

**ENHANCED VIDEO PROGRAMMING SYSTEM AND
METHOD PROVIDING A DISTRIBUTED COMMUNITY NETWORK**

5 The present invention relates to a method and apparatus for routing application layer packets of information in a network.

10 Computers have the capability to provide massive amounts of educational and entertainment information by way of the Internet. Currently, on-line systems offer a variety of different services to users, including news feeds, electronic
15 databases (either searchable by user directly on the on-line system, or downloadable to the user's own computer), private message services, electronic newsletters, real time games for play by several users at the same time, and job placement services, to name a few. However, currently most on-line communications occur merely through text. This is in contrast to the
20 audio/visual presentation of the alternative electronic medium, television. However, it is expected that as multi-media's incessant growth continues, audio/visual programs will proliferate and text will become less and less dominant in the on-line environment.

25 Even though these programs will be introduced, the Internet will remain essentially user unfriendly due to its very massiveness, organization, and randomness. Simply stated, there is no order or direction in the Internet. Specific pieces of information can be hard to find, and it is even harder to put that piece of information into a meaningful context.

30 Television, on the other hand, has been criticized for being a passive medium. Whilst interactive television systems have increased the level of user interaction, and thus, provided greater learning and entertainment opportunities, vast information resources such as databases are inaccessible from such a medium.

 The present invention seeks to close the gap between video programming and the vast information resources of the Internet.

According to a first aspect of the present invention there is provided a method for routing application layer packets of information in a network comprising:

5 receiving a packet which includes an indication of hierarchical routing information from a network service; and

 routing the packet to targets in a network based upon the hierarchical routing information.

10

 A method of an embodiment of the invention enables dynamic reconfiguration of a network for transmitting content, such as that located using URLs. The network, referred to as a distributed community network, preferably includes hubs which may logically reside on any machine and control the routing
15 of packets containing content by using hubs to assist in routing packets, the burden of routing control at corresponding server is decreased, enhancing the reliability and efficiency of the network in transmitting content and permitting access to content.

20 The present invention also extends to a method for routing application layer packets of information in a network, comprising:

 receiving a packet containing identification of a network service and content;

 routing the packet to the network service;

25 receiving from the network service the packet including hierarchical routing information; and

 routing the packet to targets based upon the hierarchical routing information.

30 According to a further aspect of the present invention there is provided apparatus for routing application layer packets of information in a network, comprising:

 receiving means for receiving a packet including an indication of hierarchical routing information from a network service;

35 determination means for determining targets in a network for receiving the packet based upon the hierarchical routing information; and

transmission means for transmitting the packet to the targets based upon the determining.

The invention also extends to apparatus for routing application layer
5 packets of information in a network comprising:

first receiving means for receiving a packet containing identification of a network service and content;

first routing means for routing the packet to the network service;

10 second receiving means for receiving from the network service the packet with hierarchical routing information; and

second routing means for routing the packet to targets based upon the hierarchical routing information.

Embodiments of the present invention will hereinafter be described, by
15 way of example, with reference to the accompanying drawings, in which;

Figure 1 is a diagram showing the receipt and decoding of video signals at a subscriber location using a method of the invention;

Figure 2 is a diagram showing an alternative embodiment to achieve the integration of Internet information with video content;

20 Figure 3 is a flow diagram of the basic software of the invention;

Figure 4 is a diagram showing an embodiment in which URLs are directly transmitted to a user;

Figure 5 shows an embodiment of a system comprising a digital cable
box;

25 Figure 6 shows an embodiment of a system including a digital T.V.;

Figure 7 shows an example of a user interface;

Figure 8 shows an example of a display providing a user interface;

Figure 9 is a diagram showing an embodiment of a system having distributed communication servers;

30 Figure 10A shows an example of a physical network configuration for a distributed community network;

Figure 10B shows an example of a physical network configuration for a particular application of a distributed community network;

35 Figure 11A shows an exemplary logical structure for a distributed community network;

Figure 11B shows an example of a physical network configuration for a distributed community network illustrating an alternative source for a video signal;

Figure 12A illustrates an example of a content push packet flow in a distributed community network;

Figure 12B shows an exemplary push/pull packet;

Figure 13 illustrates chat packet flow in a distributed community network;

Figure 14A shows an implementation of a distributed community network as a small eSchool;

Figure 14B shows an implementation of a distributed community network as a large eSchool;

Figure 14C illustrates an implementation of a distributed community network as a simple network providing video and other content; and

Figure 14D shows a distributed community network as a complex network providing video and other content.

Figure 1 illustrates an embodiment of a computer based system for receiving a video program along with embedded uniform resource locators (URLs) which direct a user's computer 16 to address locations, or web sites, on the Internet 20 to retrieve related web pages. The web pages correspond to the video presentation. The particular video programming can be delivered in analog, digital or digitally compressed formats (e.g. MPEG2) via any transmission means, including satellite, cable, wire, television broadcast or sent via the web.

The video programming is preferably created at a centralized location, for example, as content creation 4 indicated in Figure 1, for distribution to subscribers. Program creation may be accomplished by any appropriate means. After a video program is created, uniform resource locators (URLs) are embedded. In one embodiment, the URLs are embedded into the vertical blanking interval of the video programming by a URL encoder 8, as shown in Figure 1. In this embodiment, the URLs are encoded onto eight fields of line 21 of the VBI. Line 21 is the line associated with close captioning, among other things. However, the URLs may additionally and/or alternatively be embedded in other fields of the VBI, in the horizontal portion of the video, as part of the

audio channel, in any subcarrier to the video, or if the video is digital, in one of the data fields.

Although Figure 1 shows the video with the URLs broadcast over the same transmission line, the URLs may be sent down independently of the video program on a data channel. In this embodiment, the URLs may be forwarded to the remote sites either prior to initiation or during the program. Preferably, the URLs have associated time stamps which indicate to the subscriber stations when, during the video program, to display the particular web pages addressed by the URLs. Alternatively, the user can select when to call the particular web pages for display with the video program.

The particular information in line 21 is not part of the visual part of the program, and thus, is not perceptible to the human eye, thereby making it ideal to send data information to the users. Whilst the bandwidth capacity of line 21 is limited, as a system as described transmits only the URLs, and not full web pages, there is more than enough capacity. Furthermore, no additional hardware is necessary at the computer 16 to receive the video and retrieve the web pages.

20

Once the video program is created, it may be transmitted to user sites over any transmission means, including broadcast, cable, satellite, or Internet, and may reside on video servers. Furthermore, the video program, with or without embedded URLs, may be encoded onto storage means such as a video tape, for example of VHS or Beta format, or an optical disc such as CD or DVD, or any other medium.

25

Preferably, each receiver station comprises any Intel x86 machine (preferably a 486 processor, pentium processor, etc), an Apple Computer, UNIX or any other type of standard computer workstation. The local computer 16 is preferably connected to either a cable and/or broadcast television or to a local VCR or other video source. At each subscriber site, the local personal computer 16 preferably receives the cable transmission by cable connection on the back of the personal computer 16. The video/audio program may be processed for display on the computer screen using a PC card capable of displaying video signals on a computer monitor in an appropriate TV format such as PAL or

30

35

NTSC. One example of a PC card is a WinTV card. In addition to the cable connection, there is the Internet 20 connection created concurrently with the cable connection.

