



An Architecture for Residential Internet Telephony Service

Christian Huitema, Jane Cameron, Petros Mouchtaris, and Darek Smyk
Telcordia Technologies

Abstract

A new architecture that can be used for offering Internet telephony service to residential customers is introduced. The architecture addresses scalability and availability requirements of mass-market deployment of carrier-grade services and supports interconnection with SS7 for Internet telephony calls to the public switched telephone network. The architecture is based on the concept of a gateway decomposition that separates the media transformation function of today's H.323 gateways from the gateway control function of the gateways and centralizes the intelligence in a call agent. The Media Gateway Control Protocol is introduced as the protocol between the call agent that assumes the gateway control function and the gateway that provides just the media transformation function. Interworking between the architecture and the public switched telephone network, the Session Initiation Protocol, and H.323 are also discussed.

Software for making telephone calls over the Internet has recently entered the marketplace. This software allows people to make long-distance calls for the price of a local call to their Internet service provider. Solutions for providing Internet telephony are currently based on either the IETF's Session Initiation Protocol [1] or the ITU's H.323 standard [2]. Both SIP and H.323 were originally developed for establishing multimedia conferences over the Internet, and both assumed intelligent client devices; Internet telephony requires a subset of the functionality provided in each case.

To set up calls between the public switched telephone network (PSTN) and the Internet, gateway nodes have been introduced in the network. H.323 gateways provide for media transformation. The PSTN carries voice as a 64-kbps stream using pulse-code-modulation encoding, while the Internet carries voice using the Real-Time Transport Protocol (RTP) [3] and various encoding schemes over a range of bit rates. The H.323 gateway converts between the PSTN and Internet telephony voice formats.

To establish an end-to-end call between the Internet and the PSTN, H.323 gateways terminate H.323 signaling on the Internet telephony side and usually ISDN signaling on the PSTN side. The gateway also sets up the voice path for each call within the gateway by controlling the gateway resources.

Several carriers in the telecommunications industry are considering wide-scale deployments of Internet telephony service. In the short term, carriers can take advantage of arbitrage opportunities. In the long term, they can offer a wide

array of services (Internet access, telephony, multimedia conferencing) over a common IP-based network. However, wide-scale carrier deployments pose strict requirements on Internet telephony solutions that existing approaches cannot fully address:

- *Scalability.* Carriers require support for a very large number of customers (eventually millions). Existing gateways support a fairly small number of lines (up to a few thousand). The reason for the limitation is that gateways support both media transformation and media control and signaling.
- *Seamless PSTN integration.* Carriers require that their subscribers be able to set up calls the same way they do today in PSTN. Existing Internet telephony solutions require a subscriber to dial the number to connect to the gateway and then dial the destination call number. This procedure is called two-stage dialing.
- *SS7 connectivity.* Carriers require that PSTN services offered today using the Signaling System 7 (SS7) packet-based network also be offered to customers of Internet telephony. Existing H.323 gateways do not support SS7 interconnection.
- *Availability.* Carriers require very limited service downtime—on the order of a few minutes per year. Existing solutions have limited mechanisms for fail-over and cannot meet this requirement.

In this article, we propose a new architecture that addresses the limitations of the existing solutions for mass-market deployments of Internet telephony.

Gateway Decomposition

An early version of this architecture was first proposed in 1998 [4]. It is based on the concept of a gateway decomposition that separates the media transformation function from the gateway control function. In existing architectures these functions are co-resident on gateways, whereas the proposed architecture limits gateway functions mostly to media transformation and places responsibility for call establishment on a call agent located outside the gateway.

This allows a more scalable solution. The gateway decomposition also increases availability because multiple call agents can be used to control a gateway. If one call agent fails, another takes over without losing any calls. The call agent also terminates SS7 connectivity, enabling seamless integration with the PSTN; no two-stage dialing is required. SS7 connectivity also allows our architecture to offer all the services that are provided today by the PSTN.

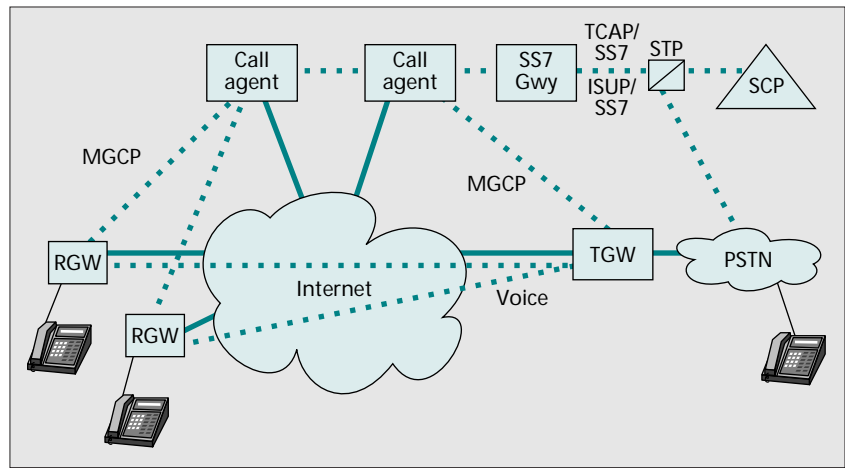
The proposed architecture assumes that most of the intelligence is inside the network, so customer premises equipment (CPE) requires only limited functionality. Internet telephony can be offered without requiring a PSTN subscriber to replace their existing CPE or buy new expensive equipment. A regular analog telephone can be used.

