

SCALEABLE COMMUNICATIONS SYSTEM

5

BACKGROUND1. Field of the Invention

The present invention relates to a method and
10 apparatus for providing video and voice communications.

2. Discussion of Related Art

VoIP (voice over internet protocol (IP)) describes
a set of facilities for managing the delivery of voice
15 information using the public Internet or private
intranets or networks, rather than using the
traditional circuit-committed protocols of the public
switched telephone network (PSTN). A major advantage
of VoIP and internet telephony is that it avoids the
20 tolls charged by commercial telephone service providers
such as MCI™ or AT&T™.

VoIP derives from ITU-T H.323, a standard promoted
by the VoIP Forum. The H.323 standard defines the
format for sending audio, video, data, or fax using IP
25 on the public Internet and within intranets. In
addition to IP, VoIP uses the well known real-time
protocol (RTP) to ensure that communications are
provided in a timely manner to avoid noticeable pauses
in voice or video communications. Using public
30 networks, it is currently difficult to control the
latency. Better service is possible with private
networks managed by an enterprise or by an Internet
telephony service provider (ITSP).

Conventional techniques that support VoIP
35 communication require voluminous software to be
downloaded to a user's personal computer. With the low

speed communications links, users require much time to download the software. For personal computers with low memory storage capability, such as personal digital assistants (PDAs), VoIP communications may not be
5 feasible given the volume of software needed.

SUMMARY

In accordance with an embodiment of the present invention, a system is provided which allows at least
10 for audio communications between a personal computer and an audio communications device, such as a telephone. Video, data, and fax communications can also be supported.

In one embodiment of the present invention,
15 communications between a personal computer and audio communications device is supported by at least one intermediate device. For communications between a personal computer and audio communications device, the personal computer communicates with a entry node, which
20 selects an intermediate server. The personal computer next communicates with the selected intermediate server, which selects a ITSP gateway. The personal computer then communicates with the ITSP gateway. The ITSP gateway provides access to a PSTN which provides
25 access to the destination audio communications device. Thereby the personal computer communicates with the destination audio communications device.

One advantage of this embodiment is the software downloaded to the personal computer is less than if the
30 personal computer had to both select an ITSP and establish communication with the ITSP.

Another advantage of this embodiment is the architecture that supports VOIP communications is scaleable so that numerous VoIP communications can be
35 facilitated.

Another advantages of this embodiment is that it can use many varieties of the H.323 standard for VoIP communications.

5 In one embodiment, for client to remote client communications, communications between client and remote client are direct. In this embodiment, client and remote client communicate without need for the client to support H.323. Thus considerable software storage is avoided.

10 Various embodiments of the present invention will be more fully understood in light of the following detailed description taken together with the accompanying drawings.

15 BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 schematically depicts an embodiment of the present invention in which a system is used to provide communications between any client and any destination audio communication device.

20 FIG. 2 depicts an exemplary relationship between ITSP gateways of an ITSP network and the system, in accordance with an embodiment of the present invention.

FIG. 3 provides a more detailed diagram of a system which is in accordance with an embodiment of the present invention.

25 FIG. 4 provides a flow diagram of a process which is in accordance with an embodiment of the invention.

FIG. 5 schematically depicts messages communicated between a client, entry node, a selected intermediate server, and selected ITSP gateway during the process of FIG. 4, in accordance with an embodiment of the invention.

30 FIG. 6 provides a suitable flow diagram of an action of the process of FIG. 4, in accordance with an embodiment of the invention.

FIG. 7 depicts a sample web site used in one embodiment, in accordance with an embodiment of the invention.

FIG. 8 provides a suitable flow diagram of an action of the process of FIG. 4, in accordance with an embodiment of the invention.

FIG. 9 provides a suitable flow diagram of an action of the process of FIG. 4, in accordance with an embodiment of the invention.

FIG. 10 schematically depicts the relationship between multiple clients and a data base, in accordance with an embodiment of the invention.

Note that use of the same reference numbers in different figures indicates the same or like elements.

DETAILED DESCRIPTION

Incorporated materials

The following table provides a list of topics and materials related to such topics. The materials are incorporated herein by reference in their entirety.

Topic	Material
UDP/IP	Internet Engineering Task Force (IETF) Request for Comments 768.
TCP/IP	"TCP/IP And Related Protocols, McGraw-Hill Series On Computer Communications", Third Edition, Ulysses D. Black, McGraw-Hill Companies, 1997, ISBN: 0079132820, and "TCP/IP Illustrated Volume 1: The Protocols", W. Richard Stevens, Addison-Wesley, 1994.
H.323 and related sub protocols such as RTP	"Digital Subscriber Signaling System No. 1 Network Layer - ISDN User-Network Interface Layer 3 Specification for Basic Call Control" (ITU-T Recommendation Q.931) and "Line Transmission of Non-Telephone Signals

	- Control Protocol for Multimedia Communication (ITU-T Recommendation H.245).
--	--

Client to destination communications device

Overview

FIG. 1 schematically depicts an embodiment of the present invention in which system 100 is used to provide audio communications between any client 105-1 to 105-N and any destination audio communication device 150-1 to 150-Z, such as a conventional telephone. In this embodiment, any client 105-1 to 105-N accesses system 100 through a public or private network such as the internet. Hereafter client 105 refers to any client 105-1 to 105-N unless otherwise specified. Hereafter destination audio communication device 150 refers to any destination audio communication device 150-1 to 150-Z unless otherwise specified.

System 100 communicates with ITSP networks 140-1 to 140-Y using a public or private network such as the internet. Hereafter ITSP networks 140 refers to any ITSP networks 140-1 to 140-Y unless otherwise specified. At least one ITSP network communicates with a public switched telephone network (PSTN), which allows for communication with the destination audio communication device 150-1 to 150-Z.

FIG. 2 depicts an exemplary relationship between ITSP gateways 145-1 to 145-N of an ITSP network 140 and system 100. An ITSP network 140 is a private network that typically includes multiple ITSP gateways 145-1 to 145-N. Typically each ITSP gateway has a unique address, such as an IP address. For example, ITSP networks are provided by IDT of New Jersey and Qwest™. Each ITSP gateway is coupled to at least one public switched telephone network (PSTN) 148. PSTNs provide either wireline or wireless access to telephone units such as any of destination audio communication devices

150-1 to 150-2. Hereafter ITSP gateway 145 refers to any ITSP gateways 145-1 to 145-N unless otherwise specified.

Typically, each ITSP gateway uses a distinct
5 implementation of the H.323 protocol to support communications.

Interface between client 105 and system 100

An exemplary embodiment of client 105 is a
10 conventional personal computer (PC). A suitable PC includes an input/output adapter, central processing unit (e.g., a microprocessor), and a memory. Suitable PCs are, for example, a IBM PS/2 personal computer, Apple Macintosh computer, UNIX-based workstation, or a
15 PDA such as a Palm VII available from 3com™. The PC further includes a display such as a computer monitor such as a super VGA type or other visual display device.

The client 105 typically has resident thereon an
20 operating system such as the Microsoft Windows NT™ or the Microsoft Windows 95™ Operating System (OS), the IBM OS/2, the Apple MAC OS, or the UNIX OS such as the HP-UX OS.

The client 105 also includes a speaker and
25 microphone as well as a sound card to support broadcasting of audible information and transmission of audible information, such as speech of the user of the PC.

The PC user with access to the internet or a
30 private network can use a web browser such as Microsoft Internet Explorer™ or Netscape Navigator™ to access a web page with graphical content. To specify a web site, the user enters a universal resource locator (URL) specifying both the server and the specific data
35 ("web page") requested. The URL may specify a hyper-text transfer protocol (HTTP) or another transfer

protocol for communicating between the server and the browser. Using the internet, the URL is transmitted to the host server which stores information corresponding to the URL.

5 In one embodiment, client 105 accesses system 100 using a website stored in the network 110. A conventional website server, which is part of the network 110, hosts the website accessed by the client 105. A suitable website server is for example a SPARC
10 Server, available from SUN Microsystems™, and uses either Microsoft NT 4.0, IIS, or UNIX operating system.

 An exemplary website is described in more detail later with respect to FIG. 7. The website server supports requests from the user for features provided
15 by the website.

