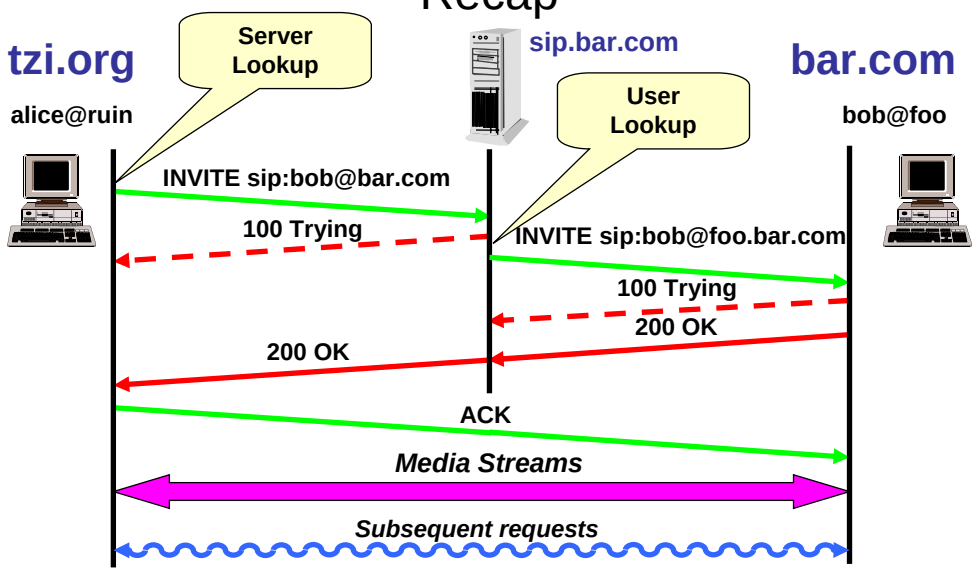
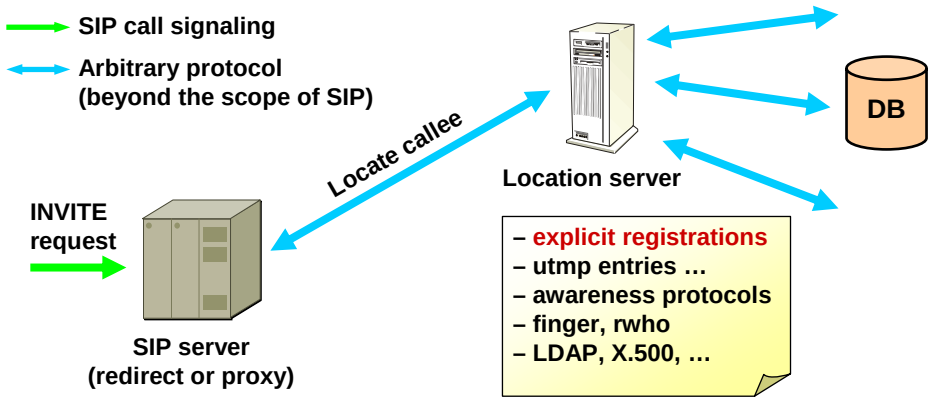
SIP Registration and Location

Recap

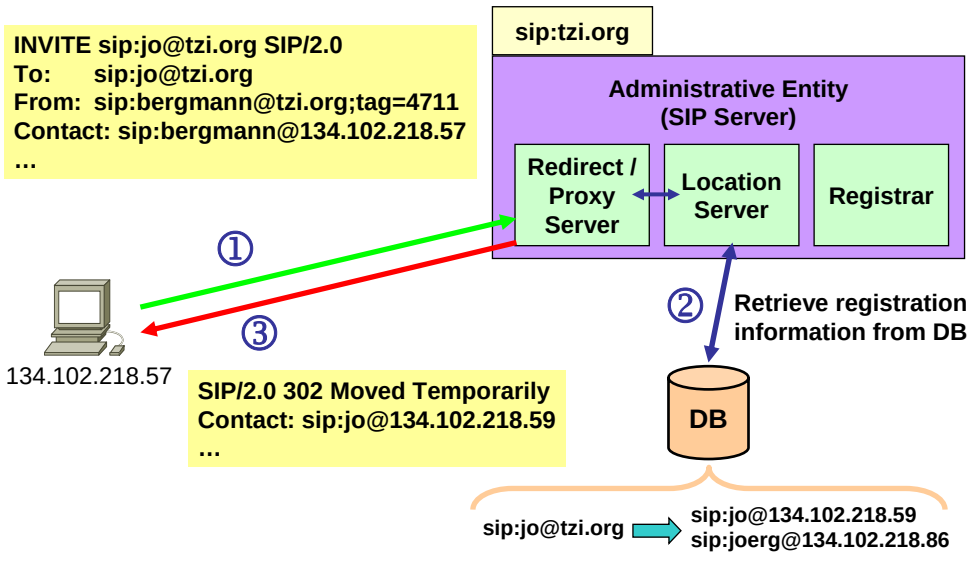


User Location

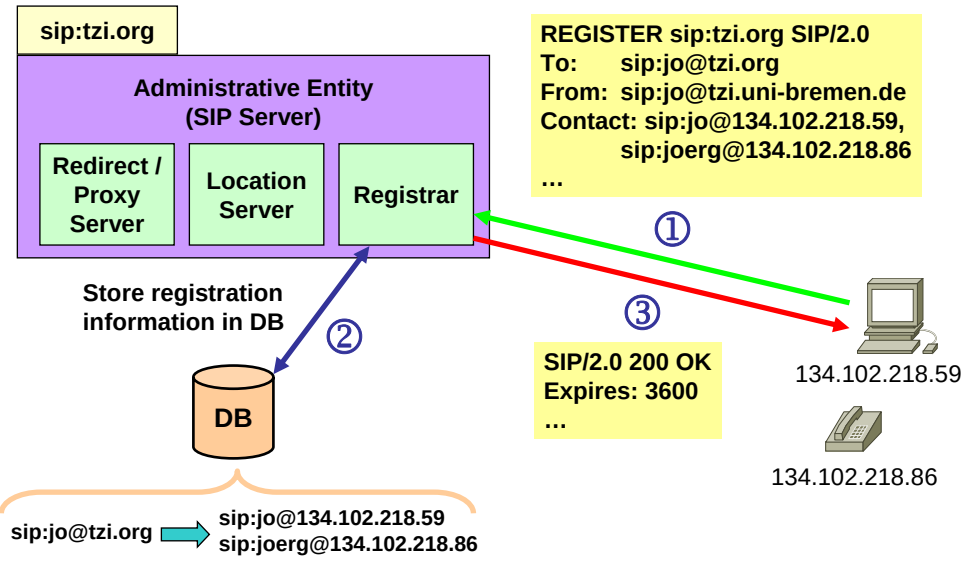


- SIP server asks location server where to find callee
- Location server returns list of contact addresses
- SIP server proxies or redirects request according to address list

