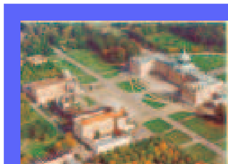
Medienengineering / Netzwerke

Lehrstuhl für Netzwerktechno-logien und multimediale Teledienste



Voice Over IP
(IETF-Protokolle)

Thomas Wolfram

mailto:thomas@wolfram.net

Universität Potsdam

Institut für Informatik

Lehrstuhl für
Netzwerktechno-
logien und
multimediale
Teledienste

Agenda

Universität
Potsdam

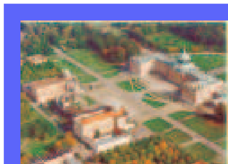
Institut für
Informatik

- Overview
- Motivation
- The IETF
- RTP
- RTCP
- SIP

Overview

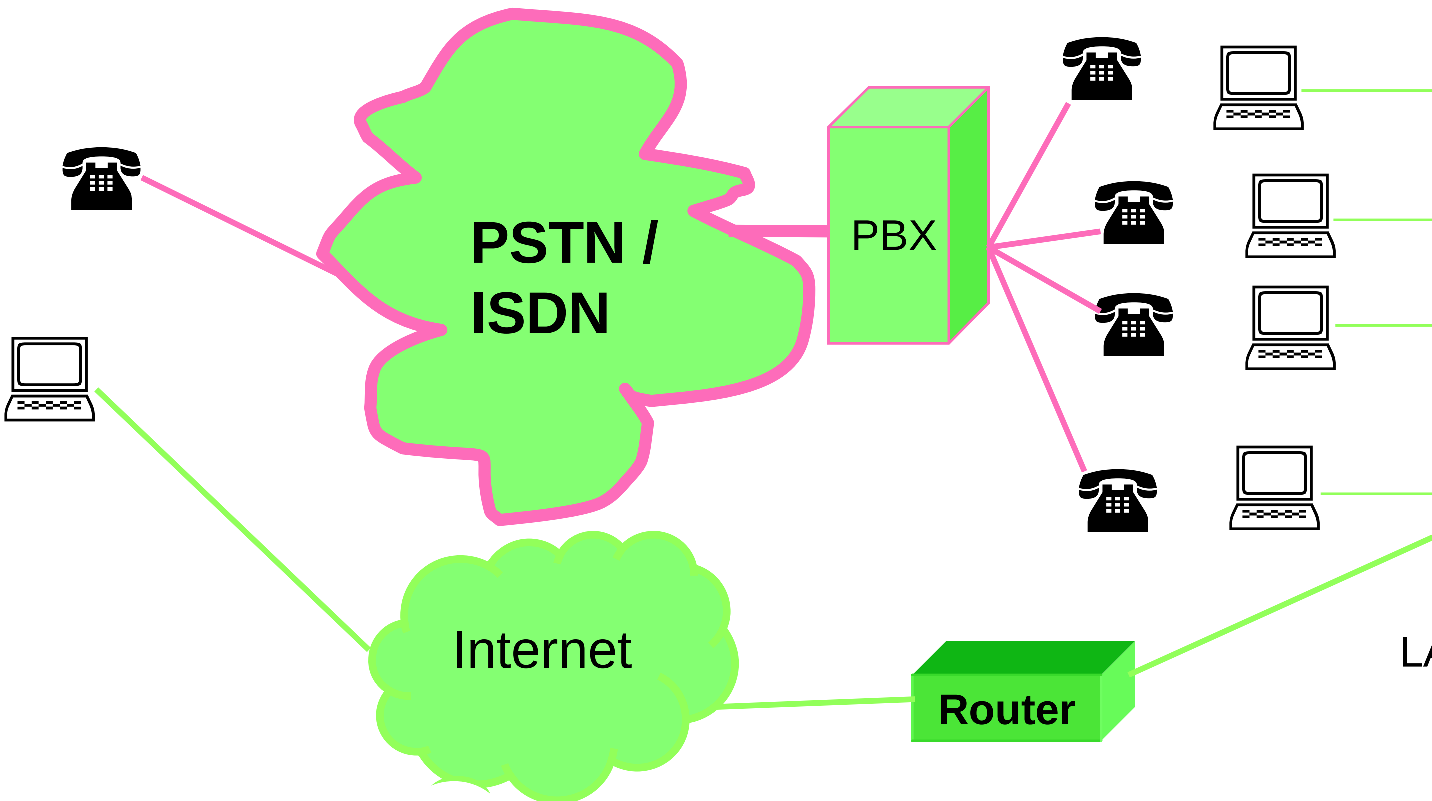
- *Definition:*
  Voice Over IP (VoIP) aka (Also Known As) **Internet Telephony** (IPT) is transporting of telephone calls over IP based networks
  - No matter whether traditional telephony devices, multimedia PCs or dedicated terminals take part in the calls
  - No matter whether the calls are entirely or only partially transmitted over the IP based network
  - Not only voice

Motivation

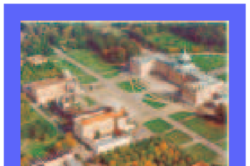


- Lower cost of ownership
  - Cost savings are the primary short-term reason to converge voice, data and video onto a single IP network
- Easy implementation of innovative services
  - Unified Messaging, Instant Messaging etc.
- In the future, Internet Telephony Service Providers (ITSP) may use a single infrastructure for providing both, Internet access and Internet telephony
  - Only data-oriented switches could be deployed for switching data as well as packetized voice
  - Multiplexing data and voice could also result in better bandwidth utilization than in today's over-engineered voice-or-nothing links

Motivation (2)

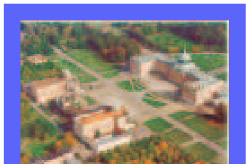


Motivation (3)



- VoIP users may also profit of its software-oriented nature:
  - Software solutions may be easily extended and integrated with other services and applications
  - E.g. whiteboarding, electronic calendar, or WWW
  - Deployment of new IP telephony services requires significantly lower investment in terms of time and money than in the traditional PSTN environment
- **But**:
  - Wide business deployment is still hindered by some problems
  - Particularly by higher delay and jitter
  - Many technical aspects of accounting, billing, charging, roaming etc. remain open yet

Technical Challenges



- ***In order to be successful in the market VoIP needs to provide the same perceived voice quality, reliability and scalability like legacy Public Switched Telephony Networks (PSTNs)!***

- Many challenges: Unlike legacy PSTN networks IP is historically best-effort only
  - Connectionless
  - Packet based
  - No delivery guarantees (packet loss)
  - No bandwidth guarantees
  - No timing guarantees (delay & jitter)

Technical Challenges (2)

Universität
Potsdam

Institut für
Informatik

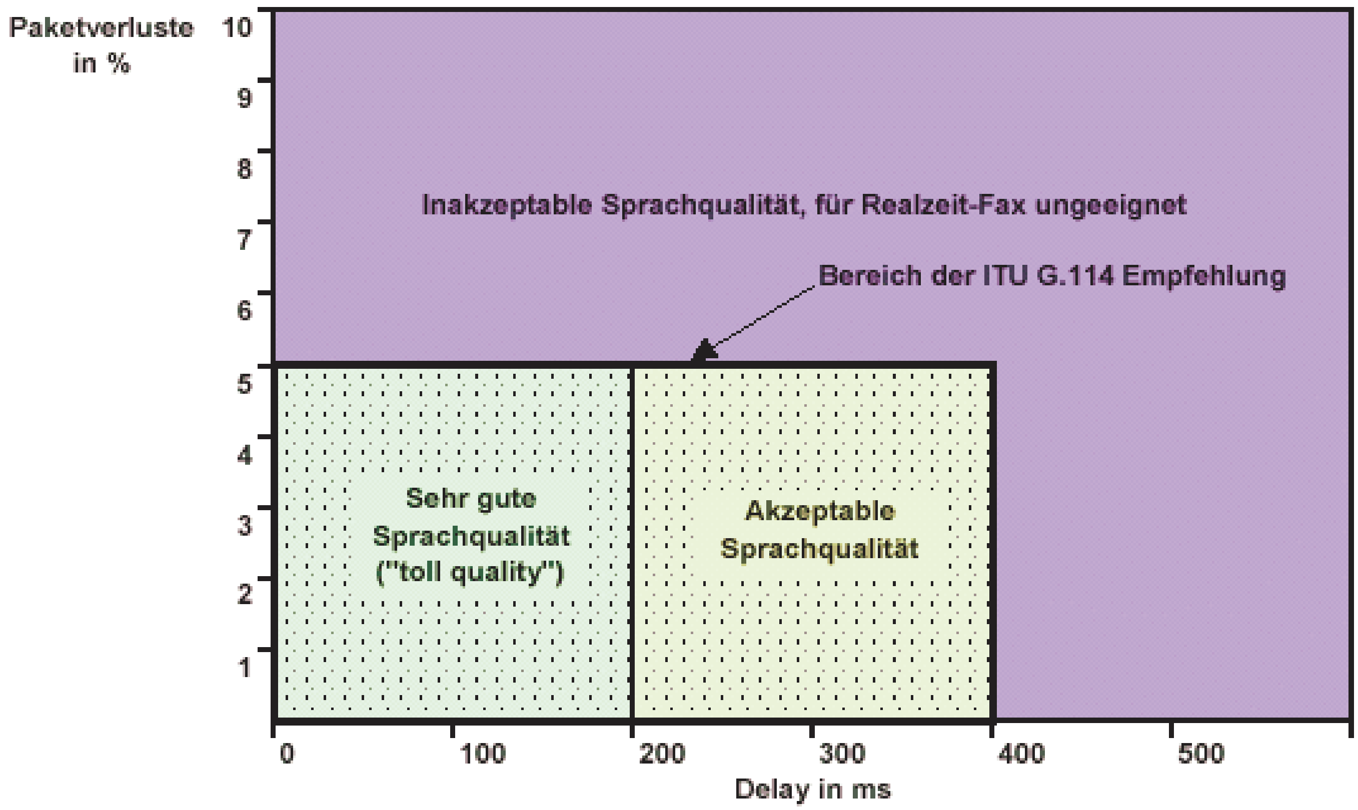
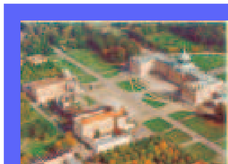