System 100

 FIG. 3 provides a more detailed diagram of system 100. System 100 includes a entry node 120 that
20 communicates with at least one intermediate server 130-1 to 130-X using a public or private network such as the internet. Each intermediate server 130-1 to 130-X is capable of communicating with every ITSP gateway 145-1 to 145-N using a public or private network such
25 as the internet. Of course, additional entry nodes 120 can be used to support more communication sessions.

 Entry node 120 of FIG. 3 monitors the activity of intermediate servers and selects an intermediate server to support audio communications from a distinct client
30 105, such as client 105-1. An exemplary embodiment of the entry node 120 is similar to the web server. The intermediate server selected by the entry node 120 to manage an audio communication from client 105 in turn selects an ITSP gateway that provides access to a PSTN
35 that provides communications with a destination audio communication device 150. Each intermediate server

130-1 to 130-X is similar to the web server. Hereafter intermediate server 130 refers to any intermediate server 130-1 to 130-X unless otherwise specified.

5 Database server

In one embodiment, a distinct database server (not depicted) uses a "Log Database" to store information on each user and processes requests for information related to a client 105 or a specific user. A suitable database server is a conventional server. Exemplary communications to and from the database server are provided in Appendix B.

Software implementation overview

15 Appendix A, an integral part of this disclosure, describes code segments that are executed by client 105, entry node server 120, and intermediate servers 130-1 to 130-X, in this embodiment. The code segments are used to perform process 400 of FIG. 4 that communications between client 105 and a destination audio communications device.

20 In action 405 of process 400, client 105 contacts entry node 120 to initiate use of system 100. Detailed description of action 405 is provided with respect to FIG. 6.

25 In action 410, the entry node 120 selects a suitable intermediate server 130. The entry node then identifies the selected intermediate server 130 to the client 105. Detailed description of action 410 is provided with respect to FIG. 8.

30 In action 415, the selected intermediate server 130 selects and establishes a communications channel with an ITSP gateway 145 and identifies the selected ITSP gateway to the client 105. Subsequently, the client 105 and ITSP gateway 145 communicate directly. As stated earlier, the ITSP gateway 145 communicates

with a PSTN, which provides communications with the destination audio communications device 150. Using the ITSP gateway 145, the client 105 communicates with the destination audio communications device 150. Detailed description of action 415 is provided with respect to FIG. 9.

In action 420, the client 105 ends the audio communications session. Detailed description of action 420 is provided below.

One advantage of this embodiment is that communication between the client 105 and destination audio communications device 150 is segmented so that selection and establishment of communications channels between the client 105 and ITSP gateway 145 is performed using entry node 120 and intermediate server 130. Thereby the software downloaded to the client 105 is less than if the client had to select and establish connection with an ITSP gateway (so called "thin client" environment).

Exemplary Messages

FIG. 5 schematically depicts messages communicated between the client 105, entry node 120, a selected intermediate server 130, and selected ITSP gateway 145 during process 400 during an exemplary operation of actions 405 to 420. Appendix D provides a description of a suitable protocol used to transmit the messages as well as a description of other messages.

Action 405

FIG. 6 provides a suitable flow diagram of action 405 (FIG. 4). In action 610, the user, using a graphical web browser such as Netscape Navigator or Microsoft Explorer, supplies a universal resource locator (URL) or uses a hyperlink to access a web site that includes functionality in accordance with an

For example, FIG. 7 depicts a sample web site 700.

In another embodiment, web site 700 provides users access to electronic mail, voice mail messages, and faxes. Websites that provide access to electronic mail, voice mail messages, and faxes are well known; see for example, Hotmail.com™, Excite.com™, and jfax.com™. Thus in this embodiment, web site 700 and system 100 allow users to respond vocally to voice mail messages, electronic mails, or faxes.

Referring to FIG. 6, in action 620, the user enters a destination phone number, for example, into field 704 of the web site 700.

In one embodiment, the phone book is stored in a data base with data base server, described earlier. Commands to the data base are described in more detail in Appendix B.

In position 708, a conventional graphical banner ad is displayed. Advertisements can be scheduled by use of a service available from Doubleclick of New York, New York, for example. Advertisements can be
5 tailored to the user's interests. By providing advertisements that are targeted to specific users, advertising revenue can be maximized. In this embodiment, advertising revenue can be used to pay for costs of communication with the destination audio
10 communications device 150.

Of course, by engaging the graphical banner advertisement, a user can purchase the product or service advertised.

In action 630, the client connects to a entry node
15 120 using, for example, the TCP/IP protocol. Action 630 corresponds to the message entitled TCP Connect 502 (FIG. 5). The IP Address and port number of the entry node 120 are preconfigured in the files downloaded in action 610. In action 630, the client communicates at
20 least the destination phone number, User ID, Password, and Session ID to the entry node 120. The User ID identifies a user and identifies information, such as described in Appendix B, associated with the user that is stored by the data base server. The Password is
25 used by the web server to determine whether to permit access to the user's information from the data base. The Session ID is used to track the status of the voice communication.

In action 640, the entry node 120 provides the
30 contact information of a selected intermediate server 130 to the client 105 using, for example, the TCP/IP protocol. Referring to FIG. 5, action 640 corresponds to the message entitled VEGA Server 504. In this embodiment, the contact information includes the IP
35 address and port number of the selected intermediate server. In this embodiment, the IP Address is a 4 byte

IPv4 address (IP version 4) of the selected intermediate server 130 and the port number is a 2 byte port number that selected intermediate server monitors for future communications from the client 105.

5 In this embodiment, the entry node 120 maintains a table of potential intermediate servers 130 as well as the percent of active communications the intermediate server 130 maintains. Appendix C, an integral part of this disclosure, provides a description of a suitable
10 technique whereby the entry node 120 monitors the activity level of intermediate servers 130. In this embodiment, the entry node 120 selects an intermediate server 130 that has the lowest activity percentage (e.g., Number of Active Sessions divided by Max Number
15 of Sessions). If all potential intermediate servers 130 are idle to the same extent, the entry node selects an intermediate server 130 having the lowest associated identification number. Other selection methods can be used, such as random selection.

20

Action 410

FIG. 8 provides a suitable flow diagram of action 410 (FIG. 4). Appendix A provides a description of programs used in action 410.

25 In action 810, the client 105 connects to the selected intermediate server 130 using the IP address and port number, both specified earlier in action 610, using for example, TCP/IP. In the example of FIG. 5, action 810 corresponds to (second occurring) "TCP
30 Connect 506".

 In action 820, the client 105 initializes connection with the selected intermediate server 130 by, for example, issuing a message entitled VSCP SETUP (item 508, FIG. 5) to the selected intermediate server
35 130 by using for example TCP/IP. Message VSCP SETUP initializes any communications between client 105 and

the selected intermediate server 130. The following table provides a description of fields used in message VSCP SETUP.

Field Name	Octet (byte) no.	Brief description
User ID	1 to 8	User ID
Password	9 to 16	password string
Client-side Media Transport Address Type	17 to 18	Type of the media transport address. IPv4 is supported although other protocols can be.
Client-side Media Transport Address	19 to 22	Client-side IP Address for media transport.
Client-side Media Transport Port Number	23 to 24	Client-side Port number for media transport.
Calling Party Number Length	25 to 26	Length of the Calling Party Number.
Calling Party Number	27 to N, where N is the length of the called party number.	Called Party Number Length. (E.164 - Maximum 128)
Called Party Number Length	N to N+1	Length of called party number. (word)

Called Party Number	N+2 to M, where M is the length of the called party number.	Called Party Number (E.164 - Maximum 128)
Client-side Protocol Type	M to M+1	Type of protocol used for media streaming. RTP is supported although other protocols can be used.
Client-side Media Type	M+2 to M+3	See below.
Client-side Payload Type	M+4 to M+5	See below.
Client-side Payload Max Frame	M+6 to M+7	Maximum size of the payload.
Client-side Silence Suppression Enable	M+8	Silence suppression enabled.
Client-side Echo Cancellation Enable	M+9	Echo cancellation enabled.

Field Client-side Media Type represents the transmitted media type.

5

Media Type	Value
Voice	0x0001
Video	0x0002
Data	0x0003

Fax	0x0004
-----	--------

Field Client-side Payload Type represents the type of compression algorithm used to encode the selected media type, such as voice, video, or data. Currently, encoded voice payloads are supported.

