Feature Interaction in Internet Telephony

Jonathan Lennox, Henning Schulzrinne
Columbia University
lennox@cs.columbia.edu, hgs@cs.columbia.edu

Abstract. While Internet telephony aims to provide services at least equal to traditional telephony, the architecture of Internet telephony is sufficiently different to make it necessary to revisit the issue of feature interaction in this context. While many basic feature interaction problems remain the same, Internet telephony adds additional complications. Complications arise since functionality tends to be more distributed, users can program the behavior of end systems and signaling systems, the distinction between end systems and network equipment largely vanishes and the trust model implicit in the PSTN architecture no longer holds. On the other hand, Internet telephony makes end point addresses plentiful and its signaling makes it easy to specify in detail the desired network behavior. Many techniques for resolving interactions in the PSTN are no longer easily applied, but several new techniques, *explicitness*, *authentication*, and *verification testing*, become possible in the Internet environment.

1 Introduction

Internet telephony is defined as the provision of telephone-like services over the Internet. Some consider it the next stage of the development of the telephone network and the first incarnation of the long-held goal of an “integrated services” network. The growth of the Internet as a platform for data delivery, and its rapidly increasing bandwidth make it desirable to create a telephone service that can run entirely over Internet protocols. Internet telephony offers the possibilities of multimedia communications, integration with other Internet services, and simplified, integrated development and operation.

In telephone networks, feature interaction occurs when several features or services, operating simultaneously, interact in such a way as to interfere with the desired operation of some of the features. This problem of feature interaction still exists in Internet telephony, and it will become an increasingly pressing problem as more, and more sophisticated, services are created and deployed in this environment. The large amount of work that has been done to understand and resolve feature interactions in traditional telephone networks will help us to understand and control interactions in Internet telephony.

Internet telephony, however, is different in many ways from the PSTN. Some of these differences help resolve or prevent feature interaction problems, as the design of new protocols, and the characteristics of the underlying network, eliminate problems associated with legacy networks and systems. However, Internet telephony also introduces some new types of interactions; it also makes several techniques for preventing or resolving interactions more difficult or impossible.

This paper generally discusses Internet telephony in terms of the Internet Engineering Task Force’s (IETF’s) architecture for it [1], centered around the Session Initiation Protocol [2, 3]. Many of the discussions and observation’s also apply to H.323 [4], an alternative protocol developed by the International Telecommunications Union (ITU). However, the IETF architecture is generally better developed in areas such as inter-provider communications, areas in which Internet telephony's differences from the PSTN in feature interaction issues are more pronounced.

This paper is organized as follows. Section 2 presents an overview of the architecture and component devices of Internet telephony. Section 3 details many of the differences between the PSTN and Internet telephony, both those that simplify service creation and allow new services, and those that make the feature interaction problem more difficult to resolve. Section 4 then discusses the applicability of existing approaches to solving feature interaction problems to the new environment. Section 6 discusses some new approaches for resolving feature interactions in the Internet. The paper looks ahead to future work in section 7.

2 Internet telephony architectural model

The architecture of Internet telephony is similar to traditional telephone networks in many ways, of course, but it also has some significant differences. Most fundamentally, Internet telephony is different from traditional telephone networks in that it, naturally, runs over the Internet, or more generally over IP networks. The most significant consequence of having this underlying network is that it provides transparent connectivity between any two devices on the network. Whereas devices in traditional networks are restricted to communicating with those devices to which they are directly connected, and the telephony protocols themselves must handle all location and routing features, Internet telephony can rely on an underlying infrastructure which provides all these capabilities automatically.

Within the Internet telephony network, we find three types of devices: end systems, gateways, and signaling servers. *End systems* are the devices on which users place and receive calls. These devices initiate and respond to signaling, and transmit and receive media. They are "smart" in that they are aware of call state; they keep track of the calls in which they are involved and the status of each of those calls. They may provide a number of services based on this call state information; for instance, *Call Waiting* or *Multiple Line* services are generally handled entirely in end systems in the Internet.

Gateways are devices which allow calls to be placed to and from other telephone networks. To other Internet telephony devices, they are not conceptually different from end systems; like end systems, they initiate and respond to signaling, and transmit and receive media. Other devices need not be aware of the existence of another network "behind" the gateway.

