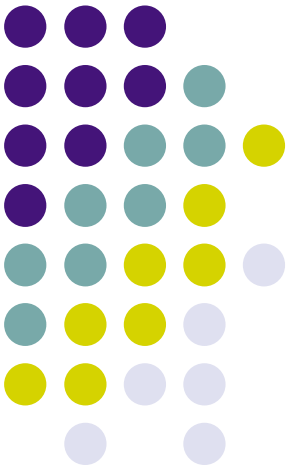
A P2P SIP Architecture
- Two Layer Approach -
draft-sipping-shim-p2p-arch-00.txt



IETF65, Dallas

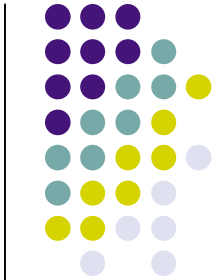
March 24, 2006

Eunsoo Shim

Sathya Narayanan

Greg Daley

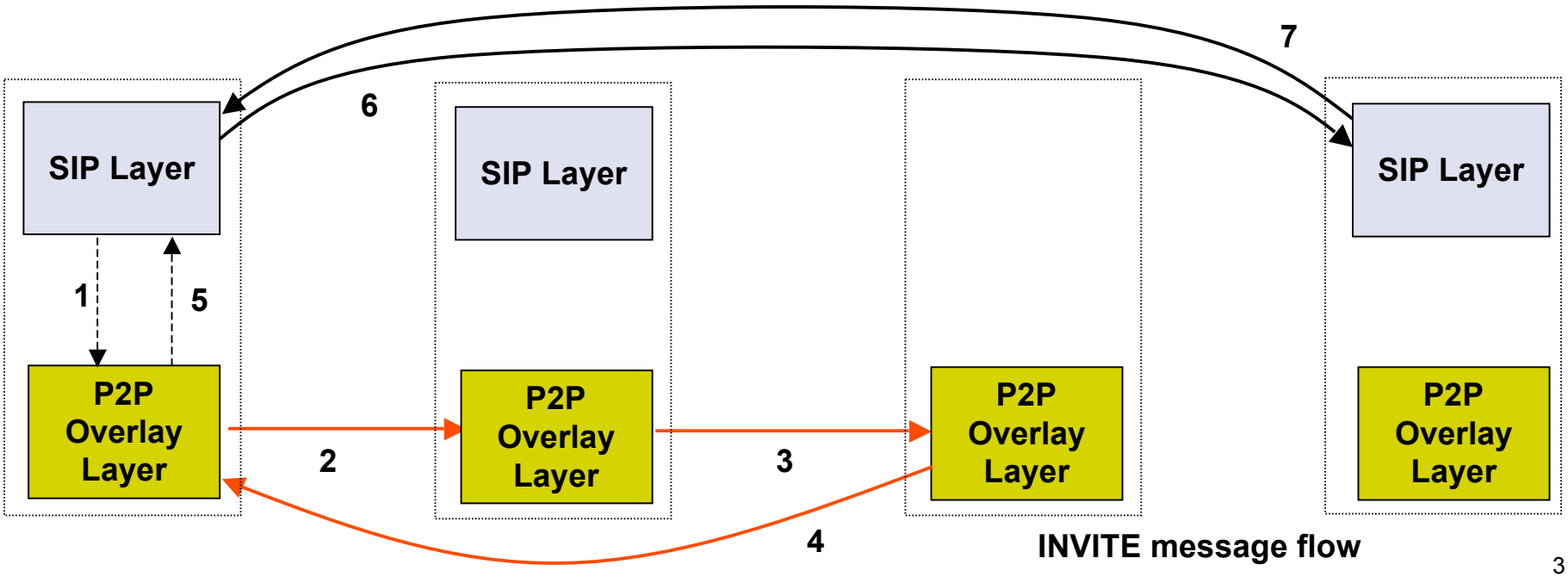
Key Aspects of the Proposed Architecture



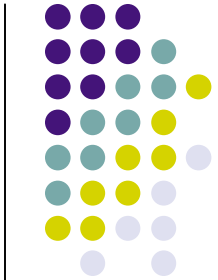
- Two Layers
  - SIP layer – call processing
  - P2P overlay layer – location service
- P2P Overlay Layer
  - Super Node, Ordinary Node
  - Authentication Server
  - Bootstrap Server
- Inter-domain call routing via Proxy
  - RFC3263

Two Layer Approach

- P2P overlay layer provides the following functions
  - P2P overlay management function - peer initiation (join), leave
  - P2P service function – placement and lookup of resources
- SIP is just an application over P2P overlay layer.
- DHT lookup messages are generic ---- independent of SIP call semantics or resource types



Why Two Layer Approach?

