Interactive Connectivity Establishment: ICE

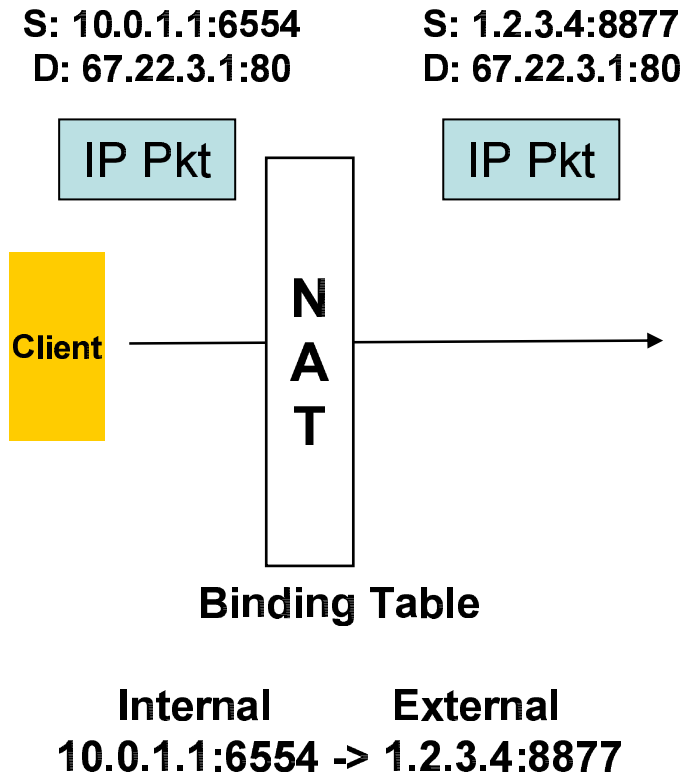
Jonathan Rosenberg
Cisco Systems

Talk Outline

- NAT Traversal Problem Statement
- ICE Overview

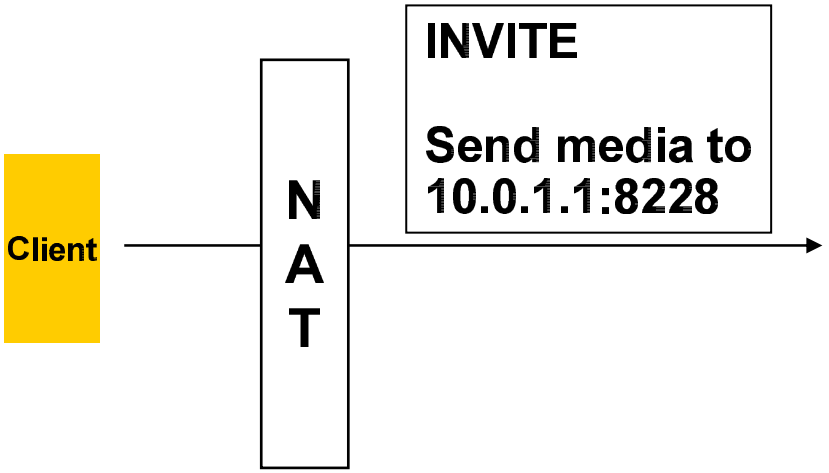
What is NAT?

- Network Address Translation (NAT)
  - Creates address binding between internal private and external public address
  - Modifies IP Addresses/Ports in Packets
  - Benefits
    - Avoids network renumbering on change of provider
    - Allows multiplexing of multiple private addresses into a single public address ($$ savings)
    - Maintains privacy of internal addresses



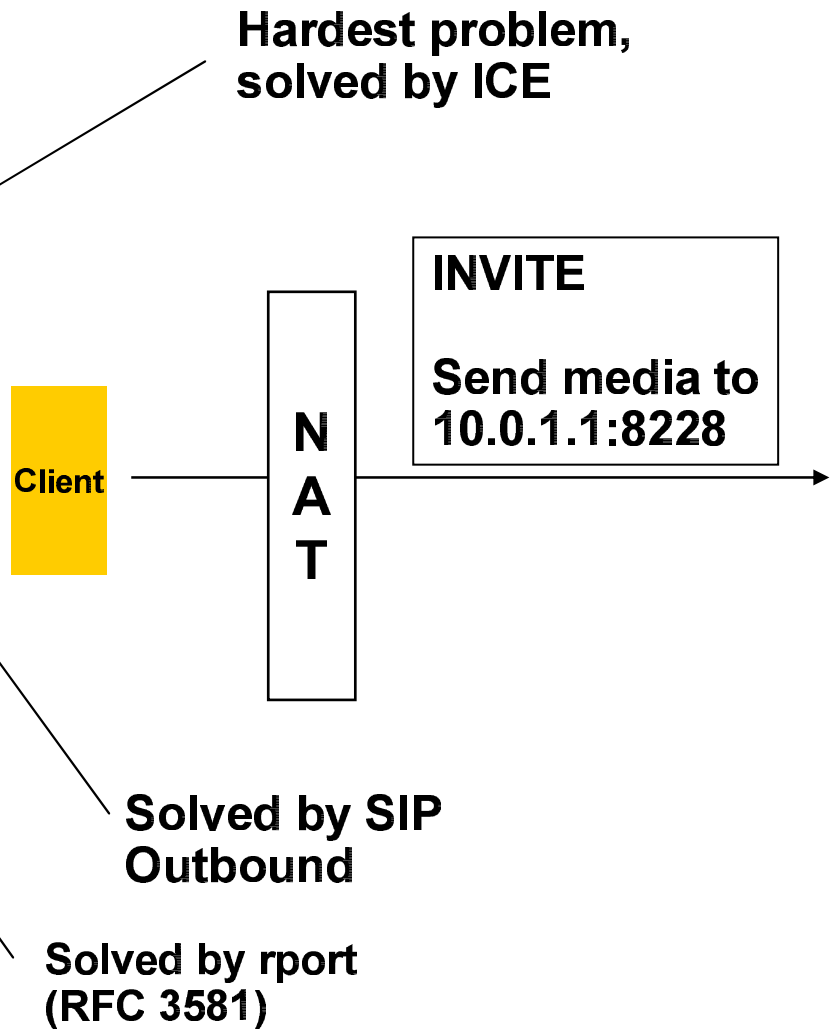
Why is this bad for SIP?

- Client will generate SIP INVITE and 200 OK responses with private addresses
  - In the SDP as the target for receipt of media
  - In the Contact of a REGISTER as the target for incoming INVITE
  - In the Via of a request as the target for the response
- Recipient will not be able to send packets to this private address
  - Media is discarded
  - Incoming calls are not delivered
  - Responses are not received



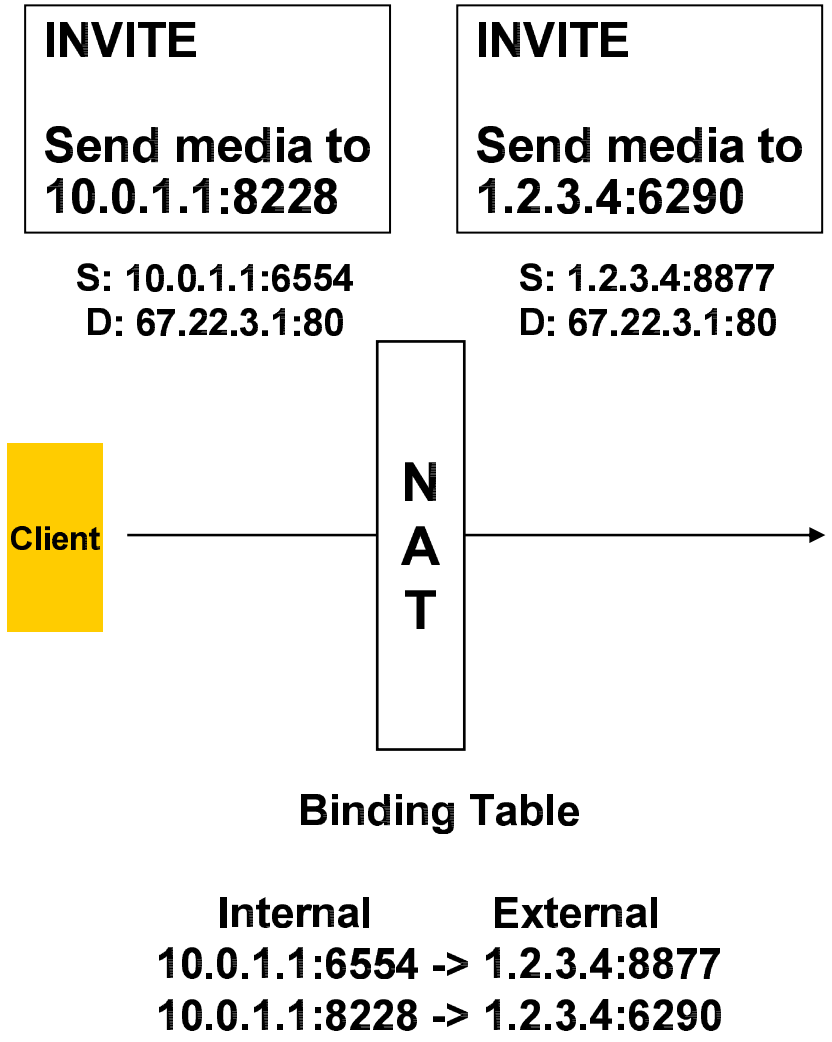
Why is this bad for SIP?