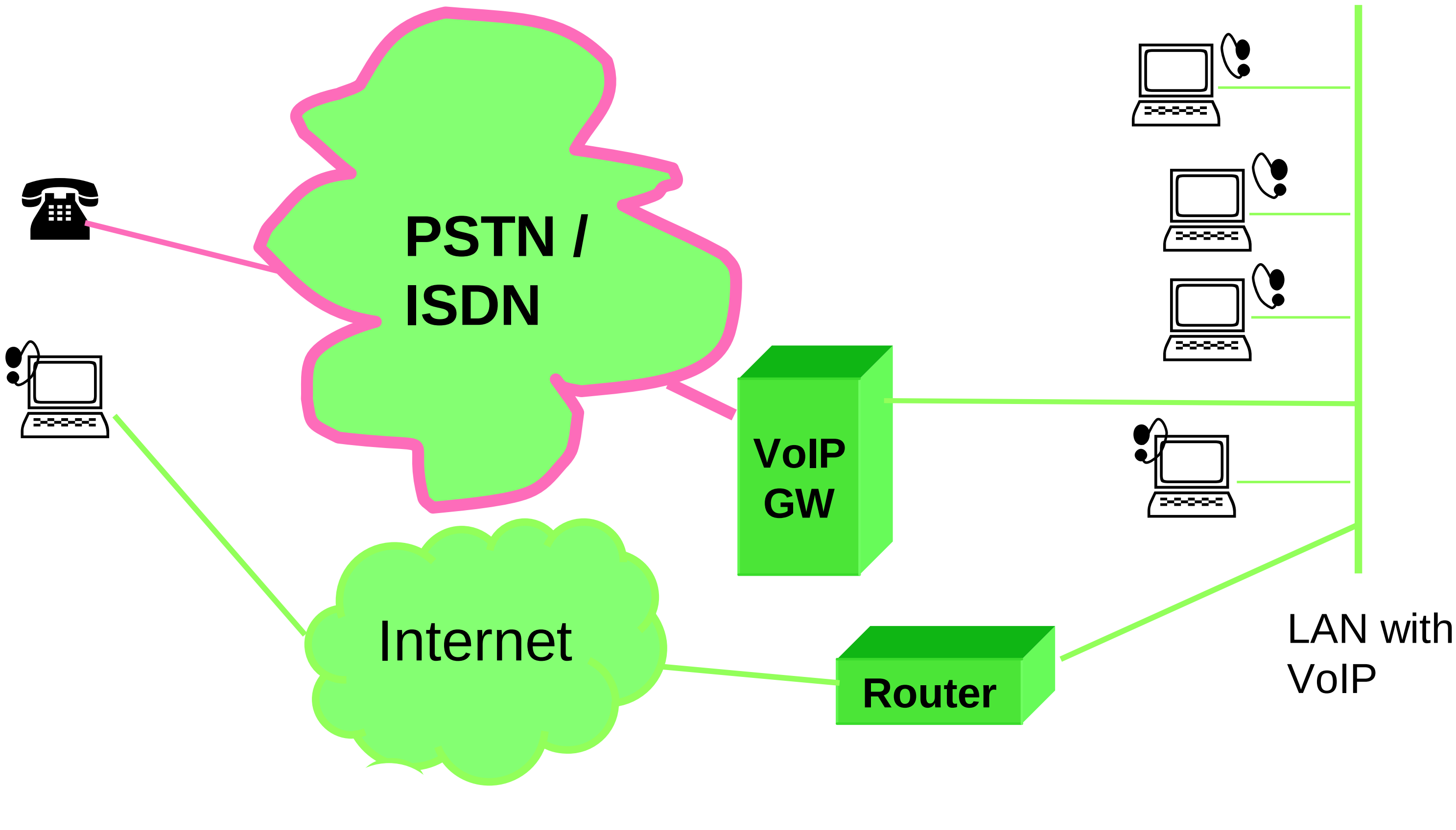
Abbildung: „Quality of Service“- Anforderungen

Long-term Visions

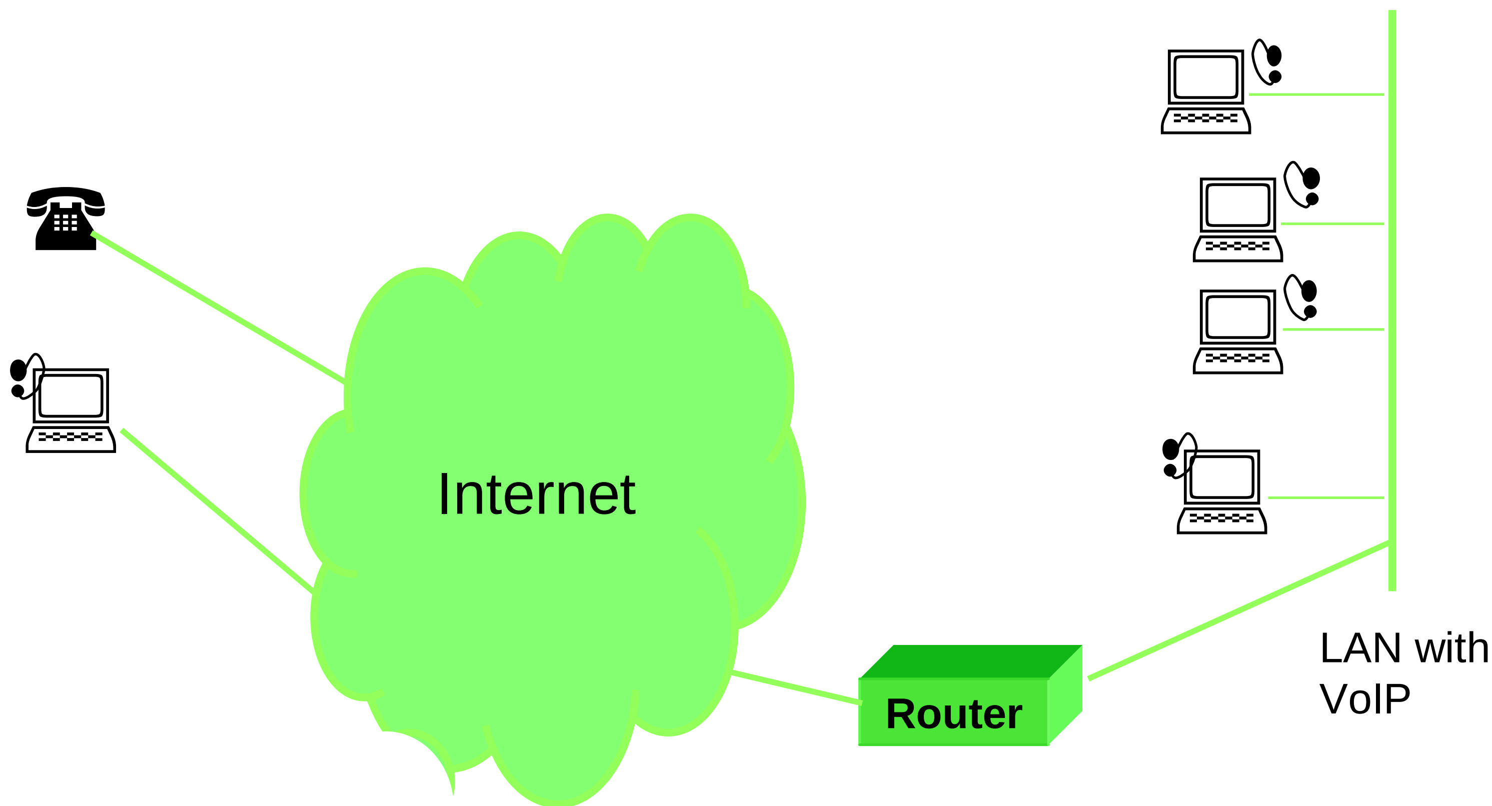


- A single global IP network for data, voice, video etc., all devices are IP-enabled including fixed and mobile phones
- Anytime, anywhere access to all message types (Unified Messaging)
- Real-time communication, you can reach the callee, no matter where the callee roams, no matter what IP device the callee is currently using
- People are identified by names or e-mail addresses, rather than by phone numbers
- Personal assistant filters, Presence management, Conferencing and collaboration, Directory services