5 The Internet 20 connection may be via high-speed line, RF, conventional modem or by way of two-way cable carrying the video programming. The local PC 16 has Internet access via, for example, an ASCII software mechanism. In an embodiment, at each subscriber site, an associated local URL decoder 12 extracts the URLs, preferably embedded in the vertical blanking interval, with the
10 use of a suitable VBI decoder device. The URL decoder 12 may be either a stand-alone unit or a card which is implemented into the personal computer 16.

 In the embodiment shown in Figure 2, the uniform resource locators (URLs) are encoded into the video as described above. Again, the URLs are
15 preferably encoded onto eight fields of line 21 of the VBI, but may also be sent independently of the video. In this embodiment, a URL decoder 24 is located at the server site rather than at the subscriber location. When the decoder 24 receives the video program signal, it strips out the URL codes on line 21 of the VBI and delivers these codes independently to an Internet server 28. The URL
20 code is then subsequently delivered over the Internet 20 to the user PC 16. Simultaneously, the video is broadcast over conventional broadcast or cable transmission means 36 to the user's personal computer 16.

 The alternative shown in Figure 4, does not use the VBI. In this
25 embodiment, the system runs an online service over the Internet 20. This service is in the form of an Internet web site 62 which provides a user-interface to a database 78 and to one or more associated data servers 90. The service provides member accounts to TV broadcasters 66 who sign up to use the illustrated system in conjunction with their broadcasts. Each member
30 broadcaster will enter the service at their computer 70 through web browser software 74 using their member account by entering various identification and password information. Once within their account, the member will be provided with a graphical user interface for pre-scheduling URLs for transmission to users 118 over a direct Internet connection 94 at particular times of day. The same
35 user interface, or a variation of it, can be used by broadcasters for live transmission 82 of URLs to users at the same time as a broadcast 86.

One example of this interface might be a scheduling calendar (daily, weekly, monthly, yearly) in which the broadcaster 66 may allocate time periods which coincide with their broadcasts 86, and during which they will send out
 5 URLs to their users to link to web pages. For each time period (for example, a particular hour long period during the day) determined by the broadcaster 66 to be a broadcast period (a period during which they want to transmit URLs that correspond to a television show being broadcast from their TV broadcast facility 110 to the external TV 114 of the user 118 at that time), the broadcaster 66 may
 10 then enter a series of URLs into an associated file ("Link File") for transmission over the Internet 20 at that time. This Link File may have a user interface such as a spreadsheet, table, or list, or it may be simply a tab-delimited or paragraph-delimited text-file. As an example, each of the records in the Link File consists of a data structure which may contain information such as:

15 (<timecode>,<URL>,<label or title>, <additional information>,<additional information>,...)

20 The above data structure is just one example. The records in the Link File preferably specify the time, Internet address (i.e. URL), label (such as an associated name), and some optional additional information, for each web page the broadcaster 66 desires to launch during a show.

25 When a broadcaster 66 modifies their calendar and/or the Link File associated with any given time period(s) in their calendar, this information is saved into the database 78 which is attached to the site 62. Each broadcaster 66 may maintain multiple calendars in the database 78 if they broadcast in different time zones, for example.

30 The database 78 provides the Link File records for upcoming time periods to a server 90, which may be one server or a distributed network of server programs on multiple computers across the network, to be utilized for scaling to large national or global audiences. The server 90 provides the Link File records, including the URLs, to the user's personal computer 16, which is connected via
 35 a network. Examples of possible networks include the public Internet 94, a direct private network, or even a wireless network.

One feature of the embodiment illustrated in Figure 4 is that one or more
broadcasters 66 may utilize the same schedule in the database 78 for their own
broadcasts 86 or during the same broadcast. For example, a network
5 broadcaster may develop a master schedule and various affiliate broadcasters
may subscribe to that schedule or copy it (in the database) and add or delete
specific URLs in the schedule for their local audiences or unique programming.
This scheme enables affiliates to insert URLs for local advertisers or local
subjects into a sequence of more general URLs provided by their network
10 broadcaster 66. In other words, the affiliate can add links that ride on the
network feed and then redistribute it to their local audiences.

The system of Figure 4 also enables personalization in the form of unique
series of URLs specific to each user's unique profile, which are directly sent over
15 the Internet 20 to each user's specific client software 106. This can be achieved
from the broadcaster 66 to each individual user 118, or to particular collections
of users. To accomplish personalization, the service may send a different
stream of URLs to each user's client software program 106. The stream of
URLs sent depends upon a user profile stored in the database 78 or the client
20 software program 106, a user profile which is built on demand or over time for
each user 118 based on criteria such as the location of the user, choices the
user makes while using a client software program 106, choices the broadcaster
66 makes during a broadcast 86, or automatic choices made by an algorithm
(such as a filter) residing on the service 62. Personalization enables each user
25 to receive URLs which are uniquely relevant to their interests, demographics,
history, or behaviour in the system.

Once the URLs have reached the personal computer 16, the operation of
all of the systems shown in Figures 1, 2 and 4 is similar.

30

In one embodiment, a JAVA enabled browser 98 as well as specialized
software 106 are installed on the computer 16. The JAVA enabled browser 98
allows the computer 16 to retrieve the web pages 102 and is presently the
preferred software, as it is platform independent, and thus, enables efficient and
35 flexible transfer of programs, images, etc., over the Internet 20. The specialized
interface software 106 (hereinafter, "client software") acts as an interface

between the video programming and the Internet functions. The client software 106 retrieves URLs from the video program (embodiment of Figure 1) or directly from the Internet connection (embodiments of Figures 2 and 4), interprets these URLs and directs the JAVA enabled browser 98 to retrieve the particular
5 relevant web pages 102. The client software 106 also synchronizes web pages to the video content for display on the user's computer 16, as shown in Figures 3 and 4 and explained in more detail below.

As explained above, the URLs may be encoded and embedded into the
10 video signal by inserting them into the vertical blanking interval (VBI).

Alternatively, the URLs may be entered by member TV broadcasters 66 along with specified times for transmitting the URLs to the user. At the appropriate times, the URLs are sent directly over the Internet to the user's PC
15 16 via the client software 106 over a direct point-to-point or multicasting connection.

The system may have the capability to detect identical URLs sent directly after one another and to cause the browser not to fetch URLs in these particular
20 cases. As shown in Figure 3, once the URL code is received at the computer, the client software 106 first interprets the URL and determines in step 42 whether the particular URL has been received previously. If it has already been received, the next received URL is interpreted for determination of prior receipt. If the particular URL has not been detected before, the software checks for
25 misspelling in step 46 and any other errors, and if errors exist, corrects these particular errors. Once again, it is determined whether the URL has been previously detected. If it has, the next URL is accessed in step 38. If the URL has not been detected, the specific URL is added to the URL list in step 54. The specific URL is then sent to the web browser, preferably a JAVA enabled
30 browser 98. Upon receipt of the URL, the browser 98, in step 58, will access the web site address 122 (Figure 4) indicated by the URL and retrieve the cited web page(s) 102 via the Internet.

