Towards Junking the PBX: Deploying IP Telephony

Wenyu Jiang, Jonathan Lennox, Henning Schulzrinne and Kundan Singh
Department of Computer Science, Columbia University
1214 Amsterdam Ave, Mail Code 0401
New York, NY 10027, USA
{wenyu,lennox,hgs,kns10}@cs.columbia.edu

ABSTRACT

We describe the architecture and implementation of our Internet telephony test-bed intended to replace the departmental PBX (telephone switch). It interworks with the traditional telephone networks via a PSTN/IP gateway. It also serves as a corporate or campus infrastructure for existing and future services like web, email, video and streaming media. Initially intended for a few users, it will eventually replace the plain old telephones from our offices, due to the cost benefit and new services it offers. We also discuss common inter-operability problems between the PBX and the gateway.

Keywords

Internet telephony deployment; VoIP test-bed; PSTN/IP interoperability; SIP

1. INTRODUCTION

Internet telephony is defined as the transport of telephone calls over the Internet. Internet telephone calls can originate from traditional phone sets through gateways, PCs using software or embedded devices ("Ethernet phones"). Most of the interest in Internet telephony is motivated by cost savings and ease of developing and integrating new services. Internet telephony integrates a variety of services provided by the current Internet and the Public Switched Telephone Network (PSTN) infrastructure. Internet telephony employs a variety of protocols, including RTP (Real-time Transport Protocol [18]) for transport of multimedia data and SIP (Session Initiation Protocol [5, 20]) or H.323 [7] for signaling, i.e., establishing and controlling sessions. SIP is designed to integrate with other Internet services, such as email, web, voice mail, instant messaging, multi-party conferencing and multimedia collaboration.

We have implemented a SIP-based software suite for Internet telephony and installed it within the Computer Science department at Columbia University, integrating it with the existing PBX infrastructure. The environment provides inter-operability with the PSTN, programmable Internet telephony services, IP-based voice mail, integration with web and email for unified messaging, multi-party multimedia conferencing, and inter-operability with existing multimedia tools. The setup allows us to extend our PBX capacity and eventually replace it, while keeping our existing phone numbers. The test-bed provides an environment where we can add new services and features, for example, accessing emails from a regular telephone. We believe that our setup can be readily used by other organizations.

Section 2 gives an overview of SIP. Section 3 details the architecture of our test-bed describing various components. PSTN Inter-operability is described in Section 4, whereas other advanced services are listed in Section 5. We analyze some scalability issues in Section 6. Finally, we summarize and point to future work in Section 8.

2. OVERVIEW OF SIP

Before we look at our architecture it is helpful to know how SIP operates[1]. Readers familiar with SIP may skip this section.

For an Internet audio call, it is sufficient for a participant to know the audio algorithms supported by the other participant and the IP address and port number at which to send the audio packets to the other participant. The problem with this is that IP addresses are hard to remember and may change if the user changes his location or machine. SIP allows use of a more high level address of the form *user@domain* for user mobility. For instance, a user can call *bob@office.com* no matter what communication device, IP address or phone number he is using currently. The current locations of the users are maintained by the SIP (location or registration) servers.

When Alice, with address *sip:alice@home.com*, wants to call Bob, *sip:bob@office.com*, her SIP phone contacts the server at *office.com*. The server knows where Bob can be reached and can either return Bob's location to Alice's phone (in redirect mode) or can itself try to contact Bob at his current location (in proxy mode). In the former case, Alice's phone retries the new location, while in the latter case the server proxies the request transparent to the caller. It is possible to encounter multiple SIP servers (either in redirect or proxy mode) in a given call attempt. A *forking proxy* can fork the call request to more than one locations, so that the first phone that is picked up gets the call, while all other phones stop ringing.

NOSSDAV'01, June 25-27, 2001, Port Jefferson, New York, USA.
Copyright 2001 ACM 1-58113-370-7/01/0006 ...$5.00.

