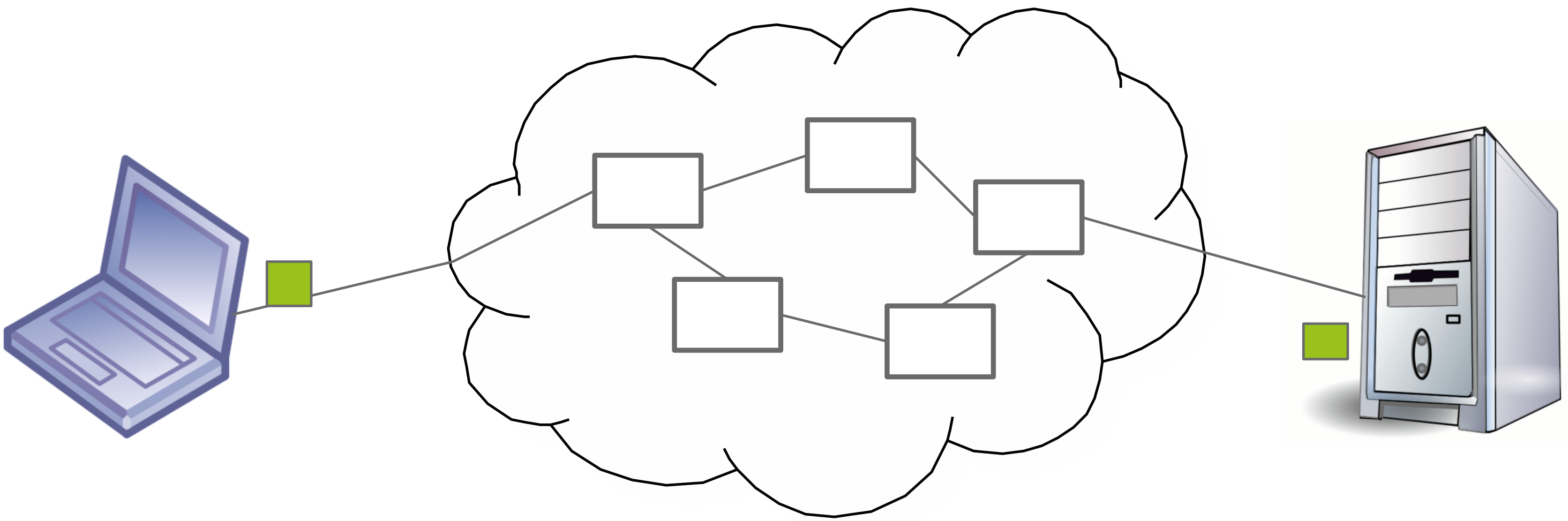
Network Address Translators (NATs) and NAT Traversal

Ari Keränen
ari.keranen@ericsson.com
Ericsson Research Finland, NomadicLab

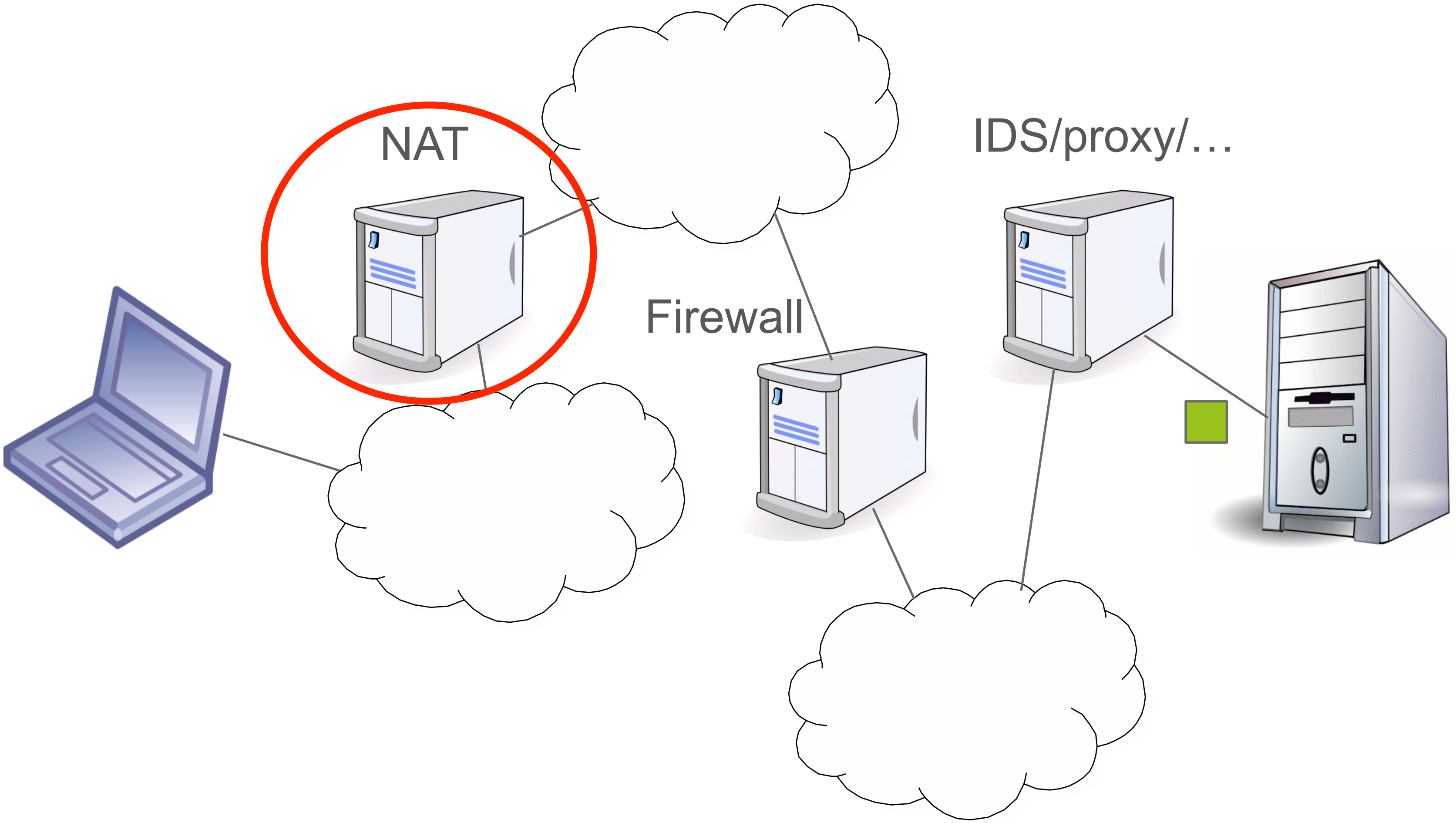
Outline

- Introduction to NATs
- NAT Behavior
  - UDP
  - TCP
- NAT Traversal
  - STUN
  - TURN
  - ICE
  - Others
- NAT64

Internet Back in the Good Old Days

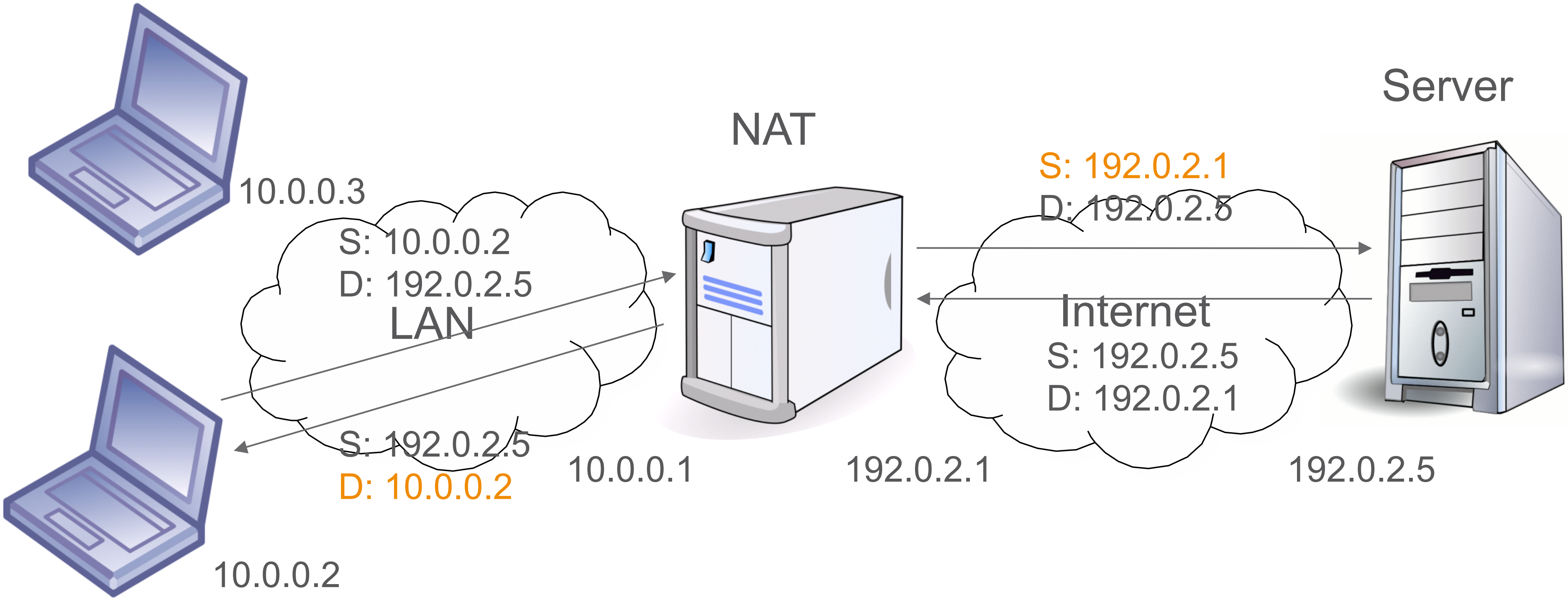


Internet Today (in practice)