- Client will generate SIP INVITE and 200 OK responses with private addresses
  - In the SDP as the target for receipt of media
  - In the Contact of a REGISTER as the target for incoming INVITE
  - In the Via of a request as the target for the response
- Recipient will not be able to send packets to this private address
  - Media is discarded
  - Incoming calls are not delivered
  - Responses are not received



What about the obvious solution?

- The NAT rewrites source IP of SIP packet, but not contents
- Why not have NAT rewrite the contents of the SIP packet also (Application Layer Gateway (ALG))?
- Numerous big problems
  - Requires SIP security mechanisms to be disabled
  - Hard to diagnose problems
  - Requires network upgrade in all NAT
  - Frequent implementation problems
  - Incentives mismatched
  - Anathema to the concept of the Internet

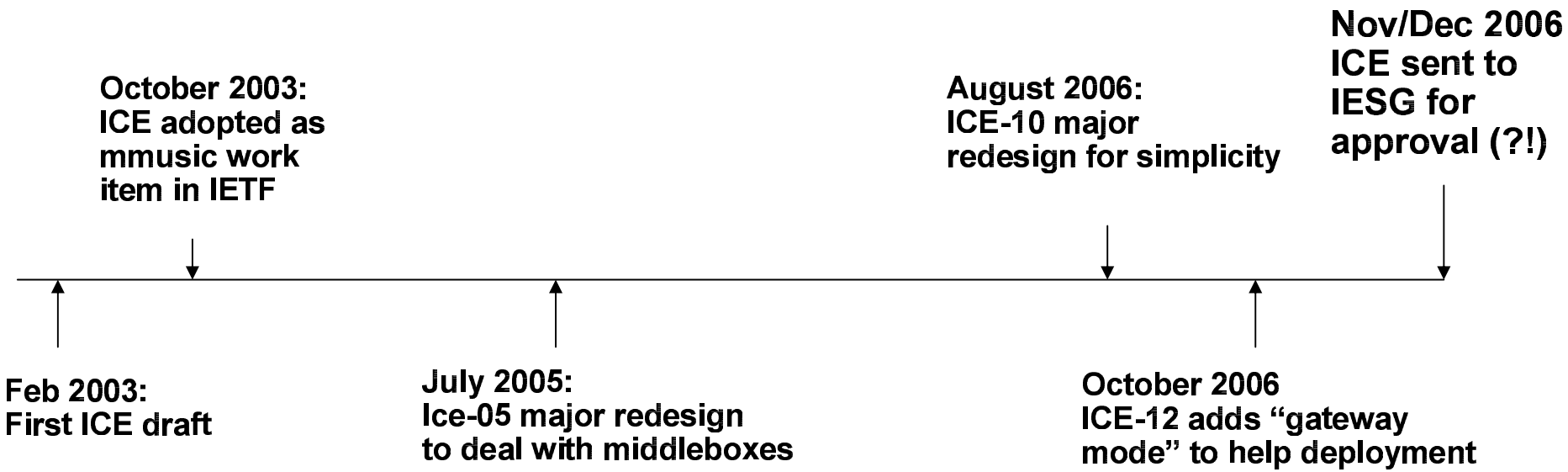


IETFs Answer: Interactive Connectivity Establishment (ICE)

1. ICE makes use of Simple Traversal Underneath NAT (STUN) and Traversal Using Relay NAT (TURN)
2. ICE is a form of p2p NAT traversal
3. ICE only requires a network to provide STUN and TURN servers
4. ICE allows for media to flow even in very challenging network conditions
5. ICE can make sure the phone doesn't ring unless media connectivity exists
6. ICE dynamically discovers the shortest path for media to travel between endpoints
7. ICE has a side effect of eliminating a key DoS attack on SIP (Voice Hammer)
8. ICE works through nearly any type of NAT and firewall
9. ICE does not require the endpoint to discover the NATs, their type, or their presence
10. ICE only uses relays in the worst case – when BOTH sides are behind symmetric NAT

Top 10 ICE Facts

ICE History and Timeline

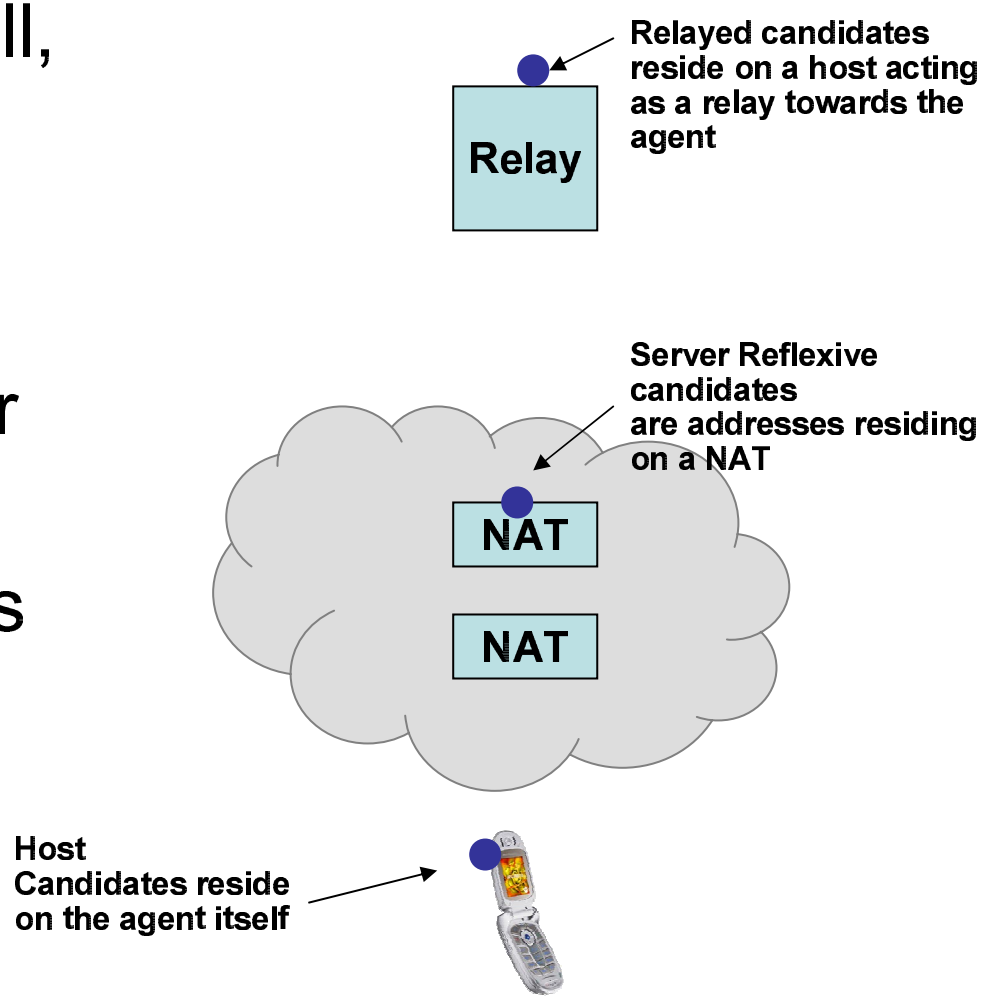


The ICE 9-Step Program to Recovery

- Step 1: Allocation
- Step 2: Prioritization
- Step 3: Initiation
- Step 4: Allocation
- Step 5: Information
- Step 6: Verification
- Step 7: Coordination
- Step 8: Communication
- Step 9: Confirmation

