



(51) International Patent Classification:
H04L 12/56 (2006.01)

(21) International Application Number:
PCT/IB2008/003186

(22) International Filing Date:
21 November 2008 (21.11.2008)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (for all designated States except US): **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FIN-02150 Espoo (FI).

(71) Applicant (for LC only): **NOKIA, INC.** [US/US]; 6000 Connection Drive, Irving, Texas 75039 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **VISALA, Kari, Antero** [FI/FI]; Lindforsinkatu 8 d 70, FIN-33720 Tampere (FI).

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

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

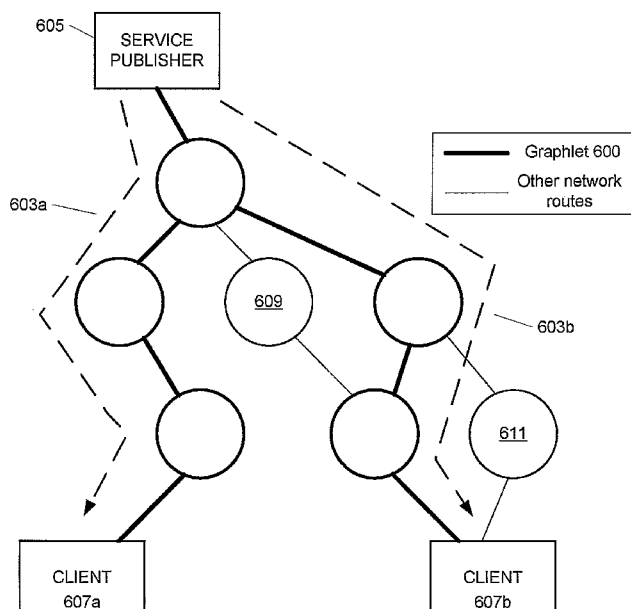
Published:

— with international search report (Art. 21(3))

(74) Agent: **WRIGHT, Bradley C.**; Banner & Witcoff, Ltd., 1100 13th Street, N.W., Suite 1200, Washington, District Of Columbia 20005-4051 (US).

(54) Title: SCALING AND LOAD-BALANCING OF DISTRIBUTED SERVICES

FIG. 6



(57) Abstract: In an active network supporting distributed services, graphlets may be used to increase the scalability and flexibility of the network. Graphlets may be used to define routes that a capsule or transmission is allowed to take for a particular service. By limiting the routes that a transmission for a service may use, resources may be allocated more equitably. Graphlets may further be scalable by using the resources provided by subscribers as they join the network.

SCALING AND LOAD-BALANCING OF DISTRIBUTED SERVICES

[01] Some aspects of the present disclosure relate to scaling and load-balancing for distributed services in a network.

BACKGROUND

[02] As use of networks like the Internet become more prevalent, applications and other types of software are beginning to migrate from a client-side only model to a more distributed on-line approach. This migration raises issues relating to the fairness in allocation of network resources and scalability.

BRIEF SUMMARY

[03] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

[04] Aspects described herein relate to a distributed service network such as a capsule-based active network that uses graphlets to define routes through which data associated with a particular service is permitted to travel. Thus, code and data relating to a service associated with a graphlet would be limited to the resources (e.g., active routers in the network) specified by the graphlet. Using graphlets, resources in an active network (e.g., processing power, storage space) may be distributed in a fair and equitable manner. Clients or services that might otherwise dominate resources may be better managed using such route and resource enforcement strategies. Graphlets may be structured similar to multicast trees and may be generated taking a variety of factors into account including locality, resources available, resources required, anticipated popularity of a service and the like. A graphlet is generally specific to a service so that resources may be divided on a per-service basis.

[05] According to another aspect, a client may subscribe to a service by sending a subscription message toward a home or proxy address of the service publisher. Upon the

subscription message intersecting a node in the graphlet corresponding to the desired service, the node may transmit code and data to the client along the path taken by the subscription message. The path may be established as a virtual channel between the node and the client for future communications relating to the service.

[06] According to another aspect, a service publisher node may use a proxy or rendezvous address instead of its actual address for publishing a service. This allows a proxy router or node to handle all communications for the service publisher. Using a rendezvous address, multiple service publisher nodes may be used to support a single service. Fault tolerance may thus be increased.

[07] According to yet another aspect, the resources of the active network may increase as more clients subscribe to services provided by the network. This allows the network to be flexible and scalable to an arbitrary number of clients. For example, a graphlet may grow dynamically as the number of simultaneous clients using the service at runtime grows. In this manner, the amount of resources consumed by the graphlet from the whole active network is proportional to the number of currently active clients of the service (this follows automatically, if the graphlet is, for example, a multicast tree where clients can dynamically join). Each client may thus bring their own resources to the graphlet thereby making the service scaleable to an arbitrary number of clients dynamically.

BRIEF DESCRIPTION OF THE DRAWINGS

[08] Certain embodiments are illustrated by way of example and not limited in the accompanying figures in which like reference numerals indicate similar elements and in which:

[09] FIG. 1 illustrates a block diagram of an example communication network in which one or more embodiments may be implemented.

[10] FIG. 2 illustrates a block diagram of an example communication device according to one or more aspects described herein.

[11] FIG. 3 illustrates an example of a capsule-based active network in which distributed services may be provided.

[12] FIG. 4 illustrates another example of a capsule-based active network in which distributed service may be provided.

[13] FIG. 5 illustrates an example of resource allocation in a distributed service network according to aspects described herein.

[14] FIGS. 6-9 illustrate examples of capsule-based active networks using graphlets according to aspects described herein.

[15] FIG. 10 illustrates an example data flow for subscribing to services in a network using graphlets according to aspects described herein.

[16] FIG. 11 illustrates an example method for subscribing to a service according to one or more aspects described herein.

[17] FIG. 12 illustrates an example method for processing communications between a service publisher and a client according to aspects described herein.

[18] FIG. 13 illustrates an example method for establishing and providing a service according to aspects described herein.

DETAILED DESCRIPTION

[19] In the following description of the various embodiments, reference is made to the accompanying drawings, which form a part hereof, and in which are shown by way of illustration various embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural and functional modifications may be made without departing from the scope of the present invention.

[20] FIG. 1 illustrates an example communication network through which various inventive principles may be practiced. A number of computers and devices including mobile communication device 105, mobile phone 110, personal digital assistant (PDA) or mobile

computer 120, personal computer (PC) 115, service provider 125 and content provider 130 may communicate with one another and with other devices through network 100. Network 100 may include wired and wireless connections and network elements, and connections over the network may include permanent or temporary connections. Communication through network 100 is not limited to the illustrated devices and may include additional mobile or fixed devices such as a video storage system, an audio/video player, a digital camera/camcorder, a positioning device such as a GPS (Global Positioning System) device or satellite, a television, an audio/video player, a radio broadcasting receiver, a set-top box (STB), a digital video recorder, remote control devices and any combination thereof.

[21] Although shown as a single network in FIG. 1 for simplicity, network 100 may include multiple networks that are interlinked so as to provide internetworked communications. Such networks may include one or more private or public packet-switched networks (e.g., the Internet), one or more private or public circuit-switched networks (e.g., a public switched telephone network), a cellular network configured to facilitate communications to and from mobile communication devices 105 and 110 (e.g., through use of base stations, mobile switching centers, etc.), a short or medium range wireless communication connection (e.g., Bluetooth®, ultra wideband (UWB), infrared, WiBree, wireless local area network (WLAN) according to one or more versions of Institute of Electrical and Electronics Engineers (IEEE) standard no. 802.11), or a high-speed wireless data network such as Evolution-Data Optimized (EV-DO) networks, Universal Mobile Telecommunications System (UMTS) networks, Long Term Evolution (LTE) networks or Enhanced Data rates for GSM Evolution (EDGE) networks. Devices 105-120 may use various communication protocols such as Internet Protocol (IP), Transmission Control Protocol (TCP), Simple Mail Transfer Protocol (SMTP) among others known in the art. Various messaging services such as Short Messaging Service (SMS) and/or Multimedia Message Service (MMS) may also be included.

[22] Devices 105-120 may be configured to interact with each other or other devices, such as content server 130 or service provider 125. In one example, mobile device 110 may include client software 165 that is configured to coordinate the transmission and reception of

information to and from content provider/server 130. In one arrangement, client software 165 may include application or server specific protocols for requesting and receiving content from content server 130. For example, client software 165 may comprise a Web browser or mobile variants thereof and content provider/server 130 may comprise a web server. Billing services (not shown) may also be included to charge access or data fees for services rendered. In one arrangement where service provider 125 provides cellular network access (e.g., a wireless service provider), client software 165 may include instructions for access and communication through the cellular network. Client software 165 may be stored in computer-readable memory 160 such as read only or random access memory in device 110 and may include instructions that cause one or more components (e.g., processor 155, a transceiver, and a display) of device 110 to perform various functions and methods including those described herein.

[23] FIG. 2 illustrates an example computing device such as mobile device 212 that may be used in network 100 of FIG. 1. Mobile device 212 may include a controller 225 connected to a user interface control 230, display 236 and other elements as illustrated. Controller 225 may include one or more processors 228 and memory 234 storing software 240. Mobile device 212 may also include a battery 250, speaker 252 and antenna 254. User interface control 230 may include controllers or adapters configured to receive input from or provide output to a keypad, touch screen, voice interface (e.g. via microphone 256), function keys, joystick, data glove, mouse and the like.

[24] Computer executable instructions and data used by processor 228 and other components of mobile device 212 may be stored in a storage facility such as memory 234. Memory 234 may comprise any type or combination of read only memory (ROM) modules or random access memory (RAM) modules, including both volatile and nonvolatile memory such as disks. Software 240 may be stored within memory 234 to provide instructions to processor 228 such that when the instructions are executed, processor 228, mobile device 212 and/or other components of mobile device 212 are caused to perform various functions or methods such as those described herein. Software may include both applications and operating system software, and may include code segments, instructions, applets, pre-

compiled code, compiled code, computer programs, program modules, engines, program logic, and combinations thereof. Computer executable instructions and data may further be stored on computer readable media including electrically erasable programmable read-only memory (EEPROM), flash memory or other memory technology, CD-ROM, DVD or other optical disk storage, magnetic cassettes, magnetic tape, magnetic storage and the like.

[25] Mobile device 212 or its various components may be configured to receive, decode and process various types of transmissions including digital broadband broadcast transmissions that are based, for example, on the Digital Video Broadcast (DVB) standard, such as DVB-H, DVB-H+, or DVB-MHP, through a specific broadcast transceiver 241. Other digital transmission formats may alternatively be used to deliver content and information of availability of supplemental services. Additionally or alternatively, mobile device 212 may be configured to receive, decode and process transmissions through FM/AM Radio transceiver 242, wireless local area network (WLAN) transceiver 243, and telecommunications transceiver 244. Transceivers 241, 242, 243 and 244 may, alternatively, include individual transmitter and receiver components.

[26] Although the above description of FIG. 2 generally relates to a mobile device, other devices or systems may include the same or similar components and perform the same or similar functions and methods. For example, a stationary computer such as PC 115 (FIG. 1) may include the components described above and may be configured to perform the same or similar functions as mobile device 212 and its components.