[1] More details at http://www.cs.columbia.edu/sip

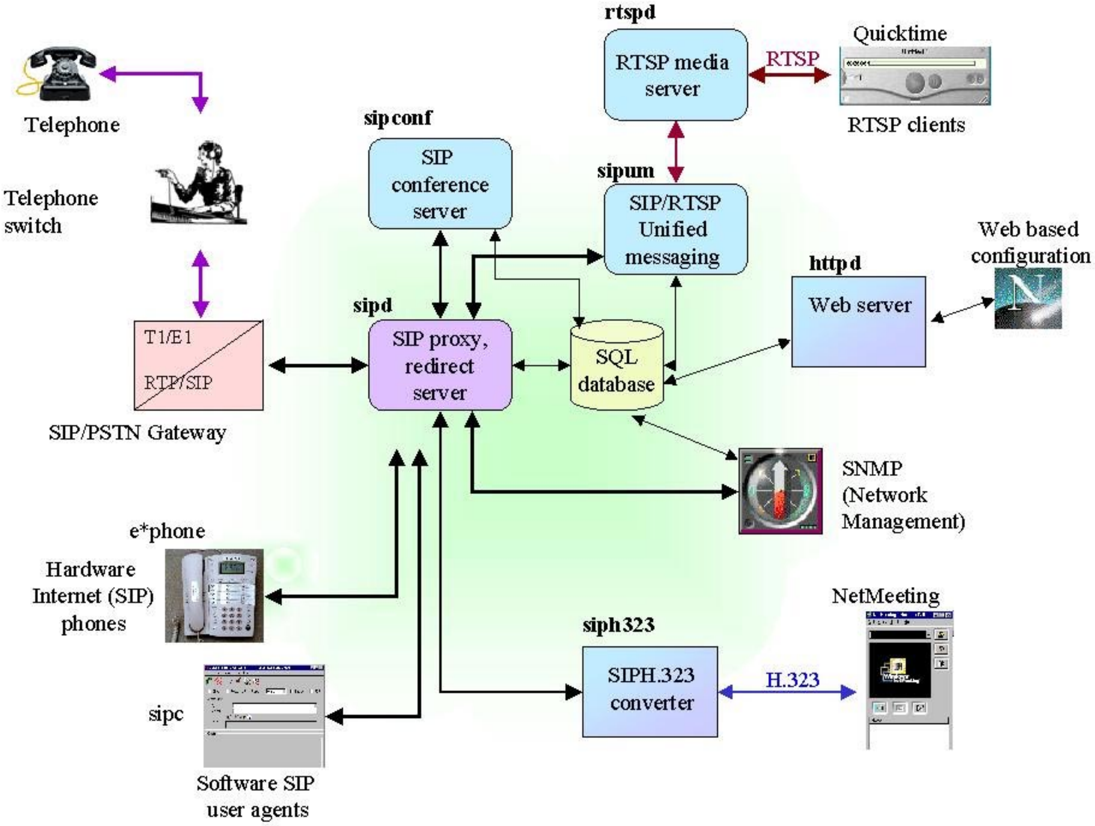

Figure 1: Architecture

SIP calls can also use "tel" URLs that identify E.164 telephone numbers [25], for example, `tel:+1-212-555-1234`.

The list of supported audio and video algorithms and the transport addresses to receive them are described using Session Description Protocol (SDP [4]), carried in SIP requests and responses.

3. ARCHITECTURE

3.1 Components

Fig. 1 shows the architecture and interaction among the components of our test bed.

SIP server: sipd is a SIP proxy, redirect and registration server.

SQL database: sipd uses the MySQL [13] database for storing the current network addresses and phone numbers where the user can be reached. The database also stores other per-user information related to voice mail and conferences.

PSTN gateway: A Cisco 2600 router with SIP/PSTN capability is connected to the departmental telephone switch (PBX) with a T1 trunk and to the department LAN.

User agents: SIP user agents (SIP UAs) allow users to interact with the system over an IP connection. They can be either hardware (Ethernet phone) or software based. Our e*phone [8] is an example of an Ethernet phone, whereas sipc [9] is software running on workstations and PCs. We also use Ethernet phones from Cisco, Pingtel and 3Com in our test bed.

Media server: rtspd is our general-purpose streaming media server, which we use for the storage and delivery of announcements and voice mail messages [24].

Unified messaging: sipum is a centralized answering machine and voice mail system [24] that uses rtspd for storing announcements and messages.

Conference server: sipconf is a centralized audio/video conference server [22].

SIP-H.323 translator: sip323 is a signaling gateway [23] between SIP and H.323. H.323 [7] is ITU-T's standard for multimedia conferencing over any packet based network. sip323 integrates popular H.323 clients such as Microsoft NetMeeting into a SIP infrastructure.

3.2 User Database

The SIP server and the SQL database form the core of the infrastructure, while the other components can be selectively enabled or disabled. For example, if an installation does not intend to use NetMeeting, it does not need sip323.

Every user of the system is given a unique identifier of the form *user@domain*, also called a canonical user identifier.

