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

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

Technische Universität München

Signalling - Introduction

Design principles:

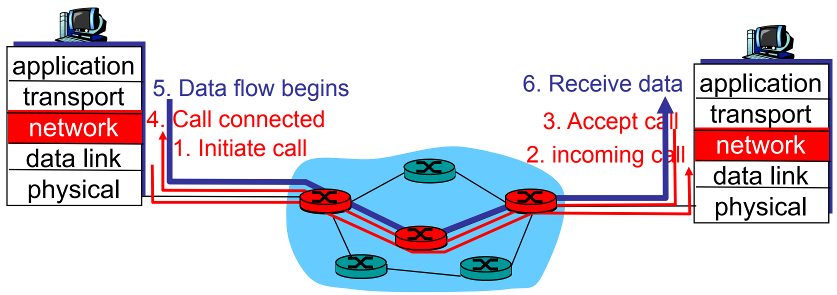
- network virtualization: overlays
- separation of data, control
- hard state versus soft state

Goals:

- identify, study common architectural components, protocol mechanisms
- what approaches do we find in network architectures?
- *synthesis:* big picture

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

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

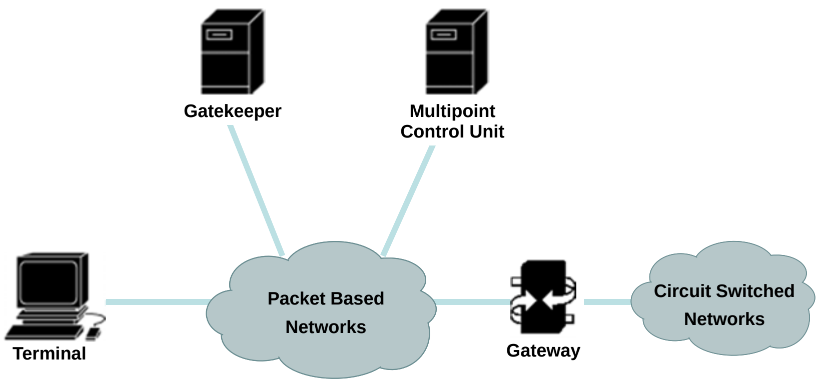
H.323

Technische Universität München

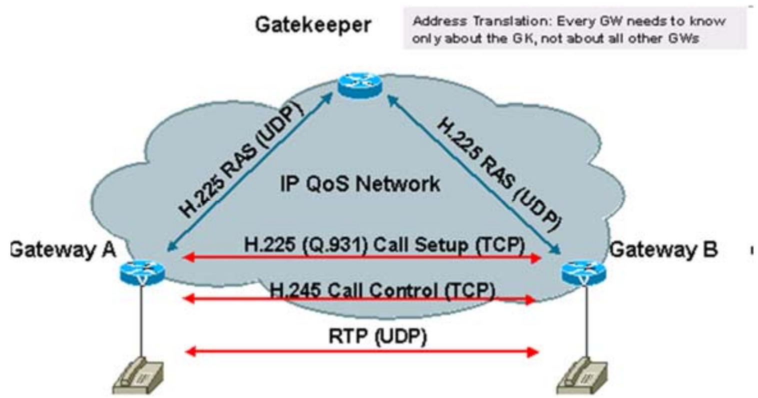
What is H.323?

- ITU-T Recommendation H.323 Version 4
  Describes terminals and other entities that provide multimedia communications services over Packet Based Networks (PBN) which may not provide a guaranteed Quality of Service. H.323 entities may provide real-time audio, video and/or data communications.
- H.323 framework defines:
  - Call establishment and teardown.
  - Audio visual or multimedia conferencing.

H.323 Components



H.225 and H.245



H.323 Terminals

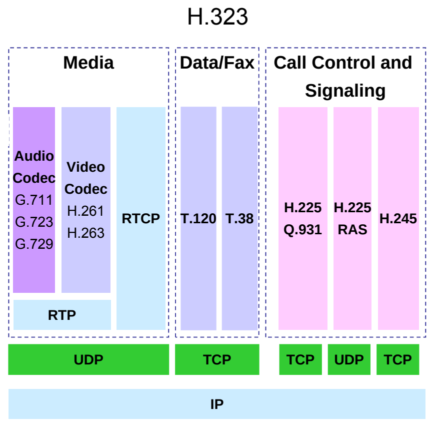
- H.323 terminals are client endpoints that must support:
  - H.225 call control signaling.
    - Call-control messages to gatekeeper (or directly to terminal), transport via TCP, port 1720
    - RAS: Registration, Admission, Status messages for address resolution and admission control
  - H.245 control channel signaling.
    - End-to-end messages between H.323 terminals.
    - Initiating and closing logical channels
    - Exchange of information on transmission capacity
  - RTP/RTCP protocols for media packets.
  - Audio codecs.
  - Video codecs support is optional.

