Internet Telephony Gateway Location

Jonathan Rosenberg Henning Schulzrinne
Bell Laboratories Columbia University

Abstract

Although the Internet was designed to handle non-real time data traffic, it is being used increasingly to carry voice and video. One important class of contributors to this growth are Internet telephones. Critical to more widespread use of Internet telephony is smooth interoperability with the existing telephone network. This interoperability comes through the use of Internet Telephony Gateway's (ITG's) which perform protocol translation between an IP network and the Public Switched Telephone Network (PSTN). In order for an IP host to call a user on the PSTN, the IP host must know the IP address of an appropriate gateway. We consider here the problem of finding these gateways. An analysis of a number of protocol architectures is presented, including hierarchical databases, multicast advertisement, routing protocols, and centralized databases. We propose a new protocol architecture, called Brokered Multicast Advertisements (BMA) which serves as a lightweight, scalable mechanism for locating ITG's. The BMA architecure is general, and can be applied to location of any service across a wide area network.

1 Introduction

Although the Internet was designed to handle non-real time data traffic, it is being used increasingly to carry voice and video. One important class of contributors to this growth are Internet telephones. Typically implemented as software in a PC, these applications allow users on IP hosts to communicate using voice. There are dozens of different applications available, many of which are free (vat [1], Nevot [2], and rat [3], for example).

Besides reducing cost, Internet telephony can bring a host of new features and capabilities to this traditional medium [4]: better user interfaces, high quality speech, integration with other tools (like web browsers and appointment books), personal mobility, and a wide assortment of supplementary services, to name a few. Despite these advantages, there are a number of barriers to more widespread use of Internet telephony. Most prominent among them is the poor quality of voice connections on the Internet [5] [6] [7]. This problem is being attacked on a number of fronts, including forward error correction to recover from loss [8] [9], adaptive playout buffers [10][11] for jitter absorption, and resource reservation for improved network QoS [12][13]. Put togetther, all of these should improve voice quality on the Internet.

This is only half the picture, however. Internet telephony is only as useful as the set of people reachable through the service. There is no doubt that the current Public Switched Telephone Network (PSTN) will remain the dominant medium for carrying telephony services for quite some time. As a result, there will continue to be users who have traditional telephones, but who do not have Internet telephones. Many of the features provided by Internet telephony don't require the other party to have Internet telephony (better user interfaces and CTI are two excellent examples). For this reason, we believe that Internet telephony to PSTN interworking is an important service.

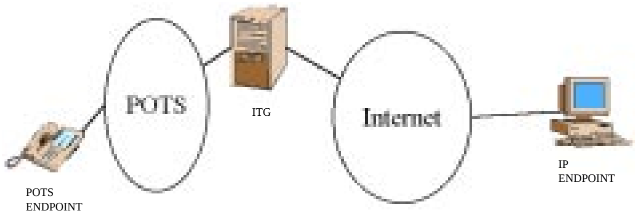

Figure 1: Scenario A

Providing connectivity between IP telephony users and PSTN end points is the function of an Internet Telephony Gateway (ITG), depicted in Figure 1. As with any device which connects two different networks, connectivity can be at several different layers. An ITG can either work at the network layer, or at the application layer. At the network layer, the ITG would translate routing and addressing information from the PSTN into IP routing and addressing information, and vice versa. As this would require significant changes to existing routers and telephone equipment, ITG's are likely to operate as application level proxies. This implies that they act as end systems on both the IP network and the PSTN. IP hosts wishing to contact a PSTN user would first contact the ITG, which would terminate the IP portion of the call and initiate a new call on the PSTN to the final destination.

With application gateways, it is necessary for the end-systems to first contact the gateway before reaching their final destination. Note that the actual user may not be aware of this operation; it is only the underlying software and hardware which must contact the ITG. The problem exists on both sides of the ITG; a PSTN user must first connect to a gateway before reaching an IP host, and an IP host must first contact a gateway before reaching a PSTN user. Because of the differences in user interfaces and network architecture, these two problems are solved in different fashions. In the remainder of this paper, we consider the location problem only from the viewpoint of an IP host.

2 Problem Definition

Locating any sort of service on the Internet is not a new problem. IP end-systems must be able to locate a DNS server, for example. This is typically done through either a static configuration, or via protocols like DHCP [14]. Unfortunately, ITG's provide a much different service than a DNS server. The nature of this service makes their location a much harder problem.

First and foremeost, it is sufficient for an IP host to use the DNS server which is closest to it, in terms of IP hops. Usually, the network administrator will run a DNS server, and configure all of its clients to use it. There are no per-access charges associated with DNS lookups. Any cost for providing DNS service (like computer depreciation) is wrapped into monthly access charges, if any. There is generally no reason for a host to use a DNS server besides the one provided by its ISP.

This situation is almost the exact opposite for an ITG. Unlike DNS services, there is a cost associated with completing the call, since the ITG must dial the final endpoint. This will cause charges to be accrued by the ITG administrator, which must then be passed back to the user. These costs depend on the distance from the ITG to the final PSTN destination, the calling plan used by the ITG, the time of day, the volume of business, etc. To reduce these costs, a client may prefer to use an ITG which is situated as close to the final PSTN callee as possible. This would result in the cheapest call between the ITG and the destination, and therefore would minimize the cost passed on to the client. We are assuming that any costs associated with the Internet portion of the connection are destination independent.

There may also be cases where a client may desire to call a PSTN endpoint, but where cost is not an issue. Instead, having the best quality for the call may be important. This kind of scenario might be typical for business calls. Due to varying delays and losses on an IP network, the best quality is probably obtained by using an ITG which is closest to the IP host, as measured in terms of delay or IP hop count. We call such a selection criteria *proximity*.