Viewers can view the integrated presentation in the following manner. As
35 mentioned above, the video signal is processed and displayed on a video

window on the PC screen using a WinTV card, for example. The corresponding audio is forwarded to the audio card and sent to the PC speakers.

5 The retrieved web pages 102, referenced by the URL, are optionally time stamped to be displayed on the computer screen when predetermined related video content is displayed in the video window, thus enhancing the video presentation by providing in-depth information related to the video content thereto. Another section on the screen is also preferably used to represent an operational control panel. This control panel provides a list of the URLs which
10 have been broadcast and correspondingly received by the computer 16. This control panel is updated to add a URL code each time a new URL code is received by the PC 16. This list gives the subscriber the flexibility to go back and retrieve particularly informative or interesting web pages that have already been displayed earlier in the program, or alternatively, to print them out for future
15 reference. Furthermore, the list may include URLs referring to web pages not displayed with the broadcast program, but which provide further information on a certain topic of interest to the viewer.

In an example, a viewer may begin watching a musical video featuring a
20 band. As the video is received by the PC 16, URLs are either being received with the video signal or are being received directly via the Internet 20 or another data channel, and are interpreted by the client software 106. Upon direction and command, the JAVA enabled browser 98 retrieves particular web pages 102 from Internet 20 web sites identified in the URLs. These web pages 102 are
25 then displayed on the video screen at particular times. So, for example, whilst the viewer is watching the music video, biographical information on the band may also be displayed adjacent to the video window. Web pages 102 may also include an upcoming concert schedule, and/or audio clips of the band's music may be downloaded from the Internet 20.

30

As another example, a user may be watching a program relating to financial news. Whilst the narrator is shown discussing high tech stocks, web pages corresponding to detailed financial performance information on high tech stocks, environment and characteristics may be displayed with the video on the
35 computer screen. If the personalization features are included, web pages associated with a particular user's stock may be fetched and displayed on the

computer screen with the video program. When the program narrator switches to a discussion on the weekly performance of the Dow Jones, web pages presenting related financial performance information may be simultaneously displayed.

5

A user may view the interactive program using a television set 114 or other display monitor in conjunction with the display screen of the personal computer 16. In this case, the relevant web pages are shown on the personal computer 16 whilst the video program is displayed on the television monitor 114. In this alternative, a cable set top box receives the television program from the multi-channel cable. The personal computer 16 also receives the video program from the multi-channel cable and extracts the URLs, embedded in the vertical blanking interval of the video signal or directly transmitted 94 over the Internet 20. The client software 106 extracts the URLs and retrieves the particular web pages as described above. The web pages are then synchronized with the particular video frames and presented to the user. It is understood that a hyperlink may exist on the web site that will allow the user to automatically load the client software and call up the specific television channel referenced in the web site. For example, someone browsing the Internet 20 may come upon a major television network's web site. It is possible then to scroll to an interesting story and then to click on an hyperlink to turn on the software which tunes the TV window to the network.

Instead of receiving the video program from a transmission means, the video program may be addressed directly from the user site if the video program, with or without embedded URLs, has been stored on appropriate means. The storage means may be a videotape in any format, such as VHS or Beta, or an optical disc in any format, such as DVD or CD-ROM. In this case, the user PC 16 and/or television 114 are connected to a video tape player, a disc drive, or other appropriate device.

Figures 5 and 6 show two alternative examples of systems which may be employed. As shown in Figure 5, a user may view an interactive program using a television set 18 or other display monitor in conjunction with a digital cable box 140. In this case, the digital cable box 140 performs the functions of the personal computer 16 shown in Figures 1, 2 and 4, and the client software is

stored in memory in the digital cable box 140. In one embodiment, the digital cable box 140 includes two tuners, thus allowing both the web page and the video program to be simultaneously viewed on the same screen. If video and web stream, however, are carried on one channel, then only one tuner is
5 necessary.

The client software retrieves URLs from the received video program, directly from the Internet connection 20 or via a separate data channel, interprets these URLs and directs the web enabled browser to retrieve the
10 particular relevant web pages, and synchronizes the retrieved web pages to the video content for display on the television 18. The relevant web pages are preferably shown in one frame of the television 18 while the video program is displayed in another frame. Alternatively, the web page can replace the video program on the display.

15

In this embodiment, the digital cable set top box 140 receives the television program from the multi-channel cable. The URLs can be encoded into the digital program channel using MPEG1, MPEG2, MPEG4, MPEG7 or any other compression video scheme. Alternatively, the URLs can be transmitted to
20 the digital cable boxes 140 from an Internet server 148. The digital cable box 140 decodes the URLs from the digital video signal or directly transmitted over the Internet 20. The client software decodes the URLs and retrieves the particular web pages as described above. Preferably, the web pages are synchronized with the particular video frames and presented to the user.

25

As with all the embodiments described above, instead of receiving the video program from a transmission means, the video program may be addressed directly from a local video source 144 if the video program, with or without embedded URLs, is stored on a storage means such as a video tape or
30 optical disc. In this embodiment, the digital cable box 140 is connected to a VCR, disc drive or other appropriate device.

Figure 6 illustrates an embodiment where a digital TV 152 is the remote reception unit and performs the functions of the personal computer, shown in
35 Figures 1, 2 and 4, and the digital cable box 140 shown in Figure 5. A processor means and memory are incorporated in the digital TV 152, and the

client software and web browser software are implemented in memory in the digital TV 152. All of the functions described above with reference to the other embodiments are performed in a similar manner by the digital TV 152 embodiment.

5

Although the digital cable box/TV 140, 18 and digital TV 152, shown in Figures 5 and 6, are incorporated into the embodiment of Figure 1, in substitution for the PC 16, they may also be substituted for the PC 16 shown in Figures 2 and 4.

10

A user may view the video and web content on one screen (in two windows), or with the video on one display screen and the web content on a separate display monitor. Alternatively, a user may access the video or web content separately. Thus, a user may branch from video to web content and vice versa.

15

The systems described herein are well-suited to the education environment. Thus, students and teachers may access one or more web servers. Software components including instructor and student user software, authoring software and database assessment software are provided. An instructor may, for example, use content creation software on a personal computer to easily integrate into the curriculum current information published on the web through an interface 156 shown in Figure 7. The instructor creates a playlist (i.e. linkfile) 160, the playlist 160 comprising a list of web pages, text notes and questions. The web sites and questions are set out in a predetermined order and can be assigned times. Preferably, the URLs identifying the web site and time stamps are sent automatically to the desktop of each student in the virtual community, either during a playback of a pre-recorded program or during a live event.

20
25
30

At each of the student workstations, the program is directed by the playlist 160. In other words, the playlist 160 provides the structure for the program. At predetermined times as indicated by the playlist 160, the browser will fetch and display a web page in a frame on the computer screen. Because program events can be set up in this manner at predetermined times, the entire program

35

and playlist can be prerecorded and stored in a web database for later access by students.

5 It will be appreciated that the students and the instructor may be located anywhere, as long as they are all connected to the web. Because a server controls the program, the instructor output comes from the server and the student workstations are automatically updated by the web server.

