US 20030051055A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2003/0051055 A1

Parrella, SR. et al. (43) Pub. Date: Mar. 13, 2003

(54) **SYSTEM AND METHOD FOR MODIFYING A DATA STREAM USING ELEMENT PARSING**

(75) Inventors: **Michael J. Parrella SR.**, Weston, CT (US); **John Lash**, Wilton, CT (US)

Correspondence Address:
Kim Kanzaki
COUDERT BROTHERS LLP
3rd Floor
600 Beach Street
San Francisco, CA 94109-1312 (US)

(73) Assignee: **NCT Group, Inc.**

(21) Appl. No.: **10/164,296**

(22) Filed: **Jun. 4, 2002**

Related U.S. Application Data

(60) Provisional application No. 60/295,721, filed on Jun. 4, 2001. Provisional application No. 60/295,672, filed on Jun. 4, 2001. Provisional application No. 60/295,676, filed on Jun. 4, 2001. Provisional application No. 60/295,720, filed on Jun. 4, 2001. Provisional application No. 60/295,671, filed on Jun. 4, 2001.

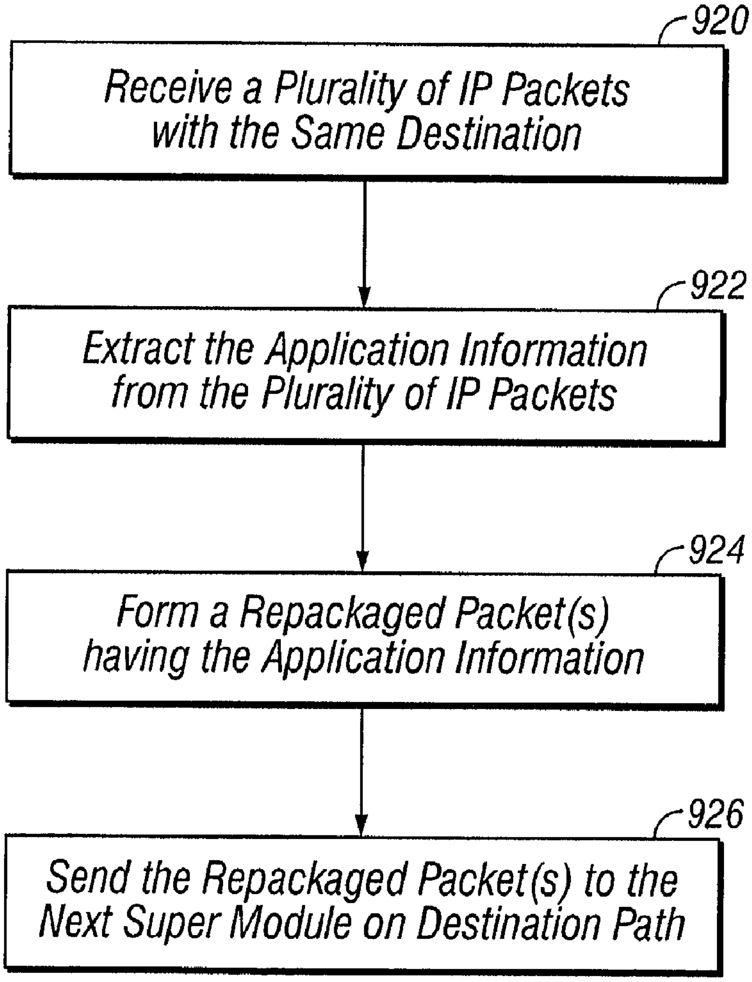
Publication Classification

(51) **Int. Cl.**^{7} **G06F 15/16**

(52) **U.S. Cl.** **709/246**; 709/203

(57) **ABSTRACT**

A system and method is provided for increasing the efficiency of information transfer in a network and for modifying application data in a data stream from a server to a user. In an exemplary embodiment application data, e.g., HTML, XML, SGML, scripts, or other software code, coming from a Web server at the request of a user, can be parsed into elements by an intermediary server located between the user's PC and the Web server. The intermediary server can modify, delete, add, search for, filter, or replace one or more of the elements based on a set of user defined rules and forward the changed application data to the user.


920
Receive a Plurality of IP Packets with the Same Destination
922
Extract the Application Information from the Plurality of IP Packets
924
Form a Repackaged Packet(s) having the Application Information
926
Send the Repackaged Packet(s) to the Next Super Module on Destination Path

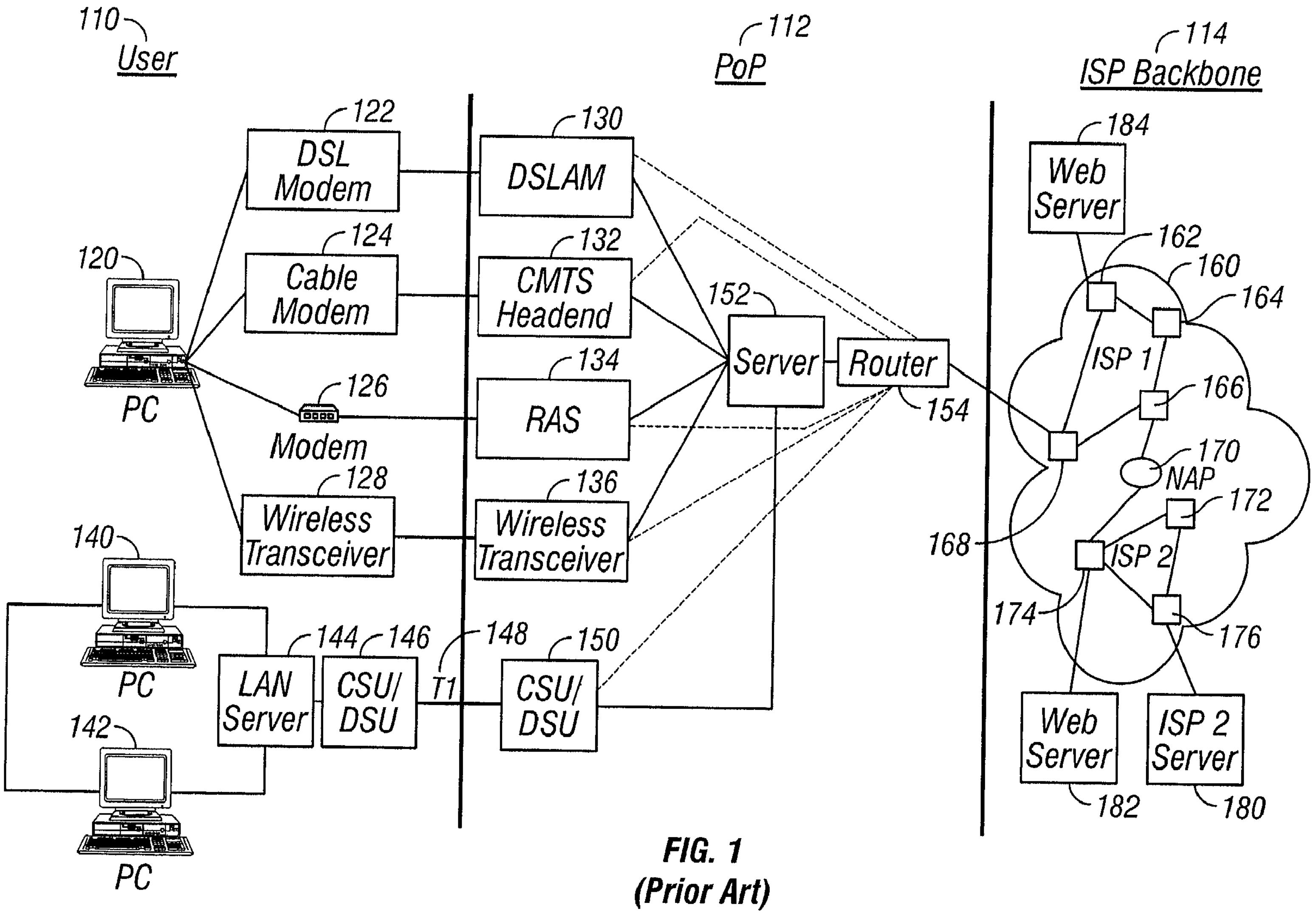


FIG. 1
(Prior Art)

