Master Course
Computer Networks
IN2097

Prof. Dr.-Ing. Georg Carle
Christian Grothoff, Ph.D.

Chair for Network Architectures and Services
Institut für Informatik
Technische Universität München
http://www.net.in.tum.de

Oral Examination

- Currently 50 students registered for examination in TUM Online

- Intended days for oral examinations:
  - Thursday 16 February
  - Thursday 8 March
  - Thursday 29 March
  - Thursday 19 April

Chair for Network Architectures and Services – Prof. Carle
Department for Computer Science
TU München

Reliable Multicast Transport

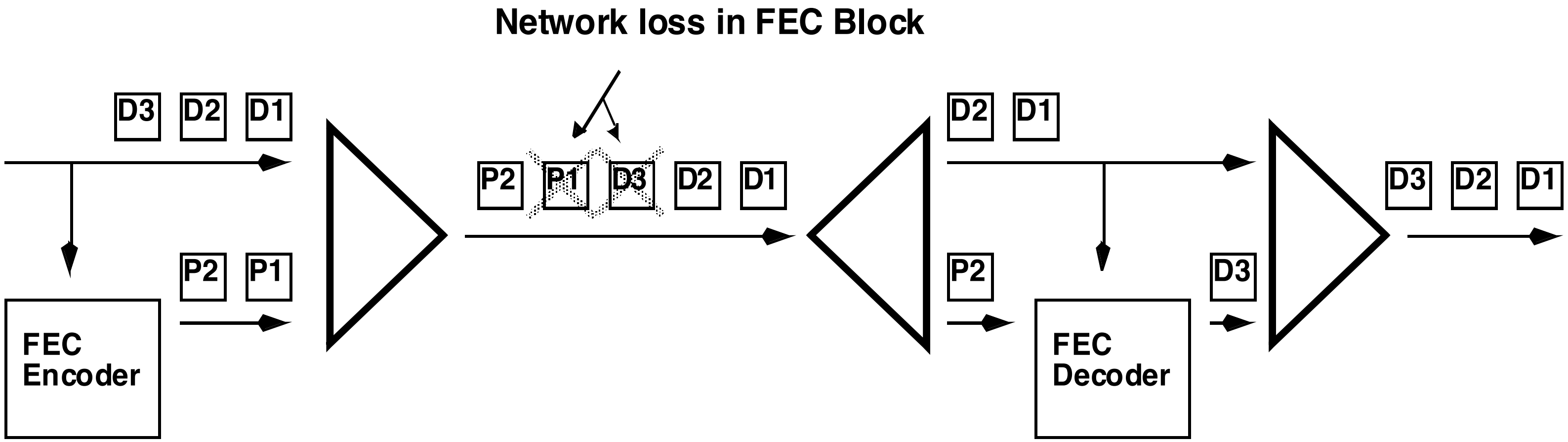
Technische Universität München

Approaches

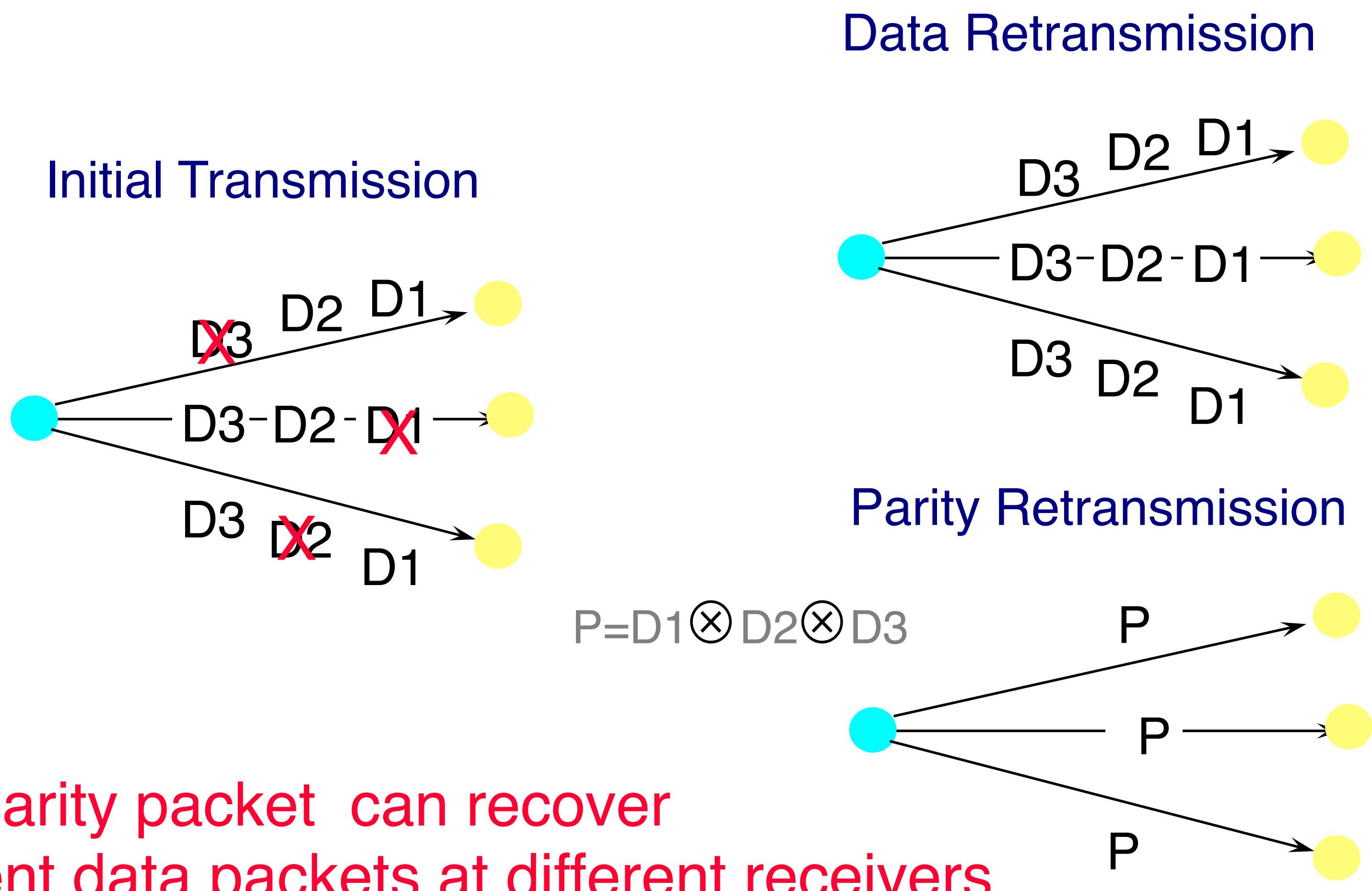
- shift responsibilities to receivers (in contrast to TCP: sender is responsible for large share of functionality)
- feedback suppression (some feedback is usually required)
- multiple multicast groups (e.g. for heterogeneity problems; can be used statically or dynamically)
- local recovery (can be used to reduce resource cost and latency)
- server-based recovery
- forward error correction (FEC)
  - FEC for unicast: frequently no particular gain
  - FEC for multicast: gain may be tremendous!

