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

Announcement for Seminars, SS 2012

- c.f. http://www.net.in.tum.de/en/teaching/ss12/seminare/
- Information and registration for topics:
  24.01.2012, 12:00-14:00, Room 03.07.023
- Future Internet - Block Seminar, 12.-13.04.2012
- Innovative Internet-Technologien und Mobilkommunikation
  Fridays 14:30 - 16:30 Uhr, Room 03.07.023
- Aerospace Networks
  Fridays 14:30 - 16:30 Uhr, Room 03.07.023
- Sensor Nodes: Operation Modes, Networks and Applications
  Block Seminar, 23.-24.07.2012
  http://www.net.in.tum.de/fileadmin/TUM/teaching/seminar_iit/SS12/Sensorknoten/Sensorknoten-SS2012.pdf

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

SIP

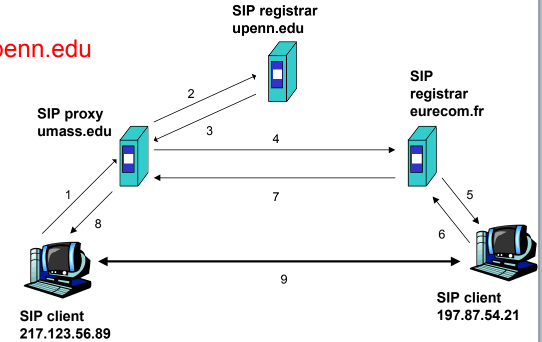
Credits in addition to
Jim Kurose and Keith Ross:

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

Technische Universität München

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 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

SIP Messages – Methods and Responses

SIP components communicate by exchanging SIP messages:

SIP Methods:

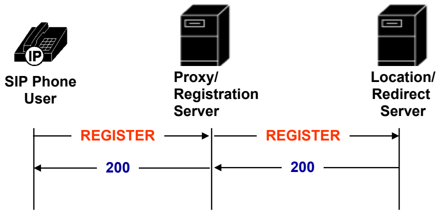
- INVITE – Initiates a call by inviting user to participate in session.
- ACK - Confirms that the client has received a final response to an INVITE request.
- BYE - Indicates termination of the call.
- CANCEL - Cancels a pending request.
- REGISTER – Registers the user agent.
- OPTIONS – Used to query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

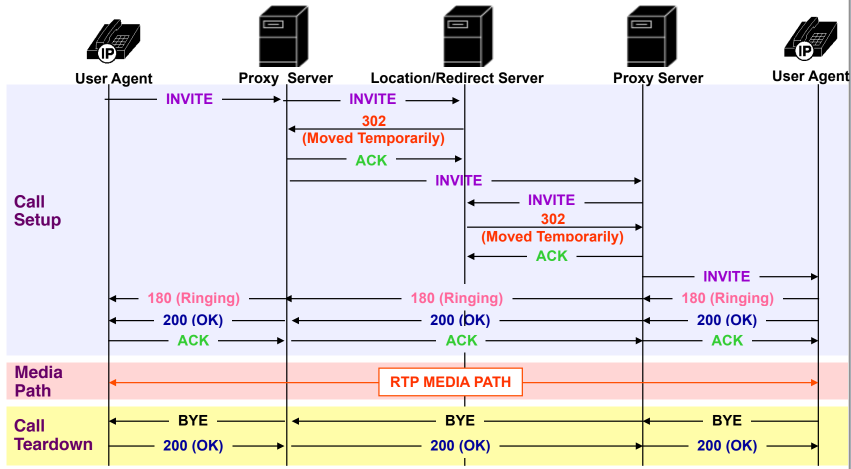
Registration

- 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.

