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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 December 2006 (07.12.2006)

PCT

(10) International Publication Number
WO 2006/130266 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/US2006/015332

(22) International Filing Date: 21 April 2006 (21.04.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/139,704 27 May 2005 (27.05.2005) US

(71) Applicant *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **ACHLIOPTAS, Demetrios**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **TRIBBLE, Eric, D.**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **PEARSON, Malcolm, E.**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **WARMAN, Leon**; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) Agents: **ALLEN, Michael, B.** et al.; c/o Sharon Rydberg, 21/2029, Microsoft Corporation, One Microsoft Way, Redmond, Washington 98052-6399 (US).

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

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

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

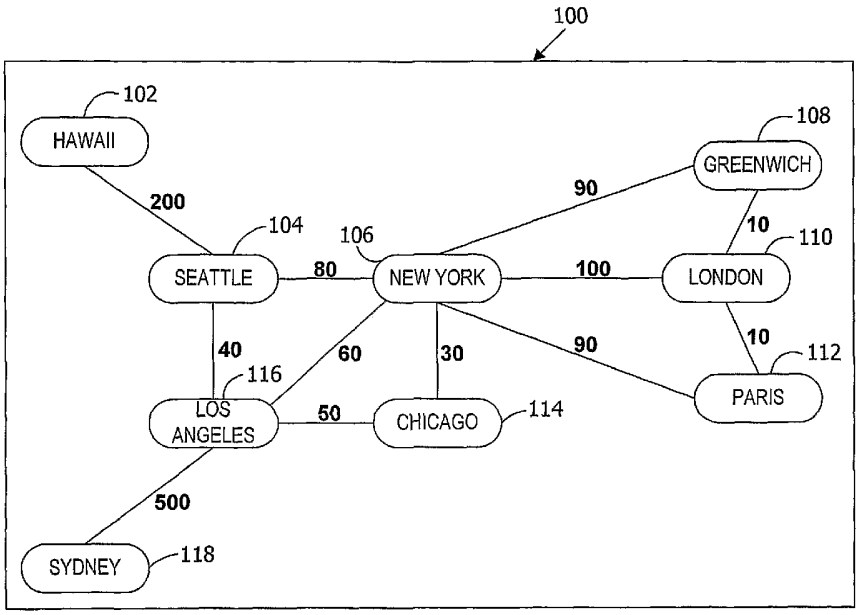
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR ROUTING MESSAGES WITHIN A MESSAGING SYSTEM



(57) Abstract: Systems and methods are described which provide enhanced stability, increased predictability, reduced transmission costs, and which conserve bandwidth in routing messages over computer networks. The systems and methods further include providing improved transmission of messages wherein the messages are transmitted to nodes closest to a target delivery node. If delivery is possible to a target node, the message transmission stops at the point of failure in the network, wherein delivery to the target node is accomplished at a later time or the message is returned to the sender.

WO 2006/130266 A2

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.

SYSTEM AND METHOD FOR ROUTING MESSAGES WITHIN A MESSAGING SYSTEM

TECHNICAL FIELD

[0001] Embodiments of the present invention relate to the field of routing messages over a network to one or more recipients. In particular, embodiments of this invention relate to systems and methods for reliably routing electronic mail over a network at minimal transfer costs; specifically when a single system to system transfer can pass responsibility for multiple recipients; more specifically in a system involving store-and-forward transfer of responsibility. Embodiments of the present invention also beneficially provide for systems and methods for reducing the costs in terms of minimizing network traffic of transferring mail addressed to multiple recipients that reside on different nodes while minimizing the amount of administrative network traffic; while preventing infinite loops due to inconsistent views of the network.

BACKGROUND OF THE INVENTION

[0002] Some prior systems and methods provide for routing mail over networks based upon network costs agreed upon by the different nodes that make up the network. However, these systems lack stability when the transfer costs vary or fluctuate, as the nodes must agree upon the transfer costs prior to the mail being transferred. These systems also pay an excessively high cost in terms of complexity or network traffic to resolve differences in the node's views of network topology. Thus, mail may be unnecessarily delayed until the costs are agreed upon.

[0003] Other systems and methods provide for routing to multiple mail recipients that calculate a routing path based upon shortest individual paths to the recipients. These methods, however, can unnecessarily use bandwidth by sending the same mail file over multiple links to multiple recipients.

[0004] Systems that rely on agreeing at the application level about network topology are unpredictable for administrators. The present system makes the system easier to diagnose because its behavior is predictable over a wider range of network conditions. The system and method of the present

invention ensure that if a message cannot reach its final destination, it will get as close as possible to the intended destination.

[0005] Accordingly, a system and method for reliably routing mail over a network to one or more recipients is desired to address one or more of these and other disadvantages.

SUMMARY OF THE INVENTION

[0006] The present invention overcomes the deficiencies of the known art by providing a system and method for routing a message over a network without requiring absolute agreement of transmission costs between a sending and a receiving node wherein the message is relayed to an available node that is the closest to the receiving node in terms of transmission costs. The system and method also optimize network bandwidth when routing a message over a network to multiple recipients residing on different nodes by delaying the forking of the message until the last node of divergence that is common to the network route to the recipients to the extent that the same view of topology is held by these nodes. The system and method also provide for improved reliability by routing mail to the closest node to the recipient's node based on availability schedules of the nodes. Additionally, the system and method of the present invention optimize the speed of routing a message to a recipient by determining the available node or nodes to which the message should be transmitted, and if a node is unavailable, selecting the next best node to which the message should be transmitted.

[0007] In accordance with the present invention, one aspect provides a computer-implemented method for transmitting a message over a computer network having a current node, a target node, and one or more intermediate nodes connected to the network between the current node and the target node. The method comprises creating a destination alternative table of the intermediate nodes located on the network having transmission costs to the target node that are less than the transmission costs of the current node to the target node, wherein the intermediate nodes are prioritized according to their minimum transmission costs. Transmission of the message is attempted to the target node. If the transmission of the message to the

target node fails, transmission of the message is attempted to at least one intermediate node as a function of its priority identified in the destination alternative table.

[0008] Another aspect of the present invention includes a message communication system for transmitting a message between the current server and the target server on a message communication network. The system comprises a current server, a target server, and one or more intermediate servers connected to the network. The system also includes a destination alternative table of the intermediate servers located on the network having transmission costs to the target server, wherein the intermediate servers are prioritized according to their minimum transmission costs. The current server being configured to execute computer-executable instructions for attempting transmission of the message to the target server. If the transmission of the message to the target server fails, the current server attempts transmission of the message to at least one intermediate server as a function of its priority identified in the destination alternative table.

[0009] Another aspect of the present invention includes one or more computer-readable media having computer-executable instructions for transmitting a message over a computer network. The network comprising a current node, a target node, and one or more intermediate nodes connected to the network between the current node and the target node. The computer-executable instructions comprise instructions for determining the transmission costs from the intermediate nodes to the target node and prioritizing the intermediate nodes according to their associated transmission costs. The instructions also comprise instructions for attempting transmission of the message to the target node; if the transmission of the message to the target node fails, attempting transmission of the message to at least one intermediate node as a function of its priority identified in the destination alternative table.

[0010] Alternatively, the invention may comprise various other methods and apparatuses.

[0011] Other features will be in part apparent and in part pointed out hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] FIG. 1 is an exemplary illustration of a network of nodes and links between the nodes in a message routing network.

[0013] FIG. 2 is a flowchart illustrating an exemplary process of one embodiment of the present invention wherein a message being conveyed over a network is transmitted to one or more recipients.

[0014] FIG. 3 is a block diagram illustrating one example of a suitable computing system environment in which the invention may be implemented.

[0015] Corresponding reference characters indicate corresponding parts throughout the drawings.