Forward Error Correction (FEC)

- k original data packets form a **Transmission Group (TG)**
- h parity packets derived from the k data packets
- any k received out of k+h are sufficient
- Assessment
  - \+ allows to recover lost packets
  - \- overhead at end-hosts
  - \- increased network load may increase loss probability



Potential Benefits of FEC



One parity packet can recover
different data packets at different receivers

Classification of Multicast Error Control

Multicast Error Recovery

Centralized Error Recovery (CER):
Source retransmits

Distributed Error Recovery (DER): retransmission by server or receiver

grouped (local):
Multicast group is
partitioned into subgroups

ungrouped (global):
All group members participate in error recovery

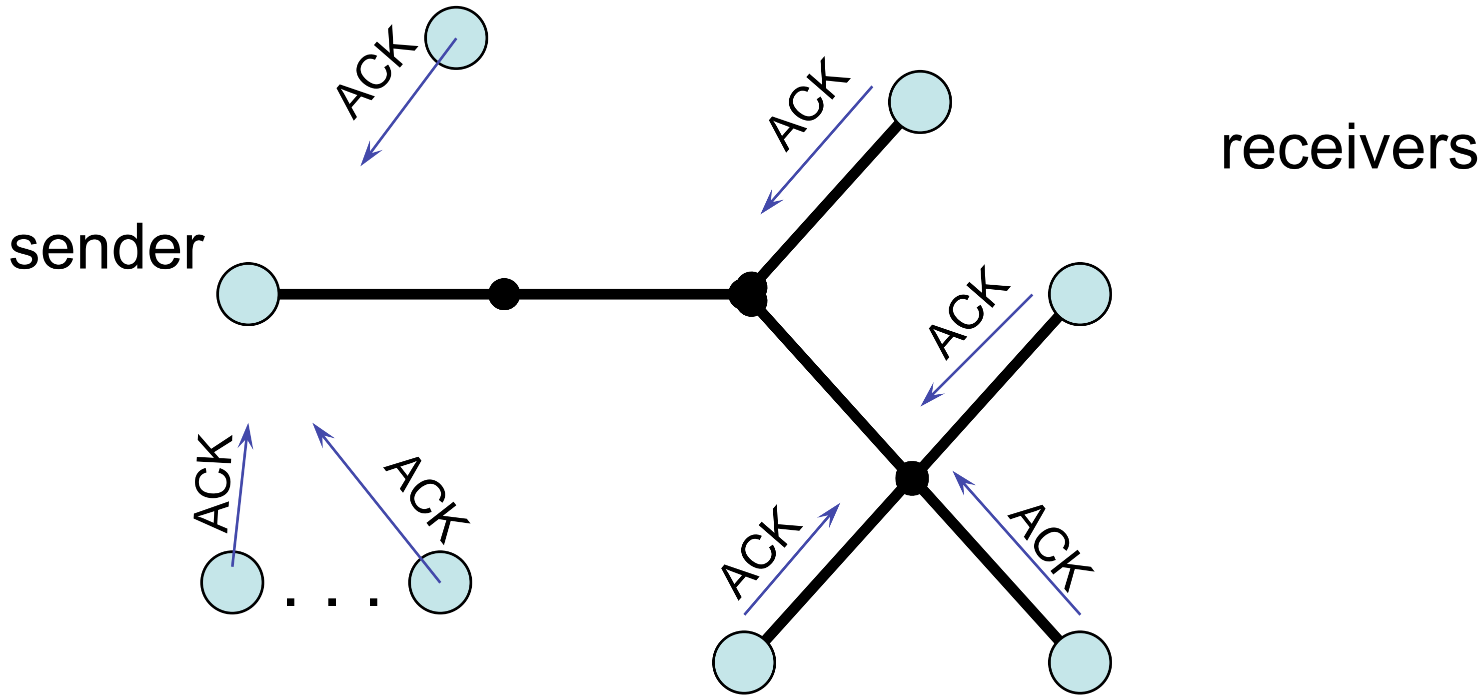
Reliable Multicast: Building Blocks

- Elements from Unicast:
  - Loss detection
    - Sender-based (ACK): 1 ACK per receiver and per packet; Sender needs a table of per-receiver ACK
    - Receiver-based (NAK): distributed over receivers; potentially only 1 NAK per lost packet
  - Loss recovery: ARQ vs. FEC
- Additional new elements for Multicast:
  - Mechanisms for control message **Implosion Avoidance**
  - Mechanisms to deal with *heterogeneous receivers*

Feedback Processing

- Assume: R Receivers, independent packet loss probability p
- Calculate feedback per packet:
  - average number of ACKs: R - pR
  - average number of NAKs: pR
  - ⇨ more ACKs than NAKs
- Processing: higher throughput for receiver-based loss detection
- Reliability needs ACKs
  (No NAK does not mean successful reception)
  - ⇨ use NAK for loss signalling
  - ⇨ use ACKs at low frequency to ensure reliability

Multicast Challenge: Feedback Implosion Problem



NAK Implosion

- Shared loss: All receivers loose same packet: All send NAK
  ⇨ NAK implosion
- Implosion avoidance techniques
  - Cluster/Hierarchy
  - Token
  - Timers

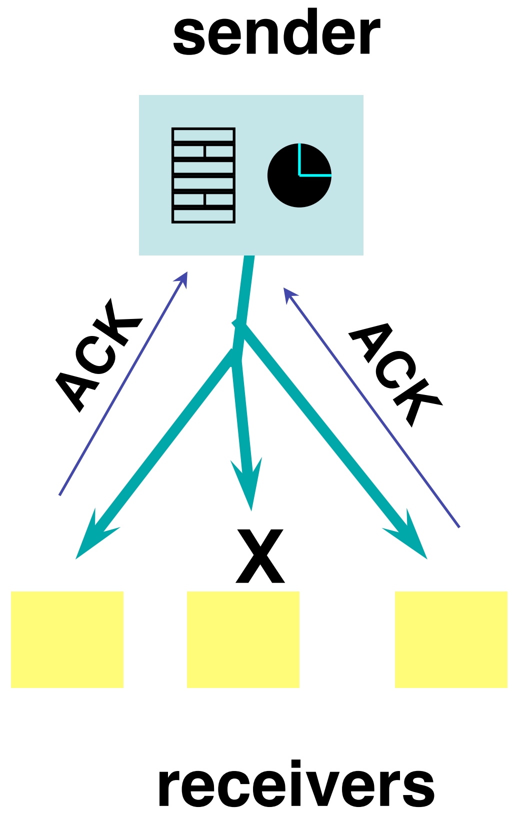
  For redundant feedback additionally:
  - Feedback suppression (e.g. multicast NAKs, receiver back off randomly)

  Drawback of implosion avoidance techniques: delay
- Fast NAKs (risk of NAK implosion):
  - Fast retransmission
  - Smaller sender/receiver buffer

  ⇨ design tradeoffs

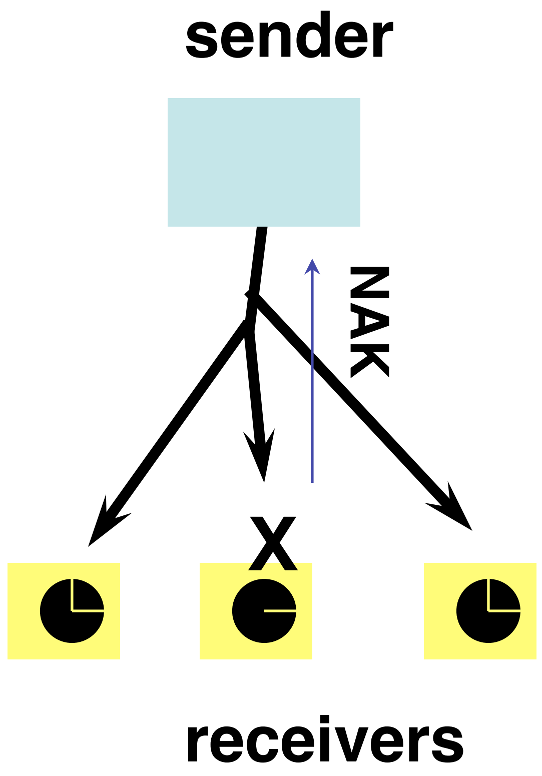
Sender Oriented Reliable Multicast

- Sender:
  multicasts all (re)transmissions
  - selective repeat
  - use of timeouts for loss detection
  - ACK table
- receiver: ACKs received packets
- Note: group membership important
- Example (historic):
  Xpress Transport Protocol (XTP)
  - extension of unicast protocol