Keeping the intelligence centralized allows for the rapid introduction of new services. New services often do not require any CPE upgrades but can be handled by simply upgrading the call agent software and making the service available to all customers that are willing to pay for it. Customers need not install any new software. This avoids the problem of CPE that supports incompatible applications and services.

The concept of centralizing intelligence deviates from today's Internet model, which assumes that most of the intelligence is located at the CPE and that the network is mostly dumb. The reason for that deviation is that our architecture is trying to address the requirements of a very reliable and highly available Internet telephony service. We believe that the requirements for such a service point to a more centralized architecture.

We expect very intelligent CPE to be connected to the network, and they may implement services that the call agent does not support. It may not be desirable to implement those services at the call agent either because it is not cost-effective to do so or because customers do not require the reliability, availability, or integration with the PSTN that the call agent offers. We expect that both the centralized architecture and a highly distributed architecture with most of the intelligence at the CPE will coexist for some time. Our architecture does not preclude customers from implementing services at their intelligent CPEs. The marketplace will determine whether both approaches will be viable in the long term and which services should be implemented in a centralized fashion.

The next section describes the architecture and its main components. It is followed by an overview of the Media Gateway Control Protocol. MGCP is a current candidate for standardizing the interface between the gateways and call agent. The last section discusses the interoperations between the proposed architecture and the PSTN, SIP, and H.323.



■ Figure 1. Network architecture. The Media Gateway Control Protocol is used by the call agent to control the residential and trunking gateways (RGWs and TGWs).

Proposed Architecture

The proposed architecture is summarized in Fig. 1. Its main components are the residential gateway (RGW), the trunking gateway (TGW), and the call agent. The call agent uses the MGCP for controlling the residential and trunking gateways. The SS7 gateway allows the call agent to interact with the SS7 signaling network of the PSTN.

Residential Gateway

The residential gateway is responsible for capturing events associated with an IP telephony subscriber (going on hook, for example) and working with the IP network to signal these events to the call agent. The RGW is also responsible for supporting RTP [3], the protocol used for communicating the voice signal end-to-end.

An RGW may have additional capabilities, depending on a customer's needs. For example, an RGW may support voice and video interfaces, home networking, applications such as remote access for energy control, and other capabilities. An RGW also supports an interface to the IP network through some type of access network. The access network may be hybrid fiber coax (HFC) or asymmetric digital subscriber line (ADSL). Asynchronous transfer mode (ATM) may also be used in the network below the IP layer.

Our architecture assumes that an RGW supports at least one PSTN line, MGCP for setting up calls, a network interface, and RTP for voice communication. The phone in Fig. 1 can be connected to the RGW via an RJ-11 jack, so customers can reuse their existing telephones. PSTN subscribers can switch to Internet telephony service just by adding the RGW between their existing telephone equipment and their existing telephone line, and get the same services (such as call forwarding, call waiting, 800 service) without observing any differences in the services they subscribe to.

We assume that the cost of an RGW is low because of the limited functionality that it provides. The RGW takes instructions from the call agent on how to react to different events. The intelligence resides in the call agent.

Trunking Gateway

The trunking gateway is responsible for bridging between the Internet and the PSTN. Calls that originate in the PSTN go through the TGW so that the voice can be converted from time-division multiplexing (TDM) format to RTP packets. Similarly, for calls that originate on the IP side and terminate on the PSTN side, the TGW needs to convert the RTP packets to the appropriate TDM format. The TGW also supports

MGCP. MGCP allows the call agent to instruct the TGW on how to proceed with the calls.

In the PSTN, voice is carried between switches by trunks of different types. The TGW needs to support functionality provided on various trunks. For example, PSTN switches perform "continuity checks" periodically to verify that a TDM trunk still works by playing tones over the trunks and ensuring that they still propagate the tones correctly. The TGW needs to support the various tones required for these continuity checks. The TGW also needs to support multifrequency (MF) trunks, which are analog trunks that were deployed before TDM trunks and that provide signaling in-band. Signaling for MF trunk needs to be detected at the TGW, and either processed or passed to the call agent for processing.

Call Agent

The call agent controls both RGWs and TGWs via the MGCP [5]. Residential and trunking gateways report events to the call agent, such as a phone going off-hook, and the call agent uses MGCP messages to instruct the RGWs and TGWs on how to continue with call processing.

The call agent handles the SS7 signaling for the trunks that interconnect the PSTN with the IP network. SS7 is a packet-based network that allows systems in the telephone network to communicate with each other. Telephone switches use the SS7 ISDN user part (ISUP) protocol for communicating with each other when setting up calls that span more than one switch. The call agent sends ISUP messages over SS7 for setting up calls between the Internet and the PSTN.

