The Royal Institute of Technology
Sweden

SIP Telephony Gateway on DTM

by
Mattias Eriksson
and
Lars Lundstedt

Examiner
Inge Jovik
The Royal Institute of Technology, Haninge

Supervisor
Michele Mizzaro
AU-system

Supervisor
Thomas Lindh
The Royal Institute of Technology, Haninge

Supervisor
Jakob Ellerstedt
Dynarc

Bachelor's thesis **1999-07-02**

AU-SYSTEM

Preface and acknowledgement

We would like to thank Team Datakom at AU-system, especially Michele Mizzaro our supervisor for his great support and dedication. We also would like to thank Jakob Ellerstedt and Olof Hagsand at Dynarc for providing DTM and Taurus for this thesis, and for their patience with our questions.

Abstract

In the near future, if you make a telephone call, it's quite likely that the Session Initiation Protocol (SIP) will be what finds the person you're trying to reach and causes their phone to ring.

The major part of this thesis has been to set up a test configuration for SIP and DTM. The test configuration includes 2 PCs with SIP user agent, 2 analog telephones, 1 PC with SIP Server/Gateway and 1 DTM router. A lot of programming had to be done for setting up the test configuration. For example we have written two own 'simple' SIP implementations.

This report also describes the Session Initiation Protocol and Dynamic synchronous Transfer Mode (DTM). SIP is a signaling protocol for establish, modify and terminate sessions, these sessions includes Internet telephone calls. DTM technology makes it possible to dynamic allocate bandwidth.

Sammanfattning (Swedish abstract)

Om du i framtiden ringer ett telefonsamtal, är det förmodligen SIP som finner den du vill nå och gör att telefonen ringer.

Den största delen av det här examensarbetet har varit att bygga upp ett testnätverk för SIP och DTM. Testnätverket innehåller 2 datorer med SIP user agent, 2 analoga telefoner, 1 dator med SIP Server/Gateway och en DTM router. En hel del programmering krävdes för att sätta upp testnätverket. Till exempel har vi skrivit två enkla SIP implementationer.

Den här rapporten beskriver också SIP och DTM. SIP är ett signaleringsprotokoll för att etablera, modifiera och terminera sessioner, dessa sessioner inkluderar Internet telefoni samtal. DTM teknologi gör det möjligt att dynamiskt allokera bandbredd.

AU-SYSTEM

Table of Contents

1. Purpose and background

Telephony and data have always lived separately, until today. The border between voice and data is decreasing rapidly, bringing them together will bring them enormous benefits.

This thesis examines how SIP could be integrated into a DTM network, regarding Quality of Services and handling of dynamic resource allocation. Currently several standards for telephony signaling exists, including SS7 for traditional telephony and SIP and H.323 for IP telephony.

SIP is a IETF Proposed Standard protocol. H.323 is a heavy ITU standard, while SIP has the ambition of being a "simpler" standard adopted to IP networks.

1.1 The purpose of this thesis

AU-system and Dynarc provided this thesis. The interests from these two companies for this commission differs. AU-system's main interest have been to get knowledge about SIP, while Dynarc wanted us to allocate bandwidth on their DTM router DS 100 with SIP.

For us (the participants) this thesis is the last step to graduate from the Royal Institute of Technology, the electrical engineering program at KTH-Haninge. The third year we concentrate our studying on Tele- and data communication. So, there was no coincident that we found this thesis very interesting.

1.2 Outline of the report

This heading conceals a short description of the contents of this report.

Chapter 1	The purpose and background for this thesis.
Chapter 2	Serves as an introduction to SIP, DTM and the surrounding environment.
Chapter 3	The implementation methods and result are presented.
Chapter 4	Conclusions.
Chapter 5	Bibliography and references.
Appendix	Information about products and applications, copyright statements and the thesis specification.

2. Introduction

2.1 IP-telephony

IP-telephony or Voice over IP (VoIP) is for the most people a rather new phenomena, and there are also two definitions of VoIP. The first one means that you use Internet for direct voice communication. The second one means that you use the IP-protocol on optional data nets with the same purpose for example LAN-telephony.

So far has the first technique been used by enthusiasts trying to call to other countries for a smaller cost without demands on the voice quality. Often a PC is used with a microphone, soundcard, speaker, and a VoIP-application. The disadvantage with the first definition is that no Quality of Service (QoS) is guaranteed.

On the Internet and in other networks, QoS is the idea that transmission rates, error rates, and other characteristics can be measured, improved, and, to some extent, guaranteed in advance. QoS is of particular concern for the continuous transmission of high bandwidth video and multimedia information. Transmitting this kind of content dependably is difficult in public networks using ordinary "best effort" protocols.

If VoIP will be interesting as a commercial service in bigger scale, it must work as good as a regular phone call. To achieve this there is a need for QoS. It must also be possibly to reach everybody without consideration of the receiving type of subscription (IP or PSTN).

There are four different types of connections for setting up the call. In all of the cases of VoIP, the UDP and RTP protocols are used. This means that the application handles the end-to-end communication without any guarantees from the net. The four different types are

1. PC to PC
2. PC to Telephone
3. Telephone to PC
4. Telephone to Telephone

4.1 Regular phones connected to PSTN

4.2 IP-telephones connected to a data net

2.2 Voice Encoding

The analog data is converted to digitalform with a codec. If the digital samples is to represent an analog waveform, the sampling rate must be at least twice the highest frequency present in the analog signal (Nyquist theorem).

Since the PSTN telephone networks eliminate frequencies over 3,3kHz, the analog speech signals are sampled at 8 kHz for Pulse Code Modulation (PCM) applications.

G.711 is the international standard for encoding telephone audio on a 64 kbps channel. It is a PCM scheme operating at 8 kHz sample rate, with 8 bits per sample. There are two different variants of G.711: A-law and mμ-law. A-law is the standard for international circuits, mμ-law is the US standard. The benefit of this kind of

techniques is that while maintaining a high quality, the amount of bandwidth is significantly reduced.

2.3 Architecture and protocols

A protocol is often defined as a set of agreements necessary to exchange data between two nodes connected directly to each other or connected to a network. A protocol is only one module in a structure of modules, together the modules sets up the protocol architecture (see Figure 1). An explanation of the two most common protocol architectures (OSI and TCP/IP) will be described below. TCP/IP is the most widely used interoperable architecture, and OSI has become the standard model for classifying communications functions.

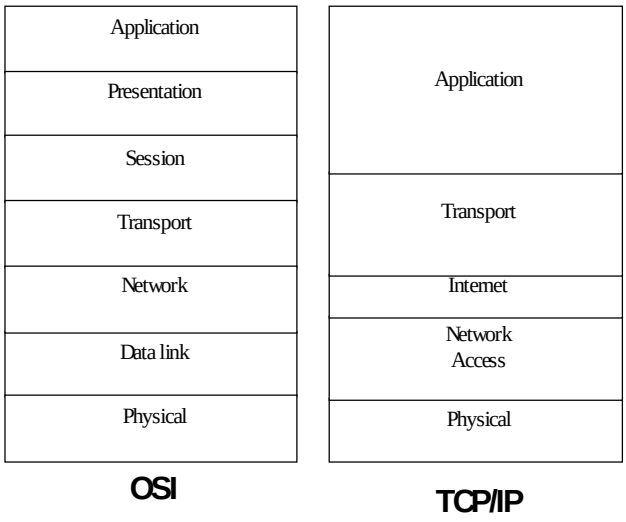


Figure 1 OSI and TCP/IP protocol architectures according to W.Stallings,Data and computer communications Fifth Edition

2.3.1 OSI Reference model

2.3.1.1 Physical layer

This layer is responsible for the mechanical, electrical, functional and procedural mechanism required for the transmission of data. It can be considered to represent the physical connection of a device to a transmission media.

2.3.1.2 Data link layer

The data link layer is responsible for the manner in which data is formatted into defined fields and the correction of any errors occurring during a transmission session.

2.3.1.3 Network layer

Provides upper layer with independence from the data transmission and switching technologies used to connect system. Responsible for establishing, maintaining and terminate connections.

2.3.1.4 Transport layer
Provides reliable and transparent transfer of data between end points. The transport layer also provides end-to-end error recovery and flow control.

2.3.1.5 Session layer
This layer is responsible for establishing and terminating data streams between network nodes. Since each data stream can represent an independent application, the session layer is also responsible for coordinating communications between different applications that require communications.

2.3.1.6 Presentation layer
Provides independence to the application processes from differences in data representation.

2.3.1.7 Application layer
Provides access to the OSI environment for users and also provides distributed information services.

2.3.2 TCP/IP Protocol suite
TCP/IP is a result of protocol research and development conducted on the experimental packet-switched network, ARPANET, funded by the Defense Advanced Research Projects Agency (DARPA), and is generally referred to as the TCP/IP protocol suite. This protocol suite consists of a large collection of protocols that have been issued as Internet standards by the Internet Architecture Board (IAB).

There is no official TCP/IP protocol model as there is in the case of OSI. However, based on the protocol standards that have been developed, the communication task for TCP/IP can be organized into five relatively independent layers (see Figure 2). The most common protocols will also be presented.

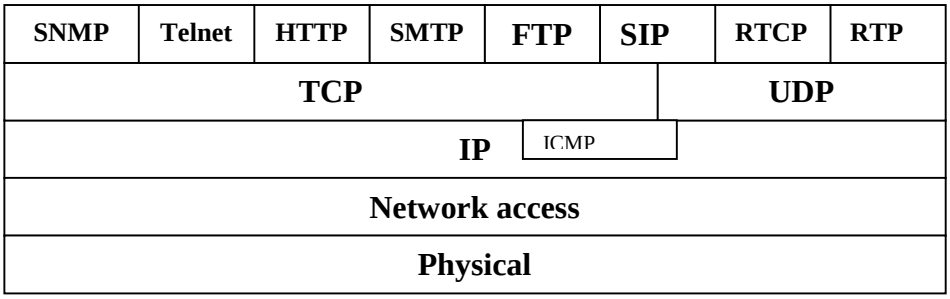


Figure 2 TCP/IP protocol model with the most common protocols and SIP

2.3.2.1 Physical layer
This Layer is responsible for the mechanical, electrical, functional and procedural mechanism required for the transmission of data. It can be considered to represent the physical connection of a device to a transmission media.

2.3.2.2 Network access layer

This layer is responsible for accepting and transmitting IP datagrams. It is also concerned with the exchange of data between an end system and the network to it is attached. The sending computer must provide the network with the address of the destination computer, so that the network may route the data to the appropriate destination. Ethernet is often used at this level. One of the most commonly used medium access control technique for bus, tree and star topologies is Carrier–Sense Multiple Access with Collision Detection (CSMA/CD).

2.3.2.3 Internet layer

This layer handles communication from one machine to the other. It accepts a request to send data from the transport layer, along with the identification of the destination. It encapsulates the transport layer data unit in an IP datagram and uses the datagram routing algorithm to determine whether to send the datagram directly onto a router. The Internet layer also handles the incoming datagrams and uses the routing algorithm to determine whether the datagram is to be processed locally or to be forwarded.

2.3.2.3.1 Internet protocol IP

IP provides a connectionless service between end systems. All the intermediate systems between the two ends just have to implement IP and the layers below IP. It is not necessary to implement higher layers. IP receives data from higher layers, and adds a header containing information related to the data received and passes it to the layer below. These packets are called IP datagrams. Each packet that travels through the Internet is treated as an independent unit of data without any relation to any other unit of data. It is for the protocol above to keep track of the order between packets and for guaranteeing that the packet arrives. The IP header contains all routing information necessary about sending the packet to the right next hop. If the packets are bigger than the maximum size that the link can handle, the IP protocol also takes care of fragmentation and reassembling of packets.