Signaling servers handle the application-level control of the routing of signaling messages. They are typically used to perform user location services; a signaling server can maintain information about where a user can currently be found, and forward or redirect call setup requests to the appropriate current location. Signaling servers are the devices which, from the point of view of feature-creation, are most similar in functionality to signal control or transfer points in the circuit-switched network; they can programmatically direct, block, or alter call signaling messages based on their own internal logic.

Figure 2 lists the Internet telephony devices and analogous devices in the PSTN.

3 Differences from the PSTN

Because of the effects of the Internet environment, Internet telephony has a number of differences from the traditional telephone networks; many of these differences will effect what sorts of features are possible, how these features are created, and how their interactions are

Internet Telephony	**PSTN**
End system	Customer-Premises Equipment, Private Branch Exchange
Gateway	Signaling gateway
Signaling server	Signal Control Point (SCP), Signal Transfer Point (STP)
Router	Signal Switching Point (SSP)

Figure 1: Comparable Components of Internet telephony and the PSTN

managed. In general, the new flexibility the Internet gives telephony allows a wide range of new possibilities; however, this flexibility also introduces new challenges.

3.1 Advantages of Internet telephony

The advantages of Internet telephony can be broadly divided into three categories, which we will discuss in detail in the subsequent sections. First of all, in section 3.1.1 we summarize the advantages that arise due to the design of Internet telephony protocols. Since there was an opportunity to design protocols "from scratch," a number of the difficulties present in traditional networks have be avoided by altering the underlying protocol architecture. Secondly, section 3.1.2 lists the advantages that arise from the infrastructure of the Internet itself. The Internet has been developed over the past decades to support a wide variety of types of services; many of these can be leveraged to provide powerful new abilities for the telephony environment. Finally, in section 3.1.3 we have those attributes of the Internet that are not as much its technical as its conceptual developments; the social and commercial evolution of the Internet has been substantially different from that of the PSTN, and this difference carries over to the social and commercial environment of Internet telephony.

3.1.1 Protocol issues

Internet telephony signaling protocols are significantly more expressive than those of the PSTN. This is particularly true compared to the limited signaling of tones and hook signals available to two-wire analog telephones. Rich signaling in Internet telephony eliminates many previous limitations on feature development. For example, an end system no longer needs to indicate its desire to transfer a call through an elaborate sequence of switchhook and DTMF tones; it can explicitly indicate to its partner the party to which the call should be transfered.

Furthermore, Internet telephony signaling is extensible, and can be extended while maintaining compatibility. As new signaling properties or events are invented, they can be added to the existing protocol in ways which can interoperate cleanly with existing implementations, either by providing richer information about the signaling information or by allowing fine-grained control over what features are required to be understood in order to understand a signaling message successfully. Internet telephony devices can also query each other to determine what properties and parameters they support. As new signaling elements and capabilities are developed, the network will be able to evolve gracefully to support advanced features without needing to undergo painful universal upgrades of an entire system.

In the past, difficulties have arisen in particular when trying to add new kinds of signaling capability, such as voice-mail control. In analog systems, only DTMF can be used, while even in ISDN, all such signaling would have to carried in user-to-user elements within existing signaling messages. In an Internet context, adding another control protocol, for example, RTSP [5] for voice mail or a presence protocol, can be done independently of the telephony signaling protocol.

Internet telephony enables the creation of new services that integrate telephone services with existing Internet protocols and services. Since Internet telephony addresses are URLs, the Internet telephony protocols have been designed so that "forwarding" or "transferring" a call to an e-mail address or a web page is not conceptually different than forwarding or transferring it to another telephone. Similarly, a signaling request can carry an arbitrary payload in its body — any media type which can be carried in MIME, the payload description mechanism of the web and e-mail, can also be carried in an Internet telephony request.

In addition, the real-time communications streams of Internet telephony sessions, while they can encompass traditional multimedia such as audio and video, are not limited to such types of communications. Because Internet telephony's signaling protocols separate the type of event (the beginning of a session, for instance) from the description of the stream, it is possible to use these same protocols to invite someone, for instance, to a multi-player game, or indeed to simultaneously invite to a game and voice communication.

