

# The TOPS Architecture for Signaling, Directory Services and Transport for Packet Telephony

*N. Anerousis, R. Gopalakrishnan, C.R. Kalmanek, A.E. Kaplan, W.T. Marshall,  
P.P. Mishra, P.Z. Onufryk, K.K. Ramakrishnan, C.J. Sreenan*

AT&T Labs - Research  
Florham Park, NJ 07932, USA

## Abstract

*Packet telephony is of increasing interest in both the telecommunications and Internet communities. The emergence of packet telephony will create new services, and presents an opportunity to rethink how conventional telephony services are implemented. We present an architecture for Telephony Over Packet networks (TOPS). TOPS allows users to move between terminals or to use mobile terminals and be reachable by the same name. TOPS users can have multiple terminals and control how calls are routed to them. TOPS allows terminals with a range of capabilities such as video, whiteboard and allows a variety of coding formats. TOPS retains the necessary information on terminal capabilities to establish appropriate communication with the remote terminal. The architecture assumes that the underlying network supports end-to-end connectivity between terminals, with an appropriate quality of service.*

*The components of TOPS are a directory service, an application layer signaling protocol and a logical channel abstraction for communication between end-systems. The directory service maps a user's name to a set of terminals where he may be reached. A user can control the translation operation by specifying profiles that customize how his name is mapped to this set of terminals. Terminal capabilities are also stored in the directory service. The application layer signaling protocol establishes and maintains call state between communicating terminals. The logical channel abstraction provides a shared end-to-end context for a call's constituent media and control streams, while isolating the applications from the details of the network transport mechanisms. We also introduce a simple encapsulation format for voice.*

## 1. Introduction

The interest in packet telephony stems from the desire to integrate multiple modes of communication on a single infrastructure. Packet networks allow the integration of data and voice and provide a straightforward way to offer a va-

riety of new communication services. Increased intelligence and more elaborate user interfaces at the terminals<sup>1</sup> allow increased flexibility in handling voice and other forms of communication. In addition, the combination of packet networks and enhanced terminal capabilities provide an opportunity to reconsider both the nature and the implementation of telephony services. Over time, we believe these services will evolve to support substantially enhanced human communication, with voice continuing to be an important mode of communication.

We present an architecture for Telephony over Packet Networks (TOPS) that exploits the flexibility provided by packet switching and by greater intelligence in end-systems. The key elements of the TOPS architecture are:

- A *directory service* to flexibly map user names to terminals where they can be reached.
- An *application layer signaling* protocol to allow end-systems to negotiate capabilities, establish and maintain call state, and support advanced telephony features.
- A *logical channel* abstraction and an efficient encapsulation format to insulate telephony applications from the underlying network transport capabilities.

A goal of the TOPS directory service is to make it more convenient to reach a user. A packet telephony service requires a directory service function that can translate between telephone numbers and network layer addresses, such as IP or ATM addresses. This simple functionality is enhanced in TOPS to allow callers to reach a person by *name* rather than address. Information is stored in the directory service about user names and the set of terminals where they can be reached. Callers obtain terminal addresses of users by issuing a name resolution query to the directory service. As users move between terminals, they can securely modify their record entries to reflect the locations where they can be reached. Thus, user mobility is a fundamental capability of the TOPS architecture. Since the di-

---

1. Throughout the paper, we use the term end-system, terminal and packet telephone interchangeably.

rectory service returns information about individual users, it provides a mechanism for them to control how a name resolution query is handled. Users can customize access to their information. The directory service also helps callers by providing them with a flexible set of call appearances and options. Compared to existing telephony features such as “personal phone numbers” and “intelligent 800-number” services, this approach is more flexible, and gives the user more control. TOPS terminals range from computers running telephony applications to low-cost packet telephony appliances. Terminal capabilities are stored in the directory and returned to the caller as part of name resolution so that appropriate communication resources can be setup. Fast terminal mobility is handled by interfacing the directory service to a *terminal tracking server* (TTS) that deals with mobility, thus isolating the user from its complexity.

Traditional telephone signaling was designed to support relatively simple terminals. The functions of routing, number translation, connection establishment, call control, terminal control, usage recording, and advanced feature support are all implemented within the network and are tightly intertwined. As a result, the complexity of the signaling software, the reliability requirements of this integrated system, and interactions among features make it difficult to deploy new services. In contrast, packet networks already provide connectivity efficiently while services are implemented at the application layer. We therefore view the remaining pieces of signaling required to establish a call as application-layer functions. An *application-layer signaling* (ALS) protocol sets up and manages the necessary state at the end-systems and supports the negotiation of terminal capabilities and media types. ALS has the advantage that it can evolve to support a wide-range of functions, including multiple media, independent of the underlying network architecture. The capability negotiation function included in ALS allows for flexibility: there is no restriction to use only predefined media channels and encodings. The ALS protocol is also used to transfer and tear-down calls, dynamically add or remove media channels and perform conferencing control functions.

The third element of the TOPS architecture is a *logical channel* (LC) abstraction and an efficient *encapsulation* format to insulate telephony applications from the underlying network transport capabilities. ALS uses logical channels as pipes for communicating both control and media data, so as to abstract the details of the underlying transport. LCs can be used to multiplex control and media channels onto a single shared context at the transport layer. The TOPS encapsulation format has been designed to contain only the essential overhead needed for transport of audio data, while allowing for synchronization among multiple media streams. This format avoids some of the fields, such as timestamps, that are typically included in a more general

real-time transport protocol such as the Internet Real-Time Protocol (RTP) [14]. However, header extensions are used to carry additional information if required.

Packet phones in TOPS use a gateway to interoperate with the public switched telephone network (PSTN). The gateway acts as an end-system for the ALS protocol on the packet network and translates signaling messages between the packet network and the PSTN. It optionally performs media transcoding.

There are a number of related efforts to design pieces of a packet telephony service. The IETF Session Initiation Protocol (SIP) [15,16] combines user location and call establishment in a single protocol. In contrast, with TOPS a directory provides a caller with options for reaching a user and the services supported prior to call establishment. This allows for evolution of the functions of user location and call establishment somewhat more independently. The ITU H.323 protocol suite [7] defines a family of protocols for multimedia communications which, unlike TOPS, emphasizes use of an intermediate entity for signaling between terminals, has considerable protocol overhead when used for ordinary telephony, and provides limited address translation facilities. The Call Management Agent (CMA) [2] of the Voice-over-IP consortium has aims similar to those of the TOPS directory server, but does not address user or terminal mobility, or signaling issues.

Figure 1 illustrates the various components of the TOPS architecture and their relationship. Users register their location with the directory server. To place a call, a user contacts the directory server to determine the locations where the called party may be reached. The directory server may consult with a terminal tracking server to reach a mobile terminal. Data and application layer signaling are exchanged directly between end-systems.

