Advanced Networking

Voice over IP

aka Multimedia in the Internet

Renato Lo Cigno

LoCigno@disi.unitn.it

VoIP: Integrating Services

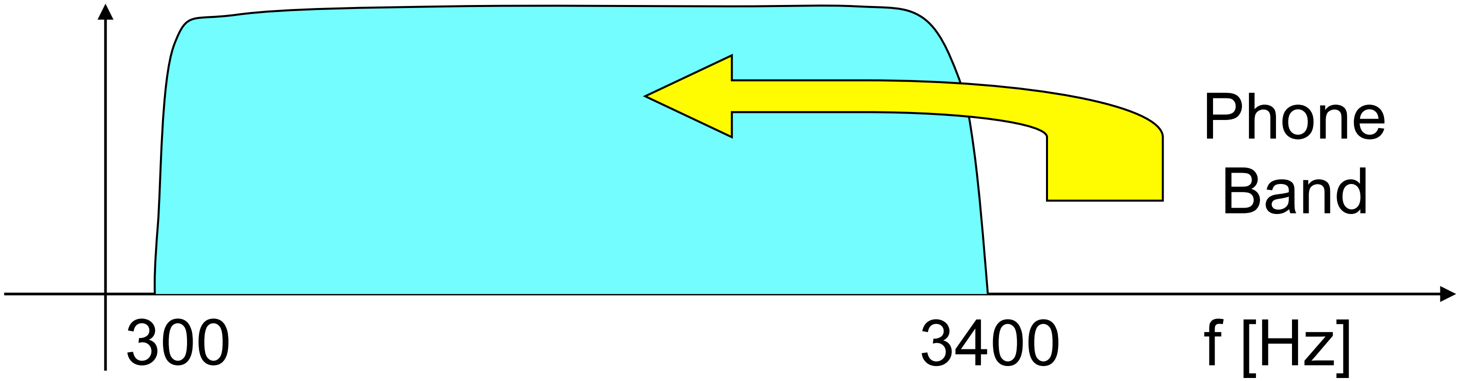
- Voice on IP Networks is just “another application”
- Nothing “special” or “specialized” as traditional telephony, where the network and the service are joint, coupled and sinergic
- VoIP is realized through end-to-end application level protocols, normally not strictly tailored for voice
- Is QoS required?

A digression: Voice & Telephony

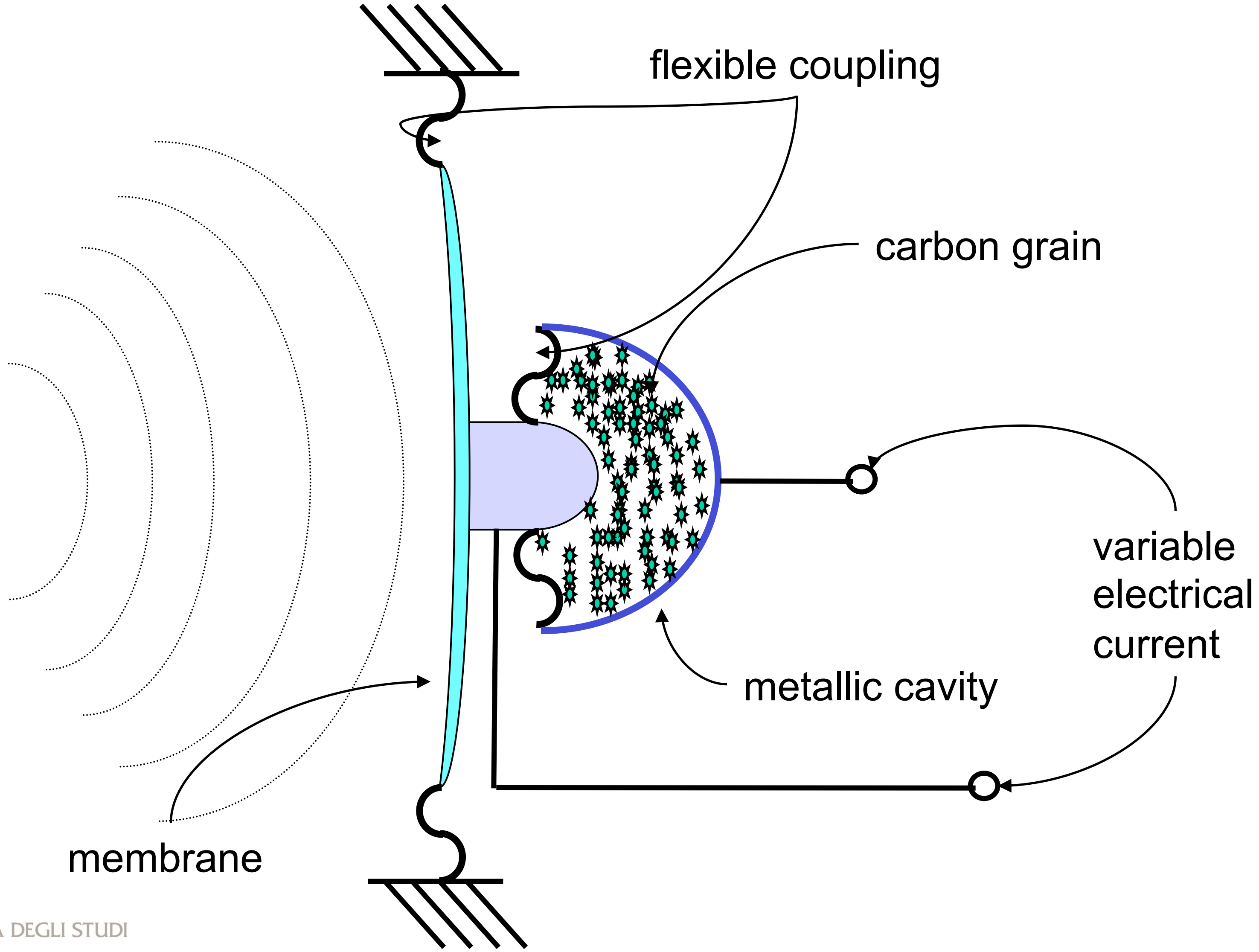
- Understanding Voice
  - What is it?
  - How is it transferred in networks?
- Understanding Telephony
  - More than voice
  - Need to replicate the services of the Plain Old Telephony Service

Voice and its transmission

- The voice signal is transmitted with analogic technique on the local loop, filtered between 300 and 3400 Hz to allow direct current for powering the phone and to limit the signal bandwidth to a known extent
- The local exchange immediately convert the analogic signal to digital PCM

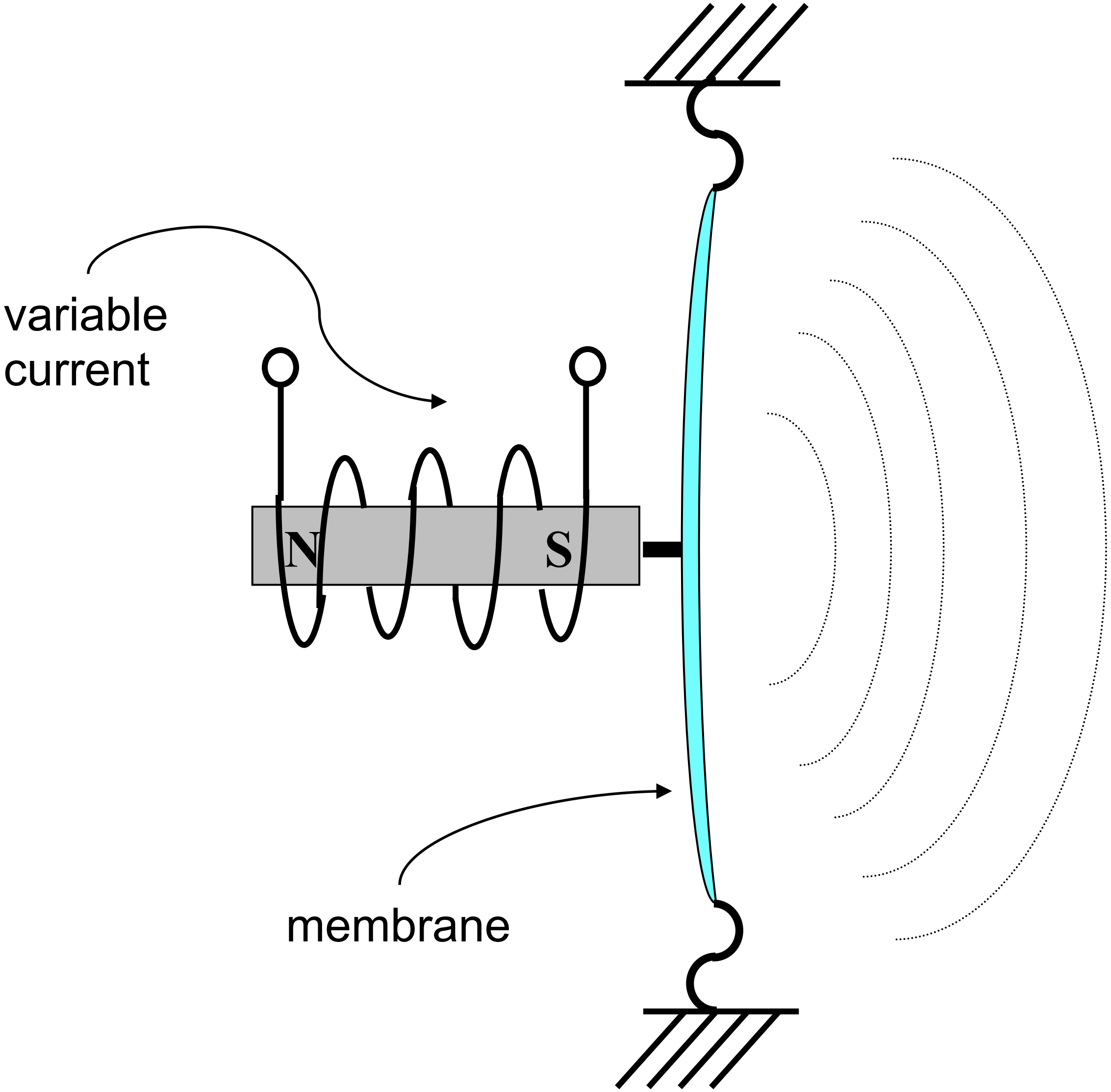


Traditional mike with carbon grains

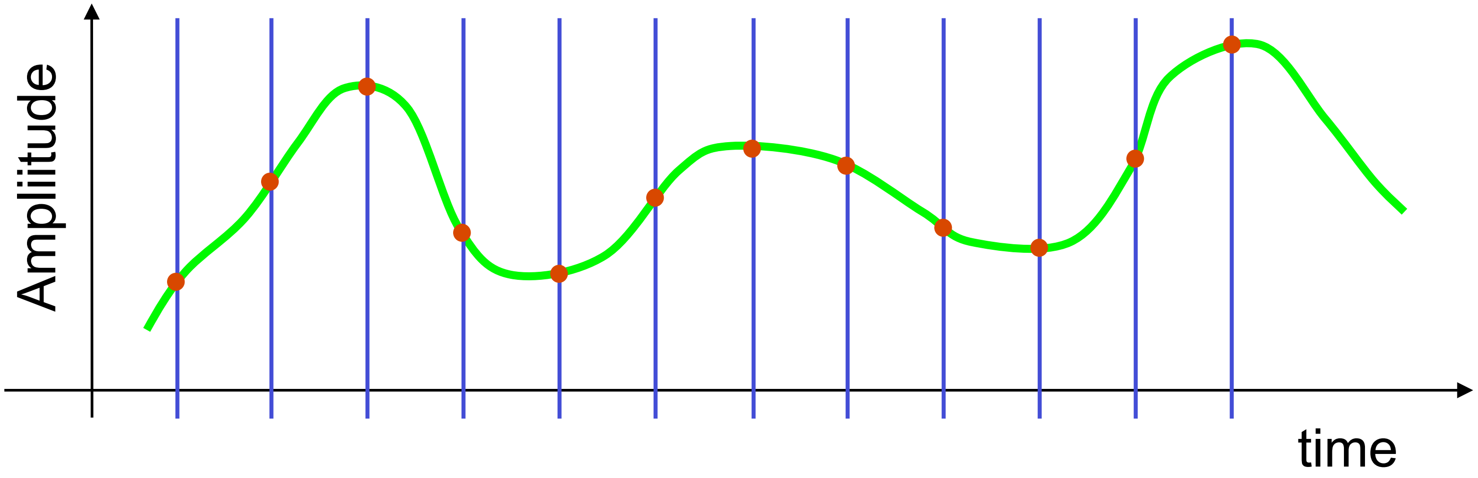


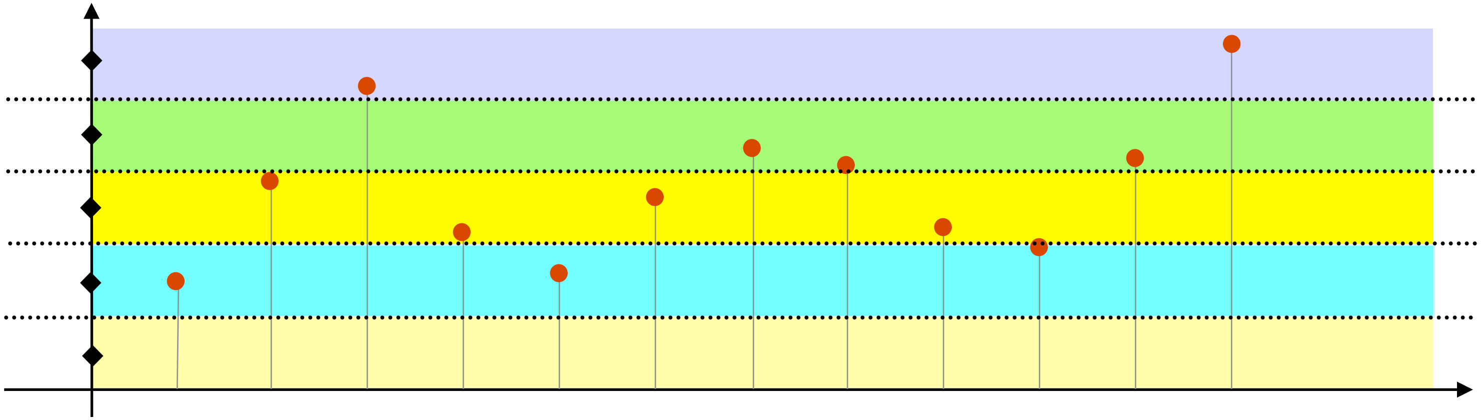
loudspeaker

- a membrane connected to a magnet within a coil
- the input signal moves the magnet which makes the membrane vibrate
- Modern electronic mike/louds are not much conceptually different



PCM: Sampling and Quantizing





PCM

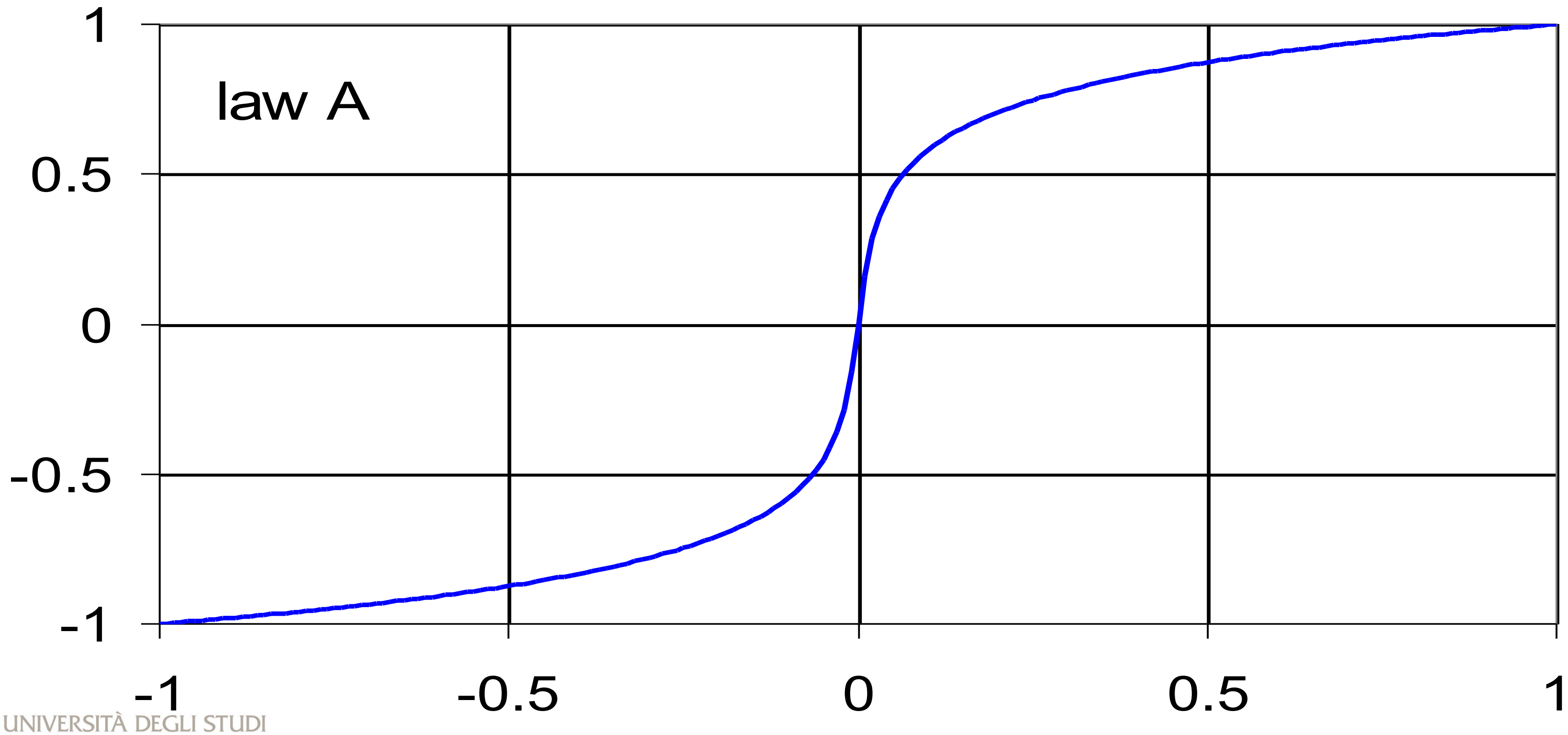
- PCM (Pulse Code Modulation) encoding is nothing else than sampling and quantizing (with non-linear quantization for telephone networks)
- Linear quantizing means equal intervals; non linear (companding) means different intervals as a function of amplitude
  - **Linear PCM: CD (~44 kHz, 16 bit =~ 1.5 Mbit/s)**
  - **Companding PCM: telephones (8kHz, 8 bit = 64kbit/s). Based on the fact that human ear sensitivity is logaritmic**

“A” Compression Law

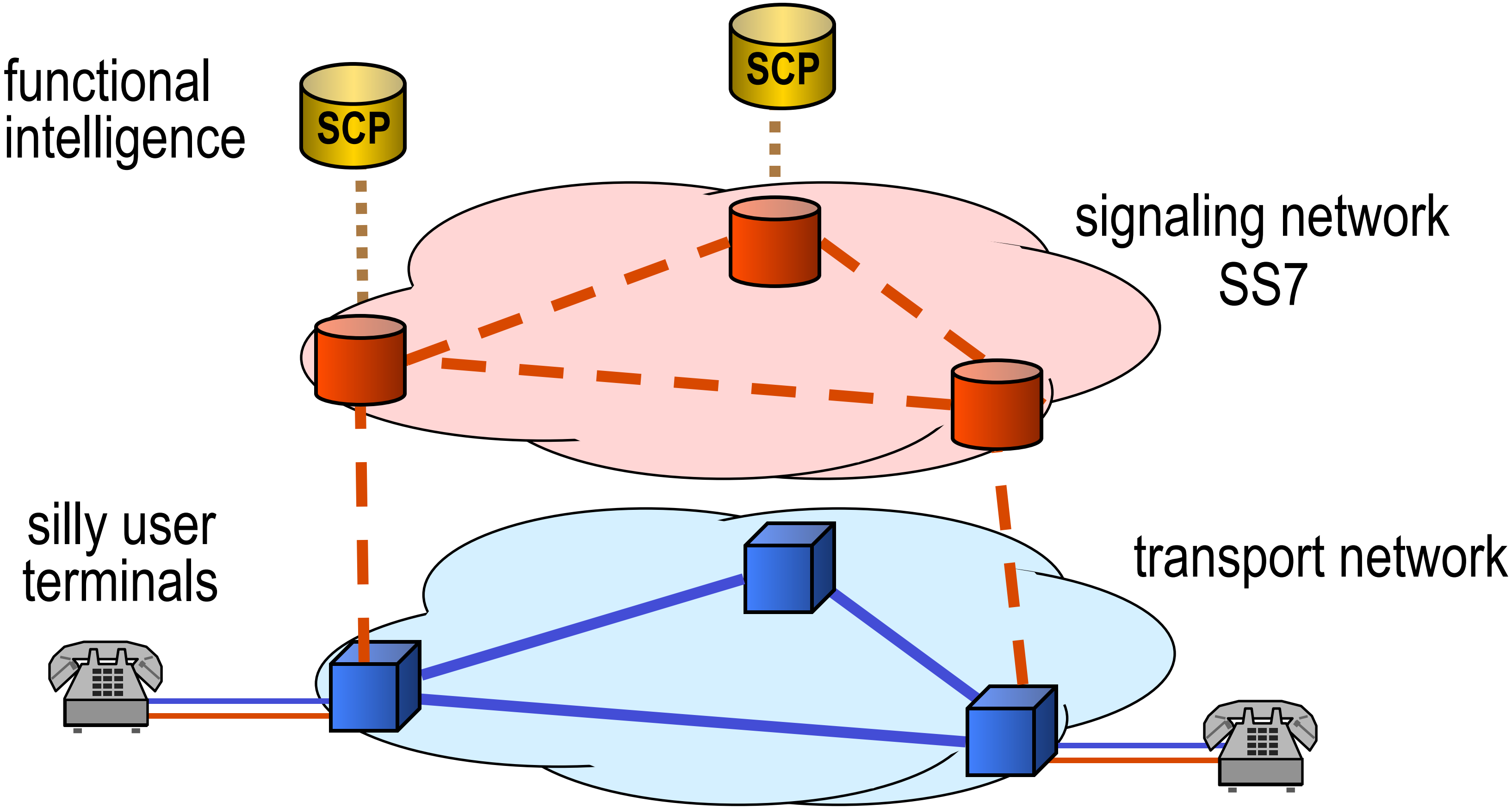
$$Y = \begin{cases} \dfrac{A}{1+\ln(A)} X & X < {}^{1}\!/\!_{A} \\ \dfrac{\operatorname{sgn}(X)}{1+\ln(A)}\left(1+\ln|AX|\right) & {}^{1}\!/\!_{A} < X < 1 \end{cases}$$

$A = 87.6$

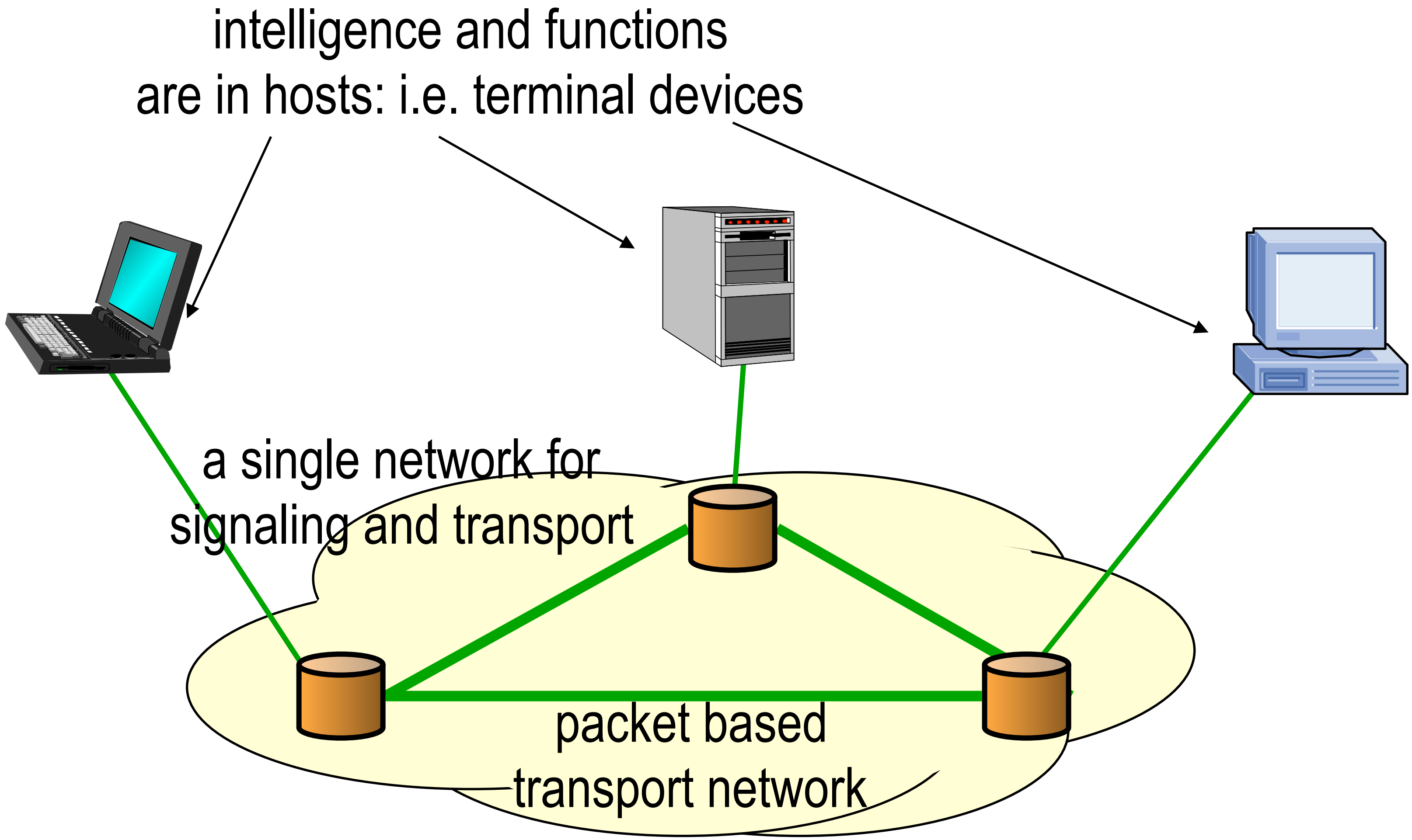
$X = {}^{V}\!/\!_{V_{\max}}$



Plain Old Telephone Service (POTS)