One major difference of the Internet's telephony protocols from those of PSTN or ISDN networks is that the protocols the user's device uses to talk to the network (user-network interface or UNI) and the protocols that network devices use (network-network interfaces or NNI) to talk to each other are identical. Indeed, Internet telephony does not make a strong distinction between user devices and network devices; a device sending a request typically is not aware (and does not need to be aware) of whether it is communicating with to a signaling server or an end system. Because of this unification, Internet telephony deployment can scale from a few individuals running their own end systems, to a giant organization providing elaborate services and user location features; and these two organizations can interoperate cleanly. What's more, this means that even a customer of a large provider can choose to bypass the provider if his current needs don't require its services; for simplicity, flexibility, reliability, or privacy reasons, users can choose to communicate with each other directly end-to-end rather than through intermediate servers, without any need to modify their end systems.

Internet telephony protocols allow for capability labeling of end systems. In traditional networks, one often encounters the problem of a voice caller accidentally reaching a fax machine or modem, or vice-versa. Internet telephony, by contrast, prevents this in two ways: first, since the media type specifications for voice and fax differ, a voice-only end system will immediately reject the call with an "unsupported media type" error. On a broader scale, an end system can identify itself by the type of communication it supports; when a caller is searching for a destination, it can specify the type of communication desired in the call, and thus network devices can automatically resolve and prevent incompatible calls.

The Internet model eliminates user-level address scarcity. Because SIP and H.323 can use e-mail-style identifiers for telephone addresses, practically infinite numbers of Internet telephony identifiers can be created by any organization which possesses a DNS domain. PSTN telephone numbers, in contrast, are a scarce resource; obtaining "throw-away" addresses is not generally possible, and when the numbers in a certain geographic are are exhausted, an expensive and intrusive re-numbering is usually required.

This lack of address scarcity has a number of important secondary consequences. Telephone numbers have become more than just identifiers of telephone end points, but have been overloaded to indicate a variety of network and end system properties. First, numbers can refer to a user, to a device, to a connection to a switch, or to a distribution point for a complex service such as a phone bank. They also in some circumstances indicate carrier selection, which party is paying, or (in some regions) whether a device is a fixed-line device, a mobile phone, or a pager. Because Internet telephony addresses are “cheap,” however, such overloading can be separated and eliminated. Thus, it is possible for each resident of a house to have his or her own address; for someone to maintain separate addresses for his general reachability, for each role that he has (home and work, for instance), and for each device that he owns; or for addresses to be assigned dynamically for temporary use, and discarded afterwards — all without imposing any more burden on the network or the numbering plan than a single telephone number would.

3.1.2 Network issues

The nature of the Internet itself engenders a number of advantages that Internet telephony has over traditional circuit-switched telephone networks.

By their nature, circuit-switched networks, if they are to enable communication among huge numbers of people, require some sort of parallel signaling mechanism which enables circuits to be established. Because communication channels cannot be constantly maintained between every pair of stations that might wish to communicate, this parallel mechanism must be “self-routed” — an originating node specifies the destination of its signaling request, and the network sees to it that the request arrives at its destination; a circuit is established while this process takes place. The Internet, however, is inherently self-routing. Both signaling and media are sent off into the network through the same mechanism; thus there is no need for two parallel infrastructures to be maintained.

Additionally, because of the end-to-end nature of the Internet, the paths by which signaling and media traverse the network can be widely disparate. While in the PSTN signaling and media can indeed travel by separate routes, the architecture of that network still requires the two types of data to traverse the same administrative domains. In the Internet, by contrast, the routes which signaling and media traverse can be entirely disparate — only the end points of the two paths need to be the same. Media packets are normally sent end-to-end — thus traveling over the “natural” route the Internet’s low-level routing protocols have established between the endpoints — whereas signaling can travel across many servers which can provide elaborate third-party services.

