Contents

Chapter 1

IP Multicast

1.1 Multicast Programming

The multicast programming model for Unix and Windows sockets is fairly straightforward. Almost all modern operating Unix and Microsoft operating systems, including Windows'95 and NT 4.0 [not sure about 3.51], support IP multicast. A few older versions of Unix need to patched or, in the case of Windows 3.1 or DOS, a multicast-capable "Winsock" stack needs to be added. Multicast sending and receiving requires no special privileges. Naturally, only UDP is supported with IP multicast.

To send data to a multicast group, the program only needs to set the destination address to that group. For example,

```
sin.sin_addr.s_addr = inet_addr("224.2.0.1");
if (connect(s, (struct sockaddr *)&sin, sizeof(sin)) < 0) {
  fatal("connect");
}
```

Any address in the range 224.0.0.0 to 239.255.255.255 can be used, but a number of addresses are reserved and should be avoided (see Table **??**. The address range of 224.x.x.x to 224.x.x.x has been set aside for dynamically allocated audio and video conferencing sessions.

By default, all packets will be sent with a time-to-live value of 1, so that they cannot travel beyond the local subnet. Multicast datagrams with a TTL of zero are restricted to the same host. The reach of larger TTL values is not as strictly defined, but typically, a TTL value of 32 gets packets to every node in the same site, 64 to the same region such as a country and 128 to the same continent.

To set the TTL value of a socket in `s` to something larger, the `setsockopt()` call is used:

```
unsigned char ttl = 16;

if (setsockopt(s, IPPROTO_IP, IP_MULTICAST_TTL, &ttl, sizeof(ttl)) < 0)) {
   perror("IP_MULTICAST_TTL");
}
```

[Admin. scoped addresses]

To receive IP multicast packets, the receiver needs to bind to a port and set socket options to the desired multicast addresses. [Can you bind to several?] The `mread.c` program in Fig. 1.1 shows how to read data from a multicast group and print out the packet content.

The program sets the `REUSE_PORT` socket option that allows several programs to bind to the same port, something not normally allowed. That way, several instances of the program running on the same host can receive packets for that group.

[Discuss: effect of bind() with multicast address. SO_REUSEPORT, RFC2133 for IPv6]

If a process subscribes to a multicast group and sends a packet to that group, it will, by default, receive a copy of that packet. This is referred to as "loop back". For some applications, getting copies of one's own transmissions just increases processing overhead and forces an application to distinguish the "echo" of its own packets from those arriving from elsewhere. Setting the socket option `IP_MULTICAST_LOOP` prevents any outgoing packets from being received locally, by any process. Unfortunately, there is no way to tell the kernel to just send multicast packets to other local processes. Looping can be safely disabled when there is only one instance of a particular application on a host, such as a router process or some other system daemon. For conferencing applications such as session announcements, disabling loopback is probably not a good idea. Video and audio tools should leave loopback enabled if more than one could sensibly be using the same multicast address at the same time, e.g., for a multi-user server running video applications across X Windows.

Remember that sending and receiving multicast packets on a local area network, including Ethernet, token ring, FDDI and ATM LAN emulation (LANE), does not require any assistance from the network.

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <errno.h>
#include <stdlib.h>
#include <unistd.h>

main(int argc, char *argv[])
{
  int s;                  /* socket */
  int n;                  /* number of bytes read */
  int one = 1;
  char buf[1500];         /* buffer for incoming packet */
  struct ip_mreq mreq;    /* multicast address */
  struct sockaddr_in sin;
  int i;

  if ((s = socket(PF_INET, SOCK_DGRAM, 0)) < 0) {
    fatal("socket");
  }
  if (setsockopt(s, SOL_SOCKET, SO_REUSEADDR, (char *)&one,
        sizeof(one)) == -1) {
    fatal("setsockopt: reuseaddr");
  }
  mreq.imr_multiaddr.s_addr = inet_addr(argv[1]);
  mreq.imr_interface.s_addr = htonl(INADDR_ANY);

  sin.sin_family      = AF_INET;
  sin.sin_addr.s_addr = mreq.imr_multiaddr.s_addr;
  sin.sin_port        = htons(atoi(argv[2]));

  if (setsockopt(s, IPPROTO_IP, IP_ADD_MEMBERSHIP, (char *)&mreq,
      sizeof(mreq)) < 0) {
    fatal("IP_ADD_MEMBERSHIP");
  }

