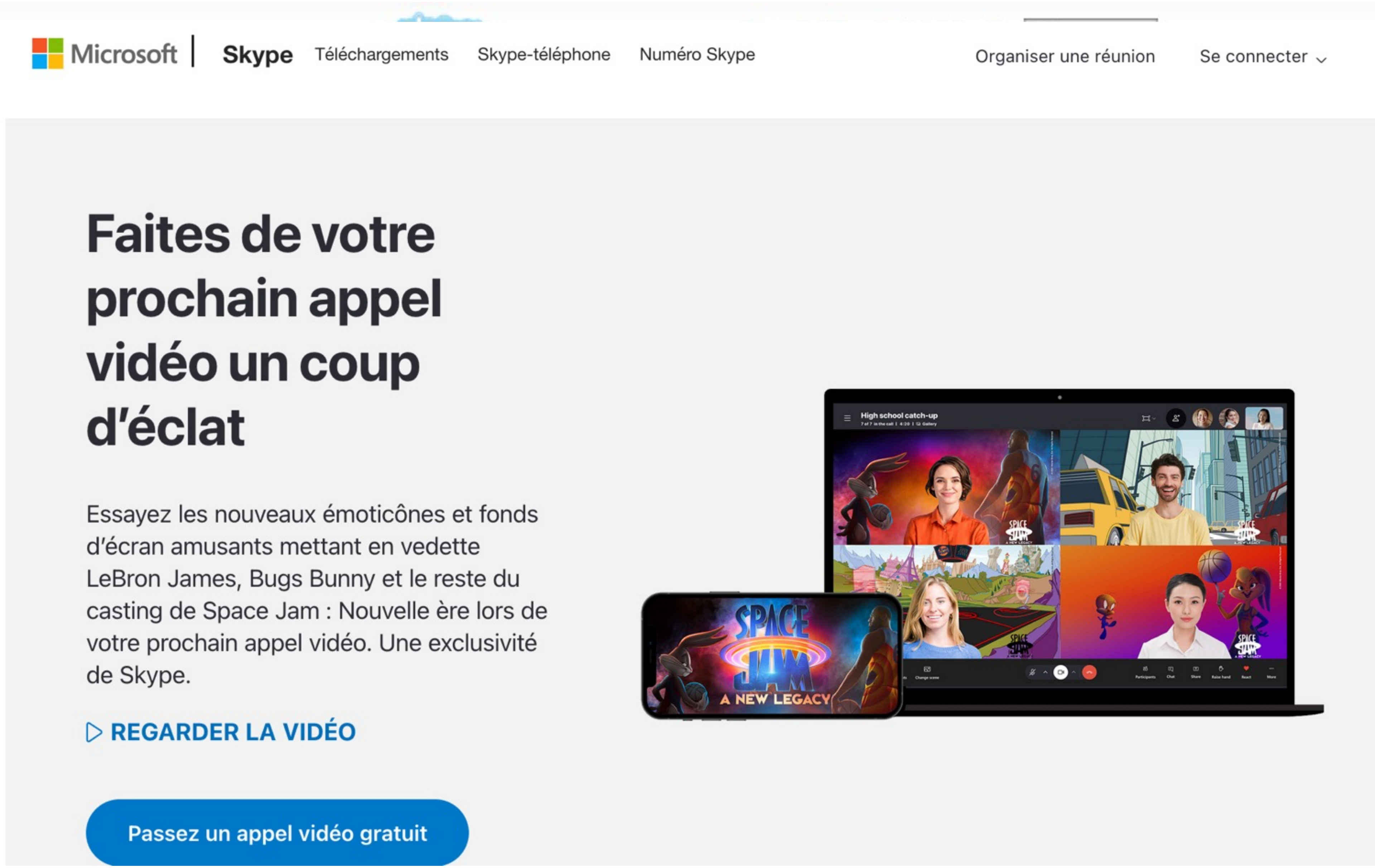


Faites de votre prochain appel vidéo un coup d'éclat

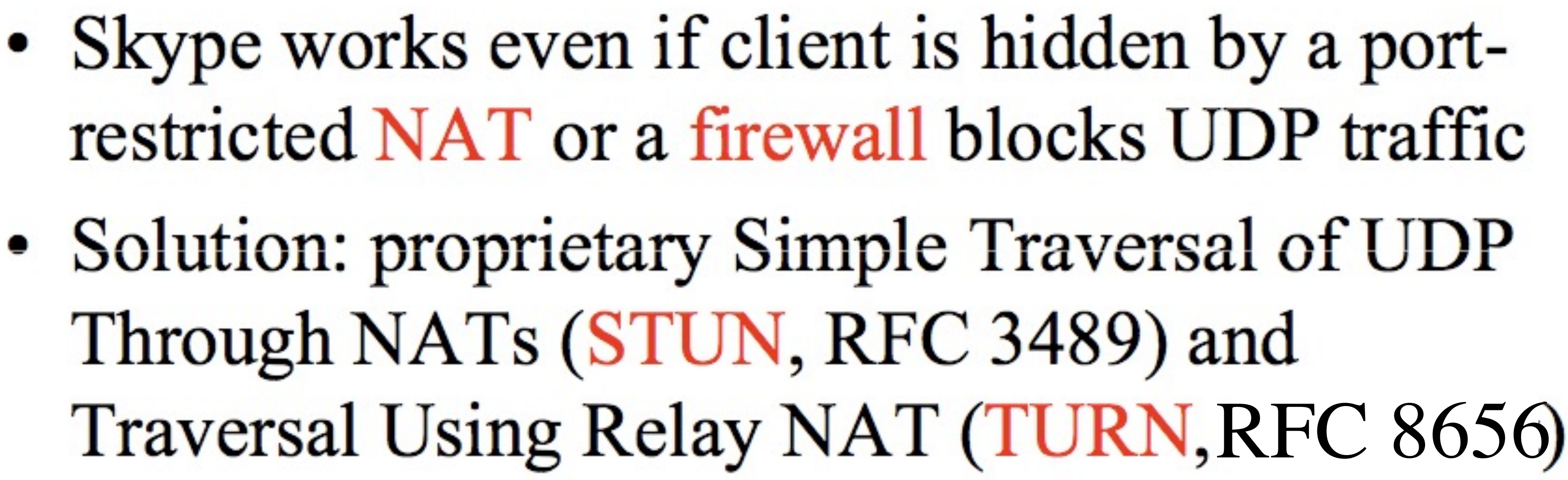
Essayez les nouveaux émoticônes et fonds d'écran amusants mettant en vedette LeBron James, Bugs Bunny et le reste du casting de Space Jam : Nouvelle ère lors de votre prochain appel vidéo. Une exclusivité de Skype.

What is Skype (before Microsoft) ?

- *Peer-to-peer inspired*, pc-to-pc, pc-to-phone, phone-to-pc VoIP and IM client (now also the engine of Microsoft TEAM)
- Developed by people who created KaZaa P2P system
- *SkypeOut* (pc-to-phone)
  - SkypeOut terms of service: governed by the laws of Luxembourg
- *SkypeIn* (phone-to-pc), voicemail
- Supported OS: ALL
- Nowadays, Skype is a "*p2p illusion*"
  - Login *server*
  - Buddy-list *server*
  - *Servers* for SkypeOut and SkypeIn
- When you have $$$ you don't need P2P technology ... just buy !

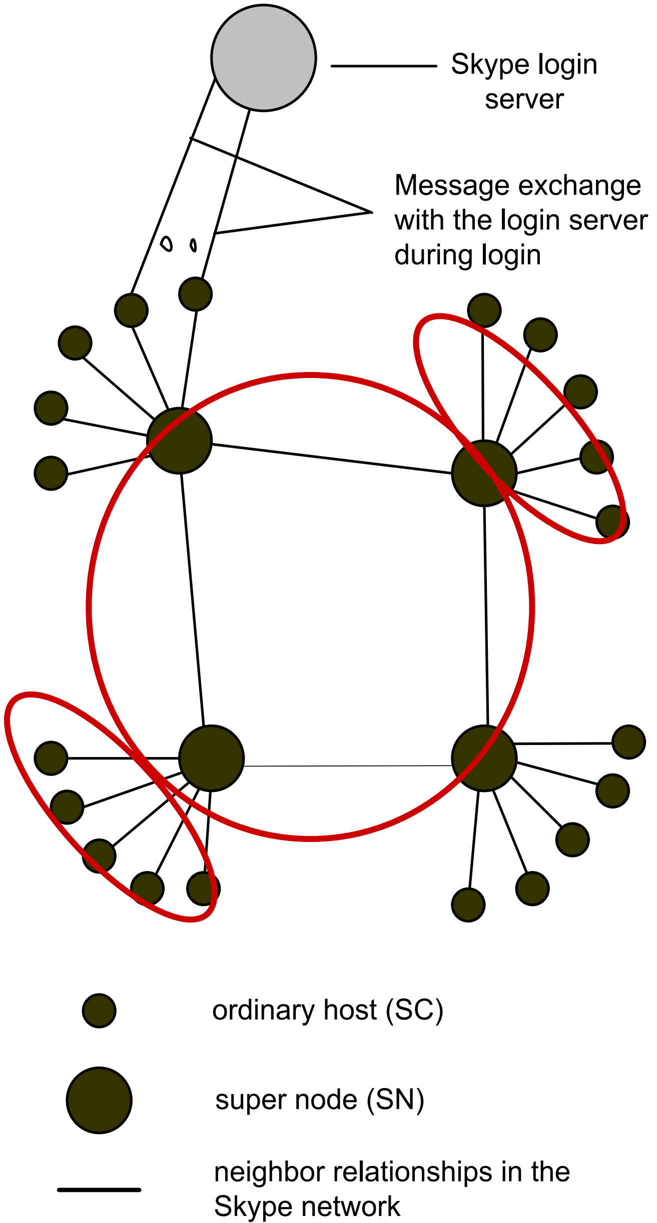
- Skype offers voice, video, chat and data transfer services over IP
- Closed design, proprietary solutions
  - Proprietary protocols, no standard
  - Communications are encrypted
  - Headers are obfuscated
  - Many anti-debugging tricks
  - A product that works well for free (beer) ?! From a company not involved on Open Source ?!
    - Is there something to hide ?
    - Impossible to scan for trojan/backdoor/malware inclusion

- Peer-to-Peer design
  - Control information is distributed among nodes
  - Data flows may run through peer nodes
- Uses both TCP and UDP
  - For signaling
  - To carry information
- Every client may become a node in the P2P overlay
- Can work even behind NAT and Firewalls



- Skype works even if client is hidden by a port-restricted NAT or a firewall blocks UDP traffic
- Solution: proprietary Simple Traversal of UDP Through NATs (STUN, RFC 3489) and Traversal Using Relay NAT (TURN,RFC 8656)

The Skype Network (2-tier)

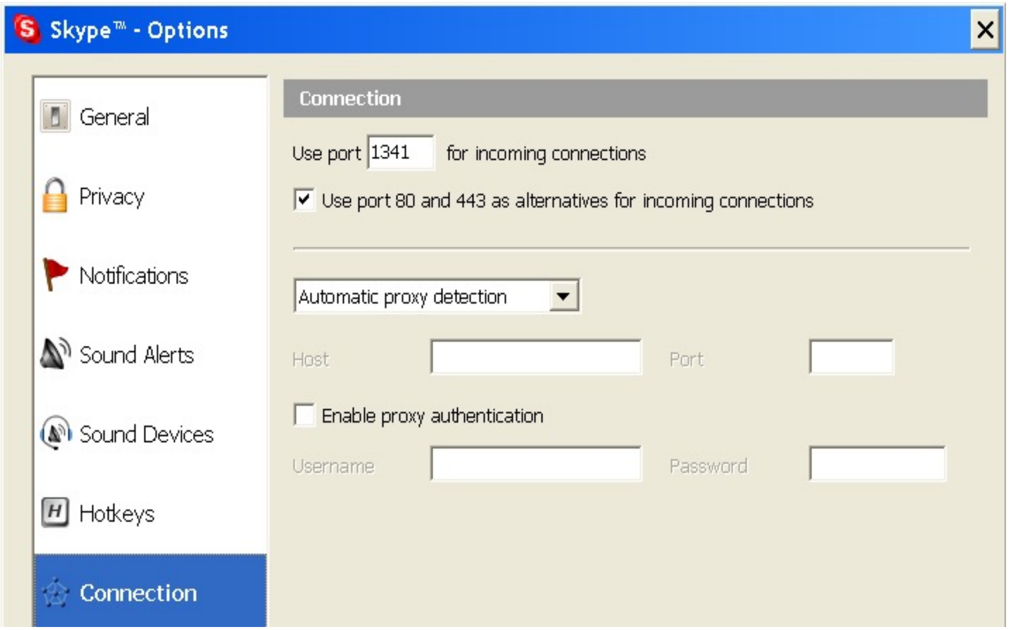

Skype login server
Message exchange with the login server during login
ordinary host (SC)
super node (SN)
neighbor relationships in the Skype network

The Skype Network hosts and servers

- Ordinary hosts (aka Skype Clients, SC)
  - A Skype client, aka YOUR SKYPE CLIENT
- Super nodes (aka Skype Super Nodes, SN)
  - A Skype client too
  - Has public IP address, ‘sufficient’ bandwidth, CPU and memory
- Login server
  - Stores Skype id’s, passwords, and buddy lists
  - Used at login for authentication
  - (Version 1.4.0.84: 212.72.49.141 and 195.215.8.141, now unreachable in 2021 ☺)

Skype Components

- Ports
  - No default listening port
  - Randomly chooses a port P on installation
  - Opens TCP and UDP listener sockets at P
  - Opens TCP listener sockets at port 80 (HTTP) and port 443 (HTTPS)