2.3.2.3.2 Internet Control Message Protocol ICMP

The IP standard specifies that a compliant implementation must also implement ICMP. ICMP provides a means for transferring messages from routers and other hosts to a host. In essence, ICMP provides feedback about problems in the communication environment. Examples of its use are: When a datagram cannot reach the destination, when the router does not have the buffering capacity to forward a datagram, and when the router can direct the station to send traffic on a shorter route. In most cases, an ICMP message is sent in response to a datagram either by a router along the datagrams path, or by the intended destination host. Although ICMP is, in effect, at the same level as IP in the TCP/IP architecture, it is a user of IP. An ICMP message is constructed and then passed down to IP, which encapsulate the message with an IP header and then transmits the resulting datagram in the usual fashion.

2.3.2.4 Transport layer

In this layer the stream of data is segmented into small data units and each packet is passed along with a destination addresses, to the layer below for transmission. The software also adds information to the packets, including port number that identify which application program sent it, as well as a checksum. This layer also regulates the

flow of information and provides reliable transport, ensuring that data arrives in sequence and with no errors.

2.3.2.4.1 Transmission Control Protocol TCP
TCP was developed to provide a reliable, connection-oriented service that supports end-to-end transmission reliability. To accomplish this task, TCP supports error detection and correction as well as flow control to regulate the flow of packets through a network. Error checking requires the computation of a Cyclic Redundancy Check (CRC) algorithm at a network node based on the contents of a packet and a comparison of the computed CRC against the CRC carried in the packet. If an error occurs TCP requests a retransmission of the faulty packets. This can result in unacceptable delays when transporting digitized voice and is rarely, if ever, used for voice transmission.

2.3.2.4.2 User Datagram Protocol UDP
UDP was developed to provide an unreliable, connectionless transport service. We should note that this is not necessary bad and adds a degree of flexibility to the protocol family. That is, if reliability is required, a higher layer, such as the application layer, can be used to ensure that messages are properly delivered. A second feature of UDP is the fact that it is a connectionless protocol. This means that instead of requiring a session to be established between two devices, transmission occurs on a best-effort basis. That is, function associated with connection setup and the exchange of status information as well as flow control procedures are avoided.

2.3.2.5 Application layer
At this level, users invoke application programs to access available services across the TCP/IP Internet. The application program chooses the kind of transport needed, which can be either messages or streams of bytes, and passes it to the transport level.

2.3.2.5.1 Real-time Transport Protocol RTP
RTP provides end-to-end delivery services for data with real-time characteristic, such as interactive audio and video. Those services include payload type identification, sequence numbering, timestamps, and delivery monitoring. Applications typically run RTP on top of UDP to make use of its multiplexing and checksum services. However, RTP may be used with other suitable underlying network or transport protocol. RTP supports data transfer to multiple destinations using multicast distribution, if provided by the underlying network.

2.3.2.5.2 Real Time Control Protocol RTCP
RTP is a simple protocol designed to carry real-time traffic and to provide a few additional services that are not present in existing transport protocols like UDP. With RTP, receivers can utilize the timestamp along with sequence numbers to better synchronize sessions. As a companion to RTP there is RTCP (RTP Control Protocol), which is used to communicate between the source and destinations. RTCP is based on the periodic transmission of control packets to all participants in the session, using the same distribution mechanism as the data packets. The underlying protocol must provide multiplexing of the data and control packets, for example, by using separated port numbers with UDP.

2.3.2.5.3 File Transfer Protocol FTP
The FTP is a mechanism for moving data files between hosts via a TCP/IP network. FTP operates as a client-server process, with the client issuing predefined commands to the server to navigate its directory structure and to upload and download files to and from the server.

2.3.2.5.4 Telnet
Telnet represents another TCP/IP client-server application. This application is designed to enable a client to access a remote computer as though the client was a terminal directly connected to the remote computer.

2.3.2.5.5 Simple Mail Transfer Protocol SMTP
The SMTP provides the data transportation mechanism for electronic messages to be routed over a TCP/IP network. This protocol is completely transparent to the user since there are no commands that govern the transfer of electronic mail via SMTP.

2.3.2.5.6 Hypertext Transmission Protocol HTTP
HTTP is the foundation protocol of the World Wide Web and can be used in any client/server application involving hypertext. The data transferred by the protocol can be plain text, hypertext, audio, images, or any Internet accessible information.

2.3.2.5.7 Simple Network Management Protocol SNMP
The SNMP provides the mechanism to transport status messages and statistical information about the operation and utilization of TCP/IP devices. In addition, with SNMP, devices can generate alarms when certain predefined thresholds are reached.

2.4 SIP

2.4.1 Signaling overview

To establish, maintain and terminate a session between two or more participants, there is a need for a common signaling protocol. There are several signaling protocols in use today, both in PSTN and in the Internet. SS7 is the name of a common signaling protocol in PSTN.

Two standards have recently emerged for signaling and control for Internet telephony. One is International Telecommunication Union (ITU) recommendation H.323, and the other is the Internet Engineering Task Force (IETF) SIP. These two protocols represent very different approaches to the same problem: H.323 embraces the more traditional circuit-switched approach to signaling based on the ISDN Q.931 protocol and earlier H-series recommendations, and SIP favors the more lightweight Internet approach based on HTTP.

2.4.2 Introduction to SIP

The Session Initiation Protocol (SIP) is a Proposed Standard protocol [1]. SIP were developed by the Multiparty Multimedia Session Control Working Group of the IETF.

SIP is a signaling protocol for establish, modify and terminate sessions or calls. These sessions can be with one or more participants. SIP sessions include Internet telephony, multimedia conferences, distance learning and similar applications.

SIP is an application layer protocol that does not require reliable transport protocol. It provides its own reliability mechanism. SIP typically runs over UDP or TCP, but it could also run over IPX, frame relay, ATM or X.25.

SIP is designed as part of the overall IETF multimedia data and control architecture. Other protocols in the IETF architecture are RSVP, RTP, RTSP, SAP and SDP. The functionality and operation of SIP does not depend on any of these protocols.

2.4.3 SIP functionality

SIP can be used in conjunction with other call setup and signaling protocols. Internet telephony gateways that connect to PSTN parties can use SIP to set up calls between them. SIP might be used to determine that a callee is reachable via the PSTN and indicate the phone number to be dialed, using an Internet to PSTN gateway.

Callers and callees are identified by SIP addresses. A SIP URL is similar to a mailto, user@host. That makes it possible to initiate the call through a web page or as a part of an email message. The user part is a user name or a telephone number. The host part is either a domain name or a numeric network address. For example sip:lars@test.com, sip:mattias@113.4.4.62 and sip:7262173@113.54.3.32

When making a SIP call, a caller first locates the appropriate server and then sends a SIP request. The most common SIP operation is the invitation. Instead of directly

reaching the intended callee, a SIP request may be redirected or may trigger a chain of new SIP requests by proxies. Users can register their locations with SIP servers.

To establish a call, an INVITE request is sent, the caller then wait for a response from the callee. A ringing message (180 Ringing) is send to the caller when the callee is trying to alert the user. If the callee agree to participate and the caller sends an ACK a session is established. If the caller no longer wants to establish the session, the caller sends a BYE request instead. The INVITE request typically contains a session description, for example using the Session Description Protocol (SDP).

SIP supports mobility for the end user, which means that a callee can move between different end systems and still be reachable.

2.4.4 The participants in SIP communications

User agent
A user agent is an application, which contains both a user agent client and user agent server. The user agent client is the calling user agent (outgoing calls) and the user agent server is the called user agent (incoming calls).

Proxy server
An intermediary program that acts both a server and a client for the purpose of making request on behalf of other clients.

Redirect server
A redirect server is a server that accepts a SIP request, maps the address into zero or more new addresses to the client.

Location server
A location server is used by a SIP redirect or proxy server to obtain information about a callee's possible location or locations.

2.4.5 SIP messages

SIP is a text-based protocol. The message syntax is and header fields are identical to HTTP/1.1 specification [2]. A SIP message is either a request or response. A request is generated by a client and a response is generated by a server.

2.4.5.1 SIP requests

The SIP request is sent from a client to a server. The request contains Request-Line, header-field and a message body. The Request-Line consists of method token, Request-URI and SIP-version. The SIP methods are listed below. The method token is case-sensitive.

INVITE
The INVITE request invites a user or service to participate in a session. The message body contains a description of the session.

ACK
The ACK request confirms that the caller has received a final response to an INVITE request, acknowledge a successful response.

OPTIONS
This method handles capability information and discover the capabilities of the receiver.

BYE
The BYE request terminates a call or call request, either a caller or callee can send a BYE request.

CANCEL
The CANCEL request cancels a pending request, but does not affect a completed request. In other words the CANCEL method terminates incomplete call requests.

REGISTER
The REGISTER method is used as a simple location service that register the current location of a user. The REGISTER method is also needed when there are several SIP servers on a single host, then only one of the servers can use the default port number.

Methods that are not supported by a proxy or redirect server, are treated as an OPTIONS method and forwarded accordingly. Methods not supported by a user agent server or registrar causes a 501 Server Failure (Not Implemented).

The header fields may be followed with a message body separated with an empty line. For ACK, INVITE and OPTIONS the message body is always a session description. The Content-Type must give the Internet media type. In the example below the message body type is the Session Description Protocol (see chapter 2.4.10.4 Session Description Protocol SDP).

Example of an INVITE request in a two-party Internet phone call

```
INVITE sip:mattias@mars.ausys.se SIP/2.0
Via: SIP/2.0/UDP swerdet.ausys.se
From: Lars <sip:lars@ausys.se>
To: Mattias <sip:mattias@ausys.se>
Call-ID: 4353456345@swerdet.ausys.se
Cseq: 1 INVITE
Subject: Hello Mattias
Content-Type: application/sdp
Content-Length: 187

v=0
o=mattias 52655765 2687637 IN IP4 172.2.2.5
s= Hello Mattias
c=IN IP4 swerdet.ausys.se
m=audio 3456 RTP/AVP 0 3 4 5
```

2.4.5.2 SIP responses

After receiving and interpreting a request message, the recipient responds with a SIP response message. The SIP response message contains Status-Line, header-field and a message body. The Status-Line consists of SIP-version , Status-Code and Reason-Phrase. The reason-phrase is a short textual description of the Status-Code. The Status-Code is a 3-digit integer code, there are six different types of the Status-Code.

1xx: Informational	Request received, continuing to process the request. For example 180 RINGING.
2xx: Success	The action was successfully received, understood, and accepted. 200 OK.
3xx: Redirection	Further action needs to be taken in order to complete the request. 302 MOVED TEMPORARILY.
4xx: Client Error	The request contains bad syntax or cannot be fulfilled at this server. 404 NOT FOUND.
5xx: Server Error	The server failed to fulfill an apparently valid request. 501 NOT IMPLEMENTED.
6xx: Global Failure	The request cannot be fulfilled at any server. 600 BUSY EVERYWHERE.