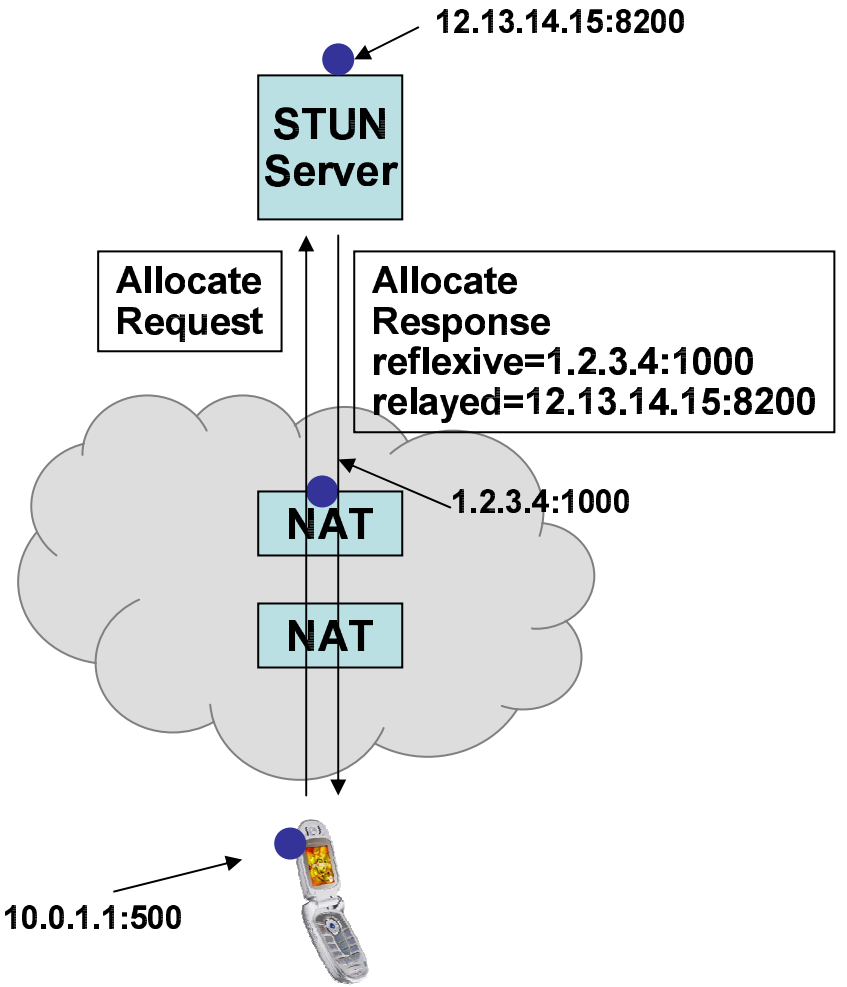
ICE Step 1: Allocation

- Before Making a Call, the Client Gathers Candidates
- Each candidate is a potential address for receiving media
- Three different types of candidates
  - Host Candidates
  - Server Reflexive Candidates
  - Relayed Candidates



Using STUN to Obtain Candidates

- Server reflexive and relayed candidates are learned by talking to a STUN server using the Relay Usage
- Client sends query to STUN relay server
- Query passes through NAT, creates bindings
- STUN relay server allocates a relayed address and also reports back source address of request to client
  - This will be the server reflexive address



Components and Media Streams

- Certain types of media streams require multiple IP addresses and ports
  - Primary example: RTP and RTCP
- Consequently, each media stream is said to have a certain number of components
  - Two for RTP
- Each component has a component-ID
  - RTP = 1
  - RTCP = 2
- Each candidate is associated with a particular component

Pacing of Allocations

- If a client has
  - Multiple interfaces
  - Multiple IP address versions
  - Multiple STUN servers
  - Multiple media streams
  - Multiple components
- This can produce a lot of allocation traffic
- Two problems
  - Network congestion
  - NAT Overload

- NAT Overload has been reported in the wild – NATs fail to maintain bindings when created too fast
- For this reason, ICE paces allocations at 1 transaction every 20ms

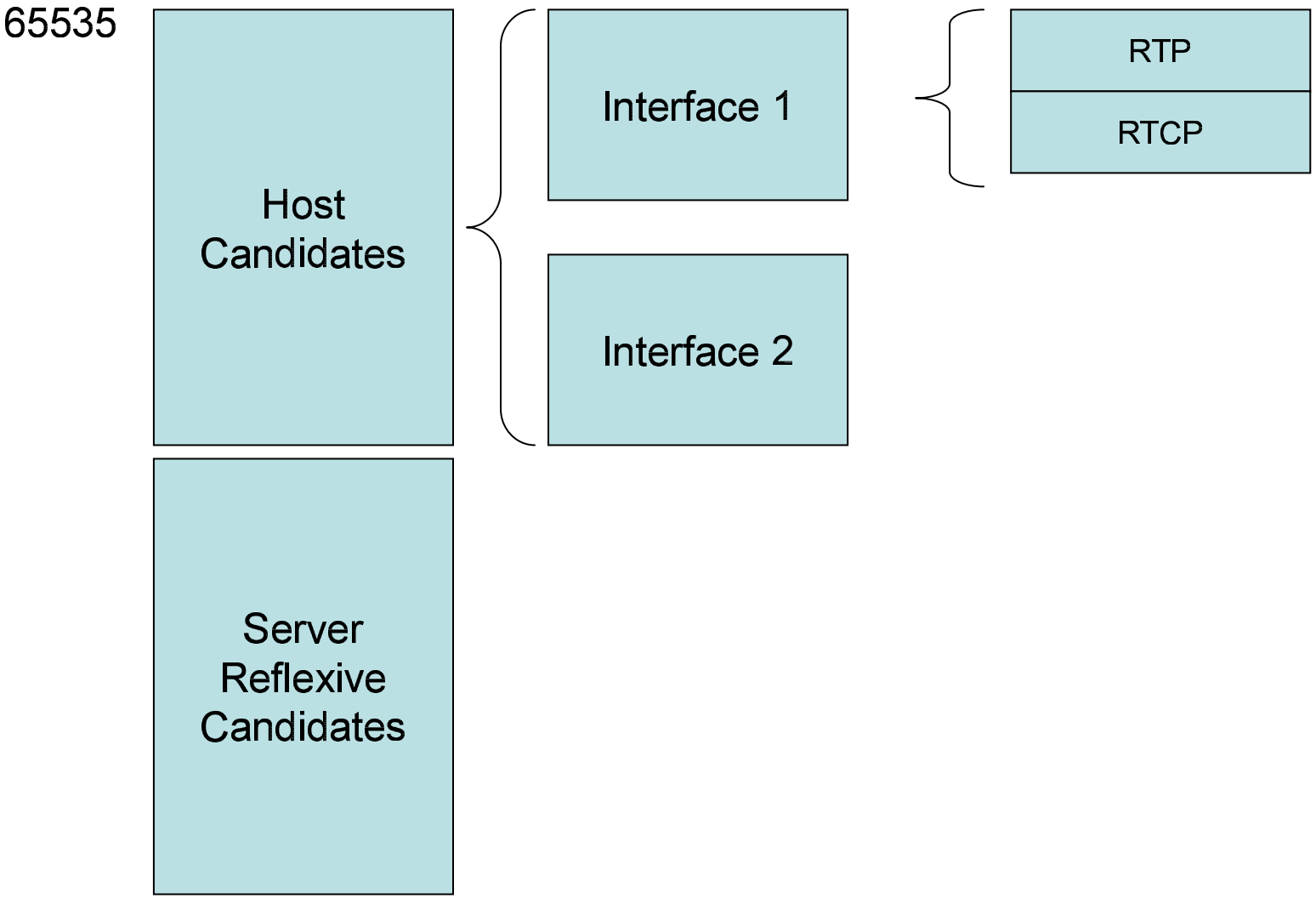
ICE Step 2: Prioritization

```
priority = (2^24)*(type preference)
          +(2^8)*(local preference)
          +(2^0)*(256 - component ID)
```