Because IP is entirely packet-based, media communication is not limited to a single fixed-rate communications channel as it is in circuit-switched network. Internet telephony can, as appropriate for the environment in which it is being used, use very-low-bitrate speech encodings, or high-bandwidth video. Multiple media sessions can also be used in a single call, and these media sessions will inherently multiplex the communications channel between the endpoints. Bandwidth usage can even vary dynamically within a call depending on network conditions, with end systems stepping down to a lower-bandwidth encoding as a network becomes more loaded, then restoring higher quality once resources are again available.

Furthermore, IP supports network-level multicast protocols, without requiring application-level devices such as bridges. This enables a number of features both at the signaling and media levels. At the signaling level, it is possible to support a number of features such as “reach any member of a group,” without needing a server to distribute the request explicitly. More interestingly, media can also be multicast; this allows multi-party conferences to be established, in a bandwidth-efficient way, without the need for a conference bridge; and the transition between “multi-party telephone calls” and “large-scale conferences” can be made seamlessly, with no distinction necessary between the two.

Finally, the Internet environment supports a number of means of strong encryption and authentication, such as the IPsec suite of protocols. These tools can secure communications and reliably guarantee that false information is not injected into end systems. Using the sophisticated algorithms and design techniques that have been developed in recent years in the fields of computer and network security, and by taking advantage of increases in processing power, communications can be made secure from eavesdroppers in manners never before possible. Security in the PSTN, by contrast, relies on the physical security of network cables and equipment; this is generally both more expensive to accomplish, and less reliable in the long run.

3.1.3 Conceptual issues

The conceptual framework of Internet services also gives rise a number of new characteristics of the Internet telephony environment. First of all, whereas the PSTN is gradually moving to an increasingly distributed environment where multiple providers must interwork and compete on an fine-grained level, the Internet is already at such a level, and shows no signs of moving away from it. Thus, services can be provided by third parties — organizations dedicated only to providing services, with no intention of providing actual voice or multimedia transport — as easily as they can be by the original provider, and indeed providers may well specialize into service provision or data transport, as these are rather separate tasks.

The broadly distributed environment also introduces some new possibilities in terms of trust models for Internet telephony. It is relatively easy in Internet telephony for a customer to proxy all his calls through a service which, for example, automatically blocks calls from known telemarketers. A traditional telephone company does not have much interest in providing such a service — and few customers would likely trust a telephone company to provide it reliably, as telemarketing calls provide the company with revenue. The introduction of the distributed network allows users to have trust relationships with organizations other than their service provider.

Additionally, the Internet environment enables programmability on a scale not seen in the telephone network. Following the precedent of web services, we see that the Internet's distributed nature will give rise to programmability on a scale unprecedented in PSTN networks. This has several causes. First of all, the rich communications media and sophisticated processing possible for even low-end users allow complex feature descriptions to be passed in real time. For example, the Call Processing Language [6, 7] that we are developing will allow users to design and upload scripts to network signaling servers. Real-time control of PSTN services, by contrast, is generally not terribly powerful; a user can typically at best set either a single parameter, or turn the feature on or off. Even when a user is specifying features off-line to his provider, he usually has only a checklist of possible features available; sophisticated controls which allow loops, branches, or user-settable timers are not possible.

The wide variety of providers available, and the fact that users will be able to use any provider of services regardless of who their data connections come from, will give service providers a strong motivation to create services which will distinguish them from their competitors. A single, standardized list of enumerated features which customers can choose among does not give service providers much to distinguish themselves from the pack, so we envision that providers will quickly develop more sophisticated, distinctive features instead.

3.2 New complications

The new features of the Internet introduce, however, a significant number of additional complications to the problem of creating and deploying features and resolving their interactions. Most of these problems are the “flip side” of new features described in the previous section;

while the new characteristics of the Internet enable new possibilities, they also increase the complexity of creating features.

The most significant of these new complications is the *distributed nature of the Internet* itself. Features can be implemented and deployed at numerous network devices, both end systems and signaling servers. What's more, these systems may well be controlled by entirely separate organizations, which may be unaware of each other or even competing, and thus will not generally be inclined to co-operate to resolve feature interactions.

Additionally, because user programmability is now possible, the new phenomenon of *features created by amateur feature designers* arises. Because new services can be created and deployed with much the same level of ease that, for example, web services can be created today — a simple service can be put together by a reasonably experienced programmer in a matter of hours — they may be created by programmers who may not consider feature interaction issues thoroughly, either through ignorance or expediency. Such distributed problems may be dismissed as the "just desserts" of customers of incompetent feature designers, but unfortunately other service providers will have to inter-operate with such services.