10 This educational embodiment integrates web content and other media with collaborative groupware functionality to create an interactive environment for students and teachers. The student may receive a traditional video lesson through a frame in his or her web browser, or from a television. Separate frames may be simultaneously provided as shown in Figure 8, which shows the browser displaying: web pages 176 automatically delivered to each student's
15 desktop with information or exercises that complement the video presentation; a chat dialogue frame 168 for conversing with the instructor and/or other students online; and an interactive playlist 164 of web pages and questions comprising the lesson.

20 In the student interface of Figure 8, each student may perform a virtual experiment, for example, during a physics lesson to learn about gravity. In addition, the students may converse with one another and with the instructor using the chat dialogue frame 168. They may also send web pages to one another and provide answers to questions from the teacher via the chat dialogue
25 frame 168 of the student interface 176. With the chat feature, students may break into subgroups for collaborative learning. Whenever a student in the group sends a message, the message is sent to the Internet server 20 and every other student in the subgroup receives and views the message in their chat dialogue frame 168.

30

The instructor, however, may retain control over the chat feature. For example, the instructor may terminate the chat feature or web push to terminate unruly on-line conversations or the sending of web pages by students.

35

The systems described herein are more powerful than conventional distance learning systems as they allow the instructor to freely and conveniently

exercise almost any type of testing strategy. The instructor may test students using a combination of the chat dialogue feature and web pages. For example, multiple choice questions and short answer questions can appear in the chat window 168. Essay questions, requiring longer answers, become web pages.

5 As mentioned above, students can perform virtual experiments on-line. Once the instructor's personal computer receives student answers, student scoring may be presented to the instructor in any format including tables, charts, diagrams, bar graphs, etc. The instructor, thus, may analyze the results and has the capability of providing real-time feedback to the students.

10

Students may also receive individualized feedback via branched interactive audio, video and/or graphics responses. For example, the workstation may branch to a particular audio response, preferably prerecorded in the instructor's own voice, based on the student response to a multiple-choice
15 question. A plurality of potential audio responses may be made available at the student's workstation, for example, by a method as described in US patent No. 5,537,141. Additionally and/or alternatively, personalized video, audio and graphics segments may be delivered and displayed to the student based on a student answer or personal profile, for example, in a manner as described in US
20 patent No. 5,724,091.

Responses to student answers may be more substantive using a memory feature comprising an algorithm which selects an interactive response to the user based not only on the student's current answer selection, but also on the
25 student's previous responses. The algorithm, preferably stored in memory at each student's workstation and under processor control, selects an output interactive response based on student responses. In an example, a student who gets three or more answers in sequence right receives a more difficult question. However, a student who fails to correctly answer one or more of the
30 three questions receives an easier question.

The system illustrated in Figure 9 is capable of servicing large numbers of users, for example, several schools. As shown, communications servers 180 distribute and route message across a LAN, WAN and the Internet. At the heart
35 of the system is a group database server 184, and this is surrounded by several communication servers 180 which each serve an area 192. Each

communication server 180 is surrounded by squares representing user stations 188. The communication servers 180 are organized in node relationships with one another.

5 Each node is responsible for serving an area 192. An area 192 is defined as a virtual location serviced by a single communication server 180 (or "com server"). An area 192 may be a single school, an office, or may consist of several actual physical locations. The defining characteristic of an area 192 is that messages sent from one member of an area 192 to another need not be
10 routed outside of the servicing com server 180.

 An area member is analogous to the frequently used term "user". For example, a "user" may be a student in an educational environment.

15 The distributed communication system shown in Figure 9 permits the dynamic addition of communication servers 180 within a group with little or no administrative tasks as well as the addition of groups within an overall communications network. A communication server group consists of several defined virtual areas 192 (preferably, consisting of no more than 250 members
20 each), each area 192 serviced by a single com server 180. This system allows members of one area 192, or group, to easily communicate with members of another area 192 or group without any configuration changes.

25 In the past, service of very large numbers of users has required large expensive servers and networks. Furthermore, as the user base increased, performance suffered and the hardware had to be upgraded to service the demand.

30 The distributed communication system allows the same, relatively inexpensive, machines to serve an ever-increasing user base. This is accomplished by routing messages from one server to another when necessary following substantially the same core pattern as IP routing and DNS lookups. If a message is for a member not belonging to the current area 192 or group, the message is routed through the distributed communication system until its
35 destination, or someone who knows the destination and can deliver the

message, is found. The destination may be cached so subsequent messages for that member or group may be more efficiently delivered.

Referring to Figure 9, if a message is posted by member "A" and is
5 intended only for the members of group 1, the message never leaves the area 1
com server. However, if the message is intended for members of area 1 and for
members of area 2, the area 1 com server forwards the message to the group
database server 184. The message is broadcast to the members of area 1 and
tagged in the group database server 184 as belonging to area 2. The message
10 is then routed to area 2 and broadcast to area 2 members. With this technique,
any member may potentially send a message to any other member. If the area
com server 180 does not recognize the destination, the message is forwarded
up the line. Each com server 180 does not need to know about any other server
180. Messages are routed until they delivered. If undeliverable, the original
15 sender is notified.

New areas 192 can be added on the fly. When a new com server 180 is
added to the network, it registers itself with the database application.
Henceforth, any message destined for the new area 192 may be routed properly
20 without altering the other area servers 180.

This method and system works for global messages or for user to user
messages. Furthermore, new groups may also be dynamically added. Once
added, each new group database server 184 registers itself with the existing
25 database servers 184. This distribution of load permits nearly unlimited
expansion with existing software and hardware. Each server manages a finite
number of members, cumulatively serving a growing community.

Users need not be informed as to the particular com server 180 they
30 should connect to. Members are directed to a single URL. The selection of the
server for user connection is determined by load balancing software. In this
manner, the network may appear to be a global network of servers or simply a
local classroom.

The architecture described, which uses database servers as routing gateways, enables the system to serve with minimum administration and configuration and with lower end, cost-effective hardware.

5 A distributed community network provides services, including those discussed above, to an arbitrarily large community of end users by distributing the load among many machines, each providing specific parts of the community. All server functions may reside on one machine in a modest context, whereas in a network of potentially millions, the web of distribution may involve additional
10 server-side hardware or even actual client machines. An implementation of the distributed community network permits routing of content to be spread among multiple machines, which eases the processing burden on the server and provides for dynamic reconfiguration of the network. The dynamic reconfigurations may thus involve adding or removing machines to or from the
15 network due to, for example, new network users or machine failures. As users and machines are added to the network, for example, those machines may include software packet switching.

 Examples of services maintained by the distributed community network
20 include a chat service, whiteboard service, and content push/pull service. A chat service involves the ability of a particular group of users to interact, such as via chat frame 168 shown in Figure 8. The interaction of a particular group of users is referred to as a room, and the distributed community network may dynamically change room assignments to add or delete users from a room.
25 Users may be assigned to a particular room based upon their user profile. A whiteboard service involves the ability of users to receive frames of video information for network collaboration among the users. The whiteboard includes a frame of video information transmitted to users within the same network chat room. The users, depending upon their drawing privileges, may make
30 modifications to the frame, and those modifications are transmitted to the other users. In this manner, the users may collaboratively draw upon the frame.