Cost is not the only reason why an IP host may prefer to use one ITG over another. In an international calling environment, the set of protocols and billing mechanisms supported by ITG's can be expected to vary. Some ITG's may support billing of IP hosts via credit cards or e-cash [15] on a per-usage basis. Others may require the IP hosts to have set up an account ahead of time. The ITG's will have to perform speech transcoding to convert the codec used by the IP host to either G.711 [16] (the standard used in the PSTN), or analog. There are many speech coders used by IP clients. These include the 8 kb/s G.729 standard [17], the 5.3/6.3 kb/s G.723 coder [18], the 16 kb/s G.728 LD-CELP coder [19], and any number of proprietary codecs. There are also a whole host of higher rate, high quality speech coders, such as G.722 [20]. Since all ITG will not support all codecs, and since IP hosts may not all implement the same baseline codec, an IP host may need to select an ITG based on its speech coder support.

In addition to the speech coder, IP hosts may utilize a range of different signalling mechanisms for initiating and terminating calls, among other actions. ITG's must be able to recognize these protocols. H.323 [21] is gaining momentum as a popular standard for IP telephony. However, it is quite complicated, and other, more lightweight protocols exist. The Session Initiation Protocol (SIP) [22] is an example of another signalling protocol for Internet telephony which is gaining acceptance. An IP host may need to select an ITG based on which signalling protocols it can understand.

Authentication and encryption are also commonly used in Internet telephony. If an IP host wishes to encrypt the portion of the call between itself and the ITG, the ITG must support the particular encryption algorithm. This, too, becomes another criteria for gateway selection.

Implicit in much of the above discussion is the fact that an ITG need not be run by the same ISP that is used by a client. In fact, it is highly unlikely. ISP's are generally local, and its customers will probably want to use ITG's to make calls to locations that are not in the ISP's area of coverage. In fact, there is no reason why the administrator of an ITG need be an ISP at all. In an open business environment, it is important for IP hosts to be able to use ITG's from whatever service provider they desire. This in and of itself can then become another criteria for selection. An IP host may prefer to use an ITG administered by some large telecommunications provider, for example.

We are now in a position to state the problem. ITG's are run by possibly independent, widely distributed service providers. These ITG's may be scattered across the world, and may implement a variety of different protocols for billing, speech transcoding, signalling, and network transport. Usage of a gateway by an IP host to complete a call to a PSTN endpoint incurs a cost, which will be passed on to the host. The host must be able to determine the IP address of a gateway which meets the requirements of the user. These requirements include (but are not limited to) cost, proximity, and protocol support. What kind of protocol architecture is necessary to allow a host to locate an ITG?

In this paper, we explore the various dimensions of this problem. Section 3 defines the requirements for an ITG location protocol. Section 4 discusses the scope of the service. Section 5 reviews existing protocol architectures (and specific implementations of them) for their suitability. Section 7 presents a new protocol architecture, Brokered Multicast Advertisements (BMA) which is well suited to the gateway location problem. We also discuss its utility in the location of general wide area services.

3 Protocol Features

Some sort of application-level protocol architecture is required to allow IP hosts to select an ITG based on any number of criteria, including cost, protocol support, and proximity to the client. Besides multi-criteria selection capabilities, there are many other desirable features for such a protocol.

First, it is desirable for the protocol to operate in a distributed fashion, avoiding central registries. Central registries tend to lead to single points of failure, concentrate network traffic, cause excessive loads on the registry, and generally don't scale. They also tend to impose security risks. One must trust that the database administrator will not corrupt the entries.

The protocol should be fast. Since finding the ITG is required before a call can be placed, the call setup times are increased by the amount of time required for this protocol to operate. Therefore, rapid operation is important.

The protocol should not require large amounts of bandwidth. It is not acceptable, for example, for an IP host to query a long list of candidate gateways. This tends to create heavy loads for the network, and also causes loading on the ITG's themselves. The protocol should also be scalable. This implies several things. First, its bandwidth needs should not be excessive as its usage becomes widespread (definition of what it means to be widespread are discussed in Section 4). Furthermore, the processing load at any end system (ITG, IP host, or other device) should not be a burden either.

The protocol should be dynamic. As new ITG's come into existence, they should become accessible to IP hosts almost immediately, without requiring some kind of human entry into a database.

Similarly, if an ITG goes down, this information should be propagated in a timely fashion. Changes in billing policies (due to some upcoming holiday or special promotion) should also be distributed rapidly.

The location protocol should be independent of the client telephony application and billing model provided by the gateway. Most importantly, it should be possible for any ITG, run by any provider, to be used by any client. This provides for an open and competitive business environment.

The protocol should be simple to implement, for ITG's and clients alike. There must also be ample security mechanisms in place. Since these protocol allow for automated selection of ITG services based on some database attributes, modification of these attributes by malicious parties can have huge business repercussions.

4 Scope of the Service

In order to properly develop a protocol architecture for locating ITG's, it is necessary to determine the scope of the service - how many clients would need to be supported, how many ITG's might eventually be deployed, and what the calling patterns of clients might be These numbers are difficult to pin down, since Internet telephony is new. This makes historical data unavailable. We therefore interpolate figures from existing statistics on telephone usage.

In a fully operational system, it should be possible for every Internet host to act as a client (not simultaneously, of course). There were 16 million Internet hosts as of January 1997, with the number increasing exponentially[1]. The U.S. telephone system handles approximately 1.6 billion calls per day [23]. Of these, 84% are local and the remainder are intra-lata and inter-lata. Since there are 153 million telephone lines deployed in the U.S., [23], this averages to around 10 calls per day per person. Of these, 1.6 are inter-lata. If all IP hosts use an ITG for their inter-lata calling, this amounts to 25 million calls per day through all ITG's. Since this is still a fraction of the total telephone calls per day, we assume most IP hosts will make inter-lata calls to PSTN endpoints, and not other IP hosts.