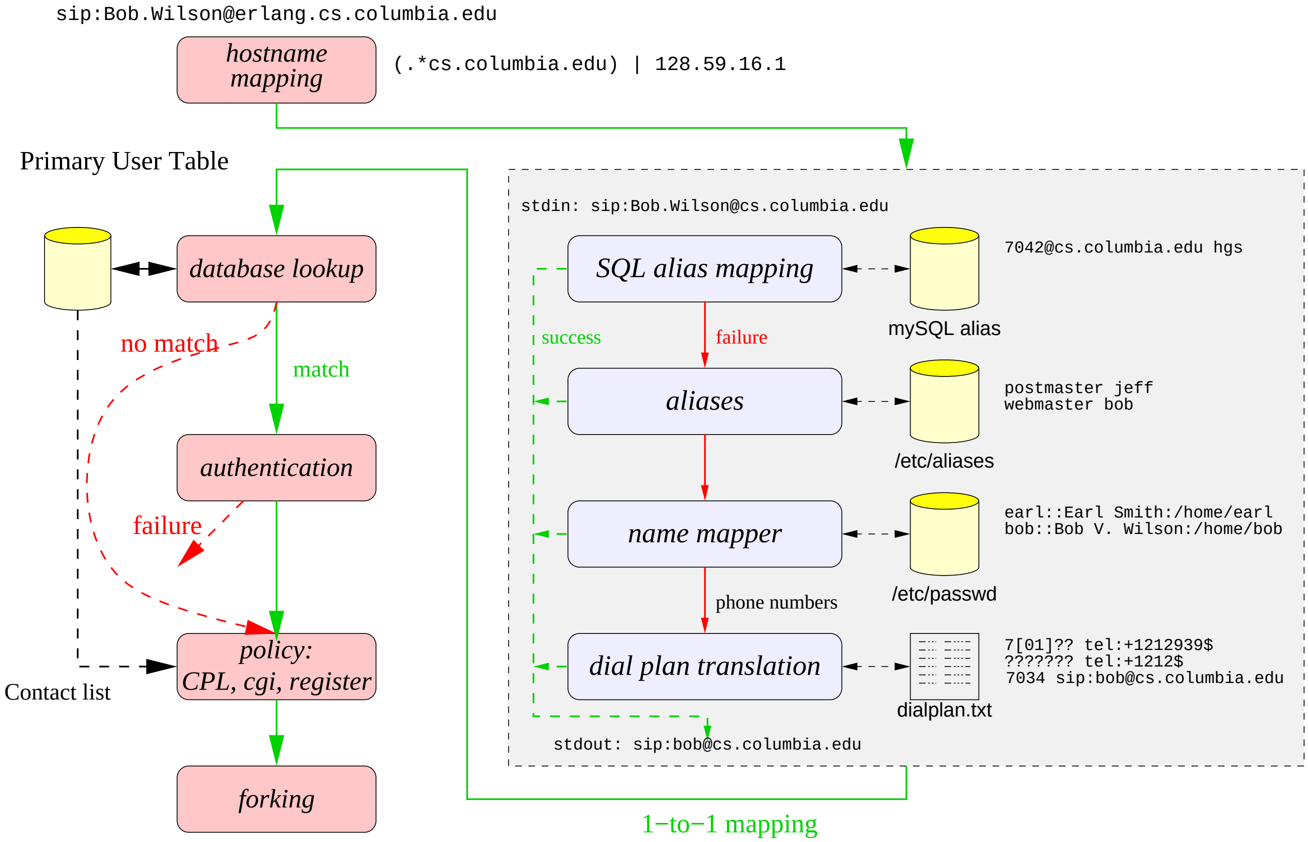

Figure 2: Canonicalization, authentication and routing for a call

Although all the local identifiers are *user@cs.columbia.edu*, the domain portion in the identifier allows for unique identification and authentication. Generally, users are assigned their email address as SIP identifiers. However, our system can also operate in "portal" mode, where a new identity is created specifically for SIP calls.

User accounts are created and managed through a web page. Users can alter their profiles such as password, authentication mechanism, or voice mail notification format. The user information is stored in the SQL database as the Primary User Table and indexed by the user identifier. The system distinguishes between regular users and administrators, in terms of access privileges.

There are other tables in the MySQL database. For instance, the `contacts` table stores the current locations of the registered users, which can be updated from the web page or by the SIP phones using SIP registration. It also contains the expiration time until which the location information needs to be refreshed, the preference value to sort multiple registered locations for the same user, and the action parameter to redirect or proxy an incoming call for this user. The `alias` table stores aliases and nicknames of all users.

3.3 Incoming Calls

An incoming call is processed as shown in Fig. 2. Here, the user sip:alice@cs.columbia.edu calls sip:Bob.Wilson@cs.columbia.edu. Through DNS SRV records, Alice finds out that the host `erlang.cs.columbia.edu` serves SIP requests for the `cs.columbia.edu` domain. We assume that Bob can be reached in many different ways, for example, as `bob`, `Bob.Wilson`, `bob_wilson`, `Bob.V.Wilson`, `webmaster`.

After validating the syntax of the call request, the server transforms the callee address to a canonical user identifier for database lookup, by first transforming the host portion and then the user name portion. For example, the domain portion, erlang.cs.columbia.edu is canonicalized to cs.columbia.edu. This is done by matching the domain portion of the request URI against a regular expression. For example,

```
(.*cs.columbia.edu) | (128.59.(1[6-9]|2[0-3]).[0-9]*)
```

maps all host names and IP addresses from 128.59.16.0 to 128.59.23.255 in the CS domain to the canonical server address of cs.columbia.edu. If the canonicalized host name does not match, the server is being used as an "outbound proxy server" and just routes the request to the SIP server for the domain, without any processing. Outbound proxy servers are useful for logging and firewall control, for example. Outbound proxies are not needed for "sip" URLs, but SIP requests with "tel" URLs need to designate such a proxy to translate the telephone number into a routable SIP identifier. This SIP identifier can either be at a PSTN gateway or be a regular *sip:user@host* URL.

The server then passes the user identifier to a coprocess called *canonicalize* which translates usernames to a canonical form. There are four ways that usernames can be translated, by SQL alias, by system alias, by name mapping and by dial plan translation. First, the canonicalize process queries the aliases table of the SQL database (described in Section 3.2) to see if there is an alias entry for the user. If there is, the alias is resolved to its canonical identifier and returned. Otherwise, the system next checks the system's email aliases file. This file typically records functional aliases such as "postmaster" or "webmaster". As a third step, the *canonicalize* name mapper function searches the system password file to see if it can deduce a username, by comparing the request URI to various combinations of the first and last name recorded in the password entry. (In the example, the name mapper determines from the system password database that the name "Bob Wilson" corresponds to the user bob.) Finally, if the user identifier looks like a telephone number, such as sip:7018@cs.columbia.edu or is a "tel" URL such as tel:7018, *canonicalize* performs dial plan transformations, which are described in more detail in Section 4.1. If none of the rules match, the user identifier is returned unchanged to the server.

The SIP server then retrieves user, contact, and policy information for the user bob@cs.columbia.edu. The policy information describes how the call is handled, for example whether it is to be proxied or redirected. Bob's preferences and policy are then executed. These may, for example, demand that a calling user be authenticated, refuse or redirect calls, or apply preferences about where Bob wants to be reached. If the server determines that Bob's current policy allows Alice's call to reach him, it contacts Bob's list of registered locations. Bob's current SIP phone rings, he picks up the handset and starts talking to Alice. When they are done, either of them can terminate the call.

Note that the SIP server normally does not store any call state information and is responsible only for forwarding requests and responses. These could be the call initiation and termination requests, or something different like an instant message. While the call initiation message goes through the SIP server, the call termination message may be directly exchanged by the two user agents without any SIP server. However, the server can insist on being in the call path for all the messages using SIP Record-Route option [5]. This is useful for call logging and accounting.