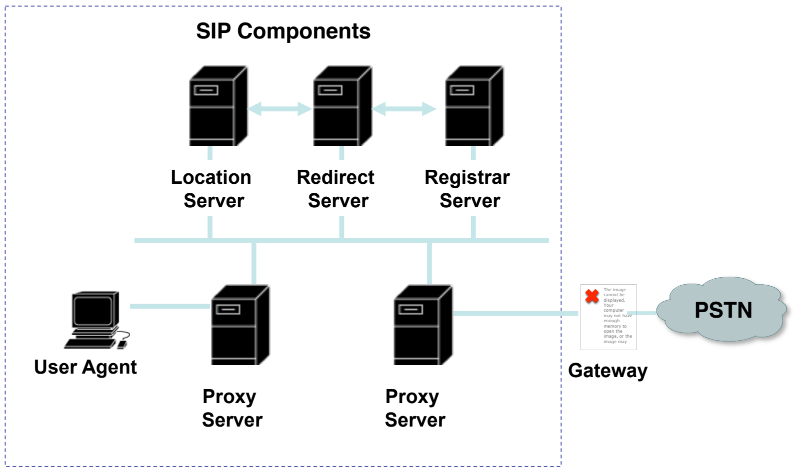


SIP Messages:
REGISTER – Registers the address listed in the To header field.
200 – OK.

Simplified SIP Call Setup and Teardown



SIP Architecture



User Agents, Proxy Server, Registrar Server

- **User Agent**: 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).
- Location can be transmitted by-value or by-reference.
- Location by reference is done by a URI that refers to a UA or proxy server
- A location Server transmits location by value in form of 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 – 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: RFC 3880
    - 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

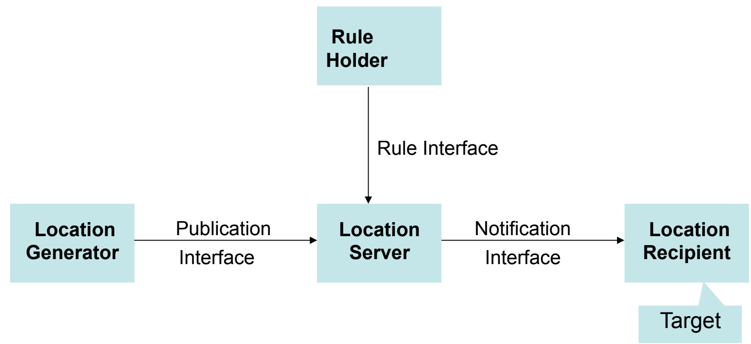
Location Information

- 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



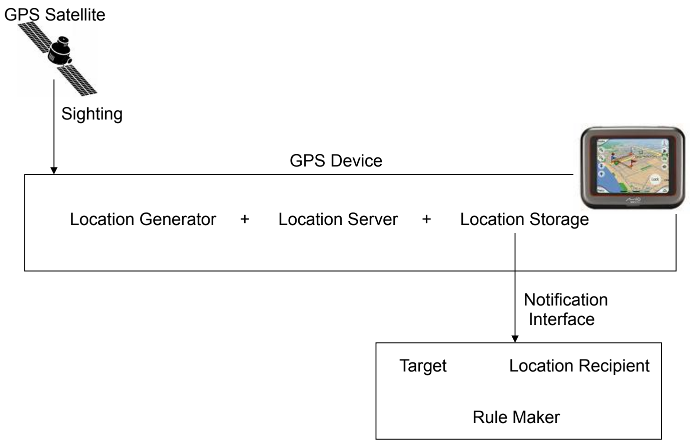
Geopriv Terminology

- 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

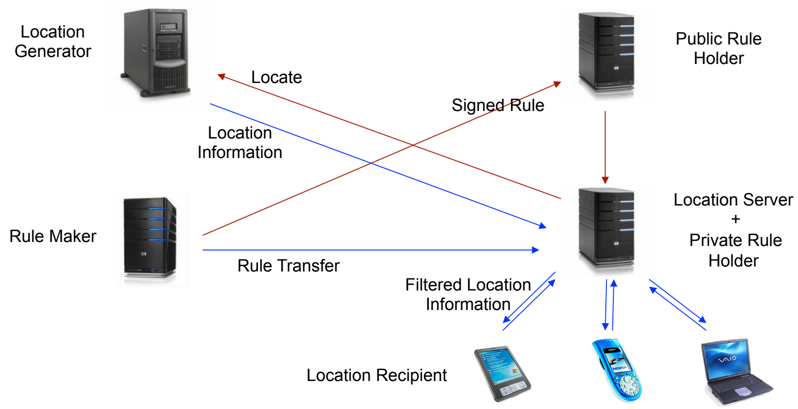
- Secure transmission of location objects
- User controlled privacy rules
- Filtering location information
- Location object carries core set of privacy rules
- Ability of user to hide real identity

