

(19) World Intellectual Property  
Organization  
International Bureau



(43) International Publication Date  
19 February 2004 (19.02.2004)

PCT

(10) International Publication Number  
**WO 2004/015950 A1**

(51) International Patent Classification<sup>7</sup>: **H04L 29/06**,  
G06F 17/30

N2J 4H2 (CA). **BOWMAN, Don** [CA/CA]; 5-407 Keats  
Way, Waterloo, Ontario N2L 5S7 (CA).

(21) International Application Number:  
PCT/CA2003/000799

(74) Agent: **BERESKIN & PARR**; 40 King Street West, 40th  
Floor, Toronto, Ontario M5H 3Y2 (CA).

(22) International Filing Date: 29 May 2003 (29.05.2003)

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU,  
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU,  
CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH,  
GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC,  
LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW,  
MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE,  
SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ,  
VC, VN, YU, ZA, ZM, ZW.

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
10/216,676 12 August 2002 (12.08.2002) US

(63) Related by continuation (CON) or continuation-in-part  
(CIP) to earlier application:  
US 10/216,676 (CON)  
Filed on 12 August 2002 (12.08.2002)

(84) Designated States (*regional*): ARIPO patent (GH, GM,  
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),  
Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),  
European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE,  
ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO,  
SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM,  
GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(71) Applicant (*for all designated States except US*): **SAND-  
VINE INCORPORATED** [CA/CA]; 408 Albert Street,  
Waterloo, Ontario N2L 3V2 (CA).

(72) Inventors; and

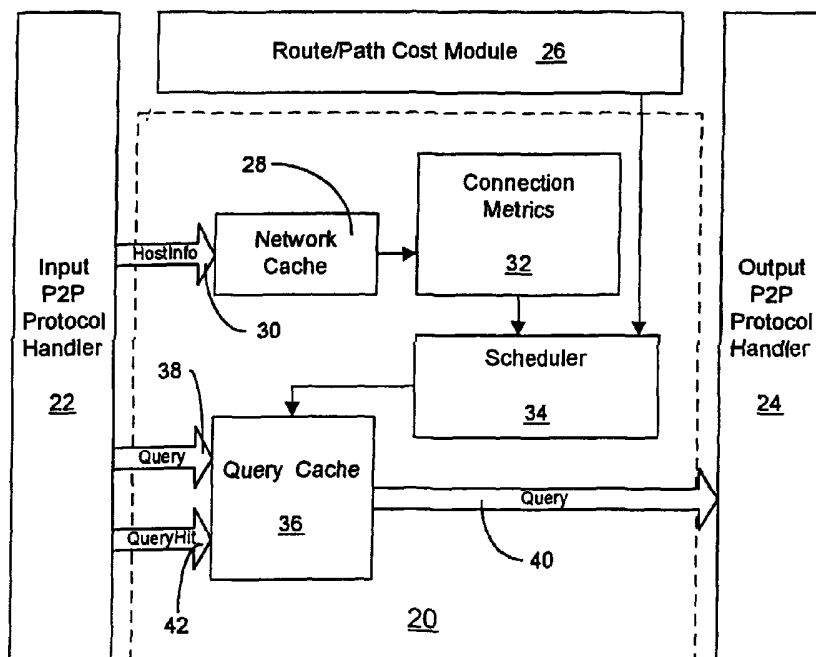
(75) Inventors/Applicants (*for US only*): **LOACH, Scot**  
[CA/CA]; 503-300 Regina Street North, Waterloo, Ontario

Published:

— with international search report

[Continued on next page]

(54) Title: HEURISTICS-BASED ROUTING OF A QUERY MESSAGE IN PEER TO PEER NETWORKS



(57) Abstract: The present invention is directed to a system and method for administering Peer to Peer (P2P) query messages within a P2P network. The present invention collects a plurality of metrics for each host that it is aware of in a P2P network, most often by a host information or query hit message. The metrics collected aid in determining a set of P2P hosts best able to fulfill a query message, without having knowledge of specific content. The metrics collected also aid in managing query messages in that they determine when to drop query messages or when to resend query messages.



- 
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

## HEURISTICS-BASED ROUTING OF A QUERY MESSAGE IN PEER TO PEER NETWORKS

**FIELD OF THE INVENTION**

5           The present invention relates to the field of Peer to Peer (P2P) messaging in a computer network. More specifically, the present invention relates to determining from which P2P hosts to request content, without the use of a content index.

10       **BACKGROUND OF THE INVENTION**

          Peer to peer (P2P) network protocols are used to exchange content among peers on a network. Generally one peer will be connected to a number of other peers. When a peer wants to find content, the peer  
15       transmits a query message to some of the peers it is connected to. This query message travels through the P2P network and elicits query responses from peers that have content that satisfies the query.

          In a broadcasting model of P2P networks, a query is sent through  
20       the network by assigning it a Time-to-live (TTL) and broadcasting it to all connected peers. The connected peers decrement the TTL and broadcast it to all of their connected peers, and the query continues through the network in this fashion until the TTL reaches zero. This method works well when each peer is connected to a small number of other peers, and when  
25       there are few cycles in the network. However, the number of times the query is forwarded grows exponentially with the number of connected peers. For a peer that connects to thousands of other peers, broadcasting queries is not feasible, since the network would become congested with messaging traffic.

One method of resolving the possibility of network congestion is to maintain an index of the content available through each connection. Queries may then be forwarded to connections that have the content being searched for. However, keeping such an index may not be feasible. For  
5 example, due to the size of the index or the dynamic nature of the content.

Thus there is a need for a system and method to intelligently route queries to connected peers, without knowing the specific content available at each peer. The present invention addresses this need.  
10

### **SUMMARY OF THE INVENTION**

The present invention relates to the routing of query messages in a peer to peer network.

15 One aspect of the present invention is a heuristics based peer to peer query routing system, the system comprising:

- a) a network cache module;
- b) a connection metrics module operatively connected to the network cache module;
- 20 c) a scheduler module operatively connected to the connection metrics module; and;
- d) a query cache module operatively connected to the scheduler module.

25 In another aspect of the present invention there is provided a method of routing messages in a peer to peer network, the method comprising the step of upon receiving a message determining the type of

the message and if the message is a query, forwarding the message to a heuristically selected host.

In another aspect of the present invention there is provided a method for managing queries in a peer to peer network device the method comprising the steps of:

- a) issuing a timer event; and
- b) if the time for a host known to the device has expired, discarding the host.

10

In another aspect of the present invention, a scheduler module is initialized by creating a set of all known hosts, and when a request is made to connect to a host providing a host from the set.

In yet another aspect of the present invention, there is provided a system for routing queries in a peer to peer network, the system utilizing heuristics to determine the best host to forward a query to.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

20

Figure 1 is a block diagram of networks connected to a Peer to Peer Optimizer (PPO);

Figure 2 is a block diagram of the components of the present invention;

Figure 3 is a block diagram of the data structures utilized by the present invention;

Figure 4 is a flowchart of the logical flow of the reception of a host information message;

Figure 5 is a flowchart of the logical flow of the reception of a query message;

Figure 6 is a flowchart of the logical flow of the reception of a query hit message

5        Figure 7 is a flowchart of the logical flow of establishing a connection;

Figure 8 is a flowchart of the logical flow of terminating a connection;

Figure 9 is a flowchart of the logical flow upon receiving a timer event; and

10        Figure 10 is a flowchart of the logical flow of a scheduler.

#### **DETAILED DESCRIPTION OF THE INVENTION**

The present invention uses heuristics to route a query to a subset of  
15    connected peers, without knowing the specific content available through  
any connected peers. The present invention may be implemented within a  
Path Optimizer for Peer-to-Peer Networks (PPO). An example of a PPO is  
disclosed in a co-pending application titled "Path Optimizer For Peer To  
Peer Networks", application number 10/138,336, filed on May 6, 2002, the  
20    entire contents of which are incorporated herein by reference

Figure 1 is a block diagram of networks connected to a PPO. 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  
25    network controlled by an ISP, a corporate network, or a University network.  
Networks 12 would typically be connected to PPO 10 via the Internet, but  
that is not a requirement of the present invention. Within networks 12 are  
one or more nodes 14. A node 14 is any computer that is capable of  
receiving or transmitting P2P data and is referred to hereinafter as a "peer".

Any network 12 that is capable of providing or requesting P2P traffic may make use of PPO 10. The present invention is preferably part of PPO 10, however it is not the intent of the inventors to restrict the present invention to use solely within PPO 10.