Origin of NATs

- Created to resolve the IPv4 address exhaustion problem
- Designed with the web in mind
  - Client/server paradigm



Different Kind of NATs

- ”Basic” Network Address Translator
  - Translates just the IP address in the packets
  - Requires multiple addresses from the NAT
    - One for each host concurrently communicating with the outside networks
  - Very uncommon today
- Network Address and Port Translator (NAPT)
  - Uses also transport layer (TCP/UDP) ports for multiplexing connections
  - Most of the current NATs are of this type
- NAT64
  - More about this later

Side-effects of NATs

- Hosts behind NATs are not reachable from the public Internet
  - Sometimes used to implement security (badly)
  - Breaks peer-to-peer (P2P, as opposed to client/server) applications
- NATs attempt to be transparent
  - Troubleshooting becomes more difficult
- NATs have state → single point of failure
- NATs may try to change addresses also in the payload (and possibly break application layer protocols)
- NATs' behavior is not deterministic
  - Difficult to build applications that work through NATs

Outline

- Introduction to NATs
- NAT Behavior
  - UDP
  - TCP
- NAT Traversal
  - STUN
  - TURN
  - ICE
  - Others
- NAT64

IETF NAT Behavior Recommendations

- Two RFCs describing how NATs **should** behave
  - RFC 4787: Network Address Translation (NAT) Behavioral Requirements for Unicast UDP
  - RFC 5382: NAT Behavioral Requirements for TCP
- Classification of current NAT behaviors
  - Old terminology was confusing (full cone, restricted cone, port restricted cone, and symmetric)
- Recommendations for NAT vendors
  - BEHAVE-compliant NATs are deterministic
- Lots of NATs implemented before the recommendations
  - Various kind of behavior found in the wild
  - Not all new NATs comply even today

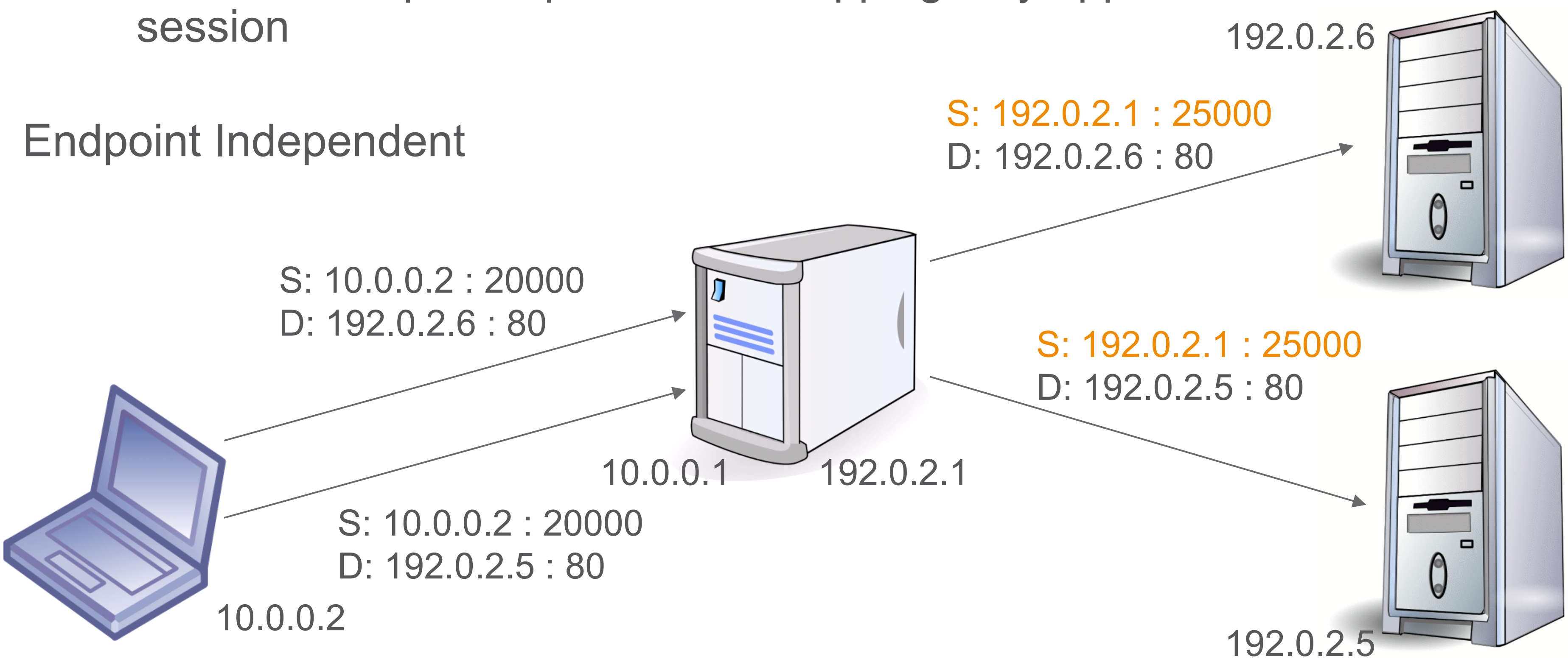
Outline

- Introduction to NATs
- NAT Behavior
  - UDP
  - TCP
- NAT Traversal
  - STUN
  - TURN
  - ICE
  - Others
- NAT64

Mapping Behavior

- For session originated on the same address and port
  - Endpoint independent: same mapping to different sessions
    - MUST use it
  - Address dependent: same mapping to sessions to the same host
  - Address and port dependent: a mapping only applies to one session