SS7 is also being used in the PSTN for communication between switches and service control points (SCPs). SCPs maintain the logic for some services. Switches get directions from the SCPs before they can continue with processing calls that use those services. For example, an SCP translates the 1-800 number before a switch can route the call to the correct destination. Similarly, the call agent interacts with SCPs over the SS7 network for services such as 1-800.

The architectural framework defined here supports all the telephony services in the PSTN today and anticipates additional services that will be introduced in the Internet telephony environment. The definition of the architectural framework takes into account that subscribers may be participating in more than one call at the same time. For example, the subscriber may have one call on hold while talking to somebody else.

The call agent supports a distributed architecture. The architecture assumes that there may be a large number of call agents controlling gateways over the Internet.

The Interface Between Call Agents — In general, a call agent can support a large number of RGWs and TGWs. However, if a provider wishes to deploy a large network that includes a very large number of RGWs and TGWs, it may be necessary to deploy several call agents. If the two RGWs participating in a two-party call are controlled by different call agents, the agents will need to coordinate to set up the call. This is very similar to two switches in the PSTN coordinating to set up a call between two parties that belong to different switches.

ISUP is used today in the telephony network between switches for setting up calls. The same interface with minor modifications can potentially be used for setting up Internet telephony calls across call agents. ISUP will need to be extended so that information about the IP addresses, ports, and codecs (described by the Session Description Protocol, or SDP) for the originating and terminating party can be exchanged between the two call agents. As will be described in detail in a later section, each gateway participating in an Internet telephony call

must have the SDP information of the other; the call agents are responsible for ensuring that this is so.

The limitation of using an ISUP extension is that it may be possible to define a simpler protocol for calls that do not enter the PSTN. ISUP supports messages for supervising the status of circuits and is in general a complex protocol. It will also be fairly difficult to extend ISUP to support more complex services, such as video conferencing and multimedia, as the Internet evolves.

SIP is another protocol being considered for communication between call agents. SIP can encapsulate the ISUP parameters and carry that information between call agents together with the SDP parameters. SIP has the advantage of being developed for handling multimedia calls over the Internet.

There are currently no efforts to standardize the call-agent-to-call-agent interface. It is actually not critical for initial deployments to have such a standard. This is because IP telephony networks belonging to different service providers will be isolated from each other, and will interoperate through the PSTN. Although that approach has limitations because of the delays added by going through at least two TGWs, it may be a viable short-term solution. This is especially true since most calls, at least initially, will be between PSTN subscribers and Internet telephony subscribers. As Internet telephony networks become more widely deployed, the need for them to interoperate will increase, at which point the need to standardize the call-agent-to-call-agent interface will become critical.

Services — There have been few discussions up to this point on how to implement enhanced services around Internet telephony. The advanced intelligent networks (AIN) paradigm can be adopted from the PSTN world. The call agent can potentially support AIN interfaces to SCPs similarly to the way it supports interfaces to ISUP. This allows for support of AIN services like 1-800 (free phone). Since the call agent has a signaling gateway to the SS7 network, it can send queries to an SCP and perform the number translations required for services like local number portability, which allows customers to change their telephone service providers and keep the same telephone number. The services that have been implemented using AIN in the PSTN can now be offered to Internet telephony subscribers.

It is clear that the call agent will initially implement some services and utilize existing telephony AIN interfaces for some enhanced services. Current telephony interfaces like ISUP and AIN cannot be easily adapted to the wide variety of services that an IP network can support (multiparty multimedia conferencing, for example). The long-term viability of Internet telephony products will depend on the set of services they can support, as well as their flexibility for quickly introducing new services.

Media Gateway Control Protocol

The Media Gateway Control Protocol is a new protocol that can be used by call agents for controlling gateways. MGCP resulted from the merger of the Simple Gateway Control Protocol [4] and Internet Protocol Device Control [6]. The IETF is currently discussing the issue of standardizing the interface between the call agent and the gateway [7]. MGCP is one of the candidates.

This section provides a high-level introduction to the protocol. The complete specification is available as an Internet draft [5]. MGCP assumes a call-control architecture where the call-control "intelligence" is outside the gateways and handled by external call-control elements, the call agent. MGCP assumes that the gateways have limited storage and functionality. It introduces the concepts of connections and endpoints for establishing end-to-end voice paths, and the concepts of events and signals for establishing and tearing down calls.

Endpoints and Connections — MGCP assumes a connection model based on endpoints and connections. *Endpoints* are sources or sinks of data. Examples include an interface on a gateway that terminates a trunk connected to a PSTN switch and an interface on a gateway that terminates an analog PSTN connection to a phone or PBX.

Connections may be either point-to-point or multipoint. A point-to-point connection is an association between two endpoints for transmitting data between them. Once this association is established for both endpoints, data transfer can occur. A multipoint connection is an association among multiple endpoints. Connections can be established over several types of bearer networks such as ATM and TCP/IP networks.

