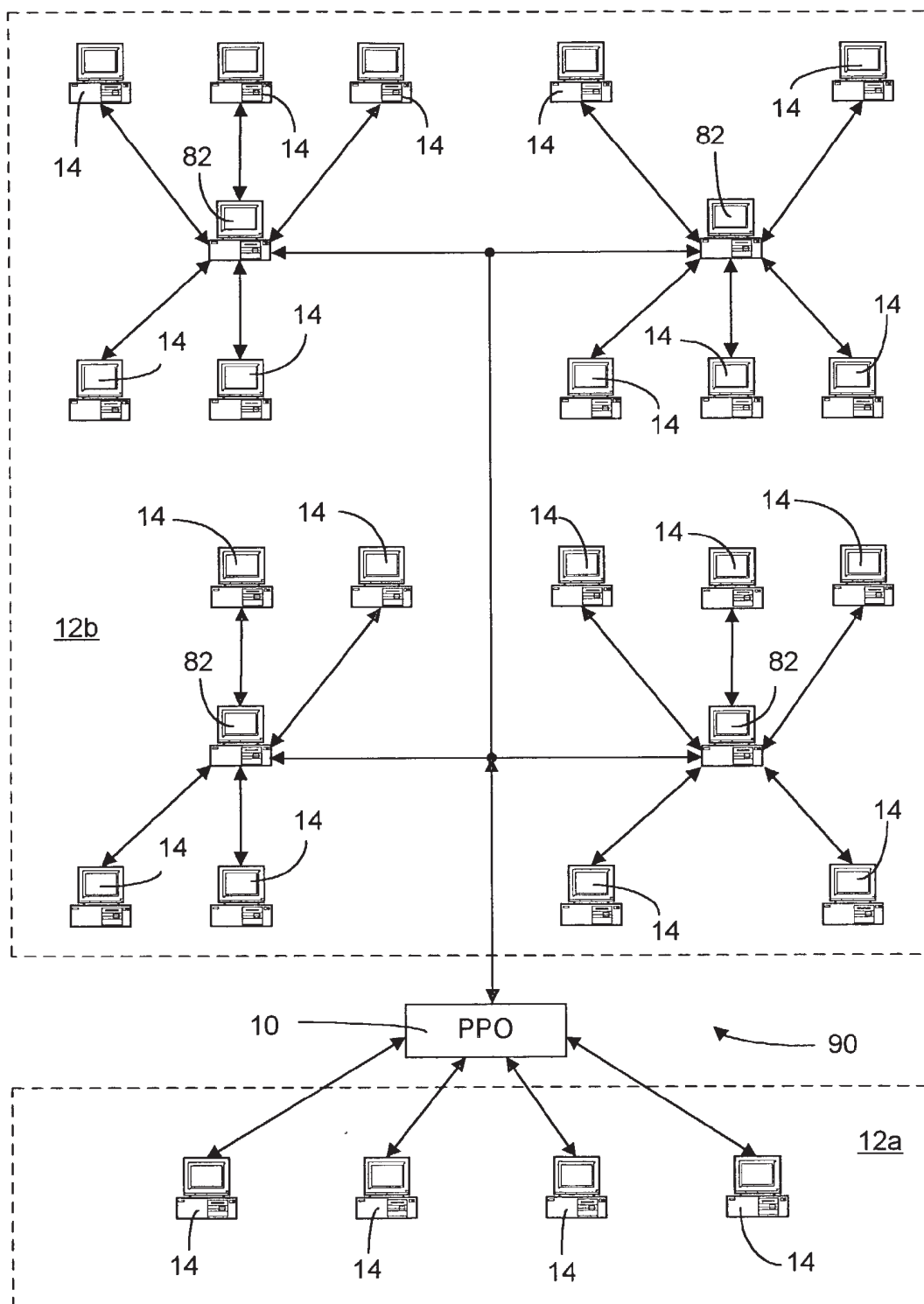


FIG. 9

FIG. 10

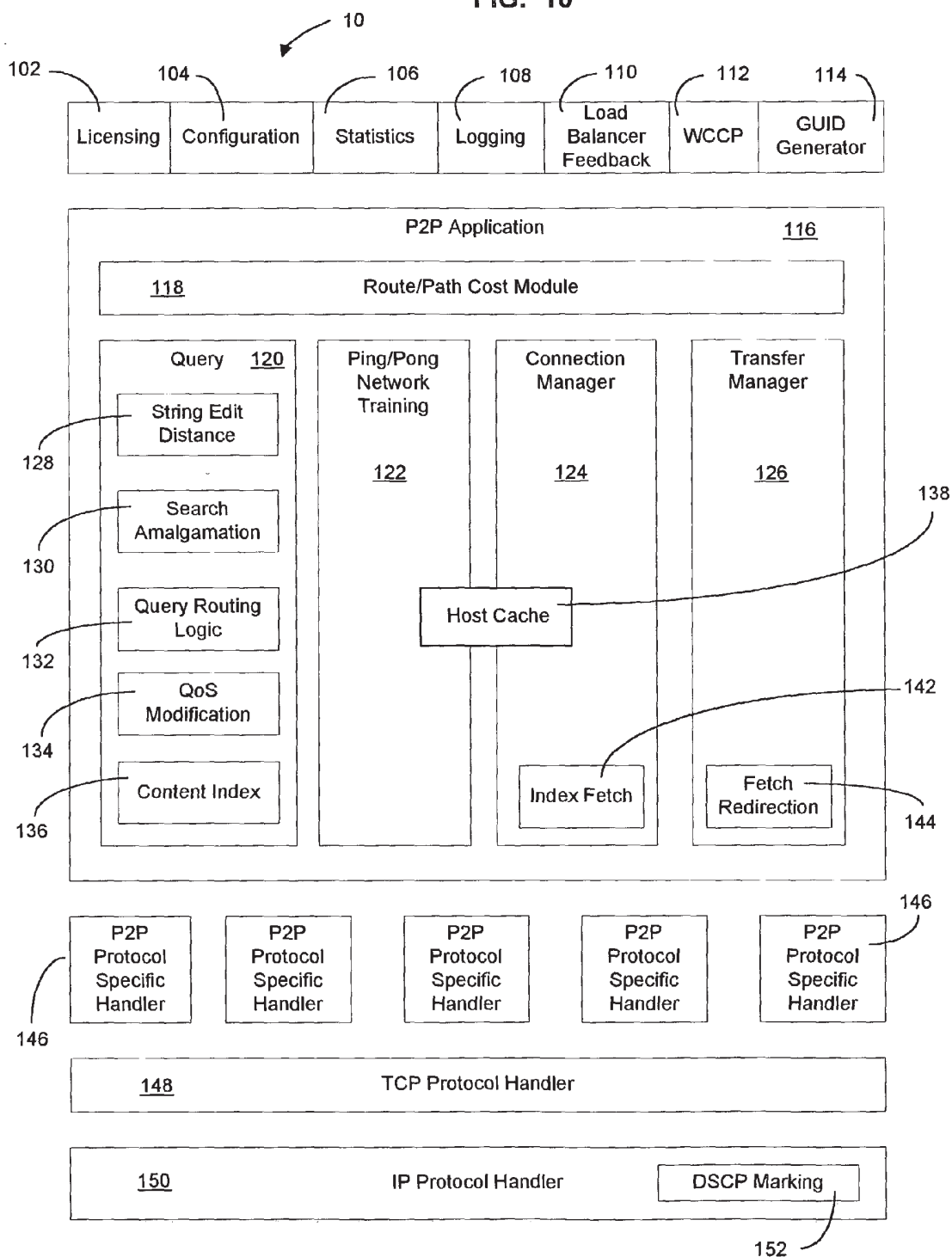


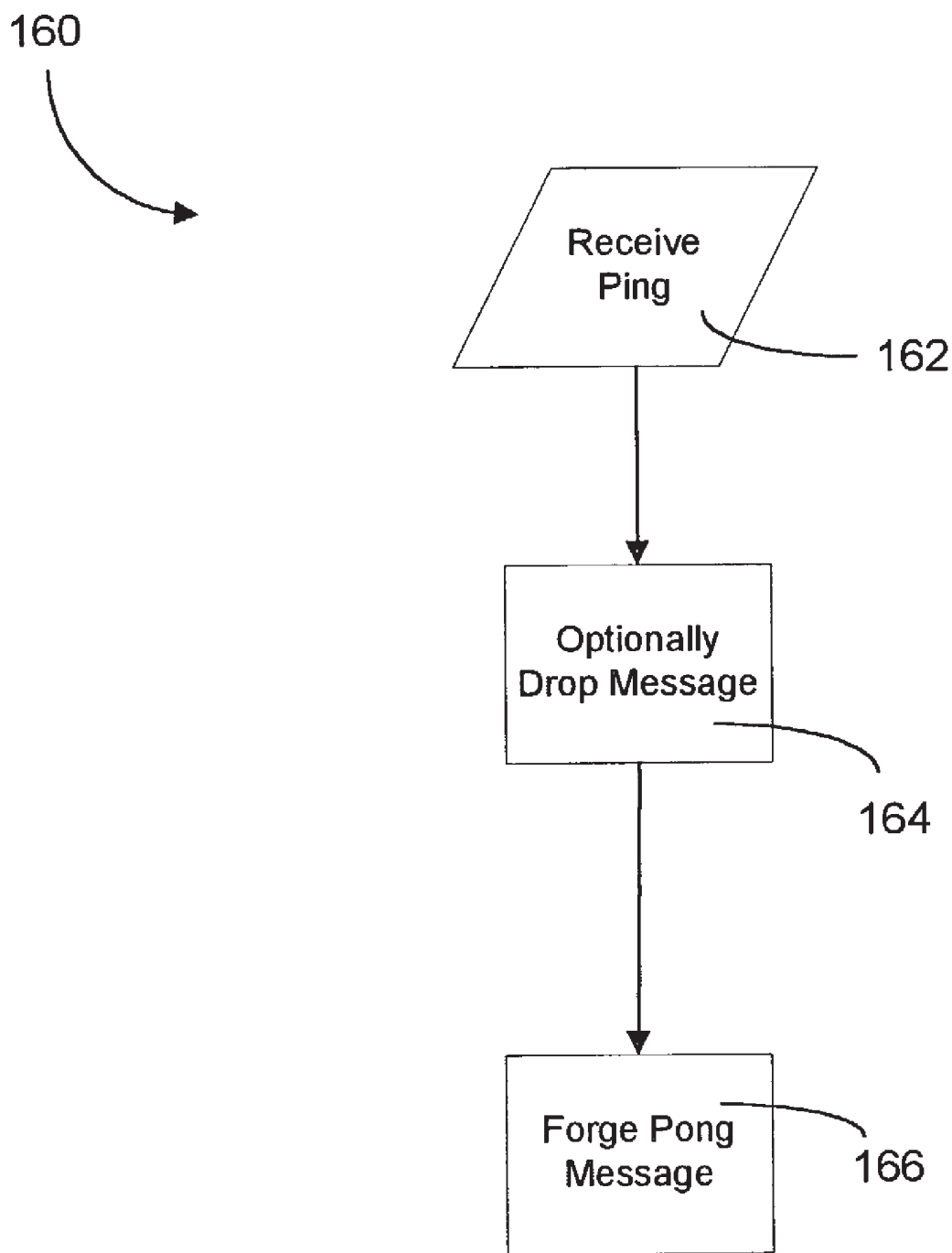
FIG. 11

FIG. 12

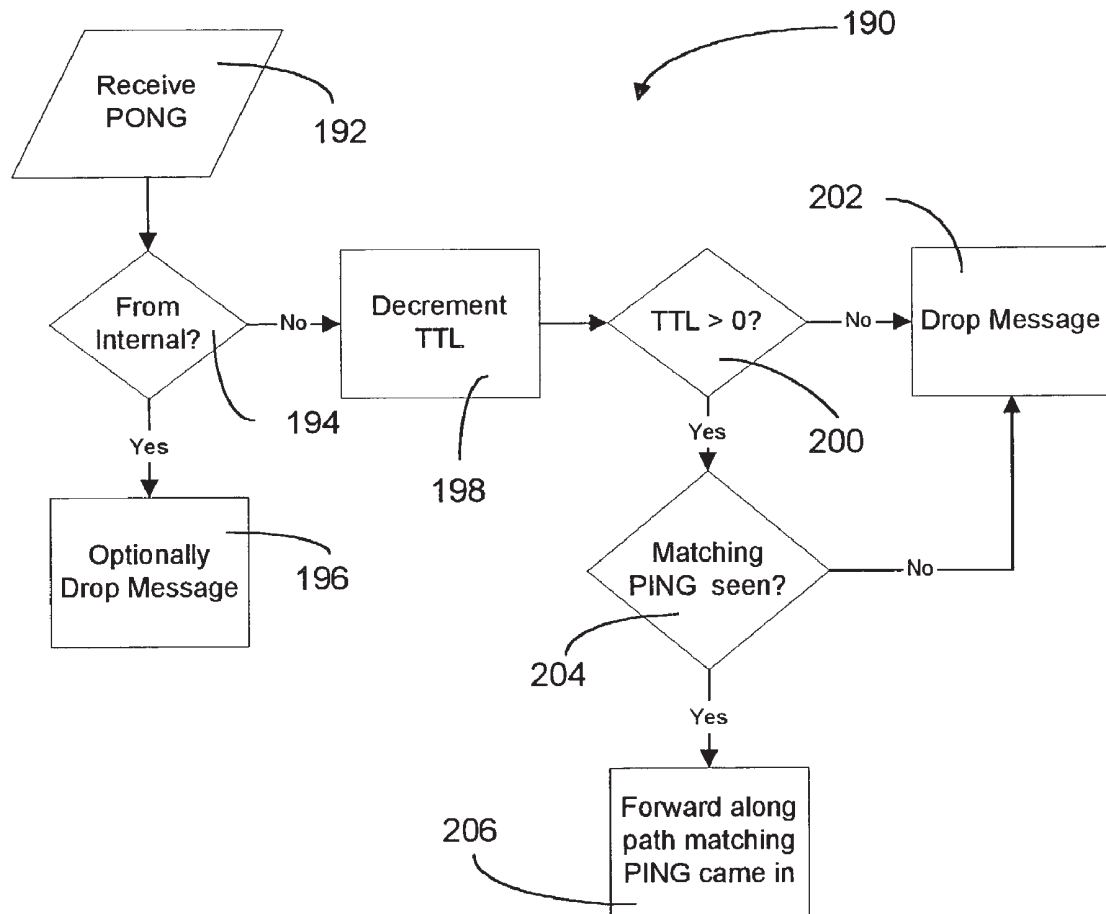


FIG. 13