5

Figure 2 is a block diagram of the components of the present invention, shown as heuristics based query routing system 20. The large arrows show the flow of data obtained from messages received from different P2P communication protocols. The small arrows show the flow of information between modules in system 20. System 20 may make use of three components in PPO 10 namely: input P2P protocol handler 22, output P2P protocol handler 24 and route/path cost module 26. Handler 22 parses a specific P2P protocol and passes the data received to components within system 20. There may be many instances of handler 22, one or more for each different protocol. Handler 24 transmits forwarded messages. Route/Path Cost Module 26 categorizes connected peers into different "classes", where each class has a cost relative to each other class. The information about the cost of a connection is used by the present invention to select a subset of connections that will be candidates for a scheduling algorithm. The invention does not require this information to work, and it is not the intent of the inventors to restrict the invention to require module 26. Similarly, as one skilled in the art will understand, handlers 22 and 24 may be replaced by equivalent components that serve the same function.

25

Handler 22 passes data from three types of P2P messages to system 20, namely:

- 1) Host Information messages 30 contain information about a peer that contains files that may be downloaded, such peers are referred to as "hosts", in a P2P network. This information varies according to the

30

specific P2P protocol, but may consist of the amount of content offered by a host, the bandwidth available to a host, the uptime of a host, etc. These messages may be called by other names within specific protocols, and protocol-specific handlers may obtain some of the information from different  
5 types of messages. For example, in the Gnutella P2P protocol, a host Information message 30 is referred to as a "Pong, and information about whether a host is behind a firewall is obtained from a query hit message.

2) Query messages 38 contain a description of content that a peer is searching for, and a globally unique identifier (GUID) that is used to  
10 route query hits back to the originating peer.

3) Query hit messages 40 contain information about content that matched a query. Each query hit message contains a list of results from a single peer, and a GUID that matches the GUID of the query. This GUID is used to route the query hit back to the originator of the query.

15

The data flow shown in Figure 2 is not intended to map to specific protocol messages, but rather show the communication of this information, which is normalized by handler 22 into the general messages types described.

20

Network cache module 28 maintains a cache of all hosts that have been detected in a P2P network. Hosts may be detected in many ways. One way to learn about hosts is by an explicit "Host Information Message" 30. For example, the Gnutella P2P protocol specifies a "Pong" message to  
25 advertise the presence of a host, and information about it. Another way to learn about hosts is by inspecting the contents of messages that are routed through module 20. For example, a "Query Hit" message 38 may have been received that has been forwarded through several hosts. This message will have the IP address and listening port of the host that sent it,  
30 and although only the final receiver of the message requires this

information, the information can be used to learn about the host that sent the query hit message 38.

Network cache module 28 saves collected information about hosts, and passes relevant information on to connection metrics module 32, which is described below. Since network cache 28 has no general way of knowing when a host is removed from the network, it removes cache entries after a certain amount of time has passed without any reference to that host. When a directly connected host disconnects from the network, that host and any hosts that were reachable through that host are removed from the cache.

Connection metrics module 32 maintains a set of heuristics based metrics for each P2P connection that is established with system 20. These metrics are obtained from network cache module 28. Some of the metrics are attributes of the connected host, and others are aggregates calculated from all hosts reachable through the connected host. Module 32 provides metrics to scheduler 34, which uses them as input to a weight-based scheduling algorithm. The specific metrics used are described below.

20

Scheduler 34 utilizes the input from connection metrics module 32 and route/path cost module 26 (if available) in a probabilistic weight-based algorithm. For example, a weighted round-robin algorithm may be used. The output of the algorithm is a sequence of connected hosts, which is used by query cache 36 to make forwarding decisions.

25

Query cache module 36 receives "Query" messages 38, makes a forwarding decision, and forwards the query to one or more specific connections vial query forwarding link 40. Query cache module 36 saves

queries for a certain amount of time to allow them to be re-sent. Module 36 also receives query hit messages 42 and uses them to track in-progress queries. Queries expire from cache 36 after an expiry time. At expiry time, if the number of returned results is under a certain threshold, the query may  
5 be forwarded again, to a different set of hosts. This scheme of query feedback results in a savings in bandwidth, since the initial number of hosts a query must be forwarded to is reduced. Only queries searching for less common content should have to be re-forwarded.

10 Figure 3 is a block diagram of the data structures utilized by system 20. One skilled in the art will appreciate that any form of data structures may be utilized to store the data required for the present invention.

System 20 comprises four tables of data, namely: network host table  
15 50, connection table 52, query table 54 and forwarded table 56. Host table 50 is utilized by network cache module 28, connection table 52 is utilized by connection metrics module 32 and both query table 54 and forwarded table 56 are utilized by query cache module 36. In Figure 3, each of the tables 50, 52, 54 and 56 is shown as a block containing a list of fields.  
20 Fields above the first horizontal line are the "index" fields, and uniquely identify a row in a table.

Host table 50 is indexed by an "IP Address" field containing the address of a host, and a "Connection" field containing the IP address of a  
25 connection. A connection may be a network 12 that contains a plurality of nodes 14 (see Figure 1). A connection may also be a single node. The "Connection" field of table 50 contains the IP address of a connection and the "IP Address" field contains the IP address of a node within the connection that can serve as a host. The connection field references a row  
30 in connection table 52, indexed by the "Connection" field of connection

table 52. Connection table 52 is described in detail later. The "Hops" field contains the number of network hops to the host. The "Content" field is the amount of content advertised by the host. This may be measured in files, bytes, or some other unit. The "Firewalled" field is either on or off, depending upon whether or not the host is behind a firewall. The "Expire Time" field is used to expire hosts from the cache. Each time there is evidence that a host is still on the network, the value of this field is set to the current time plus a constant interval, for example 10 minutes. An entry (i.e. a row), is deleted from table 50 when the current time passes the value in the expire time field.

Connection table 52 contains the heuristics that are used by scheduler 34. Table 52 is indexed by the IP address of the connected host as shown by the "Host" field. The "Total Content" field contains an aggregation of all of the values contained in the "Content" field of host table 50. This provides a measure of the amount of content available through a connection. Connections with more content are given a higher weight by scheduler 34. The "Uptime" field contains the amount of time the connected host has been connected to the P2P network. This may be used to indicate the reliability of the host. Connections with a higher uptime are given a higher weight by scheduler 34. The "Bandwidth" field contains the amount of bandwidth to a connected node. This is useful in determining how many messages can be forwarded to the connected node, since a low-bandwidth node cannot support many messages in a short time. Scheduler 34 gives a higher weight to a higher bandwidth connection. The "Queries Transmitted" field contains a count of the number of queries that have been forwarded to the connection. The "Hits Received" field contains a count of the number of query hits that have been received from the connection. The value of the "Hit Rate" field is calculated by dividing the value in the "Queries Transmitted" field with the value in the

"Hits Received" field. The value of the "Hit Rate" field indicates how well the connected network is at satisfying queries, and is a useful metric for scheduler 34. A high hit rate will be given a high weight by scheduler 34. The "% Firewalled" field is aggregated from the "Firewalled" fields of host  
5 table 50. A connection that leads to a network with a low number of firewalled hosts is given a higher weight by scheduler 34, since downloading from those hosts has a higher chance of success. The "Total Hosts" field provides an estimate of the size of the connected network. Forwarding queries to a large network generally gives a higher rate of  
10 success; therefore, larger networks are given a higher weight by scheduler 34. The "Count" field contains the number of times this host is connected. It is possible for a host to connect multiple times, and it is necessary to track this because the row should only be deleted when the count becomes zero.

15

These metrics are merely a few of the metrics that may be applied to improve the heuristics of P2P message routing. The design is such that other metrics may be easily added by adding fields (i.e. columns in the relational model) to host table 50 and connection table 52. It is not the  
20 intent of the inventors to require that the metrics suggested be always stored in a table, for example the value of the field "Hit Rate" in table 32 may be calculated as needed and not stored in table 32

Query cache module 36 utilizes two tables, query table 54 and  
25 forwarded table 56. Query table 54 is a cache of in progress queries. Each time a query is received by query cache module 36, the query is entered into table 54. The "QueryID" field contains the Global Unique Identifier (GUID) of the query, which is provided with the query. Any incoming queries that already have a GUID in table 54 are duplicates and  
30 may be dropped. The "Hits" field is initialized to 0 when a query is first

forwarded. Whenever a query hit is returned, the value in the hits field is incremented by the number of results in the query hit. A query hit may contain multiple results in response to a query. The "Retries" field is initialized to 0 when the query is first forwarded. A query may be retried if  
5 the number of hits from the query have not met a threshold value (see Figure 9). If a query is retried, the retries field is incremented. The "Data" field contains the actual query data, so that it may be retransmitted. The "Expire Time" field is used to clear the cache after a query has been in the network for a certain length of time. A time interval is set that is expected  
10 to be the maximum lifetime of a query on the network, by way of example, a minute.