  if (bind(s, (struct sockaddr *)&sin, sizeof(sin)) < 0) {
    if (errno == EADDRNOTAVAIL) {
      sin.sin_addr.s_addr = INADDR_ANY;
      if (bind(s, (struct sockaddr *)&sin, sizeof(sin)) < 0) {
        perror("bind");
        exit(1);
      }
    }
    else {
      perror("bind");
      exit(1);
    }
  }
```

Chapter 2

Internet Telephony

The use of the Internet to carry traditional “telecommunications” services such as telephony, fax, radio and television is just beginning. Before discussing some of the protocols needed to allow this to happen, it is instructive to take a closer look at the motivation for these changes. Given how much telephone service is part of the fabric of everyday life, we now need to be much more concerned with issues of scalability, reliability and quality. This chapter compares today’s Internet to the phone network and tries to motivate why Internet telephony may well replace the “innards” of the current telephone system. The next chapters will then take a closer look at the issues of carrying “telephony” data and the control of “calls”.

Following convention, we will use the term “Internet telephony” for carrying voice and fax across the Internet. It should be understood, however, that once the Internet is used to carry voice, adding other media such as video or sharing applications is simply a matter of adding bandwidth.

The term *voice-over-IP* (VoIP) or ”IP telephony” (IPtel) is also used to refer to these services, emphasizing that the common aspect is the carriage of voice in IP data packets on both the Internet and intranets.

What makes Internet telephony different from computer-based conferencing discussed in Chapter **??**? Internet telephony is dominated by two-party calls, will probably remain dominated by carrying voice and has to interoperate with a huge installed base of “end systems” (telephones and switches) dating back to the previous century. Rather than desktop computers running graphical user interfaces, most end systems even in a world dominated by Internet telephony will have to be simple, cheap single-function devices. Also, service expectations are far higher than for computer services. After all, people expect to use the telephone to call the fire department and to notify the electric utility when line power has been lost.

2.1 Introduction

The current Internet and intranets are making the transition from being a convenient additional means of communications that one can easily do without to an essential communication tool. Many in the technical and educational fields can probably function and continue to work reasonably well without an outside phone connection or PBX for a few hours, but are severely inconvenienced if the internal or external networks are unreachable.[1] Many engineers and researchers now receive far more email per day, often more than a hundred messages, than phone calls, faxes and postal mail combined. The importance is still largely limited to the technical community, however. receives about A Future/Gallop Organization poll of 972 workers from large companies found that on average, respondents received 31.8 phone calls, 13.6 e-mails, 11.2 voice mails and 8.8 faxes per day [2].

As Table 2.1 shows, the volume of email is dwarfed by daily postal deliveries and long-distance phone calls. Also, only 32% of those who go online say they would miss services [3].

Table 2.1: Communications volume per day (United States)

means of communications	year	millions/day
US Postal Service	1995	580
AT&T US phone calls	1995	200
ATM transactions	1993	20
UPS daily deliveries	1995	12
AOL email	1998	22
Federal Express	1995	2

However, just as telephone and fax have started to displace international postal mail, the Internet could have similar displacement effects as it becomes ubiquitous. (The amount of international mail from the US has been *declining* since 1992 by about 5 to 7% annually, while international call volume rose by 32% between 1992 and 1994.)

Network	date measured	traffic	Gb/s	remarks
NSFNET	end of 1994	15 TB/month	0.046	70% of total Internet traffic
Internet backbone	end of 1994	20 TB/month	0.061	
AT&T frame relay network	late 1997	5.7 TB/day	0.53	
MCI Internet backbone traffic	Nov. 1997	140 TB/week	1.8	
Total Internet traffic late 1997	2,000 TB/month	6.1		
U.S. local phone calls	1995	2,228 GDEM	271	
U.S. intrastate toll calls	1995	344 GDEM	42	
U.S. interstate toll	1995	451 GDEM	55	
U.S. switched access	1996	468.8 GDEM	57.2	
World telephony	Nov. 1996	600		

Table 2.2: Telephone and Internet traffic volumes; GDEM: giga dial equipment minutes (10^9 DEMs)

Table 2.1 compares traffic statistics for Internet and telephony services. Telephone traffic is

[1]Compare this to the effect of a prolonged phone outage documented by Wurtzel and Turner [1].

commonly measured in dial equipment minutes (DEM), that is, minutes that the local switch is busy with a given call. A single phone call racks up two DEMs for each minute of "talk time". The estimate of world telephone traffic is based on an estimate that each of the approximately 640 million phone lines in the world are used about 20 minutes per day.[2]

The table shows that telephony traffic still dwarfs Internet traffic. However, on the transatlantic and transpacific links between the U.S. and Europe and Japan, Internet traffic already approaches voice traffic. For example, in 1997, the total installed capacity for Internet services between the U.S. and Japan was 650 Mb/s as compared to 400 Mb/s for telephony. As another example, in late 1997, the data volume between the U.S. and Sweden was about twice the volume of voice traffic. (Among other reasons, it appears to be easier to send an email to Japan than trying to communicate across the language and time zone barrier.)

Also, while landline telephone usage grows by a few percent each year, Internet traffic has roughly doubled each year, even though the number of Internet connected hosts has recently increased by "only" about 30% or so per year. In addition, a large fraction of the international voice traffic is caused by fax machines. Compared to voice, fax service can be most readily replaced by Internet services.

So far the Internet has thrived on services where there is no ready competition. As we will discuss in this chapter, there are a number of reasons to make Internet technology the common platform for a truly integrated services network. Traditionally, service integration has taken place at the trench level (several fibers sharing a single duct), at the physical layer (e.g., wave-length division multiplexing (WDM) or synchronous optical networks (SONET)) or at the link layer (ISDN and ATM). Service integration at the network layer offers the same packet-based service platform with different lower-layer technologies spanning about seven orders of magnitude in link speed, from a 300 baud modem to 2.4 Gb/s.

We next describe the advantages and implications of using the Internet as a "full-service network". The provision of telephony services is discussed in Section 2.2, while radio and (cable) television are briefly covered in Section 2.3. Providing these services accentuates the inadequacy of how residential users access the Internet; Section 2.4 points out alternatives. A suitable protocol architecture for real-time Internet services is summarized in Section 2.5. Reliability is crucial for these commodity services, as pointed out in Section 2.6.

2.2 Telephony

2.2.1 A Quick Review of Telephone Terminology

Today's landline ("wired") telephone system consists of three major components: the local loop, trunks and the signaling system. The local loop is, in most cases, a pair of copper wires that connect each residence or business to the next *central office* (CO), a switching center that handles up to a few thousand phone lines. Since the local loop consists of only two wires (the so-called a/b

[2] The 165 million United States phone lines are each used about an hour a day, a number that has only increased by about 18% since 1980. The amount of interstate long-distance calls has doubled from 4 to 8 minutes during the same time, however.

Figure 2.1: Architecture of the telephone system

interface), both incoming and outgoing voice current share the wire. A *hybrid*, a special transformer, separates the two directions at both the telephone set and the central office. The local loop is designed so that it can pass audio frequencies between 300 and 3,400 Hz, sufficient for recognizing speakers. The central office rings the telephone by sending about 40 to 90 Volts of AC ("ring current") to the phone.

COs are connected by trunk (toll) lines, switched by toll switches. Trunk lines have separate wires or optical channels for each direction. Trunk lines bundle 24 (so-called T1, 1.5 Mb/s), 30 times 24 (T3, 45 Mb/s) or more phone lines into a single transmission line (see Ch. **??**). In turn, several T1 or T3 lines may be bundled into a single fiber optic cable.

There are about XX central offices in the United States, with an average of XX lines per office. A large long-distance carrier has about 130 toll switches that are connected in close to a full mesh. Routing within the phone network is typically static, with one or two alternate routes, limiting the hop count to a much smaller number than is found in the Internet.

The structure of the phone system is also reflected in the phone number. In North America, phone numbers are fixed length, consisting of a three-digit area code, a three-digit prefix and a four-digit number [CHECK TERM]. Area codes are more or less randomly assigned. Each switching center is responsible for one or more prefixes. In many European countries, area codes are variable length, with large cities having short area codes and surounding towns having related area codes.

In addition, each country or region has a one-to-three digit country code, for example 1 for North America and 42 for Switzerland.

Lifting the receiver lowers the resistance of the phone line, which is then recognized by the central office as a desire to obtain dial tone. The telephone subscriber dials by either shorting the connection between the two wires for each dial pulse or, more commonly, sending "touch tones", each consisting of a combination of two pure tones, to the central office (DTMF = dual-tone multi-frequency signaling).

Between central offices, calls are established via a separate low-speed packet network, running protocols only found in the phone system. The current version of this signaling network is known as Signaling System #7 (SS7). Details will be discussed in Section **??**.