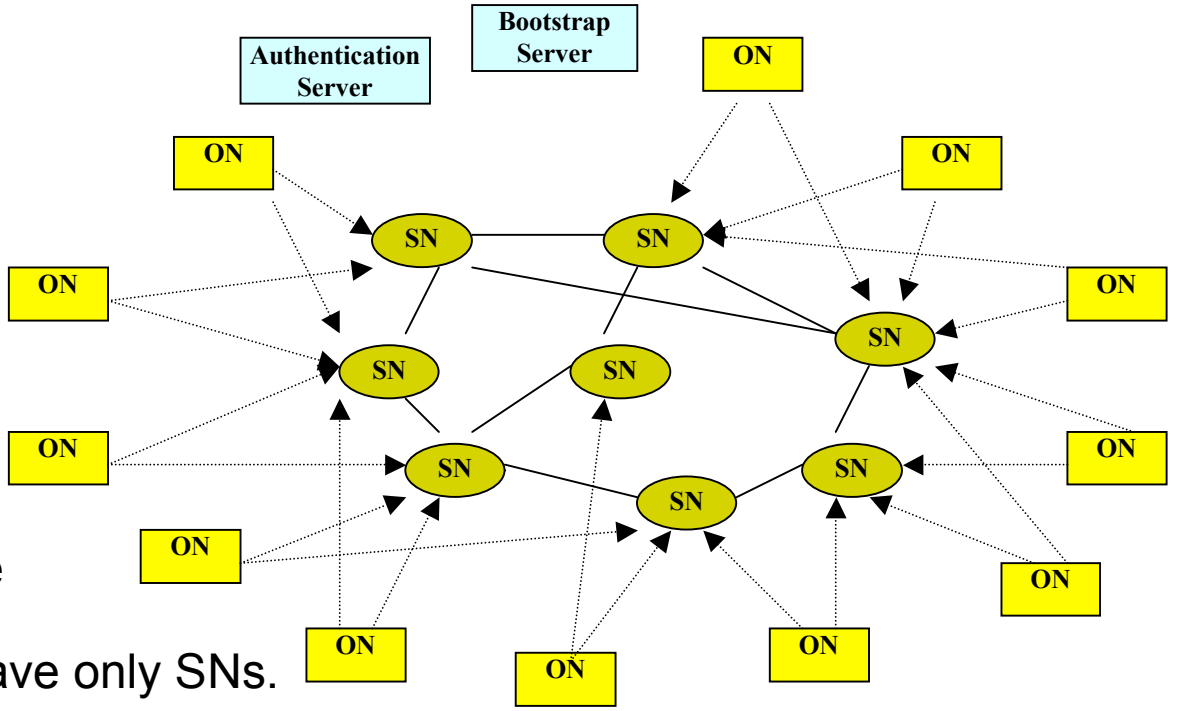


- It is the model of RFC3263
  - Location service is separate from call establishment procedures.
- Transparent interoperation with client server SIP
  - No change in SIP messages
- Clarity
  - No confusion with semantics of existing SIP messages
- Flexibility
  - Easier to support different overlay algorithms with little change in SIP messages
- Sharing the overlay network --- common lookup mechanism for many things!
  - No change in DHT operation required to support advanced features of SIP-based P2P VoIP, IM, and Presence
  - Can share the same overlay network with more applications beyond basic VoIP call or IM.
    - For example, for P2P-based conferencing later.
  - Nodes without SIP entity can participate in the overlay network.

Hierarchical Structure of P2P Overlay

- Super Node (SN): Participate in lookup routing; reachable through predefined ports and protocols by any node (most likely to have a public IP address).
- Ordinary Node (ON): Not involve in lookup routing, associate with SNs and send service requests to them.
- ON-SN hierarchy is independent of SIP UA-Proxy hierarchy.