32 bits

- Type-Preference: Preference for type (host, server reflexive, relayed)
  - Usually 0 for relayed, 126 for host
- Local Preference: Amongst candidates of same type, preference for them
  - If host is multihomed, preference by interface
  - If host has multiple STUN servers, preference for that server
- Component ID as described previously
- This algorithm is only SHOULD strength

Visualization: Priority Space



Encoding the Offer

- Each candidate is placed into an a=candidate attribute of the offer
- Each candidate line has
  - IP address and port
  - Component ID
  - Foundation
  - Transport Protocol
  - Priority
  - Type
  - “Related Address”

```
v=0
o=jdoe 2890844526 2890842807 IN IP4
10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706178 10.0.1.1
8998 typ local
a=candidate:2 1 UDP 1694498562 192.0.2.3
45664 typ srflx raddr 10.0.1.1 rport 8998
```

Encoding the Offer

- Each candidate is placed into an a=candidate attribute of the offer
- Each candidate line has
  - IP address and port
  - Component ID
  - Foundation
  - Transport Protocol
  - Priority
  - Type
  - “Related Address”

```
v=0
o=jdoe 2890844526 2890842807 IN IP4
10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706178 10.0.1.1
8998 typ local
a=candidate:2 1 UDP 1694498562 192.0.2.3
45664 typ srflx raddr 10.0.1.1 rport 8998
```

Encoding the Offer

- Each candidate is placed into an a=candidate attribute of the offer
- Each candidate line has
  - IP address and port
  - Component ID
  - Foundation
  - Transport Protocol
  - Priority
  - Type
  - “Related Address”

```
v=0
o=jdoe 2890844526 2890842807 IN IP4
10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706178 10.0.1.1
8998 typ local
a=candidate:2 1 UDP 1694498562 192.0.2.3
45664 typ srflx raddr 10.0.1.1 rport 8998
```

Encoding the Offer

- Each candidate is placed into an a=candidate attribute of the offer
- Each candidate line has
  - IP address and port
  - Component ID
  - Foundation
  - Transport Protocol
  - Priority
  - Type
  - “Related Address”

```
v=0
o=jdoe 2890844526 2890842807 IN IP4
10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706178 10.0.1.1
8998 typ local
a=candidate:2 1 UDP 1694498562 192.0.2.3
45664 typ srflx raddr 10.0.1.1 rport 8998
```

Foundation is the same for all candidates Of the same type, from the same interface And STUN server. Used as part of the Frozen algorithm (later)

Encoding the Offer

- Each candidate is placed into an a=candidate attribute of the offer
- Each candidate line has
  - IP address and port
  - Component ID
  - Foundation
  - Transport Protocol
  - Priority
  - Type
  - “Related Address”

```
v=0
o=jdoe 2890844526 2890842807 IN IP4
10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706178 10.0.1.1
8998 typ local
a=candidate:2 1 UDP 1694498562 192.0.2.3
45664 typ srflx raddr 10.0.1.1 rport 8998
```

Only UDP defined in ICE-12.
Draft-ietf-mmusic-ice-tcp
defines several TCP types and TLS

Encoding the Offer

- Each candidate is placed into an a=candidate attribute of the offer
- Each candidate line has
  - IP address and port
  - Component ID
  - Foundation
  - Transport Protocol
  - Priority
  - Type
  - “Related Address”

```
v=0
o=jdoe 2890844526 2890842807 IN IP4
10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706178 10.0.1.1
8998 typ local
a=candidate:2 1 UDP 1694498562 192.0.2.3
45664 typ srflx raddr 10.0.1.1 rport 8998
```

Encoding the Offer

- Each candidate is placed into an a=candidate attribute of the offer
- Each candidate line has
  - IP address and port
  - Component ID
  - Foundation
  - Transport Protocol
  - Priority
  - Type
  - “Related Address”

```
v=0
o=jdoe 2890844526 2890842807 IN IP4
10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706178 10.0.1.1
8998 typ local
a=candidate:2 1 UDP 1694498562 192.0.2.3
45664 typ srflx raddr 10.0.1.1 rport 8998
```

Encoding the Offer

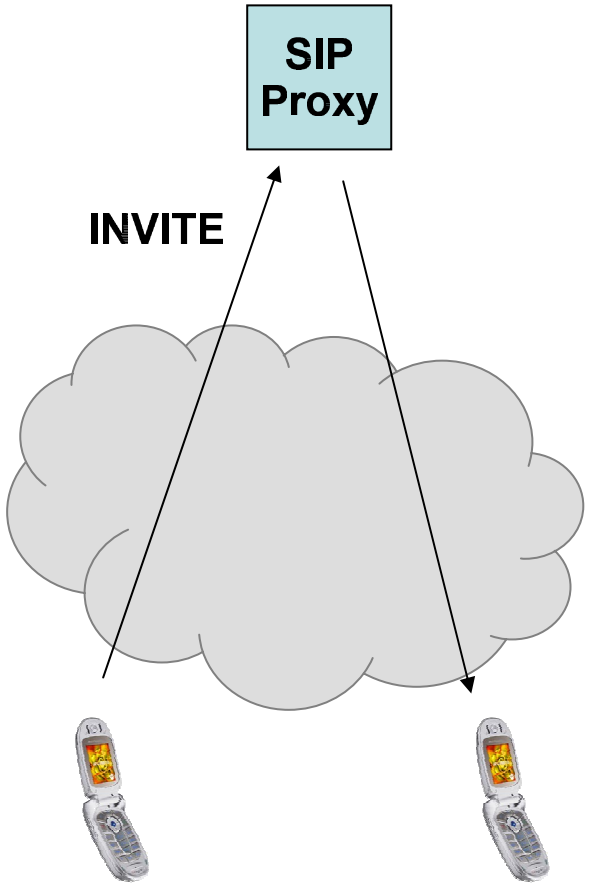
- Each candidate is placed into an a=candidate attribute of the offer
- Each candidate line has
  - IP address and port
  - Component ID
  - Foundation
  - Transport Protocol
  - Priority
  - Type
  - “Related Address”

```
v=0
o=jdoe 2890844526 2890842807 IN IP4
10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706178 10.0.1.1
8998 typ local
a=candidate:2 1 UDP 1694498562 192.0.2.3
45664 typ srflx raddr 10.0.1.1 rport 8998
```

Optional information. For relayed candidates, gives the server reflexive. For server reflexive, gives the host.

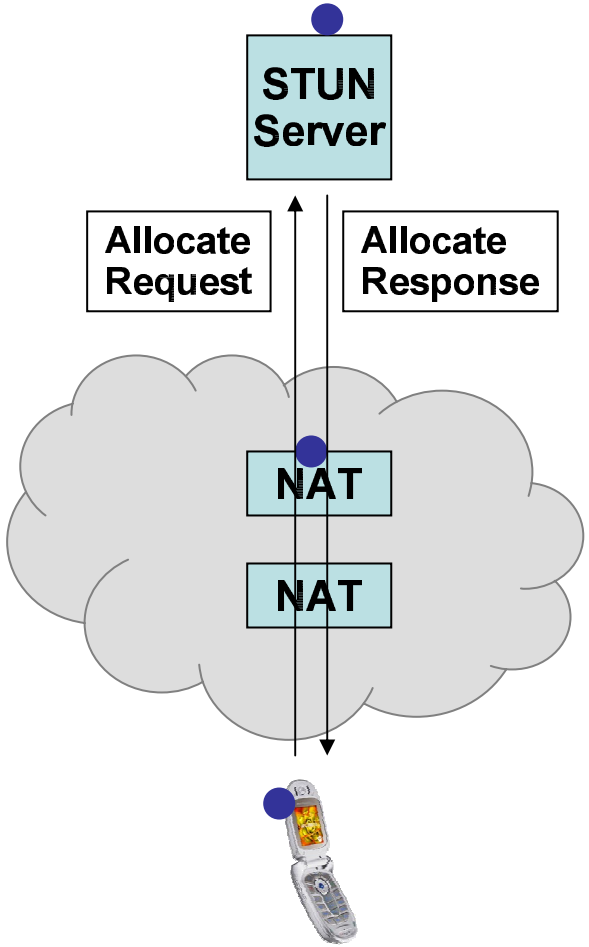
ICE Step 3: Initiation

- Caller sends a SIP INVITE as normal
- No ICE processing by proxies
- SIP itself traverses NAT using SIP outbound and rport