On a network level, the characteristics of the Internet also introduce some new complications. First of all, the fact that *media packets travel end-to-end*, without being interceptable by intermediate servers, means that intermediate servers can no longer implement a number of features transparently. For instance, ordinary signaling servers cannot listen in on calls to collect digits ("press '#' for new call"); perhaps more significantly, they cannot perform "pipe-bending" services, where an intermediate system moves one endpoint of a call from one end system to another — for example, to transfer a call — without explicitly informing the end systems of the new locations to which they should send their media packets.

Another related complication is the fact that *end systems have control of call state*. While this introduces many new possibilities for general feature creation and deployment, it also complicates issues in situations when the network wants to be able to impose control contrary to the expressed desires of an end system. For example, in traditional telephone networks, 911 (emergency) calls are usually handled specially, so that end systems cannot hang them up; the emergency operator must hang up the call before the line is cleared. If the end system controls its own states, however, it is impossible for the network to enforce this without the end system's cooperation.

Several new features of Internet telephony protocols also have the potential for dramatic feature interaction consequences with existing protocols. Probably the most dramatic of these is what is known as the *forking proxy*. A signaling server, or proxy server, can take an existing call request and transmit it in parallel to several other devices. We discuss some examples of complex interactions that can occur with this feature in section 5.1.

Another new feature is *request expiration*. A request, when it is placed, can specify how long it should be considered valid — a user might want a call to only ring for the equivalent of four rings, for example — but services on subsequent signaling servers may be programmed to do different things when the expiration time elapses.

The Internet's *lack of address scarcity* can also complicate some common features. In traditional telephone networks, where telephone numbers are difficult to obtain, a telephone number can be used, reasonably effectively, as a representative of a party's identity for such purposes as incoming or outgoing call screening. In the Internet, however, "throw-away" addresses become easy to use; someone wishing to evade a block on their address can switch to another one with minimal effort.

Related to this problem is the Internet's *trust model*. In the PSTN, telephone users generally assume that they can trust their telephone company to provide accurate information, that their telephone company will not reveal private information to third parties when inappropriate, and that the wire leading out of their house indeed connects to the telephone company and no one else. Telephone carriers, meanwhile, can assume that the signals they get from a subscriber line are indeed coming from that subscriber; and signals they get from other telephone companies are reliable and secure. All these assumptions break down when end-to-end connectivity is introduced and anybody can become an Internet Service Provider. Forging communications becomes relatively straightforward when packets may be sent from any location on the network to any other, and intercepting them, while somewhat more difficult, is still significantly more tractable than on a telephone network, due to Internet characteristics such shared-bandwidth communications channels and dynamic routing protocols. While protocols for strong authentication and encryption have been developed, deployment of a key infrastructure which would enable large-scale trust is still a long way off[1]

Additionally, features like "caller I-D blocking" become much more difficult when users cannot trust the network not to reveal calling information to recipients — and indeed cannot reliably distinguish whether they are communicating with a "network" or a "recipient."

4 Applicability of existing feature interaction work

Existing work on feature interactions is applicable to the Internet environment in some circumstances. If we consider the framework of Cameron et al. [8], single-component interactions (those where all the interacting features are implemented on the same network component) are largely the same in the Internet environment as they are in traditional telephone networks, and we expect the techniques developed to resolve these interactions to work in the new environment.

An example of single-component interaction that can be dealt with in the Internet as it is in the PSTN is Cameron et al.'s Example 1, the interaction between *Call Waiting* and *Answer Call*. These two features have conflicting definitions of what should occur when a call attempts to reach a busy line: to signal the user with a tone, or to connect the calling party to an answering service, respectively. If, in an Internet telephony environment, both these services are deployed in the same device, or in multiple devices controlled by the same organization, techniques for resolving their interaction would carry over naturally from the PSTN.