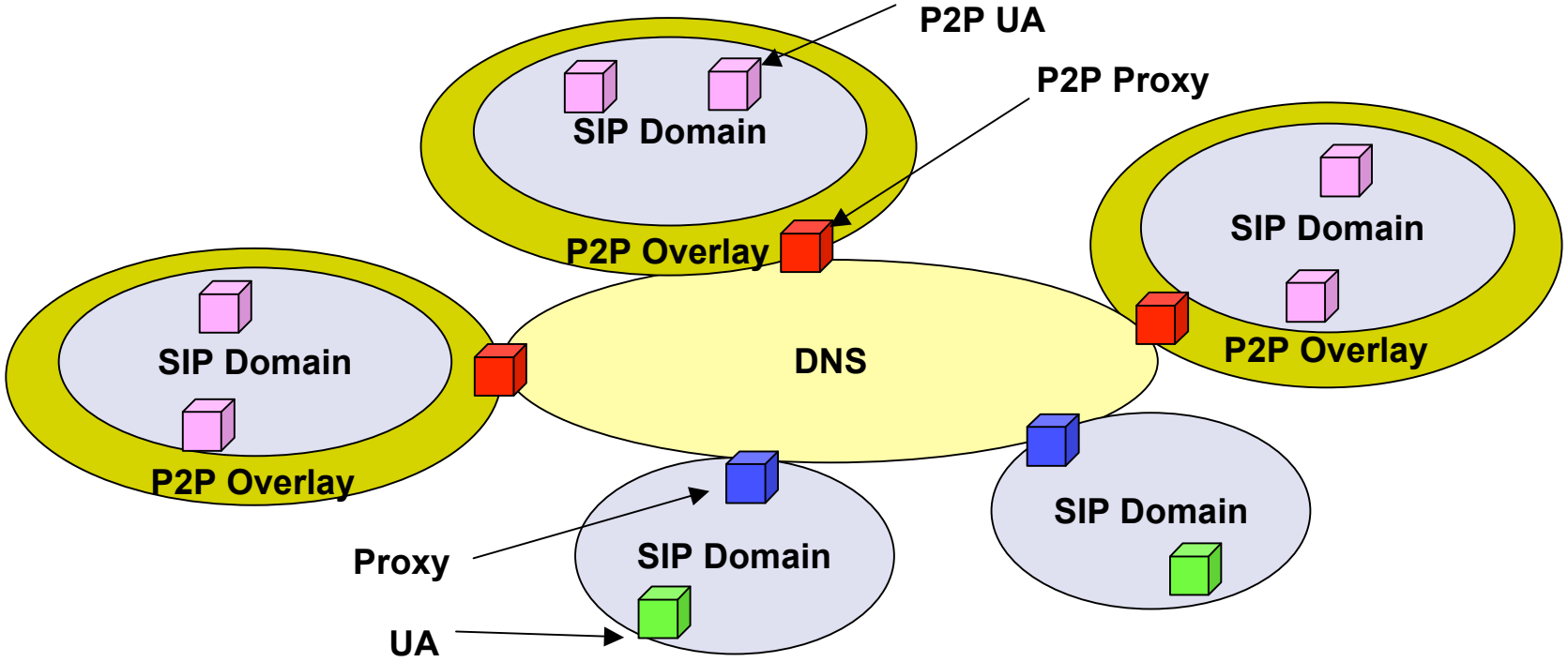
- Bootstrap Server: provides information of some of existing SNs to a new node.
- Authentication Server: authenticates user identity (password-based) and issues a certificate for user public key.



- Why Hierarchy? Peers may be heterogeneous.
- A P2P overlay network may have only SNs.

