

# ◆ The Need for Third-Party Call Control

Tsun-Chieh Chiang, Vijay K. Gurbani, and John B. Reid

*This letter examines third-party call control (3PCC) and its application to new services. The evolution of networks and services that has driven the need for 3PCC capabilities is examined along with some of the new service concepts that may make use of 3PCC. Also discussed is ongoing work within Lucent Technologies in the area of 3PCC and service concepts that make use of 3PCC.*

© 2002 Lucent Technologies Inc.

## Introduction

The advent of the Internet and the proliferation of new applications have created the opportunity for a new set of *converged* voice and data services. This convergence has resulted in the possibility of using separate networks for the control and bearer interfaces associated with a communication session. The ability to initiate a communication session independent of the endpoints connected to the session's bearer channels is generally referred to as third-party call control (3PCC). (We use the term *session* as opposed to the term *call* since the latter is associated mostly with transporting voice. A session can transport any media—voice, video, data. In a sense, a call is a voice session.) A classic example of 3PCC is *click-to-dial* from a Web page, wherein a request to establish a voice session between two parties is initiated. For example, this could be used to allow a customer to request a voice call with a sales representative from an eCommerce Web site. Click-to-dial clearly demonstrates the key attributes of third-party call control we will use here:

1. An entity that is distinct from the endpoints involved in carrying the media over the bearer initiates the session (we will refer to this entity as the "third-party call controller").
2. The third-party call controller initiates communications via a signaling interface to each of the endpoints involved in the session (that is, the endpoints need not directly establish signaling interfaces between one another but rather have a signaling relationship with the third-party call controller).
3. The third-party call controller does not carry the media stream sent over the session's bearer channels.

The ability to provide session control independently of the endpoints involved in transmitting the media greatly facilitates new service concepts. In addition to click-to-dial services, this paradigm supports flexible conference control services, network-initiated streaming content services, and the ability to provide a general set of application programming interfaces (APIs) accessible to third-party application developers. Note that a generalized 3PCC capability is independent of the media types used in a session so that this model will be extensible for services making use of media types that have yet to be defined (for example, holographic video and streaming olfactory information) without requiring any significant changes to the session control model itself.

A logical view third-party call control is shown in **Figure 1**.

### The Genealogy of 3PCC

In this section, we trace the evolution of 3PCC from traditional phone networks to today's multimedia networks that are increasingly based on the Internet protocols. Each successive generation has made 3PCC more easily realized by separating the signaling layer from the media transport.