IP services

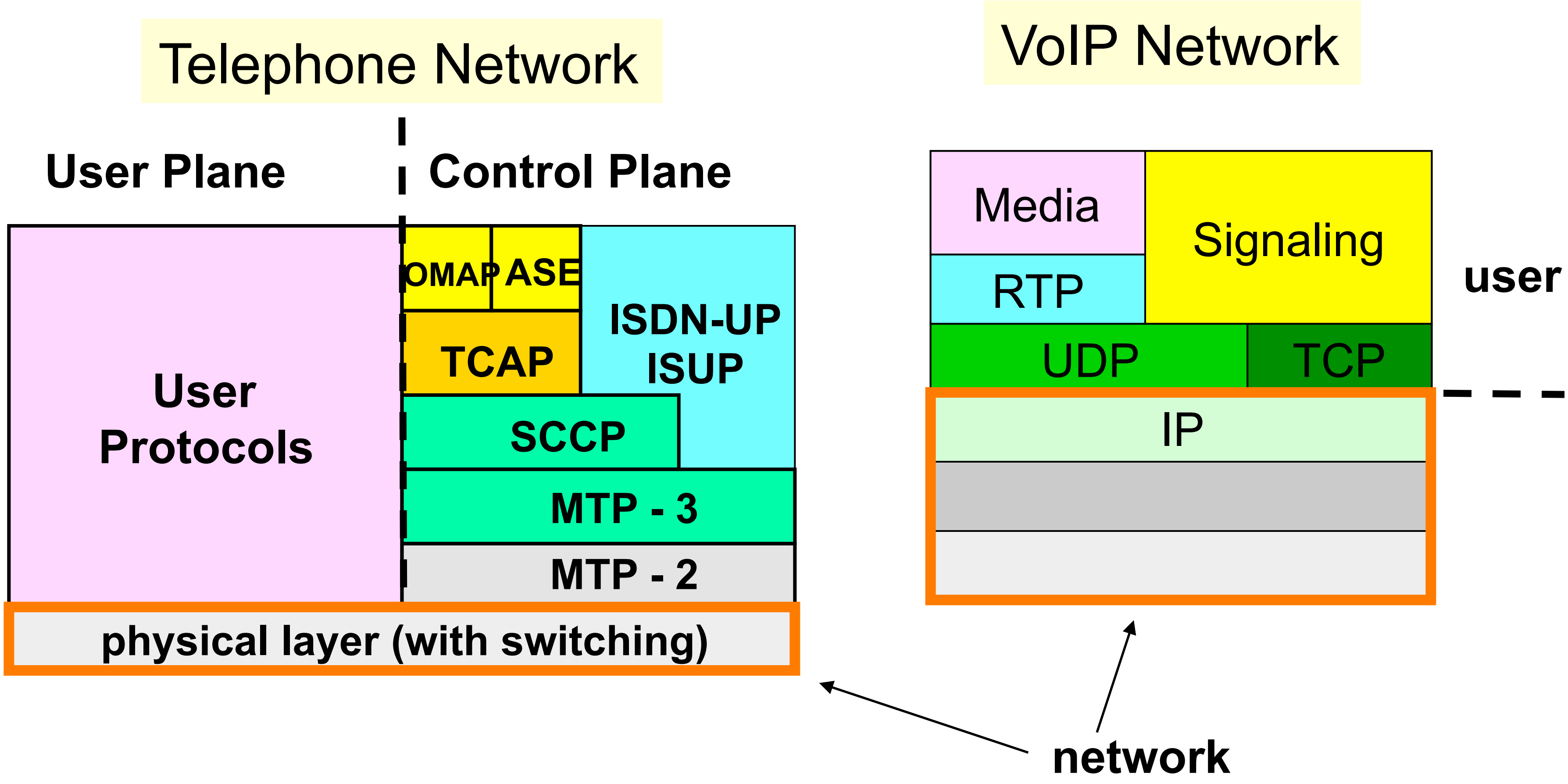


intelligence and functions are in hosts: i.e. terminal devices

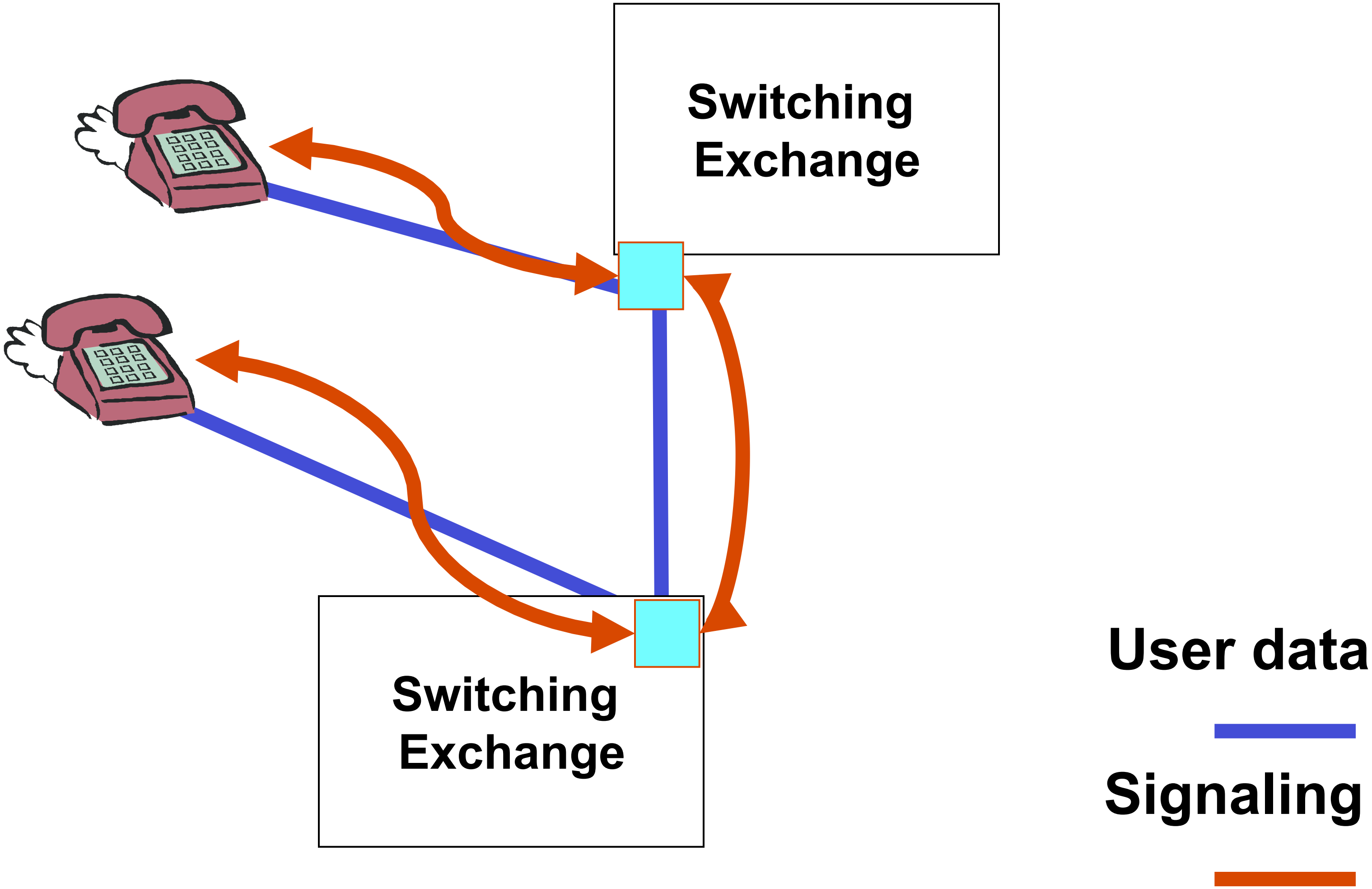
a single network for signaling and transport

packet based transport network

Architectural difference



A Telephone network ...



Switching Exchange

Switching Exchange

User data

Signaling

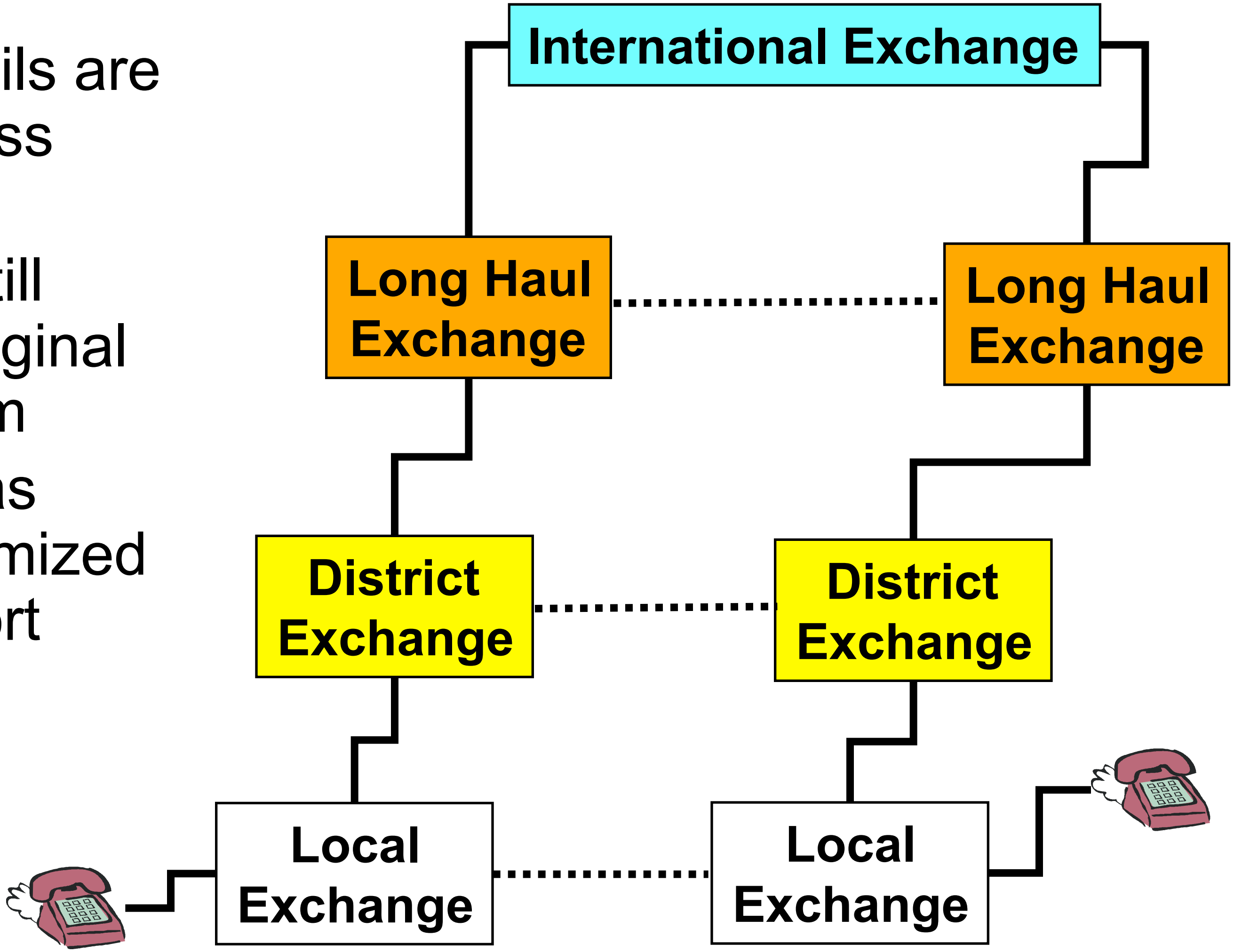
Hierarchical organization

Hierarchy levels,

Names and details are not uniform across countries.

Architecture is still biased by the original monopoly system

The structure was tailored and optimized for voice transport

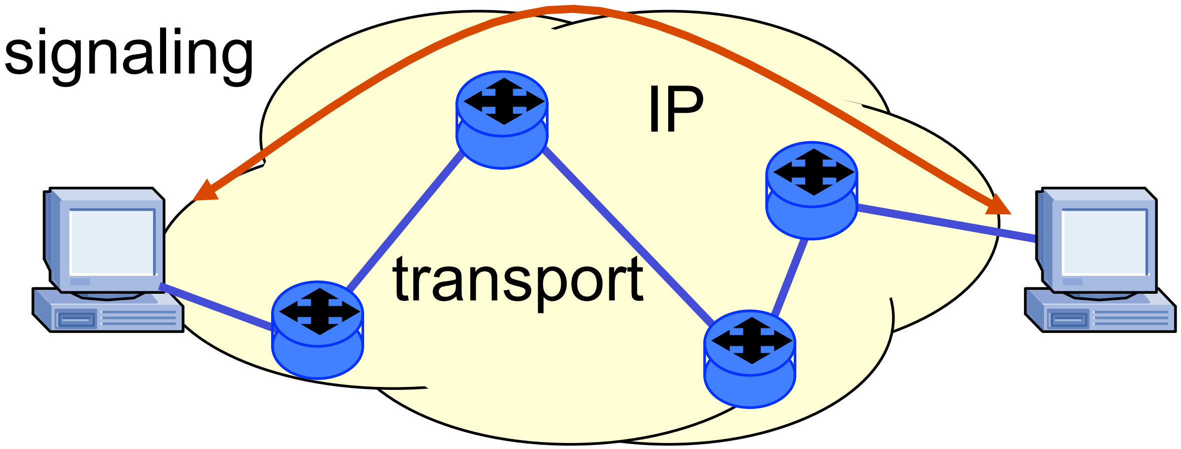


Service-specific problems

- Voice is “just another service”, but …
- Is it possible to realize e-t-e conversational services without involving the network layer?
- Signaling in telephony has application-level functionalities
  - access
  - callee identification
  - negotiation of characteristics and quality
  - billing and accounting ...
- But also control function on the transport channel
  - routing and setup
  - resource finding and reservation

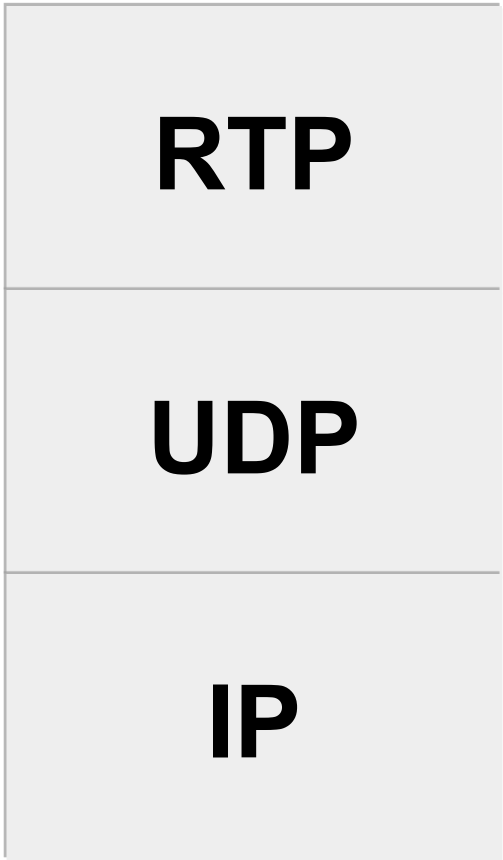
Service-specific problems

- Application level signaling are simplified by the IP e-t-e approach ☺
- Network services for the control of the channel (e.g. QoS) simply do not exist in IP ☹
- Routing is not controllable (no alternate routing), hot-swap reliability is not present, QoS control is almost impossible unless by "circuit-like" dimensioning. ☹



Real-Time Transport in IP

- Real Time (Transport) Protocol
- Developed by Audio Video Transport Working Group of IETF
- RFC 1889 obsoleted by 3550/3551
- It is an add-on to UDP building a connection-oriented unreliable channel
- Adds and header with information for:
  - Multimedia data management (coding, timestamping, etc.)
  - Error and QoS control (feedback on the reverse channel)



RTP

UDP

IP

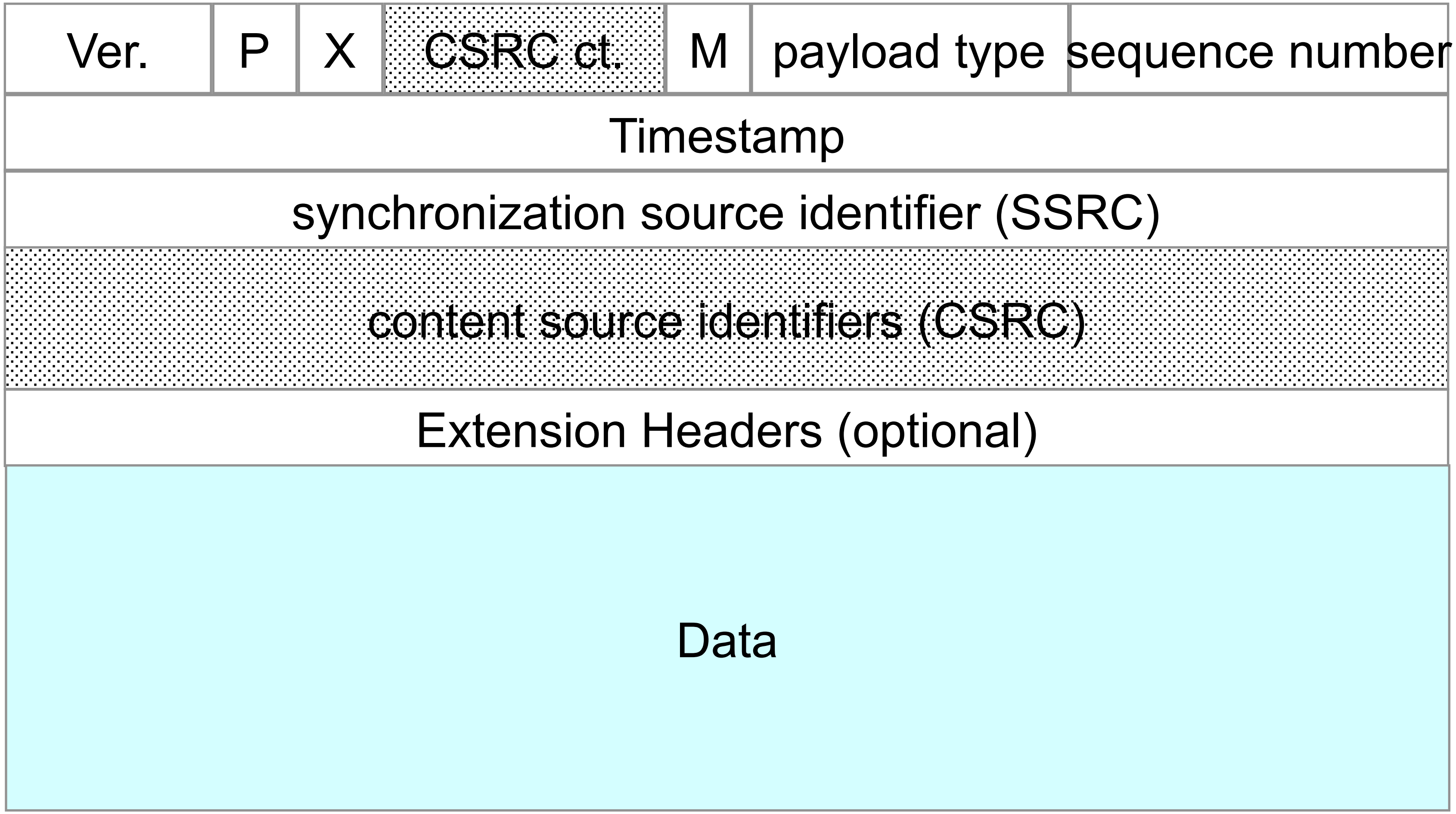
RTP: characteristics and functionalities

- Independent from the PHY (obvious!!!)
- Scalable
  - Unicast e multicast
- Defines separate logical channels for data and control
  - indeed a “pair” of protocols RTP-RTC$_{\text{ontrol}}$P
- Packet reordering at destination
- Delay jitter equalization with buffers (in addition to the playout buffer of the application)
- Sender identification
- Intra-media synchronization
- No predefined Port, but must be even

Some (more) RTP Definitions

- Transport address: network address + port (as usual)
- RTP session: communications on a pair of transport addresses (data + control)
  - A Multimedia Session requires at least Audio+Video, coordination and synchronization of different flows (RTP sessions) is up to the application
- Synchronization source (SSRC): the source of a stream of RTP packets
  - Identified by 32-bit SSRC identifier
  - All packets from the same SSRC form a single timing and sequencing space. Receivers group packets by SSRC for playback
  - Not dependent on network address
  - Examples: all packets from a camera or from a microphone
  - A participant need not use the same SSRC for all RTP sessions in a multimedia session

RTP: header format



<table>
<tr><td>Ver.</td><td>P</td><td>X</td><td>CSRC ct.</td><td>M</td><td>payload type</td><td>sequence number</td></tr>
<tr><td colspan="7">Timestamp</td></tr>
<tr><td colspan="7">synchronization source identifier (SSRC)</td></tr>
<tr><td colspan="7">content source identifiers (CSRC)</td></tr>
<tr><td colspan="7">Extension Headers (optional)</td></tr>
<tr><td colspan="7">Data</td></tr>
</table>

The RTP header (12 bytes)

- Ver.(2 bits): Version of the protocol. Current is 2
- P (1 bit): Indicate if there are extra padding bytes at the end of the RTP packet.
- X (1 bit): Extensions to the protocol used (ELH present)
- CC (4 bits): Number of CSRC identifiers that follow the fixed header
- M (1 bit): If set means that the current data has some special relevance for the application defined in a profile (external to the protocol)
- PT (7 bits): Format of the payload and its interpretation by the application

The RTP header (12 bytes)

- SSRC: Indicates the synchronization source and timing (32 bits)
- CSRC list
- – 0 to 15 items, 32 bits each
- – Identifies the contributing sources for the payload contained in this packet
- – The number of identifiers is given by the CC field.
- – Only 15 may be identified
- – CSRC identifiers are inserted by mixers, using the SSRC identifiers of contributing sources.
- Extension header: Length of the extension (EHL=extension header length) in 32bit units, excluding the 32bits of the extension header

RTCP

- Real Time Control Protocol
- Functionalities:
  - Data Distribution Control
  - Session information advertisement (during the session, not for setup)
  - QoS feedback
  - Error reporting
  - …
- RTCP messages are sent on RTP-port+1

RTCP

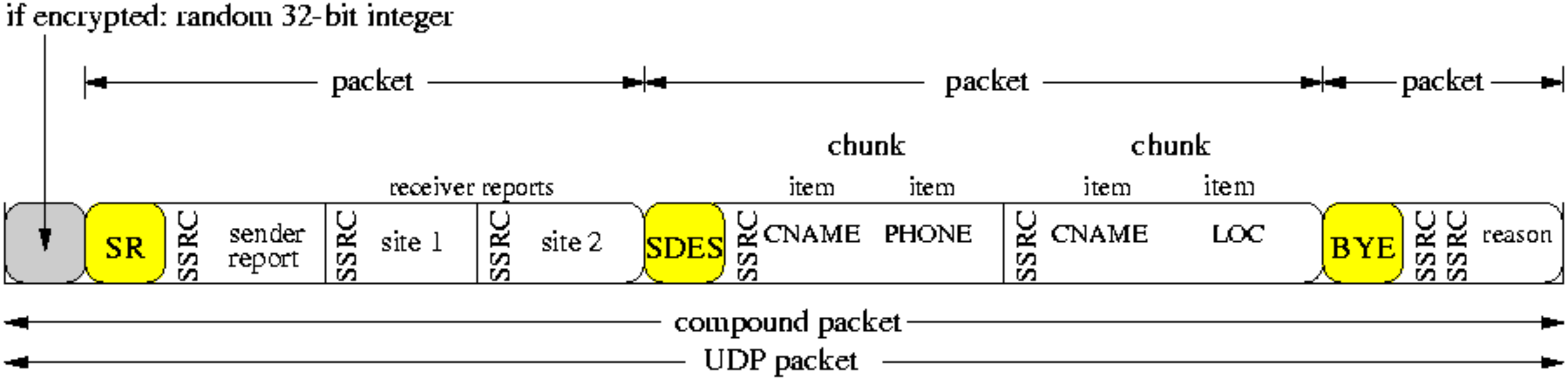
- Provide feedback on the quality of data distribution
  - Through sender and receiver reports;
  - For adaptive encoding (adaptive to network congestion);
  - Can be used to diagnose faults
- RTCP carries a persistent transport-level identifier for an RTP source, called *canonical name*, CNAME
  - Receivers use CNAME to keep track of each participant
  - And to synchronize related media streams (with the help of NTP)
- Passes participant’s identification for display

RTCP Packets

- SR: sender reports; sending and reception statistics
- RR: receiver reports; for reception statistics from multiple sources
- SDES: source description item, include CNAME
- BYE: indicates end of participation
- APP: application specific functions

Compound RTCP Packets

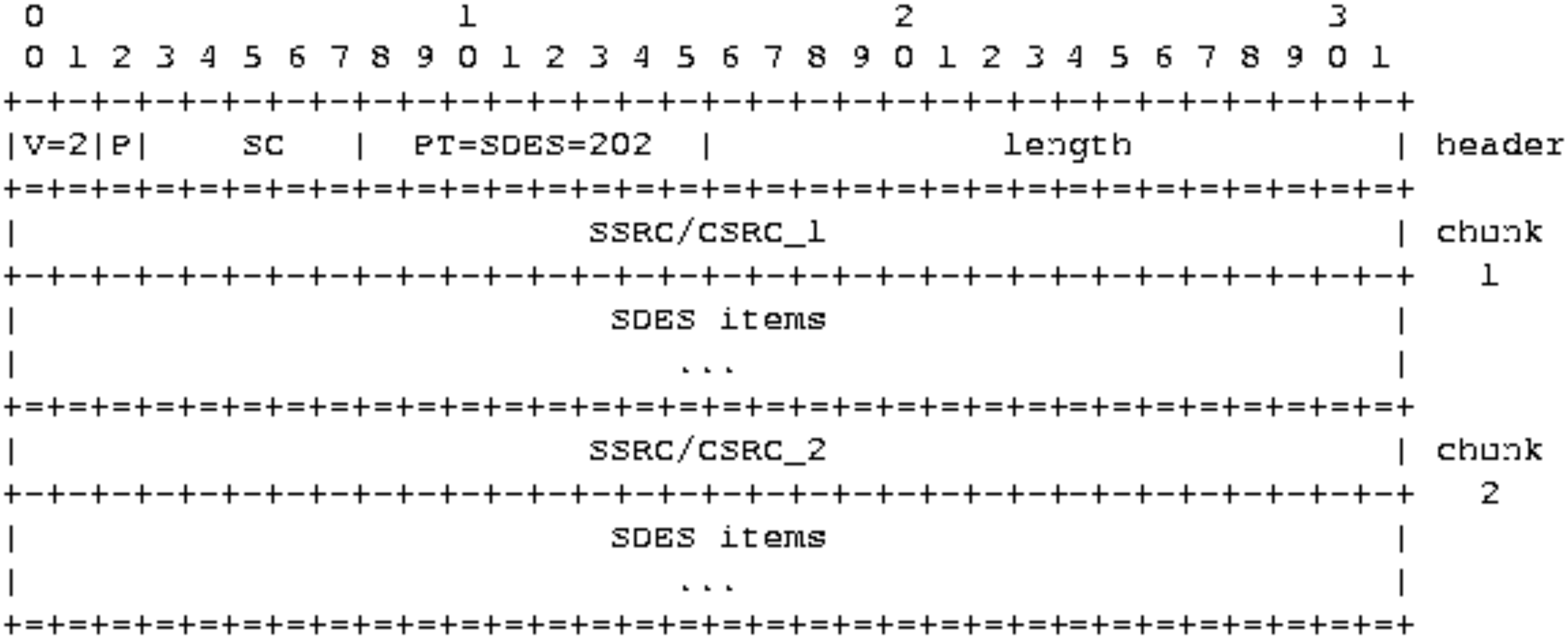
- A compound RTCP packet contains multiple RTCP messages of the previous types
- Example:



SR Packet – Cont.

