(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 January 2008 (10.01.2008)

PCT

(10) International Publication Number
WO 2008/005607 A2

(51) International Patent Classification:
G06F 15/173 (2006.01)

(21) International Application Number:
PCT/US2007/066505

(22) International Filing Date: 12 April 2007 (12.04.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/817,910	30 June 2006 (30.06.2006)	US
11/540,171	29 September 2006 (29.09.2006)	US

(71) Applicant *(for all designated States except US)*: **WITNESS SYSTEMS, INC.** [US/US]; 300 Colonial Center Parkway, Roswell, GA 30076 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **WILLIAMS, Jamie, Richard** [GB/GB]; 2 Upton Grey Drive, Fleet, Hampshire GU51 1EG (GB). **DONG, Thomas** [US/US]; 2466 Chimney Springs Drive, Marietta, GA 30062 (US). **SPOHRER, Dan** [US/US]; 109255 Kimbrall Bridge Zing, Alpharetta, GA 30022 (US). **SRINIVASA, Vijay** [IN/US]; 1712 Cattakay Lane, Suwanne, GA 30024 (US).

(74) Agent: **QUALEY, Jr., M, Paul.**; Thomas, Kayden, Horstemeyer & Risley, LLP, 100 Galleria Parkway, Suite 1750, Atlanta, GA 30339 (US).

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

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DISTRIBUTIVE DATA CAPTURE

(57) **Abstract:** Included are systems and methods for capturing screen data. At least one embodiment of a method includes receiving an indication of a communications session, wherein the communication session is associated with screen data and determining screen data to capture. Some embodiments include capturing data related to the screen data and uploading the captured data to a remote location.

WO 2008/005607 A2

DISTRIBUTIVE DATA CAPTURE

CROSS REFERENCE

[0001] This application claims the benefit of U.S. Provisional Application No. 60/817,910, filed June 30, 2006, which is hereby incorporated by reference in its entirety.

BACKGROUND

[0002] In many network configurations there exists a desire to capture data from one or more computing devices within that network. More specifically, many network configurations can include Voice over Internet Protocol (VoIP) communications. In such a configuration, users may communicate via a VoIP telephone, a softphone, and/or other communications devices. While users of the communications devices may send and receive audio data, depending on the particular configuration, the users may also desire to send other types of data in the form of text files, pictures, video files, audio files, *etc.* Additionally, while the users of communications devices may desire to send and receive the various types of data, the users, system administrators, and others may also desire to record at least a portion of the data being communicated. Additionally, these parties may also desire the ability to record other data presented to a user of a communications and/or computing device.

[0003] Additionally, in highly distributed branch networks, telephony connections can be centralized via a small number of "hub" sites or can be distributed to many or all of the "leaf" nodes of the network. The latter approach may be used for high street or retail operations where each location has a few telephone circuits from its local central office terminating on equipment at that site. There is therefore an increasing desire to provide recording systems in communications and/or data networks that are well suited to all the supported topologies. The challenge in recording data in such networks that are distributed across multiple branches is that much of the data traffic carried is entirely local to that branch. The audio packets associated with the communication do not generally leave the branch. Additionally, it is generally not desirable for the audio packets to leave the branch because there is often only limited bandwidth between the branch and corporate headquarters/data center.

[0004] Many existing IP recording solutions can require a recording device to be located at each branch so as to tap into the data at that branch. Where the number of branches is large, this becomes very expensive. When the total number of calls to be recorded is low, such a network configuration can become uneconomic, as the costs of the hardware and related support are spread across only a few recordings per day.

[0005] Additionally, using existing IP conferencing/service-observe type solutions in which conference bridges are located at the central site generally requires that the audio data be "tromboned" from the receiving site to the conference bridge and back again. In this approach, two legs of a 3-way conference (caller, agent, and recorder port) will generally be transmitted between the branch site and the central equipment. In addition to using scarce bandwidth over this link, such a configuration can use expensive resources at the central site and can impact the quality of the communication.

SUMMARY

[0006] Included are systems and methods for capturing screen data. At least one embodiment of a method includes receiving an indication of a communications session, wherein the communication session is associated with screen data and determining screen data to capture. Some embodiments include capturing data related to the screen data and uploading the captured data to a remote location.

[0007] Also included are embodiments of a computer readable medium for capturing screen data. At least one embodiment of a computer readable medium includes receiving logic configured to receive an indication of a communications session, wherein the communication session is associated with screen data and first determining logic configured to determine screen data to capture. Some embodiments include capturing logic configured to capture data related to the screen data and uploading logic configured to upload the captured data to a remote location.

[0008] Also included are embodiments of a system for capturing screen data. At least one embodiment of a system includes a receiving component configured to receive an indication of a communications session, wherein the communication session is associated with screen data and a first determining component configured to determine screen data to capture. Some embodiments include a capturing component configured to, in response to determining screen data to capture, capture data related to the screen data and an uploading component configured to upload the captured data to a remote location.

[0009] Other systems, methods, features, and advantages of this disclosure will be or become apparent to one with skill in the art upon examination of the following drawings and detailed description. It is intended that all such additional systems, methods, features,

and advantages be included within this description, be within the scope of the present disclosure.

BRIEF DESCRIPTION

[0010] Many aspects of the disclosure can be better understood with reference to the following drawings. The components in the drawings are not necessarily to scale, emphasis instead being placed upon clearly illustrating the principles of the present disclosure. Moreover, in the drawings, like reference numerals designate corresponding parts throughout the several views. While several embodiments are described in connection with these drawings, there is no intent to limit the disclosure to the embodiment or embodiments disclosed herein. On the contrary, the intent is to cover all alternatives, modifications, and equivalents.

[0011] FIG. 1 is a functional diagram illustrating an exemplary configuration of a communications network.

[0012] FIG. 2 is a functional diagram illustrating another exemplary configuration of a communications network with remotely located servers for overcoming deficiencies of the configuration from FIG. 1.

[0013] FIG. 3 is a functional diagram illustrating an exemplary communications network with locally located servers, similar to the configuration from FIG. 2.

[0014] FIG. 4 is a schematic diagram illustrating an exemplary embodiment of a computing device that may be configured to communicate via a communications network such as the networks from FIGS. 1, 2, and 3.

[0015] FIG. 5 is an embodiment of a user interface display for a Voice over Internet Protocol (VoIP) communication that may be utilized on the computing device of FIG. 4.

[0016] FIG. 6 is an embodiment of a user interface display, illustrating inclusion of a file in a communication session, similar to the user interface display from FIG. 5.

[0017] FIG. 7 is an embodiment of a display illustrating access of an unrelated application during a communications session, similar to the embodiment from FIG. 6.

[0018] FIG. 8 is an embodiment of a user interface display for capturing data associated with an application that is divorced from a communications session, similar to the embodiment from FIG. 7.

[0019] FIG. 9 is a flowchart illustrating exemplary steps that can be taken in recording a communication in a communications network, such as the networks of FIGS. 1, 2, and 3.

[0020] FIG. 10 is a flowchart illustrating exemplary steps that can be taken in recording a communication and concurrently sending the data to a server, similar to the flowchart from FIG. 9.

[0021] FIG. 11 is a flowchart illustrating exemplary actions that can be taken in capturing data associated with a communications session, similar to the flowchart from FIG. 10.

[0022] FIG. 12 is a flowchart illustrating an embodiment of compressing captured data for recording, similar to the flowchart from FIG. 11.

[0023] FIG. 13 is a flowchart illustrating exemplary actions that can be taken in removing and/or encrypting sensitive data from captured data, similar to the flowchart from FIG. 12.

DETAILED DESCRIPTION

[0024] The present disclosure makes references to a communications network with multiple outlets. Customers generally desire a selective quality system to record data associated with their in-store agents. The agents can use a heavily distributed Intelligent Contact Management (ICM) IP telephony switch, with stations spread over several of a multitude of sites. This disclosure also outlines embodiments of a plug compatible replacement for the voice capture component that can allow an application server to work in an ICM environment.

[0025] ICM generally lacks a Service Observation capability, so an alternate voice capture capability is generally desired. In such a topology, port spanning is also generally not available. Because of the lack of a service observation capability, passive-tap recording at each site could be implemented, however such a solution can be very costly. The service observation capability can also be simulated by an on-demand targeted capture of a single IP telephone station. The result can be delivered as one or more data files to a server such as an application server.

[0026] In many network environments, a computer is associated with many of the telephones at a branch office. The computer hardware can be separate from the telephone hardware, however this is not a requirement. More specifically, in an exemplary embodiment, the computer can include telecommunications capabilities and act as a telephone without additional hardware. Other configurations can include telecommunications hardware that is distinct from the computing device 104. In such a configuration, the computer and telephone hardware may be communicatively coupled, however this is not a requirement. Regardless of the configuration, there is generally a computing device (or computing logic) associated with a telephone (or telecommunications logic) in many communications networks. Indeed, in an increasing number of scenarios, the "telephone" includes a software application residing on a computing device (a "softphone") rather than a physical device in its own right.

[0027] In many cases, the computing device being used proximate to a Voice over Internet Protocol (VoIP) telephone is already, or can be connected to receive audio packets sent to and from the telephone. The computing device can therefore be used to record audio and/or other data from that telephone. Additionally, other data output from the computing device 104 can also be recorded. By installing a recording application on the computing device 104 alongside the VoIP phone, recordings can be made from that phone and/or from the computing device 104. The recordings can then be transmitted to a central site immediately or buffered locally and sent at a time of reduced network traffic.

[0028] FIG. 1 is a functional diagram illustrating an exemplary configuration of a communications network. In this exemplary embodiment, communications devices 106a and 106b are coupled to computing device 104a. Additionally, communications device 106a is coupled to local network 102a via recording device 108a. Similarly, communications device 106b is coupled to local network 102a via recording device 108b. Local network 102a is coupled to communications network 100.

[0029] Similarly, communications device 106c, as well as communications device 106d, are coupled to computing device 104b. Communications device 106c is also coupled to local network 102b via recording device 108c. Communications device 106d is coupled to local network 102 via recording device 108d. Local network 102b is coupled to communications network 100. Additionally coupled to communications network 100 is an application server 110a. As discussed above, the application server 110a can perform any of a plurality of operations. Additionally, while application server 110a is illustrated as being coupled to communications network 100, one or more application servers 216a can be configured to service specific portions of the overall network illustrated in FIG. 1. More specifically, one can conceive an application server coupled to local network 102a, as well as an application server coupled to local 210b. Other configurations can also be considered as part of this disclosure.

[0030] As discussed above, in many networking environments, a computing device is coupled, either directly or indirectly, to at least one communications device. While the exemplary embodiment of FIG. 1 illustrates two communications devices (106a and 106b) being coupled to one computing device 104a, this is a nonlimiting example, as other configurations can include one or more communications devices coupled to one or more computing devices. Similarly, while the computing devices 206 are illustrated as being separate from communications devices 106, this is a nonlimiting example. As one of ordinary skill in the art will understand, computing logic can be implemented in a communications device 106. Other configurations can include communications logic in the computing devices 206. Other configurations are also contemplated.

[0031] As illustrated in FIG. 1, recording devices 108a and 108b are coupled to communications devices 106a and 106b, respectively. Recording device 108c is coupled to local network 102b. In either configuration one or more recording device is implemented at a branch that is coupled to communications network 100. As discussed above, increased expense and network complexity can result from such a configuration.

[0032] One should also note that local networks 102a, 102b, 102c, and 102d (referred to collectively as local network 102) can include any of a plurality of different networks. More specifically one or more network or network types can be implemented, including but not limited to a Local Area Network (LAN). Similarly, communications network 100 can include one or more different networks and/or types of networks. As a nonlimiting, example, communications network 100 can include a Wide Area Network (WAN), the Internet, and/or other network.

[0033] FIG. 2 is a functional diagram illustrating another exemplary configuration of a communications network with remotely located servers for overcoming deficiencies of the configuration from FIG. 1. In this exemplary embodiment, communications device 106e is coupled to computing device 104c, as well as to local network 102c. Similarly, communications device 106f is coupled to computing device 104d, as well as local network 102c. Similarly, communications device 106g is coupled to computing device 104e, as well as local network 102d. Communications device 106h is coupled to computing device 104f, as well as local network 102d.

[0034] Local network 102d is coupled to communications network 100. Similarly, local network 102c is coupled to communications network 100. Also coupled to communications network 100 are application server 110b, capture control server 216a, and data storage 304. One should note that while data storage 214a, application server 110b, central recording system 212a, and capture control server 216a are coupled to communications network 100, these devices (and/or logic) can physically be located together at a remote site, or separately at a plurality of remote sites, regardless of the physical location of this logic, the functionality associated with these components can be configured to serve one or more branch that is coupled to communications network 100. Additionally, while data storage 214a, application server 110b, central recording system 212a, and capture control server 216a are depicted as separate devices, this is also a nonlimiting example. In at least one embodiment, one or more of these may be combined. Similarly, the functionality of these devices may also be embodied through software, firmware, and/or hardware, depending on the configuration. As such, illustration of this functionality as devices is a nonlimiting example.