### Traditional Phone Networks

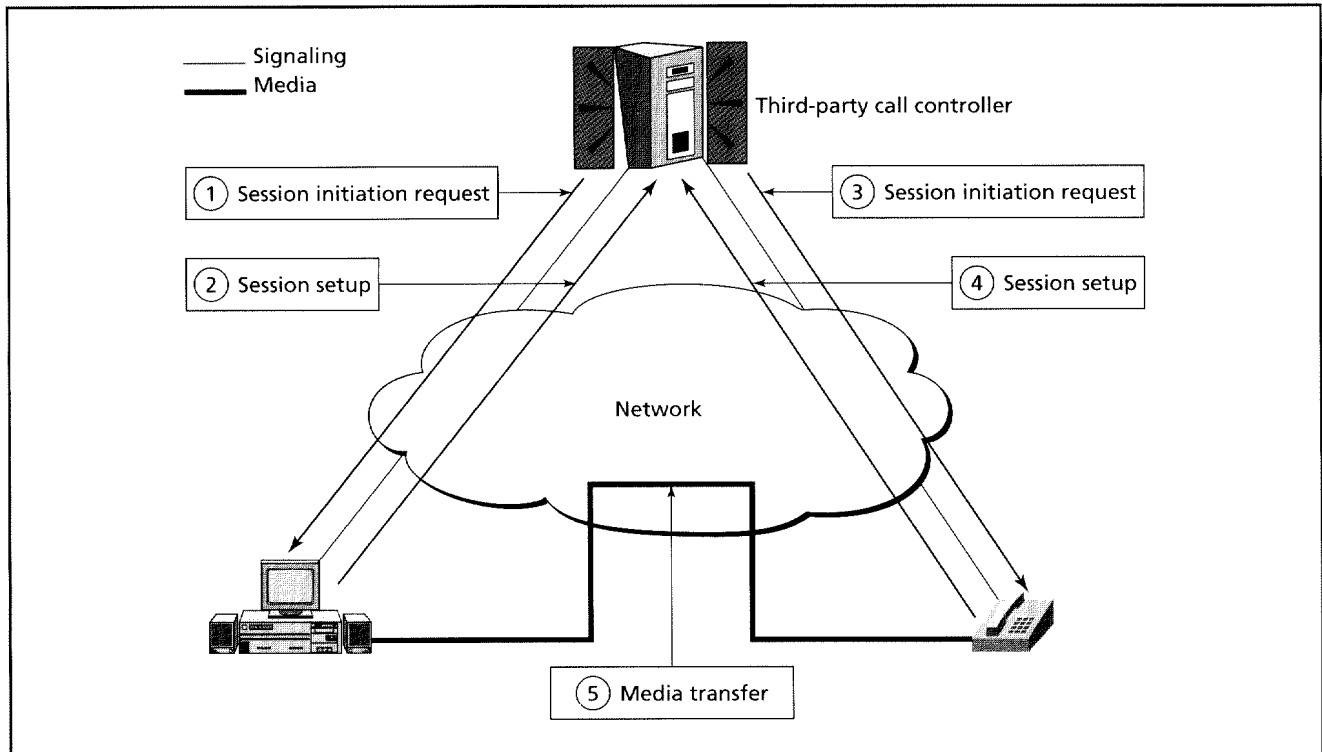
The evolution of the telephone network was built on the general premise that all calls were initiated by endpoints (telephones) that provided both the signaling (on/off hook, pulse, touch tones, ISDN Q.931 messages) and bearer interfaces involved in a call. Even with the advent of network-based resources (operator services, conference circuits, voice messaging), the basic operation of the network required that a call be initiated by an endpoint with combined media and signaling capabilities. As a result, 3PCC as defined above was not supported within the traditional PSTN—there is no standard way to initiate a call without the initiator also having to use one of its bearer channels. (Note that the ability to initiate calls from the network was included in the ETSI CS-3 IN standards, though it is not known if any implementations of 3PCC using this standard have been widely deployed.) The advent of service nodes and the integrated services digital network (ISDN) made a reasonable facsimile of 3PCC possible by using primary rate interface (PRI) connections and 2B channel transfer; however, this architecture still used additional bearer channels at the service node during setup of the call legs to each of the PSTN endpoints. Recent standards such as ETSI CS-3 have included a mechanism for initiating calls from a PSTN network. However, PSTN 3PCC solutions have not been widely deployed, and in the few known instances where 3PCC has been implemented, proprietary implementations were used (for example, within a single vendor's class 5 switch).

While the assumption that calls were initiated by devices that intended to connect to a bearer channel made sense for "plain old telephone service" (POTS),

#### Panel 1. Abbreviations, Acronyms, and Terms

3PCC—third-party call control  
AHE—Application Hosting Environment  
API—application programming interface  
ETSI—European Telecommunications Standards Institute  
ETSI CS-3—ETSI Capability Set 1  
H.248/Megaco—ITU standard for media gateway control  
H.323—ITU standard for multimedia (real-time voice, data, and video communication) across IP-based networks  
IETF—Internet Engineering Task Force  
IN—intelligent network  
IP—Internet protocol  
ISDN—integrated services digital network  
ITU—International Telecommunication Union  
ITU-T—ITU, Telecommunication Standardization Sector  
MGCP—media gateway control protocol  
NCC—Network Contact Center  
OSA—open service architecture  
Parlay—an open API specified by the Parlay Group to enable secure, public access to core capabilities of telecommunication and data networks  
PDA—personal digital assistant  
PINT—PSTN to Internet Interworking  
POTS—"plain old telephone service"  
PRI—primary rate interface  
PSTN—public switched telephone network  
Q.931—ITU-T Telecommunication Recommendation Q (switch and signaling), series No. 93  
SDP—session description protocol  
SIP—session initiation protocol  
SS7—Signaling System 7  
UA—user agent  
URL—uniform resource locator