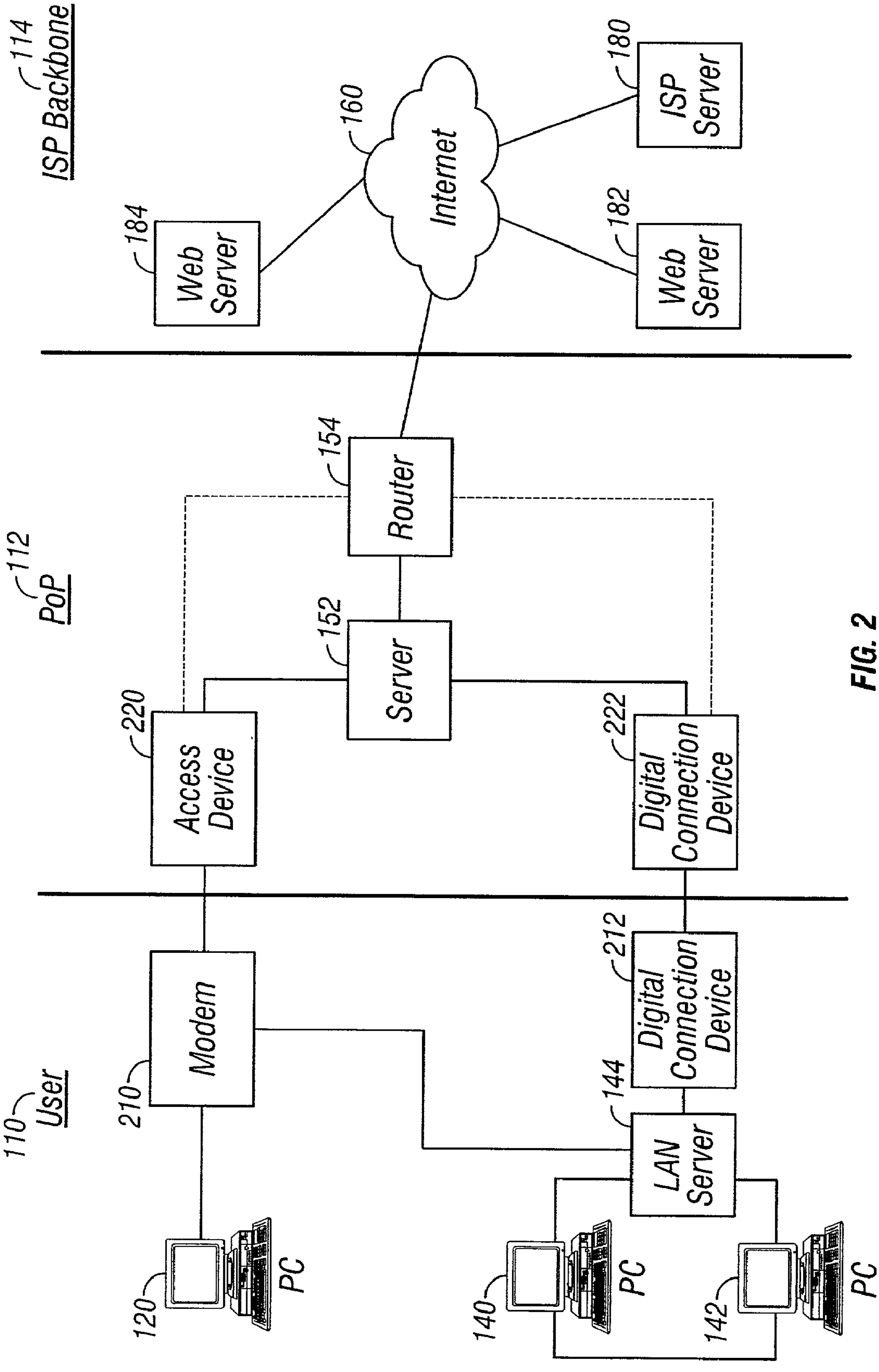

110
User
112
PoP
114
ISP Backbone
120
PC
210
Modem
140
PC
142
PC
144
LAN Server
212
Digital Connection Device
220
Access Device
152
Server
222
Digital Connection Device
154
Router
160
Internet
184
Web Server
182
Web Server
180
ISP Server

FIG. 2

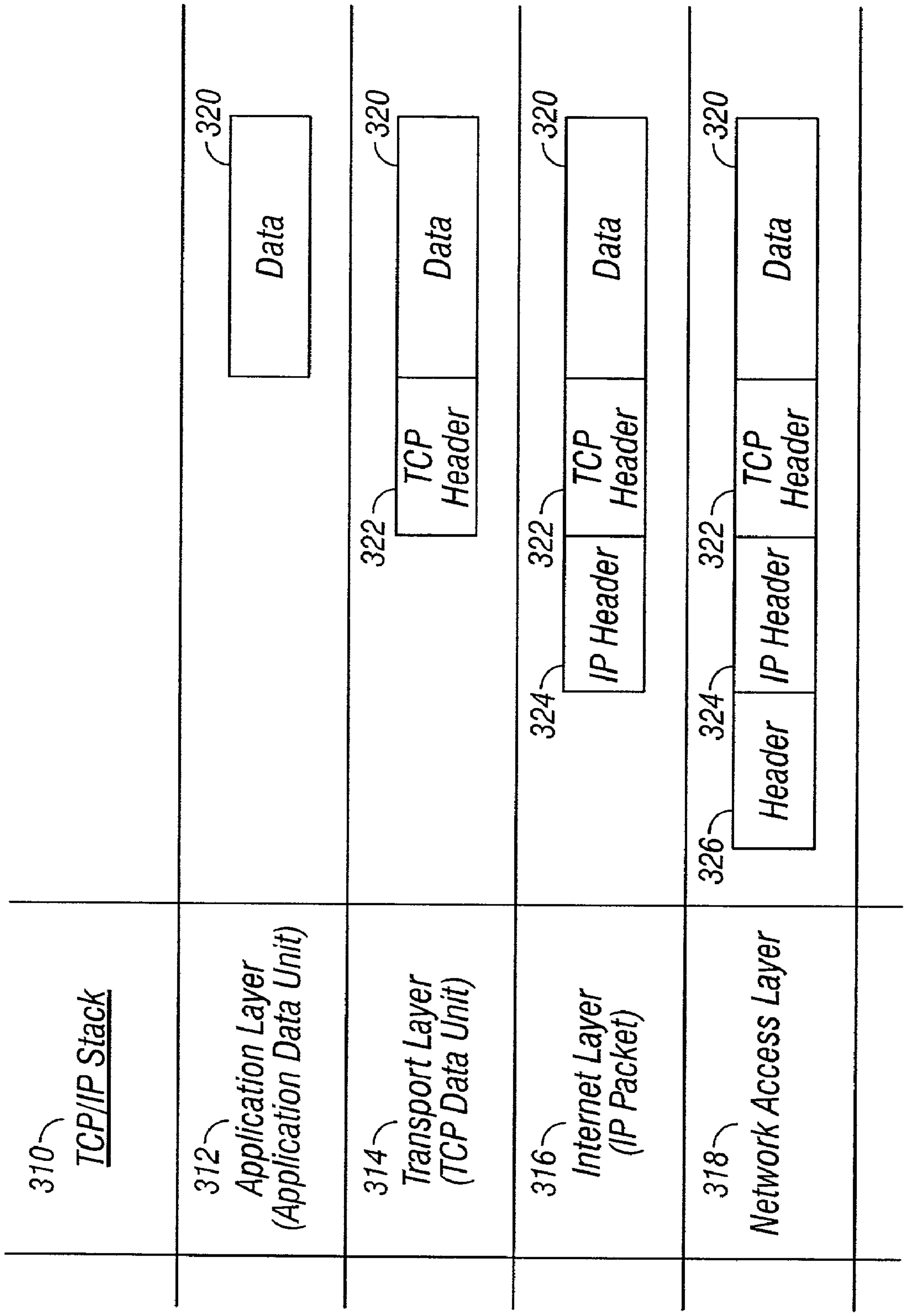

310
TCP/IP Stack
312
Application Layer
(Application Data Unit)
314
Transport Layer
(TCP Data Unit)
316
Internet Layer
(IP Packet)
318
Network Access Layer
320
Data
322
TCP Header
324
IP Header
326
Header

FIG. 3

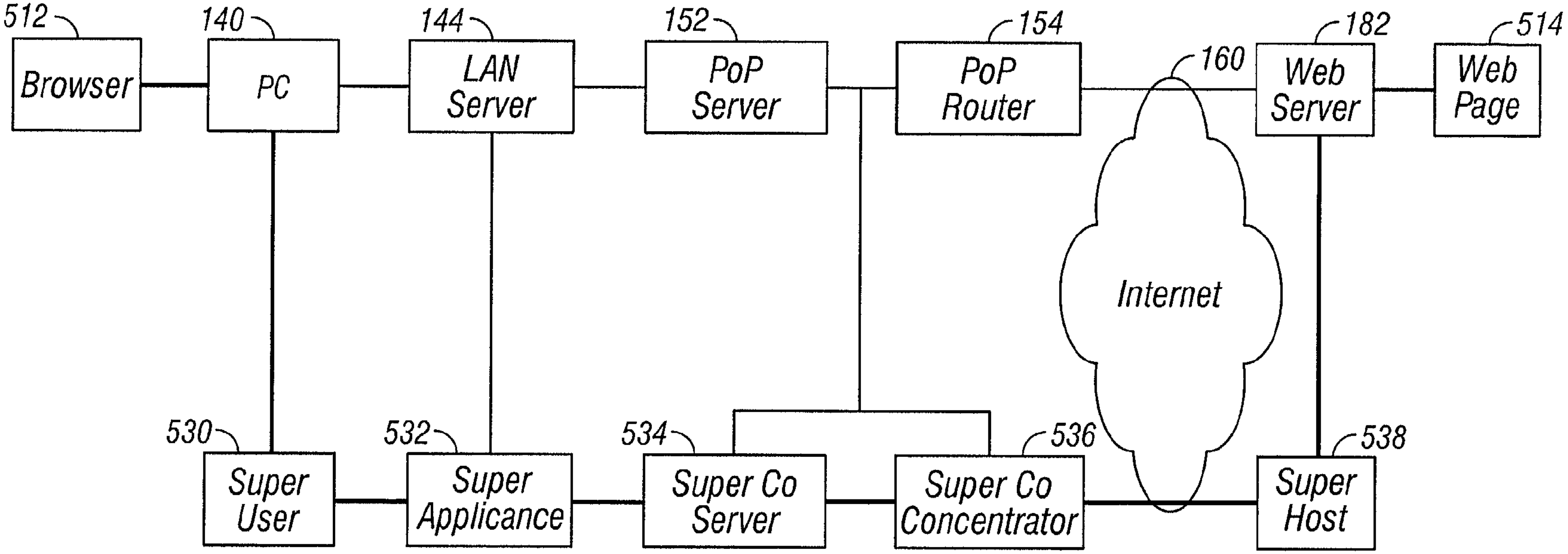

512
Browser
140
PC
144
LAN Server
152
PoP Server
154
PoP Router
160
Internet
182
Web Server
514
Web Page
530
Super User
532
Super Applicance
534
Super Co Server
536
Super Co Concentrator
538
Super Host