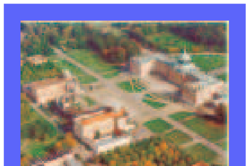
Long-term Visions (2)



Long-term Visions (3)

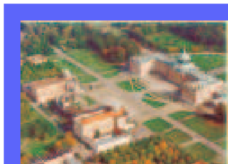


VoIP Standards



- Many different standards, e.g.:
  - ITU-T H.323 etc.
  - IETF SIP etc.
  - ITU-T und IETF MGCP
  - Vendor-specific Standards
- But: over the last couple of years, SIP and the related IETF protocols got widespread acceptance in the market
  - E.g. SIP, RTP, RTSP, RSVP etc. are supported by Windows XP out of the box

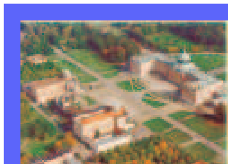
The Internet Engineering Task Force (IETF)



- Principal body engaged in the development of new Internet standard specifications
- International community of network designers, operators, vendors, and researchers
- Concerned with the evolution of the Internet architecture and the smooth operation of the Internet
- Open to any interested individual

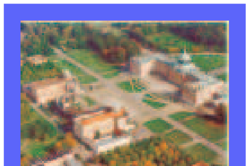
http://www.ietf.org/

Related IETF Working Groups



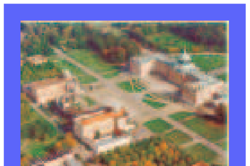
- **AVT** - Audio/Video Transport
- **SIP** - Session Initiation Protocol
- **SIPPING** - Session Initiation Proposal Investigation
- **SIMPLE** - SIP for Instant Messaging and Presence Leveraging Extensions
- **IPTEL** - IP Telephony; Naming (URI-based) and Routing for VoIP
- **MIDCOM** - Firewall/NAT Traversal
- **MMUSIC** - Multiparty Multimedia Session Control
- QoS Related: **DiffServ**, **IntServ**, **RSVP**
- PSTN legacy: SigTran, Megaco
- Interaction of PSTN and IP services: PINT, SPIRITS

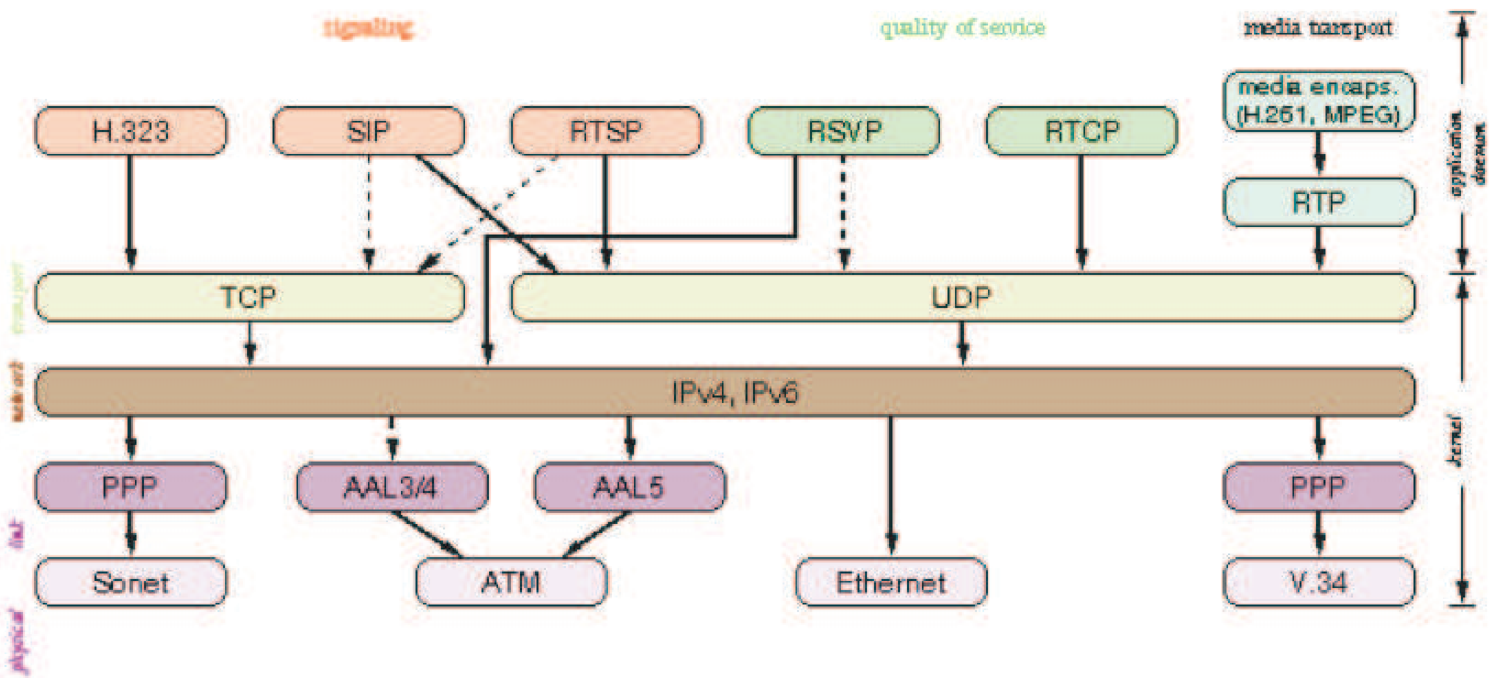
What Protocols Are Needed?



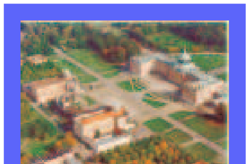
- Media Transport Protocols
  - Transmission of packetized audio/video
- Signaling protocol
  - To establish presence, locate users, set up, modify and tear down sessions
- Supporting Protocols
  - QoS, Gateway Location, Interdomain Authentication, Authorization, Accounting, address translation, etc.

VoIP: der „Protokoll-Zoo“



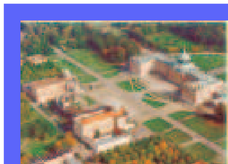


Evolution of IP-Services



- Initially
  - All users were treated equally: no privileges
- Eventually
  - Different types of usage need different treatment inside the network
- Evolving concepts (IntServ, DiffServ)
  - Priority by type of services
  - Priority by reservation
    - Per flow (RSVP)
    - Per aggregate of flows (DiffServ)

IP Service Classes



Universität Potsdam

Institut für Informatik

Best Effort
No Signaling

Prioritized
By Type of Service

Assured
By Type of Service

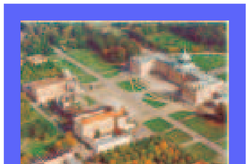
Assured
By Aggregate Reservation

Guaranteed
Per Flow Reservation

Pure packet-switching
Most scalable

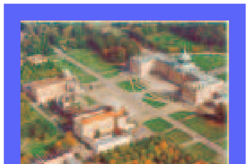
Circuit-switching
Least scalable

Real-Time Transport Protocol (RTP)



- IETF standard: RFC 3550
- Provides end-to-end delivery services for data with real-time characteristics, such as interactive audio and video
- Specifies a packet structure for packets carrying audio and video data
- RTP packet provides
  - payload type identification
  - packet sequence numbering
  - time stamping

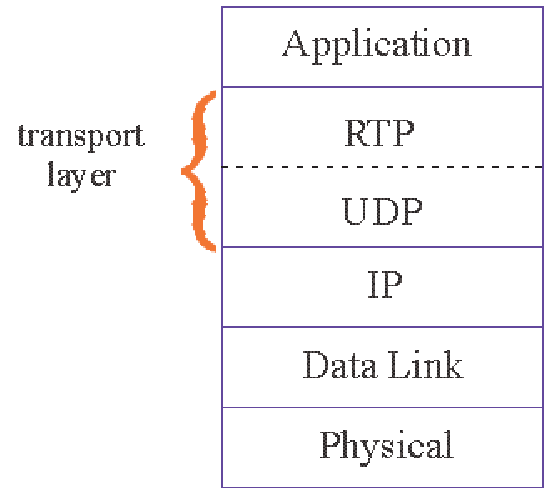
RTP (2)