this assumption became overly limiting with the evolution of enhanced services and the convergence with data interfaces. For example, conference services generally available today require that all of the participants initiate a call into a conference bridge—and that they remember the time of the conference, the bridge number, and the access code. Had 3PCC been available in the public switched telephone network (PSTN), it



**Figure 1.**  
**Logical view of 3PCC for a two-party session.**

would have been possible to have the service that reserves the bridge also initiate a call to each of the participants. Users who answered would then be automatically connected to the conference bridge.

### Softswitch Networks

The ability to separate media and signaling was further enhanced by the advent of *softswitches* around 1998. Softswitches are call controllers running on general-purpose computers and controlling network resources like PSTN-IP gateways, media servers, and IP endpoints. Softswitches straddle the PSTN and Internet protocol (IP) networks and help bridge the gap by completing calls between these two network environments. In order to efficiently do this, softswitches are designed to separate signaling and media. Media gateways are controlled by the softswitch using a master-slave protocol like media gateway control protocol (MGCP) or Megaco/H.248. Call-related signaling was rendered protocol agnostic by having softswitches convert the signaling protocols

(SS7, H.323, SIP, Q.931) to an internal canonical format.

With this built-in separation of media and signaling, softswitches are ideal for supporting converged service architectures. Softswitches provide two very important properties enabling services to make use of third-party call control: (1) they provide call control APIs that are independent of the underlying network call-signaling protocol and (2) they are the embodiment of a third-party call controller since softswitches mediate the signaling and bearer paths between endpoints without connecting to the bearer channels.

About the same time that the softswitch industry was taking shape, Internet standards such as PSTN to Internet Interworking (PINT) [3] that provide the building blocks of the first examples of converged services were being developed. PINT defined a protocol for invoking certain PSTN services from an IP network. Click-to-dial is the quintessential PINT service, since it initiated a PSTN service from an IP device—namely, the browser that displays a URL that allows a

user to establish a PSTN call between him/herself and another party. Other PINT services included initiating fax transfers and requesting access to content over the telephone (for example, using a text-to-speech converter to playback e-mail messages). While not the only medium to realize PINT services, softswitches aid the realization of these services by providing general-purpose interfaces for 3PCC that free application developers from having to directly access the underlying network signaling and media control protocols.

### **Multimedia Networks**

Today's multimedia networks, increasingly based on Internet protocols for signaling and media, not only allow the separation of the signaling from media transport, but also allow the separation of *applications* from the signaling as well as media aspects of a session. Third-party application servers can create, delete, and modify the attributes of parties participating in a call.

The session initiation protocol (SIP) [5] is a signaling protocol defined by the Internet Engineering Task Force (IETF) that is intended to support the establishment of real-time multimedia sessions in IP networks. To this end, SIP is designed to separate the negotiation of the media used in a session from the signaling involved in establishing the session. A corresponding IETF protocol, the session description protocol (SDP) [4], is used to define the media types supported by a SIP session and SDP information is transferred/negotiated between SIP endpoints involved in media transfer during session initiation. This allows for SIP to be very flexible in supporting a wide range of multimedia services—including the possibility of changing media types in mid-session. It also offers the promise of providing very efficient third-party call control: in an all-IP network an application server (a SIP 3PCC is one manifestation of an application server) can establish a session without using bearer resources in the network, whereas in traditional PSTN networks a similar capability requires an entity such as a service node to establish both signaling and bearer connections to each endpoint while establishing the session. The architecture of the SIP protocol allows for signaling endpoints to participate in session control without having to be aware of

media transfer—a flexible design that naturally supports 3PCC.