User Location



User Registration (1)



User Registration (2)

- Send REGISTER request to registrar

```
REGISTER sip:tzi.org SIP/2.0
To:      sip:jo@tzi.org
From:  sip:jo@tzi.uni-bremen.de
Contact: sip:jo@134.102.218.1,
         sip:joerg@134.102.218.86
...
```

- Request URI sip:domain
  → registrar may refuse requests for foreign domains
- To: canonical name for registered user (usually sip:user@domain)
- From: responsible person
  (may vary from To: for third party registration)
- Contact: contact information for the registered user
  → address, transport parameters, redirect/proxy, capabilities
- Specified addresses are merged with existing registrations
- Registrar denotes expiration time in Expires: header
- Client refreshes registration before expiry

Registration Expiry

- Client requests lifetime
  - Contact:-header parameter expires
  - SIP message header field Expires:
  - Relative duration (seconds)
    - RFC 2543: absolute date allowed (RFC 1123, only GMT)
  - Default if no or malformed expiry time requested: 3600 seconds
- Registrar may use lower value, indicated in 200 OK
  - Registrar must not increase expiry interval
  - May decline request with “423 Registration Too Brief”
    - Include Min-Expiry: header
- After expiration, registrar silently discards corresponding database entries

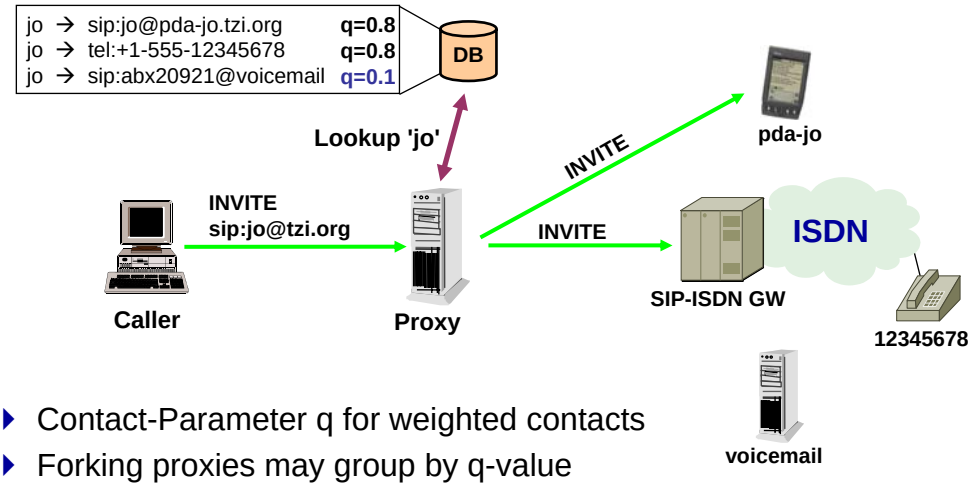
Add/update Registration Entries

- Retrieve all entries for user@domain (specified in To:-header) from database
- Compare with Contact:-addresses according to scheme-specific rules:
  - Add addresses for which no entries exist
  - Entries being equal to a contact address are updated
- Otherwise return 200 OK response
  - Include all entries for user@domain
- Response to REGISTER to include all registered Contacts

Lookup and Delete

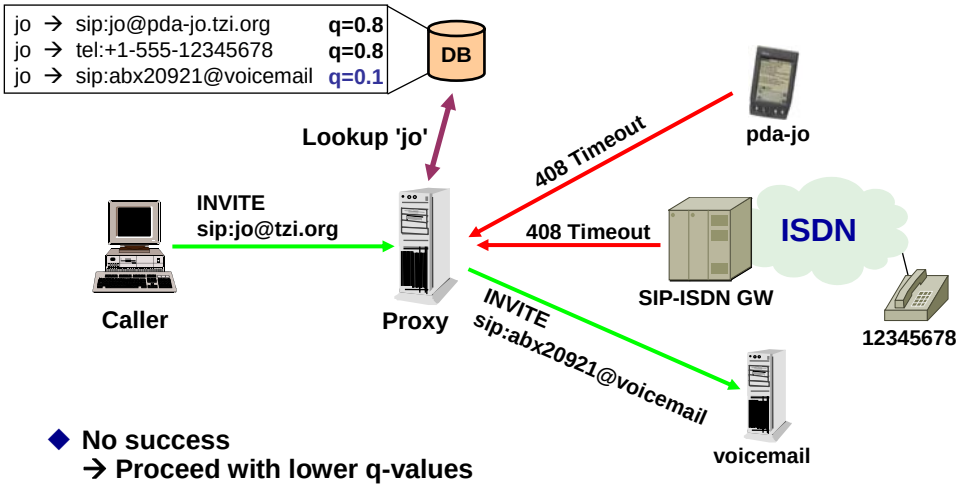
- Lookup entries:
  - No Contact:-header in request
    → Current registrations are returned in OK response
  - For further processing:
    - Client uses q-parameter to determine relative order
- Delete entries:
  - Expires: 0
    - Delete URIs specified in Contact:-Header from database
    - Delete all entries for user: Contact: *

Parallel vs. Sequential Forking



- Contact-Parameter q for weighted contacts
- Forking proxies may group by q-value
  - example: call voicemail system only if no answer from other UAs

Parallel vs. Sequential Forking



- **No success**
  → **Proceed with lower q-values**
- **Higher call setup delay**
- **Less REGISTER requests necessary**