Events and Signals — These concepts are central to MGCP. A call agent may ask to be notified about certain *events* occurring in an endpoint, such as off-hook, on-hook, flash-hook, or dialed digits, and may request that a certain *signal* be applied to an endpoint such as dial-tone, ringing, busy tone, or continuity tone.

Events and signals are grouped in packages that are supported by a particular type of endpoint. For instance, one package may support a certain group of events and signals for analog access lines, and another package may support another group of events and signals for MF trunks.

Some packages also support sequences of digits or letters, called a *dialing plan*, representing a sequence that a user is allowed to dial. Digits include numbers between 0 and 9. Letters may include the asterisk (*), the pound sign (#), and others. The call agent can ask a gateway to detect a set of digits or letters dialed by a user either by individually describing those letters or by using the “range” notation. The following is an example dialing plan:

```
[xxxxxxx | 1xxxxxxxxx | 011x.T]
```

This plan allows a user to dial any seven consecutive digits for local calls, the digit 1 followed by ten digits for long-distance calls, and 011 followed by any number of digits for international calls.

Use of Session Description Protocol

The call agent uses MGCP to provide the gateways with the description of connection parameters such as IP addresses, UDP ports, and RTP profiles. These descriptions follow the conventions defined in the Session Description Protocol, which is now an IETF proposed standard [8]. The use of SDP facilitates interoperability with the SIP, as explained in a later section.

SDP allows for description of multimedia conferences. The current version of MGCP limits SDP use to the setting of audio circuits and data access circuits. The initial session descriptions describe exactly one media, of type “audio” for audio connections. For example, the SDP information in MGCP messages may resemble the following:

```
v=0
c=IN IP4 128.96.41.1
m=audio 3456 RTP/AVP 0 96
a=rtpmap:96 G726/4
```

Gateway Control Functions

MGCP includes eight commands, which we describe here in some detail.

NotificationRequest — This command is used by the call agent for requesting notification from a gateway upon the occurrence of specified events in an endpoint. For example, a notification may be requested for an off-hook detection event in a gateway. A list of potential events includes: off hook transition, on hook transition, flash-hook, MF incoming call detected, and continuity tone detected.

The call agent can also request that the gateway collect the dialed digits. The *NotificationRequest* allows the call agent to download to the gateway a dialing plan describing all allowable digit combinations.

A call agent also includes a unique identifier in the *NotificationRequest*. The gateway includes this identifier in its *Notify* message when the requested event actually occurs. This identifier is used for tying the *NotificationRequest* to the *Notify* message that will be sent by the gateway.

Notify — Notifications are sent by the gateway via the *Notify* command in response to a *NotificationRequest* sent by the call agent to the gateway. The gateway includes, in the *Notify* command, a list of the events it observed and the unique identifier described above.

CreateConnection — The call agent uses the *CreateConnection* command for attaching an endpoint to a specific IP address and UDP port. Another *CreateConnection* request for the remote endpoint is necessary for creating an end-to-end connection with two endpoints. The call agent or the gateway may select which IP address and UDP port will be used for the endpoint.

The *CreateConnection* request specifies a *CallId* that will be used for identifying the call or session to which this connection belongs. More than one connection may actually share the same *CallId*. The *CreateConnection* request also specifies the endpoint and the parameters to be used for the connection. These parameters may include, for example, the voice encoding and compression parameters. The call agent also specifies the mode of the connection: send, receive, send/receive, conference, inactive, data, loopback, continuity test, network loopback, or network continuity test.

The *CreateConnection* request from the call agent may include a description of the remote side of the connection on the IP network. This includes not only connection parameters, such as the voice encoding, but also an IP address and UDP port. The remote connection description may be unspecified in some *CreateConnection* requests. This occurs because the call agent needs to send two *CreateConnection* requests for creating an end-to-end connection. When the first *CreateConnection* request is sent, the call agent does not yet know the remote connection descriptor. This information may be provided later via a *ModifyConnection* request.

A *CreateConnection* request may also include the parameters normally included in a *NotificationRequest*. This improves the performance of the protocol because it allows for a *CreateConnection* and a *NotificationRequest* to be sent in one message.

When the gateway acknowledges the *CreateConnection* request, it also sends to the call agent a *ConnectionId* (which uniquely identifies the connection within an endpoint) and local connection information about the IP address and UDP port it selected. The call agent can potentially select the IP address and UDP port, but the gateway may be sharing those resources for other functions, and it is preferable that the gateway does the selection.

ModifyConnection — The call agent uses the *ModifyConnection* command for changing the parameters associated with a previously established connection. The parameters in this

command are the same as in a CreateConnection request. The ConnectionId is provided by the call agent to the gateway in a ModifyConnection request.