Receiver Oriented Reliable Multicast

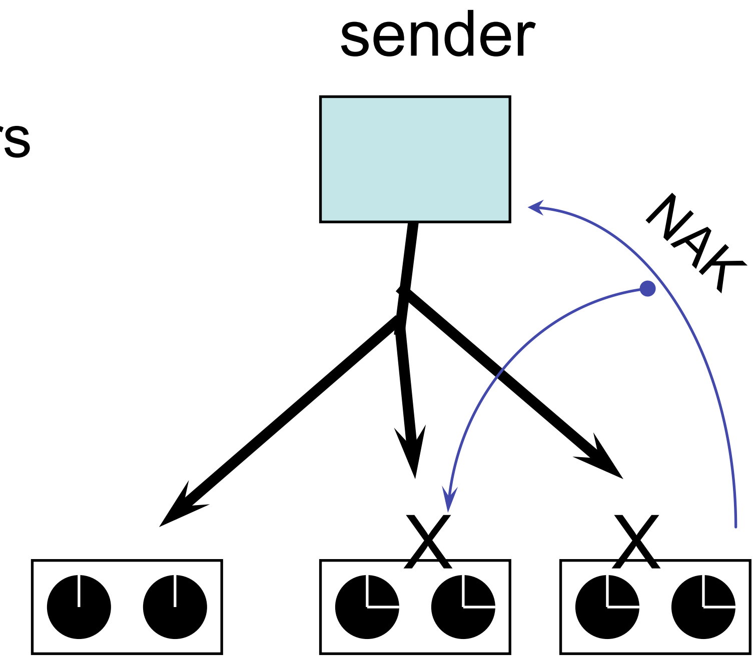
- Sender: multicasts (re)transmissions
  - selective repeat
  - responds to NAKs
- Receiver: upon detecting packet loss
  - sends pt-pt NAK
  - timers to detect lost retransmission
- Note: easy to allow joins/leaves



Feedback Suppression

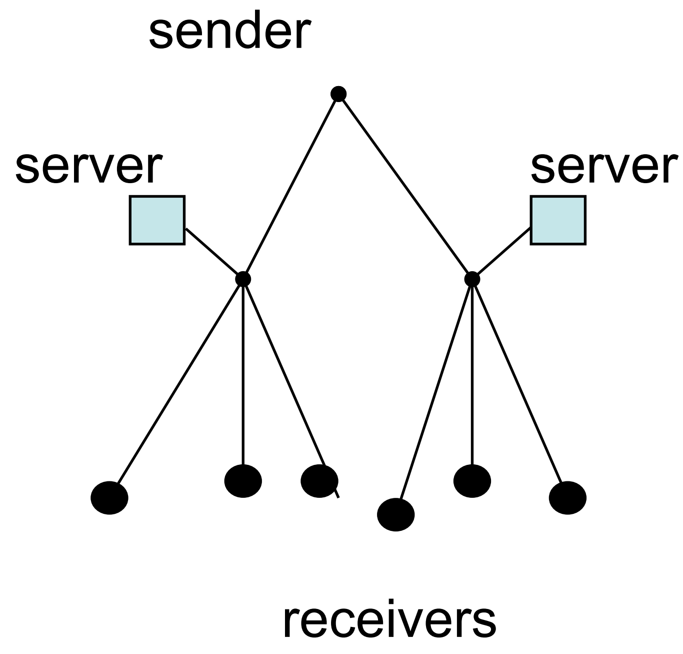
- randomly delay NAKs
- multicast to all receivers
  + reduce bandwidth
  - additional complexity at receivers (timers, etc)
  - increase latencies (timers)

- similar to CSMA/CD



Server-based Reliable Multicast

- first transmissions: multicast to all receivers and servers
- each receiver assigned to server
- servers perform loss recovery
- servers can be subset of receivers or provided by network
- can have more than 2 levels



Assessment:

- clear performance benefits
- how to configure
  - static/dynamic
  - many-many

Local Recovery

- lost packets recovered from nearby receivers

- deterministic methods
  - impose tree structure on receivers with sender as root
  - receiver goes to upstream node on tree

- self-organizing methods
  - receivers elect nearby receiver to act as retransmitter

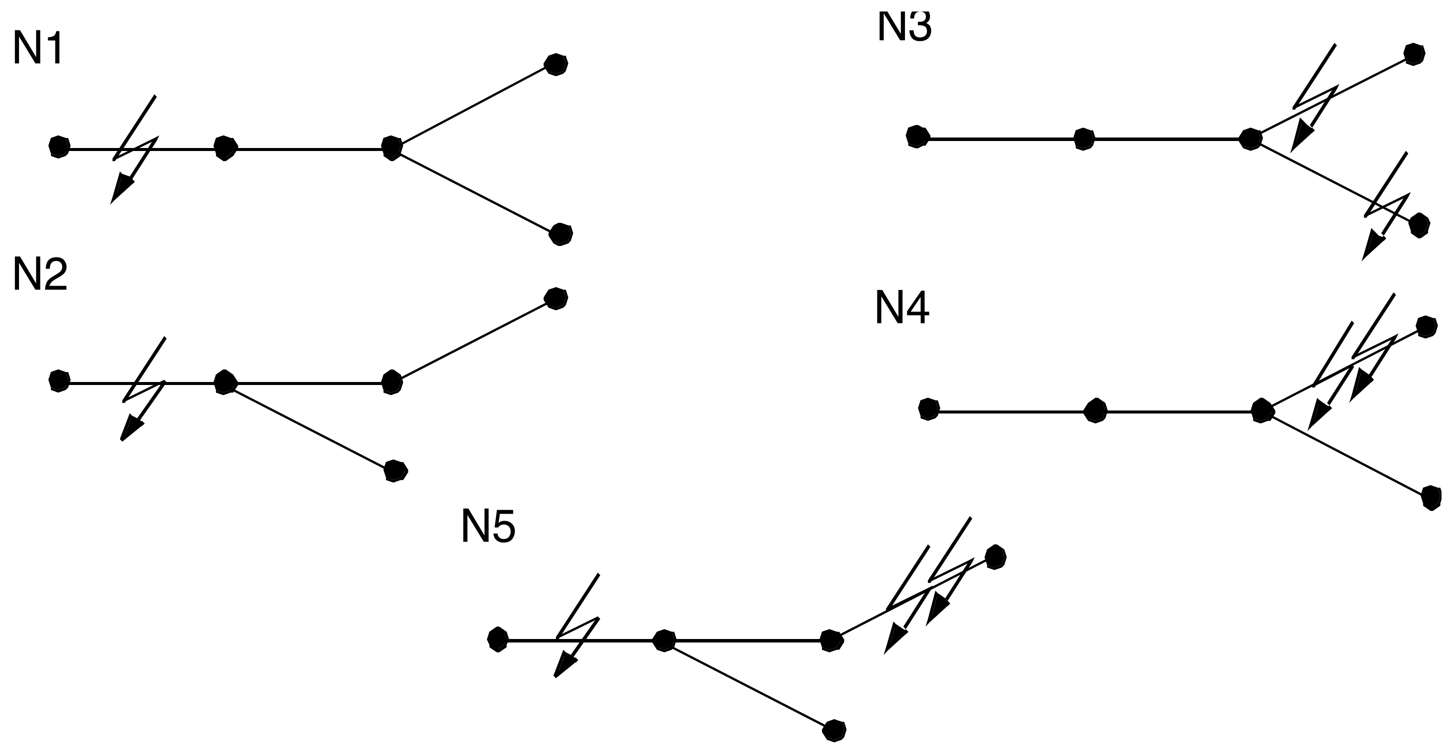
- hybrid methods

Issues with Server- and Local Based Recovery

- how to configure tree
- what constitutes a local group
- how to permit joins/leaves
- how to adapt to time-varying network conditions