Multiple-component interactions, however, are much more complicated for Internet telephony. The problem arises as features are designed and deployed by providers who do not cooperate, and have no interest in doing so; therefore, feature interaction resolution techniques such as those in [reference something or other] which depend on being able to describe features globally, and resolve their interactions at the time they are designed, are no longer practically applicable. (This is, of course, a growing problem in the PSTN as well, as increasing numbers of providers enter the market.)

5 Examples of new interactions in Internet telephony

Several varieties of new feature interactions appear in Internet telephony which either do not appear or are not as severe in traditional telephone networks. We categorize these into two types of interactions: *cooperative* interactions are those where all the parties who implement features would consider the others' actions reasonable, and would prefer to avoid

[1] In addition, many of the existing user-level certification services simply assure that the presenter of the signed request can indeed be reached by the (email) address indicated, but do not associate a legal or civil identity with a key.

an interaction if it were possible. *Adversarial* interactions, by contrast, are those where the parties involved in the call have conflicting desires, and one is trying to subvert the other's features. Roughly, cooperative interactions correspond with those that [8] describes as single-user multiple-component (SUMC) interactions; adversarial interactions are more commonly multiple-user multiple-component (MUMC) or customer-system (CUSY) interactions.

5.1 Cooperative interactions

"Cooperative" feature interactions are multiple-component feature interactions where all the components share a common goal — typically, allowing the caller to communicate with his or her intended called party — but have different and uncoordinated ways of achieving that goal. These conflicting implementations can interact in ways that can prevent the most desirable means of communication from occurring, even though it would be possible given the state of the parties involved; and can result in surprising or unpredictable consequences of deployed services.

Example 1 *Request Forking* and *Call Forward to Voicemail*
Request Forking allows an Internet telephony proxy server P to attempt to locate a user by forwarding a request to multiple destinations, A and B. The call will be connected to the first destination to pick up, and the call attempt to the others will be canceled. The interaction arises when the user to be reached is currently located at A, and another, B, has had its calls forwarded to a voicemail system. The call to B will be picked up first, as it is an automated system, and thus P will connect the call from B and cancel the call from A. The caller will never be able to reach the actual human.

Example 2 *Multiple Expiration Timers*
A SIP request may specify a length of time for which the request is valid. Difficulties arise, however, if several servers are programmed to have special behavior if the timeout elapses before the call has been definitively accepted or rejected. For example, one proxy server P_1 may be programmed to forward a call to a voicemail server when the expiration has elapsed, whereas another server P_2 may respond with a web page describing giving alternate ways of contacting the destination. If P_1 is earlier in the call path of P_2, the former server considers the latter server's response to be a definitive response to the call; and if P_2's response arrives at P_1 before its own timer expires, P_1 will forward that response back to the original caller rather than triggering its own expiration behavior. The two timers have the same nominal expiration period (the length of time specified in the request); which one executes first depends on factors such as processing time and the precision of the two servers' clocks. Therefore, there is a race condition of which of the two expiration-related services will be executed.

Example 3 *Camp-on* and *Call Forward on Busy*
Camp-on allows a caller who reaches a busy destination to continue to re-try that destination periodically until the line becomes free. However, if the destination has *Call Forward on Busy*, the call is forwarded to some alternate destination in this case, and the caller never receives the busy indication; thus there is no way to trigger the camp-on service. This is an interaction which can also arise in the PSTN, but it is more serious in Internet telephony for several reasons. First of all, because Internet telephony places so much additional power and call state knowledge into end systems, Call Forward on Busy is likely to be triggered by intelligent services implemented an end system, which may not be aware that the other party is attempting to camp on. PSTN switches which try to camp on will generally also be the location where Call Forward on Busy is implemented, and thus can resolve the interaction

locally. Furthermore, camp-on services in the Internet will generally need to be globally usable; users will not accept camp-on services which work only within one provider's network, so state cannot be shared easily among servers in a private manner either.

5.2 Adversarial interactions

"Adversarial" feature interactions, by contrast, are those where several of the parties involved — the caller, the destination, and/or either endpoint's administrator — disagree about something having to do with the call, typically about whether it should be allowed to be completed. These can be more difficult to resolve reliably than cooperative interactions, because generally parties attempting to subvert others will find ways to lie to them, or bypass them. They are also more complicated because users will generally be quite upset if the network allows their expectations about security or privacy to be violated.