Computing the number of ITG's which may be deployed is not easy. In order to minimize costs for its customers, an ITG provider may decide to deploy ITG's such that nearly every destination is reachable by a local call though some ITG. Current ISP's are faced with a similar problem; they must have points of presents (POPs) in each local calling area across the U.S.. As of August 1, 1997, America Online had approximately 1300 POPs in the U.S. (with 145 in California alone). To extrapolate to the number of POPs required worldwide, we multiply this figure by the ratio of worldwide to U.S. telephone access lines. There are 745 million telephone lines worldwide [24], so the number of worldwide POPS would be 6330. If we assume that there 2 to 3 major service providers in each POP, there could be as many as 18,000 ITG's worldwide. This estimate is rough; it is based on the number of current IP hosts, but assumes widespread IP telephony, which won't exist for many years. It also assumes current telephone line penetrations worldwide, which will change, and current POP densitites, which will also change.

5 Possible Solutions

Location of services on the Internet is not a new problem. DHCP has already been mentioned, for example. However, it is not sufficient for ITG location since it allows IP hosts to find a pre-configured service only on a LAN. This section analyzes several protocol architectures which could be used for location, and discusses their pros and cons.

5.1 Centralized Databases

In the centralized database approach, all of the information about all ITG's is located on a single computer. We consider two implementations of this architecture. The first, the Service Location Protocol, implements a centralized database which may be replicated across the Internet. Database entries are populated by unicast ITG registrations (push). The second, web search engines, use centralized databases as well, but they are populated by periodic polling of ITGs (pull).

5.1.1 Service Location Protocol

The service location protocol is under development in the srvloc working group of the IETF [25]. It allows for clients (called *user agents*) to find the location of a server providing a specific service in the clients administrative domain. This is accomplished in one of two ways. First, a client can broadcast a request for a service to a well-known multicast address. The request contains an expression describing the service required. Any servers matching the query respond back to the client. In recognition of the fact that this can cause significant bandwidth consumption, a second method for sevice location is available. First, the client discovers its Directory Agent (DA). Instead of multicasting its queries to a group, the client unicasts its queries to the DA. All servers register their services with the DA (via unicast), so that it may build up a service database. These registrations are periodically retransmitted to protect against network loss. The DA checks its database against the query, and returns a list of servers to the client. Of course, both servers and clients must now know the IP address of the DA. This is accomplished in several different ways. First, DA's multicast advertisements periodically to a well known address. Secondly, clients and servers can still use multicast queries to find a DA, just as they can use multicast queries to find any other service. Thirdly, clients and servers can always be preconfigured with the IP address of the DA. The protocol introduces the concept of *scope*. Each service is associated with some scope, which is just an arbitrary text string. Clients can request services that lie within a particular scope, and DA's can be configured to only accept registrations from servers that have a particular scope. A typical scope might be the string "math-department", so that clients in the math department of a university will only have access to services run by the department.

The service location protocol has many of the features required for gateway location. Clients can ask for services (including ITG's) which meet any set of criteria (protocol, cost, etc.). However, the protocol was designed for use only within an administrative domain. For use in a wide-area network, each ISP would need to administer its own DA. Servers would be located across the wide area, and register with each DA separately. This would effectively replicate the database.

This approach suffers from a number of problems when applied to wide area networks. First, Each server (ITG) must register

[1] Data obtained from Network Wizards, available at http://www.nw.com

with each DA separately. As the number of clients, servers, and DA's grows, the amount of traffic for registrations becomes excessive. For example, assume that there are about 1000 DA's (one per ISP) and 10,000 ITG's. If each server sends a 1 kByte registration to each particular DA once every three hours, the total network traffic for registrations alone is 7.4 Mb/s, and each server will send a packet almost once a second. DA's will need to process one registration per second. While this will not overload any networks or servers, it is needlessly high. Furthermore, the amount of traffic will grow linearly with the product of the number of ITG's and DA's.

Second, servers must know the IP address of all DA's. In a small administrative domain, this is easy. But in a wide-area Internet, this is more complex. The service location protocol defines two mechanisms for finding DA's. The first is to use increasing scope multicast searches. This approach can cause a lot of a traffic in a small administrative domain; it can be catastrophic on a wide area Internet. The other approach is for DA's to multicast advertisements about their existence. This approach is more reasonable on a wide area Internet. However, these advertisements are periodically retransmitted every three hours. With fixed soft-state refresh intervals, the bandwidth used grows linearly with the number of DA's present in the network [26]. Combined with the registration traffic, this causes a lot of control overhead.

Because of these problems, the Service Location Protocol has good bandwidth efficiency on a small scale, but does not scale well to the wide area Internet.

It does have advantages, however. Since database entries are populated by ITG push, the approach is sufficiently dynamic. The replication of these databases provides some security; it is difficult for a single person or organization to alter the records stored across all DA's. The Service Location Protocol also provides public key based authentication of server advertisements, making them impossible to forge. It easily supports multicriteria selection based on cost and protocol support, but proximity is not directly supported. It can be added in the same fashion as implemented in the proposed BMA architecture (see Section 7). Client complexity is very low, as is ITG complexity. Since the DA's are usually close to the clients they serve, the search times are generally very fast.

5.1.2 Web Indexing

World Wide Web search engines and indexing tools, such as Harvest [27] [28] and web bots can be applied to locating ITG's. This is accomplished by having an ITG publish information about the service provided (such as protocols and cost) on the web, and allowing various search engines to collect and catalog the information. Any of a number of standard search engines can then be used to find the desired gateway.

This solution, however, has some serious drawbacks. Most current search engines are based on keyword searches on indexed web pages, and lack the precision to locate a specific service precisely (anyone who has done a search which returned over ten thousand matches knows this problem). Even cooperative approaches (based on indexing meta information within the HTML page, for example) still require the search engines to periodically query a large number of servers. The web bots have no way of knowing when information at a server has changed. This means that data can remain out of date for long periods of time (this can be serious if this data reflects service cost), or web bots will have to increase the frequency of their queries to any particular server, increasing network traffic. This solution also places the burden of service location in the hands of a few, dedicated search engines. Adding more of them can ease the processing load, but at the expense of even more webbot traffic.

