US007139815B2

(12) United States Patent
Muti et al.

(10) Patent No.: US 7,139,815 B2
(45) Date of Patent: *Nov. 21, 2006

(54) SYSTEM AND METHOD FOR TRANSFERRING DATA OVER A NETWORK

(75) Inventors: **Alessandro Muti**, Redmond, WA (US); **Darshatkumar A. Shah**, Bellevue, WA (US); **Brian Joseph Moore**, Redmond, WA (US)

(73) Assignee: **Microsoft Corporation**, Redmond, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **11/005,998**

(22) Filed: **Dec. 7, 2004**

(65) **Prior Publication Data**

US 2005/0086364 A1 Apr. 21, 2005

Related U.S. Application Data

(63) Continuation of application No. 09/505,735, filed on Feb. 16, 2000.

(51) **Int. Cl.**
G06F 15/16 (2006.01)

(52) **U.S. Cl.** **709/219**; 709/232

(58) **Field of Classification Search** 709/217–219, 709/223–226; 717/168–178; 713/200–202; 726/26
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,432,781 A 7/1995 Nimon
5,434,848 A 7/1995 Chimento, Jr. et al.
5,465,331 A * 11/1995 Yang et al. 709/244
5,490,168 A * 2/1996 Phillips et al. 375/224
5,638,360 A 6/1997 Sugawara
5,708,779 A * 1/1998 Graziano et al. 709/250
5,898,673 A 4/1999 Riggan et al.
5,913,040 A 6/1999 Rakavy et al.
6,021,114 A * 2/2000 Shaffer et al. 370/232
6,078,591 A 6/2000 Kalkunte
6,078,919 A 6/2000 Ginzburg et al.
6,209,033 B1 3/2001 Datta et al.
6,272,110 B1 8/2001 Tunnicliffe et al.
6,285,662 B1 9/2001 Wattanabe et al.

(Continued)

FOREIGN PATENT DOCUMENTS

EP 0 782 302 A 7/1997

(Continued)

OTHER PUBLICATIONS

U.S. Appl. No. 09/505,735, filed Feb. 16, 2000, Muti et al.

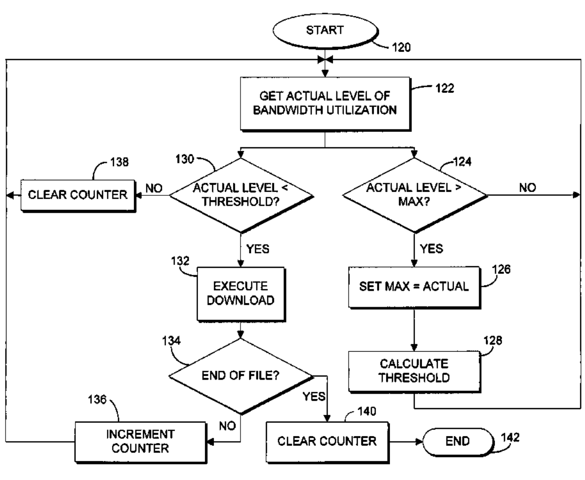
(Continued)

Primary Examiner—David Wiley
Assistant Examiner—Joseph E. Avellino
(74) *Attorney, Agent, or Firm*—Shook, Hardy & Bacon L.L.P.

(57) **ABSTRACT**

A system and method are provided for transferring a set of data over a network by monitoring the level of actual network bandwidth utilization. The method identifies a maximum monitored level of actual utilization and calculates a threshold level of utilization as a function of the maximum monitored level of utilization. If the actual level of utilization is less than the threshold level, at least a portion or segment of the data is received over the network. Each time a portion of the data is received, the monitoring step is repeated and a counter is incremented. The size of the discrete portions of the data is a function of the value of the counter and is increased when the value of the counter is greater than a predetermined value.

31 Claims, 4 Drawing Sheets



U.S. PATENT DOCUMENTS

6,427,169	B1	7/2002	Elzur	
6,442,263	B1 *	8/2002	Beaton et al.	379/142.04
6,463,468	B1	10/2002	Buch	
6,675,214	B1 *	1/2004	Stewart et al.	709/226
6,704,873	B1 *	3/2004	Underwood	726/12
6,711,137	B1 *	3/2004	Klassen et al.	370/252
6,816,456	B1 *	11/2004	Tse-Au	370/230.1
6,823,385	B1 *	11/2004	McKinnon et al.	709/226
6,920,110	B1 *	7/2005	Roberts et al.	370/235
7,016,951	B1 *	3/2006	Longworth et al.	709/224
2002/0110149	A1 *	8/2002	Roberts et al.	370/477

FOREIGN PATENT DOCUMENTS

EP	0 862 304 A	9/1998
WO	WO 00/01123	1/2000

OTHER PUBLICATIONS

Floyd, S., et al., "Link-Sharing and Resource Management Models for Packet Networks," IEEE/ACM Transactions on Networking, pp. 365-386, vol. 3, No. 4, Aug. 1995.