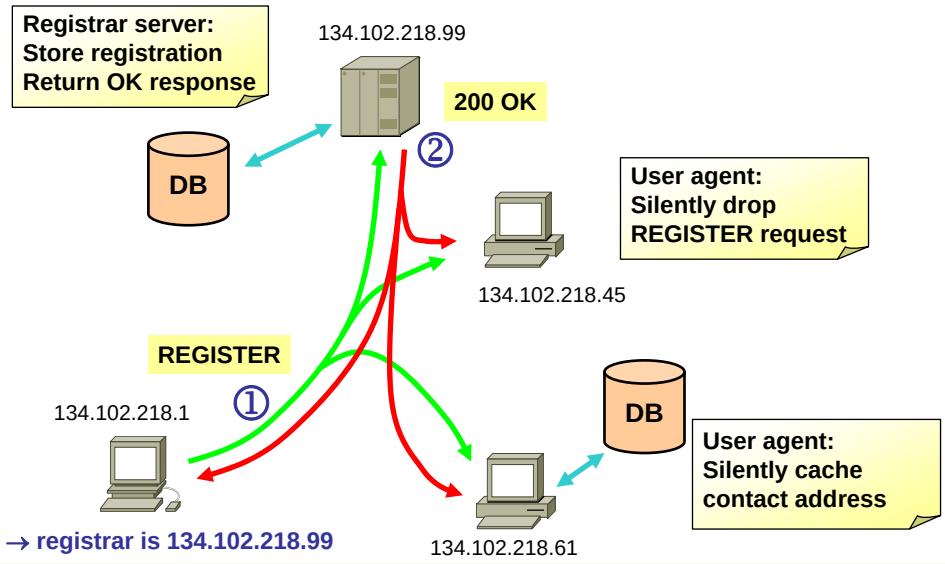
Finding a Local Registrar Server

- Multicast REGISTER request
  - Send link-local request to sip.mcast.net (224.0.1.75)
  - Use address of first server that responds with OK
  - If other OK responses, use sender addresses as fallback
- Alternatively, use configured registrar address
- Use DNS lookup on domain name to register with

Other mechanisms out-of-scope of core spec. Examples:

- DHCP
  - DNS SRV lookup on domain name obtained via DHCP
  - If no SIP server found, query A or AAAA record
- Service Location Protocol (SLP), …

Link-local Multicast Registration



Extended Registration Functions (1)

- Registrar is the initial point of contact for a UA
- Idea: provide (user-specific) configuration and other information in REGISTER response
  - Allows for centralized control using standard SIP mechanisms
- Sample services
  1. Service Route Discovery
  2. Globally Routable UA URIs
  3. UA capabilities or preferences

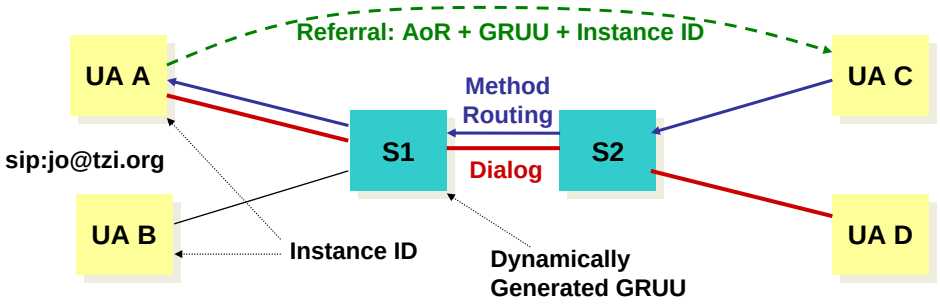
Service Route Discovery

- Dynamically configure “home service proxies” for UAs
  - Proxies to be traversed first for outbound requests
  - Routing of incoming requests can be enforced by infrastructure
  - Routing of responses implied from Via: headers
- Simple extension header returned by registrar

  Service-Route: <sip:sp1.example.com;lr>, <sip:sp2.example.com;lr>
  - Syntax similar to Route: header
- Used by UA subsequently for loose source routing
  - Included in Route: header
- Sample usages: request authorization, service provisioning, …

Globally Routable UA URI (GRUU)

- Users may be registered at multiple UAs
  - Incoming SIP requests may be forked to all UAs
  - May need to direct a request to a particular UA
    - Examples: Call Transfer, Conferencing, Presence
  - Add an instance identification for the UA: (AoR, Instance ID) pair



GRUU (2)

- GRUUs are UA SIP URIs that can be handed out to third parties
  - Resolve to the SIP domain (and thus SIP proxy) of the UA
  - Can be mapped by the SIP proxy to a specific UA
- Types of GRUUs
  - Public: the identity (the Address of Record) can be derived from the GRUU
    sip:jo@netlab.tkk.fi;gr=1234567890
  - Temporary: the identity cannot be derived
    - GRUUs may be cryptographically generated to minimize the state to be maintained

    sip:khfdshf84fkh@netlab.tkk.fi;gr
  - gr parameter indicates that the URI is a GRUU
- GRUUs are usually generated by a proxy
  - Public GRUUs may also be created by a UA, e.g., sip:jo@130.233.152.81;gr

GRUU (3)

- Registration for sip:jo@acm.org
  - UA provides Instance ID per Contact: header and indicates support

    Contact: <sip:jo@cl42.acm.org>;+sip.instance="urn:uuid:32fed8…"

    Supported: gruu
  - Registrar assigns public and temporary GRUU per Contact: header