Forwarded table 56 contains information on which connected hosts a query was forwarded to. It is filled in with the Query ID field of table 54  
15 and the Host field of table 52 whenever a query is forwarded. On a re-query, this table is checked before forwarding to a host chosen by scheduler 34. If the host is already in table 56 for the query being forwarded, the query will not be forwarded to that connection again and another host is obtained from scheduler 34.

20

We will now refer to Figures 4 to 10 which are flowcharts describing the behaviour of system 20 for each possible input. To aid the reader in understanding Figures 4 to 10, the inventors advise also consulting Figures 2 and 3 to better understand the data and logic flow of the present  
25 invention.

Referring now to Figure 4 a flowchart of the logical flow of the reception of a host information message is shown generally as process 60. A host information message 30 (see Figure 2) is received at step 62. At  
30 step 64 a test is made to determine if the IP Address provided by the host

information message 30 is in host table 50 (see Figure 3). If the IP address is in host table 50 the applicable row is determined at step 66. If the IP address is not in host table 50 a new row in host table 50 is created at step 68. At step 70 the hops, content and firewalled fields are updated for the relevant row based upon the information contained in the host information message. At step 72 the expire time field of the relevant row is set. In one embodiment this may be set at 10 minutes from the current time. At step 74 connection metrics module 32 is invoked to update its entry for the connection. Processing ends at step 76.

10

Referring now to Figure 5 a flowchart of the logical flow of the reception of a query message is shown generally as 80. Query message 38 (see Figure 2) is received at step 82. The value of "k" is the number of connections query message 38 should be forwarded to, the value of k may be selected by the user when configuring system 20. A test is then made at step 84 to determine if this is a re-query. A re-query is submitted by step 196 of Figure 9 and includes a flag indicating it is a re-query. A re-query is issued when the original query has not received enough hits to meet a threshold value. If the query is a re-query, processing moves to step 92. If the query is not a re-query, processing moves to step 86 where a test is made to determine if the query exists in query table 54, thus indicating it is a duplicate query. If the query already exists in query table 54 then the query has already been forwarded, so it is dropped at step 88. If the query is not in query table 54 it is inserted in query table 54 at step 90. Processing then moves to step 92 where the query is forwarded to a connection obtained from scheduler 34, k times. For each connection received from scheduler 34, forwarded table 56 is checked at step 94 to see if the query has already been forwarded to the connection; if it has, control returns to step 92 where a new connection is obtained from scheduler 34. At step 96 an entry is inserted into forwarded table 56 for the

forwarded query. At step 98 the value of k is decremented as the query has been forwarded to a connection. At step 100 the "Queries Transmitted" value for the connection in connection table 52 is incremented. A test is then made at step 102 to determine if the value of k is equal to zero. If it is, processing finishes at step 104. If it is not, processing returns to step 92 to obtain another connection for the query.

Referring now to Figure 6 a flowchart of the logical flow of the reception of a query hit message is shown generally as 110. Query hit message 42 (see Figure 2) is received at step 112. At step 114 a test is made to determine if a query corresponding to query hit 42 is present in query table 54, based upon the GUID of the query. If no corresponding query is present, processing ends at step 120 as this is a spurious query hit message or a query hit message from an expired query. If query table 54 does contain a corresponding query, processing moves to step 116 where the "Hits" field of query table 54 is incremented by the number of results indicated in the query hit. Processing next moves to step 118 where the "Hits Received" field of connection table 52 is updated. Processing then ends at step 120.

20

Referring now to Figure 7 a flowchart of the logical flow of establishing a connection is shown generally as 130. Beginning at step 132 system 20 has determined that a connection to a host is to be established. A test is made at step 134 to determine if there is a row having the IP address of the host, contained in the "Connection" field of connection table 32. If such a row exists, then there exist multiple connections from the host and processing moves to step 136 where the "Count" field of connection table 32 is incremented. Processing then ends at step 140. Returning to step 134, if the test is negative, this is a new host and processing moves to step 138. At step 138 a new row is added to

30

connection table 32 where the field "Connection" is set to the IP address of the host, and field "Count" is set to the value one. Processing then ends at step 140.

5 Referring now to Figure 8 a flowchart of the logical flow of terminating a connection is shown generally as 150. Beginning at step 152 a connection is disconnected. A connection would be most often disconnected by a host, however, system 20 may choose to disconnect a host based upon a variety of parameters, for example the number of  
10 connections it can maintain. At step 154 the row corresponding to the connection to be disconnected is located in connection table 52 and the value of the "Count" field is decremented at step 156. A test is then made at step 158 to determine if the value of the "Count" field is equal to zero. If the test is negative there are still connections from this host, so, processing  
15 moves to step 160 where processing ends. If the test at step 158 is positive, processing moves to step 162 where any rows corresponding to the connection are deleted from host table 50. At step 164 all rows in forwarded table 56 corresponding to the connection are deleted. Processing then moves to step 166 where the connection is removed from  
20 connection table 52.

Figure 9 is a flowchart of the logical flow upon receiving a timer event, shown generally as process 170. Process 170 begins at step 172 when a timer event is received. A timer event may be set by system 20 to  
25 trigger at any interval, for example, once every second. Upon receipt of a timer event, processing moves to step 174 where any expired rows in host table 50 are deleted. At step 176 connection metrics module 32 is invoked to update its metrics to reflect any changes. At step 178 a check is made of query table 54 to determine if any queries have expired. If there are no  
30 expired queries processing ends at step 180. If there are expired queries,

processing moves to step 182 where any expired queries are selected from query table 54. For each selected query, a test is made at step 184 to determine if the value in the "Hits" is above a threshold, if so, the query has completed successfully and processing moves to step 186. At step 186 all  
5 rows in forwarded table 56 matching the expired query are deleted. Processing then moves to step 188 where the expired query is deleted from query table 54 and processing returns to step 178 to determine if there are any more expired queries. Returning now to step 184, if the test is negative, processing moves to step 190 where a test is made to  
10 determine if the value contained in the "Retries" field of query table 54 is above a threshold. If so, the query has been retried the maximum number of times, and should not be re-tried again and processing moves to step 186. If the test at step 190 indicates the threshold has not been reached, processing moves to step 192, as the query will be retried. At step 192 the  
15 value of the field "Retries" for the query in query table 54 is updated and processing moves to step 194. At step 194 the value of the field "Expire Time" for the query in query table 54 is reset, for example to a value of ten minutes past the current time. At step 196 the query is re-issued according to the process described with relation to Figure 5. Process 170 then  
20 returns to step 178.

Referring now to Figure 10 a flowchart of the logical flow of scheduler 34 is shown generally as process 200. At step 202 scheduler 34 receives a request to initialize and moves to step 204. At step 204 a set of  
25 connected hosts is created. If route/path module 26 (see Figure 2) is present, then the sets of connected hosts are created based upon cost class. If not then a single set is created. At step 206 a weight is calculated for each connected host. The weight is calculated utilizing the connection metrics in the connection table 52. A better connection is given a higher  
30 weight. Any number of algorithms may be used to calculate the weight of a

connection. One example, using the fields (i.e. metrics) of connection table 52 follows.

To calculate the weight, each metric is scaled to a number between  
5 1 and 100 by taking the percentage of the metric compared to the  
maximum of that metric for all connections. For example, if there are three  
connections, A, B, and C, and A has total content of 3,000KB, B has total  
content of 30,000KB, and C has total content of 300,000KB, then A's total  
content for the weight equation will be 1, B's will be 10, and C's will be 100.  
10 Once these scaled metrics have been obtained, the relative weights of  
each metric may be configured by the user by specifying the relative  
importance of each metric. The following may be a good choice for some  
users as it sends queries out to networks that are most likely to provide  
results:

15

Total Content: 2

Uptime: 1

Bandwidth: 1

Hit Rate: 10

20 %Firewalled: 1

From the above:

weight = 2\*Total content + 1\*Uptime + 1\*Bandwidth + 10\*Hit Rate + 1\*  
25 Firewalled

The preceding algorithm serves as a singular example of how scheduler 34  
may determine a weight for a specific host. It is not the intent of the

inventors to restrict the present invention to this specific means of weighting a host.