2.2.2 Motivation for Internet Telephony

Compared to the current circuit-switched network controlled by a separate signaling network, using Internet technology to provide telephony services has a number of advantages:

Compression: Internet telephony allows the parties to use the encoding most appropriate for their quality needs. They may, for example, decide that for an international call, they would trade lower cost for full toll quality, while a reporter calling in her story to the radio station may go for full FM broadcast quality with little regard for price. Even without quality degradation, 5.3 kb/s (G.723.1) to 8 kb/s (G.729) are sufficient to support close to toll quality as opposed to 64 kb/s for the current landline phone network. This flexibility also has the advantage that during severe network overload, e.g., after a natural catastrophe, telephone customers can still communicate at about 3 kb/s, increasing network capacity twenty-fold.

Compression benefits the provider if services are price at a flat rate, except that it may allow better use of other services if the access bandwidth is limited. These services include emulation of a second phone line on a single access line or non-voice services such as simultaneous use of a subscriber's line for a phone call and web browsing.

Silence suppression: Sending audio as packets makes it easy to suppress silence periods, reducing bandwidth consumption by about half. The savings is more pronounced for multi-party voice conferences or for voice announcement systems. In a telephone conference call, all participants use 64 kb/s in both directions, even though they, for most of the time, only making use of the incoming voice bits.

Silence suppression and voice compression compensate for the lower bit efficiency of packetized communications and Internet telephony in particular. As we will see in more detail later, a typical voice packet may contain 50 bytes of voice samples and 40 bytes of packet headers.

Traffic separation: Sending faxes across a circuit-switched network is rather inappropriate, as this is delay-insensitive, but loss-sensitive traffic. Currently, typical fax machines use only 9.6 kb/s of an access line that could support 56 kb/s or 64 kb/s. Thus, fax traffic should be separated from voice traffic as close to the fax machine as possible and converted into either email messages or a TCP connection [4].

User identification: Standard telephone service offers, for a price, caller id indicating the number or, occasionally, name of most callers[3]. However, during a bridged multi-party conference, there is no indication of who is talking. The real-time transport protocol (RTP) [5] used for Internet telephony easily supports talker indication in both multicast and bridged configurations and can convey more detailed information if the caller desires.

User interface: Most telephones have a rather limited user interface, with at best a two-line liquid crystal display or, in the public network, cryptic commands like "*69" for call-back. Advanced features such as call-forwarding are rarely used or customized, since the sequence of steps is typically not intuitive. The graphical user interface offered by Internet telephony can be more readily customized and offer richer indications of process and progress. It is also easy even for dedicated Internet telephony "appliances" to be configured through a web page.

Security: While the Internet is generally considered insecure, it is actually far easier to tap a standard analog phone interface than a typical switched-Ethernet installation. Beyond physical security, Internet telephony makes it easy to routinely encrypt all signaling and media traffic on the network. (A Pentium 200 MHz computer can encrypt data using the DES (Digital

[3] The phone company sends a low bit-rate modem signal containing the phone number to the phone between the first and second ring.

Encryption Standard []) at a rate of XXX Mb/s.) Unless governments impose legal restrictions, traditional phone tapping, for both legal and illegal purposes, will become extremely difficult. If the government imposes restrictions, it will become difficult for all but those who do not care about the fine points of law to begin with. Detecting whether a particular voice channel is encrypted is rather difficult, particularly since sophisticated criminals can hide, using a process called steganography, an encrypted stream in an innocuous-sounding audio stream.

Computer-telephony integration: Because of the complete separation of data and control paths and the separation of end systems, computer-telephony integration (CTI) [6] is very complex, with specs [7] running to 3,300 pages. All the call handling functionality can be much more easily accomplished once the data and control path pass through intelligent, network-connected end systems. We will describe such functionality in Section 2.5.2 below.

Shared facilities: Many corporations and universities already have high-speed local area networks. Given its low bit rate, packet voice and low-bit-rate video can be readily supported on a well-designed (switched) LAN, even without explicit quality-of-service support.

Advanced services: From first experiences and protocols, it appears to be far simpler to develop and deploy advanced telephony services in a packet-switched environment than in the PSTN (public switched telephone network) [8, 9]. Internet protocols [10] that support standard CLASS (Custom Local Area Signaling Services) [11] features take only a few tens of pages to specify. They can replace both the user-to-network signaling protocols such as Q.931 as well as the network signaling (ISUP, Signaling System 7) and, through cryptography, can be made at least as secure as the existing network.

In the Internet, application-layer intelligence resides in end systems, which are typically replaced much more frequently and have an higher aggregate processing power than typical telephone switches. It is also easier to deploy services one-by-one, rather than waiting for the whole network to be upgraded. (On the other hand, implementing services in switch adjuncts makes them available immediately to all subscribers, regardless of the intelligence of the end system.) Due to implementation diversity, it may also be less likely that software faults in implementations of new features would bring down the whole network. (The Internet can also be used for service creation in the circuit-switched telephone system [8].)

Cost: One of the primary drivers for the commercial introduction of Internet telephony is the ability to offer cheaper long-distance calls. Compared to traditional telephony, costs can be reduced by improved bandwidth efficiency, ease of building services and more efficient switching (see below). However, another, probably short-lived, attraction is that currently, Internet service providers do not have to pay the local access charge imposed on long distance carriers. This charge, at an average of 4.92c per minute (1998), constitutes close to half of the cost of an average long-distance call.

Opportunity for CLECs: CLECs (Competitive local exchange carriers) or CAPs (C. access providers) are telecommunication carriers that compete with the traditional phone company, also known

Figure 2.2: Block diagram of an Ethernet-connected Internet phone

as the ILEC (incumbent LEC). In the United States, the RBOCs (regional Bell operating companies) and GTE are ILECs, while companies like MFS and some long-distance carriers have attempted to become CLECs. Since it is prohibitive to dig up the street for a second set of phone wires or run wires on poles, most CLECs lease the local loop, the copper wire from the local phone switch to the residence or business. However, local competition has not happened nearly as fast as anticipated. It has been suggested that cable TV companies and utilities may use their networks, as discussed in Section **??** to offer Internet access and Internet telephony services.

2.2.3 End Systems

One major disadvantage of Internet telephony is the cost of the end systems. It is hard to build packet voice "telephones" requiring no external power that operate over low-grade twisted pair wires several miles long at the $20 to $50 price point of a basic analog phone. However, Internet phones are not restricted to being applications on personal computers. They are natural applications for network computers and "Internet appliances". Also, a voice-only "packet phones" can be built with a single DSP with on-board A/D and D/A conversion and serial or Ethernet interface (Fig. 2.2). For business applications, Internet appliances with Ethernet interfaces are probably most appropriate, as existing business phone wiring can carry at least 10BaseT (10 Mb/s) Ethernet. For home use, a simple RS-422 or RS-485 serial interface can operate over a distance of 4,000 feet at 100 kb/s, while the ISDN 2B1Q line coding can cover 18 kft (5.5 km) at 144 kb/s. Section **??** describes a possible residential access structure.

2.2.4 Delay

Compared to circuit-switched telephony, packet voice incurs additional delays beyond the usual speed-of-light propagation delay of roughly 5 ms per 1000 km () of fiber. These additional delays are due to packetization, transmission and queueing:

Packetization delays: Packetization intervals are typically chosen between 10 and 50 ms. Shorter packetization intervals increase the header overhead, but make individual packet losses less audible and easier to conceal. Frame-based (typically, low-bitrate) codecs impose a minimum packetization delay of one codec frame, up to 30 ms, and algorithmic look-ahead of up to 7.5 ms.

Transmission delay: Each router or switch hop introduces on the order of 10 μs of transmission delay, assuming a line speed of 155 Mb/s (OC-3) and 200-byte packets.

Queueing delay: The queueing delay depends on the scheduling algorithm chosen. There are at least four choices: best-effort, priority for voice, single guaranteed class for all voice sources and per-source queueing. Best-effort service, the current policy, offers no delay bounds and no loss guarantees, but can work reasonably well in lightly-loaded high-speed networks such as most LANs.

Giving voice and other CBR-like traffic priority ensures that voice traffic does not suffer congestion loss and limits the delay to waiting for any in-progress transmissions. Voice traffic can be marked by either the IP type-of-service ("low delay") or priority bits, with appropriate, measurement-based admission control at the edges. If we do not allow preemption of lower-priority classes and assume a maximum packet size of 8,000 bytes, each router with OC-3 links adds at most 0.4 ms of delay. This estimate is likely to be on the high side, since backbone speeds are likely to be at least OC-12 within a few years and Ethernet LANs limit packets to 1,592 bytes.