FIG. 4

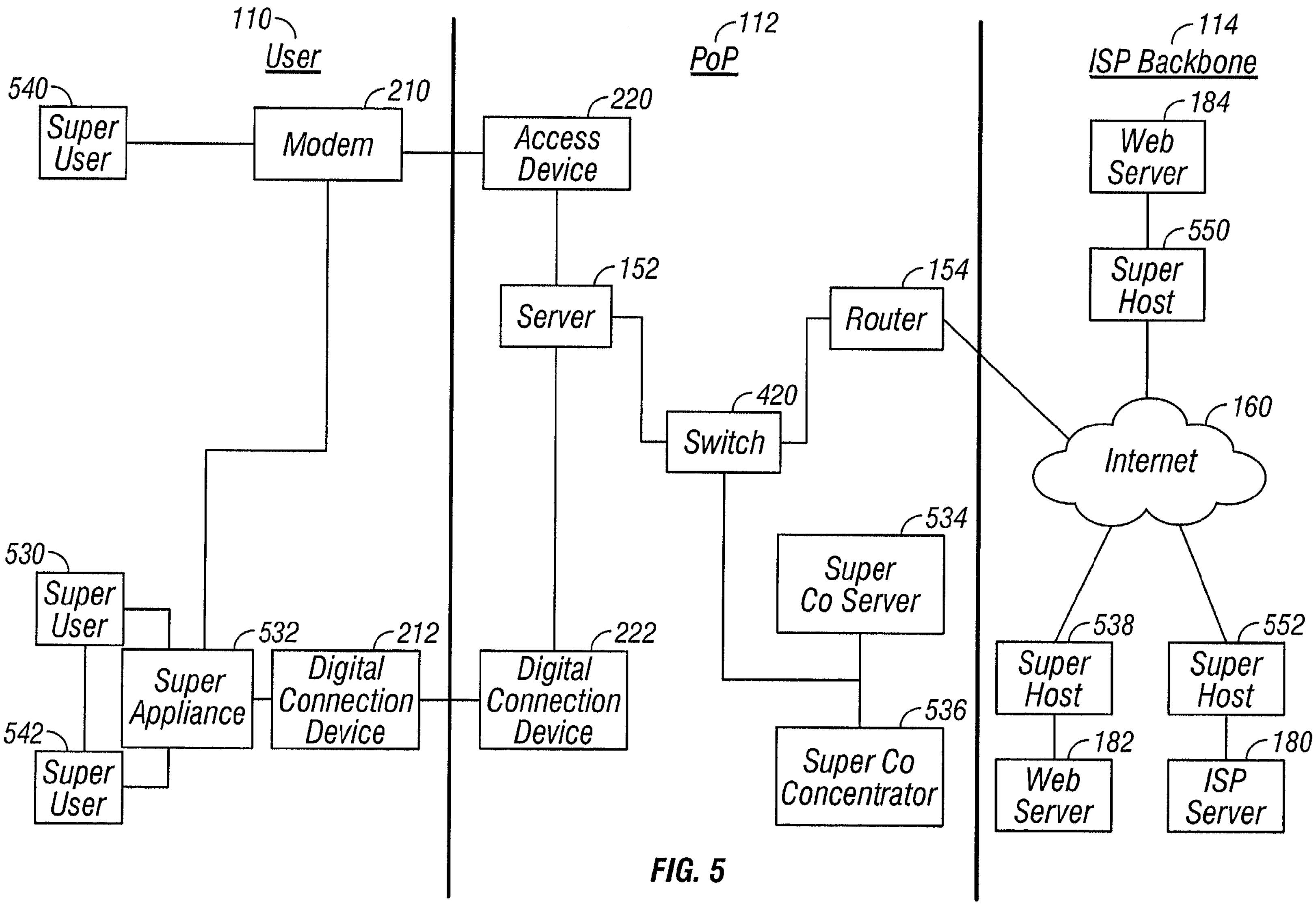

110
User
112
PoP
114
ISP Backbone
540
Super User
210
Modem
220
Access Device
184
Web Server
550
Super Host
152
Server
154
Router
420
Switch
160
Internet
530
Super User
542
Super User
532
Super Appliance
212
Digital Connection Device
222
Digital Connection Device
534
Super Co Server
536
Super Co Concentrator
538
Super Host
552
Super Host
182
Web Server
180
ISP Server

FIG. 5

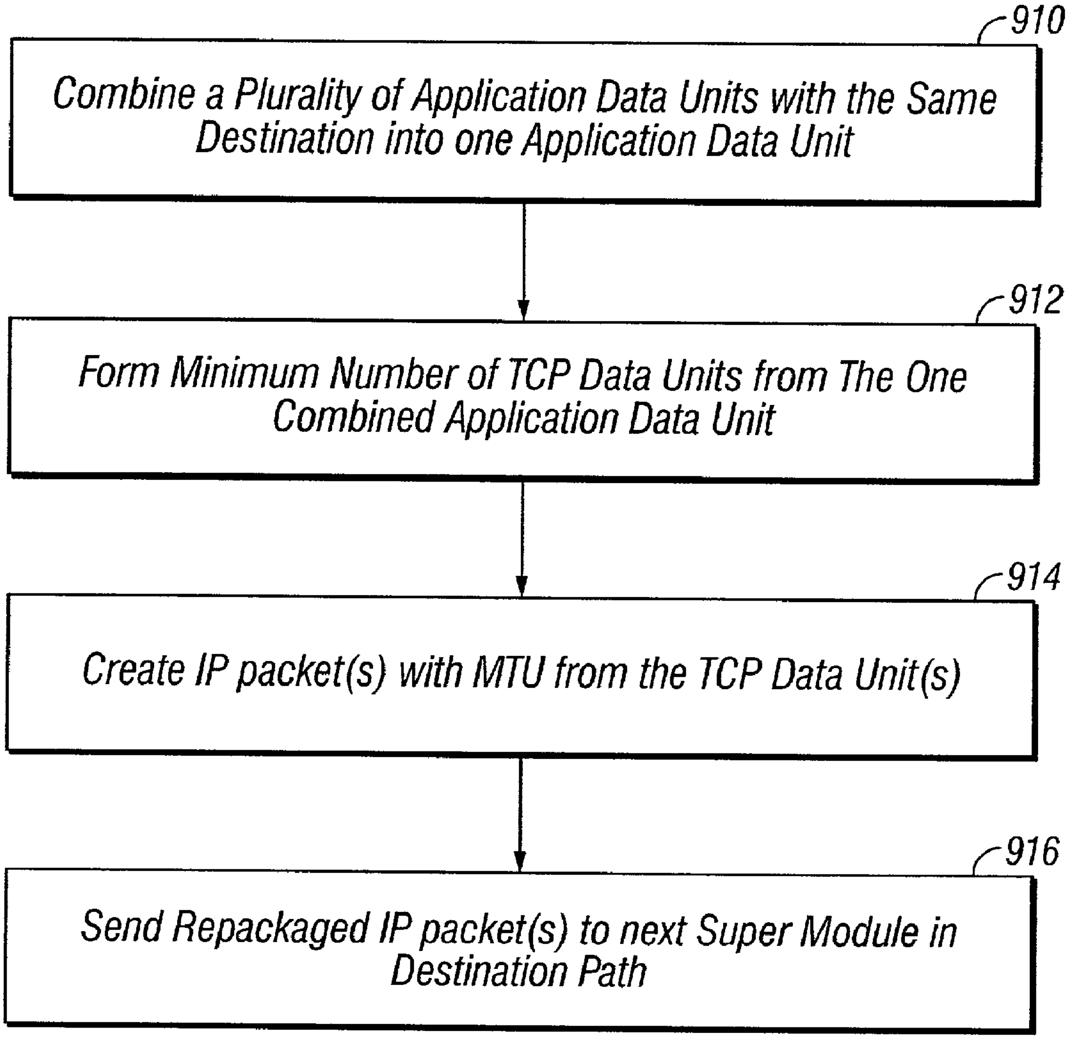

910
Combine a Plurality of Application Data Units with the Same Destination into one Application Data Unit
912
Form Minimum Number of TCP Data Units from The One Combined Application Data Unit
914
Create IP packet(s) with MTU from the TCP Data Unit(s)
916
Send Repackaged IP packet(s) to next Super Module in Destination Path

FIG. 6

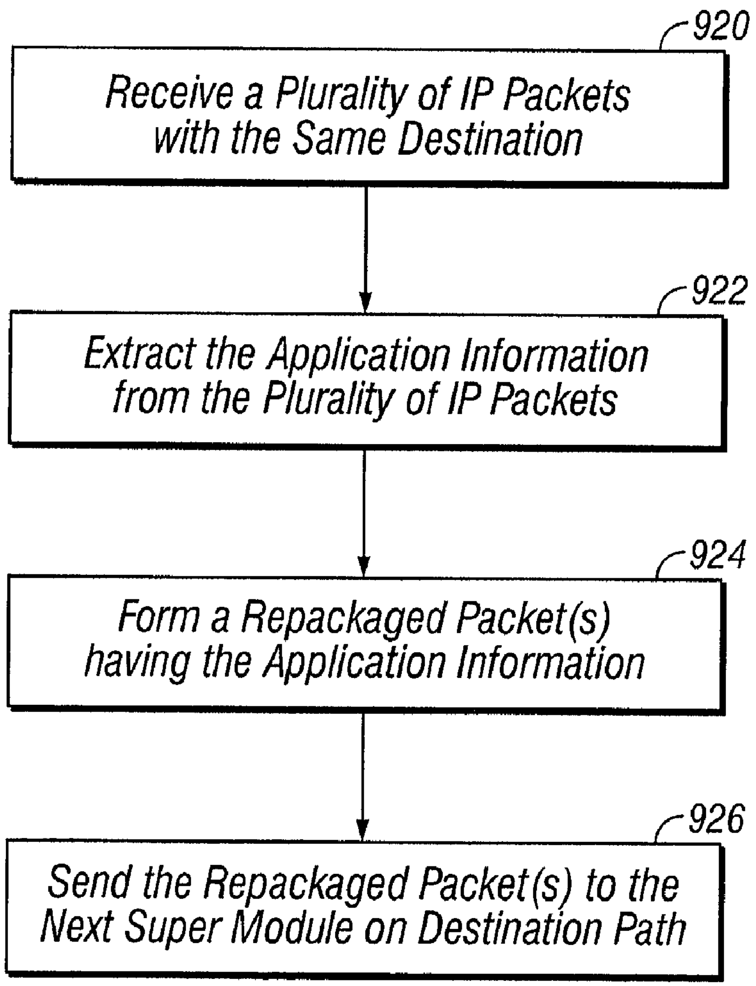

920
Receive a Plurality of IP Packets with the Same Destination
922
Extract the Application Information from the Plurality of IP Packets
924
Form a Repackaged Packet(s) having the Application Information
926
Send the Repackaged Packet(s) to the Next Super Module on Destination Path