At step 208 scheduler 34 receives requests for the "next host" in a specified cost class from query cache module 36. If no cost class information is available, then only a single class exists. At step 210 a host from the corresponding class is randomly selected, and its weight is decreased by one. At step 212 a test is made to determine if the weight associated with a host is now zero. If the weight is not zero, processing moves to step 214 where the selected host is returned to step 208. If at step 212 the weight of the selected host is zero, the host is removed from the set to which it belongs at step 216. At step 218 a test is made to determine if the set to which the host belongs is empty. If it is, the set is reinitialized at step 220 by connection metrics module 32. If the set is not empty processing moves to step 214 where the host removed at step 216 is returned.

In the disclosure the inventors make reference to various threshold values. These values will be set by default, but they may be modified by the user to best match the system requirements of the user. For example, the default value of "Retries" in query table 54 may be adjusted by the user based upon the bandwidth cost of receiving query hits balanced against the amount of memory available to store the data provided in a query hit.

Although the present invention has been described as being implemented in software, one skilled in the art will recognize that it may be implemented in hardware as well. Further, it is the intent of the inventors to include computer readable forms of the invention. Computer readable forms meaning any stored format that may be read by a computing device.

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.

**WE CLAIM**

1. A heuristics based peer to peer query routing system, said system comprising:
  - 5 a) a network cache module;
  - b) a connection metrics module operatively connected to said network cache module;
  - c) a scheduler module operatively connected to said connection metrics module; and;
  - 10 d) a query cache module operatively connected to said scheduler module.
2. The system of claim 1 further comprising: a host table database, a connection table database, a query table database and a forwarded table  
15 database.
3. The system of claim 1 further comprising means for determining a type of message received by said system and forwarding said type of message to an appropriate module.  
20
4. The system of claim 3, wherein if said message is a query message, forwarding said message to a heuristically determined host.
5. A method of routing messages in a peer to peer network, said  
25 method comprising the step of upon receiving a message determining the type of said message and if said message is a query, forwarding said message to a heuristically selected host.

6. The method of claim 5 further comprising the step of upon receiving a host information message, using the contents of said host information message to update a network cache.
- 5 7. The method of claim 5 further comprising the step of upon receiving a query message using the contents of said query message to update a query cache.
8. The method of claim 5 further comprising the step of upon receiving  
10 a query hit message using the contents of said query hit message to update a query cache.
9. The method of claim 5 wherein said host is selected based upon a weighting algorithm.
- 15 10. The method of claim 9 wherein said weighting algorithm utilizes metrics from the set of metrics consisting of: total content, uptime, bandwidth, hit rate and firewalled.
- 20 11. A method for managing queries in a peer to peer network device said method comprising the steps of:
- a) issuing a timer event; and
  - b) if the time for a host known to said device has expired, discarding said host.
- 25 12. The method of claim 11 further comprising the step:
- c) if a query forwarded by said device has expired, deleting said query from said device.

13. The method of claim 11 further comprising the step:

c) if a query issued by said device has met a threshold of hits, deleting said query from said device else re-issuing said query.

5

14. A system wherein a scheduler module is initialized by creating a set of all known hosts, and when a request is made to connect to a host providing a host from said set.

10 15. The system of claim 14 wherein the providing of a host from said set is done randomly.

16. The system of claim 14 wherein a weight is established for each host, said weight being decremented each time a host is provided, and if  
15 said weight reaches zero, removing said host from said set of known hosts.

17. The system of claim 14 wherein if said set is empty, reinitializing said set

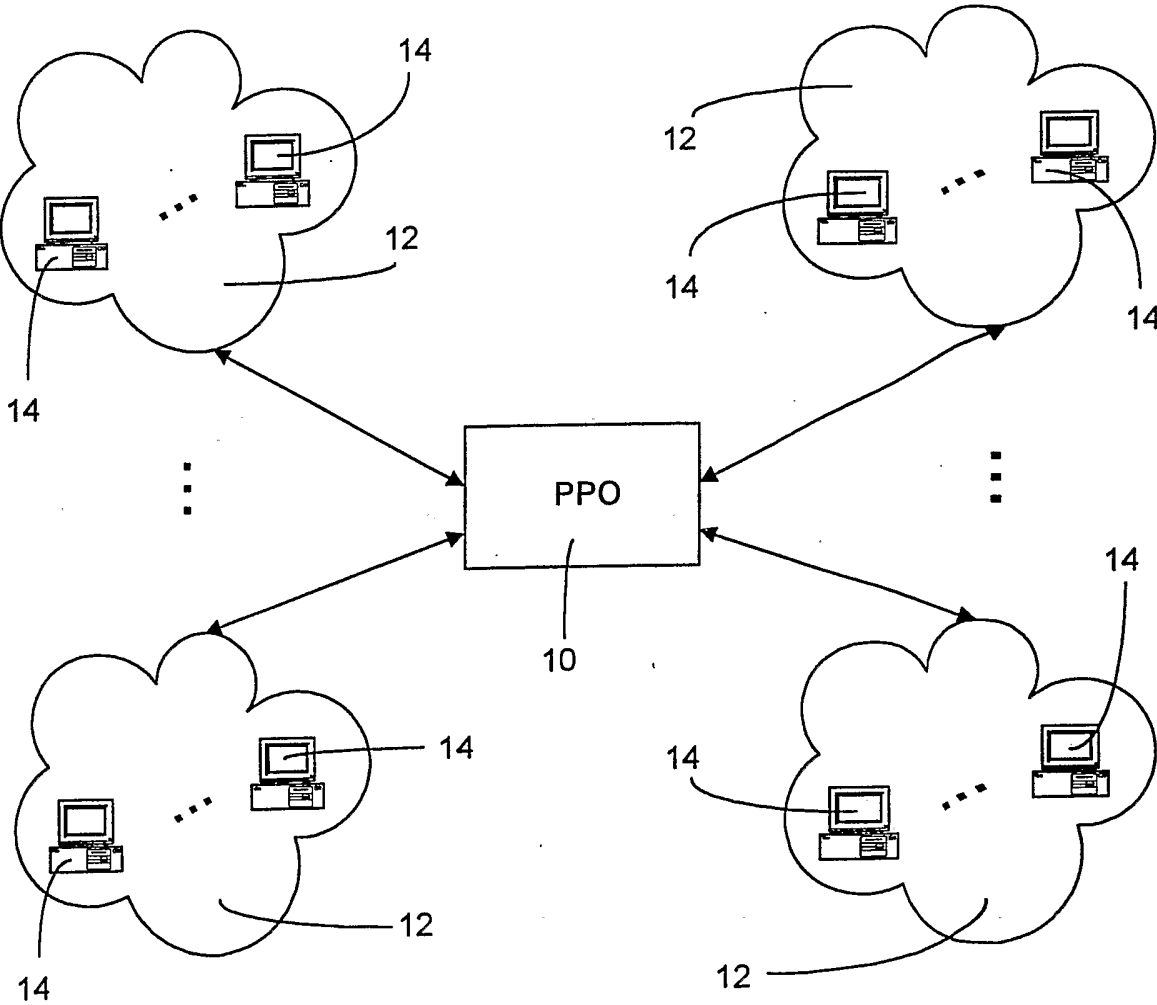
20 18. A system for routing queries in a peer to peer network, said system utilizing heuristics to determine the best host to forward a query to.

19. The system of claim 18 further comprising means for deleting expired queries.

25

20. The system of claim 18 further comprising means for re-issuing a query should said query not meet a threshold of responses.