Example of a 200 response in a two-party Internet phone call

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP swerdet.ausys.se
From: Mattias <sip:mattias@ausys.se>
To: Lars <sip:lars@ausys.se>
Call-ID: 4353456345@swerdet.ausys.se
Cseq: 1 INVITE
Contact: sip:lars@mars.ausys.se
Content-Type: application/sdp
Content-Length: 158

v=0
o=lars 3245436364 3453633533 IN IP4 172.4.5.4
s=Hello again
c=IN IP4 mars.ausys.se
m=audio 3456 RTP/AVP 0 3 4 5
```

2.4.6 SIP header fields

SIP messages use header fields to specify specific things about the participants, the path and so on. SIP header fields are similar to HTTP header fields in both syntax and semantics. The order in which header fields appears is generally of no importance, except with the exception of that header fields that are hop-to-hop must appear before any header fields, which are end-to-end. Some of SIP's header fields are used in all messages and others only when necessary. An application containing SIP does not need to understand all header fields, though it is desirable. If a SIP participant does not understand a header it simply ignores it. The total amount of SIP header fields is 37, which can be divided into four groups.

1. **General header fields** apply to both request and response messages.

2. **Entity header fields** define information about the message body or if no body is present, about the resource identified by the request.

3. **Request header fields** allow the client to pass additional information about the request, and about the client itself, to the server.

4. **Response header fields** give information about the server and about further access to the resource identified by the Request-URI.

Request-headers
Accept
Accept-Encoding
Accept-Language
Call-Id
Contact
Case
Date
Encryption
Expires
From
Record-Route
Timestamp
To
Via

General-headers
Authorization
Contact
Hide
Max-Forwards
Organization
Priority
Proxy-Authorization
Proxy-Require
Route
Require
Response-Key
Subject
User-Agent

Entity-headers
Content-Encoding
Content-Length
Content-Type

Response-headers
Allow
Proxy-Authenticate
Retry-After
Server
Unsupported
Warning
WWW-Authenticate

2.4.7 SIP example

A simple protocol exchanges for the INVITE method is shown in Figure 3 for a proxy server and in Figure 4 for a redirect server. The examples do not contain all the messages that SIP uses.

In Figure 3 the proxy server accepts the INVITE request (1), contacts the location service with all or parts of the address (2) and obtains a more precise location (3). The proxy server then issues a SIP INVITE request to the address returned by the location service (4). The user agent server alerts the user (5) and returns a success indication to the proxy server (6). The proxy server then returns the success result to the original caller (7). The caller using an ACK request, which is forwarded to the callee (8 and 9) to confirm the receipt of this message.

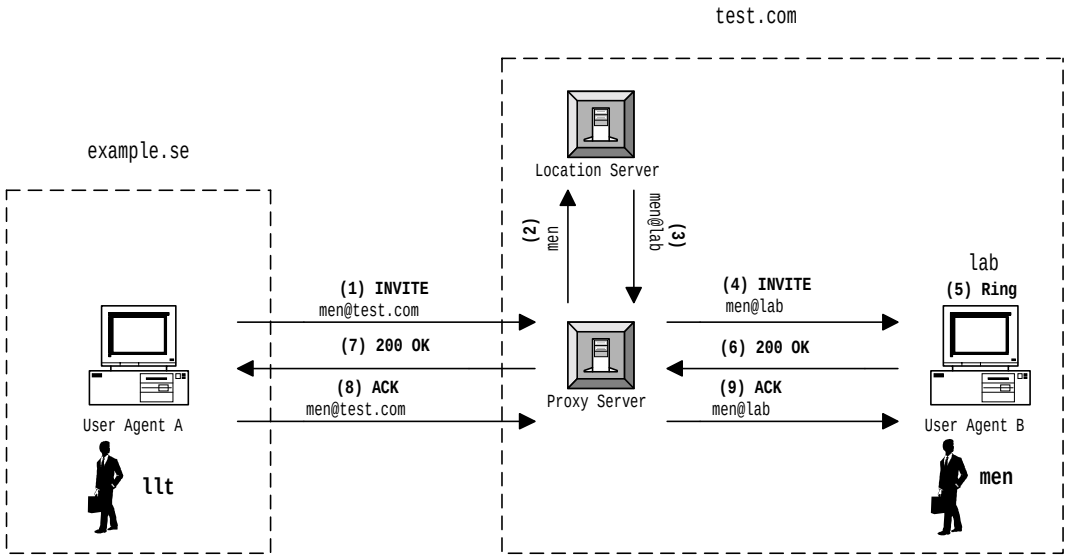


Figure 3 example of SIP proxy server

The redirect server shown in Figure 4 accepts the INVITE request (1), contacts the location service as before (2,3) and, instead of contacting the newly found address itself, returns the address to the caller (4), which is then acknowledged via an ACK request (5). The caller issues a new request, with the same call-id but a higher CSeq, to the address returned by the first server (6). The user agent server alerts the user (7). In the example, the call succeeds (8). The caller and callee complete the handshake with an ACK (9).

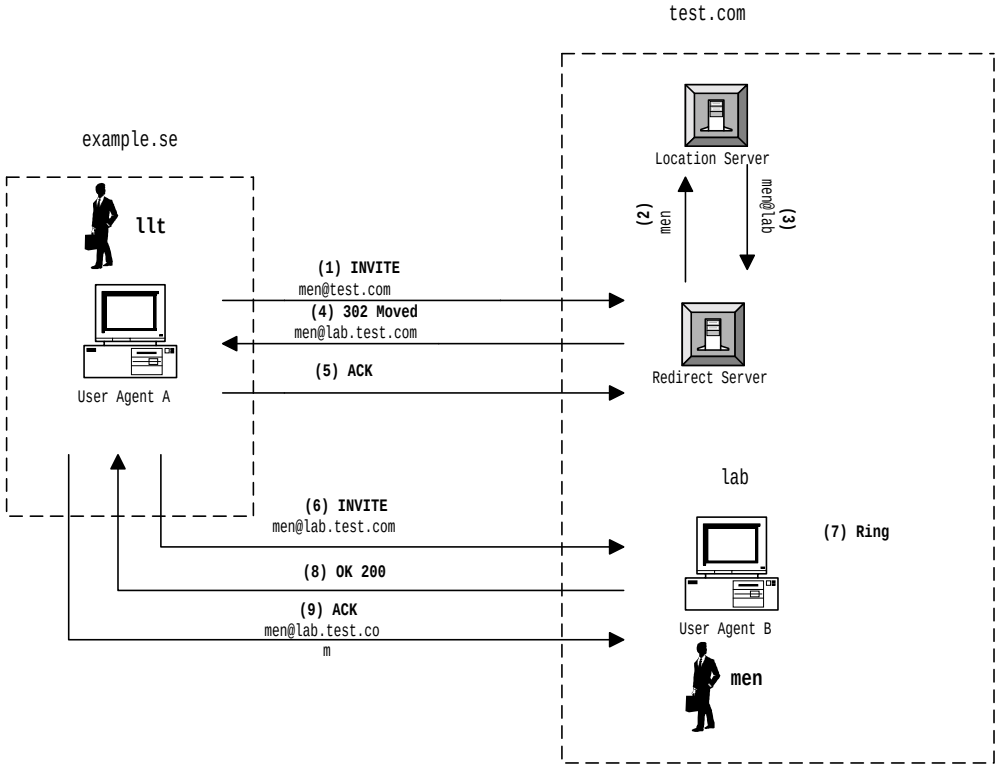


Figure 4 example of SIP redirect server

2.4.8 SIP security

SIP requests and responses can contain sensitive information about the communication patterns and communication content of individuals.
SIP supports the following forms of encryption

Encryption end-to-end
End-to-end encryption of the SIP message body and certain sensitive header fields. The keys are shared between the two user agents involved in the request.

Encryption hop-by-hop
Hop-by-hop encryption to prevent eavesdropping that tracks who is calling whom.
Hop-by-hop encryption of Via fields to hide the route a request has taken.
Hop-by-hop encryption can also encrypt messages that have been end-to-end encrypted.

All SIP implementations should support PGP-based encryption. The Pretty Good Privacy (PGP) is based on the model that the client authenticates itself with a request signed with a private key. The server can then ascertain the origin of the request if it has access to the public key, preferably signed by a trusted third party.

Even with encrypted requests there is a possibility, that an eavesdropper listens to messages and then injects unauthenticated responses that terminate, redirect or otherwise interfere with the call.

2.4.9 SIP minimal implementation

Client
All clients must be able to generate the INVITE and ACK requests. Clients must also generate and parse the Call-Id, Content-Length, Content-Type, CSeq, From and To headers. Clients must also parse the Require header. A minimal implementation must understand the protocol SDP. It must also be able to recognize the status code classes 1 through 6 and act accordingly.

Server
A minimally compliant server implementation must understand the INVITE, ACK, OPTIONS and BYE requests. A proxy server must also understand Cancel. It must parse and generate, as appropriate, the Call-Id, Content-Length, Content-Type, CSeq, Expires, From, Max-Forwards, Require, To and Via headers. It must echo the CSeq and Timestamp headers in the response. It should include the server header in its responses.

2.4.10 IETF multimedia data and control protocols

SIP is a part of the overall IETF multimedia data and control architecture currently incorporating protocols such as RTP, RSVP, RTSP, SAP, and SDP. Some of these protocols are briefly described below. Remember, the functionality and operation of SIP does not depend on any of these protocols.

2.4.10.1 Resource ReSerVation Protocol RSVP

The RSVP protocol provides a mechanism whereby applications that require guaranteed bandwidth. Shortly, RSVP reserves resources. The drawback with RSVP is that it requires that all equipment between source and destination must support RSVP.