[0035] Additionally included in the nonlimiting example of FIG. 2 is a local routing component 220. Local routing component 220 can be configured to facilitate communications from communications device 106c and 106d when communications network 100 is unavailable. More specifically, if a connection between the communications network 100 and local network 102c is severed, the local routing component (which can operate using Survivable Remote Site Telephone (SRST) and/or other technologies) can be configured to facilitate communication of the communications data to the recorder and/or storage of the communications data. Such a configuration can facilitate a local protocol based recording, which can be implemented for a primary recording mechanism and/or to provide fail-over recording protection.

[0036] FIG. 3 is a functional diagram illustrating an exemplary communications network with locally located servers, similar to the configuration from FIG. 2. More specifically, as illustrated in the nonlimiting example of FIG. 3, computing device 104g, which includes a softphone and therefore has the functionality of both a computing device and a communications device, is coupled to local network 102e. Also coupled to local network 102e is computing device 104h, which is also equipped with a softphone. Softphone enabled computing device 104i is coupled to local network 102f, as well as softphone enabled computing device 104j. Local networks 102e and 102f are coupled to communications network 100.

[0037] Also coupled to local network 102e is data storage 214b, as well as capture control server 216b. Coupled to local network 102f is central recording system 212b and application server 110c. More specifically, FIG. 3 illustrates that the functionality embodied in data storage 214b, capture control server 216b, application server 110c, and central recording system 212b can be coupled to the communications network 100 via a local network. These devices need not be remotely situated from any branch office and may be physically located at the same or different locations.

[0038] FIG. 4 is a schematic diagram illustrating an exemplary embodiment of a computing device that may be configured to communicate via a communications network such as the networks from FIGS. 1, 2, and 3. Although a wire-line communications device is illustrated, this discussion can be applied to any device configured for receiving and/or sending data. As illustrated in FIG. 4, in terms of hardware architecture, the computing device 104 includes a processor 482, volatile and nonvolatile memory 484, a display interface 494, data storage 495, and one or more input and/or output (I/O) device interface(s) 496 that are communicatively coupled via a local interface 492. The local interface 492 can include, for example but not limited to, one or more buses and/or other wired or wireless connections. The local interface 492 may have additional elements, which are omitted for simplicity, such as controllers, buffers (caches), drivers, repeaters, and receivers to enable communications. Further, the local interface may include address, control, and/or data connections to enable appropriate communications among the aforementioned components. The processor 482 may be a hardware device for executing software, particularly software stored in volatile and nonvolatile memory 484.

[0039] The processor 482 can be any custom made or commercially available processor, a central processing unit (CPU), an auxiliary processor among several processors associated with the computing device 104, a semiconductor based microprocessor (in the form of a microchip or chip set), a macroprocessor, or generally any device for executing software instructions.

[0040] The volatile and nonvolatile memory 484 can include any one or combination of volatile memory elements (*e.g.*, random access memory (RAM, such as DRAM, SRAM, SDRAM, VRAM, *etc.*)) and nonvolatile memory elements (*e.g.*, ROM, hard drive, tape, CD-ROM, *etc.*). Moreover, the memory 484 may incorporate electronic, magnetic, optical, and/or other types of storage media. Note that the volatile and nonvolatile memory 484 can also have a distributed architecture, where various components are situated remotely from one another, but can be accessed by the processor 482.

[0041] The software in volatile and nonvolatile memory 484 may include one or more separate programs, each of which includes an ordered listing of executable instructions for implementing logical functions. In the example of FIG. 4, the software in the volatile and nonvolatile memory 484 may include communications client software 499, as well as an operating system 486, and a recording cache (*e.g.*, buffer) 497. Communications client software 499 can include a screen capture daemon, a capture control daemon, a voice capture daemon, recording logic, voice recognition logic, and/or other logic. Additionally, while communications client software is illustrated in this nonlimiting example as a single piece of logic, as one of ordinary skill in the art will understand, communications logic 499 can include one or more separate software, hardware, or firmware modules. Additionally, recording cache 497 may be configured to receive and store one or more pieces of data accessed by computing device 104. The data may be part of a communication session, however this is not a requirement.

[0042] The operating system 486 may be configured to control the execution of other computer programs and may be configured to provide scheduling, input-output control, file and data management, memory management, and communication control and related services.

[0043] A system component embodied as software may also be construed as a source program, executable program (object code), script, or any other entity comprising a set of instructions to be performed. When constructed as a source program, the program is translated via a compiler, assembler, interpreter, or the like, which may or may not be included within the volatile and nonvolatile memory 484, so as to operate properly in connection with the Operating System 486.

[0044] The Input/Output devices that may be coupled to system I/O Interface(s) 496 may include input devices, for example but not limited to, a keyboard, mouse, scanner, microphone, camera, proximity device, *etc.* Further, the Input/Output devices may also include output devices, for example but not limited to, a printer, display, *etc.* Finally, the Input/Output devices may further include devices that communicate both as inputs and outputs, for instance but not limited to, a modulator/demodulator (modem; for accessing another device, system, or network), a radio frequency (RF) or other transceiver, a telephonic interface, a bridge, a router, *etc.* Similarly, network interface 488, which is coupled to local interface 492 can be configured to communication with a communications network, such as the network from FIGS. 2 and 3. While this communication may be facilitated via a communications device, such as communications device 106, this is not a requirement.

[0045] If the computing device 104 is a personal computer, workstation, or the like, the software in the volatile and nonvolatile memory 484 may further include a basic input output system (BIOS) (omitted for simplicity). The BIOS is a set of software routines that initialize and test hardware at startup, start the Operating System 486, and support the transfer of data among the hardware devices. The BIOS is stored in ROM so that the BIOS can be executed when the computing device 104 is activated.

[0046] When the computing device 104 is in operation, the processor 482 can be configured to execute software stored within the volatile and nonvolatile memory 484, to communicate data to and from the volatile and nonvolatile memory 484, and to generally control operations of the computing device 104 pursuant to the software. Software in memory, in whole or in part, is read by the processor 482, perhaps buffered within the processor 482, and then executed. Additionally, one should note that while the above description is directed to a computing device 104, other devices (such as application server 110, capture control server 216a, and central recording system 212a) can also include the components described in FIG. 4.

[0047] One should note that communications device 106 can be configured with one or more of the components and/or logic described above with respect to computing device 104. Additionally, communications device and/or computing device can include voice recognition logic, voice-to-text logic, text-to-voice logic, *etc.* (or any permutation thereof), as well as other components and/or logic for facilitating a communication.

Additionally, in some exemplary embodiments, the communications device 106 can include the computing functionality described with respect to computing device 104. Similarly, in some exemplary embodiments, the computing device 104 can include the communications functionality described with respect to communications device 106. While reference to various components and/or logic is directed to the computing device 104 or the communications device 106, as one of ordinary skill in the art will understand, these are nonlimiting examples, as such functionality can be implemented on the computing device 104, the communications device 106, or both.

[0048] One should also note that in at least one nonlimiting example, the computing device 104 and communications device 106 are configured to act as independent devices, but, because the hub/switch can be physically located inside the communications device 106, the communications device 106 can be configured to control the packet flow and copy the associated Real Time Protocol (RTP) streams, so that the desired data can be seen on the computing device's network interface. As RTP streams are addressed to the communications device 106 (or the communications device's counterparty) the RTP streams can be ignored at the hardware level in the network interface 488. However, if the network interface 488 is configured to receive data in a promiscuous mode, the network interface 488 can be configured to "snoop" the RTP streams flowing to and from an adjacent communications device 106.

[0049] As indicated above, embodiments of the computing device 104 include a screen capture daemon. Screen capture of various data related to a communication can be implemented such that the application server 110b will contact the screen capture daemon and obtain screen frames associated with a communication. Similarly, for voice capture, many communications devices, such as IP telephones generally include a small switching hub and can be wired in between the local network infrastructure 102 and the computing device 104 proximate the communications device 106. Physically, the communications device 106 can include two RJ-45 connections. One connection is connected via the building cabling back to the local network 102. The computing device 104 can be connected to the other connection via a short hook-up cable.

[0050] In operation, the screen capture daemon can be configured to capture data that is accessed by a user on computing device 104. More specifically, referring back to FIG. 3, in a communications session between computing device 104g and 104j, voice data, and/or other data may be communicated between the parties of the communication. The user of computing device 104g may desire that the user of computing device 104j view a picture, video, text file, audio file, and/or other data that may be distinct from the voice data communicated between the parties. As the users desire communication of this data, there also may be a desire to capture the data for recording purposes. To facilitate this desire, the screen capture daemon may be configured to capture screen data (which may include pictures, video files, audio files, text files, *etc.*) that is being sent to the other party (or parties) of the communication, and/or otherwise associated with one of the parties.

[0051] Additionally, depending on the particular configuration, the screen capture daemon can be configured to capture data that is sent to a recipient, as well as data that is simply being displayed during a communications session. Similarly, the screen capture daemon can be configured to capture data that is distinct from a communications session all together.

[0052] A voice capture daemon can also reside and execute on computing device 104. The voice capture daemon may be under control of the application server 110, and may start and stop RTP packet capture. The voice capture daemon can detect and isolate the two RTP streams; one directed towards the communications device 106 and one directed away from the communications device 106. Where the call is handled locally, the audio data can be encoded in G.711 protocol, but other protocols can also be utilized, such as, but not limited to the more heavily compressed G.729A protocol (often used when calls traverse communications network 100).

[0053] Referring to capture control, the application server 110 is configured to communicate with a capture control process over a TCP/IP connection. The capture control process (which can run on the application server 110 or a capture control server 216a) announces itself to the application server 110, which can then request a desired number of record and replay ports according to settings in a data file associated with the application server 110. Even though in some embodiments there is generally no concept of "record" ports, the capture control process can accept requests for an arbitrary number of record ports and the application server 110 can reply with the number of replay ports requested. If telephone replay is supported, the capture control process can then attempt to instantiate that number of communications devices 106 for replay.

[0054] The action commands that flow from the application server 110 to the capture control process can include Service Observe ON/OFF commands that specify the station, and Capture ON/OFF commands that specify a filename exposed by the application server 110. Similarly, other commands for dialing and playing back recordings can be sent from application server 110 to the capture control process.

[0055] On receipt of a service observe command, the capture control process can look up the IP address of the desired computing device 104 from a station number supplied in a lookup table that is already being maintained for screen capture. The capture control process can then arm the voice capture daemon on the computing device 104. When the capture control process receives a capture control command, the capture control process can instruct the capture control daemon to begin assembling RTP packets into audio streams.

[0056] While any encoding protocol can be used, if an RTP codec is in use under G.729A protocol, the capture control daemon can assemble 2 kilobits of audio data each second, after the capture control daemon has removed the RTP headers. The capture control daemon can repair the RTP stream in real time by removing duplicate packets, reversing out of order packets and filling any gaps with G.729A "silence." The capture control daemon can assemble a stereo pair of files; one file for transmit, and one file for receive.

[0057] If the audio is received in a G.711 protocol, the data can optionally be compressed locally at the computing device 104 to conform with the G.726 protocol and mixed into a single stream so as to reduce its bandwidth from 2-by-64 kilobits per second (kbps) to 16kbps. In general, any audio input format can be supported with a user-configurable determination of the format for conversion to and whether or not the data should be mixed into a single stream or kept as two independent streams. One should note that while the above description refers to G.729A protocol, G.711 protocol, and G.726 protocol, any encoding protocol can be used.

[0058] One should also note that any of a plurality of different encryption techniques may be used to encrypt data between a computing device 104 (and/or communications device

106) and a network (see FIGS. 1, 2, 3), as well as encrypt data locally within computing device 104 (and/or communications device 106). As a nonlimiting example, communications between the recorder and the daemons associated with a communications device 106 can be configured for encryption and decryption to provide a more secure network environment.

[0059] The capture control daemon can be configured to transfer the captured audio to the capture control process, for further processing. The RTP streams captured by the capture control daemon can be disjoint. Additionally, the application server 110 can operate in a "timed" mode and ask for capture when no call is in progress. At other times, the application server 110 can put calls on hold. The capture control daemon can use a 250 millisecond (ms), or other gap in RTP to indicate breaks between calls. Each of these call segments can be given an incrementing segment number.

[0060] Uploading can be accomplished in any of a plurality of ways. As a nonlimiting example, uploading can occur during a call segment, at the end of a call segment, at the end of recording, *etc.* (or any permutation). The first option of near-real-time optimizes the network traffic (by sending blocks of audio, stripped of the onerous RTP headers, over elastic, reliable TCP/IP pipes) without requiring the capture control daemon to maintain temporary files on the hard disk of the computing device 104. The capture control daemon can use Hypertext Transfer Protocol (HTTP), Server Message Block (SMB), a proprietary Transmission Control Protocol/Internet Protocol (TCP/IP) based protocol, or other protocol (or any permutation therein) to complete the transfer. The choice of protocol can depend on the choice of upload timing.