- RC: receiver report count
- Length: in 32-bit words – 1
- NTP ts: wallclock time, used to calculate RTT
- RTP ts: in unit and offset of RTP ts in data packets. Can be used with NTP ts for inter-media synchronization.
- Fraction lost: since the last RR or SR packet was sent. Short term loss ratio.
- LSR: last SRT time stamp; middle 32 bits of NTP timestamp.
- DLSR: delay since last SR; expressed in 1/65536 seconds between receiving the last SR packet from SSRC_n and sending this report. Source SSRC_n can compute RTT using DLSR, LSR and the reception time of the report, A.
  RTT = A – LSR – DLSR
- An application's profile can define extensions to SR or RR packets

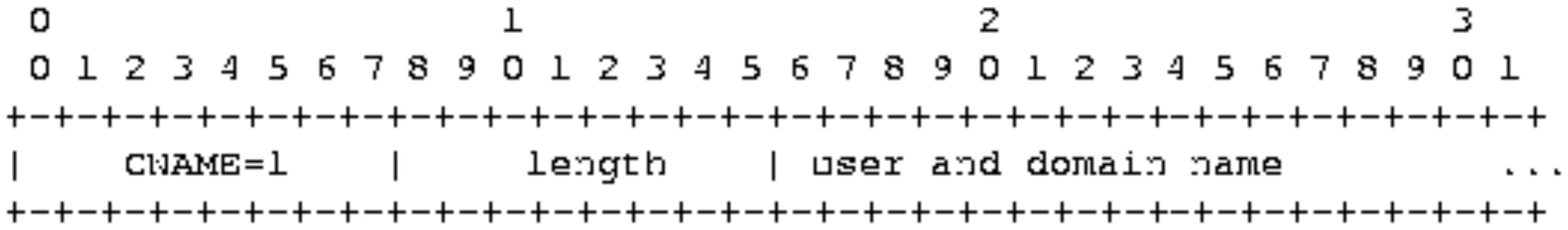
SDES Packet



```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |V=2|P|    SC   |  PT=SDES=202  |             length            | header
       +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
       |                          SSRC/CSRC_1                          | chunk
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   1
       |                           SDES items                          |
       |                              ...                              |
       +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
       |                          SSRC/CSRC_2                          | chunk
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   2
       |                           SDES items                          |
       |                              ...                              |
       +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

CNAME Item in SDES Packet

- Mandatory
- Provides a persistent identifier for a source
- Provides a binding across multiple media used by one participant in a set of related RTP sessions. CNAME should be fixed for that participant.
- SSRC is bound to CNAME
- Example: doe@sleepy.megacorp.com or doe@192.0.2.89, or, ...



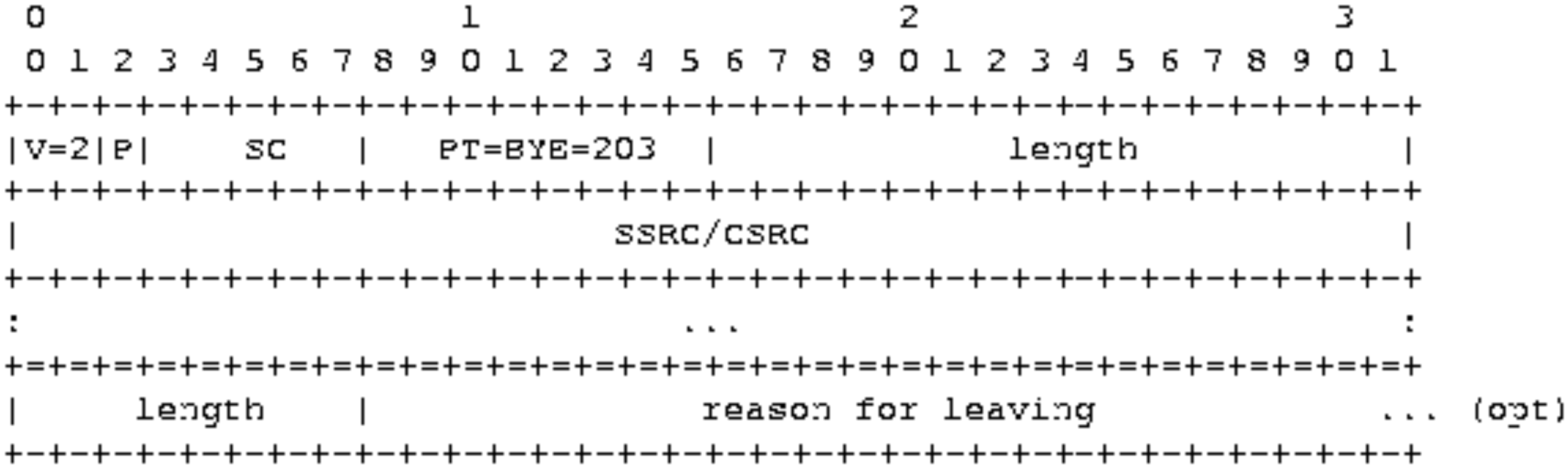
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    CNAME=1    |     length    | user and domain name        ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Other Items in SDES Packet

- NAME: user name
- EMAIL:
- PHONE:
- LOC: location
- TOOL: application or tool name
- NOTE: notice/status

BYE: Goodbye RTCP Packet

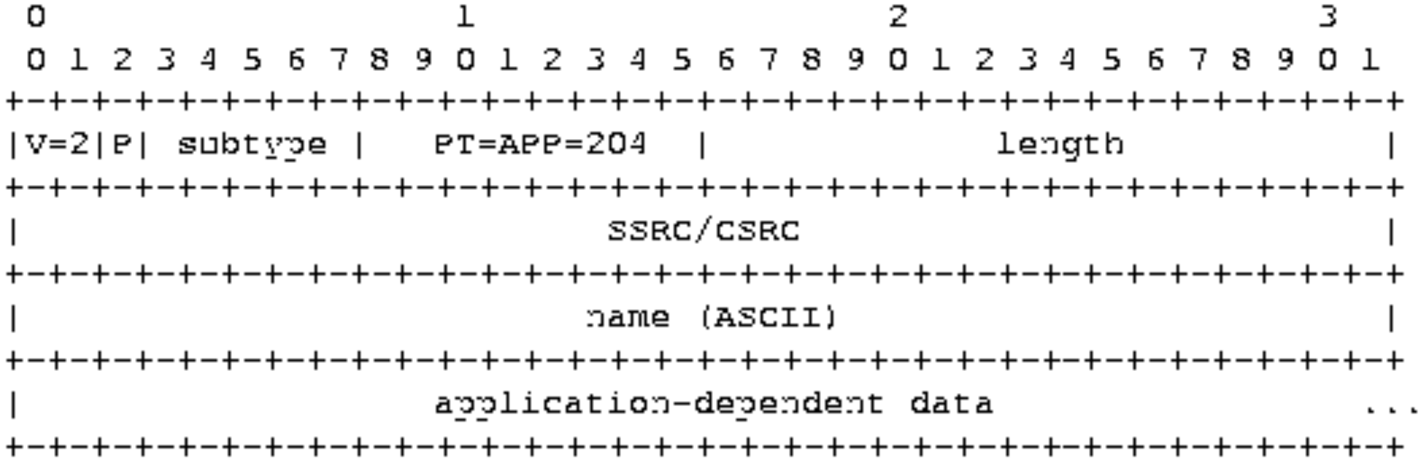
- Mixers should forward the BYE packet with SSRC/CSRC unchanged
- Reason for leaving: string field; e.g., “camera malfunction”



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|    SC   |   PT=BYE=203  |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SSRC/CSRC                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                              ...                              :
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|     length    |               reason for leaving            ... (opt)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

APP RTCP Packet

- Subtype: allows a set of APP packets to be defined under one unique name
- Name: unique name in the scope of one this application



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P| subtype |   PT=APP=204  |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SSRC/CSRC                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          name (ASCII)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   application-dependent data                ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

RTP Algorithms - I

- RTCP packets generation: need to limit the control traffic
  - Control traffic should takes about 5% of data traffic bandwidth
  - ¼ of the RTCP bandwidth is used by senders
  - Interval between RTCP packets scales linearly with the number of members in the group
  - Each compound RTCP packet must include a report packet and a SDES packet for timely feedback

RTP Algorithms - II

- SSRCs are chosen randomly and locally and can collide
- Loops introduced by mixers and translators
  - A translator may incorrectly forward a packet to the same multicast group from which it has received the packet
  - Parallel translators
- Collision avoidance of SSRC and loop detection are entangled

Example of A Profile Document

RFC3551: RTP Profile for Audio and Video Conferences with Minimal Control.

- RTP data header:
  - use one marker bit
  - No additional fixed fields
  - No RTP header extensions are defined
- RTCP
  - No additional RTCP packet types
  - No SR/RR extensions are defined
  - SDES use: CNAME is sent every reporting interval, other items should be sent only every fifth reporting interval

VoIP: Signaling Protocols

Standardized

H.323

SiP

H.248

Proprietary

skype™

wengo connecting your phone to the future

Asterisk™
(IAX, open source)

YAHOO!® MESSENGER

AOL.Instant Messenger™

CISCO SYSTEMS
(Skinny)

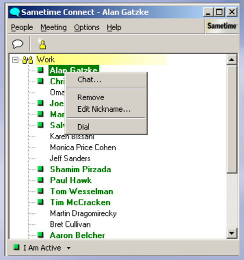
… and many many more…

Applications based on “VoIP” protocols

Collaboration

Calendar

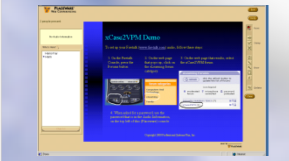
Instant Messaging



Sametime Connect - Alan Gatzke

Work
- Alan Gatzke
- Chri
- Oma
- Joe
- Mar
- Salv
- Karen Bissani
- Monica Price Cohen
- Jeff Sanders
- Shamim Pirzada
- Paul Hawk
- Tom Wesselman
- Tim McCracken
- Martin Dragomirecky
- Bret Cullivan
- Aaron Belcher

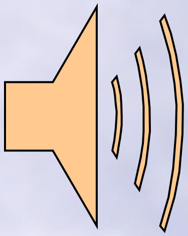
Web Application



email

Video Conferencing

Audio Conferencing



Voice Messaging

Telephone Services

Brief History of VoIP (1)

- Sharing expensive lines (end of ’90)
  - VoIP enters the enterprise market as a way to save telecom (transmission) cost by using excess data capacity for Voice
  - using the same lines for data and voice communication
    - utilizing existing Local Area Networks (LANs) and WAN connectivity for voice communication, i.e. reduce enterprises’ bill from PSTN operator
  - ITU-T promotes H.323 as protocol (ISDN-style VoIP protocol)

Brief History of VoIP (2)

- Network Convergence (beginning of ‘00)
  - network convergence:
    - data over ISDN was initially successful in some countries (Ger, J) but usage price was high and bandwidth was soon too limited
    - when Internet bandwidth became abundant – VoIP success started
  - IETF completes standardization of an Internet-style VoIP signaling protocol: Session Initiation Protocol (SIP) and media transport protocol (RTP)
  - Internet (IP) becomes the new Integrated Services Digital Network
  - Operator’s convergence began with VoIP in the backbone
  - only lately moving to the access (2005+)

Brief History of VoIP (3)

- SIP becomes the dominant VoIP protocol (‘00 until now)
  - H.323 had the earlier start, but more oriented towards local networks
  - ISDN-style H.323 was more liked by traditional operators
  - SIP is a text-based protocol on to of IP, much like HTTP and XML
    - ➢therefore easy to understand for IP and/or web experts
    - ➢SIP better suited for large scale application
    - ➢efficiency is poor
    - ➢security threats
  - but SIP became the choice of Internet community
    - Standardized by IETF

Brief History of VoIP (4)

- **breakthrough:** SIP chosen by 3GPP as basis for IMS, i.e., all multimedia services (including VoIP) in 3G
- The consumer segment becomes aware of VoIP
  - Skype clients are widespread
    - using proprietary protocols
  - consumer market is not interested in standards – only costs
  - the business model of Skype – owned by ebay – is "the whole world can talk for free" – revenue is made through arbitrage:
    - Skype out / Skype in – Gateway to PSTN
    - advertising
  - advertisements, lack of privacy/security, quality are the price consumers pay

Today’s Situation

Three VoIP market segments

1. **enterprise**
2. **public operators**
3. **consumer**

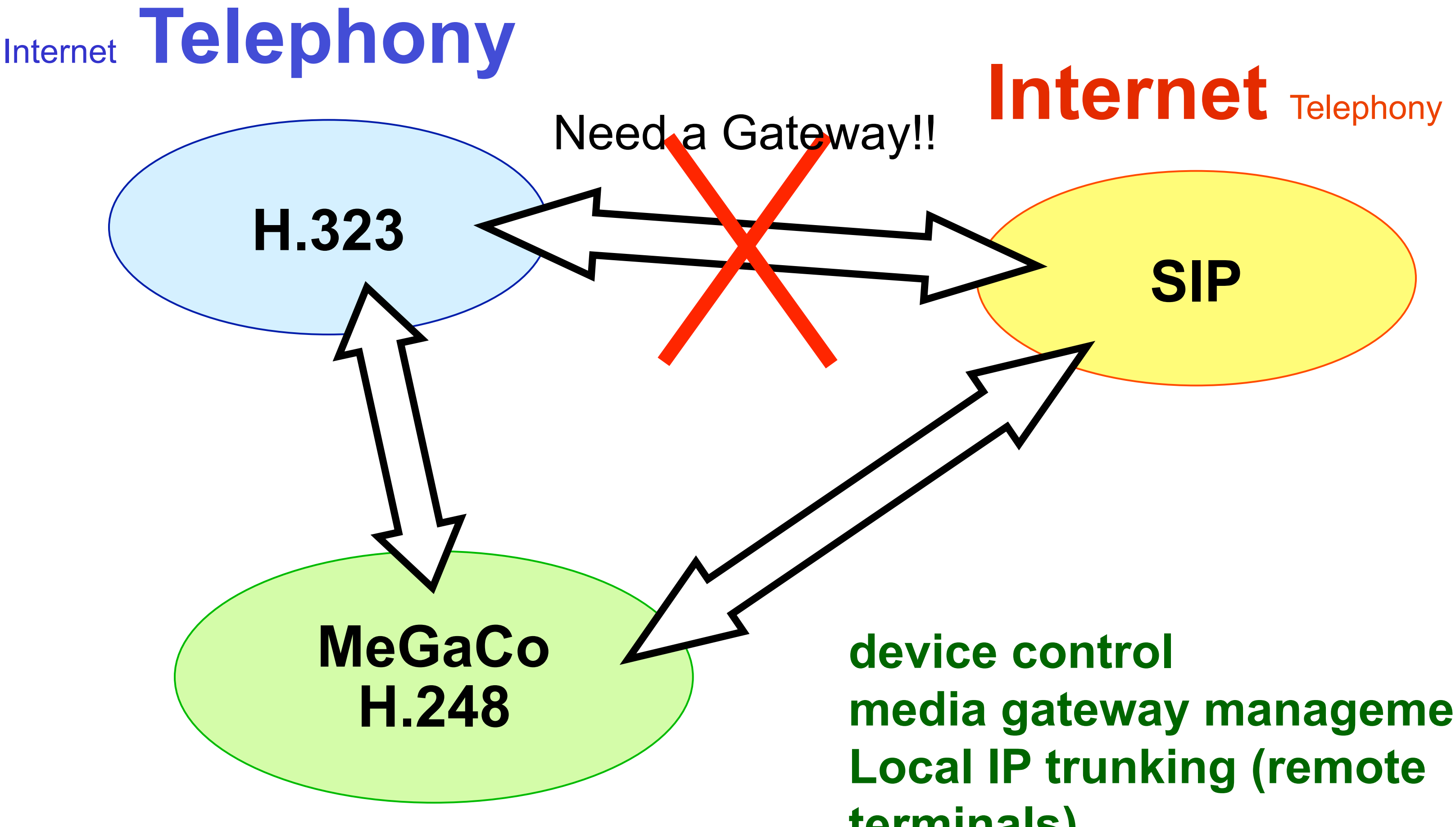
What about protocols?

- H.323 is still in the market but will probably die sooner or later ⇨ no point to get into H.323 market in 2006/7/8/9/... let alone today
- SIP is already dominating
  - today new investments are based on SIP
  - SIP large scale deployment still in the beginning
  - already dominating the corporate market
  - entering the operator market
- Proprietary protocols, e.g. Skype, are competing in consumer market only

Signaling and Service Protocols

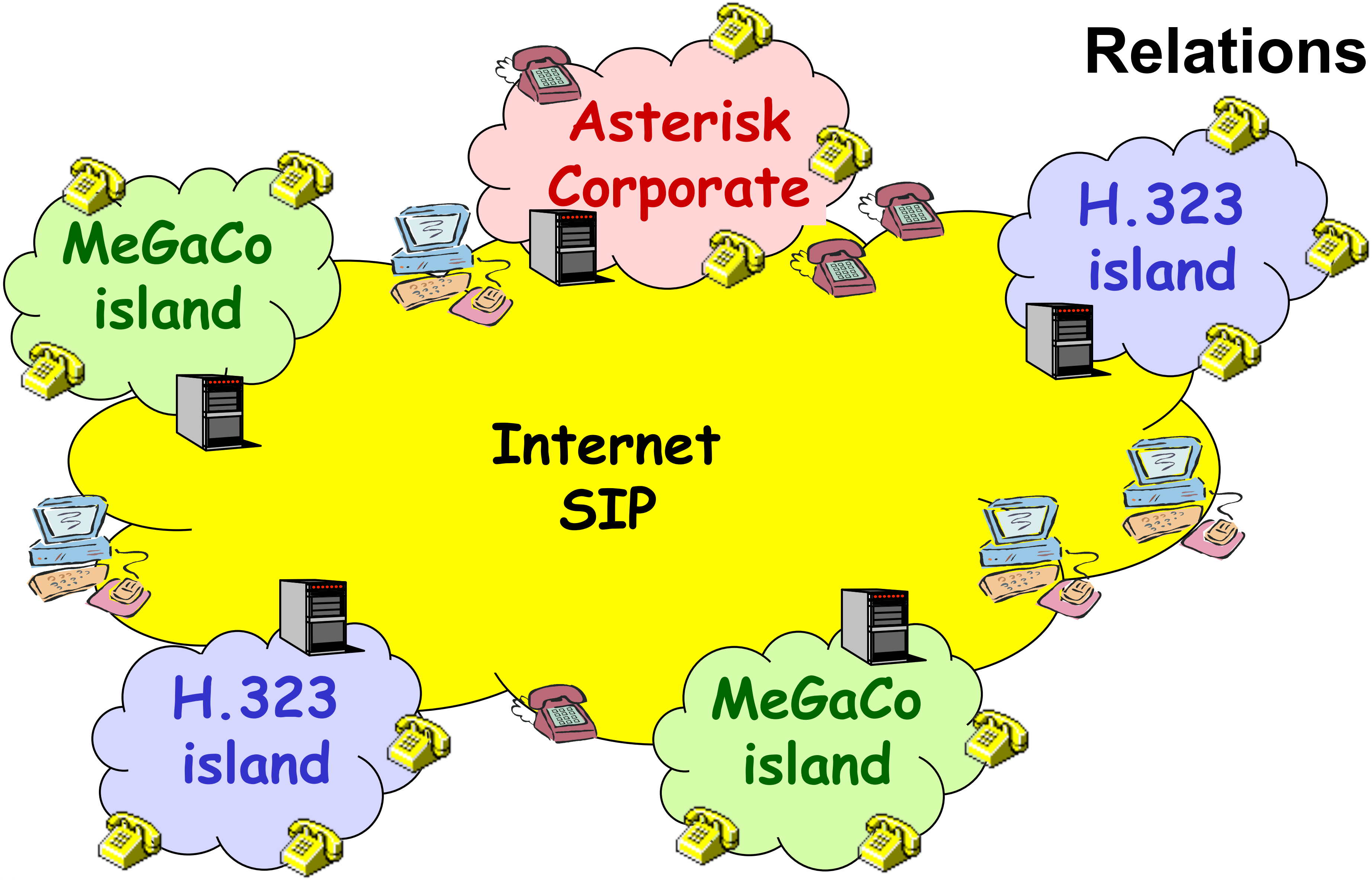
- H.323
  - Vertical, Hierarchic, Complex, Rigid, Omni-comprehensive "LAN oriented",
    not easy to integrate with PSTN
- SIP
  - Horizontale, Flat, Simple, "WAN oriented",
    impossible to integrate with PSTN (normally interoperates through a gateway based on Asterisk or similar)
- MeGaCo (H.248)
  - Vertical, Hierarchical, Complementary to H.323/SIP, Separates data and signaling for management, easy support for soft-switches
    PSTN-oriented, used locally to control media-gateways, not ment as "entire system"

Signaling Protocols



device control
media gateway management
Local IP trunking (remote terminals)

Protocols & Relations



Standard protocols: H.323

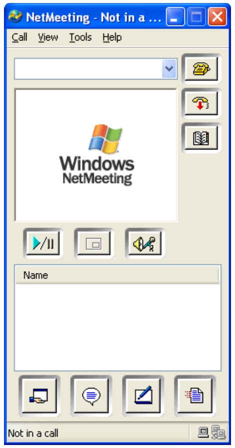
- H.323: “Packet-based multimedia communications systems”
  - recommendation from ITU-T
  - used to establish, modify, terminate multimedia sessions (e.g. VoIP calls)
  - it is based on H.320 (ISDN Videoconferencing)
    - multistage signaling
    - good interoperability with PSTN
    - it inherits its complexity
    - recent recommendations extend it to wide deployments
  - some operators deployments are still H.323-based
    - many operators have already SIP in their core network

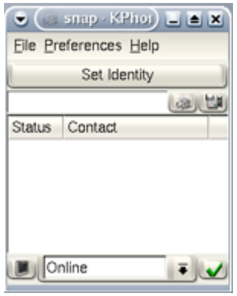
Standard protocols: SIP

- SIP: Session Initiation Protocol
  - IETF standard
  - used to establish, modify, terminate multimedia sessions (e.g. VoIP calls)
  - it is based on HTTP (light protocol)
  - it inherits its vulnerabilities
  - easily extensible
- It supports name mapping and redirection services transparently
  - personal mobility: one single externally visible identifier regardless of the network location
- Where is SIP used?
  - corporate deployments
  - 3GPP IMS (PS signaling protocol)
  - TISPAN NGN will be based on core IMS and thus on SIP as well

VoIP: architectural elements (I)

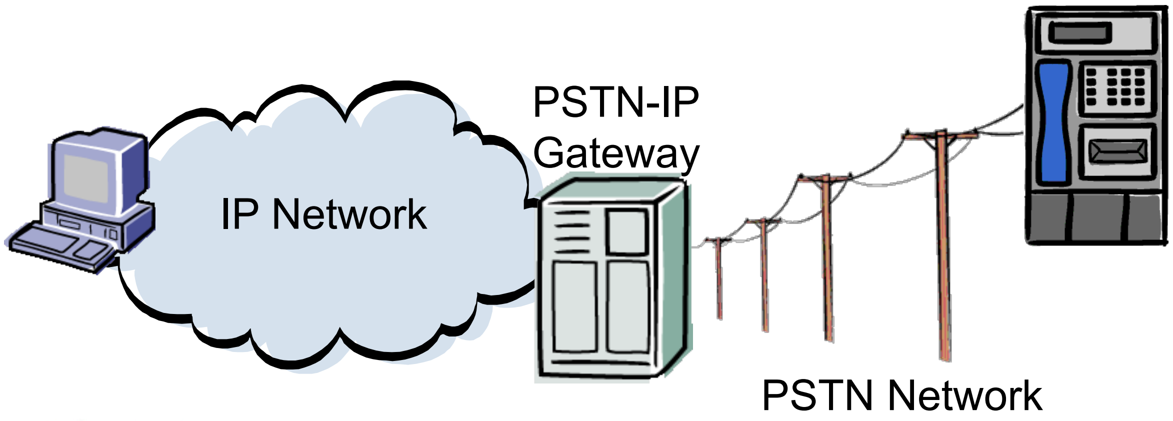
- Terminals (end-points)
  - hardware clients
  - software clients
  - optional
    - video codec
    - data transmission
    - instant message
    - presence





VoIP: architectural elements (II)

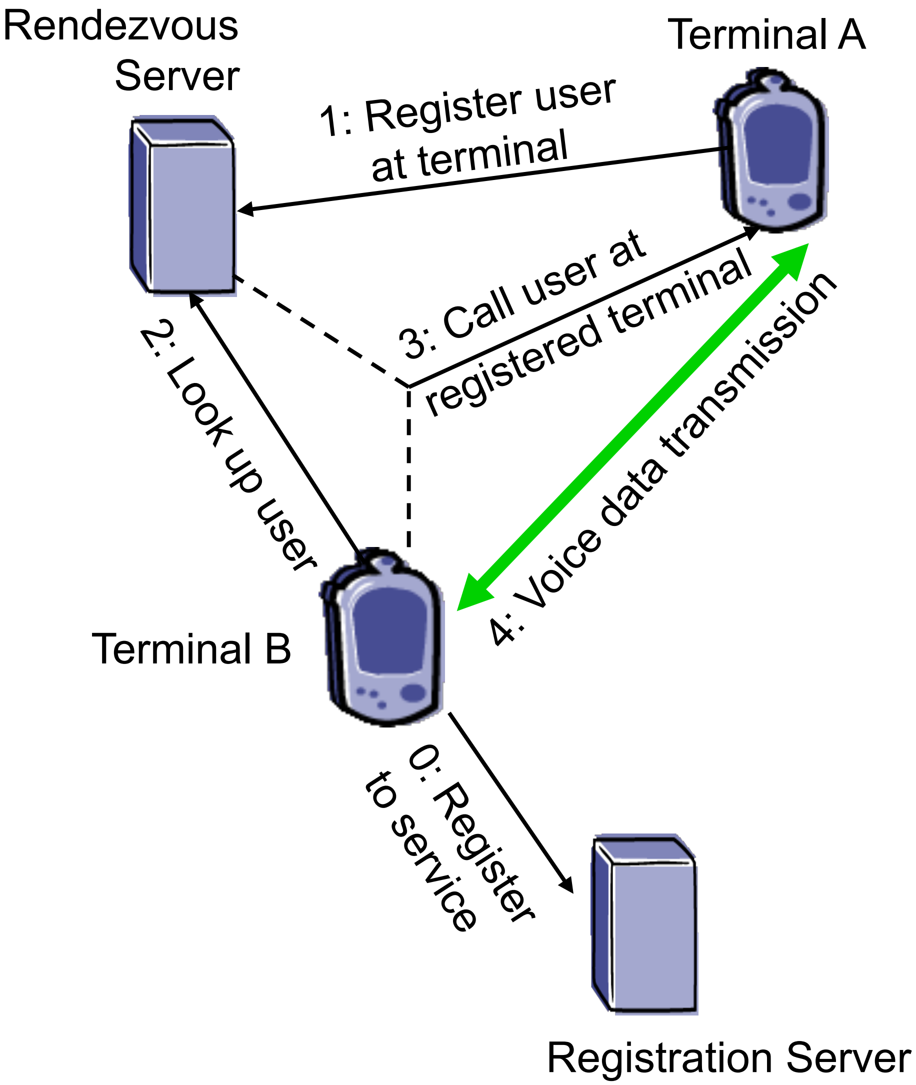
- Gateway
  - generic: an interface between two worlds
  - specific: interface between packet-based networks and circuit switched networks or between different architectures in packet-based networks (e.g. SIP—Skype)



VoIP: architectural elements (III)

- Rendezvous server
  - H.323 world: Gatekeeper
  - SIP world: Proxy server
  - Main functionalities
    - Managing entities in its domain
      - Endpoint registration
    - Address translation
      - user identity to to terminal location
    - Call routing
      - Next hop location

- Additional servers
  - application servers
    - registration servers
    - conferencing server
    - presence server
    - etc.



SIP: Session Initiation Protocol

- Defined by IETF
- RFC 2543 (first release march 1999)
  - many other RFCs ... see IETF site and later on
- Multiparty MUltimedia Session Control (Mmusic) WG
- Born from Mbone experience and as a more “Internet” alternative to H.323

IETF service vision

- First objective is connectivity
  - Transport through IP
  - Iintelligence is in hosts and not in network nodes (routers) which only switch and forward datagrams
- Scalability and Security are primary concerns … although scalability is addressed, while security …
- SIP is an umbrella protocol suite using other light mono-function protocols
  - Avoid function duplication
  - Modular development

SIP: geenral characteristics

- Client – server protocol
- The usage is: "invite" users in participation to multimedia sessions
- Uses several http-derived functionalities
- Independent from the transport layer
- Should be Scalabile, Modular and Simple
- Defining a suite is based on the use of other protocols
  - SDP: Session Description Protocol
  - SAP: Session Announcement Protocol
  - RTP/RTCP (voice/video conversational transport)
  - RTSP: Real Time Streaming Protocol (VoD like)

SIP: A General Purpose Session Control Protocol?

- SIP is not limited to IP telephony
  - SDP quite flexible
  - arbitrary payloads allowed
- Other applications relying on notion of session:
  - distributed virtual reality systems
  - network games
  - video conferencing
- Applications may leverage SIP infrastructure (Call Processing, User Location, Auth., etc.)
  - Instant Messaging and Presence
  - SIP for Appliances !?!?!?

SIP: it’s not…

- A transport Protocol
- A QoS Reservation Protocol
- A gateway Control Protocol
- It does NOT dictate ...
  - product features and services (color of your phone and distinctive ringing melodies, number of simultaneous calls your phone can handle, don’t disturb feature, ...)
  - network configuration

SIP: Architectural Elements

- Client (o end system)
  - Send SIP requests
  - Normally embedded into a SIP User Agent Server
- User Agent Server (UAS)
  - Answers incoming queries and calls
- Redirect Server
  - Redirect calls to another server
- Proxy Server
  - Send Requests to another server, including UASs

SIP: Addresses and Methods

- Addresses are URI (Universal Resource Identifier):
  - sip:jdrosen@bell-labs.com:5067
  - sip:ann:passwd@lucent.com
- 6 methods (or primitives):
  - INVITE: Starts or invite to a converence
  - BYE: Closes a participation
  - CANCEL: Terminates a search (unsuccesful) OPTIONS: Query a client on his “capabilities”
  - ACK: Accept a call (IVITE)
  - REGISTER: Registers a client onto a server, normally a proxy, include location information

SIP: Message syntax

- Derived from HTTP:

```
INVITE gerla@cs.ucla.edu SIP/2.0
From: locigno@dit.unitn.it (Renato Lo Cigno)
Subject: Next visit to L.A.
To: gerla@cs.ucla.edu (Mario Gerla)
Call-ID: 1999284605.56.86@
Content-type: application/sdp
CSeq: 4711
Content-Length: 187
```

- Make use of the Session Description Protocol (SDP)

Session Description Protocol

- Textual syntax for multimedia sessions (unicast and multicast)
- Basic characteristics
  - Describes Audio/Video flows that from the session and the related parameters
  - Includes addresses (internal ports) for the termination of different streams
  - “Commands” initial and termination times

SDP: an example

v=0 **Protocol version**

Creator and session identifier **<address type>**

o= locigno 28908044538 289080890 IN IP4 93.175.132.118

<username> **<session id>** **<version>** **<network type>** **<address>**

s=SIP Tutorial **Session name**

e=ghittino@csp.it **Email address**

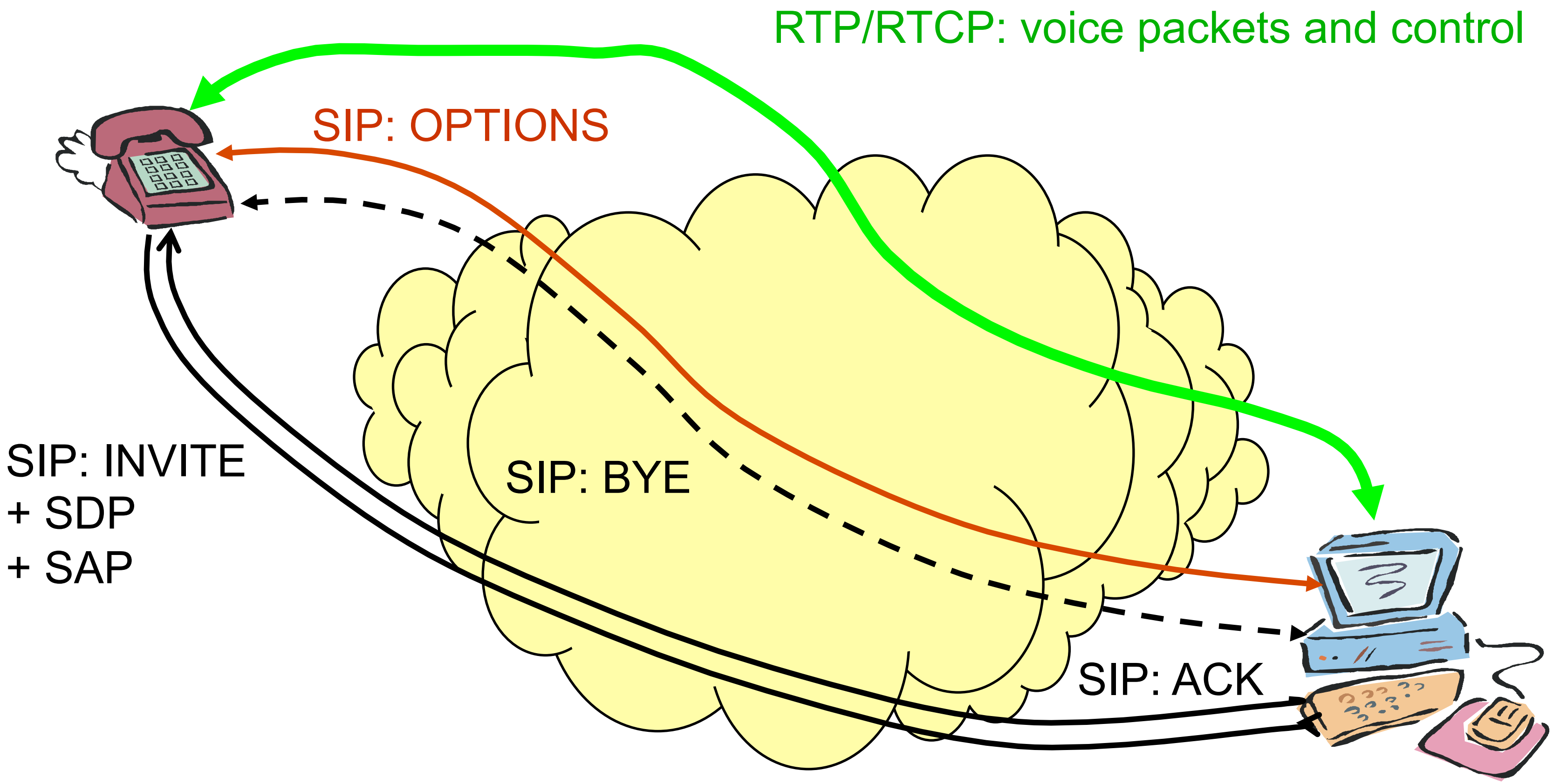
c=IN IP4 126.16.69.4 **Connection information**

t=28908044900 28908045000 **Time the session is active (start – stop)**

m=audio 49170 RTP/AVP 0 98 **Media name and transport address**

a=rtpmap:98 L16/11025/2 **Media attribute line**

SIP: Voice Call example



SIP Servers and Clients

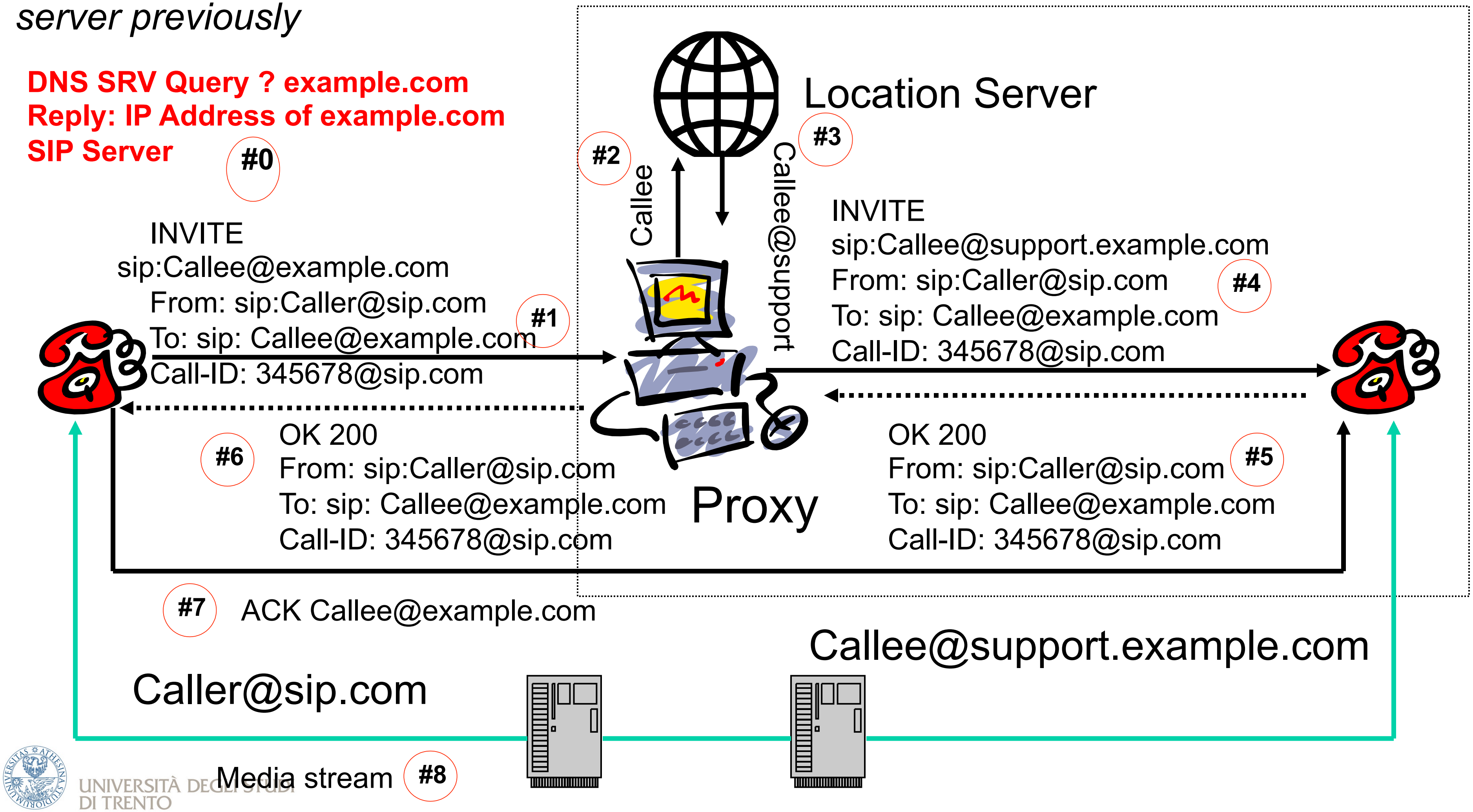
- User Agent (user application)
  - UA Client (originates calls)
  - UA Server (listens for incoming calls)
  - both SW and HW available
- SIP Proxy Server
  - relays call signaling, i.e. acts as both client and server
- SIP Redirect Server
  - redirects callers to other servers
- SIP Registrar
  - accept registration requests from users
  - maintains user’s whereabouts at a Location Server (like GSM HLR)

Proxy Server Functionality

- Serve as rendezvous point at which callees are globally reachable
- Perform routing function, i.e., determine to which UA/proxy/redirect an incoming call should be relayed
- **Allow the routing function to be programmable**
- **Forking: Several destinations may be tried for a request sequentially or in parallel**
- May serve as AAA trigger points

SIP Operation in Proxy Mode

User Caler@sip.com on left-hand side is initiating a call to Callee@example.com on right-hand side; Callee registered with his server previously



DNS SRV Query ? example.com
Reply: IP Address of example.com
SIP Server

#0

#1
INVITE
sip:Callee@example.com
From: sip:Caller@sip.com
To: sip: Callee@example.com
Call-ID: 345678@sip.com

#2
Callee

#3
Callee@support

Location Server

Proxy

#4
INVITE
sip:Callee@support.example.com
From: sip:Caller@sip.com
To: sip: Callee@example.com
Call-ID: 345678@sip.com

#5
OK 200
From: sip:Caller@sip.com
To: sip: Callee@example.com
Call-ID: 345678@sip.com

#6
OK 200
From: sip:Caller@sip.com
To: sip: Callee@example.com
Call-ID: 345678@sip.com

#7
ACK Callee@example.com

Caller@sip.com

Callee@support.example.com

#8
Media stream

SIP RFC2543 Methods

- INVITE: initiates sessions
    - Request URI indicated destination; may be changed on the path
    - session description included in message body
    - re-INVITEs used to change session state
- ACK: confirms session establishment
    - can only be used with INVITE
- BYE: terminates sessions
- CANCEL: cancels a pending INVITE
    - if a CANCEL follows a RE-INVITE the session is not torn down!
- OPTIONS: capability inquiry
    - replied as INVITE
    - may include Allow, Accept, Accept-Encoding, Accept-Language, Supported,...
- REGISTER:

SIP REGISTER Method

- REGISTER binds a permanent address to current location
- similar to registering with HLRs in GSM
- REGISTERs may be multicast
- may convey user data (e.g., CPL scripts)
- default registration timeout: 3600 s
- may be also used to cancel or query existing registrations

SIP Response Codes

- **Borrowed from HTTP: xyz explanatory text**
- **Receivers need to understand x**
- **x80 and higher codes avoid conflicts with future HTTP response codes**

- **1yz** Informational
  - 100 Trying
  - 180 Ringing (processed locally)
  - 181 Call is Being Forwarded
- **2yz** Success
  - 200 ok
- **3yz** Redirection
  - 300 Multiple Choices
  - 301 Moved Permanently
  - 302 Moved Temporarily
- **4yz** Client error
  - 400 Bad Request
  - 401 Unauthorized
  - 404 Not Found
  - 405 Method not Allowed
  - 407 Proxy Authentication Required
  - 415 Unsupported Media Type
  - 482 Loop Detected
  - 486 Busy Here
- **5yz** Server failure
  - 500 Server Internal Error
- **6yz** Global Failure
  - 600 Busy Everywhere

SIP Message Structure

Request Method

INVITE sip:UserB@there.com SIP/2.0

Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Subject: Happy Christmas
Contact: BigGuy <sip:UserA@here.com>