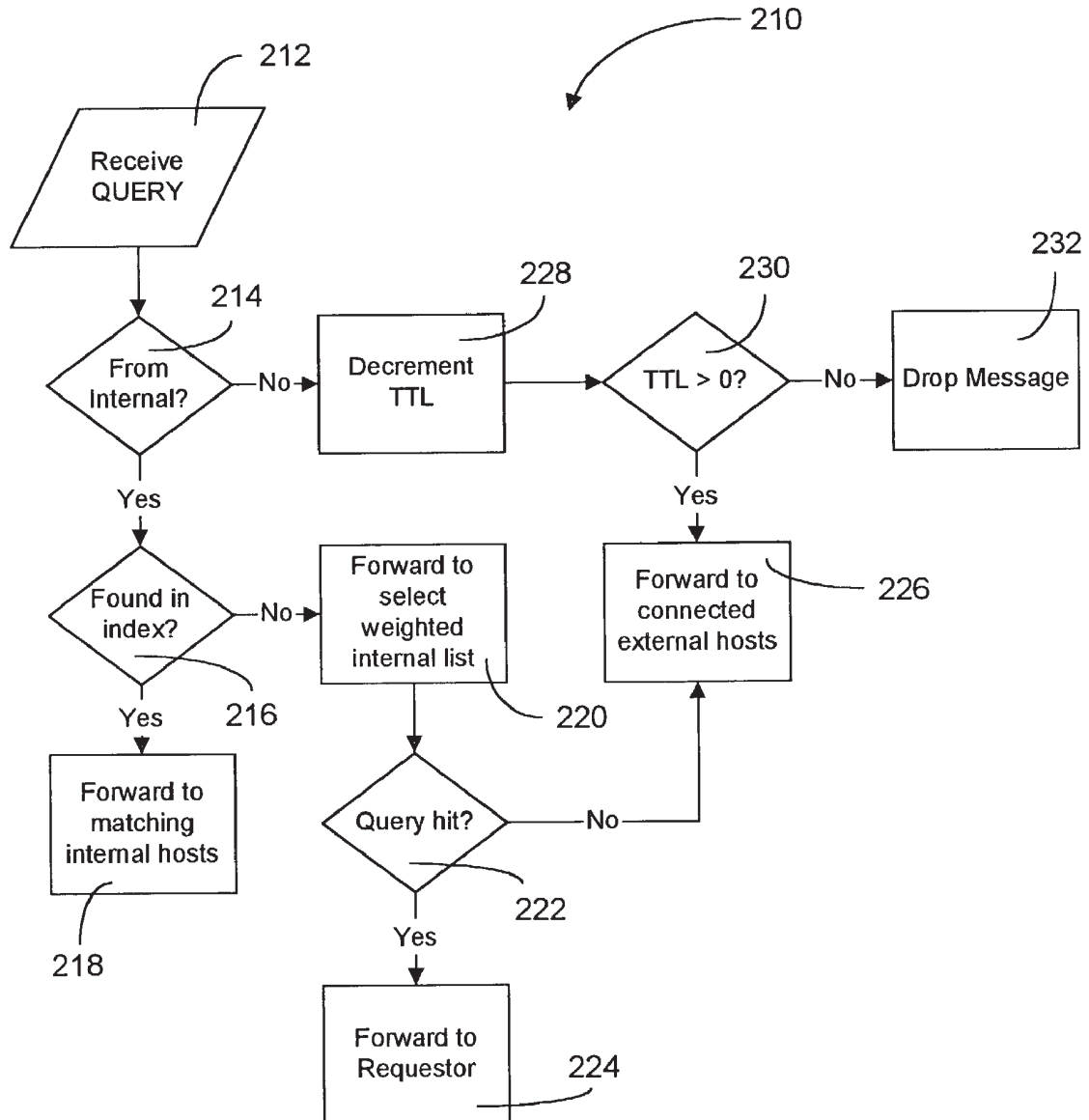


FIG. 14

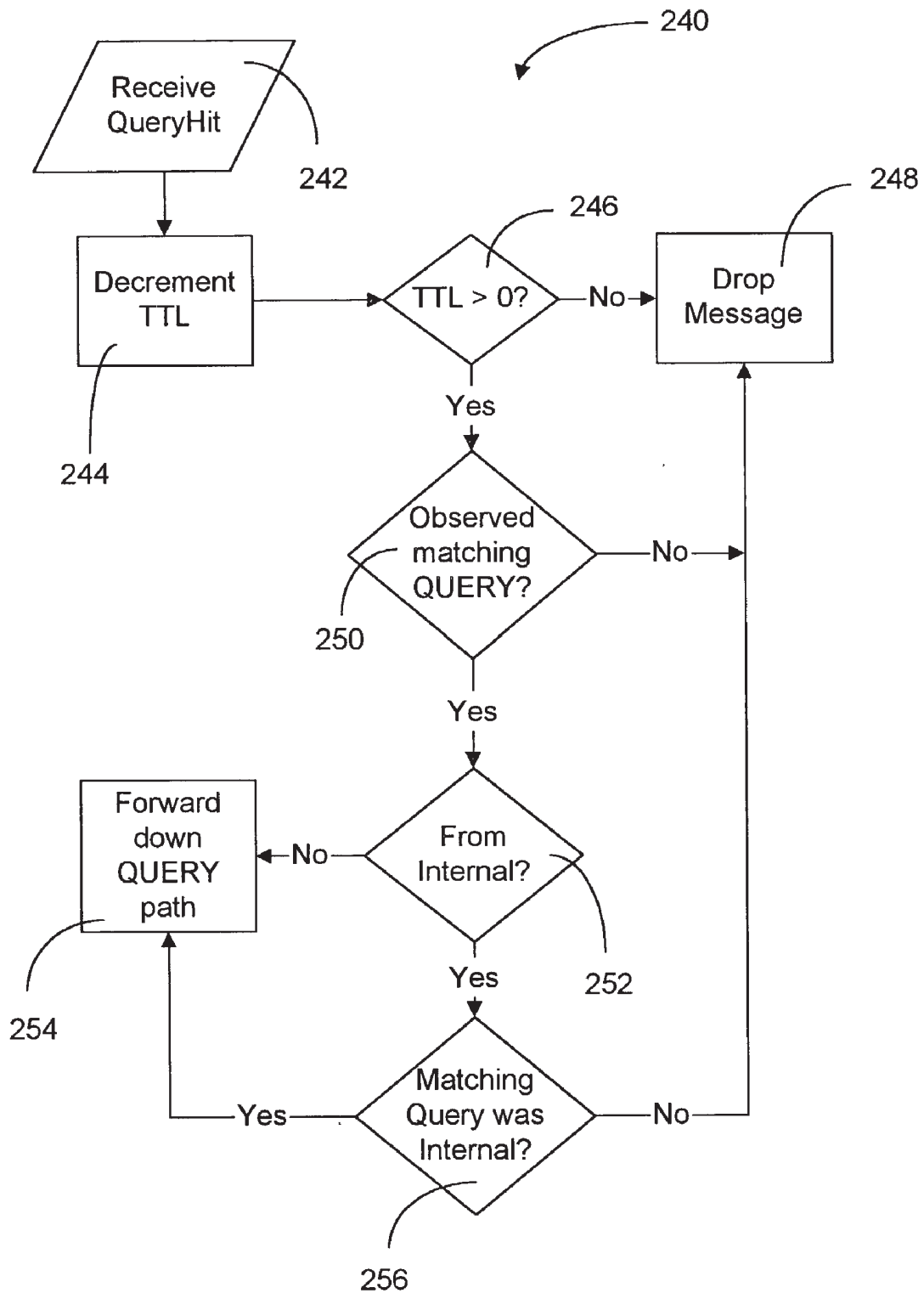
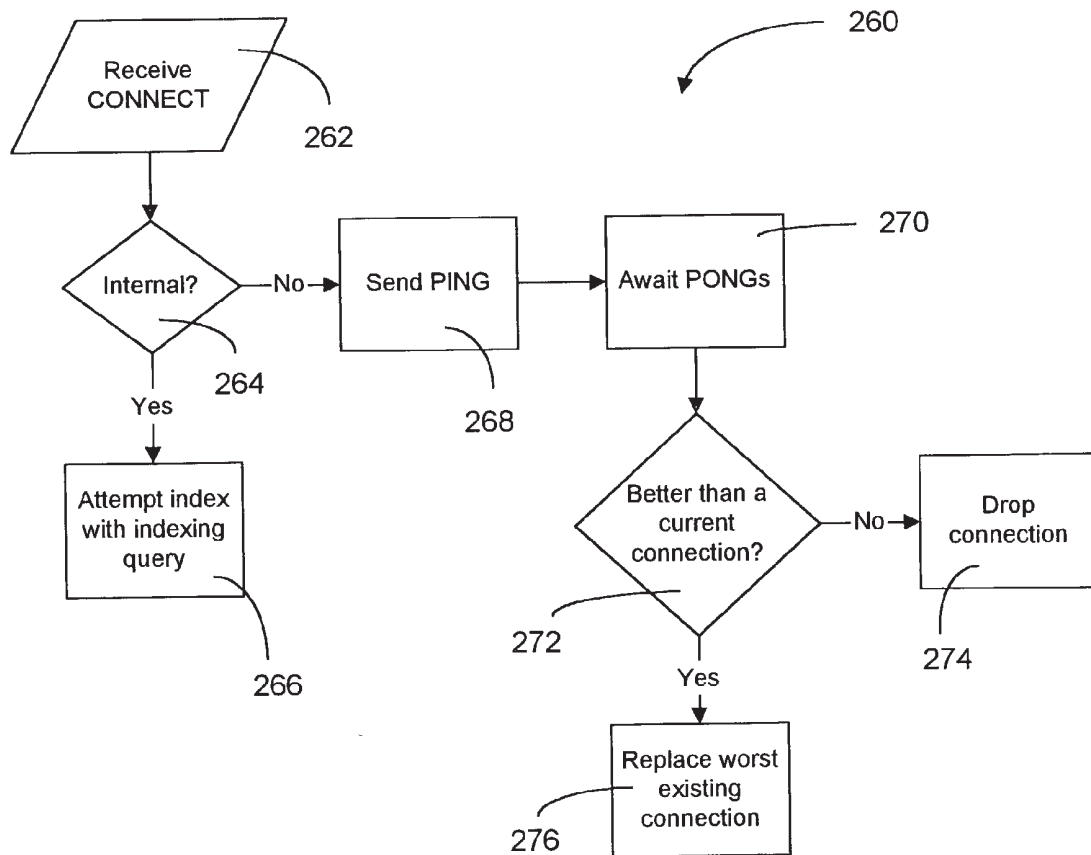


FIG. 15



PATH OPTIMIZER FOR PEER TO PEER NETWORKS

FIELD OF THE INVENTION

[0001] The present invention relates to providing a peer to peer path optimizer (PPO) to examine peer to peer network-ing messages and dynamically and transparently redirect them to a cost efficient path.