[0061] After receiving a complete stereo pair, the capture control process can copy a complete stereo pair to the portion of file share exposed by the application server 110. Before the capture control process can process the complete stereo pair, the capture control process converts the audio to a single mono audio file (such as a wav file or other audio file). The capture control process can then convert this data by decompressing the two halves from the G.729A (or other) protocol to a linear format, summing the two halves, and then converting the mixed signal back to the G.711 mu-law protocol (or G.711 A-law, or other protocol, depending on the particular configuration). This operation can be CPU intensive, so some embodiments include facilitating at least one daemon to process this data in a distributed fashion. Such an implementation could, however, lead to a four-fold increase in the amount of audio data copied from the daemon to the central server(s). In the more common case, however, where the audio is received in the G.711 mu-law protocol, the local workstation can mix and compress the data before transmission. Additionally, the capture control process can run co-resident with the application server 110, but when collecting data predominantly in the G.729A protocol, the decompression and mixing load that can be imposed on the capture control process mean that the capture control process can run on a separate server in many environments.

[0062] Instead of transmitting data during the communication, the recording of audio and/or screen data can be buffered in recording cache 497 of volatile and nonvolatile memory 484 (on data storage 495, or otherwise stored and/or accessible to the computing device 104). Additionally, transmission of the recorded audio and/or screen content from computing device 104 back to a central recording system can then be scheduled to occur at quiet periods (*e.g.*, overnight or other times of reduced network traffic). Additional processing of the data may be completed by the computing device 104 prior to and/or after transmission of the data. When used for speech recognition, the computing device 104 may tune its speech analysis algorithms to those speakers from whom the computing device 104 normally received voice data.

[0063] Additionally, for increased efficiency of data transfer, the audio and screen data may be combined over a single connection. Since screen data and/or audio data can be recorded at the computing device 104 (either together or independently), the system clock associated with the computing device 104 can be used to timestamp audio packets and on-screen changes such that the precise relationship between these is known. Other embodiments can facilitate capture of the screen data separately from the audio data. More specifically, in at least one embodiment, screen data can be captured by a first computing device 104, while the audio data is captured by a second computing device 104 (or not captured at all).

[0064] Other embodiments can combine commands to start and stop screen and audio recording, giving more efficient, simpler, and more synchronized control over the recording. Similarly, the deployment of screen and audio recording components on the computing device 104 can be combined into a single installation package such that deploying the audio recording component provides negligible additional overhead if screen capture is being deployed. If screen and/or audio data is buffered (via a rolling buffer or otherwise) at the workstation, 100% recording can be turned on at the computing device 104 with minimal realized impact on the bandwidth or load on the rest of the overall network.

[0065] The central processing system 320 can then instruct the computing device 104 to delete or forward each recording at a later time. This option allows the system to make decisions based on factors that could not be known at the start of the call, such as call duration and call outcome. Although described herein as operating under the control of a centralized quality management system with connection to a central Computer Telephony Integration (CTI) feed, the system can also be deployed with local call detection. By interpreting call setup and control information passing to and from the communications device 106, a computing device 104 can apply local rules or record some or all calls and annotate these recordings with details gleaned from the communications device 106 (*e.g.*, ANI, agent ID as well as others). These details can then be passed back to the central recording system along with the audio content.

[0066] For added security of recordings, in at least one exemplary embodiment, computing devices 206 may copy recording content to other computing devices 206 so as to provide fallback storage in the event of failure of the computing device 104 or its hard disk or attempts to tamper with the recordings.

[0067] To detect tampering and failure of the recording components, embodiments of the central recording system 212 may "heartbeat" the software on one or more computing device 104 on a regular basis to confirm that a particular computing device 104 is still operational and has not failed or been disabled. To ensure that unauthorized parties do not take control of the computing device 104 by "spoofing" the quality system, the computing device 104 may be configured with security devices such as a public key encoding system (not shown) so that only the authorized server can communicate with the computing device 104. The computing device 104 may also alert the user should the IP address of the quality server controlling the computing device 104 change. This alert can give the user an option to accept or reject this new connection.

[0068] In at least one exemplary embodiment, computing devices 104 can be configured to transmit recordings to multiple destinations if requested and/or central equipment can be configured to copy from one system to another if bandwidth between the central hubs is more readily available than between remote sites and hubs.

[0069] The embodiments disclosed herein can be implemented in hardware, software, firmware, or a combination thereof. At least one embodiment, disclosed herein is implemented in software and/or firmware that is stored in a memory and that is executed by a suitable instruction execution system. If implemented in hardware, as in an alternative embodiment embodiments disclosed herein can be implemented with any or a combination of the following technologies: a discrete logic circuit(s) having logic gates for implementing logic functions upon data signals, an application specific integrated circuit (ASIC) having appropriate combinational logic gates, a programmable gate array(s) (PGA), a field programmable gate array (FPGA), *etc.* Additional description of one or more components of this disclosure may also be found in U.S. Application No. 11/394,408, filed March 31, 2006, which is incorporated by reference in its entirety, as well as "ContactStore for Call Manager," which is also incorporated by reference in its entirety, as well as U.S. Patent Application entitled "Distributive Network Control" accorded serial number XX/XXX,XXX, which is also hereby incorporated by reference in its entirety.

[0070] FIG. 5 is an embodiment of a user interface display for a Voice over Internet Protocol (VoIP) communication that may be utilized on the computing device of FIG. 4.

As illustrated in the nonlimiting example of FIG. 5, user interface display 580 can be configured to display data related to a communication. More specifically, in data window 592, the user interface display 580 can be configured to display duration of the communication, file space used for recording, amount of data transferred to a server, destination of the recorded data, the file name(s) of the recorded data, file path(s) of the recorded data, as well as other information. In transcript window 596, a text converted transcript of the communication can be displayed. Similarly, as illustrated in files window 594, data such as video, files, images, *etc.* can be sent to another party (or parties) of the communications session and or received from a party of the communication. Additionally included in user interface display 580 are place call option 582, options option 584, and send data option 586.

[0071] One should note that, depending on the particular configuration, at least a portion of the user interface display may or may not be accessible to a party of the communication. More specifically, in at least one embodiment, one or more of the parties of the conversation may not be aware that the communication is being recorded. As such, at least a portion of the information in FIG. 5 may not be displayed to all the parties of the communication. Conversely, depending on the particular embodiment, at least a portion of the information displayed in FIG. 5 may be provided to a system administrator and/or other party that desires recording of the communication.

[0072] FIG. 6 is an embodiment of a user interface display, illustrating inclusion of a file in a communication session, similar to the user interface display from FIG. 5. As illustrated in the nonlimiting example of FIG. 6, a party to a communications session may desire to send one or more files to another party (or parties) of the communication. More specifically, by selecting send data option 586, a send data window 680 may be displayed for selecting a file to send to a party to the communications session.

[0073] One should note that in the configuration of FIG. 6, the send data window 680 is "in focus" while user interface display 580 is "out of focus." Depending on the particular configuration, screen capture daemon can be configured to capture data based on the "focus" of one or more applications. More specifically, while in some embodiments, the screen capture daemon is configured to capture data that is sent between parties of a communications session, this is not a requirement. Embodiments of the screen capture daemon can be configured to determine the "focus" of an application and capture data from the "in focus" application (or portion of application). With reference to the nonlimiting example of FIG. 6, the screen capture daemon can be configured to capture an image related to send data window 680 and/or one or more of the files being displayed in send data window 680 (whether selected or not for transmission in the communications session).

[0074] FIG. 7 is an embodiment of a display illustrating access of an unrelated application during a communications session, similar to the embodiment from FIG. 6. As illustrated in the nonlimiting example of FIG. 7, during a communications session, a user may also access one or more other applications. More specifically, a user may access the Internet via web browser 780. Access to web browser 780 may be unrelated to the communications session, however, depending on the configuration, recording of this data may be desired. As discussed with reference to FIG. 6, the screen capture daemon can be configured to determine one or more "in focus" applications (in this nonlimiting example, web browser 780) and capture data related to these applications. While the captured data can include a screenshot of the "in focus" application, this is not a requirement. In at least one configuration, other data (such as a web address, file associated with the display, *etc.*) may be included with or substituted for a screenshot of the "in focus" application.

[0075] FIG. 8 is an embodiment of a user interface display for capturing data associated with an application that is divorced from a communications session, similar to the embodiment from FIG. 7. While the nonlimiting example of FIG. 7 illustrated a configuration where the screen capture daemon can be configured to capture data that is unrelated to a communications session, FIG. 8 is included to illustrate that the screen capture daemon can be configured to capture data when a communication session is not currently in progress. More specifically, in at least one configuration, the screen capture daemon can be configured to operate upon activation of computing device 104. In such a configuration, screen capture daemon can be configured to capture data without participating in a communications session. Other configurations can link the screen capture daemon with the voice capture daemon, the capture control daemon, and/or other logic associated with facilitating and/or recording of a communications session. In such a configuration, upon completion of the communications session, the screen capture daemon can be configured to continue capturing data that is accessed on computing device 104.

[0076] FIG. 9 is a flowchart illustrating exemplary steps that can be taken in recording a communication in a communications network, such as the networks of FIGS. 1, 2, and 3. The first step in this nonlimiting example is to receive data related to a communication (block 930). The VoIP logic (which can be included in the communications software 499, 599 or both) can be configured to receive data associated with a communication, which can take the form of a user input associated with placing a call or data related to an incoming communications request. Once this data is received, the VoIP logic can begin recording data from the communication (block 932). The data from the communication can include voice data, as well as video and other types of data. More specifically, if the communication is a video conference, or similar communication, video data may be received in addition the received audio. The VoIP logic can then continue recording the communication until a user input is received or the VoIP logic determines that the communication has terminated (block 934). Once the recording is complete, the VoIP logic can be configured to store the recorded data locally (block 938). The VoIP logic can then determine a time of sparse network traffic (block 940) and send at least a portion of the recorded data to the server (block 942).

[0077] One should note that in the nonlimiting example of FIG. 9, the recorded data can be sent to a server (or related data storage) upon a determination that network traffic is sparse. In such a configuration, the entire recorded file need not be sent at one time. In at least one configuration, portions of the recorded data can be sent to the server as network resources are available. This determination can be based on a predetermined threshold of network activity, a predicted determination of network activity, or other determination.

[0078] FIG. 10 is a flowchart illustrating exemplary steps that can be taken in recording a communication and concurrently sending the data to a server, similar to the flowchart from FIG. 9. The first step in the nonlimiting example of FIG. 10 is to receive data related to a communication (block 1030). Similar to the first step in FIG. 9, this step can include receiving data related to an outgoing communication or data related to an incoming communication. Once this data is received, the VoIP logic can begin recording data from the communication (block 1032). Upon commencement of recording, the VoIP logic can begin sending at least a portion of the recorded data to a server (block 1034). The VoIP logic can then determine that a user input indicates a termination of recording or the VoIP logic can determine that the communication has terminated (block 1036). The VoIP logic can then terminate the recording.

[0079] As the flowchart from FIG. 9 illustrates an embodiment where a recording is sent to a server when network activity is low, the nonlimiting example of FIG. 10 illustrates an embodiment where the recording is sent to the server as the recording is taking place. This embodiment may be desirable when local storage of the recorded data is desired due to current network traffic. As discussed above, an option can be provided to a user to determine whether the recording is immediately stored at a server or whether the recording is queued for subsequent delivery.

[0080] Additionally, other exemplary embodiments can provide for the precise relative time-stamping of audio and screen content (*e.g.*, speech recognition can take cues from the screen activity immediately following the audio). More specifically, if the user selects "John Doe" from the list and the speech recognizers interprets voice input as either "John Doe" or "Don't know" because of the more likely scenario, the speech recognizers can infer that the former is more likely and hence gain higher accuracy.

[0081] FIG. 11 is a flowchart illustrating exemplary actions that can be taken in capturing data associated with a communications session, similar to the flowchart from FIG. 10. As illustrated in FIG. 11, the computing device 104 and/or communications device 106 can receive an indication of a communication (block 1130). The indication can take the form of a user initiating communications software, picking up a receiver of the communications device 106, or other actions. Once the indication is received, the computing device 104 and/or communications device 106 can determine at least a portion of data for capture (block 1132). As discussed above, this determination can be made based on data that is sent from one communications session party to another, based on application "focus" and/or based on other factors.

[0082] The computing device 104 and/or communications device 106 can then capture at least a portion of the data (block 1134) and send the captured data to recording cache 497 (block 1136). The computing device 104 and/or communications device 106 can then upload at least a portion of the cached data to a remote server (block 1140). As discussed above, the data can be buffered such that the data can be uploaded at a time of recording and/or at a time of reduced network traffic.