* cited by examiner

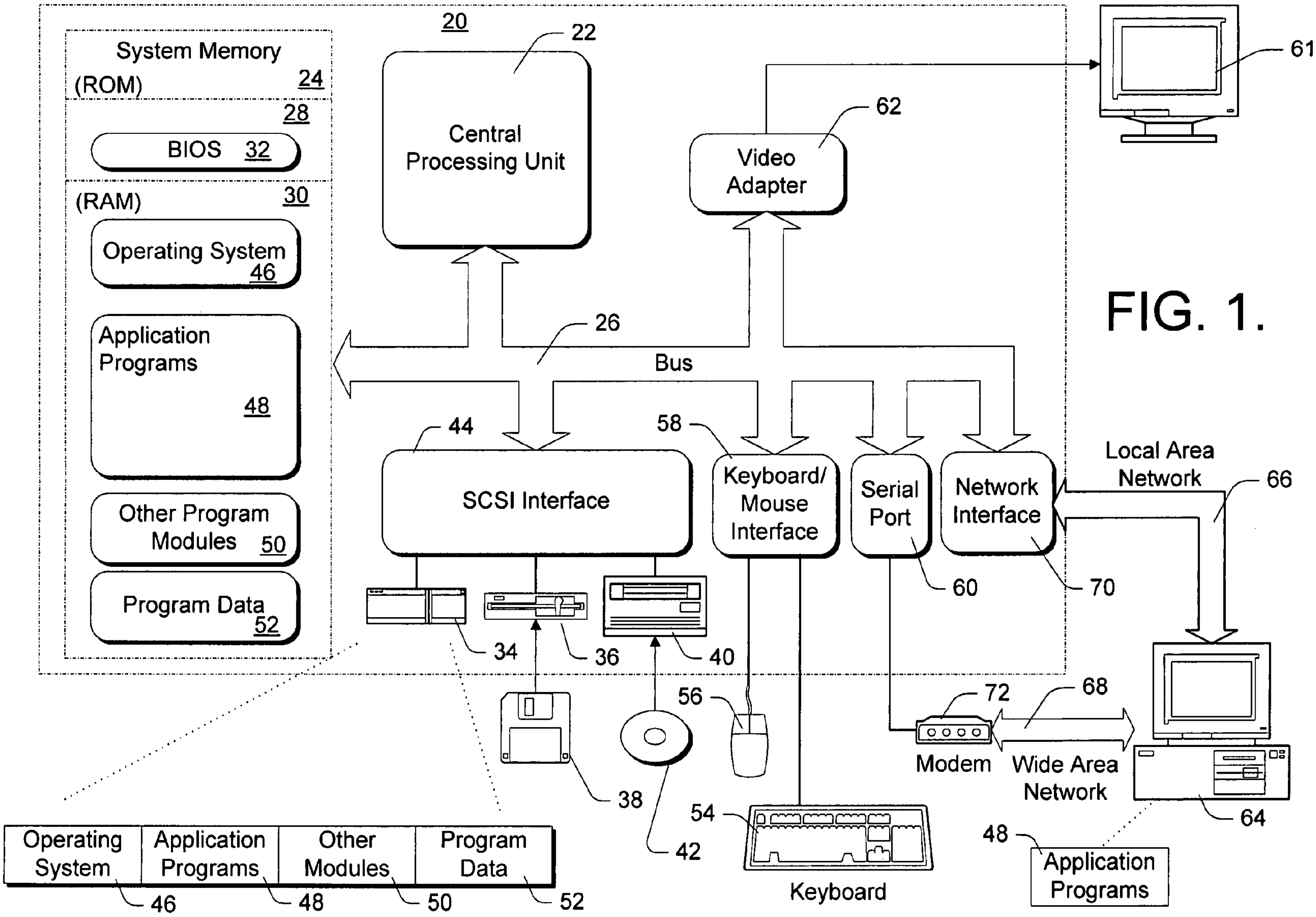

20
System Memory
(ROM) 24
28
BIOS 32
(RAM) 30
Operating System 46
Application Programs 48
Other Program Modules 50
Program Data 52
22
Central Processing Unit
62
Video Adapter
61
26
Bus
44
SCSI Interface
58
Keyboard/ Mouse Interface
Serial Port
Network Interface
Local Area Network
66
60
70
34
36
40
38
42
56
54
Keyboard
72
Modem
68
Wide Area Network
64
48
Application Programs
Operating System
Application Programs
Other Modules
Program Data
46
48
50
52

FIG. 1.

FIG. 2.

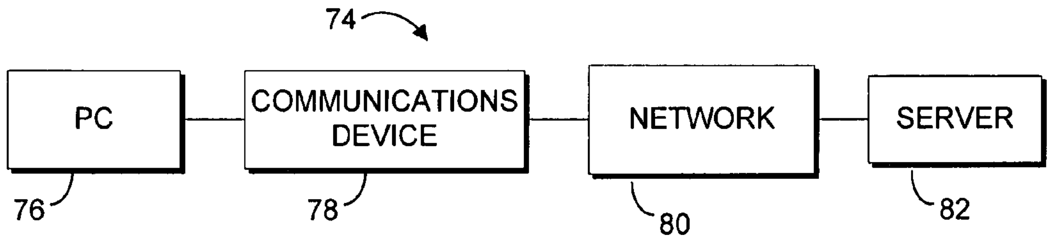


FIG. 3.

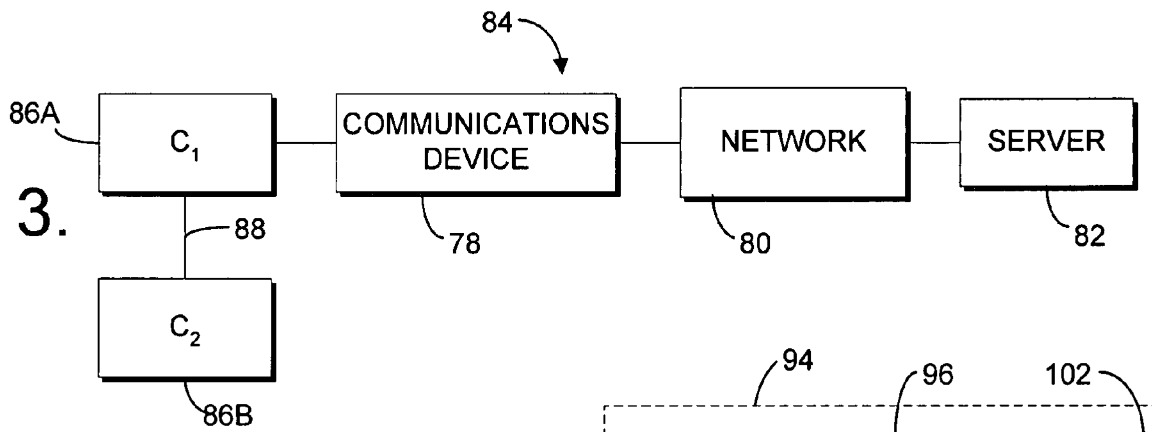


FIG. 5.

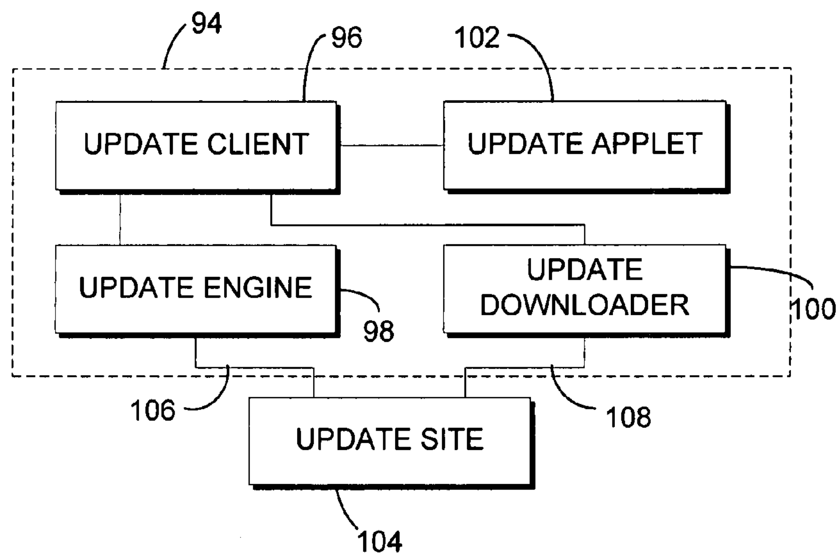


FIG. 6.