 The term "push" refers to a configuration as shown in Figures 1 to 6 for automatic delivery of content to the user. The term "pull" refers to delivery of
35 content requiring user interaction. A content push/pull service thus permits users to request content by, for example, selecting or "clicking on" an icon or

URL in order to have content transmitted to their machine. In effect, the user "pulls" content to the user machine by requesting the content. The distributed community network may provide for other network-type services as well.

5 Control over these individual services resides centrally on a server providing the services. However, packet distribution for that service may be distributed over many server or client machines implementing logical entities referred to as hubs. A hub is implemented in software, for example, and it performs routing of packets. The hubs need not implement the specific services
10 for which they distribute packets; rather, they typically need only know the routing protocol for the service. Similarly, centrally controlled routers maintain routing logic for implementing network services, and, for example, dynamically update algorithms for optimizing room assignments for a chat service without disturbing routing implemented by the hubs.

15

 In addition, rooms assignments for a chat service are organized hierarchically to facilitate distribution to a targeted audience. The chat service may be controlled by the client as in the case of a simple chat service. It may also be controlled by a database and content push/pull service making use of
20 custom knowledge about a client via a profile for use in advertising purposes. In particular, the service may select advertisements to push to a particular client or user based upon the user's profile. Clients may also belong to multiple rooms in a chat service, making it simpler for a push/pull service to create distribution channels for similar clients by subscribing them to the new rooms. The push/pull
25 service may then send the same or similar content to all members of the new room.

 Figure 10A shows an example of a physical network configuration for implementing a distributed community network 200. The configuration includes
30 client machines 204,206,208 connected through a network 202 to a hub 220 within a server 210. Network 202 may represent, for example, the Internet, a wide-area network, a local area network, or an intranet. Network 202 may also comprise a cable TV distribution medium, broadcast medium, satellite broadcast, telephone lines, fiber optics, or any other appropriate transmission
35 medium. Each of the client machines includes access to particular services managed by corresponding servers. Client machine 204 provides donut, chat,

and whiteboard services to a client or user at that machine, for example. Client machine 206 provides donut and whiteboard services. Client machine 208 is a television. A client machine may also comprise a digital TV, a TV with a digital or analog cable box, or a computer connected with a TV.

5

A donut service refers to the ability to provide content to users based upon user-profile information. The donut services specifies a "donut" of dynamic, hierarchical, shared user-profile information. In particular, it maintains either user profiles or database keys into a data repository containing the profiles. The donut may be stored in a file-type structure on a computer-readable medium such as a memory and accessed by browser programs, associated web server programs, and other applications for use in routing content to the user associated with the donut.

15 Distributed community network 200 in this example includes three servers 210, 212 and 214. Server 210 includes hub 220 and a whiteboard service 216. Server 212 includes a hub 222 and a chat service 218. Server 214 includes a hub 224 connected to two exemplary services, a donut service 226 and a content push/pull service 228. Both services 226 and 228 are connected with a data repository 230. Examples of information for storage in data repository 230 include user profiles, content for transmission to users, and web pages.

Each of the three servers 210, 212, 214 are interconnected via their respective hubs 220, 222, and 224. In addition, this embodiment includes an authoring client machine 232 connected with hub 224 in the server 214. The exemplary services 216, 218, 226, and 228 are only examples of services provided by servers. Many more services may also be provided by the server.

For the services, the servers may provide many types of content such as, for example, video, audio, and multimedia. Web browsers for communicating with the server may be located in various places such as, for example, on a user machine, in a cable head end, in a satellite operations center, or in a set-top box. The web browsers may obtain the content in realtime, or it may be prefetched and cached either locally or on the server. In addition to web browsers, other entities may obtain content.

35

A distributed community network permits processing for these exemplary services to be distributed among multiple machines, such as servers 210, 212, and 214. The routing for providing the services to client machines 204, 206, and 208 are distributed among the multiple hubs of the servers. Therefore, in order to transmit content to the client machines, the hubs in the servers provide route content from an authoring client machine 232 through network 202 to client machines 204, 206, and 208.

Use of a distributed community network distributes processing among multiple machines in order to ease the burden of providing routing to multiple client machines. It also provides a dynamic reconfigurable network by shifting routing among multiple hubs so that additional client machines may be dynamically added to the network. Each of the client machines 204, 206, and 232 may be implemented with a conventional computer with communication capability for interacting through a network. Each of the servers 210, 212, and 214 may be implemented with a typical server machine.

Figure 10B shows the use of the network of Figure 10A for a particular application. For instance, during a professional football game, a user, Bob, first switches his cable set-top box to the appropriate video channel for the game. The video originates at a television operations center, is transmitted via satellite to a cable head end, then through a cable plant to Bob's set-top box. Bob also connects to the Internet 202 on his computer and accesses the matching HyperTV web page, involving a particular network service. This web page is located on a server in the point of presence for the service. The point of presence configures Bob's machine 204 into distributed community network 200. Machine 204 opens a persistent socket on configured hub A (220) and sends a subscribe message to configured chat service 218 via hub C (222). In response to the subscribe message, the chat service 218 sends an announcement packet to all the members of the configured room via hub A (220) and the Internet 202. In addition, the chat service 218 subscribes Bob to the appropriate push/pull service 228 via hubs C (222) and B (224).

Push/pull service 228 extracts Bob's user profile donut from database 230 and pushes the values of the profile down through hub B (224) and hub A (220) through the Internet 202 to Bob's machine 204. Since Bob's donut shows that

he has previously indicated that he is a fan of the "Giants" football team, Bob is subscribed to a particular push room for Giant fans by push/pull service 228. As the game progresses, any pushed content for Giants fans and any chat messages from Bob's friends in the chat room are transmitted to him at machine
 5 204 as well.

During the game, a content producer uses the authoring client 232 to send a web page detailing the biography of a Giants' player, the quarterback, and supporting links to hub B (224). The pushed information is routed to
 10 push/pull service 228, which sends a message to Bob and all the other members of that push room by routing it first to hub B (224). Hub B (224) in turn routes it to hub A (220), which distributes it via Internet 202 to Bob's machine 204 and all his friends in the chat room. Hub B (224) might also have routed the message to other hubs that support other people in the same chat room.

15 Bob reviews the page on the Giants and decides he wants to play a prediction game about the quarterback's decisions from one of the displayed supporting links. Bob "clicks on" the link to select it and generate a pull request, which is transmitted through the persistent socket connection through the
 20 Internet 202. The request is routed via hub B (224) to push/pull service 228, which pulls the distributed object (a JAVA game) from database 230 and returns it via the same channel to Bob's machine 204.

As the game progresses, Bob makes predictions about the quarterback's
 25 behaviour. After each play, data updating the game arrives from authoring client 232 into hub B (224), and is then routed to QB game service 217 via hub A (220) and hub B (224). The game service processes the authoring client's data and sends the resultant data to Bob via hub A (220) and the Internet 202. The data is presented by the distributed object JAVA game on Bob's client machine
 30 204.

The quarterback fumbles. The producer at authoring client 232 pushes a dynamic HyperText Markup Language (DHTML) instruction in Javascript through hub B (224) and hub A (220) through the Internet 202 to Bob's machine 204. At
 35 Bob's machine, the DHTML is executed and Bob receives on his computer screen an animated image of the word Fumble!