[0083] FIG. 12 is a flowchart illustrating an embodiment of compressing captured data for recording, similar to the flowchart from FIG. 11. As illustrated in this nonlimiting example, the computing device 104 and/or communications device 106 can receive indication of a communication (block 1230). The computing device 104 and/or communications device 106 can then determine data for capture (block 1232). As discussed, the data can be determined based on data sent between parties of a communications session, based on "focus" and/or based on other criteria. The computing device 104 and/or communications device 106 can then capture at least a portion of the data (block 1234). The computing device 104 and/or communications device 10 can then buffer at least a portion of the captured data (block 1236).

[0084] The computing device 104 and/or communications device 106 can then determine at least one aspect of the captured data (block 1238). More specifically, depending on one or more criteria of the captured data, the computing device 104 and/or communications device 106 can compress the captured data in one or more different ways. More specifically, in at least one nonlimiting example, the computing device 104 and/or communications device 106 can determine if the captured data includes video data. As the clarity of text data can be compromised without significantly reducing the data being conveyed, compression of the text portions of the captured data may be implemented. Conversely, if the captured data includes video data, the video portion of the captured file(s) may not be compressed, since the clarity of video may be important understand the data that is captured.

[0085] The screen capture daemon can be configured to determine the data that is desired to be compressed as opposed to the data that is not desired to be compressed. This determination can be made based on one or more factors that could include analysis of the file name, file extension, size, embedded objects, and/or other criteria. Upon determination of the at least one aspect of the captured data (block 1238), the computing device 104 and/or communications device 106 can compress at least a portion of the captured data based on at least one predetermined compression technique (block 1240). More specifically, the computing device 104 and/or communications device 106 can be configured to execute one or more compression algorithm based on the data being compressed. As a determination of whether compression is desired was made based on the substance of data captured, the type of compression may vary, based on similar criteria. As a nonlimiting example, if the captured data includes a video file and text data, the video file may be compressed using a different compression algorithm than the text portion of the captured data. Once the captured data is compressed, the computing device 104 and/or communications device 106 can upload the compressed data (block 1242).

[0086] Additionally, other embodiments can be configured to compress the captured data based on predicted network traffic at the time of upload. More specifically, if the computing device 104 and/or communications device 106 is configured to upload the captured data immediately, compression of the captured data may depend on the current network traffic. If the current network traffic is high, a more thorough compression of the captured data may be performed. If the current network traffic is low, compression may not be necessary to conserve network resources and thus compression may be limited.

[0087] Similarly, if the computing device 104 and/or communications device 106 is configured to buffer the captured data for uploading at a time of reduced network traffic, compression can be based (at least in part) on a prediction of network traffic at the time of scheduled upload. More specifically, depending on the particular configuration, the computing device 104 and/or communications device 106 can be scheduled to upload data at 2:00 AM, when network traffic is reduced. The computing device 104 and/or communications device 106 can monitor network usage at that time and, based on the monitored data, determine a predicted network usage for compression. Other embodiments can be configured to upload when it is determined that network usage has fallen below a certain threshold. Such configurations can utilize this threshold to determine the desired compression.

[0088] One should note that although FIG. 12 illustrated compression as occurring subsequent to buffer, this is a nonlimiting example. As with other blocks in the flowcharts of this disclosure, block 1238 and 1240 can occur at a time different than illustrated in FIG. 12. As a nonlimiting example, depending on the particular embodiment, blocks 1238 and 1240 can occur prior to buffer, such that the buffered data takes up a reduced amount of space in cache. Additionally, other embodiments can include a determination of whether (and/or when) to upload the data. More specifically, in at least one embodiment an upload request can be sent to one or more components of the communications network. The data can be uploaded in response to an indication from one or more network components.

[0089] FIG. 13 is a flowchart illustrating exemplary actions that can be taken in removing and/or encrypting sensitive data from captured data, similar to the flowchart from FIG. 12. As illustrated in the nonlimiting example of FIG. 13, the computing device 104 and/or communications device 106 can receive an indication of a communication (block 1230). The computing device 104 and/or communications device 106 can then determine data for capture (block 1332). As discussed above, this determination can be made from data sent during the communication and/or other criteria, such as "focus." The computing device 104 and/or communications device 106 can then capture the determined data (block 1334). The computing device 104 and/or communications device 106 can then buffer at least a portion of the captured data (block 1336). The computing device 104 and/or communications device 106 can then remove at least a portion of the data from cache (block 1338) and determine if there is any sensitive data in the captured data (block 1340). More specifically, the computing device 104 and/or communications device 106 can determine whether the captured data includes credit card information, social security information, and/or other information that is determined to be sensitive. If the computing device 104 and/or communications device 106 determines that the captured data includes sensitive data, the computing device 104 and/or communications device 106 can encrypt/remove/block the sensitive data (block 1342).

[0090] As a nonlimiting example, depending on the particular configuration, upon a determination of the sensitive data, the sensitive data can be encrypted locally on the computing device 104 and/or communications device 106. Other configurations can include removing the sensitive data from the captured data such that the sensitive data is not recorded. Still other configurations can include capturing of the sensitive data, but blocking of the sensitive data from transmission to a remote location. Other configurations are also considered. The computing device 104 and/or communications device 106 can then upload the captured data (block 1244). Additional description related to encryption of sensitive data is provided in U.S. Application No. 11/395,514, filed March 31, 2006, which is hereby incorporated by reference in its entirety.

[0091] One should note that the flowcharts included herein show the architecture, functionality, and operation of a possible implementation of software. In this regard, each block can be interpreted to represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s). It should also be noted that in some alternative implementations, the functions noted in the blocks may occur out of the order and/or not at all. For example, two blocks shown in succession may in fact be executed substantially concurrently or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved.

[0092] One should note that any of the programs listed herein, which can include an ordered listing of executable instructions for implementing logical functions, can be embodied in any computer-readable medium for use by or in connection with an instruction execution system, apparatus, or device, such as a computer-based system, processor-containing system, or other system that can fetch the instructions from the instruction execution system, apparatus, or device and execute the instructions. In the context of this document, a "computer-readable medium" can be any means that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device. The computer readable medium can be, for example but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device. More specific examples (a nonexhaustive list) of the computer-readable medium could include an electrical connection (electronic) having one or more wires, a portable computer diskette (magnetic), a random access memory (RAM) (electronic), a read-only memory (ROM) (electronic), an erasable programmable read-only memory (EPROM or Flash memory) (electronic), an optical fiber (optical), and a portable compact disc read-only memory (CDROM) (optical). In addition, the scope of the certain embodiments of this disclosure can include embodying the functionality described in logic embodied in hardware or software-configured mediums.

[0093] One should also note that conditional language, such as, among others, "can," "could," "might," or "may," unless specifically stated otherwise, or otherwise understood within the context as used, is generally intended to convey that certain embodiments include, while other embodiments do not include, certain features, elements and/or steps. Thus, such conditional language is not generally intended to imply that features, elements and/or steps are in any way required for one or more particular embodiments or that one or more particular embodiments necessarily include logic for deciding, with or without user input or prompting, whether these features, elements and/or steps are included or are to be performed in any particular embodiment.

[0094] It should be emphasized that the above-described embodiments are merely possible examples of implementations, merely set forth for a clear understanding of the principles of this disclosure. Many variations and modifications may be made to the above-described embodiment(s) without departing substantially from the spirit and principles of the disclosure. All such modifications and variations are intended to be included herein within the scope of this disclosure.

CLAIMS

Therefore, at least the following is claimed:

1. A method for capturing screen data, comprising:

receiving an indication of a communications session, wherein the communication session is associated with screen data;

determining data to capture;

in response to determining data to capture, capturing at least a portion of the data; and

uploading the captured data to a remote location.

2. The method of claim 1, wherein the data to capture includes screen data.

3. The method of claim 2, wherein determining data to capture includes determining whether an application is active within an operating environment.

4. The method of claim 2, wherein capturing data related to the screen data includes capturing a file associated with the screen data.

5. The method of claim 1, further comprising determining whether to compress at least a portion of the captured data.

6. The method of claim 5, wherein determining whether to compress at least a portion of the captured data includes determining a data type associated with the captured data.

7. The method of claim 1, further comprising determining whether at least a portion of the captured data includes sensitive information.

8. The method of claim 7, further comprising, in response to determining that at least a portion of the captured data includes sensitive information, performing at least one of the following: encrypting the sensitive data, removing the sensitive data, and blocking the sensitive data.

9. The method of claim 1, further comprising caching at least a portion of the screen data prior to uploading the captured screen data.

10. The method of claim 1, further comprising sending an upload request prior to uploading the captured data.

11. The method of claim 1, further comprising buffering at least a portion of the captured data in a rolling buffer.

12. A computer readable medium for capturing screen data, comprising:

receiving logic configured to receive an indication of a communications session, wherein the communication session is associated with screen data;

first determining logic configured to determine screen data to capture;

capturing logic configured to, in response to determining screen data to capture, capture data related to the screen data;

caching logic configured to buffer at least a portion of the captured data; and

uploading logic configured to upload at least a portion of the captured data to a remote location.

13. The computer readable medium of claim 12, wherein the determining logic is further configured to determine whether an application is active within an operating environment.

14. The computer readable medium of claim 12, wherein the capturing logic is further configured to capture a file associated with the screen data.

15. The computer readable medium of claim 12, further comprising compressing logic configured to determine whether to compress at least a portion of the captured data.

16. The method of claim 15, wherein the compressing logic is further configured to determine a data type associated with the captured data.

17. The computer readable medium of claim 12, further comprising second determining logic configured to determine whether at least a portion of the captured data includes sensitive information.

18. The computer readable medium of claim 17, further comprising performing logic configured to, in response to determining that at least a portion of the captured data includes sensitive information, perform at least one of the following: encrypting the sensitive data, removing the sensitive data, and blocking the sensitive data.

19. A system for capturing screen data, comprising:

a receiving component configured to receive an indication of a communications session, wherein the communication session is associated with screen data;

a first determining component configured to determine screen data to capture;

a capturing component configured to, in response to determining screen data to capture, capture data related to the screen data;

a requesting component configured to send an upload request; and

an uploading component configured to, in response to receiving a positive response to the upload request, upload the captured data to a remote location.

20. The system of claim 19, wherein the determining component is further configured to determine whether an application is active within an operating environment.

21. The system of claim 19, wherein the capturing component is further configured to capture a file associated with the screen data.

22. The system of claim 19, further comprising a compressing component configured to determine whether to compress at least a portion of the captured data.

23. The system of claim 22, wherein the compressing component is further configured to determine a data type associated with the captured data.

24. The system of claim 19, further comprising a second determining component configured to determine whether at least a portion of the captured data includes sensitive information.