Scenarios



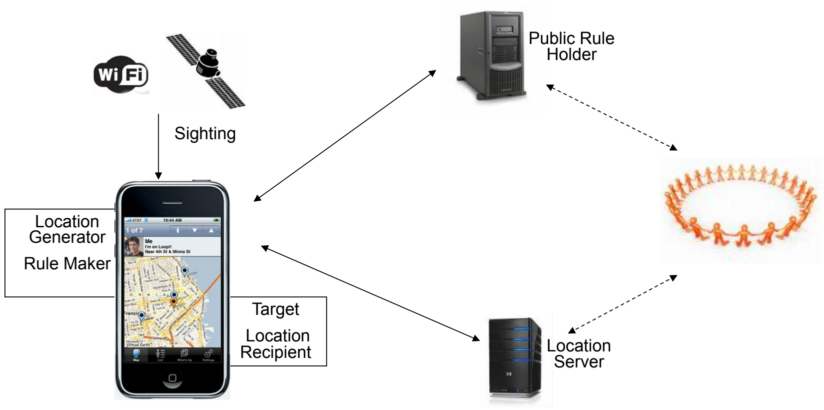
GPS Device with Internal Computing Power: Closed System

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

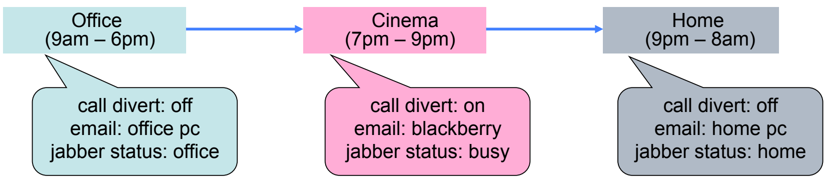
- location-info
- usage-rules
  - retransmission-allowed
  - retention-expires
  - ruleset-reference
  - note-well
- method
- provided-by

PIDF-LO Example

- PIDF-LO: RFC 4119 (RFC 5139, RFC 5491)
- c.f. http://www.voip-sos.net/tools/pidflo/