Influence of topology: Selected Scenarios for Modeling Heterogeneity

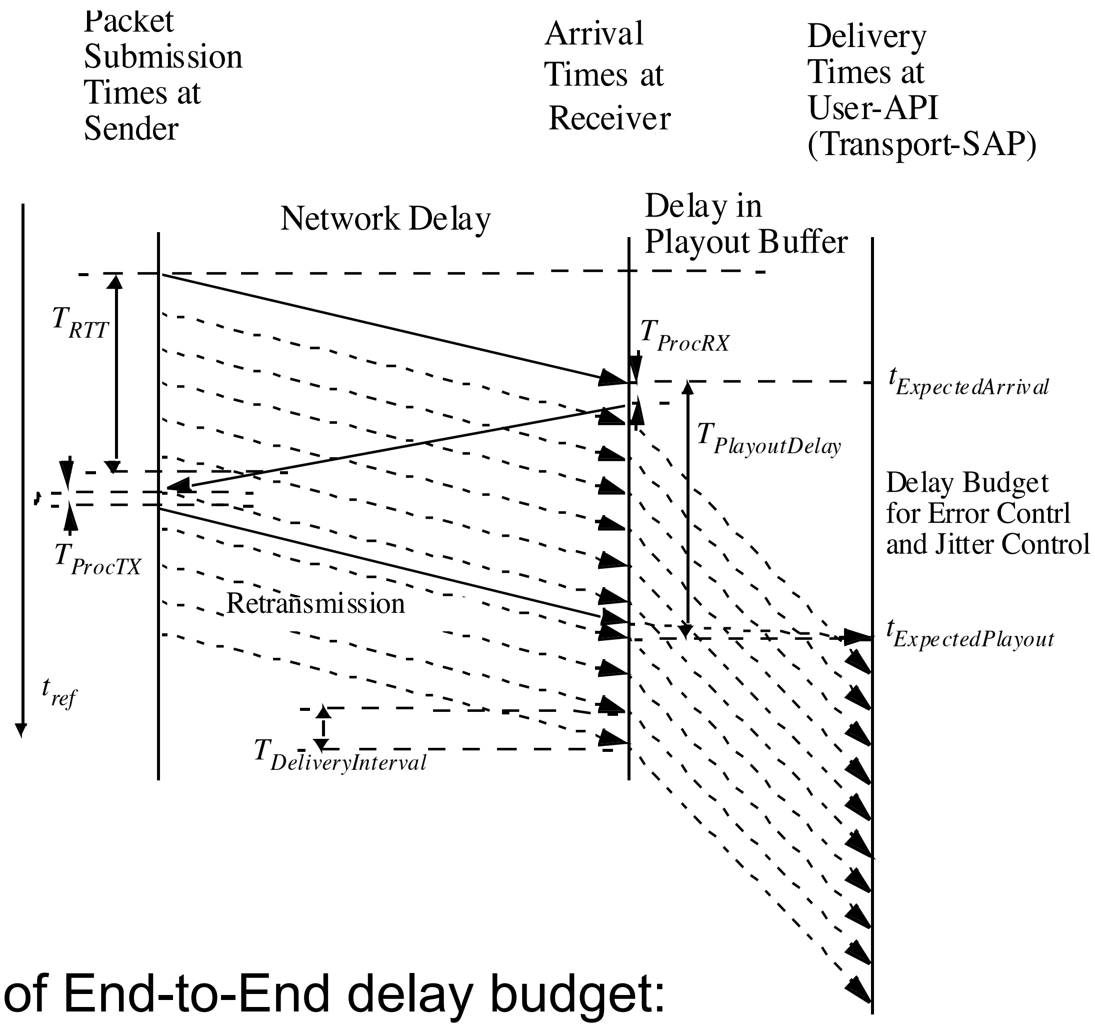
- Loss: on shared links / on individual links
- Loss: homogeneous / heterogeneous probability
- RTT: homogeneous / heterogeneous



Scenario-specific Selection of Mechanisms

- FEC is of particular benefit in the following scenarios:
  - Large groups
  - No feedback
  - Heterogeneous RTTs
  - Limited buffer
- ARQ is of particular benefit in the following scenarios:
  - Herterogeneous loss
  - Loss in shared links of multicast tree dominates
  - Small groups (Statistic by AT&T: on average < 7 participants in conference)
  - Non-interactive applications
- ARQ by local recovery:
  - large groups (good for individual losses, heterogeneous RTT)

Reliable Multicast for Audio-Visual Applications



- Exploitation of End-to-End delay budget:
  we can always trade-off reliability for delay!
  (e.g. use 10 s delay budget to get 20% loss probability down to 2%)

Chapter - Transport Layer: Summary

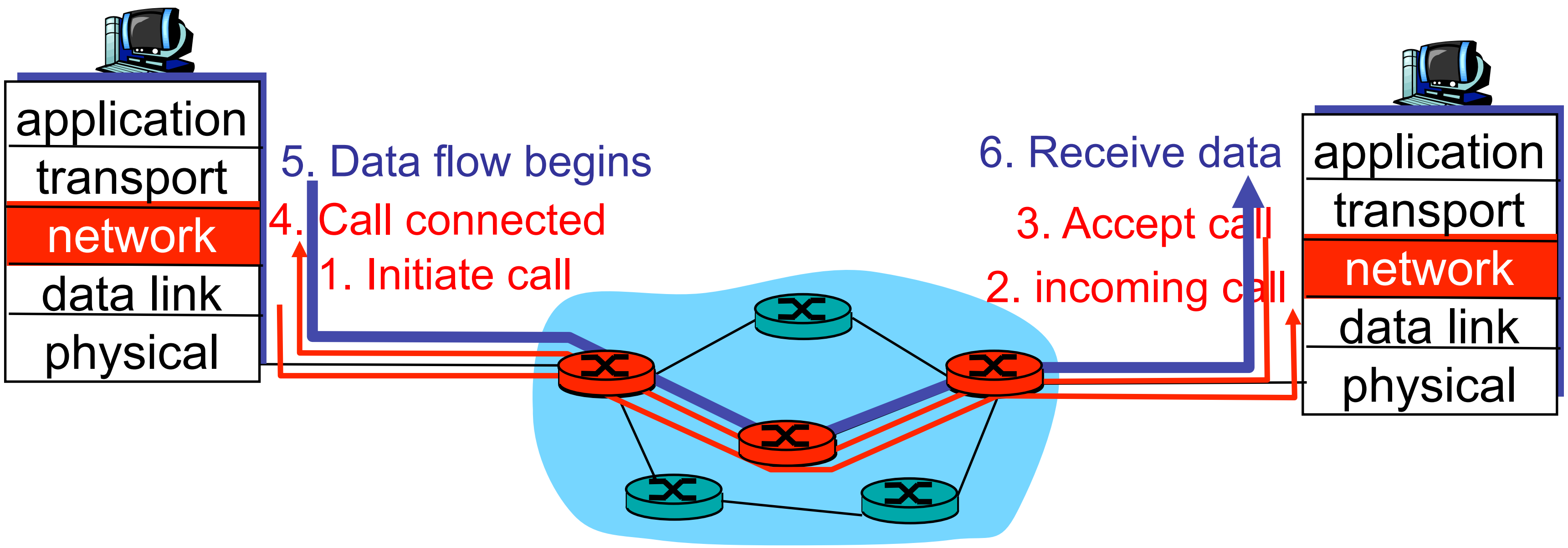
- Principles behind transport layer services:
  - multiplexing, demultiplexing
  - reliable data transfer
  - flow control
  - congestion control
- Instantiation and implementation in the Internet
  - UDP
  - TCP
  - SCTP
  - Reliable multicast protocols