Payload Type	Value
Non-Standard	0x0001
G.711 A-Law 64K	0x0002
G.711 A-Law 56K	0x0003
G.711 u-Law 64K	0x0004
G.711 u-Law 56K	0x0005
G.722 64K	0x0006
G.722 56K	0x0007
G.723.1	0x0008
G.728	0x0009
G.729	0x000a
G.729 Annex A	0x000b
IS11172	0x000c
IS13818	0x000d
G.729 Annex B	0x000e
G.729 Annex A Annex B	0x000f
G.723.1 Annex C	0x0010
GSM Full Rate	0x0011
GSM Half Rate	0x0012
GSM Enhanced Full Rate	0x0013

In action 830, the selected intermediate server 130 determines whether to allow the client 105 to begin a voice or video based communication. The selected intermediate server 130 performs authentication, which includes determining if the client 105 having the specified Userid and password is permitted to use the selected intermediate server 130.

If the client 105 is not permitted, the selected intermediate server 130 issues a RELEASE message to the client 105 (not depicted in FIG. 5). Message RELEASE is described in more detail later. Examples of reasons
5 for not permitting a client 105 include unknown User ID, incorrect password, and unsupported phone number. See below for a description of the message RELEASE.

If the client 105 is permitted, then in action 840, the selected intermediate server 130 selects an
10 ITSP gateway (hereafter "selected ITSP gateway"). An exemplary manner to select an ITSP gateway is to determine a least-cost available route for the destination phone number. Another exemplary manner to select an ITSP gateway 145 is to use a load balancing
15 method whereby the ITSP gateway 145 with the least number of active sessions is selected. Load balancing advantageously lessens the impact of failure of a single ITSP gateway 145. Another exemplary manner to select an ITSP gateway 145 is to select an ITSP
20 gateways based on a general hierarchy. The selected ITSP gateway 145 is used to make a connection with the PSTN 148, through which the destination phone number is connected. The selected ITSP gateway is identified by an IP address.

25 In action 850, the selected intermediate server 130 communicates with the selected ITSP gateway 145. A suitable technique to connect with the selected ITSP gateway 145 uses the H.323 protocol, both well known to those of skill in the art. In action 850, the Q.931
30 and H.245 protocols are conventional and described in the H.323 protocol. See for example "Digital Subscriber Signaling System No. 1 Network Layer - ISDN User-Network Interface Layer 3 Specification for Basic Call Control" (ITU-T Recommendation Q.931) and "Line
35 Transmission of Non-Telephone Signals - Control Protocol for Multimedia Communication (ITU-T

Recommendation H.245), which are incorporated herein by reference in their entirety.

As each ITSP gateway 145 may use a distinct style of H.323, the selected intermediate server 130
5 advantageously can support each distinct style of H.323. Without use of the selected intermediate server 130, the client 105 may have to support each distinct style of H.323. Such support may require voluminous software be uploaded to the client 105.

10 FIG. 9 schematically depicts operations of action 850 in more detail. In this embodiment, the communications of action 850 use the TCP/IP protocol.

In action 910, a connection with the specified destination audio communications device 150 is
15 established. In one embodiment, action 910 further includes the actions 910-1 to 910-7.

In action 910-1, the selected intermediate server 130 provides the destination PSTN phone number to the selected ITSP gateway 145 to initiate a call using the
20 conventional SETUP message which is defined by the Q.931 protocol. Action 910-1 corresponds to Q.931 SETUP 510 of FIG. 5.

In action 910-1, the selected intermediate server 130 next issues a message CALLPROCEED to the client
25 (item 512, FIG. 5). The CALLPROCEED message communicates to the client 105 that the current call request is currently in processing state. In this embodiment, message CALLPROCEED includes an 8 byte value that identifies the call. Message CALLPROCEED
30 includes a unique Call Reference Value of, e.g., 8 bytes, that identifies a particular communication session. The selected intermediate server 130 maintains the status of each ongoing session.

In action 910-2, the selected ITSP gateway 145
35 dials the destination PSTN phone number. Action 910-2 corresponds to Dial 514 of FIG. 5.

In action 910-3, the PSTN signals to the selected ITSP Gateway 145 that the destination audio communications device 150 is notifying the destination user of an incoming voice communication. Action 910-3
5 corresponds to Ringback 516 of FIG. 5.

In action 910-4, the selected ITSP gateway 145 acknowledges to the selected intermediate server 130 of notification by the PSTN to the destination audio communications device using the conventional ALERT
10 message which is defined by the Q.931 protocol. Action 910-4 corresponds to Q.931 ALERT 518 of FIG. 5.

In action 910-5, the selected intermediate server 130 signals to the client that the destination user has been alerted and provides a "Call Reference Value",
15 which is described later. Action 910-5 corresponds to the VSCP ALERT 520 of FIG. 5.

In action 910-6, the PSTN communicates to the selected ITSP gateway that the destination audio communications device has accepted the communication.
20 Action 910-6 corresponds to Answer 522 of FIG. 5.

In action 910-7, the selected ITSP gateway 145 communicates to the selected intermediate server 130 that the destination audio communications device has accepted the communication using the conventional
25 connect signal which is defined by the Q.931 protocol. Action 910-7 corresponds to Q.931 CONNECT 524 of FIG. 5.

In action 920, the communication standard between the selected intermediate server 130 and selected
30 gateway 145 is set. In one embodiment, action 920 includes actions 920-1 to 920-2.

In action 920-1, the selected intermediate server 130 communicates the compression technique specified in action 820 and the media type (e.g., voice, video,
35 data, or fax) to the selected ITSP gateway 145 using a conventional terminal capability set (TCS) message

which is defined in the H.245 protocol. Action 920-1 corresponds to H.245 TCS 526 of FIG. 5.

In action 920-2, the selected ITSP gateway 145 signals receipt of the TCS message of action 935 to the
5 selected intermediate server 130 using the conventional TCS ACK message which is defined in the H.245 protocol. Action 920-2 corresponds to the TCS ACK 528 of FIG. 5.

In action 930, a communication relationship is established between selected intermediate server 130
10 and selected ITSP gateway 145. In one embodiment, action 930 includes actions 930-1 to 930-3.

In action 930-1, the selected intermediate server 130 identifies itself as the master to the selected
ITSP gateway 145 using the conventional master slave
15 determination (MSD) message that is defined in the H.245 protocol. Action 930-1 corresponds to the H.245 MSD 530 of FIG. 5.

In action 930-2, the selected ITSP gateway 145 signals receipt of the MSD of action 945 to the
20 selected intermediate server 130 by sending the conventional MSD ACK that is defined in the H.245 protocol. Action 930-2 corresponds to the H.245 MSD ACK 532 of FIG. 5.

In action 930-3, the selected intermediate server
25 130 signals to the selected ITSP gateway 145 that the master-slave process is complete using the conventional master MSD message that is defined in the H.245 protocol. Action 930-3 corresponds to the H.245 MSD ACK 534 of FIG. 5.

30 In action 940, a voice channel is established between selected intermediate server 130 and selected ITSP gateway 145. In one embodiment, action 940 includes actions 940-1 to 940-4.

In action 940-1, the selected intermediate server
35 130 signals to the selected ITSP gateway 145 to open a voice channel from the selected intermediate server 130

signals to the selected ITSP gateway 145 using the conventional open logical channel (OLC) as defined in H.245. Action 940-1 corresponds to the H.245 OLC 536 of FIG. 5.

5 In action 940-2, the selected ITSP gateway 145 signals receipt of the OLC of action 965 to the selected intermediate server 130 using the conventional OLC ACK of H.245. Action 940-2 corresponds to the H.245 OLC ACK 538 of FIG. 5.

10 In action 940-3, the selected ITSP gateway 145 signals to the selected intermediate server to open a voice channel from the selected ITSP gateway to the selected intermediate server 130 signals using the conventional open logical channel (OLC) as defined in H.245. Action 940-3 corresponds to the H.245 OLC 540 of FIG. 5.

20 In action 940-4, the selected intermediate server 130 signals receipt of the OLC of action 975 to the selected ITSP gateway using the conventional OLC ACK as defined in H.245. Action 940-4 corresponds to the H.245 OLC ACK 542 of FIG. 5.

25 Referring to FIG. 8, in action 860, the selected intermediate server 130 notifies the client that a call is being connected by sending a CONNECT message, which corresponds to VSCP CONNECT 544 of FIG. 5. The following table provides a description of CONNECT.

Field Name	Octet no.	
Call Reference Value	1 to 8	Call Reference Value provided in the previous ALERT message.
Server-side Media Transport Address Type	9 to 10	Type of the media transport address. IPv4 is supported, although other protocols can be used.