DETAILED DESCRIPTION OF THE INVENTION

[0016] Referring first to FIG. 1, an exemplary illustration is provided of a network of nodes and links between the nodes in a message routing network in which the present invention may be implemented. A computer network 100 comprises a number of nodes or message transfer agents (MTAs) that can send and receive messages, such as electronic message, electronic mail, a packet, and the like, over links that connect the nodes. Each node may contain recipients that receive the message or mail that is routed through the network from a sending node to the target node where the recipient is located. The bold numbers adjacent to the lines that connect the various nodes of network 100 represent hypothetical transmission costs associated with transmitting a message across the links. In this example, the transmission costs associated with sending a message from node 102 (Hawaii) to node 104 (Seattle) would be 200. Costs can also be considered as being inversely proportional to network availability and thus can be viewed as the opportunity cost of using a link. In this example, fixed costs are used for purposes of illustration, but in more advanced implementations, actual costs are updated dynamically as the bandwidth is allocated/consumed.

[0017] As an illustration of routing of a message over a network, a sender of a message located on node 102 (Hawaii) sends mail addressed to a first recipient on node 108 in Greenwich, UK and a second recipient on node 112 in Paris, France. If the message were to be individually sent as two

separate messages from Hawaii to Greenwich and from Hawaii to Paris, the total transmission costs for sending the files to the two recipients would be 740 (e.g., 200+80+90 (Hawaii to Greenwich) and 200+80+90 (Hawaii to Paris). However, if the message is sent as a single file through the node links that are common to both routes on the path to the different recipients, transmission costs would be reduced and network bandwidth would be conserved. Thus, if the message was sent as a single file transmission along the common pathway from node 102 (Hawaii) to node 104 (Seattle) and also from node 104 (Seattle) to node 106 (New York), and then bifurcated into two messages at node 106 (New York) that are sent to node 108 (Greenwich) and node 112 (Paris), the total cost for transmitting the mail would be 460 (e.g., 200+80 (Hawaii to New York) and 90+90 (New York to Greenwich and New York to Paris). Network bandwidth is preserved by avoiding application level transfers to intervening nodes when these nodes are not a point of divergence and when there is not a network failure. For example, the message is transmitted directly from Hawaii to New York. However, if Hawaii is unable to reach NY, an attempt is made to transmit the message from Hawaii to Seattle.

[0018] When a message is bifurcated at a node, the node is responsible for transmitting the message to the subsequent node along the pathway to the target node. Thus, for example, if the message is to be split at a node and sent to three different nodes, the current node maintains the responsibility for continuing the transmission of the message. Once the receiving node accepts the message, it assumes responsibility for transmitting the message to the next node along the pathway to the target node, and so forth, until the message is accepted by the target node.

[0019] In one embodiment of the system and method of the present invention, the routing of a message over a network is optimized by delaying the forking of the message until the last common node of the two recipients. This embodiment beneficially reduces transmission costs, conserves available network bandwidth, and also improves reliability and speed of the recipients receiving the message by relaying the message to a common node of divergence that is closest, in terms of transmission cost, to the target nodes of the recipients. Using the illustration described above, the system and method of the present invention would delay the forking of the message at node 106

(New York) and instead forward the file to the last common node of divergence of the two recipients, which, according to FIG. 1, is node 110 (London). At node 110 (London) the message is finally forked and transmitted to node 108 (Greenwich) and node 112 (Paris). Thus, the transmission costs for transmitting the message to the two recipients would be 400 (e.g., 200+80+100 (Hawaii to London) and 10+10 (London to Greenwich and London to Paris)).

[0020] As used herein, descriptions of proximity or distance to a node refer to transmission costs required to send a message from one node to a target node. Thus, a node that is "closest" to a target node would be the node that has the lowest transmission cost or other abstract cost, such as the opportunity cost outlined above, to send a file to the target node.

[0021] Transmitting a message across a network to a node that is the closest to the recipient is significant when a failure occurs on a network that can delay the message from being transmitted. By transmitting messages as close to the recipient as possible, progress of moving toward the target node is made in the event of a network failure. In such an event, the message can be quickly transmitted to the recipient when the network becomes available. This also handles the case where not all links are available at the same time. For example, an end-to-end connection from an initiating node to a target node may never actually be available. However, in utilizing the method and system of the present invention, progress can still be made and the message will eventually get through. Furthermore, network nodes often have better or more reliable cost and routing information regarding nodes that are closer in proximity to them than do remote nodes. Thus, if a failure were to occur near the target node of a recipient, a node that is in close proximity may be able to either identify an alternative path to the recipient that is both cost efficient and ensures that the message is received by the recipient rather than being returned to the sender as non-deliverable.

[0022] Several methods can be used to determine which node a message should be transmitted to in the event that the file can not be transmitted directly to the target node.

[0023] In one embodiment, a linear path of nodes is identified that begins with the current node, ends with the target node, and contains one or

more intermediate nodes connected to the network that are between the current node and the target node. If the target node is unavailable, the file is attempted to be transmitted to the intermediate node that is closest to the target node. If that node is unavailable, an attempt is made to transmit the message to the intermediate node that is second closest to the target node, and so forth, until there are no other intermediate nodes left that have not been tried or until a threshold of a predetermined number of attempts has been met. In such an event, the message may be delayed for later resending or returned to the sender as undeliverable.

[0024] In another embodiment, nodes are identified that are within a specified transmission cost range to the target node. The nodes are then organized in priority order of their transmission costs with the lowest transmission cost having the highest priority. Thus, if a transmission attempt to the target node fails, the next attempt is to the highest priority node, i.e., the node with the lowest transmission costs. If that transmission attempt fails, a third attempt is made to the next highest priority node (i.e., the node having the second-lowest transmission costs). This process is repeated until there are no other nodes left that fall within the specified cost range, or until a threshold of a predetermined number of attempts has been met. As described above, in such an event, the message may be delayed for later resending or returned to the sender as undeliverable. In another embodiment, a second minimal transmission cost range is identified that forms a concentric circle around the first cost range. Transmission attempts are made to nodes in the second range similar to described above for the minimum cost range. Further embodiments can include additional concentric circles of transmission cost ranges.

[0025] In another embodiment, a "halving retreat" process is followed wherein a linear line of intermediate nodes is identified between the current node and the target node. If a transmission attempt to the target node fails, an attempt is made to transmit the message to an intermediate node, for example, "Node D." If the attempted transmission to Node D fails, an attempt is made to transmit the message to the intermediate node that is located half-way between the current node and Node D. This process is continued until there are no other nodes left, or until a threshold of a predetermined number

of attempts has been met. As described above, in such an event, the message may be delayed for later resending or returned to the sender as undeliverable.

[0026] In another embodiment, an accelerated retreat process may be utilized in identifying potential alternative nodes to transmit messages to. If the target node is unavailable, the accelerated retreat process may identify the node that is, for example, four nodes from the target node. If attempts to send a file to that node fail, an attempt to send the file to the node that is, for example, twelve nodes away, and so forth until there are no other nodes left, or until a threshold of a predetermined number of attempts has been met. As described above, in such an event, the message may be delayed for later resending or returned to the sender as undeliverable.

[0027] In still another embodiment, an attempt is made to send a message as close to the target node as possible based upon the operating schedules of the individual nodes. This method may be combined with any of the patterns identified above related to a retreat strategy for transmitting a message if the message cannot be transmitted directly to a target node. For example, the path between the current node that the target node consists of intermediate nodes A, B, and C, wherein A is the furthest from the target node and node C is the closest to the target node. If it is currently 3:30 p.m. and node A accepts files from 1:00 p.m. to 4:00 p.m., node B accepts files from 3:00 p.m. to 6:00p.m., and node C accepts files from 6:00 p.m. to 10 p.m., an attempt will be first made to transmit the message to node B which is the closest available node to the target node based upon operating schedules.

[0028] Another aspect of the present invention is its stability to route a message regardless of fluctuating/globally weakly consistent/inconsistent transmission costs that may be identified by different nodes of the network. Transmission costs of routing a message between network nodes are often provided by third parties to operators of the nodes or MTAs. Often, however, two nodes may not be provided with identical transmission cost information, especially at a single point in time. Thus, a recipient node may determine that a different path to a target node is shorter, i.e., costs less, than the path determined by the sending node. For example, sending node A determines that the lowest transmission cost pathway to target node D is to transmit the

message from A to B to C and finally to D. Since transmission costs are sometimes in disagreement, node C may have cost information that the shortest path to D is to first send it back to B. Thus, a problem of cycling can occur when the message is looped back and forth between node B and node C.