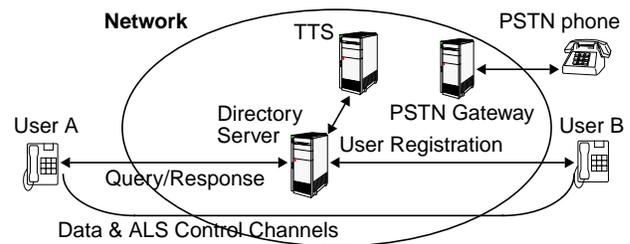


Figure 1: TOPS Components

This paper is organized as follows. Section 2 discusses the directory service. Section 3 focuses on application-layer signaling. Section 4 describes our approach to end-to-end transport of control and real-time data. Section 5 discusses interworking with the PSTN. Section 6 describes telephony services in TOPS. Section 7 presents our implementation efforts. Section 8 concludes.

## 2. TOPS Directory Service

Network directory services conveniently insulate users from dealing with network addresses. The Domain Name System [4] is an example of a directory service that translates host names to Internet addresses. The PSTN also uses directory services to support features such as 800 numbers, which allow a dialed number to be mapped to different telephone numbers based on certain rules or preferences. However, in order to reach the vast majority of PSTN subscribers, a caller needs to know the network address (telephone number) of the terminal nearest to the subscriber, e.g., his office phone, home phone, car phone, etc.

The goal of the TOPS directory service is to provide a dial-by-name capability: callers use a *distinguishing name* (DN) that uniquely identifies the person they wish to reach. The directory maps a query for a distinguishing name to one or more *call appearances*, each of which represents a terminal or other ways of “reaching” a user, such as a voice mail server or secretary. In this way the directory can support both user mobility (users can be reached at any terminal by updating their directory entry with the address of their present location) and terminal mobility (a user can receive calls at a mobile terminal even if its network address changes as it roams). In the PSTN, the “single number reach” capability is achieved by a network switch attempting to set up the call based on the translation of the name to a unique telephone number. Our approach allows the end-systems to use the information provided by the directory service directly, thus exploiting the enhanced capabilities of these terminals. The directory service also allows for the incorporation of query handling profiles that allow users to control which call appearances are returned by name resolution queries and when they are returned.

The directory is a distributed database. A hierarchical name space is used to derive the distinguishing names of users. TOPS allows multiple hierarchical name spaces to be defined over the database. In this way, a user record can be accessed through an X.500 distinguishing name (e.g., CN=Graham Bell, O=ATT, C=US) [19], an e-mail address (e.g., graham.bell@att.com) or a regular telephone number. The directory optionally supports traditional database functions. For example, it may be possible to look up users by last name, address, etc.

The naming hierarchy typically corresponds to administrative domains. Different network operators or large businesses own portions of the name space and operate their own servers for their part of the name space. The master copy of a user’s record is always stored in one authoritative server (the user’s home directory). Terminals access the directory service through a local directory server. (This server is usually the one closest to the location of the terminal and is administered by the network operator responsible for the domain to which the terminal is currently attached.) An

inter-directory server protocol is responsible for locating the home directory of the called user and fetching the appropriate record. TOPS does not allow user records to be cached by local servers; only the address of the home directory server can be cached. This ensures that the user’s call appearance information is always up-to-date.

### 2.1 Directory Contents

Figure 2 illustrates the contents of a user record in the directory database. Each record is associated with one or more DNs. The record contains information about the user (e.g. full name, address, etc.), a number of *query-handling profiles* (QHP), one or more sets of call appearances where the user can be reached, and credentials. The credentials are used by the directory to authorize a modification of the user record and by external servers to authenticate access to network services.

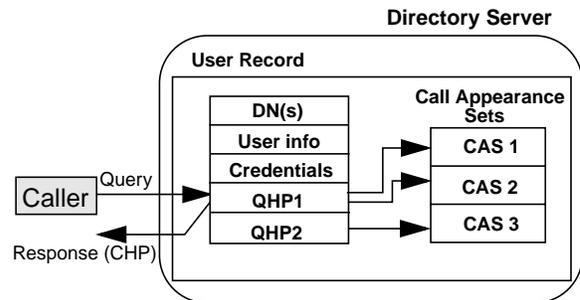


Figure 2: Structure of a user record

The query handling profile (QHP) determines the response provided to queries about this user. The QHP allows users to specify who can reach them, and where and when they can be reached. The QHP consists of a sequentially executed table of matching rules. Every rule is associated with a *call appearance set* (CAS), a group of call appearances that are returned to the querying terminal, representing the different ways in which the user can be reached. There is always a default CAS which is returned in case the QHP cannot match the incoming query to a particular rule. A call appearance set consists of a number of call appearances (see Table 1). Call appearances contain a set of fields:

- The *type* field identifies the type of the call appearance. In most cases, the type refers to a terminal or server, such as a voice mail server. In addition, a call appearance may be a pointer to another user record (DN) or indicate that the call appearance exists within another network domain, in which case a gateway must be employed.
- The *identifier* field contains optional information that can be used to identify the call appearance among others of the same type, if necessary. For example, call appearances for terminal devices may contain an electronic serial number (ESN). The ESN may be used to

**Table 1: Example of a call appearance set**

Type	Identifier	NLA	TLA	Terminal Capabilities	Application Info	Priority	Timeout	Comment
Terminal	ESN1	NLA1	TLA1	audio G.711	none	1	30 sec	work phone
Terminal	ESN2	NLA2	TLA2	audio G.711 video H.261	none	2	20 sec	home phone
TTS	TTS_ID	NLA3	TLA3	audio	ESN3	2	30 sec	mobile phone
DN	my_secy@ att.com					3		secretary
PSTN				audio G.711	(973) -555-1212	4	15 sec	parent's POTS phone
Server	VM_ID	NLA4	TLA4	audio	VM box #	5	10 sec	voice mail

support terminal mobility, in order to update the relevant call appearances when a terminal changes its network address.

- The *NLA/TLA* fields contain the network layer address of a terminal device or server, and the transport layer address of the TOPS application. The format of the addresses stored in these fields may differ depending on the network to which this terminal is attached. For telephony gateways, the *NLA/TLA* of the gateway may be statically assigned, or may be dynamically generated by the local directory server using a specialized gateway location mechanism since the address of the gateway to be used may depend on the location of the caller (or callee). The *NLA/TLA* fields are omitted when the call appearance contains another DN.
- The *terminal capabilities* field describes the media processing capabilities of the terminal (e.g., video, voice or fax), or the preferences of the user for a particular media type.
- The *application information* field is used to convey additional information to the calling application. For example, for a call from a packet telephone to a call appearance on the PSTN, the phone number associated with the DN is stored in the directory server, and passed to the gateway by the calling application.
- The *priority* field indicates the order in which call appearances should be attempted from the caller's terminal. Entries with the same priority are attempted at the same time (the equivalent of having several phones ring at once).
- The *timeout* field indicates the amount of time before trying the next call appearance if there is no answer. If there are multiple timeout values in a given priority the longest one dominates.
- A *comment* describes the nature and function of the call appearance.