BACKGROUND OF THE INVENTION

[0002] Peer to peer (P2P) networking has emerged as a popular form of exchanging data such as movies or music among individuals using the Internet. In a P2P network each computer in the network has the same responsibilities as each of the others, i.e. it is a "peer". Many variations of P2P networks have been created, at the time of writing the most prevalent being: Napster, Kazaa and Gnutella. The use of P2P for transferring large amounts of multimedia data such as movies or music has significantly increased the amount of information transmitted on the Internet.

[0003] P2P has led to increased financial pressure for network service providers. A network service provider is an entity that maintains a group of computers or nodes that form a network. Examples of networks include but are not limited to: a network controlled by an Internet Service Provider (ISP), a corporate network or a university network.

[0004] A network service provider typically must pay a fee for the traffic to and from their network.

[0005] Given the popularity of P2P networking, it is difficult for any network service provider to block P2P traffic. The network service provider is left with few choices, namely:

[0006] a) tiered bandwidth services, and the hope that users will pay for additional bandwidth, or

[0007] b) capping the amount of bandwidth available to P2P applications, which could cause dissatisfaction among the user base.

[0008] Thus, there is a need for an alternative approach, which allows a network service provider to cost effectively constrain P2P traffic through their network, while maintain-ing or improving existing performance to the user. The present invention addresses this need.

SUMMARY OF THE INVENTION

[0009] The present invention is directed to a peer to peer optimizer, the optimizer examining peer to peer messages between nodes within networks connected to the optimizer, for the purpose of optimizing behavior on each of said networks.

[0010] The present invention is also directed to a peer to peer optimizer, the optimizer examining peer to peer mes-sages between nodes within networks connected to the optimizer, for the purpose of determining a cost efficient path for each peer to peer message.

[0011] The present invention is also directed to a process for managing peer to peer messages between and within networks, the process comprising the step of determining a cost efficient path for each of the peer to peer messages.

[0012] The present invention is also directed to a computer readable medium containing instructions for managing peer to peer messages between nodes in networks, the medium comprising instructions for optimizing behavior on the net-works.

[0013] The present invention is further directed to a sys-tem for optimizing peer to peer messages between nodes within networks, the system comprising a peer to peer optimizer, the system utilizing the optimizer to examine messages between the nodes for the purpose of optimizing behavior on each of the networks.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] For a better understanding of the present invention, and to show more clearly how it can be carried into effect, reference will now be made, by way of example only, to the accompanying drawings in which:

[0015] **FIG. 1** is a block diagram of networks connected to the present invention;

[0016] **FIG. 2** is a block diagram of a plurality of nodes connected to a PPO;

[0017] **FIG. 3** is a block diagram of an ISP system;

[0018] **FIG. 4** is a block diagram of a centralized server network;

[0019] **FIG. 5** is a block diagram of a centralized server network utilizing a PPO;

[0020] **FIG. 6** is a block diagram of a decentralized server network;

[0021] **FIG. 7** is a block diagram of a decentralized P2P network utilizing a PPO;

[0022] **FIG. 8** is a block diagram of a hybrid P2P network;

[0023] **FIG. 9** is a block diagram of a hybrid P2P network utilizing a PPO;

[0024] **FIG. 10** is a block diagram of a PPO;

[0025] **FIG. 11** is a logical flow diagram illustrating the processing of a ping message;

[0026] **FIG. 12** is a logical flow diagram illustrating the processing of a pong message;

[0027] **FIG. 13** is a logical flow diagram illustrating the processing of a query message;

[0028] **FIG. 14** is a logical flow diagram illustrating the processing of a queryhit message; and

[0029] **FIG. 15** is a logical flow diagram illustrating the processing of a connect request.

DETAILED DESCRIPTION OF THE INVENTION

[0030] **FIG. 1** is a block diagram of networks connected to the present invention. Peer to Peer optimizer (PPO) **10** monitors all P2P traffic between a plurality of networks **12**. Examples of networks **12** include but are not restricted to; a network controlled by an ISP, a corporate network, or a University network. Networks **12** would typically be con-nected to PPO **10** via the Internet, but that is not a require-

ment of the present invention. Any network **12** that is capable of providing or requesting P2P traffic may make use of PPO **10**.

[0031] To minimize the cost of P2P traffic, network **12** utilizes PPO **10** to determine a cost efficient path for exchanging P2P data between nodes **14**. A node **14** is any computer that is capable of receiving or transmitting P2P data.

[0032] Referring now to FIG. 2, a block diagram of a plurality of nodes connected to a PPO **10** is shown. Each network **12a** and **12b** contains a plurality of nodes **14**. For each node **14** that it is aware of, PPO **10** maintains a cost class. Table 1 illustrates the cost class for each node of FIG. 2. Table 1.

TABLE 1

Node	Cost Class
14a	5
14b	5
14c	10
14d	250
14e	150
14f	50
14g	250

[0033] Assuming that a P2P request can be serviced within a single network such as **12a**, then typically the most cost efficient paths for P2P transfer will be within network **12a**. Examples would be connections to nodes **14a** and **14b**. However, this may not always be the case. For example a request to node **14d** may be very expensive if node **14d** which contains the data, resides halfway around the world within a corporate intranet. In such a scenario, node **14f**, within network **12b**, which contains the required data, would be a more cost efficient choice.

[0034] In determining a cost efficient path for the delivery or reception of P2P data, PPO **10** combines the cost class of each node on the end of a potential exchange of data. This combination results in a path cost value. For example, a request from node **14e** for a file on node **14a** may result in a path cost of **155**. This example is one of simple addition to the cost class of two nodes to determine a path cost. The inventor does not intend to restrict the present invention to any specific algorithm to obtain a path cost. For example, a weighting factor may be applied to nodes with a high cost class to exclude them from consideration in calculating a path cost.

[0035] In FIG. 3, a block diagram of an ISP system is shown. FIG. 3 illustrates how an Internet Service Provider (ISP) may make use of the present invention. Network **12a** is the network maintained by an ISP. Network **12a** is connected to a plurality of networks **12b** to **12n** via links **30b** to **30n**. Typically networks **12b** to **12n** would be accessible via the Internet, but they may be any form of network which contains files for P2P exchange. Network **12a** comprises a plurality of nodes **14**. P2P data is exchanged between nodes **14** within network **12a** and nodes within networks **12b** to **12n**. A node **14** may be the computer of a home user, a business computer or a corporate server connected to any of networks **12a** to **12n**. Returning to the present ISP example of FIG. 3, each node **14** within network **12a** is connected to

a communications module **20**, which allows node **14** to communicate with network **12a**. Communications module **20** may be Digital Subscriber Line Access Multiplexer (DSLAM) which is used for phone line connections. Communications module **20** may also be a Cable Modem Termination System (CMTS) which is used for cable connections. Communications module **20** may also be a module that accepts dialup connections, wireless connections or fibre optic connections. The point here being that communications module **20** connects a node **14** to network **12a**. An aggregator **22** collects the data to and from communications modules **20** and is connected to distribution router **24** for this purpose. Distribution router **24** determines where a request for information should be routed within network **12a**. Distribution router **24** is connected to core router **26**, one or more cache servers **28** and one or more P2P Path Optimizers (PPO) **10**. Core router **26** is connected to networks **12b** to **12n** most typically by InterXchange Carrier (IXC) links **30b** to **30n**. An IXC is a telecommunications company such as AT&T. A cache server **28** is a repository of information obtained from networks **12b** to **12n** that may be frequently accessed by nodes **14**. To avoid the expense of continually requesting data from networks **12b** to **12n** network **12a** may store frequently accessed information in one or more caches **28**. Most commonly this would be current versions of popular websites, but may include all forms of data. Cache **28** may also commonly be referred to as a "cache cluster" or a "cache server".

[0036] PPO **10** is where the present invention resides. PPO **10** serves to provide three main functions:

[0037] 1) Reorganize networks connected to PPO **10**. This is achieved by intercepting all P2P messages and attempting to have nodes connect to other nodes in the same cost class. This allows the networks to reorganize in two ways:

[0038] a) they become flatter as nodes connect to nodes under the control of a PPO **10** thus a tree of connections between nodes would have at most a depth of one; and

[0039] b) PPO **10** attempts to connect nodes to other nodes within a network, where without the use of a PPO **10**, connections would be random and a tree of connections between nodes may have an unlimited depth.

[0040] 2) Reduce network traffic. This is done by not broadcasting messages but instead sending them where they need to go, or dropping them if there is no need to send them on.

[0041] 3) Redirecting traffic to a cost efficient path.

[0042] Each of these functions is discussed in more detail below.