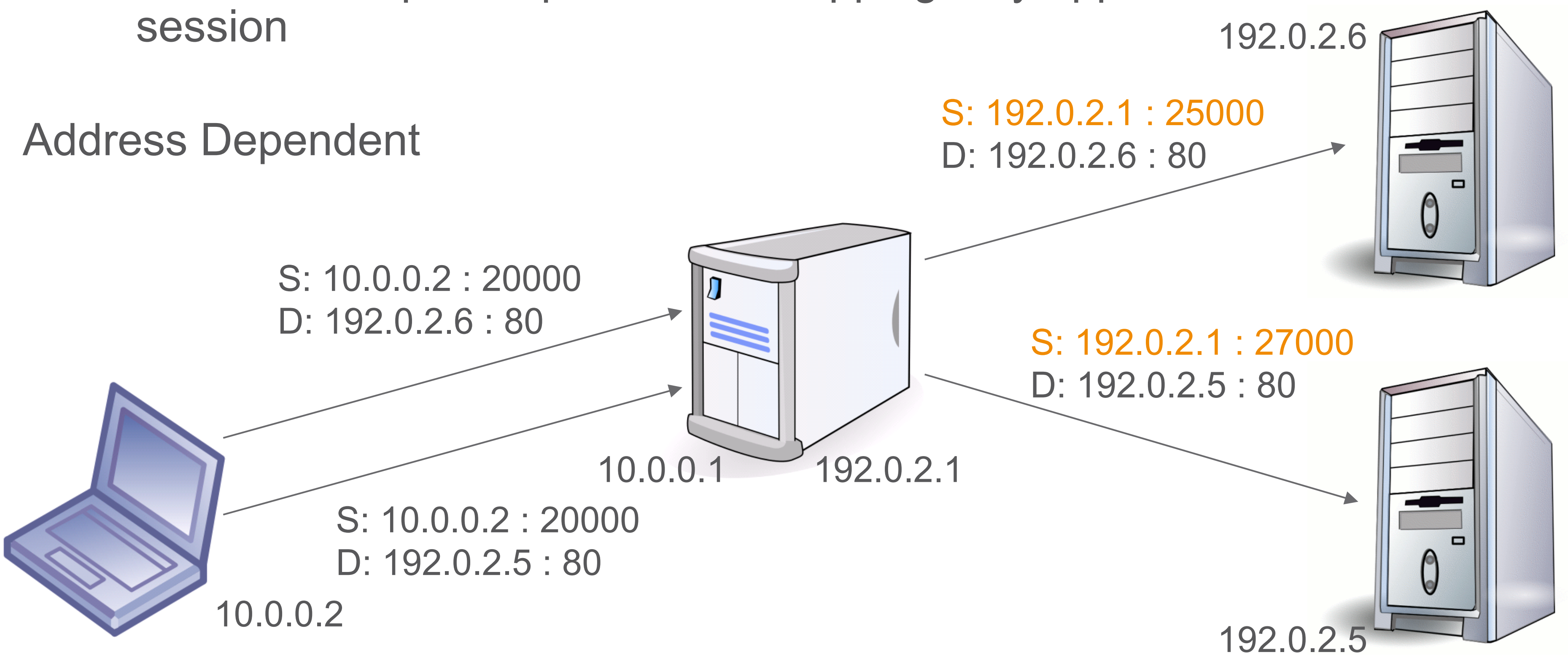
Endpoint Independent



Mapping Behavior

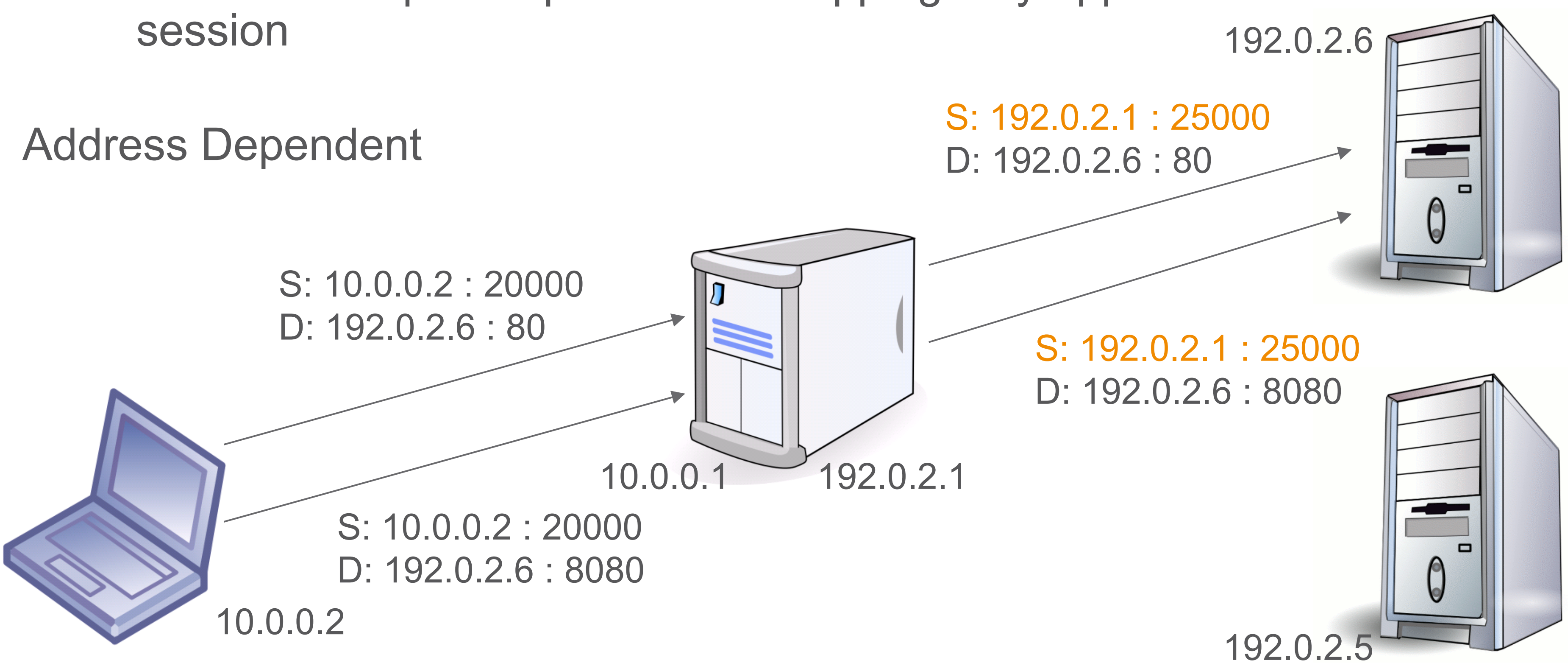
- For session originated on the same address and port
  - Endpoint independent: same mapping to different sessions
    - MUST use it
  - Address dependent: same mapping to sessions to the same host
  - Address and port dependent: a mapping only applies to one session

Address Dependent



Mapping Behavior

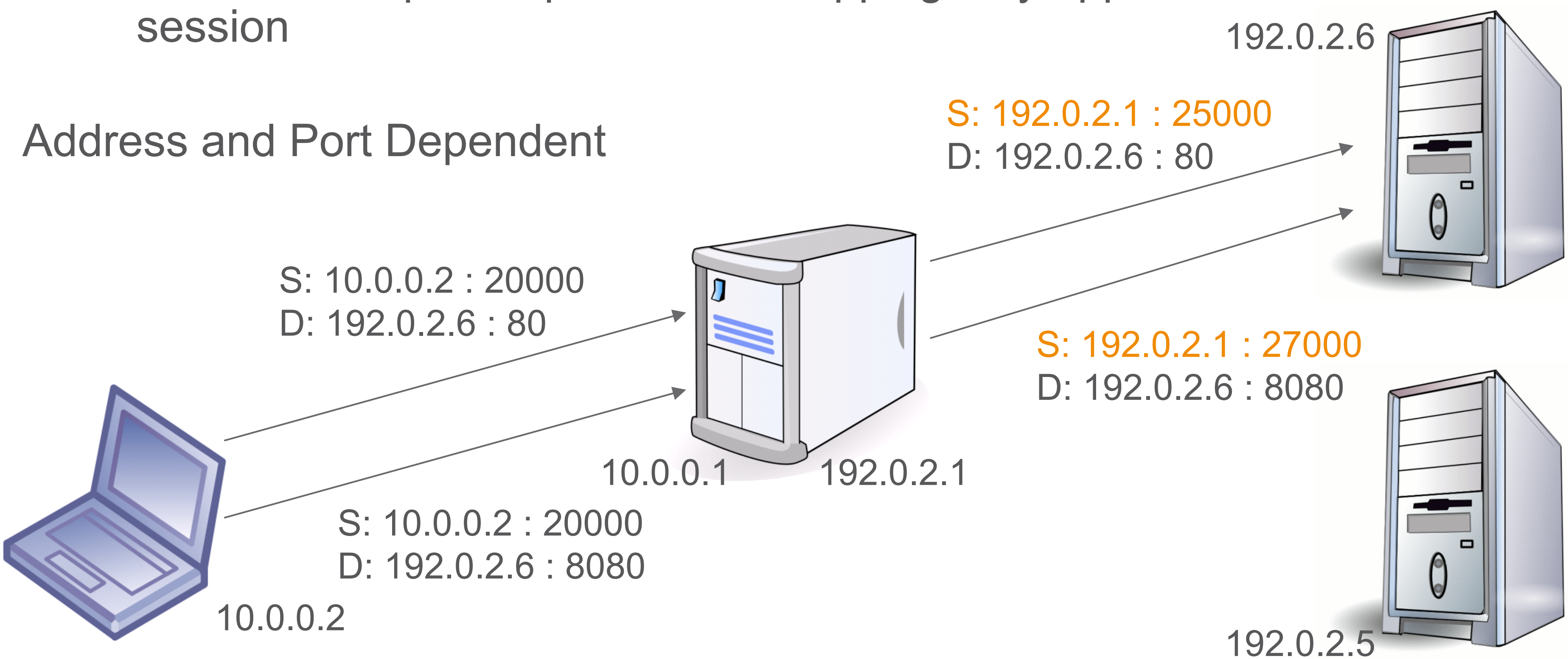
- For session originated on the same address and port
  - Endpoint independent: same mapping to different sessions
    - MUST use it
  - Address dependent: same mapping to sessions to the same host
  - Address and port dependent: a mapping only applies to one session