Table 1 above illustrates a call appearance set for a user who has indicated as his first preference his office. The caller tries this terminal first, and if no reply is received within

30 seconds, tries simultaneously the next two call appearances: the terminal at the user's home and a cellular phone (accessed indirectly through a terminal tracking server (TTS) - see Section 2.3). If these call appearances fail to provide a response, the terminal attempts to contact the person's secretary by initiating a second directory query. If this fails as well, the terminal attempts to reach a PSTN phone through a gateway. If everything else fails, the caller is connected to the callee's voice mail server.

## 2.2 Client Interactions with the Directory Service

### 2.2.1 Service Subscription

Directory records are created when a user subscribes for service. At that time, he supplies the full name, billing address, and service preferences to a service representative, who assigns a DN. To maintain compatibility with domains that use numeric addressing, such as the PSTN, a unique telephone number may be assigned as an alternative DN. Cryptographic credentials for user authentication and registration are also created at this time.

A directory entry is required only for users who wish to receive calls. An "anonymous" user who does not have a directory entry with stored credentials can also place calls as long as he has provided a means of being charged for the service (e.g., credit card).

### 2.2.2 User Registration and User Mobility

In order for a user to receive calls, one or more call appearances must be entered into the directory record. Call appearances can be configured in three different ways:

- Statically, when the service representative creates the record,
- Off-line, when the service subscriber uses directory management software to configure his record from, say, a computer,
- Dynamically, when the user registers at a new terminal to receive calls, or moves a previously registered terminal. The user may register at a terminal that retains his information (e.g., his home phone) or to a

terminal not associated with a user (e.g., a public phone).

In the last two cases, the directory must authenticate the user by using the credential information stored in the user's record. Once authenticated, the user can update or delete existing call appearances, add new ones, or update his QHP to implement a new query resolution policy.

When a directory entry is configured off-line or dynamically, a challenge-response protocol may be used to authenticate the user. The terminal device may not have a keyboard, in which case the user may employ a smart card where the necessary credentials are stored. When the smart card is used at a public terminal, the directory can be updated automatically with a new call appearance entry for this terminal. Similarly, when the user's terminal moves to a new location, it may have been authorized beforehand to update the call appearance record in the directory with its new NLA/TLA for that user (possibly with a password). The user has the flexibility to define the desired level of security for updating the directory record. In addition, the introduction of a new call appearance in the user's record or the modification of an existing one to reflect the new address of a terminal can be specified to be valid only for a specific period of time. For example, user registration at a public phone need only be valid while the smart card is in place. Explicit user deregistration triggers updates of the relevant call appearance records. Deregistration from a public terminal triggers a deletion of the corresponding call appearance.

### 2.2.3 Querying the Directory

In order to call another user, the caller queries the directory to obtain the call appearance(s) of the person he wishes to call. The only information required in a query is a DN of the called party. Callers may also supply their own DNs, the types of media to be included in the call, the capabilities of the calling terminal and an indication of urgency.

The directory also supports the concept of a user persona. In this case, the caller supplies additional information that identifies the role in which a user is being called, for example professional, social, emergency, etc. Every user persona can be handled by a different QHP. The QHP contains the logic for handling a query related to the user/persona being called. The QHP may produce different responses indicating where a user can be reached based on who sent the query, the time-of-day, and the purpose of call. In addition to providing customizability, this allows the user greater control over the privacy of information compared to a traditional directory service.

The directory service does a small amount of processing to ensure that the caller's and callee's terminals are compatible. This optimization minimizes the amount of end-to-end

capabilities negotiation during call establishment for some forms of communication, such as simple telephony.

### 2.2.4 Query Response

The response to a query is a call handling profile (CHP). A CHP is a subset of a call appearance set stored in the user's record and is determined after executing a QHP. Call appearances in the CHP are derived from call appearance records in the directory by adding or filtering out some fields.

The terminal capabilities returned with the CHP may not be the full set supported. This allows a callee to limit the terminal capabilities depending on the caller, the time of day, etc. For example, a user may restrict incoming calls from certain people to be audio only.

Fields can be added to a call appearance to implement interesting services. For example, the directory could insert an estimate of the per-minute cost for communicating with a particular call appearance. The caller could then use this information to restrict "expensive" call appearances or keep track of the accumulated cost of a call.

When the calling application receives this information, it may use the CHP directly, or it might display the CHP and await further instructions. For example, the caller may want to skip call appearances that do not meet his needs (e.g., a voice mailbox) or are expensive.

## 2.3 Support for Terminal Mobility

A directory record is updated either when a user moves between terminals or a terminal changes its point of attachment to the network and gets a new network-layer address (NLA). This works well if a user is registered at a terminal that does not move frequently; i.e., on the time scale of seconds or minutes. However, several performance problems occur for faster moving terminals. First, directory servers are usually optimized to support rapid lookups but relatively slow updates. Secondly, it may be inefficient to update the home server every time a terminal changes its NLA, because the directory server may be far from the current terminal location.

To avoid having to update the home directory service each time the terminal changes its NLA, we introduce an additional component in our architecture: a "terminal tracking server" (TTS). A mobile terminal acquires an NLA and the address of the nearest TTS from the network via auto-configuration. It then registers its electronic serial number and current NLA with the TTS. When a user registers at a terminal, the user record in the home directory is updated with a call appearance containing the network layer address of the TTS, rather than the address of the terminal. In response to a query, the home directory server contacts the TTS to obtain the current NLA of the terminal and returns this value to the caller.

As long as a terminal moves within the domain of a particular TTS, there is no need to update any entry at the directory server. When a terminal moves to a new TTS, it is necessary to:

1. update the entry at the previous TTS to point to the new TTS
2. update the directory server entry (for each user registered at the terminal) to point to the new TTS.

Support for terminal mobility also requires the underlying network layer to reroute ongoing calls as terminals move. We assume this is supported at the network or link-layer, for example using techniques such as ATM VC rerouting or mobile IP [10].