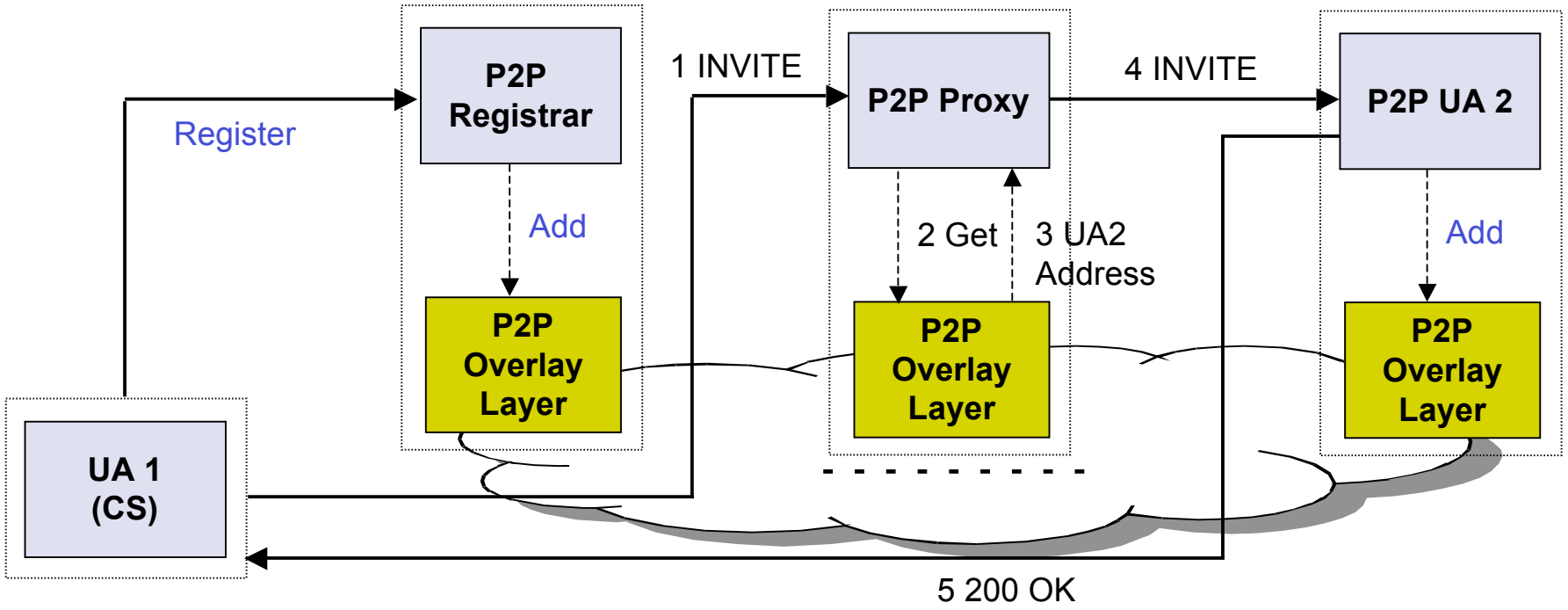
Federation of P2P SIP Networks

- Multiple independent overlay networks are allowed.
- SIP domain <=> namespace
- Single domain per overlay: A P2P overlay network supporting only one domain.
- No lookup or placement across overlay network boundaries.
- Call routing to a peer in a remote domain --- via P2P Proxies
  - Caller: alice@example.com, Callee: bob@example.net
  - The call goes through proxy of example.com and then the proxy of example.net.
  - Alice UA finds the example.com proxy via the overlay lookup.
  - The example.com proxy locates the example.net proxy via DNS.
  - The example.net proxy locates Bob UA via the overlay lookup.

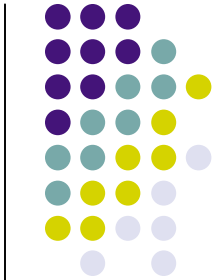


Possible P2P SIP Network Composition Scenarios

	UA	Proxy	Registrar	Role of P2P Overlay
(a)	P2P	P2P	Not Required	Replace the registrar and DNS lookup for locations of local proxies
(b)	CS	P2P	P2P	Replace the location database accessed by local proxies and the local registrar

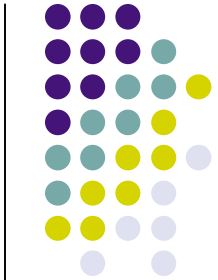


What Needs To Be Specified ?



- P2P overlay protocol
  - P2P Overlay Algorithm
  - Message syntax and state machines
  - ON <-> SN, SN <-> SN
  - ON/SN <-> Authentication Server
  - ON/SN <-> Bootstrap Server
- SIP entity behavior
  - P2P-UA Behavior
  - P2P-Proxy Behavior
  - P2P-Registrar Behavior
- Interface between SIP layer and P2P overlay layer
  - Resource records (types, formats)
    - User location, STUN/TURN server location, …
  - P2P Overlay API (semantics)

Security Considerations

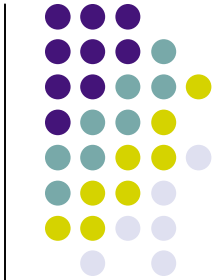


- Bootstrapping Security
  - Mutual authentication is required.
- ON-SN Authentication
  - Minimize reliance on the central login server.
- Peer Transport Security
  - Message authentication is required.
- Firewalls
  - Allow port 80, 443 as the last resort ???
- Relay for NAT Traversal
  - Defend against compromised relays by end-to-end authentication
- Registration
  - Signature for the location records
  - Privacy issue
- Authentication when central servers are not reachable
- DoS attacks
  - Defend excessive overlay traffic generation by rate limiting
- Ill behaviors of SNs
  - Messing up DHT tables
- Free riders
  - Refusing or avoiding to be a SN
- And so on ….