[27] According to some aspects, a device such as devices 105-120 of FIG. 1 or mobile device 212 of FIG. 2 may be configured to purchase and download media content. For example, mobile phones and other computing devices may have the capability to access network sites that provide music, television shows, movies, text, ring tones, other audio, video or text and the like for purchase and subsequent download. However, as with many purchases, users typically want the opportunity to preview content that they are considering buying. In many instances, a preview function and a purchase option are separate aspects of a site and require multiple user interactions to go between previewing and purchasing content.

[28] Mobile devices such as mobile device 212 and desktop or non-mobile devices like computer 115 are typically designed and configured to process many applications and significant amounts of data on a daily basis. From web browsers to video games to word processing software, developers continue to develop ever increasing numbers of applications to enhance and unlock the capabilities of such computing and communication devices. One direction in which applications are beginning to trend is distributed processing. Generally, distributed processing refers to the use of multiple processing systems or processors to provide a service or execute an application. This provides increased processing power and efficiency in executing a task.

[29] FIG. 3 illustrates a capsule-based active network in which services are provided to clients in a distributed manner. In active network 300, application code may be loaded to nodes 301 for execution and processing. Application code may be received from a client, e.g., client 305b or from a service provider/publisher 307a. A capsule, as used herein, refers generally to a data structure for transporting payload data as well as code that may be executed and used to modify the payload data. In a distributed system like network 300, a service provider/publisher 307a may provide one set of code to the network nodes 301 while a client system 305b wishing to use the service may provide another set of code to be executed in conjunction with the publisher/provider's code. For example, a service provider such as service publisher 307b may publish function calls, application protocol interfaces (API), service protocols and other generic service code segments for the service being provided. The client code may then use the service code, APIs or function calls to modify or otherwise process the payload data. Thus, a client might not need to download massive amounts of code to run an application or to access a service. Instead, the client (e.g., client 305a) might only need a portion or subset of the overall set of code that can be complemented with service code supplied by a service publisher or provider to execute an application or service. Accordingly, execution of the code and processing of data may be distributed to nodes in the network and need not be processed by the client 305a alone.

[30] Using a distributed system such as a capsule-based active network, there are occasions where one client or service may occupy a significant or disproportionate amount of

resources. As such, one client or service may slow down or halt the processing of other clients or services trying to operate on the same network. Such lack of load balancing may also be treated as a vulnerability that could make systems in the network and the network in general susceptible to exploitation and attack by hackers and other individuals. Some solutions call for certification of client code or capsules to insure that the capsules are issued by a trusted entity while others utilize time-to-live (TTL) parameters that limit the global resources consumed by the capsules. Using such methods may potentially restrict the flexibility of the distributed system, requiring that capsules obtain pre-approval or certification. Such limitations may even open new vulnerabilities (e.g., if a certification key leaks out into the public).

[31] FIG. 4 illustrates a capsule-based active network providing resource allocation to enhance scalability. Network 400 may include multiple active routers 403 that may each be configured to execute code contained in capsules published by either servers 407 or clients 405. An active router refers generally to a router that is configured to execute code for and support a service published by clients 405 or servers 407. In some configurations, an active router relates to a routing device that is able to dynamically run new code that was not hardwired into the operation of the router. Each of active routers 403 may be configured to process a different piece of code or data to complete processing more quickly and efficiently. In one configuration, active router 403a may be configured to extract and process a first piece of code from a capsule and to forward the capsule on to a second active router 403b configured to process a second or remaining piece of code from the same capsule. Capsules and code or data contained therein may be specific to a particular service. Services may be published to network 400 by a client, e.g., client 405c. Once the service (e.g., Service X) is published, one or more servers 407 may be configured to act as persistent publisher nodes of the service by publishing code and data that implement and support the service. Thus, even if client 405c stops operating in the network, the service may survive through servers 407.

[32] In one or more configurations, each service may include one or more rendezvous addresses that serve as proxy addresses for servers 407. By using rendezvous addresses, a client such as client 405b might not need to know the actual address of servers 407 to access

the published service. Rendezvous addresses may be used in instances where a first server, e.g., server 407a, passes its responsibilities to a second server such as server 407b. Instead of having to notify each client 405a, 405b and 405c of the change, the server 407a may simply notify a proxy router or server 403c corresponding to the rendezvous address. Network 400 may further include one or more passive routers 409 that are not configured to execute code, but to merely pass data along appropriate channels. Alternatively, if servers 407 are not configured to act as persistent publishers of the service (i.e., to replicate the service published by client 405a), the service might only be available for as long as the publishing client 405c or a client using the service is active and providing resources.

[33] Resources for executing and providing a service may be derived from both client allocated resources as well as server allocated resources. Allocated resources may be reserved for use by the client or server to which the resources are allocated. In the distributed network 500 of FIG. 5, for example, client 505a may be allocated processing or storage resources 511 in active router 507a, while client 505b may be allocated processing or storage resources 513 in each of active routers 507a and 507c. A client may have resources allocated to it across 1, 2, 5, 10, 14 or any number of network nodes (e.g., servers, routers, other clients). Servers 509 may be allocated resources in similar fashion. Additionally or alternatively, clients 505 and servers 509 may provide resources of their own. Resources may further include cache/memory/disk space, processor cycles (divided by a scheduler possibly), information stored in the router, and/or control of external devices attached to the router.

[34] The aggregate resources in an active network may be divided on a per-service basis to balance the load across the various active routers in the network and to insure that one service or one client does not use or occupy a disproportionate amount of resources in the network. Specifically, a routing algorithm may create routing graphs (also referred to as graphlets herein) that define where capsules for a particular service are allowed to travel. The routing graphs may be defined to distribute processing load among all of the resources in the network. That is, by limiting the path on which the capsules are allowed to travel and the active routers in which capsule code is allowed to execute, the network may insure that a

service does not expand to a point where it is dominating the resources of the entire network. Additionally or alternatively, graphlets may be service specific.

[35] Multiple methods and algorithms for distributing resources among services and generating routes for graphlets may be used. For example, a distributed hash table (DHT) like CHORD, KADEMLIA, PASTRY and TAPESTRY may be used to form a routing network as they are designed to quickly find the node in a network responsible for some key in the hash table. By routing recursively towards some key (that can be, for example, the identity of some service) in the DHT, a path from the client (or server) is created in the network of DHT nodes. In this way, multiple clients and servers can “rendezvous” at some point in the network and these paths can be combined to form a tree-like graphlet. DHT implementations typically guarantee that if the keys are evenly distributed in the key space, then each node in the DHT serves approximately the same amount of keys and the paths formed using the method described above are balanced evenly in the network. Some DHTs guarantee in addition that locality is achieved when possible. Hierarchical DHTs like CANON and CYCLONE are specifically designed to join multiple existing DHTs into a hierarchical, larger DHT with locality. Alternatively, the routes “inside” the DHT might not be directly used to form the routes, but the DHT can function as a topology layer so that each node in the DHT controls some part of the actual network and the DHT is only used to build a route to the actual network with fast routers. Each active router may serve multiple graphlet paths simultaneously and can use typical scheduling to divide their processing power among all paths that go through the active router.

[36] FIG. 6 illustrates an example graphlet defining the routes that a capsule for a service published by a service publisher may take to reach one or more subscribing clients. As described, capsules and other communications for the service may be restricted to the routes specified in the graphlet (i.e., the capsules are not allowed to use other routes or nodes). Graphlet 600, for example, defines routes 603a and 603b that a capsule may take from service publisher 605 to clients 607a and 607b, respectively. Nodes 609 and 611 are not included in graphlet 600 and thus, capsules for the service published by service publisher 605 might not be allowed to travel through and use the resources of nodes 609 and 611. The

resources of nodes 609 and 611 may, instead, be reserved for another service published by another service publisher or by service publisher 605. By dividing nodes in such a manner, the processing load of nodes 609 or 611 or any of nodes 613 may be maintained at reasonable and manageable levels. Data and code (e.g., capsules) may be sent in either direction of graphlet 600. Alternatively or additionally, one or both of nodes 609 and 611 may support multiple services by dividing its resources between the multiple services.

[37] According to one or more configurations, each graphlet/service (e.g., graphlet 600) is associated with an identity that can be used to access it from anywhere in the network. The service (publisher) may have a special role where only it can inject new code in the capsules on the path reserved from the network (but messages can, of course, be sent by both servers and clients along the path created). If only the (authorized) service can upload new code to the routers on the path, security may be enhanced. The actual path created for the graphlet 600 is dynamic and changes based on the use of the particular service. Generally, the network is responsible for and configured to perform the allocation of resources for this path (i.e., graphlet 600) based on the set of clients that have subscribed at each moment to the service. Alternatively or additionally, the service may conduct or take part in the allocation of network resources and definition of graphlet 600.

[38] If a new client should subscribe to the service corresponding to graphlet 600, the graphlet may be potentially expanded to include new active routers or nodes in the network accessible to the new subscribing client. FIG. 7, for example, illustrates the new subscription of client 707 and the addition of node 703 to the graphlet 600. In this configuration, graphlet 600 may be redefined or modified to include node 703 as part of the allowable path so that capsules are able to reach client 707 from service publisher 605. Changes to graphlet 600 may be propagated to existing nodes in graphlet 600 by the service publisher 605.

[39] FIG. 8 illustrates a capsule-based active network that includes three graphlets 801a, 801b and 801c defining routing for services published by service publisher 803a and 803b. Specifically, service publisher 803a is publishing two different services that correspond to graphlets 801a and 801b while publisher 803b is publishing a single service corresponding to graphlet 801c. Since client 805b is subscribed to services corresponding to both graphlets

801b and 801c, graphlets 801b and 801c may overlap, e.g., at node 807. In particular, service capsules transmitted by either publisher 803a or publisher 803b for services to which client 805a is subscribed may share node 807 of graphlets 801a and 801c when addressed to client 805b.

[40] In some instances, service capsules or code may join and use other graphlets in the network. For example, because client 805b is accessing two different services, service code from a first service corresponding to graphlet 801b may join graphlet 801c (corresponding to a second service) at node 807 to prevent client 805b from commandeering the resources of node 807. Accordingly, when the first service corresponding to graphlet 801b joins graphlet 801c at node 807, the original resources allocated to client 805b may be divided among the first service and the second service. That is, additional resources beyond what was originally allocated to client 805b for accessing various services might not be provided to process both code from the first service and the second service for client 805b. If node 807 were to provide additional resources for each additional service to which client 805b subscribes, client 805b may begin to consume all resources available in node 807 by subscribing to a sufficient number of services.

[41] FIG. 9 illustrates a hierarchical division of resources at a node included in multiple graphlets corresponding to multiple services accessed by a single client. Node 905 in an active network may divide its resources between two graphlets or services, Service X and Service Y. Service code and processing for service X would be performed using resources 901a while service code and processing for service Y would be performed using resources 901b. Node 905 may further divide resources for a service, such as resources 901a for service X, among the various subscribers of the service. For example, client 903a may be allocated resources 909 while another client such as client 903b may be allocated resources 911 for accessing service X.

[42] In some instances, a client such as client 903a may access both services X and Y, causing the potential for client 903a to occupy both resources 901a and 901b. To eliminate the potential of a single client occupying significant resources in a single node, service Y, when accessed by client 903a, may be joined to service X. That is, processing for service Y

on behalf of client 903a may share resources 901 used by client 903a for accessing service X. Allocation of resources 901a between services X and Y may be determined based on priority or importance. Priority or importance may be defined between the services (e.g., by the publishers), based on an amount of data or code processed, by the client or based on combinations thereof. Allocated resources may include processor cycles, storage space, network bandwidth and the like. Code for different services may be executed in different logical spaces or sandboxes in an active router (e.g., node 905) to limit interaction between the services and to aid in the fair division of resources. If client 903c were to access service Y, however, processing may occur in node 905 using resources 901b since client 903c is not accessing both services X and Y. Using the above allocation strategy, an active network may insure that a single client or a single service does not overrun available resources.