2.4.10.2 Real Time Streaming Protocol RTSP

The RTSP protocol establishes and controls either a single or several time-synchronized streams of continuous media as audio and video. RTSP acts like a network remote control for multimedia servers.

2.4.10.3 Session Announcement Protocol SAP

Advertising multimedia sessions via multicast. SAP should be used for sessions of some public interest where the participants are not known in advance, otherwise a better mechanism is to explicitly invite them using SIP.

2.4.10.4 Session Description Protocol SDP

SDP is intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of session initiation.
SDP is in widespread use in SAP, SIP, H.332, and RTSP.

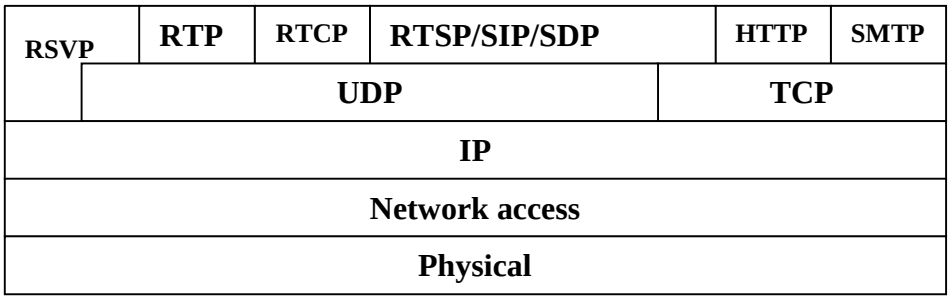


Figure 5 IETF multimedia data and control protocols

2.5 DTM

DTM, Dynamic synchronous transfer mode, is a broadband network architecture developed at the Royal Institute of Technology in Sweden. It is an attempt to combine the advantages of circuit switching and packet switching, in that it is based on fast circuit switching. Because of the inherent simplicity of circuit switching, DTM is a cost-effective technology scaling to high-speed fiber optical links. DTM is designed to support various types of traffic and can be used directly for application-to-application communication, or as a carrier network for other protocols [3].

Compared to ATM and SONET/SDH, DTM offers a higher network utilization rate and provides the support for new, advanced services at a much lower cost. Furthermore, DTM's simplicity and distributed architecture allows for unique scaling properties.

2.5.1 Fast circuit switching

Fast circuit switching was proposed for use in telephone systems already in the early 1980s. A fast circuit switched telephone network attempts to allocate a transmission path of a given data rate to a set of network users only when they are actively transmitting information. This means that a circuit is established for every burst of information. When silence is detected, the transmission capacity is quickly reallocated to other users [4]. It has been shown that the signaling delay associated with creation and tear down of communication channels determines much of the efficiency of fast circuit switching. DTM is therefore designed to create channels fast, within a few hundreds of milliseconds. Even though a DTM network may have the potential to create a channel for every message, this approach is probably not suitable for all traffic classes. Rather, it is up to the user to decide whether to establish a channel per information burst or to keep the channel established even during idle periods.

2.5.2 DTM time slots

DTM is based on a simple circuit switched scheme on a shared link, such as a ring (see Figure 6) or a bus. The link capacity is divided into cycles of 125 microseconds, which is further divided into 64-bit time slots (see Figure 7). Access to the link is based on time and space division multiplexing: in order for a node to communicate with other nodes, it has to own time slots on the link segments between the nodes. A small portion of time slots is reserved for sending the control information used for controlling access to the link and for setting up and taking down channels.

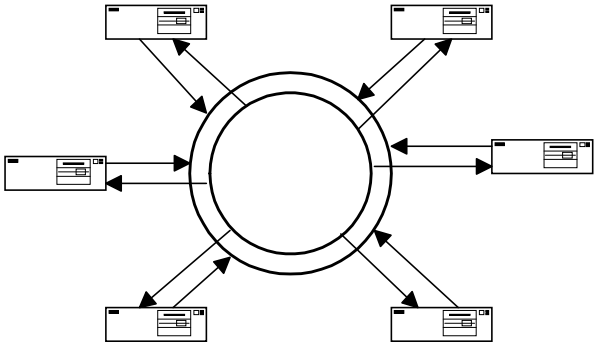


Figure 6 typical ring configuration

The slots are separated into data and control slots. Each node has access to at least one control slot, which is used for sending control information to the other nodes. At system start up, the data slots are allocated to the nodes according to some predefined distribution. This means that each node owns a portion of data slots. A node needs ownership of a slot to send data in it, and the ownership of slots may change dynamically among the nodes during the operation of the network.

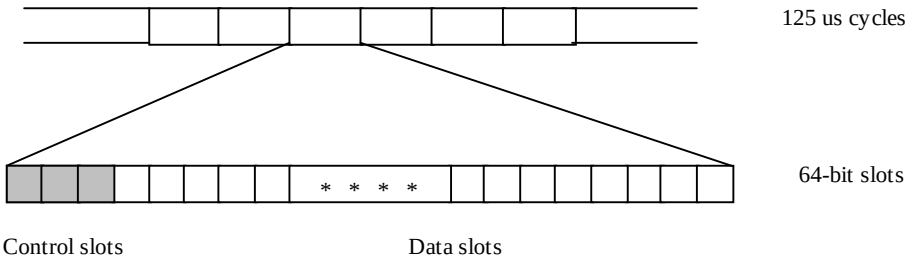


Figure 7 cycles and slots

2.5.3 DTM slot allocation

DTM uses a distributed algorithm for slot reallocation, where the pool of free slots is distributed among the nodes.

At the reception of a user request, the node first checks its local pool to see if it has slots enough to satisfy the request and, if so, immediately sends a channel establishment message to the next hop. Otherwise the node first has to request more slots from the other nodes on the bus.

Each node maintains a status table that contains information free slots in other nodes, and when more slots are needed the node consults its status table to decide which node to ask for slots. Every node regularly sends out status messages with information about its local pool.

2.5.4 DTM channels

The DTM channels differ from ordinary circuits in several ways, some examples below.

Simplex
DTM channels are simplex, making it possible to achieve high bandwidth utilization with asymmetric traffic.

Multirate
A DTM channel is a dynamic resource that can be set up with a bandwidth ranging from 512 kbps in quantum steps of 512 kbps up to the full capacity of the link. To each channel a set of network time slots are initially associated. The slots may be changed dynamically during the lifetime of the channel, which enables efficient network management and billing capabilities.

Multicast
DTM channels are multicast by nature. This means that any channel at any given time can occupy one sender and any number of receivers. Over the network, any number of multicasting groups can be active simultaneously. This is important when distributing video or other multicast services.

Fast establishment
Signaling delay associated with creation and deletion of communication channels determines much of the efficiency of fast circuit switching. Therefore, DTM is designed to create channels fast. Channels are set up in less than a millisecond.

A node creates a channel by allocating a set of data slots for the channel and by sending a channel establishment control message. The control message is addressed either to a single node or to a multicast group, and announces that the channel has been created and what slot it uses. To create a channel, slots must be allocated at sender and at each switch node along the channel's route. Thus, switch nodes allocate slots for a channel on behalf of the sender. The switch nodes then start switching the