H.323 is an "Umbrella" Specification

Media
- H.261 and H.263 – Video codecs.
- G.711, G.723, G.729 – Audio codecs.
- RTP/RTCP – Media transport and control protocol.

Data/Fax
- T.120 – Data conferencing.
- T.38 – Fax.

Call Control and Signaling
- H.245 - Capabilities advertisement, media channel establishment, and conference control.
- H.225
- Q.931 - call signaling and call setup.
- RAS – registration, admission control, status messages with a gatekeeper.



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

SIP

Credits in addition to
Jim Kurose and Keith Ross:

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

Technische Universität München

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

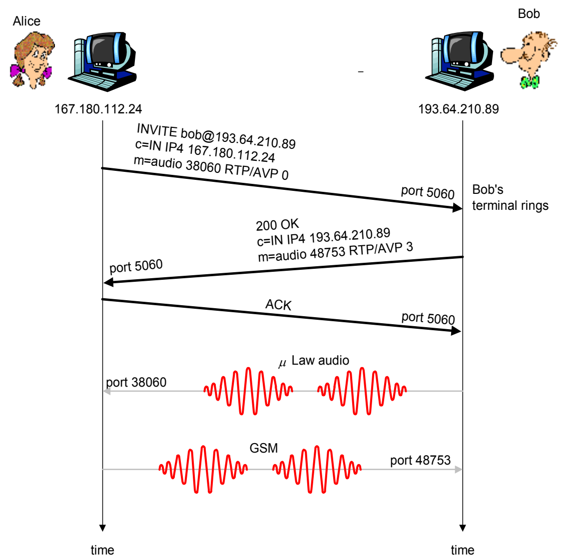
SIP

- 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 for TCP and UDP, and 5061 for TSL-over-TCP.

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
```

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

- 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)
- Session Description Protocol (SDP), RFC 4566 is a format for describing streaming media initialization parameters.

c=connection information
m= (media name and transport address)

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 server

- 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:
- Used by a UA to indicate its current IP address and the URLs for which it would like to receive calls.

```
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.
  - each proxy includes its address as VIA address
- 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

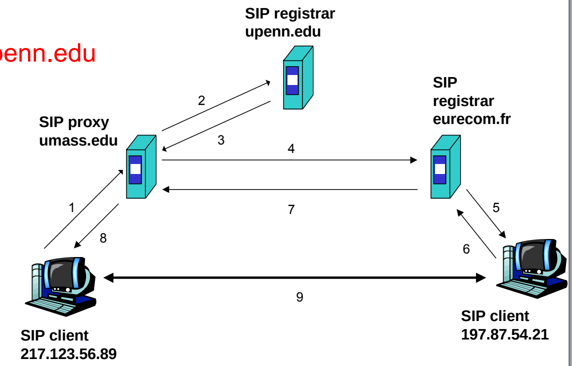
Example

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

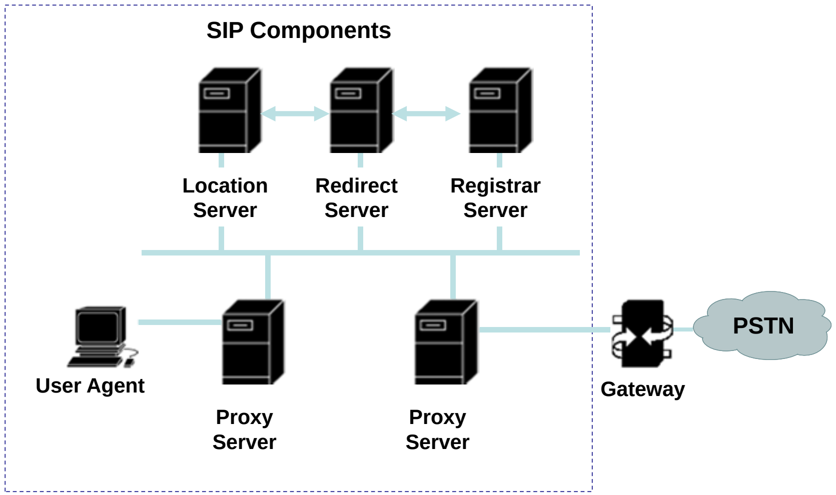
(9) media sent directly between clients.