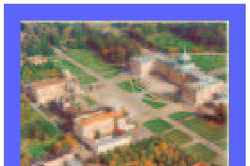
Universität Potsdam

Institut für Informatik

- RTP runs in the end systems
- RTP packets are encapsulated in UDP segments
- Provides a transport-layer interface ("OSI-Layer 4b")

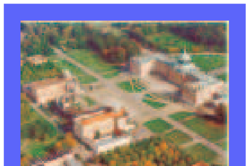


RTP (3)



- RTP does **not** provide
  - Any mechanism to ensure timely delivery of data
  - Any quality of service guarantees
- RTP encapsulation is only seen at the end systems
  - Not recognized by routers as a special kind of datagram
  - Routers providing best-effort service do not make any special effort to ensure that RTP packets arrive at the destination in a timely matter
- Allows for each voice or video device to have its own RTP stream
- Supports unicast as well as multicast applications

RTP Header



Payload Type	Sequence Number	Timestamp	Synchronization Source Identifer	Miscellaneous Fields

RTP Header

- **Payload Type (8 bits):** Indicates type of encoding currently being used. If sender changes encoding in middle of conference, sender
- Informs the receiver through this payload type field

Audio Payload Types:

- Payload type 0: PCM mu-law, 64 kbps
- Payload type 3: GSM, 13 kbps
- Payload type 7: LPC, 2.4 kbps
- Payload type 18: G.729, 8 kbps

Video Payload Types:

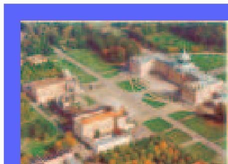
- Payload type 26: Motion JPEG
- Payload type 26: Motion JPEG
- Payload type 31: H.261
- Payload type 32: MPEG 1
- Payload type 33: MPEG 2

- **Sequence Number (16 bits):** Increments by one for each RTP packet sent, and may be used to detect packet loss and to restore packet sequence

Universität Potsdam

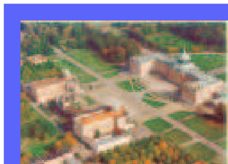
Institut für Informatik

RTP Header (2)



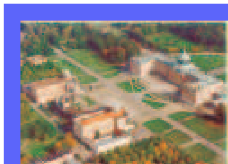
- **Timestamp field (32 bits):**
  - Reflects the sampling instant of the first byte in the RTP data packet
  - Clock frequency is dependent on the format of data carried as payload
    - E.g. if application generates chunks of 160 encoded samples, then timestamp increases by 160 for each RTP packet when source is active
  - Timestamp clock continues to increase at constant rate when source is inactive
- **SSRC field (32 bits):**
  - Identifies the source of the RTP stream
  - Each stream in a RTP session should have a distinct SSRC

RTP Example

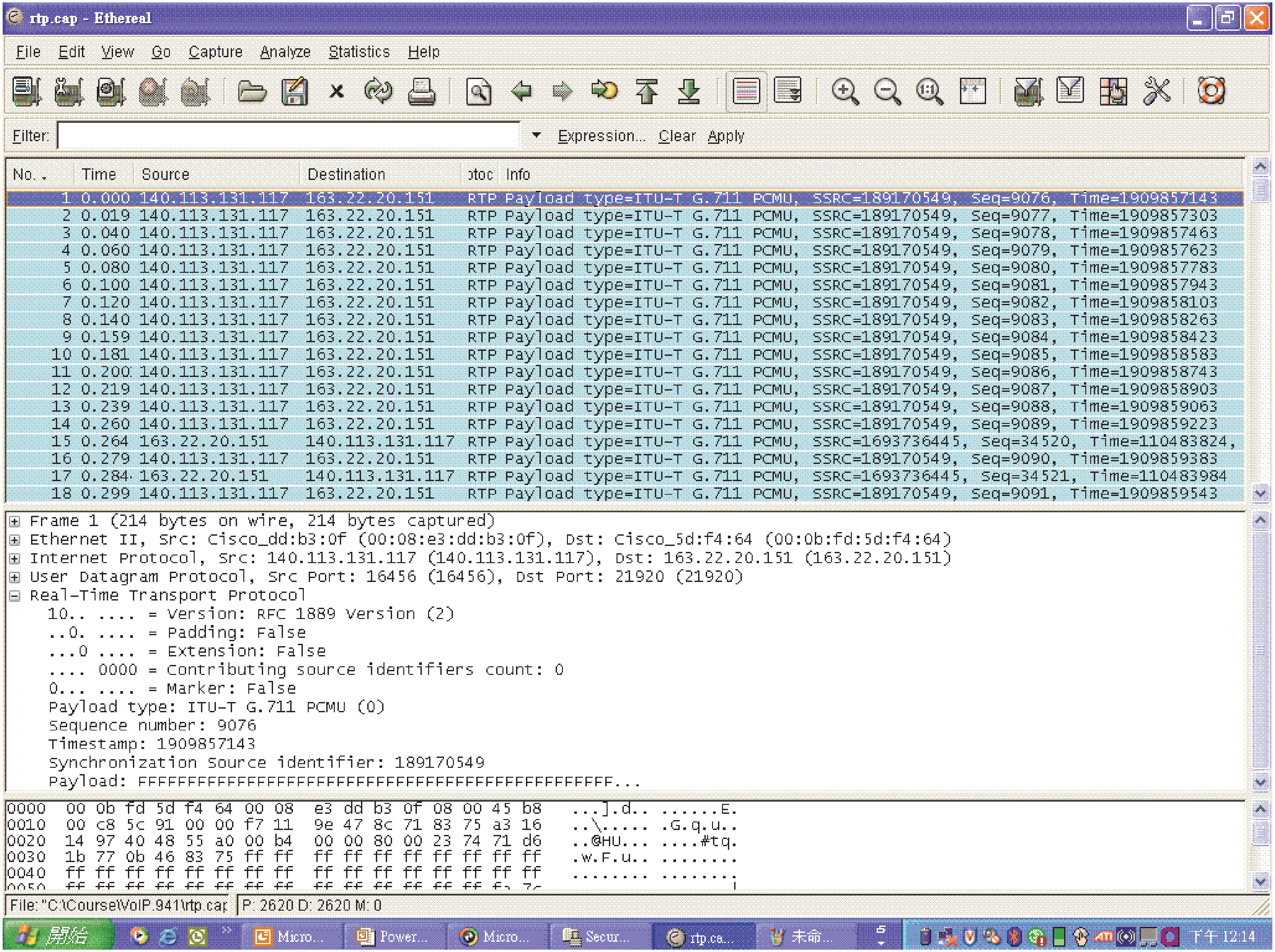


- When sending 64 kbps PCM-encoded voice over RTP:
  - Application collects the encoded data in chunks, e.g., every 20 msec = 160 bytes in a chunk
  - Audio chunk along with the RTP header forms the RTP packet, which is encapsulated into a UDP segment
  - RTP header indicates type of audio encoding in each packet (Sender can change encoding during a conference)
  - RTP header also contains sequence numbers and timestamps

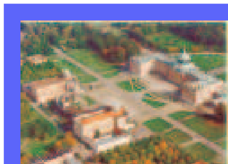
Homework #1



- Go to NCNU library and download G.114 spec of ITU-T. Write a brief summary about what you learn from it.
- Compare RFC 1889 and RFC 3550 and briefly describe their difference. (You need not write down everything. Just describe those that you feel to be important.)
- Send your homework to me via email:
  - To: Quincy.Wu@gmail.com
  - Subject: [VoIP Homework 1]
  - Describe your student number and name, and the answers to the above questions
  - Due: 11PM, March 8 (Wednesday)