Server-side Media Transport Address	11 to 14	Address (IP Address, in this embodiment) of the selected ITSP gateway.
Server-side Media Transport Port Number	15 to 16	Port number of the selected ITSP gateway.
Called Party Number Length	17 to 18	Length of the Called Party Number.
Called Party Number	19 to N, where N is the length of the called party number.	Called Party Number (E.164 - Maximum 128)
Server Side Protocol type	N to N+1	Protocol used for media streaming. RTP is used, although other protocols can be used.
Server-side Media Type	N+1 to N+2	Similar to the client-side media type described earlier.
Server-side Payload Type	N+3 to N+4	Similar to the Client-side Payload Type described earlier.
Server-side Payload Max Frame	N+5 to N+6	Maximum size of the payload.
Server-side Silence Suppression Enable	N+7	Silence suppression enabled.

Server-side Echo Cancellation Enable	N+8	Echo cancellation enabled.
--	-----	-------------------------------

Action 415

In action 415 (FIG. 4), the client 105 communicates with the selected ITSP gateway 145 using, for example, the RTP protocol. RTP is an independent standard protocol for encoding and transmitting multimedia streaming that is well known to those of ordinary skill in the art. Action 415 corresponds to RTP Data 546 of FIG. 5. Of course, other protocols can be used besides RTP, such as any other proprietary media streaming protocol.

In accordance with the RTP standard, program "strtp.dll", described in Appendix A, provides transmission of encoded voice. An exemplary technique to encode and decode voice is available from the DSP Group of Santa Clara, CA. In this embodiment, program "tsd2.dll" includes the exemplary technique to encode and decode voice.

Thus voice streams are communicated directly between the client and selected ITSP gateway. This direct communication reduces delay of voice packet communications. Such delays introduce noticeable delays in voice communications.

In one embodiment, brief audio ads can be provided to users through the speaker of the client 105 prior to commencing an audio communication. The audio ads generate revenue that can be used to subsidize costs of communicating with the destination audio communications device 150.

Action 420

In action 420 (FIG. 4), the voice communication ends. For example, a client 105 issues a RELEASE (548,

FIG. 5) to the selected intermediate server 130 when a user desires to terminate a talk. Similarly, the selected intermediate server 130 issues a RELEASE to the client 105 using, for example, TCP/IP, when a call recipient wishes to terminate a talk. In this embodiment, the "VEGA Server Source Files" described in Appendix A are used in action 420.

One exemplary manner to terminate a voice communication is as follows. The selected intermediate server 130 signals to terminate the H.323 session using for example a H.245 ESC message as defined in the H.245 standard to the selected ITSP gateway (see H.245 ESC 550 of FIG. 5). The selected ITSP gateway 145 then signals to terminate the H.323 session using for example a H.245 ESC message as defined in the H.245 standard to the selected intermediate server 130 (see H.245 ESC 552 of FIG. 5). Next, the selected ITSP gateway 145 issues a hangup command to the PSTN to discontinue a session (see Hangup 554 of FIG. 5). Next, the selected intermediate server 130 communicates to the selected ITSP gateway 145 to terminate the session using for example the Q.931 disconnect message defined in the Q.931 protocol (see Q.931 disconnect 556 of FIG. 5).

Thereafter, the selected intermediate server 130 frees resources such as other temporary allocated data structures used for the ended session for another session. Also the selected intermediate server 130 logs the end of the call to a Log Database. The Log Database records the Start Time, End Time, Success/Failure, Caller, Callee phone number. The Log Database is stored by the database server, described earlier and in Appendix B.

DTMF support

When the user presses a digit on the website 700 (for example, for voice mail), the client 105 sends the digit to the selected intermediate server 130 using the DTMF message. The selected intermediate server 130, in turn, transfers the DTMF message to the selected ITSP gateway using an H.245 message.

The following table provides parameters of the DTMF message.

Field Name	Octet no.	Brief Description
Call Reference Value	1 to 8	Described earlier.
DTMF String Length	9 to 10	The length of the DTMF String.
DTMF String	11 to N, where N is the length of the DTMF String.	DTMF characters. A maximum length of the DTMF String is, for example, 128 bytes.

10

Communication halts

At any time, communication halt can be generated by either the client, a selected intermediate server 130, or a remote client using VSCP RELEASE. For example, some communications or messages between a client and selected intermediate server 130 or remote client must be received within an allotted time. If the communications or messages are not received within an allotted time, the receiver issues a RELEASE message to the communication/ message sender to discontinue any communication on an initiated or active session.

The RELEASE message is used to end a TCP connection. For example, the RELEASE message is used if the selected intermediate server or client do not receive a message, such as SETUP, CALLPROCEED, ALERT,

or CONNECT, in an allotted time. The following table provides parameters of the RELEASE message.

Field Name	Octet no.
Call Reference Value	1-8
Reason	9-10

5 The Call Reference Value field identifies a particular communication session. The following table provides potential values and associated meanings for each Reason field.

Reason	Value
Unknown reason	0x0001
No permission	0x0002
Unreachable Destination	0x0003
Destination Rejection	0x0004
User	0x0005

10 Client to remote client

In client to remote client communication, the client and remote client communicate directly using the internet or a private network. The clients communicate using messages TCP connect, VSCP SETUP, VSCP
 15 CALLPROCEED, VSCP ALERT, VSCP CONNECT, RTP Data, and VSCP RELEASE described earlier at least with respect to FIG. 5. Thus in this embodiment, each of client and remote client use only the "Client side programs", described in Appendix A. The client initiates
 20 communication with a remote client using the TCP connect message, in which the IP address and port of the remote client are specified. Thus, for example, the client and remote client are each identified by an IP address and port number.

25 Advantageously, each client does not need to support the full H.323 protocol, which would be

necessary to communicate with an ITSP gateway. Rather, for client to remote client communications, only the RTP needs to be supported. Thus, the software needed by each client is minimized. Should the client request
5 communication with a destination audio communications device, a selected intermediate server supports any necessary distinct implementation of the H.323 protocol needed to communicate with any ITSP gateway.

10

Modifications

The above-described embodiments of the present invention are merely meant to be illustrative and not limiting. It will thus be obvious to those skilled in the art that various changes and modifications may be
15 made without departing from this invention in its broader aspects. For example, system 100 could easily support video, data, or fax communications. Therefore, the appended claims encompass all such changes and modifications as fall within the true scope of this
20 invention.

APPENDIX A

Entry node programs

A program "vgk.c" is executed by the entry node 120 to support communications between intermediate
5 servers and any client. Program "vgk.c" is used in action 405 (FIGs. 4 and 8).

Client side programs

The "Client side programs" are executed by the
10 client 105 and support any communications to and from the client 105. Collectively, the client side programs are used in actions 610 and 620 of FIG. 6. In this embodiment, code segments provided in tables entitled
"Java files", "Resource Files", "Microsoft COM Wrapper
15 Class(Internet Explorer only)", "Microsoft COM (Common Object Model) related files for Internet Explorer",
"Netscape Plugin related files for Netscape Communicator", "C++ Compiler(Microsoft Visual C++ 6.0) related files", "C++/Header Files", and "External
20 Dependencies" are uploaded to a client.

Program "vegacomm.cpp", "vegacomm2.cpp", and "vegacomm.h" are compiled together to make up the main client side application.

Together, all of code segments listed under the
25 C++/Header Files table make up "vscp.dll". Executing program "vscp.dsw" commences compilation of "vscp.dll".

Java files are used to build the Java applet. It is the main module of the client application. All the direct user interaction and user interface are
30 implemented in these files.

Resource files contain the user interface and multimedia object definitions.

The COM files are used as an interface between Java Applet and the *.dll files.

35 The Netscape Plugin files allow control the java applet to control the sound card. Netscape plugin

files for Netscape serves the same role that the COM module does for MS Internet Explorer.

- File "strtp.dll" supports RTP communication and voice encoding/decoding functions between the client
- 5 105 and any destination device (for example, PC to phone communications) or any remote client (for example, PC to PC communications).