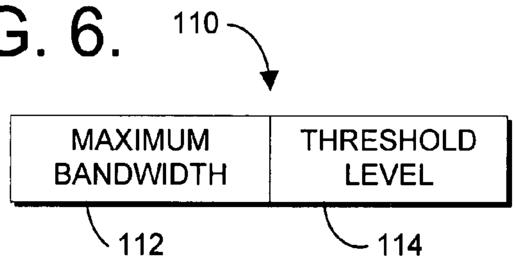


FIG. 4.

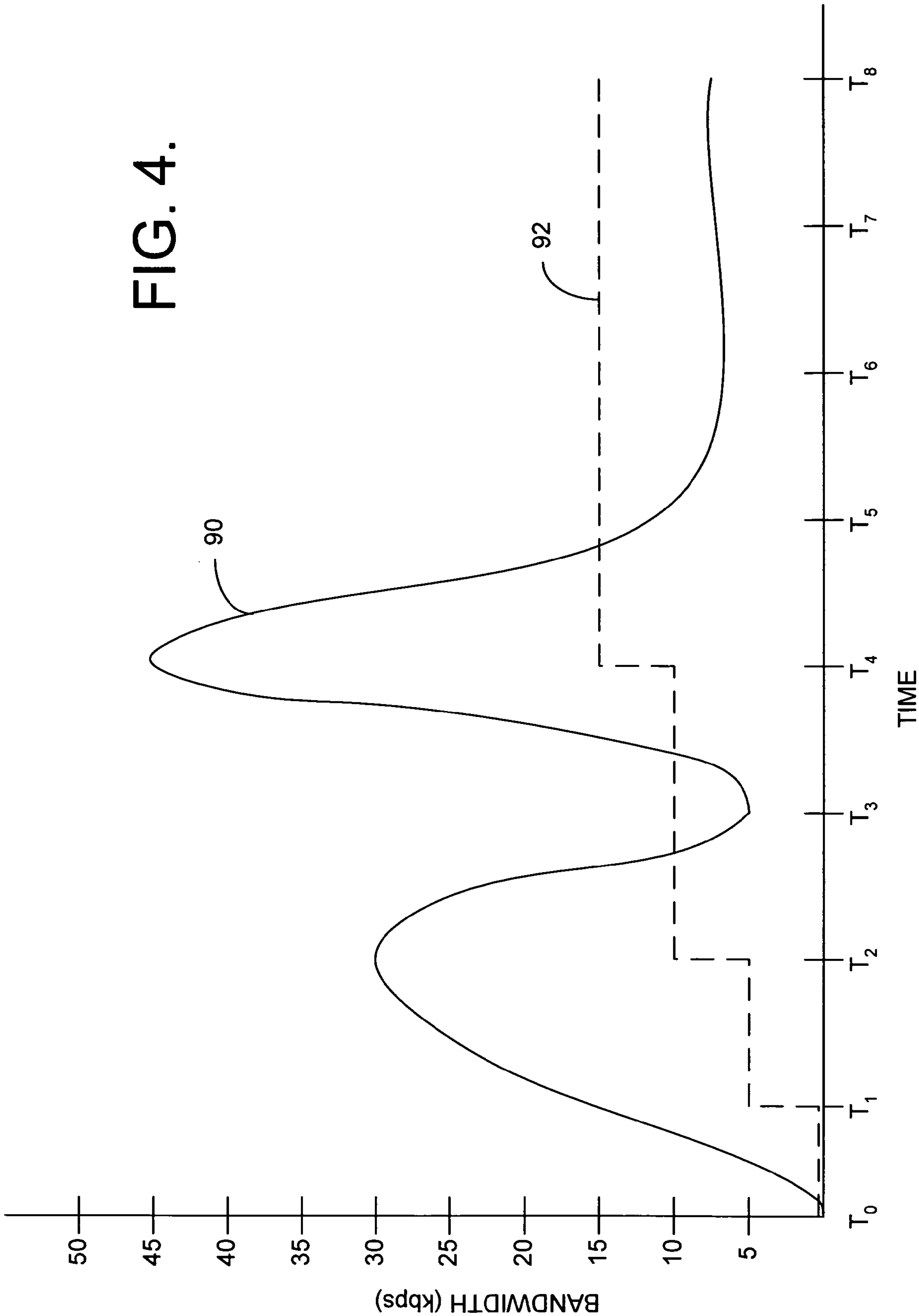
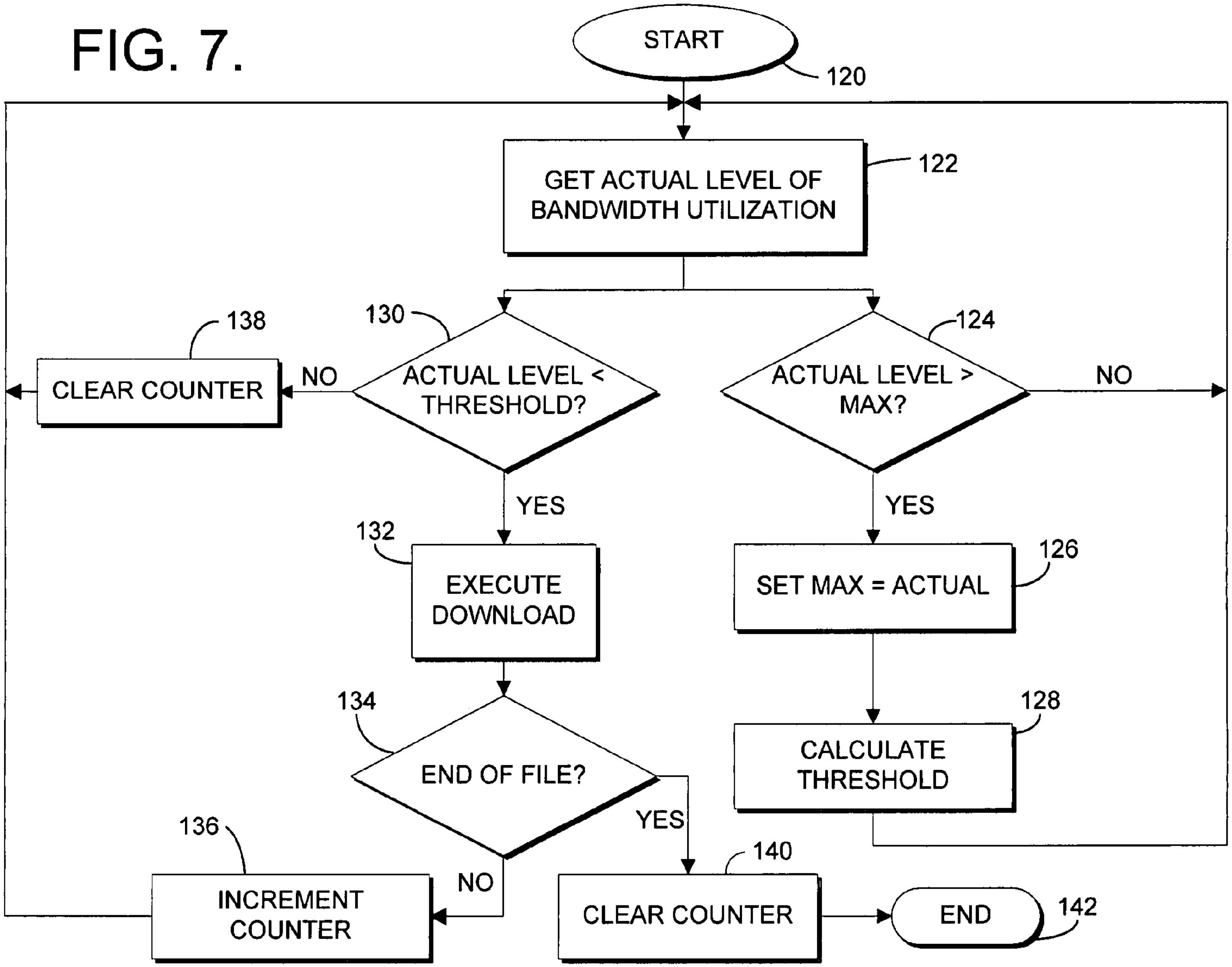