An alternative which avoids the problems introduced by simple priority schemes is to treat all voice connections as a single flow for purposes of reservation and scheduling, allocating, by some variation of weighted fair queueing, the appropriate aggregate bandwidth to that class. The delay bound is similar to the one discussed in the paragraph above. Unfortunately, this type of allocation is not supported by the current version of RSVP.

Finally, each flow can be admitted and schedule individually, with the concomitant overhead, but the same delay bound.

Given the per-hop delays, total end-to-end delays for the 40 ms packetization interval typical for low-bit rate voice should consist of the propagation delay plus about 44 ms, of which 4 ms are the total transmission delays for a ten-hop path. For sample-based codecs only, IP-over-ATM or native ATM systems can reduce the packetization delay by transmitting ATM cells as they are filled rather than waiting for a whole packet.

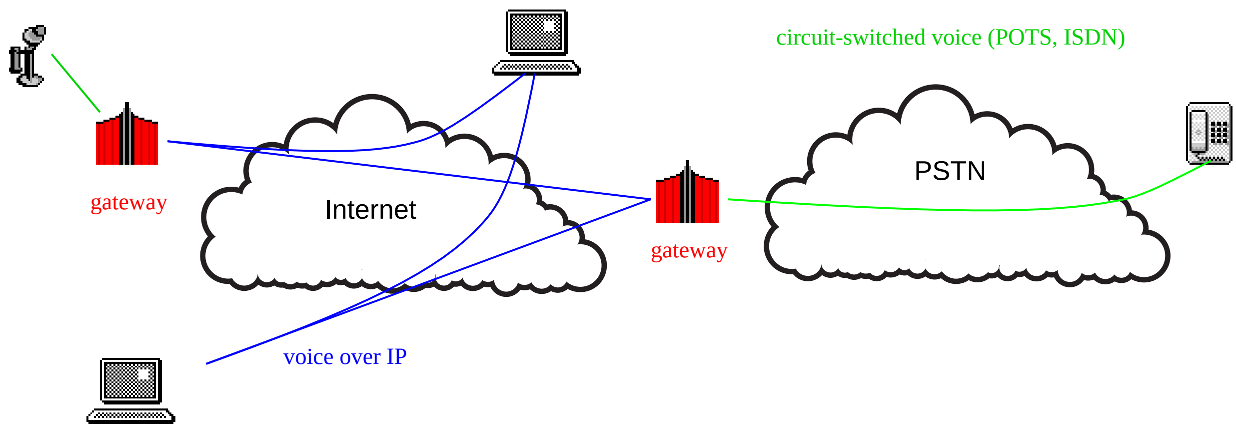
For end-to-end packet telephony, there are no transitions between two-wire and four-wire cabling and thus, no hybrids. This avoids the electrical echo introduced by hybrids that otherwise limits voice circuits to end-to-end delays of about 45 ms without echo cancellation [12] [13, p. 673]. This leaves the acoustical feedback at the handset, which is generally negligible, and allows serviceable connection quality with delays of several hundred milliseconds[**?**]. Naturally, if speakerphones are used, echo cancellation is required.

2.2.5 Interoperation with Circuit-Switched Telephony

It is clear that the 700 million or so telephones in the world will not be converted any time soon to packet telephones. (There are only about 200 million computers of all kinds in the world.) Dumb, cheap end systems will continue to be appropriate in many circumstances. The corner gas station is not likely to install a coin PC next to the car vac. Thus, interoperation between "classical" telephony and packet telephony is an important consideration. This can take place in various ways, as shown in Fig. 2.3. We can distinguish the following cases, which can be combined as needed:

end-to-end packet: End systems such as network computers, dedicated "Internet phones" or PCs packetize audio; the packets are delivered to one or more similar end systems for playback.

Figure 2.3: Possible interactions between Internet telephony and POTS



tail-end hop off: Packet networks are used for long-haul voice transmission, while standard circuit-switched voice circuits connect the CPE (telephones) to the packet telephony gateways. This can be used both for individual voice circuits as well as for PBX interconnect. Tail-end hop off allows to bypass long-distance phone companies as well as to connect POTS (plain old telephony service) devices to packet audio end systems. There exist "Internet telephony set-top boxes" that connect to a cable modem via an Ethernet interface. This device has a subscriber loop circuit that allows to plug in the regular house phone line. Dial tone and ring current are generated within the device and make it appear to old-style analog phones that they are connected to the central office.

local packet delivery: Voice is generated by packet audio end systems, but carried as circuit-switched voice over leased or public facilities in the wide area. An example is the "packet PBX" connecting to the PSTN.

term	originating system	wide-area	local-loop termination
	POTS	packet	POTS
	POTS	packet	packet
	packet	packet	packet
tail-end hop off	packet	packet	POTS
	packet	POTS	packet
	packet	POTS	POTS

2.3 Radio and Television

The FCC reports that as of December 1996, 12,140 AM/FM radio stations (4,857 AM, 5,419 commercial FM, 1,864 educational FM) are broadcasting in the United States. In other countries,

Table 2.3: Intensity of communications activities

	year	source	minutes/day
long-distance calls, US	1994	[14]	14
America Online (AOL)	1997	[15]	32
local phone calls, US	1994	[14]	38
total phone line usage	1994	[14]	52
Internet	1995	[16]	57
Television, Sweden	1996	[17]	120
television, USA	1996	[17]	240

particularly of smaller area and with government-run or public stations, the number of radio stations is much smaller, for example, only about 100 major stations in Germany. If all such stations were to be made available via Internet multicast at FM quality (56 kb/s), this would take 680 Mb/s. Since many of these stations broadcast identical programs, the likely number of channels transmitted nationwide is much lower. More realistically, offering the 31 channels of audio carried by the DirecTV satellite service would add only 1.7 Mb/s of Internet traffic, while the 45 FM channels in New York City would add 2.5 Mb/s.

Carrying radio services over the Internet allows improved directory services, easy addition of side information such as content labeling, bandwidth diversity for different kinds of programming as well as the reception of more diverse programming particularly in more sparsely populated areas. Labeling also makes it easy to have receivers assemble custom programs for listening at some later time ("time-shifting"). (Instead of constructing a special-purpose digital radio system it would seem to be preferable to have a general packet radio, with some fraction of the bandwidth set aside for distribution services, both audio and other content types.)

Similarly, given enough bandwidth, there are substantial advantages of distributing entertainment video, both broadcast and video-on-demand services, using Internet protocols, in particular MPEG over RTP [18]. This, in combination with an Internet-based stream control protocol such as RTSP [19], appears to be both more flexible and simpler than using MPEG as a new network protocol and DSM-CC as a control protocol.

2.4 Internet Access

The services just described impose new requirements on network access. The current residential Internet access architecture, based on modems, is unsatisfactory in several respects:

- The local phone switches were not designed as leased-line switches. While the average local voice call lasts 2 to 5 minutes [20] or 4.28 minutes according to [21], the average Internet call already lasts for 17 to 21 minutes [20], depending on the phone company surveyed. This has already lead to isolated inability to obtain dial tone in some central offices. As shown in

Table 2.3, if carried by the Internet, telephony and broadcast services can add several hours to phone line usage.

- The current standard flat-rate fee of US$19.95 for unlimited Internet access is based on a multiplexing model [22, 23] where 200–300 concurrent users share a T1 access line (costing about $3000 per month), for an average rate of 5 kb/s to 7 kb/s. A modem is provisioned for every 10–15 customers, depreciated at about $10/month, with ISP line charges of about $20. Again, with continuous use of the Internet for video entertainment, interactive games or telephony increases, Table 2.3 suggests that during the busy hours of the early evening, almost every subscriber will be trying to connect to the modem pool. Also, the average data rate will exceed the per-user rate.

  For example, a January 1997 study commissioned by AT&T Worldnet reports that between 6 pm and 9 pm, users manage to connect to a modem 93.4% of the time. Users calling AOL during the same hours, meanwhile, manage to connect just 36.7 percent of the time[24].

- For applications like Internet telephony, remote monitoring and control, the end system should be permanently powered up and connected to the Internet in order not to have to pre-arrange phone calls by email.

- Since data traffic, including fax, has exceeded voice traffic since 1995 [25], it makes sense to design the network for the predominant traffic.

It was hoped that ISDN might improve data access, as it avoids some of the signaling difficulties of analog POTS and offers higher speeds, yet configuration complexity, hardware costs, the necessity to maintain an analog line for communications during power outages, comparatively high tariffs, as well as the decreasing gap in speed between analog modems and a single ISDN basic rate channel have limited the appeal of ISDN, at least in the US. Clearly, ISDN does not address the congestion problem at the local office.

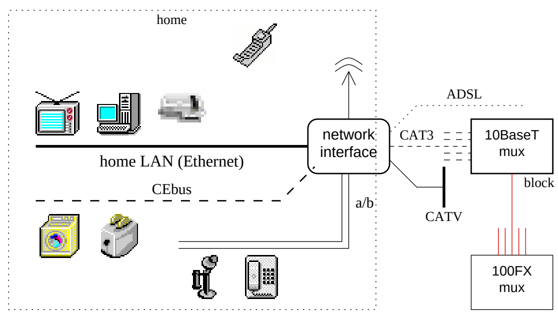
Table 2.4: Cost comparison of switches and routers

switching method	ports	capacity (Gb/s)	cents/kb/s	$/interface
10BaseT Ethernet hub	16	0.16	0.1	9.4
100BaseT Ethernet hub	16	1.6	0.05	46
10BaseT Ethernet switch	24	0.24	0.8	77
100BaseTX Ethernet switch	8	0.80	0.15	156
router		2.1	16.0	
local ATM switch	16	2.48	1.0	1581
PBX	256	0.02	218.	140
Lucent 5ESS local (no AIN)	5,000	0.32	469.	300
Lucent 5ESS local (AIN)	20,000	1.28	273.	175
Lucent 4ESS toll	45,000	2.88	1527.	

From an economical standpoint, common data switches are far more cost-effective for switching bits than either PBXs or traditional telephone switches. A rough estimate of the cost per kb/s and per port of switching is given in Table 2.4. On average, a PBX port costs about $567 in 1996, including the telephone set.

Note that costs for the 5ESS and other telephone switches are hard to quantify since the vendors do not generally release price lists and since much of the cost is in software upgrades. The 4ESS prices are based on the number of simultaneous phone calls, not the number of lines.

Figure 2.4: Residential Internet access



Given these considerations, traffic should be packetized as close to the end user as possible. Thus, a suitable architecture for residential access to the Internet should look similar to the current corporate LAN environment, as sketched in Fig. 2.4. The figure indicates three alternatives for access, namely ADSL, CATV [26], and Ethernet. The first two have been explored extensively elsewhere. However, for densely populated areas or apartment buildings, an Ethernet-based approach may be more cost-effective. Since the median loop length between network termination and central office of 1.7 miles (9 kft) (1973) [13, p. 328] exceeds twisted-pair Ethernet specifications, a direct connection between homes and the central office is not possible. Instead a more layered approach needs to be taken. For example, several dozen homes or apartments would be connected to an Ethernet switch or hub, located no more than the CAT-3 cabling distance limit of 328 feet from the network termination unit. The switches would, in turn, connect through fiber to the neighborhood switch. This architecture has the advantage that a mix of low-bandwidth and high-bandwidth customers can be accommodated without running additional wires. Since switch costs are dominated by interface counts rather than bandwidth, this mechanism offers much higher per-user bandwidth (particularly peak bandwidth), yet switching costs are similar to today's.