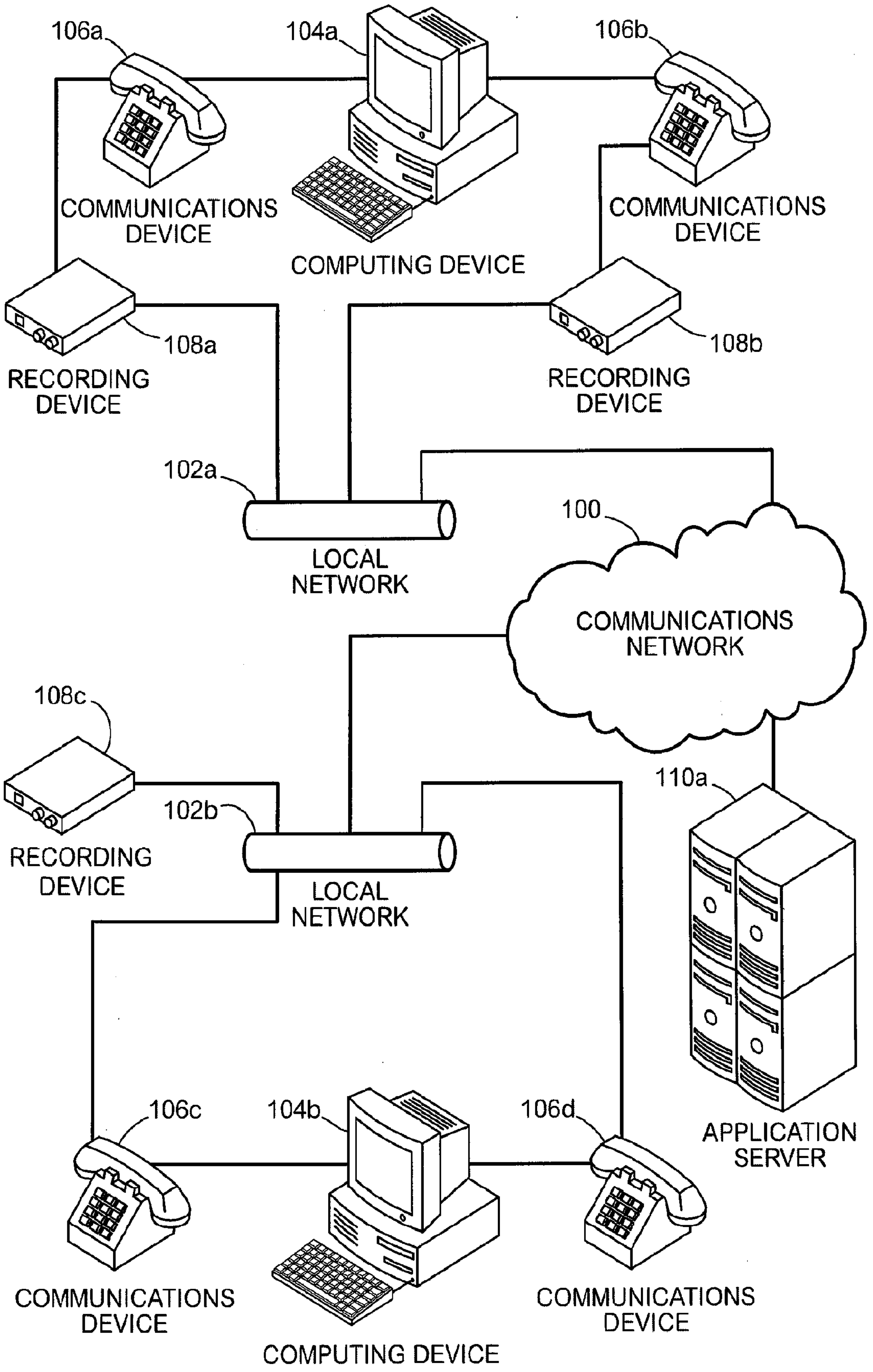

106a
104a
106b
COMMUNICATIONS DEVICE
COMPUTING DEVICE
COMMUNICATIONS DEVICE
108a
RECORDING DEVICE
108b
RECORDING DEVICE
102a
100
LOCAL NETWORK
COMMUNICATIONS NETWORK
108c
RECORDING DEVICE
102b
LOCAL NETWORK
110a
APPLICATION SERVER
106c
104b
106d
COMMUNICATIONS DEVICE
COMPUTING DEVICE
COMMUNICATIONS DEVICE

FIG. 1

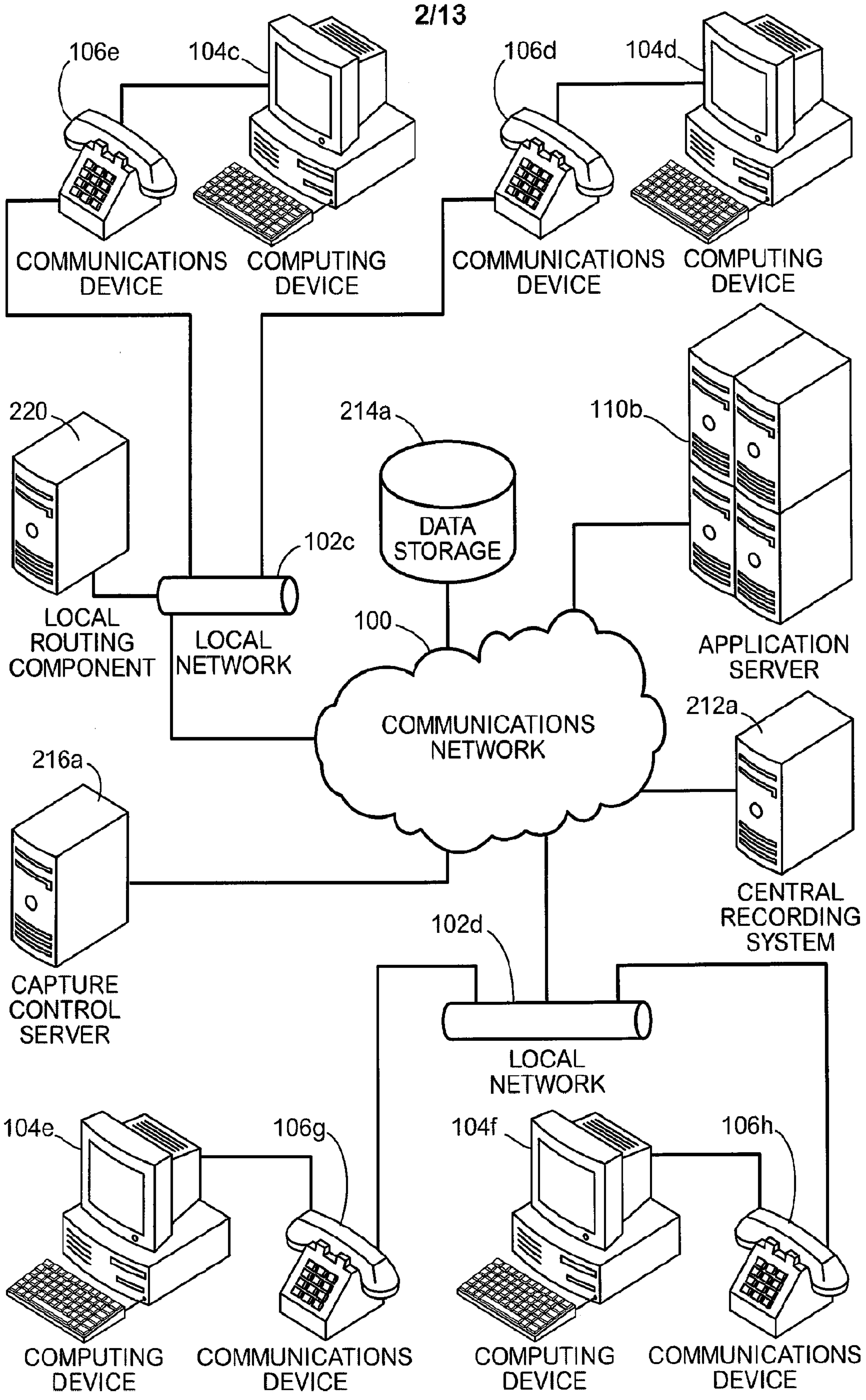

106e
104c
106d
104d
COMMUNICATIONS DEVICE
COMPUTING DEVICE
COMMUNICATIONS DEVICE
COMPUTING DEVICE
220
214a
110b
DATA STORAGE
102c
LOCAL ROUTING COMPONENT
LOCAL NETWORK
100
APPLICATION SERVER
COMMUNICATIONS NETWORK
212a
216a
CENTRAL RECORDING SYSTEM
102d
CAPTURE CONTROL SERVER
LOCAL NETWORK
104e
106g
104f
106h
COMPUTING DEVICE
COMMUNICATIONS DEVICE
COMPUTING DEVICE
COMMUNICATIONS DEVICE

FIG. 2

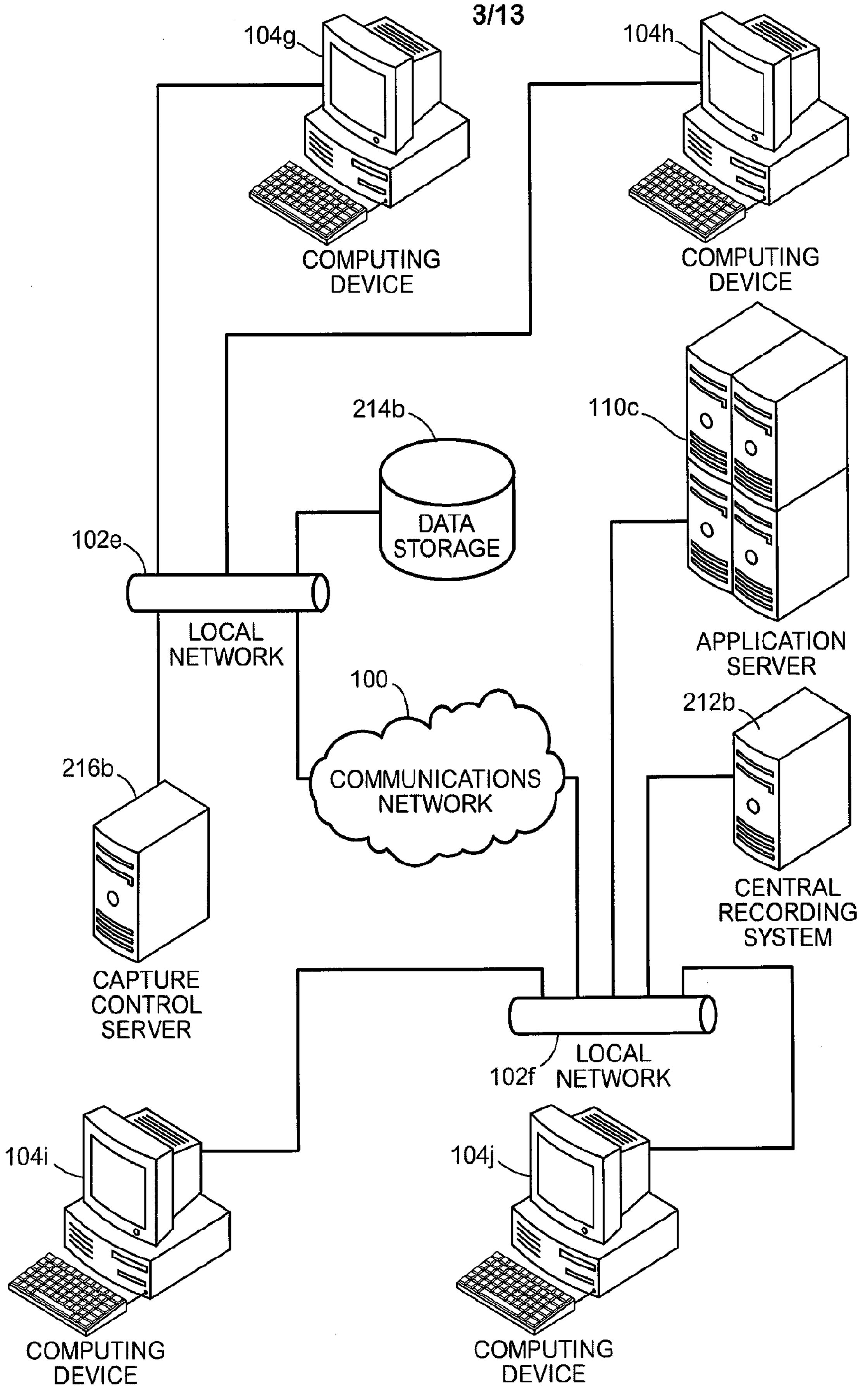
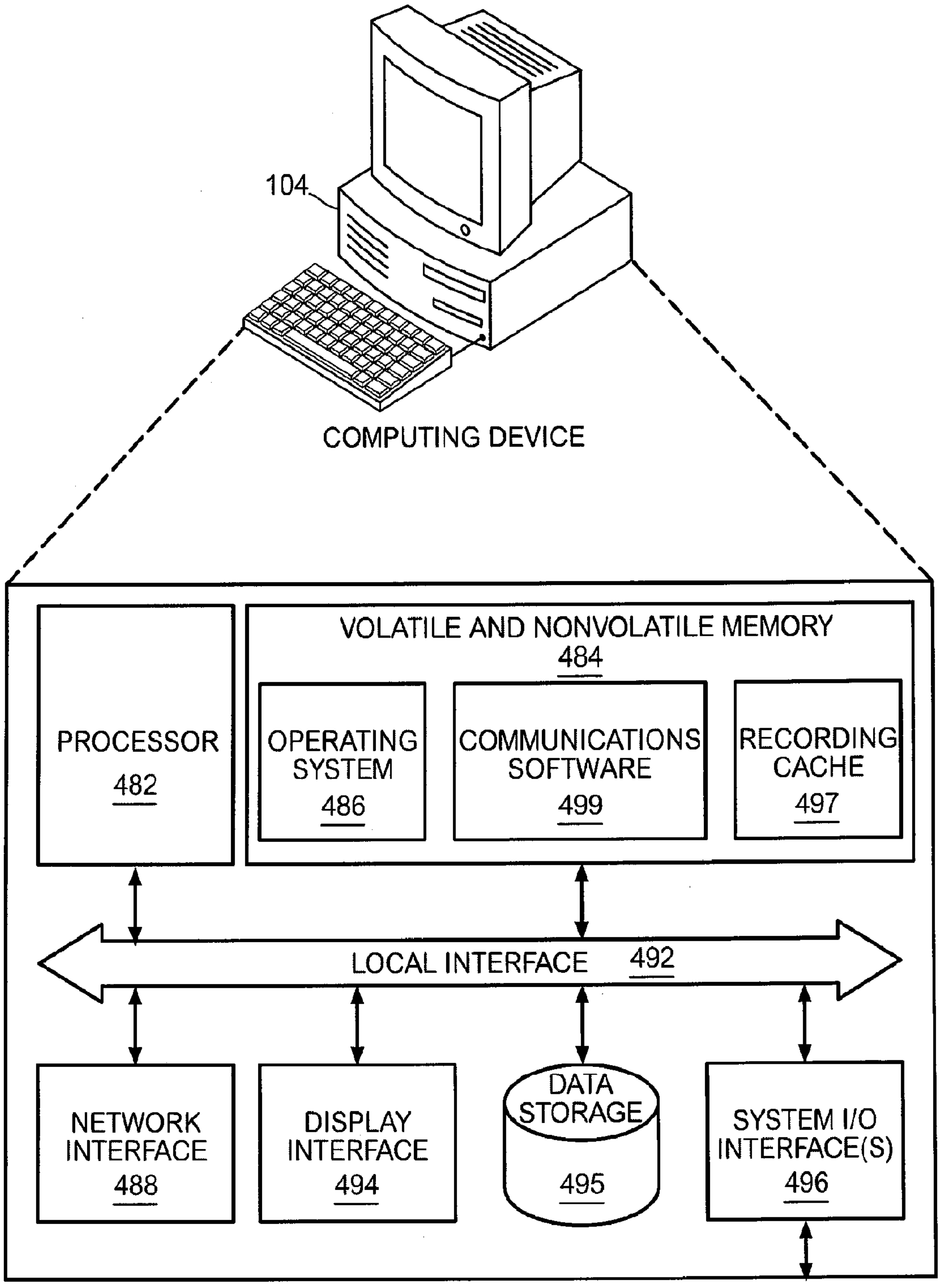
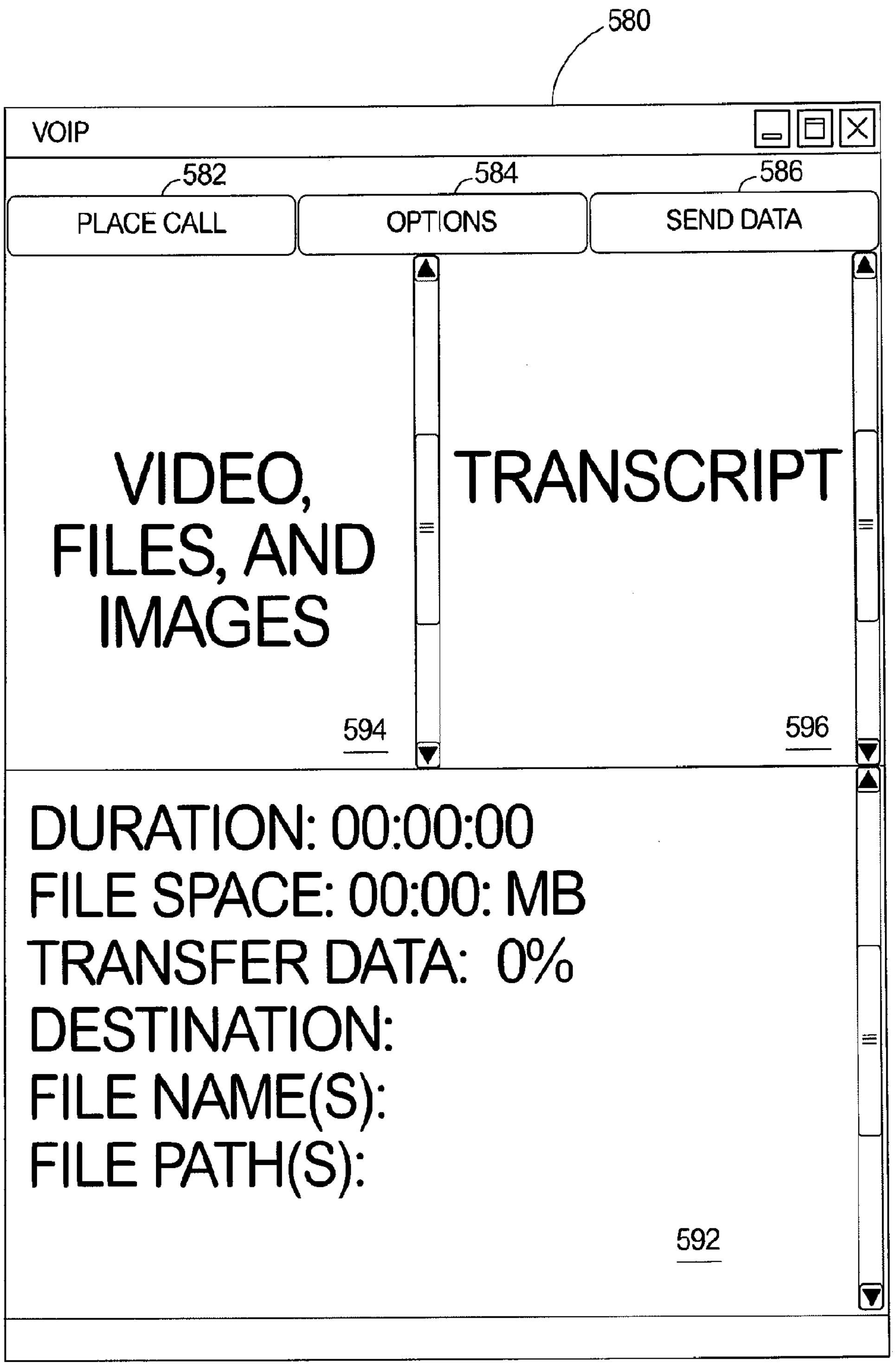