### 3. Application-layer signaling

Signaling in the telephone network supports multiple functions, including routing, number translation, connection establishment, call control, terminal control, usage recording and advanced feature support. In packet networks, routing, connectivity and resource management are already supported by existing network layer mechanisms. Moreover, intelligent end-systems reduce the need for network entities to be involved in terminal control. In our view, telephony applications communicate with each other directly, using an application-layer signaling protocol for control, and media streams for data. In this section, we describe the application-layer signaling (ALS) protocol that is used in the TOPS architecture.

The primary functions supported by ALS are establishing and maintaining call state between applications. ALS is used at an end-system to setup a call, establish and manage media channels, negotiate parameters for these channels, and to implement additional service features. The protocol has a small and simple set of basic messages augmented by optional messages through the use of header extensions. ALS supports call management for a variety of call types, ranging from simple, point-to-point voice calls, to multi-party, multimedia calls, to calls that involve other application services, such as a game or storage server. We now describe use of ALS in point-to-point telephony and multimedia calls.

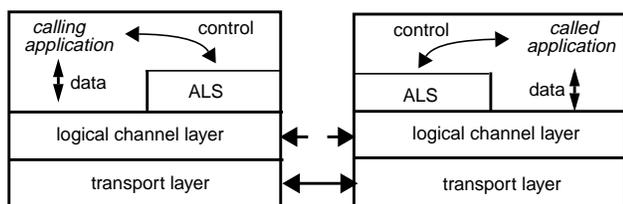


Figure 3: ALS protocol layering

All communication between the calling and called parties takes place over logical channels (LC) that are used to dis-

tinguish information flows within a call. Each call has a default LC for exchanging ALS messages, and one or more LCs for media data. Figure 3 illustrates the protocol relationship between ALS and the logical channel layer; details of logical channel operation are provided in Section 4. LCs may share a transport context while providing different semantics to the constituent flows. For example, the ALS protocol may use an LC that provides reliable and ordered message delivery, whereas an audio stream requires an LC that transports real-time data while meeting the performance constraints of that audio stream.

ALS is invoked after the calling terminal has queried the directory service to obtain one or more call appearances for the called party. The sequence of message exchanges are shown in Figure 4. An INVITE message is sent to the default LC at the address returned in the call appearance of the directory response. The message contains the distinguishing name of the called party, and optional information identifying the calling party, subject and urgency. A key function of the invitation message is capability negotiation which is used by the two parties to agree upon the number of media channels for media data, as well as their attributes such as media type, framing, encoding, priority and transport related parameters. The ALS invitation contains the set of logical channels and their attributes desired by the caller. These values are derived by taking into account the called-party capability information provided in the directory reply.

The called party application replies with a RESPONSE message that indicates which of the desired media channels are available, as well as additional information that the called party application wishes to provide to the calling party application. If no channels are available, the ALS exchange terminates. If the response indicates that a set of channels that the calling party considers necessary are available, the calling party application then obtains the network resources required to support these channels. If it succeeds, it sends a RING message to the called party; otherwise the ALS protocol terminates due to lack of resources. The RING message indicates the set of media channels that were allocated. When RING is received, the called user is alerted by a local mechanism. When the user accepts the call, an ANSWER message is sent to the calling party application. The ANSWER can contain the set of media that the user chose to accept. The called user can also indicate in the ANSWER message that he wishes to decline the call and a reason e.g., user is busy. This exchange is all that is necessary to establish call state between the parties and allow communication to commence. We note that the invitation/response performs capability negotiation and provides additional information about the called party application in a single message exchange. For example, the RESPONSE may indicate that the called party does not

wish to be disturbed. Once this initial handshake has completed, the calling party application can allocate network resources if desired. This occurs prior to alerting the called party, as in the PSTN.

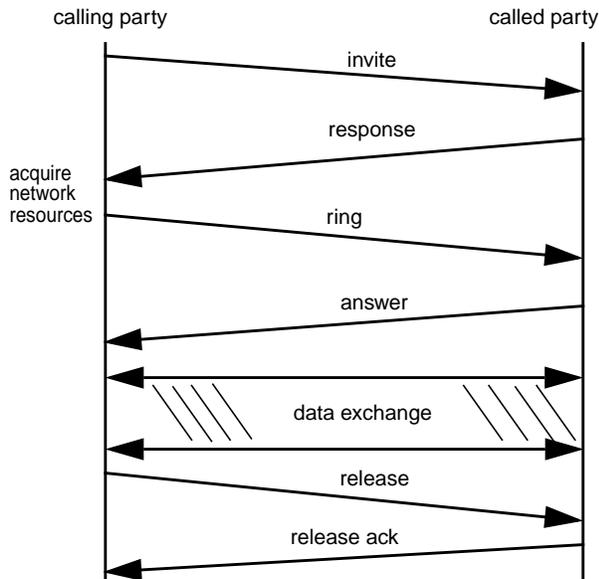


Figure 4: Fundamental ALS message exchange

Table 2 lists ALS messages. During a call, either party can issue an ALS request to modify the call state. This can be to setup a new logical channel, to renegotiate parameters for an existing logical channel, or to teardown a logical channel. The called party’s reply is used to accept, reject or negotiate the proposed change. To enable this negotiation of the characteristics of the flow with the network, it is desirable that the network layer provide a convenient means for negotiation and renegotiation. An example of a lightweight network layer signaling protocol that allows end-systems to efficiently and conveniently negotiate quality of service characteristics is UNITE [18]. It supports both negotiation and renegotiation in a uniform manner..

To terminate a call, either the calling or called party must issue a RELEASE message to its peer, which in turn must send an acknowledgment. At that point the coders are stopped, logical channels are torn down, and related transport resources are freed up. Note that all ALS messages contain a calling party-generated call identifier, allowing the context of each message to be easily determined.

#### 4. Logical Channels and Encapsulation

A goal of TOPS is that packet telephony applications are shielded from details of the underlying transport and network layers. The logical channel (LC) abstraction provides this independence by presenting a uniform communications interface to the application layer. LCs are used to carry application-layer signaling and media data, and support

Table 2: Key ALS messages

Type	Function
INVITE	invite called party to enter call; negotiate call state
RESPONSE	supply called party’s reply; negotiate call state
RING	alert called party
ANSWER	notify caller the result of alerting the remote user
MODIFY	specify desired changes to the call state e.g. adding a media channel
MOD-ACK	reply, selecting acceptable changes from a modify request
RELEASE	notify intention to terminate the call
REL-ACK	acknowledge call termination; delete call state

multiplexing of related data streams onto a single transport connection. For an ordinary telephone call this can reduce the delay of call establishment by multiplexing signaling and data on a single transport connection. Multiplexing of LCs allows temporal ordering to be preserved among streams, without the need for explicit inter-stream synchronization. Use of multiplexing has implications for meeting LC quality of service requirements. The LC layer allows applications to control which LCs are multiplexed, considering their different quality of service, reliability and delivery order requirements.