rtp.cap - Ethereal
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply
No. Time Source Destination otoc Info
1 0.000 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9076, Time=1909857143
2 0.019 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9077, Time=1909857303
3 0.040 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9078, Time=1909857463
4 0.060 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9079, Time=1909857623
5 0.080 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9080, Time=1909857783
6 0.100 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9081, Time=1909857943
7 0.120 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9082, Time=1909858103
8 0.140 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9083, Time=1909858263
9 0.159 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9084, Time=1909858423
10 0.181 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9085, Time=1909858583
11 0.200 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9086, Time=1909858743
12 0.219 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9087, Time=1909858903
13 0.239 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9088, Time=1909859063
14 0.260 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9089, Time=1909859223
15 0.264 163.22.20.151 140.113.131.117 RTP Payload type=ITU-T G.711 PCMU, SSRC=1693736445, Seq=34520, Time=110483824,
16 0.279 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9090, Time=1909859383
17 0.284 163.22.20.151 140.113.131.117 RTP Payload type=ITU-T G.711 PCMU, SSRC=1693736445, Seq=34521, Time=110483984
18 0.299 140.113.131.117 163.22.20.151 RTP Payload type=ITU-T G.711 PCMU, SSRC=189170549, Seq=9091, Time=1909859543
Frame 1 (214 bytes on wire, 214 bytes captured)
Ethernet II, Src: Cisco_dd:b3:0f (00:08:e3:dd:b3:0f), Dst: Cisco_5d:f4:64 (00:0b:fd:5d:f4:64)
Internet Protocol, Src: 140.113.131.117 (140.113.131.117), Dst: 163.22.20.151 (163.22.20.151)
User Datagram Protocol, Src Port: 16456 (16456), Dst Port: 21920 (21920)
Real-Time Transport Protocol
10.. .... = Version: RFC 1889 Version (2)
..0. .... = Padding: False
...0 .... = Extension: False
.... 0000 = Contributing source identifiers count: 0
0... .... = Marker: False
Payload type: ITU-T G.711 PCMU (0)
Sequence number: 9076
Timestamp: 1909857143
Synchronization Source identifier: 189170549
Payload: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF...
0000 00 0b fd 5d f4 64 00 08 e3 dd b3 0f 08 00 45 b8 ...].d.. ......E.
0010 00 c8 5c 91 00 00 f7 11 9e 47 8c 71 83 75 a3 16 ..\..... .G.q.u..
0020 14 97 40 48 55 a0 00 b4 00 00 80 00 23 74 71 d6 ..@HU... ....#tq.
0030 1b 77 0b 46 83 75 ff ff ff ff ff ff ff ff ff ff .w.F.u.. ....
0040 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ....
File: "C:\Course\VoIP.941\rtp.cap P: 2620 D: 2620 M: 0
開始
Micro...
Power...
Micro...
Secur...
rtp.ca...
未命...
下午 12:14

Real-Time Control Protocol (RTCP)



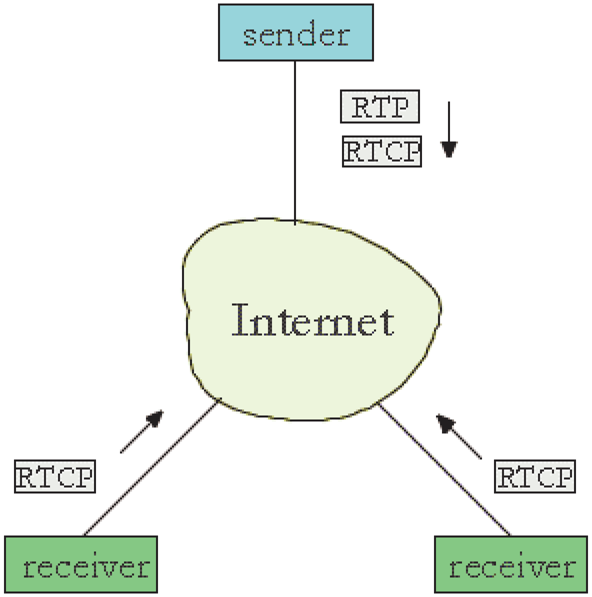
- Works in conjunction with RTP
- Each participant in RTP session periodically transmits RTCP control packets to all other participants
- Each RTCP packet contains sender and/or receiver reports
  - Report statistics useful to application
- Statistics include number of packets sent, number of packets lost, interarrival jitter, etc.
- Feedback can be used to control performance
  - Sender may modify its transmissions based on feedback (incl. Bandwidth scaling)

Universität Potsdam

Institut für Informatik

RTCP (2)

- For an RTP session there is typically a single multicast address; all RTP and RTCP packets belonging to the session use it
- RTP and RTCP packets are distinguished from each other through the use of distinct port numbers (RTCP port = RTP port + 1)
- To limit traffic, each participant reduces its RTCP traffic as the number of conference participants increases



RTCP Packet Types

Receiver reception report fields:

- SSRC of each stream received
- Fraction of packets lost
- Last sequence number in the stream
- Interarrival time between packets

Sender transmission report fields:

- SSRC of the RTP stream
- Timespamp and “wall clock” time of most recent packet
- Number of packets sent
- Number of bytes sent

- Each RTCP packet coming from sender or receiver is sent to the multicast tree that connects all participants

Synchronization of Streams

- Consider videoconferencing app for which each sender generates one RTP stream for video and one for audio
- Timestamps in RTP packets tied to the video and audio sampling clocks
  - Not tied to the "real-time" clock
- Receivers can use this association to synchronize the playout of the audio and video streams
- Each RTCP sender-report packet contains (for the most recently generated packet in the associated RTP stream):
  - Timestamp of the RTP packet
  - Wall-clock time for when packet was created

Session Initiation Protocol (SIP)

- Application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants
- Sessions include Internet telephone calls, multimedia distribution, and multimedia conferences
- RFC 3261
  - http://www.ietf.org/html.charters/sip-charter.html
- SIP allows participants to agree on a set of compatible media types
- Provides user location, Authentication and Authorization services
- SIP runs on top of several different transport protocols

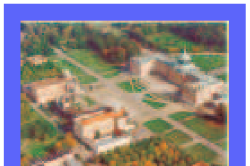
Universität Potsdam

Institut für Informatik

SIP History

- Work began in 1995 in the IETF MMUSIC WG
  - 02/1996: draft-ietf-mmusic-sip-00: 15 ASCII pages, one request type
- 03/1999: RFC 2543, 153 ASCII pages, 6 methods
- 11/1999: SIP WG formed
- 2001: SIP implementations widely available
  - Windows XP ships with built-in SIP support
- 06/2002: RFC 3261, 269 ASCII pages, 6 methods
- Compared with
  - SMTP (RFC 821) has 68 ASCII pages, 14 methods
  - POP3 (RFC 1939) has 23 ASCII pages, 13 methods
  - IMAP4 (RFC 2060) has 82 ASCII pages, 25 methods
  - BGP4 (RFC 4271) has 104 ASCII pages
  - HTTP 1.1 (RFC 2616) has 176 ASCII pages

SIP Services

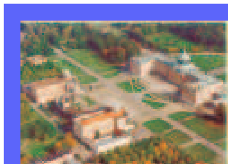


Setting up a call

- Provides mechanisms:
  - for caller to let callee know she wants to establish a call
  - so that caller and callee can agree on media type and encoding
  - Provides mechanisms to end call

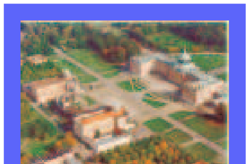
- Determine current IP address of callee
  - Maps mnemonic identifier to current IP address
- Call management
  - Add new media streams during call
  - Change encoding during call
  - Invite others
  - Transfer and hold calls

SIP Characteristics



- SIP borrows much of the syntax and semantics from HTTP
- SIP messages are ASCII-readable, just like HTTP messages
- SIP requires all messages to be acknowledged → runs over UDP or TCP

SIP Messages – Methods and Responses

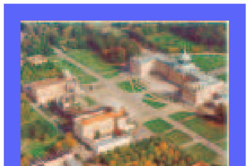


- SIP Methods:
  - INVITE: Initiates a call by inviting user to participate in session.
  - ACK: Confirms that the client has received a final response to an INVITE request.
  - BYE: Indicates termination of the call.
  - CANCEL: Cancels a pending request.
  - REGISTER: Registers the user agent.
  - OPTIONS: Used to query the capabilities of a server.

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

→ **SIP components communicate by exchanging SIP messages**

SIP Addressing

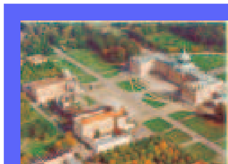


- The SIP address is identified by a SIP URL, in the format: user@host
- Examples of SIP URLs:
  - sip:tw@iptel.org
  - sip:tw@192.168.10.1
  - sip:14083831088@fokus.gmd.de
  - sip:dbaron@mit.edu
  - sip:csie@ncnu.edu.tw (?)