[43] In one or more arrangements, a graphlet may be formed as a source-based multicast tree. In source-based multicast trees, data messages might only be allowed to be sent from the source to multiple receivers or, alternatively, each participant in the multicast can send messages that everybody subscribed to the multicast channel then receives. With graphlets, however, the tree (or more generally, a graph) formed in the network is more concrete and limits the possible paths taken by messages, even among subscribers. Stated differently, a packet sent by a service publisher across a graphlet may be limited to less than all subscribers of the service whereas using a multicast tree, a packet is typically not limited and would be sent to all subscribers. Additionally, the network may separate the publishers/service from the subscribers/clients so that only publishers (possibly authenticated) can send capsules to the network. As noted, the capsules may contain new code that is uploaded to the active routers on the path reserved in the network for the graphlet. Subscribers may send messages along the tree but they might not be allowed to send capsules. Thus, graphlets may also impose rules such as what types of data may be transmitted through the graphlet by a subscriber or a service publisher. Further, active code may change the semantics of a router from a simple multicast so that messages are processed along the path and possibly not forwarded to all parts of the tree (when the code is run inside the active router, the active router should generally know its location in the graphlet's communication tree and at which directions the service can be found).

[44] FIG. 10 illustrates a network having graphlet 1015 through which a service may be provided by service publisher 1001 to clients 1005. A client such as client 1005b may subscribe to an advertised service by transmitting a subscription message toward the source of the multicast tree or graphlet 1015 (i.e., service publisher 1001). The path along which the subscription message travels may be reserved as a virtual channel 1025. Virtual channels such as virtual channel 1025 may be needed since a service publisher 1001 might not know which clients will want to subscribe to a service corresponding to graphlet 1015. Thus, the graphlet 1015 might not have a route to every client. The virtual channel 1025 allows a client to reach and access the graphlet and the service. The subscription message path (i.e., virtual channel 1025) will normally eventually join or intersect graphlet 1015. For example, virtual channel 1025 and graphlet 1015 intersect at active router 1020. Active router 1020 may be pre-configured with service code provided by service publisher 1001 and may transmit capsule code intended for execution in client 1005b to client 1005b along the virtual channel 1025 upon receiving the subscription message. The capsule code may include an applet or code segment that operates as a local representative or interface in client 1005b for the subscribed service. Subsequent communications between client 1005b and service publisher 1001 may be transmitted along the virtual channel 1025 and the graphlet 1015.

[45] Once the virtual channel 1025 reaches a node in the graphlet 1015, the virtual channel may become part of the graphlet. Virtual channel 1025 may remain a part of the graphlet 1015 as long as the client 1005b is using the service. Accordingly, each client 1005 may "bring his own branch" to the graphlet 1015, thereby adding resources to the graphlet 1015. The graphlet 1015 may thus grow dynamically as a function of the number of the clients and the service can scale to arbitrary number of clients.

[46] In one or more configurations, instead of using a service publisher's actual or home address, the service publisher may instead use a proxy or rendezvous router and address to facilitate communication with clients. For example, in FIG. 10, router 1030 may be configured to act as a rendezvous router for service publishers 1001 and 1002. Using a rendezvous or proxy address, multiple service publishers such as publishers 1001 and 1002 may be used to provide a service, thus sharing the communication and processing load.

Additionally, the use of multiple service publishers increases fault tolerance, allowing for failover in the event one service publisher (e.g., service publisher 1002) fails (in contrast to having a single service publisher and not being able to recover). Clients 1005 may subscribe to a service in a manner similar to that described above, but instead of transmitting subscription messages and other communications to the home address of a publisher node (e.g., publisher 1001), communications are directed to the address of proxy router 1030 which may then forward the transmissions to an appropriate service publisher node such as node 1001 or 1002.

[47] Alternatively or additionally, graphlets may be formed using multicast routing topology (e.g., for advertising services) based on distributed hash table (DHT) strategies generally known in the art. For example, DHTs may be used as overlay networks (and in some cases on a network level (e.g., HERMES)) for multicast and publisher/subscriber systems to provide load balancing of tree state information to all nodes of the DHT and a guarantee that new subscribers will join existing (multicast) trees early/locally and an increased probability of cache hits by using the same nodes for the same trees. While some active routers have been used together with multicast (for example to filter incoming stream based on subscriber preferences), capsule based active networks have not been built using DHT routing strategies, especially those that are scalable.

[48] FIG. 11 illustrates an example method by which a client may subscribe to a service and the service's corresponding graphlet. In step 1100, a client may receive an advertisement from a service publisher identifying an available service. The service advertisement may be broadcast to clients that have subscribed for such advertisements or to all clients in the network. The service advertisement may include a source address for the service. In one example, the source address may be the home address of the service publisher. Alternatively, the source address may be a rendezvous or proxy address of a proxy router or node in the network. In step 1105, the client may transmit a subscription request addressed to either the home address or the rendezvous address of the service publisher. As discussed, the subscription request will eventually intersect a part of the graphlet of the desired service. Once that occurs, the client may receive service code and/or data from an active router at

which the intersection occurs in step 1110. In step 1115, the client may execute the service code to activate or update a local representative or interface in the client device. In particular, the local representative may be configured to process data and code according to protocols used by the service publisher. A service publisher may thus dynamically and transparently (to the client) update the protocols or code used in the service by transmitting updates to the client and active nodes through the graphlet. The use of a local interface also allows the service publisher to use whatever protocol it desires without requiring the client to have a priori knowledge of the protocol. A virtual channel may further be established between the active router and the client to facilitate future communications in step 1120. For example, the client may store information specifying the nodes in the virtual channel so that the local representative or another component of the client may access the service through the graphlet.

[49] In one example, a distributing computing project such as SETI@home may use graphlets to distribute updated protocols and interfaces used by each client computer or other device registered to process scientific data for the project. Additionally, active routers in the network may be used to process and check results provided by the client computers.

[50] FIG. 12 illustrates an example method in which an active router processes communications for a service in accordance with a graphlet. In step 1200, the active router may receive a subscription request corresponding to a service and graphlet supported by the active router. For example, the active router may provide resources for processing data and code associated with the service. In step 1205, the active router may retrieve code and data associated with the requested service. The code may be published to each active router by the service publisher prior to the service being advertised so that the active router may set up corresponding protocols and execute relevant code segments prior to the service becoming active. In step 1210, the active router may determine a virtual channel or route from which the subscription request arrived. In one example, the router may make such a determination by extracting route information from the subscription request message. In step 1215, the active router may subsequently transmit the code and data to the client along the determined channel. Optionally, the identified virtual channel may be added to an existing graphlet for the requested service in step 1217.

[51] Once a client has been subscribed to a service supported by the active router, the active router may begin receiving communications from the client in step 1220. The communications may identify a service to which the communication corresponds. Using the identification, the active router may determine a graphlet corresponding to the service. As previously noted, the graphlet may define the paths and nodes to which transmissions for the service are limited. This insures that a service or client does not consume a disproportionate or disruptive amount of resources in the network. In step 1225, the active router may determine whether the communication includes code that needs to be executed at the active router. For example, a capsule may specify whether it includes code that needs to be executed at a distributed system or if the code is intended for the service publisher. If code is to be executed at the router, the router may identify resources allocated to the service or to the client or both in step 1230. Once identified, the resources may be used to process the capsule code in step 1235. If the communication does not include code to be executed at the router or if communications are directed to the service publisher, the router may forward the communications to another node or to the service publisher in accordance with the graphlet in step 1240. Similarly, if code is received from the service publisher, the router may determine if code needs to be executed at the router and forward communications on to the intended recipients. In one or more configurations, the communications may be forwarded along a virtual channel or route established between the router and a client.

[52] In one or more configurations, the router or another network entity such as the service publisher may determine whether a client has unsubscribed to the service. If so, the router or the other network entity may remove the virtual channel from the graphlet corresponding to the service.

[53] FIG. 13 illustrates an example method for establishing a service in an active network. In step 1300, a service publisher may receive service information from a client or other system wishing to publish a new service. The information may include service code and data for providing the service to clients. In step 1305, the service publisher may establish a graphlet for the service. The service publisher may establish the graphlet in a variety of ways and taking into account a variety of factors such as locality of known clients, fair distribution

of resources, anticipated popularity of the service and the like. For example, as discussed herein, an initial graphlet may be formed using multicast routing strategies such as DHT and the like. In step 1310, the service publisher may broadcast the availability of the service throughout the network. The service publisher may further transmit service capsules to each active router in the graphlet in step 1315. The service capsules may include code for execution at the active router or code intended for a client (e.g., local interface code) or both. In step 1320, the service publisher may receive notifications of a new client subscribing to the service. In step 1325, the service publisher may add the client's information to a database of subscribers. Optionally, the service publisher may modify the graphlet to insure the most efficient routing of messages to and from the client in step 1330. For example, a virtual channel used by the client to access the graphlet (i.e. if the client cannot directly access a node in the graphlet) may be added to the graphlet upon the client subscribing. In step 1335 and 1340, the service publisher may subsequently receive and process communications from the subscribed clients. In one or more configurations, the service publisher may also instruct an active router in the graphlet to forward service code for a local interface/representative to the client.

[54] Active routers may execute code in a variety of manners. In one example, an active router may process code using a sandbox or other execution environment to reduce the likelihood of services intermingling in resources. Execution environments may include JAVA Virtual Machine (JVM) environments and the like. Using such execution environments, resource usage may be readily and easily managed between different services. Additionally, security risks may be better managed in a contained environment. Security may further be enhanced using code that is signed by a service author to insure the integrity of the data.

[55] Graphlets may be used in a variety of applications including caching data for data dissemination, creating real-time media, content scaling, filtering, access control, scientific computing and mobile/offline computing. For example, a graphlet may be used to form an inverted multicast where active clients may produce a single media stream together. A distributed audio mixing network may be used, for instance, to allow clients to stream audio

to a communication tree where active routers produce a single audio stream from the streams provided by the multiple clients. In another example, graphlets may be used to scale content for particular devices based on their capabilities. Active routers in the graphlets may modify the content in accordance with a client to which the content is being sent.

[56] Additionally or alternatively, graphlets may be formed as peer-to-peer networks, i.e., between clients without network level active routers. This architecture is similar to existing peer-to-peer networking protocols such as BITTORRENT. Services may be accessed using a file (e.g., a .torrent file) that points to a tracker controlling the formation of the peer-to-peer overlay and the relaying of information about the locations of other clients to each joining client. Instead of communicating parts of a shared file (like in BITTORRENT), the clients may form a tree-like communication channel rooted at a seeder node. The seeder node may then transmit capsules containing client-side code to each of the clients. A seeder node refers to a node in the network that provides data or publishes services to other nodes or clients in the network.

[57] It should be understood that any of the method steps, procedures or functions described herein may be implemented using one or more processors in combination with executable instructions that cause the processors and other components to perform the method steps, procedures or functions. As used herein, the terms "processor" and "computer" whether used alone or in combination with executable instructions stored in a memory or other computer-readable storage medium should be understood to encompass any of various types of well-known computing structures including but not limited to one or more microprocessors, special-purpose computer chips, field-programmable gate arrays (FPGAs), controllers, application-specific integrated circuits (ASICs), combinations of hardware/firmware/software, or other special or general-purpose processing circuitry.