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:

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.
- INFO – Used to carry out-of-band information, such as DTMF (Dual-tone multi-frequency) digits.

SIP Responses:

- 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:

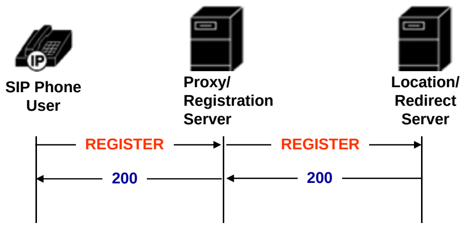
```
-----------------------------------------------------------------
                          SIP Header
-----------------------------------------------------------------
INVITE sip:5120@192.168.36.180 SIP/2.0
Via: SIP/2.0/UDP 192.168.6.21:5060
From: sip:5121@192.168.6.21
To: <sip:5120@192.168.36.180>
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

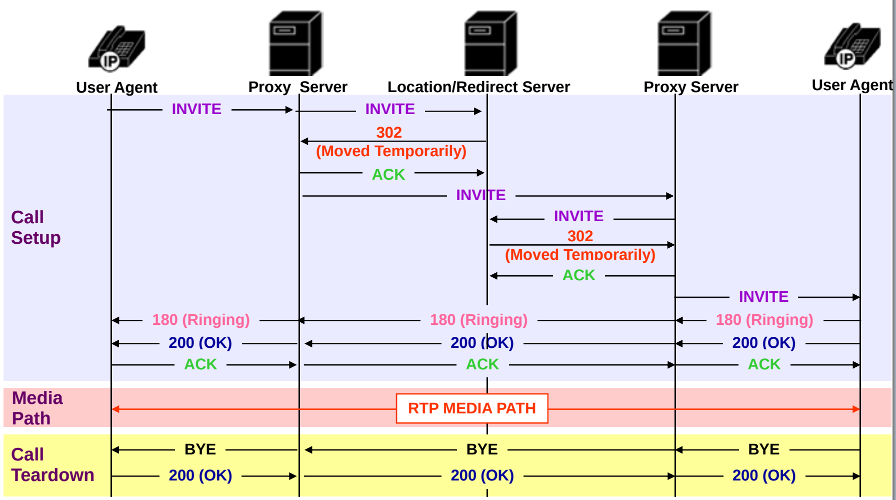
Registration

- 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/

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

Location Information and IETF GeoPriv Working Group

credits:

Milind Nimesh, Columbia University

Technische Universität München

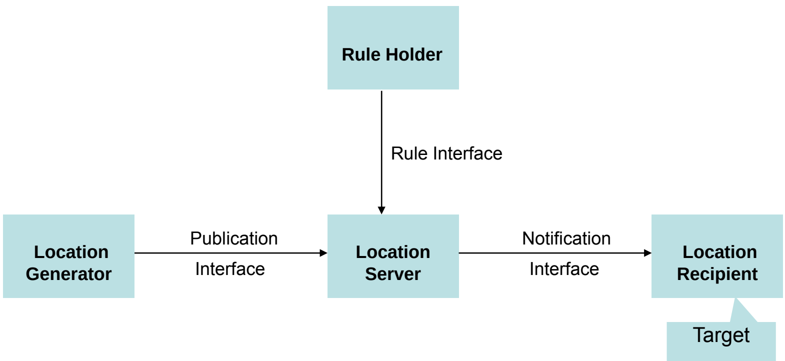
Location Information

- Describes physical position of a person or device:
  - geographical
  - civic (i.e., address)
  - descriptive (e.g. library, airport)
- Formatting and transfer of location information – relatively easy
- Privacy and security – complex
- Application:
  - emergency services
  - resource management
  - social networking
  - search
  - navigation

IETF Geopriv Working Group

- Geographic Location/Privacy working group
- Primary tasks for this working group
  - assess authorization, integrity and privacy requirements
  - select standardized location information format
    - enhance format → availability of security & privacy methods
  - authorization of: requester, responders, proxies
- Goal: transferring location information: private + secure

Geopriv Entities



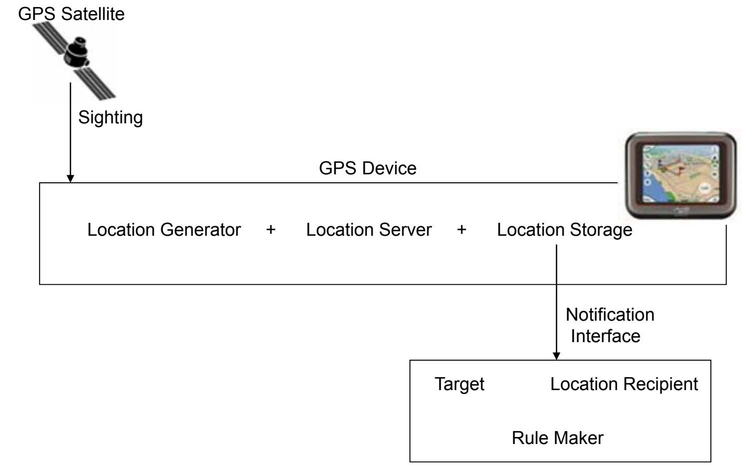
Geopriv Terminology

- Location Object: conveys location information + privacy rules
- Rule Maker: creates rules → governs access to location information
- Target: person/entity whose location communicated