Users register their network location with a registration server, typically colocated with a proxy server. A single user name can be registered at any number of devices.

4. PSTN INTER-OPERATION

4.1 Dialplan

PSTN subscribers are identified by telephone numbers rather than email or IP addresses. A PSTN user can reach the gateway by dialing any of the extensions assigned to the gateway's T1 line. For example, our PBX has assigned extensions in the range 7130-7139 to the gateway. So anybody who dials (212) 939-7134 reaches the gateway at extension 7134. On the gateway, we need to define a voice over IP call-leg specifier (called a dial peer). An example where the SIP server's IP address is 128.59.19.62[2], is as follows:

```
dial-peer voice 1 voip
 preference 1
 destination-pattern 713[0-9]
 voice-class codec 1
 session protocol sipv2
 session target ipv4:128.59.19.62
```

The following example is a POTS (Plain Old Telephone Service) dial peer for specifying 7-digit local calls from SIP to PSTN.

```
dial-peer voice 1005 pots
 preference 6
 destination-pattern 8.......
 no digit-strip
 port 1/0:1
```

A "." is a wildcard for any digit, and "8" is the prefix the user must dial to reach a number outside our PBX. The "preference" parameter is used to match dial peers in a certain order.

When a call comes in from the PSTN, the gateway can react in one of the two modes, direct-inward-dialing (DID) or no-DID. In DID mode, the incoming trunk delivers the destination extension to the PBX or gateway. So a call to 7134 is forwarded to the SIP server as sip:7134@128.59.19.62. The SIP server maintains a mapping between the telephone number and the user identifier. The mapping is called a dialplan. For example, 7134 can be mapped to sip:bob@cs.columbia.edu so that the above call reaches Bob at his SIP phone (see Section 3.3). For the 713x range, the DID mode can support only up to 10 users. In the no-DID mode, the gateway will prompt the caller with a second dial tone. After the caller dials a new extension, it is captured and forwarded to the SIP server. The differences between the two modes are summarized below.

Mode	usage	advantages
DID	dial directly	simpler dialing from PSTN
No-DID	dial extension	supports more users

In the reverse direction, when a SIP user dials a telephone number, e.g., sip:9397040@cs.columbia.edu, the SIP server transforms the telephone number to the telephone subscriber tel:+12129397040. Also, as is typical for PBXs, the same number can be dialed in a number of different ways, for example, as a four-digit extension (7042), as a local phone number (939-7042) or as a global number (1-212-939-7042), with country code. In addition, PBXs often designate a digit such as 8, 9 or 0 to reach an outside line. Thus, for IP telephones, which often follow the mobile phone model of requiring an explicit indication of the end of a phone number, a large number of variations need to be unified into a single global number which can then be used to determine the appropriate gateway.

This model is reflected in the following sample dialplan used for our server, where both a 4-digit extension and a 7-digit local number are mapped to a canonical format with country code, area code, and local number. The symbol "$" is substituted by the matched string on the left column, while "?" matches a single digit and "*" matches any digit string.

```
# Intra-department calls
7[01]??     tel:+1212939$
# Local (same area code) calls
???????     tel:+1212$
# Numbers prefixed by '8' are treated the same
(8)???????  tel:+1212$
# International numbers
(011)*      tel:+$
(8011)*     tel:+$
```

[2] The IP addresses and net masks are not necessarily real.

The server then locates the appropriate gateway to route the call to the PSTN. For an organization with a small number of gateways, a static table, as currently used in sipd, is sufficient. If networks of IP telephony gateways are deployed, more complex routing protocols such as TRIP [17] may become essential. TRIP allows to route the call to the optimal gateway, e.g., the one closest to the destination.

In our system, each local user is assigned a “gateway class”, such as faculty, staff or student. The gateway mapping table contains mappings such as the following:

```
(+1212939)7[01]??  full,guest  sip:$@gateway.office.com
```

Here, any call to 7000 through 7199 made by users from the “full” and “guest” classes will be routed to the gateway gateway.office.com.

The server may terminate the call if the caller does not have sufficient privileges. For example, anybody may be allowed to make intra-department and toll-free telephone calls but only the faculty and administrative staff may be allowed to make local or long-distance calls using the gateway.

4.2 Connecting to the PBX

PBX (Private Branch eXchange) is used in many corporations and universities. It centralizes telephone management, consoliates external trunk lines and voice mail. Our PBX is a Nortel Meridian Option 11C. It has an external T1 line to the public telephone network, capable of 24 incoming/outgoing calls. It also has an internal T1 line to connect with the PSTN/IP gateway. With this topology, a user can make IP telephone calls from either an analog phone (whether inside or outside the department) or a SIP UA.

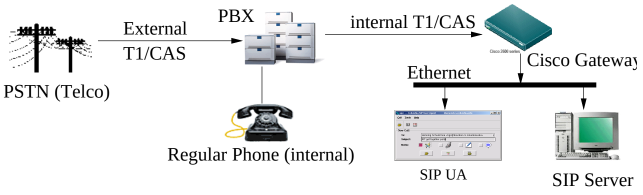

Figure 3: PBX set-up; an incoming call flow

During deployment, we encountered quite a few problems that are worth describing. Many of them have to do with the proprietary and arcane nature of PBX systems.