Content-Type: application/sdp
Content-Length: 147

Message Header Fields

v=0
o=UserA 2890844526 2890844526 IN IP4 here.com
s=Session SDP
c=IN IP4 100.101.102.103
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000

Payload

Response Status

SIP/2.0 200 OK

Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>;tag=65a35
Call-ID: 12345601@here.com
CSeq: 1 INVITE
Subject: Happy Christmas
Contact: LittleGuy <sip:UserB@there.com>
Content-Type: application/sdp
Content-Length: 134

v=0
o=UserB 2890844527 2890844527 IN IP4 there.com
s=Session SDP
c=IN IP4 110.111.112.113
t=0 0
m=audio 3456 RTP/AVP 0

"receive RTP G.711-encoded audio at 100.101.102.103:49172"

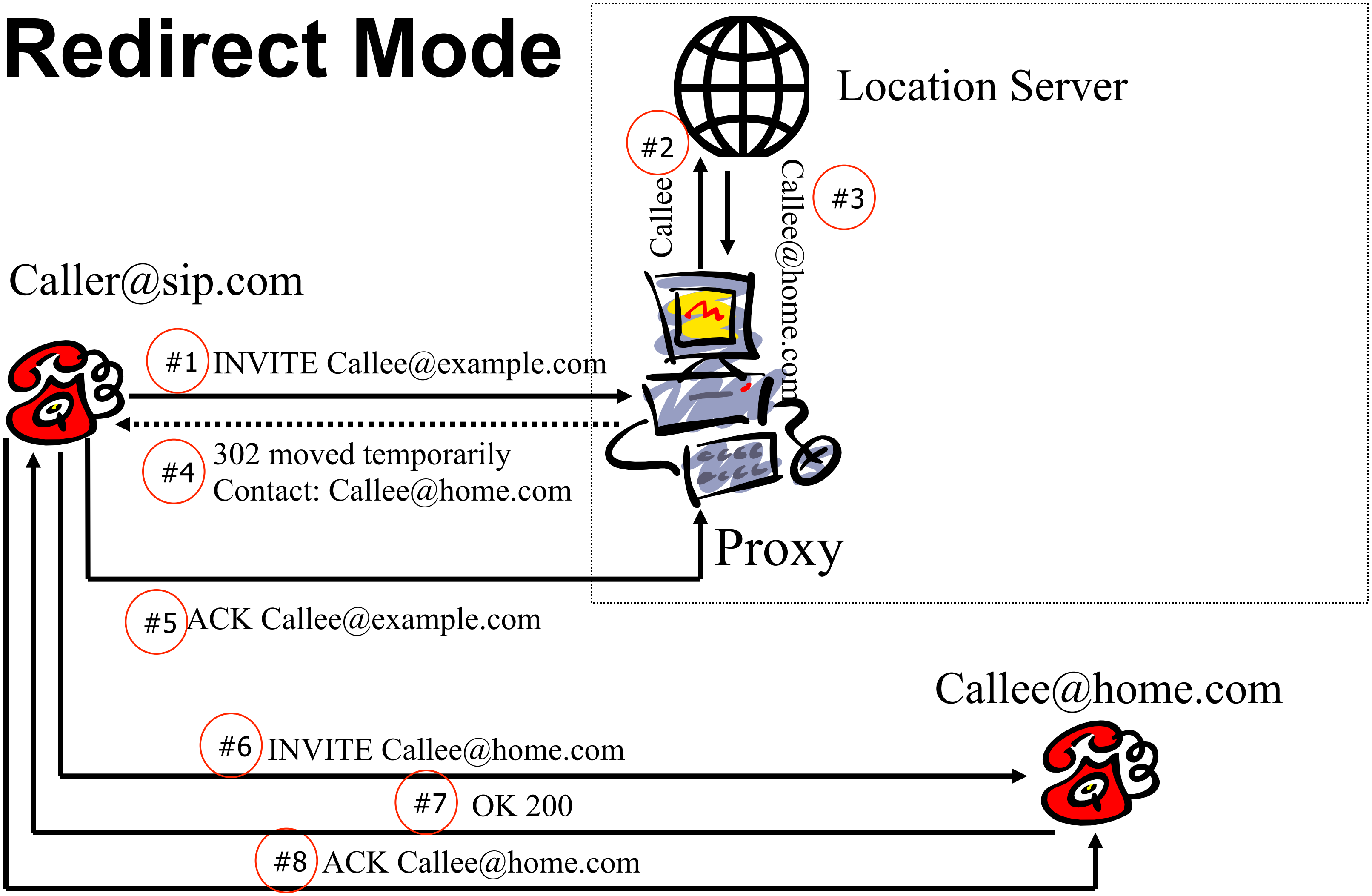
SIP Addresses

- URLs used to identify a call party a human being or an automated service
- examples:
  - sip:voicemail@examples.com?subject=callme
  - sip:sales@bigcom.com; geo.position:=48.54_-123.84_120
- must include host, may include user name, port number, parameters (e.g., transport), etc.
- may be embedded in Webpages, email signatures, printed on your business card, etc.
- address space unlimited
- non-SIP URLs can be used as well (mailto:, http:, ...)

SIP Server -- Proxy versus Redirection

- A SIP server may either **proxy** or **redirect** a request
  - statically configured
  - dynamically determined (CPL).
- Redirection
  - a user moves or changes her provider (PSTN: “The number you have dialed is not available.”)
  - caller does not need to try the original server next time. Stateless.
- Proxy useful if
  - forking, AAA, firewall control needed
  - proxying grants more control to the server

SIP Operation in Redirect Mode



Advanced Networking

Skype

Renato Lo Cigno
Renato.LoCigno@disi.unitn.it

Credits for part of the original material to Saverio Niccolini NEC Heidelberg

Skype characteristics

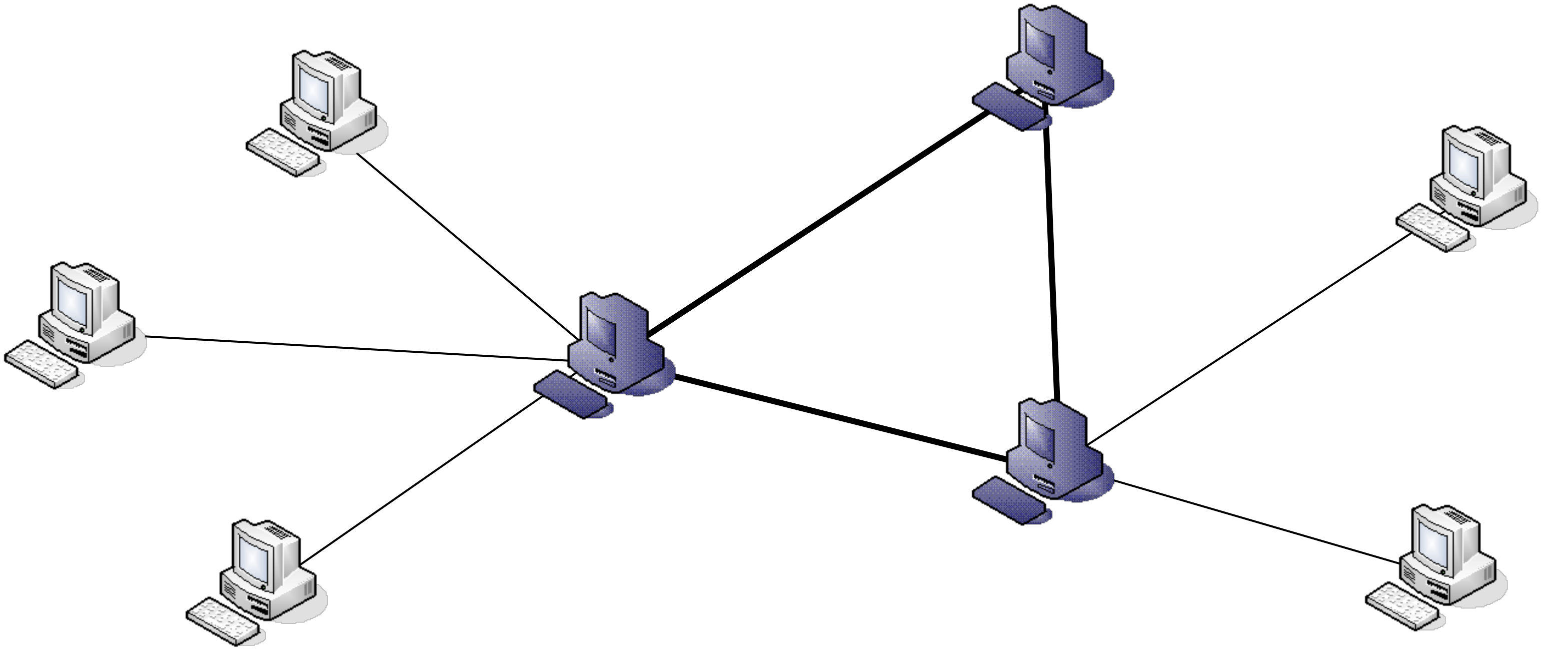
- **Skype is a well known P2P program for real time communications**
  - **Voice calls**
  - **Video (from version 2.0)**
  - **File sharing and instant messaging when in a call**
- **Seems to work with no problems in all network conditions compared to similar P2P applications**
- **One of the reasons of its success is its ability to work in network scenarios with middleboxes**
  - **such as firewalls and Network Address Translators (NATs)**
  - **usually, this is a problem for P2P applications**

How Skype works

- Skype overlay network
  - network structure
  - entities involved
- Skype function analysis
- Lesson learned
- Skype security analysis
  - Binary
  - Network protocol
  - Skype authentication
  - Traffic encryption

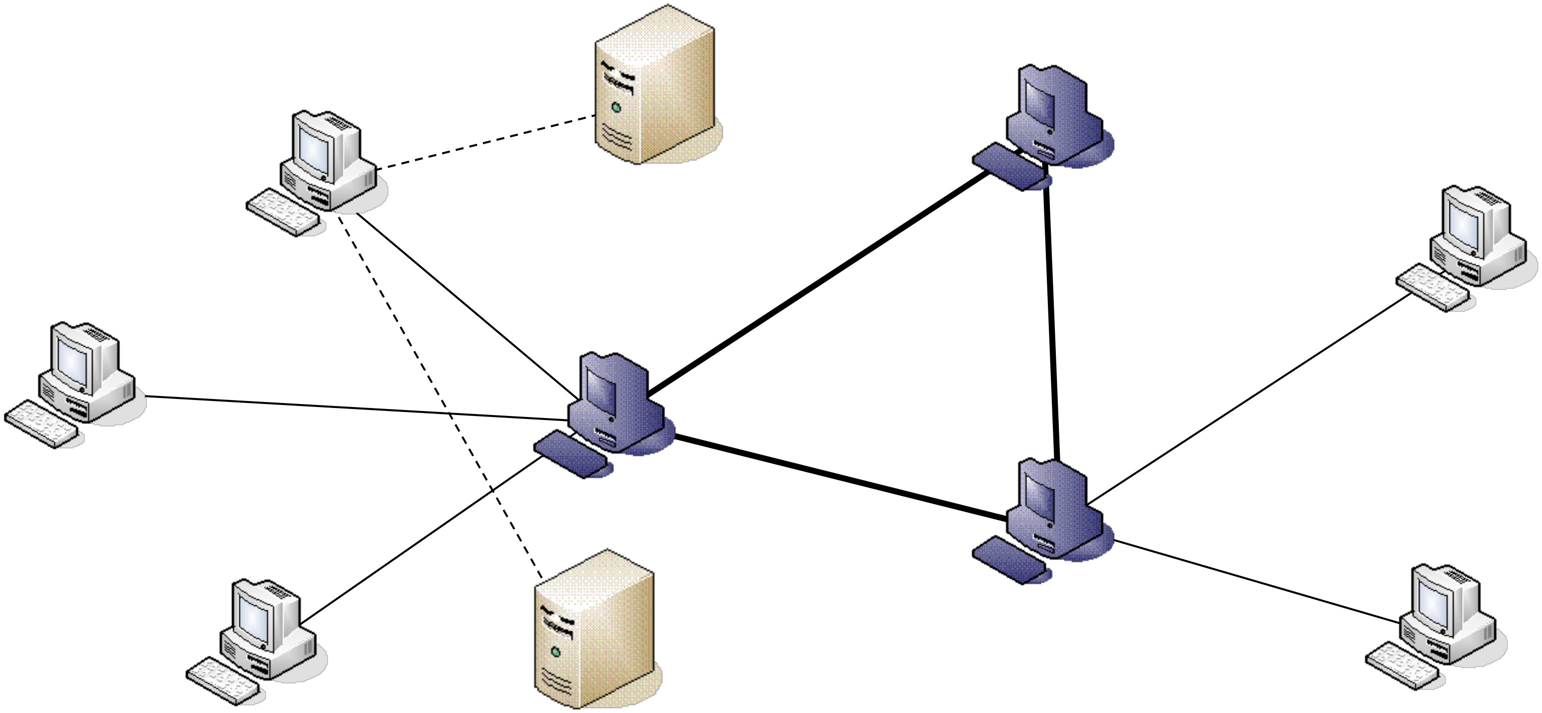
Skype overlay network (I)

- **Skype network relies on distributed nodes:**
  - **Skype Clients (SCs)**
  - **Supernodes (SNs)**



Skype overlay network (II)

- **Although there are also centralized entities:**
  - **HTTP Server**
  - **Login Server**



Skype overlay network (III)

Skype Client

- used to place voice calls and send instant messages
- connection to skype network possible through a supernode (SN)
- connection with the SN (via TCP) maintained for the whole time the client is on-line
- client configuration and SN addresses are stored locally and refreshed periodically to maintain a coherent view of Skype network

Skype overlay network (IV)

Supernode

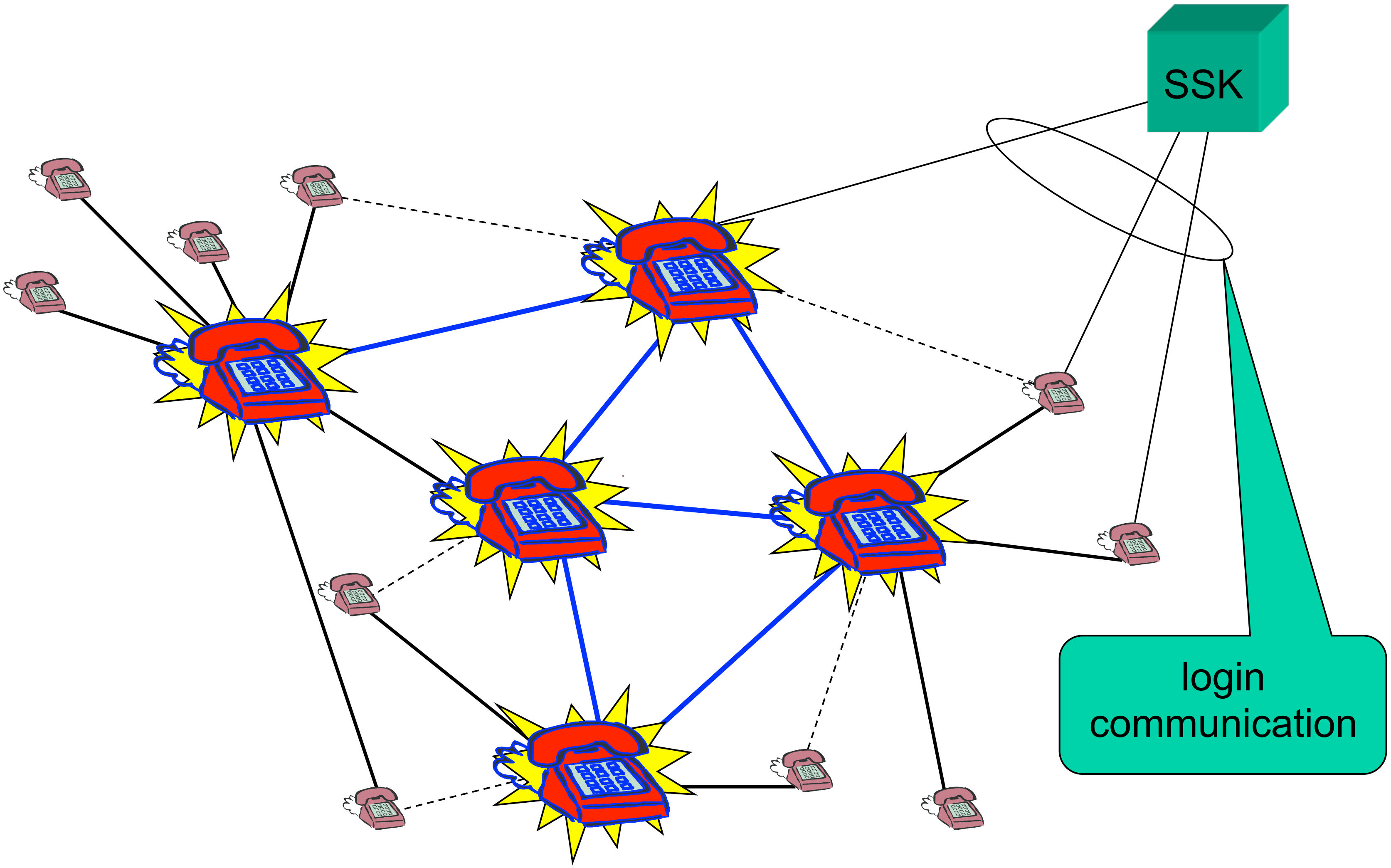
- Normal Skype Client that can accepts incoming TCP connections, with enough CPU, memory and bandwidth
- There are also a number of “default” Supernodes, used to increase network robustness and stability

Skype overlay network (V)

Servers

- Login server ensures that names are unique across Skype namespace. Also central point for authentication
- HTTP Server used by clients to check for updates

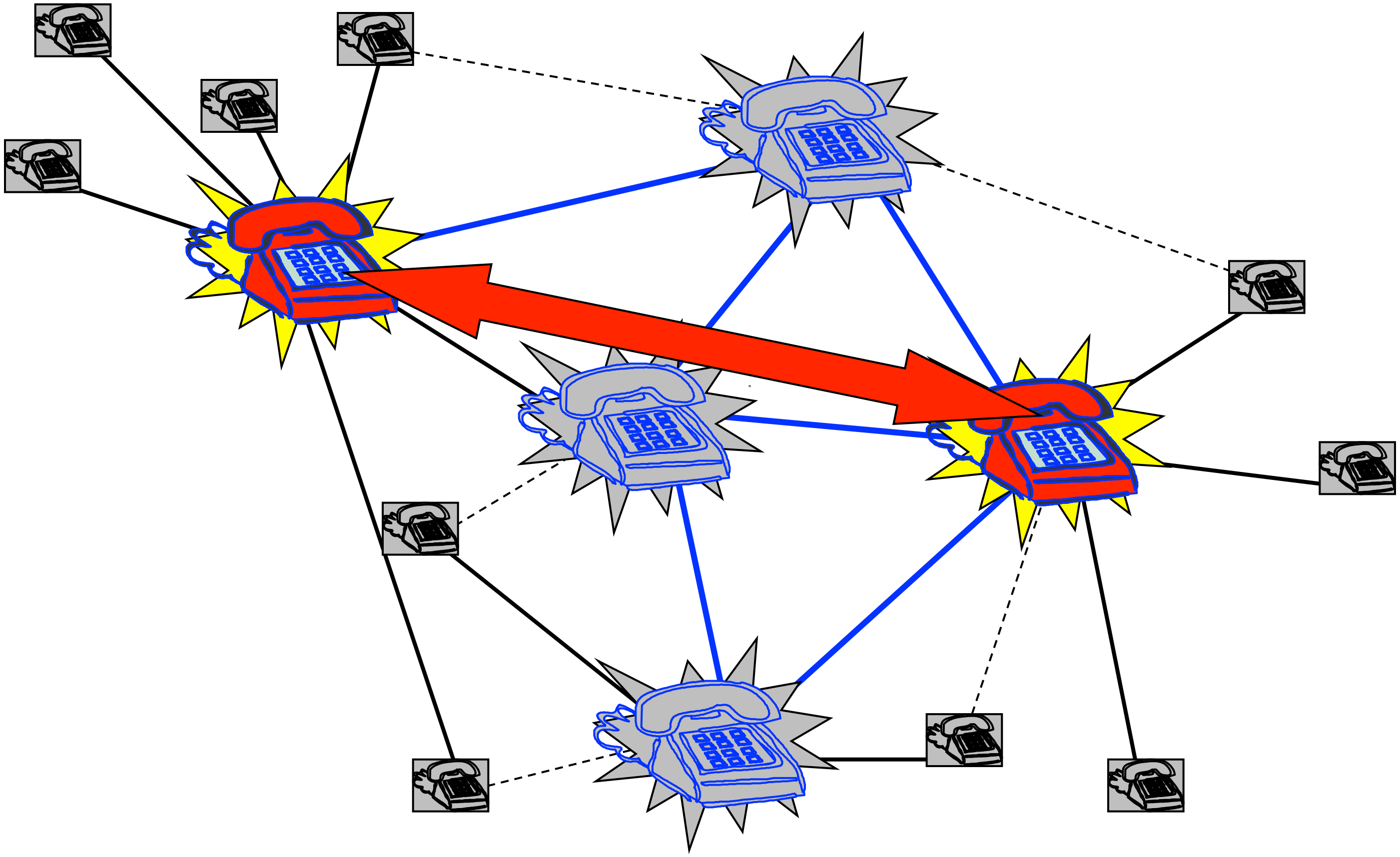
Topology



—— default connection

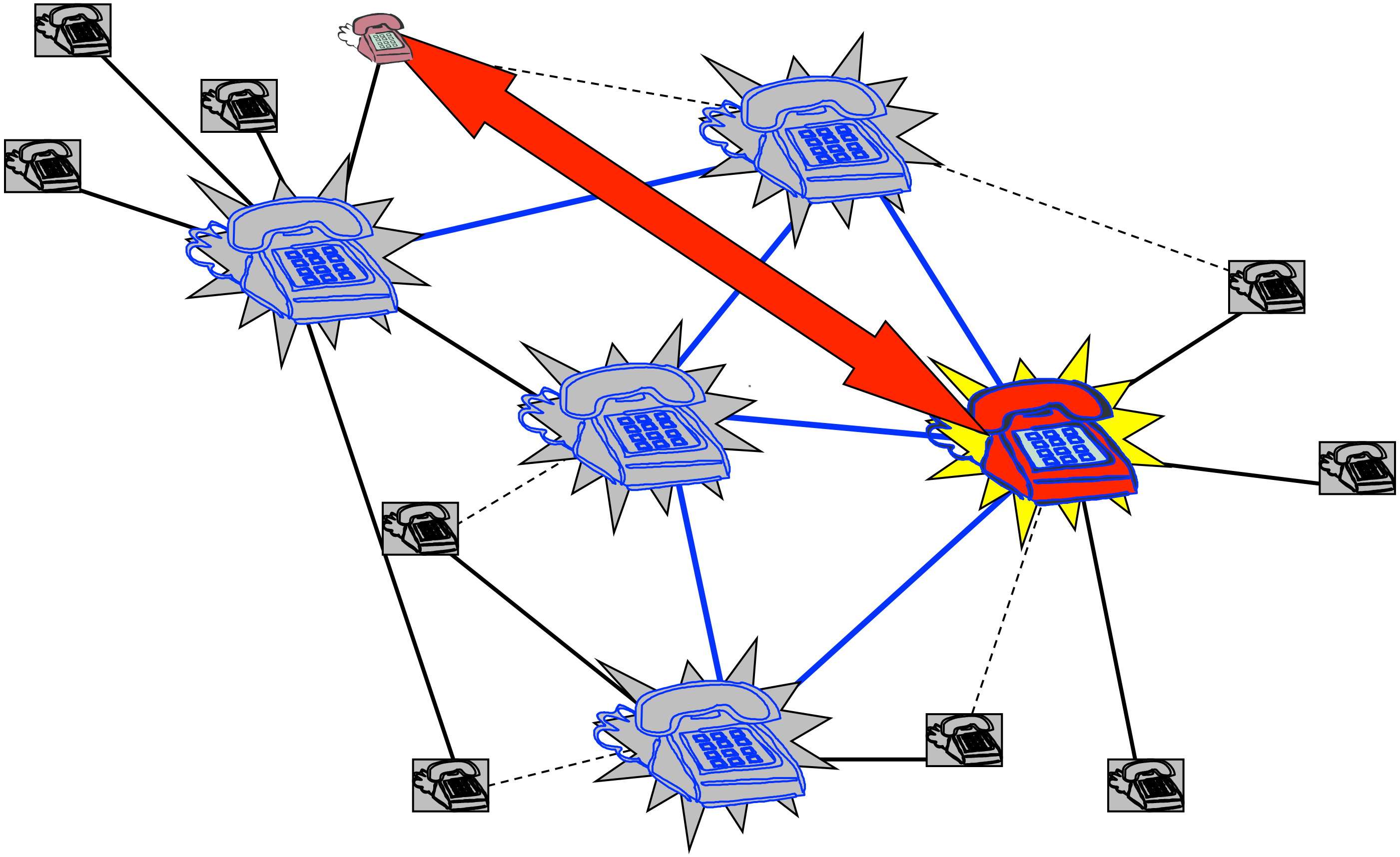
Topology: calls

Supernodes communicate directly ...



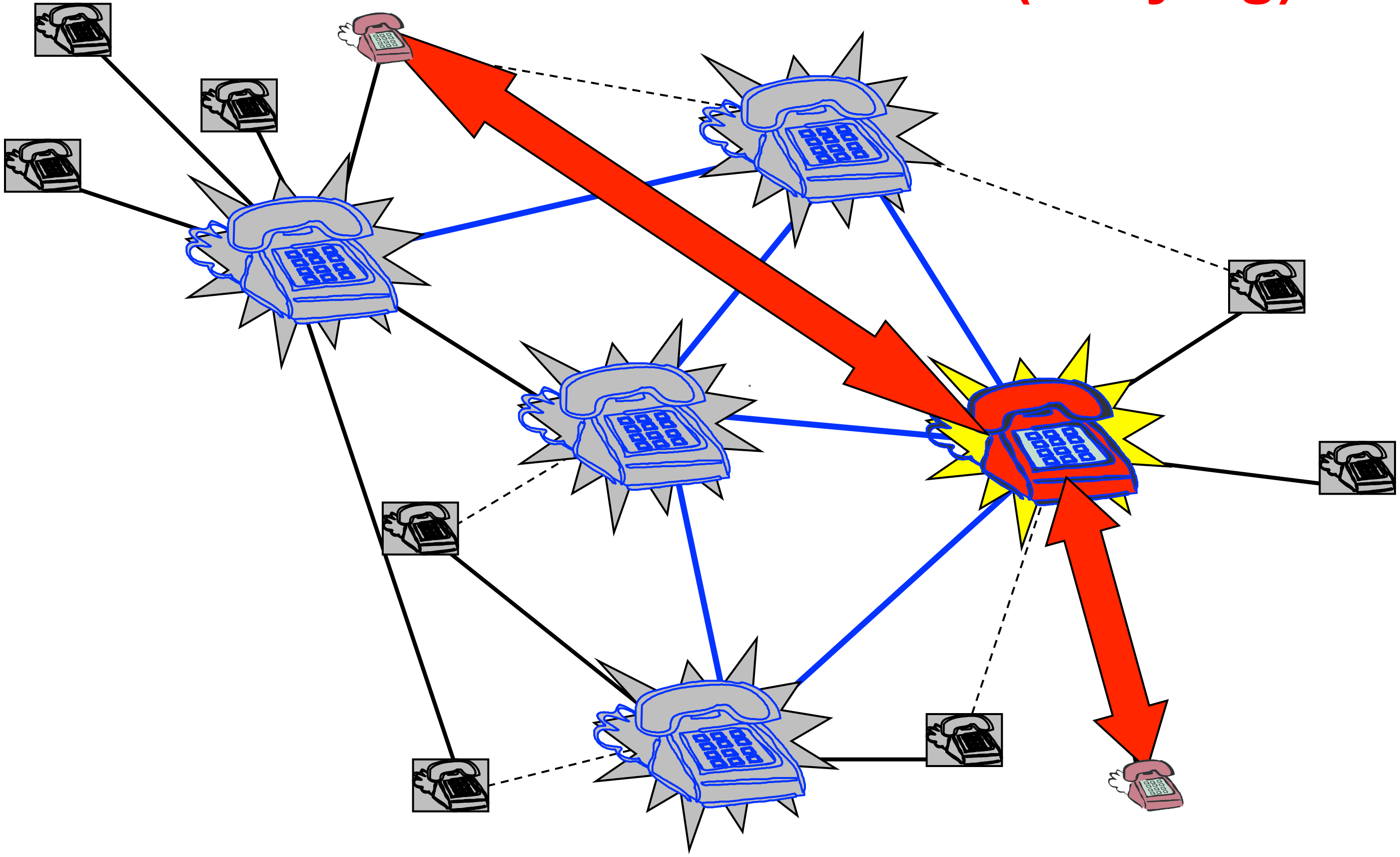
Topology: calls

... also with normal peers



Topology: calls

normal nodes require a supernode intermediation (relaying)



Some caratteristics

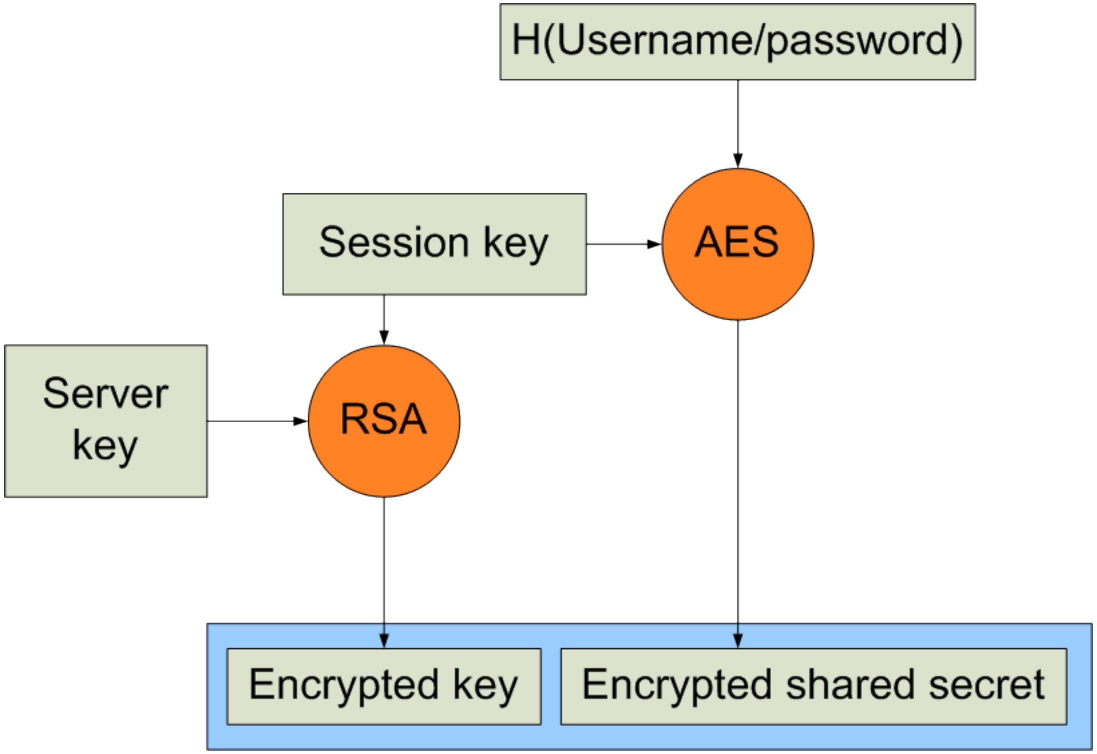
- CODECs
  - Default is a wideband (8 kHz-16kHz sampling) resulting in a transmission rate of 40 kbit/s in each direction (140 pck/s with payload of 67 bytes)
  - Quality in normal conditions is very good, much better than PCM telephony
  - No narrowband coding is provided, congestion is not considered a problem generated by skype
  - Under lab conditions over UDP the system works well even with only 16--20 kbit/s; below 12 kbit/s the system cannot work

Some Characteristics

- Ports