[58] The methods and features recited herein may further be implemented through any number of computer readable media that are able to store computer readable instructions. Examples of computer readable media that may be used include RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, DVD or other optical disk storage, magnetic cassettes, magnetic tape, magnetic storage and the like.

[59] Additionally or alternatively, in at least some embodiments, the methods and features recited herein may be implemented through one or more integrated circuits (ICs). An integrated circuit may, for example, be a microprocessor that accesses programming instructions or other data stored in a read only memory (ROM). In some such embodiments, the ROM stores programming instructions that cause the IC to perform operations according to one or more of the methods described herein. In at least some other embodiments, one or more the methods described herein are hardwired into an IC. In other words, the IC is in such cases an application specific integrated circuit (ASIC) having gates and other logic dedicated to the calculations and other operations described herein. In still other embodiments, the IC may perform some operations based on execution of programming instructions read from ROM or RAM, with other operations hardwired into gates and other logic of IC. Further, the IC may output image data to a display buffer.

[60] Although specific examples of carrying out the invention have been described, those skilled in the art will appreciate that there are numerous variations and permutations of the above-described systems and methods that are contained within the spirit and scope of the invention as set forth in the appended claims. Additionally, numerous other embodiments, modifications and variations within the scope and spirit of the appended claims will occur to persons of ordinary skill in the art from a review of this disclosure.

We Claim:

1. A method comprising:
establishing a graphlet for transmitting data associated with a service published in a network, wherein the graphlet is specific to the service and defines one or more routes to which transmission of data for the service is restricted; and
transmitting code to one or more nodes in the graphlet, wherein the code is configured for execution at each of the one or more nodes.
2. The method of claim 1, wherein the graphlet further includes one or more passive nodes that are not configured to execute the code.
3. The method of claim 1, further comprising advertising the service to one or more clients in the network.
4. The method of claim 1, wherein the code includes client-side code configured to establish a local interface in each subscribing client.
5. The method of claim 1, wherein establishing the graphlet includes determining resources to allocate to the service, wherein the resources include resources at the one or more nodes.
6. The method of claim 1, further comprising designating a proxy node different from the service publisher to receive communications relating to the service.
7. The method of claim 1, wherein the code includes a service protocol and wherein the method further comprises updating the service protocol by transmitting new code to the one or more nodes in the graphlet.
8. A method comprising:
determining, at a device, a source address of a service;

transmitting a subscription message addressed to the source address;
receiving client code for execution at the device; and
executing the client code to generate a local interface for accessing the service through a graphlet, wherein the graphlet defines one or more routes through nodes in a network to which transmissions associated with the service are restricted.

9. The method of claim 8, wherein the source address of the service includes an address of a publisher of the service.

10. The method of claim 8, wherein the source address includes an address of a proxy node different from a publisher of the service.

11. The method of claim 8, wherein the local interface defines communication protocols for communicating with a publisher of the service.

12. The method of claim 8, wherein transmitting the subscription message to the source address includes transmitting the subscription message to a node in the graphlet along a virtual channel.

13. The method of claim 12, wherein the client code is received through the virtual channel.

14. A method comprising:

receiving, at a node in a network, a communication from a client, the communication addressed to a publisher of a first service; and

transmitting the communication using a graphlet specific to the first service, wherein the graphlet defines one or more routes through nodes in the network to which transmissions associated with the first service are restricted.

15. The method of claim 14, wherein the graphlet includes less than all nodes in the network.

16. The method of claim 14, further comprising:
receiving code from the publisher of the first service; and
executing the code to update at least one service protocol associated with the first service.
17. The method of claim 14, further comprising dividing resources of the node between the first service and a second service.
18. The method of claim 17, wherein first code associated with the first service is executed in a first virtual environment of the node separate from a second virtual environment of the node in which second code associated with the second service is executed.
19. An apparatus comprising:
a processor; and
memory storing computer readable instructions that, when executed, cause the apparatus to:
establish a graphlet for transmitting data associated with a service published in a network, wherein the graphlet is specific to the service and defines one or more routes to which transmission of data for the service is restricted; and
transmit code to one or more nodes in the graphlet, wherein the code is configured for execution at each of the one or more nodes.
20. The apparatus of claim 19, wherein the graphlet further includes one or more passive nodes that are not configured to execute the code.
21. The apparatus of claim 19, wherein the code includes client-side code configured to establish a local interface in each subscribing client.

22. The apparatus of claim 1, wherein the memory further stores computer readable instructions that, when executed, cause the apparatus to determine resources to allocate to the service, wherein the resources include resources of the one or more nodes.
23. An apparatus comprising:
a processor; and
memory storing computer readable instructions that, when executed, cause the apparatus to:
determine a source address of a service;
transmit a subscription message addressed to the source address;
receive client code for execution at the apparatus; and
execute the client code to generate a local interface for accessing the service through a graphlet, wherein the graphlet defines one or more routes through nodes of a network to which transmissions associated with the service are restricted.
24. The apparatus of claim 23, wherein the source address of the service includes an address of a publisher of the service.
25. The apparatus of claim 23, wherein the source address includes an address of a proxy node different from a publisher of the service.
26. The apparatus of claim 23, wherein the apparatus transmits the subscription message to the source address by transmitting the subscription message to a node in the graphlet along a virtual channel.
27. An apparatus comprising:
a processor; and
memory storing computer readable instructions that, when executed, cause the apparatus to:
receive, at a node in a network, a communication from a client, the communication addressed to a publisher of a first service; and

transmit the communication using a graphlet specific to the first service, wherein the graphlet defines one or more routes through nodes in the network to which transmissions associated with the first service are restricted.

28. The apparatus of claim 27, wherein the graphlet includes less than all nodes in the network.

29. The apparatus of claim 27, wherein the computer readable instructions, when executed, further cause the apparatus to:

receive code from the publisher of the first service; and

execute the code to update at least one service protocol associated with the first service.

30. One or more computer readable media storing computer readable instructions that, when executed, cause an apparatus to:

establish a graphlet for transmitting data associated with a service published in a network, wherein the graphlet is specific to the service and defines one or more routes to which transmission of data for the service is restricted; and

transmit code to one or more nodes in the graphlet, wherein the code is configured for execution at each of the one or more nodes.

31. The one or more computer readable media of claim 30, wherein the graphlet further includes one or more passive nodes that are not configured to execute the code.

32. The one or more computer readable media of claim 30, wherein the code includes client-side code configured to establish a local interface in each subscribing client.

33. One or more computer readable media storing computer readable instructions that, when executed, cause an apparatus to:

determine a source address of a service;

transmit a subscription message addressed to the source address;

receive client code for execution at the apparatus; and
execute the client code to generate a local interface for accessing the service through a graphlet, wherein the graphlet defines one or more routes through nodes in a network to which transmissions associated with the service are restricted.

34. The one or more computer readable media of claim 33, wherein the source address of the service includes an address of a publisher of the service.

35. The one or more computer readable media of claim 33, wherein the source address includes an address of a proxy node different from a publisher of the service.

36. One or more computer readable media storing computer readable instructions that, when executed, cause an apparatus to:

receive, at a node in a network, a communication from a client, the communication addressed to a publisher of a first service; and

transmit the communication using a graphlet specific to the first service, wherein the graphlet defines one or more routes through nodes in the network to which transmissions associated with the first service are restricted.

37. The one or more computer readable media of claim 36, wherein the graphlet includes less than all nodes in the network.

38. The one or more computer readable media of claim 36, wherein the computer readable instructions, when executed, further cause the apparatus to:

receive code from the publisher of the first service; and

execute the code to update at least one service protocol associated with the first service.