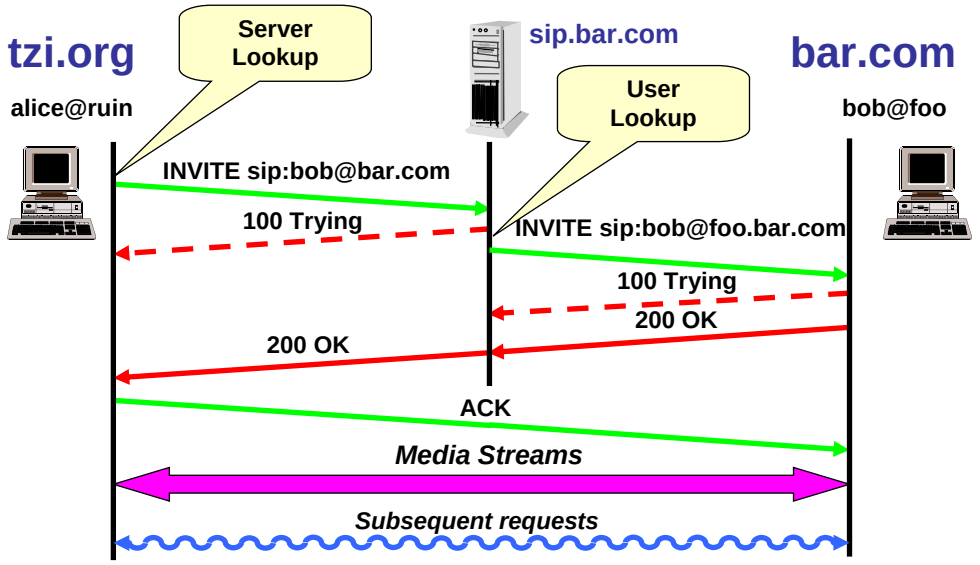
    Contact: <sip:jo@cl42.acm.org>;pub-gruu="sip:jo@cl42.acm.org;gr="urn:uuid:32fed..."
    ;temp-gruu="sip:dsiah-0=@acm.org;gr";+sip.instance="urn:uuid:32fed8…"
  - Valid for the lifetime of this UA's registration
  - Local proxies ensure proper routing to the right UA
- UA may place GRUU in Contact: header in later requests

  Contact: <sip:dsiah-0=@acm.org;gr>
- Remote UAs will use GRUU to send messages to

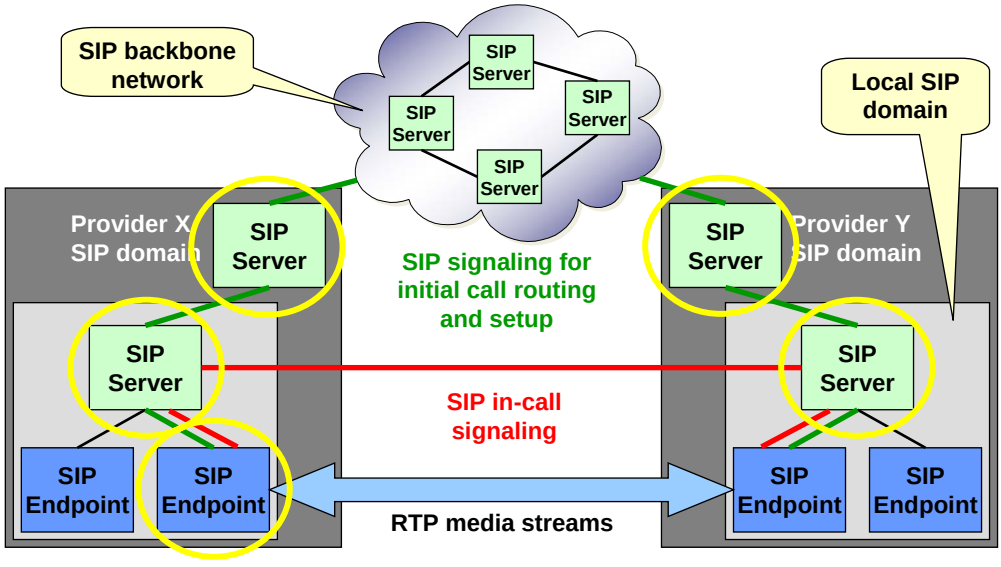
  REFER sip:bergmann@tzi.org SIP/2.0

  Refer-To: <sip:dsiah-0=@acm.org;gr>

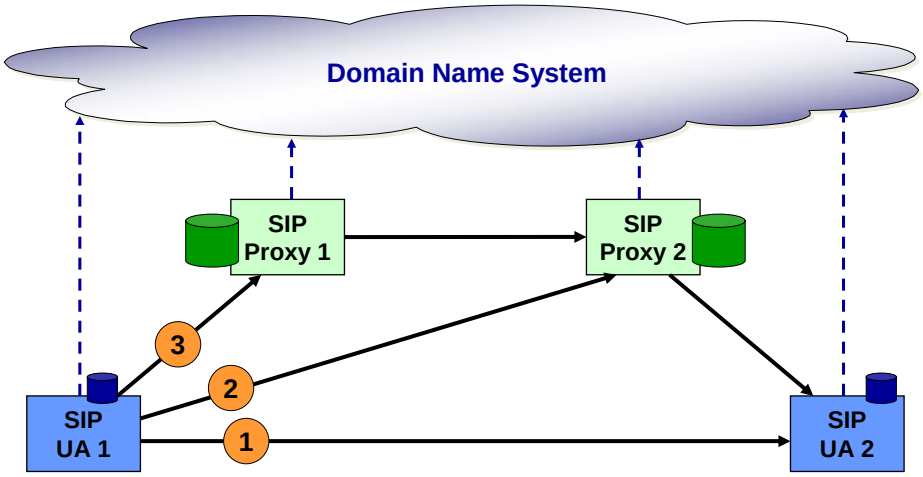
Recap (Part II)