Lastly, web-based searching is designed for human interaction, and not for automated processes. They do not provide a simple mechanism by which a process can find the cheapest gateway.

5.2 Distributed Databases

The solutions in this section implement distributed databases. The information about ITG's is scattered across the network, residing in computers which are local to the ITG's whose data they contain. All of these approaches suffer from similar scalability problems due to the difficulty of organizing and searching through the distributed entries. Three protocols are analyzed: DNS, X.500, and whois++.

5.2.1 DNS

The Domain Name System [29] is used to map host names to IP addresses. It has also been used for mapping a service in a domain to a server which can provide that service [30]. Most relevant to ITG location problem is the tpc.int subdomain [31]. The tpc.int subdomain allows a host to register a fax machine as providing fax service to a certain set of telephone numbers. An IP host wishing to send a fax constructs a domain name based on the fax number, and can then find the IP address of an email to fax gateway. The construction of the domain name from telephone number is done by assigning each digit to a subdomain, in reverse order. For example, a fax gateway in the 415 area code in the U.S. (country code 1) would construct its domain name as 5.1.4.1.tpc.int. If there are multiple gateways servicing any particular telephone area, the DNS server will contain multiple records, one for each ITG.

Application of the tpc.int domain to ITG location is done trivially. Consider a new domain, called itg.int. An ITG located in the 415 area code would have a DNS name 5.1.4.1.itg.int. A host wishing to call a PSTN endpoint with a given number would first lookup the entire exchange (+1 415 822) to find a match (if there is one; some countries have flat number spaces within an area code). If there is no match, the host looks up the area code (+1 415), and then the country code (+1) if necessary. The assumption is that a gateway located in a particular area code is likely to provide the cheapest calls to that area code; generally a reasonable assumption. This approach essentially divides up ITG's hierarchically following the strict boundaries of telephone number allocations, and places them into the database based on that hierarchy.

The data contained in the resource records for each ITG can take many forms. In the simplest implementation, it can just contain the IP address of the gateway. Unfortunately, this provides no information on the protocol capabilities or cost structure provided by the ITG. Clients receiving the list of gateways would need to query each gateway separately (possibly in parallel), to determine its capabilities and cost. This approach does not scale. Consider the scenario where there is no gateway in the 609 area code in the U.S.. In order to find the next closest gateway, the client would need to look up 1.itg.int, which contains all ITG's in the U.S..

This is likely a long list (several thousand). Returning the list to the user requires a lot of network resources (both in terms of bandwidth and DNS processing time). Querying each of the thousand servers directly is most certainly inappropriate. This problem will arise when the tree of ITG's in the telephone hierarchy is "unbalanced". Since the number of telephone lines in different area codes in the U.S. is most certainly unbalanced, one would expect the distribution of ITG's to follow the same pattern.

Several enhancements to the simple implementation can be made to improve performance. First, DNS implements caching. This means that once a set of records has been transferred to a user, its local DNS server will have them for a short time. Since users generally make calls to the same telephone numbers, future ITG searches can be done straight from the cache at the local DNS server. This relieves some of the network congestion and DNS processing requirements.

It is also possible for DNS records to contain more than just the IP address of the ITG. In fact, resource records which can contain the "kitchen sink" - including MIME data, ASN.1 data, and ASCII, are being defined [32]. These records can be used to contain compact descriptions of the protocol support, capabilities, and cost structure of the ITG. As a result, when a host receives a list of resource records for ITG's in a particular exchange, country, or area code, it no longer needs to query the ITG's directly for more information. It can perform a local computation to choose the ITG which meets its needs. This helps reduce network bandwidth consumption, but also increases it because the resource records are now larger. The use of caching probably results in a net savings.

Caching and more complete resource records do not resolve the problem of "unbalanced trees" discussed above, which will still result in transfers of large numbers of resource records.. There are two fixes to this. First, the tree can be kept in an unbalanced form, but the searches can be done in a non-hierarchical fashion (i.e., first looking up the exchange, then the area code, and then the country code is hierarchical) The advantage of the hierarchical approach to searching is that it doesn't require a host to have any information about proximities between exchanges, area, and country codes. If such information does exist (by direct interaction with the user, or as part of a database packaged with the IP telephony software), arbitrary search patterns can be constructed. These patterns can search area codes around the target area code, and if no suitable gateway is found, give up. Such searches are more bandwidth- friendly, and can be done in parallel, causing only small increases in call setup times.

An alternate fix is to attempt to rebalance the tree. This can be done in two ways. The first is to allow the set of ITG's listed in various area codes to "overlap". Consider the example above, where no gateway exists in the 609 area code. If a gateway does exist in the nearby 908 area code, it can be listed as a record in the 9.0.6.1.itg.int domain in addition to the 8.0.9.1.itg.int domain. This will allow the client to find a gateway which services the 609 area code without having to query the entire U.S. The second way is to restrict (or prune) the number of ITG which may appear in any domain; 1.itg.int, for example.

These *overlap* and *restriction* policies suffer from a serious practical drawback. It is difficult to really determine the set of ITG closest to any particular country and area code. The PSTN is a fully connected network. Any ITG can complete a call to any PSTN destination. It is in the interests of an ITG provider to have its ITG's listed in as many DNS records as possible, in the hopes of increasing business. If any restrictions or overlaps are to be implemented, they will impact the business of the ITG provider. Who is to decide which ITG's are to be located in the different sections of the database?

As with any non-replicated database, DNS has security problems. One must hope that the authority maintaining the itg.int domain does not tamper with the entries. The lack of authentication in the DNS update protocol makes this concern even greater. In fact, DNS servers have recently been the subject of attacks which have attempted to change the name to IP address mappings for business gain [33].