T1 line type: A T1 line can be either channelized or PRI [1, pages 446-447]. The former supports 24 DS-0 (64 kb/s PCM) voice channels, and uses Channel Associated Signaling (CAS). CAS is a form of in-band signaling, where some bits in each voice channel are “robbed” for signaling, hence the nickname robbed-bit signaling. In comparison, PRI supports 23 DS-0 B (voice) channels plus 1 DS-0 D (signaling) channel, and uses an out-of-band signaling method known as Common Channel Signaling (CCS). PRI is a form of business grade narrow-band ISDN. Channelized T1 has more voice channels, but each channel is not full 64 kb/s, and it is not guaranteed to provide advanced features such as Caller-ID. We use channelized T1 in our PBX for both T1 lines. The PRI service may require additional hardware in the PBX’s T1 line-card.

Line type	voice channels	signaling	caller-ID
Channelized	24, robbed-bit	CAS	maybe
PRI (ISDN)	23B + 1D	CCS	yes

T1 line characteristics: First, T1 lines can use several different line codings, including AMI or B8ZS [1, pages 175-182]. We recommend B8ZS because it provides a full 64 kb/s for each DS-0 channel, whereas AMI steals one out of every eight bits (leading to 56 kb/s channel), thus degrading the voice quality. The line coding is *not* always independent of the line type. For instance, AMI cannot be chosen with PRI, because PRI requires a full 64 kb/s channel. Second, one needs to select a framing type, usually either Super Frame (SF), also known as D4, or ESF (Extended SF) [1, pages 210-216]. We choose ESF, which is more advanced and should be supported on most PBX systems.

Trunk type: The most popular trunk types are DID (Direct Inward Dial) and TIE. A TIE line is a bi-directional trunk line. The name TIE comes from the fact that the trunk line “ties” two nodes together. We recommend configuring the T1 line as a TIE trunk, because it allows both DID (incoming) and outgoing calls.

Channel type: The channel type can be data, voice-only, or data/voice. This is a crucial parameter. If a channelized T1 line is used on a Meridian system, the channel type must be set to voice-only, otherwise, IP-to-PSTN calls may fail as the PBX could treat a call as data transmission. In a Nortel Meridian PBX system, this parameter is named DSEL (Data SELector).

Access permissions: Nortel Meridian systems use a concept called Network Class Of Service (NCOS). Typically, a low NCOS means low access permission. For example, 1 may indicate internal or local call only, and 7 may indicate all long-distance allowed. So if one wants to restrict his SIP phones to local PSTN (outgoing[3]) calls only, he should specify a NCOS of 1 for the internal T1 line.

For incoming calls, however, the scheme is less obvious: when a call arrives at the PBX, whether incoming or outgoing, the calling entity’s NCOS is compared against the callee’s NCOS in the PBX call routing table. Note that the callee in this case is *not* the internal T1 line, but the 713x range - a virtual entity. The calling entity for an incoming call is the external T1 trunk, and it usually has a NCOS of 0. The call goes through only if the caller’s NCOS is high enough. In our test-bed, the 713x range is a virtual phone number range, being a part of what is called Coordinated Dialing Plan (CDP). One must ensure that the routing entry for this CDP (713x) range has an NCOS value less than or equal to the caller’s NCOS, so that incoming calls can be accepted. Therefore we use an NCOS of 0 for the CDP entry.

These issues are summarized in Table 1.

[3] Here outgoing and incoming are viewed from the perspective of the PBX (or the department).

T1 attributes	common choices	recommended
Line Type	Channelized, PRI (ISDN)	PRI, used channelized
Line Coding	AMI, B8ZS	B8ZS
Framing	D4(SF), ESF	ESF
Trunk Type	DID, TIE	TIE
Channel Type	data, voice-only, voice-Data	voice-only

Table 1: Summary of key attributes

4.3 Security Issues

We need to deal with three security-related issues, namely user registrations, remote callers and access to the PSTN. First, user registrations need to be authenticated to prevent unauthorized users from redirecting calls to themselves or elsewhere. We use digest authentication, where a shared secret between the server and the client is verified via challenge-response.

Secondly, a local user may choose to force remote callers to be authenticated. We cannot rely on a public key infrastructure, so we chose a more pragmatic, albeit less secure, approach. Our authentication goal is to establish a consistent mapping between a caller's SIP identity and her email identity. If a caller is unknown, a mail message is sent to the same identifier, treated as an email address. The mail message contains a randomly generated password and a link to the original called SIP URL. The caller simply retries the call after receiving the email message and stores the secret for future use. This ensures that the SIP caller is indeed identical to the corresponding email address. (One approach that does not work is to simply have the callee issue an INVITE in the reverse direction. This could be easily abused to cause somebody to make nuisance calls to a third party.)

It should be noted that it is much harder to use call filtering to prevent VoIP crank calls than their PSTN equivalent since Internet identifiers are abundant and cheap. However, it is possible to at least restrict unknown callers to, say, daytime hours or leaving voicemail.

Finally, we need to restrict access to the PSTN gateway. In most cases only an outgoing call incurs a toll charge. The last line of security in such case is the PBX, but the PSTN/IP gateway and the SIP proxy server are in general much more flexible and programmable. Currently, our gateway does not have user authentication and authorization capability, so we delegate this functionality to the SIP proxy server, so that only authorized users can make calls. However, if a user discovers the gateway's IP address, he can still bypass the proxy and make "free" calls. To enforce security without adding security code to the gateway, we can make the gateway "reject" direct-dialed calls. Since our gateway is a fully functional Cisco 2600 router with IOS (Internetwork Operating System), we can use the IOS Access Control Lists (ACL) to accept SIP requests only from the proxy, but accept UDP media streams from all potential users. The following example IOS ACL blocks certain inbound traffic on the gateway 128.59.19.61. Its subnet has a net mask of 255.255.255.0, hence its ACL's reverse mask is 0.0.0.255. The gateway would allow UDP media packets from machines on the same subnet with port number range 512 to 65535. However, since SIP requests are typically carried on UDP port 5060 (inside 512-65535), to reject "direct-dialed" SIP calls, our ACL allows a UDP packet only if its destination is not the gateway (128.59.19.61)'s SIP port (5060). Note that the SIP proxy server (128.59.19.62)'s request will still be honored because ACL rules are evaluated in the same order they are defined.

```
interface FastEthernet0/0
 ip address 128.59.19.61 255.255.255.0
 ip access-group 101 in
...
access-list 101 permit ip host 128.59.19.62 any
access-list 101 permit udp 128.59.19.0 0.0.0.255 \
   range 512 65535 host 128.59.19.61 neq 5060
```