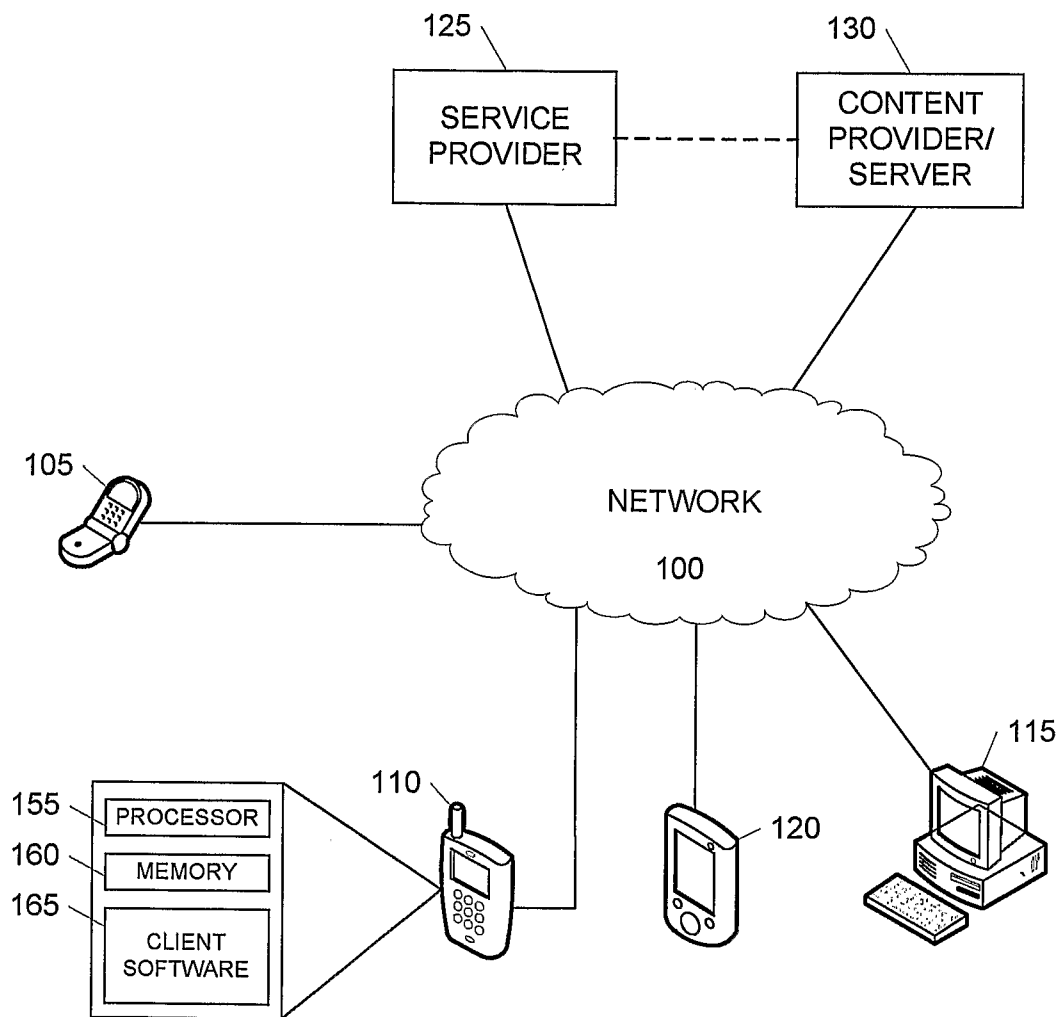


FIG. 1

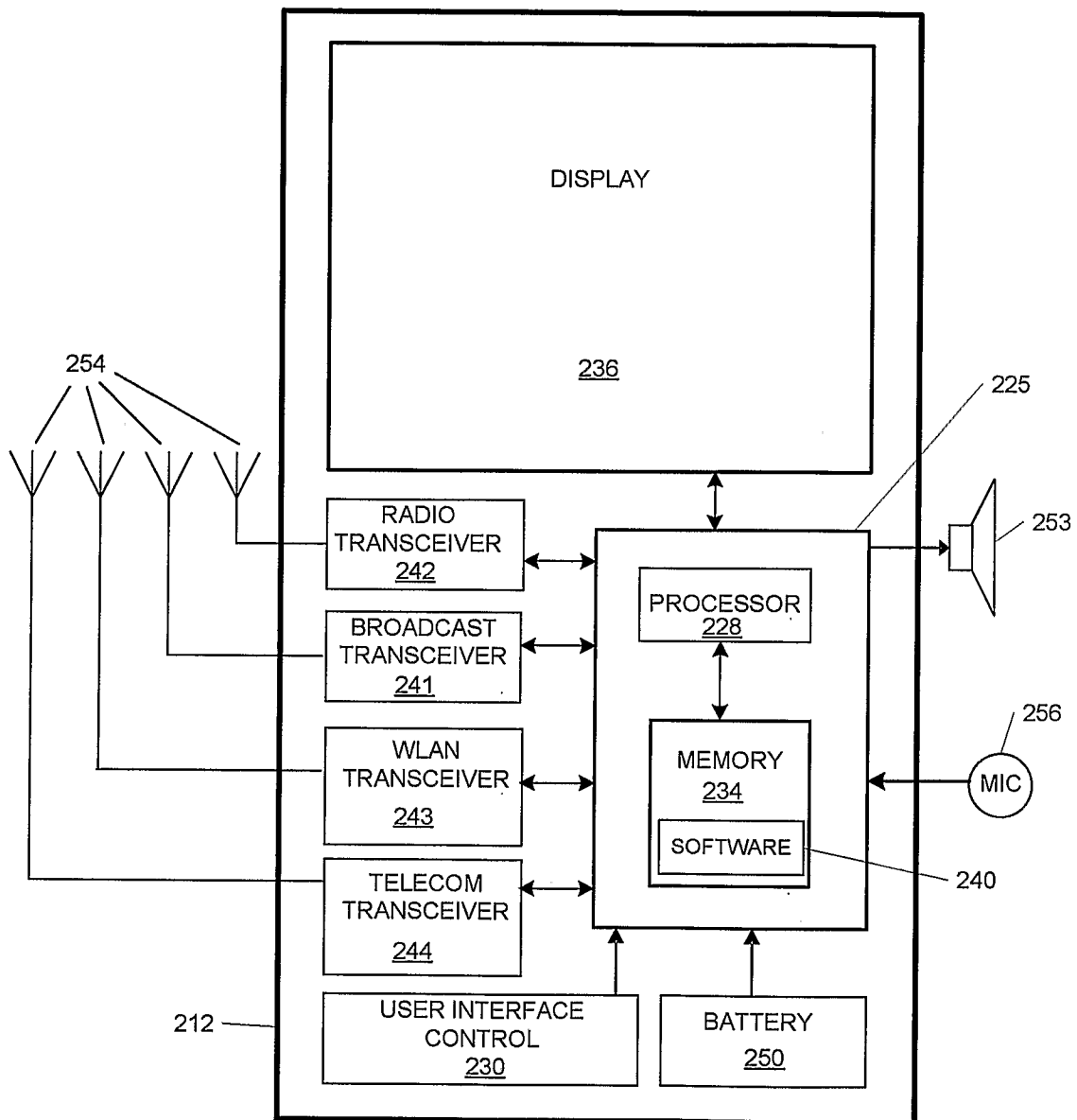


FIG. 2

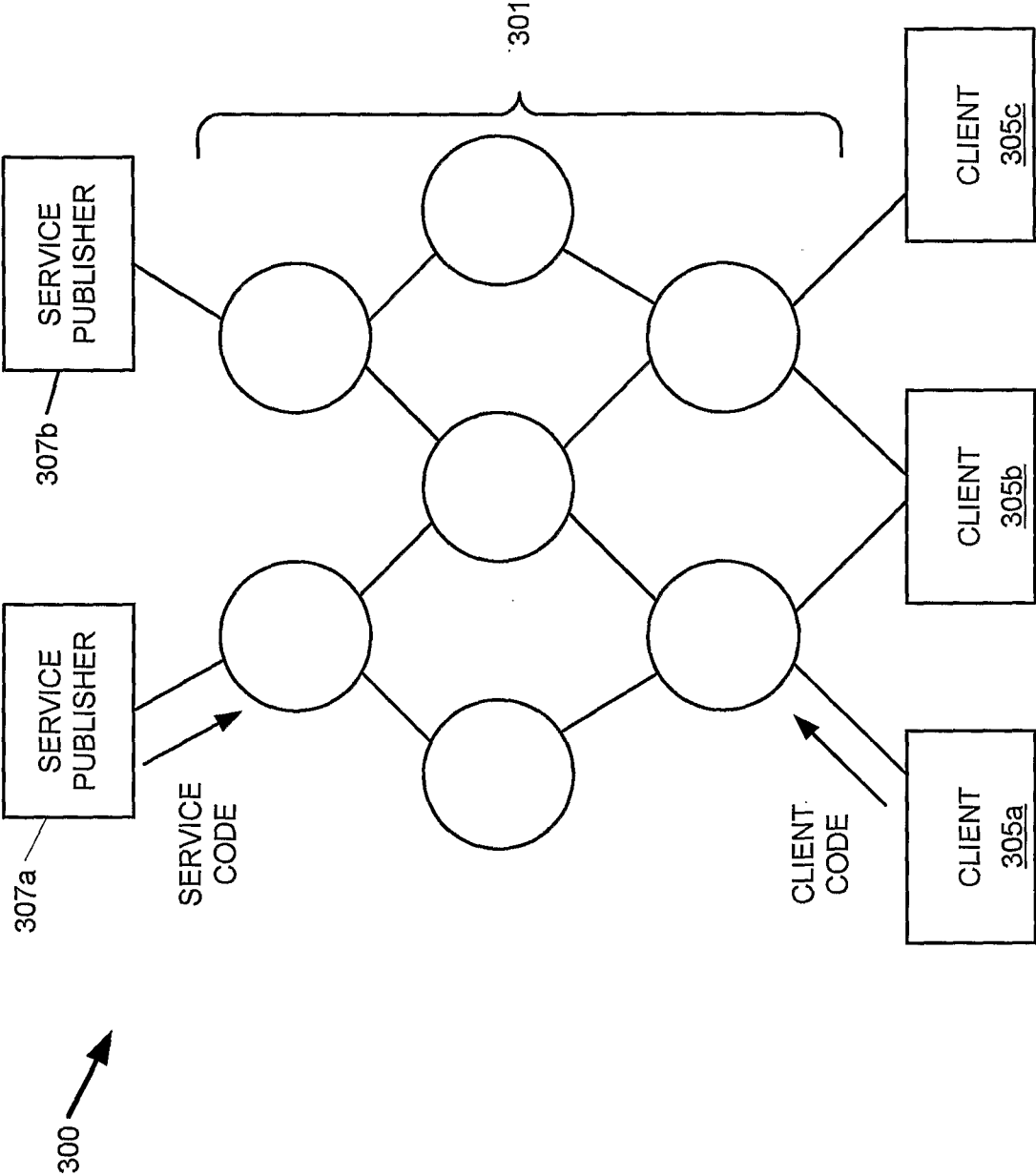


FIG. 3

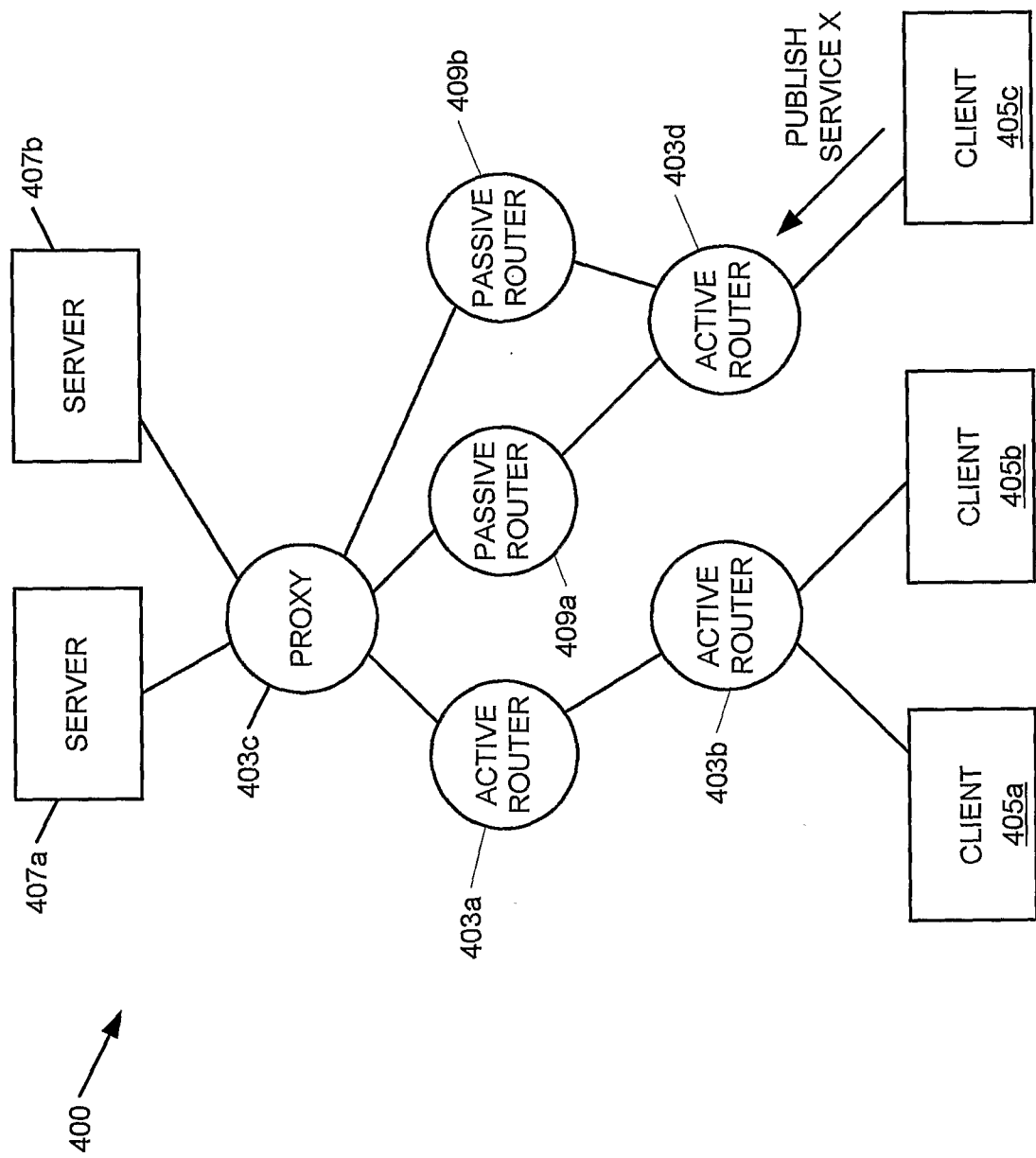