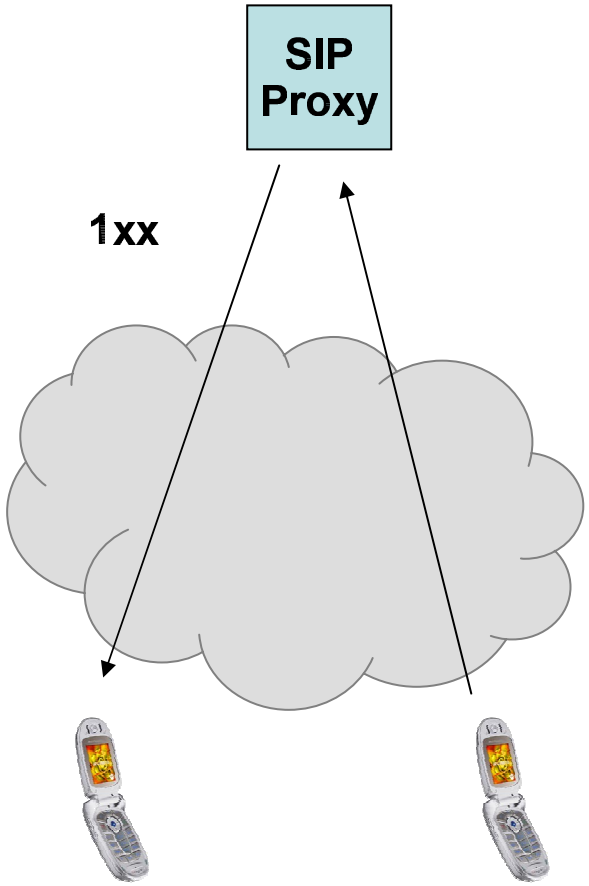
ICE Step 4: Allocation

- Called party does exactly same processing as caller and obtains its candidates
- Recommended to not yet ring the phone!



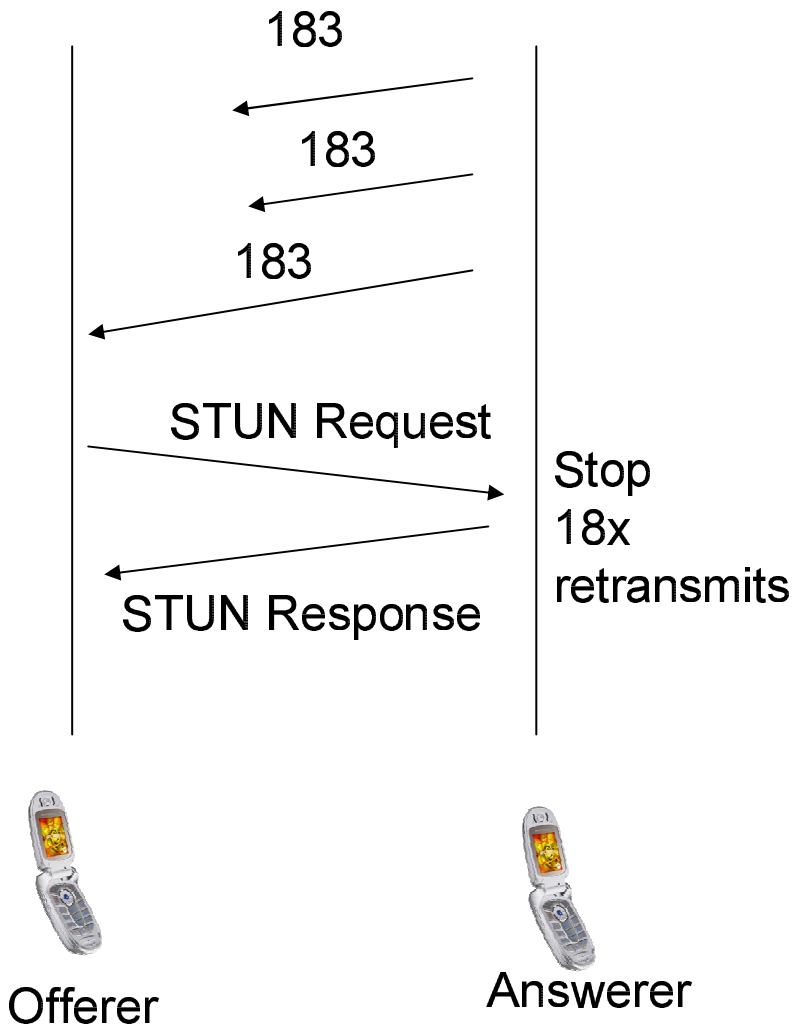
ICE Step 5: Information

- Caller sends a provisional response containing its SDP with candidates and priorities
  - Can also happen in 2xx, but this flow is “best”
- Provisional response is periodically retransmitted
- As with INVITE, no processing by proxies
- Phone has still not rung yet



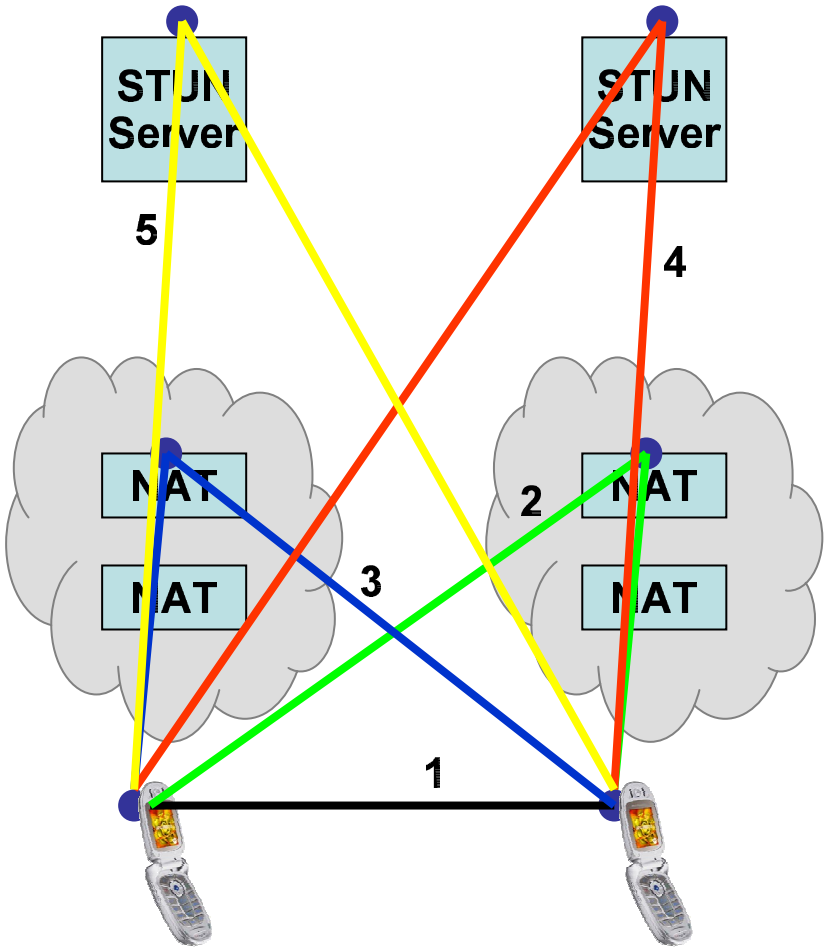
Reliability without PRACK

- ICE allows an optimization for reliable 1xx without PRACK!
- Answerer retransmits 18x as if PRACK was being used
- However, when it receives STUN Binding Request, ceases retransmits
  - Positive indication of receipt of 18x by offerer!
- Admittedly kind of hoakey – using media path message to influence SIP state machinery