### **Case Studies**

In this section, we present two case studies involving the use of 3PCC within Lucent products.

#### **Click-to-Dial Softswitch Service Implementation**

The 3PCC mechanism was leveraged in the creation of Network Contact Center (NCC [6]), a Lucent Softswitch-based service. NCC is the next-generation contact center, leveraging the Internet and the PSTN to make it easy for the customer to contact a business. The 3PCC mechanism was involved when the customer clicked on a URL on the business's Web page requesting to talk to a customer service representative. This event reached the NCC running on the Lucent Softswitch, which resulted in a customer service representative being alerted first based on the profile of the customer (for example, a Spanish-speaking representative was alerted if the customer's profile indicated such a preference). When the customer service representative picked up his/her phone, the Softswitch then proceeded to call the customer, thus connecting two disparate parties.

#### **Third-Party Call Control with SIP**

The Mobility IN architecture organization has developed a third-party call controller (based on the "second attempt" scenario [1]). Although the IETF draft on SIP-based 3PCC described three different call flows, we were only able to successfully implement one of them, given the limitations of the currently available SIP user agents (UAs) on the market (none of the hard or soft SIP phones in our lab supported an INVITE without any SDP at the time this work was done). Therefore, our SIP 3PCC implementation made use of a special "held SDP" in the initial INVITE message to the SIP UA from the third-party call controller. A "held SDP" means that the initial call is placed on hold at the onset. The "200 OK" response from the first SIP UA includes the SDP it intends to use for the subsequent media transfer (that is, a non-null SDP header). The third-party call controller forwards this SDP information to the second SIP UA in an INVITE message. When the third-party call controller receives the "200 OK" response from the second SIP

UA, it forwards this (potentially) new SDP to the first SIP UA in yet another INVITE (referred to as a “re-INVITE”). After this point, media flows directly between the SIP UAs; however, the third-party call controller is still in the signaling path between the UAs. Once either UA terminates the session by sending a BYE request, the third-party call controller will generate another BYE message to terminate the call leg to other party.

A more robust version of the SIP-based 3PCC described above is being included in the *PacketIN*<sup>®</sup> Application Hosting Environment (AHE) release available by the summer of 2002. (It is expected that the 3PCC included in the *PacketIN* AHE will support the preferred call flow (“Third Flow”) described in [1].) This will include a set of APIs delivered with the Enhanced Service Authoring Environment (eSAE) to allow application developers to access these 3PCC capabilities. If expectations are realized, in future releases of the *PacketIN* AHE additional capabilities will be added to support conference control, mid-session call control, and generalized back-to-back SIP UA capabilities. Furthermore, these capabilities will be available through Parlay/OSA APIs as part of the Intelligent Services Gateway, which is also being built on the *PacketIN* AHE. These standard APIs should enable and encourage Web-based applications to make use of 3PCC capabilities when expanding to include real-time sessions as part of their Web services. Lucent’s Mobility IN organization is currently involved in a customer trial of our third-party call controller in conjunction with the *PacketIN* AHE and its associated *iSIP* SIP proxy server. This trial will include a classic click-to-dial service off of an active Web page that will establish a session between two SIP UA endpoints. In addition, the trial will include integration with an instant messaging/presence application, *Rear View Mirror* (RVM), which is part of the TeamPortal™ service [2] concept development from Bell Labs’ Software Production Research Department. This integration allows for status information regarding current SIP sessions to be shown in the RVM client windows so that others in a work group can know who may be involved in a SIP session in real time. In addition, the RVM client win-

dow will support a click-to-dial capability that will initiate a SIP VoIP session between “buddies.” This will be implemented by having the RVM server access the third-party call controller’s APIs to establish the VoIP session.