[0029] Instead of delaying transmitting a message until the nodes agree upon transmission costs, the present invention monitors or tracks the number of nodes through which a file passes on its path to the target node. For example, in an email document, the nodes the file passes through on its path from the sender node to the target node are added to the SMTP header, or other mechanism for other systems. For example, X.400 utilizes the idea of an Envelope for this purpose. Thus, if the number of nodes through which a file passes on its path to a target node exceeds a minimum threshold, the node halts further transmittance for a period of time (a settling period) to permit the transmittance costs to settle across the network. After the settling period has passed, the message is attempted to be transmitted to the target node according to the shortest path (least transmission cost) and the number of nodes through which a file passes is continued to be monitored. If transmission costs have settled after the period of time has lapsed, the file is successfully sent to the target node where it is received by the addressee. However, if the transmission costs are still unsettled and cycling of the file continues, the cycling of the file is terminated when the number of nodes through which a file passes on its path to a target node meets a maximum threshold. If the maximum threshold is reached, the file is marked as not deliverable and/or a non-delivery report is appended to the file and it is returned to the sender.

[0030] In one embodiment, the minimum threshold is 10 or more nodes. In another embodiment, the minimum threshold is 10 to 15 nodes.

[0031] In one embodiment the settling period is up to one hour. In another embodiment, the settling period is up to 30 minutes. In still another embodiment, the settling period is up to 15 minutes.

[0032] In one embodiment, the maximum threshold is 20 or more nodes. In another embodiment, the maximum threshold is 25 to 35 nodes.

[0033] In one embodiment, the computer network of the present invention comprises nodes connected to the internet. In another embodiment, the computer network of the present invention is a corporate or university network crossing multiple countries. In another embodiment, the computer network of the present invention is a corporate or university network crossing multiple states. In another embodiment, the computer network of the present invention is a corporate or university network crossing multiple cities. In still another embodiment, the computer network of the present invention is a corporate or university network crossing multiple buildings of a single city or corporate or university campus.

[0034] By monitoring or tracking the number of nodes through which a file passes instead of delaying the sending of the message until costs are completely settled results in a more stable routing system and method that transmit messages to recipients, without delay, in the event minor transmittance cost disparities are present across the network. The system and method of the present invention therefore do not delay transmitting messages until all nodes agree on costs, but rather transmit the files to nodes that are closer to the target node. As previously discussed, nodes have better transmission cost and network status information for nodes that are in close proximity than more remote nodes. Thus, by transmitting messages to nodes that are closer to the target node, the files are not delayed, and the nodes that are closer to the target node are more likely to have more accurate cost and network status information that enable it to successfully transmit the file to the target node.

[0035] Referring next to FIG. 2, a flowchart illustrating an exemplary process of one embodiment of the present invention is provided wherein a message being conveyed over a network is transmitted to either one or more recipients. The flowchart describes the process carried out by a node (hereinafter referred to as the "current node"). At 202, the current node receives a message that is to be sent to one or more recipients residing at one or more target nodes other than the current node. The current node is either the original node to send a message or it is an intermediate node connected to the network between the originally sending node and the target node.

[0036] At 203, least cost paths and points of divergence of nodes on the network are computed. These methods are known in the art and are not limited to any particular computation. Ongoing monitoring of changes in system configuration is conducted and the least cost paths and points of divergence are recomputed as the system configuration changes.

[0037] At 204, a determination is made whether the message has been transferred to MIN number of nodes, wherein MIN is a predetermined minimum threshold number. If the number of nodes the file has been transmitted to exceeds the minimum threshold number, the file is set aside at 206 for a predetermined Settling Period to permit the transmission costs to settle.

[0038] At 208, a determination is made whether the message has been transferred to MAX number of nodes, wherein MAX is a predetermined maximum threshold number. If the number of nodes the file has been transmitted to equals the maximum threshold number, the file is returned to the sending party and marked as non-deliverable at 210. In one embodiment, a non-delivery report is provided to the sender that further describes the delivery failure for the file.

[0039] At 212, the current node builds a Destination Table that identifies the transmission costs associated with sending message to the nodes that are present in the network. The Destination Alternative Table further prioritizes the nodes on the network according to the minimum transmission costs associated with transmitting messages to them from the current node. A determination is made at 214 whether the message is to be sent to one or more recipients residing on a single node, or whether the file is to be sent to two or more recipients residing on two or more different nodes.

[0040] If it is determined that the message is to be sent to one or more recipients residing on a single node, the message is added to the queue at 216 for transmission to a destination node.

[0041] A connection manager algorithm determines that the mail should be sent to a destination node at 218. The connection manager algorithm first attempts to transmit the message directly to the target node that is identified on the message at 220. If the file cannot be directly sent to the target node, either because there is not a direct connection, or because the

target node cannot accept the file for some reason (e.g., component failure, bandwidth limitations, network failure, configuration errors, lower level system policy and the like), the file is transmitted to the highest priority node identified in the Destination Alternative Table that has not yet been tried by the current node. At 222, the current node determines if the file is successfully transmitted to the destination node within X tries, wherein X is a predetermined number for transmission attempts by the current node. Alternatively, or in addition to a predetermined number of transmission attempts, X may include a predetermined time limit within which the message must be sent. If the file is successfully transmitted to the destination node, the current node's responsibilities have been fulfilled and the process ends.

[0042] If the current node determines at 222 that the file was not successfully transmitted to the destination node within X tries (or a time limit for X has been exceeded), the current node determines if attempts have been made to transmit the mail to all the destination nodes identified in the Destinations Alternative Table at 224. If transmissions have not been attempted to all of the identified destination nodes, the current node attempts to send the message to the highest priority node at 220 that has not yet been attempted. If delivery has been attempted to all of the identified destinations nodes, the current node determines if delivery to all nodes has been attempted Y times at 226, wherein Y is a predetermined number of transmission attempts that are made by the current node. Alternatively, or in addition to a predetermined number of transmission attempts, the Y may include a predetermined time limit within which the message must be sent. If the current node has not attempted delivery Y times (or a time limit for Y has been exceeded), the message is set aside for a specified time period at 228 to permit the network to be repaired. After the specified time period has elapsed, the connection manager again determines that the mail should be sent to a destination node at 218.

[0043] If the current node has attempted delivery Y times or a predetermined time limit within which the current node must deliver the message has been exceeded, the current node determines whether the message is addressed to multiple destinations with a common node of

divergence at 230. If so, the file is requeued for the individual destinations rather than the node of divergence at 232 and added to the queue at 216.

[0044] If the current node determines at 230 that the message is addressed to one or more recipients residing at a single target node, the file is returned to the sending party and marked as non-deliverable at 210. In one embodiment, a non-delivery report is provided to the sender that further describes the delivery failure for the file.

[0045] If the current node determines at 214 that the message is addressed to multiple recipients residing at two or more different nodes, a Minimal Spanning Tree is created at 234 wherein the current node is the root. In one embodiment, the Minimal Spanning Tree is computed in advance and used for transmitting multiple different messages. A Spanning Tree Local Partition Table is created at 236. It is a Minimal Spanning Tree which encodes the minimize cost spanning tree for the network. The Minimal Spanning Tree is used to find the points of divergence. Utilizing the Minimal Spanning Tee and the Spanning Tree Local Partition Table, the current node identifies the last node of divergence for the recipients at 238. If at 240 it is determined that the current node is the last node of divergence for the recipients, at 242 the message is forked to the target nodes of the recipients and the message is placed on the queue at 216.

[0046] If the current node is not a node of divergence, the current node groups the recipient destination nodes at 244 according to the last node of divergence for each group and places the grouped destinations on the queue at 216.