FIG. 1



2/10

FIG. 2

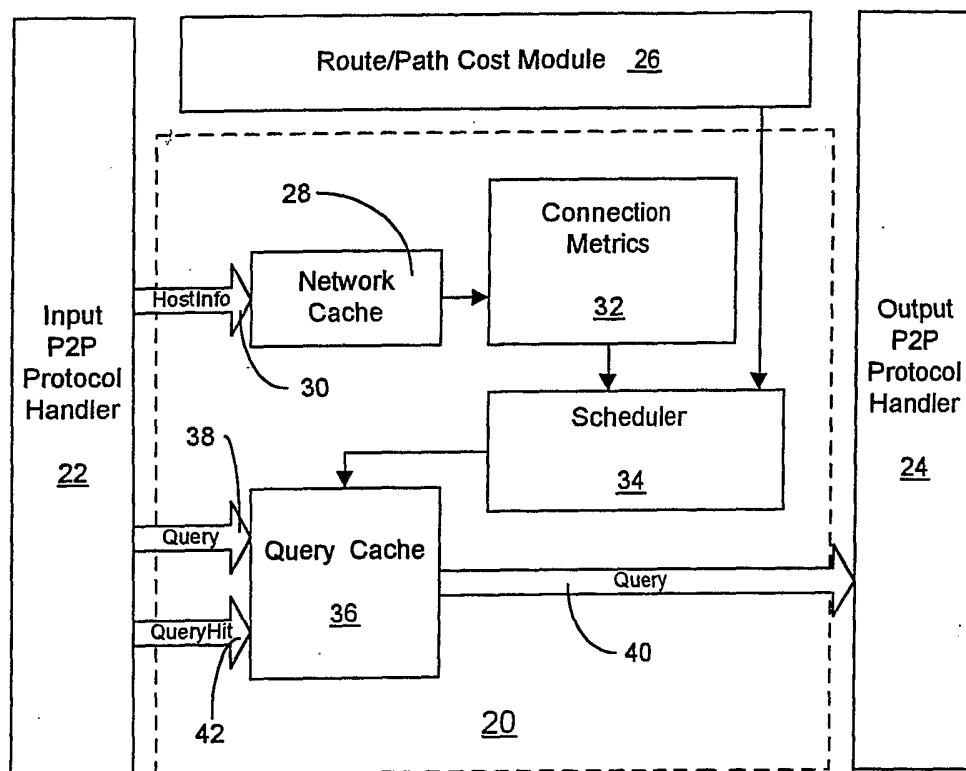
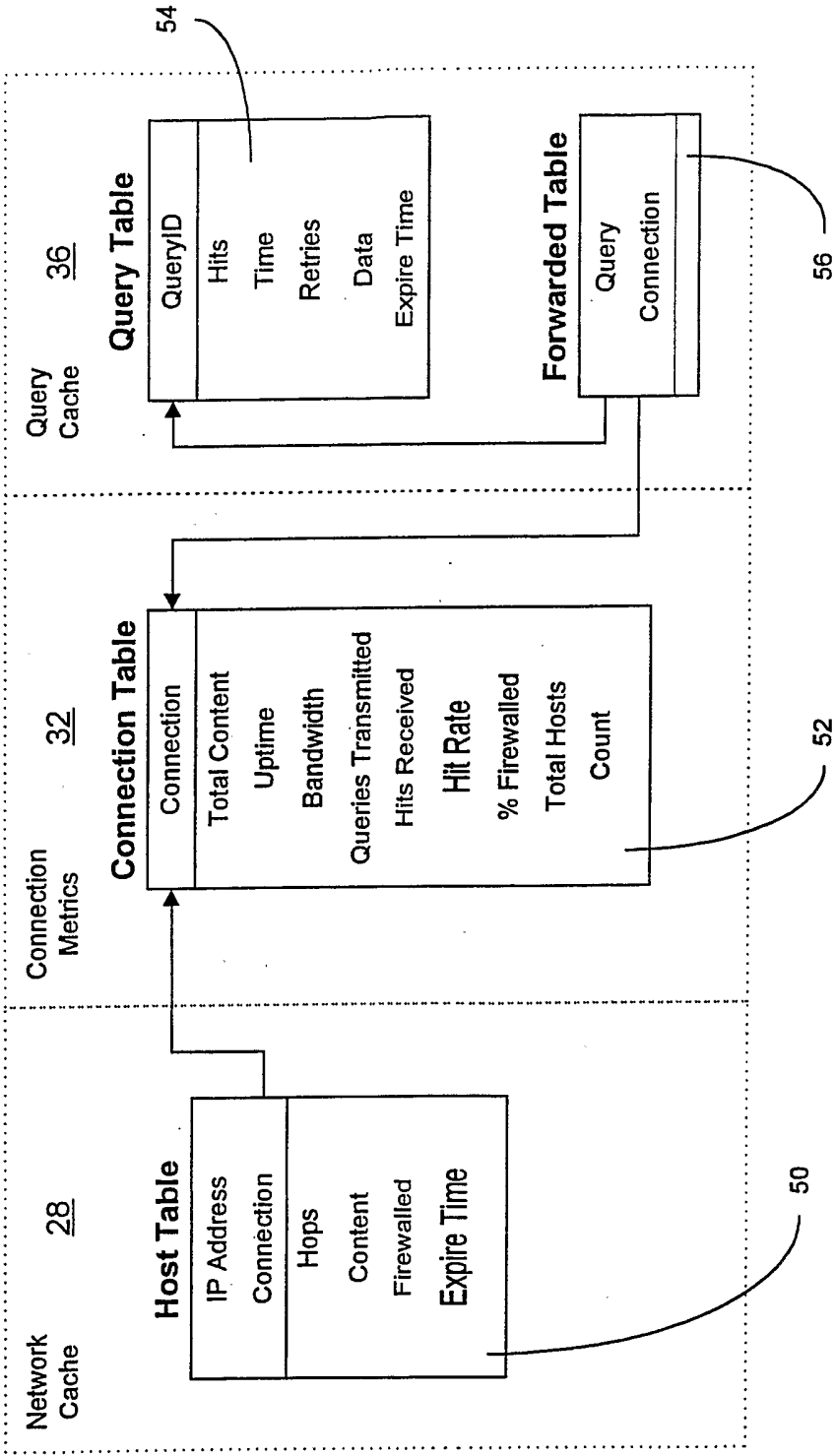
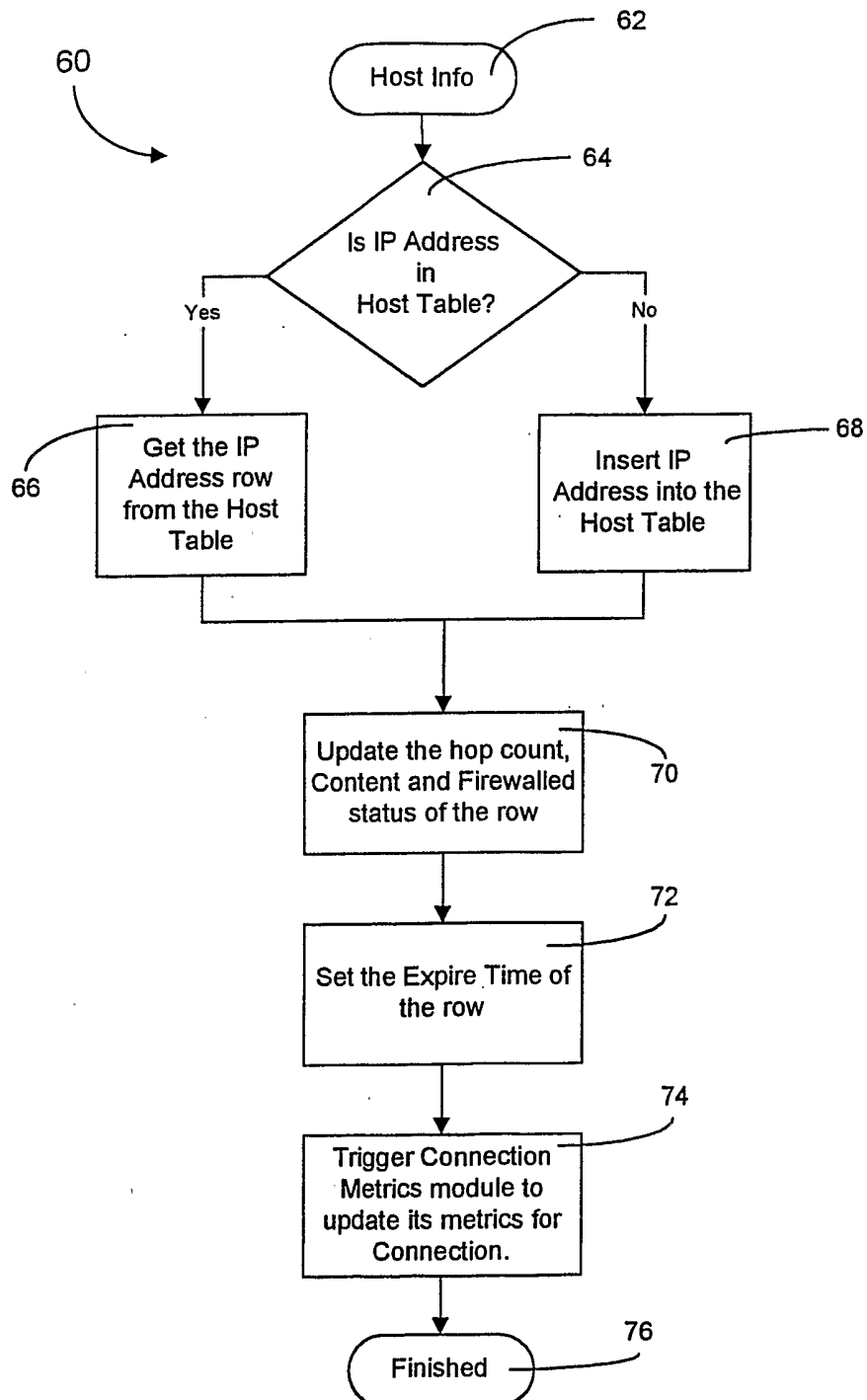