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
    entity="pres:sample@example.com">
<tuple id="0815">
<status>
 <gp:geopriv>
  <gp:location-info><!--  location information is inserted here --></gp:location-info>
  <gp:usage-rules>
    <gp:retransmission-allowed>no</gp:retransmission-allowed>
    <gp:retention-expiry>2010-08-10T09:00:10+02:00</gp:retention-expiry>
  </gp:usage-rules>
 </gp:geopriv>
</status>
<timestamp>2010-08-10T08:31:00+02:00</timestamp>
</tuple>
</presence>
```


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

GeoPriv RFCs

- RFC 3693: Geopriv Requirements, 2004 (Informational), Updated by RFC 6280
- RFC 3694: Threat Analysis of the Geopriv Protocol, 2004 (Informational), Updated by RFC 6280
- RFC 3825: Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information, 2004 (Proposed Standard), Obsoleted by RFC 6225
- RFC 4079: A Presence Architecture for the Distribution of GEOPRIV Location Objects, 2005 (Informational)
- RFC 4119: A Presence-based GEOPRIV Location Object Format, 2005 (Proposed Standard), Updated by RFC 5139, RFC 5491
- RFC 4589: Location Types Registry, 2006 (Proposed Standard)
- RFC 4676: Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information, 2006 (Proposed Standard), Obsoleted by RFC 4776
- RFC 4745, Common Policy: A Document Format for Expressing Privacy Preferences, 2007 (Proposed Standard)
- RFC 4776: Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information, 2006 (Proposed Standard), Updated by RFC 5774

GeoPriv RFCs

- RFC 5139: Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO), 2008 (Proposed Standard)
- RFC 5491: GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations 2009 (Proposed Standard)
- RFC 5580: Carrying Location Objects in RADIUS and Diameter, 2009 (Proposed Standard)
- RFC 5606: Implications of 'retransmission-allowed' for SIP Location Conveyance, 2009 (Informational)
- RFC 5687: GEOPRIV Layer 7 Location Configuration Protocol: Problem Statement and Requirements, 2010 (Informational)
- RFC 5774: Considerations for Civic Addresses in the Presence Information Data Format Location Object (PIDF-LO): Guidelines and IANA Registry Definition, 2010 (Best Current Practice)
- RFC 5808: Requirements for a Location-by-Reference Mechanism, 2010 (Informational)

GeoPriv RFCs

- RFC 5870: A Uniform Resource Identifier for Geographic Locations ('geo' URI), 2010 (Proposed Standard)
- RFC 5985: HTTP-Enabled Location Delivery (HELD), 2010 (Proposed Standard)
- RFC 5986: Discovering the Local Location Information Server (LIS), 2010 (Proposed Standard)
- RFC 6155: Use of Device Identity in HTTP-Enabled Location Delivery (HELD), 2011 (Proposed Standard)