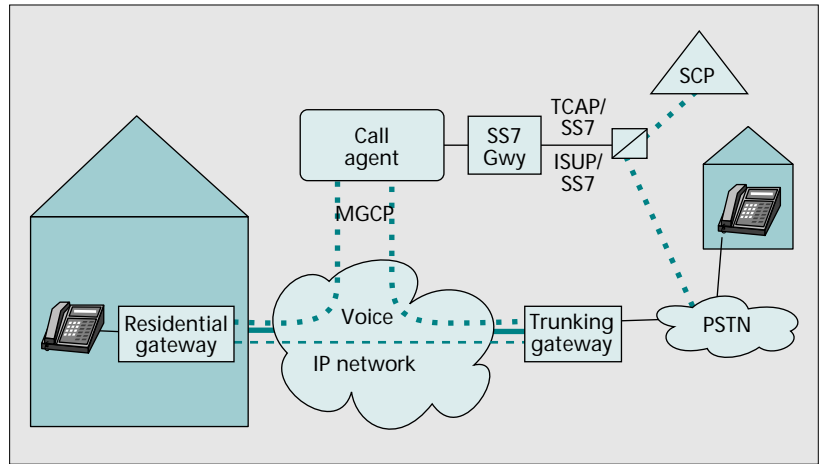
The ModifyConnection can be used for providing information about the other end of the connection (through the remote connection descriptor), for activating or deactivating a connection, and for changing the parameters of a connection.

DeleteConnection — The call agent can use this command to delete an existing connection. When the gateway acknowledges a DeleteConnection request, it includes a list of parameters about the status of the connection in the response. These parameters include numbers of packets and octets sent and received, number of packets lost, inter-arrival jitter, and average transmission delay.

The DeleteConnection command may also be sent by a gateway to the call agent for indicating that a connection can no longer be sustained because of a hardware failure in the gateway.

AuditEndpoint — The call agent can use this command for getting details about the status of an endpoint or a list of endpoints. Information that can be audited includes requested events, dialing plan, and connection identifiers. The response from the gateway includes the requested information.

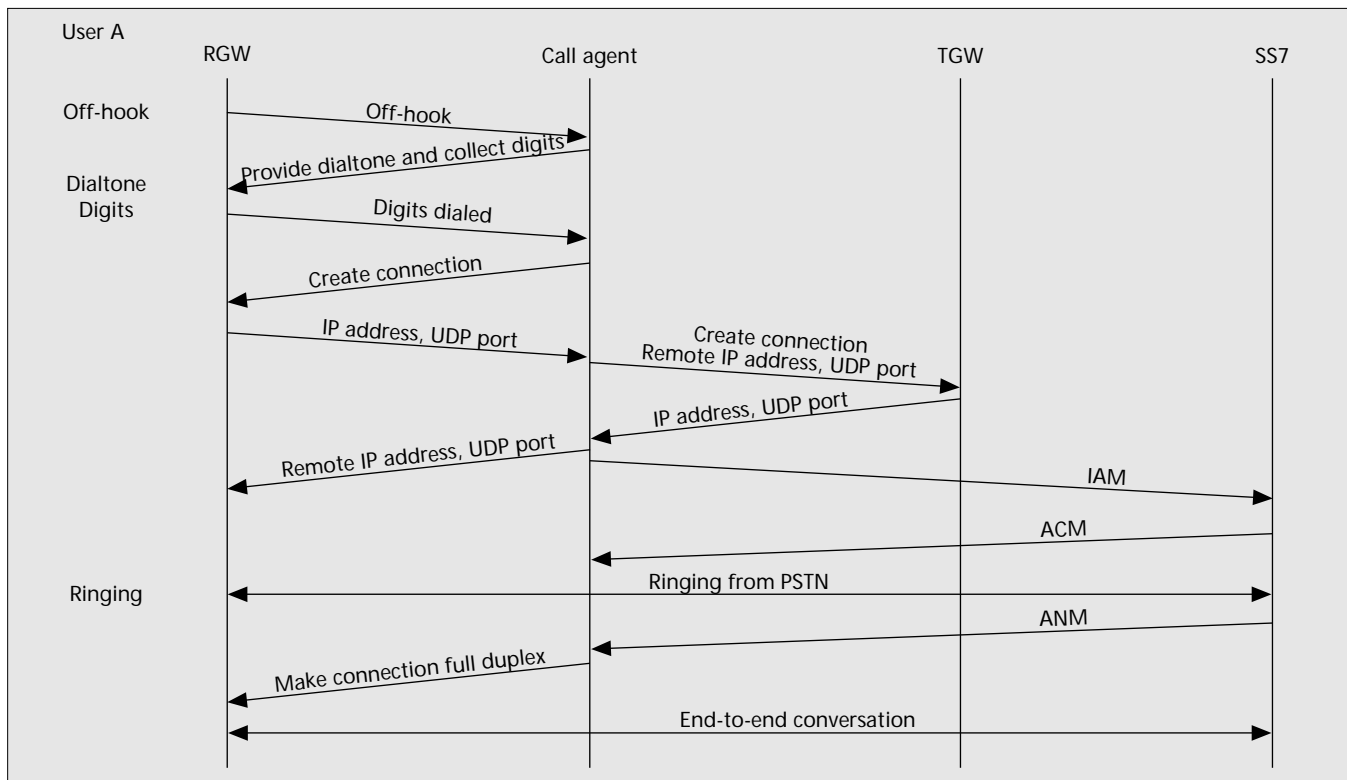
AuditConnection — The call agent can use AuditConnection for retrieving information related to a specific connection of an endpoint identified by a ConnectionId. The information