All the calls from the gateway are forwarded to the SIP server. This configuration, along with the SIP Record-Route mechanism that forces subsequent requests within a call to traverse a designated set of proxies, allows call logging and billing services to be part of the SIP server.

5. OTHER SERVICES

This section describes other services provided by the system.

5.1 Programmable Call Handling

When receiving an incoming call request, the SIP server finds the current user location and either proxies, redirects or rejects the call initiation message. Although this simple model satisfies most of the needs, some advanced users may want a more complex scenario. For example, "reach me at my office phone during office hours and call me at my home after office hours, or don't disturb me when a telemarketer calls." This can be implemented by uploading a piece of software on the server, which governs its behavior based on the time-of-day or caller identification. SIP allows many different ways to achieve this, for example, via the XML based Call Processing Language (CPL [16, 11]) and the SIP Common Gateway Interface [12]. The latter is similar to HTTP-CGI. SIP-CGI scripts can be written in any language. Our SIP server, sipd, supports SIP-CGI. A CPL implementation is in progress. The piece of software which alters the server behavior, either a SIP-CGI or a CPL script, can be uploaded to the server using a SIP UA such as sipc, or edited from a web page. An example of a service is a calendar-based call routing system. Calendaring and scheduling information formatted as iCal [2] is combined with a policy file and then converted into a CPL script, which is uploaded to the server. The policy expresses rules such as "if Joe calls and I'm busy (according to my calendar), forward to secretary". Services can also be driven by caller preferences [21], where the caller indicates desired call routing and handling behavior. For example, a caller may request that calls are not forked or that calls are not routed to voicemail or attendants.

5.2 Unified Messaging

Answering machines and voice mail systems are crucial PSTN components. They are equally important in an Internet telephony environment. Installing the voice mail service on every SIP phone is inefficient and inconvenient if the user has many phones. Secondly, it may not work if the user has calls forwarded to many different devices. Also, end-system-based answering machines place a high premium on the reliability of those end systems. Centralized voice mail systems have an advantage in the centralized management of user accounts and configuration. An Internet-based voice mail system can be integrated easily with other Internet services like email, web, video mails and fax, giving an unified messaging environment. Moreover, it can use the existing protocols and tools, like SIP and RTSP (Real Time Streaming Protocol [19]).

Our system uses SIP for signaling and RTSP for storage and retrieval of voice messages as described in [24]. The user gets an email notification when a new message arrives. The user messages are also listed on a web page, where they can be played by just a mouse-click. Alternatively, an RTSP client such as Apple's QuickTime can be used to play back the message. Using streaming media to deliver voicemail avoids having to download the whole message while traveling, for example.

5.3 Multi-party Conferencing

Multi-party conferencing is also an important telephony service, provided in the PSTN by conference bridges. Our Internet telephony environment employs a SIP conference server with audio and video capabilities. The conferences can be set up via a web interface.

Every conference is identified by an address similar to the canonical user identifier, e.g., staffmeet@cs.columbia.edu. Users join the conference by dialing that conference address. The dialplan in the SIP server can map telephone numbers to the conference addresses, so that regular PSTN users can also take part in conferences. This requires dynamic modification of the dialplan.

The system can be extended to provide dial-out conferences instead of the traditional dial-in conferences. In this mode, the conference server itself invites the participants at the start of a pre-configured conference.

The SQL database stores various conference attributes and can be updated from a web page. These include the conference identifier, duration and schedule, authentication mechanism for restricted conferences, limit on the number of participants, types of media allowed, and so on. The participant list can be further restricted by defining different capabilities for different set of participants. For instance, one may want a conference where anybody can listen but only users *@cs.columbia.edu* can send media. Authentication of PSTN phones requires some form of voice interface, which we are currently adding.

6. SCALABILITY

This section describes some of the scalability issues which may be encountered when we extend the system to large scale environment with thousands of registered users. The distributed nature of the components in our environment allows putting more components to meet the needs of a large-scale user base. For example, multiple conference servers can be installed, with each running only tens of active conferences.

For scaling proxy servers, we make use of the DNS SRV capability in SIP. A DNS SRV [3] resource record lists a set of servers, ordered by priority, for each service and domain. For example,

```
example.com
_sip._udp 0 40 a.example.com
          0 40 b.example.com
          0 20 c.example.com
          1  0 backup.somewhere.com
```

indicates that the servers a, b, c should be used if possible, with backup.somewhere.com as the backup server. Within the three primary servers, a and b are to receive a combined total of 80% of the requests, while c, presumably a slower server, should get the remaining 20%. Weighted randomization by the client is used to achieve this distribution.

However, simple randomization is not sufficient since servers need to share access to the same registration information. Thus, in the example above, each server would have to replicate incoming REGISTER requests to all other servers or update a common shared database. In either case, updates or lookups would quickly become the bottleneck. (In SIP, clients typically register once an hour, thus, for a wireless operator with one million phones, the database has to process about 280 requests per second.)