Population of the DNS records can take place in one of two ways. ITG's can be manually entered into the database by administrators. Alternatively, protocols exist which can allow ITG's to update the DNS records themselves as they change [34][35]. This makes DNS sufficiently dynamic, as long as DNS cache entries are timed out frequently.

Finally, it is difficult for a client to choose an ITG based on proximity. The only way to locate a nearby gateway is for a client to construct the name for the country code, area code, and exchange where the client is located. The DNS lookup on this name would then yield a nearby ITG. This measure of distance is only very approximate, since the number of hops away on an IP network, or network delays, are not strongly related to geographical proximity.

In conclusion, DNS can be used quite efficiently for ITG location on a smaller scale. The various enhancements discussed above lead to a bandwidth efficient, fast, and flexible scheme based on existing standards. The latter results in ease of implementation and rapid deployment. On a larger scale, however, the number of ITG's located in various exchanges, area, and country codes may have to be limited to restrict bandwidth usage and DNS server loading. This introduces complex legal and political problems which are best avoided. On either large or small scale, DNS is not very efficient at finding the closest gateway to a host, and has security drawbacks. It is extremely simple for clients and ITG's alike, as it is based on existing and well understood standards.

5.2.2 LDAP and X.500

LDAP [36] is the Lightweight Directory Access Protocol, which is used to query databases. X.500 [37] is a large, distributed database which can be used to store information about nearly anything. LDAPv3 [38] is a new version of LDAP which adds support for improved security, extensibility, and server referrals.

The architecture of a distributed X.500 database is much like DNS. Portions are scattered over the wide area Internet. It makes sense to organize the database using the same hierarchy for assigning names in tpc.int. An ITG's distinguished name would be constructed from its country code, area code, exchange, followed by a unique identifier. In the ideal model, a client would use LDAP to generate a query for an ITG which meets its requirements, and LDAP would return with a single gateway for use.

Having an LDAP database return a single result would require the client to specify information to aid in the search, such as the destination telephone number, expected call duration, time of day, etc. Such client-side hints are not supported in LDAP. LDAP also does not currently support maximum and minimum operators on attributes, which is necessary for indicating the desire for the cheapest gateway. The current lack of support for caching, shadow and backup directories (although this is coming [39] [40] [41] [42]) are further practical problems with LDAP. LDAP is also fairly complex.

Architecturally, an LDAP database suffers the same problem as a DNS server. A client would have to perform the same kind of "longest prefix match" searches as in DNS. It would first ask for all entries which meet the protocol constraints, and lie in the subtree defined by the country code, area code, and exchange. The returned entries would include cost structure attribute-value pairs, which could be used for a local minimization search among the returned entries. If no satisfactory result is found, the search is redone at either a higher level in the hierarchy, or at a sibling subtree. This entails the same traffic and processing loads as discussed above for DNS. Unlike DNS, however, the absense of caching will amplify these problems substantially.

The whois++ protocol [43] is another database query protocol, similar to LDAP. Unlike LDAP, it does not require a hierarchically organized database. Database entries are indexed, creating *centroids* which represent the union of the values for all attributes for all entries. By exchanging centroids, database servers can determine which other servers might be a match for a particular query. This allows for more flexibility in the organization of the database servers, but at the expense of increased network traffic to process a query.

5.3 Multicast Advertisements - SAP

The Session Announcement Protocol (SAP) is used to advertise multicast sessions on the Mbone [44]. Clients of the protocol send announcements describing the Mbone session (time of ocurrence, media supported, etc.) on a well known multicast address. The transmission rate of announcements is scaled back linearly with the number of other clients sending announcements. This allows for rapid refresh of announcements when the client population is small, and bounded bandwidth utilization when it is large. This approach is also used to control RTCP packet transmission rates in RTP [45].

This basic mechanism can be extended to ITG location trivially. Instead of advertising Mbone sessions, ITG's advertise the details of the service they provide. IP hosts who wish to make a call to a PSTN destination join the multicast group and collect announcements. As the usage of the protocol increases, different multicast groups can be used for gateways located in different countries. This would decrease the time required to collect announcements from relevant gateways. In order to avoid contacting the servers directly, the advertisements should contain sufficient information so that the clients can decide on an ITG based solely on them.

The biggest drawback to SAP is that PC's may require a large amount of time to collect all of the advertisements. Most PC's are not kept on 24 hours a day. Even with caching of advertisements (an absolute must), a host may never be on during the transmission of an advertisement. This is especially true for users who turn the machines on for a specific task (such as making an IP phone call), and then turn it off. Furthermore, when a user first obtains the software to collect advertisements, it may take several hours before hearing from a gatway which meets the user's requirements. This may mean a telephony application cannot really operate properly until a few hours after boot, which is unacceptable.

Caching of advertisements also imposes significant memory requirements on the end systems. The processing requirements for searching these advertisements may also become an undue burden. These arguments become even more persuasive when one considers that future Internet telephones may not all be PC's - cheaper, simpler machines like the NetPC or even standalone IP telephones will have very limited (if any) non-volatile storage.

6 Routing

In some sense, the gateway location problem is much like a routing problem: it requires the dissemination of information about reachability (in this case, telephone number reachability), and attributes characterizing those routes. Scalability and wide area operating are a must. Existing wide area routing protocols provide this kind of function. In this section, we briefly focus on how BGP [46] might be used for telephony gateway location.

Each ITG acts as a BGP router.It can either run IBGP between itself and the regular EBGP routers for the autonomous system, or use an interior gateway protocol,such as OSPF [47] to distribute its routes. A new extension for BGP, called Multiprotocol BGP [48] allows for the EBGP router to advertise non-IP routes, such as those for a set of telephone numbers. BGP also allows routes to be propagated with attribute tags. These could be used to carry information such as supported codecs, protocols, and billing methods of the gateway.