channel, by copying the channel's slots from the incoming to the outgoing bus. An attempt to establish a multi-hop channel fails if any of the switch nodes involved cannot allocate the required amount of slots. In this case another route has to be tried.

2.5.5 DTM by Dynarc

Dynarc has developed a transmission and switching technology that enables the true convergence of data- and telecommunication services in a single network. Based on fast circuit switching, DTM inherently supports real-time guarantees across the network, crucial for carrier-class services such as voice and streaming video. In addition, the distributed resource management scheme and the optimized DTM signaling make DTM highly efficient for data traffic. The protocol is designed to be used in integrated services networks, thus combining the quality of circuit switching with the flexibility of packet switching. For the first time, strict QoS in IP networks can be realized.

Dynarc has developed DTM to provide an efficient and flexible link layer for high performance IP networks. With features like dynamic bandwidth allocation, constant delays and inherent multicast capabilities, DTM ensures high utilization with strict QoS. This is done at a very low overhead cost by separating DTM control information from data and an efficient multiplexing scheme of IP packets onto DTM channels.

Dynarc's DTM system provides two methods of setting up channels:

Dynamic channel management
Traffic-driven channel establishment provides a best-effort service allowing capacity sharing through statistical multiplexing of channels. For IP traffic, channels are set up on a next-hop basis ("macro flows"),

Controlled channel management
Channels are created and managed by higher layers, for example through explicit reservations and network management. Such channels provide a basis for supporting applications with QoS demands, traffic separation for virtual private networking etc.

Dynarc develops products that integrate DTM in existing networks according to a smooth migration strategy. By using layer 3 switching, data is forwarded across different network platforms. Currently, Dynarc provides support for IP routing, Ethernet bridging, VLAN, and E1/T1 tunneling, and DTM will run on dark fiber, DWDM as well as SDH/SONET infrastructures, on a wide range of network platforms. Dynarc has developed a modular software architecture, allowing for simple and timely adaptation to different platforms and different protocols.

3. Implementation methods and result

The test configuration will be able to setup both a voice path between an ordinary analogue telephone and a computer based IP telephone using headset and/or between computer based IP telephones. The signaling protocol SIP, sets up and terminates the connections. SIP also allocate/free bandwidth on the DTM link. When a connection between computer and telephone is established, the voice from the telephone is delivered to the computer headset, not from the headset to telephone. The voice from the telephone to headset is not telephone quality because that the Dialogic Voice Board D/41ESC doesn't use G.711 when encoding, that the audio application Taurus expect.

3.1 The hardware configuration

The following equipment is used in the test configuration.
2 Analogue telephones
1 Gateway/Server PC (NT)
1 Dialogic D/41ESC 4-port voice processing board
2 PC with soundcard and headset (WIN98)
1 DTM router, DS100 (NETBSD)
Fiber cables between gateway and a DS 100
DTM LAN card for gateway PC
Figure 8 shows the evaluation setup:

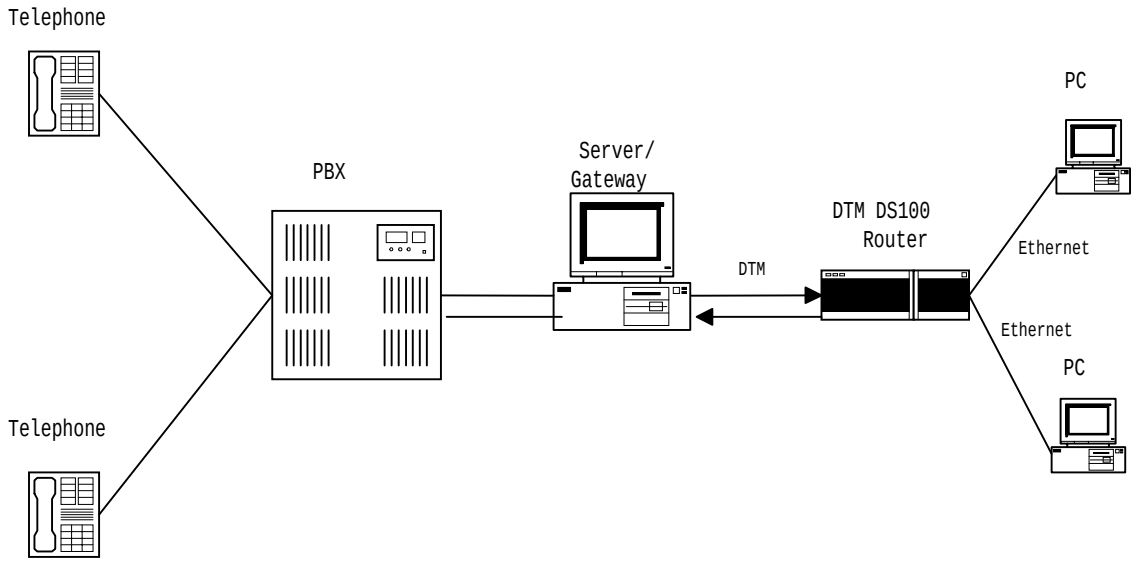


Figure 8 the test setup

3.2 The software development

One of the conditions before this thesis started was that a simple SIP user agent and server implementations would be available to install and configure. Instead of wait for the SIP implementations, that never showed up, we decided with our supervisors to write the SIP user agent and the SIP Server/Gateway implementation of our own. A lot of programming had to be done to setup the test configuration, the main programming activity follow:
Implemented a “simple” SIP user agent.
Implemented a “simple” SIP server/gateway.
Integrated SIP Server/Gateway with Dialogic APIs for the gateway.
Integrated and configured an existing voice and RTP application (Taurus) with SIP Server/Gateway.
Integrated Taurus with SIP user agent.
Implemented dynamic bandwidth allocation on DTM in the SIP server.

3.2.1 SIP User agent

The user agent includes a simple SIP implementation for signaling, a RTP and audio application for the voice data. The user agent is equipped with a soundcard and a headset with microphone.

We implemented our simple version of SIP user agent in the two Windows 98 PC’s. It is a client/server application. The server part accepts request and sends back responses. The client program sends SIP requests. Different threads are therefor active depending on, if the useragent are the calling user or the called user.

When executing the SIP user agent application, a client thread writes the text ”Press ’i’ for send INVITE” in a command prompt and a server thread is active listening for an incoming INVITE (see Figure 9). If the key ’i’ is pressed, the client thread ask for host IP-address and destination IP-address and then sends an INVITE. The client thread then waits for a response OK/BYE. If OK is received the user can send ACK or BYE. ACK establish a session and the audio application Taurus is executed. The session and Taurus are terminated when the other side in the session press the key ’c’ for close the session. When the SIP user agent already established a session, a byeserver thread automatically sends BYE to new callers. When the session is terminated the application is again ready for receiving or sending INVITE. If an INVITE is received a server thread instead of the client thread handles the establishing and termination of the session.

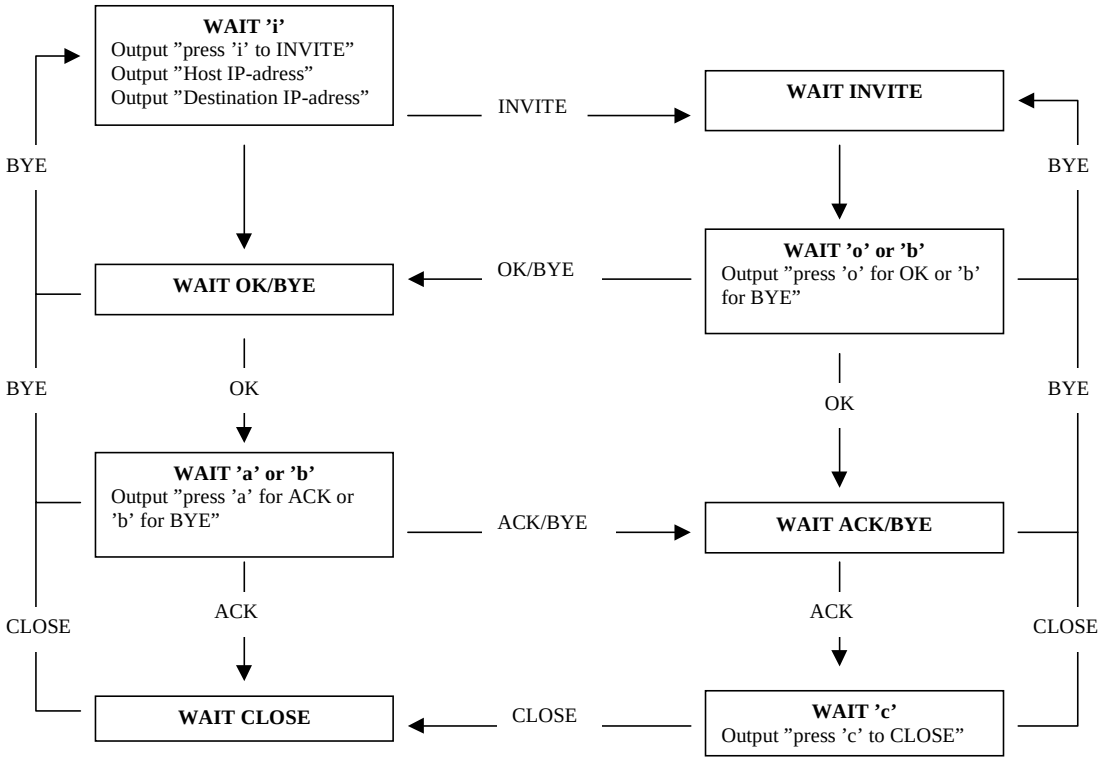


Figure 9 SIP User agent sequence diagram