- Using Protocol: protocol carrying location object
- Viewer: consumes location information but does not pass information further
- c.f. RFC 3693

Geopriv Requirements

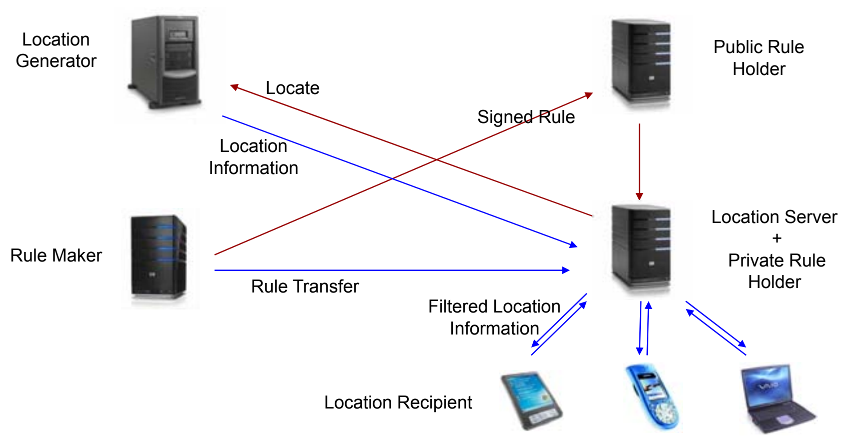
- Secure transmission of location objects
- User controlled privacy rules
- Filtering location information
- Location object carries core set of privacy rules
- Ability of user to hide real identity

Scenarios



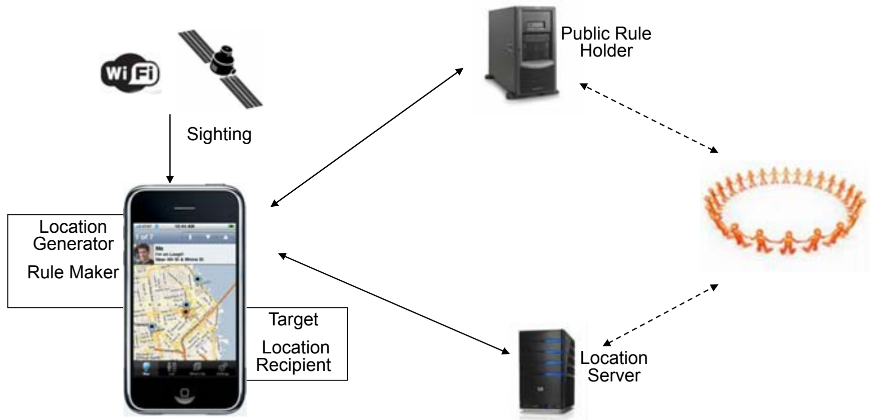
GPS Device with Internal Computing Power: Closed System

Scenarios



Mobile Communities and Location-Based Services

Applications: Social Networking



Location Configuration

Configuring the location of a device, using means such as:

- DHCP extensions
  - RFC3825 : Option 123, geo-coordinate based location
  - RFC4776 : Option 99, civic address
- Link Layer Discovery Protocol - Media Endpoint Discovery
  - LLDP - a vendor-neutral Layer 2 protocol that allows a network device to advertise its identity and capabilities on the local network.
    IEEE standard 802.1AB-2005 in May 2005.
    Supersedes proprietary protocols like Cisco Discovery Protocol,
  - auto-discovery of LAN information (system id, port id, VLAN id, DiffServ settings, …) ⇨ plug & play
  - cisco discovery protocol: switch broadcasts switch/port id
    - switch → floor, port → room ⇨ room level accuracy
- HTTP Enabled Location Delivery
  - device retrieves location from Location Information Server (LIS)
  - assumption: device & LIS present in same admin domain;
    find LIS by DHCP, IPv6 anycast, …
- Applications ⇨ emergency 911, VoiP, location based applications

Security Considerations

- Traffic Analysis
  - attacks on target and privacy violations
- Securing the Privacy Rules
  - rules accessible to LS
  - authenticated using signature
- Emergency Case
  - handling authentication failure
- Identities & Anonymity

Presence Information Data Format - PIDF

- XML based object format to communicate presence information
  ⇨ Instant Messaging (IM)
- PIDF extension to carry geographical information:
- Extended PIDF encapsulates