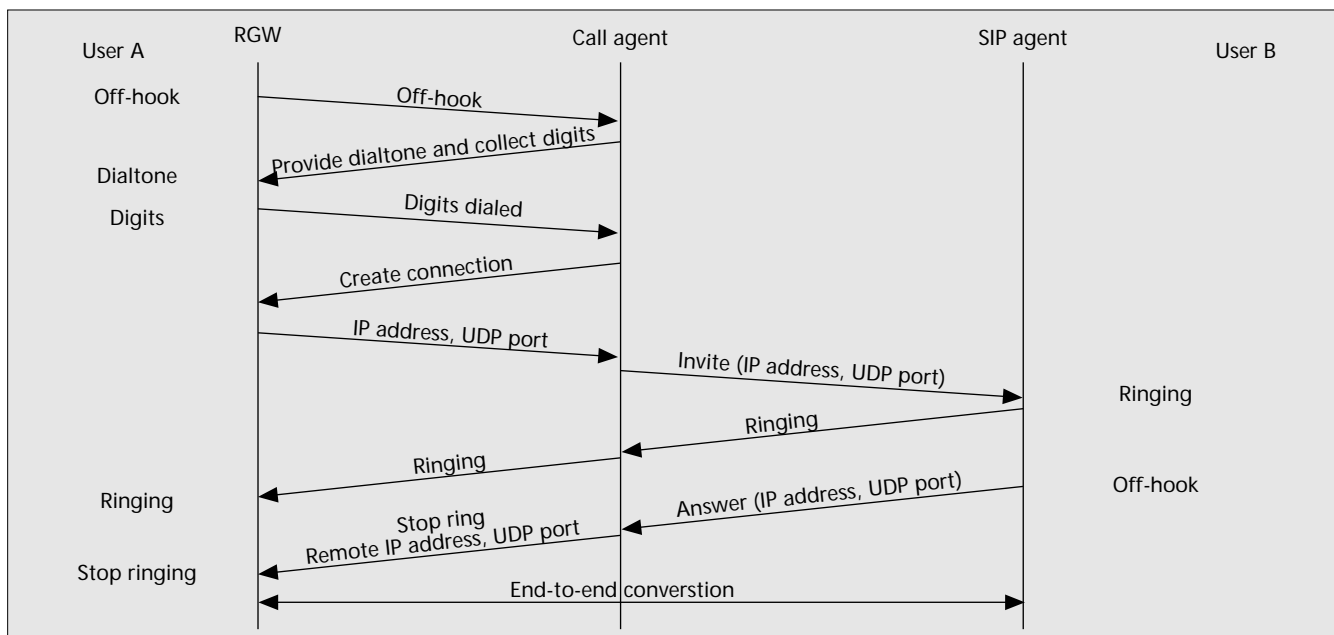
■ Figure 2. Integration of Media Gateway Control Protocol with the public switched telephone network (PSTN).

that can be retrieved includes call ID, local and remote connection descriptors, local connection parameters, and connection mode. The gateway response to the AuditConnection request includes the requested information.

RestartInProgress — The gateway uses this command to signal that an endpoint, or a group of endpoints, is taken in or out of service. The parameters of the RestartInProgress message indicate the group of endpoints that the message applies to. The RestartInProgress message also includes a parameter that specifies the type of restart. A graceful restart indicates that the endpoints will be taken out of service after a specified delay. A forced restart indicates that the endpoints are taken immediately out of service. A delayed restart indicates that the service will be restored after the specified delay.



■ Figure 3. Call flow illustrating MGCP interoperation with PSTN in a call originating in the IP network. It begins when User A goes off-hook and the residential gateway (RGW) sends a Notify message of the event to the call agent.



■ Figure 4. Call flow illustrating a call that originates from an MGCP-controlled gateway and terminated to a terminal that supports SIP. The call agent routes the call to the terminating side by sending a SIP invitation to the SIP agent of the called party.

MGCP Integration with Other Architectures

We conclude by showing how our MGCP-based network can interoperate with other networks, specifically PSTN, SIP, and H.323.

PSTN

Integration with the PSTN is critical for our architecture, and we have ensured that it happens in a manner that is transparent to service subscribers. In the first years of deployment, we expect that most calls originating in the IP network will terminate in the PSTN and, similarly, that calls terminating in the IP network will have originated from the PSTN.

Figure 2 shows how our architecture accomplishes the integration with the PSTN. Figure 3 illustrates a call flow that originates in the IP network and is destined for the PSTN. Message acknowledgments are omitted to make the figure more readable.

MGCP is used by the call agent for controlling the caller's RGW and the TGW that interconnects the IP network with the PSTN. The call agent exchanges ISUP messages over the SS7 network to terminate the call in the PSTN.