Ethernet appears advantageous as the local access technology since PC interfaces are cheap, operate over a variety of media and, unlike ATM, they allow easy addition of more devices on a multiple-access LAN.

Even with packet connectivity to the individual residence, easy connection of existing telephone equipment is imperative. Thus, Fig. 2.4 shows the network termination unit with a built-in a/b (two-wire) telephone interface. This could be readily implemented within a single DSP that would act as a simple packet voice module. It would also implement DTMF recognition for user-to-network signaling based on, say, SIP [10], described briefly below.

2.5 Protocol Architecture

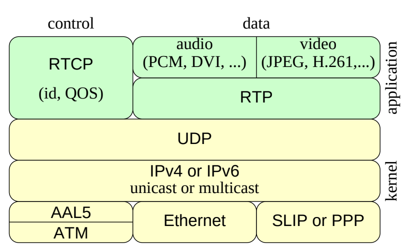
We now present an overall Internet protocol architecture that can support telephony and other continuous-media services such as “Internet radio” and “Internet TV” [27].

2.5.1 Data Transport

For transporting real-time data across the Internet, the accepted end-to-end protocol is the Real-Time Transport Protocol (RTP) [5, 28]. It is also used by the ITU-T H.323 teleconferencing recommendation. RTP is a thin protocol providing support for applications with real-time properties, including timing reconstruction, loss detection, security and content identification. An associated control protocol called RTCP provides support for real-time conferencing for large groups within an internet, including source identification and support for gateways (like audio and video bridges) and multicast-to-unicast translators. It offers quality-of-service feedback from receivers to the multicast group as well as support for the synchronization of different media streams.

While UDP/IP is its initial target networking environment, efforts have been made to make RTP transport-independent so that it could be used, say, over CLNP, IPX or other protocols. RTP is currently also in experimental use directly over AAL5 using native ATM services. This protocol stack is shown in Fig. 2.5. RTP is described in detail in Chapter **??**.

Figure 2.5: RTP-based protocol stack



For low-speed links and highly compressed audio, the combined stack consisting of IP, UDP and RTP add 40 bytes to every packet, while 20 ms of 8 kb/s audio only take up 20 bytes. Thus, just like for TCP, header compression is desirable [29].

As an end-to-end protocol, RTP cannot guarantee a certain quality of service. The resource reservation protocol, RSVP [30], might be used to allocate resources to either individual streams or a group of streams. Unlike in the telephone network, establishing connectivity and allocating resources are two distinct operations. Typically, the caller would first “ring” the callee. Only if the called party wishes to communicate and once a set of media has been agreed upon, would RSVP be used to reserve resources. This runs the risk of being denied the necessary network resources, but in any network dimensioned for telephone service, voice calls should be blocked extremely infrequently. Given that the resource needs are only known once the parties agree on the media and their quality, this order is also the only feasible one.

Since packet audio flows have a relatively low, constant bit rate, with deterministic arrival patterns, it may make more sense to treat all voice calls on a link as a single stream for scheduling rather than managing several thousand individual reservations, with the attendant state space and refresh overhead. As indicated earlier, it may be sufficient to simply give this class of service priority. In some cases it may also be advantageous for the end systems to reserve only the minimum necessary bandwidth, and obtain additional throughput and improved audio quality through best-effort flows.

2.5.2 Signaling

In the architecture described here, the traditional telephony signaling protocols, Q.931 for ISDN user-to-network signaling and SS7 for network-to-network signaling, are replaced by a single, much simpler signaling protocol. One such solution, the session initiation protocol (SIP) [31], can establish multimedia conversations with one or more parties. Instead of telephone numbers, it uses addresses of the form user@domain or user@host. In many cases, this address would be identical to a person's email address. Table 2.5 compares the properties of email addresses to standard telephone numbers.

Table 2.5: Comparison of (U.S.) telephone numbers and email addresses

feature	phone number	email
mnemonic	no (except for some 1-800)	name@organization
multiple per person	no	easy
avg. characters	≈ 12	22
location-independent	1-700	yes: j.doe@ieee.org
carrier $\neq$ naming	maybe	yes
directory	411, 1-555, switchboard.com	LDAP [32]

(Note: average email address size is estimated from the IEEE ComSoc TCCC mailing list, an international list with about 500 members.)

SIP offers the standard PBX, ISDN [33] or CLASS functionality, including call forwarding, call waiting, caller ID, call transfer, camp-on[4], call park [5], and call pickup [6]. Many of these features actually require no signaling support at all, but can be implemented by end system software. SIP is designed as a variant of HTTP/1.1[35], which allows easy reuse of HTTP security and authentication, content labeling and payment negotiation features.

We used the SIP features to implement a calendar-based call handler. The call processing software accesses a user's personal appointment calendar and answers the phone accordingly. The user can define categories of callers and preset, based on the calendar entry, whether and where

[4]"Camp-on allows an attendant-originated or extended call to a busy single-line voice station to automatically wait at the called station until it becomes free while the attendant is free to handle other calls." [34]

[5]"Allows user to put a call on hold and then retrieve the call from another station within the system". [34]

[6]"Allows stations to answer calls to other extension numbers within the user specified call pickup group" [34]

their calls are forwarded. The information released to the caller if calls are not forwarded may range, for example, from “is currently not available” to “John Smith is in a meeting until 3 pm in room 5621 with Jane Doe”, depending on the caller’s identity. In the near future, this will be integrated with the call processing language, a state-based scripting language that allows to construct voice-mail systems or automatic call handling systems in a few lines of code. It also manages the translation between ISDN calls and Internet telephony calls.

2.6 Reliability

Clearly, if Internet technologies are to play a larger role in providing telephony service, they have to offer at least the same reliability and manageability as the older circuit-switched technologies they are to replace. The target is set fairly high: a typical local telephone switch is out of service for about 120 seconds a year, of which 25% are for outages less than 2 minutes. On average, a subscriber can expect his or her line to be unavailable for 85 seconds during the year, of which 34 seconds are scheduled outages, where only outages of more than two minutes are counted.[7]

Unfortunately, only limited reliability information is available for online and Internet services. America Online reports [36] scheduled and unscheduled outages of 1% or 88 *hours* a year for 1996, down from 3.5% or 307 hours the year before.

A more packet-oriented architecture, as presented earlier, would remove this restriction, as well as modems that are sometimes subject to locking up or not releasing a line. Typical local network switching equipment has actualized MTBFs of 170,000 to 210,000 hours (19 to 24 years); from anecdotal evidence, hubs and Ethernet switches rarely fail as a whole, rather, individual interfaces may.

Recent reported large-scale Internet outages were due to either misconfiguration, as in the case of the AOL BGP router collapse in 1996, or a local power failure without adequate uninterruptable power supplies, as when a major POP on the Stanford University campus was out of service for a better part of a day. Many of the Mbone routing failures are due to router misconfiguration, for example injecting all unicast routes into the multicast routing protocol. Also, it is clearly harder to maintain telephone-level uptime when traffic doubles every few months and host counts double yearly.

However, there are a number of obvious improvements that are necessary for Internet services to approach telephone-level reliability:

- Software upgrades should be possible without taking down a router or switch.
- Router configuration must be made simpler, with checking against local rules that make catastrophic failure less likely.

[7] These numbers are drawn from the FCC quality of service reports for 3Q93 through 3Q95 and are for USWest, one of the regional Bell operating companies (RBOCs) with about 13.8 million access lines. It appears that the difference between the per-switch and per-line figures largely explained by the two-minute threshold. An average switch serves 8,600 lines.

- Closer integration of network functionality into the operating system should reduce end-system difficulties. Much of the complexity of configuring current PCs for Internet usage appears to stem from having to configure the modem and multiple protocol stacks. Widespread use of DHCP [37] and IPv6 autoconfiguration [38], as well as eliminating the modem, should make the network invisible.

- While backbone networks feature redundant links, POPs and access links are often single point of failures. Also, different providers often peer at only a small number of points. Particularly the latter problem must be remedied to increase the number of true end-to-end alternate paths.

In the long run, tools like traceroute and ping, as well as relatively simple management protocols (SNMP) and applications with built-in reporting mechanisms such as provided by RTCP probably make Internet service more manageable than many traditional POTS installations.