BANDWIDTH (kbps)
50
45
40
35
30
25
20
15
10
5
TIME
T_0
T_1
T_2
T_3
T_4
T_5
T_6
T_7
T_8
90
92

FIG. 7.


START
120
GET ACTUAL LEVEL OF BANDWIDTH UTILIZATION
122
130
ACTUAL LEVEL < THRESHOLD?
124
ACTUAL LEVEL > MAX?
NO
138
CLEAR COUNTER
NO
YES
YES
132
EXECUTE DOWNLOAD
SET MAX = ACTUAL
126
134
END OF FILE?
128
CALCULATE THRESHOLD
YES
136
INCREMENT COUNTER
NO
140
CLEAR COUNTER
END
142

SYSTEM AND METHOD FOR TRANSFERRING DATA OVER A NETWORK

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of application Ser. No. 09/505,735, filed Feb. 16, 2000.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not applicable.

TECHNICAL FIELD

The present invention relates to computer software and, more particularly, to a system and method for transferring data over a network such as the Internet.

BACKGROUND OF THE INVENTION

A computer system in which one or more client machines communicate with one or more servers over a network is a common arrangement. For example, there are millions of personal computers (PCs) connected to the Internet for communication with various servers. Many of these PC users acquire new or updated software on their PCs by downloading the software from a remote server over the Internet.

Conventional methods of downloading new and updated software to a PC over a network such as the Internet interfere with the PC user's ability to browse the Internet or to otherwise communicate over the network during the download. With increased processing speeds for PCs, it is the bandwidth to the Internet which is becoming a significant bottleneck to the user's browsing experience. Because conventional downloads typically occupy 100% of the user's available communication bandwidth until the download is complete, the user is unable to utilize the network for other tasks during the download. Moreover, if the download is interrupted for some reason (e.g., the network connection is lost), it may be necessary to restart the download from the beginning of the file, thus wasting significant network resources. Other conventional downloading software permits the user to schedule the download for a designated time. In any event, conventional methods for downloading software over a network such as the Internet severely limit the user's ability to engage in other network activity.

Providing software updates is beneficial to PC users because it corrects problems and improves the performance of their PCs. Additionally, the software vendor benefits from a reduction in the number of product support calls, which in turn reduces the substantial resources the vendor might otherwise allocate to handling such calls.

Microsoft Corporation provides a software update service in connection with its WINDOWS 98 operating system under the mark "WINDOWS UPDATE". By accessing the "WINDOWS UPDATE" Internet website, WINDOWS 98 users can have their system evaluated and download both critical and non-critical software updates (e.g., a fix or a patch) over the Internet. Upon determining there are updates available that are not already loaded on the user's PC, the user is notified that such updates are available, and the user is prompted to select one or more of them for downloading over the Internet. While this is a convenient method for providing software updates to users, such downloads could

interfere with other network activity. Moreover, it is likely that many users will not take the initiative to navigate to the "WINDOWS UPDATE" website and download operating system updates for their PCs. Consequently, users who have not yet downloaded critical updates are more likely to encounter problems with their PCs, and they will likely call the vendor's product support line when a problem arises.

Accordingly, there is a need for an effective method for downloading software updates over a network so that interference with other network activity is minimized. One potential solution is simply to increase network bandwidth. However, increased bandwidth is expensive and has historically been followed by increased user demands. Thus, even as available bandwidth increases with technological advances, there will continue to be a need to eliminate interference with other network activity because of the corresponding increases in user demands.

SUMMARY OF THE INVENTION

The present invention is directed to a system, method, and one or more computer-readable media for transferring a set of data over a network. In one aspect, an exemplary embodiment of the invention is directed to a method. The method includes the steps of monitoring the level of actual network bandwidth utilization and identifying a maximum monitored level of actual utilization. Then, the method calculates a threshold level of utilization as a function of the maximum monitored level of utilization. If the actual level is less than the threshold level, a number of discrete portions of the set of data are separately received over the network. The monitoring step is repeated and a counter is incremented each time a discrete portion of the set of data is received. The size of the discrete portions of the data may be a function of the value of the counter. The method increases the size of the discrete portions of data when the value of the counter is greater than a predetermined value.

In another aspect, an exemplary embodiment of the invention takes the form of one or more computer-readable media having computer-useable instructions embodied thereon for performing a method for transferring a set of data over a network. The method includes the steps of monitoring the level of actual network bandwidth utilization and identifying a maximum monitored level of actual utilization. The method then calculates a threshold level of utilization as a function of the maximum monitored level of utilization. If the actual level is less than the threshold level, a number of discrete portions of the set of data are separately received over the network. The monitoring step is repeated and a counter is incremented each time a discrete portion of the data is received over the network. The size of the discrete portions of the data is a function of the value of the counter and is increased when the value of the counter is greater than a predetermined value.