When the caller goes off-hook, the RGW sends a Notify message to the call agent that an off-hook event has occurred. The call agent immediately acknowledges that notification and examines the services associated with an off-hook action. The call agent provides a dialing plan to the gateway, and requests that the gateway play a dial-tone via a NotificationRequest message. The RGW immediately acknowledges that request.

The RGW starts accumulating digits according to the dialing plan received in the NotificationRequest. When the user has dialed a sufficient set of digits, the RGW sends the digits to the call agent via the Notify command. The call agent immediately acknowledges the notification. The call agent will then seize the incoming circuit by asking the RGW to bind the originating telephone line to an IP address and port with a CreateConnection message. The connection is created in receive-only mode. The RGW immediately acknowledges the creation, sending back the identification of the newly created connection and the session description used to receive audio data that includes the IP address and UDP port to be used for

the connection. The RGW also specifies the specific encoding scheme used for the audio.

The call agent, having seized the incoming line and completed a routing lookup to identify the outgoing gateway, must now seize the outgoing trunk. It does so by asking the TGW to bind the outgoing trunk to an IP address and port with a CreateConnection. The CreateConnection command sent to the TGW differs from the CreateConnection command sent to the RGW because the endpoint identifier points toward the trunk that will be used for routing the call. The CreateConnection message carries the session description returned by the RGW, and the connection to be created is bidirectional.

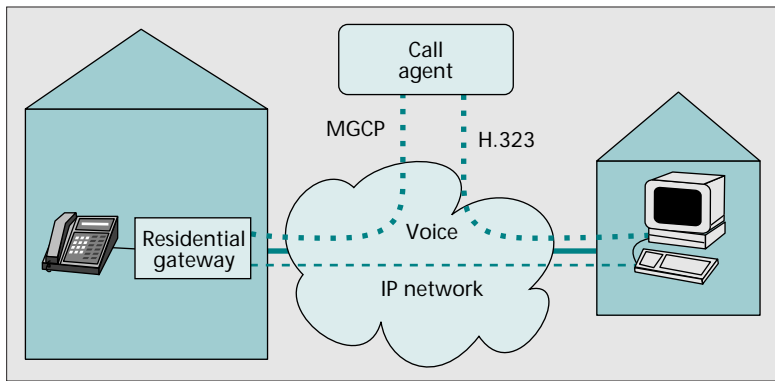
The TGW acknowledges the connection request, sending in the session description its own parameters such as address, ports, and RTP profile. The call agent then relays the IP information to the RGW, using a ModifyConnection request. The gateway immediately acknowledges the modification. At this stage, the call agent has established a half-duplex transmission path.

The call agent then sends an ISUP initial address message (IAM) to the SS7 network. After some time, it receives an address complete message (ACM) indicating that the called party is being alerted. In this flow, we assume that ringing for this call is generated in the PSTN for both the originating and terminating party. This is the most common approach in today's PSTN.

When the called party goes off-hook answering the phone, the call agent will receive an ISUP answer message (ANM) indicating that the call was answered. At that point, the call agent will send a ModifyConnection message to the RGW, to place the connection in full-duplex mode. The gateway will acknowledge the request. At this point, a bidirectional connection is established and the two parties can start talking.

Session Initiation Protocol

SIP is a protocol developed by IETF for establishing sessions [1], and it can be used for establishing Internet telephony calls. The main difference between SIP and MGCP is that MGCP includes functionality for controlling gateways and allocating resources within the gateway. We expect there to be uses for both protocols, and for that reason MGCP has been made compatible with SIP. MGCP and SIP are both text-based protocols and share the use of SDP. Service providers deploying Internet telephony services will need to select the



■ Figure 5. *Integration of the Media Gateway Control Protocol with H.323.*

protocol that best fits the requirements and architecture of the specific deployment.

Figure 4 demonstrates how a call agent can support calls between SIP-based terminals and MGCP-controlled gateways. Specifically, it shows how a call that originates from an MGCP-controlled gateway can be terminated to a terminal that supports SIP. Message acknowledgments are omitted from the figure to make it more readable.

In this flow, MGCP is used by the call agent to control the RGW of the caller. SIP is being used between the call agent and the SIP agent of the called party.

When the caller goes off-hook, the RGW notifies the call agent, plays dial-tone, and collects digits, as in Fig. 4. The call agent then creates the connection on the originating side in receive-only mode, as was shown in an earlier section, and must now route the call to the terminating side. It does so by sending a SIP invitation to the SIP agent of the called party. The invite message includes the IP address and UDP port of the caller. The SIP agent sends a response indicating that it is ringing the user. The call agent then sends a NotificationRequest to the caller's RGW requesting it to provide a ringing tone to the caller. The RGW acknowledges the message.

After the called party goes off-hook, the SIP agent sends a response message to the call agent indicating that the called party has accepted the call. The call agent acknowledges receipt of the response. The call agent then sends a ModifyConnection to the RGW of the caller providing the IP address and UDP port of the called party. The message also instructs the RGW to look for an on-hook transition and asks the RGW to place the connection in full-duplex mode. The RGW will acknowledge the request. At this point, the connection is established.

