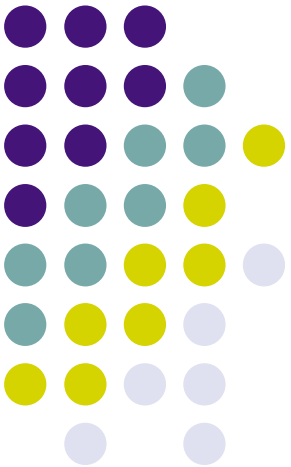
A P2P SIP Architecture
- Two Layer Approach -
draft-sipping-shim-p2p-arch-00.txt



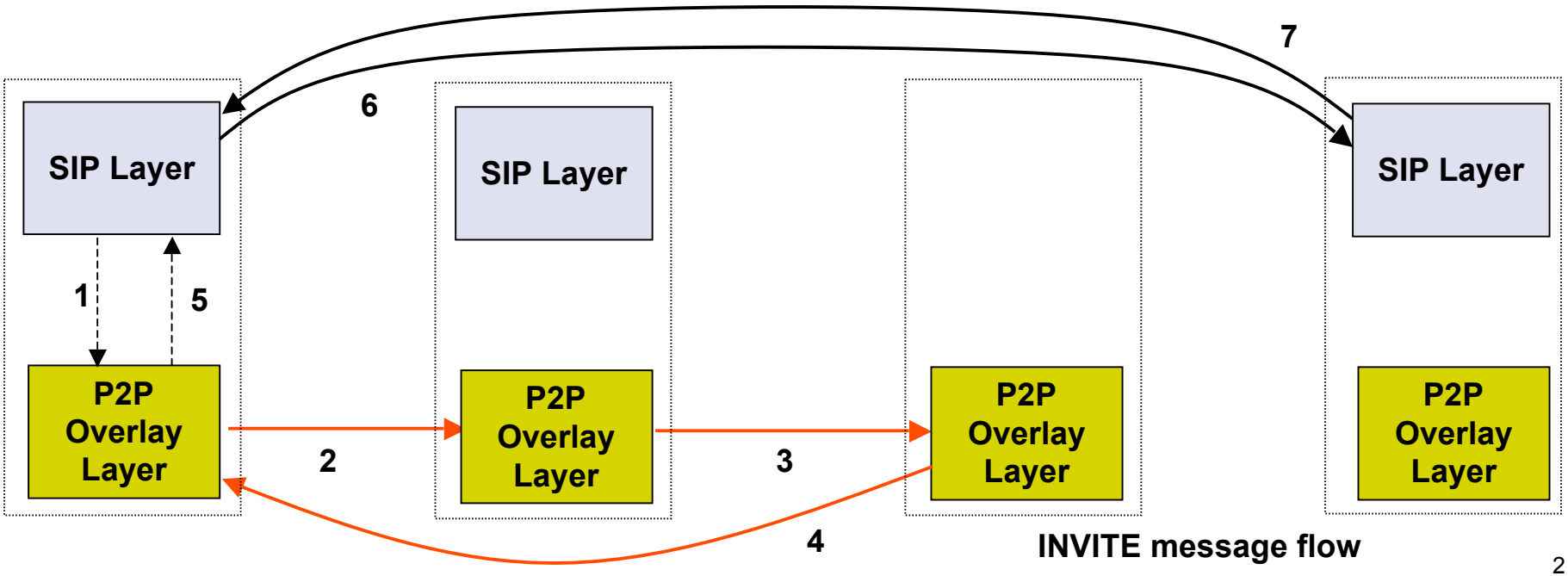
IETF65, Dallas
March 25, 2006

Eunsoo Shim
Sathya Narayanan
Greg Daley

Panasonic Digital Networking Laboratory

Two Layer Approach

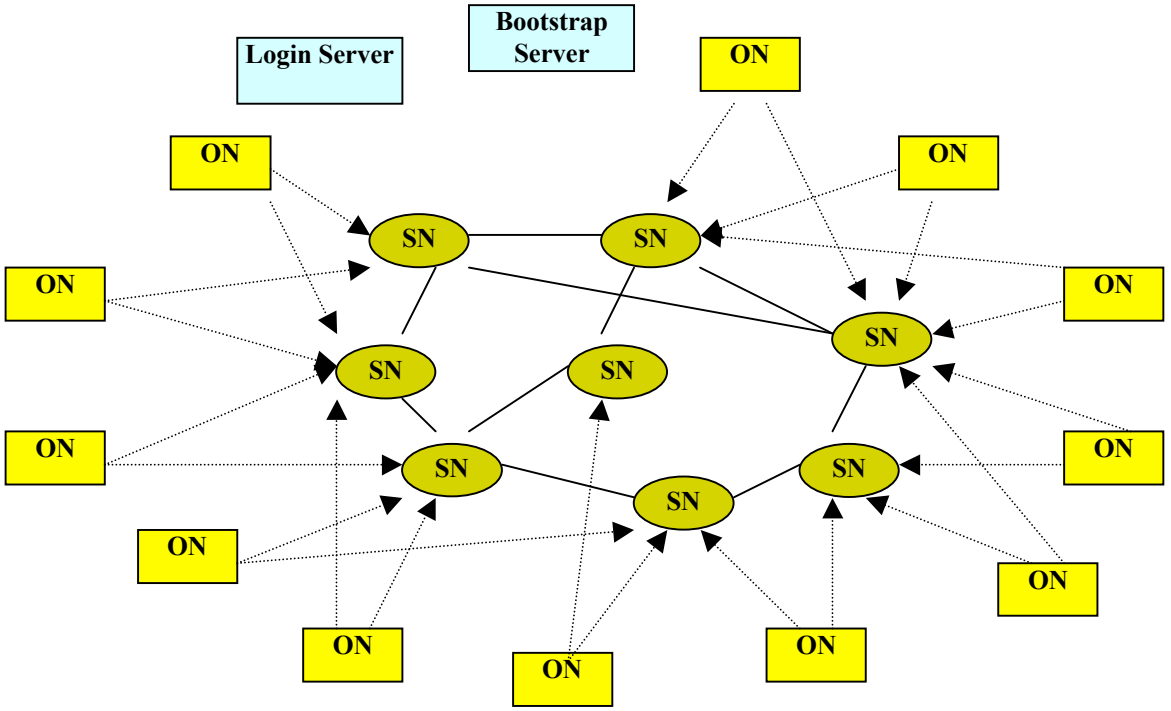
- P2P overlay layer provides the following functions
  - P2P overlay management function - peer initiation (join), leave
  - P2P service function – placement and lookup of resources
- SIP is just an application over P2P overlay layer.