FIG. 3



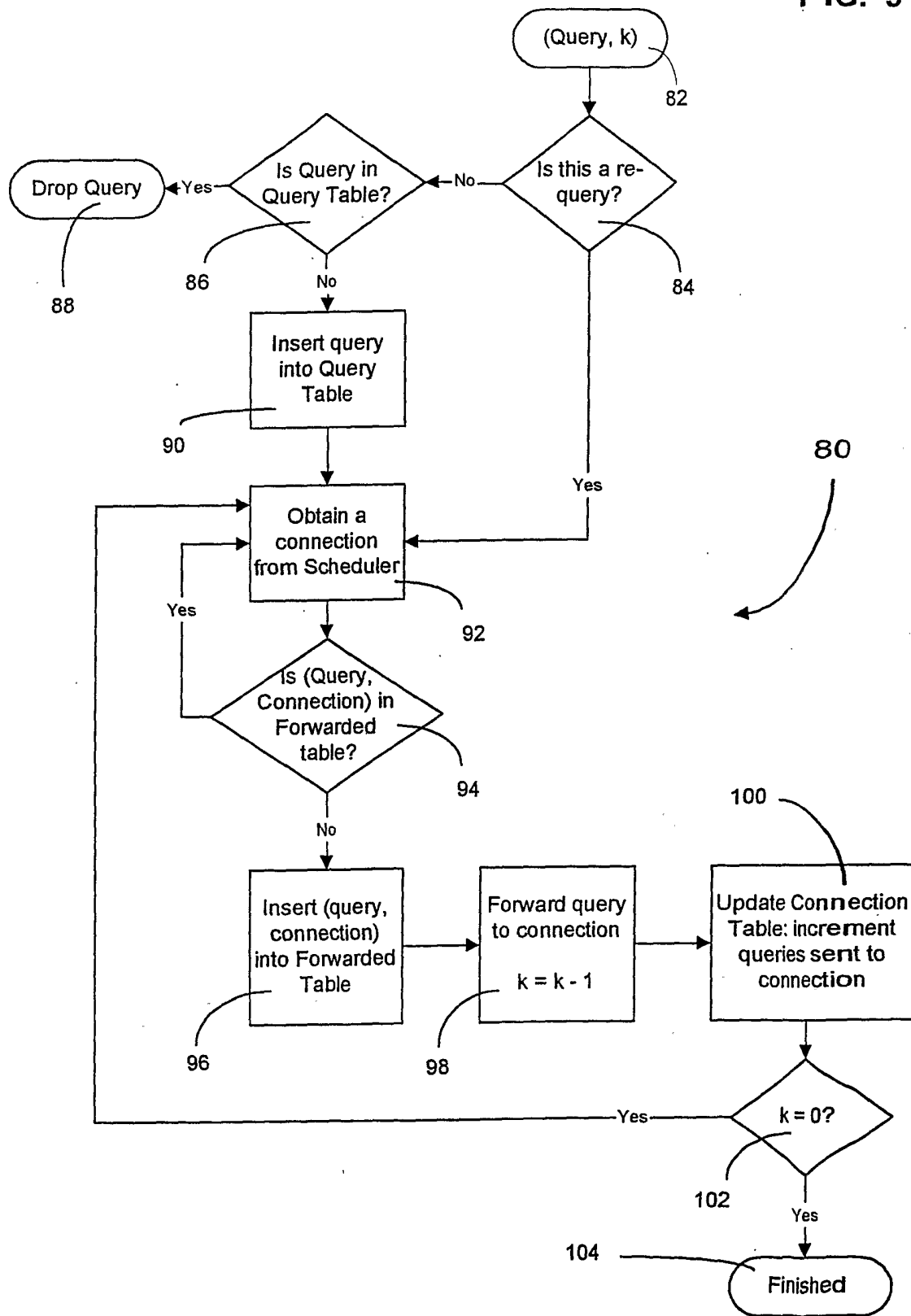
4/10

FIG. 4



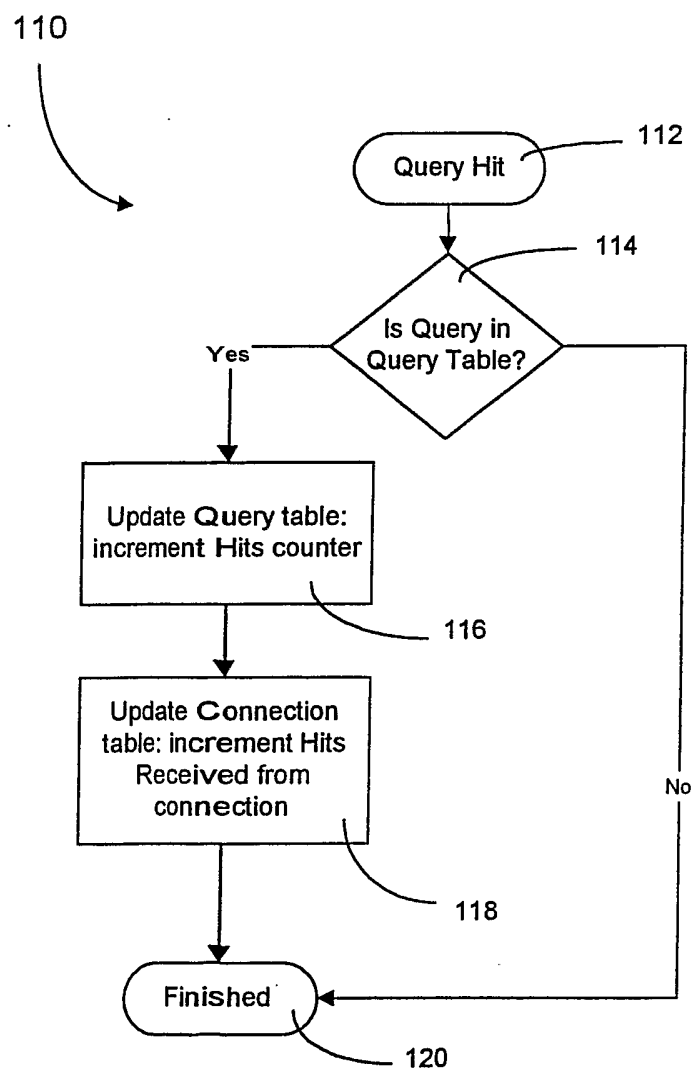
5/10

FIG. 5



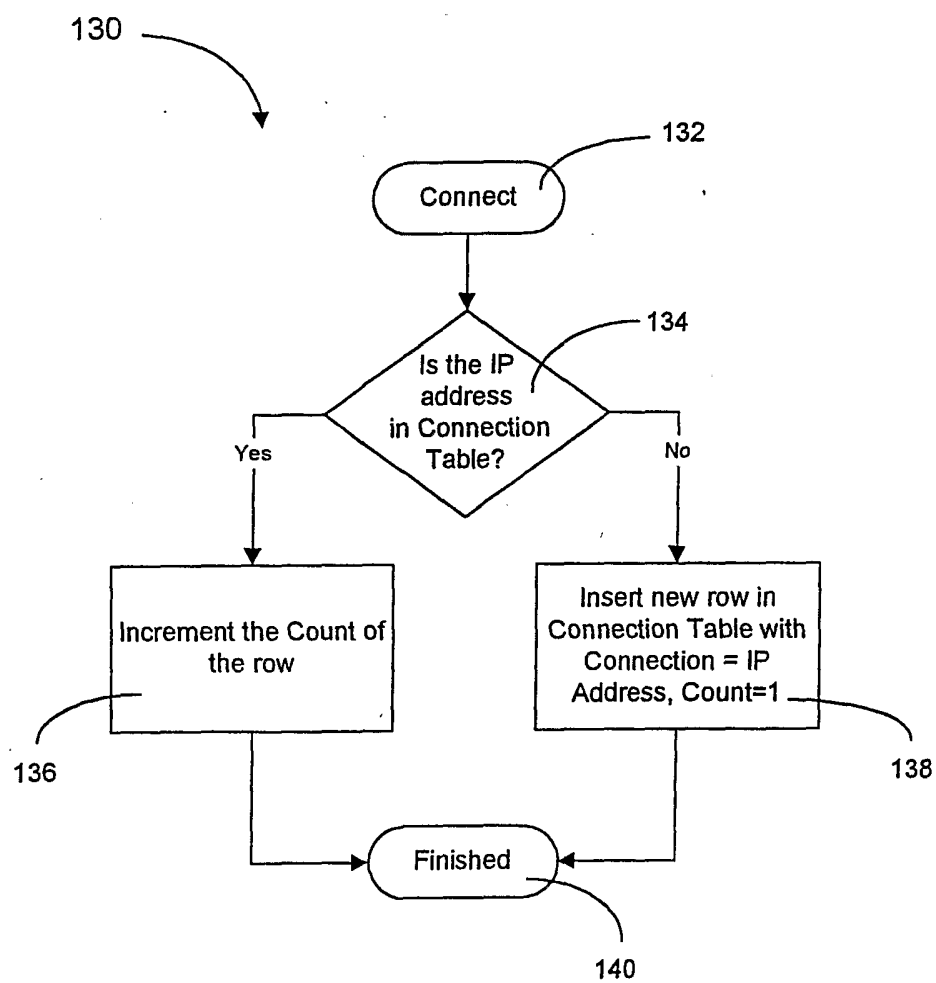
6/10

FIG. 6



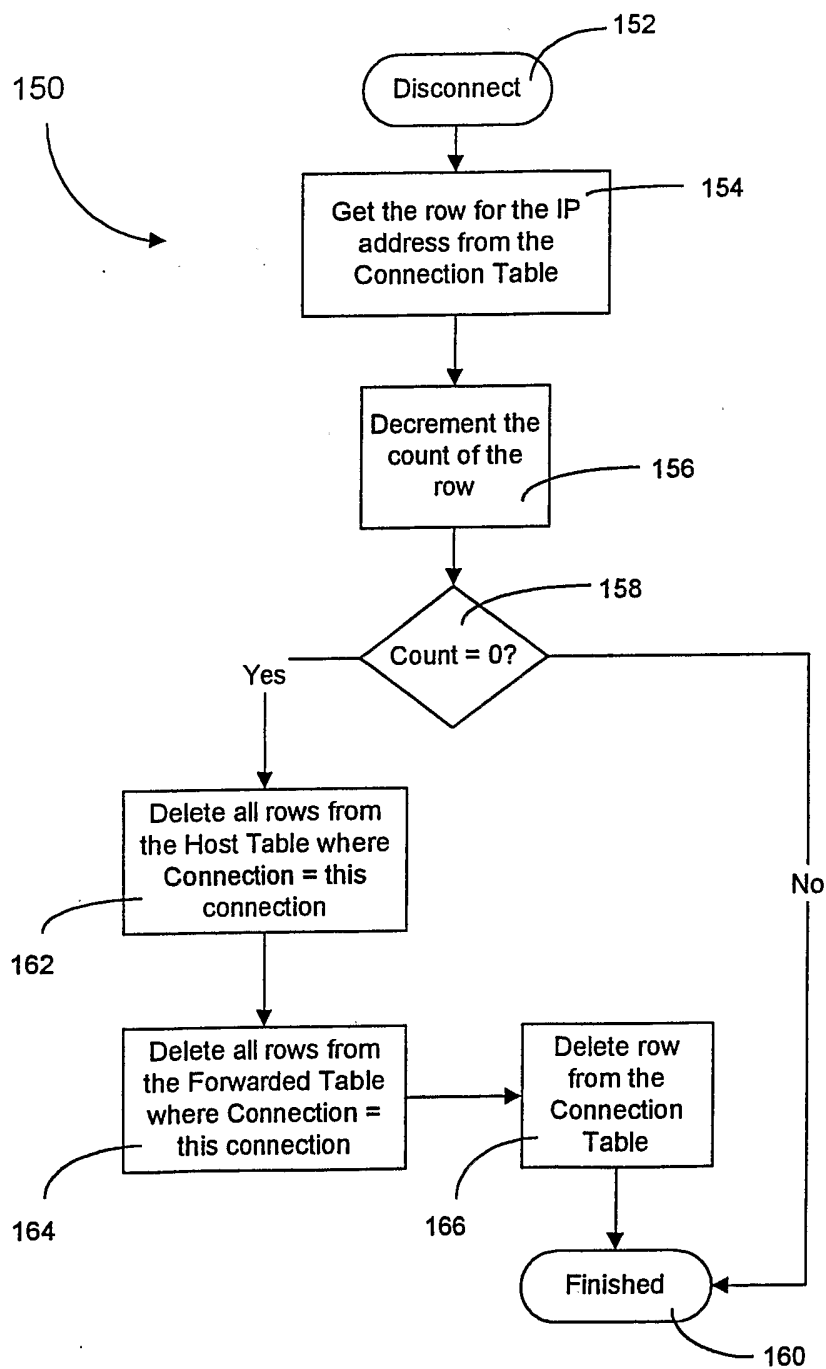
7/10

FIG. 7



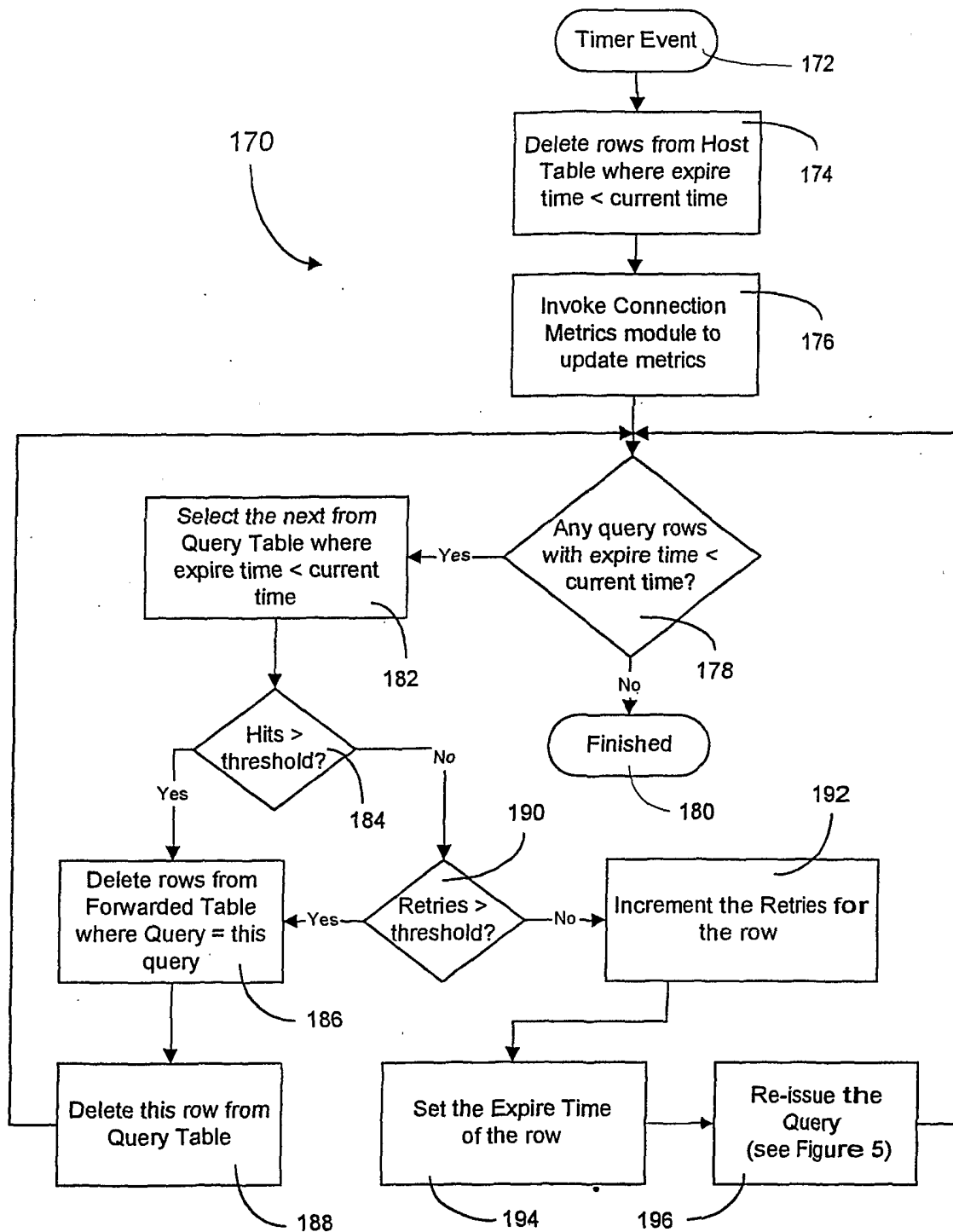
8/10

FIG. 8



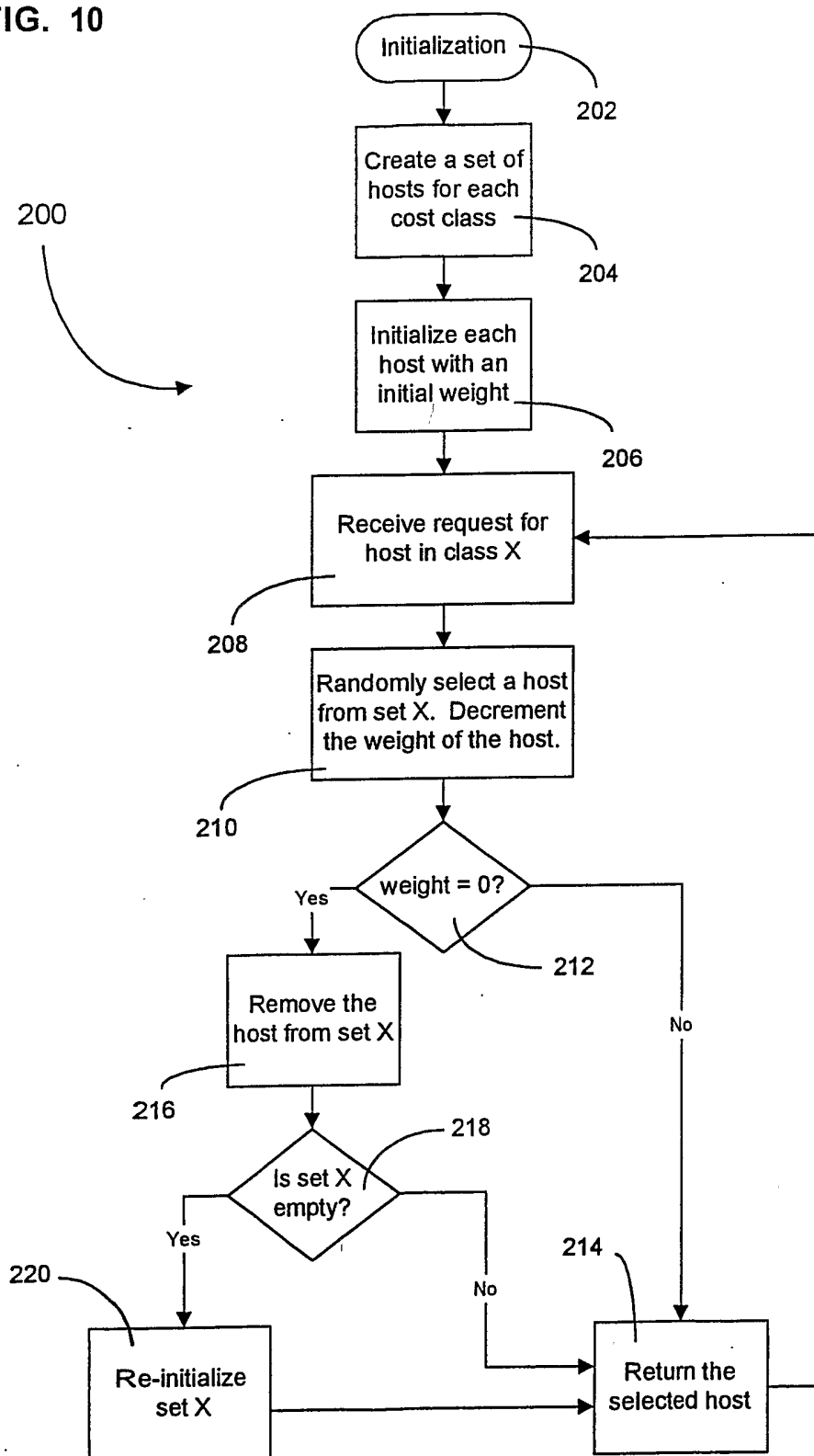
9/10

FIG. 9



10/10

FIG. 10



## INTERNATIONAL SEARCH REPORT

International Application No

PCT/CA 03/00799

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 H04L29/06 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 7 H04L 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)

EPO-Internal

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category * | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                                                                                                                  | Relevant to claim No. |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| X          | CRESPO, GARCIA-MOLINA: "Routing Indices for Peer-to-Peer Systems"<br>ICDCS 2002 22ND INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS,<br>2 July 2002 (2002-07-02), - 5 July 2002 (2002-07-05) XP002252824<br>Vienna, Austria                                              | 1-8,<br>18-20         |
| A          | abstract<br>page 3, left-hand column, paragraph 3.1.QUERY.PROCESSING.IN.A.DISTRIBUTED.SEAR<br>CH.P2P.S<br>page 4, left-hand column, paragraph 4.ROUTING.INDICES<br>page 7, left-hand column, paragraph 7.1.MODELING.SEARCH.MECHANISMS.IN.A.P2P.SY<br>STEM - page 8<br>-----<br>-/-- | 9,10                  |