Chair for Network Architectures and Services – Prof. Carle
Department for Computer Science
TU München

Signalling

Virtual circuits: signaling protocols

- used to set up, maintain teardown VC
- used in (G)MPLS, ATM, frame-relay, X.25
- not used in today's Internet at L3 (network layer)



Signaling

signaling: exchange of messages among network entities to enable (provide service) to connection/call

- before, during, after connection/call
  - call setup and teardown (state)
  - call maintenance (state)
  - measurement, billing (state)
- between
  - end-user <-> network
  - end-user <-> end-user
  - network element <-> network element
- examples
  - Q.921, SS7 (Signaling System no. 7): telephone network
  - Q.2931: ATM
  - RSVP (Resource Reservation Protocol)
  - H.323: Internet telephony
  - **SIP** (Session Initiation Protocol): Internet telephony

SIP

Credits in addition to
Jim Kurose and Keith Ross:

Julie Chan, Vovida Networks.
Milind Nimesh, Columbia University
Christian Hoene, University of Tübingen

SIP: Session Initiation Protocol [RFC 3261]

SIP long-term vision:

all telephone calls, video conference calls take place over Internet

- people are identified by names or e-mail addresses, rather than by phone numbers
- you can reach callee, no matter where callee roams, no matter what IP device callee is currently using

SIP key person:

Henning Schulzrinne, Columbia University

- M. Handley, H. Schulzrinne, and E. Schooler, "SIP: session initiation protocol," Internet Draft, Internet Engineering Task Force, March 1997. Work in progress.
- H. Schulzrinne, A comprehensive multimedia control architecture for the Internet, 1997

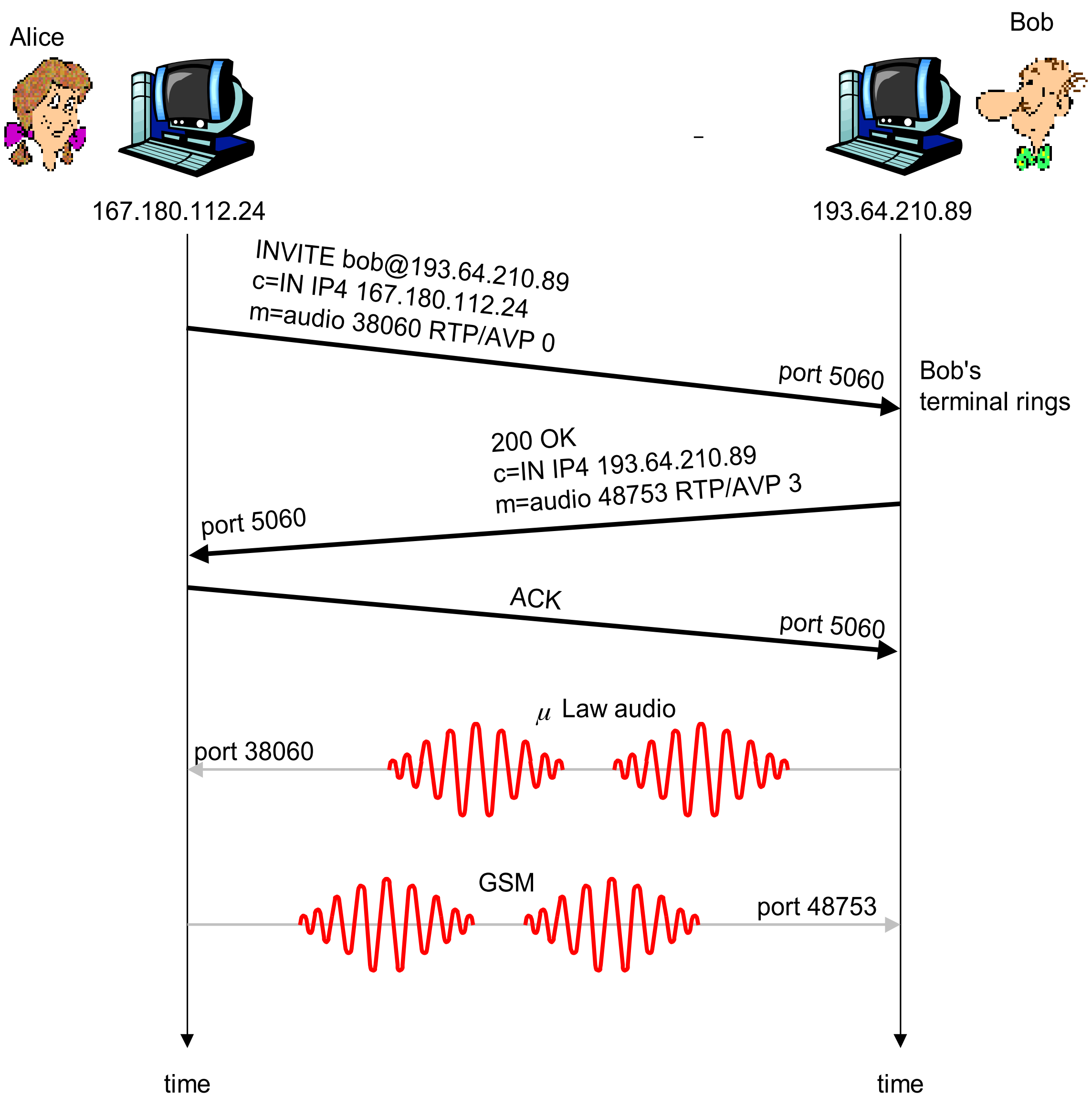
SIP

- IETF RFC 2543: Session Initiation Protocol –
  An application layer signalling protocol that defines initiation, modification and termination of interactive, multimedia communication *sessions* between users.
- Sessions include
  - voice
  - video
  - chat
  - interactive games
  - virtual reality
- SIP is a text-based protocol, similar to HTTP and SMTP.

SIP Services

- Setting up a call, SIP provides mechanism
  - for caller to let callee know she wants to establish a call
  - so caller, callee can agree on media type, encoding
  - to end call
- determine current IP address of callee:
  - maps mnemonic identifier to current IP address
- call management:
  - add new media streams during call
  - change encoding during call
  - invite others
  - transfer, hold calls

Setting up a call to known IP address



- Alice's SIP invite message indicates her port number, IP address, encoding she prefers to receive (e.g. AVP 0: PCM ulaw)
- Bob's 200 OK message indicates his port number, IP address, preferred encoding (e.g. AVP 3: GSM)
- SIP is an out-of-band signalling protocol
- SIP messages can be sent over TCP or UDP. (All messages are ack'ed)
- default SIP port number is 5060.

Setting up a call (more)

- codec negotiation:
  - suppose Bob doesn't have PCM ulaw encoder.
  - Bob will instead reply with 606 Not Acceptable Reply, listing his encoders
  - Alice can then send new INVITE message, advertising different encoder
- rejecting a call
  - Bob can reject with replies "busy," "gone," "payment required," "forbidden"
- media can be sent over RTP or some other protocol

Example of SIP message

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

- Here we don't know Bob's IP address. Intermediate SIP servers needed.
- Alice sends, receives SIP messages using SIP default port 5060
- Via: header specifies intermediate server(s)

Notes:

- HTTP message syntax
- sdp = session description protocol
- Call-ID is unique for every call.