Yet another aspect of the invention is directed to a system for transferring a set of data over a network. The system includes a monitoring component, an identifying component, a calculating component, a receiving component, and a counter. The monitoring component monitors the level of actual network bandwidth utilization. The identifying component identifies a maximum monitored level of actual utilization. The calculating component calculates a threshold level of utilization as a function of the maximum monitored level of utilization. The receiving component separately receives a number of discrete portions of the set of data over the network if the actual level is less than the threshold level. The counter is incremented each time a discrete portion of the data is received over the network. The size of the discrete portions of the data is a function of the value of the counter and is increased when the value of the counter is greater than a predetermined value.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

The present invention is described in detail below with reference to the attached drawing figures, wherein:

FIG. **1** is a block diagram of a suitable computing system environment for use in implementing the present invention;

FIG. **2** is a block diagram of a system including a client machine which communicates with a server over a network;

FIG. **3** is a block diagram similar to FIG. **2** but with first and second client machines connected to one another via an intranet;

FIG. **4** is a diagram illustrating network bandwidth utilization over time with a curve representing the actual network bandwidth utilization and a broken line representing a threshold level of utilization calculated in accordance with a preferred embodiment of the present invention;

FIG. **5** is a block diagram illustrating the architecture of a client machine in accordance with a preferred embodiment of the present invention;

FIG. **6** is a block diagram of a data structure according to a preferred embodiment of the present invention; and

FIG. **7** is a flowchart representative of a computer program for transferring data over a network in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides a system and method for transferring data over a network such as the Internet. FIG. **1** illustrates an example of a suitable computing system environment in which the invention may be implemented. The computing system environment is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment.

The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

The invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

With reference to FIG. **1**, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer **20**. Components of computer **20** include, but are not limited to, a processing unit **22**, a system memory **24**, and a system bus **26** that couples various system components including the system memory to the processing unit **22**. The system bus **26** may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

Computer **20** typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer **20** and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer **20**. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

The system memory **24** includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) **28** and random access memory (RAM) **30**. A basic input/output system **32** (BIOS), containing the basic routines that help to transfer information between elements within computer **20**, such as during start-up, is typically stored in ROM **28**. RAM **30** typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit **22**. By way of example, and not limitation, FIG. **1** illustrates operating system **46**, application programs **48**, other program modules **50**, and program data **52**.

The computer **20** may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, FIG. **1** illustrates a hard disk drive **34** that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive **36** that reads from or writes to removable, nonvolatile magnetic disk **38**, and an optical disk drive **40** that reads from or writes to a removable, nonvolatile optical disk **42** such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital video disks, digital video tape, Bernoulli cartridges, solid state RAM, solid state ROM, and the like. The hard disk drive **34**, magnetic disk drive **36**, and optical disk drive **40** are typically connected to the system bus **26** by a Small Computer System Interface (SCSI) **44**. Alternatively, the hard disk drive **34**, magnetic disk drive **36** and optical disk drive **40** may be connected to the system bus **26** by a hard disk drive interface, a magnetic disk drive interface, and an optical drive interface, respectively.

The drives and their associated computer storage media discussed above and illustrated in FIG. **1**, provide storage of computer readable instructions, data structures, program modules and other data for the computer **20**. In FIG. **1**, for example, hard disk drive **34** is illustrated as storing operating system **46**, application programs **48**, other program modules **50**, and program data **52**. Note that these components can either be the same as or different from operating system **46**, application programs **48**, other program modules **50**, and program data **52**. A user may enter commands and information into the computer **20** through input devices such as a keyboard **54** and pointing device **56**, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit **22** through a user input interface **58** or a serial port interface **60** that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor **61** or other type of display device is also connected to the system bus **26** via an interface, such as a video adapter **62**. In addition to the monitor **61**, computers may also include other peripheral output devices such as speakers and printers, which may be connected through an output peripheral interface.

The computer **20** may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer **64**. The remote computer **64** may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer **20**, although only a memory storage device has been illustrated in FIG. **1**. The logical connections depicted in FIG. **1** include a local area network (LAN) **66** and a wide area network (WAN) **68**, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the computer **20** is connected to the LAN **66** through a network interface or adapter **70**. When used in a WAN networking environment, the computer **20** typically includes a modem **72** or other means for establishing communications over the WAN **68**, such as the Internet. The modem **72**, which may be internal or external, may be connected to the system bus **26** via the serial port interface **60** or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer **20**, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. **1** illustrates remote application programs **48** as residing on memory device **64**. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Although many other internal components of the computer **20** are not shown, those of ordinary skill in the art will appreciate that such components and the interconnection are well known. Accordingly, additional details concerning the internal construction of the computer **20** need not be disclosed in connection with the present invention.

Those skilled in the art will understand that program modules such as the operating system **46**, application programs **50** and data **52** are provided to the computer **20** via one of its memory storage devices, which may include ROM **28**, RAM **30**, hard disk drive **34**, magnetic disk drive **36** or optical disk drive **40**. Preferably, the hard disk drive **34** is used to store data **52** and programs, including the operating system **46** and application programs **48**.

When the computer **20** is turned on or reset, the BIOS **32**, which is stored in the ROM **28** instructs the processing unit **22** to load the operating system from the hard disk drive **34** into the RAM **30**. Once the operating system **46** is loaded in RAM **30**, the processing unit **22** executes the operating system code and causes the visual elements associated with the user interface of the operating system **46** to be displayed on the monitor **61**. When an application program **48** is opened by a user, the program code and relevant data are read from the hard disk drive **34** and stored in RAM **30**.

With reference to FIG. **2**, an exemplary system for practicing the present invention is designated generally by reference numeral **74**. The system **74** includes a client machine such as a PC **76** and a communications device **78** such as a modem for accessing a network **80** (e.g., the Internet). A server **82** is also connected to the network **80** in a conventional manner. The PC **76** and modem **78** establish a connection with the server **82** over the network **80** to facilitate the transfer of data between the client and the server.

A system **84** set forth in FIG. **3** is similar to the system **74** of FIG. **2**, except that the system **84** includes a first client **86**A and a second client **86**B which are connected to one another over a link **88** such as an intranet. As with the system **74**, the communications device **78** (e.g., a modem) establishes a connection between the first client **86**A and the network **80**. Accordingly, the clients **86**A, **86**B can exchange data with the server **82** over the network **80**.

Those skilled in the art will readily appreciate that the present invention encompasses many possible variations to the systems **74**, **84** shown in FIGS. **2** and **3**. Of course, any number of client and server machines may be connected to the network **80** (either directly or indirectly). For example, the system **84** could include more than two client machines, one of which would act as a proxy for the others. In that case, the client machines could be connected to one another over an intranet, and the network card in the proxy machine could be connected to a router which is linked to the network **80** over a **T1** line. Moreover, the network **80** could be the Internet or any other public or private network. Additionally, the communications device **78** may be part of the PC **76** or an independent hardware device.

In one application of the present invention, the client machine is the PC **76**, the device **78** is a 56 Kbps (Kilobits per second) modem, the network **80** is the Internet, and the server **82** is a website accessible over the Internet for downloading software updates for the operating system of the PC. Referring to FIG. **4**, a curve **90** represents the level of actual network bandwidth utilization for a client PC accessing the Internet over a period of time. The curve **90** rises from T_0 through T_2 to reflect the flurry of network activity which typically occurs as the user initially accesses the Internet. Then, after a brief decrease in network activity at T_3, the client's actual utilization again rises dramatically to a peak at T_4. This peak could represent any number of possible actions initiated by the client, such as the downloading of a webpage or the opening of an email message. While the user at the client machine reads the web page or the email message, or perhaps listens to music over the Internet, the actual network utilization decreases to a significantly lower level from T_5 through T_8.