104g
COMPUTING DEVICE
104h
COMPUTING DEVICE
214b
DATA STORAGE
110c
APPLICATION SERVER
102e
LOCAL NETWORK
100
COMMUNICATIONS NETWORK
212b
CENTRAL RECORDING SYSTEM
216b
CAPTURE CONTROL SERVER
102f
LOCAL NETWORK
104i
COMPUTING DEVICE
104j
COMPUTING DEVICE

FIG. 3



104
COMPUTING DEVICE
PROCESSOR 482
VOLATILE AND NONVOLATILE MEMORY 484
OPERATING SYSTEM 486
COMMUNICATIONS SOFTWARE 499
RECORDING CACHE 497
LOCAL INTERFACE 492
NETWORK INTERFACE 488
DISPLAY INTERFACE 494
DATA STORAGE 495
SYSTEM I/O INTERFACE(S) 496

FIG. 4



DURATION: 00:00:00
FILE SPACE: 00:00: MB
TRANSFER DATA: 0%
DESTINATION:
FILE NAME(S):
FILE PATH(S):

FIG. 5

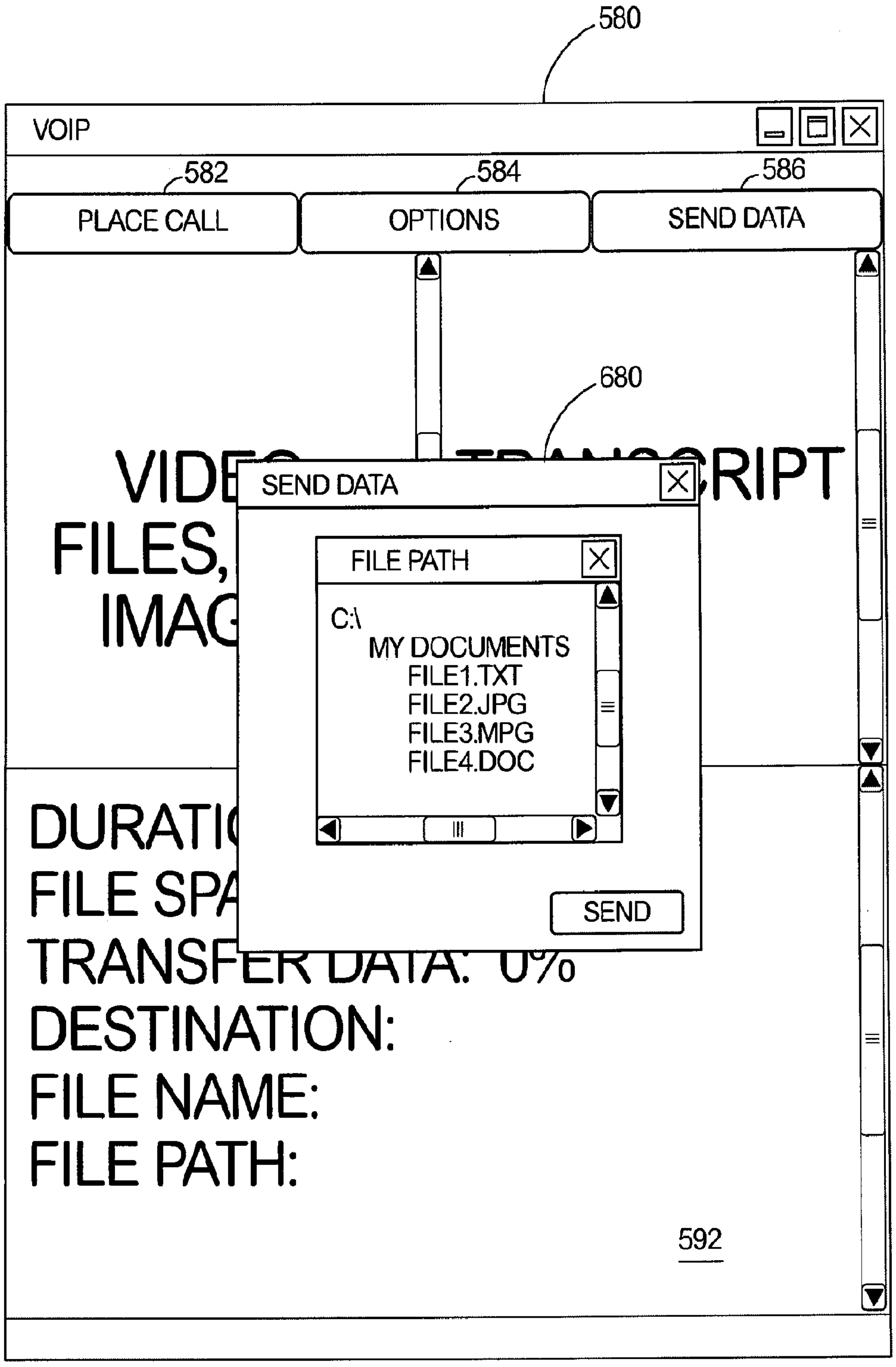
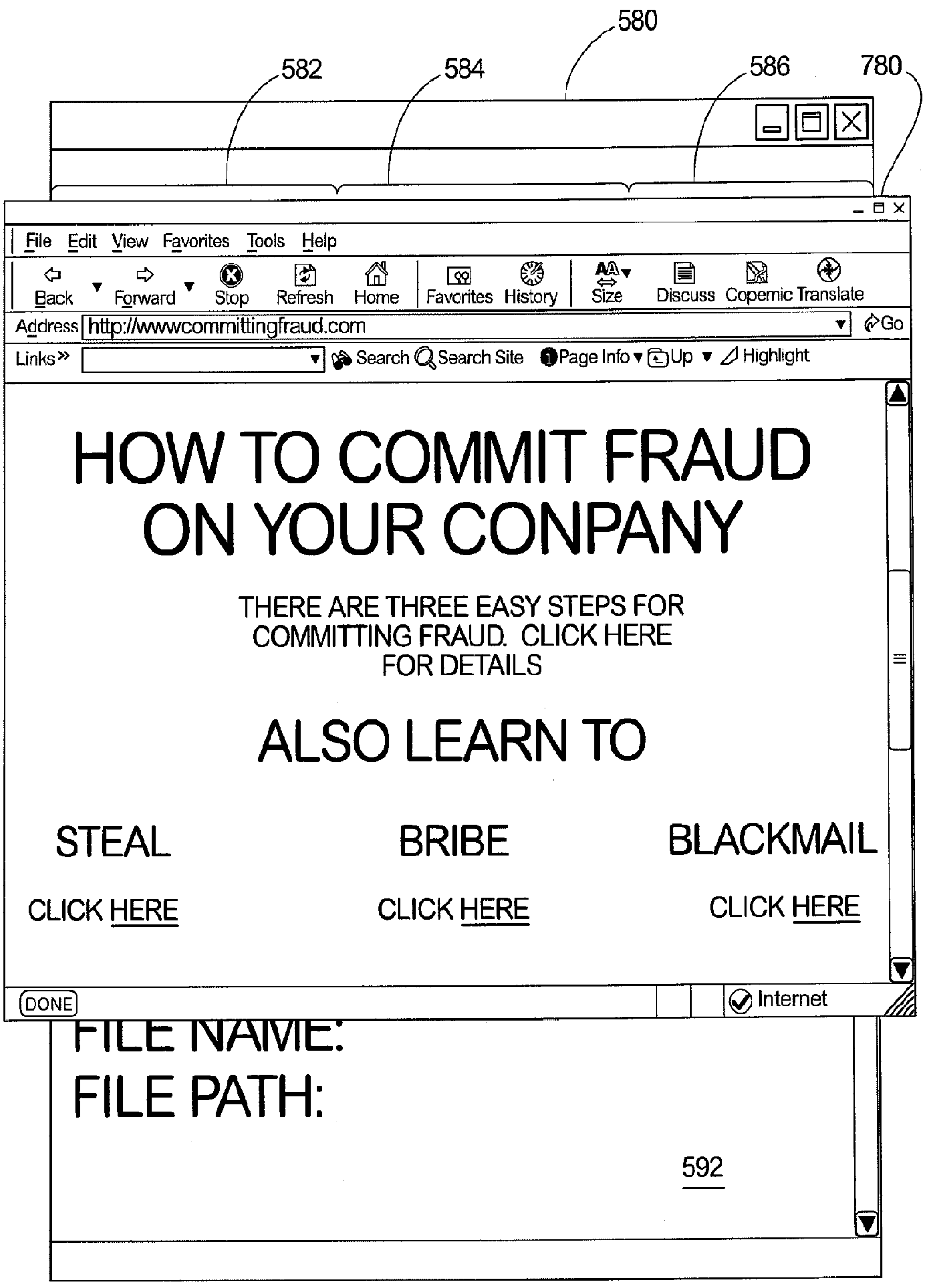

