SPEED SENSITIVE CONTENT DELIVERY IN A CLIENT-SERVER NETWORK

TECHNICAL FIELD

This invention relates to measured data flow, and more particularly to connecting a client to a particular destination based on the connection speed of the client.

BACKGROUND

To provide greater access to the Internet, the communication protocols and languages utilized by the clients and servers have become standardized. These protocols include the Hyper-Text Transfer Protocol (HTTP), which is the communication protocol used for communications between clients and servers, and the Transfer Control Protocol/Internet Protocol (TCP/IP). The TCP portion is the transport-specific protocol for communication between computers or applications. IP is a lower-level protocol than TCP and facilitates the transmission of data packets between remotely-located machines. TCP is a transport-level protocol that operates on top of IP. TCP provides a full-duplex byte stream between applications, whether they reside on the same machine or on remotely-located machines.

166166

G00002

TCP ensures that transmitted data packets are received in the same order in which they were transmitted.

Remote terminals or computers may connect to the Internet via an Internet Service Provider (ISP) using a variety of connection methods and speeds. These may include, among other things, a dial-up modem at 28.8 kilobits per second (Kbps), dial-up at 56 Kpbs, an Integrated Services Digital Network (ISDN) line, a Digital Subscriber Line (DSL), a cable modem, or a T1 line. Although each of these methods allow for connection to the Internet, each method does so at differing speeds. For example, a typical dial-up modem connects to the Internet at 56 Kbps while a T1 line may connect at 1.544 million bits per seconds (Mbps). The increased bandwidth of the higher speed connections allows more information to be downloaded in a shorter period of time.

Web designers are taking advantage of the proliferation of high bandwidth connections to the Internet by creating more content rich websites. This allows more information, multimedia, and features to be provided to the users. However, for users that are connecting to the Internet using a slower speed connection, these high-content websites take an exceedingly long period of time to

load. Many users allow a limited period of time for a website to download. If a website is not fully downloaded within this limited time period, the user typically logs off the Internet or proceeds to another website. Thus, web-designers are forced to decide between providing content rich sites for the high-bandwidth connections and losing the low-bandwidth connections or decreasing the amount of information provided to everyone.

SUMMARY

According to one aspect of this invention, there is provided a method as defined in claim 1 of the appended claims.

According to another aspect of this invention, there is provided a method as defined in claim 7.

According to a further aspect of this invention, there is provided an apparatus as defined in claim 12.

A method of enhancing data delivery may comprise determining a response time of a remote terminal, automatically selecting a destination address optimized for the response time, and connecting the remote terminal to the optimized destination address.

In some variations, the method may further comprise determining a data flow rate. Additionally or in the alternative, the method may comprise determining the response time based on at least one of monitoring a latency in the TCP protocols, a user's connection speed, network congestion, the timing of a handshake. Moreover, the method may correspond the destination address to a selected address and the response time.

A method of connecting a remote terminal to a server may comprise determining a response time of a remote terminal, communicating the response time to a requested address on the server, selecting a destination address corresponding to the requested address and the response time, and communicating the selected destination address to the remote terminal.

With this variation, response time may, for example, be determined based on at least one of data flow from the remote terminal, connection speed, and network congestion.

The requested address of this method may include a plurality of sub-addresses each of which corresponds to and is optimized for a response time range. The method may additionally or in the alternative comprise connecting the remote terminal to the selected destination address.

An apparatus includes instructions residing on a machine-readable storage medium for use in a machine system to handle a plurality of instructions. The instructions may cause the machine to determine a response time of a remote terminal, communicate the response time to a requested address on the server, select a destination address corresponding to the requested address and the response time, and communicate the selected destination address to the remote terminal.

In some variations, the apparatus instructions may cause the machine to connect the remote terminal to the selected destination address. In this variation and other variations, the response time may be affected by at least one of a connection speed, network congestion, and on the timing of a handshake.

DESCRIPTION OF DRAWINGS

These and other features and advantages of the invention will become more apparent upon reading the following detailed description and upon reference to the accompanying drawings.

Figure 1 is a block diagram of a network server connection using a client interface device according to an embodiment of the present invention.

Figure 2 illustrates a destination address including a plurality of sub-addresses optimized for specific connection speeds according to an embodiment of the present invention.

Figure 3 is a diagram of the exchange between the client, the client interface device, and the server according to one embodiment of the present invention.

Figure 4 is a flow chart of the process used by the client interface device to optimize the connection between the client and the server according to an embodiment of the present invention.

DETAILED DESCRIPTION

Figure 1 is a block diagram of a network server connection 100 using a client interface device 110 according to an embodiment of the present invention. A client 105 communicates with the client interface device 110. The client 105 may be a remote computer connected to a network such as a local area network, a wide area network, the Internet, or the like. For purposes of illustration, the invention will be described herein with a connection to the Internet. The client interface device 110 communicates with the server 115. The client interface device 110 relays information from the client 105 to the server 115 in both directions. The client interface device 110 may also directly connect the client 105 to the server 115.