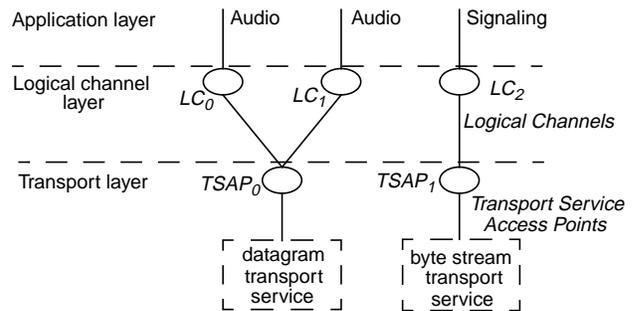


Figure 5: Mapping of logical channels to transport services

Figure 5 shows a mapping of logical channels to transport services for a packet telephony application comprising three LCs - two for audio and one for application-layer signaling. The audio LCs are multiplexed onto a transport service access point (TSAP) that uses best effort datagram transport service. The signaling LC is mapped to a reliable byte stream transport service. This illustrates the function of multiplexing application data over available transport services. LCs are mapped to TSAPs and multiplexing is achieved by allowing LCs to share a TSAP. On the receive

side, the LC layer demultiplexes data arriving on a given TSAP. An implementation is described in Section 7.

Demultiplexing data requires messages to include a header identifying their destination LC. TOPS defines a header format which is used to encapsulate audio packets, provide LC identifiers, and other packet telephony specific fields. The design is simple and does not make assumptions about the network layer used. In most packet telephony implementations, an application reads blocks of audio samples from an A/D converter, encodes them using a negotiated format, adds an encapsulation header, and arranges for transmission. The receiving application reassembles packets, performs delay adaptation, and writes decoded audio samples to a D/A converter. Since the size of the audio blocks is fairly small due to delay constraints, the TOPS encapsulation scheme contains only essential information in the header thus minimizing overhead. The encapsulation format is generic and may be used for control messages and other media data.

The format of the encapsulation header is shown in Figure 6. An 8-bit logical channel ID (LCID) allows demultiplexing of LCs that share a transport connection. This is followed by a 7-bit sequence number which is used for reassembly and delay adaptation. Each packet contains an 8-bit coding field in the header which defines the compression and coding format used by the audio data in the packet. This field allows the coding format to be changed on-the-fly without the need for a separate control message exchange. These basic fields provide the minimum information needed for audio encapsulation. Flexibility is achieved through use of a header extension bit (X), allowing an application to provide additional header information, possibly using type-length-value fields. For example, it can provide timestamps for inter-media synchronization.

8	7	1	1	7	8
LCID	SEQ #	X	M	RESRVD	CODING
Octet 1	Octet 2	Octet 3		Octet 4	

LCID: Logical channel ID  
 SEQ# Sequence number  
 M: Marker bit  
 RESRVD: Reserved for future use

**Figure 6: Packet Header Format**

The encapsulation scheme also supports synchronization of generic control operations for a media LC by using a marker (M) bit in the header. As an example of the use of this bit, consider the synchronization of audio packets with related control packets, assuming that the control and audio LCs are not multiplexed together. This could be used to enable switching to a new set of Huffman tables for use with a decoder. Since the control messages may experience different delays than the audio data, a mechanism is needed

for synchronization. This is achieved by setting the marker bit in an audio packet indicating to the receiver that a control packet was transmitted and that the action it specifies must be performed before the received audio data may be used. The receiver is required to acknowledge the receipt of the control packet. Once acknowledged, the sender clears the marker bit in subsequent audio packets. On detecting this transition the receiver sends an acknowledgment to indicate the end of synchronization. This handshake has the restriction that only a single control operation can be outstanding at any time. It also requires sequential delivery semantics for the messages on the control and data LCs.

## 5. Interworking with the PSTN

Interworking with the PSTN is provided by telephony gateways that convert between the packet stream and its voice encoding, and TDM (time division multiplexing) 8-bit mu-law format used by the PSTN.

### 5.1 Directory Support

For a call from a packet terminal to the PSTN, the user may have specified either a telephone number or the DN of a user, such as an X.500 name, that has a PSTN telephone number as one of its associated call appearances. In either case, the directory service returns the address of a telephony gateway rather than the address of a terminal in the call appearance. When a query contains a telephone number, the local directory server may need to map from local dialing plan numbers to a standard E.164 address. Since a gateway is likely to handle a range of telephone numbers and since there may be multiple gateways serving the same range, the directory service may use specialized mechanisms for storing and searching for PSTN telephone numbers and for locating gateways. When a query contains a DN that is not a telephone number, the application information field in the gateway's call appearance contains the telephone number to be used by the gateway to place a PSTN call.

For calls from the PSTN to a packet terminal, the subscriber dials a regular telephone number that is routed through the PSTN to a gateway. The gateway then tries to resolve the telephone number to a user inside the TOPS domain. Since the information conveyed from the PSTN is in numeric format only, it is necessary to assign an additional DN in the form of a (personal) telephone number. This number can be used from the TOPS domain as an alternative DN. Once the user record has been located in the directory, a call handling profile is returned to the gateway to complete the call. When the called number answers, the gateway bridges the connection between the two domains. The gateway generates ringback or busy tones.

### 5.2 ALS Support

ALS is used to establish calls between packet terminals and gateways. Each gateway acts as an ALS proxy, processing

messages on the packet network and taking appropriate action on the telephone network side. Figure 7 shows calls originating both in the packet network and the telephone network.

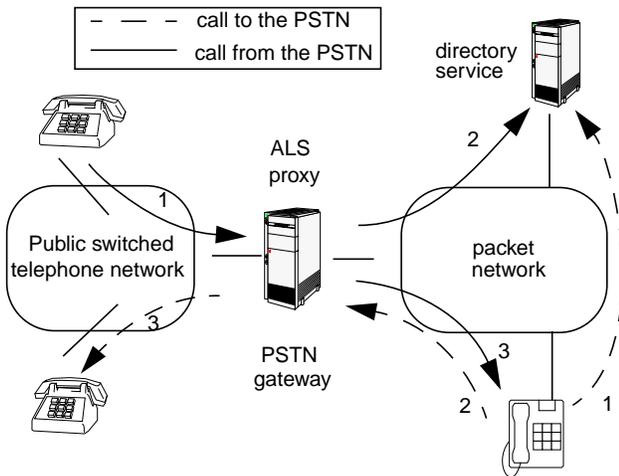


Figure 7: Calling to/from the PSTN

To establish a call the terminal initiates a regular ALS exchange with the gateway. The telephone number returned by the directory is included in the ALS invitation as a message extension. The gateway terminates the ALS protocol, translates messages to actions required on the telephone network side, responds appropriately to the called party, and maintains call state for each active call. This signaling interworking is illustrated in Figure 8 for the case of a Q.931 ISDN gateway. A gateway provides media transcoding between the encodings used by the packet telephone and the PSTN when necessary.