Connection

Use port 1341 for incoming connections

☑ Use port 80 and 443 as alternatives for incoming connections

Automatic proxy detection

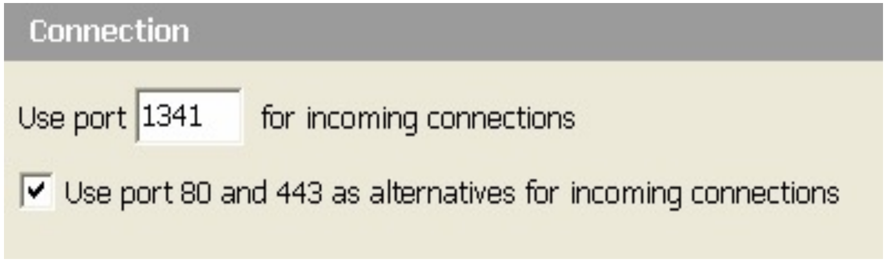
Host

Port

□ Enable proxy authentication

Username

Password



Connection

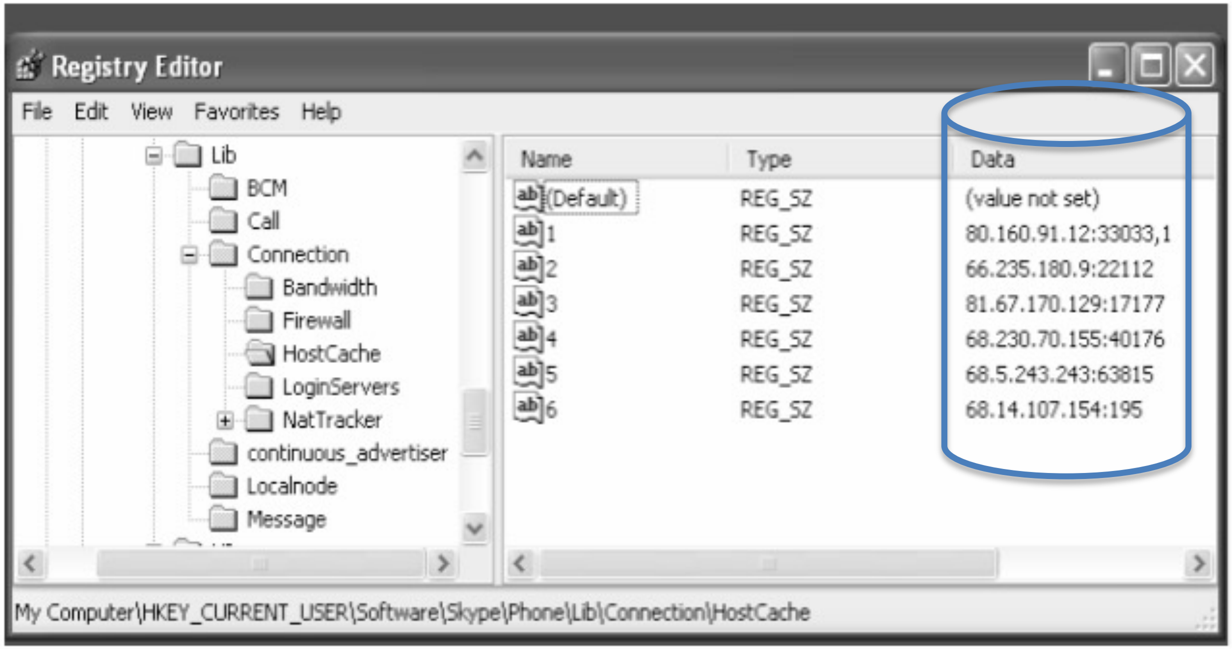
Use port 1341 for incoming connections

☑ Use port 80 and 443 as alternatives for incoming connections

Skype Components (contd…)

- Host Cache (HC) (kind of *finger table / bucket list*)
  - IP address and port number of **online** Skype super nodes (SNs)
  - Maximum size: 200 entries
  - Login server IP address and port number
  - Host Caches in an Windows 7 skype installation ☺

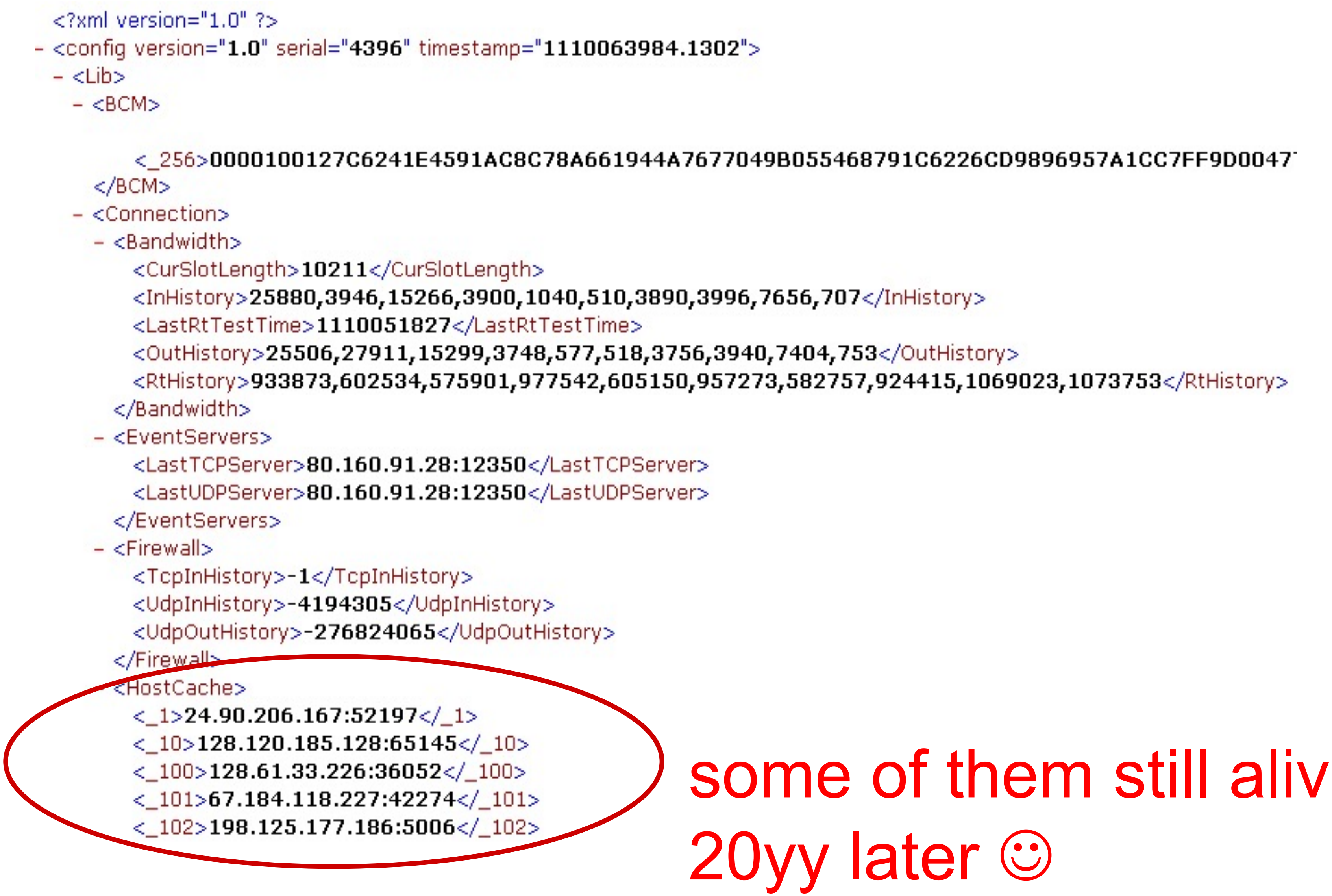
    C:\Documents and Settings\All Users\Application Data\Skype



Name	Type	Data
(Default)	REG_SZ	(value not set)
1	REG_SZ	80.160.91.12:33033,1
2	REG_SZ	66.235.180.9:22112
3	REG_SZ	81.67.170.129:17177
4	REG_SZ	68.230.70.155:40176
5	REG_SZ	68.5.243.243:63815
6	REG_SZ	68.14.107.154:195

My Computer\HKEY_CURRENT_USER\Software\Skype\Phone\Lib\Connection\HostCache

Skype Host Caches (shared.xml)



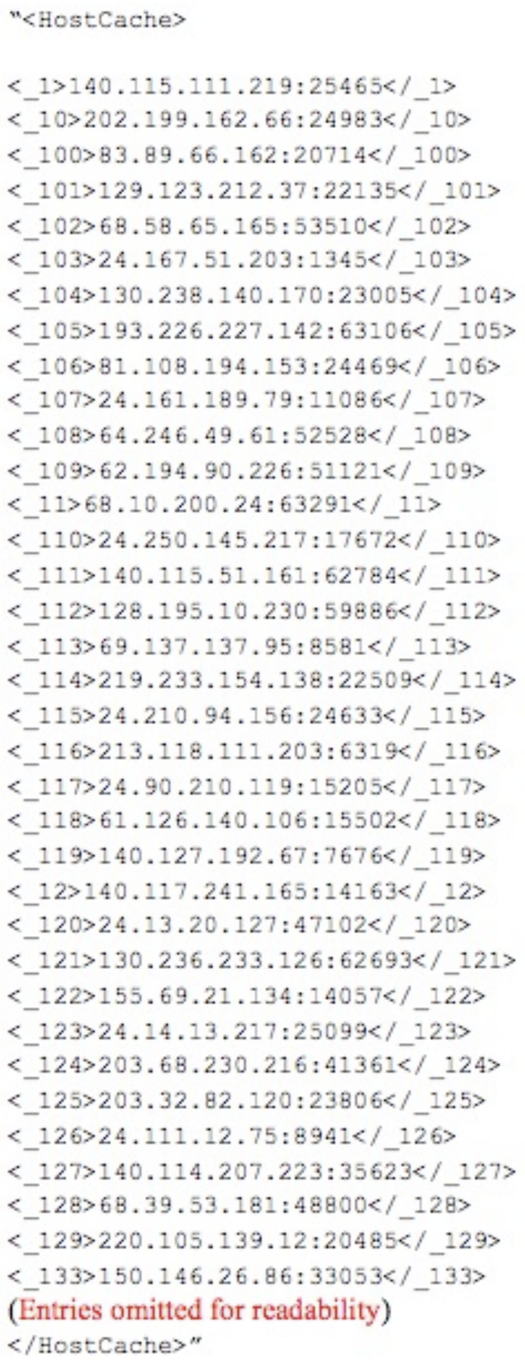
```
  <?xml version="1.0" ?>
- <config version="1.0" serial="4396" timestamp="1110063984.1302">
  - <Lib>
    - <BCM>

          <_256>0000100127C6241E4591AC8C78A661944A7677049B055468791C6226CD9896957A1CC7FF9D0047
      </BCM>
    - <Connection>
      - <Bandwidth>
          <CurSlotLength>10211</CurSlotLength>
          <InHistory>25880,3946,15266,3900,1040,510,3890,3996,7656,707</InHistory>
          <LastRtTestTime>1110051827</LastRtTestTime>
          <OutHistory>25506,27911,15299,3748,577,518,3756,3940,7404,753</OutHistory>
          <RtHistory>933873,602534,575901,977542,605150,957273,582757,924415,1069023,1073753</RtHistory>
        </Bandwidth>
      - <EventServers>
          <LastTCPServer>80.160.91.28:12350</LastTCPServer>
          <LastUDPServer>80.160.91.28:12350</LastUDPServer>
        </EventServers>
      - <Firewall>
          <TcpInHistory>-1</TcpInHistory>
          <UdpInHistory>-4194305</UdpInHistory>
          <UdpOutHistory>-276824065</UdpOutHistory>
        </Firewall>
      - <HostCache>
          <_1>24.90.206.167:52197</_1>
          <_10>128.120.185.128:65145</_10>
          <_100>128.61.33.226:36052</_100>
          <_101>67.184.118.227:42274</_101>
          <_102>198.125.177.186:5006</_102>
```

some of them still alive 20yy later ☺

only P2P feature

(present at the beginning of the Skype life):



```
"<HostCache>

<_1>140.115.111.219:25465</_1>
<_10>202.199.162.66:24983</_10>
<_100>83.89.66.162:20714</_100>
<_101>129.123.212.37:22135</_101>
<_102>68.58.65.165:53510</_102>
<_103>24.167.51.203:1345</_103>
<_104>130.238.140.170:23005</_104>
<_105>193.226.227.142:63106</_105>
<_106>81.108.194.153:24469</_106>
<_107>24.161.189.79:11086</_107>
<_108>64.246.49.61:52528</_108>
<_109>62.194.90.226:51121</_109>
<_11>68.10.200.24:63291</_11>
<_110>24.250.145.217:17672</_110>
<_111>140.115.51.161:62784</_111>
<_112>128.195.10.230:59886</_112>
<_113>69.137.137.95:8581</_113>
<_114>219.233.154.138:22509</_114>
<_115>24.210.94.156:24633</_115>
<_116>213.118.111.203:6319</_116>
<_117>24.90.210.119:15205</_117>
<_118>61.126.140.106:15502</_118>
<_119>140.127.192.67:7676</_119>
<_12>140.117.241.165:14163</_12>
<_120>24.13.20.127:47102</_120>
<_121>130.236.233.126:62693</_121>
<_122>155.69.21.134:14057</_122>
<_123>24.14.13.217:25099</_123>
<_124>203.68.230.216:41361</_124>
<_125>203.32.82.120:23806</_125>
<_126>24.111.12.75:8941</_126>
<_127>140.114.207.223:35623</_127>
<_128>68.39.53.181:48800</_128>
<_129>220.105.139.12:20485</_129>
<_133>150.146.26.86:33053</_133>
(Entries omitted for readability)
</HostCache>"
```

Your Skype Client could enter in the HOST CACHE of many others Skype Clients if you are measured as capable to host a Skype Super Node

This was a huge problem for our *Security System Administrators*

Skype Components (Continued)