Name translation and user location

- caller wants to call callee, but only has callee's name or e-mail address.
- need to get IP address of callee's current host:
  - user moves around
  - DHCP protocol
  - user has different IP devices (PC, PDA, car device)
- result can be based on:
  - time of day (work, home)
  - caller (e.g. don't want boss to call you at home)
  - status of callee (calls sent to voicemail when callee is already talking to someone)

Service provided by SIP servers:

- SIP registrar server
- SIP proxy server

SIP Registrar

- when Bob starts SIP client, client sends SIP REGISTER message to Bob's registrar server

  (similar function needed by Instant Messaging)
- registrar analogous to authoritative DNS server

Register Message:

```
REGISTER sip:domain.com SIP/2.0
Via: SIP/2.0/UDP 193.64.210.89
From: sip:bob@domain.com
To: sip:bob@domain.com
Expires: 3600
```

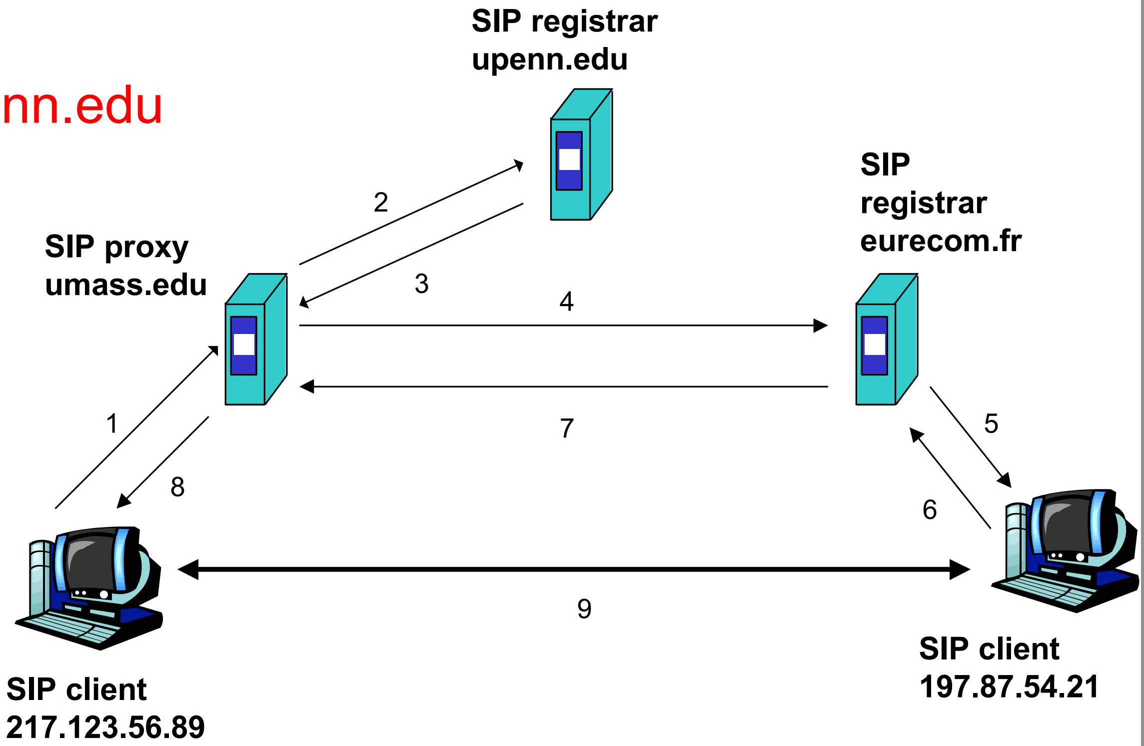
SIP Proxy

- Alice sends invite message to her proxy server
  - contains address sip:bob@domain.com
- proxy responsible for routing SIP messages to callee
  - possibly through multiple proxies.
- callee sends response back through the same set of proxies.
- proxy returns SIP response message to Alice
  - contains Bob's IP address
- proxy analogous to local DNS server

Example

Caller jim@umass.edu
places a call to keith@upenn.edu

(1) Jim sends INVITE message to umass SIP proxy.
(2) Proxy forwards request to upenn registrar server.
(3) upenn server returns redirect response, indicating that it should try keith@eurecom.fr

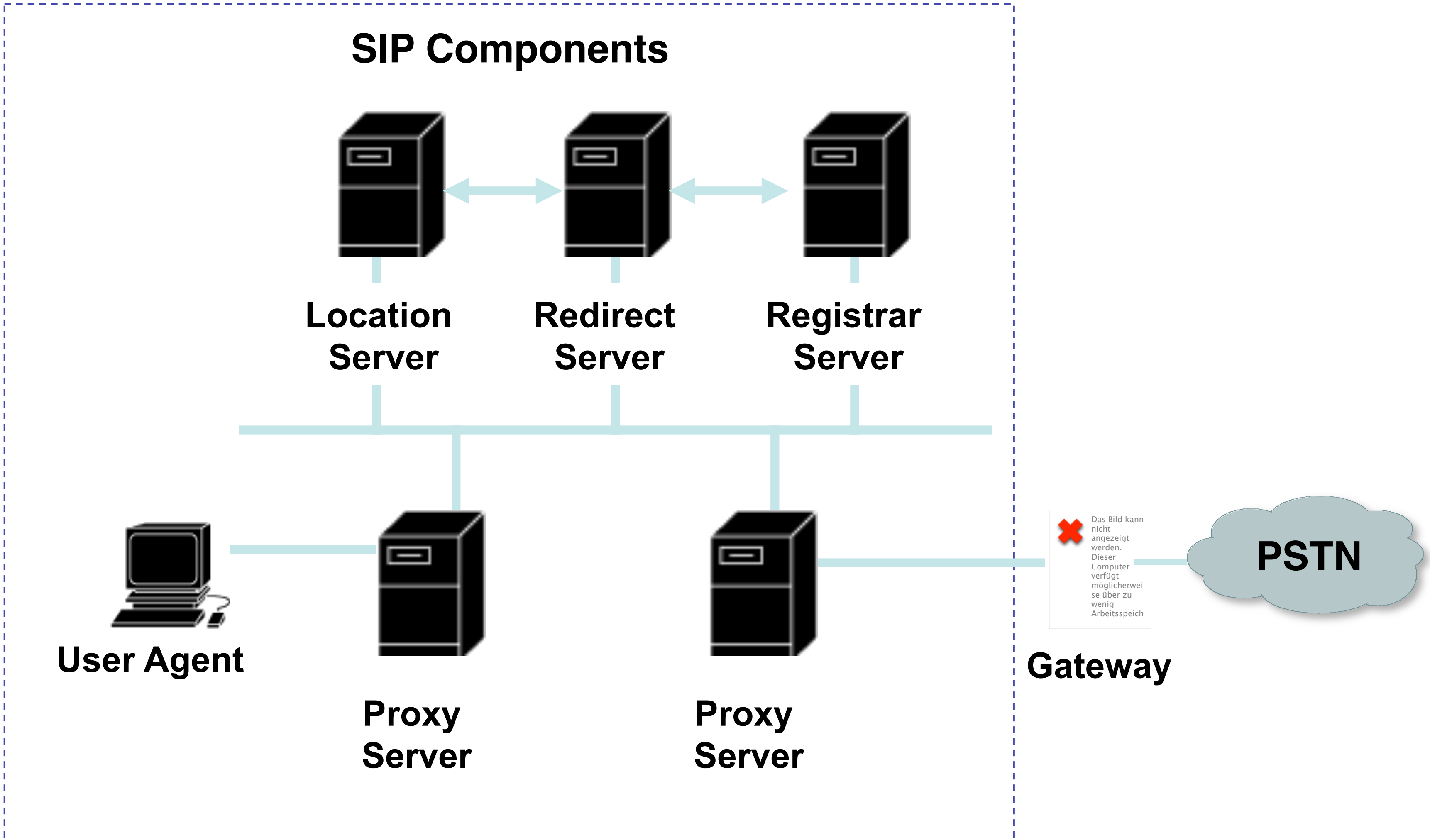


(4) umass proxy sends INVITE to eurecom registrar.
(5) eurecom registrar forwards INVITE to 197.87.54.21, which is running keith's SIP client.
(6-8) SIP response sent back
(9) media sent directly between clients.
Note: SIP ack messages not shown.