Java files		
File	Description	Comment
AnswerDialog.java	Yes/No message dialog class	
BrowserType.java	class for detecting surrounding browser type	
ContactInfo.java	Contains personal information and location information of Client.	
DialDialog.java	Dialing Dialog class	
DialDialogEvent.java	Dialing Dialog event class	
MySocket.java	socket utility class	
MyUtil.java	utility class for frequently used methods	
VegaCommEvent.java	VegaCommEvent class	
VegaCommWrapper.java	browser-independent class for internal vscp methods	
Web2PhoneApplet.java	main Java Applet class which communicates with Browsers and vscp engine (plugin or	

	COM)	
Web2PhoneHelper.java	Front-end class for Web2PhoneApplet class	

Resource Files		
File	Description	Comment
vscp.rc	vscp dll resource template, contains version information and resource data	
VegaComm.rgs	resource file for registering VegaComm COM class to Windows registry	
busy.wav	busy sound	Microsoft PCM 8000 Hz, 8 bit, mono format
dialtone.wav	dial tone sound	Microsoft PCM 8000 Hz, 8 bit, mono format
ringback.wav	ring sound	Microsoft PCM 8000 Hz, 8 bit, mono format
Tone0.wav	button 0 sound	Microsoft PCM 8000 Hz, 8 bit, mono format
Tone1.wav	button 1 sound	Microsoft PCM 8000 Hz, 8 bit, mono format
Tone2.wav	button 2 sound	Microsoft PCM 8000 Hz, 8 bit, mono format
Tone3.wav	button 3 sound	Microsoft PCM 8000

		Hz, 8 bit, mono format
Tone4.wav	button 4 sound	Microsoft PCM 8000 Hz, 8 bit, mono format
Tone5.wav	button 5 sound	Microsoft PCM 8000 Hz, 8 bit, mono format
Tone6.wav	button 6 sound	Microsoft PCM 8000 Hz, 8 bit, mono format
Tone7.wav	button 7 sound	Microsoft PCM 8000 Hz, 8 bit, mono. format
Tone8.wav	button 8 sound	Microsoft PCM 8000 Hz, 8 bit, mono format
Tone9.wav	button 9 sound	Microsoft PCM 8000 Hz, 8 bit, mono format
Tonestar.wav	button star(*) sound	Microsoft PCM 8000 Hz, 8 bit, mono format
Tonepnd.wav	button pound(#) sound	Microsoft PCM 8000 Hz, 8 bit, mono format

Microsoft COM Wrapper Class(Internet Explorer only)		
File	Description	Comment
_IVegaCommEvent.java	IVegaCommEvent wrapper for MS Java	generated from vscp.dll by Microsoft Visual J++ COM Wrapper
IVegaComm.java	IVegaComm wrapper for MS	generated from vscp.dll

	Java	
tagCodecConstants.java	Codec constants definition class	generated from vscp.dll
tagStatusConstants.java	status constants definition class	generated from vscp.dll
VegaComm.java	VegaComm wrapper for MS Java	generated from vscp.dll

Microsoft COM(Common Object Model) related files for Internet Explorer		
File	Description	Comment
vscp.idl	IDL(Interface Definition Language) file which defines the COM interfaces exposed to external Java classes	
vscpCP.h	C stub file for COM Connection Point interface	generated from vscp.idl
vscp.h	C stub file for COM interface	generated from vscp.idl
vscp_i.c	C source file which has declarations for IID(Interface ID) and CLSID(Class ID)	generated from vscp.idl
vscp_p.c	C source file which contains proxy stub code for Distributed COM	generated from vscp.idl not using.

Netscape Plugin related files for Netscape Communicator

File	Description	Comment
npvscp.cpp	C language implementation file for basic Netscape Plugin methods, VegaPlugin methods, VegaCommEvent methods	uses COM internally
npwin.cpp	Win32 interface for Netscape Plugin	from Netscape PluginSDK source
stubs.c	stub files for Netscape LiveConnect (Java to Plugin communication) technology	
VegaPlugin.java	JRI (Java Raw Interface) declaration file for Netscape Plugin methods	
VegaCommEvent.java	JRI (Java Raw Interface) declaration file for status update events	
com_serome_VegaCommEvent.c	Cross-language Interface file for status update events methods which calls external java	generated from VegaCommEvent.java

	classes	
com_serome_VegaCommEvent.h	Header file for status update events methods which calls external java classes	generated from VegaCommEvent.java
com_serome_VegaPlugin.c	Cross-language Interface file for Netscape Plugin methods which is called by external java classes	generated from VegaPlugin.java
com_serome_VegaPlugin.h	Header file for Netscape Plugin methods which is called by external java classes	generated from VegaPlugin.java
SinkObj.h	JRI<->COM communication class(CSinkObj)	

C++ Compiler(Microsoft Visual C++ 6.0) related files		
File	Description	Comment
vscp.def	declares DLL export functions for use with COM and Netscape Plugin	
vscp.clw	C language class information file	MSVC internal
vscp.dsp	Project File information	

vscp.dsw	Project Workspace information	
stdafx.cpp	files for standard AFX features	MSVC internal

C++/Header Files		
File	Description	Comment
VegaComm.cpp	implementation file for key features (CvegaComm)	main file
VegaComm2.cpp	implementation file for key features (CvegaComm)	
VegaComm.h	header file for VegaComm.cpp, VegaComm2.cpp	
WaveCtrl.cpp	Manipulates Microphone and Speaker hardware devices (CwaveCtrl)	Recording & Playing
WaveCtrl.h	header file for WaveCtrl.cpp	
stvscpp.cpp	Frames/Deframes VSCP packets	
stvscpp.h	VSCP header files	
vscpp.cpp	dll main file	
vscpp.h	dll main header file	

External Dependencies		
File	Description	Comment
stnet.h	network API functions	
stnet.dll	network API functions dll	
sterror.h	stnet error definition	
strtp.h	RTP header file	
strtp.dll	RTP DLL	Supports RTP

		communication and voice encoding/decoding functions.
stg711.h	G.711, G.723 codec DLL header file	
tsd2.dll	G.711, G.723 codec DLL	

Intermediate server programs

The "VEGA Server Source Files" are executed by the selected intermediate server and support any

5 communications to and from the selected intermediate server.

VEGA Server Source Files		
File	Description	Comment
Child.c	Child process main routine and thread routine which handles each call	
h323stub.c	H.323 stack calling routines for VEGA server	
Sendvscp.c	VSCP packets sending routines	
Utils.c	Miscellaneous routines	
Vserver.c	Main routine which creates child processes and routes incoming calls to child processes	
stdb.c	Performs user authentication by looking up the DB	
Stdebug.c	Debugging/Logging facility	

H.323 programs

The programs of the "H.323 Stack Packages" are executed by the selected intermediate server and support any H.323 level communications between the selected intermediate server and the ITSP gateway. The programs of the H.323 Stack Packages are used to support an H.323 protocol based communication in process 850 of FIG. 9.

10

H.323 Stack Packages		
Package	Description	Comment
Stasn	ASN.1 facilities using OSS ASN.1 package	
sth245	H.245 abstraction layer (Session management)	
sth323m	H.323 Multi-session (Toplevel API)	
Stnet	TCP/IP abstraction layer	
stq931	Q.931 and H.225.0 abstraction layer (Call signaling)	
Stras	RAS abstraction layer (Registration, Admission and Status)	
Strtp	RTP abstraction layer (Real-Time Protocol)	
Stwin32	Win32 functions ported on Solaris 2.x	

The following table provides a correspondence between actions of process 400 and the programs described in Appendix A.

Actions	Programs
405, 410, 415, 420.	C++/Header Files, C++ Compiler(Microsoft Visual C++ 6.0) related files, Netscape Plugin related files for Netscape Communicator, Microsoft COM(Common Object Model) related files for Internet Explorer Microsoft COM Wrapper Class(Internet Explorer only), Resource Files, and Java files.
415	External Dependencies
410, 415, and 420	Intermediate Server Source Files
850	H.323 Stack Packages

APPENDIX B: DATABASE

The following commands are issued by the web server to the data base server. The data base server
 5 replies using responses, described herein. FIG. 10 schematically depicts the relationship between clients 105-i and 105-j and the data base 1000. In one embodiment, the commands are transmitted using TCP/IP.

Commands

10 The following table shows commands and associated values.

Command	Value
LOGIN	0x10
REGISTER	0x11
LOGOFF	0x12
GETSTATUS	0x13
UNREGISTER	0x14
ADDUSER	0x20
DELETEUSER	0x21
MODIFYUSER	0x22
GETUSER	0x23
ADDCONTACT	0x30
DELETECONTACT	0x31
MODIFYCONTACT	0x32
GETCONTACTLIST	0x33
GETCONTACTINFO	0x34
FINDUSERNAME	0x50

Return Codes

The following table shows the return codes and
 15 associated values. The return codes are generated in response to a command.