Mapping Behavior

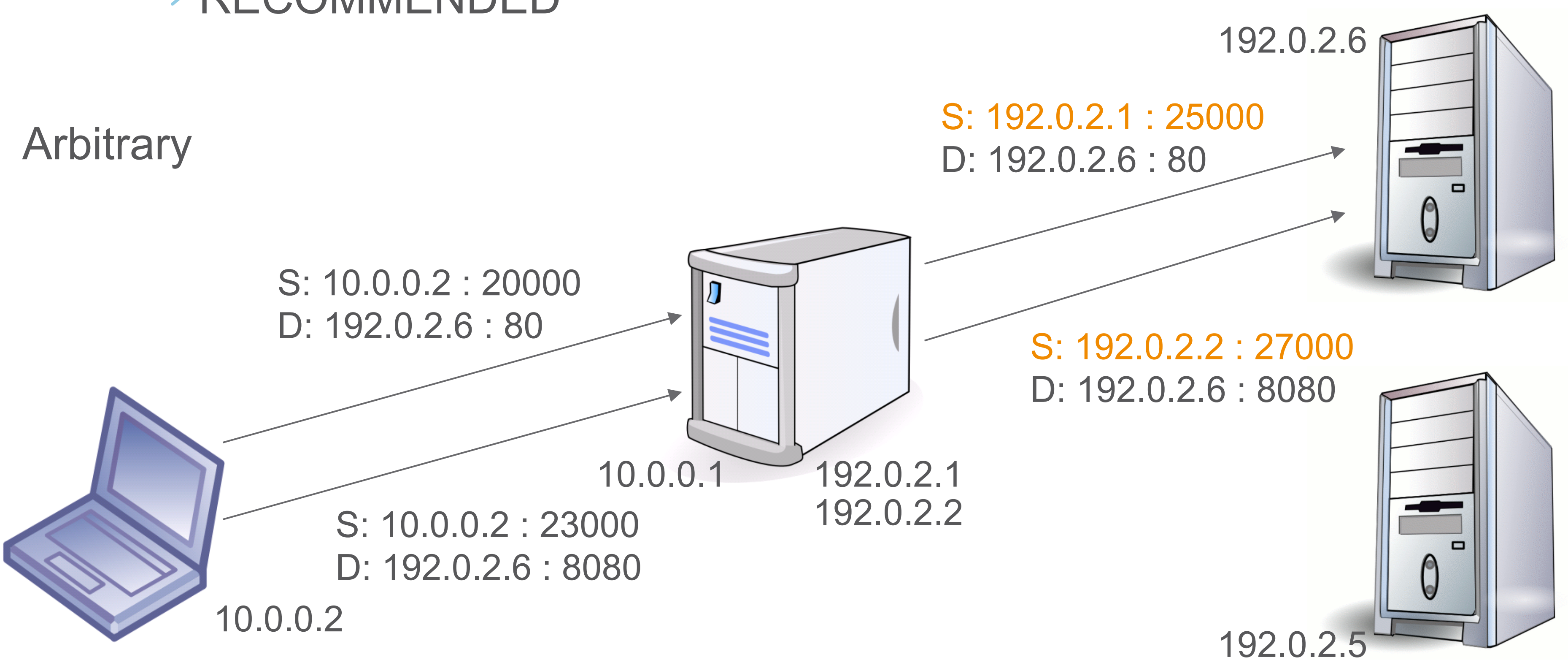
- For session originated on the same address and port
  - Endpoint independent: same mapping to different sessions
    - MUST use it
  - Address dependent: same mapping to sessions to the same host
  - Address and port dependent: a mapping only applies to one session

Address and Port Dependent



IP Address Pooling Behavior

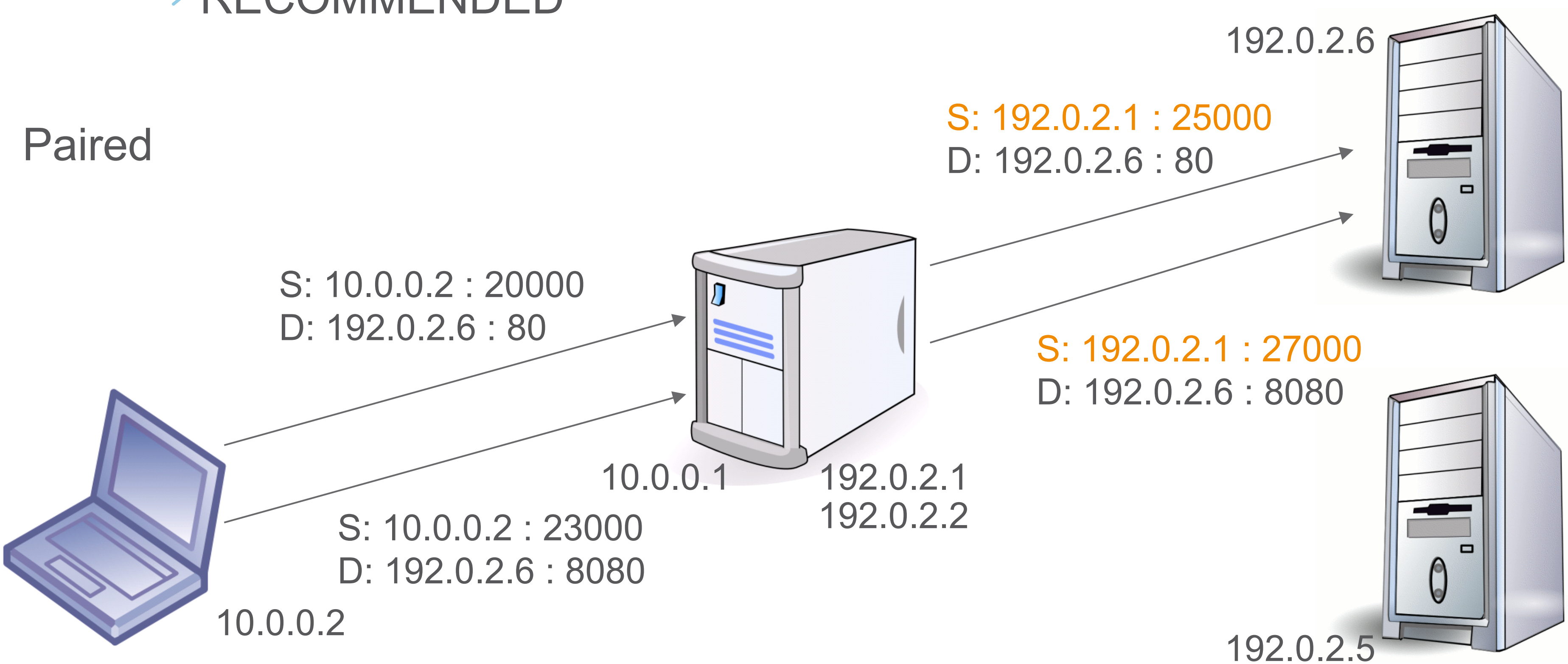
- NATs with a pool of external IP addresses
  - Arbitrary: an endpoint may have simultaneous mappings corresponding to different external IP addresses of the NAT
  - Paired: same external IP address of the NAT
    - RECOMMENDED



IP Address Pooling Behavior

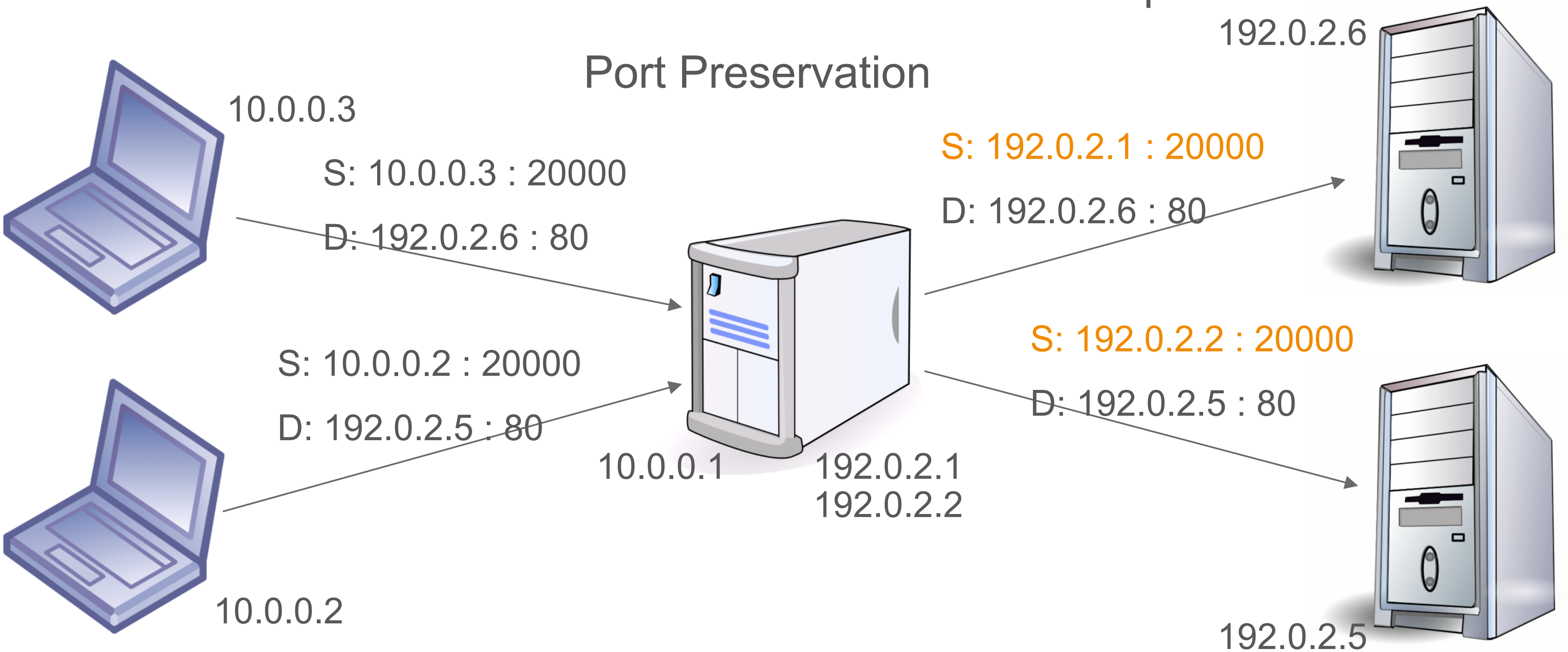
- NATs with a pool of external IP addresses
  - Arbitrary: an endpoint may have simultaneous mappings corresponding to different external IP addresses of the NAT
  - Paired: same external IP address of the NAT
    - RECOMMENDED