[0047] In one embodiment of the present invention, as described in FIG. 2, the last node of divergence may not be available or cannot accept messages that are attempted to be sent to it. In such a case, the present invention forks the message at the current node in order to ensure that the message is successfully transmitted to the recipient at the expense of additional transmission costs and bandwidth. Thus, while the first attempt(s) is to transmit a message in a manner that delays forking a file until the last node of divergence, if delaying forking would either cause the file to be delayed in delivery or undeliverable, the present system and method instead fork the message earlier, thereby transmitting the file to the recipient. Thus,

the file is forked earlier as it is typically believed that it would be more valuable for the recipient to receive the message than to save the additional transmission costs and bandwidth associated with delaying the file and forking the file at the last node of divergence. An example of this is illustrated in FIG. 1 where the current node is node 106 (New York) which is attempting to forward the message to node 110 (London) which is the last node of divergence to recipients residing at node 108 (Greenwich) and node 112 (Paris). If node 106 (New York) cannot forward the message to node 110 (London), it forks the message recipients at 230 and places two messages on the queue wherein one message is transmitted to node 108 (Greenwich) and the other message is transmitted to node 112 (Paris). Thus, the transmission costs increase from 120 to 180, but instead of delaying the message or returning the file to the sender as non-deliverable, the file is sent directly to the recipients nodes of node 108 (Greenwich) and node 112 (Paris).

[0048] In another embodiment, the system and method maintain a cache of all recently failed connections. In another embodiment, the system and method maintain a cache of all recently failed connections grouped by the adjacent node that they are reached through. These embodiments can be utilized to avoid attempting to transmit message to recently failed nodes and thereby avoid delaying the transmittance of a file. In one embodiment, a cache is maintained for failed connections occurring within a predetermined time period. In one embodiment, the time period is the most recent six hours. In another embodiment, a cache is maintained for failed connections occurring within the most recent hour. In another embodiment, a cache is maintained for failed connections occurring within the most recent thirty minutes.

[0049] In one embodiment, attempts are made to simultaneously transmit messages to multiple alternative nodes instead of opening connections to alternatives one at a time. In this embodiment, the node that is closest to the ultimate target and succeeds within a predetermined time, e.g., one second, is selected. If none succeed in the predetermined time period, the node that first successfully accepts the message is used. If none succeed, then a longer time period can be used.

[0050] In one embodiment, if the current connection is "slow," i.e., the bytes/second is significantly lower than average, then an attempt to open a connection to a "closer" node is made to transmit a second message. If the "closer" node is at or above "average," sending the message to the "slow" node should be discontinued in favor of the "closer" node.

[0051] FIG. 3 shows one example of a general purpose computing device in the form of a computer 130. In one embodiment of the invention, a computer such as the computer 130 is suitable for use in the other figures illustrated and described herein. Computer 130 has one or more processors or processing units 132 and a system memory 134. In the illustrated embodiment, a system bus 136 couples various system components including the system memory 134 to the processors 132. The bus 136 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

[0052] The computer 130 typically has at least some form of computer readable media. Computer readable media, which include both volatile and nonvolatile media, removable and non-removable media, may be any available medium that may be accessed by computer 130. By way of example and not limitation, computer readable media comprise computer storage media and communication media. Computer storage media include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, message structures, program modules or other message. For example, computer storage media include RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium that may be used to store the desired information and that may be accessed by computer 130. Communication media typically embody

computer readable instructions, message structures, program modules, or other message in a modulated message signal such as a carrier wave or other transport mechanism and include any information delivery media. Those skilled in the art are familiar with the modulated message signal, which has one or more of its characteristics set or changed in such a manner as to encode information in the signal. Wired media, such as a wired network or direct-wired connection, and wireless media, such as acoustic, RF, infrared, and other wireless media, are examples of communication media. Combinations of any of the above are also included within the scope of computer readable media.

[0053] The system memory 134 includes computer storage media in the form of removable and/or non-removable, volatile and/or nonvolatile memory. In the illustrated embodiment, system memory 134 includes read only memory (ROM) 138 and random access memory (RAM) 140. A basic input/output system 142 (BIOS), containing the basic routines that help to transfer information between elements within computer 130, such as during start-up, is typically stored in ROM 138. RAM 140 typically contains message and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 132. By way of example, and not limitation, FIG. 3 illustrates operating system 144, application programs 146, other program modules 148, and program message 150.

[0054] The computer 130 may also include other removable/non-removable, volatile/nonvolatile computer storage media. For example, FIG. 3 illustrates a hard disk drive 154 that reads from or writes to non-removable, nonvolatile magnetic media. FIG. 3 also shows a magnetic disk drive 156 that reads from or writes to a removable, nonvolatile magnetic disk 158, and an optical disk drive 160 that reads from or writes to a removable, nonvolatile optical disk 162 such as a CD-ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that may be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 154, and magnetic disk drive 156 and optical disk drive 160

are typically connected to the system bus 136 by a non-volatile memory interface, such as interface 166.

[0055] The drives or other mass storage devices and their associated computer storage media discussed above and illustrated in FIG. 3, provide storage of computer readable instructions, message structures, program modules and other message for the computer 130. In FIG. 3, for example, hard disk drive 154 is illustrated as storing operating system 170, application programs 172, other program modules 174, and program message 176. Note that these components may either be the same as or different from operating system 144, application programs 146, other program modules 148, and program message 150. Operating system 170, application programs 172, other program modules 174, and program message 176 are given different numbers here to illustrate that, at a minimum, they are different copies.

[0056] A user may enter commands and information into computer 130 through input devices or user interface selection devices such as a keyboard 180 and a pointing device 182 (e.g., a mouse, trackball, pen, or touch pad). Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are connected to processing unit 132 through a user input interface 184 that is coupled to system bus 136, but may be connected by other interface and bus structures, such as a parallel port, game port, or a Universal Serial Bus (USB). A monitor 188 or other type of display device is also connected to system bus 136 via an interface, such as a video interface 190. In addition to the monitor 188, computers often include other peripheral output devices (not shown) such as a printer and speakers, which may be connected through an output peripheral interface (not shown).

[0057] The computer 130 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 194. The remote computer 194 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 130. The logical connections depicted in FIG. 3 include a local area network (LAN) 196 and a wide area network (WAN) 198, but may also include

other networks. LAN 136 and/or WAN 138 may be a wired network, a wireless network, a combination thereof, and so on. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and global computer networks (e.g., the Internet).

[0058] When used in a local area networking environment, computer 130 is connected to the LAN 196 through a network interface or adapter 186. When used in a wide area networking environment, computer 130 typically includes a modem 178 or other means for establishing communications over the WAN 198, such as the Internet. The modem 178, which may be internal or external, is connected to system bus 136 via the user input interface 184, or other appropriate mechanism. In a networked environment, program modules depicted relative to computer 130, or portions thereof, may be stored in a remote memory storage device (not shown). By way of example, and not limitation, FIG. 3 illustrates remote application programs 192 as residing on the memory device. The network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0059] Generally, the message processors of computer 130 are programmed by means of instructions stored at different times in the various computer-readable storage media of the computer. Programs and operating systems are typically distributed, for example, on floppy disks or CD-ROMs. From there, they are installed or loaded into the secondary memory of a computer. At execution, they are loaded at least partially into the computer's primary electronic memory. The invention described herein includes these and other various types of computer-readable storage media when such media contain instructions or programs for implementing the steps described below in conjunction with a microprocessor or other message processor. The invention also includes the computer itself when programmed according to the methods and techniques described herein.

[0060] For purposes of illustration, programs and other executable program components, such as the operating system, are illustrated herein as discrete blocks. It is recognized, however, that such programs and components reside at various times in different storage components of the computer, and are executed by the message processor(s) of the computer.