- DHT lookup messages are generic ---- independent of SIP call semantics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



Hierarchical Structure of P2P Overlay

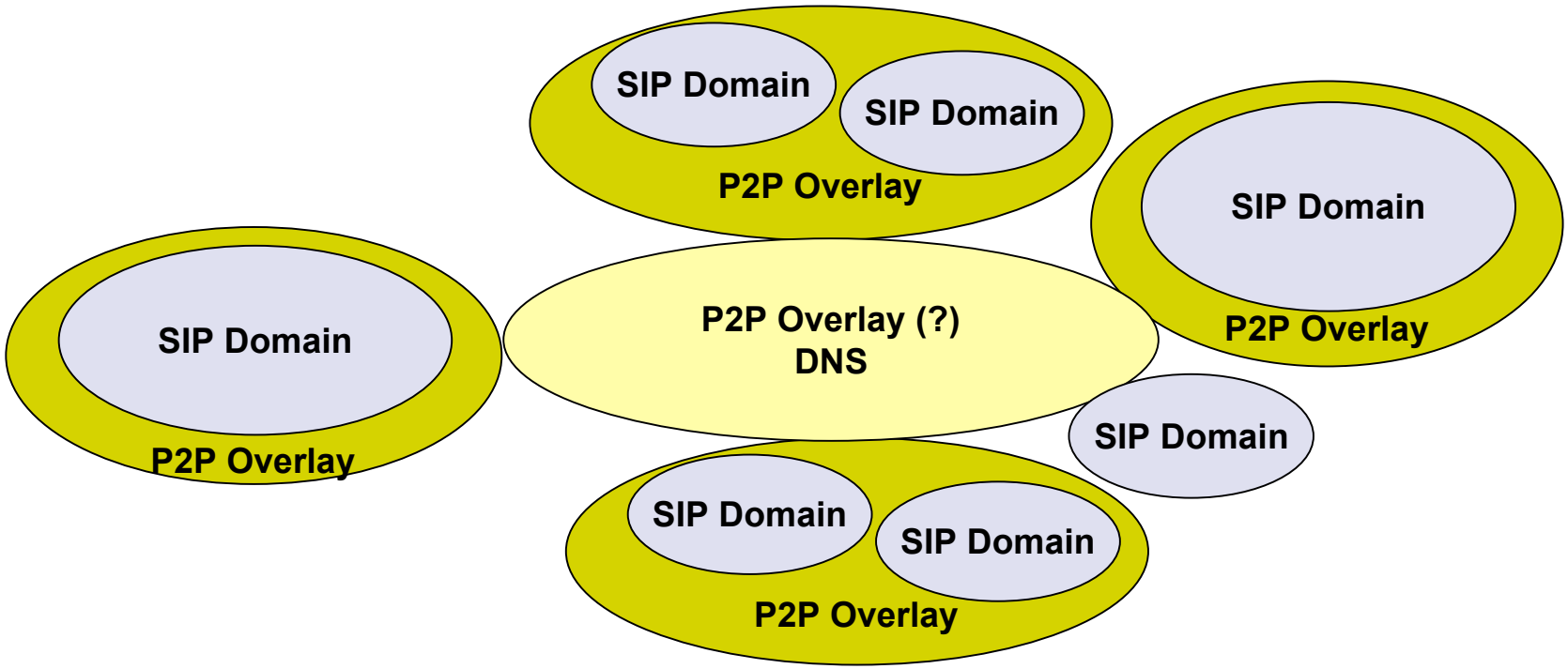
- Super Node (SN): Participate in DHT; reachable through predefined ports and protocols by any node (most likely to have a public IP address).
- Ordinary Node (ON): Not in DHT, associate with SNs and send service requests to them.
  - A P2P overlay network may have only SNs.
  - ON-SN hierarchy is independent of SIP UA-Proxy hierarchy.