Command	Value
RETCODE_SUCCESS	0x00
RETCODE_INVALID_USERID	0x01
RETCODE_INVALID_CONTACTID	0x02
RETCODE_INVALID_USERNAME	0x03
RETCODE_INVALID_PASSWORD	0x04
RETCODE_INVALID_IPADDRESS	0x05
RETCODE_INVALID_PORT	0x06
RETCODE_SERVER_ERROR	0x07
RETCODE_ALREADY_EXIST	0x08
RETCODE_INVALID_FOLDERID	0x09
RETCODE_INVALID_FOLDERNAME	0x0a
RETCODE_LOGIN_FIRST	0x0b
RETCODE_INVALID_SESSIONID	0x0c
RETCODE_ALREADY_CONNETED	0x0d
RETCODE_NO_OWNER	0x0e

LOGIN

The LOGON command is a command to logon to the selected intermediate server. The LOGON command format is as follows.

5

Field Name	Octet no.	format
Command	1	0x10
User Name	2-11	Null-terminated ASCII String
Password	12-21	Null-terminated ASCII String

Response to LOGIN

The format of the response to the LOGIN command follows.

Field Name	Octet no.
Return Code	1
Session ID	2-33

The Return Codes are any of: RETCODE_SUCCESS, RETCODE_INVALID_USERNAME, RETCODE_INVALID_PASSWORD, or RETCODE_SERVER_ERROR. The Session ID is described
5 earlier.

REGISTER

The REGISTER command registers the client's IP Address/Port information to the database. The REGISTER
10 command format is as follows.

Field Name	Octet no.
Command	1
Session ID	2-33
IP Address	34-37
Port Number	38-39
Override Flag	40

Command: 0x11

Session ID: described earlier

IP Address (IPv4)

15 Port Number: Port Number of selected intermediate server

Override flag is either

NO_OVERRIDE (0x0) : Do not replace the information if already exists

20 OVERRIDE (0x01): Replace the information if already exists.

Response to REGISTER

The format of the response to the REGISTER command follows.

Field Name	Octet no.
Return Code	1
User ID	2-5

The Return Code is any of: RETCODE_SUCCESS,
 5 RETCODE_INVALID_SESSIONID, RETCODE_INVALID_IPADDRESS,
 RETCODE_INVALID_PORT, RETCODE_ALREADY_CONNECTED, or
 RETCODE_SERVER_ERROR.

The User ID is described earlier.

10

LOGOFF

LOGOFF command specifies which session is to end.
 The LOGOFF command format is as follows.

Field Name	Octet no.	Brief description
Command	1	0x12
Session ID	2-33	Described earlier.

15 Response to LOGOFF

No Response

GETSTATUS

GETSTATUS command is a request for status
 information of a user The GETSTATUS command format is
 20 as follows.

Field Name	Octet no.	Brief description
Command	1	0x13

User ID	2-5	Described earlier
User Name	6-15	Described earlier

Response to GETSTATUS

The format of the response to the GETSTATUS command follows.

Field Name	Octet no.
Return Code	1
Status	2
IP Address	3-6
Port Number	7-8

5

The Return Code is any of the following:
 RETCODE_SUCCESS, RETCODE_INVALID_USERID,
 RETCODE_INVALID_USERNAME, or RETCODE_SERVER_ERROR.

The Status is any of the following:

- 10 STATUS_NOTCONNECTED (0x00): User client not registered
 STATUS_CONNECTED (0x01): User client registered
 IP Address is described earlier.
 Port Number is described earlier.

15 UNREGISTER

The UNREGISTER command is a request to unregister a client's IP Address/Port information from the database. The UNREGISTER command format is as follows.

Field Name	Octet no.	
Command	1	0x14
User ID	2-5	Described earlier.
User Name	6-15	Described earlier.

20 Response to UNREGISTER

No Response.

ADDUSER

The ADDUSER command is a request to add a new user to the database and provides associated information on the new user. The ADDUSER command format is as follows.

Field Name	Octet no.	Brief description
Command	1	0x20
User Name	2-11	
First Name	12-21	
Last Name	22-41	
Address1	42-81	
Address2	82-121	
City	122-141	
State	142-143	
Zip Code	144-153	
Phone1	154-173	
Phone2	174-193	
Fax	194-213	
Email	214-277	
Password	278-287	Described earlier.
Publish Option	288	Publish Option 'O': Publish user information in the public directory 'N': Do not publish user information

Response to ADDUSER

The response to the ADDUSER command is the following format.

Field Name	Octet no.
Return Code	1

The Return Code is any of the following:

5. RETCODE_SUCCESS, RETCODE_INVALID_USERNAME, RETCODE_INVALID_PASSWORD, RETCODE_ALREADY_EXIST, or RETCODE_SERVER_ERROR.

DELETEUSER

- 10 The DELETEUSER command is a request to delete a user from the database. The DELETEUSER command format is as follows.

		Brief description
Command	1	0x21
Session ID	2-33	Described earlier

15

Response to DELETEUSER

The response to the DELETEUSER command is the following format.

	Octet no.
Return Code	1

The Return Code is any of RETCODE_SUCCESS, RETCODE_INVALID_SESSIONID, or RETCODE_SERVER_ERROR.

MODIFYUSER

- 5 The MODIFYUSER command is a request to modify user information in the database. The MODIFYUSER command format is as follows. Any field that is not to change is 0x00.

	Octet Number	Brief Description
Command	1	0x22
User Name	2-11	
First Name	12-21	
Last Name	22-41	
Address1	42-81	
Address2	82-121	
City	122-141	
State	142-143	
Zip Code	144-153	
Phone1	154-173	
Phone2	174-193	
Fax	194-213	
Email	214-277	
Password	278-287	Described earlier.
Publish Option	288	Publish Option 'O': Publish user information in the public directory; or 'N': Do not publish user information.

10 Response to MODIFYUSER

The response to the MODIFYUSER command is the following format.

	Octet
Return Code	1

The Return Code is any of the following:

- ```
5 RETCODE_SUCCESS, RETCODE_INVALID_SESSIONID, or
 RETCODE_SERVER_ERROR,
```

## GETUSERINFO

The GETUSERINFO command is a request to retrieve  
10 user information from the database. The GETUSERINFO  
command format is as follows.

|            | Octet no. | Brief Description  |
|------------|-----------|--------------------|
| Command    | 1         | 0x23               |
| Session ID | 2-33      | Described earlier. |

## 15 Response to GETUSERINFO

The response to the GETUSERINFO command is the following format.

|             | Octet no. | Brief Description |
|-------------|-----------|-------------------|
| Return Code | 1         |                   |
| User Name   | 2-11      |                   |
| First Name  | 12-21     |                   |
| Last Name   | 22-41     |                   |

|                |         |                    |
|----------------|---------|--------------------|
| Address1       | 42-81   |                    |
| Address2       | 82-121  |                    |
| City           | 122-141 |                    |
| State          | 142-143 |                    |
| Zip Code       | 144-153 |                    |
| Phone1         | 154-173 |                    |
| Phone2         | 174-193 |                    |
| Fax            | 194-213 |                    |
| Email          | 214-277 |                    |
| Password       | 278-287 | Described below.   |
| Publish Option | 288     | Described earlier. |

The Return Code is any of the following:  
 RETCODE\_SUCCESS, RETCODE\_INVALID\_SESSIONID, or  
 RETCODE\_SERVER\_ERROR.

5

#### ADDCONTACTINFO

The ADDCONTACTINFO command is a request to add a new contact to the user's contact list stored the database. The ADDCONTACTINFO command format is as follows.