  - 80 (HTTP) e 443 (HTTPS) on TCP for signalin, random choice on UDP or TCP for voice
  - Ports are announced on the P2P network
- Encryption
  - All communications are AES (Advanced Encryption Standard) encoded

Skype Encryption

- **Authentication**
  - **At login time the client generate a RSA session key and uses it to encrypt his credentials.**
  - **Then encrypts the session key using the server's public key**
  - **and sends this information to the login server**



Some Characteristics

- Host Cache
  - List of supernodes (IP, Port) used to make the search phase faster
  - Roughly 200 entries dynamically updated
  - If the host cache is void skype does not work (some defaults entry are there from the beginning)
  - Une of the critical points for skype functioning
  - The idea is not new to P2P networks and answer to the bootstrap problem ... albeit in a naive way

Skype functions analysis

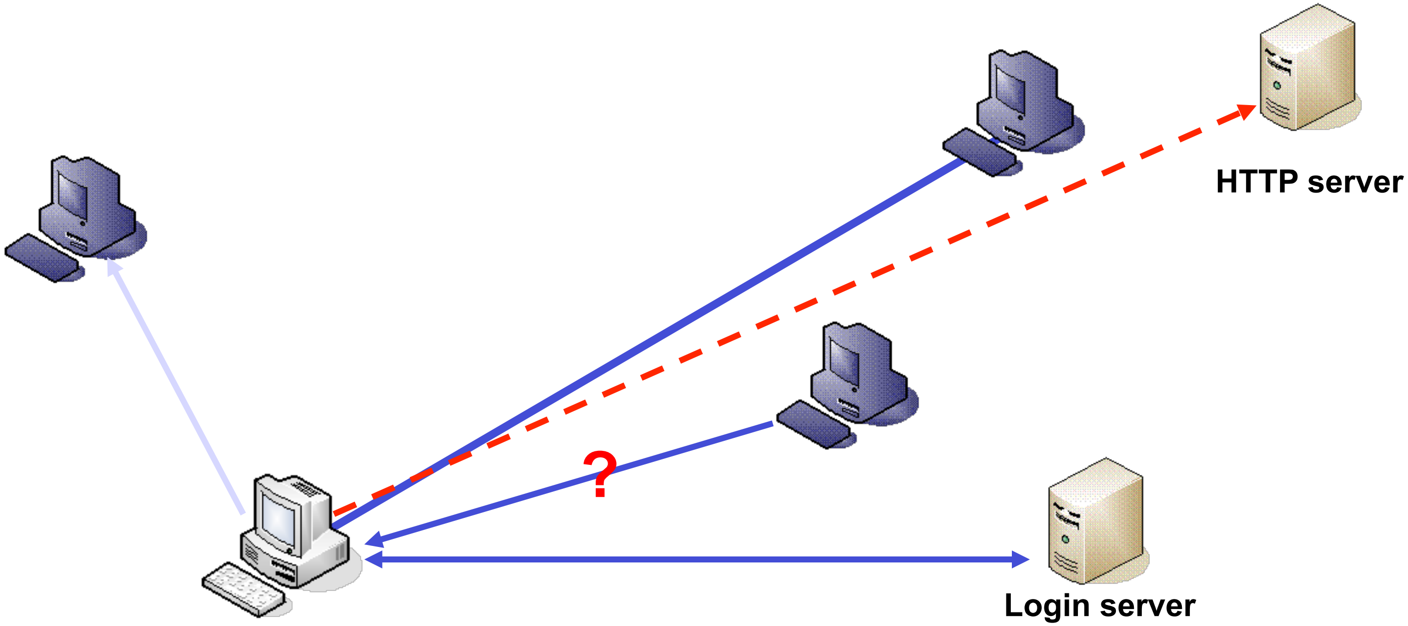
- **Essentials**
  - Login
  - Search
  - Buddy list signaling
  - Call establishment

Login function

- **Join and maintain overlay network:**
  - **Interaction with central servers**
    - **login server manage authentication and ensures unique names**
    - **HTTP server ensures client software updates**
  - **Refresh of shared.xml**
    - **file stored on the client containing SNs list and parameters identifying middlebox**
  - **Network tests if joining client can act as a SN**

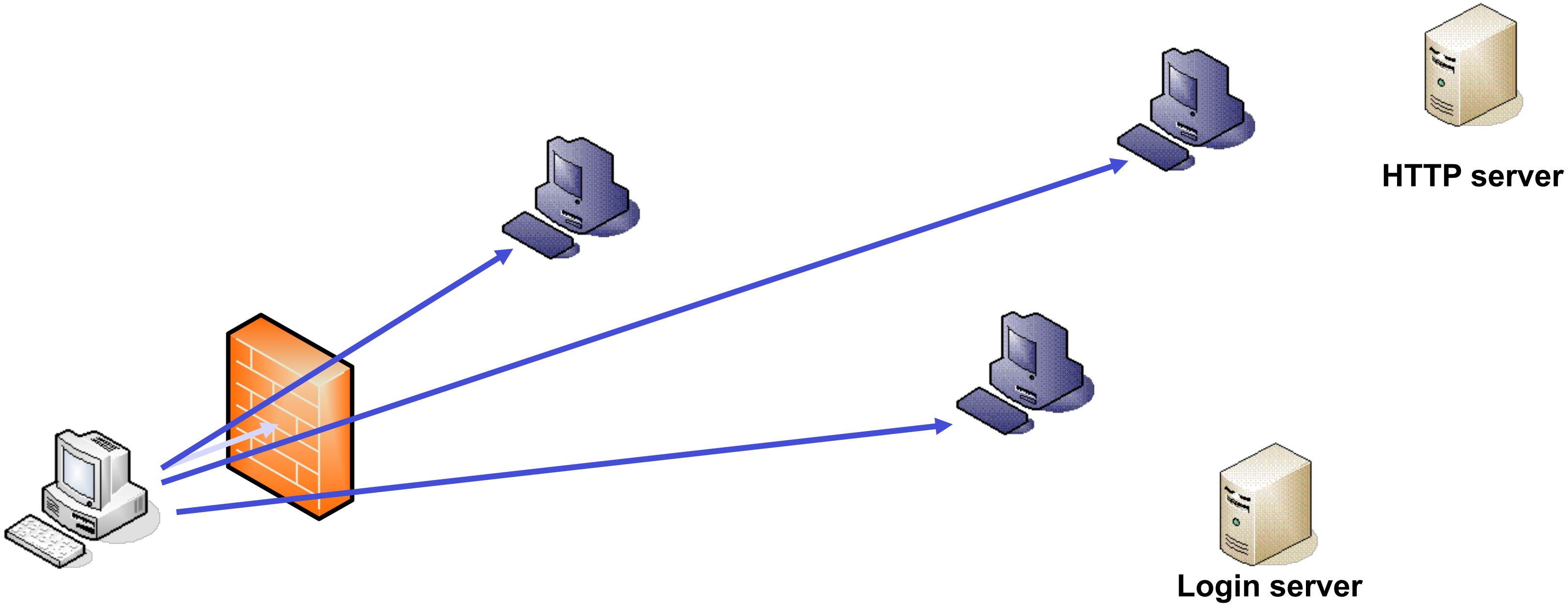
Login procedure

- At startup the client contacts the HTTP server to check for updates
- Sends UDP datagram to a -default SN- to refresh the list of supernodes
- Connects via TCP to a SN (connection maintained throughout Skype session) and exchanges info on online nodes
- Verify username and password via TCP with the Login server
- Another SN tests if client can act as a SN



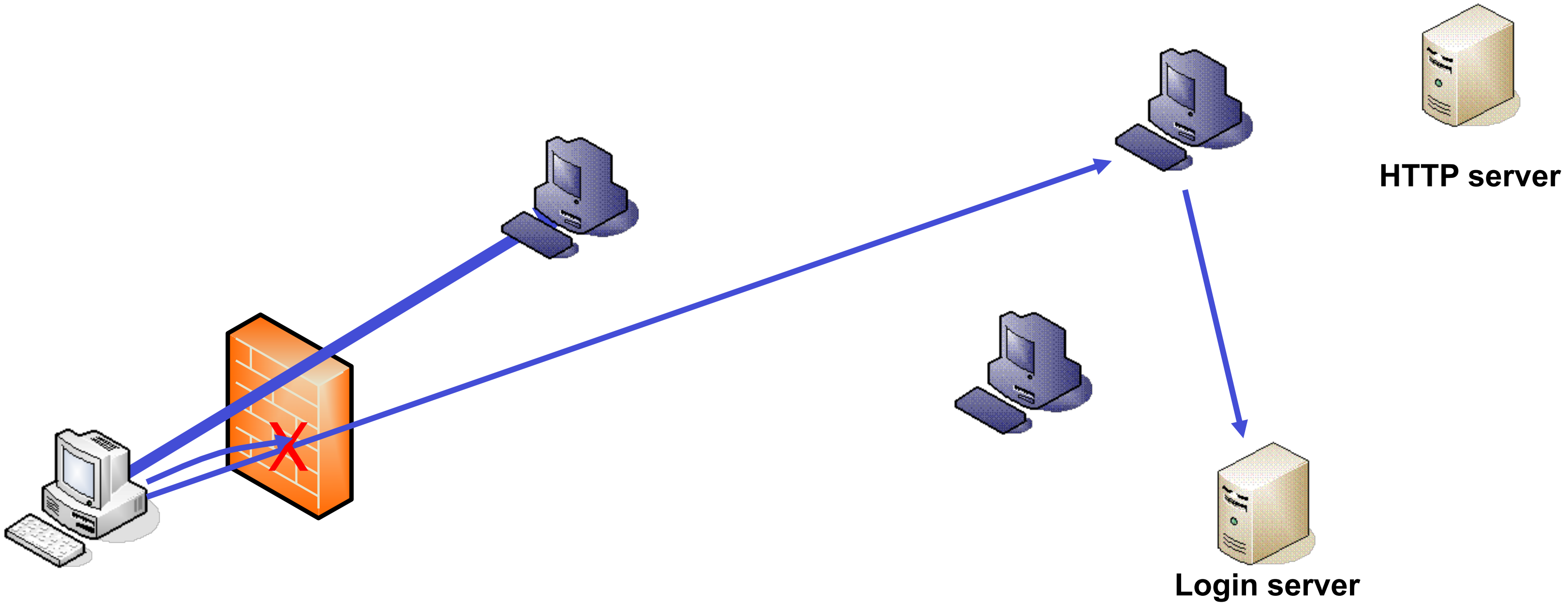
Login: Firewall blocks UDP

- Firewall prevents UDP exchange for SN list refreshing
- Client establishes several TCP connections with SNs to gather information, when finished all but one are torn down



Login: Firewall blocks Login sever

- After connection with the SN, attempt to connect with the Login server fails
- Client connect to the Login using a SN as a relay



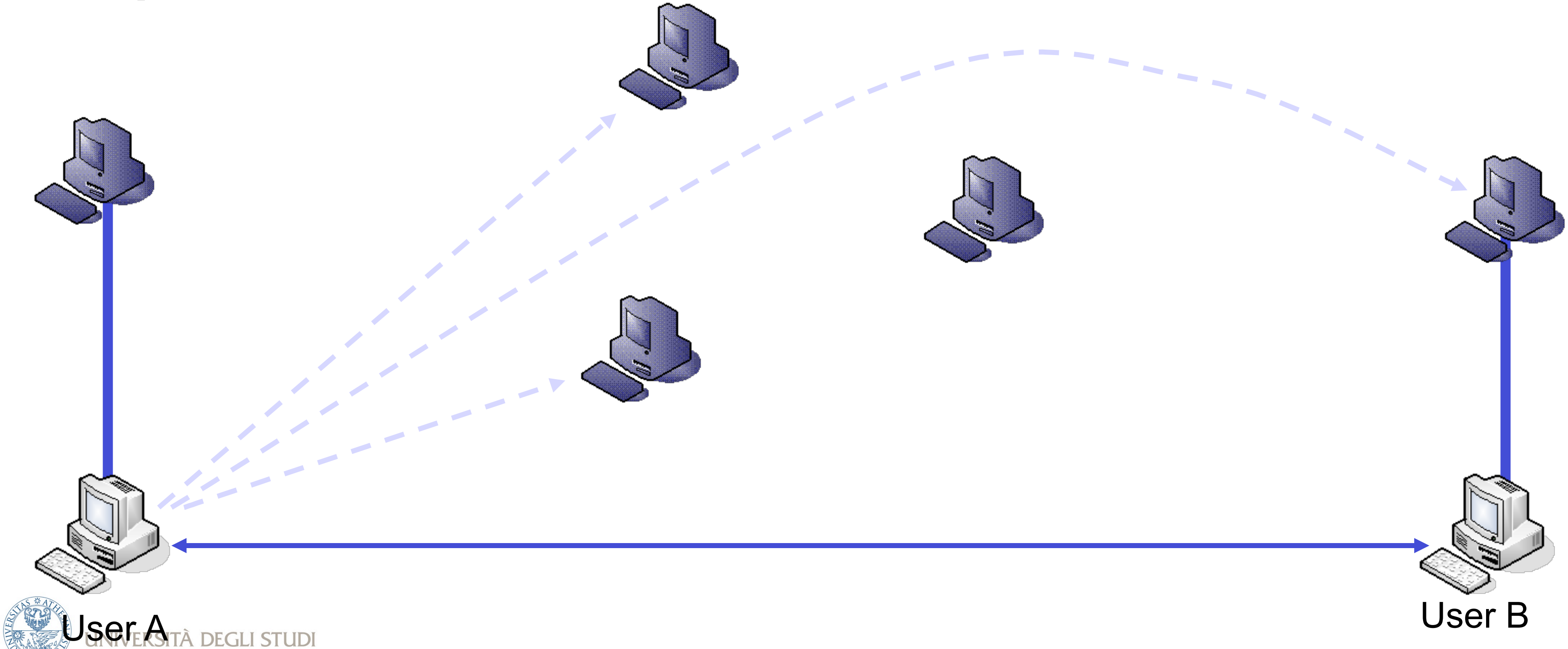
Search function

- Procedure performed when a user wants to add someone to his buddy’s list and communicate for the first time

- Search is performed using username as key
  - possible since names are unique
  - this is why there is the need for central servers

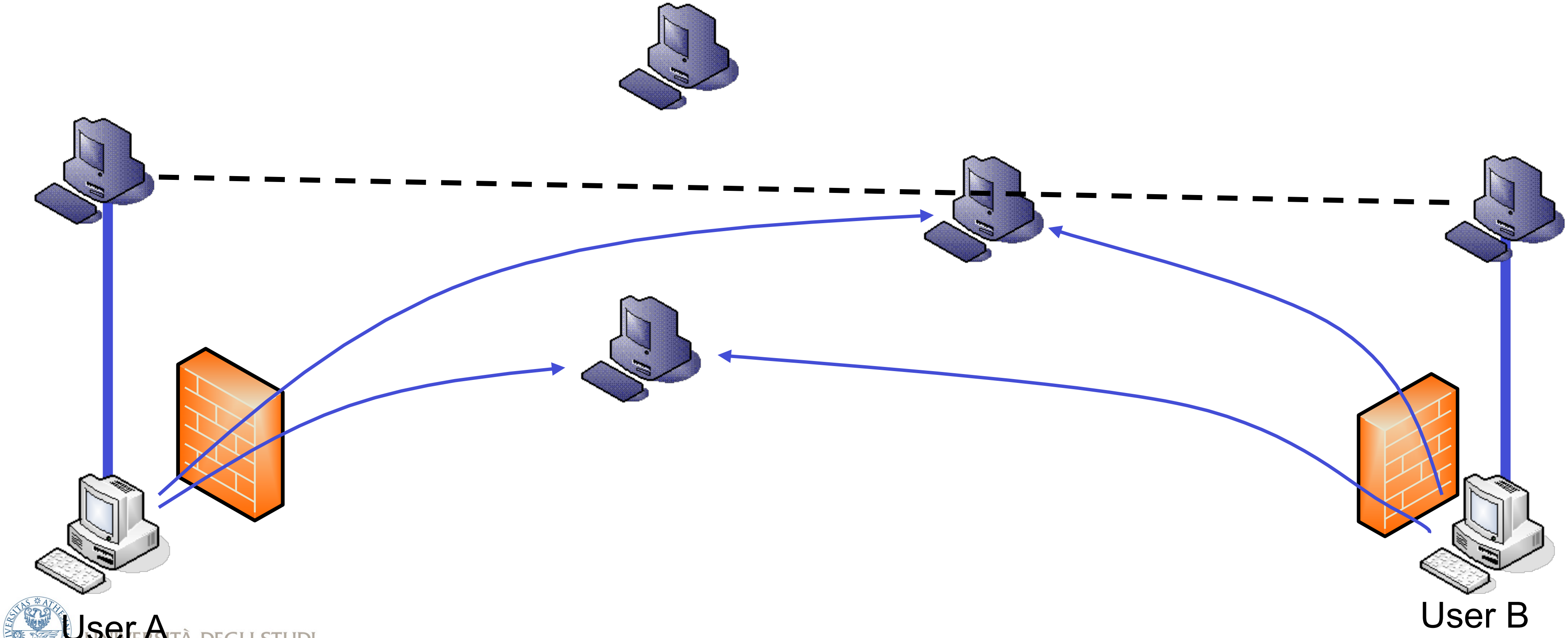
Search procedure

- User A exchanges info with its SN and gather 3 SNs addresses
- A query the 3 SNs via UDP asking if they know the public IP of B
- Once A gets the address of B authorization exchange is performed



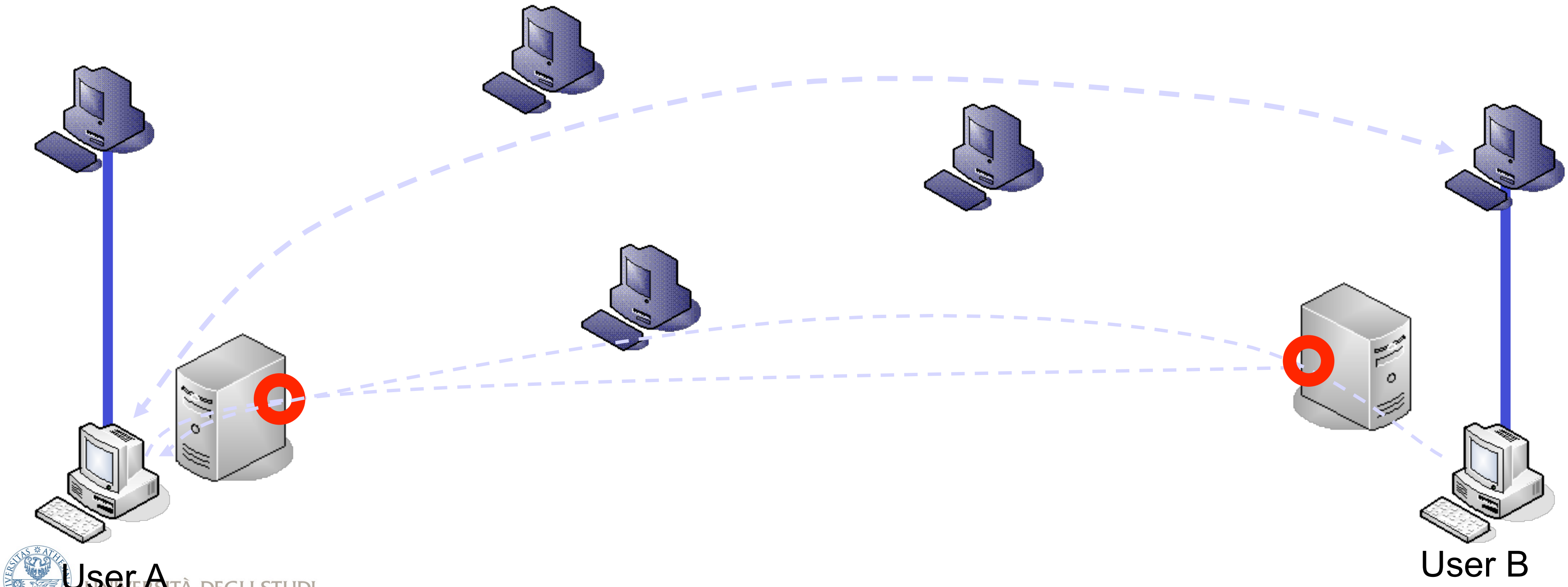
Search: Firewall blocks UDP

- Firewall blocks UDP
  - preventing direct connection w/ the SNs or another user
  - the SN of A communicate to B (via his SN) the address of A
- Both A and B establish TCP connections with the same 2 SN to exchange authorization



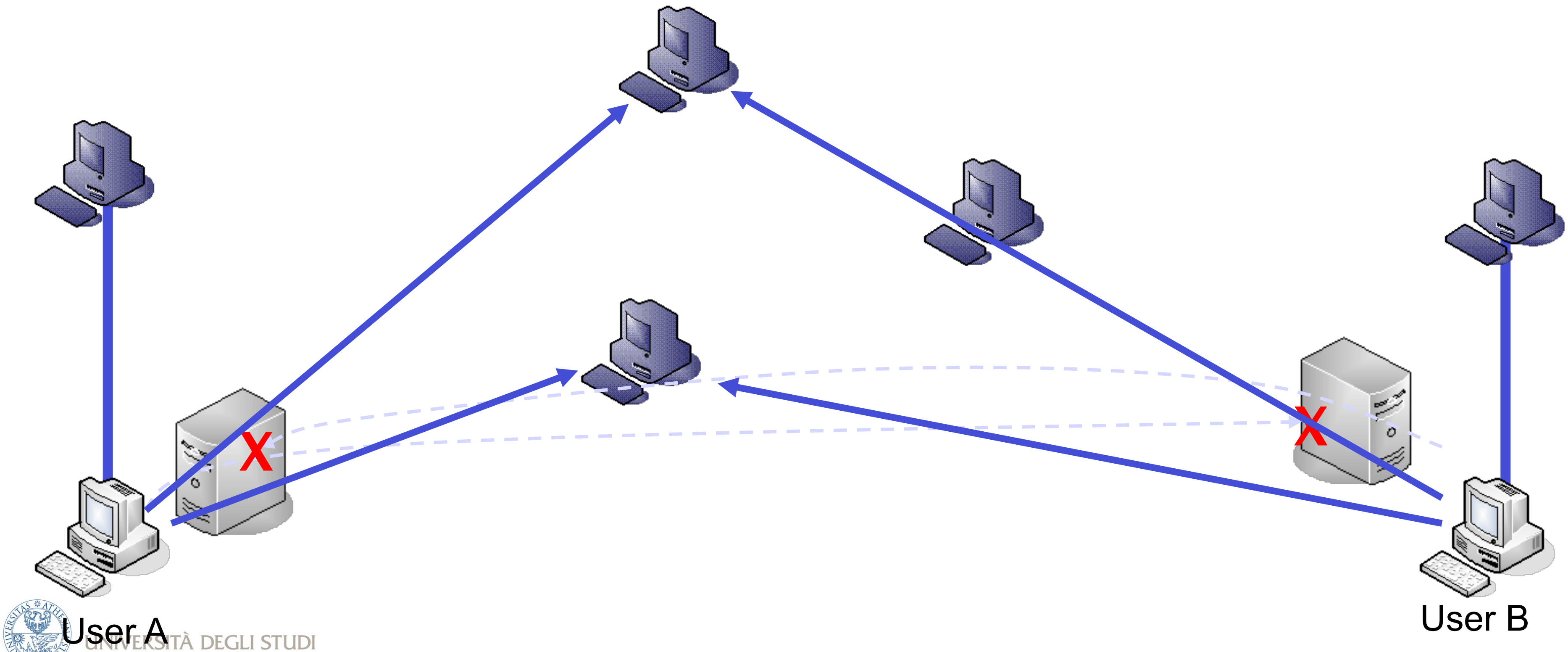
Search: Port restricted NAT

- Once user A gather the address of SN of B, sends a UDP query containing his external address. SN of B replies with user B external address.
- User A send an UDP datagram to user B external address in order to create a mapping in his NAT, anyway packet will be filtered by NAT of B
- User B does the same but this datagram reaches user A
- Once exchanged authorization a TCP connection via 2 SNs as relay is established, as depicted in previous slide



Search: Symmetric NAT

- Clients try the technique depicted for Port restricted NAT
  - but it fails due to symmetric NAT behvior
- Clients exchange authorization via TCP using 2 SNs as relay

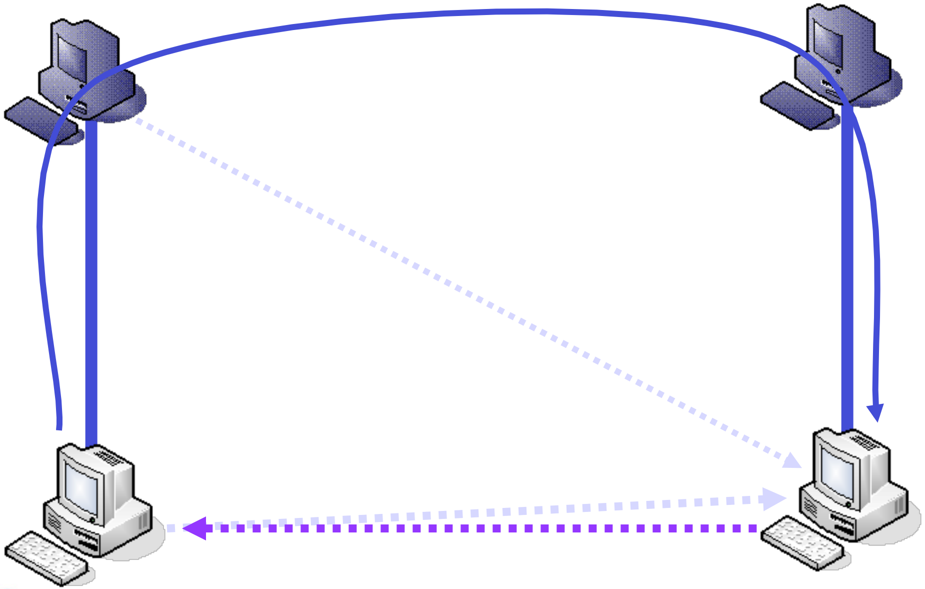


Buddy list signaling

- Buddy list is a list of “friend” users
- Skype allow a user to know if buddies are online/offline
  - overlay network informs buddies when user change status

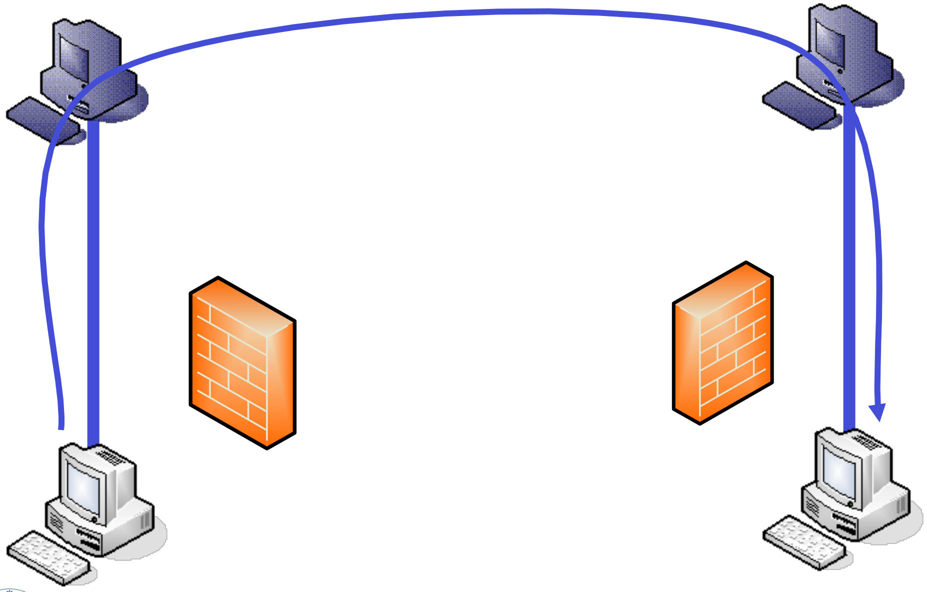
Buddy List signaling procedure

- A user going on-line informs his buddies either directly using UDP or via the SNs.
- When going off-line, a user tear down the TCP connection with the SN.
- The SN informs via UDP the buddies that the user is going off-line
- To have a confirmation buddies try to ping the user.



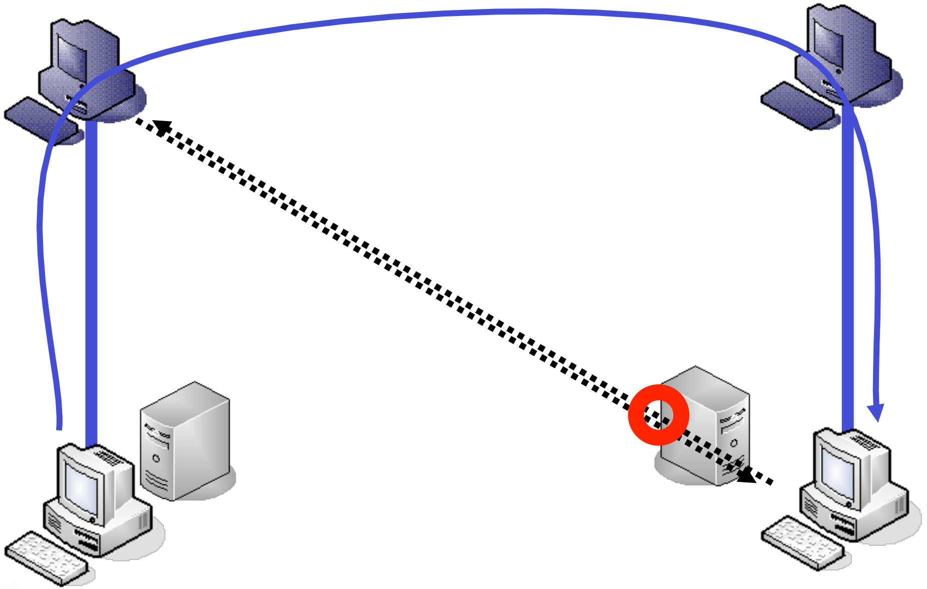
Buddy List signaling: Firewall blocking UDP

- **Since UDP traffic is blocked, on-line/off-line signalling is performed via the SNs**



Buddy List signaling: Port restricted NAT