SIP consists of a few RFCs

RFC	Description
2976	The SIP INFO Method
3361	DHCP Option for SIP Servers
3310	Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)
3311	The Session Initiation Protocol UPDATE Method
3420	Internet Media Type message/sipfrag
3325	Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks
3323	A Privacy Mechanism for the Session Initiation Protocol (SIP)
3428	Session Initiation Protocol Extension for Instant Messaging
3326	The Reason Header Field for the Session Initiation Protocol (SIP)
3327	Session Initiation Protocol Extension for Registering Non-Adjacent Contacts
3329	Security Mechanism Agreement for the Session Initiation Protocol (SIP) Sessions
3313	Private Session Initiation Protocol (SIP)Extensions for Media Authorization
3486	Compressing the Session Initiation Protocol
3515	The Session Initiation Protocol (SIP) Refer Method
3319	Dynamic Host Configuration Protocol (DHCPv6)Options for Session Initiation Protocol (SIP) Servers
3581	An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing
3608	Session Initiation Protocol Extension Header Field for Service Route Discovery During Registration
3853	S/MIME AES Requirement for SIP
3840	Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)
3841	Caller Preferences for the Session Initiation Protocol (SIP)
3891	The Session Inititation Protocol (SIP) 'Replaces' Header
3892	The SIP Referred-By Mechanism
3893	SIP Authenticated Identity Body (AIB) Format
3903	An Event State Publication Extension to the Session Initiation Protocol (SIP)
3911	The Session Inititation Protocol (SIP) 'Join' Header
3968	The Internet Assigned Number Authority (IANA) Header Field Parameter Registry for the Session Initiation Protocol (SIP)
3969	The Internet Assigned Number Authority (IANA) Universal Resource Identifier (URI) Parameter Registry for the Session Initiation Protocol (SIP)
4032	Update to the Session Initiation Protocol (SIP) Preconditions Framework
4028	Session Timers in the Session Initiation Protocol (SIP)
4092	Usage of the Session Description Protocol (SDP) Alternative Network Address Types (ANAT) Semantics in the Session Initiation Protocol (SIP)
4168	The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP)
4244	An Extension to the Session Initiation Protocol (SIP) for Request History Information
4320	Actions Addressing Identified Issues with the Session Initiation Protocol's (SIP) non-INVITE Transaction
4321	Problems identified associated with the Session Initiation Protocol's (SIP) non-INVITE Transaction
4412	Communications Resource Priority for the Session Initiation Protocol (SIP)
4488	Suppression of Session Initiation Protocol (SIP) REFER Method Implicit Subscription
4508	Conveying Feature Tags with Session Initiation Protocol (SIP) REFER Method
4483	A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages
4485	Guidelines for Authors of Extensions to the Session Initiation Protocol (SIP)

SIP Architecture



User Agents

- An application that initiates, receives and terminates calls.
  - User Agent Clients (UAC) – An entity that initiates a call.
  - User Agent Server (UAS) – An entity that receives a call.

  - Both UAC and UAS can terminate a call.

Proxy Server

- An intermediary program that acts as both
  a server and a client
  to make requests on behalf of other clients.
- Requests are serviced internally or passed on,
  possibly after translation, to other servers.
- Interprets, rewrites or translates a request message
  before forwarding it.

Registrar Server

- A server that accepts REGISTER requests.
- The registrar server may support authentication.
- A registrar server is typically co-located with a proxy or redirect server and may offer location services.

Redirect Server

- A server that accepts a SIP request, maps the address into zero or more new addresses and returns these addresses to the client.
- Unlike proxy server, the redirect server does not initiate own SIP requests
- Unlike a user agent server, the redirect server does not accept or terminate calls.
- The redirect server generates 3xx responses to requests it receives, directing the client to contact an alternate set of URIs.
- In some architectures it may be desirable to reduce the processing load on proxy servers that are responsible for routing requests, and improve signaling path robustness, by relying on redirection.
- Redirection allows servers to push routing information for a request back to the client, thereby taking themselves out of the loop of further messaging while still aiding in locating the target of the request.
  - When the originator of the request receives the redirection, it will send a new request based on the URI(s) it has received.
  - By propagating URIs from the core of the network to its edges, redirection allows for considerable network scalability.
- C.f. iterative (non-recursive) DNS queries

Location Server

- A location server is used by a SIP redirect or proxy server to obtain information about a called party’s possible location(s).

- A location Server is a logical IP server that transmits a Presence Information Data Format - Location Object (PIDF-LO).
- A PIDF-LO is an XML Scheme for carrying geographic location of a target.
- As stated in RFC 3693, location often must be kept private. The Location Object (PIDF-LO) contains rules which provides guidance to the Location Recipient and controls onward distribution and retention of the location.

SIP Messages – Methods and Responses

SIP components communicate by exchanging SIP messages:

<u>SIP Methods:</u>

- INVITE – Initiates a call by inviting user to participate in session.
- ACK - Confirms that the client has received a final response to an INVITE request.
- BYE - Indicates termination of the call.
- CANCEL - Cancels a pending request.
- REGISTER – Registers the user agent.
- OPTIONS – Used to query the capabilities of a server.
- INFO – Used to carry out-of-band information, such as DTMF (Dual-tone multi-frequency) digits.

<u>SIP Responses:</u>

- 1xx - Informational Messages.
- 2xx - Successful Responses.
- 3xx - Redirection Responses.
- 4xx - Request Failure Responses.
- 5xx - Server Failure Responses.
- 6xx - Global Failures Responses.

SIP Headers

- SIP borrows much of the syntax and semantics from HTTP.
- A SIP messages looks like an HTTP message:
  message formatting, header and MIME support.
- An example SIP header:

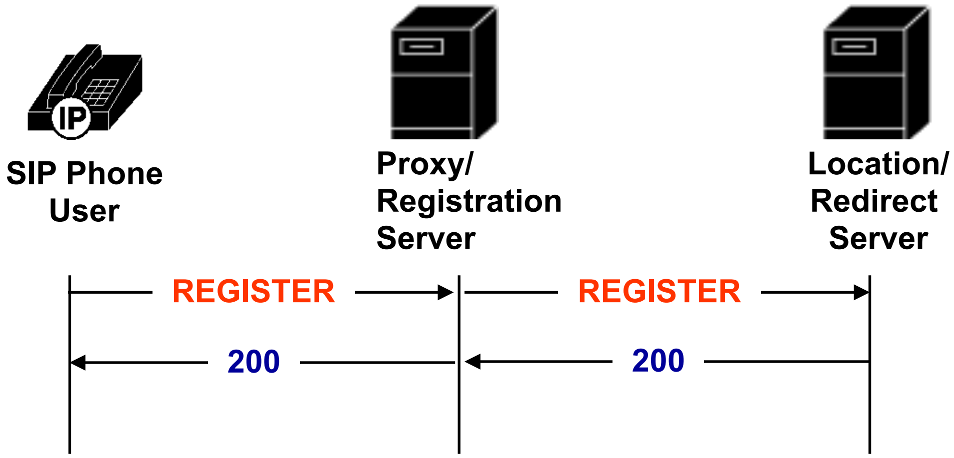
```
----------------------------------------------------------------
                        SIP Header
----------------------------------------------------------------
INVITE sip:5120@192.168.36.180 SIP/2.0
Via: SIP/2.0/UDP 192.168.6.21:5060
From: sip:5121@192.168.6.21
To: <sip:5120@192.168.36.180>
Call-ID: c2943000-e0563-2a1ce-2e323931@192.168.6.21
CSeq: 100 INVITE
Expires: 180
User-Agent: Cisco IP Phone/ Rev. 1/ SIP enabled
Accept: application/sdp
Contact: sip:5121@192.168.6.21:5060
Content-Type: application/sdp
```