- RFC 6225: Dynamic Host Configuration Protocol Options for Coordinate-Based Location Configuration Information, 2011 (Proposed Standard)
- RFC 6280: An Architecture for Location and Location Privacy in Internet Applications, 2011 (Best Current Practice)

GeoPriv Tools

c.f. http://trac.tools.ietf.org/wg/geopriv/trac/wiki/GeoprivTools

- Open Source LIS: A PHP-based HELD server with a Java-based client, http://held-location.sourceforge.net/
- The Internet Geolocation Toolkit: A multi-platform, multi-protocol C++ library for geolocation access, http://igtk.sourceforge.net/
- ECRITdroid: An emergency calling client for Android. Doesn't do GEOPRIV now (just LoST/ECRIT), but should soon, in order to be fully ECRIT-compliant, http://ecritdroid.googlecode.com/
- Online DHCP encoders: An AJAX tool for encoding location values for use in the DHCP location options; http://geopriv.dreamhosters.com/dhcloc/
- Firefox implementation of W3C Geolocation API: supports a limited profile of HELD. To enable: Go to "about:config"; set "geo.wifi.protocol" to "1"; set "geo.wifi.uri" to URL of HELD server, https://bugzilla.mozilla.org/show_bug.cgi?id=545001
- CommScope LIS: commercial LIS, http://www.commscope.com

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

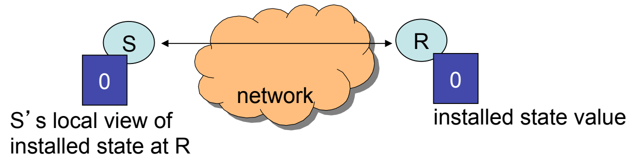
- state *installed* by receiver on receipt of setup (trigger) message from sender (typically, an endpoint)
  - sender also sends periodic *refresh* message: indicating receiver should continue to maintain state
- state *removed* by receiver via timeout, in absence of refresh message from sender
- default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

State: senders, receivers

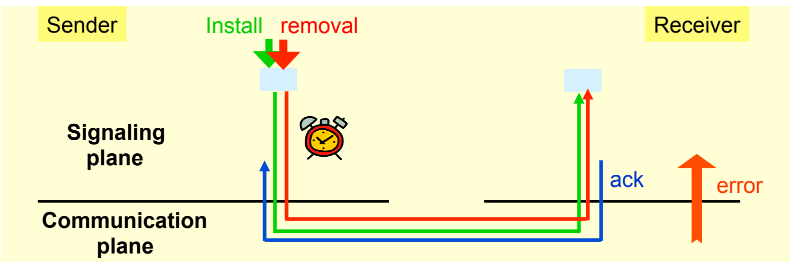
- sender: network node that *(re)generates* signaling (control) messages to install, keep-alive, remove state from other nodes
- receiver: node that creates, maintains, removes state based on signaling messages *received* from sender

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

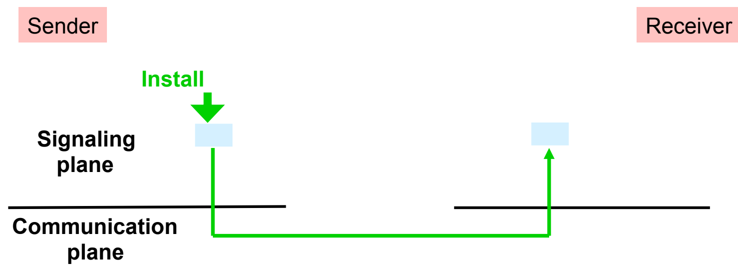


Hard-state signaling



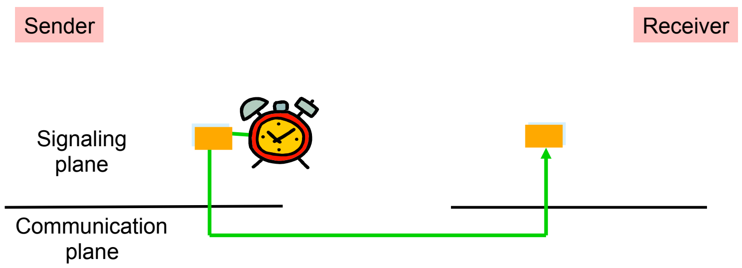
- reliable signaling
- state removal by request
- requires additional error handling
  - e.g., sender failure

Soft-state signaling



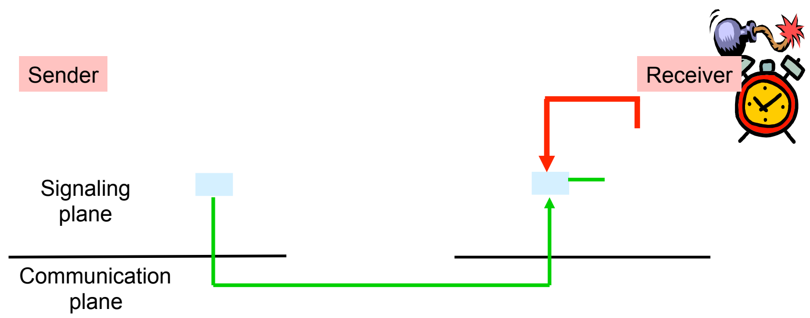
- best effort signaling

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

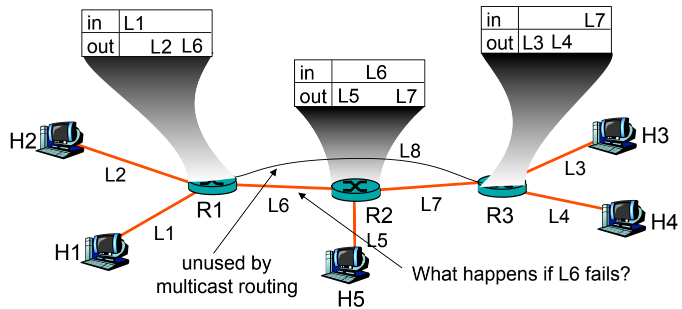
Soft-state: claims

- "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?

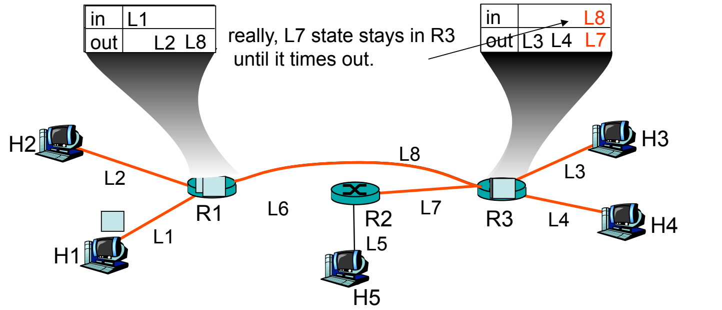
Soft-state: "easy" handling of changes

- 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