The relatively low level of actual network bandwidth utilization shown from T_5 through T_8 (FIG. **4**) is sometimes referred to as "network idle." This concept differs from "machine idle," which occurs when a PC user is not currently using the keyboard or mouse. If the machine remains idle for a period of time, a screen saver may be invoked. However, network utilization may be high at the same time the machine is idle (e.g., during a download), and the network interface may be "idle" when the machine is not. Thus, the present invention is directed to the transfer of a set of data, such as a software update, over a network at a time when the network utilization is relatively low. This transfer of data is intended to be transparent to the user, and the user's machine need not be idle during the transfer.

As described below in connection with FIG. **7**, the present invention monitors the level of actual utilization represented by the curve **90** (FIG. **4**) and calculates a threshold level of utilization below which the transfer of data between the client and server is not likely to interfere with other network activity. In a preferred embodiment, the threshold level is equal to approximately 33% of the maximum detected level of actual network utilization. As shown in FIG. **4**, the threshold level is generally designated by reference numeral **92** and is initially set to zero. Network activity is periodically sampled, and a new threshold level is calculated each time a new maximum is identified. At T_1, the level of actual bandwidth usage is 15 Kbps, which is a new maximum, so the threshold **92** is set to 5 Kbps. The level of actual bandwidth usage is again sampled at T_2, and another new maximum of 30 Kbps is identified. Consequently, the new threshold **92** is calculated to be 10 Kbps.

When the level of actual usage **90** drops below the threshold level **92**, as at T_3, a brief transfer of data should not significantly interfere with other network activity. The transfer of data could be initiated immediately upon first detecting that the actual usage is below the threshold level. However, a more conservative approach, in that it is more deferential to the user, is to detect at least two consecutive occurrences of low usage before initiating a download. In any event, a new maximum of 45 Kbps is detected at T_4, thereby triggering the calculation of an updated threshold level of 15 Kbps. Consequently, if any data transfer were initiated at T_3, then it would be immediately suspended at T_4. At T_5, the actual usage **90** once again drops below the threshold level **92**. This time, the level of actual usage **90** remains between 5–10 Kbps, which is less than the threshold level of 15 Kbps, from T_5 to T_8. Thus, the transfer of data over the network would not interfere with other network activity during this time period.

The preferred architecture of the client of the present invention is set forth in FIG. **5**. A client **94** includes an update client **96**, an update engine **98**, an update downloader **100** and an update applet **102**. The client **94** is connected to a server such as via an update website **104** over a pair of links **106**, **108**, which are preferably HTTP (Hypertext Transfer Protocol) links. The update client **96** is responsible for notifications and personalization options for the user. The update engine **98** controls content interaction with the update site **104** as well as detection, installation and logging of software updates. The update downloader **100** is responsible for implementing the packet background downloading/throttling and incoming packet queue management. The update applet **102** determines the configuration of update notifications to the user and other interactions with the user. The backend content and catalogs are available from the update site **104**.

The present invention utilizes a data structure **110** illustrated in FIG. **6**. The data structure **110** includes a first field **112** which contains the maximum identified level of actual bandwidth utilization and a second field **114** which contains the threshold level of utilization. As shown in FIG. **4**, both the maximum identified level and the threshold level can vary over time depending on the level of actual utilization. As mentioned above, the threshold level is calculated as a function of the maximum identified level of actual usage. Therefore, a new threshold level is calculated each time a new maximum level is detected.

FIG. **7** sets forth a flowchart representative of a computer program for selectively transferring data over a network in accordance with a preferred embodiment of the present invention. The program is started at step **120** when certain predetermined conditions are present. Typically, the client PC should be turned "on" and connected to the network before starting the program. While it would be possible for the program to establish the connection between the PC and the network, some PC users may be startled by the unexpected sound of their modem dialing up the network on its own. It is also possible that the client may be connected to another network (e.g., a private network) so that access to a webserver is not possible until the client disconnects from the private network and reconnects to the Internet. It can be confirmed that the client and server are connected to the same network by "pinging" the server and receiving an acknowledgment from the server. Those skilled in the art can readily determine which preliminary conditions would be appropriate in a given situation.

Once the client is connected to the same network as the server, the level of actual network bandwidth utilization is obtained at step **122**. As shown in FIG. **4**, the actual usage is sampled periodically (e.g., every five seconds). Preferably, this is done by tracking the number of incoming and outgoing packets over a given time period (e.g., one second) at the communications interface between the client and the network. For a PC running the "WINDOWS 98" operating system, this information is available using the "iphlpapi" function call.

Relatively large files provide a more accurate basis for measuring the level of actual network bandwidth usage at the network interface. For example, it is difficult to accurately measure the download time for a 200 byte file because the overhead of measurement is significant and the small file does not saturate the available bandwidth. Consequently, the present invention preferably updates the maximum level of utilization only when the measurement of the actual level of utilization is based on the transfer speed of a file that is at least 4 KB in size.

Preferably, each of the "virtual devices" is enumerated to ensure accurate monitoring, especially in a system which is more complex than the system **74**. The term "virtual device," as used herein, refers to any networking device with a device driver, such as modems (including cable modems and ISDN modems), network cards and DSLs (Digital Subscriber

Lines). Each channel of a 2-channel ISDN modem is considered to be a separate virtual device for the purposes of the present invention.

Importantly, the level of actual network bandwidth utilization is monitored rather than simply relying on the rated or listed throughput of the network device. The rated value is not reliable because, as a practical matter, the maximum throughput achieved by a modem is less than its listed throughput and may be different for each network session. Variables such as the quality of the telephone connection typically result in a maximum available bandwidth of less than 50 Kbps for a 56 Kbps modem.

Moreover, rated values cannot be assumed to be accurate when a client is connected to the network by a proxy. In the system **84**, there is a peer-to-peer connection between the first client **86**A and the second client **86**B over the intranet **88**, and the first client **86**A is in turn linked to the network **80** by a modem **78**. Assuming a 56 Kbps modem and a 10 Mbps intranet, the modem **78** is the bottleneck to the network **80**. Thus, if the first client **86**A were utilizing 40 Kbps of the network bandwidth at the same time the second client **86**B is utilizing only 5 Kbps of the bandwidth, there would not be sufficient network bandwidth available to download a set of data (e.g., a software update) from the server **82** to either client without degrading their other network activity. Moreover, it would not be accurate to assume either that the second client **86**B has a maximum throughput of 10 Mbps or that substantial bandwidth is available because the second client **86**B is only using 5 Kbps of the network interface. Here, in the case of remote access, the level of actual bandwidth utilization for the second client **86**B can be obtained from the RAS (Remote Access Service) device table. Thus, monitoring the actual throughput at each virtual device advantageously adapts to configuration changes between sessions and is therefore a more reliable approach.