  - preexisting location information formats
  - security & policy control
- Protocols capable of carrying XML or MIME types suitable
- Security: MIME-level → S/MIME

PIDF Elements

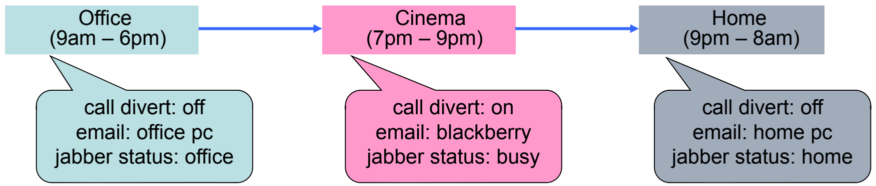
Baseline: RFC 3863

- entity
- contact (how to contact the person)
- timestamp
- status
- tuple (provide a way of segmenting presence information)

Extensions: RFC 4119

- location-info
- usage-rules
  - retransmission-allowed
  - retention-expires
  - ruleset-reference
  - note-well
- method
- provided-by

Location Type Registry



- Describes places of humans or end systems
- Application
  - define location-based actions
  - e.g. if loc = “classroom” then cell phone ringer = off
  - e.g. if loc = “cinema” then call divert = on
- Location coordinate knowledge ≠ context
- airport, arena, bank, bar, bus-station, club, hospital, library….
- ⇨ Prediction:
  most communication will be presence-initiated or pre-scheduled

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

Maintaining network state

Design Principles

Goals:

- identify, study common architectural components, protocol mechanisms
- what approaches do we find in network architectures?
- *synthesis:* big picture

7 design principles:

- network virtualization: overlays
- separation of data, control ⇨signalling
- **hard state versus soft state**
- randomization
- indirection
- multiplexing
- design for scale

Maintaining network state

state: information *stored* in network nodes by network protocols

- updated when network “conditions” change
- stored in multiple nodes
- often associated with end-system generated call or session
- examples:
  - ATM switches maintain lists of VCs: bandwidth allocations, VCI/VPI input-output mappings
  - RSVP routers maintain lists of upstream sender IDs, downstream receiver reservations
  - TCP: Sequence numbers, timer values, RTT estimates

Hard-state

- state *installed* by receiver on receipt of *setup message* from sender
- state *removed* by receiver on receipt of *teardown message* from sender