Global SIP Architecture



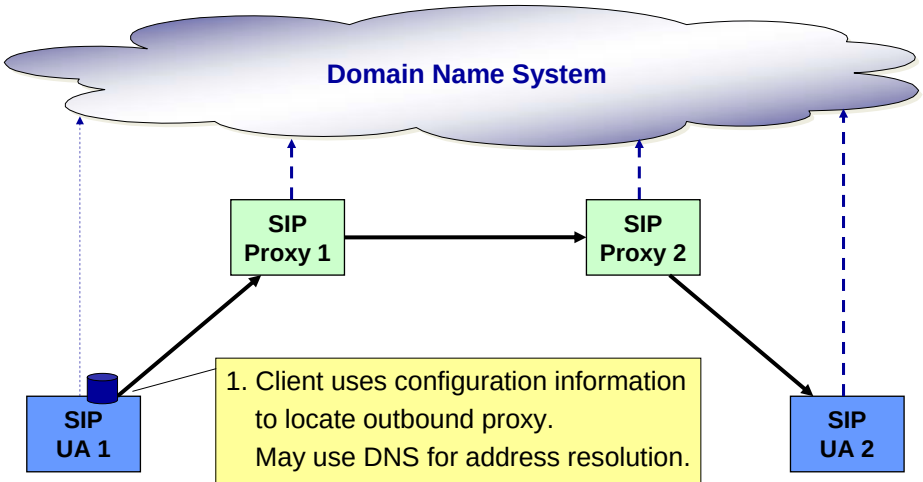
SIP Address Resolution Steps



sip:ob@tzi.org

sip:jo@acm.org

SIP Address Resolution Steps

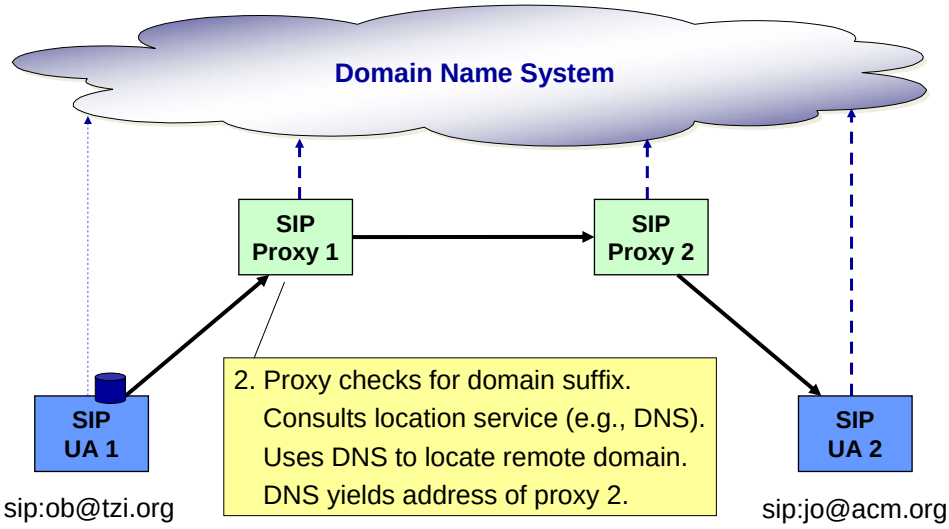


1. Client uses configuration information to locate outbound proxy.
   May use DNS for address resolution.
   All requests sent to proxy.