SIP Addressing

- The SIP address is identified by a SIP URL, in the format: user@host.
- Examples of SIP URLs:
  - sip:user@domain.com
  - sip:user@192.168.10.1
  - sip:14083831088@domain.com

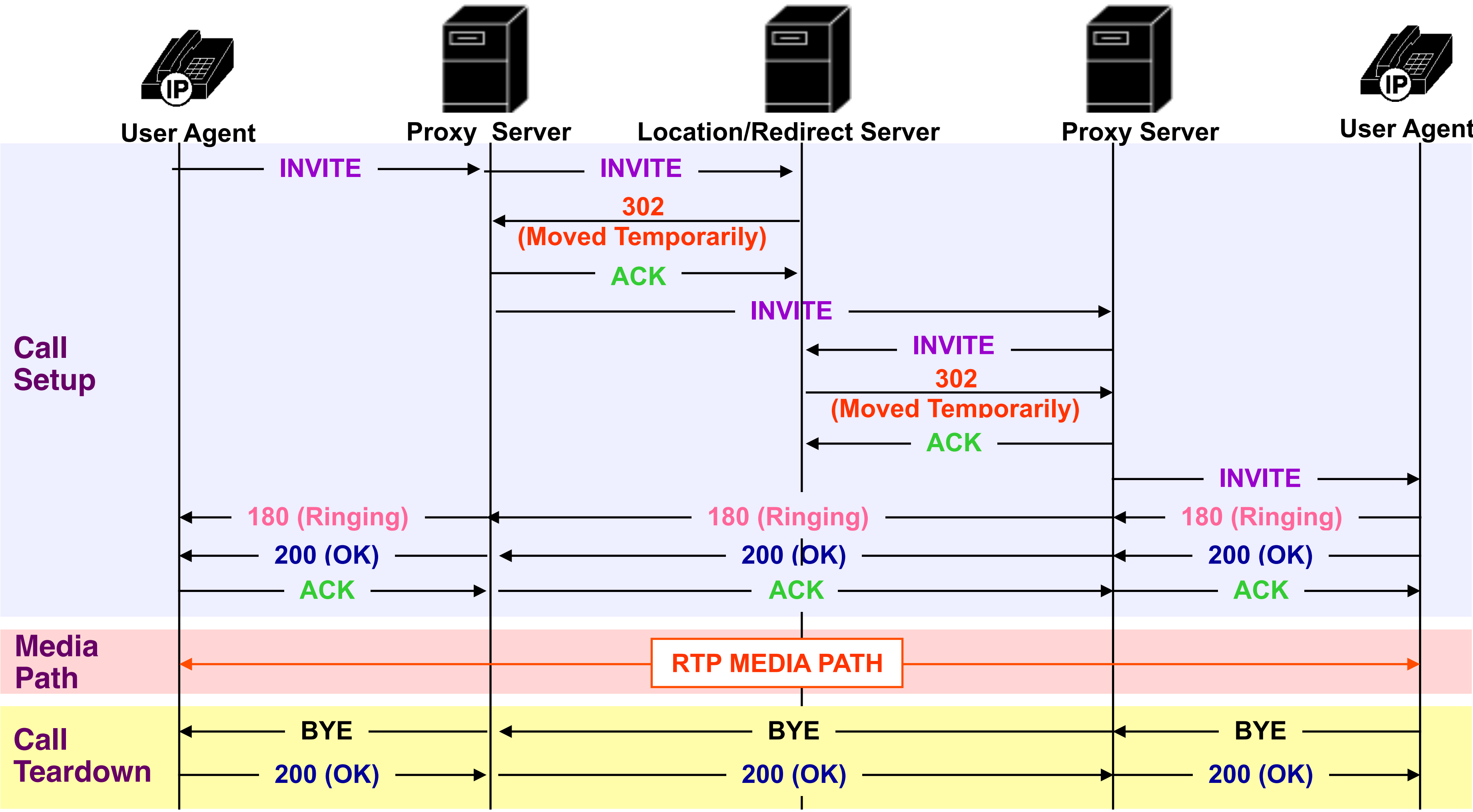
Registration

- Each time a user turns on the SIP user client (SIP IP Phone, PC, or other SIP device), the client registers with the proxy/ registration server.
- Registration can also occur when the SIP user client needs to inform the proxy/registration server of its location.
- The registration information is periodically refreshed and each user client must re-register with the proxy/registration server.
- Typically the proxy/registration server will forward this information to be saved in the location/redirect server.



SIP Messages:
REGISTER – Registers the address listed in the To header field.
200 – OK.

Simplified SIP Call Setup and Teardown



SIP – Design Framework

- SIP was designed for:
  - Integration with existing IETF protocols.
  - Scalability and simplicity.
  - Mobility.
  - Easy feature and service creation.

Integration with IETF Protocols

- Other IETF protocol standards can be used to build a SIP based application. SIP works with existing IETF protocols, for example:
  - RTP Real Time Protocol - to transport real time data and provide QOS feedback.
  - SDP Session Description Protocol – for describing multimedia sessions.
  - RSVP - to reserve network resources.
  - RTSP Real Time Streaming Protocol - for controlling delivery of streaming media.
  - SAP Session Advertisement Protocol - for advertising multimedia session via multicast.
  - MIME – Multipurpose Internet Mail Extension – describing content on the Internet.
  - COPS – Common Open Policy Service.
  - OSP – Open Settlement Protocol.

Scalability and Simplicity

- Scalability:
  The SIP architecture is scalable, flexible and distributed.
  - Functionality such as proxying, redirection, location, or registration can reside in different physical servers.
  - Distributed functionality allows new processes to be added without affecting other components.

- Simplicity:
  SIP is designed to be:
  - "Fast and simple in the core."
  - "Smarter with less volume at the edge."
  - Text based for easy implementation and debugging.

Feature Creation

- SIP can support these features and applications:
  - Basic call features (call waiting, call forwarding, call blocking etc.)
  - Unified messaging (the integration of different streams of communication - e-mail, SMS, Fax, voice, video, etc. - into a single unified message store, accessible from a variety of different devices.)
  - Call forking
  - Click to talk
  - Presence
  - Instant messaging
  - Find me / Follow me

Feature Creation (2)

- A SIP based system can support rapid feature and service creation
- For example, features and services can be created using:
  - Common Gateway Interface (CGI).
    - A standard for interfacing external applications with information servers, such as Web servers (or SIP servers).
      A CGI program is executed in real-time, so that it can output dynamic information.
  - Call Processing Language (CPL).
    - Jonathan Lennox, Xiaotao Wu, Henning Schulzrinne: RFC3880
    - Designed to be implementable on either network servers or user agents. Meant to be simple, extensible, easily edited by graphical clients, and independent of operating system or signalling protocol. Suitable for running on a server where users may not be allowed to execute arbitrary programs, as it has no variables, loops, or ability to run external programs.
    - Syntactically, CPL scripts are represented by XML documents.

References

- For more information on SIP:
- IETF
  - http://www.ietf.org/html.charters/sip-charter.html
- Henning Schulzrinne's SIP page
  - http://www.cs.columbia.edu/~hgs/sip/