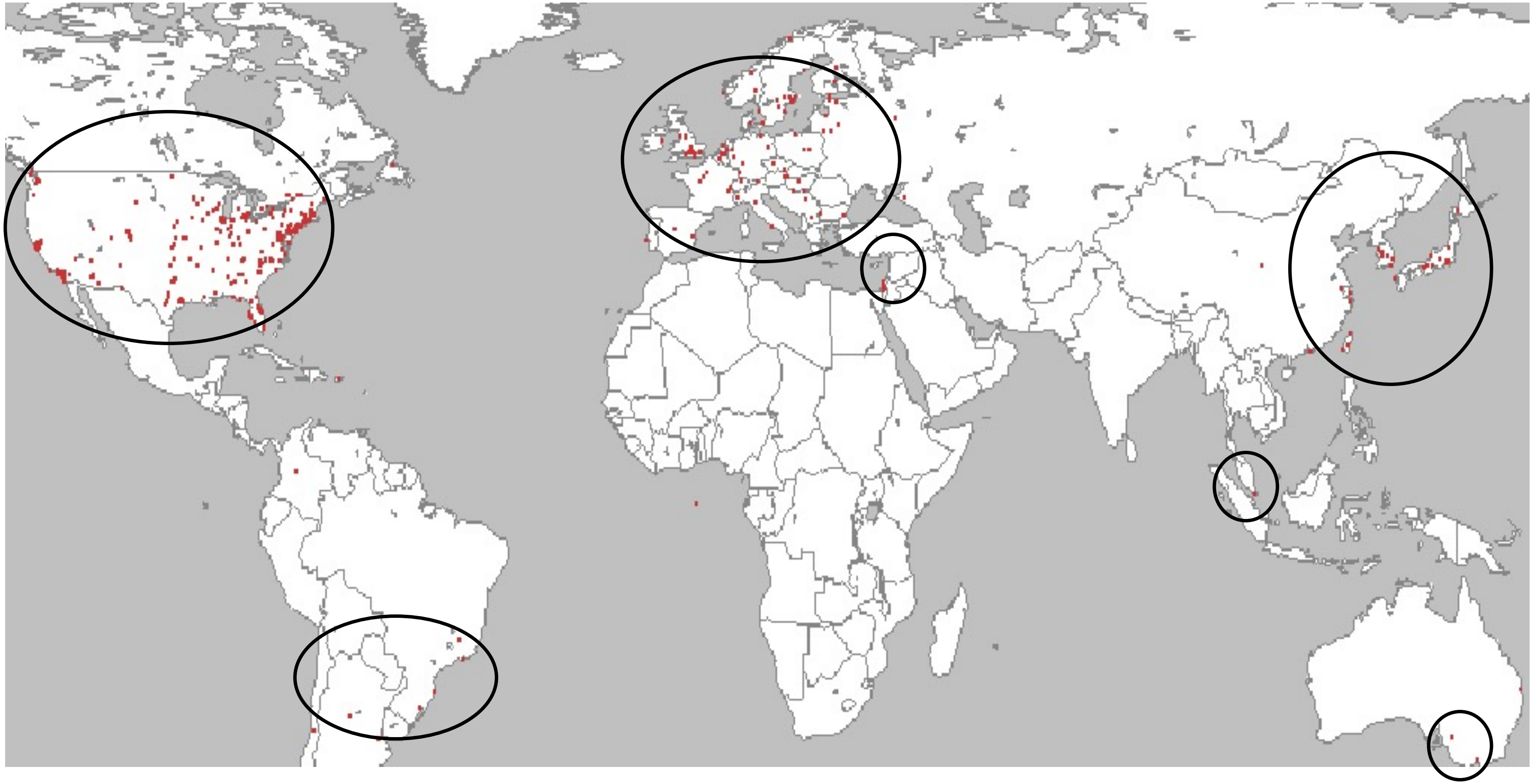
- Codecs (all Global IPSound plus VOIP standard G.729)
  - Wide band codecs (50-8,000 Hz)
  - iLBC (packet size: 20 and 30 ms bitrate: 15.2 kbps and 13.3 kbps)
  - iSAC (packet size: 30-60 ms bitrate: 10-32 kbps)
  - G.729 for SkypeOut?
- Buddy list
  - Stored in ‘config.xml’ file, in Windows 7
    - C:\Documents and Settings\<XP user>\Application Data\Skype\<skype user id>

```
<CentralStorage>
<LastBackoff>0</LastBackoff>
<LastFailure>0</LastFailure>
<LastSync>1120325519</LastSync>
<NeedSync>0</NeedSync>
<SyncSet>
<u>
<skypebuddy1>f384d3a0:1</skypebuddy1>
<skypebuddy2>7d1dafc4:1</skypebuddy2>
```

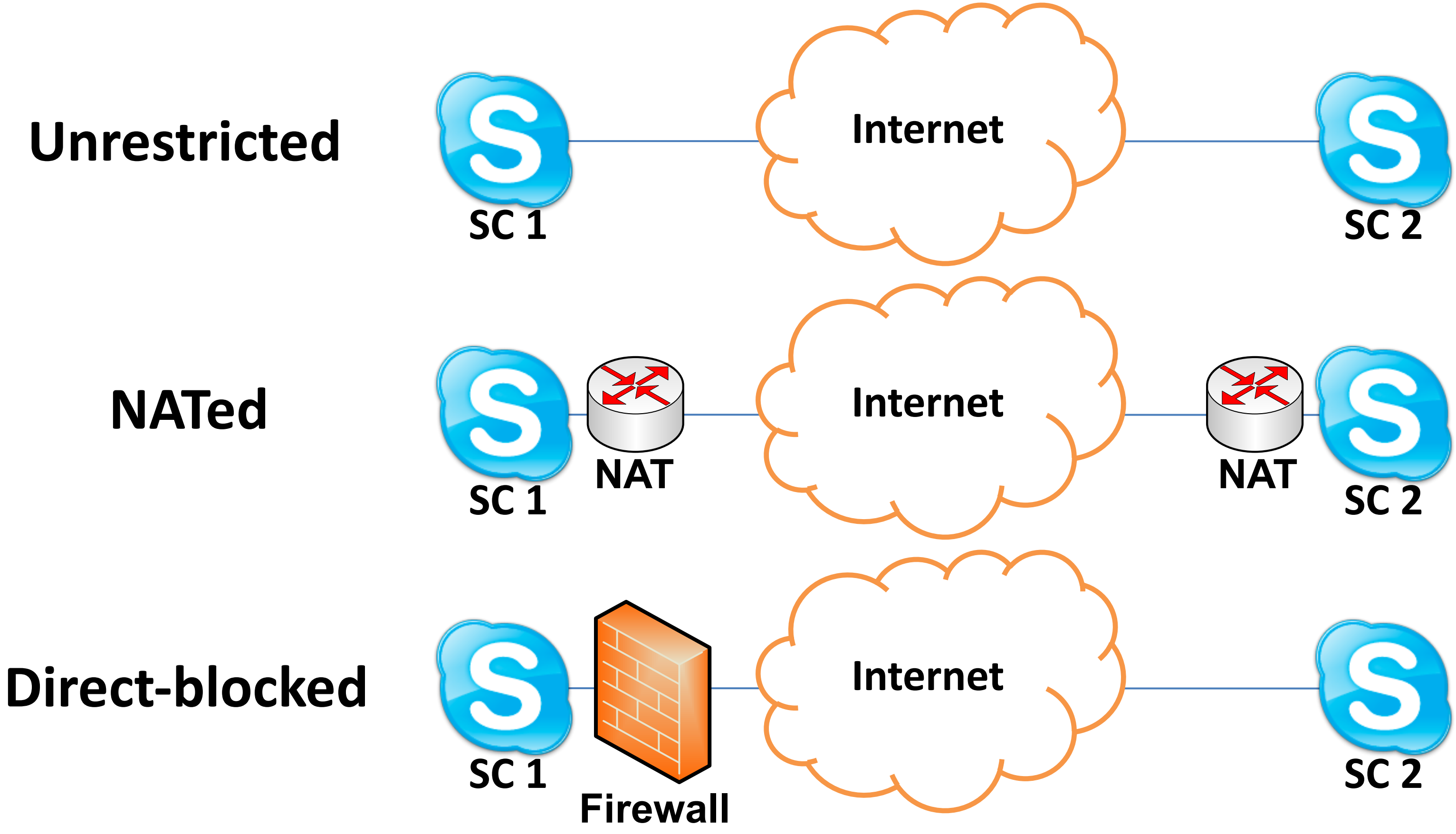
Skype Super Nodes

- **Skype Super Node.** A node with which a Skype client establishes a TCP connection at login
- More than 10K successful login attempts over four days (i.e. nodes are running)
- 35% hostnames had a .edu suffix
  - 102 universities
- Super Nodes IP distribution:
  - US 83.7%, Asia 8.9%, Europe 7.1%
- Top 20 nodes received 43.8% of the total connections
- Top 100 nodes: 70.5%

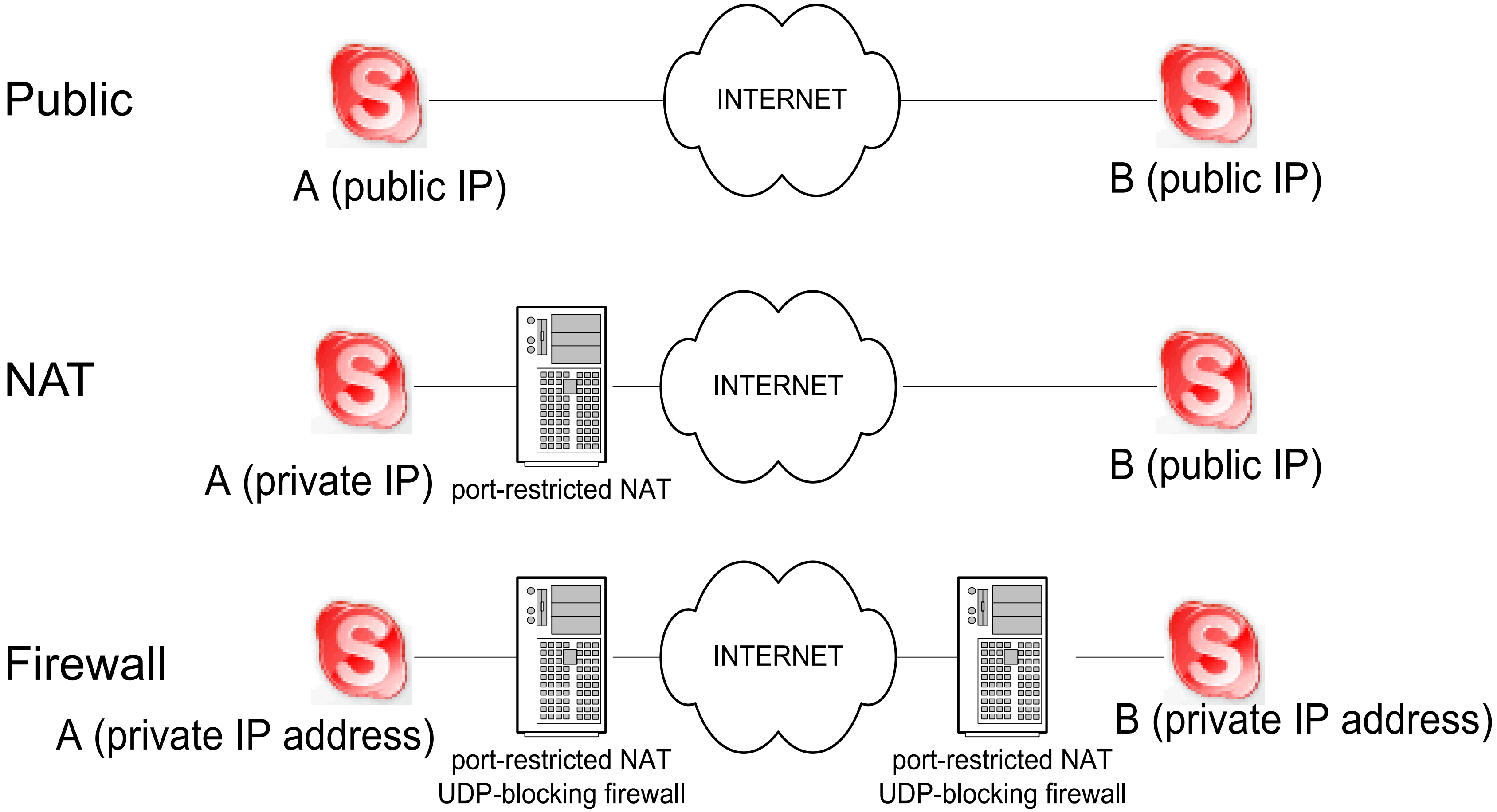
Skype Super Node Map (V1.4)



Experimental Setup: 3 cases to study



Experimental Setup



Public

A (public IP)

INTERNET

B (public IP)

NAT

A (private IP)

port-restricted NAT

INTERNET

B (public IP)

Firewall

A (private IP address)

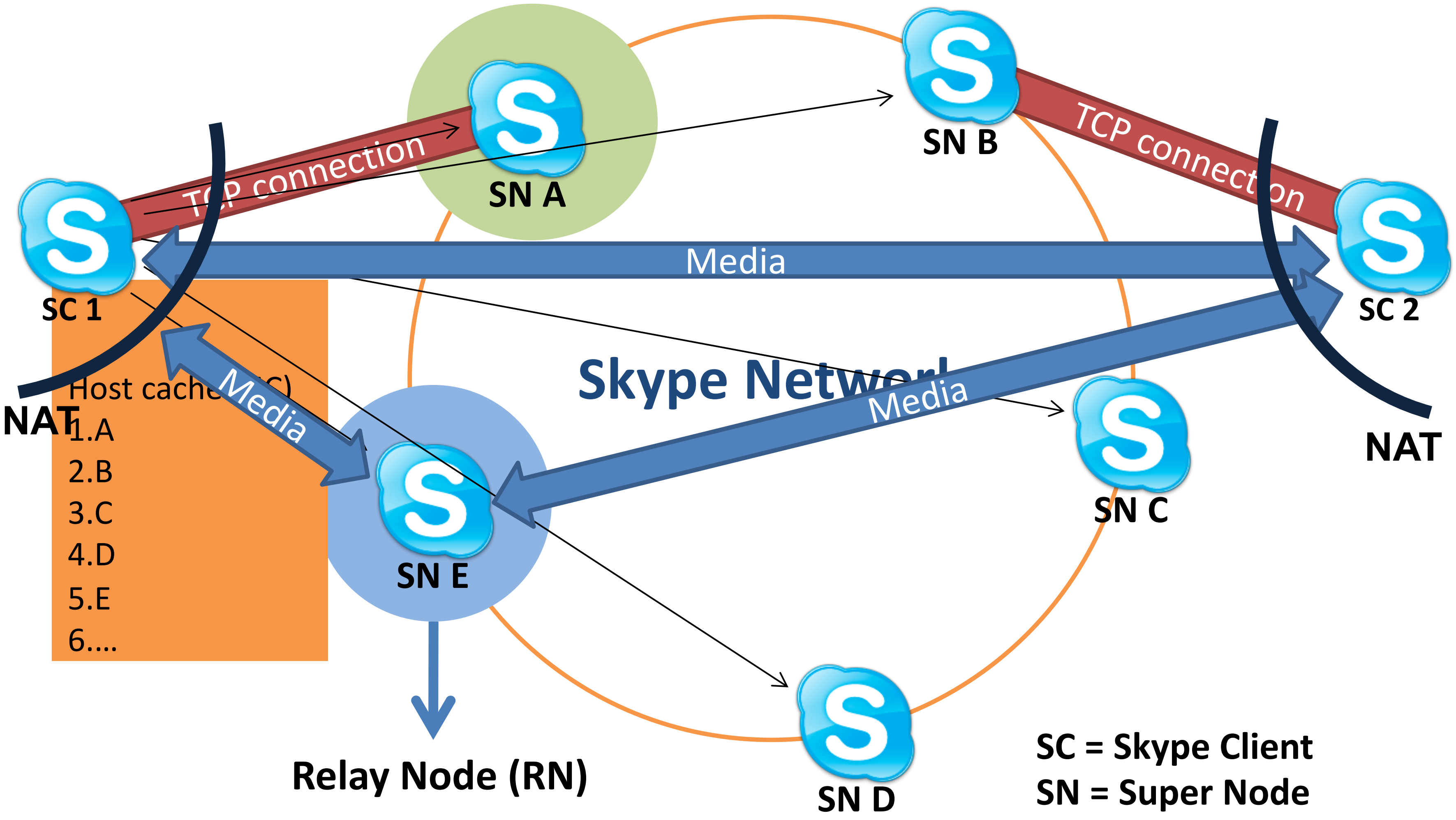
port-restricted NAT
UDP-blocking firewall

INTERNET

port-restricted NAT
UDP-blocking firewall

B (private IP address)

In one slide….