From the telephone network's point of view, a gateway is a termination point for a set of telephone numbers. A call originating on the PSTN and destined for one of the TOPS terminals is routed to the gateway and terminated there. Having queried the directory, the gateway issues an ALS invitation to the destination terminal, performs signaling translation and maintains call state throughout the call.

## 6. Implementing Telephony Services in TOPS

As an example of the migration of existing services to a packet network, consider the implementation of the familiar *call waiting* feature. In the POTS network, an indication of a second incoming call is provided through the insertion of special tones by the local telephone switch into the call that is being interrupted. This feature not only requires the local telephone switch to insert tones and detect user responses (i.e., a flash hook), but also requires it to maintain the signaling state for two simultaneous calls. The ability of a packet telephone to support multiple simultaneous streams, coupled with its processing capabilities and en-

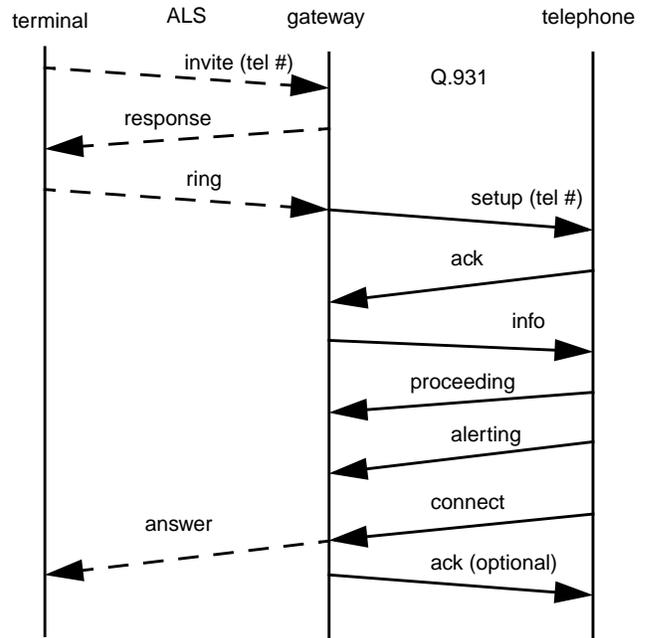


Figure 8: Example of ALS proxy interworking

hanced user interface, allows this service to be implemented entirely at the end-system. When a call is placed to a packet telephone which is already participating in a call, the called packet telephone may decide how to handle the incoming call request. It may locally generate a call waiting tone, as in current call waiting, or on packet telephones with an enhanced user interface it may create a pop-up icon with information about the call. This pop-up icon may include information about the call such as the caller's name, the topic of the call, the importance of the call, and so on. Other common POTS services such as: call forwarding, call transfer, three-way calling, and return-call may similarly be implemented entirely by end-systems.

Packet networks allow traditional POTS services to be enhanced without modification to core network switches or their software. An example of such an enhancement is *n-way call waiting*. In this service a packet telephone's enhanced user interface may be used to accept multiple incoming calls and allow a user to cycle through them in an arbitrary manner. We also envision more imaginative capabilities enabled by the richer signaling allowed by packet networks. For example, a meek caller may indicate in the invite request that he wishes to be notified if the called party is already in a call. With this notification, he is given the option of whether or not to interrupt the called party with a ring message, allowing him to choose to not disturb the call in progress.

Thus far we have considered traditional voice calls, but it is clear that in the future, telephone calls will expand to multimedia calls. For example, while users speak they may share a common whiteboard, view the same video clip, or

participate in a video call. This creates the need for parties to be able to discover compatible media formats when placing a call, and dynamically modify media formats during a call. Network directory servers may be used to provide information on media formats supported by a called party, simplifying the task of capability negotiation during call setup.

Packet communication is opening the door for a variety of new services. An example is that when a multimedia call is established, the caller may wish to move to a terminal with different capabilities, while saving the context of the current communication. For example a caller may answer a call in the bedroom saying “Hold-on, I’ll take your call on my video phone in the living room.” Such a “teleporting” [12] service requires dynamic adaptation of media formats and possibly the transfer of state. Dynamic adaptation of media formats is also required when a call is transferred, for example by a receptionist, who supports a different set of media formats than those required by the ultimate destination of the call. Multimedia calls may benefit from a “virtual hold” service in which parties involved in a call are allowed to pause for an extended time period and later resume the call where they left off. This requires saving of the call state and the ability to reinstate it once the call is resumed. To conserve network resources, connections used by media streams should be released during a virtual hold. This is conveniently achieved in packet networks where the mechanism used to modify a call’s state may be separate from the mechanism used to establish end-to-end connectivity.

Many of the standard POTS services can be supported in TOPS by appropriate customization of the query handling profile for each distinguishing name, or the call handling profile returned for each query. For example, POTS supports various flavors of call forwarding in which a call placed to a certain telephone number is forwarded to another number either by default, or if a user is busy, or if he does not answer. Supporting default call forwarding in TOPS is easy; a query to the directory service for a particular user name returns the address corresponding to the packet telephone (or perhaps a voice mailbox) to which the user would like the call forwarded. Subsequently, the caller’s terminal uses ALS to setup a call to this address.

Forwarding a call when a user is busy, is accomplished by having the directory service return two NLAs in response to the initial query with a call handling profile that indicates which one to try first. If ALS fails to get a busy response to the RING sent to the first NLA, it tries the next NLA.

The simultaneous ring feature provided by PBXs is emulated by having the directory service return a call handling profile that has more than one address at the same priority. ALS then attempts to establish a call to all of these addresses simultaneously. Services similar to the “advanced 800

number” service can also be emulated using the directory service. This service supports sophisticated policies for translation of an “800” number to a terminating telephone number. The directory can for example do time-of-day based routing. The QHP provides a flexible way of customizing these features by allowing individual users to specify the behavior and policies they need.

In addition to customizing the QHP, a user may control the service features he receives by suitably customizing the behavior of his telephony application. For example, when there is an incoming call to a user who is already involved in another call, ALS determines based on a user service profile whether to provide the user with an indication of the incoming call, return a busy indicator to the caller, or to provide the caller access to a voice mailbox.

## 7. Implementation

A packet telephony system has been implemented to gain experience with our architecture. The initial focus for voice transport is native ATM, since it provides support for quality of service and offers good delay characteristics. Control information, such as directory access and ALS, is exchanged over native ATM or classical IP-over-ATM.