- On-line/off-line signaling is performed in a way similar to that depicted in previous slide.
- As a difference after the change of status, buddies query the SN of the user for confirmation.

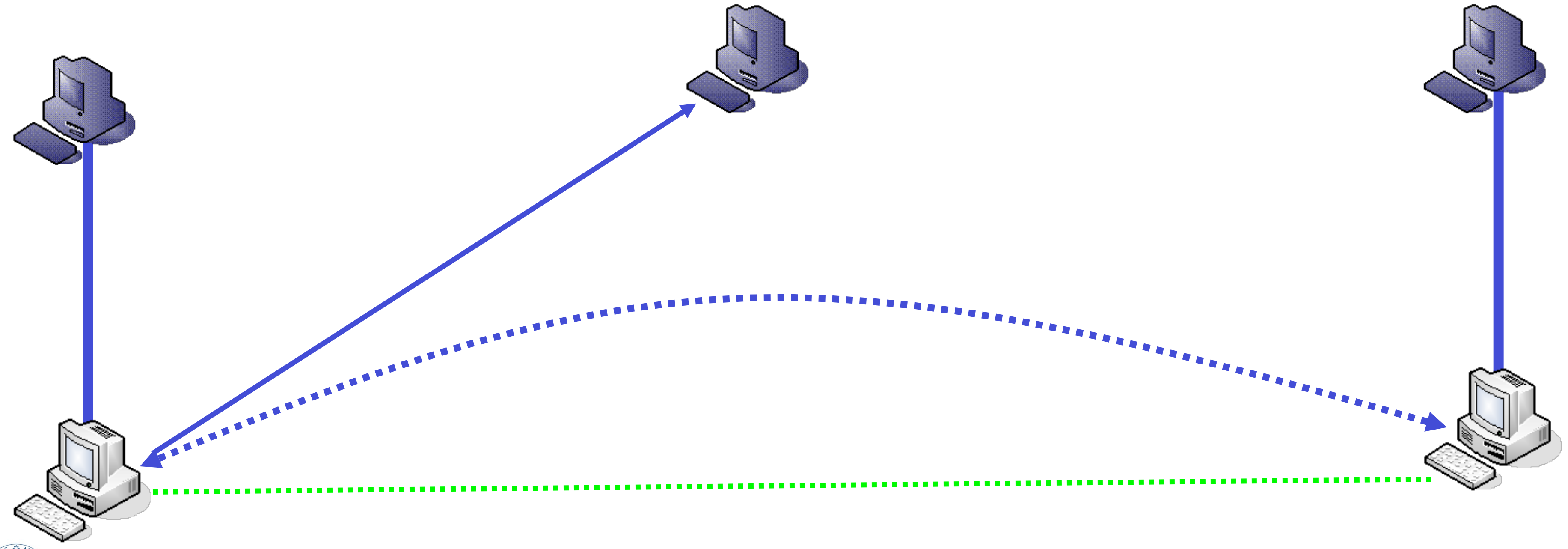


Call establishment function

- **Signaling performed using TCP connection**
  - **overlay network used only if otherwise impossible**
- **Media carried over UDP when possible**
  - **in case relay servers are used**

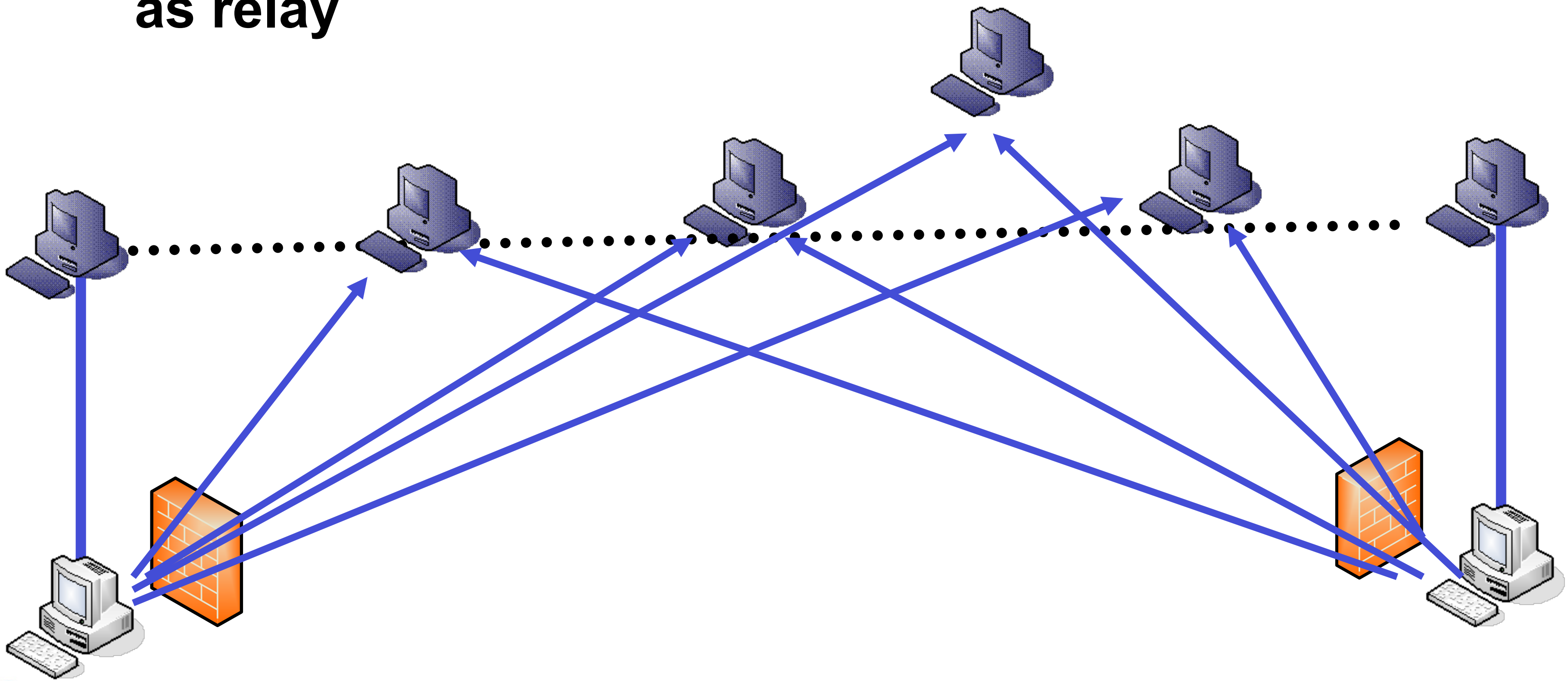
Call establishment procedure

- User A wants to call user B, so he query some SNs for user B address.
- Once he gets user B address they exchange signaling over TCP
- Voice traffic carried via UDP



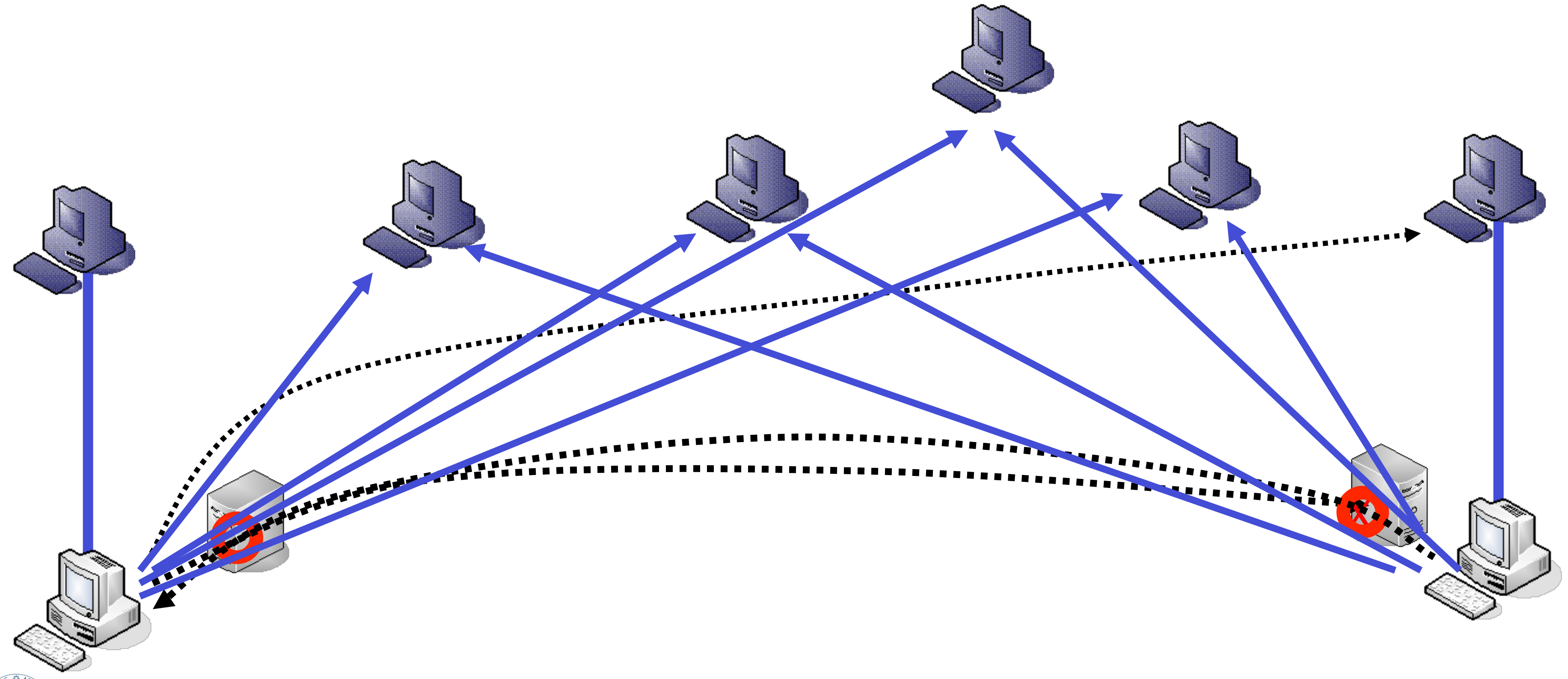
Call establishment: firewall blocks UDP

- Signaling exchanges are performed by the SNs on behalf of the users
- Media exchange is performed via TCP using 4 SNs as relay



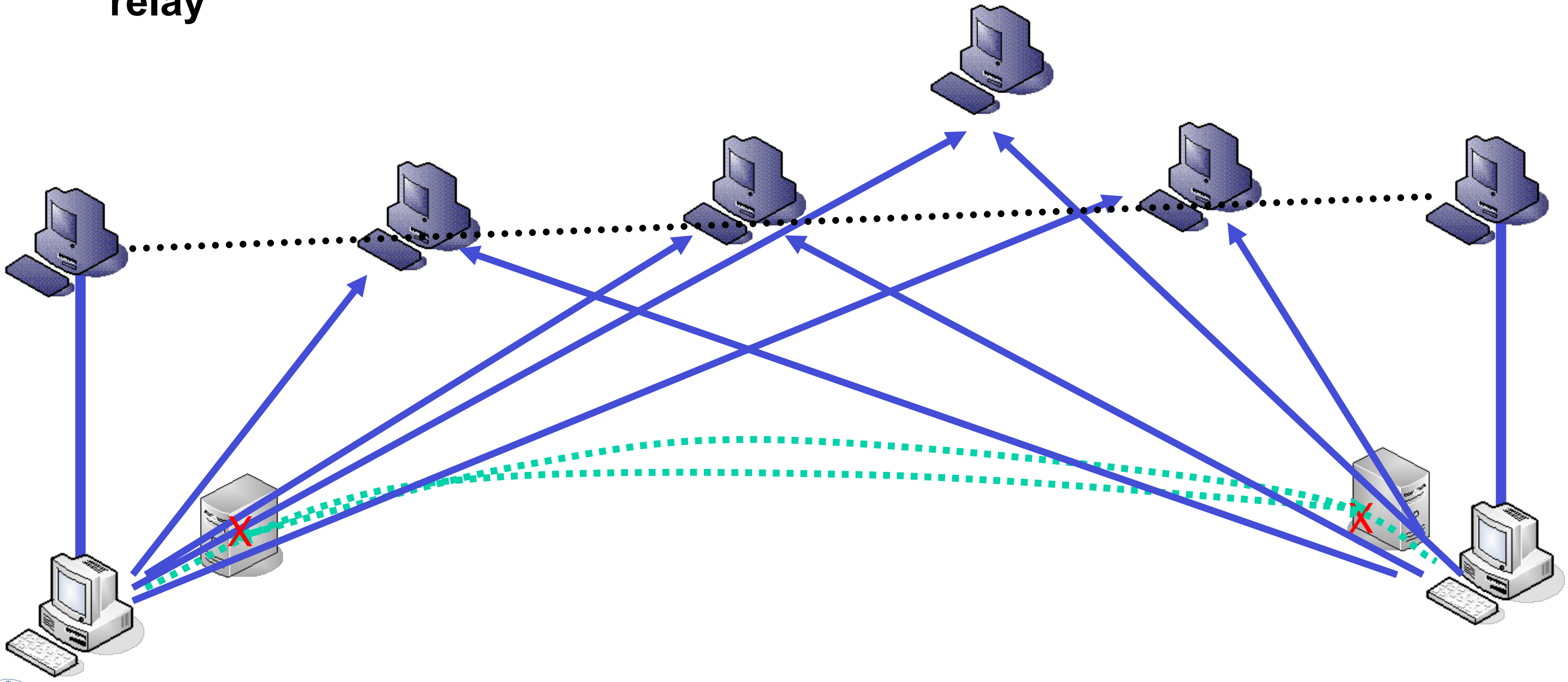
Call establishment: Port restricted NAT

- Once User A gets the address of the SN responsible for user B he queries for his address. SN informs B that user A wants to call him, and tells external address of B to A.
- A and B establish UDP flow using reverse hole punching
- They also establish TCP connection using 4 SNs as relay



Call establishment: Symmetric NAT

- User A and B communicate their addresses via their SNs
- They try reverse hole punching but it won’t work because of NATs restrictions
- To establish the media and signalling channel they will use 4 SNs as relay



Lesson learned

- **Traversal is well possible in many cases without explicit signaling to the middlebox**
  - **open public access network**
  - **protected enterprise networks**
- **Reverse hole punching and tunneling techniques workarounds allow Peer-to-peer communications in almost every scenario**
  - **Skype only fails completely if firewall blocks TCP but in fact that is a very uncommon case**
- **Explicit middlebox signaling protocols (like IETF MIDCOM MIB, CheckPoint OPSEC, NEC's SIMCO) are still required for**
  - **highly protected access network**
  - **applying security policies by network operator**
  - **anyway Skype will undermine many of these policies**
- **Skype tries to use IP network instead of overlay**
  - **SNs can't assure constant presence**
  - **avoid overlay congestion**

Advanced Networking

P2P Voice Applications beyond skype

Renato Lo Cigno
Renato.LoCigno@disi.unitn.it

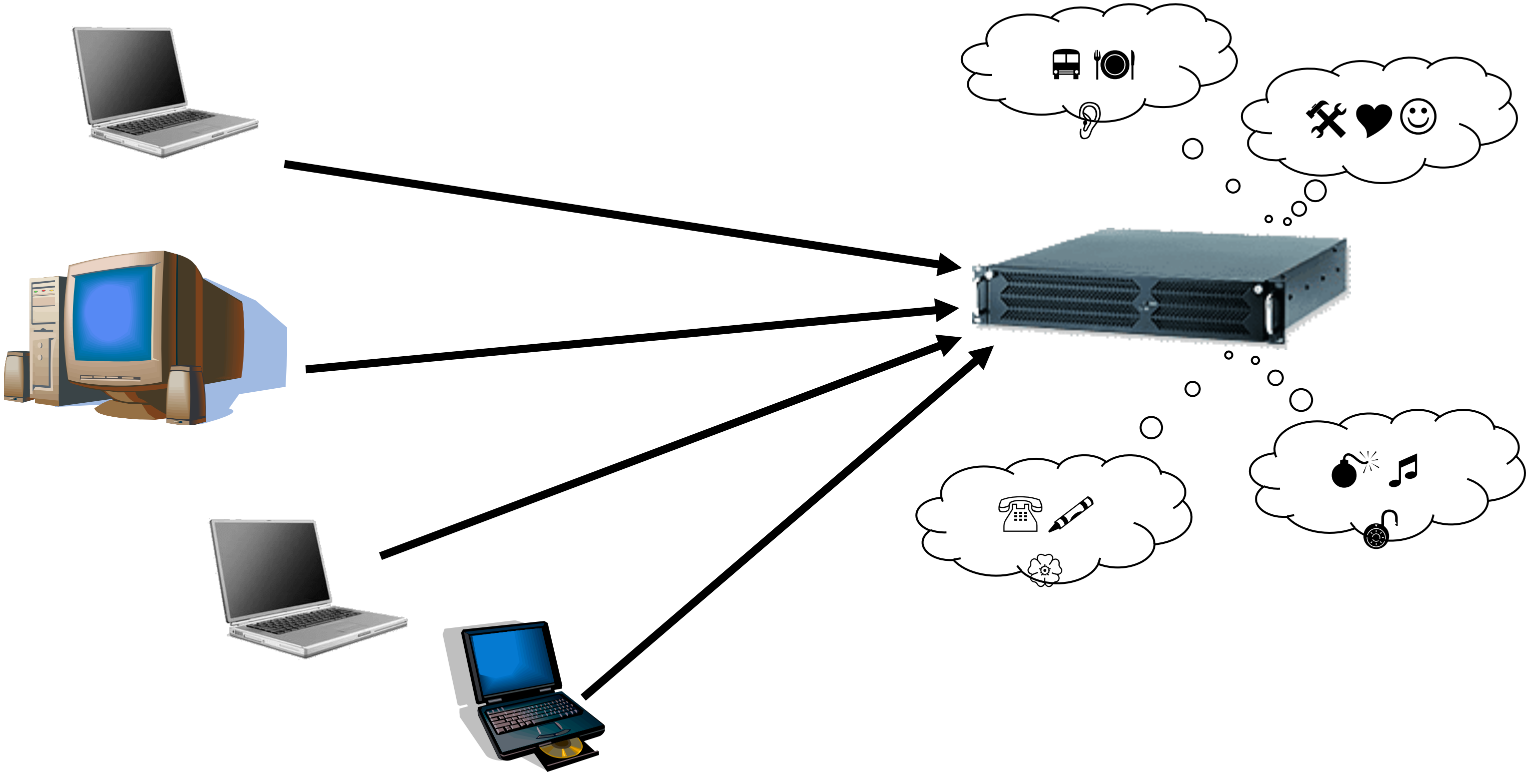
Credits for part of the original material to Saverio Niccolini NEC Heidelberg

What is Peer-to-Peer (not specific to VoIP)?

- **Peer-to-Peer (P2P) paradigm**
  - **Fundamentally different than client server**
  - **Nodes cooperate with each other**
    - **to provide (collectively) the functionality a central server would provide**
  - **Not all nodes provide all services/know everything, but as a group they do**

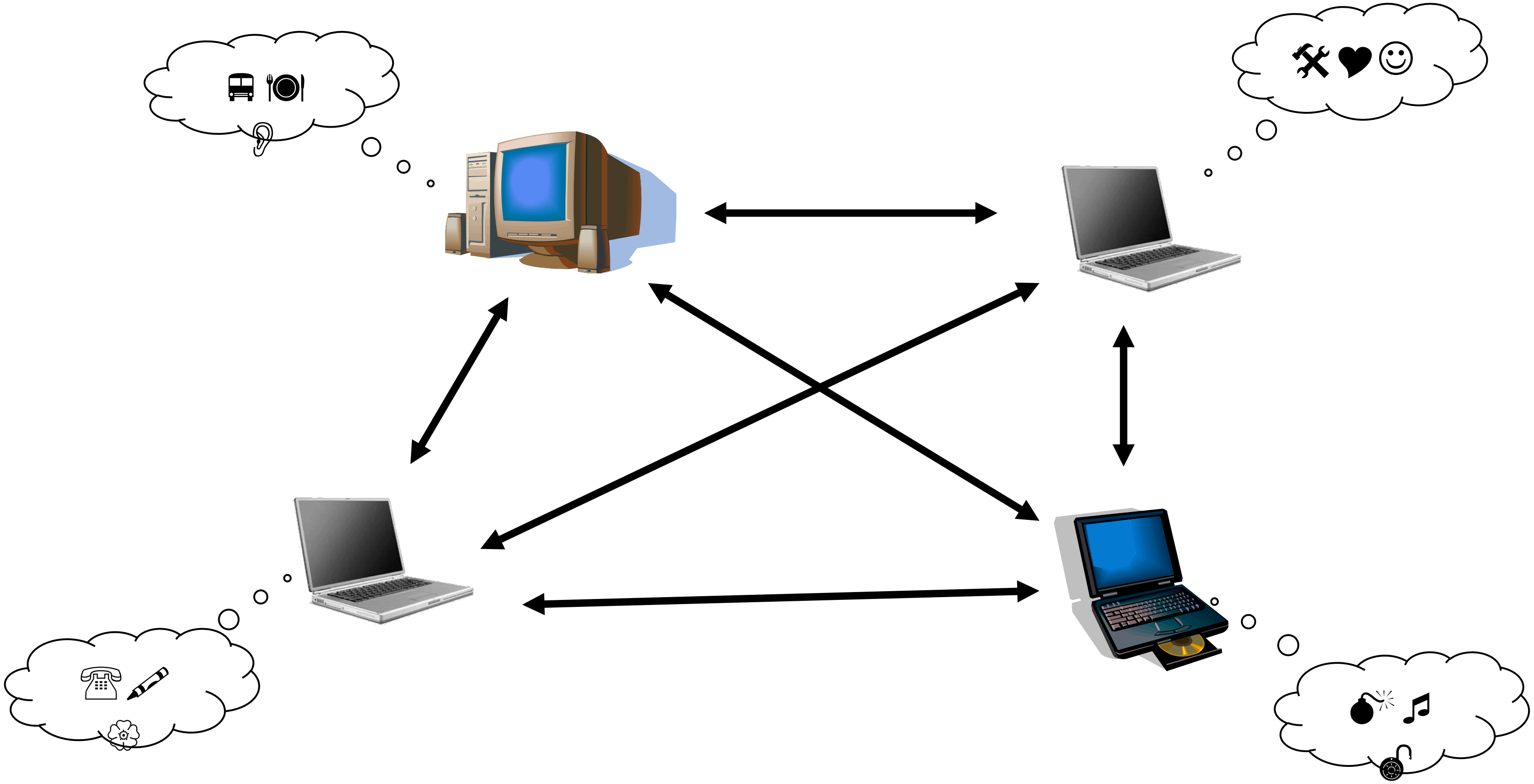
What is Peer-to-Peer (not specific to VoIP)?

Client-Server

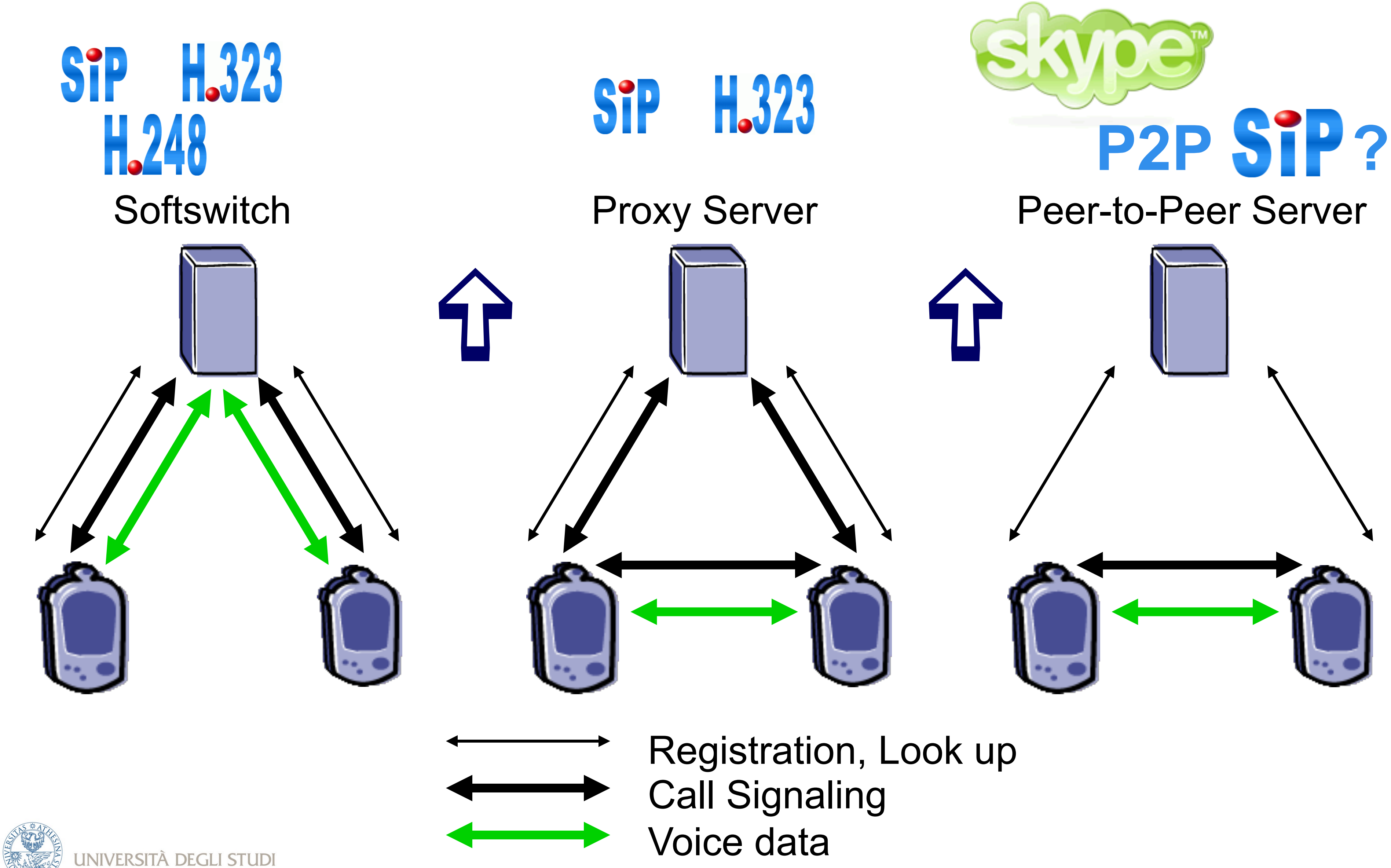


What is Peer-to-Peer (not specific to VoIP)?

Peer-to-Peer



Towards VoIP P2P: Evolution



Why P2P?

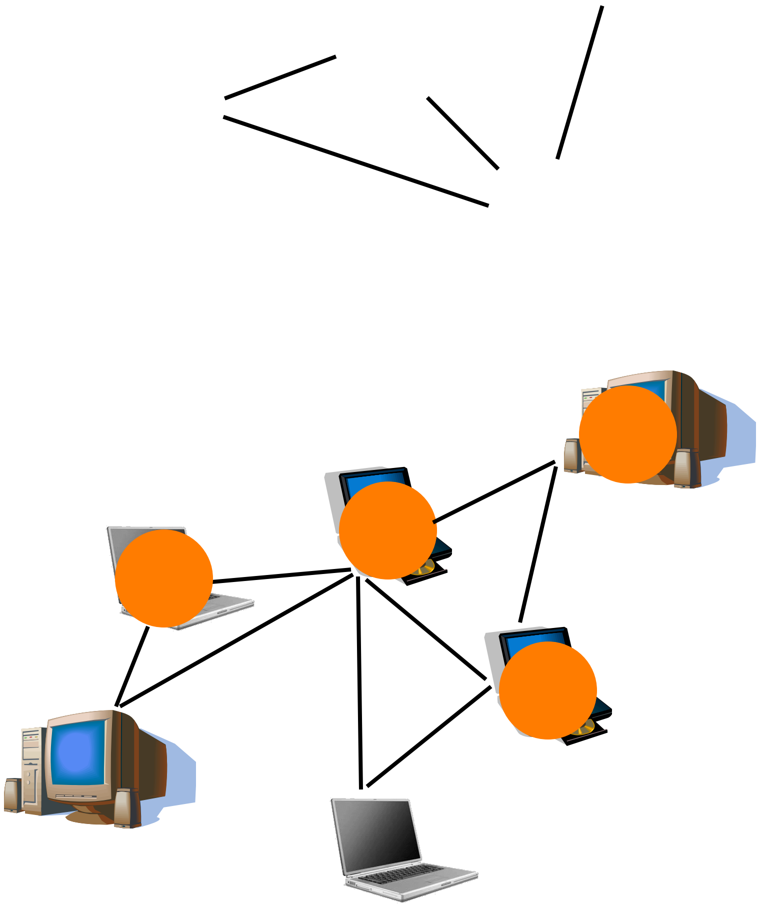
- Infrastructure independence
  - No central servers (up to a a certain limit)
  - Don’t need direct connectivity (up to a certain limit)
- Simple discovery and setup
- Privacy
- Highly scalable
- Lack of central control
- Dynamic DNS doesn’t offer all of this

P2P Basics

- Most ~~famous~~ infamous use of P2P is file sharing
  - Each user stores some number of files on the network, ask peers for the file
- Can also share other resources or services, no need to be files
- Connected to each other in a logical network called an overlay

Overlay Network

- **Collection of nodes, connected logically in some way**
- **The connections in the overlay are frequently not related to those in the physical network**

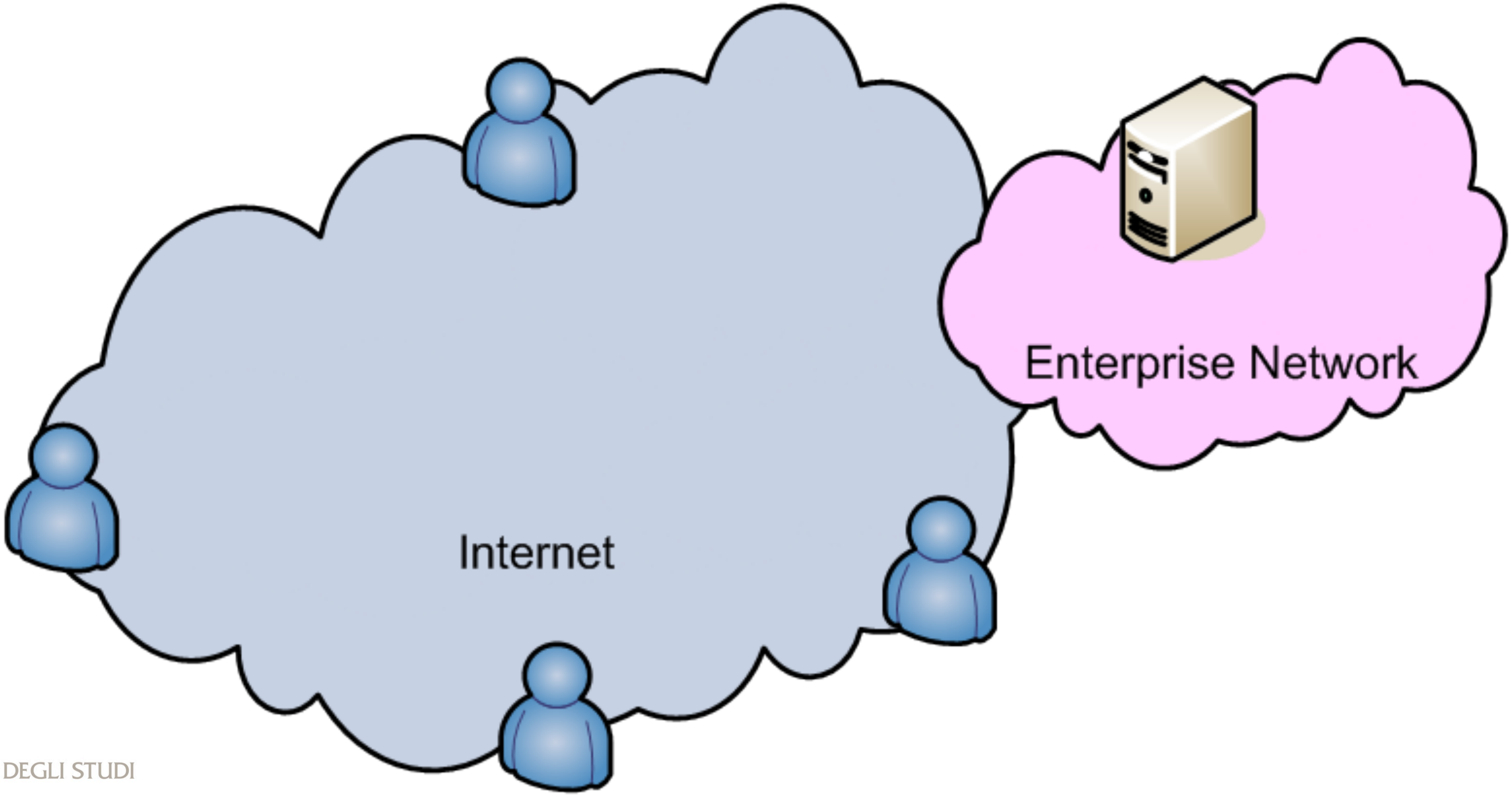


Motivating Cases

- Small deployments
    - Distributed remote office solutions
        - different from centralized VPN
    - Better enforcement of security
    - Lack of resources
- Limited/No Internet connectivity
- Ad-Hoc groups
- Censorship or impeded access
- Large scale decentralized communications
    - Skype (sort of)

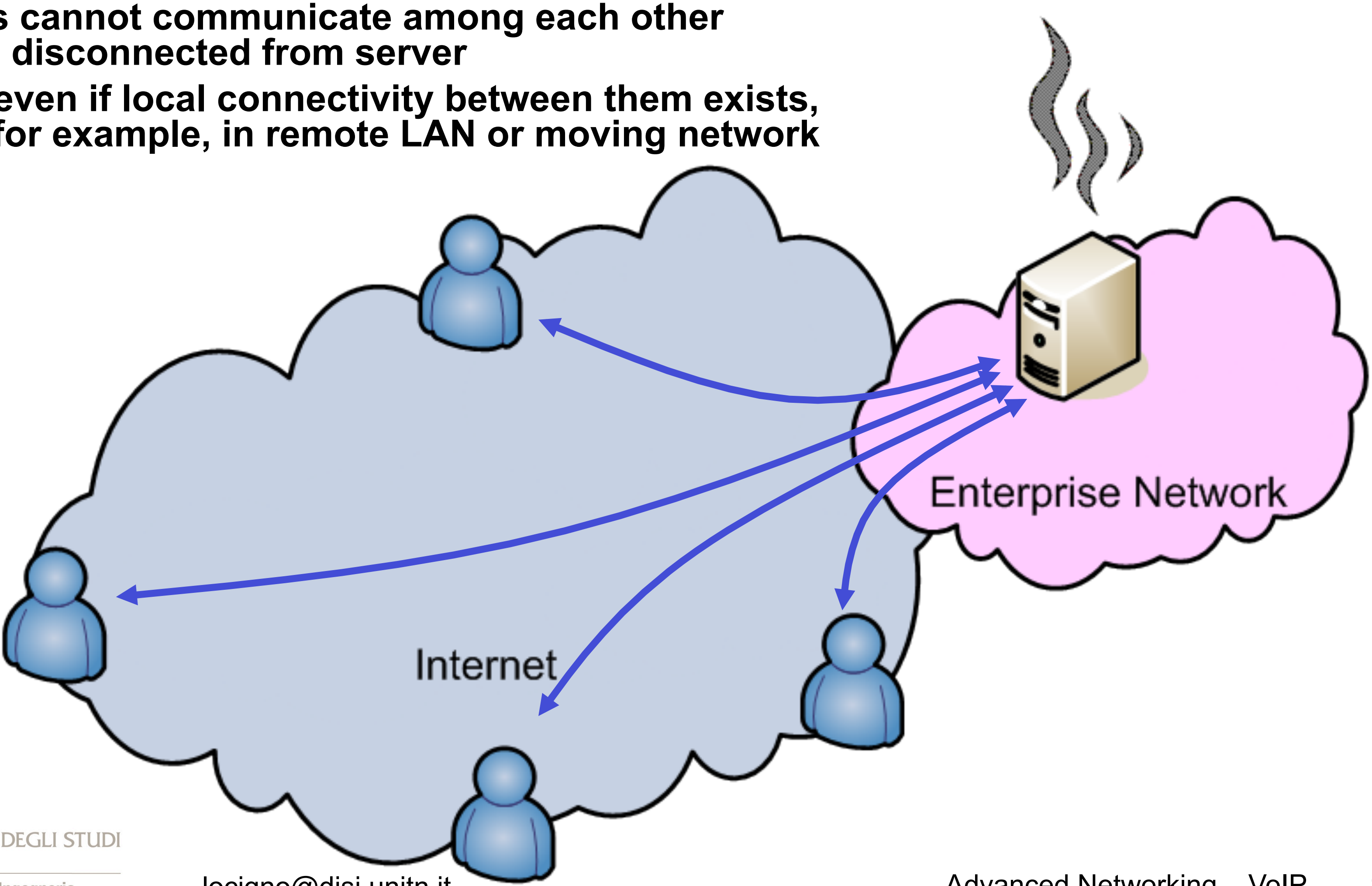
Distributed remote office solutions

- **Road warriors need virtual office network**
- **Collaborative network between employees**
- **Employees need access to company data as well**



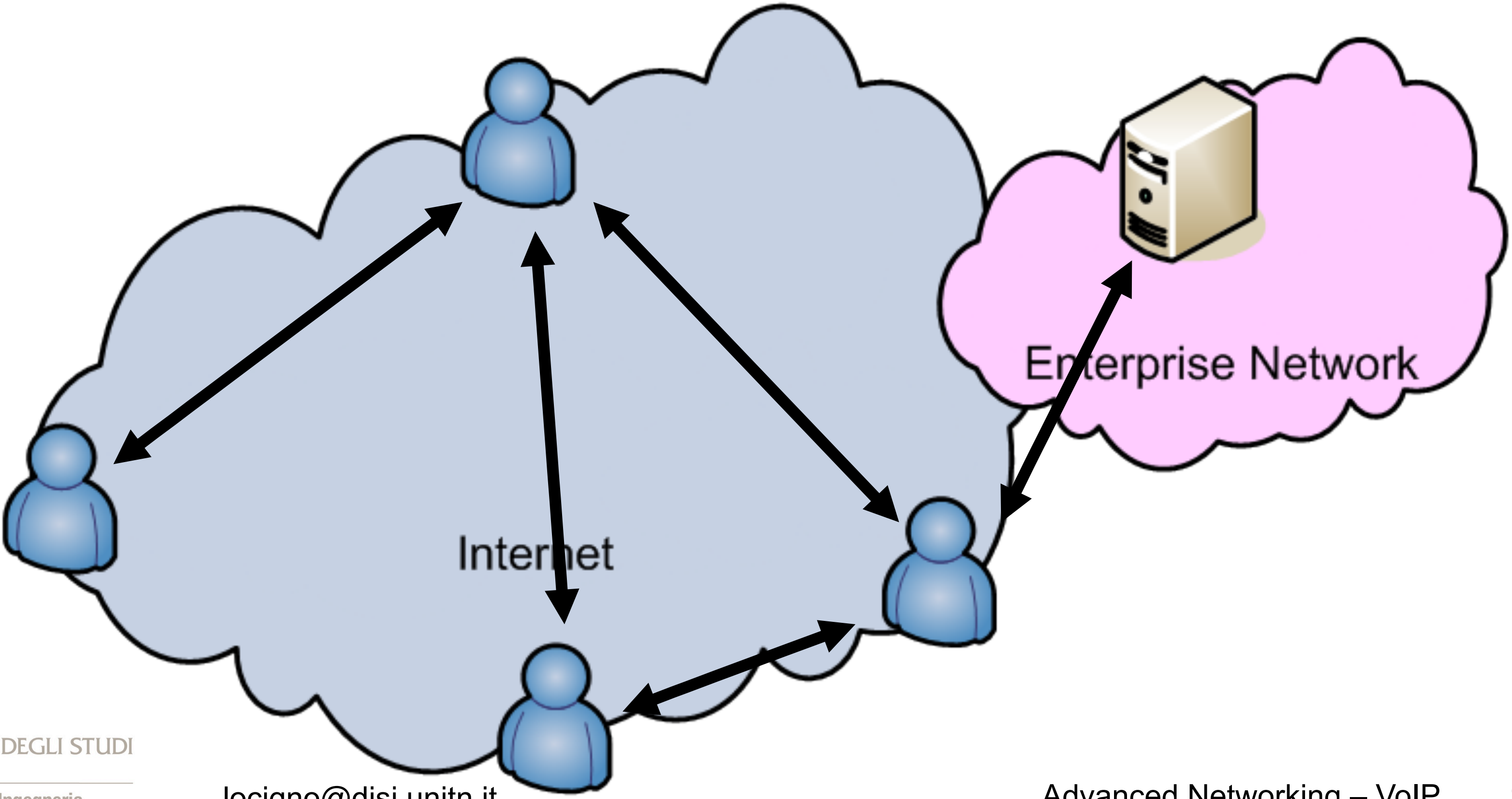
Background: Conventional VPN

- Provides private and secure connections over the public network.
- All users connect to this server: server is data hub.