Peer Initiation

- Any peer starts as an ordinary node (ON).
1. Bootstrap – discovering peers in the overlay
   - Service location (multicast)
   - Cached addresses
   - Last good addresses
   - Preconfigured bootstrap server
2. ON-SN Association
   - Contacting SN – UDP, TCP, Fallback Transport
   - Mutual return reachability test
3. Authentication
   - If ON does not have a certificate, contact the login server and receives the certificate.
   - ON-SN mutual authentication
4. NAT/FW Traversal
   - Create address bindings for inbound SIP messages
   - ICE (STUN, TURN)

Post Initiation Tasks

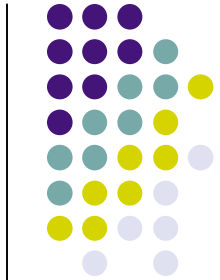


- Registration
  - Publish contacts (tuples of transport protocol, IP address, port) in the overlay
- Becoming a Super Node
  - Self-selected dynamically and automatically.
  - MUST be able to receive overlay messages from other SNs on predetermined protocols and ports.
  - SHOULD be online stably.
  - SHOULD have sufficient physical resources.

P2P Overlay API

- *get*(in overlay_id, in *name,* out *records,* out *error*)
- *add*(in overlay_id, in *name*, in *record,* in *lifetime,* in *option,* out *error*)
- *update*(in overlay_id, in *name*, in *record,* in *lifetime,* in *option,* out *error*)
- *remove*(in overlay_id, in *name,* out *error*)

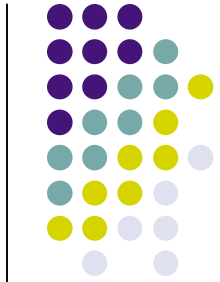
Resource Record – User Location



```
<resource>                                          ------ header of a resource
<version>1.0</version>                              ------ resource format version
<type>user location</type>                          ------ type of the resource
<key>19873761ab24</key>                             ------ key for the resource
<lifetime> 3600 </lifetime>                         ------ lifetime of the record
<timestamp>19809832142</timestamp>                  ------ indicate which is more recent
<user_URI> user@example.com </user_URI>
<location>
  <node_IP>178.14.234.21</node_IP>                  --- the IP address at which the user can be
    reached
  <transport>TCP5060 UDP5060 TCP80 TCP443</transport>  --- the list of ports the UA is
    listening to
</location>
<location>
  <node_IP>192.168.0.100</node_IP>                  --- the IP address at which the user can be
    reached
  <transport>TCP5060 UDP5060 TCP80 TCP443</transport>  --- the list of ports the UA is
    listening to
</location>
</resource>
```

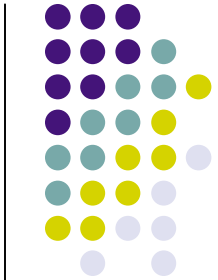
- Resource name: <type> + <user_URI>
- Resource key = hash(resource name)

Resource Record – Proxy Location



```
<resource>                                  ------ header of a resource
<version>1.0</version>                      ------ resource format version
<type>proxy location</type>                 ------ type of the resource
<key>19873761ab24</key>                     ------ key for the resource
<lifetime> 36000 </lifetime>                ------ lifetime of the record
<timestamp>198023422</timestamp>            ------ indicate which is more recent
<domain> example.com </domain>
<location>
  <node_IP>178.14.234.21</node_IP>          --- the IP address at which the user can be
    reached
  <transport>TCP5060 UDP5060 TCP80 TCP443</transport>   --- the list of ports the proxy is
    listening to
</location>
<location>
  <node_IP>192.168.0.100</node_IP>          --- the IP address at which the user can be
    reached
  <transport>TCP5060 UDP5060 TCP80 TCP443</transport>   --- the list of ports the proxy is
    listening to
</location>
</resource>
```

- Resource name: <type> + <domain>
- Resource key = hash(resource name)



Acknowledgements

- Henning Schulzrinne
- Salman Baset