### 7.1 TOPS Terminals

The prototype system consists of a set of packet telephones and servers interconnected by an ATM network infrastructure, as shown in Figure 9. There are three types of tele-

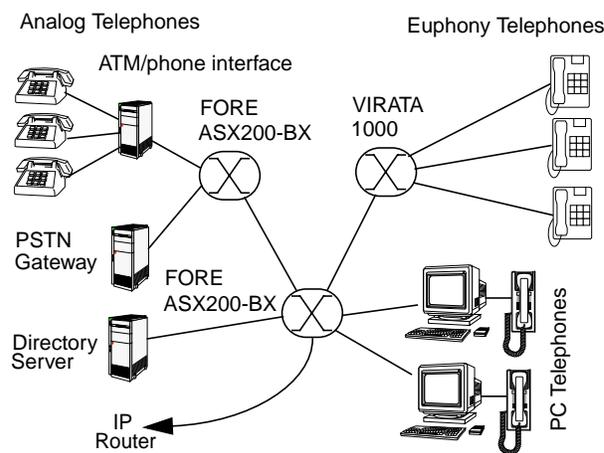


Figure 9: Current packet telephony testbed

phones: PCs running a telephone application, custom built ATM telephone appliances, and analog phones that are controlled and interfaced to ATM by a PC. A server provides a gateway to the PBX (and hence PSTN), and a directory server is used for queries to ATM addresses.

The PC telephone is implemented as an application over Linux. The hardware configuration consists of a Pentium based PC with full-duplex Gravis Ultrasound and Sound-

Blaster audio cards sound cards and a ZeitNet 155 Mbps ATM card. The application uses the Open Sound System (OSS) API [13]. It supports a variety of audio coders including G.711, G.723.1, G.726, G.728 and G.729. Both UDP as well as native ATM AAL5 are used as audio transport protocols. In addition, the software provides a graphical X windows user interface for call control, including invitation, release, call transfer, etc. A user level LC library and ALS signaling have been implemented over the Linux ATM API [1].

A custom built ATM telephone appliance has been designed and is being used experimentally. It is based on the Euphony processor [5,6] developed by AT&T Labs and LSI Logic. Euphony is a highly integrated RISC processor that incorporates signal processing functions, a 25 Mbps ATM interface, a digital audio interface, and all the system logic required to build a complete system (e.g., DRAM/SRAM controller, five DMA channels, counter timer, etc.) into a single VLSI device. The ATM telephone hardware consists of a Euphony, SRAM, Flash, A/D and D/A, UART, and a 25 Mbps ATM physical layer device. Due to its high level of integration, the entire system is packaged in a standard telephone case. The software running on the ATM telephone consists of the telephony application, VxWorks real-time operating system, and ATM signaling code. The ATM telephone augments the traditional POTS telephone user interface, which consists of a handset, a switch hook and a 12 button keypad, with two additional buttons and three LEDs. Features such as LCD screen, mouse, or keyboard may be connected to the base unit through a serial port on the back of the ATM telephone. For example, it is used for a prototype 10 inch color touch screen with a keyboard and mouse, allowing access to advanced services.

The third type of device is an off-the-shelf analog telephone connected to a PC-based ATM/telephone interface card [8]. In this arrangement audio data is handled entirely within the interface card, which provides audio A/D and D/A, ATM access and AAL functions. The card forwards off/on-hook and key press events to a control program on the PC which is responsible for generating appropriate dial tones, directory access and call signaling. Users dial using either unique user identifiers or telephone numbers which are passed to the directory server. Special key press sequences are provided for user registration and deregistration, to take advantage of user-mobility and caller-ID-based directory features.

### 7.2 LC and ALS Implementation

The LC layer is implemented as a user-level library and exports a synchronous receive and send API to the application. The LC layer runs on top of native ATM AAL5, UDP, and TCP. Applications use a common API to create and use LCs independent of which of the above transport protocols

are used. For TCP, which is a byte stream protocol, the LC layer transparently prefixes each message with a length field to preserve LC message boundaries. Currently the LC mechanism has been implemented and tested in Linux and is being ported to Wind River's VxWorks operating system.

A subset of the ALS protocol has been implemented over VxWorks. This protocol is used by a phone application to setup calls between the Euphony ATM terminals. ALS responds to on-hook and off-hook events, performs digit collection, does directory lookup, and does call setup. The application also has modules for generating dialtone, touch tones, and error tones.

### 7.3 Directory Implementation

The TOPS directory server is shown in Figure 10. It is organized around a database of user records. Clients can access the directory service using different protocols (the current implementation supports both IP and native ATM). A protocol adapter module produces a directory query from a message (e.g., a UDP datagram or an AAL5 frame) and similarly constructs a response with the results of the query.

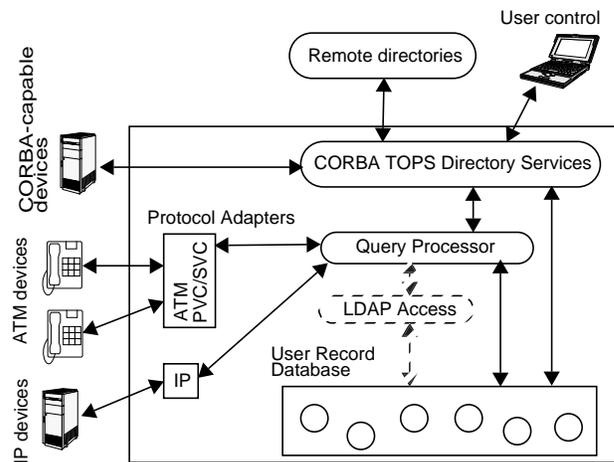


Figure 10: Directory Software Architecture

Every query is passed to a query processor that identifies a user record in the database. It uses the persona information in the query (if present) to select the appropriate QHP to execute. The query processing interface is also accessible as a CORBA service [11] for clients that have this capability.

In order to submit a query, the client uses a simple message protocol (shown in Table 3). Each message is of the form: `cmd_name attrname1=attrval1 attrname2=attrval2 ...`, where `cmd_name` identifies the action to be taken by the directory. Parameters are specified as attribute name/value pairs.

User profiles can be created and modified through a separate CORBA service. For convenience, a user can modify his profile using a Web browser, that loads a Java applet

**Table 3: Directory Command Interface**

Command	Supplied Attributes	Description
Query	Callee, Caller, Persona, Terminal Caps.	Submits a query to the directory and returns a set of call appearances that can be used by the terminal to locate the user.
Create Update	ESN, Address, Priority, Timeout, Description	Creates or updates a call appearance record when the user or a terminal moves.
Find	First/Last Name, Email, etc.	Returns a list of DNs from the local database that match the user supplied predicates (white pages service)

from the server that supports an interface for profile modification. Another set of interfaces is used to access remote directories in order to fetch user records, update call appearances, etc.