When a client wishes to contact a gateway, it queries some router in its autonomous system. This router has built up a table of routes to various telephone numbers. It can find a match for the query, and return the resulting IP address of the gateway to the client.

The principle difficulty with this approach is that the normal BGP operations of route aggregation and route selection, done at intermediate BGP routers, cannot be performed. This is because the decision about which gateway to use needs to be based on client expressed preferences. In routing terminology, this means that routing policy is only made by clients. An intermediate router cannot choose one gateway over another when deciding which to propagate, since some client downstream may require the features supported by one and not the other. Similarly, aggregation can only be performed if all of the attributes of the gateways are identical - likely an infrequent event. These problems stem from the fact that ITG's are application level devices, terminating application level protocols. BGP was never meant to distribute application level gateway attributes, but rather network level attributes.

Without aggregation and route selection, BGP becomes merely a flooding protocol, distributing the attributes of every gateway to every other gateway and router on the Internet. In that case, it makes more sense to use a protocol optimized for such flooding operations, instead of burdening an already overloaded and stressed mechanism.

7 Proposed Solution: BMA - Brokered Multicast Advertisements

We propose here a new protocol architecure which is a hybrid of some of the ones discussed above. It combines the scalable advertisement mechanisms of SAP, the powerful client server query protocols of LDAP, X.500, and the Service Location Protocol, and the hierarchical naming space for ITG's in DNS. We call our architecture Brokered Multicast Advertisements (BMA).

The system is depicted in Figure 2. It is composed of a number of components. The *client* is an IP host who wishes to make an IP to PSTN telephone call. A *broker* is similar to a DA in the Service Location Protocol. It has access to a large database of ITG's. A client wishing to find an ITG meeting some criteria unicasts a query to the broker. The broker searches its database, finds one or more gateways meeting the criteria, and unicasts the result back to the client. Brokers are replicated across the Internet. Like a DNS server, one would generally be provided by each ISP (although this is not required). A client would know its broker through DHCP or static configuration. ITG's are scattered across the wide-area network. Using methods similar to those in SAP, they multicast advertisements about themselves to one of several well known multicast groups. Brokers join these multicast groups, and collect announcements, storing them in the local database.

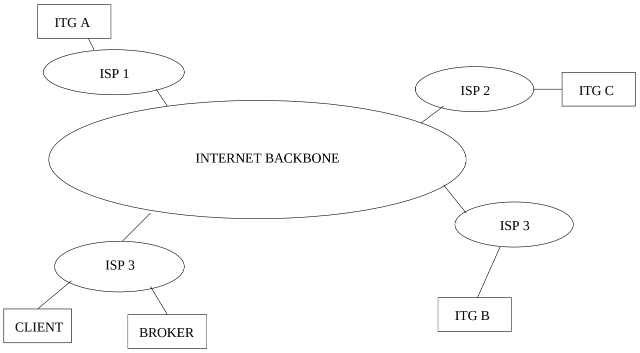

Figure 2: BMA Architecture

By using scalable wide area multicast for distributing ITG advertisements, both deficiencies of the Service Location Protocol are eliminated. ITG's do not need to know the IP addresses of brokers, nor do they need to generate multiple advertisements, one for each broker. By using a broker, clients are relieved from the burden of collection, storage and processing of advertisements, as in X.500 and DNS. They also no longer require multicast capabilities, as in SAP. By replicating brokers at each ISP, lookup delays are minimized. Additional brokers can be added if processing overhead at a broker becomes unacceptable. Any backend for storage and accessing of databases used by the broker can be used, as they are transparent to the protocol. This means that existing database tools, such as LDAP and X.500, can be used to store advertisements.

The architecture is highly scalable. Wide area network traffic is restricted to multicast where the bandwidth is tightly controlled. Client-broker interactions are unicast, but are localized. Even though the scalable multicast advertisements restrict bandwidth usage, they allow for rapid updates of advertisements which have changed. This makes the architecture more dynamic than any of the others discussed so far. Both client and ITG implementations are simple, at the expense of moderate complexity at brokers.

The architecture is also very general. Any client can use any ITG, as long as the broker stores the advertisement from that ITG. This also allows administrators of brokers to implement local *policy*. This facet of the architecture is discussed in Section 7.2.

In fact, the BMA architecture is well suited to location of any service offered on the wide area network. Brokers can be established for different services, and more multicast addresses assigned. Non-network ervices like hotels, movie theaters, restaurants can all be located, in addition to more traditional network services, like ITG's, media bridges, and media servers.

The following sections detail the functionalities of the brokers and ITG's in more detail.

7.1 ITG Behavior

ITG's advertise their attributes using multicast.

A set of multicast addresses are defined. Each ITG takes the country code it is located in, and hashes it to one of them, on which it will advertise. By using separate multicast groups for different country codes, a broker can quickly learn about a specific set of gateways. It can also, based on policy, ignore certain multicast groups if it likes.

In order to determine the frequency of advertisement transmissions to the group, each ITG also joins the multicast group it sends on. Each ITG keeps track of the IP addresses of the other ITG's who have sent advertisements to the group. Based on this, it can obtain a group size estimate, L. Each ITG also stores the size (in bits), of its advertisements, S. To maintain the rate of advertisements to a group at R packets per second, each ITG sends an advertisement with a nominal period of LS/R. Bandwidth usage can be further reduced by increasing this nominal period for advertisements which have already been transmitted before. Randomization of the period is also required to avoid synchronization effects [49].

This approach is a variation on the control mechanisms used in RTP [45]. It maintains the cap at R bits per second, but allows lower transmission rates when the advertisements contain no new data.

As a further enhancement to improve scalability, timer reconsideration should be used [50]. This algorithm requires end-systems to recheck the validity of their event timers before sending a packet. The validity is based on whether the perceived multicast group size (L) has changed since the timer was set. If the perceived group size has grown, packet transmission is delayed in accordance with the most recent group size estimate. This approach will help reduce packet transmissions from servers which boot up, join their multicast group, and initially see only one server: themselves.