[0043] Although the example of FIG. 3 applies to the network of an ISP, it is not the intent for the inventors to restrict the use of the present invention to an ISP network. Any network for which an entity wishes to control P2P traffic in a cost efficient manner may make use of the present invention. As discussed above examples would be a corporate network or a University network or any commercial use of a large network, such as the hotel industry.

[0044] Before describing in detail the structure of PPO 10, we will refer first to how it may be utilized in a variety of P2P models.

[0045] Referring now to FIG. 4, a block diagram of a centralized server network is shown generally as 40. Network 40 is an example of a first generation P2P network, such as Napster. Napster is an Internet service that was originally designed to permit users to exchange MP3 music files. In network 40, a central server 42 connects a plurality of nodes 14 via connections 44. Connections 44 would typically be connections via the Internet. A node 14 sends a request for a file to central server 42 via a connection 44. Central server 42 provides a reply via connection 44 indicating on which node 14 the requested file resides. In essence central server 42 contains a directory of all files available for access on all nodes 14. In acting on the reply, the requesting node 14 establishes a connection with its peer node 14 that contains the file and requests a copy of the file, as is shown in transfer link 46.

[0046] To explain how FIG. 4 may make use of the present invention we refer now to FIG. 5 where a block diagram of a centralized server network utilizing a PPO is shown generally as 50. In network 50, PPO 10 examines search requests sent to central server 42. If PPO 10 is aware of the requested file, it will provide the requester with a cost efficient path to the file. If PPO 10 is not aware of the file it will utilize alternatives to direct the requestor to the file. These alternatives are discussed in detail later. In determining which node 14 to direct the request to, PPO 10 makes use of cost class information. Cost class information for a node would typically be determined by metrics such as the speed of the connection to the node (e.g. bandwidth and distance) and the monetary cost of using a connection to the node. Cost class information would typically reflect a monetary cost to obtain a file from a specific node 14. An administrator of the present invention may set their own cost class to a node 14 or PPO 10 may set them by default or determine them dynamically. Whatever the method of establishing the cost class for a node, the point is that each node has a cost associated with it and PPO 10 utilizes this information to provide a cost efficient path for exchanging P2P data.

[0047] Referring to FIG. 6, a block diagram of a decentralized server network is shown generally as 60. Network 60 utilizes a distributed model where each node 14 is equal and there is no central server 42 as with network 40 (FIG. 4). Network 60 may be considered to be a second generation P2P network, an example of which would be Gnutella. Gnutella provides a file sharing service for many types of information and is not directed solely to the exchange of multimedia files. Each node 14 tries to maintain some number of connections 44 to other nodes 14 at all times. Requests for information are sent with a Time to Live (TTL) field that is decremented and then forwarded by each node 14 to all other nodes 14 to which it is connected. When the TTL value reaches zero, the request is dropped. This type of network has been shown to have significant scaling issues, as requests for information will degrade network performance. As with network 40 when a requested file is located a direct connection is made between two nodes to transfer the requested data as shown by transfer link 46.

[0048] In the present invention the topology of network 60 is reconfigured as shown in FIG. 7. FIG. 7 is a block diagram of a decentralized P2P network utilizing PPO 10 and is shown generally as 70. In network 70 when P2P communication is sent between networks 12a and 12b PPO 10 examines it. PPO 10 then determines a cost efficient manner to deal with the communication. It is not the intent of the inventor to restrict the use of the present invention to only two networks 12a and 12b as shown in FIG. 7.

[0049] Referring now to FIG. 8 a block diagram of a hybrid P2P network is shown generally as 80. This network topology may be referred to as third generation P2P where some nodes are elected as "supernodes" or "ultra peers" and serve as the traffic coordinator for the other nodes. In FIG. 8, supernodes are designated with the feature number 82 and are connected to each other. This model is utilized by P2P services such as Fasttrack, Kazaa, Morpheus and Grokster. The supernodes 82 change dynamically as bandwidth and network topology change. Any node 14 may be a supernode 82.

[0050] Referring now to FIG. 9, a block diagram of a hybrid P2P network utilizing PPO 10 is shown generally as 90. In network 90, PPO 10 acts as a supernode between networks 12a and 12b. All nodes within network 12a will see PPO 10 as their supernode and thus as their path to network 12b. Nodes 14 within network 12a may also be supernodes within network 12a (not shown).

[0051] With regard to the topologies of the networks shown in FIGS. 4 to 9, it is the intent of the inventor to simply illustrate how the present invention may be utilized in existing P2P networks. It is not the intent of the inventor to restrict the present invention to the networks shown, but rather to provide examples of the diversity of the present invention.

[0052] Referring now to FIG. 10 a block diagram of a PPO is shown generally as 10. As one skilled in the art can appreciate, PPO 10 may be implemented in many different ways. The structure of PPO 10 as shown in FIG. 10 serves only as an example of one implementation that may be used to examine and manage P2P communications. We will now describe the components of PPO 10 as illustrated in FIG. 10 in more detail.

[0053] Licensing module 102 is responsible for enforcing the maximum number of concurrent users of PPO 10 for which the customer (i.e. the owner of a PPO 10) has paid a license fee. Configuration module 104 maintains the configuration of PPO 10, such as the sub-networks and IP addresses of the nodes that reside within a network 12. Statistics module 106 maintains the statistics for PPO 10, such as the number of files redirected and the number of concurrent users. Logging module 108 is responsible for logging functions, such as when PPO 10 was started up or shut down and when the number of licenses was exceeded. Load balancer feedback module 110 provides a negative feedback loop to an external load balancer so that multiple PPO's under the control of a customer will receive equal traffic. WCCP module 112 operates with the Cisco Web Cache Communication Protocol (WCCP) to ensure that a router, such as distribution router 24 of FIG. 3 sends only P2P communications to one or more PPO's 10. As one

skilled in the art can appreciate, a number of methods may be used to direct P2P traffic to a PPO 10, such as recognizing specific port addresses or context sensitive scanning of packets. WCCP serves only as one example. GUID generator 114 generates a globally unique identifier for each sender of a P2P packet to avoid the possibility of looping back to the original sender of the packet and to also uniquely identify messages that have been received.

[0054] P2P application 116 acts as the control program for PPO 10. Application 116 comprises: route/path cost module 118, query module 120, ping/pong network training module 122, connection manager module 124 and transfer manager module 126. Route/path cost module 118 assigns a path cost to each proposed connection based upon the cost class of each node in the connection.

[0055] Query module 120 comprises: string edit distance module 128, search amalgamation module 130, query routing logic module 132, QoS modification module 134 and content index module 136. String edit distance module 128 determines the similarity between the name of a requested file and the filenames known to PPO 10. Search amalgamation module 130 utilizes string edit distance module 128 to map the name of a requested file to the known files available, regardless of cost class. Query routing logic module 132 routes queries for a file to the nodes that are likely to contain the requested file. Module 132 maintains a list of all messages to and from a network 12. By maintaining such a list, module 132 may quickly drop spurious messages, such as requests for data that have not been acknowledged. QoS modification module 134 rewrites the routing information of module 132 to select a cost efficient path determined by route/path cost module 118. Routing information includes QoS parameters such as stated bandwidth and uptime. The purpose of rewriting routing information is to provide the requestor with a path to a file or files that make the most efficient use of network resources. By doing so a message may be redirected. Content index 136 maintains an index of content available for access in nodes 14 within networks 12. Content index 136 also contains the cost class for each node in which the content resides. Typically such content will be a file but may also include forms of data such as streaming media. It is not the intent of the inventor to restrict the use of the term "file" to any form of P2P data that may be examined by or transmitted through PPO 10.

[0056] Ping/Pong network training module 122 serves to fill host cache 138 with IP addresses of nodes 14 based upon the Ping messages received by PPO 10 from nodes 14. Ping/Pong network training module 122 sends a plurality of Pong messages in response to a Ping message in an attempt to train a network sending a Ping message. Pong messages are sent by PPO 10 for each node 14 that is in the same cost class as the sender of the Ping that PPO 10 is aware of. This use of multiple Pong messages serves to train the network that sent the Ping. This training provides the sending network with nodes other than those for which PPO 10 wishes to restrict traffic.

[0057] When a connection is established between a node 14 and PPO 10, connection manager 124 maintains the connection until the node 14 drops the connection. Index fetch module 142 is responsible for obtaining content names and adding them to content index 136.

[0058] Transfer manager 126 is in essence a proxy that handles the exchange of P2P data. Manager 126 utilizes fetch redirection module 144 to redirect a request for content to a node with a lower path cost. A node 14 may make a request for a specific file on another node 14. If that file is available via a more cost efficient path, fetch redirection module 144 will silently direct the request to another node having a more cost efficient path.