Figure 2 illustrates a destination group 200 including a destination address 205 and a plurality of sub-addresses 210-225 optimized for specific connection speeds. Web designers are continually placing more and more information on a web page. This allows the web designers to increase the amount of information delivered to the client, and also allows for the creation of a more entertaining website using streaming video, music, and large graphic files. Those clients 105 connecting to the server 115 with a high bandwidth connection can enjoy the full features of the website.

The destination address 205 may be designed for optimal viewing by a client 105 connecting to the server 115 using a connection having 1.5 Mbps or higher. With this connection speed, the client 105 is able to view all of the content of the destination address 205 without undue delays caused by downloading. However, if a slower speed client 105 attempted to connect to the high bandwidth designed destination address 205, the client 105 would require a large amount of time to load the contents of the page. This would result in longer delays and a reduction in the quality of service provided the client 105. Therefore, web designers may create sub-addresses which are designed for slower speed

connections. A first sub-address 210 may be designed for clients connecting at 28.8 Kbps, a second sub-address 215 may be designed for clients connecting at 56.6 Kbps, a third sub-address 220 may be designed for clients connecting at 128 Kbps, and a fourth sub-address 225 may be designed for clients connecting at 512 Kbps. Of course, the number of sub-address may vary site to site. Each of the sub-addresses 210-225 includes a reduced amount of content to allow the site to load within an acceptable period of time under the designated connection speed. To achieve this, the sub-address 210-225 may remove some multimedia content, use reduced resolution, delete some content, or other techniques to reduce the bandwidth necessary. If the destination address 205 is a graphics file, the sub-address 210-225 may be the same graphics file at respectively lower resolutions. The location of the sub-addresses 210-225 may be provided to a client 105 or client interface device 110 attempted to access the destination address 205.

Figure 3 is a diagram of the exchange between the client 105, the client interface device 110, and the server 115 according to one embodiment of the present invention. To initiate a connection to the server 115, the client 105 requests a connection by transmitting a SYN packet 305.

Although the client 105 is sending the SYN packet to the server 115, the SYN packet will be received by the client interface device 110. The client interface device 110 then responds to the client 105 with a SYN/ACK packet 310. The client interface device begins timing upon transmission of the SYN/ACK packet 310. After receiving the SYN/ACK packet 310, the client 105 responds with an ACK 315 back to the client interface device 110. Once the ACK 315 is received, the client interface device 110 stops timing and calculates the elapsed time required for the communication to the client 105. Based on this elapsed time, the client interface device 110 is able to determine a profile for the client 105. The profile may include the data flow rate. Several factors may affect the profile of the client 105, including the connection speed and network congestion.

Once the client 105 has successfully established a connection, the client 105 may issue a GET request 320 to the server 115. The GET request 320 may be requesting a website, a file, or other destination. Once received, the client interface device 110 acknowledges the GET request 320 with an ACK 325 signal.

To complete the GET request 320, the client interface device 110 establishes a connection with the

server 115. The client interface device 110 establishes the connection by transmitting a SYN packet 330 to the server 115. The server 115 receives the SYN packet 330 and responds with a SYN/ACK packet 335. After receiving the SYN/ACK packet 335, the client interface device 110 responds with an ACK 340 back to the server 115.

Now that the connection to the server 115 is established, the client interface device 110 examines the GET request 320 and determine if the client 105 can successfully complete this request in the necessary time frame. The client interface device 110 uses the client profile to determine the timing requirements. The client interface device 110 then creates a GET request based of the GET request 320 of the client 105 and the calculated client profile. If the client 105 is able to handle the original GET request 320, the client interface device 110 will simply forward that as the GET request 345 which is sent to the server 115. However, if the client 105 cannot handle the original GET request 320, the client interface device will determine a revised GET request 345 based on the client profile. For example, the original GET request may be for the destination address 205, but the client 105 may only have a 128 Kbps profile. In this circumstance,

the client interface device 110 will rewrite the request to send the client to the sub-address 220. Thus, the GET request 345 sent to the server 115 will be for the sub-address 220. When the server 115 receives the GET request 345, the server 115 sends an ACK 350 to acknowledge the request.

After the server 115 receives the GET request 345, normal TCP/IP and HTTP protocols facilitate client request fulfillment. During this process, the client interface device 110 may create a more detailed profile of the client connection speed. Of course, the client interface device 110 may also directly connect the client 105 to the server 115.

Figure 4 is a flow chart of the process 400 used by the client interface device 110 to optimize the connection between the client 105 and the server 115 according to an embodiment of the present invention. The process 400 begins in START block 405. Proceeding to block 410, the client interface device 110 determines the response time of the client 105. This allows the client interface device 110 to estimate a connection speed of the client 105.

Proceeding to block 415, the client interface device 110 receives a resource request from the client 105.