580
VOIP
582
PLACE CALL
584
OPTIONS
586
SEND DATA
680
VIDE
FILES,
IMAG
RIPT
SEND DATA
FILE PATH
C:\
MY DOCUMENTS
FILE1.TXT
FILE2.JPG
FILE3.MPG
FILE4.DOC
SEND
DURATI
FILE SPA
TRANSFER DATA: 0%
DESTINATION:
FILE NAME:
FILE PATH:
592

FIG. 6



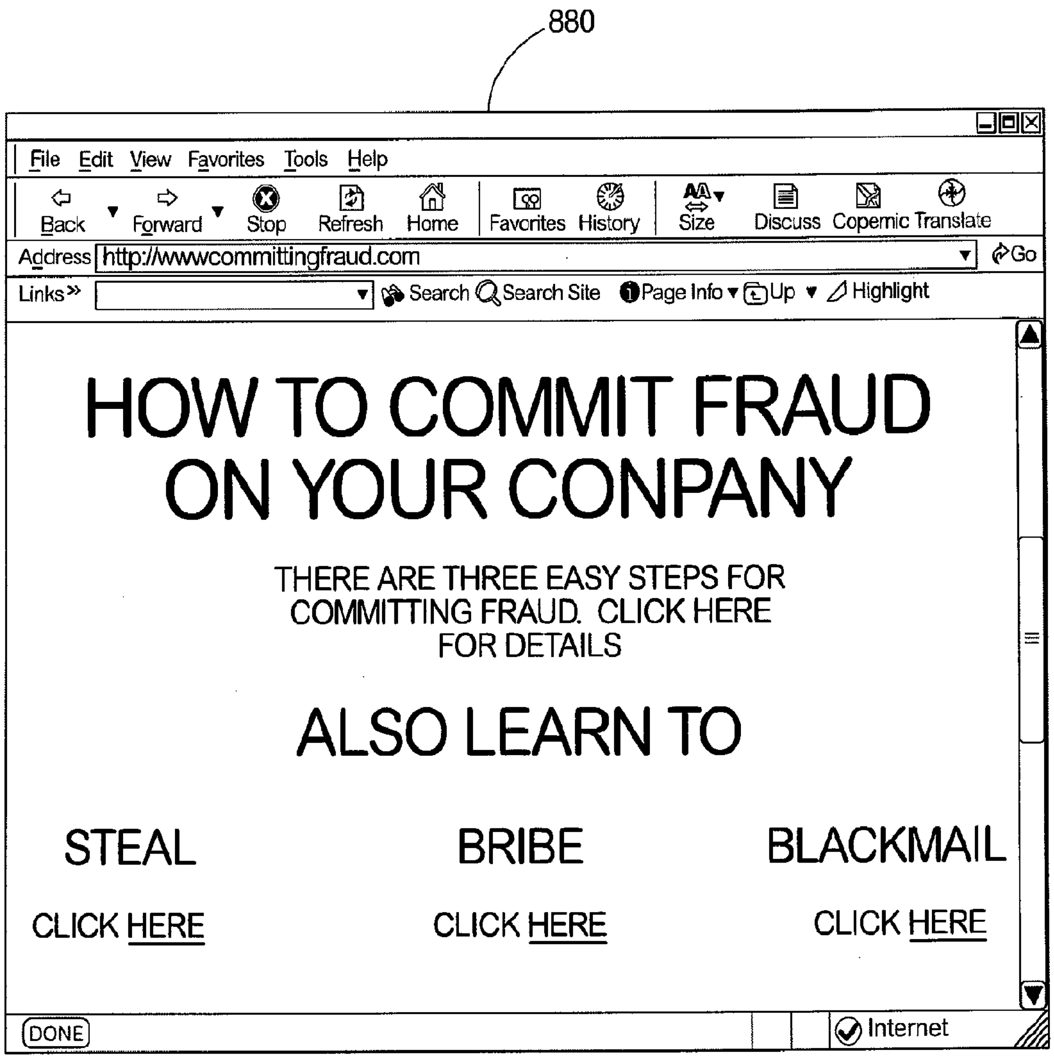
HOW TO COMMIT FRAUD ON YOUR CONPANY

THERE ARE THREE EASY STEPS FOR COMMITTING FRAUD. CLICK HERE FOR DETAILS

ALSO LEARN TO

STEAL	BRIBE	BLACKMAIL
CLICK HERE	CLICK HERE	CLICK HERE

FIG. 7


880
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Favorites History Size Discuss Copernic Translate
Address
Go
Links
Search Search Site Page Info Up Highlight
DONE
Internet

http://wwwcommittingfraud.com

HOW TO COMMIT FRAUD ON YOUR CONPANY

THERE ARE THREE EASY STEPS FOR COMMITTING FRAUD. CLICK HERE FOR DETAILS

ALSO LEARN TO

STEAL	BRIBE	BLACKMAIL
CLICK HERE	CLICK HERE	CLICK HERE

FIG. 8

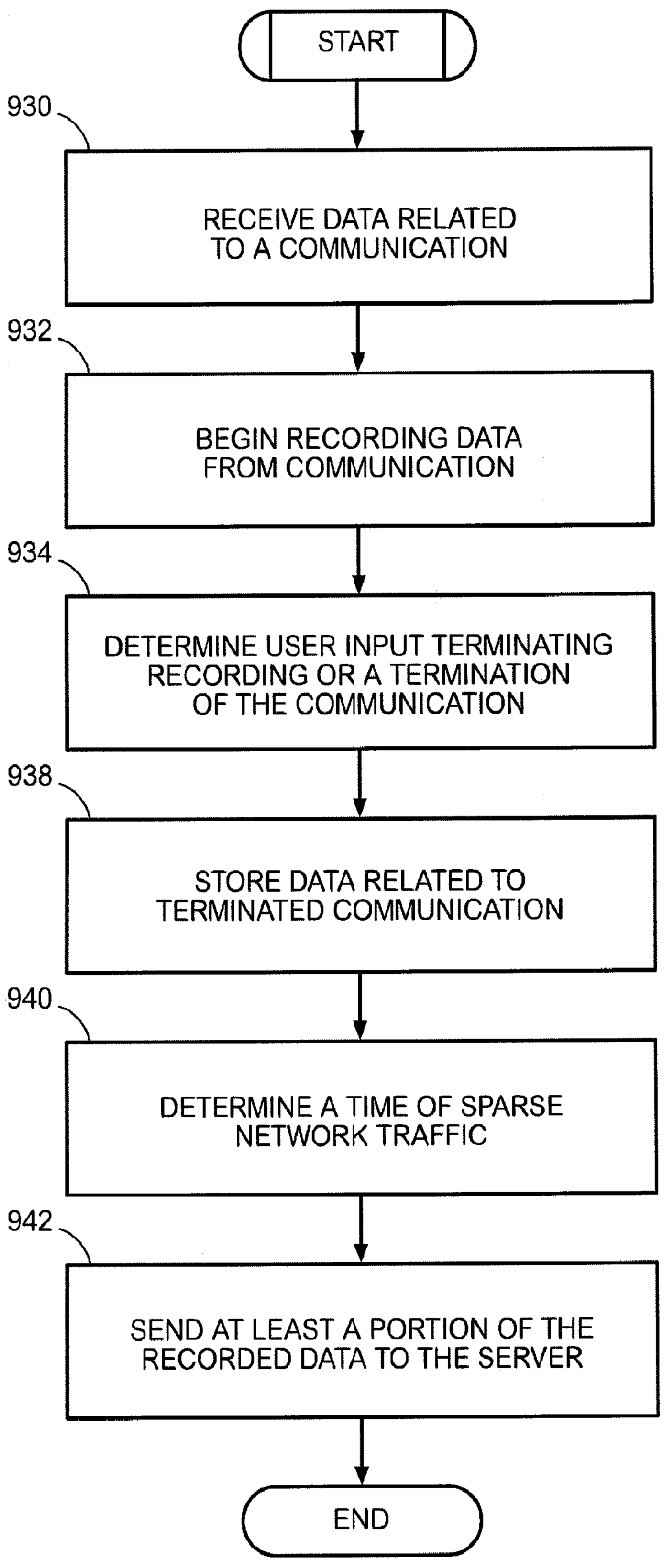

START
930
RECEIVE DATA RELATED TO A COMMUNICATION
932
BEGIN RECORDING DATA FROM COMMUNICATION
934
DETERMINE USER INPUT TERMINATING RECORDING OR A TERMINATION OF THE COMMUNICATION
938
STORE DATA RELATED TO TERMINATED COMMUNICATION
940
DETERMINE A TIME OF SPARSE NETWORK TRAFFIC
942
SEND AT LEAST A PORTION OF THE RECORDED DATA TO THE SERVER
END

FIG. 9

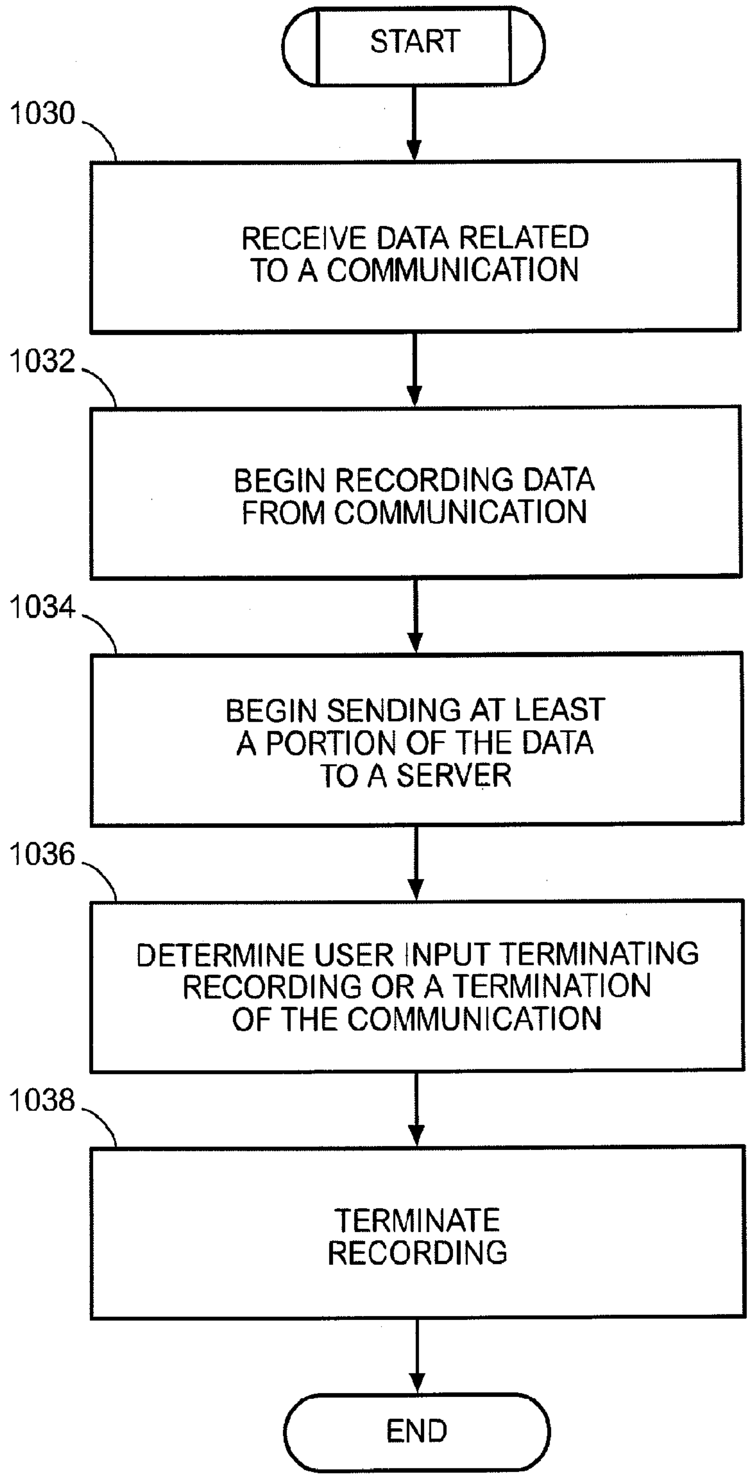

START
1030
RECEIVE DATA RELATED TO A COMMUNICATION
1032
BEGIN RECORDING DATA FROM COMMUNICATION
1034
BEGIN SENDING AT LEAST A PORTION OF THE DATA TO A SERVER
1036
DETERMINE USER INPUT TERMINATING RECORDING OR A TERMINATION OF THE COMMUNICATION
1038
TERMINATE RECORDING
END