☒ 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.

"&amp;" document member of the same patent family

Date of the actual completion of the international search

28 August 2003

Date of mailing of the international search report

16.01.04

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

Lefebvre, L

## INTERNATIONAL SEARCH REPORT

International Application No.

PCT/CA 03/00799

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

| Category ° | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                                                                                                                                      | Relevant to claim No.  |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| X          | TANG, XU, MAHALINGAM: "Peersearch: Efficient Information Retrieval in Peer-to-Peer Networks"<br>HEWLETT-PACKARD,<br>12 July 2002 (2002-07-12), XP002252825                                                                                                                                              | 5                      |
| A          | page 1, paragraph 1.INTRODUCTION<br>page 6, paragraph 5.RELATED.WORK<br>-----                                                                                                                                                                                                                           | 1-4,<br>6-10,<br>18-20 |
| A          | SAROIU, KRISHNA GUMMADI, GRIBBLE: "A measurement study of Peer to Peer file sharing systems"<br>PROCEEDINGS OF MULTIMEDIA COMPUTING AND NETWORKING 2002 MMCN'02,<br>January 2002 (2002-01), XP002252826<br>San Jose,CA,USA<br>paragraph<br>[4.RECOMENDATIONS.TO.PEER.RO.PEER.SYSTEM.D ESIGNES]<br>----- | 1-10,<br>18-20         |
| A          | SAM JOSEPH: "Neurogrid: Semantically Routing queries in Peer to Peer Networks"<br>PEER TO PEER COMPUTING AND NETWORKING 2002,<br>vol. 2376, no. 2002,<br>19 May 2002 (2002-05-19), - 24 May 2002 (2002-05-24) pages 202-214, XP002252827<br>Pisa, Italy<br>the whole document<br>-----                  | 1-10,<br>18-20         |

# INTERNATIONAL SEARCH REPORT

International application No.  
PCT/CA 03/00799

## Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:  
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:
3. ☐ Claims Nos.:  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

## Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-10, 18-20

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest.
- ☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-10,18-20

Routing of query messages of Peer to Peer networks using  
heuristics based criteria  
---

2. claims: 11-13

Managing queries stored in a device used in a Peer to Peer  
network  
---

3. claims: 14-17

Scheduler apparatus  
---