FIG. 7

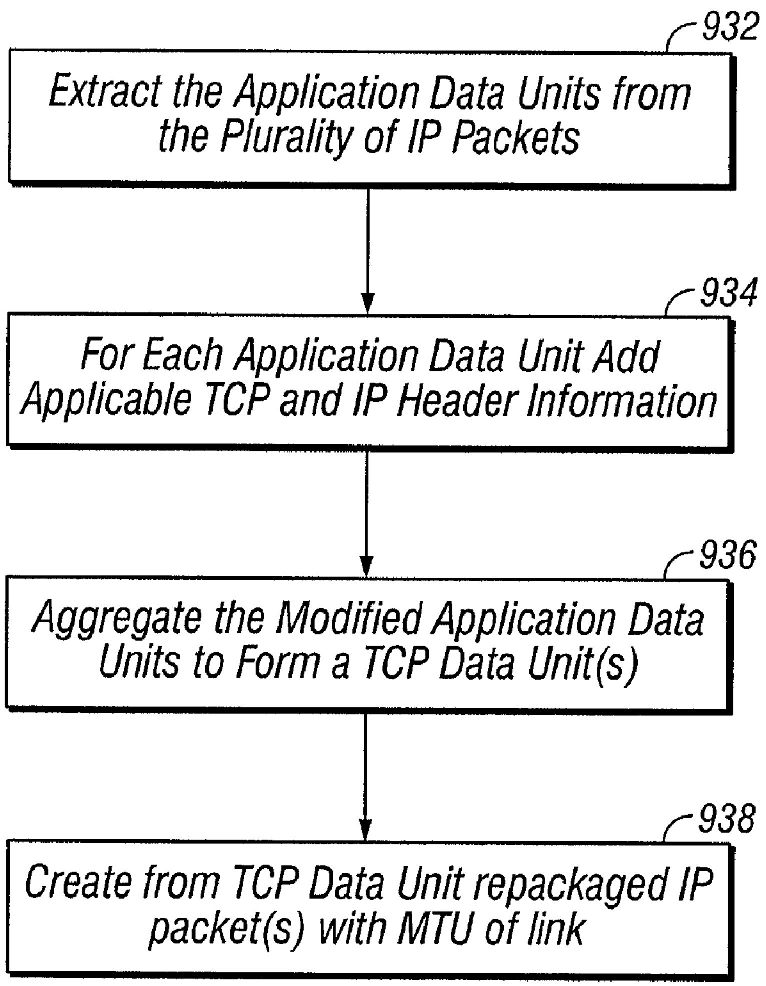

932
Extract the Application Data Units from the Plurality of IP Packets
934
For Each Application Data Unit Add Applicable TCP and IP Header Information
936
Aggregate the Modified Application Data Units to Form a TCP Data Unit(s)
938
Create from TCP Data Unit repackaged IP packet(s) with MTU of link

FIG. 8

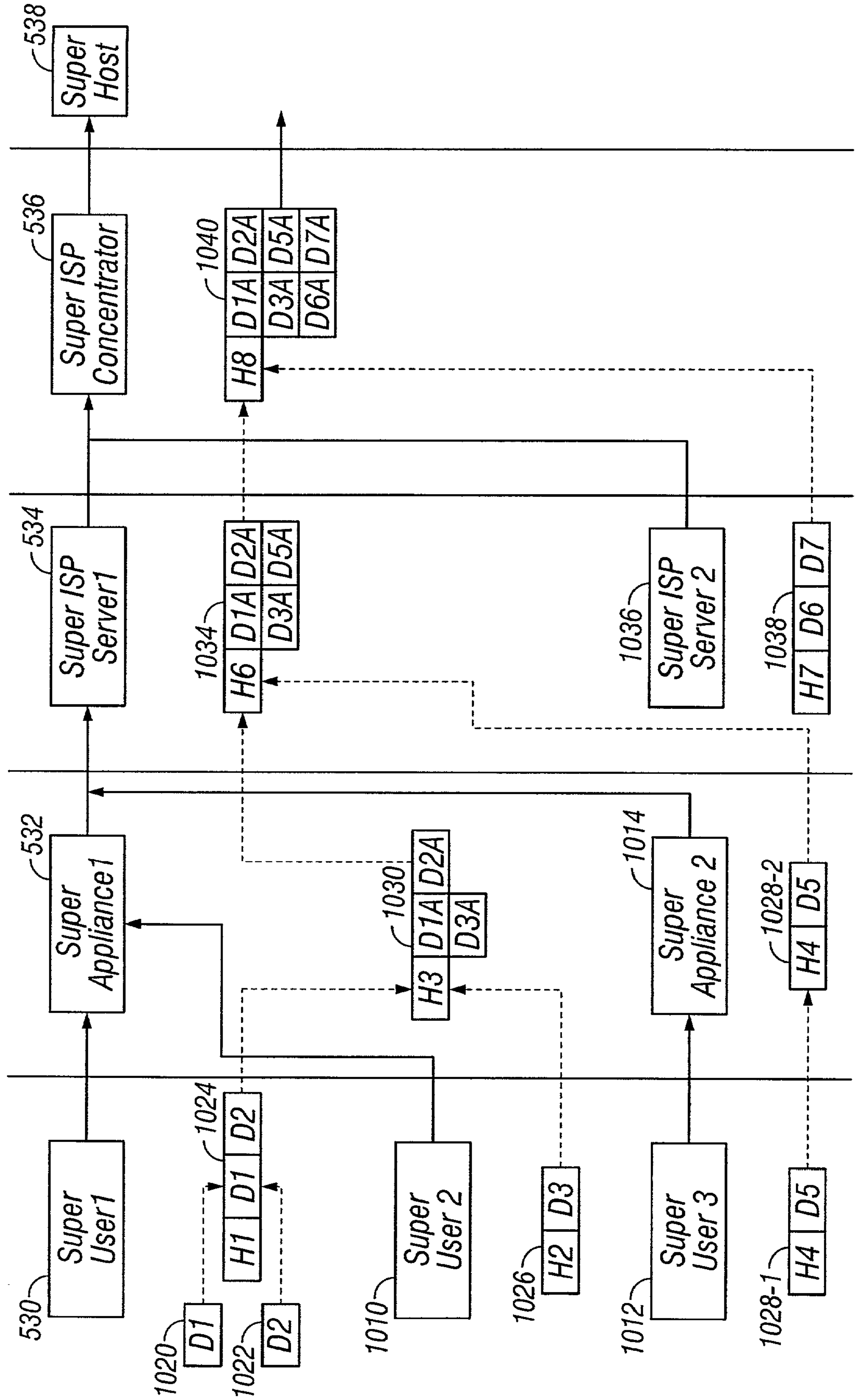

530
Super User1
532
Super Appliance1
534
Super ISP Server1
536
Super ISP Concentrator
538
Super Host
1020
D1
1022
D2
1024
H1
D1
D2
1030
H3
D1A
D2A
D3A
1034
H6
D1A
D2A
D3A
D5A
1040
H8
D1A
D2A
D3A
D5A
D6A
D7A
1010
Super User 2
1026
H2
D3
1012
Super User 3
1014
Super Appliance 2
1036
Super ISP Server 2
1028-1
H4
D5
1028-2
H4
D5
1038
H7
D6
D7

FIG. 9

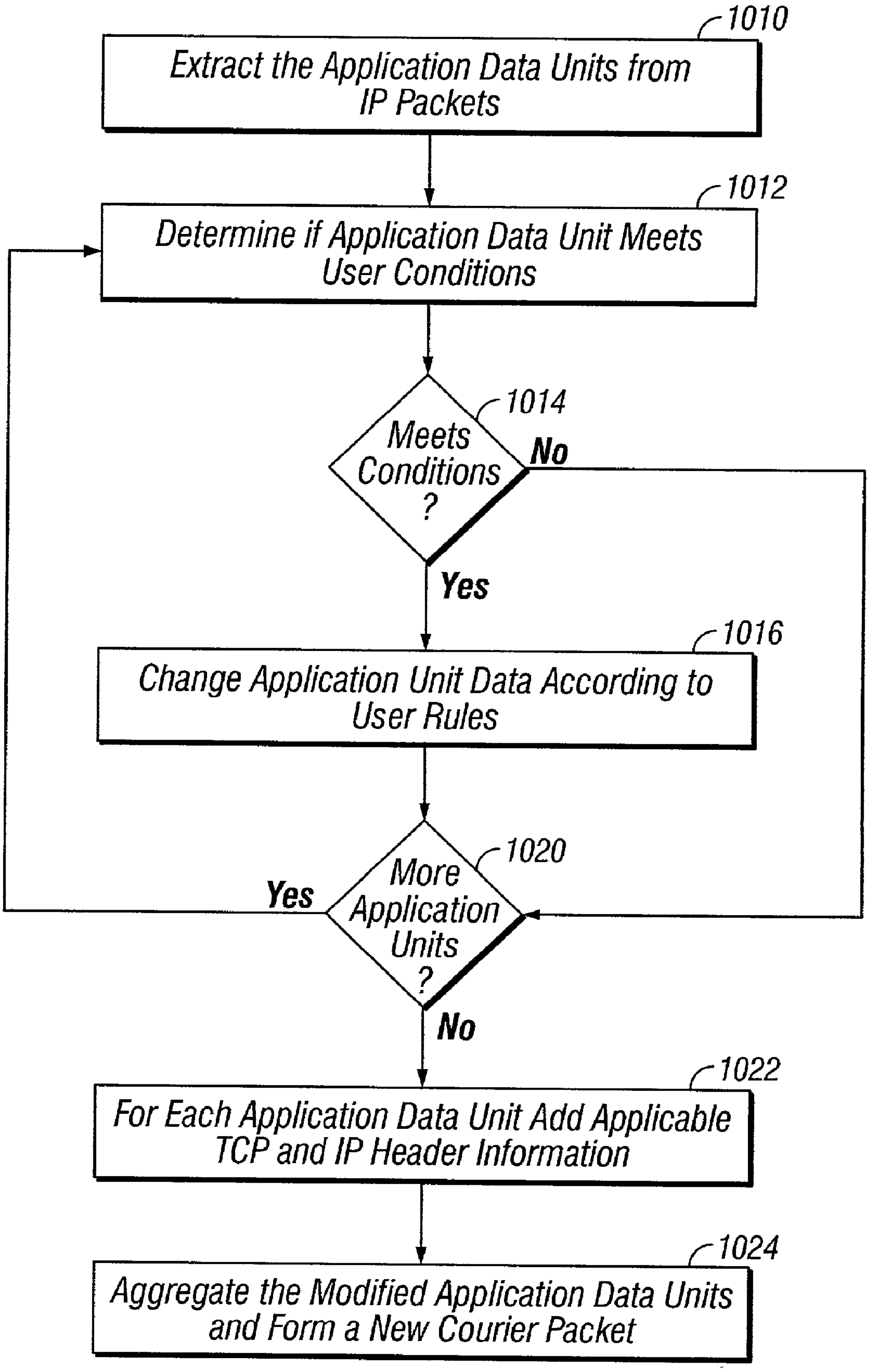

1010
Extract the Application Data Units from IP Packets
1012
Determine if Application Data Unit Meets User Conditions
1014
Meets Conditions ?
No
Yes
1016
Change Application Unit Data According to User Rules
1020
More Application Units ?
Yes
No
1022
For Each Application Data Unit Add Applicable TCP and IP Header Information
1024
Aggregate the Modified Application Data Units and Form a New Courier Packet

FIG. 10

SYSTEM AND METHOD FOR MODIFYING A DATA STREAM USING ELEMENT PARSING

CROSS REFERENCES

[0001] The following copending, commonly assigned applications are incorporated herein by reference in their entirety: U.S. Utility Application entitled, "System And Method For Increasing the Effective Bandwidth of a Communications Network", by Michael J. Parrella, Sr., et al., filed Jun. 4, 2002, Attorney Docket No. 20275-0003; and U.S. Utility Application entitled, "System And Method For Reducing The Time to Deliver Information from a Communications Network To a User", by Michael J. Parrella, Sr., et al., filed Jun. 4, 2002, Attorney Docket No. 20275-0004.