[0059] A plurality of P2P protocol specific handlers 146 are responsible for maintaining a specific P2P protocol, for example Gnutella or Fasttrack. Transmission Control Protocol (TCP) handler 148 ensures the maintenance of correct TCP behaviour. Similarly, Internet Protocol (IP) handler 150 serves the same purpose for IP. It is not the intent of the inventor to restrict the present invention to the use of TCP and IP. These serve only as an example. As one skilled in the art can appreciate any number of communication protocols may be used, including, but not restricted to: ATM, UDP, and wireless.

[0060] Differentiated Services Code Point (DSCP) marking module 152, utilizes Differentiated Services (DiffServ or DS) to specify IP packets by class so that certain types of packets get precedence over others. For example a limit may be imposed on the number of P2P packets allowed to enter or leave a network 12. Such a feature is optional but may be used by networks that find P2P data is consuming too much of their bandwidth. As one skilled in the art can appreciate any number of schemes such as packet snooping or recognizing specific port addresses may be utilized to identify P2P traffic. It is not the intent of the inventor to restrict the ability to limit P2P traffic to the DSCP solution.

[0061] PPO 10 optimizes behavior between and within the networks 12 to which it is connected. Behavior is the ability to create, destroy, modify or ignore messages. Behavior optimizes future behavior of each network 12, not just the current message. An example of creating a message is a false pong. An example of destroying a message is deleting a message that has already been answered or in the case of Gnutella, a message whose TTL has expired. Modification is not limited to QoS modification module 134. For example, search amalgamation module 130 may modify messages to reflect the closest filename as determined by string edit distance module 128. In the case of a specific protocol, for example Gnutella, modification may include overwriting the TTL portion of the message when forwarding the message. Similarly the GUID for a message may be changed if needed. In essence, depending upon the protocol, PPO 10 may modify messages as required to optimize network behavior. An example of ignoring a message is to ignore a query request to a node in a network, as traffic from that network has been restricted.

[0062] In order for PPO 10 to examine and act upon P2P requests, it must be aware of a variety of P2P protocols. This functionality is handled by P2P protocol specific handlers 146.

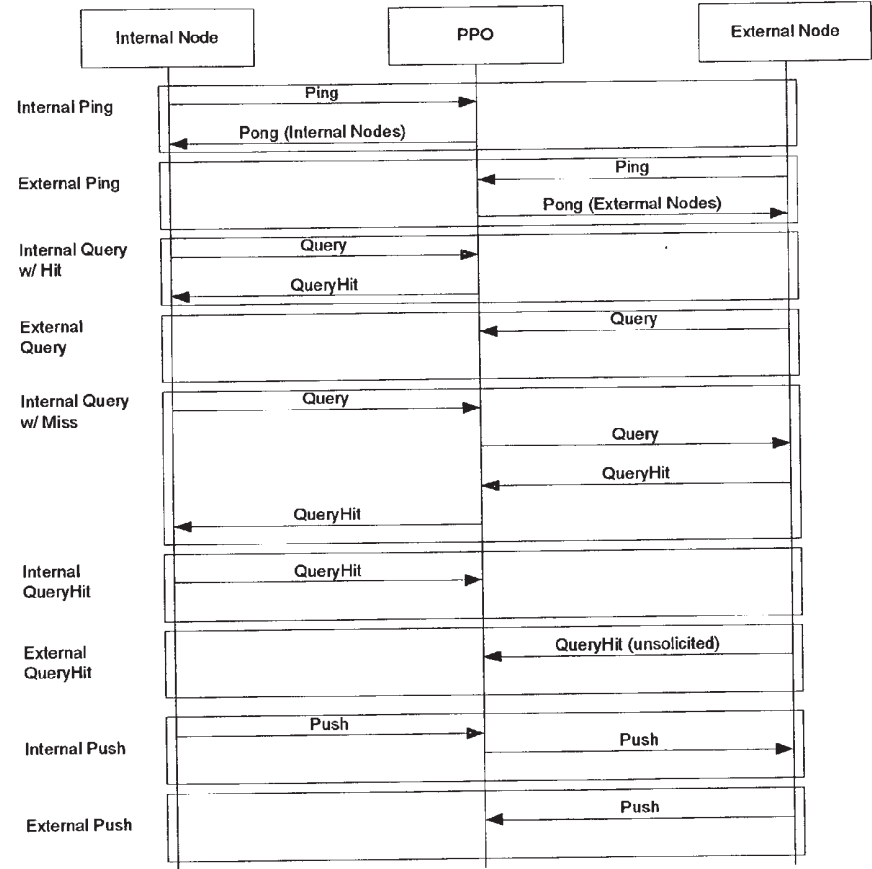
[0063] By way of example we refer next to how a P2P protocol specific handler 146 may interface with the Gnutella protocol. It is not the intent of the inventor to restrict the present invention to work simply with the Gnutella protocol, but rather to provide a practical example of how the present invention may deal with P2P requests.

[0064] The Gnutella protocol has five message types, namely: ping, pong, query, queryhit and push. How a handler 146 handles each of these messages is shown in the following Chart 1. In Chart 1, the term “internal node” refers to a node 14 within network 12a of the ISP example of FIG. 3. The term “external node” refers to a node 14 within a network 12b to 12n of FIG. 3. By the use of the terms “internal node” and “external node” the inventor means to show how PPO 10 may be used to examine and redirect P2P

traffic between nodes in an “internal” network such as an ISP and nodes in an “external” network such as a plurality of sites on the Internet.

[0065] Chart 1 illustrates how PPO 10 may be used to examine and redirect P2P traffic between network. In the following description of Chart 1, we advise the reader refer to FIGS. 3 and 10 as well as Chart 1.

Chart 1



[0066] A ping message is used to determine if a node 14 is active, and helps to establish a database of active nodes in host cache 138 of FIG. 10. PPO 10 responds to a ping message with a pong message. To avoid identifying a node 14 within network 12a, PPO 10 would typically provide a forged pong message. The forged pong message would indicate the number of files shared and the amount of data shared within network 12a containing the pinged node 14, as well as the IP address and port of the pinged node. Similarly in this example, PPO 10 does not forward pong messages, however it does receive them and adds them to the list of nodes in host cache 138 from which it may obtain data.

[0067] A query message is a search message containing a fragment of a filename, in other words, a request for data. In the present example, incoming query messages from an external node are dropped, thus appearing to be a query miss and thereby avoiding servicing a P2P request from a network 12b to 12n. It is not the intent of the inventor to require that query messages be dropped, it is simply one method that may be used to restrict unwanted P2P traffic into network 12a. Implementations utilizing PPO 10 may choose to allow free flow of all messages or to provide a limited amount of traffic. Query messages from a node 14 within network 12a are forwarded first to the nodes 14 containing the requested file that have a cost efficient path. Typically these would be nodes 14 within network 12a, but that may not always be the case. The nodes 14 having the requested data will then respond with queryhit messages. If there are no matches for the request for data, or if no queryhit message is returned, then the query message is sent to a random set of nodes 14 within network 12a. One method of determining the random set of nodes 14 to receive the query message would be to use a weighted probabilistic function such as a round robin method based upon the number of files available from each node 14. In this way, the query does not always go to the node 14 having the largest number of files. If there is still no match, the query is forwarded to nodes 14 having the lowest path cost in networks 12b to 12n.

[0068] A queryhit message is a response to a query message. Incoming queryhit messages from nodes in networks 12b to 12n are forwarded to the appropriate node 14 within network 12a. Incoming queryhit messages from nodes 14 within network 12a are forwarded back to the requesting node within network 12a and not sent out to networks 12b to 12n.

[0069] A push message is used when the transmitting node has a firewall and the receiving node does not. The receiving node sends a push message, which causes the transmitting node to open a connection directly to the receiving node. Incoming push requests may be optionally dropped by PPO 10 and are propagated unchanged on the way out of network 12a.

[0070] By way of example on how the present invention may be utilized to provide support for the Gnutella protocol, we will now refer to logical flow diagrams 11 to 15. As with the previous discussion with regard to Chart 1, we will be referring to the components of FIG. 3 by way of example.

[0071] Referring now to FIG. 11, a logical flow diagram illustrating the processing of a ping message is shown generally as 160. Beginning at step 162, a ping message is received. At step 164 the ping message is optionally dropped and if dropped, is not propagated within network 12a. At step 166 a forged Pong message is created. The forged pong response may contain the number of files available for P2P exchange within network 12a. The forged pong message

may be sent to each node to connected to PPO 10 in order to train a network as described earlier with reference to ping/pong training module 122 of FIG. 10.