Referring again to FIG. **7**, the maximum detected level of actual usage is initially set to zero. Then, at step **124**, it is determined whether the actual usage is greater than the current maximum. If not, the actual usage is sampled again at step **122**. However, if the actual usage is greater than the current maximum, then a new maximum is set equal to the current actual level at step **126**. Next, a threshold level is calculated as a function of the maximum at step **128**. The threshold is initially set to zero, and a new threshold is calculated each time a new maximum is detected. As a general proposition, the threshold level could be anywhere from approximately 5%-80% of the maximum detected level. However, the presently preferred threshold level for Internet applications is approximately 33% of the maximum detected level. Once the new threshold level has been calculated, the actual usage is sampled again at step **122**.

It should be noted that the present invention calculates the threshold level as a percentage of actual network bandwidth in Kilobits per second, as opposed to some other parameter, such as the percentage of time the network connection is busy over a given period. For example, a client machine utilizing its network connection 75% of the time over the course of an hour is not necessarily using 75% of its available bandwidth. In fact, if the user at the client PC is merely reading a webpage, reading email, listening to music or the like, it is likely that the level of actual bandwidth utilization is well below the maximum available bandwidth. Those skilled in the art will understand that the present invention, which is implemented at the client level, is fundamentally different from methods for limiting bandwidth at the hardware level (e.g., sockets).

Each time the level of actual network bandwidth utilization is obtained at step **122**, it is also compared to the current threshold level. At step **130**, it is determined whether the actual usage is less than the threshold level. If so, a software update (or other set of data) available from the server can be downloaded over the network to the client. To minimize interference with any other present or future network activity of the client, the download is preferably performed by downloading the software update in several segments. Accordingly, at step **132**, one segment of the file is downloaded from the server to the client. If the client's network activity increases after downloading the first segment, the download of the entire file can be suspended until the actual usage drops back down below the threshold level. However, if the network is idle for an extended period of time, the download can be accelerated by downloading progressively larger segments of the file over the network.

A counting device is provided in a preferred embodiment of the present invention to facilitate the downloading of progressively larger file segments. For example, a 144 KB file may be downloaded from the server to the client over the Internet by first transferring an 8 KB segment of the file at step **132** of FIG. **7**. A small segment such as 8 KB has been found to be an effective default because it uses 100% of the pipeline for a short period of time and permits frequent monitoring of actual usage between segments. Even if a user clicks on an URL (Uniform Resource Locator) during the download of an 8 KB block, there will be no noticeable delay to the user. Moreover, in the event an interruption occurs, 8 KB is the most data that can be lost.

At step **134**, it is determined whether the entire file has been downloaded. Since the 8 KB block was the first segment of the 144 KB file, the entire file has not been downloaded and the counter, which is initially set to zero, is incremented to "1" at step **136**. The counter keeps track of the number of segments that have been downloaded consecutively without interruption so that the size of the segments can be increased so long as network activity remains low. As an example, the size of the segments could be increased to 16 KB if the counter is equal to "2", and the size of the segments could be increased to 32 KB if the counter is equal to "4". This would enable the 144 KB file to be downloaded in seven progressively larger segments as opposed to 18 of the smaller 8 KB segments, provided network activity stays below the threshold level. Although the larger segments take longer to download and therefore result in less frequent monitoring of the actual usage, this approach takes advantage of any extended periods of low network activity. Typically a header accompanies each segment, so transferring a smaller number of larger segments also reduces network overhead by reducing the total number of headers transferred over the network. Presently, a 64 KB segment is the largest segment that is preferred when using a 56 Kbps modem.

After incrementing the counter at step **136**, steps **122** and **130** are repeated. If the level of actual usage is no longer less than the threshold level, the file download is suspended and the counter is cleared (i.e., set to zero) at step **138**. Then, the actual utilization is sampled again at step **122**. In the event the download is suspended, the download will be resumed from the point of suspension. However, if the actual usage remains below the threshold level, the next segment is downloaded at step **132**, and the process is repeated until it is determined at step **134** that the entire file has been downloaded. If so, the counter is cleared at step **140** and the loop ends at step **142**.

It should be noted that the progressive download feature of the present invention may require filtering or a similar technique to overcome the limitations of some existing servers. While progressive downloads have been widely available at the FTP (file transfer protocol) level, they were not available at the HTTP level until HTTP 1.1 servers became available. As discussed above, the present invention utilizes a byte range approach with 8 KB file segments as a default. For the first segment, a byte range of "0–7999" is specified in the download request from the client, the range "8000–15999" is specified in the download request for the second segment, and so on. However, the byte range portion of the request would be lost an HTTP 1.0 server is encountered, and the entire file would be downloaded at once. One way to avoid this problem is to use an ISAPI (Internet Server Application Programming Interface) filter to simulate an HTTP 1.1 byte request across an HTTP 1.0 proxy. Then, even if an HTTP 1.0 server is encountered, the byte range information will not be lost.

Of course, the preferred size of the segment or byte range can vary dramatically based on factors such as the speed of the modem. The foregoing examples (e.g., the default segment size of 8 KB) are based on the assumption that a large number of users will access the network with 56 Kbps modems. However, it may be appropriate to assume that a large number of users will access the network with 28 Kbps modems. In that case, the preferred default segment size would be 4 KB rather than 8 KB. Likewise, the progressively larger segments would be 8 KB and 16 KB rather than 16 KB and 32 KB. Those skilled in the art will appreciate that the use of a variety of other segment sizes is also contemplated by the present invention.

In a preferred embodiment of the present invention, the monitoring of actual bandwidth usage during an extensive download may be given more weight than the monitoring of actual bandwidth usage before or after such a download. That is, if a predetermined number of segments are downloaded consecutively, the average level of usage for each of the segments is adopted as the new "maximum" level. For example, if the predetermined number of segments is 10, the average level of usage during the downloading of ten consecutive segments would replace the previous maximum level even if the average level is less than the previous maximum value. Moreover, if 11 segments are consecutively downloaded, the average level would be calculated for segments 2–11 so that a moving average of 10 downloads determines both a new maximum level and a new threshold level.

Using a moving average to identify the maximum level advantageously provides a dynamic increase or decrease to the threshold level in response to changing network traffic conditions without compromising the accuracy of the measurement. Preferably, the maximum level will remain frozen at the most recent moving average (of the predetermined number of segments) until another extensive download occurs. In other words, a maximum level based on an extensive download is deemed to be a more accurate reflection of available bandwidth than the initial maximum level measured prior to a download. Likewise, the download of an isolated segment may not provide an accurate reflection of available bandwidth. To keep track of the number of consecutive downloads for the moving average, a counter such as the one referenced in FIG. **7** may be used.