We solve this problem in our two-stage scaling architecture, shown in Fig. 4. We divide a domain server group into two parts, a set of stateless proxy servers and a set of clusters. The first set of proxy servers perform very simple stateless request routing. For example, they may route based on a hash of the user identifier, with each hashing range mapping to a particular second-stage server, or, for reliability, a cluster of servers. The cluster member is again determined via the DNS SRV. In the figure, each letter of the alphabet gets its own server cluster, so that the request for bob@example.com is routed to a server in the "b" cluster. The second-stage server then performs the actual request processing.

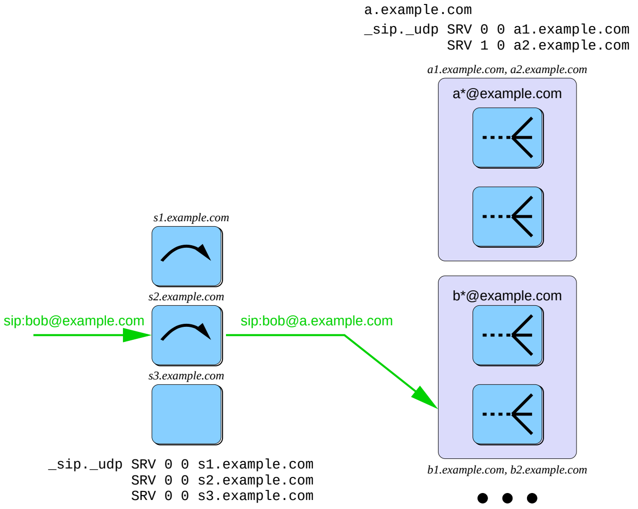

Figure 4: SIP scalability using DNS SRV

The architecture has the advantage that it scales for both call setup and registration. Each server in a cluster forwards each registration, via multicast or unicast, to all other members in a cluster, but since registrations are split by weight, not all servers for the domain need to see every registration. This architecture scales to any desired processing load and user population size. (A related scaling technique is used by large-scale email.)

Moreover, SIP's redirect feature can also be used to delegate requests to less loaded servers.

The SIP server and the unified messaging server do not handle the real-time media packets, so their load is relatively low. On the other hand, the RTSP media server and the SIP conference server do need to handle the media streams. However, use of multiple servers distributes the load and is scalable.

The primary bottleneck is the SIP-PSTN gateway. Given that a single T1 line can support only 24 simultaneous calls, larger systems will require more T1 lines between the gateway and the PBX. If cost permits, installing multiple gateways is an option. Alternatively, some PBXs can now be equipped with Ethernet interfaces. In the future, it appears likely that carriers will offer gateway services, often called "IP centrex", removing the need for smaller organizations to have their own gateways. (The term IP centrex is somewhat misleading since, unlike for traditional centrex, media streams between extensions of the same organization do not have to be sent to the IP centrex.)

The LAN bandwidth could be another bottleneck. For gateway calls alone, if all 24 voice channels on the T1 trunk are active, without silence suppression, and assuming 64 kb/s PCM encoding, the voice traffic is 24×64 kb/s $= 1.536$ Mb/s in each direction. The total traffic is $1.536 \times 2 = 3.072$ Mb/s in full duplex. IP/UDP/RTP has an overhead of about 40 bytes per packet, so the gross traffic will be somewhat higher. With a 20 ms packet interval, each PCM packet is $160+40 = 200$ bytes, leading to a gross load of $200/160 \times 3.072 = 3.84$ Mb/s. With a 40 ms packet interval, gross load is 3.456 Mb/s. 3-4 Mb/s may represent a medium load and cause increased delay for the gateway on a 10 Mb/s Ethernet interface. Choosing a low bit-rate codec such as G.729 [6] (8 kb/s) reduces the traffic dramatically to 384 kb/s, not counting IP/UDP/RTP overhead. Low bit-rate codecs are, however, more sensitive to overhead. With 20 ms packet interval, gross traffic becomes $(20+40)/20 \times 384 = 1.152$ Mb/s. 40 ms is more efficient, leading to $(40+40)/40 \times 384 = 768$ kb/s. Silence suppression can reduce the traffic further by at least half, since voice activity factor is typically 40-45% [10]. Therefore, LAN bandwidth should not be a bottleneck for a few tens of simultaneous calls if we use a fast network interface (e.g., 100 Mb/s), a low bit-rate codec, silence suppression, or a combination of these options.

7. RELATED WORK

Internet telephony has become an active area, with a number of companies such as Net2Phone, DialPad and MediaRing providing PC-to-PC and PC-to-phone calls. Their objective is mainly to provide low-cost call service to PSTN from the public Internet, whereas our architecture is well-suited for Internet telephony infrastructure within an organization. This is initially intended to minimize telephone infrastructure and service costs for an organization. Internal calls can be carried over IP with virtually no added cost. The infrastructure is connected to the public Internet as well as the external PSTN network. It allows external callers on the Internet to reach a SIP user or an internal PSTN number within the organization for free.

8. CONCLUSIONS AND FUTURE WORK

We have described the architecture of our Internet telephony installation consisting of the SIP server, SIP-PSTN gateway, RTSP media server, unified messaging server, conferencing server and SIP-H.323 translator.

The test-bed is initially intended for small scale experiments within the department and later to be extended to a campus-wide Internet telephony environment. A similar architecture can be deployed at other campus and organization networks who want to benefit from the services provided by Internet telephony, in particular SIP.