Figure 11A shows an example of a logic structure 240 for the distributed community network 200 shown in Figure 10A. Logic structure 240 illustrates logical relationships amongst entities within the physical distributed community network 200. In particular, an arrow represents a class/subclass relationship, and a diamond represents a relationship identifying an attribute of an entity. A client 244 may correspond with one of the client machines 204, 206, and 208 and it subscribes to a hub 248, which may correspond with hub 220, 222, or 224 within one of the servers 210, 212, and 214. A server 246 may correspond with one of the servers 210, 212, and 214, and it manages a service 250, which may correspond with one of the services 216, 218, 226, or 228.

Client 244 is a type of participant 243, which is a client machine subscribed to a particular service such a service 250. The client's status as participant 242 identifies it as a member of a particular room for a chat service. A directory 254 through a connection 252 provides communication for locating users for participant 242. In particular, connection 252 is a network connection between two participants over which packets can be sent, and directory 254 provides instructions for identifying which participants are to receive particular content, where they are located on the network, and which content they are to receive.

Router 256 maintains map of a portion of the distributed community network for resolving routing questions from hubs, adding new participants to the network, and adding participants to services, which involves adding connections. In particular, hubs request routing decisions from router 256, and in response router 256 provides instructions to the hubs for their use in routing packets. For example, if hub A (220) failed, router 256 can issue instructions to the remaining hubs to avoid routing packets through hub A (220).

Directory 254 includes information used in routing information among client machines, and it provides information to hub 248 for use in routing information. The term directory is used to refer to a room for a chat service or any other service. Directory 254 is a hierarchical collection of a directory, terminated with a room and it has a subdir (set of directories). A room is a directory and identifies a group of clients subscribed to a service, such as a chat

service. A room has clients (set of participants) and hubs (set of hubs) where hubs is a subset of clients.

Packet 258 is a structured piece of information delivered from one client
5 to another. It may include any type of content for various services such as a
push/pull service or a chat service. For example, in the chat service client 244
creates a packet when the user enters information and sends it to a hub
associated with the client's machine. The hub sends that packet to all the
connections subscribed to that room. If the connections include another hub,
10 the process iterates, thus distributing the packet to the room over multiple hubs.
Packet 258 identifies hierarchical routing information including a source
(participant), dir (directory), target (participant), conn (connection), and a route
(set of participants).

15 The packet with the routing information may include a variety of types of
information as its payload or transmitted content. The packet may include data
to be displayed, for example, to a user in a particular frame on a display device.
The packet may also include one or more distributed executable objects, and
the objects may include data, executable code, or data in combination with
20 executable code. The executable code may include, for example, a JAVA
applet or any type of entity specifying executable code.

Hub 248 includes software for routing packets, and it may be used to
push an application or content to multiple users. Examples of such applications
25 or content include a whiteboard, a JAVA game, chat messages, text, and a file.
By distributing hubs among multiple machines, the burden on a server of routing
packets is decreased. Also, hub 248 includes a connection to a router 256 for
obtaining routing information, if necessary. Table 1 includes an example of hub
logic for use by hub 248 in routing packet 258; this logic may be implemented in
30 software or firmware modules for execution by a corresponding machine.

Table 1: Hub Logic

 on receive Packet p

```

5      // find my Directory dir
      if ! p.dir in my dirs
        my dirs[p.dir] = my routers.get(p.dir)
      d = my dirs[p.dir]

10     // handle targeted and broadcast separately
      if p.target is empty

          // send it to all the clients in the room
          for i in d.clients (recursive)
15             if i.conn != p.conn
                send p to i.conn

        else

20             // see if I have a direct connection to the target
            for i in d.clients (recursive)
                if i.id==p.target
                    send p to i.conn
                    done // it's delivered

25             // we don't have the client, so send it to
            // all the relevant hubs
            for i in d.hubs (recursive)

30                 // if it has routing info, route
                if i.id in p.route
                    send p to i.conn
                    done // it's en-route

35             // otherwise, send it to all my sub-hubs
            else if p.route is empty
  
```

```

    send p to i.conn
    // we have to keep going, since it
    // may not p.target may not be in i

```

```

5      if we got this far, we failed to deliver
      send <not found> to p.source via p.connection

```

Figure 11B is an exemplary diagram of a physical network configuration 259 for a distributed community network illustrating an alternative source for a video signal. Configuration 259 illustrates the distribution of video or television content using cable television signals. An operations center 261 generates a cable television signal having content for distribution and transmits the cable television signal over a satellite link 263 to one or more cable head ends 265. Cable head end 265 may include a web browser for interacting with a network such as the Internet 202. Cable head end 265 receives the cable television signal from satellite link 263 and distributes the signal over cable lines to client machines 267. Alternatively, the operations center may transmit over a satellite link to the client machines, foregoing the cable route. Client machines 267 may represent televisions or any machines capable of displaying cable television signals (such as a personal computer with a TV card or a module for processing TV signals for display). The client machines 267 may also include connection to a network such as the Internet 202 for implementing a distributed community network.

25

Figure 12A shows an example of content push packet flow 260 in a distributed community network 200 for implementing a content push/pull service. The content push packet flow involves transmission of content via packets through the logic structure 240 shown in Figure 11A as implemented in the physical distributed community network configuration 200 shown in Figures 10A and 10B. As described above, a system may provide to a user both video and other media content, such as any content available via a URI or a client-side script such as a JAVA script. The type of media content pushed to a user may depend upon the user's profile.

35

The media content may include, for example, video, audio, combined video and audio, or multimedia content. When transmitting different types of media content, such as both video and audio, the media content may be transmitted from the same or different sources. In addition, the content may be
5 transmitted from a wide variety of sources such as, for example, television, broadcast television, cable, satellite, local video, and local CD-ROM or digital versatile disk (DVD). The local content may be stored on the hard disk drive of a user's machine.

10 As shown in Figure 12A, a content push/pull packet flow 260 includes interaction of multiple machines transmitting the packets. The packets may originate from various sources such as a web browser on a client machine, a cable head end, or a server. Client machines 284, 276, and 268 may correspond with client machines shown in Figure 10A and client logic elements
15 shown in Figure 11A. Client 284 includes a browser program which provides frames for the following services: media 286, chat 290, and content 288. Client 276 likewise includes a browser program providing frames for media 280, chat 282, and content 278 services. Client 268 includes frames for media 270, chat 272, and content 274 services. Client 284, 276, and 268, as shown, typically
20 have the same services or access to the same services; alternatively, they may each implement different services. Also, they may implement different network services than those shown, such as a whiteboard service as identified above.

As shown in Figure 8, a chat service may include an associated chat
25 frame 168 on a user's machine; content may include web pages 176; and media frames may include transmitted video programs. Also, as explained in the embodiments described above, the implementation of a chat service may use a digital television, digital cable box, or personal computer. In addition, the program from the program sources, such as those identified above, and the
30 content from a network service, such as a chat service or a push/pull service, may be provided on the same or different physical machines. For example, both the program and the content may be provided on a television or on a personal computer, or they may be provided on separate physical machines such as providing the program on a television and providing the content for the network
35 service on an associated personal computer.