Following request and responses are supported in the implementation INVITE, ACK, OK, BYE.

The SIP user agent is written in C using Microsoft Visual C++ and the source code includes threads, semaphores, sockets and processes.

3.2.2 SIP Server/Gateway

The SIP Server/Gateway sets up and terminates connections and handles the DTM dynamic bandwidth allocation. The gateway has the task to handle and forward call setup signaling and voice-path data in full duplex. It handles the Dialogic API. The gateway is implemented in the SIP Server/Gateway and uses RTP as the protocol for voice data.

The server is designed to handle several connections at the same time. The SIP Server/Gateway is not asynchronous, therefor it must be only one request at the time. Which means that when a connection is established the SIP Server/Gateway is ready for the next one.

The server is also integrated with the Dialogic API. The server has three threads working with dialogic functions, one thread to dial out on PSTN, two threads listening on different channels for incoming calls from PSTN (Answer0 and Answer1).

The SIP Server/Gateway also contains a server thread that will create a server worker when there is an incoming request from a workstation. The SIP server/gateway also have a client thread that responds to requests. To handle all threads there is a thread named mainservthread that communicates with the threads and starts new dial and client threads. The thread structure is shown in Figure 10.

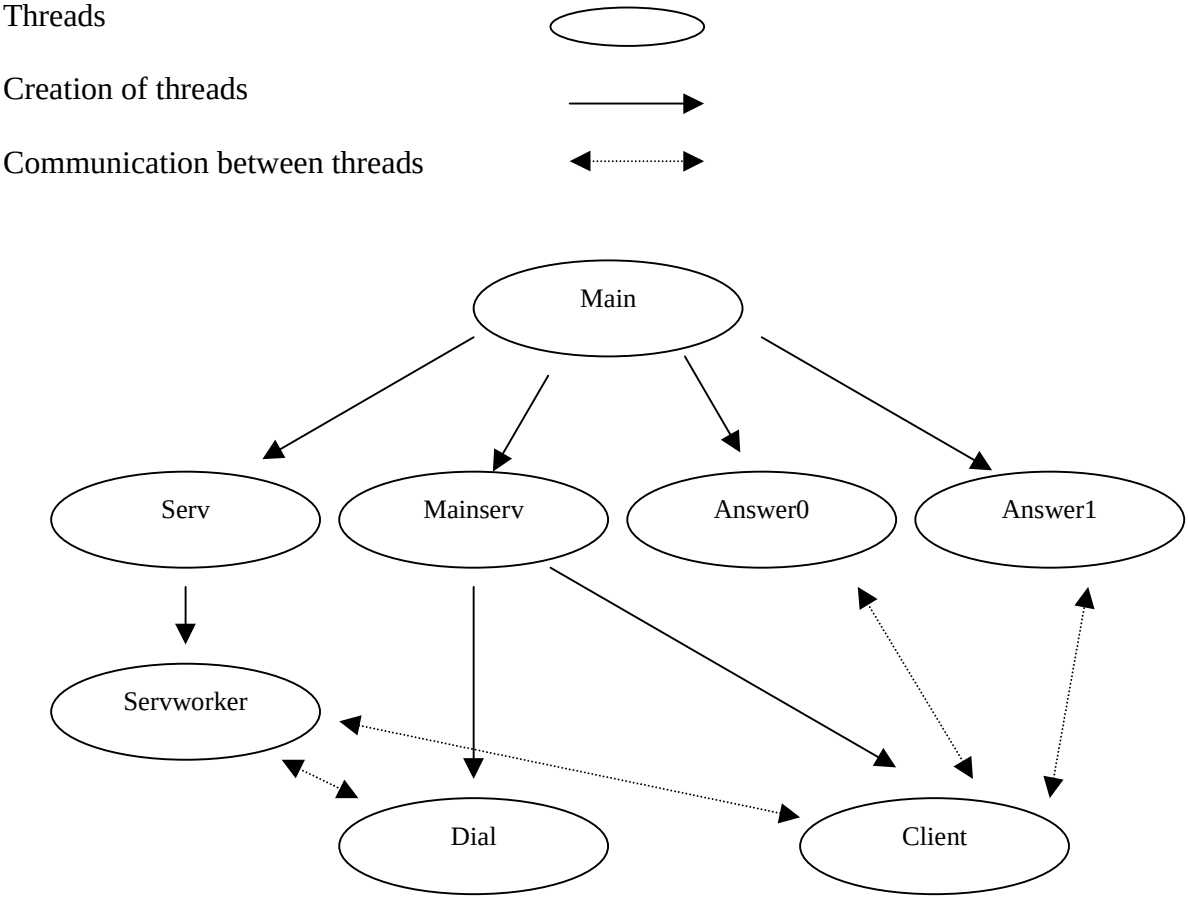


Figure 10 SIP Server/Gateway thread structure

The server is able to handle the SIP commands INVITE, ACK, BYE and OK, and to forward the information that it has received from the participants. The server handles two-party-calls, which means no multicast.

3.2.3 INVITE examples in the test setup

There are three kinds of INVITE examples in the test setup. Figure 11 shows the INVITE example between workstations and then there are two examples of INVITE between a Telephone and a workstation (see Figure 12 and Figure 13).

When the server receives an INVITE from a workstation (A) it looks at the information received from the caller. If the server recognize the IP-number in the To field it sends an INVITE to workstation (B) matching that IP-number. The server then waits for an OK from B, and forwards it to A. When A send an ACK the server forwards ACK to B and then allocate the necessary bandwidth on the DTM-link.
If either A or B sends a BYE under the setup or after the setup, the connection is terminated. When a connection is established it is the called person who terminates the connection with a BYE.

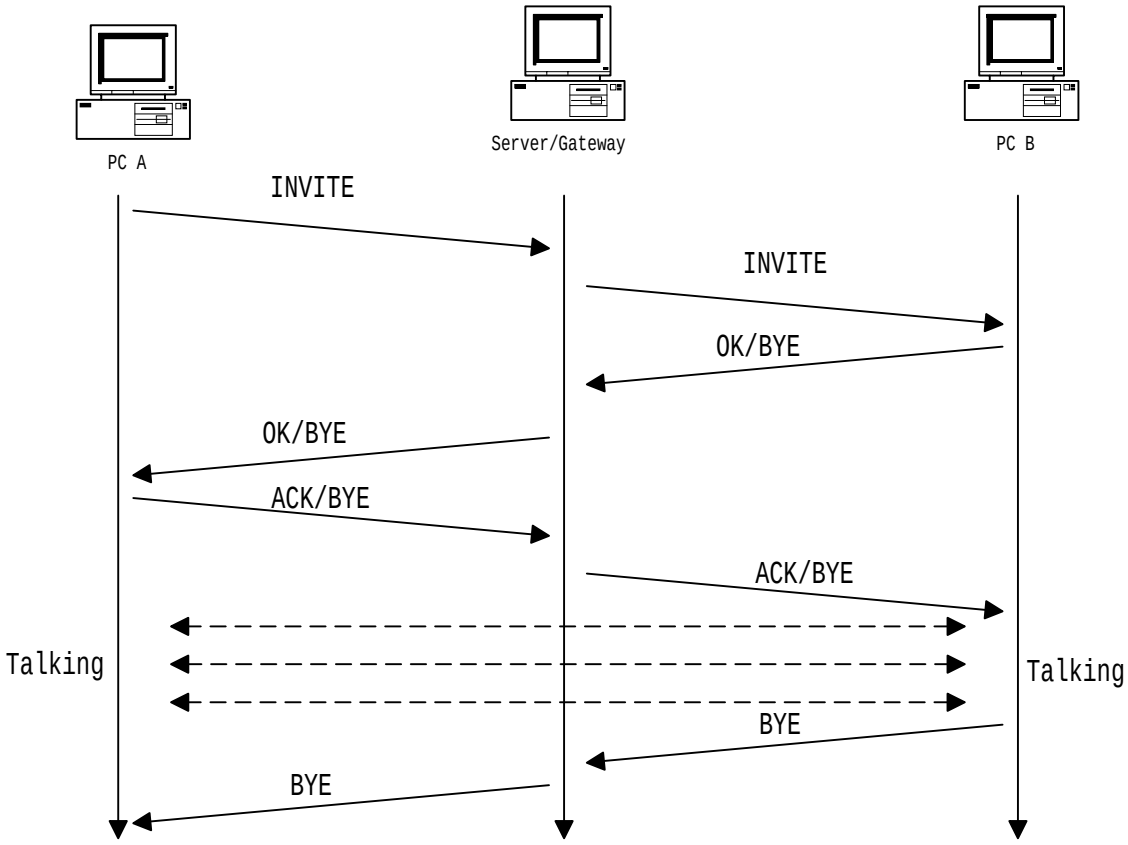


Figure 11 INVITE example between workstations

If there is an incoming call from PSTN or a workstation wants to call out to PSTN, the server acts like a user agent, starting Taurus after an ACK is received. To terminate from a “regular” telephone the server waits for a DTMF tone from the telephone.

The signaling procedure when a telephone INVITE a workstation will be presented in Figure 12. The user offhook the telephone and dials the specific number. The SIP Server/Gateway listen on the dialogic PSTN channels, passes the INVITE to the workstation (B) represented by the incoming telephone number. B then chooses to send OK or BYE, if an OK is send the server respond with an ACK and the session is established.

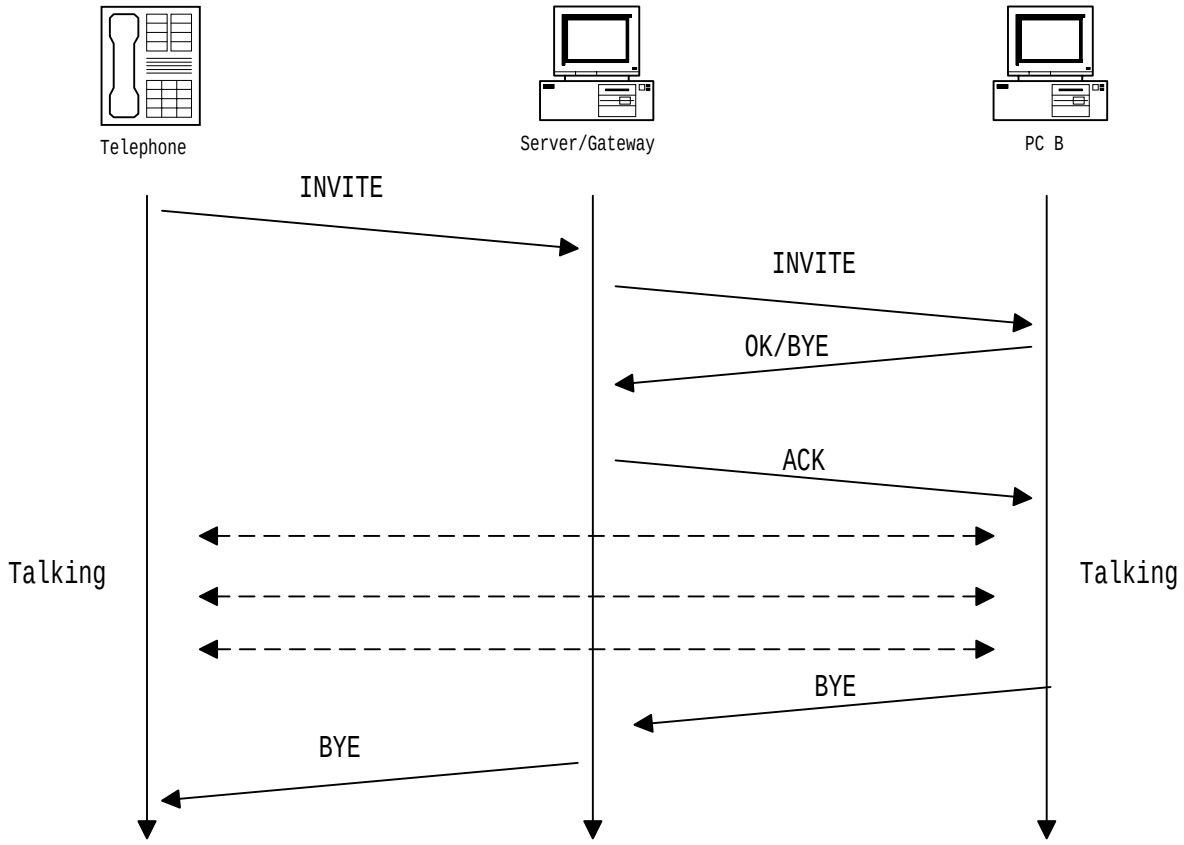


Figure 12 INIVITE example Telephone to workstation

The signaling procedure when a workstation (A) INVITE a telephone will be presented in Figure 13. A sends an INVITE to the server that responds with an OK, then the workstation sends an ACK or BYE. If an ACK is received at the server it dials the number that is represented by the incoming IP address.

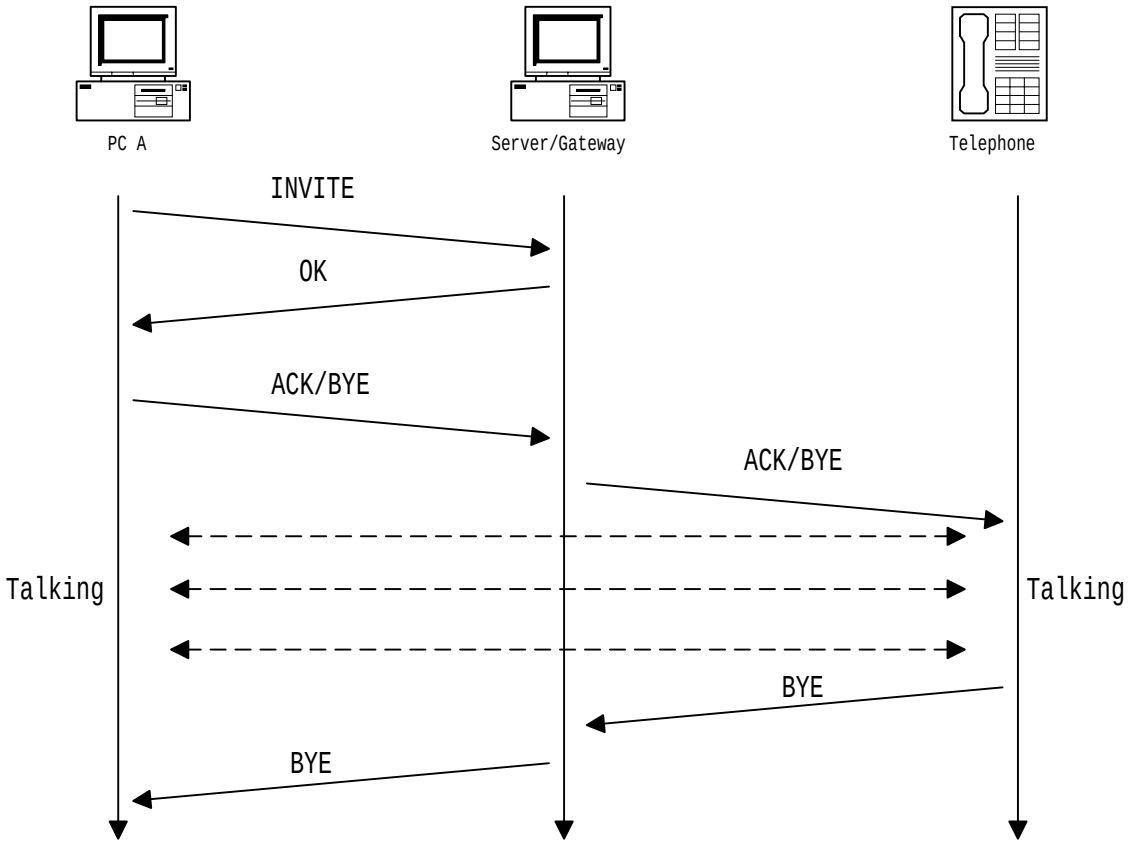


Figure 13 INVITE example workstation to Telephone

To start the telephone application the SIP server/gateway has to start a new process containing the modified Taurus. The changes in Taurus were made because instead of working with the soundcard, Taurus must work with the Dialogic hardware. Unfortunately there were some problems to do that, so Dialogic stores the data in a shared memory between SIP server/gateway and Taurus. Taurus then takes the stored data and forwards it in RTP packets to the other involved participant.

The software on the SIP server/gateway part was designed in Microsoft Visual C++, containing eight different threads, dealing with separate task. When the server doesn't have anything to do five threads are active, and if the server is active at least 7 threads are going. Besides shared memory and threads, the server contains processes, semaphores, sockets and events, the last used by Dialogic.

3.2.3 The Dialogic implementation

The Dialogic hardware that is implemented in the server is a D41/ESC-board, with 4 different channels, ready to handle a connection at each channel. AU-system had already chosen this board for the test setup. The D41/ESC board does not support G.711. The D41/ESC board is communicating using analog line or SCBus and there is no direct connection to IP line. The audio application Taurus expect the voice to be encoded with G.711, therefor the voice quality is lower than telephone quality.

Dialogic thinks that a better choice might have been the DM3 IPLink board that supports G.723. G.723 is designed for video conferencing/telephony over standard phone lines, and is optimized for real time encode and decode. But still there would be a missmatch for us, because the audio application we used, Taurus expect the voice to be encoded with G.711. From the authors point a view the best choice would be to use a complete IP-PSTN Gateway product.

In the test setup the Dialogic board works like the soundcard against Taurus. This means that the Dialogic board records the incoming voice and stores the data into a shared memory between the board and Taurus. Dialogic also supply a software API, this API contains a lots of different functions, which gives the programmer a lots of approaches to solve the programming task.

This example of the Dialogic API listen on one channel for an incoming call, and then records the voice.