10

|            | Octet no. | Brief Description         |
|------------|-----------|---------------------------|
| Command    | 1         | 0x30                      |
| Session ID | 2-33      | Described earlier.        |
| Folder ID  | 34-35     | 2 byte Folder Identifier. |
| User Name  | 36-45     |                           |
| First Name | 46-65     |                           |

|             |         |                  |
|-------------|---------|------------------|
| Last Name   | 66-85   |                  |
| Phone1 Type | 86      | Described below. |
| Phone1      | 87-106  |                  |
| Phone2 Type | 107     | Described below. |
| Phone2      | 108-127 |                  |
| Phone3 Type | 128     | Described below. |
| Phone3      | 129-148 |                  |
| Phone4 Type | 149     | Described below. |
| Phone4      | 150-169 |                  |
| Email       | 170-233 |                  |

Phone Types are any of the following:

HOME\_PHONE: Home Phone Number

WORK\_PHONE: Work Phone Number

5 CELL\_PHONE: Cell Phone Number

PAGER\_PHONE: Pager Number

OTHER\_PHONE: Other Number

#### Response to ADDCONTACTINFO

10 The response to the ADDCONTACTINFO command is the following format.

|             | Octet no. |
|-------------|-----------|
| Return Code | 1         |
| Contact ID  | 2-5       |

The Return Code is any of the following:

RETCODE\_SUCCESS, RETCODE\_INVALID\_SESSIONID,

15 RETCODE\_INVALID\_USERNAME, RETCODE\_ALREADY\_EXIST, or  
RETCODE\_SERVER\_ERROR.

The Contact ID is a 4 byte Contact Identifier.

#### DELETECONTACTINFO

The DELETECONTACTINFO command is a request to delete contact information from a user's contact list that is stored in the database. The DELETECONTACTINFO command format is as follows.

5

|            | Octet | Brief Description  |
|------------|-------|--------------------|
| Command    | 1     | 0x31               |
| Session ID | 2-33  | Described earlier. |
| Contact ID | 34-37 | Described earlier. |

#### Response to DELETECONTACTINFO

The response to the DELETECONTACTINFO command is the following format.

|             |   |   |   |   |   |   |   |              |
|-------------|---|---|---|---|---|---|---|--------------|
| 8           | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Octet<br>no. |
| Return Code |   |   |   |   |   |   |   | 1            |

10

The Return Code is any of the following:  
 RETCODE\_SUCCESS, RETCODE\_INVALID\_SESSIONID,  
 RETCODE\_INVALID\_CONTACTID, RETCODE\_NO\_OWNER, or  
 RETCODE\_SERVER\_ERROR.

15

#### MODIFYCONTACTINFO

The MODIFYCONTACTINFO command is a request to modify a contact in a user's contact list that is stored in the database. The MODIFYCONTACTINFO command  
 20 format is as follows.

|  | Octet | Brief Description |
|--|-------|-------------------|
|--|-------|-------------------|

|             |         |      |
|-------------|---------|------|
| Command     | 1       | 0x32 |
| Session ID  | 2-33    |      |
| Contact ID  | 34-37   |      |
| Folder ID   | 38-41   |      |
| User Name   | 42-51   |      |
| First Name  | 52-71   |      |
| Last Name   | 72-91   |      |
| Phone1 Type | 92      |      |
| Phone1      | 93-112  |      |
| Phone2 Type | 113     |      |
| Phone2      | 114-133 |      |
| Phone3 Type | 134     |      |
| Phone3      | 135-154 |      |
| Phone4 Type | 155     |      |
| Phone4      | 156-175 |      |
| Email       | 176-239 |      |

#### Response to MODIFYCONTACTINFO

The response to the MODIFYCONTACTINFO command is a one octet "Return Code".

- 5        The Return Code is any of the following:  
 RETCODE\_SUCCESS, RETCODE\_INVALID\_SESSIONID,  
 RETCODE\_INVALID\_CONTACTID, RETCODE\_INVALID\_USERNAME,  
 RETCODE\_ALREADY\_EXIST, RETCODE\_NO\_OWNER, or  
 RETCODE\_SERVER\_ERROR.

10

#### GETCONTACTLIST

The GETCONTACTLIST command is a request for a list of contacts from the user's contact list based on provided search criteria. The GETCONTACTLIST command  
 15    format is as follows.

|                       | Octet no. | Brief description                                                                                                                              |
|-----------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------|
| Command               | 1         | 0x33                                                                                                                                           |
| Session ID            | 2-33      |                                                                                                                                                |
| Folder ID             | 34-35     |                                                                                                                                                |
| Starting Index        | 36-37     | Part of the contacts that satisfies the search criteria. The retrieved contacts that has the index less than the starting index are discarded. |
| # of Entries          | 38-39     | Number of entries to retrieve.                                                                                                                 |
| First Index Character | 40        | Retrieve the contacts that has a user name that starts with an alphabet between the first and last character.                                  |
| Last Index Character  | 41        | Retrieve the contacts that has a user name that starts with an alphabet between the first and last character.                                  |
| Search String         | 87-150    | A search string that is matched against the contact list.                                                                                      |

|             |     |                                                                                                                                                  |
|-------------|-----|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Sort Option | 151 | The Sort Option is any of SORT_LASTNAME_ASC, SORT_LASTNAME_DES, SORT_FIRSTNAME_ASC, SORT_FIRSTNAME_DES, SORT_USERNAME_ASC, or SORT USERNAME DES. |
|-------------|-----|--------------------------------------------------------------------------------------------------------------------------------------------------|

### Response to GETCONTACTLIST

The response to GETCONTACTLIST is a header  
5 followed by multiple contact entries.

#### Header

|                                      | Octet no. |
|--------------------------------------|-----------|
| # of Total Entries                   | 1-2       |
| # of Entries returned for this query | 3-4       |

#### Each Contact Entry

|                 | Octet | Brief description      |
|-----------------|-------|------------------------|
| Contact ID      | 1-4   | 4 byte Contact         |
| Contact User ID | 5-8   | 4 byte Contact User ID |
| First Name      | 9-28  |                        |
| Last Name       | 29-48 |                        |
| Phone1 Type     | 49    |                        |
| Phone1          | 50-69 |                        |
| Phone2 Type     | 70    |                        |
| Phone2          | 71-90 |                        |
| Phone3 Type     | 91    |                        |

|             |         |                                 |
|-------------|---------|---------------------------------|
| Phone3      | 92-111  |                                 |
| Phone4 Type | 112     |                                 |
| Phone4      | 113-132 |                                 |
| Email       | 133-196 | Null-terminated<br>ASCII string |

The Phone Types are any of the following:

- HOME\_PHONE (Home Phone Number),
- WORK\_PHONE (Work Phone Number),
- 5 CELL\_PHONE (Cell Phone Number),
- PAGER\_PHONE (Pager Number),
- OTHER\_PHONE (Other Number).

#### GETCONTACTINFO

- 10 The GETCONTACTINFO command is a request for detailed information of a specific contact entry. The GETCONTACTINFO command format is as follows.

|            | Octet | Brief<br>description |
|------------|-------|----------------------|
| Command    | 1     | 0x34                 |
| Session ID | 2-33  |                      |
| User ID    | 34-37 |                      |
| Contact ID | 38-41 |                      |

15

#### Response to GETCONTACTINFO

The response to the GETCONTACTINFO command is the following format.

|                   | Octet   | Brief<br>description |
|-------------------|---------|----------------------|
| Return Code       | 1       |                      |
| Contact User ID   | 2-5     |                      |
| Contact User Name | 6-15    |                      |
| First Name        | 16-35   |                      |
| Last Name         | 36-45   |                      |
| Phone1 Type       | 46      |                      |
| Phone1            | 47-76   |                      |
| Phone2 Type       | 77      |                      |
| Phone2            | 78-97   |                      |
| Phone3 Type       | 98      |                      |
| Phone3            | 99-118  |                      |
| Phone4 Type       | 119     |                      |
| Phone4            | 120-139 |                      |
| Email             | 140-203 |                      |

The Return Code is any of the following:

RETCODE\_SUCCESS, RETCODE\_INVALID\_SESSIONID,  
 RETCODE\_INVALID\_USERID, RETCODE\_INVALID\_CONTACTID,  
 5 RETCODE\_NO\_OWNER, RETCODE\_SERVER\_ERROR

The Phone Types are any of the following:

HOME\_PHONE (Home Phone Number)  
 WORK\_PHONE (Work Phone Number)  
 10 CELL\_PHONE (Cell Phone Number)  
 PAGER\_PHONE (Pager Number)  
 OTHER\_PHONE (Other Number)

#### FINDUSERNAME

The FINDUSERNAME command is a request for a list of users that match the first, last name and email. The FINDUSERNAME command format is as follows.

|            | Octet  | Brief description |
|------------|--------|-------------------|
| Command    | 1      | 0x50              |
| First Name | 2-21   |                   |
| Last Name  | 22-41  |                   |
| Email      | 42-105 |                   |

5 Response to FINDUSERNAME

The response to the FINDUSERNAME command includes a header followed by multiple contact entries.