A particular problem with packet telephony is the need for power at the end systems, and, if the architecture of Fig. 2.4 is adopted, at the various multiplexing points within the network. Fortunately, end systems can be build so that they consume almost no power in standby mode and should be able to function on a small rechargeable battery for days, given the current operating times of small cellular phones. Similarly, a typical 24-port Ethernet switch consumes about 30 W of power, so that it can be operated with a typical 1 kWh sealed lead-acid battery for more than 30 hours.

2.7 Summary

Internet services have the promise of being the foundation of an integrated services, packet-switched networks that delivers not only web pages and email to its users, but also replaces parts of the telephone system or cable television. Within the last few years, many of the necessary protocols and architectures have emerged to realize this vision. However, the most important factor may not be protocols, but new residential access methods, increased reliability and sufficient backbone capacity.

2.7.1 Resources

The Internatioal Telecommunications Union (ITU) (www.itu.int) publishes an annual survey *ITU Direction of Traffic: Trends in international telephone tariffs* that summarizes volume and growth of telephone traffic. The ITU web site contains basic indicators such as population, GDP, main telephone lines and main lines per 100 people.

For the United States, the Industry Analysis Division of the Common Carrier Bureau within the Federal Communications Commission (FCC) publishes the *Trends in Telephone Service*, available through the FCC web site (www.fcc.gov). It contains detailed statistics on call volumes, tariffs and historical trends.

2.8 Problems and Experiments

- The United States has about 200 million people above the age of 12. Assume that each teenager and adult makes about an hour of long distance and an hour of local calls per day. Compute the total bandwidth needed to carry these conversations assuming that each phone call takes a bandwidth of 32 kb/s. Why is this computation not good enough to design a network?

- Experiment with an existing Internet telephony application. Can you measure how long it takes to set up a call? What is the delay? (Hint: You need two workstations in the same room for this experiment. You might find a stereo cassette recorder handy that can record audio from both the sending and receiving side. Alternatively, you can use a sound recorder tool on a third workstation to record both.)

- Find out (e.g., from manufacturer's web pages) how much power a typical PC, laptop and Ethernet hub consume. A typical 12-Volt car battery has a capacity of XXX Ah or XXX Wh. In other words, this battery can sustain a load of XXX W for XX hours or a load of XXX W for XX hours. How long could you run the PC and other equipment on that battery? Find out how long typical office UPS (uninterruptible power supplies) last. What are the implications for Internet telephony?

- Can you suggest how E911 (enhanced emergency) service might work for Internet telephony? In this service, the dispatcher can see which address you are calling from. Depending on the address, a different 911 calling center will be contacted. What are the difficulties for dial-up users?

- Find out how many outside phone lines your workplace has and estimate how many minutes a day each employee makes or receives outside phone calls. In a typical PBX, about a third of calls are internal, with the remainder split evenly between incoming and outgoing external calls. A typical business line on a PBX is in use 10 minutes of the hour. This is referred to us a load of 6 CCS. CCS is a measure of telephony traffic representing hundred call seconds per hour. How much does your workplace differ?

- In many countries, you can choose a long-distance telephone provider or even use a different one for each phone call by dialing a prefix (10 and a three-digit number in the United States). How could one implement a similar service for Internet telephony?

- Find out if your Internet service providers reveals or guarantee its reliability. If so, is reliability measured?

- Design an experiment that tests reachability from your workstation to a group of sites by periodically "ping"ing it or running `traceroute` to get more details. To reduce traffic, you should only run the test every few minutes. In your experiments, do you find that connectivity problems are local, in the middle of the network or at the destination?

Appendix A

Projects

Below, we describe projects that are suitable for classes taught using this book. In combination or by requiring additional literature research, analysis or measurements, they can also be used for undergraduate and Masters research projects. Some projects require special hardware resources, but a subset can usually be implemented by simulating the interface. Most of these projects can be done on any computer platform, including Unix workstations or 32-bit Windows and NT versions. Most projects can be programmed in Java, C/C++. Some parts lend themselves to scripting languages like Tcl/Tk. Commercial products are mentioned as examples; in almost all cases, there are equivalent products available from other vendors. A number of projects entail modifications of existing audio and video tools. The RTP web resources (www.cs.columbia.edu/˜hgs/rtp) enumerates a number of such tools that are available as source and run on most common operating system platforms. The web pages accompanying the book contain updated resources for these projects.

A.1 Internet Telephone Gateway

There are a number of devices that convert the a/b interface of an analog phone line to line-level audio inputs and outputs, e.g., the T-311 by Teltone.

Construct a software driver to connect the analog phone line interface to an IP telephony server, allowing dial-in and dial-out. A basic version connects every incoming phone call to a fixed Internet address and allows outgoing calls to any phone number.

For group projects, this gateway can be extended so that a phone caller can "dial up" an Internet address. The phone answers "*Welcome to the Internet telephony gateway service. Please enter the email address or extension, using * for the '@' sign*". The user then "types" in the beginning of a name or email address using the phone's numeric keys (447 for hgs, for example), with the text-to-speech facility offering the list of ambiguous choices. You should be able to resolve both ambiguous user names (e.g., from the `/etc/passwd` file) and ambiguous host names (e.g., by acquiring a zone dump of the .com entries).

Consider also the restriction of outgoing phone calls. Your system must handle the case when no outgoing phone line is available.

A.2 Web by phone

Using the telephone gateway described in Project A.1, build a web-by-phone server that recognizes touch-tone (DTMF) digits and then reads a web page aloud using text-to-speech software such as the Bell Labs TTS system (www.bell-labs.com). If necessary, the project can be implemented without the gateway hardware on any system with audio I/O: the microphone input replaces the audio channel from the phone, while the loudspeaker output produces the acoustic prompts normally heard through the telephone. A DTMF generator comes in handy for testing; it is available for a few dollars from your local electronics store. The freely available Lynx textual web browser makes a good tool for pre-processing web content for reading aloud, particularly in its configuration for blind users.

A.3 Email by phone

This project is similar to the previous one, but instead of web pages, its goal is to read and respond to email through the phone. The project should support navigation through message folders. The server can either retrieve messages into a file or access a remote email server while leaving the messages on the mail server. The `movemail` program that is available, for example, for the Netscape browser or the POP protocol [**?**] can retrieve messages into a local file. IMAP [] manages a remote message store. The program should be able to read subject lines and the sender of the email. When the message is being read, the caller should be able to stop, back up and cancel the reading. Possible extensions include the ability to only read messages marked as "urgent" in the subject line or those personally addressed to the recipient rather than to a mailing list. Additional features include the ability to respond to a message, using a MIME [] voice attachment recorded through the phone, and to play back messages containing an audio attachment.

A.4 Multimedia jukebox: catalogue

A CD ROM jukebox is a mechanized storage system for CD ROMs. A robotic arm moves CDs into one or more CD readers or writers. If you have such a CD ROM jukebox available. Using existing low-layer SCSI interface, write a program that catalogues the jukebox CD ROMs into a line-per-CD text file or a web page. With that, write a program that locates and loads CDs by title. Allow the user to associate a particular waveform signature with an audio CD, so that CDs can be found after having been added and removed.

A.5 Multimedia jukebox: audio CDs

Using the jukebox described in the previous project and available raw-disk reading routines, write a program that reads the data from the disk and serves it using an existing RTSP media server (see the book web page).

A.6 Conversion of AVI to H.261

Using the H.261 codec in `vic` and the cinepak codec in the `xanim` program, convert standard AVI files to H.261, to allow streaming across the network using the RTSP server.

A.7 MPEG payload format handler

Write a module that splits an MPEG file into individual frames and wraps each into the necessary RTP payload header (see RFC xxx). Integrate into the RTSP media server for generating streaming MPEG.

A.8 Integration of LPC codec into audio media agent

Integrate the LPC-10 low-rate audio codec by Andy Fingerhut, Washington University into NeVoT, the author's network voice terminal. If you have some signal processing background, you might attempt to optimize. Evaluate the processing requirements and audio quality, as a function of loss and packet size and compare it to the current LPC codec found in `vat` and `NeVoT`.

A.9 Integration of MPEG Layer-3 audio codec

Integrate the MPEG layer 3 (www.iis.fhg.de/audio) high-quality audio codec into NeVoT or similar audio tool.

A.10 Subjective performance of audio codecs