```
/*  open a channel with chdev as descriptor */
chdev = dx_open("dxxxB1C1",NULL)

/* Set event mask to receive ring events */
dx_setevtmsk(chdev, DM_RINGS)

/* check for ring event for infinite */
dx_getevt(chdev,&eblk,INFINITE)
eblk.ev_event == DE_RINGS

/* set the voice channel off-hook */
dx_sethook(chdev, DX_OFFHOOK, EV_SYNC)

/* recording voice */
dx_rec(chdev,&iott, &tpt, RM_TONE |EV_SYNC)

/* set the voice channel on-hook */
dx_sethook(chdev, DX_ONHOOK, EV_SYNC)

/*  close the channel  */
dx_close(chdev)
```

The first task for the Dialogic board was to answer an incoming call, report to server that it has received a call. If the signaling results in a session, the Dialogic board starts to record the incoming voice. The second task for the Dialogic board was to dial an outbound telephone. If the Dialogic board is told to dial, SIP first checks if it recognize the telephone number and the Dialogic API makes it possible to dial the outbound telephone and to record the voice.

To close a connection with a BYE from a telephone, the Dialogic board has silence detection. The intention was to listen for a couple of seconds of silence and then assume that the telephone gone onhook. This was not possible because AU-systems PBX (Private Branch Exchange) did not support that technique. Instead the Dialogic board disconnect when a DTMF tone is received from the telephone.

3.2.4 Modified version of Taurus

In the server/gateway PC, Taurus had to be modified to read from the shared memory where Dialogic stores the voice from the telephone. Taurus expects by default the size of the incoming packets to be 10ms, with 8 kHz sample rate and 8 bits per sample the packet size is expected to be 80 bytes.

The Dialogic D/41ESC Voice Processing Board cannot record voice in that small buffers without add a clicking noise, when changing the record buffer. Therefor two 80 000 bytes circular buffers were used recording the voice from the telephone. Then Taurus at the server/gateway polls and read from the shared memory between Dialogic and Taurus, and sends 80 bytes packets to the Taurus application running on the user agent PC.

When recording in the shared memory between Taurus and SIP server/gateway, the recording function in Dialogic API records the speech in the two circular buffers (see Figure 14). Taurus then start poll the first buffer to see if there is any new data to send to the other participants. If there is new data Taurus send it and step forward 80 bytes and look in the beginning of the next 80 bytes. Because of that Taurus always look in the next 80 bytes of the buffer, Taurus immediately detect if there are any new data recorded. When the first buffer is full, the record function starts to recording in the second buffer, and Taurus sends the stored data. When one buffer is full, the record function starts to fill the other buffer.

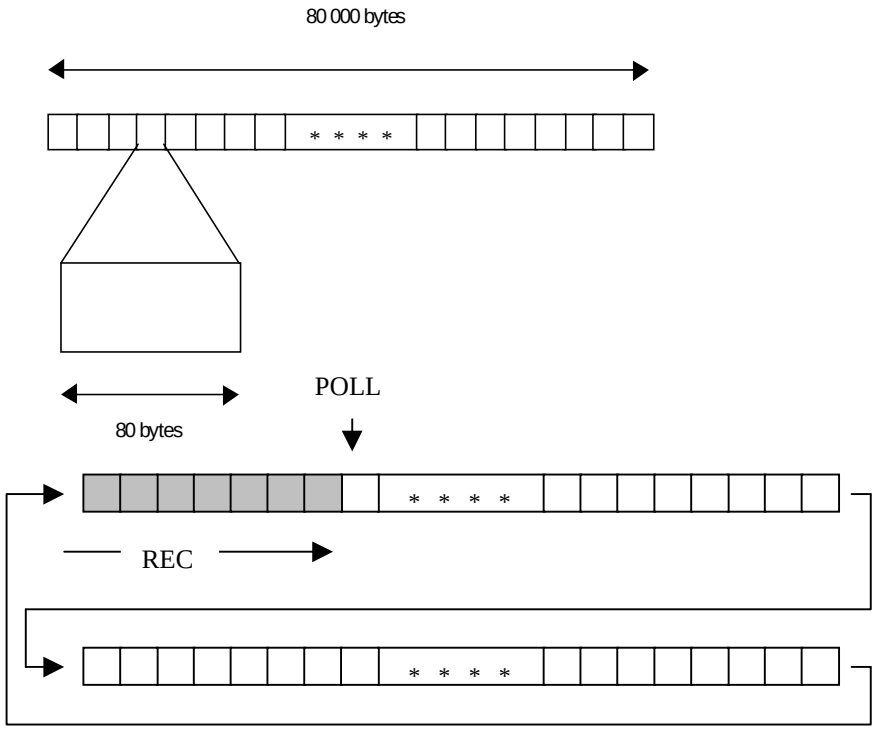


Figure 14 shared memory buffers

The Taurus modification was done using Microsoft Visual C++, to compile and configure Taurus properly cygwin was used.

3.2.5 DTM bandwidth allocation

The bandwidth allocation on DTM is implemented in the SIP server/gateway. The implementation using a simple algorithm, 1 slot allocated when there is no conversation and 10 slots whenever there is a conversation transfer over the DTM-link.

To allocate bandwidth locally on the server/gateway, there is only need for the Microsoft Visual C++ function system() with the appropriate command line.

The allocation on the DS100 is executed with a remote shell command from the SIP Server/Gateway setting up the number of slots and the timeout.

The remote shell command has to wait for the previous remote shell command to close his socket. A solution to the problem could be to implement a program on the DTM router DS100 that communicates with the SIP server/gateway, that make several remote shell commands unnecessary.

4. Conclusion

The future of IP-telephony looks bright. Many companies now have realized the possibilities with this technique. The benefits of transport voice and data on the same network are probably the main reason. The IP-telephony technique is still rather new, the development of it has come far in a short period of time and what we see now it's just the beginning.

The Session Initiation Protocol is sufficient enough to set up sessions, without being too complicated to understand. The interest of SIP is increasing and when SIP has been an accepted protocol, several companies will use SIP in their products instead of H.323.

Dynamic synchronous Transfer Mode is also a new technique that will have a breakthrough if the market realize all the benefits with DTM. DTM allows capacity allocation to vary with the traffic. A great advantage with DTM is the possibility to transport both data and voice over IP with high QoS. The DTM router will act like a router or a switch depending on the type of traffic.

With our experience of SIP and DTM, we can say that SIP is a simple protocol to penetrate and we needed about five weeks to write two simple SIP versions. We have nothing negative to say about the DTM router we used in our network. We had no problems to integrate SIP with DTM, so, maybe could SIP as signaling over DTM be the future for real-time applications.

We have set up the test network with a functional SIP signaling over DTM with dynamic allocation of bandwidth. We have also integrated the Dialogic API in the SIP server/gateway. Because we never got the promised SIP implementations, a major part of our work became to write our SIP user agent and Sip server/gateway. That led the work to contain more programming and less evaluation.

The result of the thesis was that we are able to set up and terminate a voice path between PC to PC, PC to Telephone and Telephone to PC. It is also possibly to transport voice over the established connection without any remarkable latency.

4.1 Problems and solutions

During the work several problem emerged. Some of the unsolved problems and suggestions how to solve them are listed below.

When the SIP Server/Gateway do remote shell (rsh) command to the DTM router DS100 to allocate bandwidth, the SIP Server/Gateway must wait for about 1 min before the next rsh command to the DS100. This is because rsh command uses a socket and the socket waits about one minute to close.

Solution: Write a program on the DS100 that communicate with the SIP Server/Gateway.

The creation of Taurus can fail after Taurus has been terminated earlier, because the TerminateProcess function that terminates Taurus does not terminate DLL's properly.

Solution: Integrate Taurus in the SIP Server/Gateway program or communicate between the processes and make Taurus terminate itself after a session.

There were also problems between dsound.dll in Win NT and dsound.lib in Microsoft Visual C++. They did not match each other even though we had the newest version of them both. This was not any problem with dsound in Windows 98.

Our solution: Everything in Taurus that referred to dsound was removed, this was okay because in the Server/Gateway Taurus worked with the Dialogic board instead of the soundcard

4.2 Tips for further improvements

Make our SIP programs more dynamic to fit other kinds of setups and networks.

Enlarge the simple SIP implementations.

Integrate Taurus in the SIP programs, both in the SIP user agents and the SIP Server/Gateway.

Change the Dialogic board D41/ESC to a complete IP-PSTN Gateway product.

Implement a program on the DS100 that communicates with the Server/Gateway to improve the dynamic bandwidth allocation.

5. Bibliography and references

Our general sources

W.Stallings, Data and computer communications Fifth Edition

D.Minoli and E.Minoli, Delivering Voice over IP Networks

G.Held, Voice over Data Networks

IETF
http://www.ietf.org

M.Handley, H.Schulzrinne, E.Schooler and J.Rosenberg, Internet Draft SIP: Session Initiation Protocol <draft-ietf-mmusic-sip-12.txt>

M.Handley, V.Jacobson, Internet Draft SDP: Session Description Protocol <draft-ietf-mmusic-sdp-02.txt>

H.Schulzrinne, S.Casner, R.Frederick and V.Jacobson, RFC 1889 RTP: A Transport Protocol for Real-Time Applications

H.Schulzrinne, R.Lanphier and A.Rao, RFC 2326 Real Time Streaming Protocol (RTSP)

Connected: An Internet Encyclopedia, G.711 Protocol Overview,
http://cie.bilkent.edu.tr/Topics/127.htm

Dialogic
http://www.dialogic.com

Codec central, G.723
http://www.terran-int.com/CodecCentral/Codecs/G723.html

Robust Audio Tool
http://www-mice.cs.ucl.ac.uk/multimedia/projects/rat/

H.Schulzrinne, Session Initiation Protocol(SIP)
http://www.cs.columbia.edu/~hgs/sip/

J. Crowcroft, M.Handley, I.Wakeman, Internetworking Multimedia, SIP, SAP, SDP, RSVP and RTSP
http://www.cs.ucl.ac.uk/staff/J.Crowcroft/mmbook/book

J.Skansholm, C++ direkt

J.Skansholm and U.Bilting, Vägen till C

B.Lewis and D.J.Berg Multithreaded Programming with Pthreads

J.Richter, Advanced Windows Third Edition

I.Gratte, Grunderna i C++

J.Luthman, Föreläsningsanteckningar Processprogrammering 98/99

Our specific references

[1] M.Handley, H.Schulzrinne, E.Schooler and J.Rosenberg, Standards Track RFC 2543 SIP:Session Initiation Protocol

[2] R.Fielding, J.Gettys, J.Mogul, H.Frystyk and T.Berners-Lee, RFC 2068 Hypertext Transfer Protocol -- HTTP/1.1

[3] Dynarc, http://www.dynarc.se, June 99

[4] C.Bohm, M.Hidell, P.Lindgren, L.Ramfelt and P.Sjödin, Fast Circuit Switching for the Next Generation of High Performance Networks

Appendix

A. Audio applications

The audio application used in this thesis was Taurus. In the beginning of the work Robust Audio Tool (RAT) was used, but it had too much latency compared to Taurus. The latency in RAT might already been solved.

A1. Taurus

Taurus uses the protocols RTP and UDP/IP. Taurus is simplex, like RTP, therefor two files have to be executed, one for sending audio and one for receive audio. Taurus is not a commercial product. Taurus is developed by the Multimedia Communication group at Ericsson Business Communication.

A2. Robust Audio Tool RAT

RAT is a network audio tool that allows users to participate in audio conferences over the internet. These can be between two participants directly, or between a group of participants on a common multicast group. RAT is based on IETF standards, using RTP above UDP/IP as its transport protocol. RAT is distributed free with source code(see Appendix B3 for copyright statement). RAT is developed by The UCL Networked Multimedia Research Group

A3. Robust Audio Tool Copyright Statement

B. Dialogic

The D41/ESC voice board was already choosen before this thesis started. A better choice might the DM3 IPLink board that supports G.723. Yet another choice would be to use a complete IP-PSTN Gateway product