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



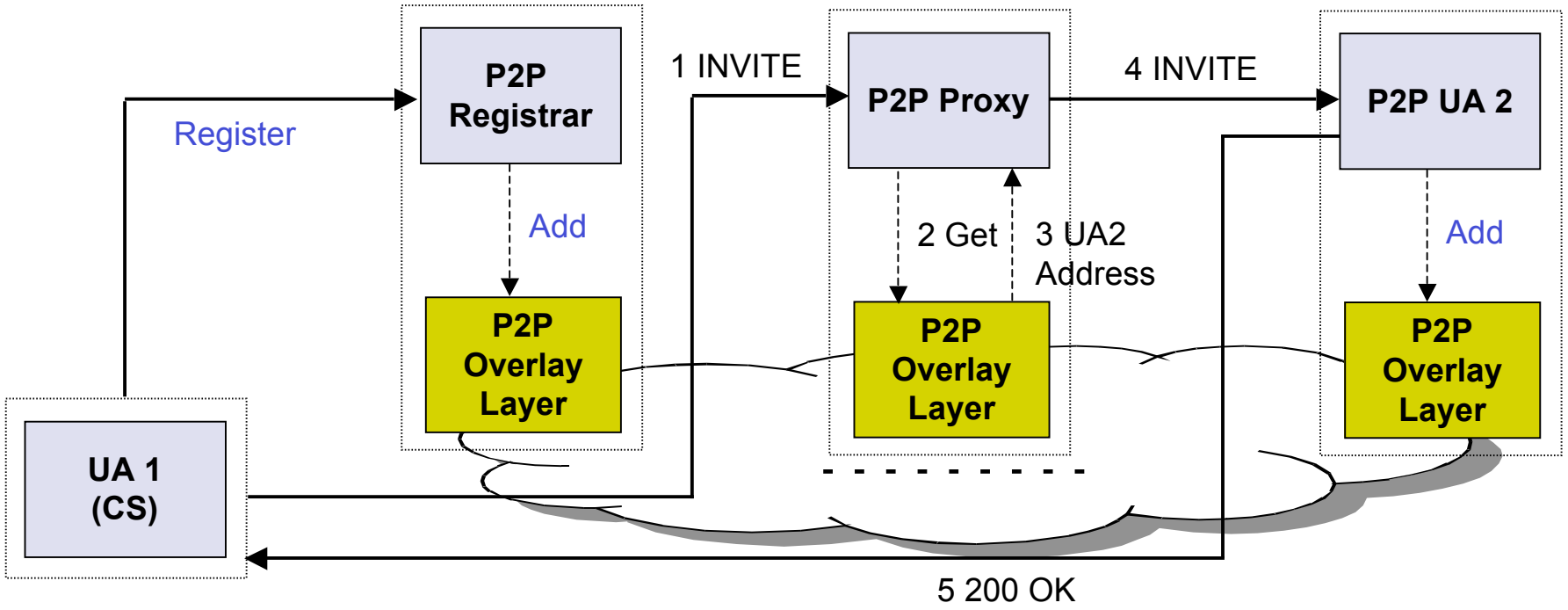
Federation of P2P SIP Networks

- Multiple independent overlay networks are allowed.
  - Overlay identifier ≠ domain name
- Single-domain overlay: A P2P overlay network supporting only one domain.
- Multi-domain overlay: A P2P overlay network supporting multiple domains.
- No lookup or placement across overlay network boundaries (?)
  - Call routing to a peer in a remote overlay network --- via P2P Proxies



Possible P2P SIP Network Composition Scenarios

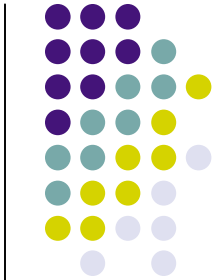
	UA	Proxy	Registrar	Role of P2P Overlay
(a)	P2P	P2P	Not Required	Replace the registrar and DNS lookup for locations of local proxies
(b)	CS	P2P	P2P	Replace the location database accessed by local proxies and the local registrar



Peer Initiation

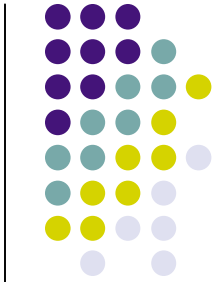
- Any peer starts as an ordinary node (ON).
1. Bootstrap – discovering peers in the overlay
   - Service location (multicast)
   - Cached addresses
   - Last good addresses
   - Preconfigured bootstrap server
2. ON-SN Association
   - Contacting SN – UDP, TCP, Fallback Transport
   - Mutual return reachability 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



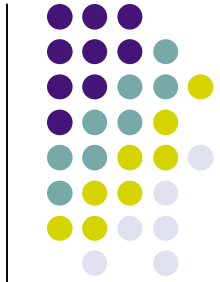
- Registration
  - Publish contacts (tuples of transport protocol, IP address, port) in the overlay
- Becoming a Super Node
  - Self-selected dynamically and automatically.
  - MUST be able to receive overlay messages from other SNs on predetermined protocols and ports.