Another aspect of the Software Production Research Department’s TeamPortal suite is an application called *handiMessenger*, which provides the ability to establish various communication sessions from a wireless personal digital assistant (PDA). One potentially interesting extension to the *handiMessenger* service would be to interface it to the SIP third-party call controller to allow SIP sessions to be initiated from a wireless PDA. In the future, as the SIP third-party call controller is enhanced it is expected that it will support the ability to do general conference control (for example, on-demand drops and adds of various call legs) from applications such as *handiMessenger*. This will yield the best of both worlds with respect to providing conferencing capabilities via an intuitively obvious data interface—users can be added without the host dropping off the conference connection and a user’s PDA-resident “phonebook” data can be accessed directly, without having to enter number strings of 10 or more digits. Note that not all wireless PDAs will have the ability to participate in media sessions; the end user may choose to use another device to process the session’s media while reserving the PDA as a convenient 3PCC appliance.

## Summary

The 3PCC concept is not new; the essentials for third-party call control—namely, separation of signaling and transport—have been present in the traditional telephone network for decades. We believe the evolving signaling protocols of today’s multimedia network greatly facilitate the open, standard use of 3PCC in delivering value-added, converged services. This facilitation should apply to a centralized network model using the softswitch as a controller as well as an endpoint-centric model espoused by SIP, using a UA as the controller. In either case the flexibility provided by third-party call control will undoubtedly enable the development of many as yet unforeseen novel services.

## References

- [1] G. Camarillo, J. Peterson, J. Rosenberg, and H. Schulzrinne, "Third-Party Call Control in SIP," IETF Draft, Mar. 2, 2001, <<http://search.ietf.org/internet-drafts/draft-rosenberg-sip-3pcc-02.txt>>.
- [2] R. Colbert, D. S. Compton, R. L. Hackbarth, J. D. Herbsleb, L. A. Hoadley, and G. J. Wills, "Advanced Services: Changing How We Communicate," Bell Labs Tech. J., 6:1 (2001), 211–228.
- [3] L. Conroy and S. Petrack, "The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services," IETF RFC 2848, June 2000, <<http://www.ietf.org/rfc/rfc2848.txt>>.
- [4] M. Handley and V. Jacobson, "SDP: Session Description Protocol," IETF RFC 2327, Apr. 1998, <<http://www.ietf.org/rfc/rfc2327.txt>>.
- [5] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol," IETF RFC 2543, Mar. 1999, <<http://www.ietf.org/rfc/rfc2543.txt>>.
- [6] Network Contact Center Project at Bell Labs Research, <<http://www-db-in.research.bell-labs.com/projects/ncc/>>.

*(Manuscript approved April 2002)*

*TSUN-CHIEH CHIANG is a member of technical staff in the Architecture and Next-Generation Evolution Department in Mobility Solutions at Lucent Technologies in Naperville, Illinois. He holds M.S. and Ph.D. degrees in computer science and electrical engineering from the Illinois Institute of Technology in Chicago. His work focuses on the research, prototyping, and development of telephony services for wireline, wireless, and packet networks. Currently, Dr. Chiang is working on the design and prototyping of services related to WAP, SIP, and IM. He has five patents pending in the U.S. Patent and Trademark Office.*



*VIJAY K. GURBANI is a distinguished member of technical staff in the Architecture and Next-Generation Evolution Department in Mobility Solutions at Lucent Technologies in Naperville, Illinois. He has been involved in Lucent's Internet Call Waiting offer and also in creating services on the Lucent Softswitch, mapping the IN Call Model to SIP, using SIP as an enabler for IN services, and implementing a SIP proxy server for*



*demonstrating converged services. He has a B.Sc. and an M.Sc. in computer science from Bradley University, Peoria, Illinois, and is currently completing his Ph.D. research in the same field at the Illinois Institute of Technology, Chicago. His thesis is entitled "Enabling Services Through Protocol Interworking."*

*JOHN B. REID is a technical manager in the IN Architecture and Next-Generation Evolution Department in Mobility Solutions in Naperville, Illinois. The team he manages is responsible for developing requirements for the MiLife™ SurePay offer under offer-level systems engineering for IN Mobility solutions. He has a B.S. degree in computer engineering from the University of Michigan, Ann Arbor, and an M.S. degree in computer science from Stanford University, Palo Alto, California. ♦*