- Server is bottleneck, server is single point of failure.
- Users cannot communicate among each other when disconnected from server
  - even if local connectivity between them exists, for example, in remote LAN or moving network

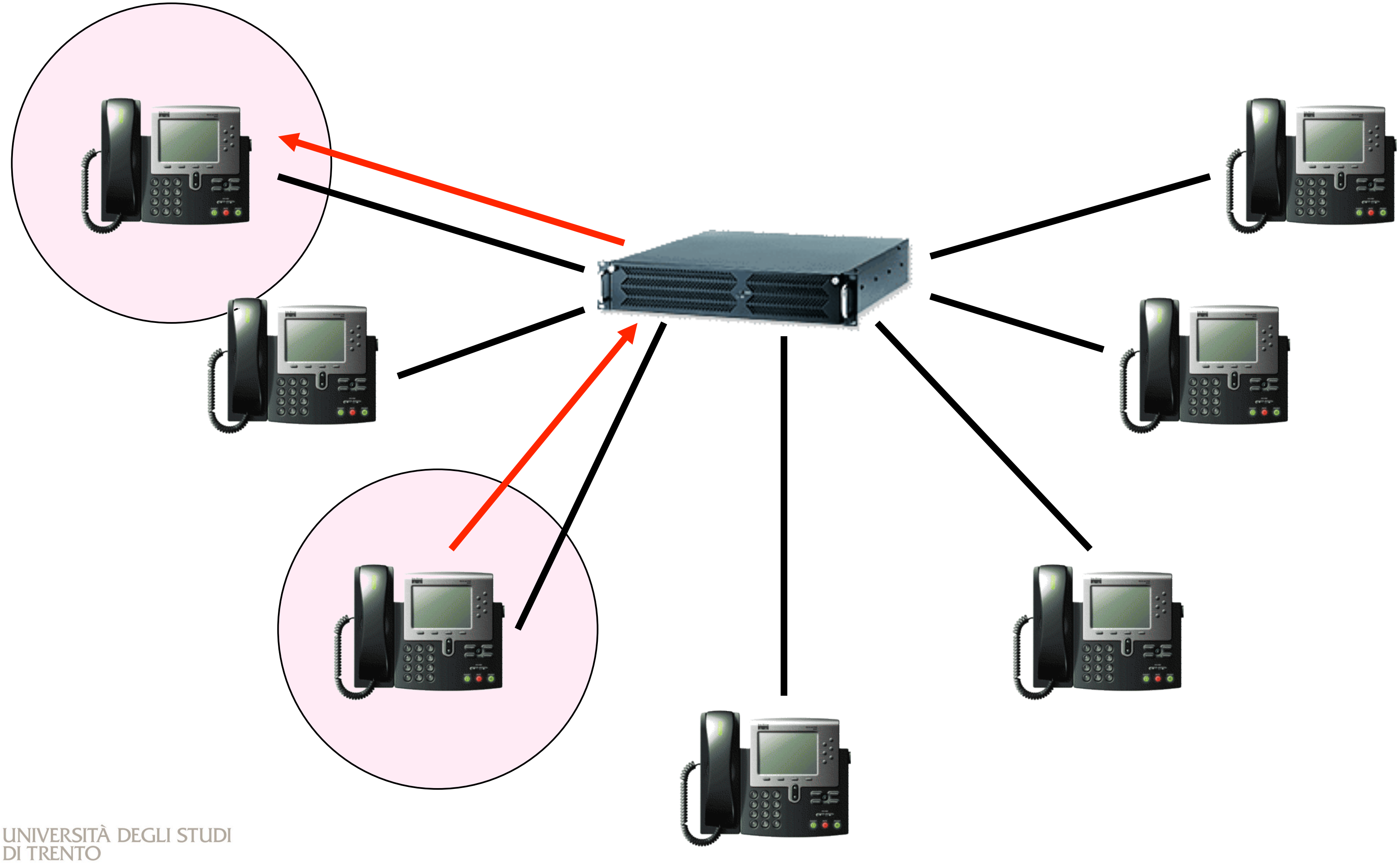


Background: Peer-To-Peer Networks

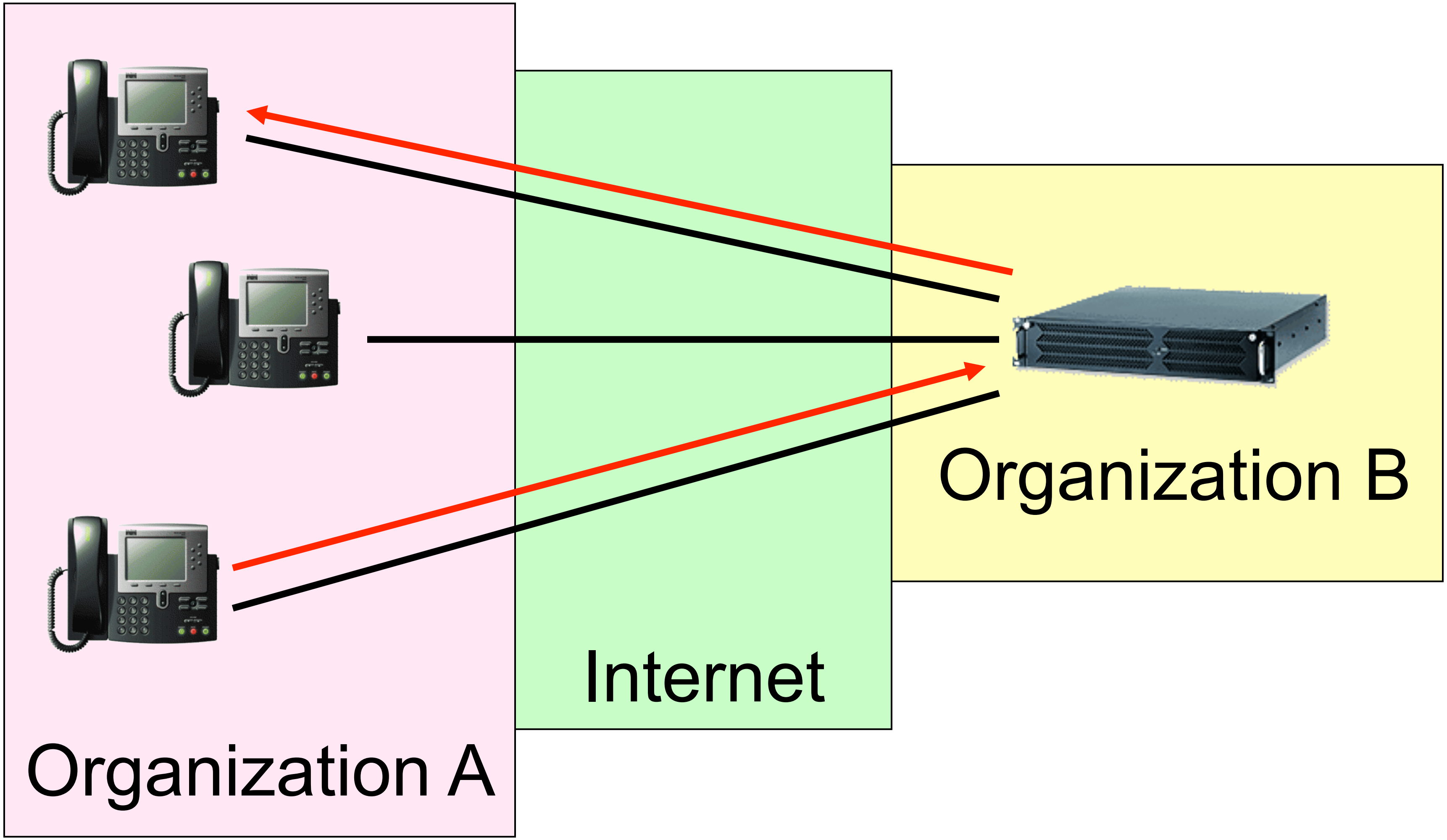
- flexible network
- no data hub



Client-Server Session



Problem with Remote Server



VoP2P Standardization

- SIP is already compatible with the P2P paradigm
  - need to substitute the register and proxy servers with distributed functions and data bases
- There are several proposals
  - If the idea is a winner … some of them will survive

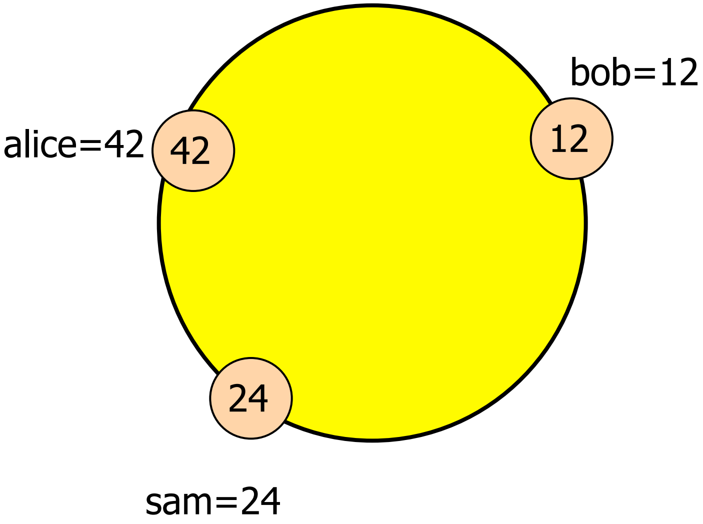
SIPeer

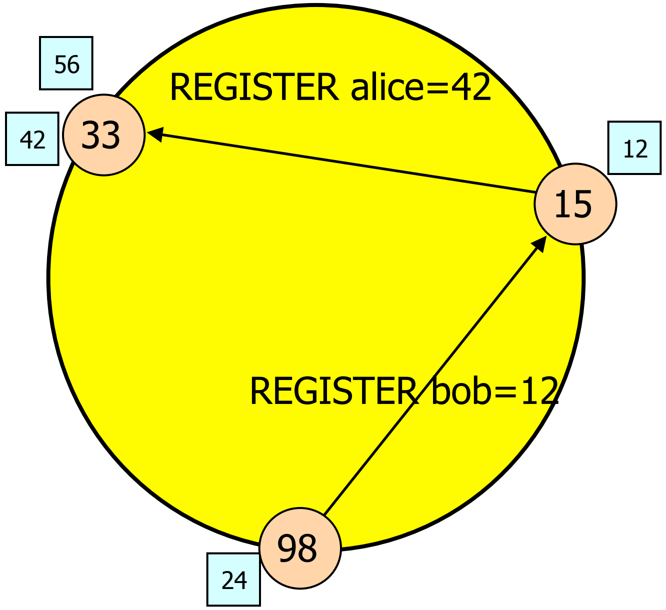
- Goals
  - Progetto P2P basato sulle primitive SIP
  - Nessuna necessità di configurazione
  - Audio conferenza e messaggistica
  - Interoperabile con i sistemi SIP esistenti
- In qualche modo si può dire ispirato a Skype
- Uso di sistemi di ricerca distribuita esistenti DHT (Distributed Hash Tables)
  - Key=hash(user@domain)

User Search: Examples

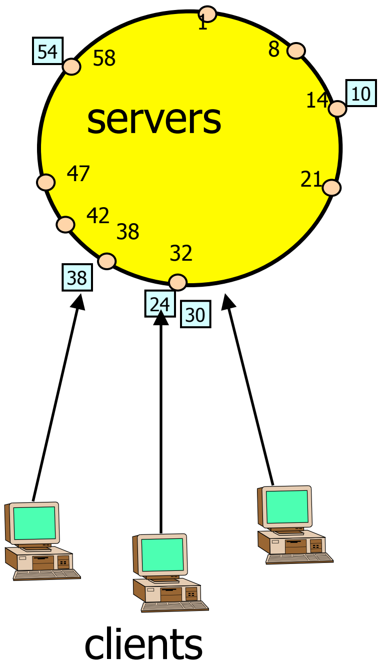
- No “REGISTER”
  - Compute a key based on the user ID
  - Nodes are connected to the P2P overlay based on the User ID
  - One node ⇔ One user

With “REGISTER”
  - The user REGISTERs with nodes that handles his key
  - Need periodic refreshing
  - Enable off-line services (voice-mail, messaging, …)

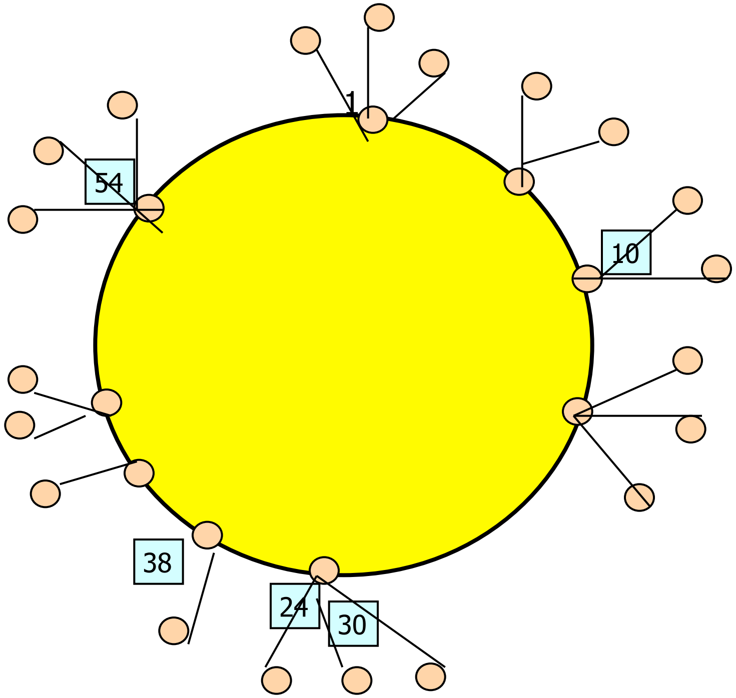




Several design alternatives



Mixed models with servers handling DHTs



Hierarchical supernodes a-la-skype

P2P real-time: Users perspective

- Ease of usage
- No user configuration required
- Working across all networking environments
  - Network Address Translators (NATs)
  - Firewalls (FWs)
- P2P real-time applications are not standard-based but they “just work”
- Different user experience with respect to standard-based real-time applications
  - e.g. H.323-based or SIP-based

Identification of issues with P2P SIP

- Goal
  - Identify potential issues of SIP-based P2P communication related to Middleboxes (NAT and firewall) traversal
    - to be considered when designing standards for a SIP-based P2P infrastructure
- Non-Goals
  - Constrain a future P2P SIP architecture in any way
  - Still we need to list potential communication steps that might raise issues
  - Those steps are not necessary part of the final SIP-based P2P solution
  - Suggest NAT traversal methods to be selected for P2P solution

Potential Communication Steps

- Steps considered
  - middlebox detection
  - registration
  - search for relays
  - address lookup
  - call setup
  - call termination
- Not all steps might be necessary
- Several steps may be combined into one

Middlebox Detection

- Detect Middleboxes
  - on the signaling path
  - on the data path
- Communication means detection for
  - registration
  - incoming / outgoing signaling
  - data streaming to and from other terminals or relays
- Checks to be performed
  - sending and receiving UDP packets
  - opening incoming and outgoing TCP connections
  - use of certain fixed port numbers
  - the option to relay or tunnel signaling messages and streamed data
- NAT parameter detection
  - full cone, half cone, etc…

Registration

- Authentication of the user
- Notification of communication capability and willingness
- Registration of contact parameters
- Notification of service provisioning capability and willingness

Further Steps

- Search and Connect Relay
  - Candidate relays may be suggested by infrastructure
- Address Lookup
  - Per-call lookup
  - Buddy list lookup
- Connection Establishment and Termination

Middlebox Traversal Methods

- **Tunneling**
  - in highly restricted environments only
  - controversial:
    - HTTP and DNS tunneling are not legitimate
    - TURN could be OK
- **Network-initiated Middlebox Signaling**
  - not the right choice for P2P SIP
- **Terminal-initiated Middlebox Signaling**
  - several methods known

Terminal-initiated Middlebox Signaling

- Standards
  - STUN (IETF RFC3489)
  - UPnP (UPnP Forum)
  - SOCKS (IETF RFC 1928)
  - RSIP (IETF RFC 3103)
- Under development
  - STUN update (IETF behave WG)
  - ICE (IETF mmusic WG)
  - NSIS (IETF nsis WG)
- Middlebox traversal using relays
  - STUN relay (previously TURN) (IETF mmusic WG)

Open Issues for SIP-based P2P

- SIP-unrelated
  - middlebox detection beyond UDP
- SIP-related
  - terminal reachability
  - communication service requirements
  - communication service offers
- The relevance of these issues strongly depends on the choice of P2P architecture