FIG. 10

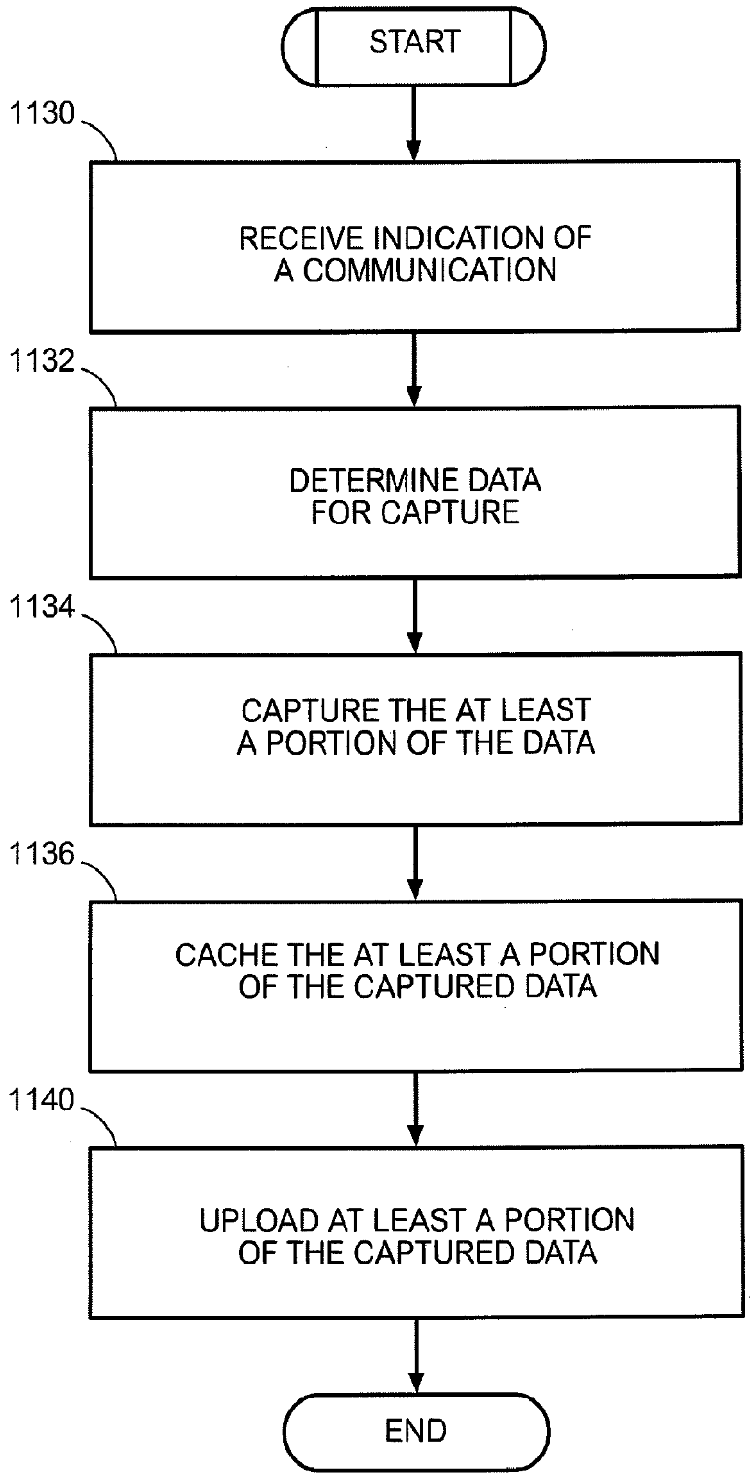

START
1130
RECEIVE INDICATION OF A COMMUNICATION
1132
DETERMINE DATA FOR CAPTURE
1134
CAPTURE THE AT LEAST A PORTION OF THE DATA
1136
CACHE THE AT LEAST A PORTION OF THE CAPTURED DATA
1140
UPLOAD AT LEAST A PORTION OF THE CAPTURED DATA
END

FIG. 11

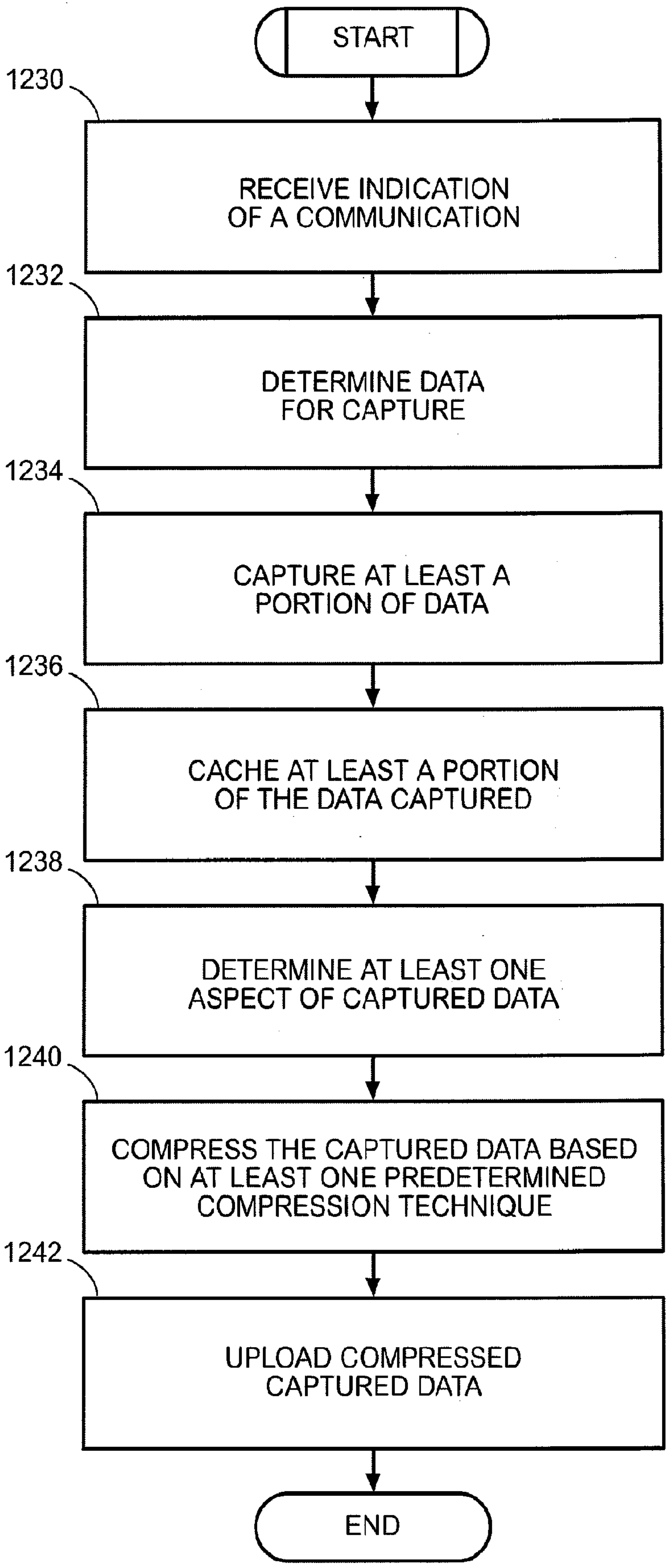

START
1230
RECEIVE INDICATION OF A COMMUNICATION
1232
DETERMINE DATA FOR CAPTURE
1234
CAPTURE AT LEAST A PORTION OF DATA
1236
CACHE AT LEAST A PORTION OF THE DATA CAPTURED
1238
DETERMINE AT LEAST ONE ASPECT OF CAPTURED DATA
1240
COMPRESS THE CAPTURED DATA BASED ON AT LEAST ONE PREDETERMINED COMPRESSION TECHNIQUE
1242
UPLOAD COMPRESSED CAPTURED DATA
END

FIG. 12

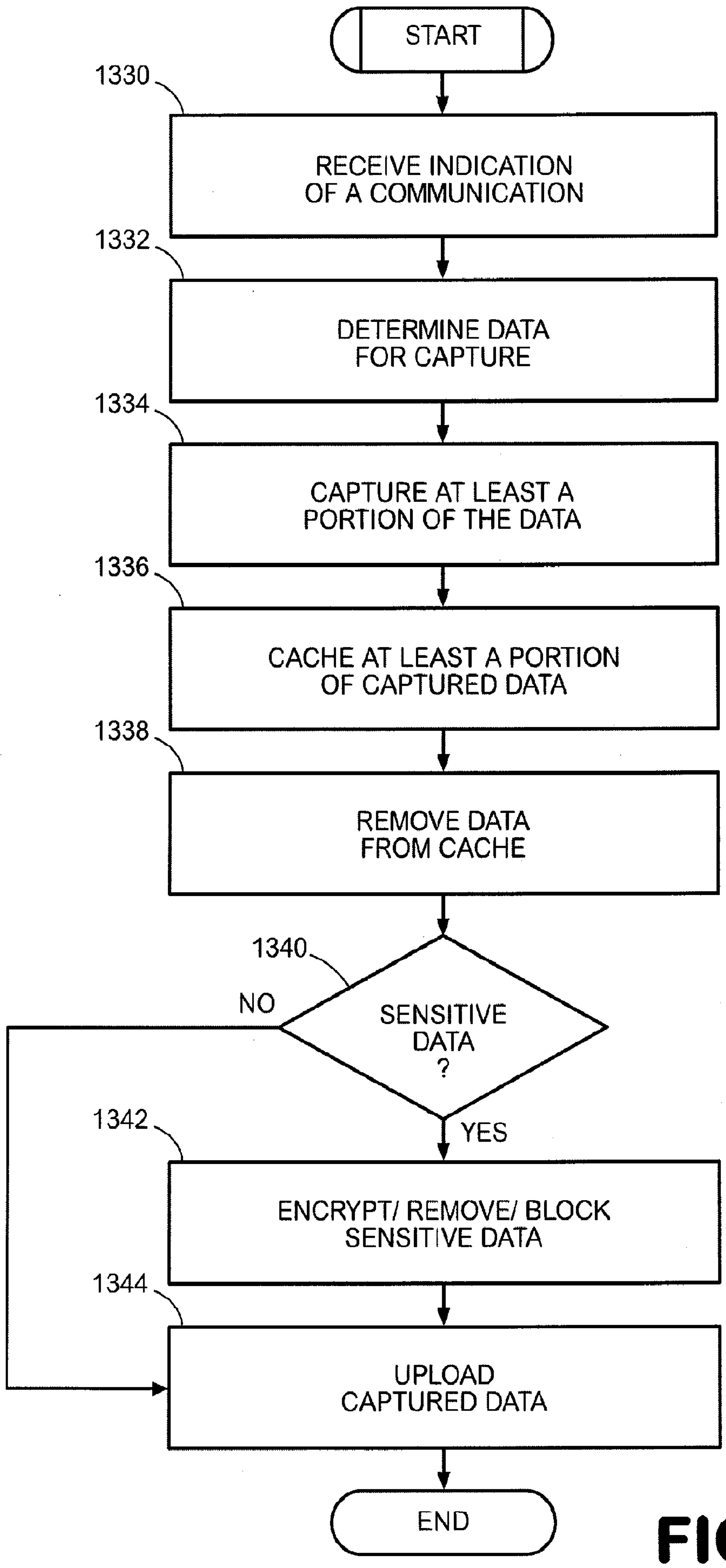

START
1330
RECEIVE INDICATION OF A COMMUNICATION
1332
DETERMINE DATA FOR CAPTURE
1334
CAPTURE AT LEAST A PORTION OF THE DATA
1336
CACHE AT LEAST A PORTION OF CAPTURED DATA
1338
REMOVE DATA FROM CACHE
1340
SENSITIVE DATA ?
NO
YES
1342
ENCRYPT/ REMOVE/ BLOCK SENSITIVE DATA
1344
UPLOAD CAPTURED DATA
END

FIG. 13