Evaluate the subjective performance of an audio codec using the diagnostic rhyme test (DRT) (Section **??**), A-B comparison or other technique. Compare to the measured SNR.

A.11 Impact of packet loss on codecs

Measure how long it takes for an audio codec such as G.723.1, G.729, DVI or GSM to recover from packet loss: compare the evolving codec state with and without packet loss and count the number of packets until the output has converged again to the same value as without packet loss.

A.12 Recording function for audio and video media agents

Add a function to an audio or video agent that records incoming and outgoing packets into an rtpdump file. This is useful when recording events for later re-broadcast.

A.13 Screen grabber for video media agent

Add functionality that allows to grab screen or single window into a video media agent. This greatly improves the quality of sending PowerPoint or Adobe Acrobat reader slides compared to doing scan conversion (see glossary) and then converting back into digital format again. Consider using GIF compression. Design an RTP payload format that ensures that large GIF frames are properly segmented across packets. Consider whether to periodically re-transmit parts of the image to compensate for packet loss.

A.14 RTP translator

Using the RTP library (www.cs.columbia.edu/˜rtp), build an RTP translator to allow configurable translation between audio formats, using the audio encoding libraries contained in NeVoT. For example, an incoming high-bitrate stream using PCM (μ-law) encoding could be translated to GSM.

A.15 RTP aggregator

Write an RTP translator that aggregates several RTP media streams into a single aggregate stream (see Sec. **??**).

A.16 Integration of RSVP into a media agent

Integrate RSVP capabilities into NeVoT or other multimedia agent. You can either install and use the existing Solaris or FreeBSD implementation or, more ambitiously, write an RSVP client that implements the RSVP protocol on top of UDP in user space. This client only needs to generate and understand RSVP messages, not perform local packet scheduling or admission control. Consider integrating a dynamic scheme so that the media agent attempts a reservation at a lower rate if the reservation fails.

A.17 Media agent reliability enhancements

Add packet-loss compensation to an audio or video media agent, using forward error correction (FEC) mechanisms and lower-fidelity delayed versions. Measure the efficiency of the FEC in the face of real Internet loss traces and compare the perceived quality.

A.18 VCR interface for audio/video tools (RTSP client)

Build a Java or Tcl/Tk user interface that implements "VCR" functionality for on-demand access to stored audio and video using RTSP. The user interface contains the standard VCR control buttons (record, play, pause, rewind, fast forward), as well as a slider or similar element for positioning (by elapsed time) within the movie. The audio and video stream are delivered by an existing RTSP server; streams are played back using standard audio and video media agents.

A.19 Java RTSP server

A.20 Audio stereo placement

Enhance an audio media agent to allow placing speakers acoustically at different parts of the stereo space, using, for example, delay and channel balance.

A.21 Measurements and comparisons of audio and video codecs for networks

Evaluate and compare the performance of audio and video codecs both through objective and subjective measurements. Include the performance for random and correlated packet losses, including loss traces collected from the Internet.

Prepare and present a wide variety of standard audio and video sequences, showing different qualities and the effects of impairments, to be used for comparison (A/B tests) and teaching.

A.22 Loss Measurement Tool

Create a tool that combines traceroute and ping to indicate where in the path between source and destination packets are lost and delayed. Present results graphically and try your tool on some domestic and international routes.

A.23 H.263 implementation

Based on existing H.261 implementations, implement an H.263 video codec for a video agent.

A.24 Clock measurements

Measure clock accuracy and drift of audio and video clocks on workstations and PCs.

A.25 Multimedia server

Develop a module for the RTSP media-on-demand server that reads and writes Microsoft ASF audio and video files.

A.26 Digital editing of live content

Design and develop one or more modules for creating digital effects for live video. Effects include title overlay, wipes, fades, insert or chroma keying.

A.27 Classroom question manager

Design and develop an application that supports ”raising your hand” in a distributed, Internet-based classroom. The mechanism should scale to very large classes, with hundreds of participants. Thus, instead of a central server, multicast UDP should be used. Students should be able to indicate the nature of the question so that an instructor can group or delay questions. An instructor should be able to call on students out-of-order and cancel questions. Calling on a student enables that student's audio and video transmission. Other students should, in general, be able to see the list of waiting students. Consider using the PMM (pattern-matching multicast) enabled audio and video applications NeVoT and vic or NeViT. Your application has to deal with packet loss, but can assume that the computer clocks of all participants are synchronized, so that there is no ambiguity as to who raised her hand first. This application could also be useful for meetings, “Internet TV” talk shows with audience participation and town-hall-style gatherings.

A.28 RTP conformance tester

Develop a tool and test suite that tests RTP compliance and allows to “stress test” RTP implementations. Consider tests for loops, collisions and invalid packet formats (e.g., wrong SDES lengths).

A.29 RTP-based audio mixer

Currently, only a single application can use the audio device. Based on the NeVoT library, develop a daemon that multiples requests for an audio device. Modify xplay and playtool to use this to play audio files and/or develop a Netscape plug-in that plays audio files using this mechanism or direct write. This could also serve as the basis for a remote audio capability similar to the X window system.

A.30 Background eliminator

Construct a system that only transmits a (moving) person, without the background and allows to reassemble a virtual audience at the receiver. Re-use the video codecs already available, such as H.261.

(Some image processing background is helpful for this project.)

A.31 Remote-control web and conferencing camera

There a number of conferencing cameras (such as the Canon VC-C1) that have a motorized zoom, pan and tilt which can be controlled remotely through a serial port. Serial port interfaces can be written in C or Tcl. As part of the project, write a generic control library in C or Tcl that allows to control the camera. Once a basic control interface API has been written, there are several choices for functionality:

- Implement a forms and cgi web interface using Perl or Tcl, to retrieve images (*web cam*). Accomodate several "concurrent" users. If an image for a particular position has been acquired recently (within a setable interval), return it immediately, otherwise, queue the request and serve requests in order or using motion optimization. The web cam visitor should be able to click on a particular part of the image to zoom in or re-center the camera.

- Live camera control: Create a Tcl/Tk application that allows a camera operator to position the camera for a live feed (e.g., to tape a lecture). The video is distributed to the network via a video media agent. The Tcl/Tk application should allow both relative ("tilt down 15 degrees") and absolute positioning ("pan to 5 degrees left-of-center"). It should be possible to store the current setting by name, so that one can quickly zoom in on a student in a class, for example. Extension: The camera operator points the mouse to a point of the existing image and the camera either zooms in or re-centers on that point. (This may require modifying the media agent to recognize mouse clicks in the displayed image.)

- Integrate camera control into the existing media server, using the GET_PARAMETER and SET_PARAMETER RTSP commands.

A.32 Directory Services for Internet Telephony

Compare the existing directory services for Internet telephony (Netscape/Insoft IS411, Vocaltec, Four11, Microsoft ULS, ...). Some are documented, some need to be reverse-engineered. Consider distributed alternatives that allow sub-grouping and graphical representations (rooms, maps, VRML?). Install an implementation of LDAP and construct an interface to a SIP server.

A.33 "Buddy List"

Using SIP, implement a "buddy list", where a user can see who of her friends are logged on, active (i.e., with keyboard activity) or in some other user-defined state at the time. It should be easy to invite a subset of your "buddies" to a multimedia conference.

A.34 Calendar interface for Internet telephone

Write an interface to a calendar program (such as Schedule+, Sun calendar manager (cm), the Unix Common Desktop Environment (CDE) calendar, a vcalendar compliant program, ...) that automatically forwards or answers calls ("I'm in a meeting until 4:30", "I'm on vacation until August 26"). Allow the calendar owner to define keywords that govern behavior, e.g., "private" may disable forwarding calls. Use existing privacy indications, such as "Show Time and Text", "Show Time Only", "Show Nothing" to govern the amount of detail to be provided to the caller; allow configuration based on user groups. For example, the user might configure the application so that all callers from cs.foo-state.edu are shown time only, while a select group of people identified by their email addresses are shown full details. Everybody else is shown nothing (available/not available). You could use an addressbook to look up forwarding details. (For example, "meeting at Jane Sullivan" would look up her phone number or email address for forwarding the call.)

Your program should be able to deal with overlapping appointments. For example, if the user is traveling to LA and has a meeting there, indicating the end of the meeting may not be particularly interesting, but rather the time to call back would be the time of return from the trip.

Consider parsing the calendar entry to understand header fields, e.g., a Location: line would indicate the appropriate forwarding location.

The application should be called as

busy *caller*

The application should return messages in the format of SIP cgi responses, for example:

```
Status: 480 Meeting with John Doe
Location: j.doe@cs.bar.edu
Retry-After: Mon, 9 Feb 1998 17:37:17 +0100
Content-Length: 0
```

If there is currently no meeting scheduled, return 200 OK.

A.35 Scheduling of repeated events

Develop a language that expresses sequences of events like daily newscasts, (almost) weekly lectures and the like in a space-efficient manner. Take into account daylight savings time shifts.

A.36 Bandwidth estimation and measurement

Based on sample packets (ICMP, UDP/RTP, or TCP) or the received data stream, attempt to estimate the bottleneck bandwidth for point-to-point links. Estimate the bandwidth distribution for multicast groups.

A.37 RTCP for network management

Add RTCP capability to the tkined network manager and display tool, to allow monitoring of reception quality in a local or wide area network.

A.38 Service differentiation for data and real-time

Investigate options, such as priority bits, to differentiate data and real-time services without using resource reservations.

A.39 Call controller

Using Java or Tcl/Tk, implement a SIP call controller. The program should support both calling and called party using SIP for signaling. Your project should be modular, consisting of a parser and generator for SIP requests and responses, a module for parsing and generating SDP descriptions, and a user interface for initiating and receiving calls. The call controller starts the necessary applications (media agents) to receive and send audio and video. The controller also terminates these applications (by sending them an appropriate signal). You can use the media agents described on the RTP page, e.g., NeVoT, vic, rat and vat. Initially, it is sufficient to simply start these agents, with parameters supplied from the command line. For example, to set up a video connection to port 3456 at `foo.bar.com`, the vic video tool is started as `vic foo.bar.com/3456`.

A.40 H.323 Implementation

Implement (a subset of) H.323v2 (Section **??**, the Internet telephony standard. H.323 is supported by Intel Internet Video Phone and Microsoft NetMeeting, among others, but only on Windows'95.

The project only has to worry about the signaling aspects, since there are existing RTP tools that can exchange data with these H.323 implementations; they just cannot set up a call.

The project might start by implementing routines that parse and generate the Q.931 messages described in H.225.0 and then proceed to parse and generate the ASN.1 (PER) encoding.

A.41 Network reliability

Design and implement a network reliability monitor. Test reachability of a number of sites periodically (e.g., via ping every ten minutes). If a site is not reachable, try to determine, using traceroute, whether this is a local problem or a problem affecting only one site or a problem affecting an Internet backbone.

A.42 Audio and video delay measurements

Using the phone system or a radio as a reference, measure the end-to-end delay of both commercial and research network audio (and maybe video) tools on different platforms. Determine whether delays are primarily due to the network, operating system and sound hardware or application. There are several possible approaches to do these measurements. In the simple case where receiver and sender workstation are in the same room, it suffices to send an on-off waveform and measure the delay of the "on" or "off" edge between sender and receiver. Record the sender input and receiver output onto two channels of a stereo tape deck or PC sound recorder tool and visually or automatically measure the delay based on inspection of the waveform.

When sender and receiver are separated, use a radio or telephone to feed the same audio signal to the sender and the stereo recorder. Since the codecs will distort the signal, investigate the use of correlation analysis to estimate the delay. When using a telephone, a white noise source can be expected to work best.

A.43 Delay sensitivity

Set up a human-factors experiment that introduces artificial delay between two test subjects that can only communicate via Internet telephony. Have the subjects evaluate the perceived quality as a function of the delay. Using a silence detector, measure the distribution of the length of talkspurts as a function of end-to-end delay. Count the occurences of double talking, i.e., where both subjects are talking at the same time. Consider consulting a student from the psychology department at your school for guidance on experimental design and a sample coordination task.

A.44 VRML

Experiment with a communicative VRML space. As you approach somebody in 3-D space and turn towards him, you can hear and see that person more clearly. Consider using multicast groups for communications.

Bibliography

[1] A. H. Wurtzel and C. Turner, “Latent functions of the telephone: what missing the extension means,” in *The Social Impact of the Telephone* (I. de Sola Pool, ed.), pp. 246–261, Cambridge, Massachusetts: MIT Press, 1977.

[2] G. Gillespie, “Warning: Too much info hazardous to your health,” *IEEE Institute*, June 1997.

[3] Times Mirror Center for The People and The Press, “Survey about computer use,” Oct. 1995.

[4] C. Malamud and M. Rose, “Principles of operation for the TPC.INT subdomain: Remote printing – technical procedures,” Request for Comments (Experimental) 1528, Internet Engineering Task Force, Oct. 1993. (Obsoletes RFC1486).

[5] 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.

[6] C. Strathmeyer, “Feature topic: Computer telephony,” *IEEE Communications Magazine*, vol. 34, Apr. 1996.

[7] Versit Consortium, “Computer telephony integration (CTI) encyclopedia,” Tech. Rep. Release 1.0, Versit Consortium, Oct. 1996.

[8] C. Low, “The internet telephony red herring,” in *Proceedings of Global Internet*, (London, England), pp. 72–80, Nov. 1996.

[9] C. Low, “Integrating communications services,” *IEEE Communications Magazine*, vol. 35, pp. –, June 1997.

[10] M. Handley, H. Schulzrinne, and E. Schooler, “SIP: Session initiation protocol,” Internet Draft, Internet Engineering Task Force, Dec. 1996. Work in progress.

[11] P. Moulton and J. Moulton, “Telecommunications technical fundamentals,” technical handout, The Moulton Company, Columbia, Maryland, 1996. see http://www.moultonco.com/semnotes/telecomm/teladd.htm.

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

[13] R. F. Rey, ed., *Engineering and Operations in the Bell System*. Murray Hill, New Jersey: AT&T Bell Laboratories, 1984.

[14] Industry Analysis Division, Common Carrier Bureau, "Trends in telephone service," tech. rep., Federal Communications Commission, Washington, D.C., May 1996.

[15] L. Zuckerman, "America Online moves to placate angry users," *New York Times*, Jan. 1997. CyberTimes.

[16] Emerging Technologies Research Group, "The american internet user survey," survey, Find/SVP, New York, Oct. 1996. Reported in *Tampa Tribune*, January 12, 1996.

[17] D. Lavin, "Tailoring new products is becoming more complex," *Wall Street Journal*, Aug. 1996. European Technology Section.

[18] D. Hoffman, G. Fernando, and V. Goyal, "RTP payload format for MPEG1/MPEG2 video," Request for Comments (Proposed Standard) 2038, Internet Engineering Task Force, Oct. 1996. (Obsoleted by RFC2250).

[19] H. Schulzrinne, "A real-time stream control protocol (RTSP')," Internet Draft, Internet Engineering Task Force, Dec. 1996. Work in progress.

[20] Bell Atlantic, "Report of Bell Atlantic on internet traffic," FCC report, Bell Atlantic, Mar. 1996.

[21] M. Mayer, "The telephone and the uses of time," in *The Social Impact of the Telephone* (I. de Sola Pool, ed.), pp. 225–245, Cambridge, Massachusetts: MIT Press, 1977.

[22] D. H. Dennis, "The inet-access frequently asked questions list," Dec. 1996.

[23] R. Stiennon, "So you want to be an internet provider?," *TechKnow Times*, vol. 2, May 1996.

[24] D. Lazarus, "AOL refugees call on ATT WorldNet," *Wired News*, Jan. 1997.

[25] A. Dorman (Pacific Bell), "Simba newswire," Mar. 1995.

[26] C. Bisdikian, K. Maruyama, D. I. Seidman, and D. N. Serpanos, "Cable access beyond the hype: on residential broadband data services over HFC networks," *IEEE Communications Magazine*, vol. 34, pp. 128–135, Nov. 1996.

[27] H. Schulzrinne, "Internet services: from electronic mail to real-time multimedia," in *Proc. of KIVS (Kommunikation in Verteilten Systemen)* (K. Franke, U. Hübner, and W. Kalfa, eds.), Informatik aktuell, (Chemnitz, Germany), pp. 21–34, Gesellschaft für Informatik, Springer Verlag, Feb. 1995.

[28] H. Schulzrinne, "RTP profile for audio and video conferences with minimal control," Request for Comments (Proposed Standard) 1890, Internet Engineering Task Force, Jan. 1996.

[29] S. Casner and V. Jacobson, “Compressing IP/UDP/RTP headers for low-speed serial links,” Internet Draft, Internet Engineering Task Force, Nov. 1996. Work in progress.

[30] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, “RSVP: a new resource ReSerVation protocol,” *IEEE Network*, vol. 7, pp. 8–18, Sept. 1993.

[31] M. Handley, H. Schulzrinne, and E. Schooler, “SIP: Session initiation protocol,” Internet Draft, Internet Engineering Task Force, Feb. 1998. Work in progress.

[32] 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).

[33] G. C. Kessler and P. V. Southwick, *ISDN: concepts, facilities and services*. New York, New York: McGraw-Hill, 1997.

[34] G. W. Weber, “Centrex feature comparison for small business,” memorandum, AT&T, San Ramon, CA, July 1993. see also http://www.first-tel.com/ltlfeat.htm.

[35] R. Fielding, J. Gettys, J. Mogul, H. Nielsen, and T. Berners-Lee, “Hypertext transfer protocol – HTTP/1.1,” Request for Comments (Proposed Standard) 2068, Internet Engineering Task Force, Jan. 1997.

[36] D. Lazarus, “AOL’s email system crashes twice in one day,” *Wired News*, Jan. 1997.

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

[38] S. Thomson and T. Narten, “IPv6 stateless address autoconfiguration,” Request for Comments (Proposed Standard) 1971, Internet Engineering Task Force, Aug. 1996.