  - SHOULD be online stably.
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

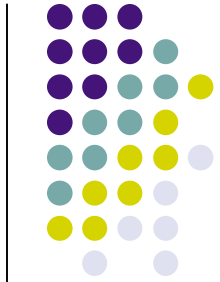
Resource Record – User Location



```
<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)

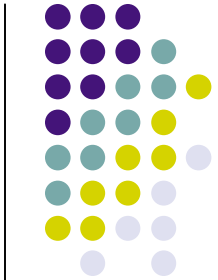
Resource Record – Proxy Location



```
<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>
```

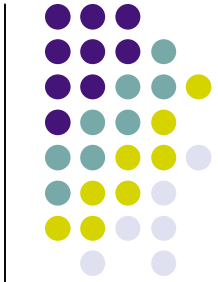
- Resource name: <type> + <domain>
- Resource key = hash(resource name)

Security Considerations



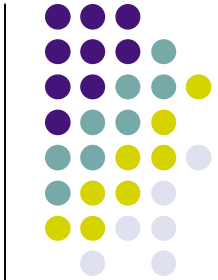
- Bootstrapping Security
  - Mutual authentication is required.
- ON-SN Authentication
  - Minimize reliance on the central login server.
- Peer Transport Security
  - Message authentication is required.
- Firewalls
  - Allow port 80, 443 as the last resort ???
- Relay for NAT Traversal
  - Defend against 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 ….

Pros & Cons of the Two Layer Approach



- Pros
  - Transparent interoperation with client server SIP
  - 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.
- Cons
  - No optimization of search and overlay maintenance by using SIP call semantics
- Challenge
  - Defining a new protocol
    - Can we reuse any existing protocols?

What Needs To Be Specified ?



- P2P overlay protocol
  - P2P Overlay Algorithm
  - Message syntax and state machines
  - ON <-> SN, SN <-> SN
  - ON/SN <-> Login 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)
  - P2P Overlay API (semantics)

Acknowledgements

- Henning Schulzrinne
- Salman Baset