B1. D/41ESC-Euro International 4-Port Voice Processing Board

Features and Benefits

- Four independent voice processing ports in a single PC ISA slot support low- to medium-density voice systems
- Reduces inventory requirements and replaces country-specific four-port boards with one assembly. D/41ESC-Euro is approved for use in all CTR21 member countries in Europe.
- SCSA™ SCbus™ connectivity enables applications requiring switching and allows access to additional resource cards such as fax, text-to-speech, and automatic speech recognition
- Supports UNIX®, Windows NT® and Windows® 95 including TAPI/WAVE®
- A-law or µ-law voice coding at dynamically selectable data rates, 24 Kb/s to 64 Kb/s, selectable on a channel-by-channel basis for optimal tradeoff between disk storage and voice quality
- Dialogic downloadable signal and call processing firmware, SpringWare™, provides easy feature enhancement and field-proven performance based on over three million installed ports
- International Caller ID capability via on-hook audio path. Supports Bellcore CLASS™, UK CLI, and other international protocols.
- PerfectDigit™ DTMF (touchtone) provides reliable detection during voice playback — allows callers to "type-ahead" through menus
- Patented outbound call progress analysis monitors outgoing call status quickly and accurately
- Configure multiple boards in a single PC for easy and cost-effective system expansion and to build scalable systems from 4 to 64 ports
- C-language application program interfaces (APIs) for MS-DOS®, UNIX, OS/2®, Windows NT, and Windows 95 shorten your development cycle so you can get your applications to market faster
- Support for Global Dial Pulse Detection (DPD™) pulse-to-tone conversion software
- Supports software-based speech technologies, include TextTalk™ TTS software and speech recognition
- Supports PBXpert™, a free utility that simplifies switch integration
- Supports legacy PBX switches that utilize Earth Break recall signaling.

Applications

- Voice messaging/auto attendant
- Interactive voice response
- Audiotex
- Inbound and outbound telemarketing
- Operator services
- Dictation

- Auto dialers
- Telecomputing servers
- Notification systems
- On-line data entry/query

Dialogic voice products offer a rich set of advanced features, including state-of-the-art DSP technology and signal processing algorithms, for building the core of any computer telephony system. With industry-standard ISA bus expansion boards and a variety of channel densities to choose from, you can integrate Dialogic voice products easily into exactly the type of system you require at a price and performance level unmatched in the computer telephony industry.

The D/41ESC-Euro board

- connects directly to analog loop start telephone lines
- offers application controlled call answering
- detects touchtones
- plays voice messages to a caller and digitizes, compresses and records voice signals
- places outbound calls and automatically monitors their progress
- all in real time on four independent channels.

Software Support

The D/41ESC-Euro is supported by Dialogic System Software and Software Development Kits for many popular operating systems including MS-DOS®, OS/2®, UNIX®, Windows NT® and Windows® 95. These packages contain a set of tools for developing complex multichannel applications.

B2. Dialogic System Software and SDK for Windows NT

FEATURES & BENEFITS

- Single programming paradigm for all computer telephony technologies
- Scalability allows an application to run on small or large systems
- Binary compatible API with Dialogic Development Kit for Windows® 95; provides further scalabilty to a smaller, lower cost operating system
- Multiple compiler support including Microsoft® Visual C++ 2.0 and higher; Borland C++® 4.5 and higher.
- TAPI and WAVE API support
- .WAV file play and record
- SMP support
- Supports threads, optimizing system resource usage
- Synchronous and asynchronous programming models
- Dynamic Link Library (DLL) reduces version dependencies
- Bulk data handled by driver/library leaving application free from real-time constraints
- Interrupt-driven architecture optimizes use of system resources
- Downloadable firmware offers optimum performance, reliability, and upgradability
- Software architecture reduces memory use and lowers maintenance costs
- State-of-the-art development support
- SCbus™ support

APPLICATIONS

- CPE
  -Unified messaging/voice messaging
  -Interactive voice response
  -LAN computer telephony services
  -Audiotex
  -Operator services
  -Telemarketing
  -Fax-on-demand
  -Dictation
  -Auto dialers
  -Notification systems
  -On-line data entry/query
- Telephone network
  -Operator services
  -Advanced Intelligent Network (AIN)/Intelligent Peripherals
  -Voice controlled dialing
  -Fax-on-demand
  -Fax broadcast
  -Automatic intercept credit card processing

The Dialogic System Software and SDK for Windows NT® provides a complete call processing development environment for Windows NT applications. The package is suitable for implementing robust and richly featured computer telephony systems that can reside on premises or in the telephone network.

The Dialogic System Software and SDK for Windows NT is designed for multithreaded, multitasking applications support providing improved interprocess communication and better use of system resources. Applications can be designed with a thread controlling each voice channel or multiple channels handled in a single thread.
Both synchronous and asynchronous programming models are supported. The synchronous programming model uses blocking functions that allow each channel to be controlled from a separate process or thread. This allows you to dynamically assign distinct applications to different channels in real time. The asynchronous programming model enables control of multiple voice channels from within a single process. This allows the development of complex applications where multiple tasks must be coordinated simultaneously. The asynchronous programming model supports both polled and callback event management.
A powerful yet flexible API makes it easy to write applications but still allows for low-level control through the use of parameters within each function.
Additionally, a DLL allows for greater efficiency and makes it easy for developers to take advantage of third-party facilities.
The package supports a variety of application interfaces, from the C language interface provided, to point-and-click application generators available from Dialogic Partners.

C. SIP Full Copyright Statement

Full Copyright Statement

AU-SYSTEM

D. Thesis specification

Specification	1999-03-19
Document number	Rev Reference PB1
Prepared Michele Mizzaro	Approved

SIP Telephony Gateway on DTM

Revision history

Rev.	Date	By	Comment
A	99-01-26	Lars-Göran Nilsson	Rev. A
PB1	99-03-19	Michele Mizzaro	Change figure modified text

AU-SYSTEM

Introduction

This document specifies the student project for the evaluation of a SIP - DTM integration. Currently several standards for telephony signaling exists, including SS7 for traditional telephony and SIP and H.323 for IP telephony.
SIP is a new IETF proposal in the process of being standardised. H.323 is a heavy ITU standard, while SIP has the ambition of being a "simpler" standard adopted to IP networks.
This student project examines how SIP could be integrated into a DTM network, regarding Quality of Services and handling of dynamic resource allocation.

Time schedule

W.9-12	Initial studying Learn about protocols: SIP, RTP, RTCP, DTM, UDP, IP. Learn about Dialogic´s Voice board.
W.13-22	Test setup and programming Implement a small SIP gateway in the server Integrate the RTP software in the server Implement a small SIP client in the PC phones Integrate the RTP software in the client Evaluate!
W.23-26	Report with evaluation results How does SIP co-operates with DTM in an IP network? What Quality of Service is achieved? Is there any fall-backs with this solution?

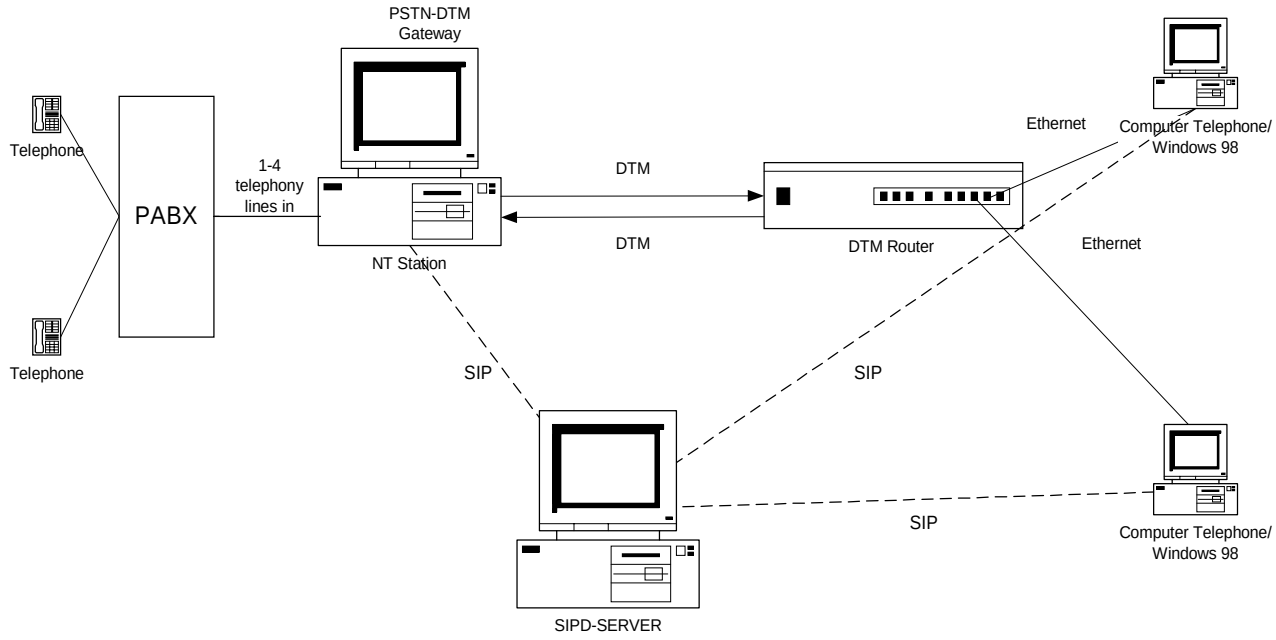
Technical configuration

The test configuration will be able to setup a voice-path between an ordinary analogue telephone and a computer based IP telephone. The student will evaluate how the QoS, latency and resource allocation is handled in a PSTN-SIP gateway using DTM as a carrier on the IP network.

The following equipment is needed for the configuration.

- 2 Analogue telephones
- 1 Gateway PC
- 1 Dialogic D/41D 4-port voice processing board
- 2 PC with SIP and RTP protocol implementations
- 1 DTM router, DS100
- Fibre cables between gateway an DS 100
- DTM LAN card for gateway PC
- SIP server NT or Solaris or other ????

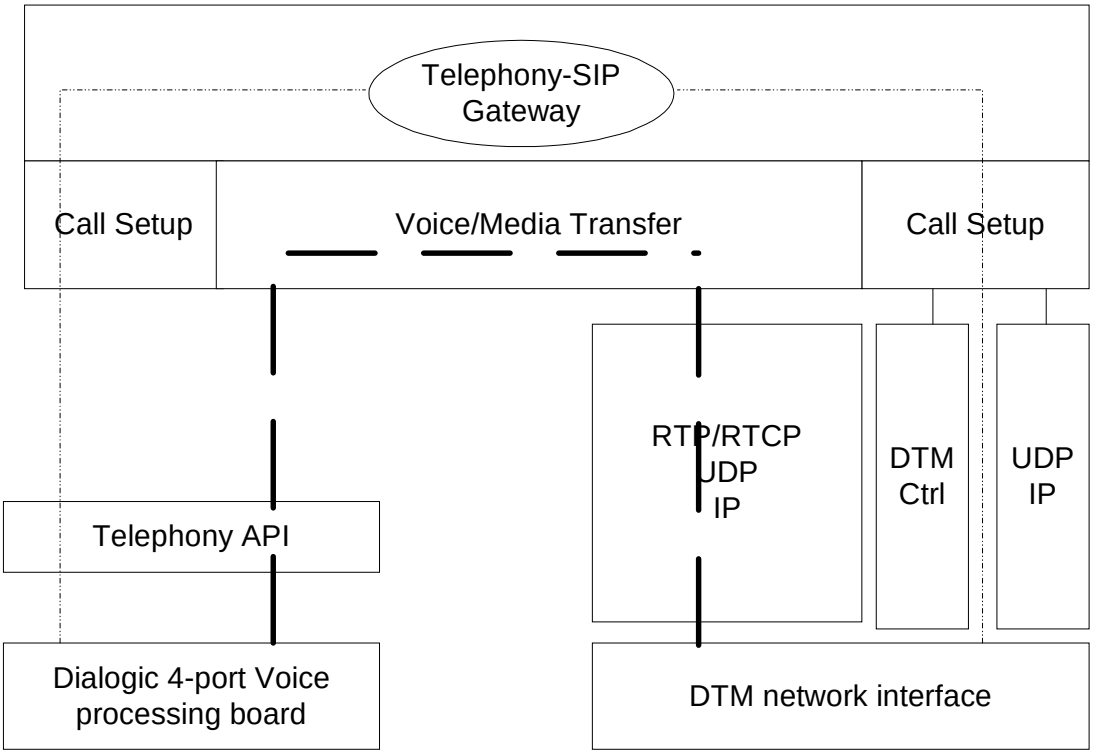
The following figure shows the evaluation setup:



Software Architecture

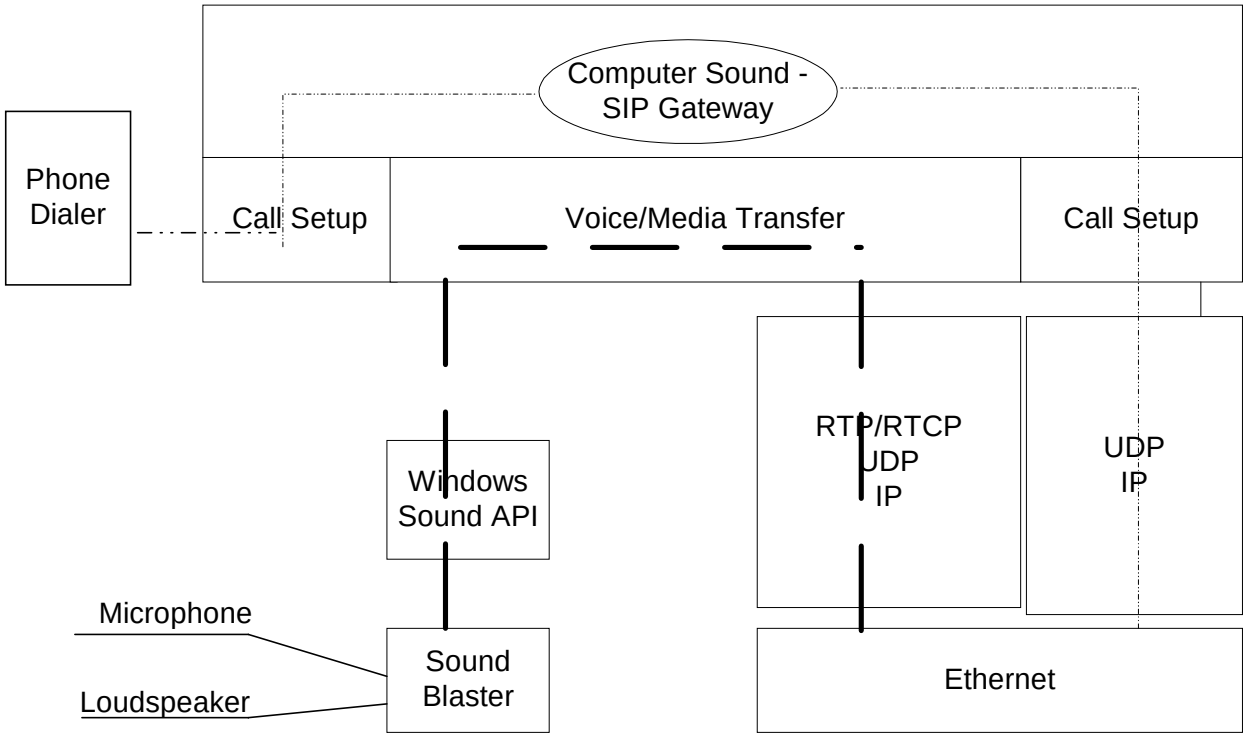
Gateway

The gateway has the task to handle and forward call setup signalling and voice-path data in full duplex. It handles the Dialogic programming interface, implements a simple SIP and uses RTP as the protocol for voice data.



Client

The computer telephone includes a simple SIP implementation for signalling and an RTP implementation for the voice data. The client is equipped with a SoundBlaster II compatible sound card and a head-set with microphone.



Programming activities
The two alternatives depend on how the SIP SW is implemented.

Alternative 1
The software development in this student project is limited to the following:

1. Make a simple symmetric SIP core implementation that can be used both in the gateway and in the client.
2. Add gateway specific functions to the SIP core aimed for the gateway.
3. Add client specific functions to the SIP core aimed for the computer telephone.
4. Integrate an existing RTP implementation both in the gateway and in the client.
5. Integrate the RTP with Dialogic APIs for the gateway.
6. Integrate the RTP with Windows Sound APIs in the client.

Alternative 2 – SIPD server with SIP clients
The software development in this student project is limited to the following:

1. Install and configure SIP server SW. Possibly some porting needs to be done.
2. Add gateway specific functions to the SIP core aimed for the gateway. An initial approach is to have static allocation of bandwidth, see also point 7.
3. Add client specific functions to the SIP client SW (Java or Tcl) aimed for the computer telephone.
4. Integrate an existing RTP implementation both in the gateway and in the client.
5. Integrate the RTP with Dialogic APIs for the gateway.
6. Integrate the RTP with Windows Sound APIs in the client.
7. Implement the SIP to DTM dynamic bandwidth allocation.