The resource request may be for a website, a file, or other data to be transferred to the client 105. The process 400 then proceeds to block 420. In block 420, the client interface device 110 determines if the resource request can be fulfilled based on the measured response time of the client 105. If the request can be fulfilled, the process 400 proceeds along the YES branch to block 425. In block 425, the resource request is transmitted to the server 115.

Returning to block 420, if the resource request cannot be fulfilled, the process 400 proceeds along the NO branch to block 430. In block 430, the client interface device rewrites the resource request to optimize the client loading based on the measured response time. For example, a client 105 may request a website www.website.com which is optimized for connection using 1.5 Mbps. However, the measured response time may indicate the client 105 has a response time of 56 Kbps. In this case, the client interface device 110 may rewrite the request to www.website.com/56 which is the same website content optimized for 56 Kbps.

After the request or rewritten request is sent to the server 115, the process 400 proceeds to block 435. In block 435, the client interface device 110 connects the server 115 to the client 105. During the connection, the

client interface device 110 may act as a conduit, continually measuring the client response time or may directly connect the server 115 to the client 105. The process 400 then terminates in END block 440.

Numerous variations and modifications of the invention will become readily apparent to those skilled in the art. Accordingly, the invention may be embodied in other specific forms without departing from its spirit or essential characteristics.

CLAIMS

1. A method of enhancing data delivery comprising:

sending a first packet from a client interface to a remote terminal at a first time;

receiving at the client interface a second packet from the remote terminal at a second time;

determining a response time of the remote terminal at the client interface based on a time period elapsing between the first time and the second time;

using said response time to determine a connection speed between the remote terminal and the client interface;

providing a plurality of destination addresses, said destination addresses relating to different amounts of content at a server coupled to the client interface, each destination address being optimized for a specific connection speed;

based on said determined connection speed, automatically selecting a destination address from said plurality of destination addresses, said selected destination address being optimized for said determined connection speed; and

connecting the remote terminal to the selected destination address.

2. The method of Claim 1, further comprising determining a data flow rate from the determined response time of the remote terminal, wherein determining the response time comprises:

starting a timer at the first time when the client interface sends the first packet to the remote terminal; and

stopping the timer at the second time when the client interface receives the second acknowledgement packet from the remote terminal.

3. The method of Claim 1, further comprising determining the response time by monitoring a latency in the TCP protocols.

4. The method of Claim 1, further comprising determining network congestion based on the determined response time.

5. The method of Claim 1, further comprising determining the response time based on the timing of a handshake between the remote terminal and the client interface.

6. The method of Claim 1, wherein selecting the destination address from a plurality of addresses is based on a requested address by the remote terminal and the determined response time.

7. A method of connecting a remote terminal to a server comprising:

sending a first packet from a client interface to the remote terminal;

receiving at the client interface a second packet from the remote terminal;

determining a response time of the remote terminal at the client interface based on a time period between the first packet being sent and the second packet being received;

using said response time to determine a connection speed between the remote terminal and the client interface;

providing a plurality of destination addresses, said destination addresses relating to different amounts of content at a server coupled to the client interface, each destination address being optimized for a specific connection speed;

receiving a request for a destination address;

based on said determined connection speed, selecting a destination address corresponding to the request; and

communicating data from the selected destination address to the remote terminal.

8. The method of Claim 7, further comprising determining a data flow rate from the remote terminal based on the response time.

9. The method of Claim 7, wherein the requested destination address includes a main destination address and a plurality of sub-addresses, each of said sub-addresses corresponding to a connection speed and optimized for said connection speed.

10. The method of Claim 7, further comprising determining a network congestion based on the determined response time.

11. The method of Claim 7, further comprising connecting the remote terminal to the selected destination address.

12. An apparatus, including instructions residing on a machine-readable storage medium, for use in a machine system to handle a plurality of instructions, the instructions causing the machine system to:

send a first packet from a client interface to the remote terminal;

receive at the client interface a second packet from the remote terminal;

determine a response time of the remote terminal at the client interface based on a time period between the

first packet being sent and the second packet being received;

use said response time to determine a connection speed between the remote terminal and the client interface;

provide a plurality of destination addresses, said destination addresses relating to different amounts of content at a server coupled to the client interface, each destination address being optimized for a specific connection speed;

receive a request for a destination address;

based on said determined connection speed, select a destination address corresponding to the request; and

communicate the selected destination address to the remote terminal.

13. The apparatus of Claim 12, wherein the instructions further cause the machine system to connect the remote terminal to the selected destination address.

14. The apparatus of Claim 12, wherein the response time is affected by network congestion.

15. The apparatus of Claim 12, wherein the response time is determined based on the timing of a handshake between the remote terminal and the client interface.

16. The method of Claim 1, further comprising:

receiving a request from the remote terminal for a destination address at the server; and

rewriting the request to include the selected destination address optimized for the determined connection speed of the remote terminal.