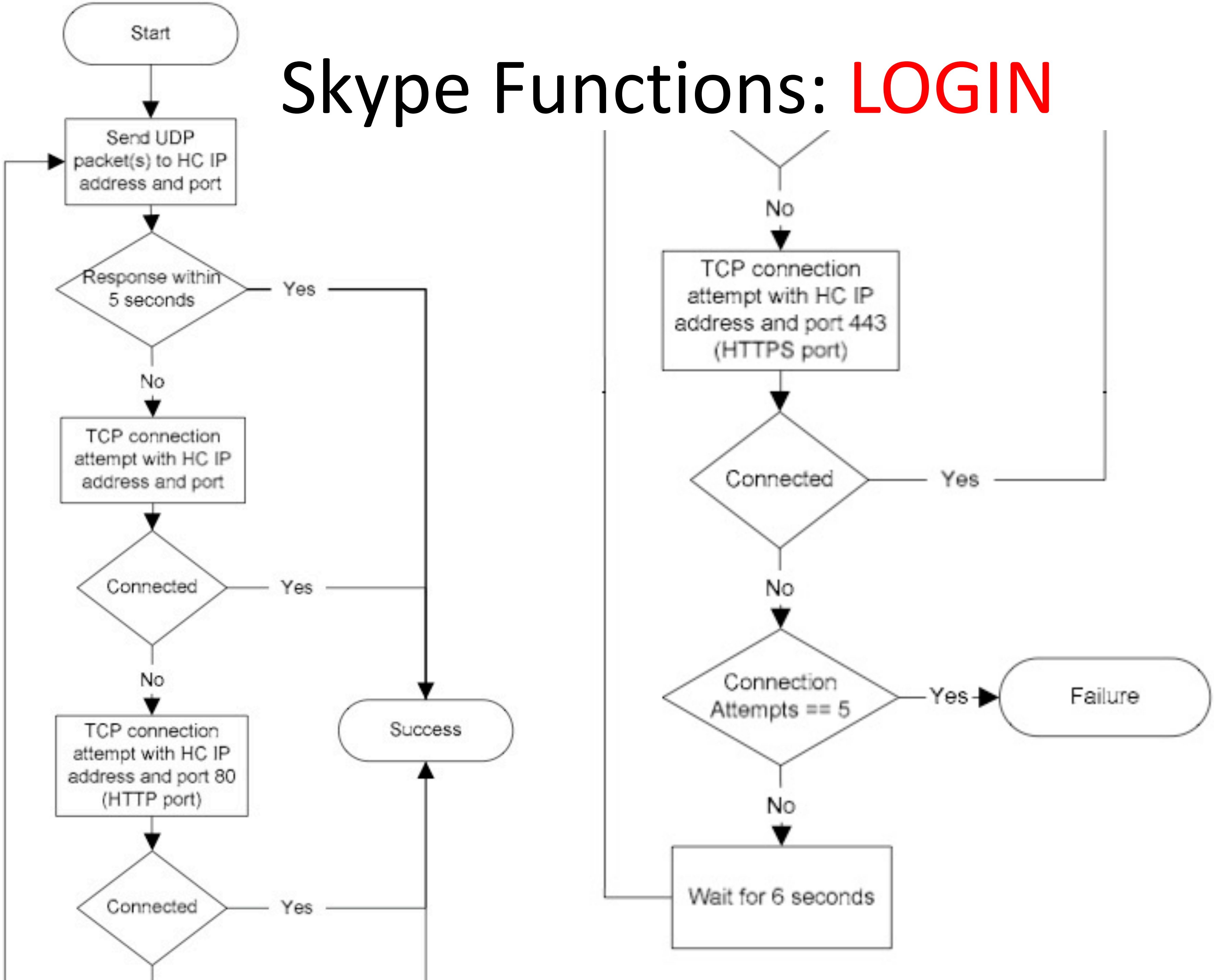
SC = Skype Client
SN = Super Node

Skype Functions: LOGIN

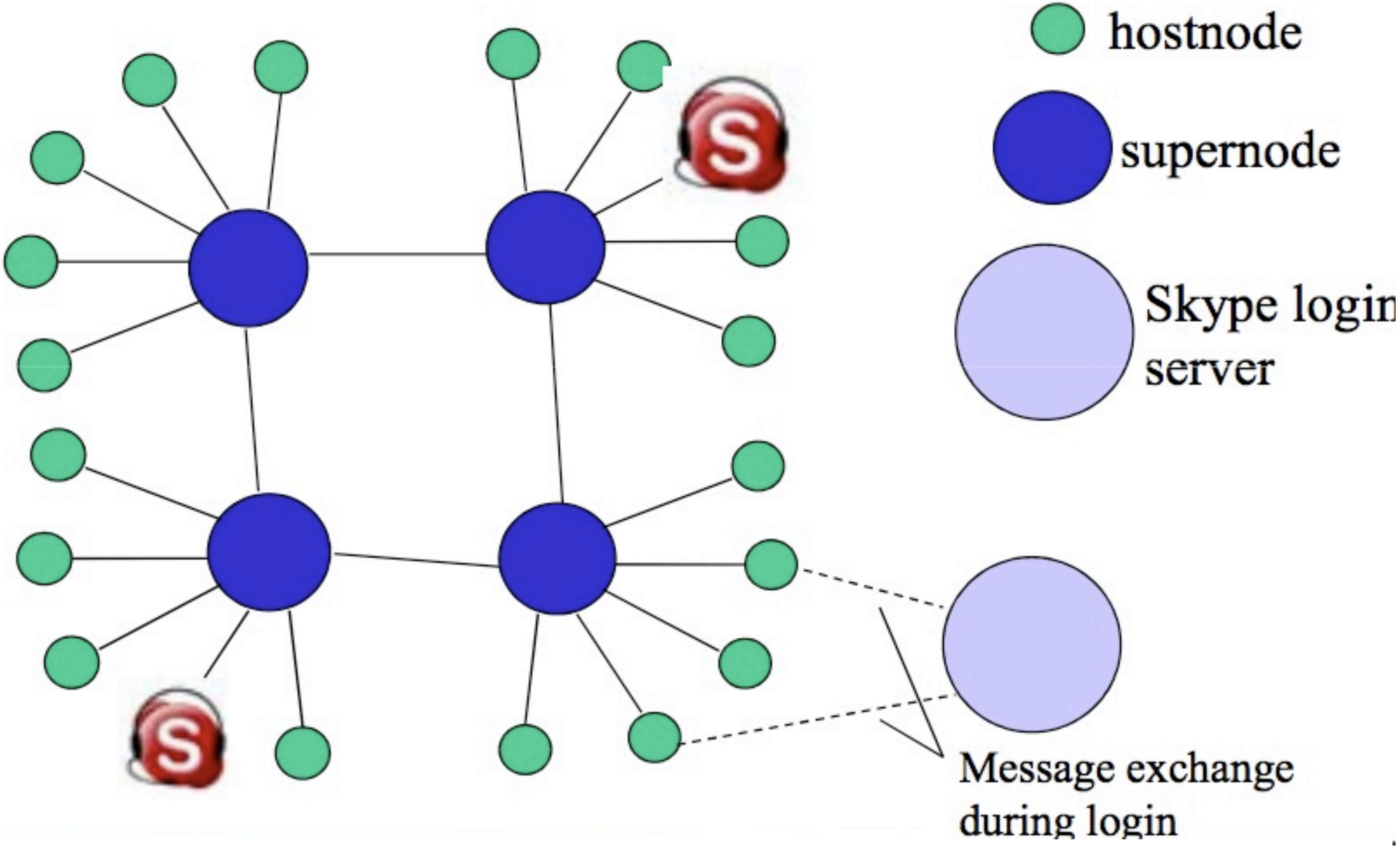
- Establishes a UDP/TCP connection with one or more SN
- Authenticates with the login server and gets a certified public key
- Login is “hard-coded” in the Skype Client

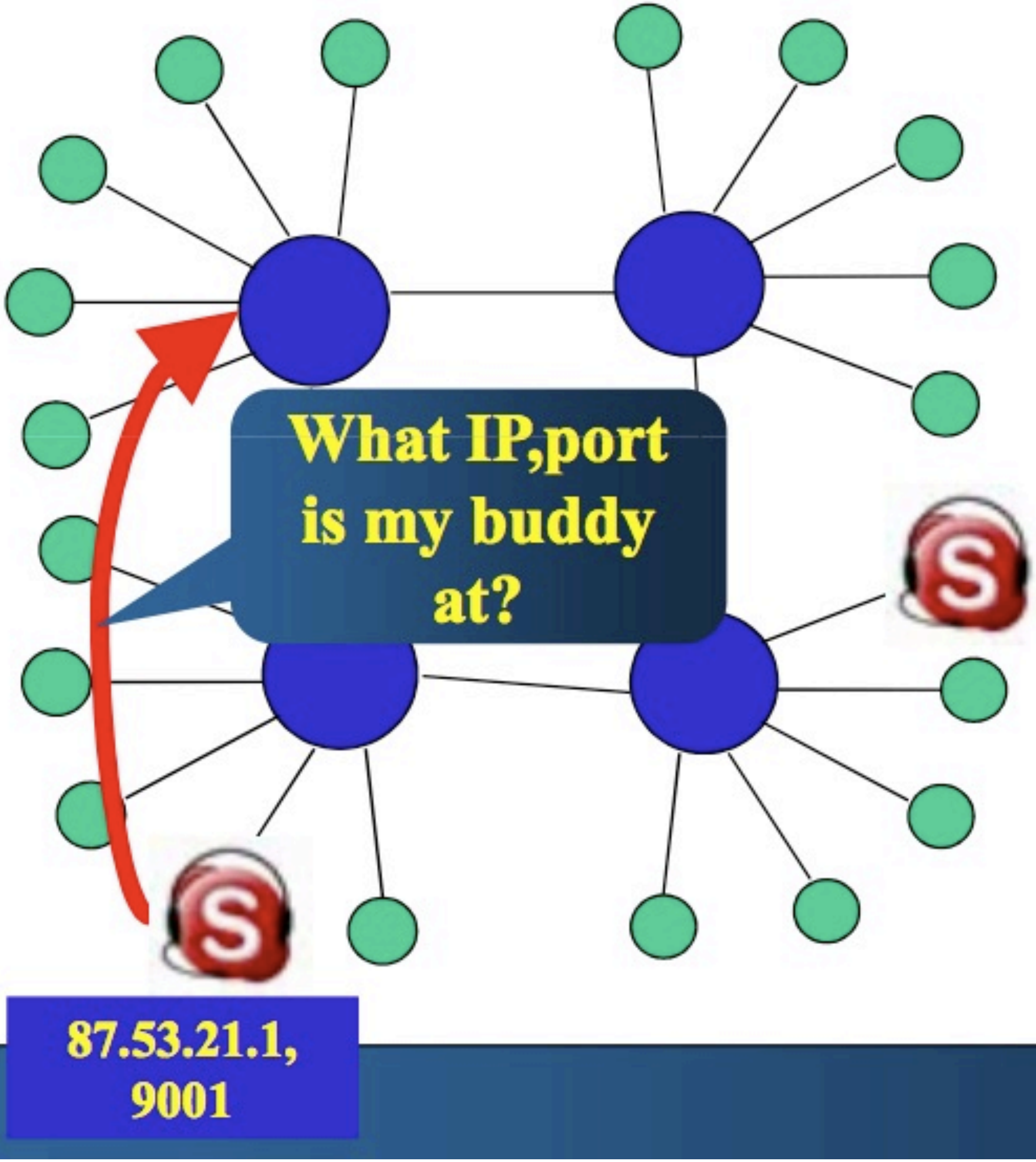
IP address:port	Reverse lookup result	Authority section
66.235.180.9:33033	sss1.skype.net	ns1.hopone.net
66.235.181.9:33033	No PTR result	ns1.hopone.net
212.72.49.143:33033	No PTR result	ns07.customer.eu.level3.net
195.215.8.145:33033	No PTR result	ns3.DK.net
64.246.49.60:33033	rs-64-246-49-60.ev1.net	ns2.ev1.net
64.246.49.61:33033	rs-64-246-49-61.ev1.net	ns2.ev1.net
64.246.48.23:33033	ev1s-64-246-48-23.ev1servers.net	ns1.ev1.net

Skype Functions: LOGIN


Start
Send UDP packet(s) to HC IP address and port
Response within 5 seconds
Yes
No
TCP connection attempt with HC IP address and port
Connected
Yes
No
TCP connection attempt with HC IP address and port 80 (HTTP port)
Connected
Yes
Success
No
TCP connection attempt with HC IP address and port 443 (HTTPS port)
Connected
Yes
No
Connection Attempts == 5
Yes
Failure
No
Wait for 6 seconds