H.323

H.323 is a family of protocols that ITU has defined for establishing Internet telephony calls. H.323 was originally defined for video-conferencing applications over a LAN and has been extended for Internet telephony. H.323 is suitable for enterprise environments where the number of users is not excessive and most users have a PC running compatible applications. The H.323 model assumes that the CPE or gateway is intelligent. MGCP can work with H.323 for supporting intelligent CPE equipment like a PC that may be attached to the network. Figure 5 illustrates the integration. We are currently evaluating whether interoperation of MGCP with H.323 can meet the performance requirements of Internet telephony.

There are some concerns when interoperating with H.323 that need to be researched further. H.323 is a complex protocol that includes H.245 for control, H.225 for connection establishment, and H.450 for supplementary services. The issues posed by its wide-scale deployment have been described by Schulzrinne and Rosenberg [9]. H.323 is currently defined to be transmitted over TCP/IP. Timers have long been part of

standard implementations of TCP. The response and scalability requirements required by Internet telephony cannot be met in wide-scale deployments over large geographic areas. The use of TCP is a significant limitation, although ITU is working on adopting the use of UDP.

Acknowledgments

We would like to thank the authors of the MGCP specification, Mauricio Arango, Andrew Dugan, Isaac Elliott, and Scott Pickett, for their efforts in developing the MGCP by merging SGCP and IPDC. The architecture described in this article reflects the efforts of several individuals within Telcordia Technologies. The list of individuals that have contributed to its development is too long to be included here.

■

References

- [1] M. Handley *et al.*, "SIP: Session Initiation Protocol," IETF RFC 2543, Mar. 1999.
- [2] ITU-T Rec. H.323, "Packet-Based Multimedia Communications Systems," Feb. 1998.
- [3] H. Schulzrinne *et al.*, "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 1889, Jan. 1996.
- [4] M. Arango and C. Huitema, "Simple Gateway Control Protocol," Internet draft, Sept. 1998, work in progress, draft-huitema-sgcp-v1-02.txt.
- [5] M. Arango *et al.*, "Media Gateway Control Protocol (MGCP)," IETF Internet draft, Oct. 1998.
- [6] IP Device Control Protocol, set of six documents distributed as Internet drafts in 1998.
- [7] F. Cuervo *et al.*, "SS7-Internet Interworking — Architectural Framework," IETF Internet draft, July, 1998.
- [8] M. Handley and V. Jacobson, "SDP: Session Description Protocol," IETF RFC 2327, Apr. 1998.
- [9] H. Schulzrinne and J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony," *Proc. 1998 Wksp. Network and Op. Sys. Support for Digital Audio and Video (NOSSDAV'98)*, July 1998, Cambridge, England.

Biographies

CHRISTIAN HUITEMA (huitema@research.telcordia.com) is currently chief scientist in the Internet Architecture Research Laboratory at Telcordia Technologies (formerly Bellcore). His role is to animate Internet-related research. He is currently working on Internet quality of service and Internet telephony. He has written a fairly large number of scientific publications, articles, and conference communications, as well as three books. He studied at the Ecole Polytechnique in Paris from 1972 to 1975 and obtained in 1985 a Doctorat es Sciences from the Universite Pierre et Marie Curie (Paris 6). He was a member of the Internet Architecture Board (IAB) from 1991 to 1996, its chair between April 1993 and July 1995, and was elected a trustee of the Internet Society in May 1995.

JANE CAMERON (jcameron@telcordia.com) joined Telcordia Technologies in 1986. Since then she has worked on protocols specification and design, distributed systems and distributed databases, and call processing, in particular feature interactions. Most recently she has worked on Internet telephony and customer care systems. Jane is presently a chief scientist in Telcordia's Applied Research Laboratory. Prior to joining Bellcore she was in academia. She has a Ph.D. in mathematics.

PETROS MOUCHTARIS (pmouchta@telcordia.com) is director of Internet services architecture research at Telcordia Technologies. He focuses on architecture of solutions supporting Internet telephony and other services over packet-based networks. His interests also include signaling, multimedia services, and network control. Petros in the past has managed development groups at Pacific Bell and Oracle. He received his M.S. and Ph.D. in electrical engineering from the California Institute of Technology and his Diploma in electrical engineering from the National Technical University of Athens, Greece.

DAREK SMYK (dsmyk@telcordia.com) is the director of the next generation network evolution group at Telcordia Technologies. He is responsible for designing the network architecture supporting innovative network services and applications. He has been employed at Telcordia Technologies since 1985. Prior experience included four years at Standard and Poors, New York City, managing the development of real-time stock and commodity ticker processing systems. He holds nine patents in the areas of distributed computing, intelligent networks, the Internet, and convergence of video, voice and data networks. In 1998, in recognition of his accomplishments, he was named a Bellcore Fellow. He holds a Master's degree in computer science, and studied at Carnegie-Mellon University, Rutgers University, and Technical University of Wroclaw.