Paired



Port Assignment

- Port preservation: preserves the port as long as there are available IP addresses in the NAT's pool
- Port overloading: the port is preserved always, even without available IP addresses in the NAT's pool
  - The NAT translates based on the source of the response

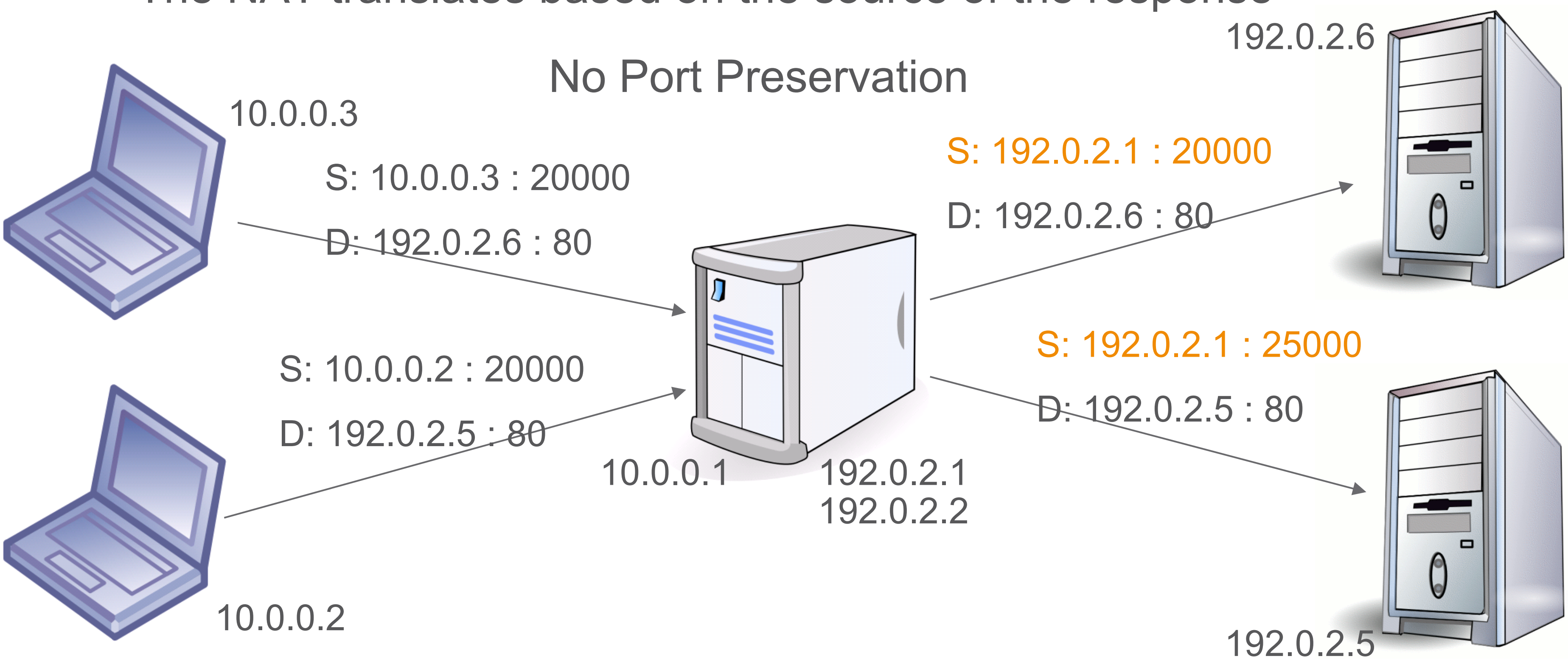


Port Assignment

› Port preservation: preserves the port as long as there are available IP addresses in the NAT's pool

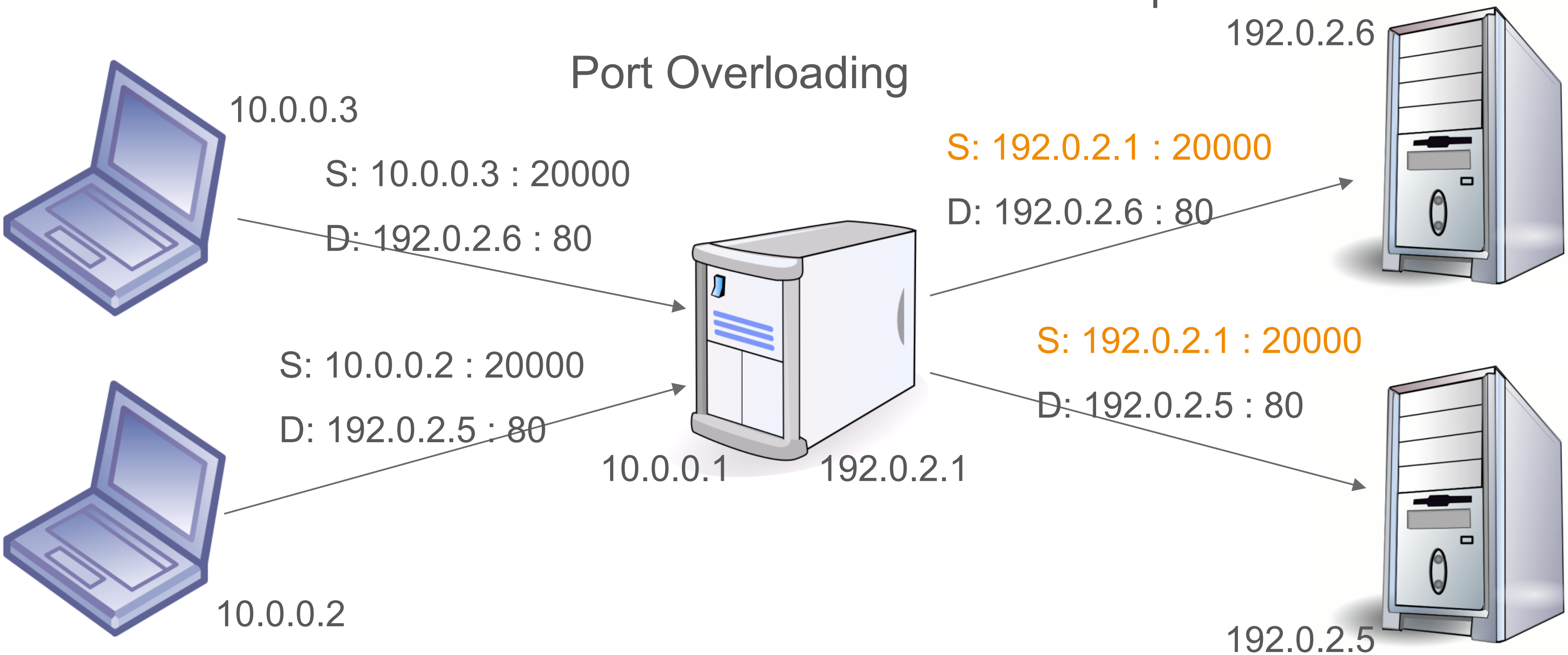
› Port overloading: the port is preserved always, even without available IP addresses in the NAT's pool