## Header

|              |   |   |   |   |   |   |   |              |
|--------------|---|---|---|---|---|---|---|--------------|
| 8            | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Octet<br>no. |
| # of Entries |   |   |   |   |   |   |   | 1-2          |

10 Each User Entry

|            |   |   |   |   |   |   |   |              |
|------------|---|---|---|---|---|---|---|--------------|
| 8          | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Octet<br>no. |
| User Name  |   |   |   |   |   |   |   | 1-10         |
| First Name |   |   |   |   |   |   |   | 11-30        |
| Last Name  |   |   |   |   |   |   |   | 31-50        |

Each of User Name, First Name, Last Name are Null-terminated ASCII strings.

### APPENDIX C

The entry node maintains the IP Address, Port numbers, and the status information of selected  
 5 intermediate servers. The entry node further routes a client's call request to an intermediate server based on the status information.

The entry node can be used for disabling or enabling intermediate servers. In the case of a  
 10 failure of an intermediate server, a backup intermediate server can easily replace the dysfunctional intermediate server.

The entry node maintains a TABLE of information for all the intermediate servers. The entry node  
 15 periodically broadcasts a REPORT\_REQUEST packet to the intermediate servers, and all the intermediate servers register/report their information in response to the packet. Because the REPORT\_REQUEST packet is broadcasted periodically, new intermediate servers can  
 20 be added without having to restart the entry node. In this embodiment, the REPORT\_REQUEST is transmitted using UDP (User Datagram Protocol) over IP.

The REPORT\_REQUEST packet includes an IP address and a port number. The IP Address is, e.g., a 4 byte  
 25 IPv4 Address of the entry node. The port number is, e.g., a 2 byte port number that the entry node listens for response.

A SERVER\_REPORT packet is transmitted by an intermediate server in response to the REPORT\_REQUEST.  
 30 It contains the IP Address/Port information and the load information of the intermediate server 130. In this embodiment, the SERVER\_REPORT is transmitted using UDP/TCP/IP.

The following table describes fields in the  
 35 SERVER REPORT packet.

| Field Name | Octet no. |
|------------|-----------|
|------------|-----------|

|                           |      |
|---------------------------|------|
| IP Address                | 1-4  |
| Port Number               | 5-6  |
| Number of Active Sessions | 7-8  |
| Max Number of Sessions    | 9-10 |

The IP Address is, e.g., a 4 byte IPv4 Address of the corresponding intermediate server. The Port Number is, e.g., 2 byte port number that the corresponding  
5 intermediate server listens for communications from the client. This port number is passed to the client upon request by the entry node. The Number of Active Sessions is the number of calls that the corresponding intermediate server is handling. This information is  
10 used by the entry node for distributing the client's call requests. The Max Number of Sessions is a maximum number of concurrent calls that the corresponding intermediate server is capable of handling.

## APPENDIX D: Messages

### Message transmission protocol

- 5 A description of a suitable transport topology for the messages is provided. In this embodiment, messages are transmitted using TCP/IP. The following conventional TCP header is used to transport all messages.

| Field Name                       | Octet no. |
|----------------------------------|-----------|
| Protocol Discriminator (hi-byte) | 1         |
| Protocol Discriminator (lo-byte) | 2         |
| Content Length (hi-byte)         | 3         |
| Content Length (lo-byte)         | 4         |

- 10 The Protocol Discriminator species byte orders. Byte order is the order used for forming a 'word'. Some platforms use the higher byte first, and some use the lower byte first. For example, to express the number '1', it is 00 01 on some systems, and 01 00 on others.
- 15 Byte orders are different in different CPU platforms. For example, Sun SPARC uses different byte order from Intel's. For network applications, which typically would pass data to another heterogeneous system, it is important to let the remote party which byte order
- 20 scheme is being used.

In this embodiment, the Protocol Discriminator is 0x5354. The Content Length represents a total length of the packet, i.e., the header and + message content together.

25

### Message header

The following is a header for messages.

| Field Name | Octet no. |
|------------|-----------|
|------------|-----------|

|                                |   |
|--------------------------------|---|
| Message Type (hi-byte)         | 1 |
| Message Type (lo-byte)         | 2 |
| Version Info - Major (hi-byte) | 3 |
| Version Info - Major (hi-byte) | 4 |
| Version Info - Minor (hi-byte) | 5 |
| Version Info - Minor (lo-byte) | 6 |

The Message Type field species the type of message. The following tables provides associated values and messages.

5

| Message Type | Value  |
|--------------|--------|
| SETUP        | 0x0001 |
| CALLPROCEED  | 0x0002 |
| CONNECT      | 0x0003 |
| ALERT        | 0x0004 |
| RELEASE      | 0x0005 |
| DTMF         | 0x0006 |
| STATUS       | 0x0007 |

The Version Info field specifies the Version Number of the message. Because the message protocol definitions may change in the future, by using the version info, the receiver can reject a message if it is a higher version than it supports.

10

#### Various Messages

The following message can be used, but is not shown in FIG. 5.

15

STATUS. The client uses the STATUS message to report current status information to the selected intermediate server. The following table provides parameters of the STATUS message.

| Field Name | Octet no. |
|------------|-----------|
|------------|-----------|

|                      |      |
|----------------------|------|
| Call Reference Value | 1-8  |
| Status               | 9-10 |

The Call Reference Value field specifies a unique identifier (number) of the call session. The Status field species the status of the client. For example a  
5 value of 0x00 species the connection to the client is  
alive



# CLAIMS

What is claimed is:

1. A method of providing communications between a  
5 client and destination audio receiver, the method  
comprising the acts of:  
accessing a web site;  
providing link information to a entry node;  
providing link information to an intermediate  
10 node;  
accessing the intermediate node;  
providing link information to a gateway node;  
and  
accessing the gateway node.  
15
2. The method of Claim 1, wherein the act of  
providing link information to an intermediate node  
comprises determining an intermediate node with the  
lowest activity level.  
20
3. The method of Claim 1, wherein the act of  
providing link information to the gateway node further  
comprises determining a least cost gateway.
- 25 4. The method of Claim 1, wherein the gateway node  
comprises an ITSP gateway.
5. The method of Claim 1, further comprising using a  
PSTN associated with the destination audio receiver.  
30
6. The method of Claim 5, wherein the destination  
audio receiver comprises a remote client.
7. The method of Claim 1, wherein the intermediate  
35 node establishes H.323 based communication with the  
gateway node.

8. The method of Claim 1, wherein the web site provides advertising and wherein the communications are free to select users.

5

9. An apparatus that provides communications between a client and destination audio receiver, the apparatus comprising:

10 at least one gateway server that is capable of communicating with the destination audio receiver;

at least one intermediate server that is capable of communicating with the at least one gateway server;

15 an entry node that is capable of communicating with the at least one intermediate server and the client, wherein

the client signals the entry node to select an intermediate server,

20 the selected intermediate server selects a gateway server,

the selected intermediate server initiates communications with the gateway server, and

25 the client and destination audio receiver communicate without use of the entry node.

10. The apparatus of Claim 9, wherein the destination  
30 audio receiver comprises a remote client.

11. The apparatus of Claim 9, wherein the apparatus provides free audio communications between the client and destination audio receiver by providing advertising  
35 to the client.



12. The apparatus of Claim 11, wherein the advertising comprises video advertisements.
13. The apparatus of Claim 11, wherein the advertising  
5 comprises audio advertisements.
14. The apparatus of Claim 9, wherein the destination audio receiver comprises a remote client.
- 10 15. A method of providing free audio communications to a user, the acts comprising:  
    providing an advertisement to the user;  
    providing a destination audio receiver  
    identifier; and  
15      providing a signal path for audio  
    communications using the destination audio  
    receiver identifier.
16. The method of Claim 15, wherein the providing an  
20 advertisement further includes:  
    using a website to provide a visual  
    advertisement.
17. The method of Claim 15, wherein the providing an  
25 advertisement further includes:  
    generating an audio advertisement.
18. The method of Claim 15, wherein the providing a  
signal path further includes:  
30      providing link information to a entry node;  
    providing link information to an intermediate  
    node;  
    accessing the intermediate node;  
    providing link information to a gateway node;  
35      and  
    accessing the gateway node.



19. The method of Claim 15 wherein the destination audio receiver comprises a remote client.
- 5 20. The method of Claim 19 wherein the destination audio receiver identifier comprises an IP address of the remote client.