[0061] Although described in connection with an exemplary computing system environment, including computer 130, the invention is operational with numerous other general purpose or special purpose computing system environments or configurations. The computing system environment is not intended to suggest any limitation as to the scope of use or functionality of the invention. Moreover, the computing system environment should not be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, mobile telephones, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

[0062] The invention may be described in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices. Generally, program modules include, but are not limited to, routines, programs, objects, components, and message structures that perform particular tasks or implement particular abstract message types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

[0063] An interface in the context of a software architecture includes a software module, component, code portion, or other sequence of computer-executable instructions. The interface includes, for example, a first module accessing a second module to perform computing tasks on behalf of the first module. The first and second modules include, in one example, application programming interfaces (APIs) such as provided by operating systems, component object model (COM) interfaces (e.g., for peer-to-peer application

communication), and extensible markup language metamessage interchange format (XMI) interfaces (e.g., for communication between web services).

[0064] The interface may be a tightly coupled, synchronous implementation such as in Java 2 Platform Enterprise Edition (J2EE), COM, or distributed COM (DCOM) examples. Alternatively or in addition, the interface may be a loosely coupled, asynchronous implementation such as in a web service (e.g., using the simple object access protocol). In general, the interface includes any combination of the following characteristics: tightly coupled, loosely coupled, synchronous, and asynchronous. Further, the interface may conform to a standard protocol, a proprietary protocol, or any combination of standard and proprietary protocols.

[0065] The interfaces described herein may all be part of a single interface or may be implemented as separate interfaces or any combination therein. The interfaces may execute locally or remotely to provide functionality. Further, the interfaces may include additional or less functionality than illustrated or described herein.

[0066] The following examples further illustrate the invention.

Example – Routing of Electronic Mail Over a Network

Initialization

[0067] During initialization, there are two message structures to be computed. These are used throughout the execution of the process. If the underlying message changes, these message structures can simply be re-computed and used from that point forward.

Destination Alternative Table

[0068] For every node within the organization, delivery nodes that are closer to the current machine are determined and placed on a list or table (the destination alternative table). The list is ordered according to their minimal transmittal cost to the target in question. This includes all the nodes on the minimal cost path between the current node and the target node. This can also include other nodes, on other paths from the source to the target as long as their minimal cost to the target is less than the transmittal cost from

the current node to the destination node. Finally, this can include nodes that are on a path to the target, but not on a path from the current node to the target. Here, distances are according to the link costs defined between nodes, accounting for link bridges.

[0069] It is a heuristic decision of how inclusive the algorithm is to be. Including a larger set will allow the system to try as many alterative steps as possible to reach the target; whereby increasing the chance of success, but reducing the overall efficiency in the face of link failures. Being more inclusive allows us to account for the likelihood that the underlying node link records are just a model of the underlying topology and do not represent every aspect of the physical network topology.

Minimal spanning tree

[0070] A minimal spanning tree is computed for the network nodes/links with the current node as the root. Additionally, a spanning tree local partition table is built. For each destination D, the next hop is determined (first adjacent node N in the path in the minimal spanning tree from the current node to the target D).

Handling inconsistent global message

[0071] The algorithm assumes that network node/link message is globally consistent. In practice, this will frequently be true because the message is replicated aggressively and because it changes slowly, only under operator control. However, it is not guaranteed to be true due to replication latency.

[0072] The system will track the number of times a message has been transferred from node to node, using, for example, the SMTP Received: header. If the transfer count exceeds a first threshold, the mail will be delayed, giving replication a chance to settle. If the transfer count exceeds a maximum threshold, the system is presumed to be broken and the mail is returned to the sender with information about the failed delivery, such as a non-delivery report (NDR).

[0073] A benefit of the present algorithm is its need for minimal information about nodes that are further away from it. As a result, it is less sensitive to the inconsistencies in the global node/link message.

Single Destination Case

[0074] In the Single Destination Case, mail is routed to the address of an individual recipient. There are no concerns about fan-out issues of recipients at multiple nodes because there is only one recipient. This may have occurred because the mail was originally addressed to an individual recipient, or may have occurred because the mail was originally addressed to multiple recipients, and the mail has fanned out.

[0075] At the time of routing, the message is placed on a queue for delivery to the destination node.

[0076] At the time of relay (when the system wishes to move mail from the local machine to a remote machine), a connection manager algorithm determines that the mail should be relayed to destination T.

[0077] The mail is first attempted to be transmitted to the destination of the node of the recipient. If the transmission to the destination node fails, additional attempts are made to transmit the mail as a function of the minimal transfer cost to every alternative intermediate node in the destination alternative table. These additional transfer attempts are performed in priority order of the minimal transfer cost between the intermediate node and the destination node. This has the effect of moving the mail to the node closest to the destination if the destination itself cannot be reached.

[0078] If a destination cannot be reached, TCP will determine that the connection cannot be made within a period of time. For example, if the connection cannot be made for 20 seconds, each site has on average 2 bridgeheads, and the diameter of the network is up to 10, then a latency of 400 seconds, or 7 minutes would be required to attempt all of the alternative sites. This can be further optimized by keeping a cache of connections that were recently attempted and failed (e.g., failures in the last 30 minutes), and not attempting to connect to those sites.

[0079] If no connection can be made, in the Single Destination Case, the queue of messages is put aside and a later attempt to transmit the

queue is performed in anticipation that the network will have been repaired at that later time.

The Multiple Destination Case

[0080] In some cases, mail is addressed to more than one recipient residing on different nodes. In this case, the network traffic optimal approach is to relay the mail across the links in the minimal spanning tree computed when the system started.

[0081] The mail recipients are grouped according to the last node where the path to the recipient's node first diverges. This node is the diverging node that has the highest priority on the destination alternative table (e.g., the minimal cost between the diverging node and the destination nodes). A fork of the message is created at the diverging node and the mail is placed on the queue for the diverging node for the recipients that take that path. The message must be forked the message when two recipients have paths that are only concurrent at the current node. It is advantageous to delay forking the message as much as possible, so given three recipients, the mail is transmitted to a closer "next hop" to avoid an additional fork.

[0082] The node can be efficiently computed using the spanning tree local partition table described in the initialization section. Using the partition table, the recipients of the mail are grouped according to the next hop in the minimal spanning tree to the destination for the recipient. A fork of the message is created for each of the groups of the message; and the mail is queued to the node that appears on the path within the minimal spanning tree from the current node to the target node, that is furthest from the current node (i.e., the node having the minimal cost to the destination node(s)).

[0083] Once the mail has been queued, transmission proceeds as described in the Single Destination Case above.

[0084] If a connection cannot be made to any of the diverging node alternatives for more than one target, a forking of the message is performed at the current node and the mail is queued for the new targets that appear as a result of the forking. In this way, if a node is not available the message is forked at the current node to allow the mail a greater chance of delivery. This

does reduce efficiency where a node is not available, but can permit some or all of the mail to be successfully transmitted.

[0085] The order of execution or performance of the methods illustrated and described herein is not essential, unless otherwise specified. That is, elements of the methods may be performed in any order, unless otherwise specified, and that the methods may include more or less elements than those disclosed herein. For example, it is contemplated that executing or performing a particular element before, contemporaneously with, or after another element is within the scope of the invention.

[0086] When introducing elements of the present invention or the embodiment(s) thereof, the articles "a," "an," "the," and "said" are intended to mean that there are one or more of the elements. The terms "comprising," "including," and "having" are intended to be inclusive and mean that there may be additional elements other than the listed elements.

[0087] In view of the above, it will be seen that the several objects of the invention are achieved and other advantageous results attained.

[0088] As various changes could be made in the above constructions, products, and methods without departing from the scope of the invention, it is intended that all matter contained in the above description and shown in the accompanying drawings shall be interpreted as illustrative and not in a limiting sense.

WHAT IS CLAIMED IS:

1. A computer-implemented method for transmitting a message over a computer network, said network having a current node, a target node, and one or more intermediate nodes connected to the network between the current node and the target node, the method comprising:

creating a destination alternative table of the intermediate nodes located on the network having transmission costs to the target node that are less than the transmission costs of the current node to the target node, wherein the intermediate nodes are prioritized according to their minimum transmission costs;

attempting transmission of the message to the target node; and

if the transmission of the message to the target node fails, attempting transmission of the message to at least one intermediate node as a function of its priority identified in the destination alternative table.