Universität
Potsdam

Institut für
Informatik

Example of SIP message

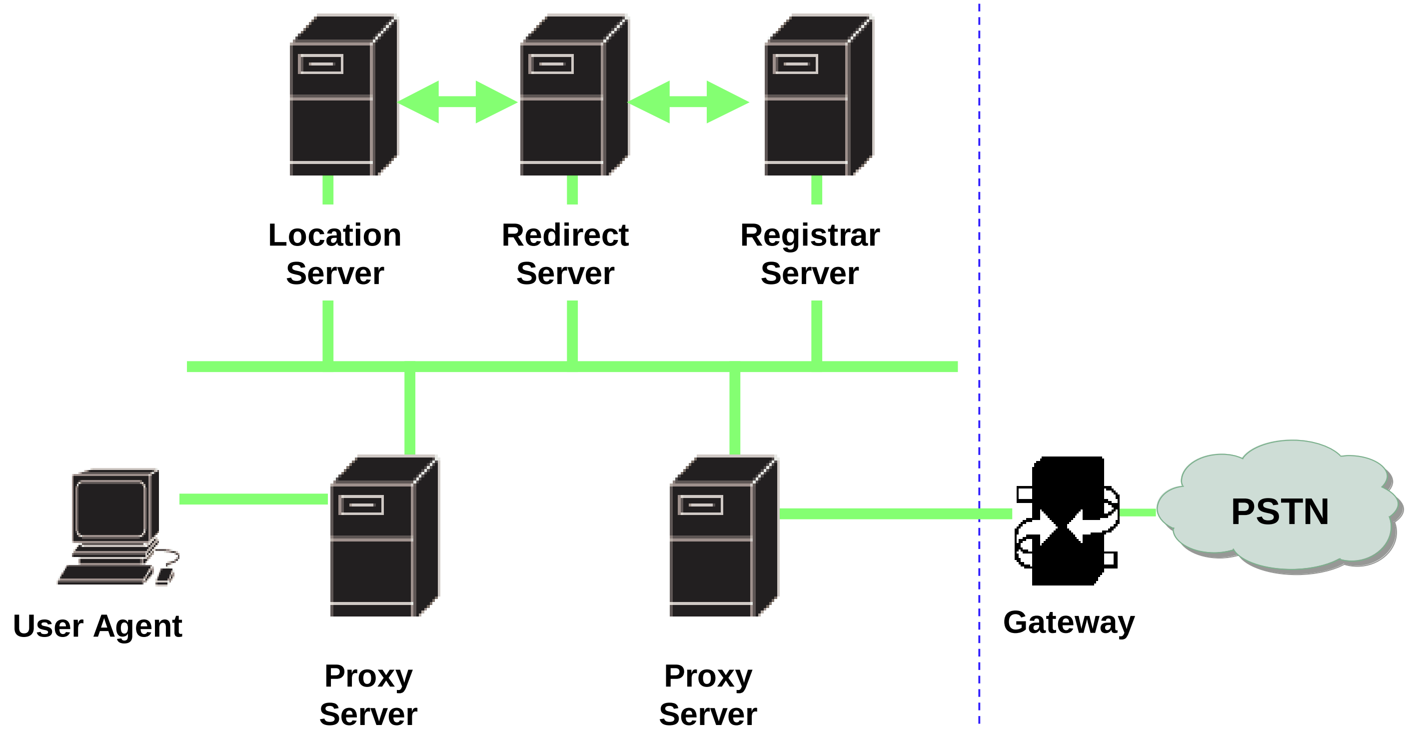


```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

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

- HTTP message syntax
- Header includes SIP version
- Call-ID is unique

SIP Distributed Architecture



User Agents

Universität Potsdam

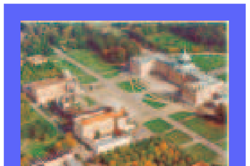
Institut für Informatik

- 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

Process for Establishing Communication

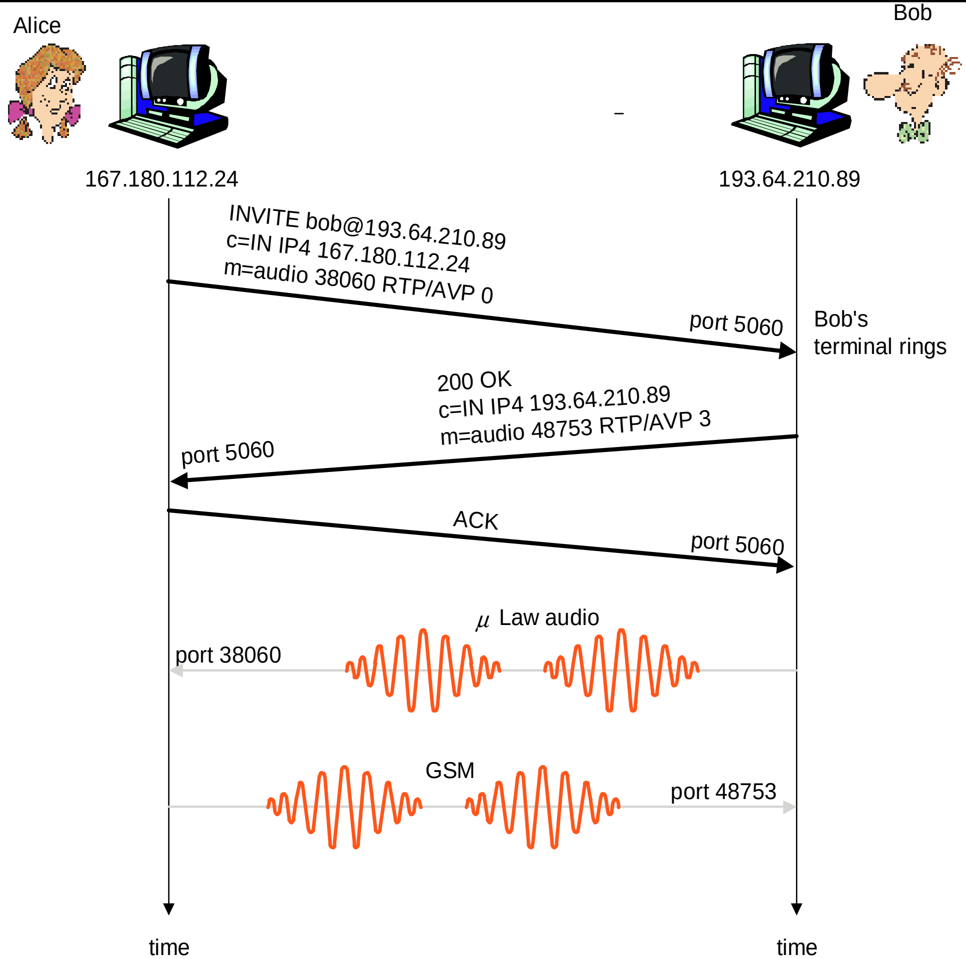
- Establishing communication using SIP usually occurs in six steps:
  1. Registering, initiating and locating the user
  2. Determine the media to use – involves delivering a description of the session that the user is invited to
  3. Determine the willingness of the called party to communicate – the called party must send a response message to indicate willingness to communicate – accept or reject
  4. Call setup
  5. Call modification or handling – example, call transfer (optional)
  6. Call termination

Setting up a call to a known IP address



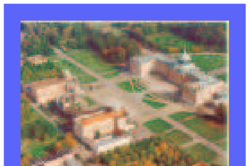
Universität Potsdam

Institut für Informatik



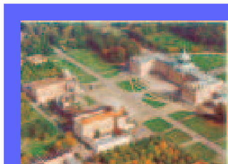
- Alice's SIP invite message indicates her port number & IP address. Indicates encoding that Alice prefers to receive (PCM μ-law)
- Bob's 200 OK message indicates his port number, IP address & preferred encoding (GSM)
- SIP messages can be sent over TCP or UDP; here sent over RTP/UDP
- Default SIP port number is 5060

Setting up a call (more)



- Codec negotiation:
  - Suppose Bob doesn't have PCM μ-law encoder
  - Bob will instead reply with a response 488 Not Acceptable Here and list encoders he can use
  - Alice can then send a new INVITE message, advertising an appropriate encoder
- Rejecting the call
  - Bob's device can reject with responses
    - 486 Busy Here
    - 410 Gone
    - 402 payment required
    - 403 Forbidden
    - 404 Not Found

Name translation and user location



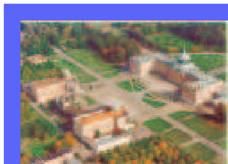
- Caller wants to call callee, but only has callee’s name or e-mail address.
- Need to get IP address of callee’s current host:
  - user moves around
  - DHCP protocol
  - user has different IP devices (PC, PDA, car device)
- Result can be based on:
  - Time of day (work, home)
  - Caller (don’t want 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**

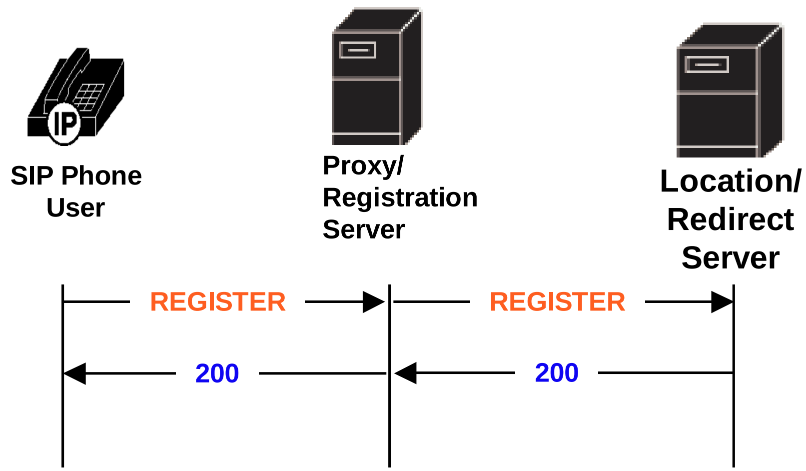
Registrar Server

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

Registration

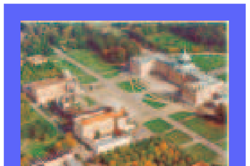


- Each time a user turns on the SIP user client (SIP IP Phone, PC), the client registers with the proxy/registration server
- Registration can also occur when the SIP user client needs to inform the proxy/registration server of its location
- The registration information is periodically refreshed and 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.

SIP Registrar



Universität
Potsdam

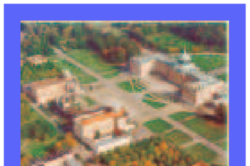
Institut für
Informatik

- When Bob starts SIP client, client sends SIP REGISTER message (with its current IP address) to Bob's registrar server

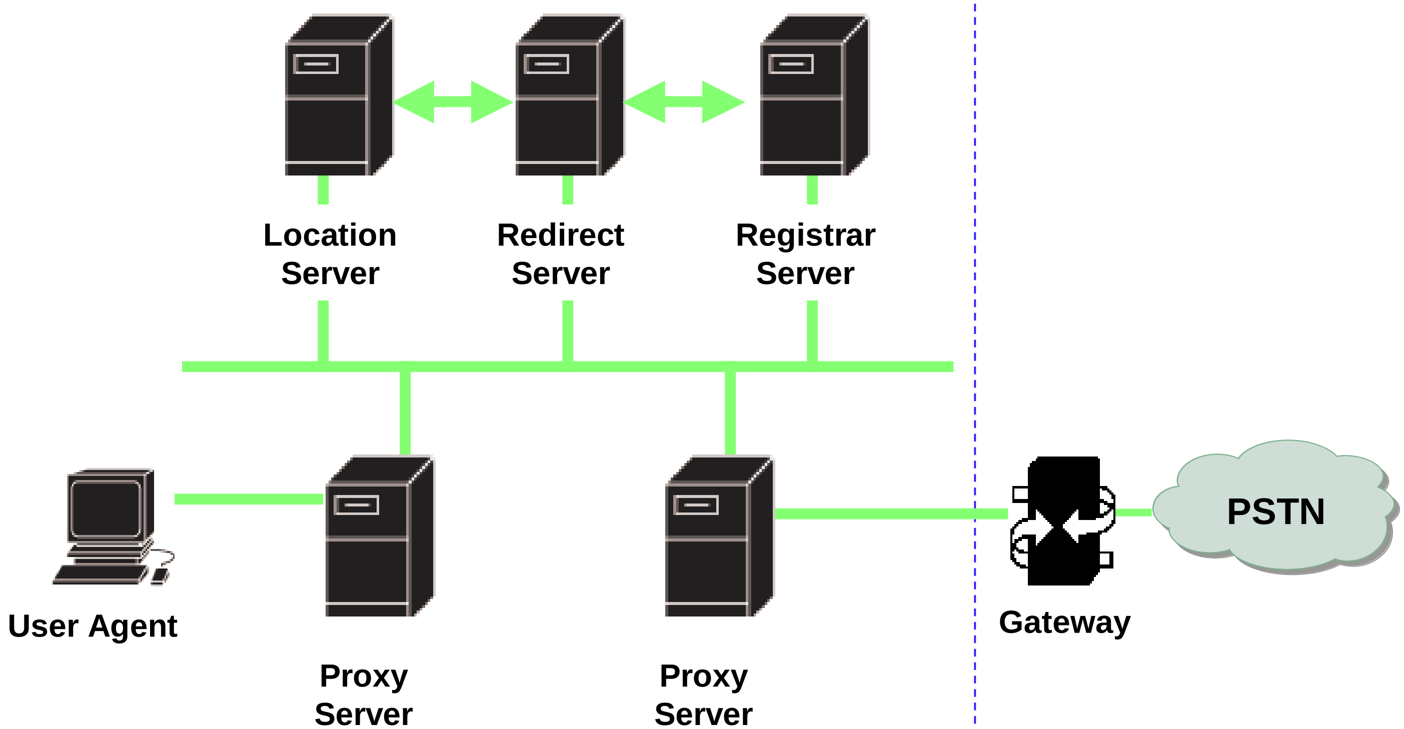
Register Message:

```
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 Distributed Architecture



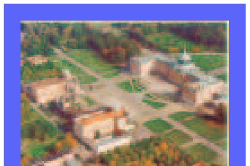
SIP Components



Proxy Server

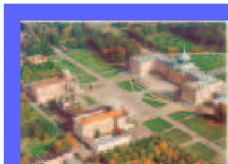
- An intermediary program that acts as both a server and a client to make requests on behalf of other clients
- Requests are serviced internally or by passing them on, possibly after translation, to other servers
- Interprets, rewrites or translates a request message before forwarding it

Proxy Server (2)



- Alice send's invite message to her proxy server
  - Destination 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, in reverse order
- Proxy returns SIP response message to Alice
  - With Bob's IP address in the Contact header field

200 OK Response



SIP/2.0 200 OK

Via: SIP/2.0/UDP server10.biloxi.com
;branch=z9hG4bKnashds8;received=192.0.2.3

Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2

Via: SIP/2.0/UDP pc33.atlanta.com
;branch=z9hG4bK776asdhds ;received=192.0.2.1

To: Bob <sip:bob@biloxi.com>;tag=a6c85cf

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710@pc33.atlanta.com

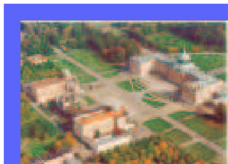
CSeq: 314159 INVITE

Contact: <sip:bob@192.0.2.4>

Content-Type: application/sdp

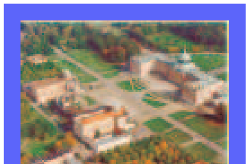
Content-Length: 131

Redirect Server



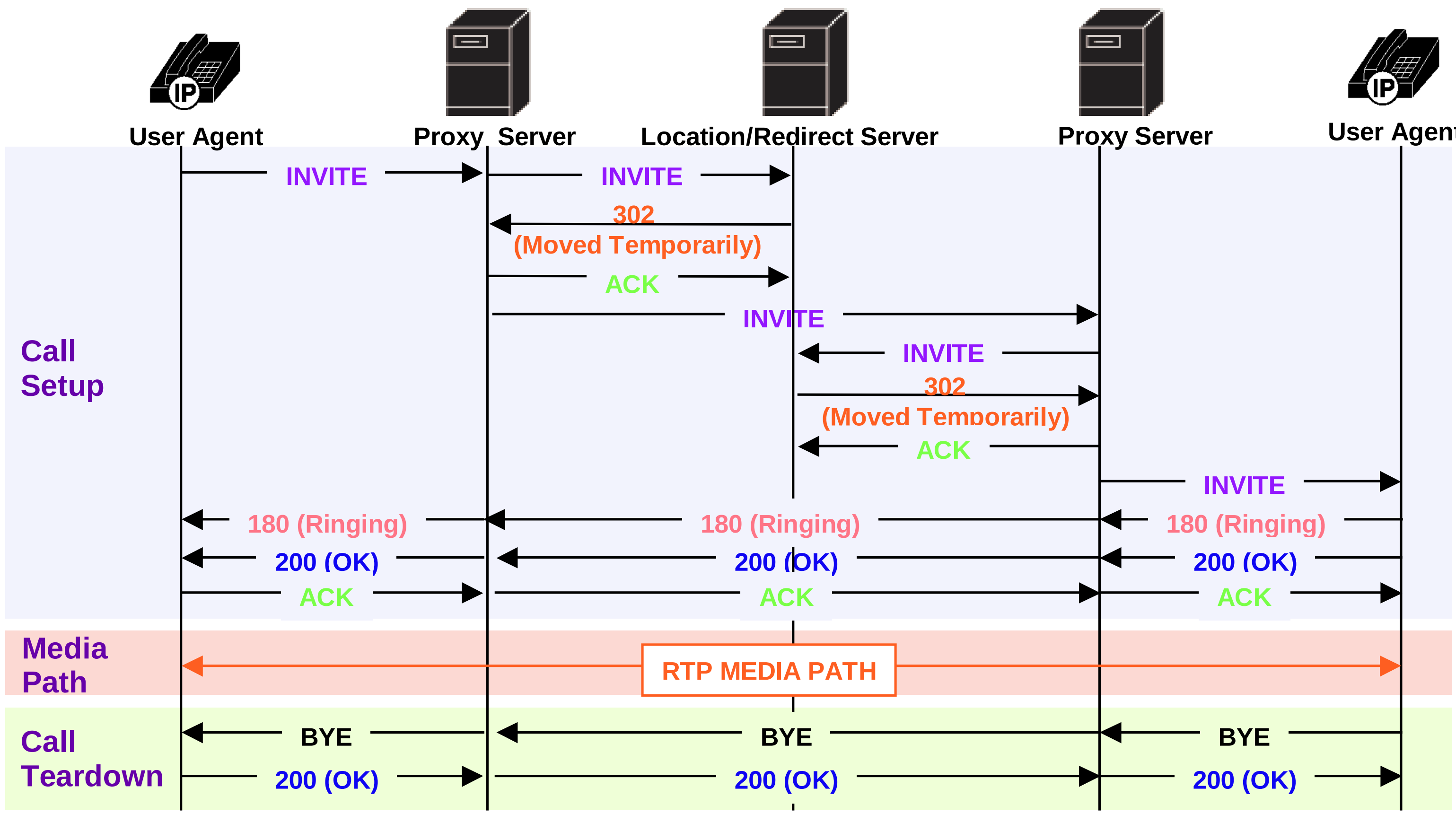
- A server that accepts a SIP request, maps the address into zero or more new addresses and returns these addresses to the client
- Unlike a proxy server, the redirect server does not initiate its own SIP request
- Unlike a user agent server, the redirect server does not accept or terminate calls