[0072] Referring now to FIG. 12, a logical flow diagram illustrating the processing of a pong message is shown generally as 190. Process 190 begins at step 192 with the receipt of a pong message. At step 194 a test is made to determine if the message is from a node 14 within network 12a. If the message is from a node 14 within network 12a the message may be optionally dropped at step 196. If the message is from a node 14 in a network 12b to 12n, processing moves to step 198 where the TTL for the message is decremented. Processing then moves to step 200 where a test is made to determine the current value of the TTL. If the TTL has expired, the message is dropped at step 202. If the TTL has not expired, processing moves to step 204 where a test is made to determine if a ping message to match the pong message has been received. PPO 10 stores messages it receives under the control of query routing logic module 132 (FIG. 10). Typically a message is not stored for long as most P2P requests for information are resolved within less than a minute. If no matching ping message is found, the pong message is dropped at step 202. If a matching ping message has been seen, then the pong message is forwarded to the source of the original ping message at step 206.

[0073] In the above description of FIG. 12, the inventor makes reference to Time To Life (TTL). TTL information is utilized by the Gnutella protocol, but not by all other protocols. For other protocols not recognizing TTL, the logic of FIG. 11 would be modified to remove steps 198, and 200. Thus control would flow from step 194 in the negative case directly to step 204.

[0074] Referring now to FIG. 13, a logical flow diagram illustrating the processing of a query message is shown generally as 210. Process 210 begins at step 212 where a query message is received. At step 214 a test is made to determine if the query message came from a node 14 within network 12a. If this is the case processing moves to step 216 where a test is made to determine if the requested file is contained within network 12a as indicated by content index 136 (FIG. 10). If the file is contained within network 12a then processing moves to step 218 where the query message is forwarded to nodes 14 having the lowest cost class within network 12a. If the file is not found within network 12a then processing moves to step 220 where the query message is forwarded to a select weighted list of nodes 14 within network 12a. The intent here being that content index 136 may not be current and the requested file may reside within network 12a. One method of determining the set of nodes 14 to send the query message to would be to use a weighted probabilistic function such as a round robin method based upon the number of files available from each node 14 within network 12a. In this way, the query does not always go to a node 14 having the largest number of files.

[0075] A test is next made at step 222 to determine if the file has been located on a node 14 within network 12a. If the file has been located the location information is forwarded to the originator of the query message at step 224. If at step 222 the file has not been located, the query message is forwarded to a weighted subset of connected nodes having the lowest cost class in networks 12b to 12n at step 226. As mentioned before, a weighted round robin scheme may be utilized to select the nodes 14 in networks 12b to 12n to receive the query. A connected node is one that has established a communication path with PPO 10, for example via TCP/IP.

Returning to step 214 if the query message is not from a node 14 within network 12a, processing moves to step 228 where the TTL value of the message is decremented. A test is then made at step 230 to determine if the TTL value for the message is greater than zero. If it is not, then the message is dropped at step 232 and processing ends. If the TTL value is less than or equal to zero then processing moves to step 226 where the query message is forwarded to all connected nodes in networks 12b to 12n. Optionally, if the query is from a node in networks 12b to 12n, the query may simply be dropped or returned to the requesting node at step 226, thus not requiring PPO 10 to forward the query to connected nodes.

[0076] As discussed above with reference to FIG. 12, if the communication protocol does not make use of TTL, then steps 228, 230 and 232 would be deleted. The negative case from step 214 would then flow to step 226.

[0077] Referring now to FIG. 14, a logical flow diagram illustrating the processing of a queryhit message is shown generally as 240. Process 240 begins at step 242 where a queryhit message is received by PPO 10. Processing moves to step 244 where the TTL for the queryhit message is decremented. At step 246 if the TTL is less than or equal to zero then the message is dropped at step 248. If the TTL is greater than zero, processing moves to step 250 where a test is performed to determine if a matching query message had been received for the queryhit message. PPO 10 stores messages it receives under the control of query routing logic module 132 (FIG. 10). Typically a message is not stored for long as most P2P requests for information are resolved within less than a minute. If no matching query message was received, processing moves to step 248 where the message is dropped. If a matching query message was received, processing moves to step 252 where a test is made to determine if the queryhit message was from a node 14 within network 12a. If not, the queryhit message is then optionally forwarded to the node that made the original query at step 254. If at step 252 the queryhit message is determined to have come from a node 14 within network 12a, then processing moves to step 256. At step 256 a test is made to determine if the original query message corresponding to the queryhit message was from a node 14 within network 12a. If so, processing moves to step 254 where the message is forwarded to the node that made the original query. If not, processing moves to step 248 where the message is dropped.

[0078] Referring now to FIG. 15 a logical flow diagram illustrating the processing of a connect request is shown generally as 260. Any node 14 may request a connection with any other node 14 at step 262. At step 264 a test is made to determine if the request is from a node 14 within network 12a. If so, connection manager 124 (see FIG. 10) attempts to service the query through query module 120 (see FIG. 10) to determine a cost efficient path within network 12a, at step 266. If at step 264 it is determined that the connect request is not from a node 14 within network 12a, processing moves to step 268. At step 268 a ping message is sent to connected nodes 14 within networks 12b to 12n. At step 270 one or more pongs are received and a decision is made at step 272 which connection to a specific external node 14 should be utilized. Step 272 may utilize a variety of methods to determine which connections to keep and which to drop. Typically, step 272 would maintain connections based upon the amount of data, cost class, and the total number of connections that may be maintained. If at step 272 a node 14 is found to be no better than an existing connection, it is dropped at step 274. If at step 272 a better connection is found, it is added to content index 136 at step 276.

[0079] Although this disclosure and the claims appended hereto make use of the terms query, queryhit, ping, pong, push and connect, it is not the intent of the inventor for these terms to be specifically associated with the Gnutella protocol. To the inventor the term query is analogous to a request for data and queryhit to a reply to a query, indicating that the data has been located. A ping is a standard computer communications term and is short for Packet Internet Groper; in essence it is a message to determine whether a specific computer in a network is accessible. A pong is a response to a ping. A push is a message sent directly to a node that is protected by a firewall. A push is used to request a direct connection between the node behind the firewall and the node sending the push message so that the node behind the firewall can "push" data to the requesting node. A connect is a connection between two nodes.

[0080] Although the disclosure refers to a PPO within an ISP by way of example, it is not the intent of the inventor to restrict the invention to such a configuration. For example a PPO may be used within any network, including networks utilized by corporations to exchange data with their employees or customers. Further, multiple PPO's may be utilized to provide redundancy in case one PPO fails and also to provide load balancing. In the case of a network 12 utilizing a single PPO, if the PPO failed, network 12 would revert to the status quo without the PPO; i.e. all P2P messages are exchanged with no decision made on who should service the request.

[0081] Although the invention has been described with reference to certain specific embodiments, various modifications thereof will be apparent to those skilled in the art without departing from the spirit and scope of the invention as outlined in the claims appended hereto.

I claim:

1. A peer to peer optimizer, said optimizer examining peer to peer messages between nodes within networks connected to said optimizer, for the purpose of optimizing behavior on each of said networks.
2. The optimizer of claim 1 wherein said optimizer utilizes path cost information in optimizing behavior.
3. The optimizer of claim 2 wherein said path cost information is derived from cost class information.
4. The optimizer of claim 1 wherein a peer to peer message may be redirected.
5. The optimizer of claim 1 wherein a peer to peer message may be modified.
6. The optimizer of claim 1 wherein a peer to peer message may be forwarded.
7. The optimizer of claim 1 wherein a peer to peer message may not be forwarded.
8. The optimizer of claim 1 wherein new peer to peer messages may be created.
9. The optimizer of claim 1 wherein if a peer to peer message is a query message, forwarding said query message to an optimized set of nodes.
10. The optimizer of claim 9 further comprising a content index.
11. The optimizer of claim 10 wherein said content index comprises a mapping of content to nodes.
12. The optimizer of claim 11 wherein said content index is used to limit the number of nodes a query message is forwarded to.
13. The optimizer of claim 11 wherein said content index is used to determine said optimized set of nodes.
14. The optimizer of claim 13 wherein said optimized set of nodes is selected based upon path cost.

15. The optimizer of claim 13 wherein if no nodes are found in said content index to satisfy said query message, forwarding said query message to a set of nodes determined by a weighted probabilistic function.

16. The optimizer of claim 1 wherein if a peer to peer message is a queryhit message, said optimizer modifying quality of service parameters of said queryhit message.

17. The optimizer of claim 1 wherein if a peer to peer message is a ping message from a first node, said optimizer creating one or more pong messages to nodes in the same cost class as said first node.