- *default assumption:* state valid unless told otherwise
  - in practice: failsafe-mechanisms (to remove orphaned state) in case of sender failure e.g., receiver-to-sender "heartbeat": is this state still valid?
- examples:
  - Q.2931 (ATM Signaling)
  - ST-II (Internet hard-state signaling protocol - outdated)
  - TCP

Soft-state

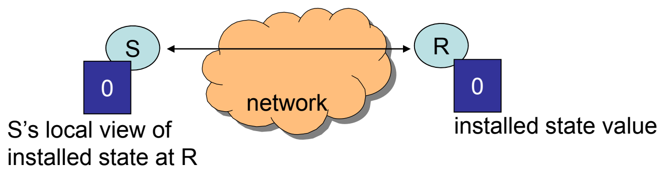
- state *installed* by receiver on receipt of setup (trigger) message from sender (typically, an endpoint)
  - sender also sends periodic *refresh* message: indicating receiver should continue to maintain state
- state *removed* by receiver via timeout, in absence of refresh message from sender
- default assumption: state becomes invalid unless refreshed
  - in practice: explicit state removal (*teardown*) messages also used
- examples:
  - RSVP, RTP/RTCP, IGMP

State: senders, receivers

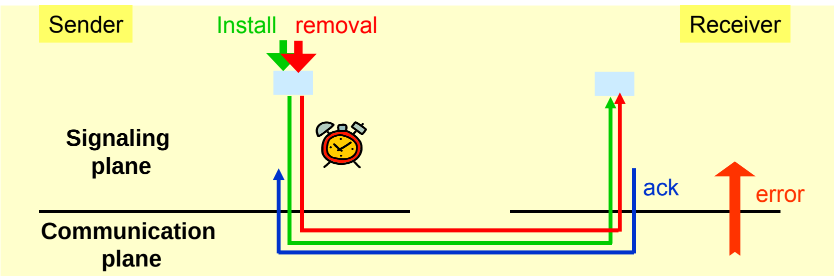
- sender: network node that *(re)generates* signaling (control) messages to install, keep-alive, remove state from other nodes
- receiver: node that creates, maintains, removes state based on signaling messages *received* from sender

Let's build a signaling protocol

- *S:* state *S*ender (state installer)
- *R:* state *R*eceiver (state holder)
- desired functionality:
  - S: set values in R to 1 when state "installed", set to 0 when state "not installed"
  - if other side is down, state is not installed (0)
  - initial condition: state not installed

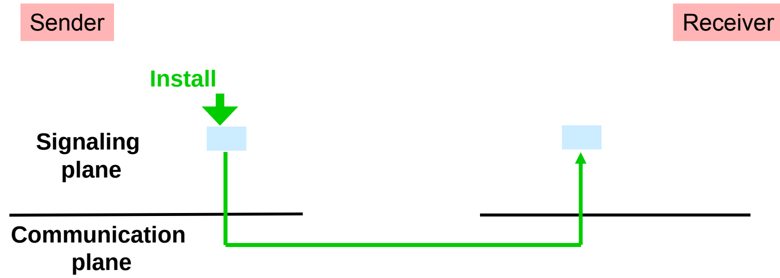


Hard-state signaling



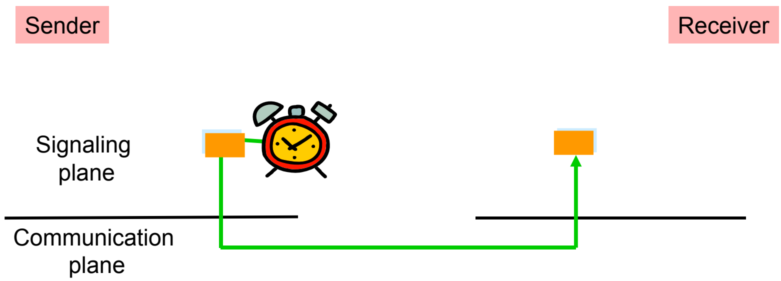
- reliable signaling
- state removal by request
- requires additional error handling
  - e.g., sender failure

Soft-state signaling



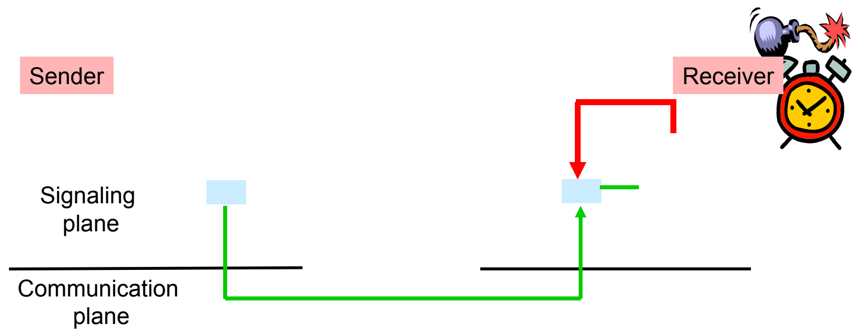
- best effort signaling

Soft-state signaling



- best effort signaling
- refresh timer, periodic refresh

Soft-state signaling



- best effort signaling
- refresh timer, periodic refresh
- state time-out timer, state removal only by time-out

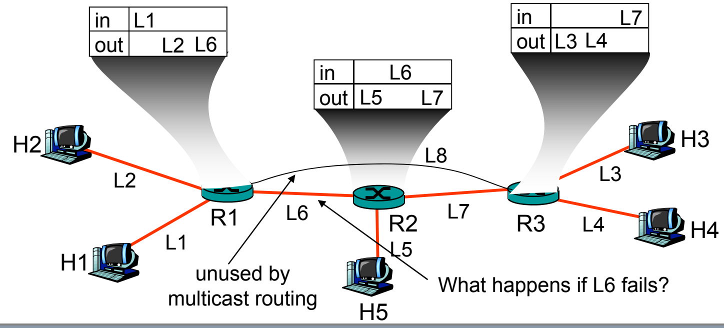
Soft-state: claims