The present invention has been described in connection with exemplary embodiments which are intended to be illustrative rather than restrictive. For example, the invention has been described in the context of downloading software updates for the operating system of the PC. The software updates could just as easily relate to many other types of software, such as updates to video games on the PC. Moreover, the data to be downloaded may include text, image and/or audio data.

Alternative embodiments of the present invention will become apparent to those skilled in the art to which it pertains upon review of the specification, including the drawing figures. Accordingly, the scope of the present invention is defined by the appended claims rather than the foregoing description.

The invention claimed is:

1. A method, implemented in one or more computing devices, of transferring a set of data over a network, the method comprising:
- monitoring the level of actual network bandwidth utilization of a communications device;
- automatically identifying a maximum monitored level, wherein the maximum monitored level is a maximum of the monitored level of actual network bandwidth utilization of the communications device;
- calculating a threshold level of utilization as a function of the maximum monitored level of utilization;
- if the actual level is less than the threshold level, separately receiving a plurality of discrete portions of the set of data over the network;
- repeating at least said monitoring step and incrementing a counter each time a discrete portion of the data is received over the network, wherein the size of the discrete portions of the data is a function of the value of the counter; and
- increasing the size of the discrete portions of data when the value of the counter is greater than a predetermined value.

2. The method of claim **1**, wherein a client receives the data over the network from a server.

3. The method of claim **2**, wherein said monitoring occurs at the interface between the client and the network.

4. The method of claim **1**, wherein the network is the Internet.

5. The method of claim **1**, wherein the threshold level is equal to a predetermined percentage of the maximum monitored level.

6. The method of claim **1**, wherein the set of data includes a software update.

7. The method of claim **1**, further comprising clearing the counter after receiving all the plurality of discrete portions of the data over the network.

8. The method of claim **1**, further comprising clearing the counter if the level of actual utilization becomes greater than the threshold level.

9. The method of claim **1**, further comprising suspending the receipt of discrete portions of the data if the level of actual utilization becomes greater than the threshold level.

10. The method of claim **9**, further comprising resuming the receipt of discrete portions of the data from the point of suspension when the level of actual utilization becomes less than the threshold level.

11. The method of claim **1**, further comprising:
- identifying a maximum level of utilization during receipt of the set of data; and
- calculating a threshold level of utilization for the set of data as a function of the maximum level of utilization identified during receipt of the set of data.

12. The method of claim **11**, wherein said identifying step includes estimating the maximum level of utilization during receipt of the set of data by calculating an average level of utilization for the set of data upon repeating said monitoring step a predetermined number of times during receipt of the set of data.

13. The method of claim **11**, further comprising receiving at least a portion of the set of data over the network if the actual level is less than the threshold level of the set of data.

14. The method of claim **11**, further comprising receiving at least a portion of a second set of data over the network if the actual level is less than the threshold level of the set of data.

15. One or more computer-readable storage media having computer-useable instructions embodied thereon for performing a method comprising:

monitoring the level of actual network bandwidth utilization of a communications device;

automatically identifying a maximum monitored level, wherein the maximum monitored level is a maximum of the monitored level of actual network bandwidth utilization of the communications device;

calculating a threshold level of utilization as a function of the maximum monitored level of utilization;

if the actual level is less than the threshold level, separately receiving a plurality of discrete portions of the set of data over the network;

repeating at least said monitoring step and incrementing a counter each time a discrete portion of the data is received over the network, wherein the size of the discrete portions of the data is a function of the value of the counter; and

increasing the size of the discrete portions of data when the value of the counter is greater than a predetermined value.

16. A computer system having a memory, an operating system and a central processor, said processor being operable to use the instructions embodied on the computer-readable media of claim **15**.

17. A system for transferring a set of data over a network, the system comprising:

a monitoring component operative to monitor the level of actual network bandwidth utilization of a communications device;

an identifying component operative to automatically identify a maximum monitored level, wherein the maximum monitored level is a maximum of the monitored level of actual network bandwidth utilization of the communications device;

a calculating component operative to calculate a threshold level of utilization as a function of the maximum monitored level of utilization;

a receiving component operative to separately receive a plurality of discrete portions of the set of data over the network if the actual level is less than the threshold level; and

a counter operable to be incremented each time a discrete portion of the data is received over the network, wherein the size of the discrete portions of the data is a function of the value of the counter and is increased when the value of the counter is greater than a predetermined value.

18. The system of claim **17**, wherein a client receives the data over the network from a server.

19. The system of claim **18**, wherein the monitoring component is operative to monitor the level of actual network bandwidth utilization at the interface between the client and the network.

20. The system of claim **17**, wherein the network is the Internet.

21. The system of claim **17**, wherein the threshold level is equal to a predetermined percentage of the maximum monitored level.

22. The system of claim **17**, wherein the set of data includes a software update.

23. The system of claim **17**, wherein the counter is cleared after the receiving component receives all the plurality of discrete portions of the data over the network.

24. The system of claim **17**, wherein the counter is cleared if the level of actual utilization becomes greater than the threshold level.

25. The system of claim **17**, wherein the receiving component suspends the receipt of discrete portions of the data if the level of actual utilization becomes greater than the threshold level.

26. The system of claim **25**, wherein the receiving component resumes the receipt of discrete portions of the data from the point of suspension when the level of actual utilization becomes less than the threshold level.

27. The system of claim **17**, wherein the identifying component identifies a maximum level of utilization during receipt of the set of data, and the calculating component calculates a threshold level of utilization for the set of data as a function of the maximum level of utilization identified during receipt of the set of data.

28. The system of claim **27**, wherein the identifying component estimates the maximum level of utilization during receipt of the set of data by calculating an average level of utilization for the set of data upon repeating said monitoring a predetermined number of times during receipt of the set of data.

29. The system of claim **27**, wherein the receiving component receives at least a portion of the set of data over the network if the actual level is less than the threshold level of the set of data.

30. The system of claim **27**, wherein the receiving component receives at least a portion of a second set of data over the network if the actual level is less than the threshold level of the set of data.

31. A system for transferring a set of data over a network, the system including computer readable storage media comprising:

means for monitoring the level of actual network bandwidth utilization of a communications device;

means for automatically identifying a maximum monitored level, wherein the maximum monitored level is a maximum of the monitored level of actual network bandwidth utilization of the communications device;

means for calculating a threshold level of utilization as a function of the maximum monitored level of utilization;

means for separately receiving a plurality of discrete portions of the set of data over the network if the actual level is less than the threshold level;

means for repeating at least said monitoring step and incrementing a counter each time a discrete portion of the data is received over the network, wherein the size of the discrete portions of the data is a function of the value of the counter; and

means for increasing the size of the discrete portions of data when the value of the counter is greater than a predetermined value.

* * * * *