18. The optimizer of claim 17 wherein cost class information for a plurality of nodes is stored in a host cache.

19. The optimizer of claim 1 wherein if a peer to peer message is a connect, and if said connect is better than an existing connection, dropping said existing connection for said connect.

20. The optimizer of claim 3 wherein said cost class information is set by an administrator of said optimizer.

21. The optimizer of claim 3 wherein said cost class information is set by said optimizer.

22. A peer to peer optimizer, said optimizer examining peer to peer messages between nodes within networks connected to said optimizer, for the purpose of determining a cost efficient path for each of said peer to peer messages.

23. The optimizer of claim 22 wherein said optimizer in determining said cost efficient path combines a cost class for each of two nodes to create a path cost.

24. The optimizer of claim 22 wherein if a peer to peer message is not initially directed to said cost efficient path, redirecting said peer to peer message to said cost efficient path.

25. The optimizer of claim 22 wherein if a peer to peer message is not from a node in a first network and is a query message, forwarding said query message to one or more nodes not in said first network.

26. The optimizer of claim 22 wherein if a peer to peer message is from a node in a first network and is a query message and if said query message is unsuccessful, forwarding said query message to a weighted list of nodes in said first network.

27. The optimizer of claim 22 wherein if a peer to peer message is not from a node in a first network and is a query message, ignoring said query message.

28. The optimizer of claim 22 wherein if a peer to peer message is a queryhit message, determining if a matching query message has been observed and if so forwarding said queryhit message to the originator of said matching query message.

29. The optimizer of claim 22 wherein if a peer to peer message is a query message said optimizer determining;

a) if said query is from a node in a first network, determining if said query may be serviced by one or more nodes in said first network, if so, forwarding said query to said one or more nodes in said first network; or

b) if said query is from a node not in said first network, dropping said query.

30. The optimizer of claim 22 wherein if a peer to peer message is a queryhit message, forwarding said queryhit message to the originator of a matching query message.

31. The optimizer of claim 22 wherein if a peer to peer message is a ping message from a node in a first network dropping said ping message so that it does not propagate from said first network.

32. The optimizer of claim 22 wherein if a peer to peer message is a pong and a matching ping message has not been seen, dropping said pong.

33. The optimizer of claim 22 wherein if a peer to peer message is a ping, said optimizer optionally dropping said ping.

34. The optimizer of claim 33 wherein if said optimizer has not dropped said ping, said optimizer providing a forged pong message.

35. The optimizer of claim 22 wherein if a peer to peer message is a connect, and if said connect is better than an existing connection, dropping said existing connection for said connect.

36. The optimizer of claim 23 wherein said cost class is set by an administrator of said optimizer.

37. The optimizer of claim 23 wherein said cost class is set by said optimizer.

38. The optimizer of claim 22 wherein each peer to peer message is examined to obtain information about the originator of said message.

39. The optimizer of claim 38 wherein said information is stored by said optimizer in a host cache.

40. A process for managing peer to peer messages between and within networks, said process comprising the step of determining a cost efficient path for each of said peer to peer messages.

41. The process of claim 40 further comprising the step of utilizing a message to obtain information about the originator of said message.

42. The process of claim 40 further comprising the step of matching a requested filename filenames stored in a content index.

43. The process of claim 42 further comprising the step of utilizing a content index in determining said cost efficient path.

44. The process of claim 40 further comprising the step of utilizing cost class information to determine said cost efficient path.

45. The process of claim 40 further comprising the step of if a peer to peer message is not initially directed to said cost efficient path, redirecting said peer to peer message to said cost efficient path.

46. The process of claim 40 further comprising the step of modifying a peer to peer message.

47. The process of claim 40 further comprising the step of forwarding a peer to peer message.

48. The process of claim 40 further comprising the step of not forwarding a peer to peer message.

49. The process of claim 40 further comprising the step of creating a new peer to peer message.

50. The process of claim 40 further comprising the step that if said peer to peer message is a query message, forwarding said query message to an optimized set of nodes.

51. A computer readable medium containing instructions for managing peer to peer messages between nodes in networks, said medium comprising instructions for optimizing behavior on said networks.

52. The medium of claim 51 further comprising instructions to utilize path cost information in optimizing behavior.

53. The medium of claim 52 further comprising instructions to derive said path cost information from cost class information.

54. The medium of claim 51 further comprising instructions for redirecting a peer to peer message.

55. The medium of claim 51 further comprising instructions for modifying a peer to peer message.

56. The medium of claim 51 further comprising instructions for forwarding a peer to peer message.

57. The medium of claim 51 further comprising instructions for not forwarding a peer to peer message.

58. The medium of claim 51 further comprising instructions for creating a new peer to peer message.

59. The medium of claim 51 further comprising instructions to determine if a peer to peer message is a query message, and if so, forwarding said query message to an optimized set of nodes.

60. The medium of claim 59 further comprising instructions to implement a content index.

62. The medium of claim 60 further comprising instructions to map content to nodes.

63. The medium of claim 60 further comprising instructions to utilize said content index to limit the number of nodes a query message is forwarded to.

64. The medium of claim 63 further comprising instructions to utilize said content index to determine said optimized set of nodes.

65. The medium of claim 64 further comprising instructions to select said optimized set of nodes based upon path cost.

66. The medium of claim 64 further comprising instructions to determine that if no nodes are found in said content index to satisfy said query message, forwarding said query message to a set of nodes determined by a weighted probabilistic function.

67. The medium of claim 51 further comprising instructions to determine if a peer to peer message is a queryhit message, if so, modifying quality of service parameters of said query hit message.

68. The medium of claim 51 further comprising instructions to determine if a peer to peer message is a ping message from a first node, if so, creating one or more pong messages to nodes in the same cost class as said first node.

69. The medium of claim 53 further comprising instructions for storing said cost class information in a host cache.

70. The medium of claim 51 further comprising instructions to determine if a peer to peer message is a connect, if so, and if said connect is better than an existing connection, dropping said existing connection for said connect.

71. The medium of claim 69 further comprising instructions to permit an administrator to set said cost class information.

72. The medium of claim 69 further comprising instructions to set said cost class information without human intervention.

73. The medium of claim 51 further comprising instructions for determining if a message is from a node in a first network, and instructions for preventing said message from reaching a node in a second network.

80. The medium of claim 51 further comprising instructions for utilizing a message to obtain information about the node from which said message originated.

81. The medium of claim 80 further comprising instructions to populate a host cache with said information.

82. The medium of claim 51 further comprising instructions to determine if a peer to peer message is not initially directed to a cost efficient path, redirecting said peer to peer message to said cost efficient path.

83. A system for optimizing peer to peer messages between nodes within networks, said system comprising a peer to peer optimizer, said system utilizing said optimizer to examine messages between said nodes for the purpose of optimizing behavior on each of said networks.

84. The system of claim 83 wherein said system further comprises means to utilize path cost information in optimizing behavior.

85. The system of claim 84 wherein said system comprises means to derive said path cost information from cost class information.

86. The system of claim 83 further comprising means for redirecting a peer to peer message.

87. The system of claim 83 further comprising means for modifying a peer to peer message.

88. The system of claim 83 further comprising means for forwarding a peer to peer message.

89. The system of claim 83 further comprising means for not forwarding a peer to peer message.

90. The system of claim 83 further comprising means for creating a new peer to peer message.

91. The system of claim 83 further comprising means to determine if a peer to peer message is a query message, and if so, utilizing means to forward said query message to an optimized set of nodes.

92. The system of claim 91 further comprising a content index.

93. The system of claim 92 wherein said content index comprises a mapping of content to nodes.

94. The system of claim 93 wherein said content index is used to limit the number of nodes a query message is forwarded to.

95. The system of claim 93 further comprising means to utilize said content index to determine said optimized set of nodes.

96. The system of claim 95 further comprising means to select said optimized set of nodes based upon path cost.

97. The system of claim 95 further comprising means to determine if any nodes are found in said content index to satisfy said query message, if no nodes were found, utilizing means for forwarding said query message to a set of nodes determined by means of calculating a weighted probabilistic function.

98. The system of claim 83 further comprising means to determine if a peer to peer message is a queryhit message, and if so utilizing means for modifying the quality of service parameters of said query hit message.

99. The system of claim 83 further comprising means to determine if a peer to peer message is a ping message and if so and if said peer to peer message is a ping message from a first node, utilizing means to create and send one or more pong messages to nodes in the same cost class as said first node.

100. The system of claim 99 further comprising cost class information for a plurality of nodes, said cost class information stored in a host cache.

101. The system of claim 83 further comprising means for determining if a peer to peer message is a connect, and if said connect is better than an existing connection, means for dropping said existing connection for said connect.

* * * * *