As shown in Figure 12A for the push/pull service flow, authoring client 262 includes an authoring application 264 for creating a packet 266. An authoring application includes any program for use in creating a packet containing or identifying particular content. A server 292 provides for routing of packet 266. In particular, it includes a hub 294 coupled to a content push/pull service 296 and a chat service 298. The routing of packets and their content for the exemplary content push/pull service is further explained in Table 2 illustrating packet structure throughout the flow.

10 Authoring client 262, using authoring application 264, creates packet P-0 (step 266) and transmits it to server 292 where it is received by hub 294 as packet P-1 (step 306). Hub 294 routes the packet to an individual target. In particular, content push/pull service 296 sends packet P-2 to an entire directory via hub 294 (step 308). Hub 294 routes packet P-3 to all members of the
15 directory (step 306), and each client receives the packet P-4 (steps 312,314,316). Hub 294 may include software for routing packets, as illustrated by the logic in Table 1. The packets identify information in the header used by the hub logic in routing the packets for the push/pull service.

20 Figure 12B shows an example of a push/pull packet content 271 for use in a distributed community network 260 as packet P-4. Packet 271 includes a primary URI 273 for use by client machines 262, 268, 276, and 284 in retrieving content to be pushed to the machines. Packet 271 may also include one or more URIs for use by the client machines in pulling content. For example,
25 packet 271 includes a URI 275 for a first pull item, a URI 277 for a second pull item, and a URI 279 for a third pull item. The pull items may include an element to be displayed to the user, such as an icon, such that when the user "clicks on" or selects the item, the corresponding client machine uses the URI associated with the displayed item to retrieve content and display the content to the user.
30 Packet 271 may include multiple URIs for pushed content, as well as the multiple URIs for pulled content.

Table 2

	P-0	P-1	P-2	P-3	P-4
source	2000	2000	2000	2000	2000
service	content push/pull	content push/pull	content push/pull	content push/pull	content push/pull
target	200	200	-	-	-
dir	5	5	5	5	5
action	PUSH	PUSH	PUSH	PUSH	PUSH
note	author 262 creates packet P-0 containing a URI to be displayed and URIs to pull information	hub 294 routes packet P-1 to individual target, service 296	service 296 sends packet P-2 to entire directory via hub 294	hub 294 routes packet P-3 to all members of the directory	each client 268, 276, 284 in the directory receives packet P-4, including the URI to be displayed and the URIs to pull information

Figure 13 shows an example of the chat packet flow 330 in a distributed community network 200. This diagram includes the same logic elements as are shown in Figure 12A with routing for flow of a chat packet using a chat service 298 and implementations include the various alternatives identified above with respect to Figure 12A.

The routing of packets and their content for the exemplary chat service is further explained in Table 3 illustrating packet structure throughout the flow. In this example, client 284 creates a new chat message as packet P-0 (step 332). The client may use an application associated with the chat service to enter information, such as typing in a text string using a keyboard, and send it in packet form. The application may include, for example, scripts associated with the chat frame on the user machine.

Packet P-0 from client machine 284 is directed to the server 292 where it is received as packet P-1 (step 334). Hub 294 routes the packet to an individual target as identified in the header information of the packet. In particular, chat

service 298, using hub 294, sends packet P-2 to the entire directory (step 336). Hub 294 routes packet P-3 to all members of the directory (step 338), in which case is received by each of the client machines in the directory as packet P-4 (steps 340, 342, 344). Hub 294 may include software for routing packets, as
 5 illustrated by the logic in Table 1.

Programs in each client machine 268, 276, 284 associated with the chat service, such as scripts associated with the chat frames in the client machine, receive packet P-4 and perform necessary processing to extract the content
 10 originally created by the user at machine 284 (packet P-0) and display that content in the chat frames 272, 282, 290.

Table 3

	P-0	P-1	P-2	P-3	P-4
source	3000	3000	3000	3000	3000
service	chat	chat	chat	chat	chat
target	201	201	-	-	-
dir	5	5	5	5	5
Action	SAY	SAY	SAY	SAY	SAY
note	client at machine 284 enters new chat message, packet P-0	hub 294 routes packet P-1 to individual target, chat service 298	chat service 298 sends packet P-2 to entire directory via hub 294	hub 294 routes packet P-3 to all members of the directory	each client 268, 276, 284 in the directory receives packet P-4

15 Figures 14A to 14D shows examples of implementations for distributed community network 200. Figure 14A is a diagram of a distributed community network 350 for a small eSchool. The network 350 includes, in this example, a server 352 providing service for all chat rooms and whiteboards and is coupled to client machines 354, 356, 358, and 360. Each client machine may therefore
 20 access chat room and whiteboard services as managed by the server. Server 352 may include a hub having software for routing information, as illustrated by the logic in Table 1. Server 352 may also include a connection with a network such as the Internet 202.

Figure 14B is a diagram of an exemplary implementation of a distributed community network 200 such as a large eSchool 362. Configuration 362 includes a router 364 connecting servers 366 and 368. Each server manages a separate chat room and may exchange information through router 364. Two clients 370 and 372 are connected with a server 366 for the first chat room. Two additional clients 374 and 376 are connected with server 368 for the second chat room. Servers 366 and 368 may include hubs having software for routing information, as illustrated by the logic in Table 1. Servers 366 and 368 may include a connection with a network such as the Internet 202.

10

Figure 14C shows an implementation of a distributed community network 200 as a simple configuration 378 providing video and other content. Configuration 378 includes a server 380 coupled to clients 382, 384, 386, and 388. Server 380 in this example includes two directories identified as teams 1 and 2. One directory includes clients 382 and 384, and the other includes clients 386 and 388. These directories identify rooms for participants of a chat service maintained by server 380. Server 380 also identifies the type of content to be transmitted to the clients in its directory by being pushed to the clients by a content push/pull service. In this example, a first advertisement is to be transmitted to clients 382 and 386, and a second advertisement is to be transmitted to clients 384 and 388. Therefore, server 380 via a directory structure may specifically identify the type of content to be transmitted to particular clients. Server 380 may include a hub having software for routing information, as illustrated by the logic in Table 1. Server 380 may also include a connection with a network such as the Internet 202.

25

Figure 14D shows an implementation of a distributed community network as a complex configuration 390 providing video and other content. Configuration 390 includes a server 392 and a server 394 both connected with a server 396. Server 396 is connected with hubs 398 and 400. Hub 398 is connected with clients 402 and 404, and hub 400 is connected with clients 406 and 408. Servers 392, 394, and 396 may include hubs having software for routing information, as illustrated by the logic in Table 1.

35

Configuration 390 also includes its own directory structure for the routing of information. Server 396 routes to team 1 or hub 398, and provides separate

advertisements for hubs 398 and 400. The teams identify rooms for participants of a chat service, and the advertisements refer to particular content to be pushed to users by a content push/pull service. The directory for server 394 includes a first advertisement to be transmitted to server 396 and a second
5 advertisement to be transmitted to server 396. The directory for hub 398 includes a first team for clients 402 and 404, and hub 400 and server 396. It further includes the first advertisement to be transmitted to client 402, and a second advertisement to be transmitted to client 404. The directory for hub 400 includes a first team for clients 406 and 408 and hub 398. It also includes a first
10 advertisement to be transmitted to client 406, and a second advertisement to be transmitted to client 408. Servers 392, 394, and 396 may also include a connection with a network such as the Internet 202.