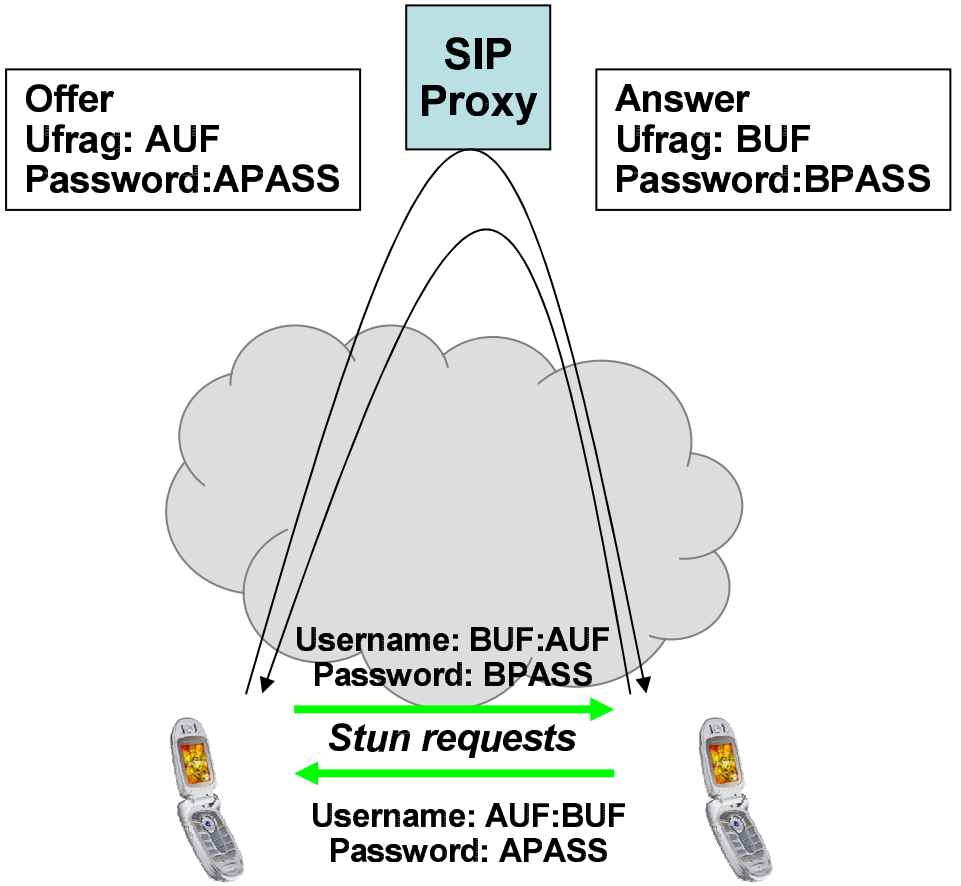
ICE Step 6: Verification

- Each agent pairs up its candidates (local) with its peers (remote) to form candidate pairs
- Each agent sends a connectivity check every 20ms, in pair priority order
  - Binding Request from the local candidate to the remote candidate
- Upon receipt of the request the peer agent generates a response
  - Contains a mapped address indicating the source IP and port seen in the request
- If the response is received the check has succeeded



Authenticating STUN

- STUN Connectivity checks are authenticated and integrity protected
- Authentication is based on a username and password
- Username is constructed by combining username fragments exchanged in offer and answer separated by colon
- Password is exchanged in offer/answer
- Username and password are same for all candidates in a media stream



Pairing up Candidates

O-P: Offerers Priority
A-P: Answerers Priority

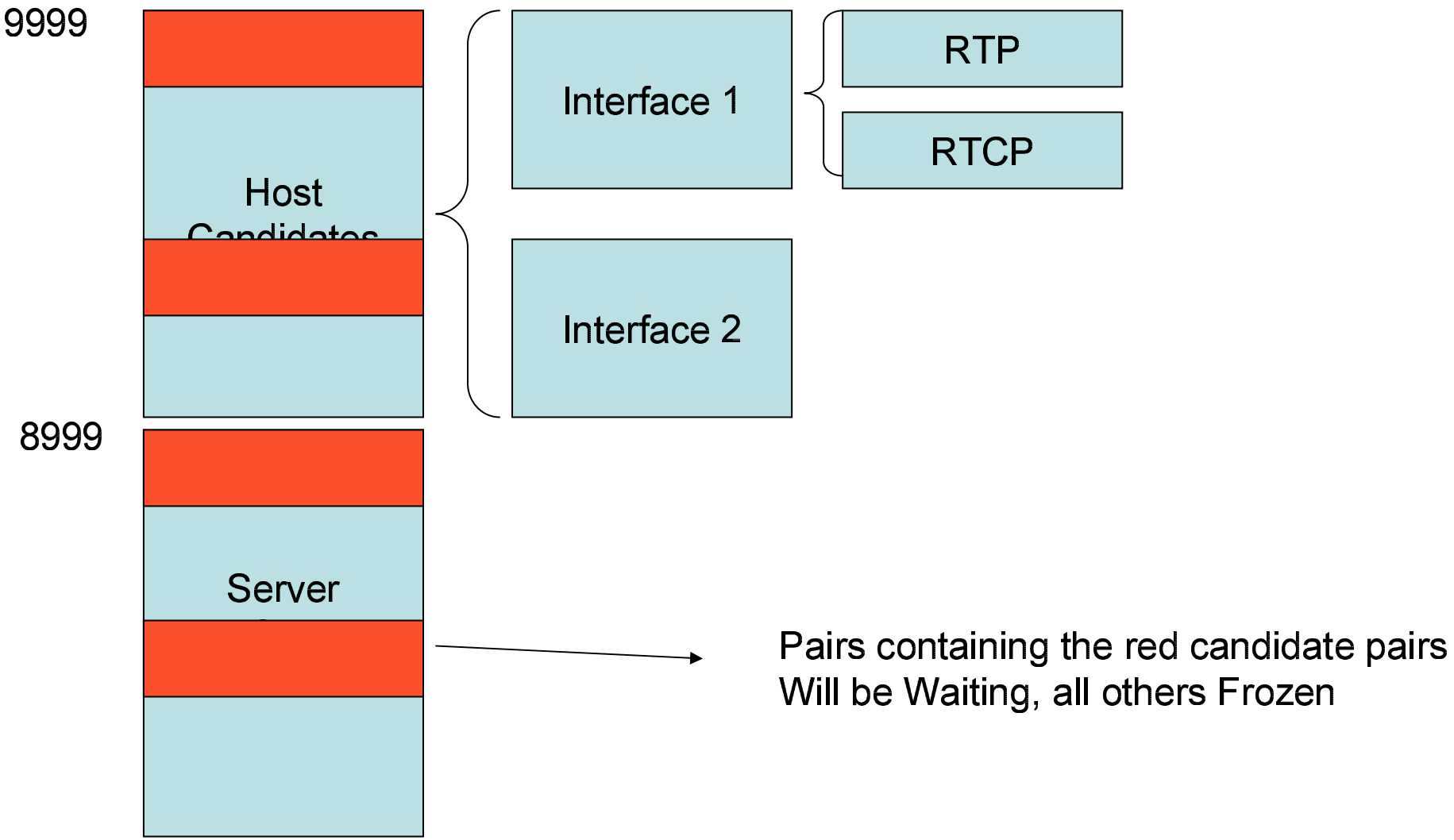
pair priority = 2^32*MIN(O-P,A-P) + 2*MAX(O-P,A-P) + (O-P>A-P?1:0)

- Pairs are sorted in order of decreasing pair priority
- Each agent will end up with the same list
- Last term serves as a tie breaker
- Min/Max results in highest priority for pair with two host RTP candidates, lowest for pair with two relayed RTCP