Example 4 *Outgoing Call Screening* and *Call Forwarding*
Outgoing Call Screening blocks calls at an originating party based on the address to which a call is placed. However, even if a Call Screening service blocks calls to an address X, another signaling server, downstream from the location where the blocked is imposed, may forward calls originally directed to a non-blocked address Y to the blocked address X. This interaction also appears in the PSTN, of course (and this description is largely taken from [8]), but the ability to easily change addresses and get easy call forwarding on the Internet makes this problem much more significant in the Internet environment.

Example 5 *Outgoing Call Screening* and *End-to-end connectivity*
Because the Internet provides end-to-end connectivity, enforcement of *Outgoing Call Screening* policy is difficult for another reason. A signaling server cannot force calls to be placed through it; because the Internet telephony UNI and NNI protocols are identical, and because any device can talk to any other, an end system can be programmed to communicate directly with the remote party, bypassing local administrative controls entirely.

Example 6 *Incoming Call Screening* and *Polymorphic Identity*
Incoming Call Screening allows a called party — either in a signaling server or an end system — to reject calls from certain callers automatically. Because Internet telephony addresses are cheap, however, and because the caller can switch the identity he presents in his call request, he can easily alter the address he presents as his own in order to evade the screening lists the destination has programmed her phone to reject.

Example 7 *Incoming Call Screening* and *Anonymity*
Even in the absence of a malicious caller, *Incoming Call Screening* can be complicated by a caller's legitimate desire for anonymity. Because the trust model of the Internet does not allow a user to be sure that a network provider will hide the information like caller ID, if a user wishes to be anonymous he must avoid sending all identifying information in the signaling information in the first place — and for assured anonymity will likely have to use an anonymizing server run by a trusted third party, which will hide all information, including the sender's IP address for transmission of media packets and signaling. In the PSTN, a destination switch can easily apply *Incoming Call Screening* and *Caller I-D Blocking* services simultaneously; and both the caller and destination can trust this switch to apply their service reliably. In Internet telephony, however, there are not generally such mutually-trusted third parties, so for anonymous calls the critical information is simply not to the network. There is no reliable way to screen anonymized calls other than simply rejecting all of them.

6 New Approaches for Managing Internet Interactions

Though Internet telephony brings about new feature interactions, it also presents new possibilities for managing or resolving these interactions. The flexibility of the signaling protocols, and the underlying infrastructure of the Internet, can be exploited to resolve or prevent interactions in a manner which maintains and extends the powerful new characteristics of the Internet telephony architecture.

6.1 Explicitness

Many of the interactions which we have categorized as "cooperative" can be prevented or made less likely by making explicit the actions being taken, and their desired effects. Because the Internet telephony protocols are extensible, it is possible to add parameters which tell downstream servers what actual actions are desired; such parameters are currently being standardized [9]. If a call is intended to only reach a human, for instance, it is possible to specify that the call should not be forwarded to a station which has registered with a "voicemail" attribute; intelligent services which would otherwise forward a call to voicemail should know to return a "not currently available" status code instead. Similarly, a call wishing to camp on to the actual user to be contacted could specify "do-not-forward" so as to get back a "busy" response rather than have the call be forwarded against their wishes. The difficulty with this solution is that it can complicate the creation of services significantly; service creators need not only to determine what it is they wish to do, but to determine whether those actions are compatible with the preferences the caller specified with the call. Also, this explicitness requires that the receiver know about the attributes the caller desires; a call may specify "want to reach only the family goldfish," but the recipient is unlikely to be able to do anything useful with this if "goldfish" is not a recognized category.

6.2 Universal authentication

Many of the problems introduced by polymorphic identities and identity forging can be resolved by insisting on strong authentication of requests. Whereas a generic address can easily be used once and thrown away, and indeed a user can claim to be someone else, the barrier toward obtaining certificates giving actual signed identity information is much higher, and presumably widely-trusted certification authorities can be relied upon to be sufficiently consistent in their identification of users that call screening services can use this information to block callers. Unfortunately, all of this infrastructure fails if users accept non-authenticated calls; and authentication is far from being sufficiently widespread enough for it to be practical to accept only authenticated ones. However, we hope that the growth of Internet telephony will help be a driving force for widespread authentication to finally become widely deployed on the Internet.