FIG. 4

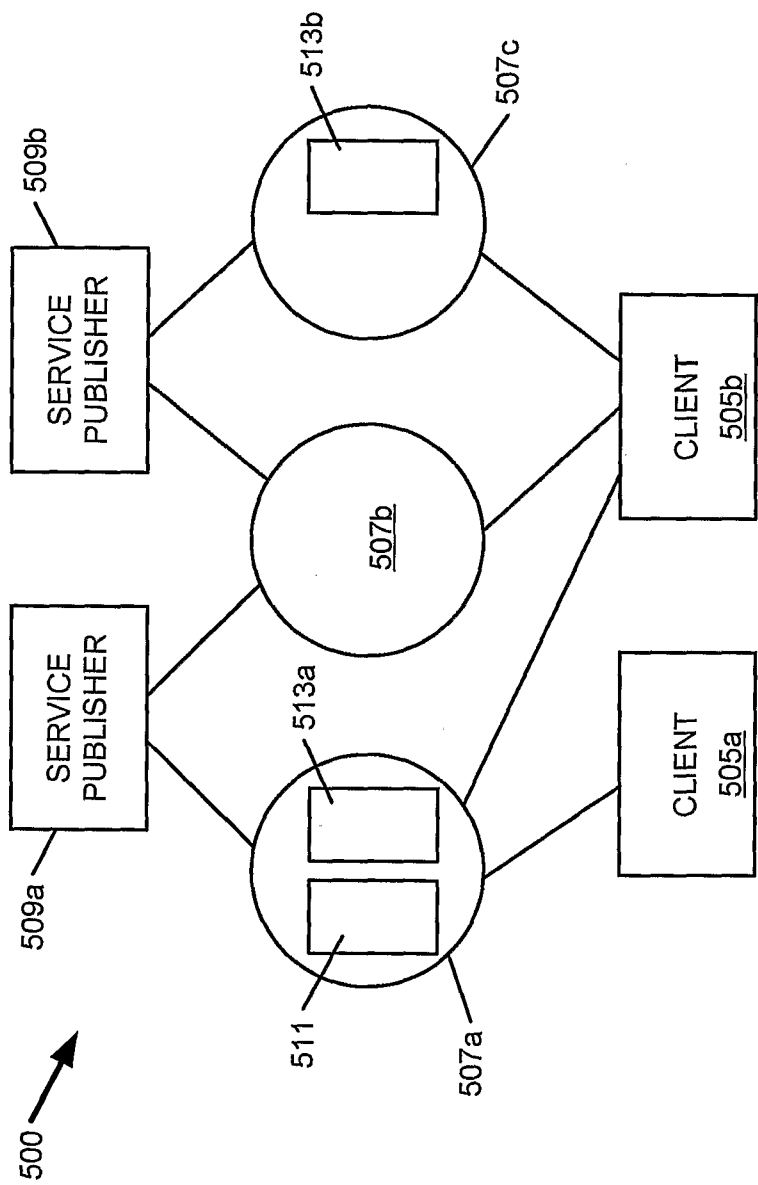


FIG. 5

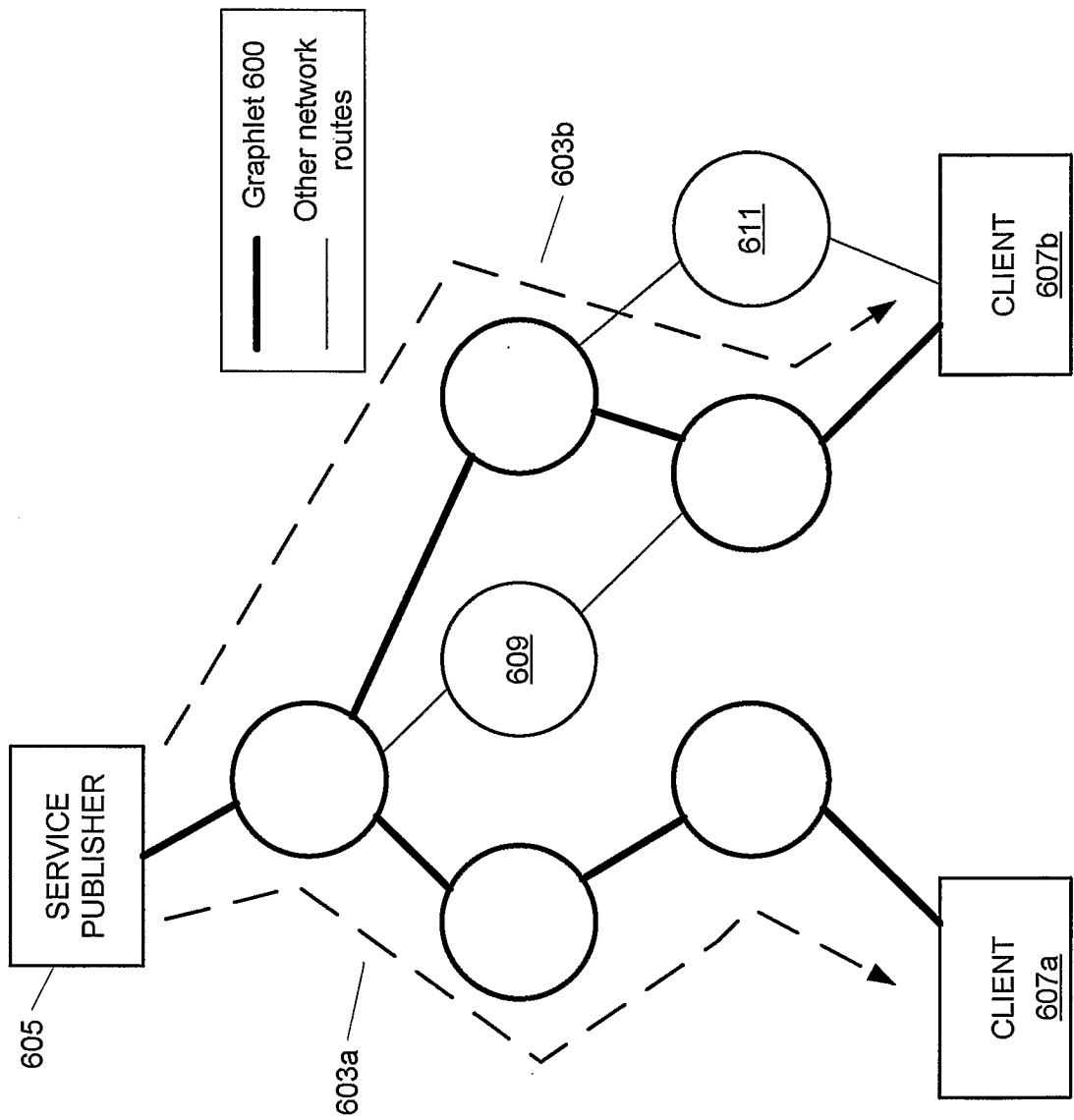


FIG. 6

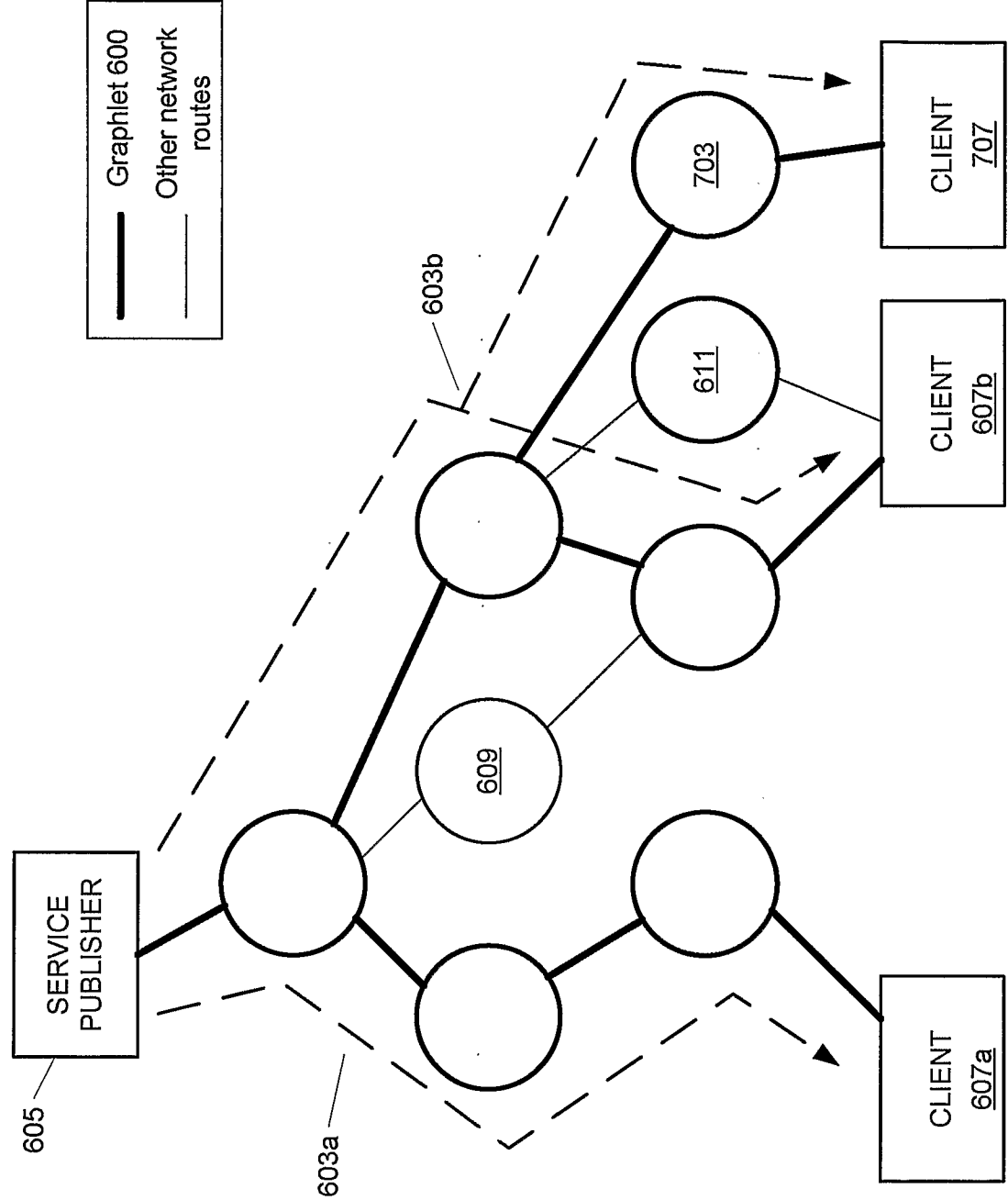
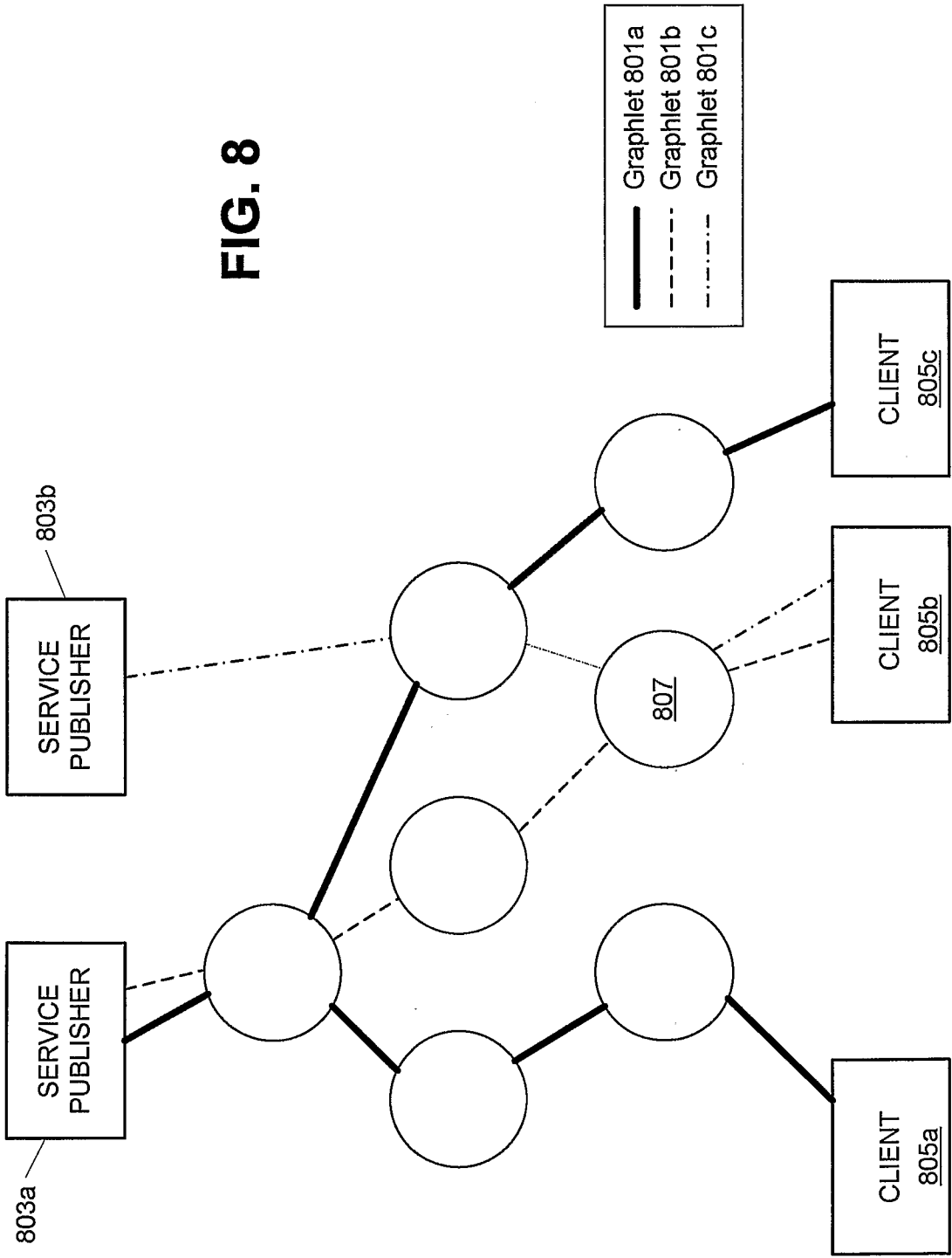