[0002] This application claims priority from and incorporates by reference in its entirety U.S. Provisional Application Serial No. 60/295,721, titled "System and Method for Improving the Effective Bandwidth of a Communication Device", by Michael J. Parrella et. al., filed Jun. 4, 2001, U.S. Provisional Application Serial No. 60/295,672, titled "Method and System Providing Compression/Decompression of Communication Data", by Michael J. Parrella et al., filed Jun. 4, 2001, U.S. Provisional Application Serial No. 60,295,676, titled "System and Method Providing Packaging of Parseable Data Elements in a Network Communication", by Michael J. Parrella et al., filed Jun. 4, 2001, U.S. Provisional Application Serial No. 60/295,720, titled "Bi-Directional File Transfer Multiplier", by Michael J. Parrella et al., filed Jun. 4, 2001, U.S. Provisional Application Serial No. 60/295,671, titled "Modification of a Data Stream Using Element Parsing", by Michael J. Parrella et al., filed Jun. 4, 2001.

FIELD OF THE INVENTION

[0003] The invention relates generally to the field of communications, and in particular to the efficient transfer of information over a computer network.

BACKGROUND OF THE INVENTION

[0004] The Internet has grown considerably in its scope of use over the past decades from a research network between governments and universities to a means of conducting both personal and commercial transactions by both businesses and individuals. The Internet was originally designed to be unstructured so that in the event of a breakdown the probability of completing a communication was high. The method of transferring information is based on a concept similar to sending letters through the mail. A message may be broken up into multiple TCP/IP packets (i.e., letters) and sent to an addressee. Like letters, each packet may take a different path to get to the addressee. While the many small packets over many paths approach provides relatively inexpensive access by a user to, for example, many Web sites, it is considerably slower than a point-to-point connection between a user and a Web site.

[0005] **FIG. 1** is a block diagram showing a user connection to the Internet of the prior art. In general a user **110** connects to the Internet via a point-of-presence (PoP) **112** traditionally operated by an Internet Service Provider (ISP). The PoP is connected to the ISP's backbone network **114**, e.g., ISP1. Multiple ISP backbone networks, e.g., ISP1 and ISP2, are connected together by Network Access Points, e.g., NAP **170**, to form the Internet "cloud"**160**.

[0006] More specifically, a single user at a personal computer (PC) **120** has several choices to connect to the PoP **112** such as a direct subscriber line (DSL) modem **122**, a TV cable modem **124**, a standard dial-up modem **126**, or a wireless transceiver **128** on, for example, a fixed wireless PC or mobile telephone. The term personal computer or PC is used herein to describe any device with a processor and a memory and is not limited to a traditional desktop PC. At the PoP **112** there will be a corresponding access device for each type of modem (or transceiver) to receive/send the data from/to the user **10**. For the DSL modem **122**, the PoP **112** has a digital subscriber line access multiplexer (DSLAM) as its access device. For the cable modem **124**, the PoP **112** has a cable modem termination system (CTMS) headend as its access device. DSL and cable modem connections allow hundreds of kilo bits per second (Kbps) and are considerably faster than the standard dial up modem **126** whose data is received at the PoP **112** by a dial-up remote access server (RAS) **134**. The wireless transceiver **128** could be part of a personal digital assistant (PDA) or mobile telephone and is connected to a wireless transceiver **136**, e.g., a base station, at the PoP **112**.

[0007] A business user (or a person with a home office) may have a local area network (LAN), e.g., PCs' **140** and **142** connected to LAN server **144** by Ethernet links. The business user may have a T1 (1.544 Mbps), a fractional T1 connection or a faster connection to the PoP **112**. The data from the LAN server **144** is sent via a router (not shown) to a digital connection device, e.g., a channel service unit/data service unit (CSU/DSU) **146**, which in turn sends the digital data via a T1 (or fractional T1) line **148** to a CSU/DSU at the PoP **112**.

[0008] The PoP **112** may include an ISP server **152** to which the DSLAM **130**, CTMS Headend **132**, RAS **134**, wireless transceiver **136**, or CSU/DSU **150**, is connected. The ISP server **152** may provide user services such as E-mail, Usenet, or Domain Name Service (DNS). Alternatively, the DSLAM **130**, CTMS Headend **132**, RAS **134**, wireless transceiver **136**, or CSU/DSU **150** may bypass the ISP server **152** and are connected directly to the router **154** (dashed lines). The server **152** is connected to a router **154** which connects the PoP **112** to ISP1's backbone having, e.g., routers **162**, **164**, **166**, and **168**. ISP1's backbone is connected to another ISP's backbone (ISP 2) having, e.g., routers **172**, **174**, and **176**, via NAP **170**. ISP2 has an ISP2 server **180** which offers competing user services, such as E-mail and user Web hosting. Connected to the Internet "cloud"**160** are Web servers **182** and **184**, which provide on-line content to user **110**.

[0009] While the Internet provides the basically functionality to perform commercial transactions for both businesses and individuals, the significant time delay in the transfer of information between, for example, a Web server and a business or individual user is a substantial problem. For example a user at PC **120** wants information from a Web site at Web server **182**. There are many "hops" for the data to travel back from Web server **182** to user PC **120**. Also because information is being "mailed" back in packets, the packets travel back typically through different paths. These different paths are shared with other users packets and some paths may be slow. Hence there is a significant time delay even if there were sufficient capacity in all the links between Web server **182** and user **120**. However, because there are also choke points, i.e., where the traffic exceeds the capacity, there is even further delay.

[**0010**] Two major choke points are the last and second to last mile. The last mile is from the PoP **112** to the user **110**. This is readily evident when the user **120** is using a dial up modem with a maximum speed of 56 Kbps. Even with a DSL modem of about 512 Kbps downloading graphics may be unpleasantly slow. The second to last mile is between the ISPs. An ISP with PoP **112** may connect via its backbone **114** to a higher level ISP (not shown) to get regional/national/global coverage. As an increase in bandwidth to the higher level ISP increases the local ISP's costs, the local ISP with, for example PoP **112**, may instead reduce the amount of bandwidth available to user **110**. The effect is that there is more traffic than link capacity between Web server **182** and PC **120** and hence a significant delay problem. In today's fast pace world this problem is greatly hindering the use of the Internet as a commercial vehicle.

[**0011**] In addition to the choke points and inefficiencies of traditional TCP/IP traffic, there is a lot of noise traffic. Like junk mail the traffic routes become clogged and the user is inundated with unwanted information. Since web sites and ISPs may receive funding from advertisers, their interests may diverge from the commercial user who is looking for targeted information and does not need nor want the distractions.

[**0012**] Filtering or "ad blocking" by a user's web browser of, for example, pop-up windows, banner ads, and other annoying advertisements, is well known in the arts. And while a corporate server may block selected URL's or IP addresses, the burden is still on the user's browser to do the filtering.

[**0013**] Therefore not only is there is a need for improving the efficiency of the transfer of information over a communications network, e.g., the Internet, but there needs to a way of reducing the undesirable data traffic to a user.

SUMMARY OF THE INVENTION

[**0014**] The present invention provides a system and method for increasing the efficiency of information transfer in a network and for modifying application data in a data stream from a server to a user. In an exemplary embodiment application data, e.g., HTML, XML, SGML, scripts, or other software code, coming from a Web server at the request of a user, can be parsed into elements by an intermediary server located between the user's PC and the Web server. The intermediary server can modify, delete, add, search for, filter, or replace one or more of the elements based on a set of user defined rules and forward the changed application data to the user. When the intermediary server is close to the Web server and filters out much of the user specific undesirable data, e.g., banner ads, the overall effective network bandwidth is also increased.

[**0015**] One embodiment of the present invention includes a method for changing application data sent by a first computer system to a second computer system via a communications network, wherein the second computer has a browser for displaying the application data. The method includes: parsing the application data into elements by the first computer; if an element of the elements satisfies a predetermined user condition, changing the element according to a predetermined action, wherein the changing includes replacing, modifying, and adding; and sending the changed element to the browser.

[**0016**] Another embodiment of the present invention includes a method for changing application data by an intermediary computer in a data stream from a first computer to a second computer. The method includes: extracting application data received from at least one IP packet; determining if a part of the application data meets a predefined user condition; responsive to the part meeting the predefined user condition, changing the part according to a predefined user rule; combining the changed part with other application data and forming at least one new IP packet; and sending the new IP packet.

[**0017**] Yet another embodiment of the present invention includes a system for modifying application data elements in a data stream. The system includes: a first super module for receiving at least one application data elements in the data stream; a decision module for analyzing the application data element according to a set of predetermined user rules and for modifying the application data element when predetermined conditions are met; a repackaging module for creating a courier packet using the modified data element; and a second super module for receiving the courier packet.

[**0018**] These and other embodiments, features, aspects and advantages of the invention will become better understood with regard to the following description, appended claims and accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[**0019**] **FIG. 1** is a block diagram showing a user connection to the Internet of the prior art;

[**0020**] **FIG. 2** is a simplified, but expanded, block diagram of **FIG. 1** and is used to help explain the present invention.