7.2 Brokers

A broker is a device which joins the multicast groups used by ITG's to advertise their service. As advertisements are received, they are placed in a local database. The broker also accepts *queries* from clients which specify the desired features for an ITG. This query is applied to the database, and the results are placed in a *response*, which is sent back to the client.

There are several options for the query-response protocol between a client and the broker. LDAP can be used (with alterations in syntax and semantics), as can the query-response protocol used in the Service Location Protocol.

An important aspect of broker behavior is policy. Based on any criteria, an administrator can program a broker to drop advertisements from certain ITG's, based on the values of any attributes in the advertisements. Some possible policies include: (1) Dropping advertisements from ITG's run by competitors, (2) Dropping advertisements from ITG's which do not contain satisfactory authentication, (3) Dropping advertisements from ITG's whose administrators "misbehave", by advertising false information, violating the scalable multicast rules, etc., or (4) Dropping advertisements from ITG's which do not use protocols mandated by the administrator of the broker.

This capability for policy is absent when a single database (distributed or centralized) such as X.500 or DNS, is used for storage. It can be implemented in the replicated database architectures, such as SAP and the Service Location Protocol. The use of brokers and policy creates the opportunity for a new business: brokering. Since clients can choose brokers by static configuration, they can choose a broker based on an ad on a web page, for example. Brokers can attract business by offering the largest databases, best authentication, etc.

The main limitation to scalability for the brokers is the processing burden to search a large number of records. If the number of servers for a particular service begins to exceed several tens of thousands, the storage requirements for them, and the time for even a single search of the database for a match, can become excessive. In this scenario, brokers always have the option of implementing policy to restrict the size of the database. As faster machines and bigger disks become available, these policies can be lifted. Since the protocol does not mandate the database structure used to satisfy queries, local instances of distributed database and search engines can be used to speed up accesses.

8 Conclusion

We presented the problem of Internet Telephony Gateway (ITG) location, and showed that solving this problem is important for the success of Internet telephony as a service. We show that it is important for IP hosts to find the IP address of an ITG which meets a wide range of constraints, including cost for completing a call to a specific destination on the PSTN, proximity to the client, and protocol support. We analyzed a number of existing resource discovery protocols, including DNS, X.500, the Service Location Protocol, whois++, web search engines, SAP, and BGP. We also introduced a new protocol architecture which we call Brokered Multicast Advertising (BMA). It is highly scalable, bandwidth efficient, and simple. We conclude that BMA is the most effective solution for gateway location, but that DNS can be made to work quite well for smaller number of ITG's.

The BMA architecture is being proposed as an extension to the current Service Location Protocol [51]. Although designed for ITG location, BMA is well suited to location of a wide range of services, of which ITG's are just an example.

9 Acknowledgements

The authors would like to Santosh Krishnan, T.V. Lakshman and Dheeraj Sanghi for their thoughts and insights on various aspects of this problem. A special thanks go to Bernhard Suter, who contributed many of the ideas in this paper.

References

[1] V. Jacobson, "vat - lbnl audio conferencing tool," Tech. Rep., LBNL, Available at http://www-nrg.ee.lbl.gov/vat.

[2] H. Schulzrinne, "Nevot: Network voice terminal," Tech. Rep., Available at ftp://gaia.cs.umass.edu/pub/hgschulz/nevot.

[3] C. Perkins, "Rat: The robust audio tool," Tech. Rep., UCL, Available at http://www-mice.cs.ucl.ac.uk/mice/rat.

[4] Henning Schulzrinne, "Re-engineering the telephone system," in *Proc. of IEEE Singapore International Conference on Networks (SICON)*, Singapore, Apr. 1997.

[5] J.C. Bolot, "End-to-end packet delay and loss behavior in the internet," in *SIGCOMM Symposium on Communications Architectures and Protocols*, San Francisco, CA, Sept. 1993, pp. 289–298.

[6] N. Macemchuk and S. Lo, "Measurement and interpretation of voice traffic on the internet," in *Conference Record of the International Conference on Communications (ICC)*, Montreal, Canada, June 1997.

[7] Colin Perkins and Jon Crowcroft, "Real time audio and video transmission of IEEE GLOBECOM'96 over the Internet," Technical report, University College London, London, England, Nov. 1996.

[8] Jean-Chrysostome Bolot and Andres Vega Garcia, "Control mechanisms for packet audio in the internet," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, San Fransisco, California, Mar. 1996.

[9] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, Apr. 1997.

[10] W. Montgomery, "Techniques for packet voice synchronization," *Journal of Selected Areas in Communications*, vol. SAC-6, no. 1, Dec. 1983.

[11] Ramachandran Ramjee, Jim Kurose, Don Towsley, and Henning Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Toronto, Canada, June 1994, pp. 680–688, IEEE Computer Society Press, Los Alamitos, California.

[12] R. Braden, L. Zhang, and S. Berson, "Resource reservation protocol (RSVP) – version 1 functional specification," Internet Draft, Internet Engineering Task Force, Nov. 1995, Work in progress.

[13] P. Pan and H. Schulzrinne, "A simple reservation mechanism for the internet," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, 1998, submitted.

[14] R. Droms, "Dynamic host configuration protocol," Request for Comments (Proposed Standard) 1541, Internet Engineering Task Force, Oct. 1993, (Obsoletes RFC1531); (Obsoleted by RFC2131).

[15] Stefan Brands, "Electronic cash on the Internet," in *Proc. of the Internet Society 1995 Symposium on Network and Distributed System Security*, San Diego, California, Feb. 1995.

[16] ITU-T, *Recommendation G.711: Pulse Code Modulation (PCM) of Voice Frequencies*, 1988.