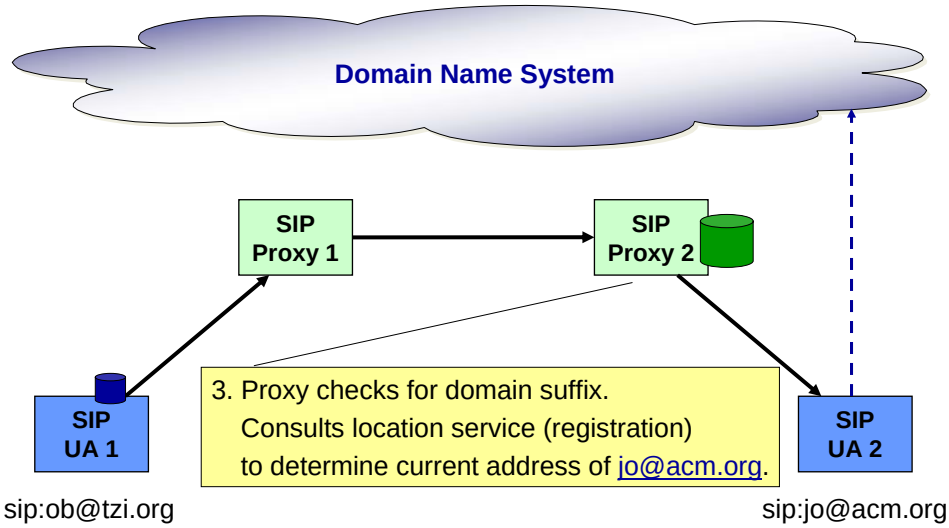
sip:ob@tzi.org

sip:jo@acm.org

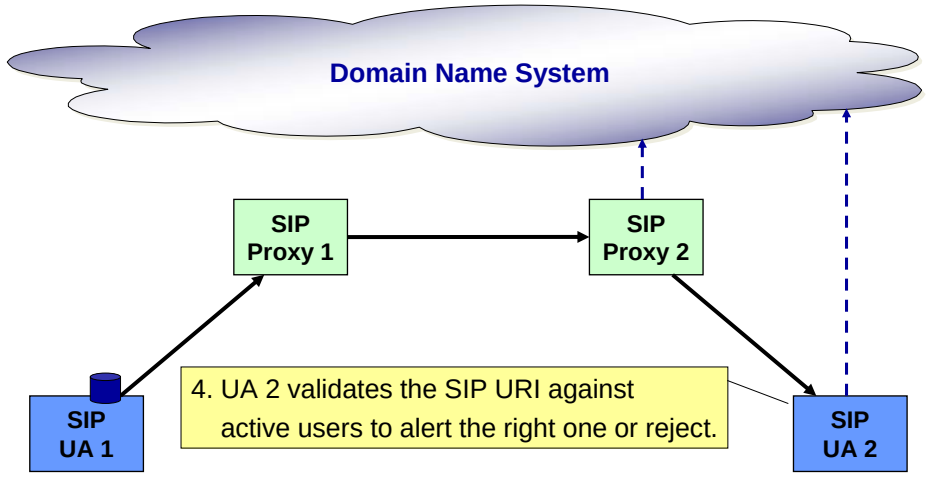
SIP Address Resolution Steps



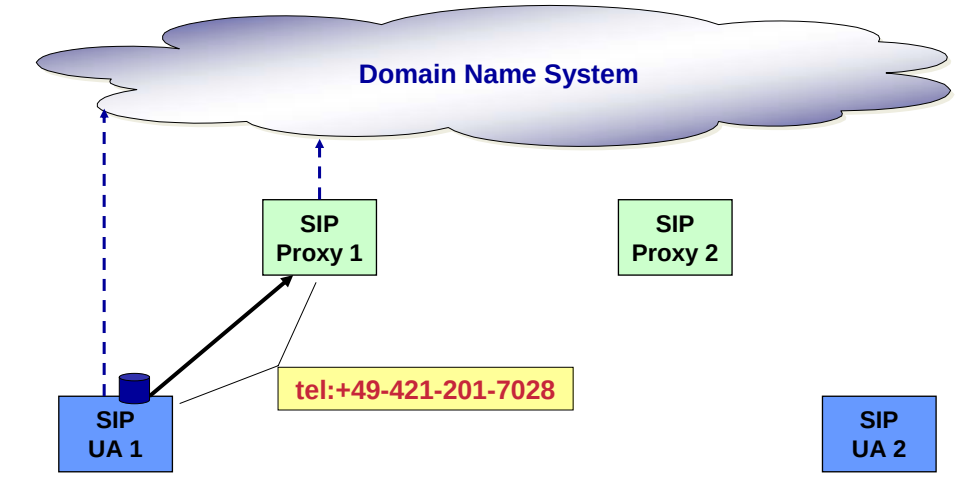
SIP Address Resolution Steps



SIP Address Resolution Steps



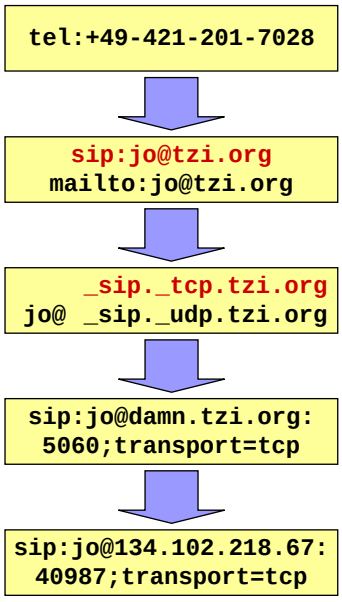
Address Resolution for Phone Numbers



Finding the Next Hop for tel: URIs

- **UAC may use a (manually) configured outbound proxy**
  - Outbound proxy may also have be learned upon registration
- If request URI contains IP address and port, message can be sent directly
- Otherwise, **determine next hop SIP server via DNS**
  - Use NAPTR RR (SIP+D2U/D2T/D2S, SIPS+D2T/D2S) to obtain SRV records
  - Query for SRV RR: _sip._udp, _sip._tcp, _sips._tcp for all supported transport protocols
  - If entries found, try as specified in RFC 2782
    - If no entries found, use UDP for sip: URIs and TCP for sips: URIs
- **Query A or AAAA records for IP address**

(ENUM) DNS Lookup Process

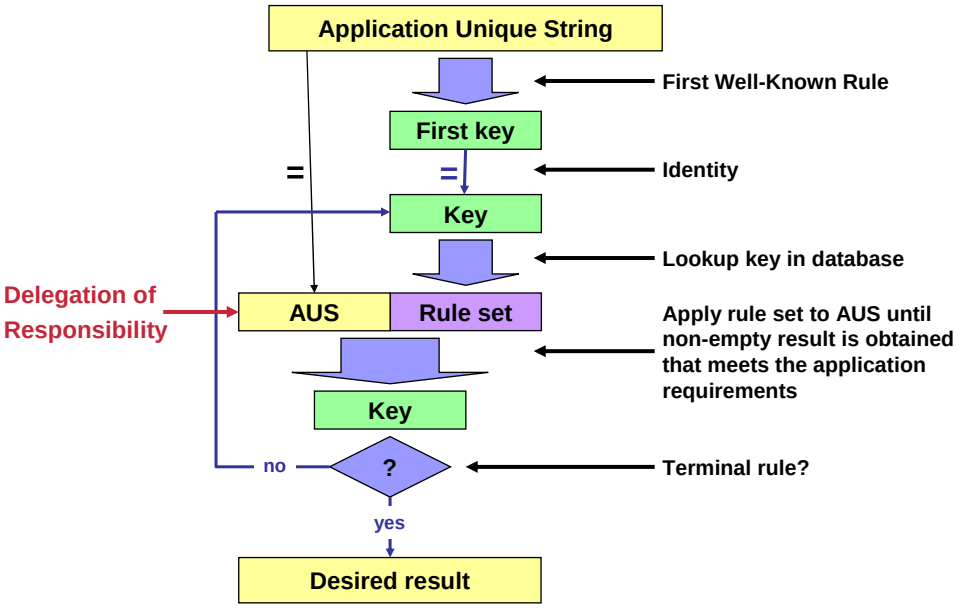


- ENUM service lookup
- Uses DNS NAPTR Resource Records
- NAPTR RR lookup to select preferred SIP services
- Based upon transport protocols and TLS
- SRV RR lookup for load balancing
- May or may not yield IP address
- A or AAAA lookup to determine IP address(es) associated with name

Background

- Background: Dynamic Delegation Discovery System (DDDS)
  - RFC 3401, 3402, 3403, RFC 3761, RFC 3764
- Abstract concept
  - Map Application Unique String (AUS) to some data
  - Based upon a (distributed) database
  - Indexed by keys
  - Lookup yields rule set: matching + substitution rules (regexp-like)
  - Result may point to a different location in the database
  - Algorithm: repeated substitution applied to AUS until a terminal symbol is reached
- Effect: Allow delegation of responsibility to store “some data” for an AUS (and thus to delegate mapping)

Step 1: DDDS General Operation



Step 1: NAPTR Resource Records

- Uses DDDS with DNS as distributed database
  - Key: DNS name

Order	Preference	Flags	Services	regexp	Replacement

- Order: absolute sorting of rules; first matching one is used
- Preference: application-specific priority indication
- Flags: control rule processing; e.g. indicate terminal rule
- Services: available services on this delegation path
- regexp: substitution rule for Application Unique String

- Replacement: replacement for DNS name (= key)

Step 1: ENUM Lookup

- Application unique string: tel:+49-421-201-7028
  - +49-421-201-7028 ➔ +494212017028
- First well-known rule: identity
- Database key: transformation into valid DNS name
  1. Remove leading '+': 494212017028
  2. Put dots between digits: 4.9.4.2.1.2.0.1.7.0.2.8
  3. Reverse order of digits: 8.2.0.7.1.0.2.1.2.4.9.4
  4. Add 'e164.arpa': 8.2.0.7.1.0.2.1.2.4.9.4.e164.arpa
  - Yields the domain name used to query for NAPTR records
- Flags: 'u' to indicate terminal rule
- Service: E2U+servicespec[+servicespec]…
- ENUM services: sip, h323, pres, …

Step 1: SIP Lookup for ENUM

Type	Order	Preference	Flags	Services	regexp	Replacement

- Example: SIP (and other) entries for a user

$ORIGIN 8.2.0.7.1.0.2.1.2.4.9.4.e164.arpa

IN NAPTR	10	100	u	E2U+sip	!^.*$!sip:jo@tzi.org!	.
IN NAPTR	10	101	u	E2U+pres	!^.*$!pres:jo@tzi.org!	.
IN NAPTR	10	102	u	E2U+msg	!^.*$!mailto:jo@tzi.org!	.