[**0021**] **FIG. 3** shows the TCP/IP protocol stack and the associated data units for each layer.

[**0022**] **FIG. 4** is a block diagram of the communication path between a browser and a web server of an embodiment of the present invention.

[**0023**] **FIG. 5** is a block diagram of the super modules inserted in the conventional system of **FIG. 2** of an embodiment of the present invention.

[**0024**] **FIG. 6** is a flowchart for repackaging a plurality of application data units at a Super User of an embodiment of the present invention.

[**0025**] **FIG. 7** is a flowchart for repackaging a plurality of received IP packets at a super module of another embodiment of the present invention.

[**0026**] **FIG. 8** explains in more detail steps **922** and **924** of **FIG. 7**. At step **932** the application data units are extracted from the IP packets.

[**0027**] **FIG. 9** shows an example of courier packets from a Super User to a Super Host of an aspect of the present invention.

[0028] **FIG. 10** is a flow chart for changing the application data units at a super module of one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0029] In the following description, numerous specific details are set forth to provide a more thorough description of the specific embodiments of the invention. It is apparent, however, to one skilled in the art, that the invention may be practiced without all the specific details given below. In other instances, well known features have not been described in detail so as not to obscure the invention.

[0030] In order for individuals and businesses to use the Internet as an effective commercial vehicle, the time for a user to request and receive information must be significantly reduced compared to the typical times that occur today. The present invention provides both a "super" system that may be overlaid on parts of the Internet infrastructure and techniques to increase information flow in the network, which, either separately or in combination, significantly reduce the user's wait time for information from, for example, Web sites or other users.

[0031] **FIG. 2** is a simplified, but expanded, block diagram of **FIG. 1** and is used to help explain the present invention. Where applicable the same labels are used in **FIG. 2** as in **FIG. 1**. The modem **210** includes the DSL modem **122**, cable modem **124**, dial-up modem **126**, and wireless transceiver **128** of **FIG. 1**. Likewise the access device **220** includes the corresponding DSLAM **130**, CMTS Headend **132**, RAS **134**, and wireless transceiver **136** of **FIG. 1**. The digital connection devices **212** and **222** include the CSU/DSU devices **146** and **150**, and in addition include, satellite, ISDN or ATM connection devices. **FIG. 2** has an additional connection between LAN server **144** and modem **210**, to illustrate another option for a LAN to connect to the PoP **112** besides the digital connection device **212**. Most of the computer and network systems shown in **FIG. 2**, communicate using the standardized Transport Communication Protocol/Internet Protocol (TCP/IP) protocol.

[0032] **FIG. 3** shows the TCP/IP protocol stack and the associated data units for each layer. The TCP/IP protocol stack **310** includes an application layer **312**, transport layer **314**, Internet layer **316**, and network access layer **318**. The application layer receives the application or user data **320**, one block or unit of data, which we will call an application data unit. For example, a user request for a Web page would be one application data unit. There are numerous application level protocols in TCP/IP, including Simple Mail Transfer Protocol (SMTP) and Post Office Protocol (POP) used for e-mail, Hyper Text Transfer Protocol (HTTP) used for the World-Wide-Web, and File Transfer Protocol (FTP).

[0033] The transport layer **314** includes the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP is a connection oriented protocol that provides a reliable virtual circuit between the source and destination. TCP guarantees to the applications that use it to deliver the stream of bytes in the order they were sent without duplication or data loss even if the IP package delivery service is unreliable. The transport layer adds control information via a TCP header **322** to the data **320** and this called a TCP data unit. UDP does not guarantee packet delivery and applications which use UDP must provide their own means of verifying delivery.

[0034] The Internet layer **316** is named because of the inter-networking emphasis of TCP/IP. This is a connectionless layer that sends and receives the Internet Protocol (IP) packets. While the IP packet has the original source address and ultimate destination address of the IP packet, the IP layer at a particular node routes the IP packet to the next node without any knowledge, if the packet reaches its ultimate destination. The IP packet includes an IP Header **324** added to the TCP data unit (TCP header **322** and data **320**).

[0035] The network access layer **318** is the bottom layer that deals with the physical transfer of the IP packets. The network access layer **318** groups together the existing Data Link and Physical Layer standards rather than defining its own. This layer defines the network hardware and device drivers. A header **326** and a trailer (not shown) are added to the IP packet to allow for the physical transfer of the IP packet over a communications line.

[0036] An example of the use of the TCP/IP protocol in **FIG. 2** is a user at the PC **140** requesting a web page from web server **182**. The user through his browser creates a user request for a Web page, i.e., application data unit **320** (**FIG. 3**), at the application layer **312**. The TCP/IP stack **310** creates one or more TCP data units where each TCP data unit has part of the application data unit **320** with a TCP header **322** appended to it. The transport layer **314** at PC **140** establishes a peer-to-peer connection, i.e., a virtual circuit, with the TCP the transport layer **314** at web server **182**. Each TCP data unit is divided into one or more IP packets. The IP packets are sent to LAN server **144** and then to PoP server **152**, where they are then sent out to the Internet **154** via PoP router **154**. The IP packets proceed through multiple paths on Internet **160** and arrive at web server **182**. The transport layer **314** at web server **182** then reassembles the TCP data units from the IP packets and passes the TCP data units to application layer **312** to reassemble the user request. The user request to get the web page is then executed. To send the web page back to the user, the same TCP virtual circuit may be used between the transport layers of the Web server **182** and PC **140**. The web page then is broken up into TCP data units, which are in turn broken up into IP packets and sent via Internet **160**, PoP router **154**, PoP server **152**, LAN server **144**, to PC **140**.

[0037] **FIG. 4** is a block diagram of the communication path between a browser and a web server of an embodiment of the present invention. The conventional exchange between browser **512** and web server **182**, when a user using browser **512** requests a Web page **514** from web server **182**, was described above. An embodiment of the present invention creates a plurality of "super" modules, including Super User **540**, Super Appliance **532**, super Central Office (CO) server **534**, Super CO Concentrator **536**, and Super Host **538**, that provides an alternative super freeway path to exchange data between browser **512** and web server **182**. The user request for Web page **514** is sent by browser **512**, executing on PC **140**, to Super User software **530** also running on PC **140**. Super User **530** then sends the user request to Super Appliance software **532** running on LAN server **144** (or in an alternative embodiment executing on its own server). Super Appliance **532** then sends the user request to Super CO Server **534**, which sends the request to Super CO Concentrator **536**. The Super CO Server **534** and Super CO Concentrator **536** may be standalone servers or may be software that runs on PoP server **152**. Super CO Concentrator **536** sends the user request via Internet **160** to Super Host **538** which may have its own server (or in an alternative embodiment Super Host **538** is software that runs on web server **182**). The user request proceeds from Super Host **548** to web server **182**, which retrieves web page **154** from a web site running on web server **182** (the web server **182** may include a Web farm of servers and multiple Web sites). The web page **514** then proceeds back to browser **512** via Super Host **538**, Super CO Concentrator **536**, Super CO Server **534**, Super Appliance **532**, and Super User **530**.

[**0038**] In other embodiments, one or more of the super modules may be missing, for example, the Super Appliance **532**. In the case of a missing Super Appliance **532**, Super CO Server **534** exchanges information with Super User **530** through LAN server **144**. Another example is if Super Host **548** was not present, then web server **182** exchanges information with Super CO Concentrator **536**. Thus if a super module is missing, the corresponding normal module, e.g., PC **140**, LAN server **144**, PoP server **150**, PoP router **154**, and web server **182**, is used instead. All or some of the super modules can be used and as long as there is at least one communication link between at least two different super modules, the information flow across the link improves significantly. Additionally, more super modules can deployed to extend the granularity of the super layer over the network.

[**0039**] **FIG. 5** is a block diagram of the super modules inserted in the conventional system of **FIG. 2** of an embodiment of the present invention. The same labels are used in **FIG. 4** as in **FIG. 2** where the devices are the same or similar. Super User **540** is connected through modem **210** is connected to PoP Server **152** via access device **220**. A local area network having Super User **530**, Super User **542**, and Super Appliance **532** is connected to modem **210** or digital connection device **212**, where digital connection device **212** is connected to PoP server **152** by digital connection device **222**. Super Appliance **532** includes software executing on LAN server **144**. Server **152** is connected to router **154** via switch **420**, which detours the packet traffic to Super CO Server **534** and Super CO Concentrator **536**. Router **154** is connected to the Internet cloud **160**. From Internet **160**, traffic can go to Super Host **538** connected to web server **182** or to Super Host **550** connected to web server **184** or to Super Host **552** connected to ISP Server **180**.

[**0040**] Super System Components

[**0041**] Described below is one embodiment of each of the components of the super system of **FIG. 5**, including Super User **540**, Super Appliance **532**, Super CO Server **534**, Super CO Concentrator **536**, and Super Host **538**.

[**0042**] The Super User **530** includes software which resides on the user's PC, e.g., PC **140**. A browser, e.g., Microsoft's Internet Explorer, is set to proxy to the Super User **530**, so that all browser requests for data are supplied from the Super User **530**. In addition, all user requests via the browser are sent to the Super User **530**. Hence the browser is isolated from the rest of the network by the Super User. The Super User caches all the data the user has requested in a local cache on the user's PC, so that when the user requests the data again, it may be retrieved locally, if available, from the local cache. If the data that is cached exceeds a predetermined file size, then the Super User analyzes all the data in the local cache and deletes the data that is least likely to be used. For example, a conventional least recently used algorithm may be used to discard old data. Some of the software function of Super User **540** are:

> [**0043**] 1. Caching: If the browser requests data that exists in the local cache and the data meets the cache life requirements, then the data is supplied from the local cache. Otherwise the data is retrieved from the nearest super module cache, e.g., the Super Appliance **532** or Super CO Server **534**, Super CO Concentrator **536**, or Super Host **538**, where the updated data is available or if not available from any super cache then from the Web server. Each data element has a cache life, that is how long it can be used from a cache before it needs to be refreshed.
>
> [**0044**] 2. Refreshing the Cache: When the Super User PC is idle (not actively retrieving data from the Internet), the Super User checks the local cache and automatically refreshes data that is reaching its cache life. The Super User, using Artificial Intelligence (AI) or other techniques, prioritizes the refreshing based on what it determines the user is most likely to request. For example, the Super User can keep a count on how often a user accesses a web page. A higher count would indicate that the user is more likely to request that web page in the future, and the Super User would automatically refresh that page.
>
> [**0045**] 3. Pre-fetching: Using AI or other techniques the Super User, during idle times, pre-fetches web pages (i.e., retrieves web pages that the user has not yet asked for) that have a high probability of being needed by the user. For example, if a user is viewing some pages on a catalog site, then there is a high probability that the user will view other pages on the site in the same category. The Super User would pre-fetch these pages. The pre-fetching increases the probability that the user will get the data from the local cache.
>
> [**0046**] 4. Courier packets (described later) are packaged and the packaged data compressed by the Super User before being sent to the Super Appliance or Super CO Server. Courier packets are un-packaged and the un-packaged data uncompressed by the Super User before being sent to the browser.