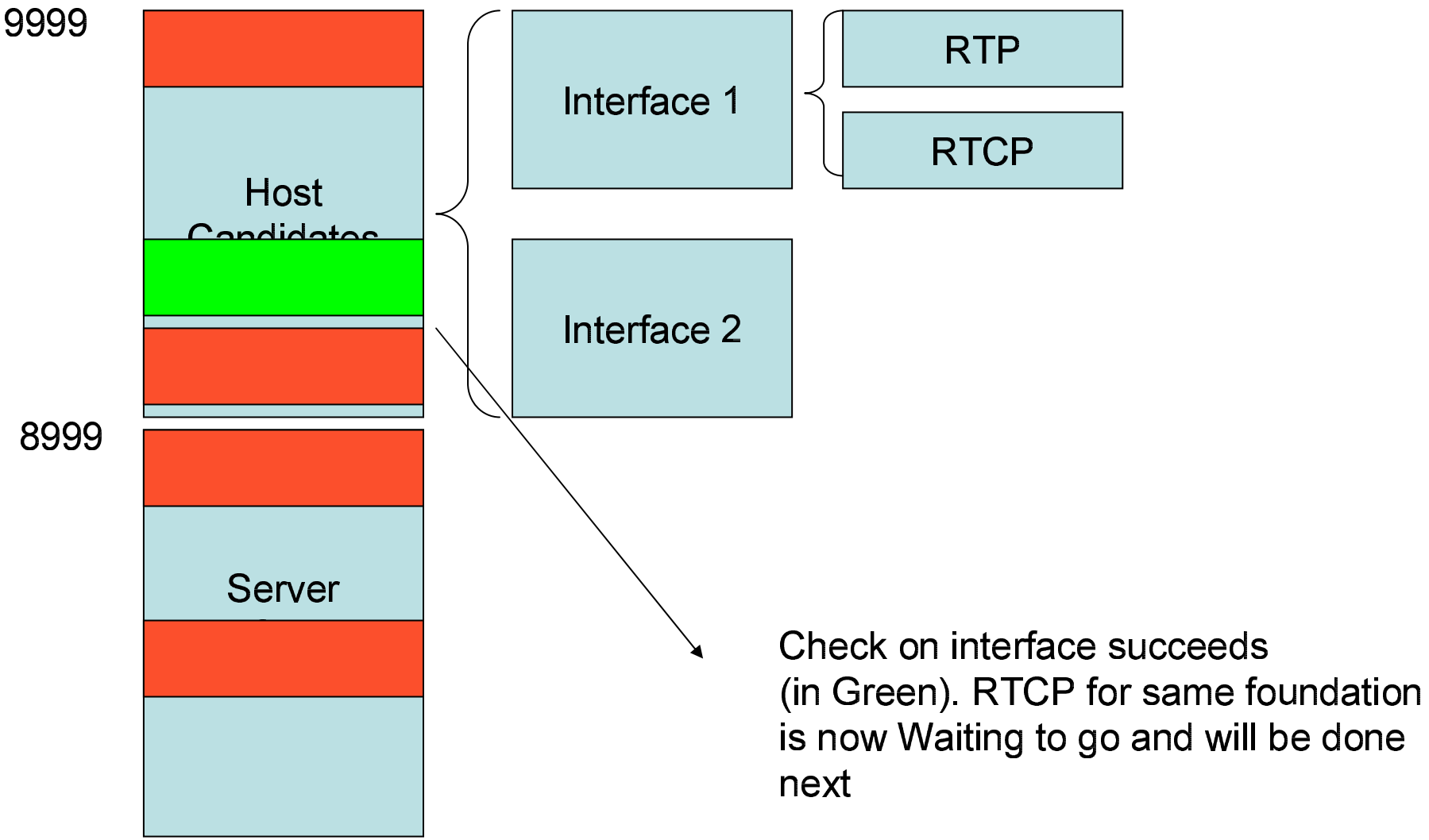
Frozen Algorithm

- ICE provides an optimization called the *Frozen* algorithm
- Applicable when checks need to be done for RTP and RTCP, or when there is multimedia
- Main idea is to use the results of a previous check to predict the likelihood of a future one working
- Basic algorithm
  - First, check the RTP candidate pairs for the audio stream
  - Once one succeeds, then check the similar RTCP candidate
  - If that succeeds, then check the RTP and then RTCP for similar candidates for video
  - Candidates are similar when they are of the same type and obtained from the same interface and STUN server
    - Same foundation

Visualizing Frozen Algorithm

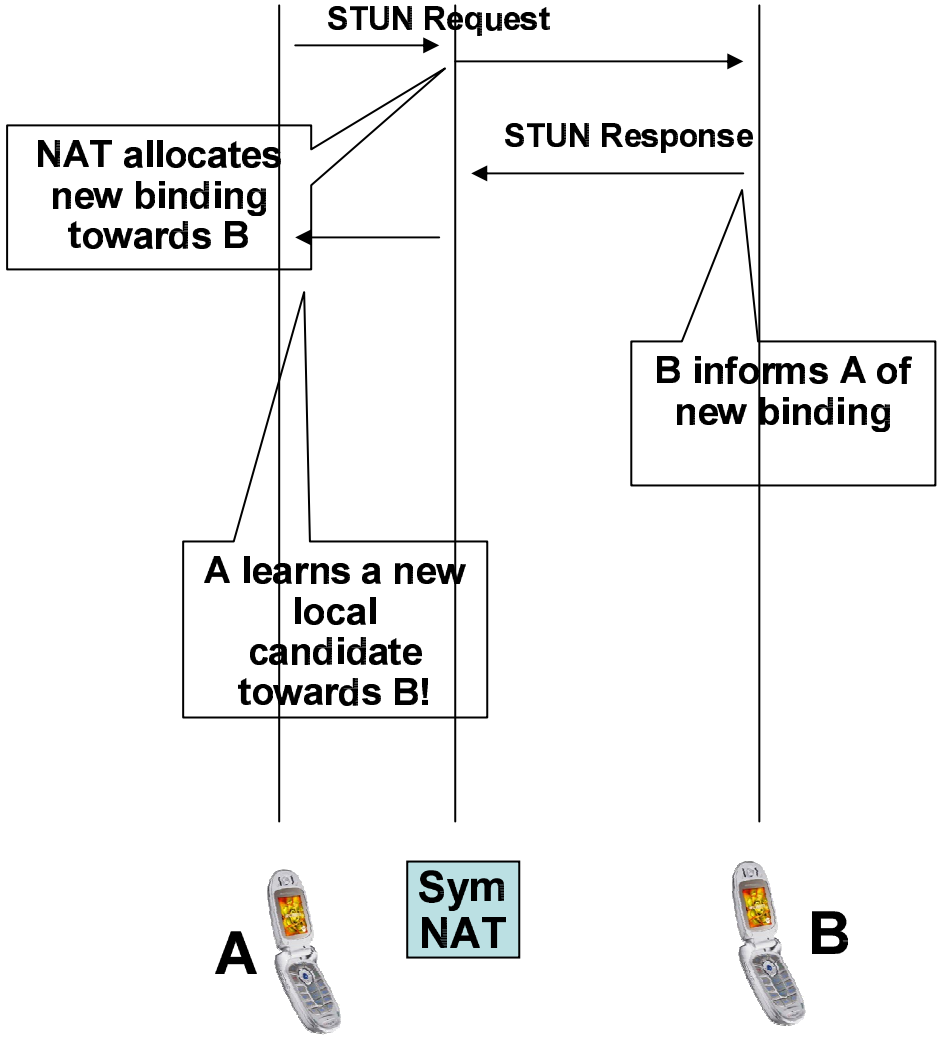


Visualizing Frozen Algorithm



Peer Reflexive Candidates

- Connectivity checks can produce additional candidates
  - Peer reflexive candidates
- Typically happens when there is a symmetric NAT between users
- Peer reflexive candidate will be discovered by both users
  - For user A, from the Response
  - For user B, from the Request
- Allows direct media even in the presence of symmetric NAT!