2. The method of claim 1, wherein

if the attempt to transmit the message to the target node fails, attempting transmission of the message to the intermediate node having the highest priority; and

if the attempt to transmit the message to the highest priority intermediate node fails, attempting transmission of the message to intermediate nodes of successively decreasing priority.

3. The method of claim 1, wherein the message is transmitted over a computer network from the current node to at least two target nodes, the method further comprising:

creating a minimum spanning tree for the network with the current node as a root;

determining a last node of divergence from the minimum spanning tree of at least two recipients residing on different nodes;

grouping recipients according to a last node of divergence; and

transmitting the message to last node of divergence.

4. The method of claim 3, further comprising,

determining if the last node of divergence can transmit the message to the target node; and

diverging the message from the current node for transmission to the target node when it is determined that the last node of divergence cannot transmit the message to the target node.

5. The method of claim 1, further comprising tracking the number of transmissions between nodes for the message;

wherein the number of transmissions exceeds a minimum threshold, delaying additional transmissions for a settling period; and

wherein the number of transmissions exceeds a maximum threshold, return message as undeliverable.

6. The method of claim 1, wherein the transmission of the message to the target node fails, further comprising:

determining the available transmission schedules of network nodes; and

attempting transmission of the message to an intermediate node according to its priority identified in the destination alternative table and its availability schedule.

7. The method of claim 1, wherein one or more computer-readable media have computer-executable instructions for performing the computer-executable method of claim 1.

8. A message communication system for transmitting a message having a current server, a target server, and one or more intermediate servers connected to the network between the current server and the target server on a message communication network, comprising:

a destination alternative table of the intermediate servers located on the network having transmission costs to the target server, wherein the intermediate servers are prioritized according to their minimum transmission costs;

the current server being configured to execute computer-executable instructions for:

attempting transmission of a message to the target server; and

if the transmission of the message to the target server fails, attempting transmission of the message to at least one intermediate server as a function of its priority identified in the destination alternative table.

9. The system of claim 8, wherein said current server is further configured to execute computer-executable instructions for:

attempting transmission of the message to the intermediate node having the highest priority if the attempt to transmit the message to the target node fails; and

attempting transmission of the message to intermediate nodes of successively decreasing priority if the attempt to transmit the message to the highest priority intermediate node fails.

10. The system of claim 8, wherein the message is transmitted over a computer network from the current node to at least two target nodes, said current server is further configured to execute computer-executable instructions for:

creating a minimum spanning tree for the network of servers with the current server as a root;

determining a last server of divergence from the minimum spanning tree of at least two recipients residing on different servers; a grouping recipients according to a last server of divergence; and transmitting the message to last node of divergence.

11. The system of claim 10, further comprising,

determining if the last node of divergence can transmit the message to the target node; and

diverging the message from the current node for transmission to the target node when it is determined that the last node of divergence cannot transmit the message to the target node.

12. The system of claim 8, further comprising tracking the number of transmissions between nodes for the message;

wherein the number of transmissions exceeds a minimum threshold, delaying additional transmissions for a settling period; and

wherein the number of transmissions exceeds a maximum threshold, return message as undeliverable.

13. The system of claim 8, wherein the transmission of the message to the target node fails, further comprising:

determining the available transmission schedules of network nodes; and

attempting transmission of the message to an intermediate node according to its priority identified in the destination alternative table and its availability schedule.

14. One or more computer-readable media having computer-executable instructions for transmitting a message over a computer network, said network having a current node, a target node, and one or more intermediate nodes connected to the network between the current node and the target node, comprising:

instructions for determining the transmission costs from the intermediate nodes to the target node and prioritizing the intermediate nodes according to their associated transmission costs; and

instructions for attempting transmission of the message to the target node and if the transmission of the message to the target node fails, attempting transmission of the message to at least one intermediate node as a function of its priority identified in the destination alternative table.

15. The computer-readable media of claim 14, further comprising:

instructions for attempting transmission of the message to the intermediate node having the highest priority if the attempt to transmit the message to the target node fails; and

instructions for attempting transmission of the message to intermediate nodes of successively decreasing priority if the attempt to transmit the message to the highest priority intermediate node fails.

16. The computer-readable media of claim 14, wherein the message is transmitted over a computer network from the current node to at least two target nodes, the computer-readable media further comprising:

instructions for creating a minimum spanning tree for the network with the current node as a root;

instructions for determining a last node of divergence from the minimum spanning tree of at least two recipients residing on different nodes;

instructions for grouping recipients according to a last node of divergence; and

instructions for attempting transmission of the message to the last node of divergence.

17. The computer-readable media of claim 14, further comprising:

instructions for determining if the last node of divergence can transmit the message to the target node and diverging the message from the current node for transmission to the target node when it is determined that the last node of divergence cannot transmit the message to the target node.

18. The computer-readable media of claim 14, further comprising:

instructions for tracking the number of transmissions between nodes for the message;

instructions for delaying additional transmissions for a settling period wherein the number of transmissions exceeds a minimum threshold; and

instructions for returning the message as undeliverable wherein the number of transmissions exceeds a maximum threshold.

19. The computer-readable media of claim 14, wherein the transmission of the message to the target node fails, further comprising:

instructions for determining the available transmission schedules of network nodes; and

instructions for attempting transmission of the message to an intermediate node according to its priority identified in the destination alternative table and its availability schedule.

20. The computer-readable media of claim 14, further comprising:

instructions for maintaining a cache of all nodes having recently failed connections wherein the cache is utilized to avoid attempting to transmit the message to recently failed nodes.