[17] ITU-T, *Recommendation G.729: Coding of Speech at 8 kbit/s using Conjugate Structure Algebraic Code Excited Linear Prediction (CS-ACELP)*, Mar. 1996.

[18] ITU-T, *Recommendation G.723.1: Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbits/s*, Mar. 1996.

[19] ITU-T, *Recommendation G.728: Coding of Speech at 16 kbit/s Using Low Delay Code Excited Linear Prediction*, Sept. 1992.

[20] ITU-T, *Recommendation G.722: 7 kHz audio-coding within 64 kbit/s*, 1988.

[21] ITU-T, *Recommendation H.323 - Visual Telephone Systems and Equipment for Local Area Networks which Provide Non-Guaranteed Quality of Service*, February 1996.

[22] Mark Handley, Henning Schulzrinne, and Eve Schooler, "SIP: Session initiation protocol," Internet Draft, Internet Engineering Task Force, Dec. 1996, Work in progress.

[23] "Preliminary statistics of communications common carriers," Tech. Rep., Federal Communications Commission, 1996.

[24] "World telecommunications development report," Tech. Rep., ITU, 1996.

[25] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, "Service location protocol," Request for Comments (Proposed Standard) 2165, Internet Engineering Task Force, June 1997.

[26] Puneet Sharma, Deborah Estrin, Sally Floyd, and Van Jacobson, "Scalable timers for soft state protocols," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Kobe, Japan, Apr. 1997.

[27] C.M. Bowman, P. Danzig, D. Hardy, U. Manber, M. Schwartz, and D. Wessels, "Harvest: A scalable, customizable discovery and access system," Tech. Rep. CU-CS-732-94, Dept. of Computer Science, U. Colorado, Mar. 1995.

[28] C.M. Bowman, P. Danzig, D. Hardy, U. Manber, and M. Schwartz, "The harvest information discovery and access system," *Computer Networks and ISDN Systems*, , no. 28, pp. 119–125, 1995.

[29] P. Mockapetris, "Domain names - concepts and facilities," Request for Comments (Standard) STD 13, 1034, Internet Engineering Task Force, Nov. 1987, (Obsoletes RFC882); (Obsoletes RFC883); (Obsoletes RFC973); (Obsoleted by RFC2065); (Updated by RFC1101); (Updated by RFC1876); (Updated by RFC1982); (Updated by RFC2181).

[30] A. Gulbrandsen and P. Vixie, "A DNS RR for specifying the location of services (DNS SRV)," Request for Comments (Experimental) 2052, Internet Engineering Task Force, Oct. 1996.

[31] C. Malamud and M. Rose, "Principles of operation for the TPC.INT subdomain: General principles and policy," Request for Comments (Informational) 1530, Internet Engineering Task Force, Oct. 1993.

[32] D. Eastlake 3d, "The kitchen sink resource record," (internet draft), Internet Engineering Task Force, Apr. 1997, Work in Progress.

[33] Peter Wayner, "Private domain register upsets internic again," *New York Times Cybertimes*, July 1997.

[34] J. Bound, Y. Rekhter, S. Thomson, and P. Vixie, "Dynamic updates in the domain name system (DNS UPDATE)," Request for Comments (Proposed Standard) 2136, Internet Engineering Task Force, Apr. 1997, (Obsoletes RFC1035).

[35] D. Eastlake, "Secure domain name system dynamic update," Request for Comments (Proposed Standard) 2137, Internet Engineering Task Force, Apr. 1997, (Obsoletes RFC1035).

[36] W. Yeong, T. Howes, and S. Kille, "Lightweight directory access protocol," Request for Comments (Draft Standard) 1777, Internet Engineering Task Force, Mar. 1995, (Obsoletes RFC1487).

[37] ITU-T, *Recommendation X.500: The Directory: Overview of concepts, models, and services*, Nov. 1993.

[38] M. Wahl, T. Howes, and S. Kille, "Lightweight directory access protocol (v3)," (internet draft), Internet Engineering Task Force, July 1997, Work In Progress.

[39] R. Weiser and E. Stokes, "Ldap replication requirements," (internet draft), Internet Engineering Task Force, July 1997, Work in Progress.

[40] S. Jain, U. Srinivasan, and G. Good, "Schema for replication information," (internet draft), Internet Engineering Task Force, July 1997, Work in Progress.

[41] C. Weider and J. Strassner, "Multi-master replication protocol," (internet draft), Internet Engineering Task Force, July 1997, Work in Progress.

[42] G. Good, "Definition of an object class to hold ldap change records," (internet draft), Internet Engineering Task Force, July 1997, Work in Progress.

[43] L. Daigle P. Faltstrom, S. Newell, "Architecture of the whois++ service," (internet draft), Internet Engineering Task Force, Mar. 1997, Work in Progress.

[44] M. Handley, "Sap - session announcement protocol," (internet draft), Internet Engineering Task Force, Nov. 1996, Work in Progress.

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

[46] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," Request for Comments (Draft Standard) 1771, Internet Engineering Task Force, Mar. 1995, (Obsoletes RFC1654).

[47] J. Moy, "OSPF version 2," Request for Comments (Draft Standard) 2178, Internet Engineering Task Force, July 1997, (Obsoletes RFC1583).

[48] D. Katz, Y. Rekhter, T. Bates, and R. Chandra, "Multiprotocol extensions for BGP-4," Internet Draft, Internet Engineering Task Force, Sept. 1997, Work in progress.

[49] Sally Floyd and Van Jacobson, "The synchronization of periodic routing messages," *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 122–136, Apr. 1994.

[50] Jonathan Rosenberg and Henning Schulzrinne, "Timer reconsideration for enhanced RTP scalability," Internet Draft, Internet Engineering Task Force, July 1997, Work in progress.

[51] Jonathan Rosenberg, Bernd Suter, and Henning Schulzrinne, "Wide area network service location," Internet Draft, Internet Engineering Task Force, July 1997, Work in progress.