Fig. 7



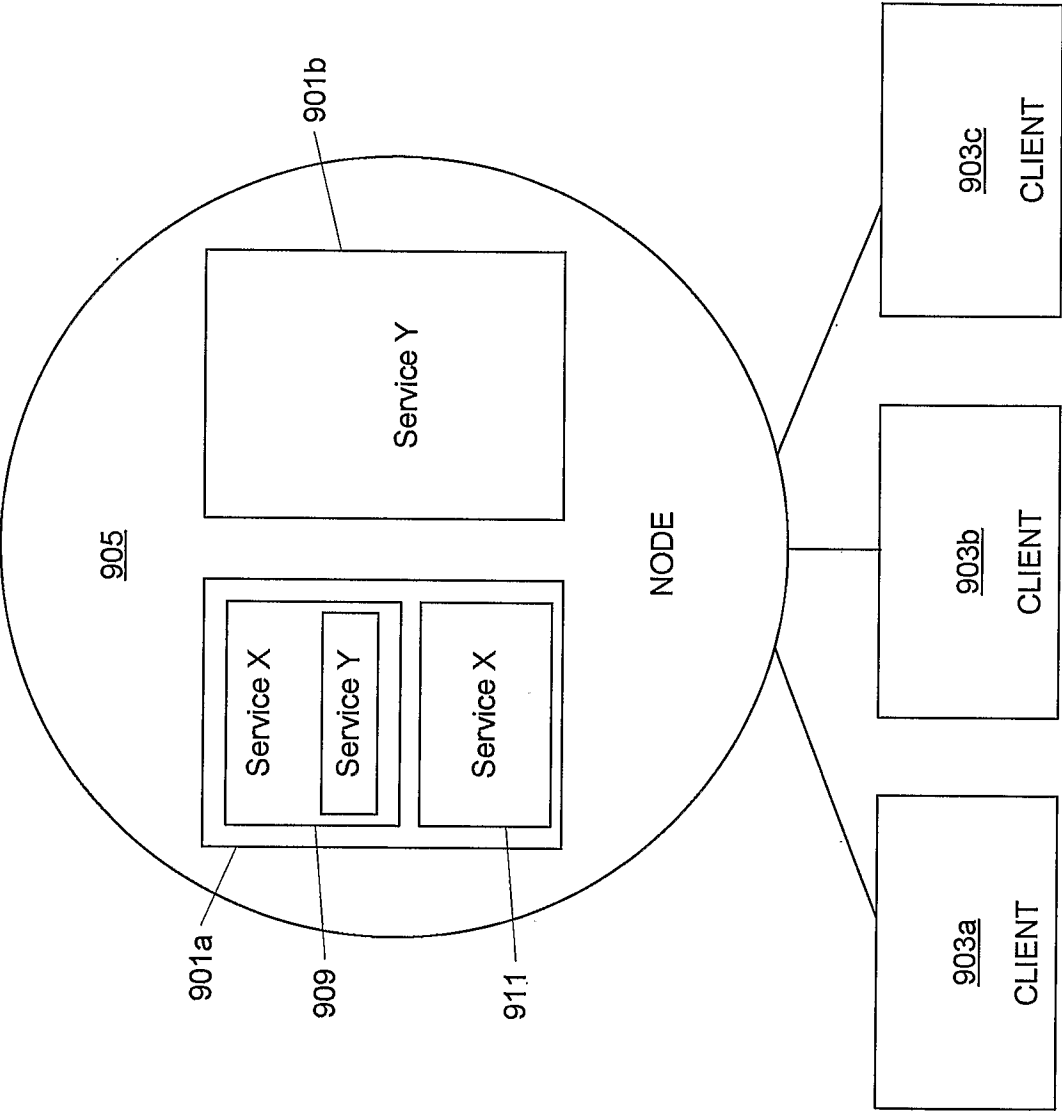


FIG. 9

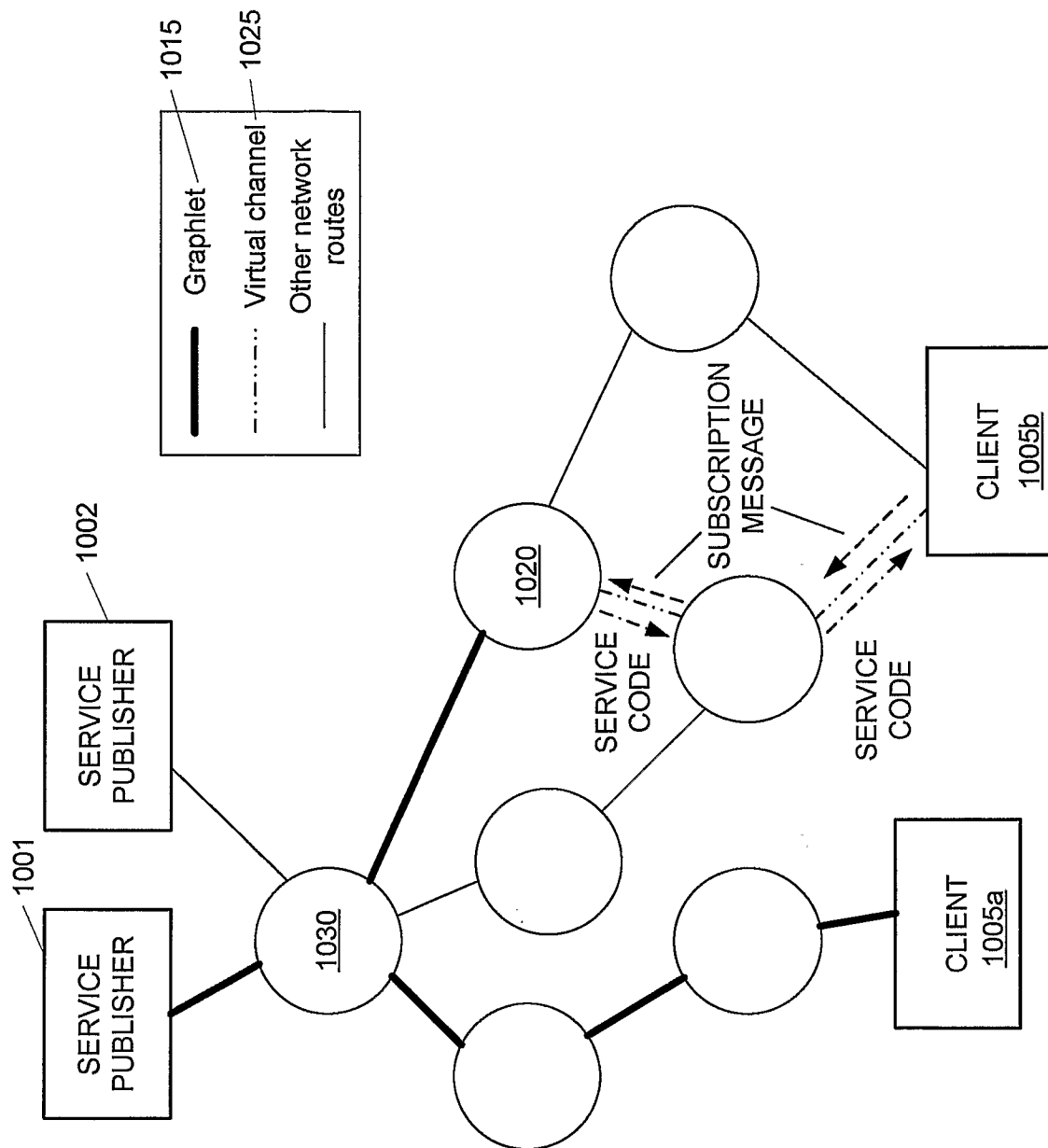
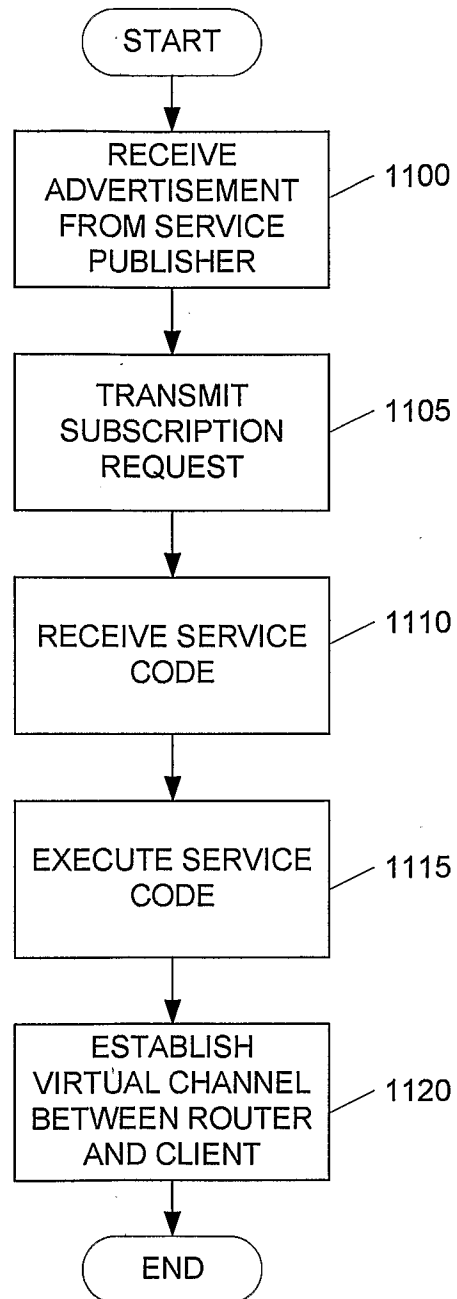
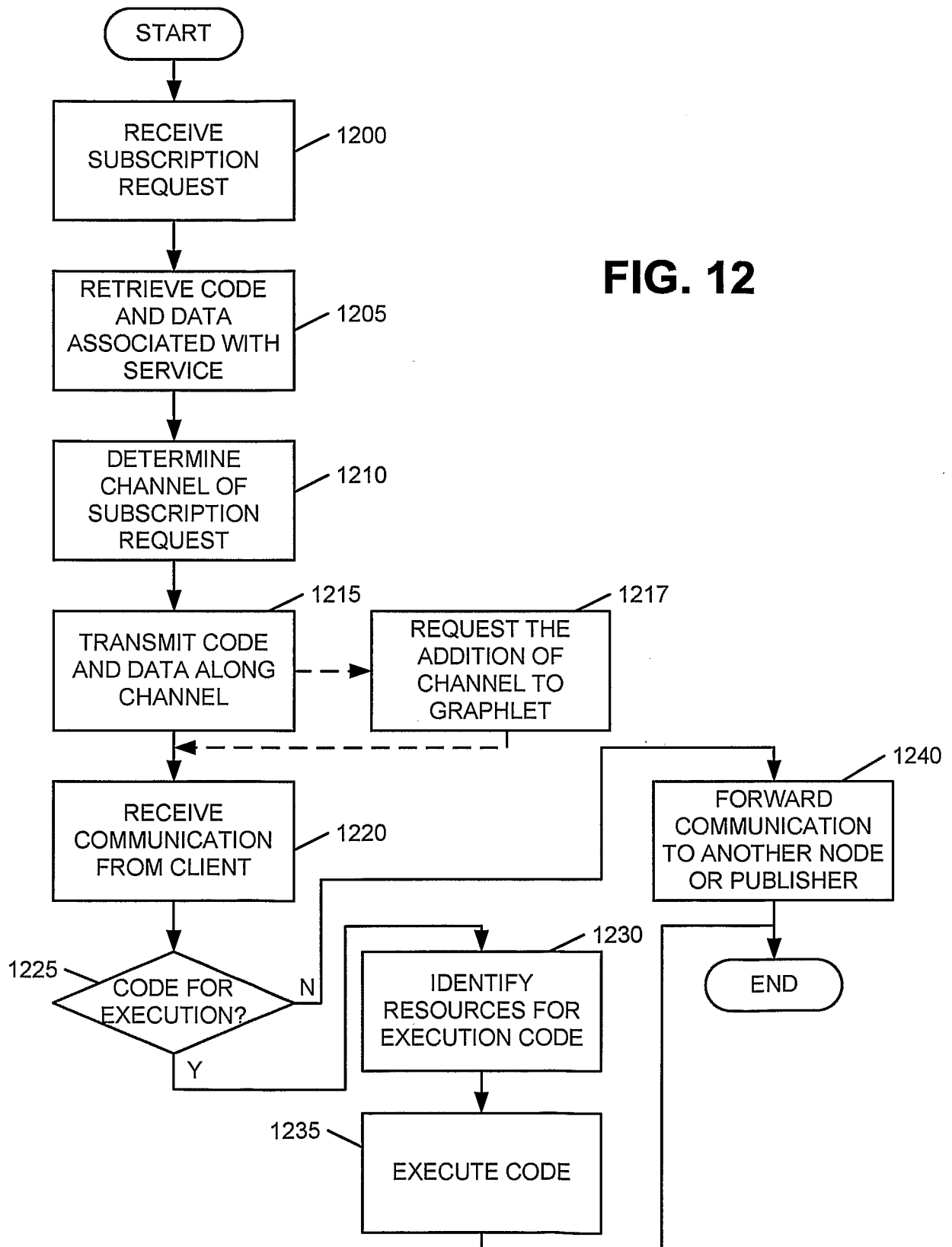
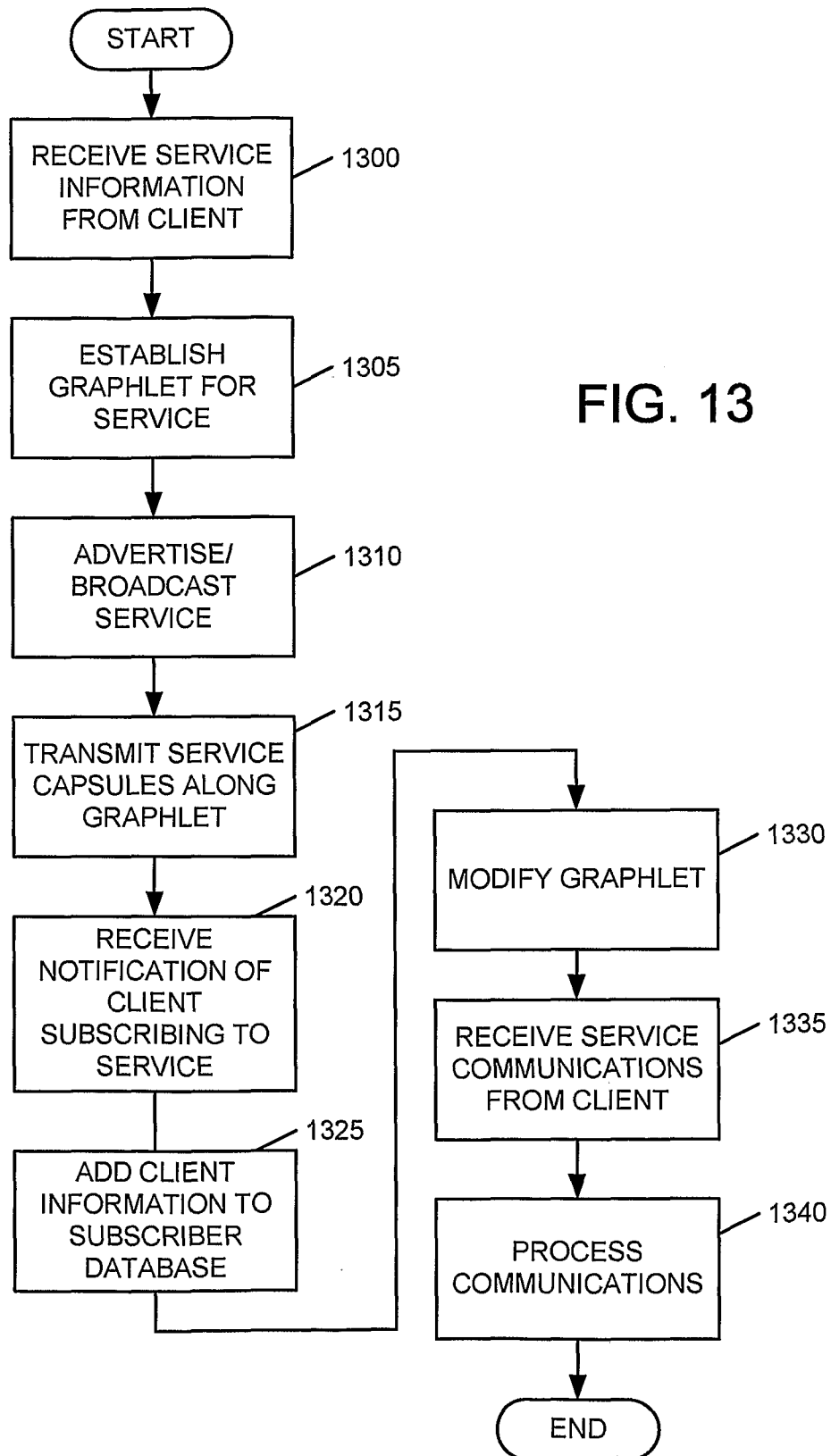


FIG. 10

**FIG. 11**





INTERNATIONAL SEARCH REPORT

International application No

PCT/IB2008/003186

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04L12/56

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)

H04L

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, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WETHERALL D J ET AL: "ANTS: a toolkit for building and dynamically deploying network protocols" OPEN ARCHITECTURES AND NETWORK PROGRAMMING, 1998 IEEE SAN FRANCISCO, CA, USA 3-4 APRIL 1998, NEW YORK, NY, USA, IEEE, US, 3 April 1998 (1998-04-03), pages 117-129, XP010272584 ISBN: 978-0-7803-4783-0 page 117, right-hand column, line 35 - page 122, left-hand column, line 14 page 124; figures 7-9 ----- -/--	1-38



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

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

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

& document member of the same patent family

Date of the actual completion of the international search

25 September 2009

Date of mailing of the international search report

08/10/2009

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040,
 Fax: (+31-70) 340-3016

Authorized officer