ICE Step 7: Coordination

- ICE needs to finalize on a candidate pair for each component of each media stream
  - More than one may work
- Each agent needs to conclude on the same set of pairs
- Finalization takes place without SIP signaling – all through STUN

Agent Roles

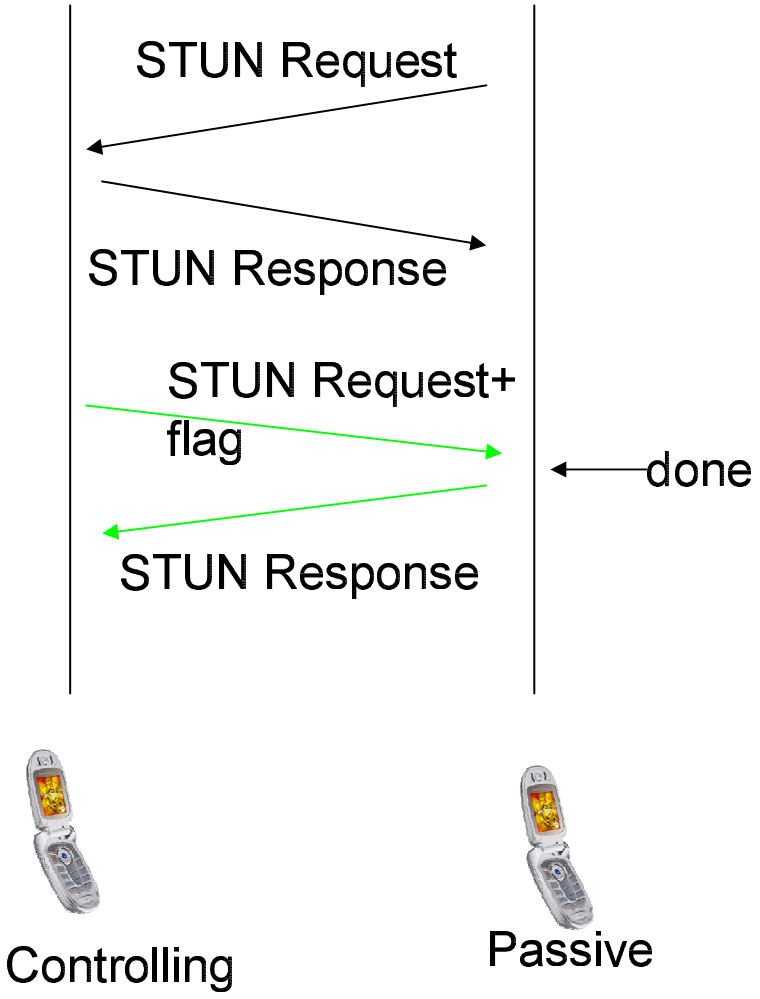
- One agent acts as the controlling agent, the other as the passive agent
- Controlling agent is normally the offerer, unless offerer signals it only supports passive role (see later)
- Controlling agent responsible for
  - Deciding when STUN checks should finish
  - Deciding which pairs to use once it is finished

Why not just use the first pair?

- ICE checks proceed in priority order
  - So why not just stop once the first check succeeds, and use that?
- Several reasons
  - Packet loss on a higher priority check may delay it from finishing – giving checks more time may produce better results
  - An agent may have other criteria for choosing pairs (for example – RTT estimates!)

Signaling Completion

- When controlling agent is done, it inserts a flag into a STUN check
- If passive agent had successfully completed a check in reverse direction, it stops checks for that component of that stream
- Both agents use the pair generated by the check that included the flag
- When ‘done’ – ring the phone!

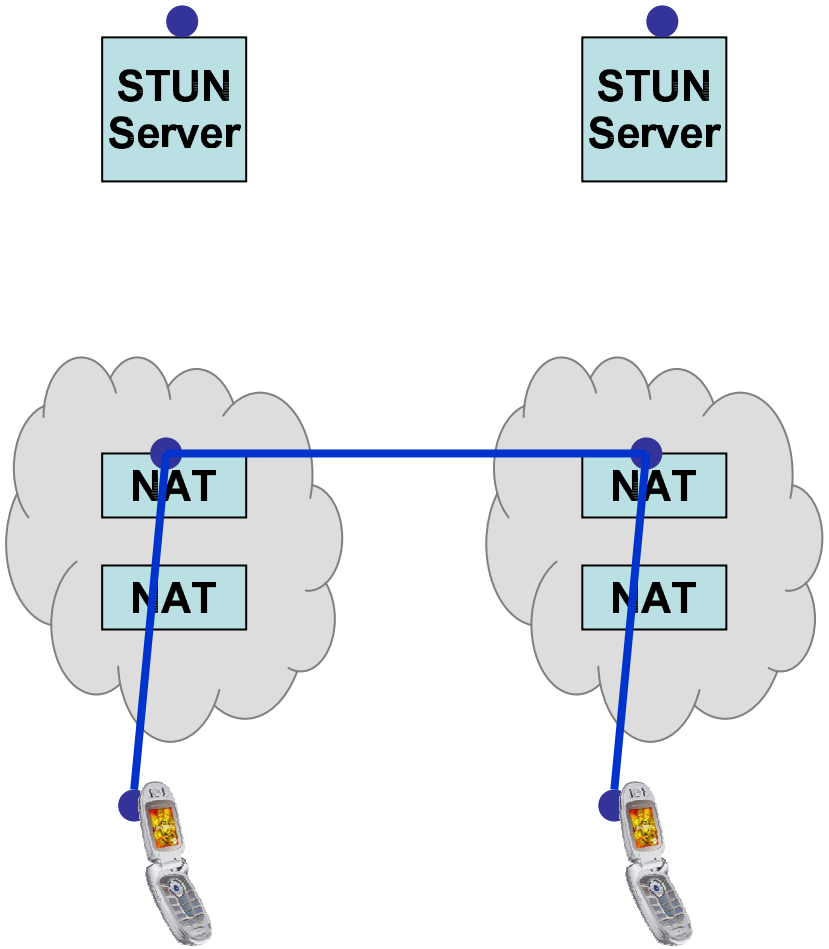


ICE for Gateways

- ICE Supports a modality known as “passive-only”
- Used for endpoints that always have public IP
  - PSTN gateways
  - Media servers
  - Conference servers
- These endpoints need to run ICE for ICE to be used, but don’t themselves have a “NAT problem”
- An agent signals its “passive-only” in SDP
- If both agents are “passive-only” ICE is not used at all
- A passive agent has a single candidate (host only) and only needs to
  - Receive a STUN check and send a response
  - Generate “triggered checks” in the reverse direction
  - No state machinery or pair priorities or anything else

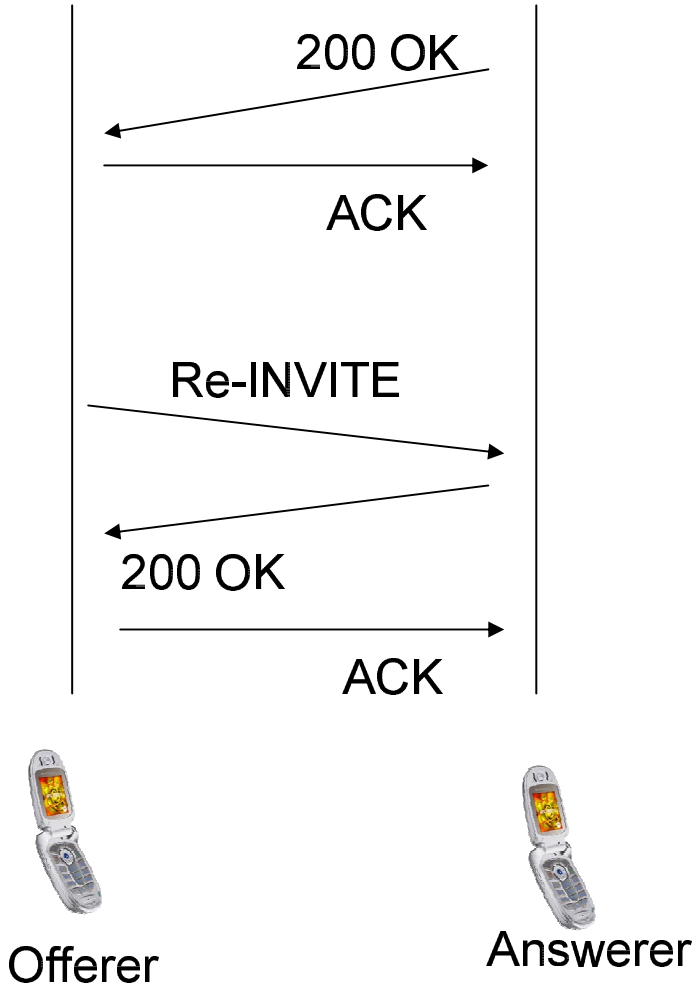
ICE Step 8: Communication

- Media can flow in each direction once pairs have been selected by the controlling agent for each component
- Allows “early media” in both directions



ICE Step 9: Confirmation

- 200 OK and ACK work as normal
  - 200 mirrors SDP from provisional
- If m/c-line in original INVITE didn't match candidate pairs selected by ICE, controlling agent does a re-INVITE to place them in m/c-line
- Re-INVITE ensures that 'middleboxes' have the correct media address
  - QoS installation (i.e., IMS or Packetcable)
  - Diagnostic tools
  - Monitoring applications
  - Firewalls



Questions?