FIG. 1

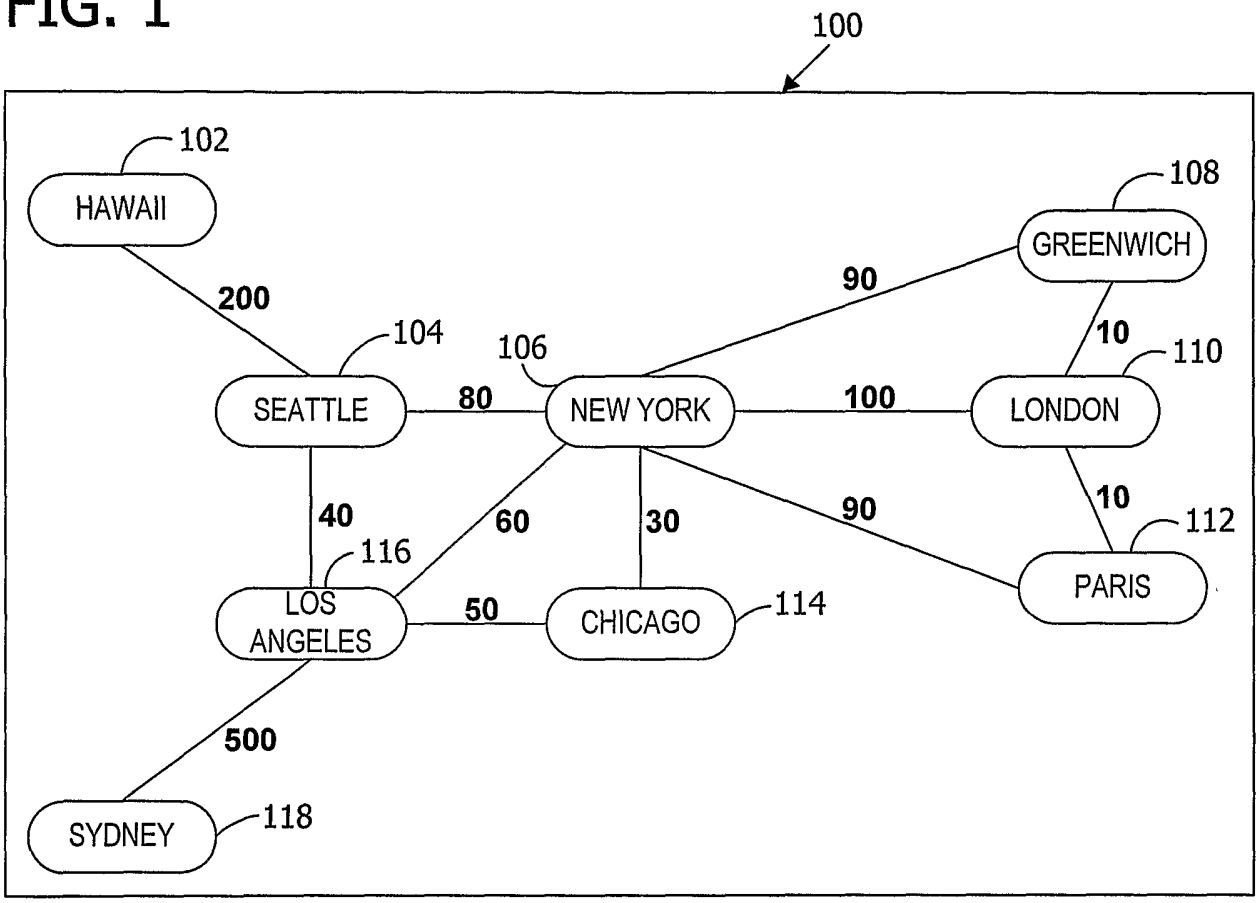

100
102
HAWAII
200
104
SEATTLE
106
NEW YORK
80
108
GREENWICH
90
10
110
LONDON
100
40
60
30
90
10
112
PARIS
116
LOS ANGELES
50
CHICAGO
114
500
SYDNEY
118

FIG. 2A

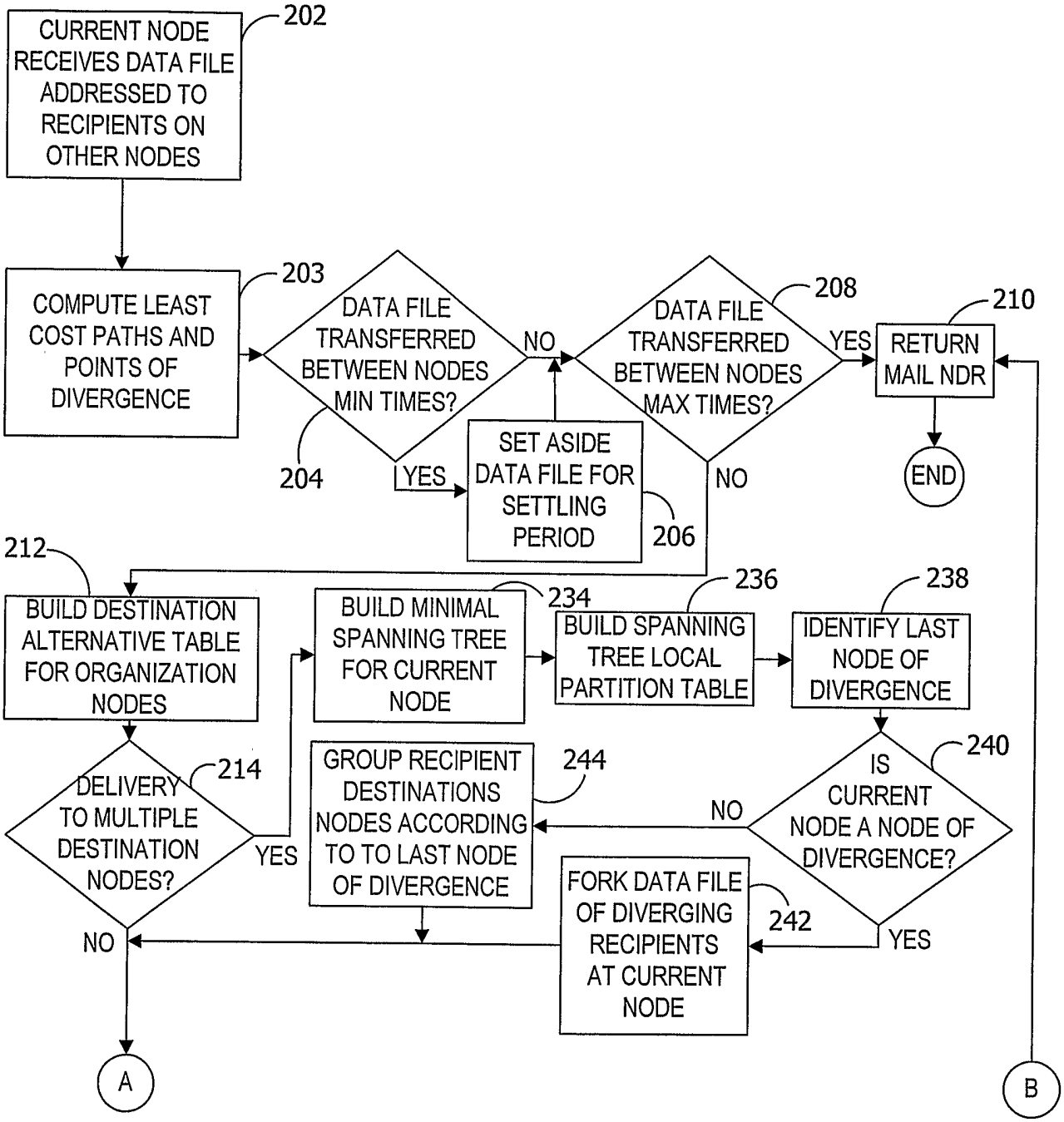

CURRENT NODE RECEIVES DATA FILE ADDRESSED TO RECIPIENTS ON OTHER NODES
202
COMPUTE LEAST COST PATHS AND POINTS OF DIVERGENCE
203
DATA FILE TRANSFERRED BETWEEN NODES MIN TIMES?
204
NO
YES
SET ASIDE DATA FILE FOR SETTLING PERIOD
206
DATA FILE TRANSFERRED BETWEEN NODES MAX TIMES?
208
YES
NO
RETURN MAIL NDR
210
END
212
BUILD DESTINATION ALTERNATIVE TABLE FOR ORGANIZATION NODES
DELIVERY TO MULTIPLE DESTINATION NODES?
214
YES
NO
A
BUILD MINIMAL SPANNING TREE FOR CURRENT NODE
234
BUILD SPANNING TREE LOCAL PARTITION TABLE
236
IDENTIFY LAST NODE OF DIVERGENCE
238
IS CURRENT NODE A NODE OF DIVERGENCE?
240
NO
YES
GROUP RECIPIENT DESTINATIONS NODES ACCORDING TO TO LAST NODE OF DIVERGENCE
244
FORK DATA FILE OF DIVERGING RECIPIENTS AT CURRENT NODE
242
B