Lupia, Sergio

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2008/003186

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WETHERALL DAVID: "Active network vision and reality: lessons from a capsule-based system" PROCEEDINGS OF THE ACM SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, XX, XX, vol. 34, no. 5, 1 December 1999 (1999-12-01), pages 64-79, XP002197921 page 64, right-hand column, lines 41-43 -----	2,20,31
A	GOURHANT Y ET AL: "RETOUR D'EXPERIENCE SUR DIFFERENTES APPLICATIONS DE RESEAU ACTIF//APPLICATIONS BASED ON ACTIVE NETWORKS ASSESSMENT AND PERSPECTIVES" ANNALES DES TELECOMMUNICATIONS - ANNALS OF TELECOMMUNICATIONS, GET LAVOISIER, PARIS, FR, vol. 59, no. 5/06, 1 May 2004 (2004-05-01), pages 590-614, XP001217803 ISSN: 0003-4347 pages 12-13 -----	3,6,10,25,35
A	F. BELGHOUL AND O. MARCE: "Optimised Routing in Hybrid Active Networks" OPENARCH 2002, 5TH IEEE CONFERENCE ON OPEN ARCHITECTURES AND NETWORK PROGRAMMING, [Online] 28 June 2002 (2002-06-28), - 29 June 2002 (2002-06-29) XP002547441 NEW YORK, USA Retrieved from the Internet: URL:http://www.eurecom.fr/util/publi/download.en.htm?id=997 [retrieved on 2009-09-23] Chapters 1-3, 5,7 -----	1,2,8,14,19,20,23,27,30,31,33,36
A	EP 1 408 660 A (CIT ALCATEL [FR]) 14 April 2004 (2004-04-14) paragraphs [0005] - [0023] -----	2,5,20,22,31

Information on patent family members

PCT/IB2008/003186

Form PCT/ISA/210 (patent family annex) (April 2005)