Location Server

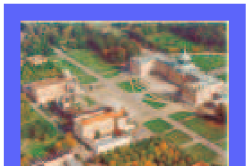


- A location server is used by a SIP redirect or proxy server to obtain information about a called party’s possible location(s)

Simplified SIP Call Setup and Teardown

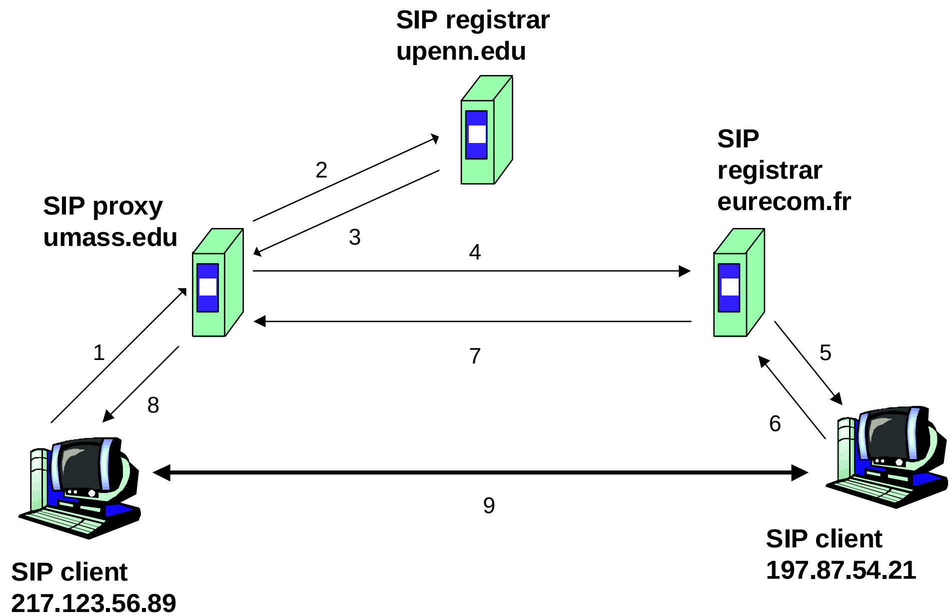


Example



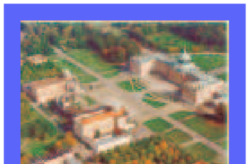
Caller jim@umass.edu places a call to keith@upenn.edu

(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



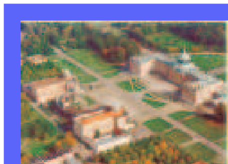
(4) umass proxy sends INVITE to eurecom registrar. (5) eurecom regristrar forwards INVITE to 197.87.54.21, which is running keith's SIP client. (6-8) SIP response sent back (9) media sent directly between clients.

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 (1)

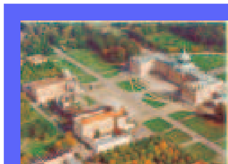


Universität Potsdam

Institut für Informatik

- Other IETF protocol standards can be used to build a SIP based application
- SIP can work with existing IETF protocols, for example:
  - RSVP - to reserve network resources
  - RTP - to transport real time data and provide QOS feedback.
  - RTSP (Real Time Streaming Protocol) - for controlling delivery of streaming media
  - SAP (Session Advertisement Protocol) - for advertising multimedia session via multicast

Integration with IETF Protocols (2)

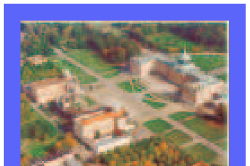


- SDP (Session Description Protocol) – for describing multimedia sessions
- MIME (Multipurpose Internet Mail Extension) – defacto standard for describing content on the Internet
- HTTP (Hypertext Transfer Protocol) - HTTP is the standard protocol used for serving web pages over the Internet
- COPS (Common Open Policy Service)
- OSP (Open Settlement Protocol)

Scalability

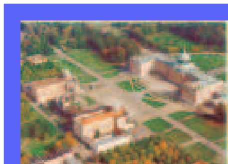
- The SIP architecture is scalable, flexible and distributed
  – Functionality such as proxying, redirection, location, or registration can reside in different physical servers
  – Distributed functionality allows new processes to be added without affecting other components

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

Mobility

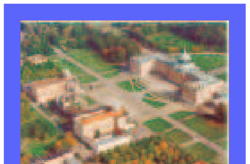


- SIP supports user mobility by proxying and redirecting requests to a user's current location
- The user can be using a PC at work, PC at home, wireless phone, IP phone, or regular phone
- The user must register their current location.
- The proxy server will forward calls to the user's current location
- Example mobility applications include presence and call forking

Universität
Potsdam

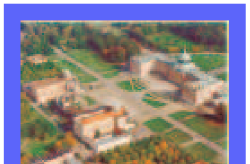
Institut für
Informatik

Feature Creation



- A SIP based system can support rapid feature and service creations
- For example, features and services can be created using:
  - Common Gateway Interface (CGI)
  - Call Processing Language (CPL)
  - VoiceXML

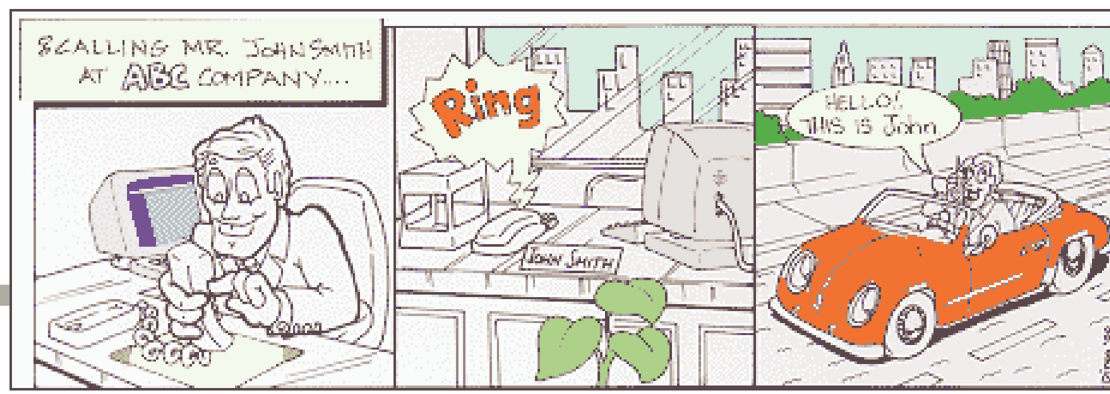
Feature Creation (2)



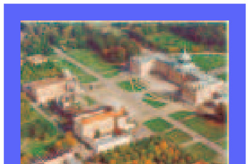
Universität Potsdam

Institut für Informatik

- SIP can support these features and applications:
  - Basic call features (call waiting, call forwarding, call blocking etc.)
  - Unified messaging
  - Call forking
  - Click to talk
  - Presence
  - Instant messaging
  - Find me / Follow me



Reference



Universität
Potsdam

Institut für
Informatik

- http://www.iptel.org
- http://www.cs.columbia.edu/sip/
- http://www.ietf.org/html.charters/sip-charter.html
- http://www.sipcenter.com/
- http://www.sipforum.com/
- http://www.voipwatch.com