- “Systems built on soft-state are robust” [Raman 99]
- “Soft-state protocols provide .. greater robustness to changes in the underlying network conditions…” [Sharma 97]
- “obviates the need for complex error handling software” [Balakrishnan 99]

What does this mean?

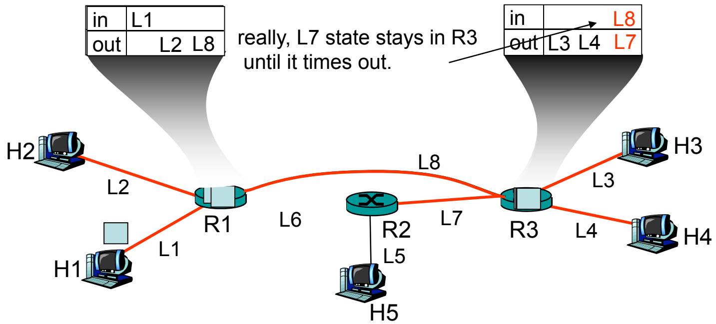
Soft-state: “easy” handling of changes

- Periodic refresh: if network “conditions” change, refresh will re-establish state under new conditions
- example: RSVP/routing interaction: if routes change (nodes fail) RSVP PATH refresh will *re-establish* state along new path



Soft-state: “easy” handling of changes

- L6 goes down, multicast routing reconfigures but…
- H1 data no longer reaches H3, H4, H5 (no sender or receiver state for L8)
- H1 refreshes PATH, establishes *new* state for L8 in R1, R3
- H4 refreshes RESV, propagates upstream to H1, establishes new receiver state for H4 in R1, R3



Soft-state: “easy” handling of changes

- “recovery” performed transparently to end-system by normal refresh procedures
- no need for network to signal failure/change to end system, or end system to respond to specific error
- less signaling (volume, types of messages) than hard-state from network to end-system but…
- more signaling (volume) than hard-state from end-system to network for refreshes

Soft-state: refreshes

- refresh messages serve many purposes:
  - trigger: first time state-installation
  - refresh: refresh state known to exist (“I am still here”)
  - <lack of refresh>: remove state (“I am gone”)
- challenge: all refresh messages unreliable
  - problem: what happens if first PATH message gets lost?
    - copy of PATH message only sent after refresh interval
  - would like triggers to result in state-installation a.s.a.p.
  - enhancement: add receiver-to-sender refresh_ACK for triggers
  - sender initiates retransmission if no refresh_ACK is received after short timeout
  - e.g., see paper “Staged Refresh Timers for RSVP” by Ping Pan and Henning Schulzrinne
  - approach also applicable to other soft-state protocols

Signaling Spectrum