We will continue with integration of additional services. For example, SIP-based instant messaging and presence will allow a standard way to send instant messages and form buddy lists [14, 15]. Combining presence and Internet telephony offers improved services, reducing, for example, the number of failed call attempts or involuntary redirects to voice mail. We are implemementing a VoiceXML [26] browser that allows us to easily implement services such as retrieving email, including voicemail, via traditional phones, but also simplifies the task of building voice menus, making such services available to small organizations. (VoiceXML is an XML DTD that mimics HTML forms input via DTMF or speech recognition.) SNMP (Simple Network Management Protocol) based monitoring and control of the SIP server has been implemented.

We are currently instrumenting our proxy and conference server to better understand how to build highly scalable systems. The performance evaluation of SIP servers is more difficult than that of, say, web servers since finding the maximum operating rate is complicated by SIP's use of UDP, causing packet loss and retransmissions under overload. Also, the workload is likely to differ dramatically between registration-bound mobility service and script-processing-bound service engines.

A commercial deployment involves many other issues related to security, billing and quality of service. One of the disadvantages of current PBXs is that it is hard to tailor their billing information to local needs, usually resulting in having to manually inspect paper printouts. We are implementing call logging in sipd via an SQL database interface. Also, enhanced security and encryption may be needed when operating over the Internet. Interworking with the corporate firewalls and Network Address Translators (NATs) is another challenge. We are also planning developing a Windows CE version of our SIP UA, making it possible to integrate wireless PDAs into the infrastructure.

Finally, a short-term goal is to deploy the system throughout the Computer Science department and then be able to replace our PBX. In the long term, we will provide direct SIP services for integrated access, and address Quality of Service issues.

9. ACKNOWLEDGMENTS

We would like to thank Enlai Chu of Cisco, Keane Chin of SIP Communications Inc., and Sarmistha Dutta of Columbia University for their help with the Cisco gateway and departmental PBX, Xiaotao Wu for sipc and Tarun Kapoor for the MySQL database installation.

This work was supported by equipment and research grants from 3Com, Cisco, Clarent Communications, Lucent, SIPcomm, and Sylantro.

10. REFERENCES

[1] J. Bellamy. *Digital Telephony.* John Wiley & Sons, New York, 1991.

[2] F. Dawson and D. Stenerson. Internet calendaring and scheduling core object specification (icalendar). Request for Comments 2445, Internet Engineering Task Force, Nov. 1998.

[3] A. Gulbrandsen, P. Vixie, and L. Esibov. A DNS RR for specifying the location of services (DNS SRV). Request for Comments 2782, Internet Engineering Task Force, Feb. 2000.

[4] M. Handley and V. Jacobson. SDP: session description protocol. Request for Comments 2327, Internet Engineering Task Force, Apr. 1998.

[5] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. SIP: session initiation protocol. Request for Comments 2543, Internet Engineering Task Force, Mar. 1999.

[6] International Telecommunication Union. Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction. Recommendation G.729, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Mar. 1996.

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

[8] IRT Lab, Columbia University. E*phone home page. http://www.cs.columbia.edu/~hgs/ephone/.

[9] IRT Lab, Columbia University. sipc home page. http://www.cs.columbia.edu/IRT/software/sipc.

[10] W. Jiang and H. Schulzrinne. Analysis of on-off patterns in VoIP and their effect on voice traffic aggregation. In *International Conference on Computer Communication and Network*, Las Vegas, Nevada, Oct. 2000.

[11] J. Lennox and H. Schulzrinne. CPL: a language for user control of internet telephony services. Internet Draft, Internet Engineering Task Force, July 2000. Work in progress.

[12] J. Lennox, H. Schulzrinne, and J. Rosenberg. Common gateway interface for SIP. Request for Comments 3050, Internet Engineering Task Force, Jan. 2001.

[13] MySQL AB Co. MySQL home page. http://www.mysql.com.

[14] J. Rosenberg et al. SIP extensions for instant messaging. Internet Draft, Internet Engineering Task Force, Apr. 2001. Work in progress.

[15] J. Rosenberg et al. SIP extensions for presence. Internet Draft, Internet Engineering Task Force, Apr. 2001. Work in progress.

[16] J. Rosenberg, J. Lennox, and H. Schulzrinne. Programming Internet telephony services. *IEEE Network*, 13(3):42–49, May/June 1999.

[17] J. Rosenberg and H. Schulzrinne. A framework for telephony routing over IP. Request for Comments 2871, Internet Engineering Task Force, June 2000.

[18] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: a transport protocol for real-time applications. Request for Comments 1889, Internet Engineering Task Force, Jan. 1996.

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

[20] H. Schulzrinne and J. Rosenberg. Internet telephony: Architecture and protocols – an IETF perspective. *Computer Networks and ISDN Systems*, 31(3):237–255, Feb. 1999.

[21] H. Schulzrinne and J. Rosenberg. SIP caller preferences and callee capabilities. Internet Draft, Internet Engineering Task Force, Nov. 2000. Work in progress.

[22] K. Singh, G. Nair, and H. Schulzrinne. Centralized conferencing using SIP. In *Internet Telephony Workshop 2001*, New York, Apr. 2001.

[23] K. Singh and H. Schulzrinne. Interworking between SIP/SDP and H.323. In *Proceedings of the 1st IP-Telephony Workshop (IPtel 2000)*, Berlin, Germany, Apr. 2000.

[24] K. Singh and H. Schulzrinne. Unified messaging using SIP and RTSP. In *IP Telecom Services Workshop*, page 7, Atlanta, Georgia, Sept. 2000.

[25] A. Vaha-Sipila. URLs for telephone calls. Request for Comments 2806, Internet Engineering Task Force, Apr. 2000.

[26] VoiceXML Forum. Voicexml home page. http://www.voicexml.org/.