[**0047**] The Super Appliance **532** includes software executing on LAN server **144**. Some of the functions performed by the Super Appliance **532** includes, firewall security, global caching, teaming, smart hosting, and email management. Further function performed by the Super Appliance software include:

> [**0048**] 1. If the Super Appliance is attached to a Super CO Server, then all the data transmitted between them is compressed and packaged into courier packets, otherwise standard Internet requests are used and the responses are packaged into courier packets before the responses are sent to the Super User.

[0049] 2. The Super Appliance also automatically copies and maintains web sites that are used frequently by its users.

[0050] 3. If the Super Appliance is attached to a Super CO Server, then it updates its copy of the web sites only when notified of changes from the Super CO Server. If the Super Appliance is not attached to a Super CO Server then it checks for updates of the web sites during idle times and/or during periodically predetermined intervals.

[0051] 4. If Super Users are attached to the Super Appliance then all data responses are transmitted in compressed format to the Super Users. If regular users are attached to the Super Appliance, then the data responses are decompressed in the Super Appliance and sent to the users. If the Super User is maintaining web sites, then anytime a web page is updated on the Super Appliance a notification is sent to the Super User so that the Super User may request the change.

[0052] 5. The Super User will also notify the Super Appliance of information about the user's PC monitor density so that adjustments can be made to the graphics transmitted over the local area network. Sending high density graphics to a monitor that can not display the graphics is a waste of network resources. The software in the Super Appliance adjusts the graphics density before transmitting the data.

[0053] 6. If more than one Super User requests the same data, then the Super Appliance implodes the request and sends only one request to the next super module, e.g., the Super CO Server. If there is not another super module between the Super Appliance and the Web site, then the request is still imploded and a standard TCP/IP request is made. When the response to the imploded request is received then the data is exploded by the Super Appliance and the data is sent to the appropriate Super Users.

[0054] The more web sites that are maintained at the Super Appliance the more the access speed for web pages approaches the local area network speed. The more web pages maintained at the Super User the more the web access speed approaches hard disk access speed. The more web pages that can be copied and maintained on the Super Appliance and the Super User, the less the last mile becomes a bottleneck for response time.

[0055] The Super CO Server **534** is the bridge between the Internet backbone **114** and the user **110**. One objective of the Super CO Server **534** is to minimize the traffic between the user and the Internet. The Super CO Server accomplishes this by copying the web sites accessed by the super or normal users via the Super CO Server. The more web sites that are hosted on the Super CO Server, the more the network is optimized by reducing the movement of data across the network. If the web sites that are hosted at Super CO Server come from web sites stored on a Super CO Concentrator **536**, the Super CO Server **534** requests updated web pages whenever notified by the Super CO Concentrator **536** that the web pages have changed. Web pages from the Super CO Concentrator **536** are stored in compressed and repackaged format. If the web sites that are hosted on the Super CO Server are not stored in the Super CO Concentrator, then the Super CO Server checks at predetermined intervals for changes in the web site at the hosting web server. The Super CO Server keeps a log of the web sites that are hosted on every Super Appliance **532** cache. As changes occur to web sites that exist on a Super Appliance cache, a notification is sent to that Super Appliance that changes have occurred and that the Super Appliance should request updated copies of the changed web pages. As data is received from a non Super CO Concentrator site it is compressed, packaged and stored on the Super CO Server. The Super CO Server determines from its request logs the web sites that are being accessed by its users and determines which web sites to copy and maintain at the Super CO Server **534** cache. The Super CO Server will also delete sites that are not being used. If a web site is not being stored and maintained, the web page is maintained in a separate global cache so that if it is requested again it can be supplied from the global cache. A correct balance needs to be maintained between the global cache and the web hosting. The global cache and Super CO Server can be implemented as one cache and managed separately or implemented as two separate caches. If a web page is requested from a Super Appliance then the web page is sent in super compressed and repackaged format, otherwise the web page is decompressed and sent to the requesting user. The super module closest to the user unpackages any repackaged formats and decompresses the data so that it is sent to the user in native form. The super module closest to the user also caches the information in non-compressed and non-packaged format. The optimizations used are related to the amount of compression applied to the variable data (usually text) and the amount of variable data on the web page. The more Rich Data formats are used on the Internet the more optimization is achieved. Flash software, files, java programs, java scripts etc. are all stored at the Super CO Server.

[0056] The data requests from the Super Appliances that are not satisfied by the Super CO Server cache are sent to the Super CO Concentrator **536** that is responsible for servicing the URL (web site) requested. The requests are packaged compressed and imploded according to the optimization schemes. In one embodiment, the first level of data implosion occurs at the Super CO Server. In an alternative embodiment implosion is done by the Super Appliance. The Super CO Server is organized by ISP geography so that duplicate usage characteristics that are regionally oriented can be imploded on request and exploded on response. All requests and imploded requests that cannot be responded to by data in the Super CO Server's cache are passed to the Super CO Concentrator.

[0057] The Super CO Concentrator **536** is organized by Web sites (URL's). This increases the probability that Web site data that users need will be in the CO concentrator's cache. It also increase the probability that requests can be imploded and network traffic can be reduced. Each Super CO Concentrator is responsible for caching and interfacing with the Super Hosts, e.g. **538**, and other non Super Host web sites. For non Super Host web sites, Super CO Concentrator **536** is the first super module encountered and the initial repackaging, first compression, final implosion, first explosion, the conversion of all graphics to an optimized compression format, such as PNG or proprietary compression algorithms, and the first level of super caching occurs.

This is also where all the checking and refreshing occurs for the other super modules. As data from the Web sites is refreshed and updated the Super CO Servers are notified so that all caches can be updated and refreshed.

[0058] The Web server hosts one or more web sites that are attached to the Internet. The Super Host, i.e., Super Host **538**, replies to requests made from the Super CO Concentrators, e.g., **536**. Each time a request is made for a down load of any web site hosted on the Web server, the Super Host **538** retrieves the web pages from the Web server and compresses and packages the contents before sending it to the requesting Super CO Concentrator. This improves the efficiency of the web transport by the effective compression rate and by sending a single data block for all the requested web page data. Each piece of information is analyzed and compressed using techniques that best perform for the specific type of data. As each Super CO Concentrator request is received, the Super Host records the IP address of the Super CO Concentrator. The Super Host checks the web sites contained on the Web server and sends notifications of any changed web pages to any Super CO Concentrator that has requested data from the web sites historically. This allows the Super CO Concentrator to know when it needs to refresh its version of the Web site and minimizes Web traffic by allowing the Super CO Concentrator to service user requests for web pages directly from its version of the web page in the Super CO Concentrator's cache. The only time the Super CO Concentrator version of the web page needs to be refreshed is when it has changed. This allows for minimized traffic from the web hosting sites to the ISP sites. There are many ISP sites accessing data at each web site. This is a step in moving web sites to the outer fringe of the Internet and bringing compression and packaging to the inner workings of the Internet. The challenge of moving web sites to the outer fringes of the Internet is to make sure data is current, the interlocking of the super module caches insures this.

[0059] Repackaging

[0060] Typical web pages today contain a HyperText Markup Language (HTML) document, and many embedded images. The conventional behavior for a browser is to fetch the base HTML document, and then, after receipt of the base HTML document, the browser does a second fetch of the many embedded objects, which are typically located on the same web server. Each embedded object, i.e., application data unit, is put into a TCP data unit and each TCP data unit is divided into one or more IP packets. Sending many TCP/IP packets for the many embedded objects rather than, e.g., one large TCP/IP packet, means that the network spends more time than is necessary in sending the control data, in other words, the control data/time to application data/time ratio is too large. It is more efficient to combine the many embedded objects into one large application data unit and then create one (or at least a minimum number of) large TCP data unit. For the one large TCP data unit the maximum transmission unit (MTU) for the link between this sender super module and the next receiver super module is used for the IP packet(s). The sender super module will try to minimize the number of IP packets sent by trying to make each IP packet as close to the MTU as practical. For each link between a super module sender and a super module receiver the MTU is determined for that link and the size of the IP packets may change. Unlike the prior art where the lowest common denominator MTU among all the MTUs of communication links between the user and Web server is normally used, in this embodiment, the MTU of each link is used.

[0061] In one embodiment of the present invention application data units, e.g., users requests and Web server responses, are repackaged (or unpackaged) into a larger (or multiple smaller) modified application data unit(s), when necessary, at each super module, e.g., Super User, Super Appliance, super Central Office (CO) server, Super CO Concentrator, and Super Host. For example, let's combine two IP packets into one IP packet, which is one example of a "courier" packet. The first IP packet has a first IP header, a first TCP header, and a first application data unit. The second IP packet has a second IP header, a second TCP header, and a second application data unit. A first modified application data unit is created which has the first application data unit and a first pseudo header having control data from the first IP Header and first TCP header, such as source address, source and destination ports and other control information needed to reconstruct the first IP packet. A second modified application data unit is created which has the second application data unit and a second pseudo header having control data from the second IP Header and second TCP header, such as source address, source and destination ports and other control information needed to reconstruct the second IP packet. A combined application data unit is made having the first modified application data unit concatenated to the second modified application data unit. A new TCP header and IP header are added to the combined application data unit and the courier packet is formed. Thus necessary control information is embedded in the combined application data unit and the TCP/IP protocol is used to move the combined application data unit between a super module sender and a super module receiver. When the receiver is not a super module the combined application data unit is unbundled and the first IP packet and second IP packet are recreated and sent to the normal receiver by the super module sender.