It will be appreciated that modifications in, and variations of, the
15 embodiments described above may be made within the scope of the present invention as defined by the appended claims.

**CLAIMS**

1. A method for routing application layer packets of information in a network comprising:
 - 5 receiving a packet which includes an indication of hierarchical routing information from a network service; and
 - routing the packet to targets in a network based upon the hierarchical routing information.
- 10 2. A method as claimed in Claim 1, further comprising determining targets in the network for receiving the packet based upon the hierarchical routing information by receiving routing instructions for use in determining the targets.
- 15 3. A method as claimed in Claim 1 or Claim 2, wherein said packet includes content, and further comprising transmitting the packet to the target using a content push/pull service.
- 20 4. A method as claimed in Claim 3, wherein the transmitting step comprises transmitting the packet including a uniform resource identifier for use in pushing content to the user.
- 25 5. A method as claimed in Claim 3, wherein the transmitting step comprises transmitting the packet including a uniform resource identifier for selection by the user for pulling content.
6. A method as claimed in any preceding claim, further comprising determining targets in a wide-area network, a local area network, an intranet, or the Internet for receiving the packets.
- 30 7. A method as claimed in any preceding claim, wherein the received packet contains video, audio, or multimedia content.
8. A method as claimed in any preceding claim, further comprising requesting routing instructions for transmitting the packet.



9. A method as claimed in any preceding claim, further comprising transmitting a program to the targets.

5 10. A method as claimed in Claim 9, further comprising transmitting the program and the packet for simultaneous processing and display of the program and the content on a display monitor associated with the targets.

10 11. A method as claimed in Claim 9, further comprising transmitting the program and the packet for simultaneous processing and display of the program and the content on a television associated with the targets.

12. A method as claimed in Claim 9, further comprising:
transmitting the program for display of the program on a television
associated with the targets; and
15 transmitting the packet for display of the content on a display monitor associated with the targets.

20 13. A method as claimed in any preceding claim, further comprising accessing a directory in order to obtain routing instructions for the packet.

14. A method as claimed in Claim 13, wherein the received packet includes an indication of the directory for use in obtaining the routing instructions.

25 15. A method as claimed in any preceding claim, wherein the received packet includes an indication of an action associated with a service.

16. A method as claimed in any preceding claim, wherein the received packet contains executable code, data, or executable code with associated data.

30 17. A method as claimed in any preceding claim, wherein the received packet contains an object, the object specifying executable code, data, or executable code with associated data.

35 18. A method for routing application layer packets of information in a network, comprising:



receiving a packet containing identification of a network service and content;

routing the packet to the network service;

receiving from the network service the packet including hierarchical
5 routing information; and
routing the packet to targets based upon hierarchical routing information.

19. A method as claimed in Claim 18, wherein the packet is received with an
identification of a push/pull service; an address used to push content to the
10 targets, and an address for the client targets to pull content.

20. A method as claimed in Claim 18 or Claim 19, wherein the packet is
received with an identification of a chat service.

15 21. Apparatus for routing application layer packets of information in a
network, comprising:

receiving means for receiving a packet including an indication of
hierarchical routing information from a network service;

determination means for determining targets in a network for receiving the
20 packet based upon the hierarchical routing information; and

transmission means for transmitting the packet to the targets based upon
the determining.

22. Apparatus as claimed in Claim 21, wherein the determination means
25 comprises means for receiving routing instructions for use in determining the
targets.

23. Apparatus as claimed in Claim 21 or Claim 22, wherein the transmission
means comprises means for transmitting the packet including content using a
30 content push/pull service.

24. Apparatus as claimed in any of Claims 21 to 23, wherein the transmission
means comprises means for transmitting the packet including a uniform
resource identifier for use in pushing content to a user.



25. Apparatus as claimed in any of Claims 21 to 24, wherein the transmission means comprises means for transmitting the packet including a uniform resource identifier for selection by a user for pulling content.

5 26. Apparatus as claimed in any of Claims 21 to 25, wherein the determination means comprises means for determining the targets in a wide-area network, a local area network, an intranet, or the Internet.

10 27. Apparatus as claimed in any of Claims 21 to 26, wherein the receiving means comprises means for receiving the packet containing video, audio, or multimedia content.

28. Apparatus as claimed in any of Claims 21 to 27, further comprising means for requesting routing instructions for transmitting the packet.

15

29. Apparatus as claimed in any of Claims 21 to 28, further comprising a program source for transmitting a program to the targets.

20 30. Apparatus as claimed in Claim 29, further comprising means for transmitting the program and the packet for simultaneous processing and display of the program and the content on a display monitor associated with the targets.

25 31. Apparatus as claimed in Claim 29, further comprising means for transmitting the program and the packet for simultaneous processing and display of the program and the content on a television associated with the targets.

30 32. Apparatus as claimed in Claim 29, further comprising:
means for transmitting the program for display of the program on a television associated with the targets; and
means for transmitting the packet for display of the content on a display monitor associated with the targets.



33. Apparatus as claimed in any of Claims 21 to 32, further comprising means for accessing a directory in order to obtain routing instructions for the packet.

5 34. Apparatus as claimed in Claim 33, wherein said receiving means comprises means for receiving the packet including an indication of the directory for use in obtaining the routing instructions.

10 35. Apparatus as claimed in any of Claims 21 to 34, wherein the receiving means comprises means for receiving the packet including an indication of an action associated with the service.

15 36. Apparatus as claimed in any of Claims 21 to 35, wherein the receiving means comprises means for receiving the packet containing executable code, data, or executable code with associated data.

20 37. Apparatus as claimed in any of Claims 21 to 36, wherein the receiving means comprises means for receiving the packet containing an object, the object specifying executable code, data, or executable code with associated date.

38. Apparatus for routing application layer packets of information in a network comprising:
first receiving means for receiving a packet containing identification of a
25 network service and content;
first routing means for routing the packet to the network service;
second receiving means for receiving from the network service the packet with hierarchical routing information; and
second routing means for routing the packet to targets based upon the
30 hierarchical routing information.

39. Apparatus as claimed in Claim 38, wherein the first receiving means comprises means for receiving with the packet an identification of a push/pull service, an address used to push content to the targets, and an address for the
35 targets to pull content.



40. Apparatus as claimed in Claim 38 or Claim 39, wherein the first receiving means comprises means for receiving with the packet an identification of a chat service.

5 41. A system for routing application layer packets of information in a network, comprising:

a network;

a plurality of client machines connected to the network;

an authoring client machine; and

10 a server connected to the network and the authoring client machine,

wherein the server comprises:

receiving means for receiving from the authoring client machine a packet including an indication of hierarchical routing information and content using a particular network service;

15 determination means for determining targets among the plurality of client machines in the network for receiving the packet based upon the hierarchical routing information; and

transmission means for transmitting the packet to the targets based upon the determining.