  – The NAT translates based on the source of the response



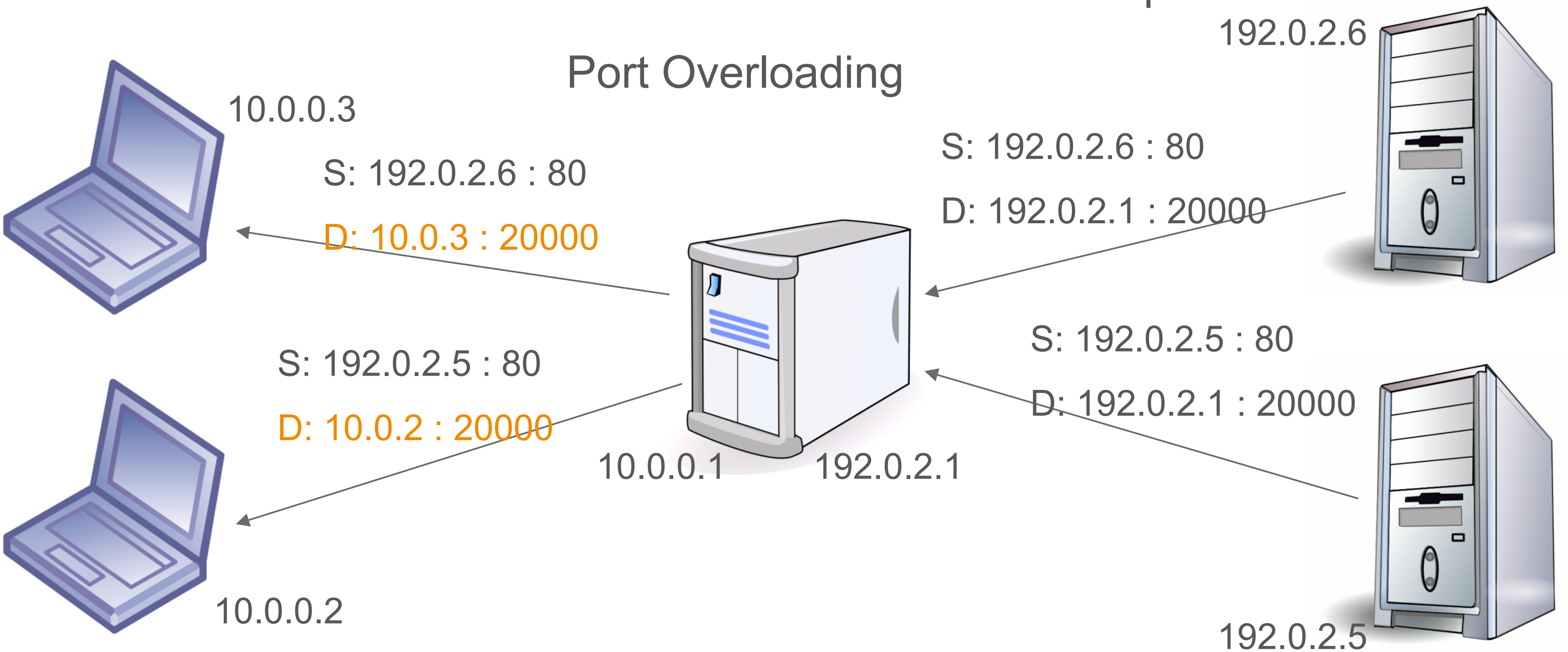
Port Assignment

- Port preservation: preserves the port as long as there are available IP addresses in the NAT's pool
- Port overloading: the port is preserved always, even without available IP addresses in the NAT's pool
  - The NAT translates based on the source of the response



Port Assignment

- Port preservation: preserves the port as long as there are available IP addresses in the NAT's pool
- Port overloading: the port is preserved always, even without available IP addresses in the NAT's pool
  - The NAT translates based on the source of the response



Port Ranges

- 1 – 1023 Well known
- 1024 – 49151 Registered
- 49152 – 65535 Dynamic / Private

- RECOMMENDED to preserve the following ranges
  - 1 – 1023
  - 1024 – 65535

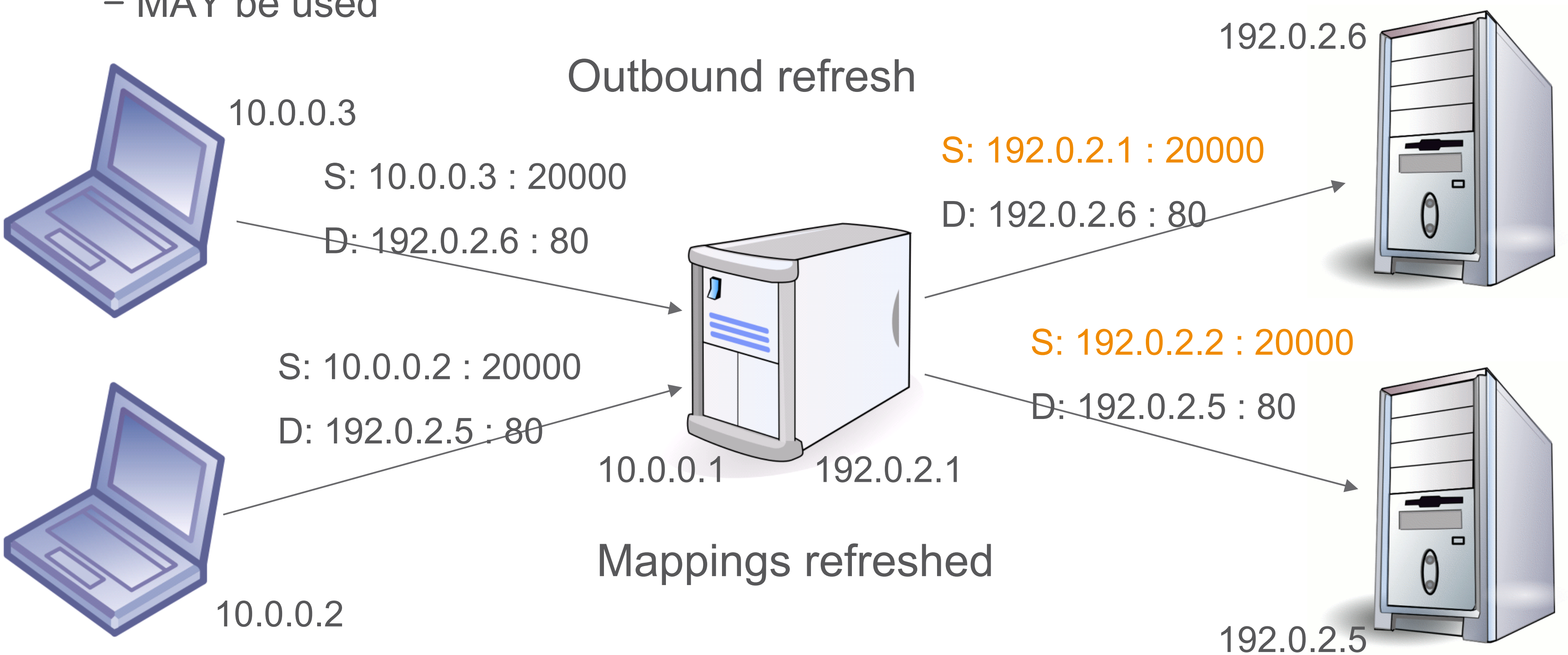
- Port overloading MUST NOT be used
  - Problems when two internal hosts connect to the same external host
- It is RECOMMENDED that NATs preserve port parity (even/odd)
- No requirement for port contiguity

Mapping Timeout

- NAT mappings need to be eventually discarded in order to re-use NAT's public address-port pairs
  - Usually idle connections result in mapping timeout
- NAT UDP mapping MUST NOT expire in less than 2 minutes
- NATs can have application-specific timers
  - Well-known ports
- It is RECOMMENDED to use more than 5 minutes
  - However, ~100 seconds is common and even shorter than 30 second timeouts have been seen in practice

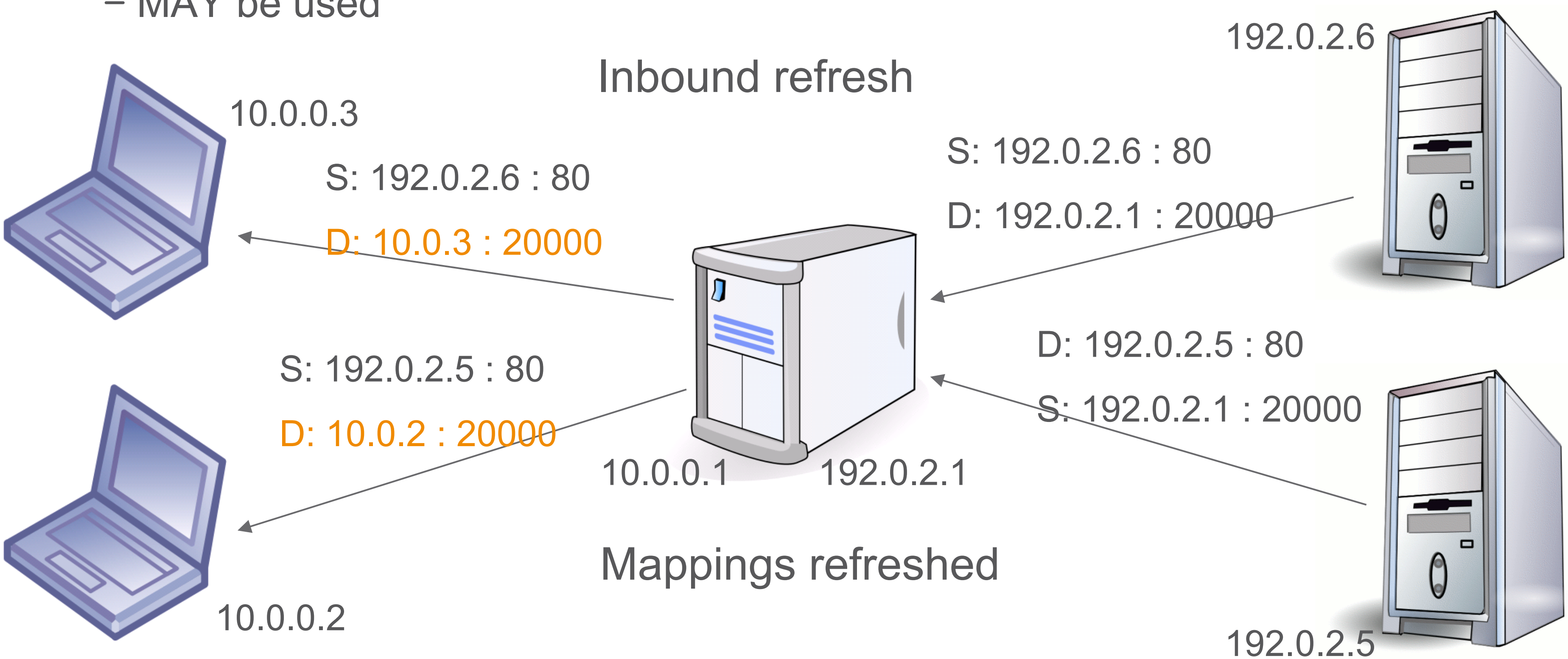
Mapping Refresh

- NAT outbound refresh: packets from the internal to the external interface
  - MUST be used
- NAT inbound refresh: packets from the external to the internal interface (attackers may keep the mapping from expiring)
  - MAY be used