Mapping to a Domain Name

Step 2: SIP NAPTR for Transport Selection

Type	Order	Preference	Flags	Services	regexp	Replacement

- Example: Secure SIP preferred over SIP, TCP over UDP
- Looking up <sip:jo@tzi.org>

$ORIGIN tzi.org

IN NAPTR	50	100	u	D2T+SIPS		_sips._tcp.tzi.org
IN NAPTR	90	100	u	D2T+SIP		_sip._tcp.tzi.org
IN NAPTR	100	100	u	D2U+SIP		_sip._udp.tzi.org

Steps 3 and 4: SRV and A Resource Records

- Example: SIP load balancing across three servers

$ORIGIN _sip._tcp.tzi.org

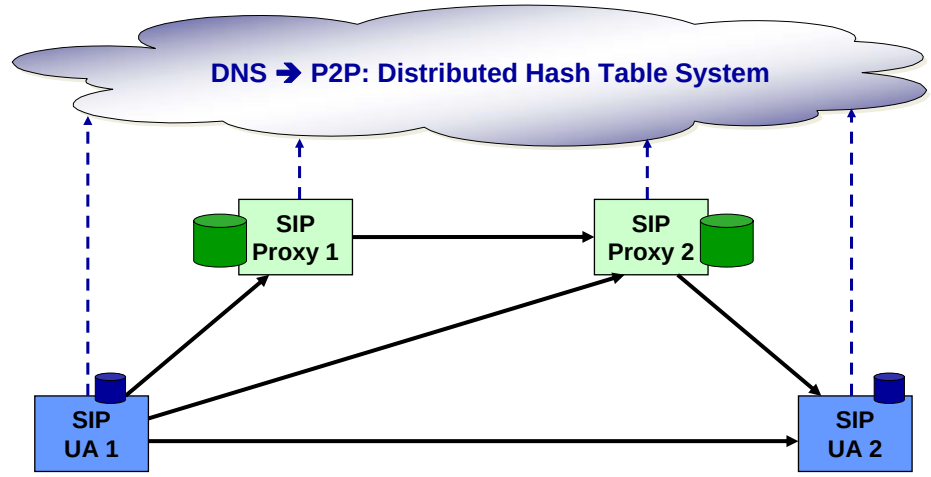
IN SRV	0	1	5060	damn.tzi.org
IN SRV	0	2	5060	rasen.tzi.org
IN SRV	0	4	50600	rasen.tzi.org

- Finally: lookup of A records for rasen.tzi.org
- Then send SIP message to 134.102.218.67

Alternative Address Resolution Schemes

- Telephony Routing for IP (TRIP) [RFC 3219]
  - BGP-4-based routing protocol to find gateways
- Hierarchical Routing
  - See H.323 Gatekeeper Hierarchy across NRENs
- Static routing
  - SIP-based IP PBXes with statically configured prefix routing
- Peer-to-peer address resolution
  - Relying on a *different* distributed data base than DNS
- …

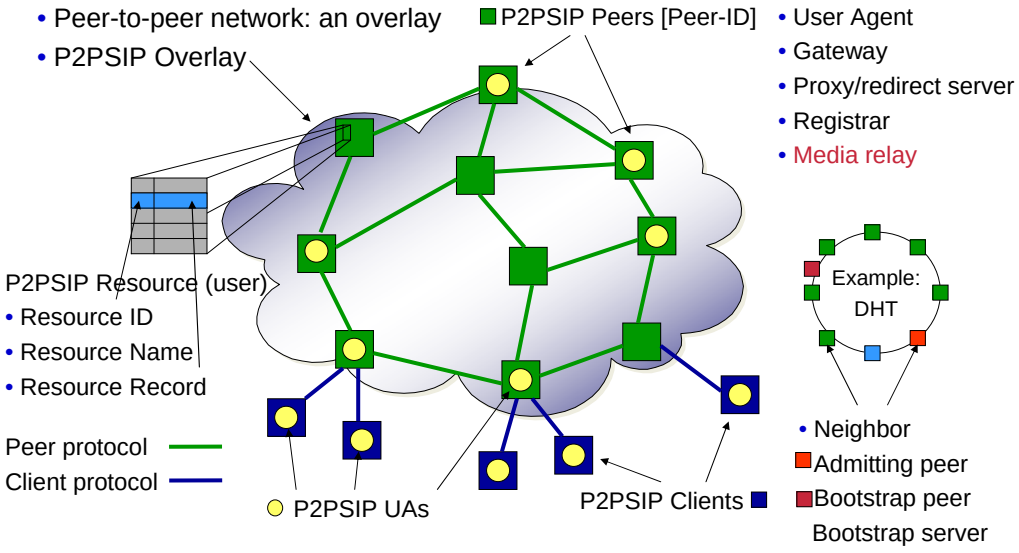
Example for Peer-to-Peer SIP



Peer-to-Peer SIP Core Functions

- Creating an Overlay of SIP nodes (P2PSIP peers)
- Mapping AoRs to a (set of) current user locations
  - Distributed database (storing and retrieving data)
  - Location service
  - Avoid the need to run servers
- Providing a relayed transport for SIP messages
- Further services
  - Services also need to be identified and located in the overlay
  - Examples
    - NAT/firewall traversal (STUN servers, relays)
    - Voicemail storage
    - Conferencing support (media server, focus)
    - Gateways (e.g., PSTN)