In the future we plan to integrate the TOPS user database within an X.500 directory service. The main advantage of this is that user records for TOPS can be kept together with information for other services. In this case, the TOPS directory server will access the user record through an X.500 access protocol such as LDAP [9] rather than from local disk.

#### 7.4 Implementation of a PSTN Gateway

To interwork between TOPS phones and PSTN phones, a gateway has been implemented. It uses a PC card [3] that terminates a PSTN connection and moves the voice data to/from an ATM VC that is setup between a packet phone and the gateway.

Telephone numbers can be handled in one of two ways. When calling a packet phone from the PSTN, telephone numbers assigned to packet phones are also assigned to a gateway. Dialing a packet phone from the PSTN results in a call being setup to the gateway. The gateway consults the directory to map the dialed number to an ATM VC, and arranges for the hardware to move the voice data between the telephone circuit and the VC. Alternatively, the gateway itself is assigned a PSTN telephone number. When this number is called from the PSTN, the gateway provides a secondary dialtone to the caller. The caller enters the name of the called party via the telephone keypad. The gateway maps this name to an ATM VC and sets up the voice path as before.

To call out to the PSTN, the initiating terminal connects to the gateway and uses ALS to convey the target PSTN telephone number. The gateway initiates a voice call on the PSTN to the specified number and sets up the voice path as before. The gateway may also provide secondary dialtone, allowing the caller to dial directly.

## 8. Summary

TOPS enhances existing telephone services. It leverages the combination of packet networking and more sophisticated end-systems to enable a variety of creative new services. TOPS extends traditional directory services to

support powerful location management functions. The directory service maps from a user's distinguishing name to one or more terminals, servers or other entities to which a call should be routed. Users have the ability to customize the translation using a profile which allows them to be associated with multiple terminals, and to screen and route calls addressed to them. By using a name as the basis for reaching a user, the TOPS architecture directly supports both user and terminal mobility.

TOPS incorporates an application layer signaling (ALS) protocol that supports call establishment, terminal capability negotiation and session management for the duration of the call. ALS supports multiple simultaneous sessions, which allows traditional services such as three-way calling to be easily implemented. ALS supports multiple media streams to be associated with a call. TOPS uses an encapsulation format that is simple and efficient for real-time transport. A logical channel mechanism allows control and media streams to be multiplexed over an single transport association, and to be managed as a single, application-layer context. For example, one or more control channels may be used to manage the constituent media streams of a single ALS context. This allows the application to be isolated from the characteristics of the underlying transport layer (such as a datagram or connection oriented transport) or network layer. TOPS only assumes that the network supports establishment of end-to-end connectivity between terminals, with the appropriate quality of service to carry the real-time media streams.

TOPS interfaces packet telephony services with the public switched telephone network through a telephone gateway. The gateway performs conversion from packet to circuit, and acts as a signaling gateway between ALS and traditional telephone signaling.

Conferencing of multiple call participants is an important service that must continue to be supported as telephony evolves to packet networks. TOPS supports a variety of conferencing modes. For details see [17].

We believe that mechanisms for authorization and usage recording must be developed that allow service providers to charge users for telephony services. We are presently exploring these issues.

## References

- [1] W. Almesberger, "ATM over Linux documentation", <http://ircwww.epfl.ch/linux-atm/doc.html>.
- [2] O. Kahane and S. Petrack. "Call Management Agent System: Requirements, Function, Architecture and Protocol", IMTC Voice over IP Forum submission VoIP97-010, 1997.
- [3] Dialogics Corp., "More About Computer Telephony". <http://www.dialogic.com/company/3602web.htm>.
- [4] P. Mockapetris. "Domain names: Concepts and facilities", IETF RFC 882, 1983.
- [5] Onufryk, P. Z. "Euphony: A Signal Processor for ATM," EE Times, pp. 54-80, January 20 1997.
- [6] Onufryk, P. Z. "Euphony: An Embedded RISC Processor for Low Cost ATM Networking and Signal Processing," Design SuperCon97: Digital Communications Design Conference, pp. C112-1 to C112-16, 1997.
- [7] ITU-T. "Recommendation H.323: Visual Telephone Systems and Equipment for Local Area Networks Which Provide a Non-guaranteed Quality of Service", 1996.
- [8] Inno-Media Logic Ltd., Building Telephony Switches and Systems from 1,000 to 100,000+ ports with Computer Telephony Chassis linked over an ATM infrastructure. [http://www.impl-cti.com/index\\_doc.htm](http://www.impl-cti.com/index_doc.htm).
- [9] M. Wahl, T. Howes, S. Kille. "Lightweight Directory Access Protocol (v3)", IETF draft <draft-ietf-asid-ldapv3-protocol-09>, November 1997.
- [10] P. Bhagwat, S.K. Tripathi, C. Perkins, "Network Layer Mobility: an Architecture and Survey", IEEE Personal Communication, 3 (3), June 1996.
- [11] Object Management Group, "The Common Object Request Broker Architecture and Specification", Rev. 1.2, Dec. 1993.
- [12] T. Richardson, F. Bennett, G. Mapp, A. Hopper, "Teleporting in an X Window System Environment", IEEE Personal Communications Magazine, Vol. 1, No. 3, pp 6-12, September 1994.
- [13] Open Sound System (OSS/Linux). <http://www.4front-tech.com/linux.html>.
- [14] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson. "RTP: A Transport Protocol for Real-Time Applications", IETF RFC 1889, January 1996.
- [15] H. Schulzrinne, "A comprehensive multimedia control architecture for the Internet", 7th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 97), May 1997.
- [16] M. Handley, H. Schulzrinne, E. Schooler. "SIP: Session Initiation protocol", IETF draft <draft-ietf-mmusic-sip-04>, November 1997.
- [17] N. Anerousis, R. Gopalakrishnan, C.R. Kalmanek, A.E. Kaplan, W.T. Marshall, P.P. Mishra, P.Z. Onufryk, K.K. Ramakrishnan, C.J. Sreenan. "TOPS: An Architecture for Telephony over Packet Networks", IEEE Journal on Selected Areas in Communications, to appear, 1<sup>st</sup> quarter 1999.
- [18] G. Hjalmtysson and K.K. Ramakrishnan. "UNITE: An Architecture for Lightweight Signaling in ATM Networks", *Proceedings of the IEEE Infocom '98*, San Francisco, CA, April 1998.
- [19] ITU-T, "The Directory: Overview of Concepts, Models and Service", Rec. X.500, 1993.