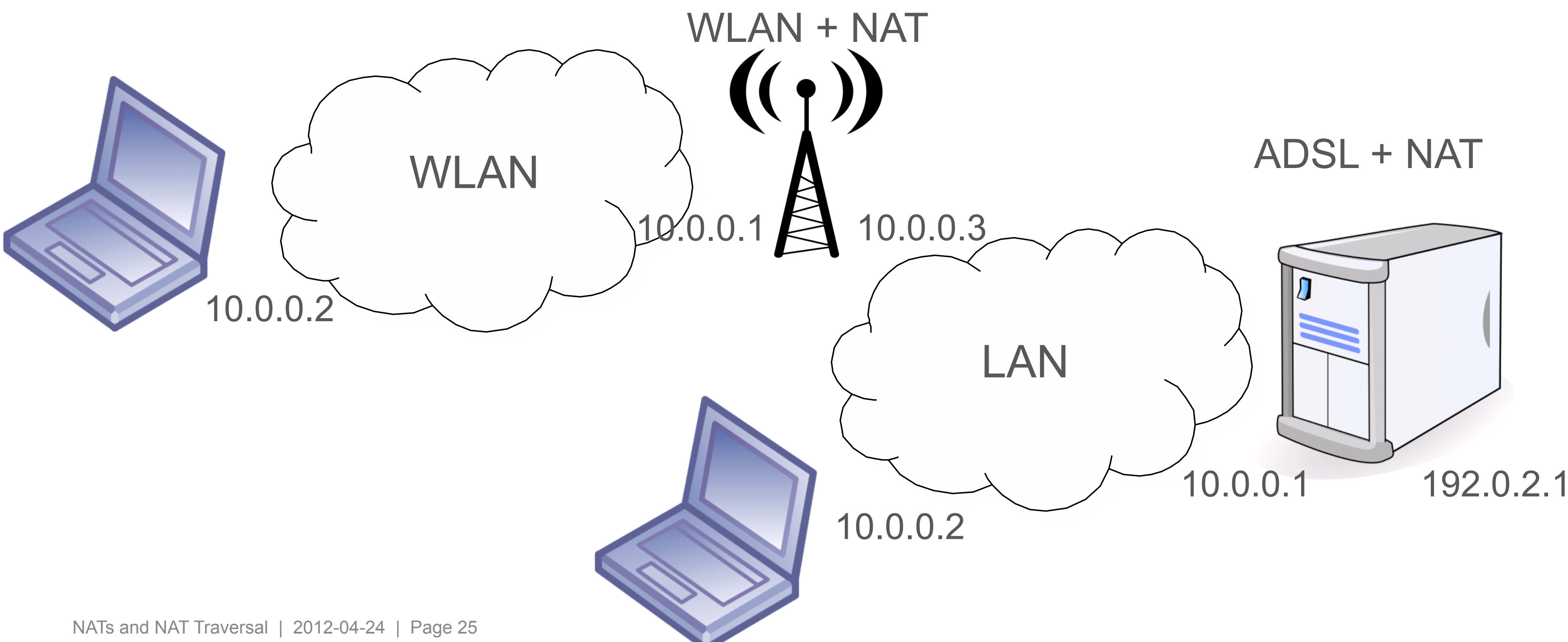
Mapping Refresh

- NAT outbound refresh: packets from the internal to the external interface
  - MUST be used
- NAT inbound refresh: packets from the external to the internal interface (attackers may keep the mapping from expiring)
  - MAY be used



External Address Spaces

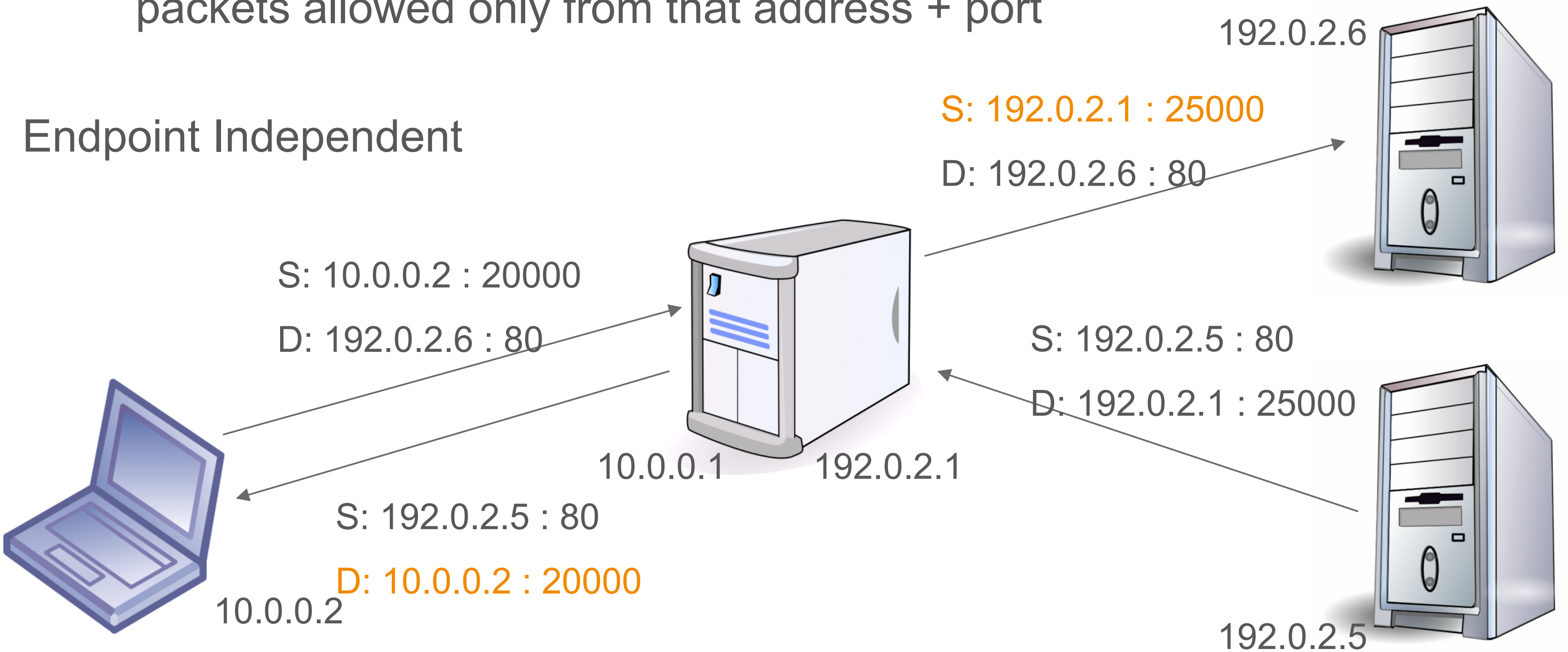
› NATs MUST be able to handle external address spaces that overlap with the internal address space
  - Internal nodes cannot communicate directly with external nodes that have the same address as another internal node
  - However, they can use STUN techniques