[0062] **FIG. 6** is a flowchart for repackaging a plurality of application data units at a Super User of an embodiment of the present invention. At step **910** a Super User combines a plurality of application data units with the same destination into one application data unit. For example, multiple user requests to a web server, are combined. At step **912** one TCP data unit (or a minimum number of TCP data units) is formed from the one application data unit. At step **914** one IP packet (or the minimum number of IP packets), i.e., courier packet(s), are created, where each IP packet is filled to be as close as possible to the MTU number of bytes for the link or until a forwarding timer T has expired. At step **916** the courier packet(s) are sent to the next super module, e.g., the Super Appliance or Super CO Server, in the destination path.

[0063] **FIG. 7** is a flowchart for repackaging a plurality of received IP packets at a super module of another embodiment of the present invention. At step **920** the super module receives a plurality of IP packets with the same destination. At step **922** the application information is extracted from the plurality of IP packets. At step **924** the extracted application is used to form a repackaged packet(s) (i.e., a courier packet(s)). At step **924** the repackaged packet(s) is sent on its way to the next super module in the path to the common destination.

[**0064**] **FIG. 8** explains in more detail steps **922** and **924** of **FIG. 7**. At step **932** the application data units are extracted from the IP packets. For each application data unit the related TCP header and IP header control information is examined. And the applicable control information, e.g., the source, source and destination ports, and data length, are added to the corresponding application data unit to form a modified application data unit (step **934**). At step **936** the modified application data units are aggregated to form one TCP data unit (or a minimum number of TCP data units). At step **938** new repackaged IP packet(s) is formed from the TCP data unit using the MTU of the link between the sender and receiver super modules.

[**0065**] The decision on whether to form at step **936** one large TCP data unit or multiple small TCP data units is dynamically determined depending on the traffic load on the link leaving the sender super module. For example, if the link is near capacity then it is more efficient to send multiple small TCP data units, and hence small IP packets, then one (or several) large IP packets, which would have to wait.

[**0066**] **FIG. 9** shows an example of courier packets from a Super User to a Super Host of an aspect of the present invention. Super User **530** combines user requests **1020** and **1022**, i.e., application data units D1 and D2, into a courier packet **1024** according to the flowchart in **FIG. 6**. Super User **1010** has its user request D3 in IP packet **1026** and Super User **1012** has a user request D5 in IP packet **1028**. Both of these single Super User requests are repackaged to courier packets and sent to the appropriate Super Appliance. At the first Super Appliance **530**, courier packet **1024** and IP packet **1026** are received and repackaged according to the flowchart in **FIG. 7** to form larger appliance courier packet **1030**. Appliance courier packet **1030** has, for example, application data unit D1 which has been modified (D1A) to include control information from TCP and IP header Hi of IP packet **1024**. The second Super Appliance **1014** receives courier packet **1028-1**, does not change it (**1028-2**) and forwards it to Super CO Server **534**. The Super CO Server **534** receives appliance courier packet **1030** from Super Appliance **532** and courier packet **1028-2** from Super Appliance **1014**. Courier packets **1030** and **1028-2** are repackaged according to the flowchart in **FIG. 7** to form CO courier packet **1034**, which is sent to Super CO Concentrator **536**. Super CO Server **1036** has CO courier packet **1038** which is also sent to Super CO Concentrator **536**. Super CO Concentrator **536** repackages CO courier packets **1034** and **1038** to CO concentrator courier packet **1040**, which is sent to Super Host **538**. The Super Host unpacks CO concentrator courier packet **1040** to get user requests D1, D2, D3, D4, D5, D6, and D7 (e.g., HTTP or FTP requests) and the requests are sent to the Web server. The repackaging according to **FIGS. 6, 7** and **8** also occurs for the data responses from the web server to the Super Host **538** back to Super User **530** via Super CO Concentrator **536**, Super CO Server **534**, and Super Appliance **532**.

[**0067**] Changing the Application Data

[**0068**] As can be seen from the above discussion of repackaging, the application data units are examined many times as they proceed back as courier packets from the Super Host **533** to the Super User, e.g., **530**. Since between any two super modules courier packets are used the flowcharts given in **FIGS. 7 and 8** are used to receive courier packets and, if necessary, to repackage them as new courier packets. In each case the application data is extracted. If the application data is HTML, then the application data can be parsed into programming elements and IF-THEN rules applied (i.e., if a condition holds then performs a predetermined action). Another embodiment may use a scripting language such as Perl (Practical Extraction and Reporting Language) which would look for patterns in the application data and perform certain actions such as deletion, modification, replacement, or addition to the data that fit the pattern. The rules to delete, add, modify, or replace elements may be based on any desirable user criteria including content, advertising, intended audience, user, human resources, timing, context, law, geography, IP address, source, file size or type or political content.

[**0069**] For illustration purposes, the HTML code for a banner ad rotating through several pictures is used as an example of application data that may be returned as a response to a user request for a web page. There is code for an event handler to trigger the banner display based on some event, such as going to the Web page:

[**0070**] <body on Load="rotate Banner" ('images/Banner1.jpg')">

[**0071**] Next there is code to display the first banner image (i.e., Banner1.jpg):

[**0072**] <table>

[**0073**] <tr><td><img name="banner" src="images/Banner1.jpg></td>

[**0074**] </tr>

[**0075**] </table>

[**0076**] Lastly there is a function rotateBanner( ) which recursively calls itself every 5 seconds and changes the "src" property above, thus displaying a new banner image:

```
function rotateBanner (BannerSrc)
{ var Timer ID
    // swap the picture
     document.banner.src = BannerSrc;
         // wait for timeout and call self to swap next picture
     if (BannerSrc = = images/Banner1.jpg")
    TimerID = setTimeout ("rotateBanner ('images/Banner2.jpg')",
    5000);
     else if (BannerSrc = = images/Banner2.jpg")
    TimerID = setTimeout ("rotateBanner ('images/Banner3.jpg')",
    5000);
         . . . . .
}
```

[**0077**] The above example shows standard programming constructs and can be parsed by numerous software programs available to one of ordinary skill in the arts. Once the programming constructs and variables are parsed, these elements can be manipulated by user defined rules. Thus the user has an ability to filter or modify the data he/she has requested using any super module in the path from Web server to browser.

[**0078**] While the application data stream modification software may be part of any super module, in a preferred embodiment it is located on the Super CO Server **534**, Super CO Concentrator **536**, or Super Host **538** (**FIG. 4**), i.e., on the Internet side of the last mile (between POP Server **152** and LAN Server **144**). By removing the banner ads and superfluous graphics, traffic over the last mile is reduced. By removing the banner ads at the Super Host **538**, for example, would save the unnecessary data traffic over the Internet **160**. In addition there is the ability at the Super CO server **534** to use other change rules besides the user. For example, a government entity may want to replace the banner ads with public service announcements.

[**0079**] **FIG. 10** is a flow chart for changing the application data units at a super module of one embodiment of the present invention. At step **1010**, the application data units are extracted from the incoming IP packets (see step **932** of **FIG. 8**), which could be normal IP packets coming from a normal module or courier packets coming from another super module. At step **1012** one application data unit is analyzed and the user's set of IF-THEN rules are checked. When the application data unit meets the "IF" condition of the user's rules "THEN" the data may be deleted, modified, or replaced (step **1016**). If there are more application data units in the courier packet, then step **1012** is repeated. Otherwise, at step **1022**, the previously extracted pseudo TCP and IP header information is added to each application data unit. These application data units are then aggregated, and a new TCP and IP header added to form a new courier packet (Step **1024**). The new courier packet is sent to the next super module.

[**0080**] Therefore, as the application data in the courier packets pass through each super module, they are dynamically evaluated and changed according to user defined rules. In another embodiment the application data is examined and changed according to the user defined rules in one or more of the super module's super cache.

CONCLUSION

[**0081**] Although specific embodiments of the invention have been described, various modifications, alterations, alternative constructions, and equivalents are also encompassed within the scope of the invention. The described invention is not restricted to operation within certain specific data processing environments, but is free to operate within a plurality of data processing environments. Additionally, although the invention has been described using a particular series of transactions and steps, it should be apparent to those skilled in the art that the scope of the invention is not limited to the described series of transactions and steps.

[**0082**] Further, while the invention has been described using a particular combination of hardware and software, it should be recognized that other combinations of hardware and software are also within the scope of the invention. The invention may be implemented only in hardware or only in software or using combinations thereof.

[**0083**] The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will, however, be evident that additions, subtractions, deletions, and other modifications and changes may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the claims.

What is claimed is:

1. A method for changing application data sent by a first computer system to a second computer system via a communications network, wherein said second computer has a browser for displaying said application data, said method comprising:

parsing said application data into elements by said first computer;

if an element of said elements satisfies a predetermined user condition, changing said element according to a predetermined action, wherein said changing includes replacing, modifying, and adding; and

sending said changed element to said browser.

2. The method of claim 1 wherein changing further includes deleting and filtering.

3. The method of claim 1 wherein said predetermined user condition and predetermined action is based on content, advertising, intended audience, user, human resources, timing, context, law, geography, IP address, source, file size or type or political content.

4. A method for changing application data by an intermediary computer in a data stream from a first computer to a second computer, comprising:

extracting application data received from at least one IP packet;

determining if a part of said application data meets a predefined user condition;

responsive to said part meeting said predefined user condition, changing said part according to a predefined user rule;

combining said changed part with other application data and forming at least one new IP packet; and

sending said new IP packet.

5. The method of claim 4 wherein said application data is HTML information.

6. The method of claim 4 wherein said predefined user condition includes existence of HTML code for a banner ad, and said predefined user rule is selected from a group consisting of deleting said HTML code, substituting a banner ad for another product, and adding a public announcement.

7. A system for modifying application data elements in a data stream, comprising:

a first super module for receiving at least one application data elements in said data stream;

a decision module for analyzing said application data element according to a set of predetermined user rules and for modifying said application data element when predetermined conditions are met;

a repackaging module for creating a courier packet using said modified data element; and

a second super module for receiving said courier packet.

8. The system of claim 7 wherein said set of predetermined user rules is based on content, advertising, intended audience, user, human resources, timing, context, law, geography, IP address, source, file size or type or political content.

* * * * *