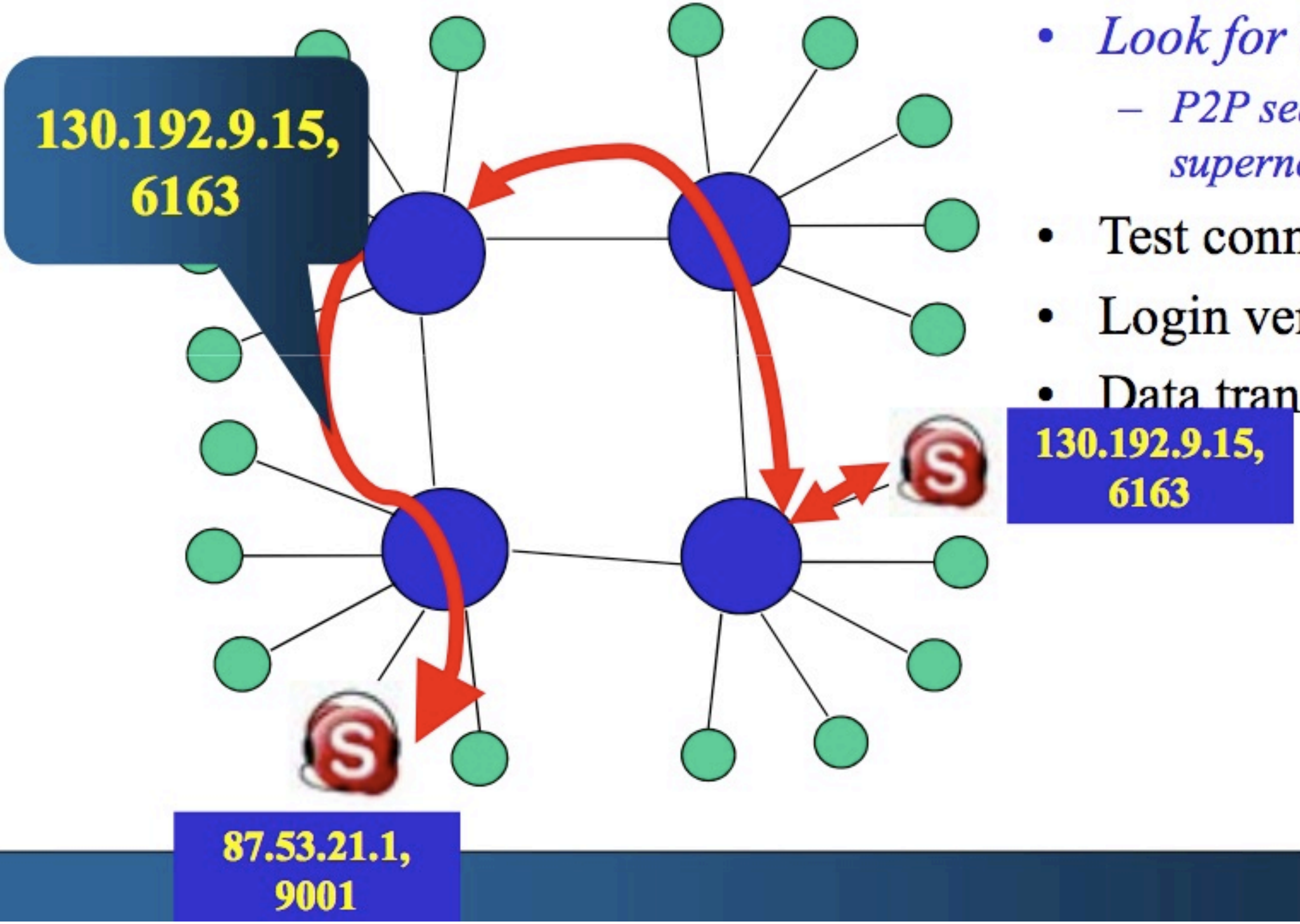
Case 1: the two SC have a visible IP



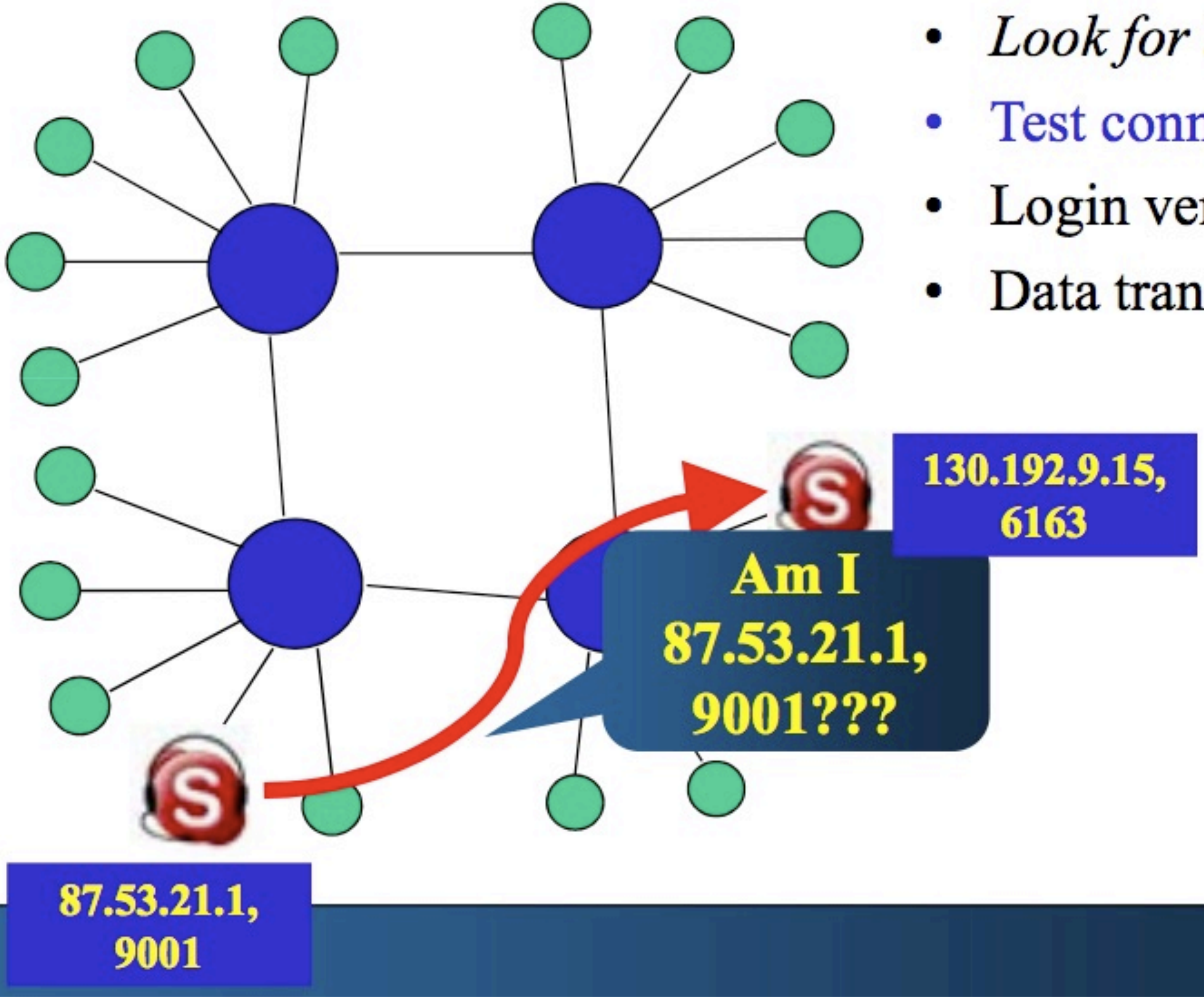


- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission

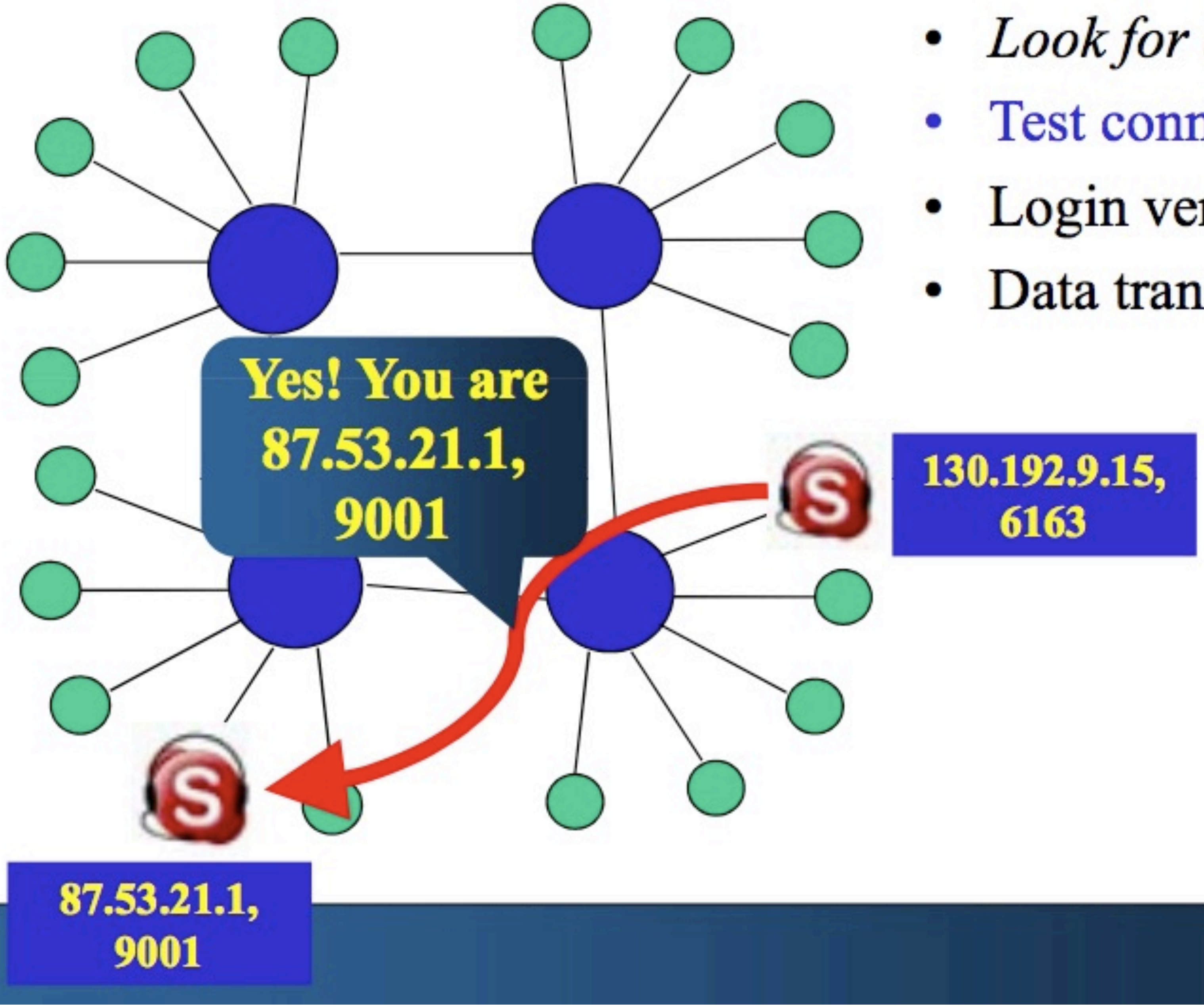
- *Look for buddy info*
  - *P2P search through supernodes*
- Test connectivity
- Login versus party
- Data transmission



- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission



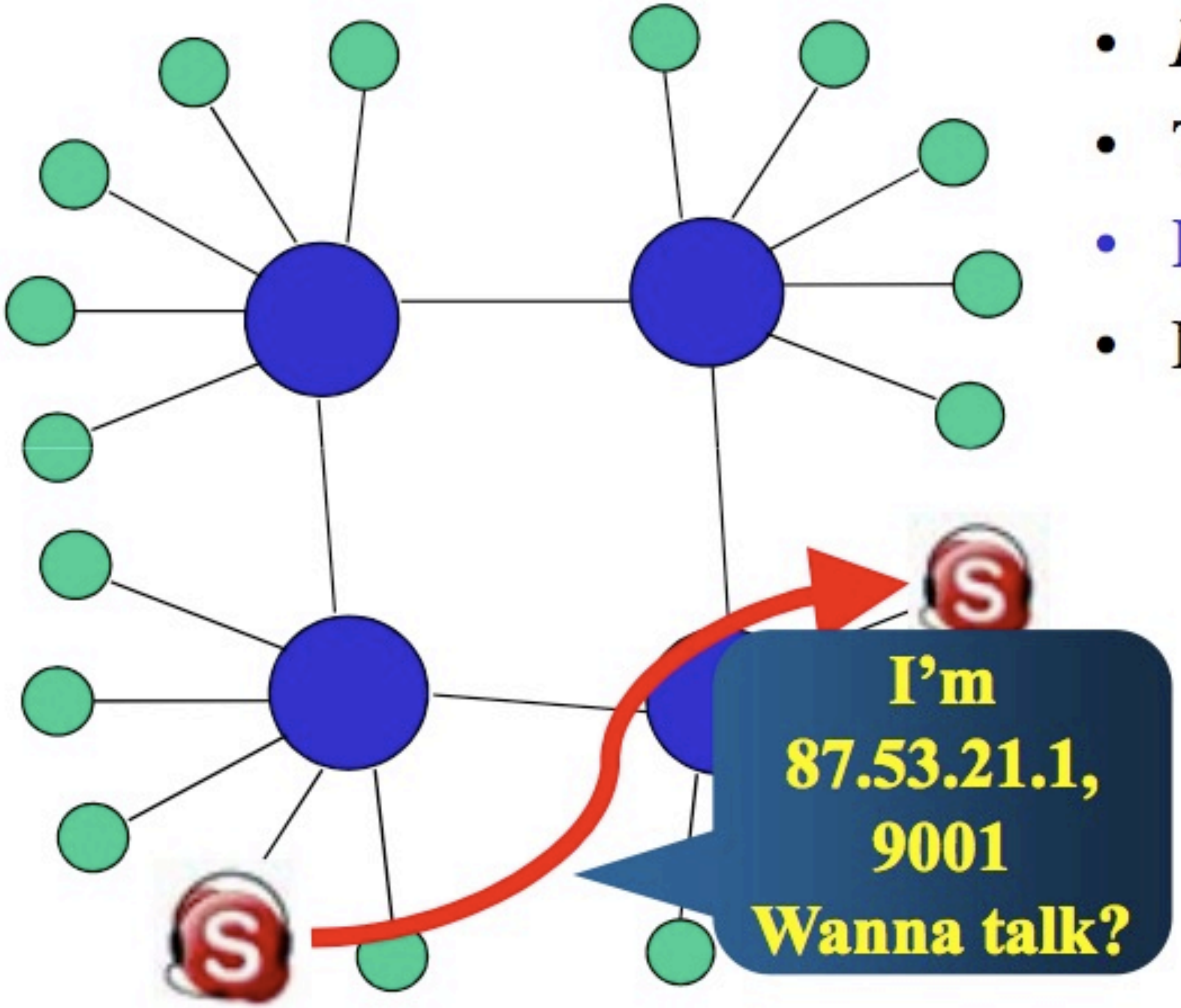
- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission



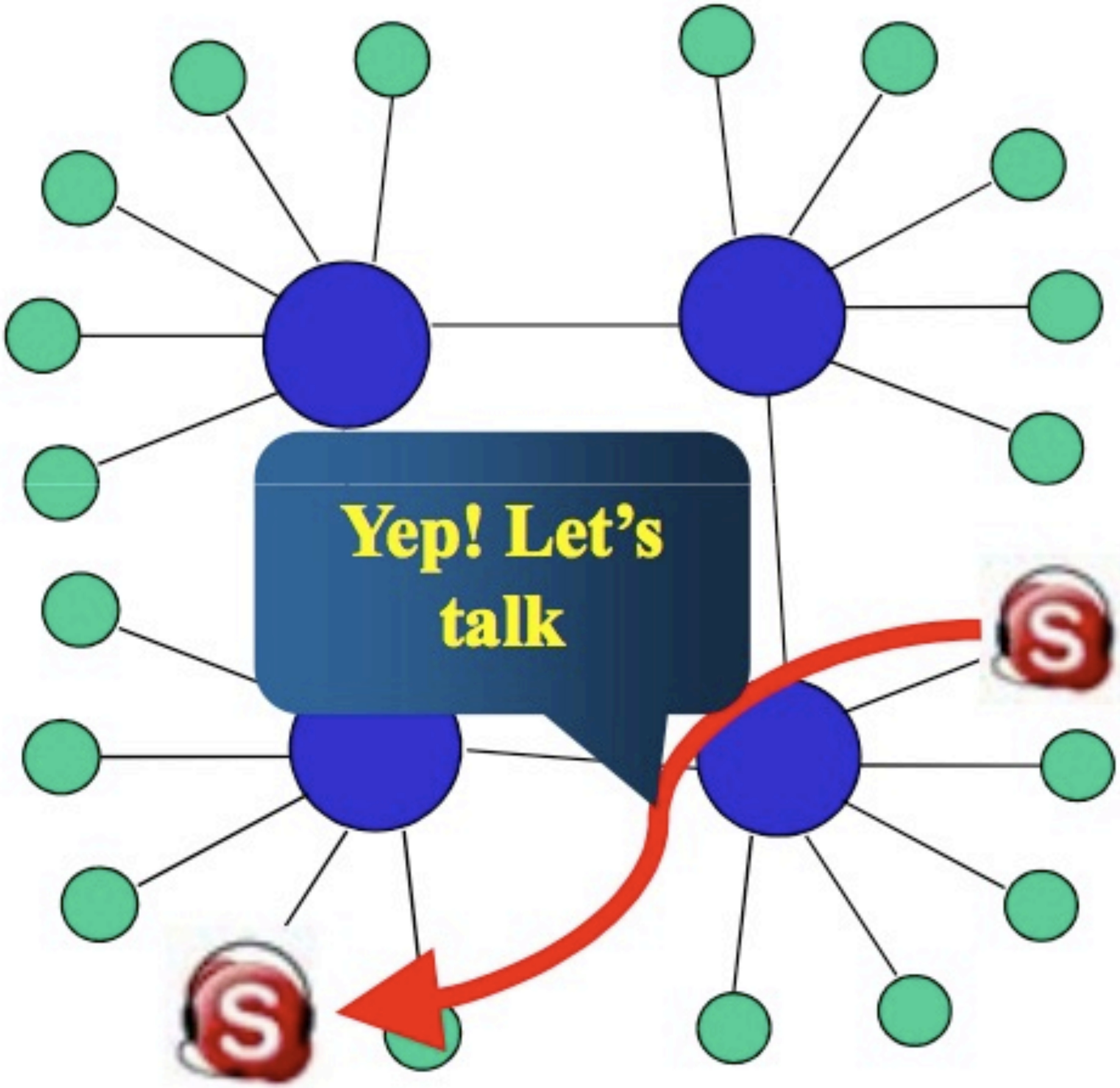
Yes! You are 87.53.21.1, 9001

130.192.9.15, 6163

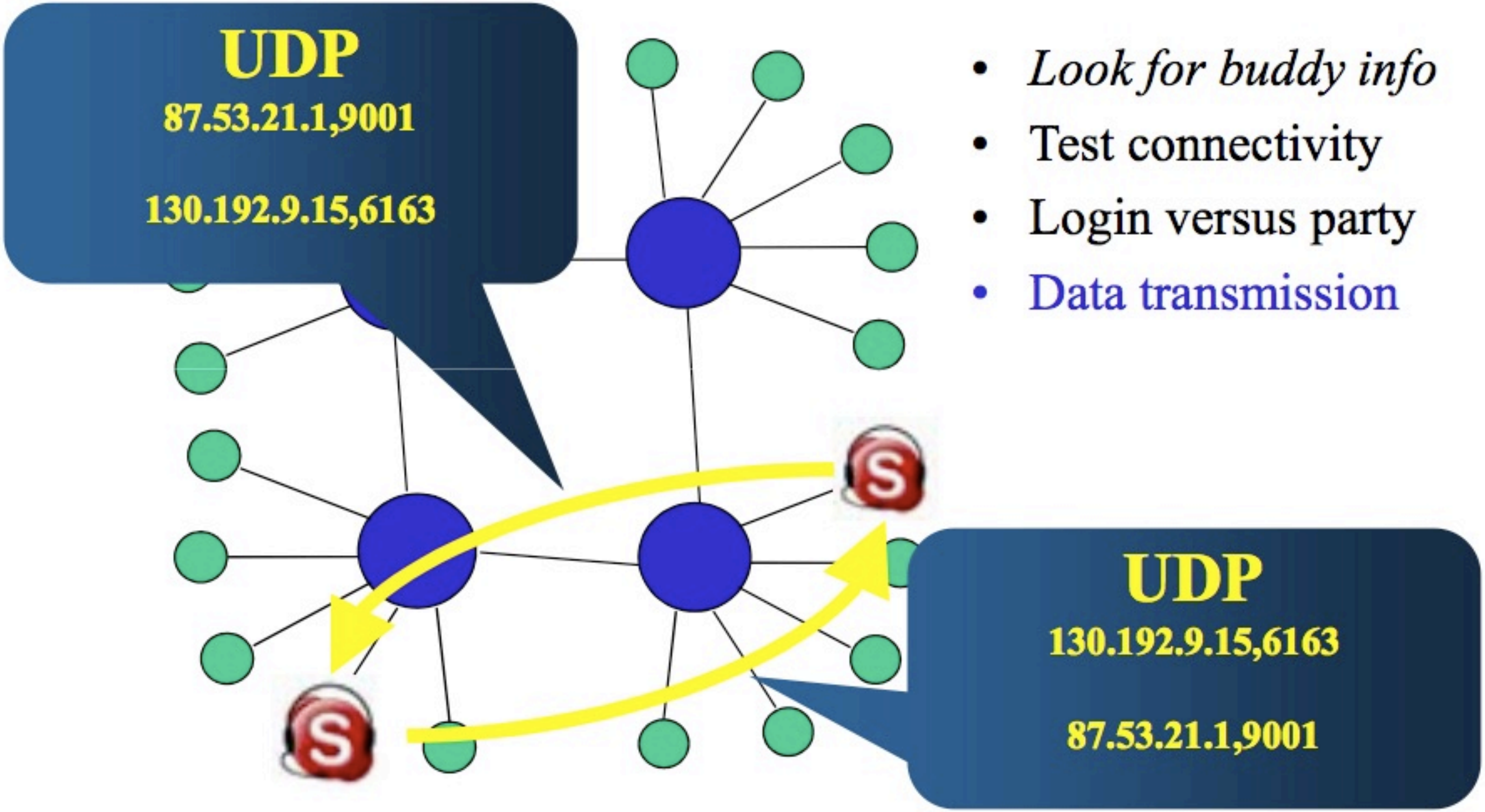
87.53.21.1, 9001



- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission

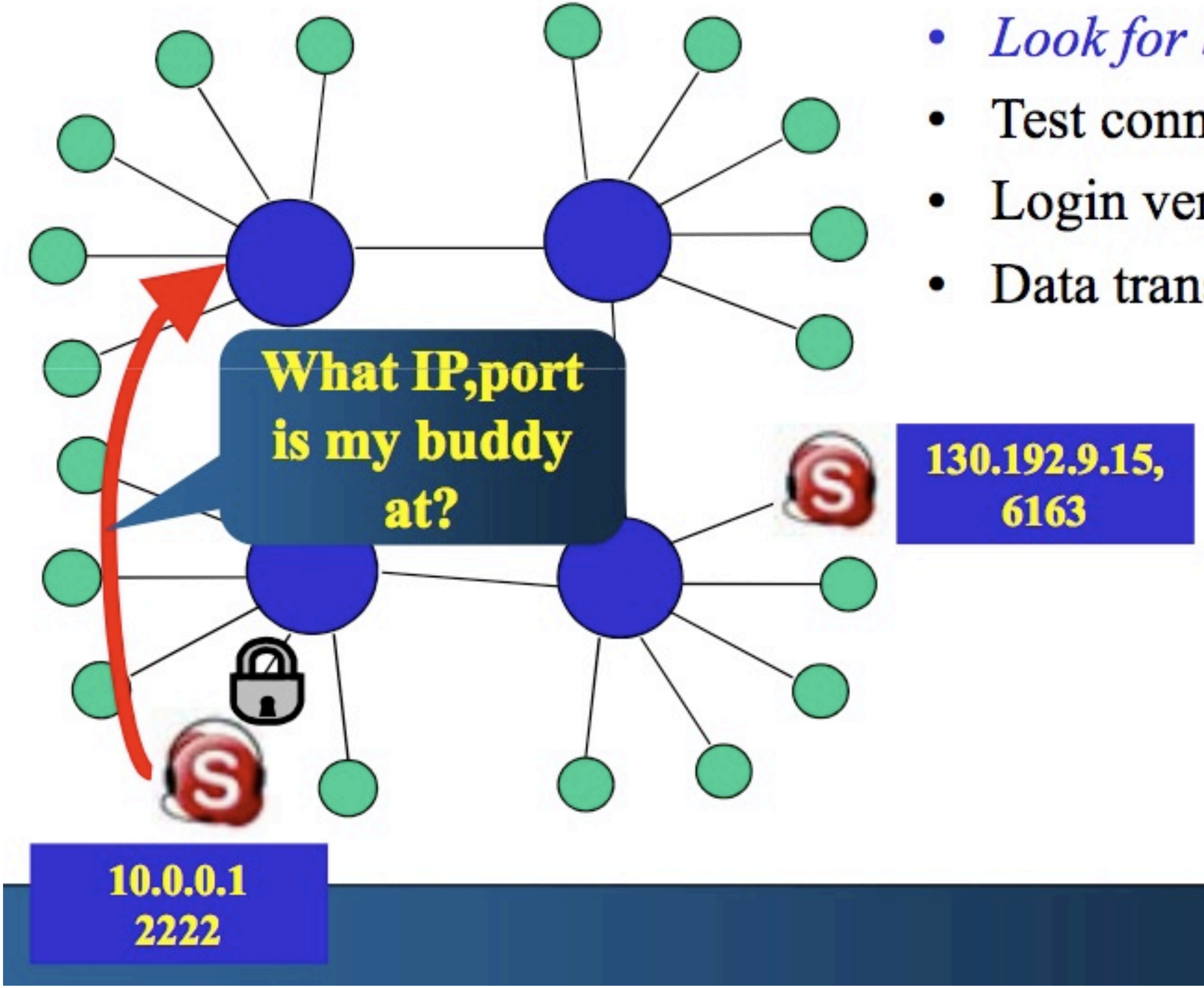


- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission

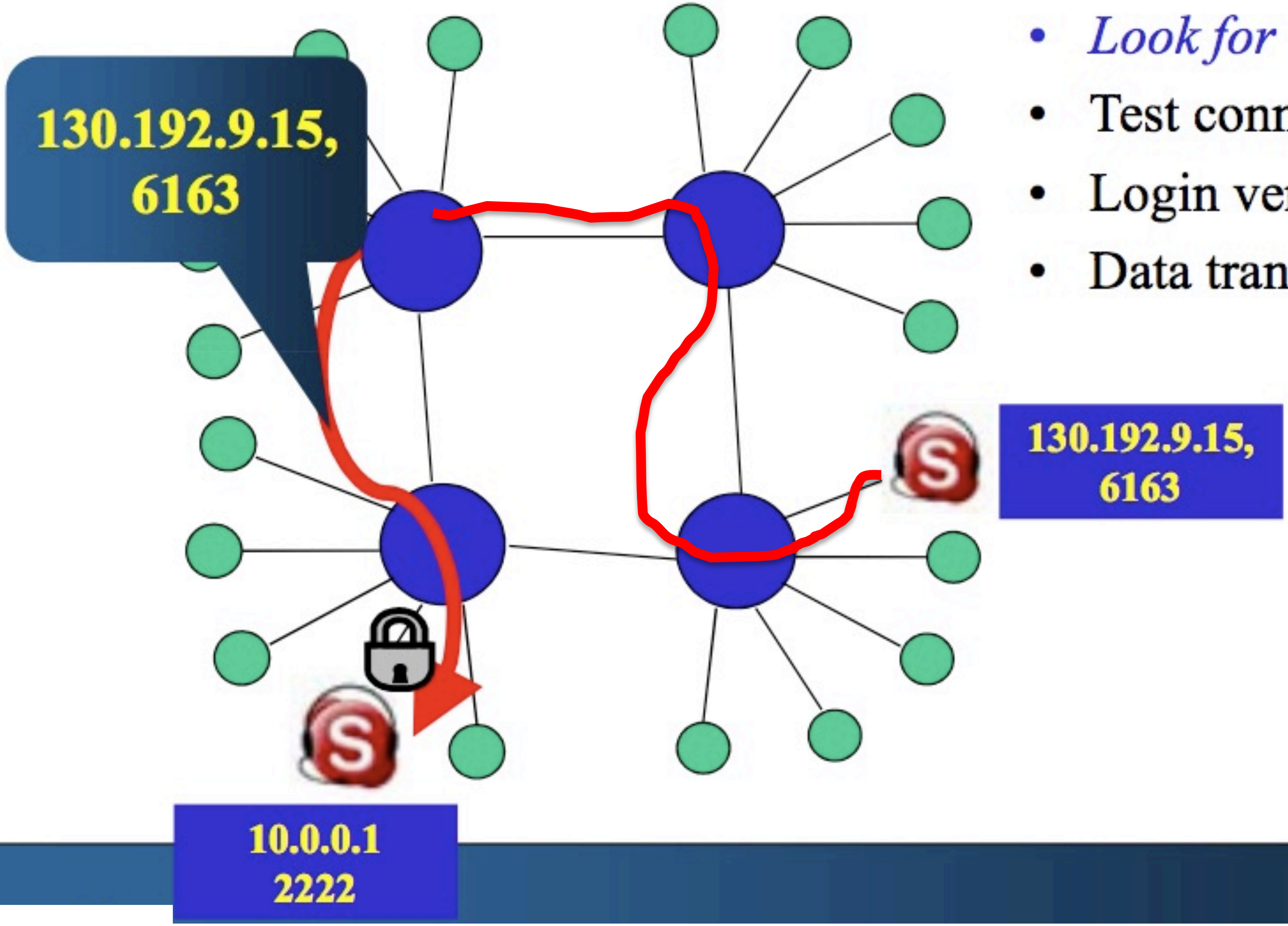


- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission

Case 2: one SC does not have a visible IP

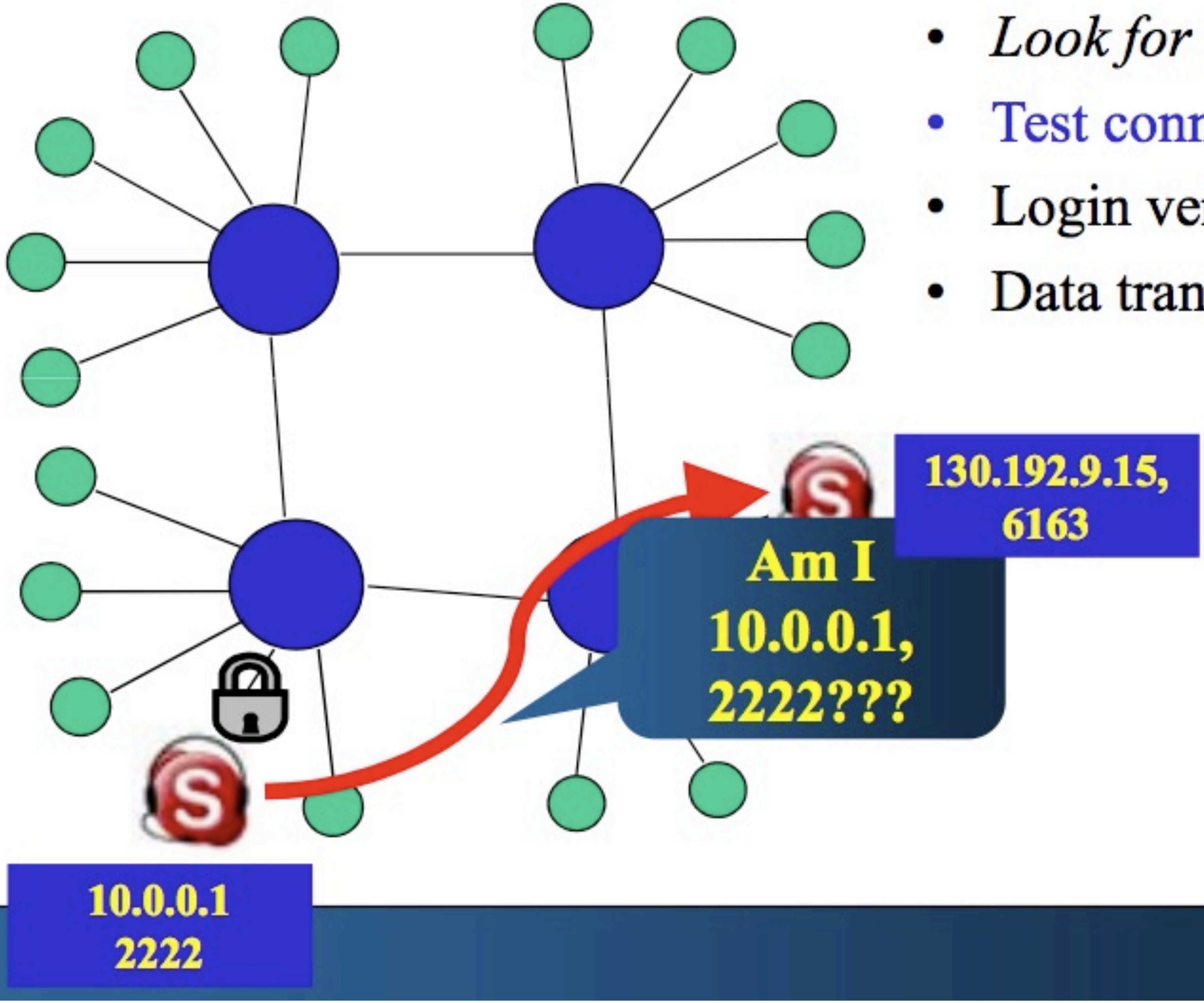


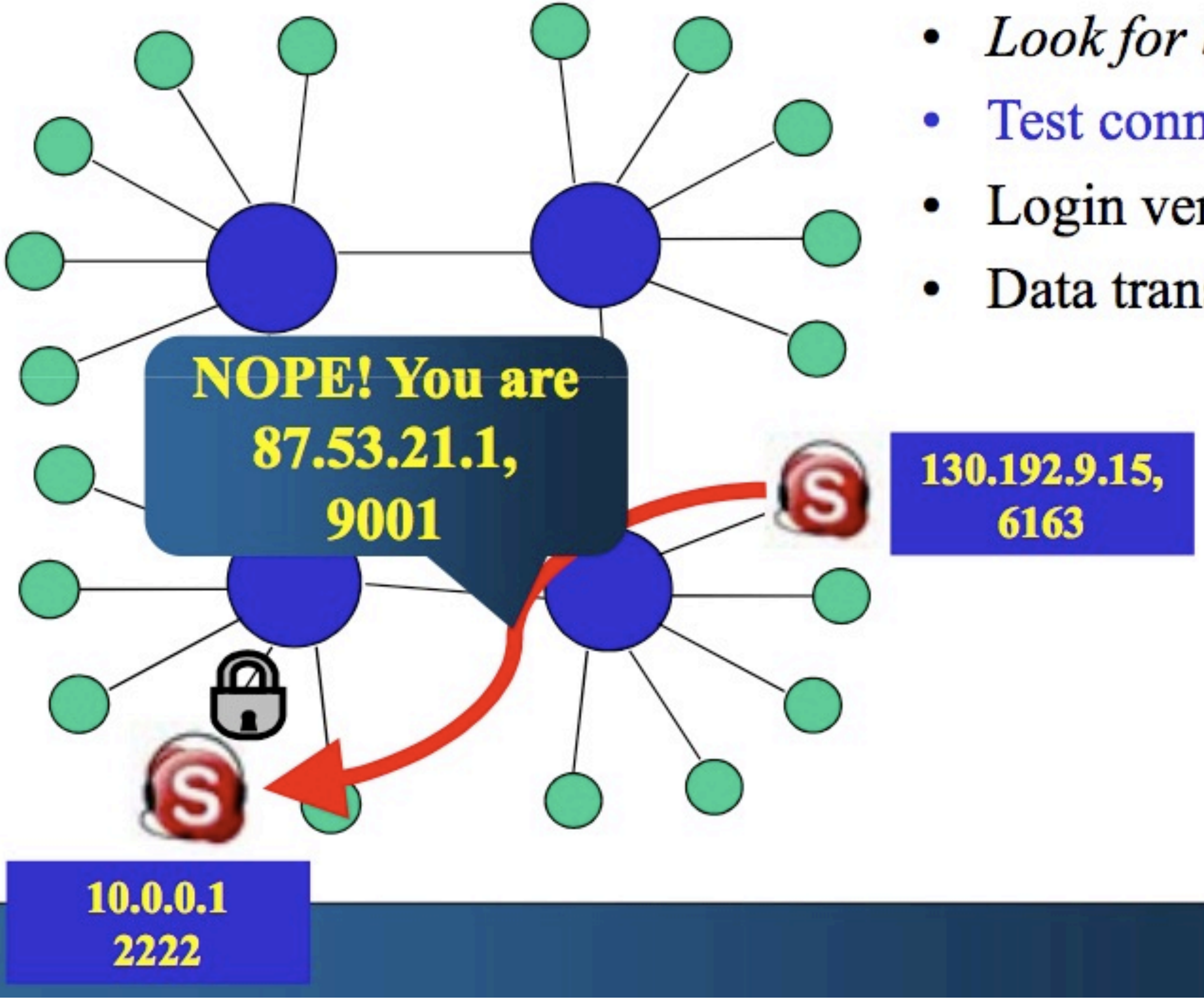
- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission



- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission

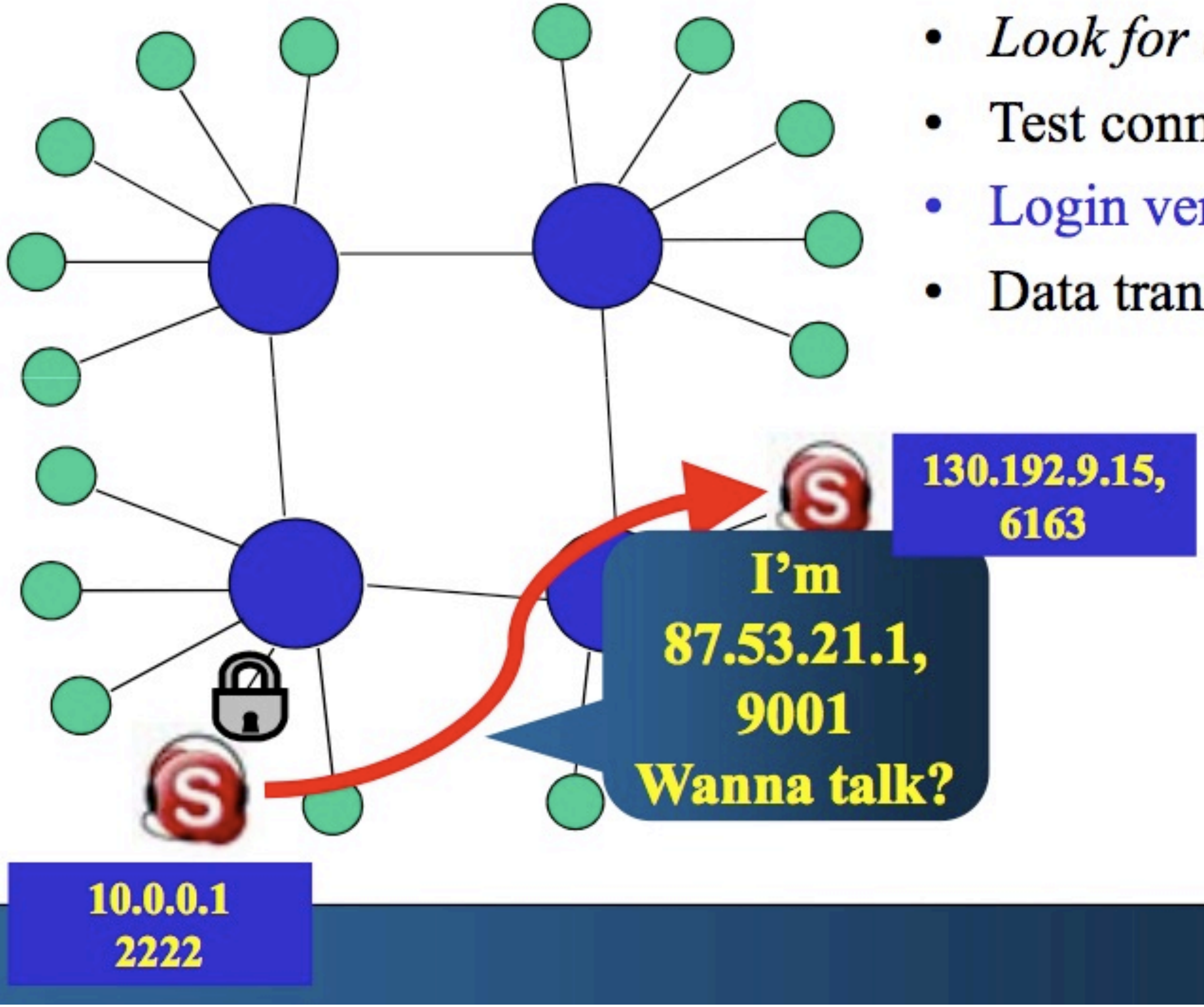
- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission



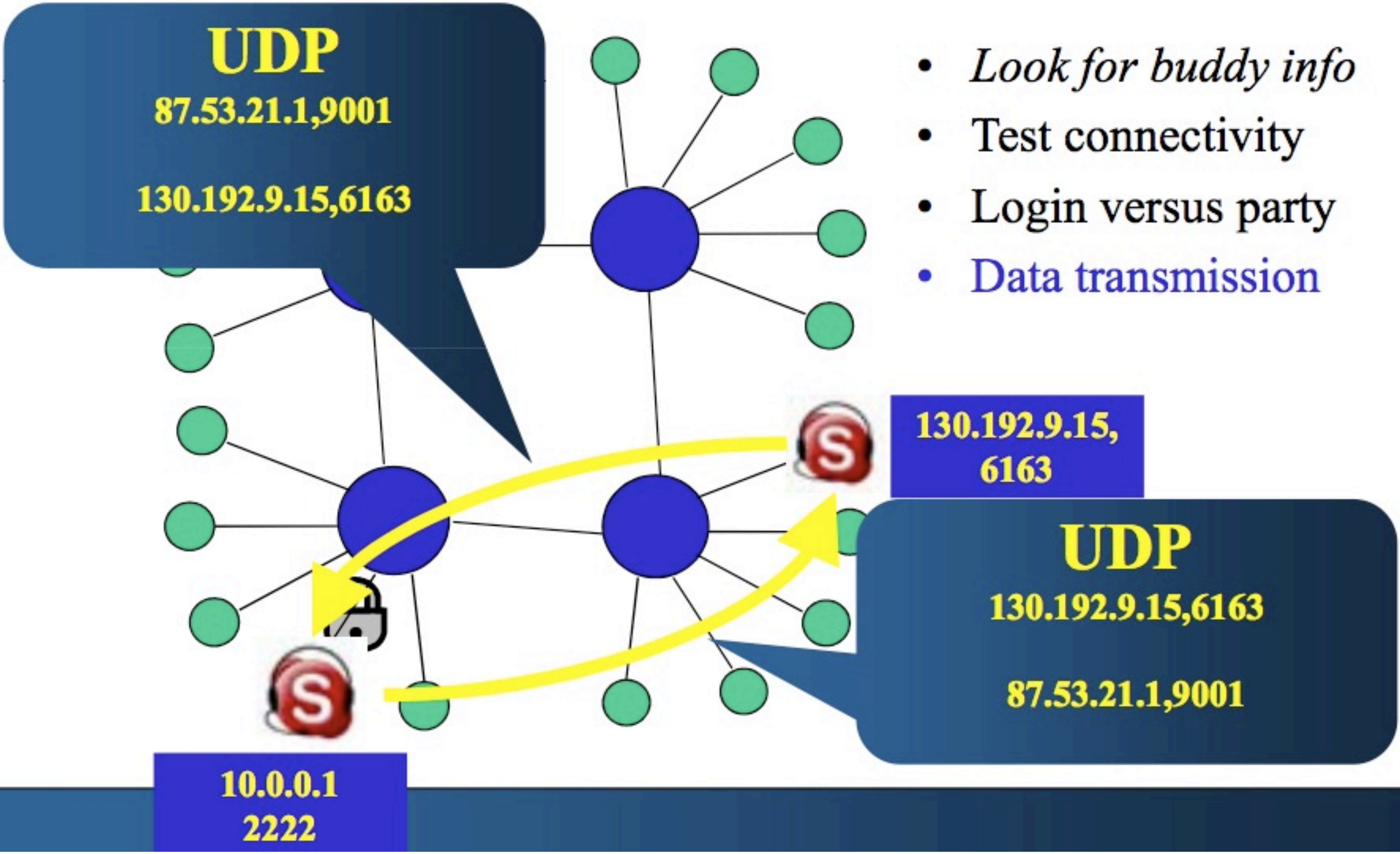


- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission

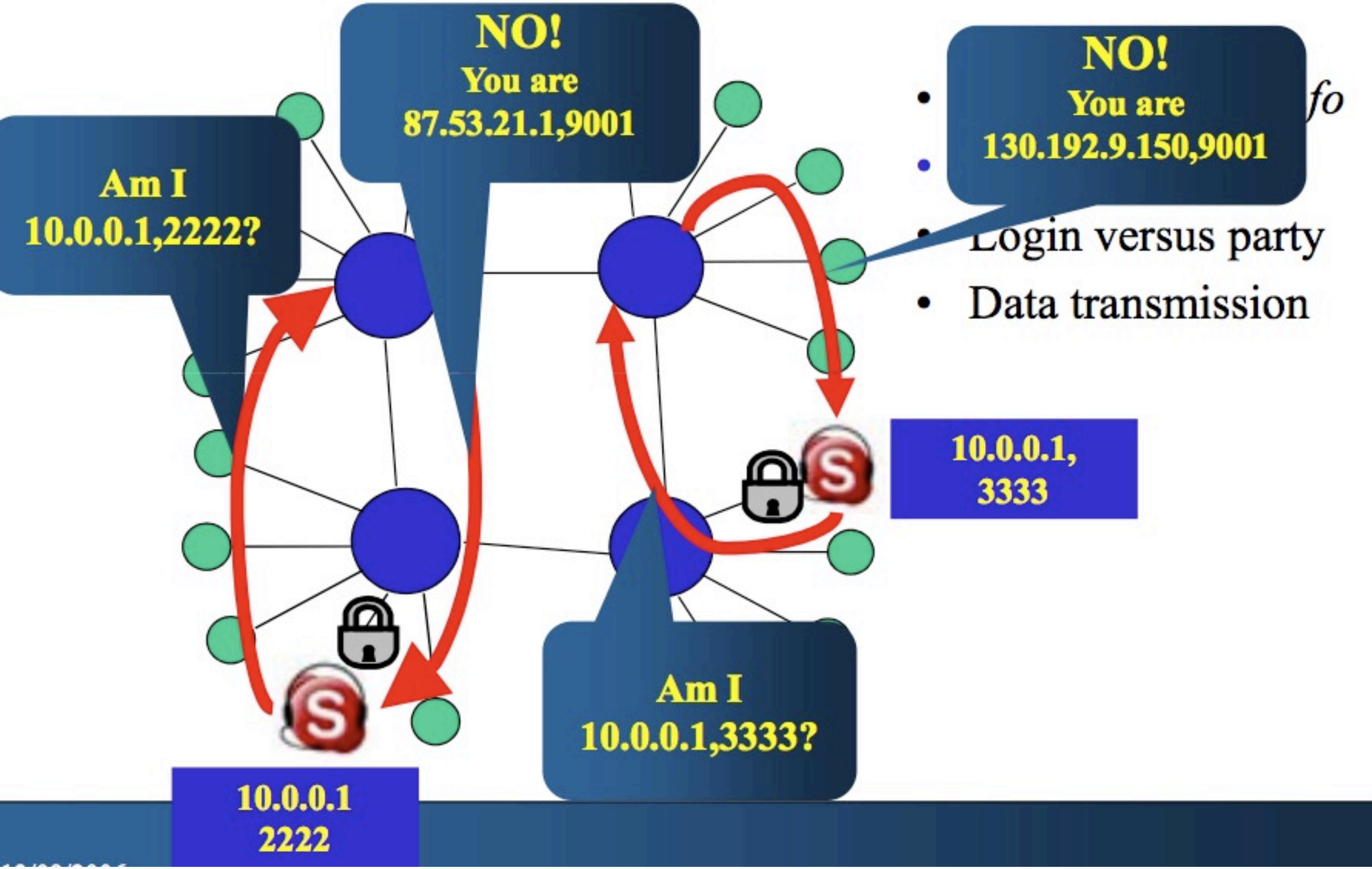
- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission



- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission

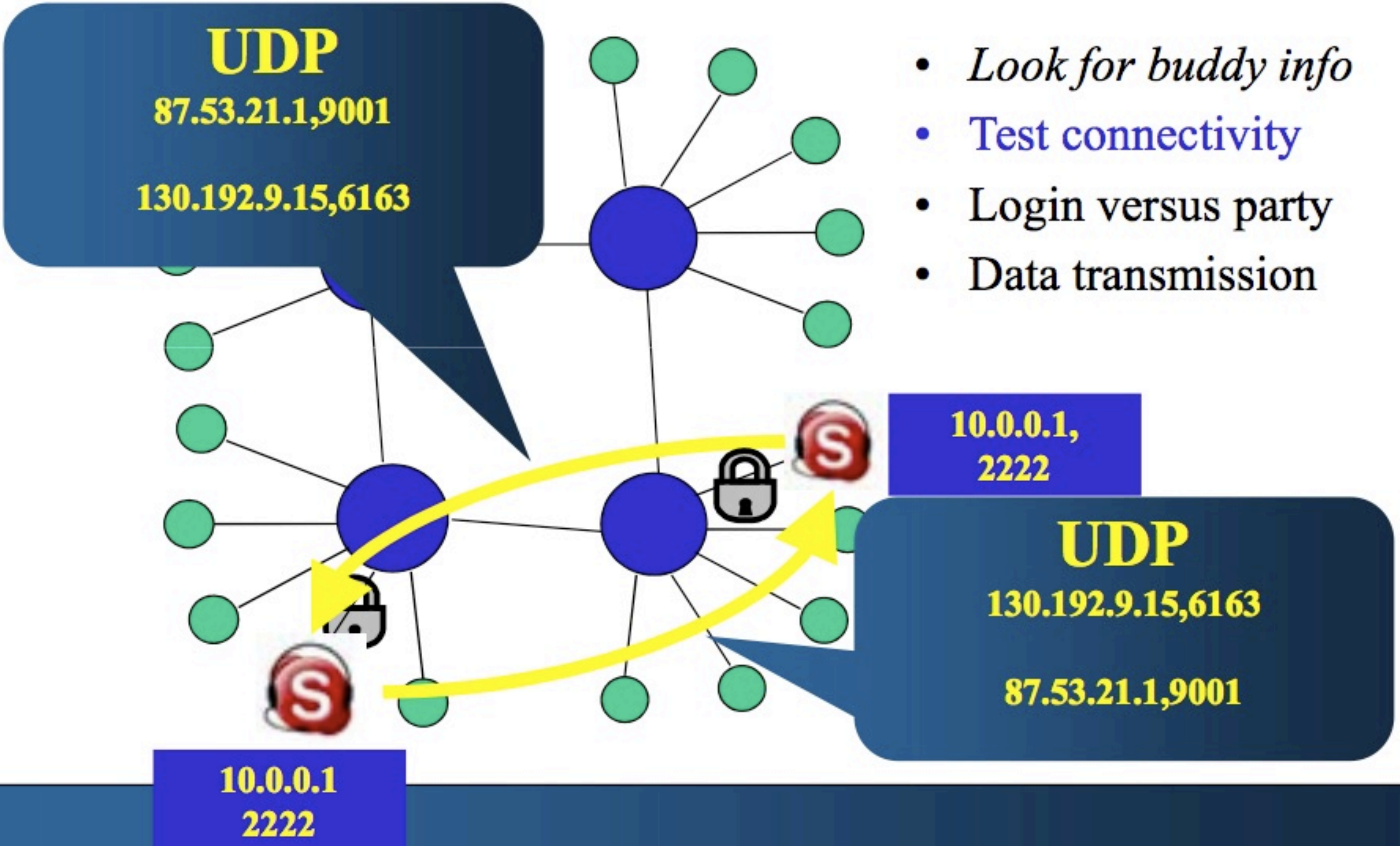


Case 3: the two SCs does not have a visible IP



- Login versus party
- Data transmission

- *Look for buddy info*
- Test connectivity
- Login versus party
- Data transmission



UDP

87.53.21.1,9001

130.192.9.15,6163

10.0.0.1,
2222

UDP

130.192.9.15,6163

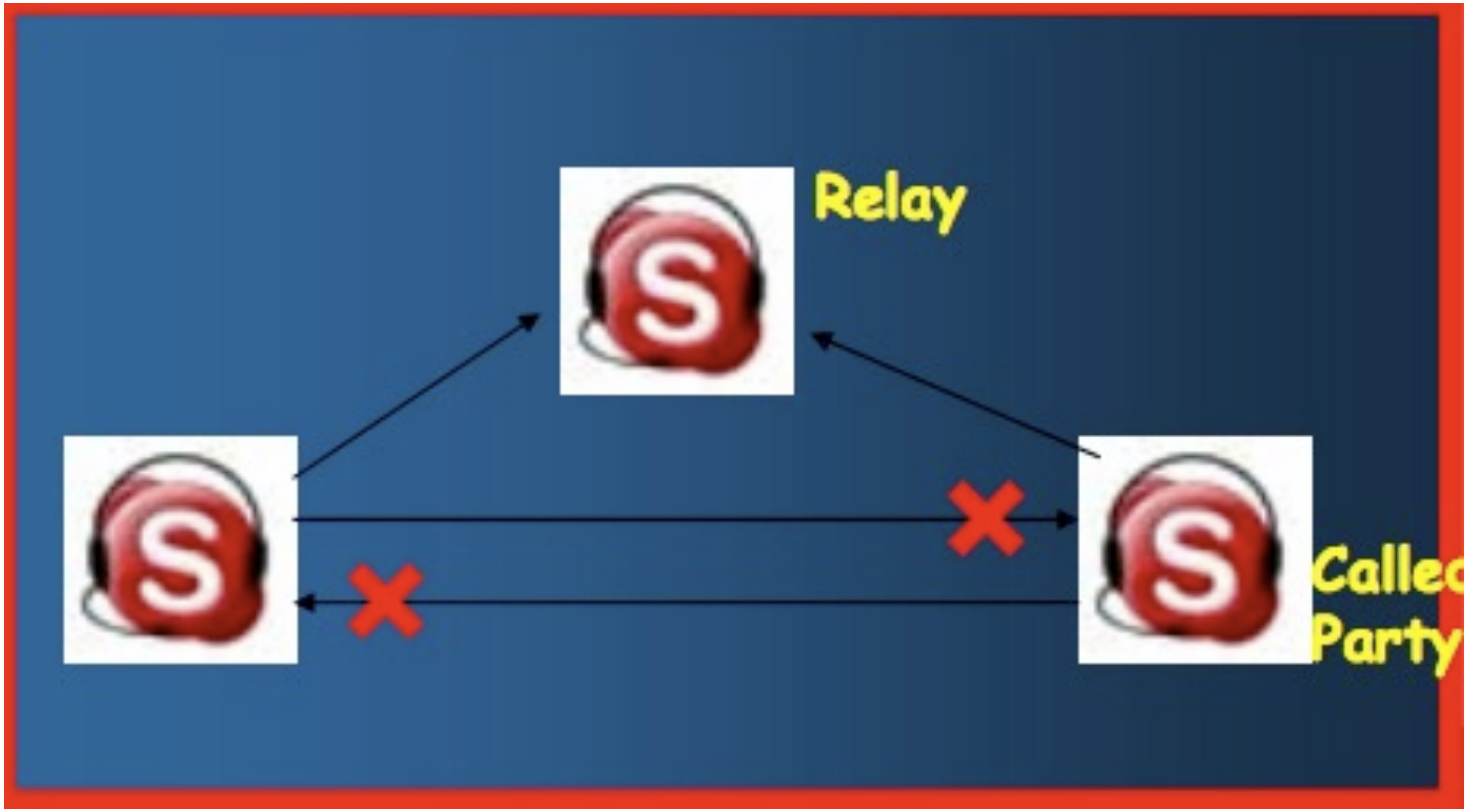
87.53.21.1,9001

10.0.0.1
2222

In case of caller/callee firewalled

- If a *firewall* block UDP traffic
  - Clients cannot use UDP to talk each other
- If the callee has a public IP address, then
  - TCP is used by the caller
  - If TCP is firewalled, try port 80 and 443 (HTTPS)
- If the callee is firewalled/NATed
  - NAT traversal cannot work with TCP
  - Perform a **TCP Relay** (TURN protocol)

TCP Relay (Turn protocol)


Relay
Called
Party