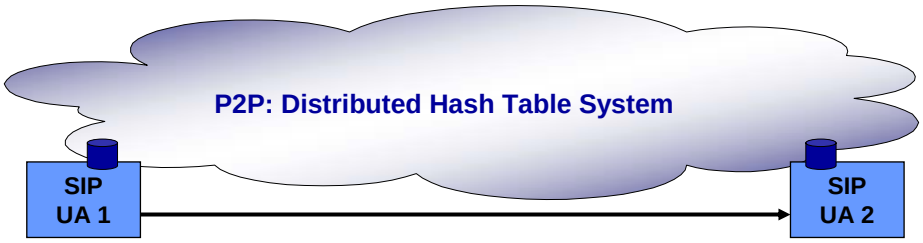
P2PSIP: Draft Definitions



P2PSIP Protocols

- Peer Protocol: between peers for distributed system maintenance
  - Enrolling a peer and inserting a peer into the overlay
  - Enrolling a user/resource and inserting a user/resource record
  - Retrieving a resource record about a user/resource
- Client Protocol: between clients and peers
  - Strict subset of the peer protocol
  - Enrolling and inserting a user/resource in the overlay
  - Retrieving a resource record about a user/resource from the overlay
- Both protocols may or may not use SIP as a basis
- SIP for signaling
  - May or may not pass through P2PSIP peers (proxy vs. redirection)
- RTP for media streams
  - May be reflected in media relays (RTP translators)

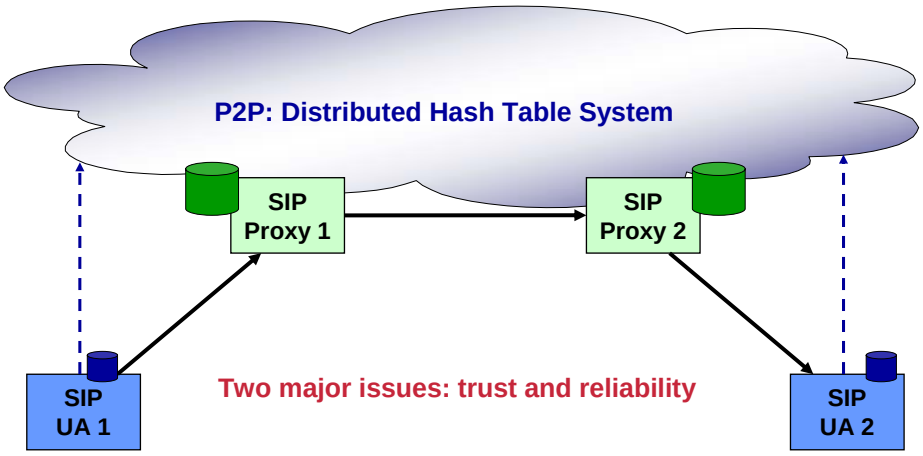
UAs as P2PSIP Peers



sip:ob@tzi.org

sip:jo@acm.org

Proxies as P2PSIP Peers, UAs a clients



sip:ob@tzi.org

sip:jo@acm.org

P2PSIP NAT Traversal Thoughts

- Approach 1: Relying on Superpeers
  - Superpeers are “elected” from the set of P2PSIP peers
  - Must be located in the public Internet (no NAT)
  - Communications between NATed peers then via superpeers
- Approach 2: All peers are equal
  - Create a partial mesh between peers
    - Assuming there will always be some publicly reachable peers to start with
  - Mesh to be organized in a structured manner (e.g., using some DHT)
  - Route messages along the edges
    - Assuming that all peers will be connected via at least one edge and are thus reachable
- Media sessions: use “traditional” SIP mechanisms (STUN, ICE)

Some P2PSIP Questions

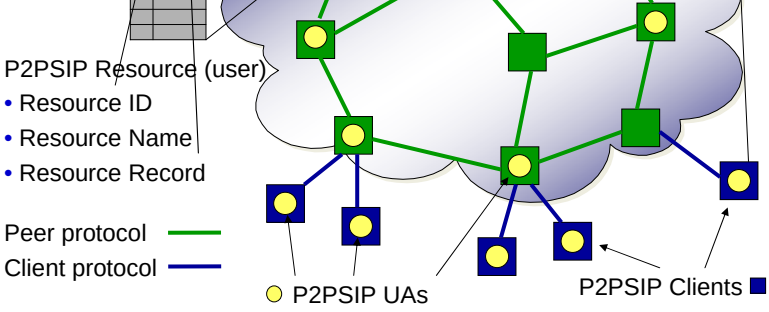
- Selecting between Multiple Peers offering the Same Service
  - And distinguishing services from users
    (do not want to lose the capability of forking)
- Visibility of Messages to Intermediate Peers
- Using C/S SIP and P2PSIP Simultaneously in a Single UA
- Clients, Peers, and Services
  - Do all peers providing routing, storage, and all other services, or do only some peers provide certain services?
  - What services, if any, must all peers provide?
  - Do we need clients as a discrete class, or do SIP UAs and/or low- function peers completely satisfy the requirements?
  - How we can we describe the capacity of a peer for delivering a given service?
- Relationships of Domains to Overlays
  - Can there be names from more than one domain in a single overlay?
  - Can there be names from one domain in more than a single overlay? If so, how do we route Client/Server SIP requests to the right overlay?
  - Can the domain of an AoR be in more than one overlay?
  - Should we have a "default overlay" to search for peers in many domains?

P2PSIP Client: A node participating in a P2PSIP Overlay that is less capable than a P2PSIP Peer in some way. The role of a P2PSIP Client is still under debate, with a number of competing proposals, and some have suggested removing the concept entirely (see the discussion on this later in the document). If clients exist, then it has been agreed that they do have the ability to add, modify, inspect, and delete information in the overlay. Note that the term client does not imply that this node is a SIP UAC. Some have suggested that the word 'client' be changed to something else to avoid both this confusion and the implication of a client- server relationship.

ons

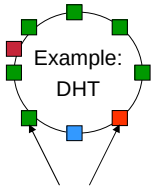
er-ID]

- User Agent
- Gateway
- Proxy/redirect server
- Registrar
- Media relay



P2PSIP Resource (user)
- Resource ID
- Resource Name
- Resource Record

Peer protocol
Client protocol



- Neighbor
- Admitting peer
- Bootstrap peer

Bootstrap server

P2PSIP UAs

P2PSIP Clients