6.3 Network-level administrative restriction

Administrative restrictions in the Internet cannot generally be reliably applied at the application level. If users have end-to-end connectivity available, it is not generally possible to prevent them from taking advantage of this connectivity by imposing restrictions solely at the application layer. Therefore, network-layer administrative restrictions such as firewalls must be used to limit end-to-end connectivity in order to impose administrative controls; these restrictions also have the advantage that they automatically apply to *all* Internet services, not just a limited subset of them. Network-level and application-level restrictions can also be used in concert; for instance, an Internet telephony signaling server, if it decided to allow a call, could instruct a firewall to open up the appropriate ports to allow the media associated with the call to flow.

6.4 Verification testing

Finally, the most direct way of ensuring correct operation of features is to test them directly. It is for third parties to establish services which automatically, at your request, place calls to you with various parameters or conditions enabled, to allow you to confirm explicitly that your features work the way you desire. As such providers gain more experience into the sorts of conditions that are likely to cause problems with services, they can expand their suites of testing tools to cover more esoteric interaction conditions. Thus, it should be possible to verify features and resolve their interactions in the real environment in which they are deployed, rather than attempting to analyze and categorize all possible consequences of a feature beforehand.

7 Conclusion

Feature interactions in Internet telephony are a serious issue, which feature developers will need to consider as they develop services for this new environment. There is a temptation, as Internet telephony “re-invents” the telephone network, to discard the lessons learned from the experience of traditional networks; however, it is clear that feature creation in the Internet must learn from prior experience of creation of telephony services. If these lessons are learned, however, problems of feature interaction will be manageable and can be dealt with efficiently.

The architecture of the Internet makes some of the feature interaction management techniques developed for traditional circuit-switched networks impractical. The distributed nature of feature creation, in particular, means that it will not generally be possible to describe all features globally before designing and implementing them. The Internet also, however, makes new techniques for dealing with interactions possible. These new techniques, and new applications of existing techniques in the new environment, will be a fruitful area for future research.

Acknowledgments

We would like to thank Yow-Jian Lin and Jonathan Rosenberg for their discussions and comments. We are also grateful to the members of the Internet Real-Time research group of the Columbia University Computer Science Department for comments and feedback on a presentation of an early version of this paper.

References

[1] H. Schulzrinne and J. Rosenberg, “Internet telephony: Architecture and protocols – an IETF perspective,” *Computer Networks and ISDN Systems*, vol. 31, pp. 237–255, Feb. 1999.

[2] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, “SIP: session initiation protocol,” Request for Comments (Proposed Standard) 2543, Internet Engineering Task Force, Mar. 1999.

[3] H. Schulzrinne and J. Rosenberg, “The session initiation protocol: Providing advanced telephony services across the internet,” *Bell Labs Technical Journal*, vol. 3, pp. 144–160, October-December 1998.

[4] International Telecommunication Union, “Packet based multimedia communication systems,” Recommendation H.323, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.

[5] H. Schulzrinne, A. Rao, and R. Lanphier, “Real time streaming protocol (RTSP),” Request for Comments (Proposed Standard) 2326, Internet Engineering Task Force, Apr. 1998.

[6] J. Lennox and H. Schulzrinne, “CPL: a language for user control of internet telephony services,” Internet Draft, Internet Engineering Task Force, Mar. 1999. Work in progress.

[7] J. Rosenberg, J. Lennox, and H. Schulzrinne, “Programming internet telephony services,” *IEEE Network*, vol. 13, pp. 42–49, May/June 1999.

[8] E. J. Cameron, N. D. Griffeth, Y.-J. Lin, M. E. Nilson, W. K. Schure, and H. Velthuijsen, “A feature interaction benchmark for IN and beyond,” *Feature Interactions in Telecommunications Systems, IOS Press*, pp. 1–23, 1994.

[9] H. Schulzrinne and J. Rosenberg, “SIP caller preferences and callee capabilities,” Internet Draft, Internet Engineering Task Force, Mar. 1999. Work in progress.