FIG. 2B

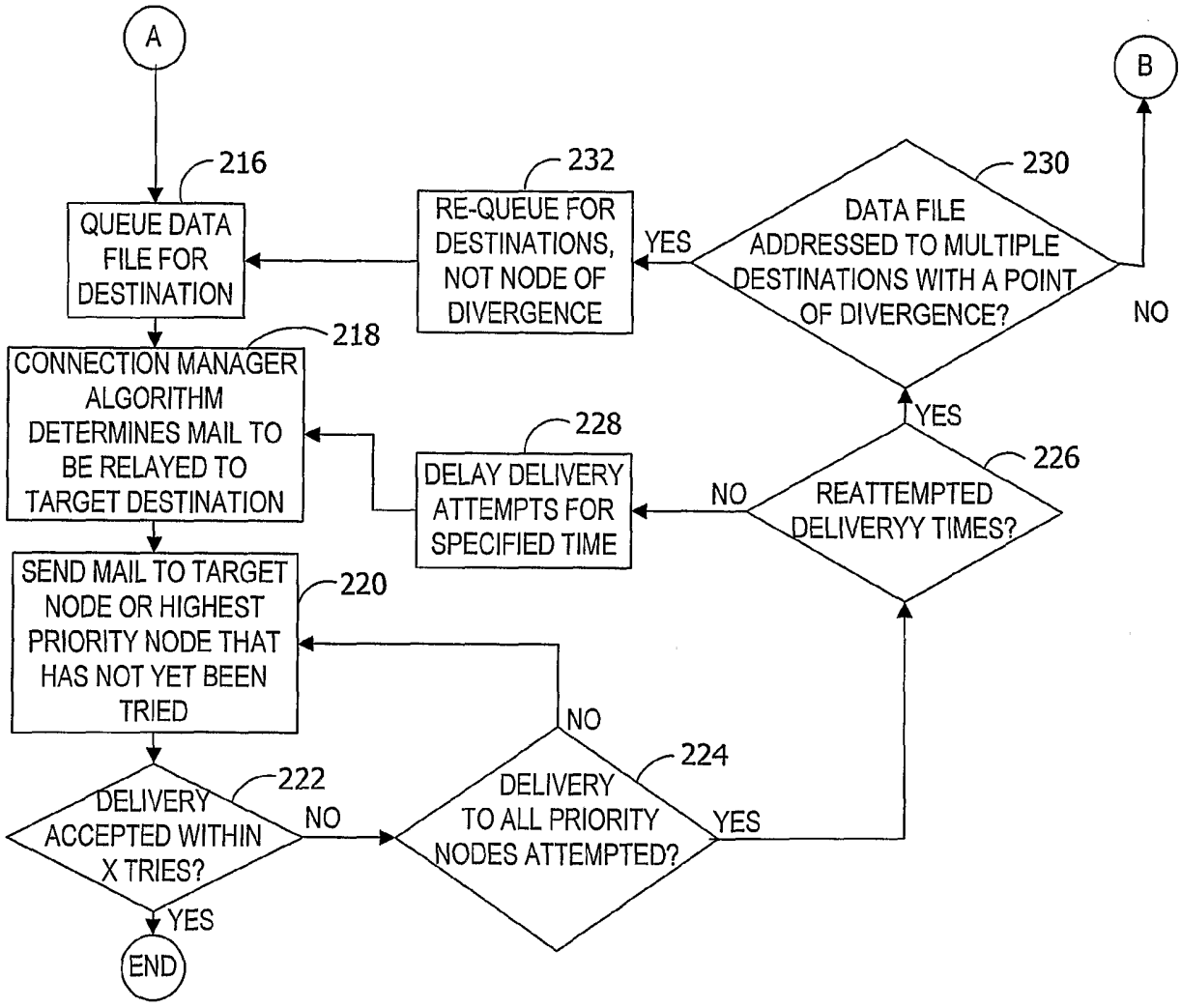
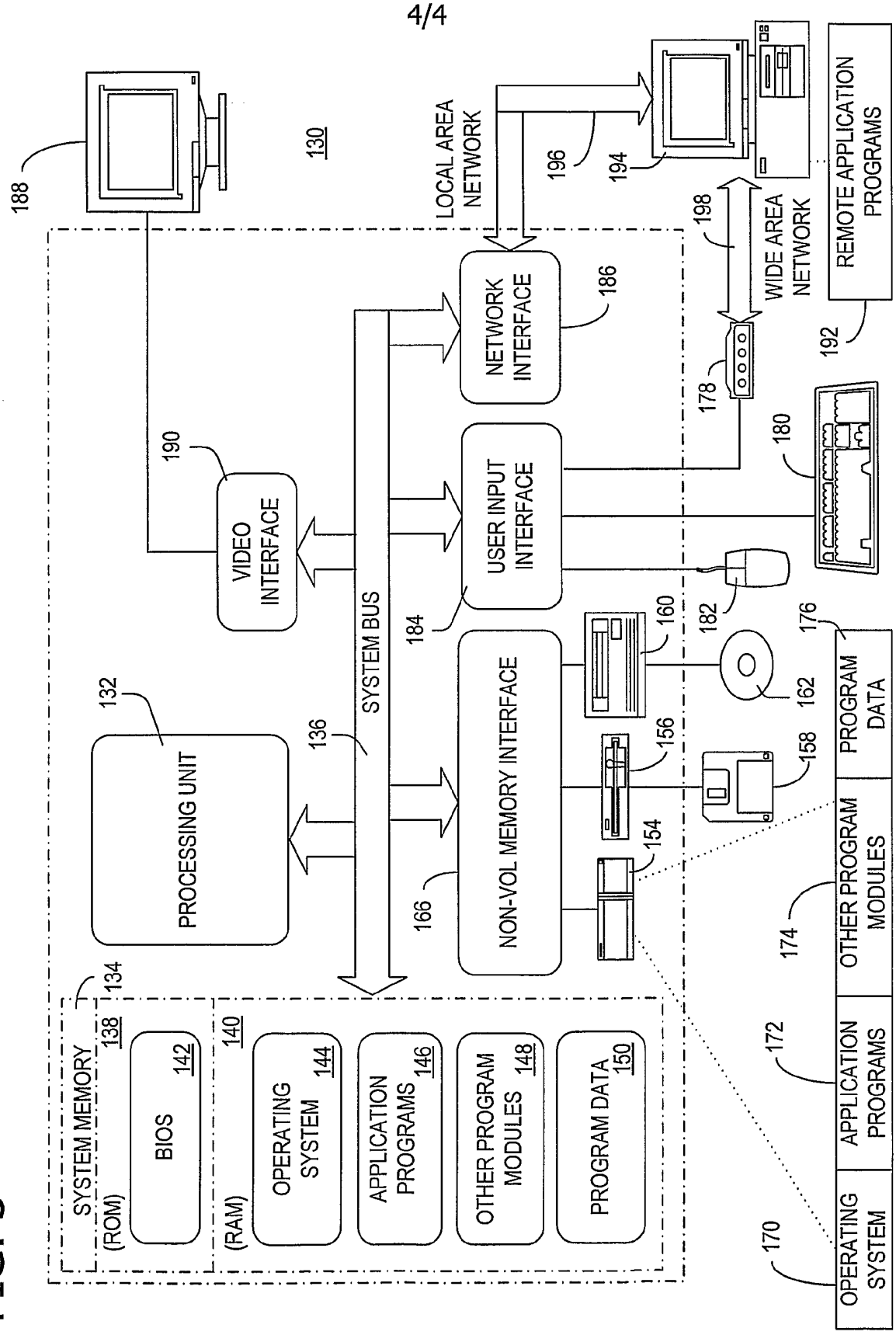

A
B
216
QUEUE DATA FILE FOR DESTINATION
232
RE-QUEUE FOR DESTINATIONS, NOT NODE OF DIVERGENCE
YES
230
DATA FILE ADDRESSED TO MULTIPLE DESTINATIONS WITH A POINT OF DIVERGENCE?
NO
218
CONNECTION MANAGER ALGORITHM DETERMINES MAIL TO BE RELAYED TO TARGET DESTINATION
228
DELAY DELIVERY ATTEMPTS FOR SPECIFIED TIME
NO
226
REATTEMPTED DELIVERYY TIMES?
YES
220
SEND MAIL TO TARGET NODE OR HIGHEST PRIORITY NODE THAT HAS NOT YET BEEN TRIED
NO
224
DELIVERY TO ALL PRIORITY NODES ATTEMPTED?
YES
222
DELIVERY ACCEPTED WITHIN X TRIES?
NO
YES
END

FIG. 3


130
188
LOCAL AREA NETWORK
196
194
198
WIDE AREA NETWORK
REMOTE APPLICATION PROGRAMS
192
NETWORK INTERFACE
186
178
180
190
VIDEO INTERFACE
USER INPUT INTERFACE
184
182
160
176
PROGRAM DATA
162
132
PROCESSING UNIT
136
SYSTEM BUS
156
158
154
166
NON-VOL MEMORY INTERFACE
174
OTHER PROGRAM MODULES
172
APPLICATION PROGRAMS
170
OPERATING SYSTEM
134
SYSTEM MEMORY
138
(ROM)
BIOS 142
140
(RAM)
OPERATING SYSTEM 144
APPLICATION PROGRAMS 146
OTHER PROGRAM MODULES 148
PROGRAM DATA 150