Filtering Behavior

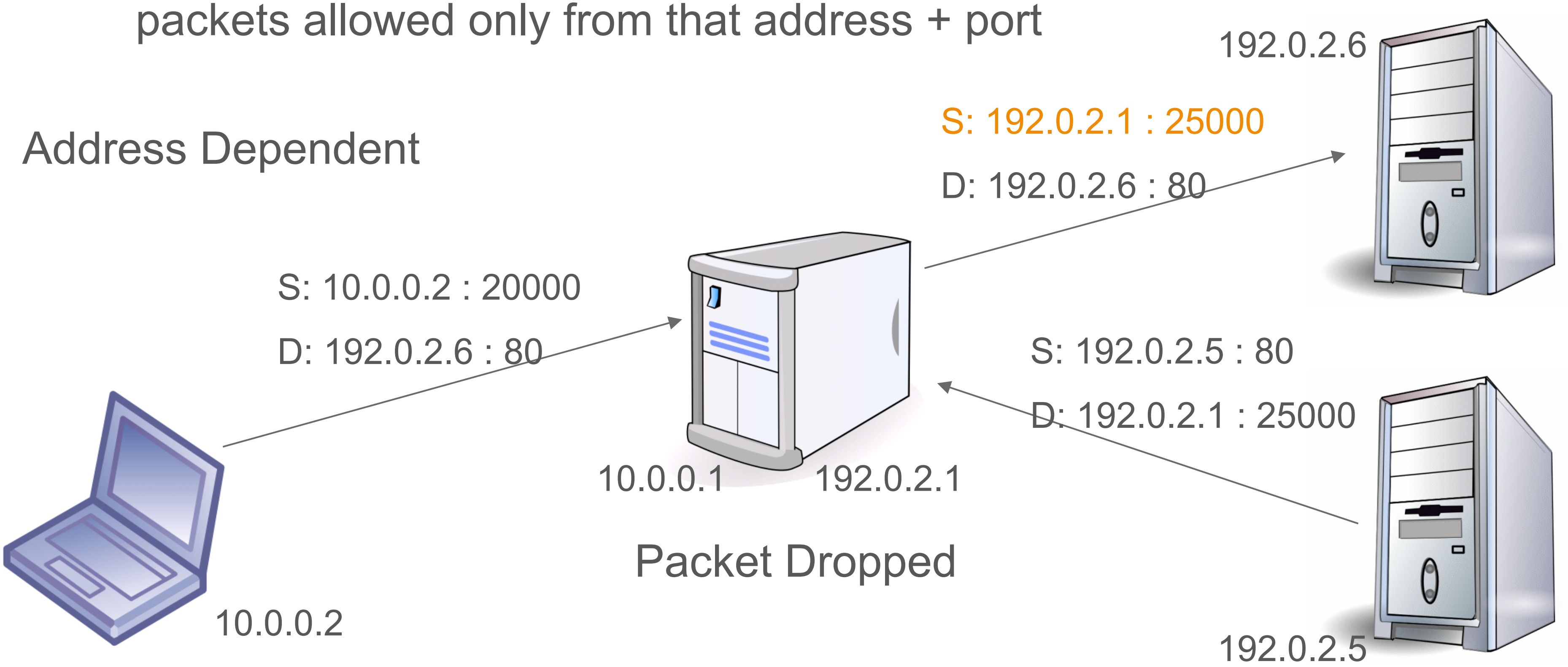
- Endpoint independent: any packets allowed back
- Address dependent: external hosts can return packets
- Address and port dependent
  - Packets sent to an address + port → incoming packets allowed only from that address + port

Endpoint Independent



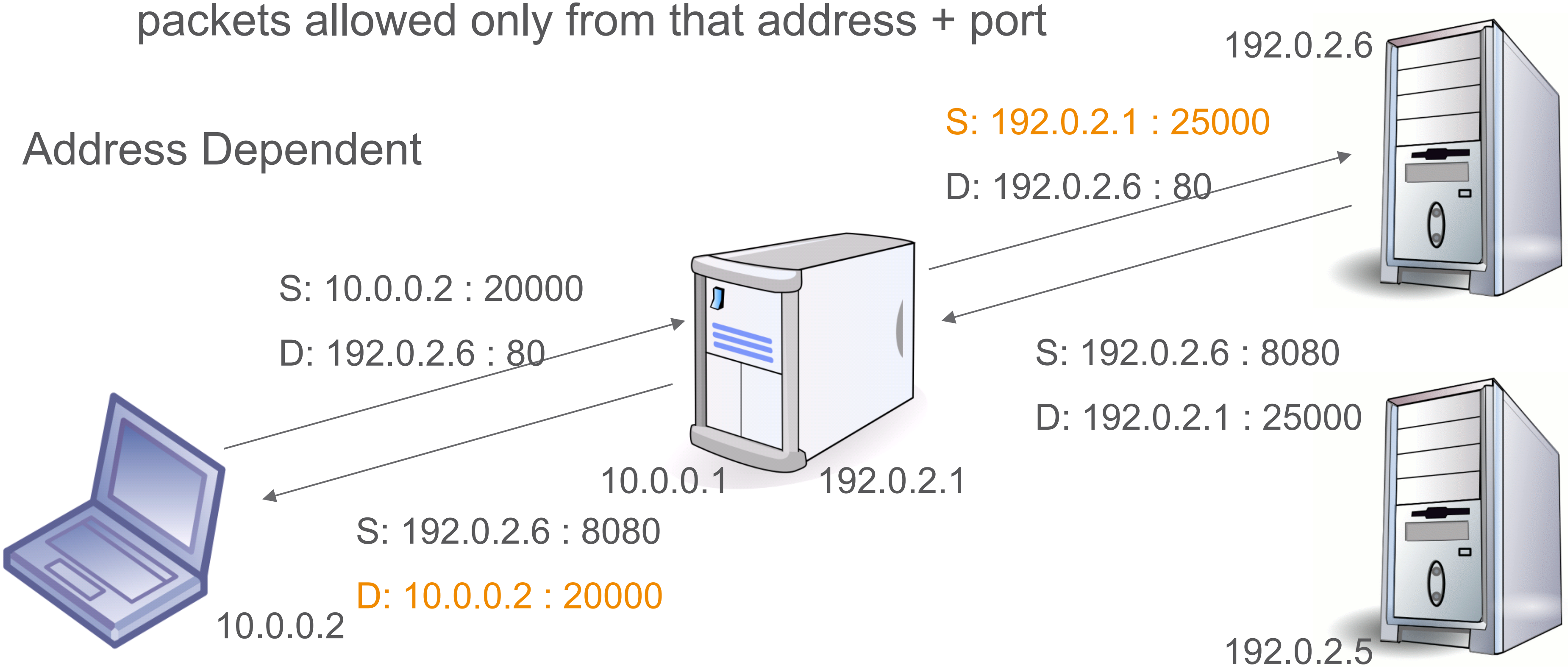
Filtering Behavior

- Endpoint independent: any packets allowed back
- Address dependent: external hosts can return packets
- Address and port dependent
  - Packets sent to an address + port → incoming packets allowed only from that address + port



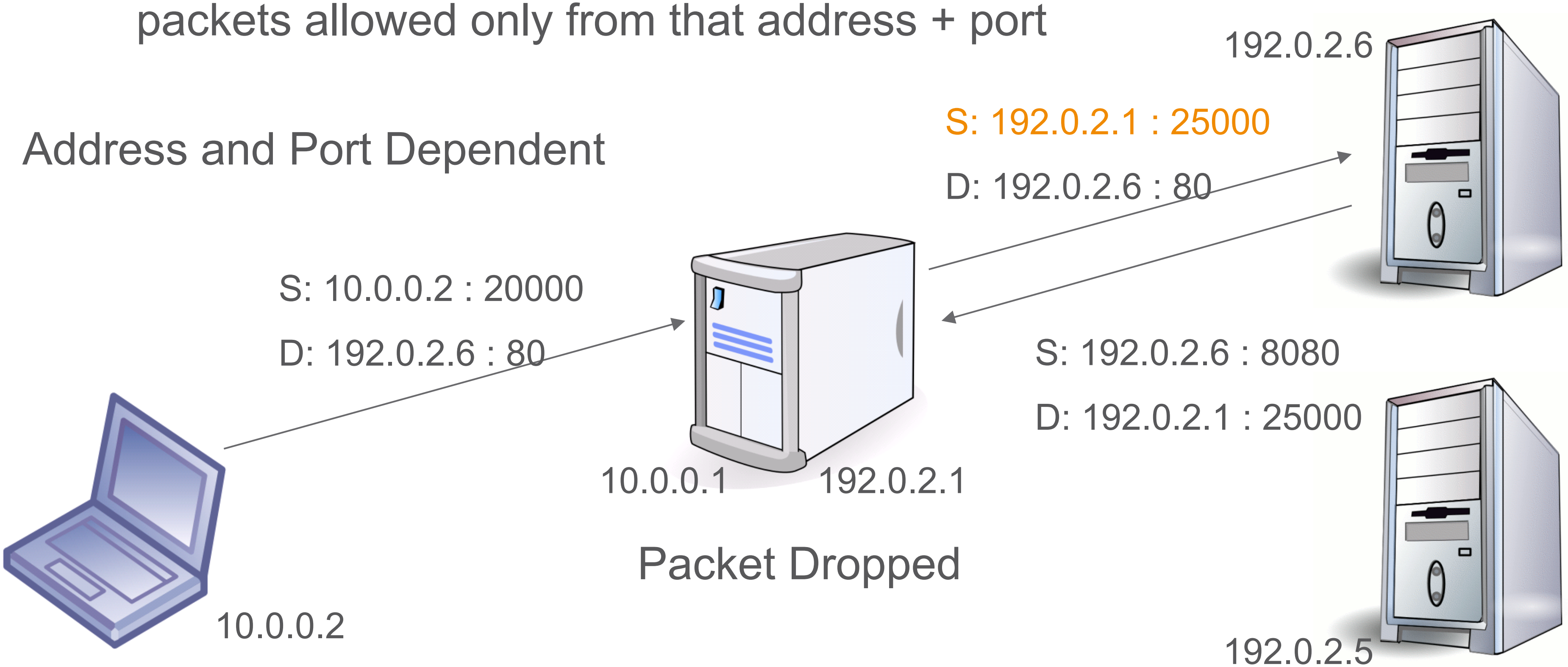
Filtering Behavior

- Endpoint independent: any packets allowed back
- Address dependent: external hosts can return packets
- Address and port dependent
  - Packets sent to an address + port → incoming packets allowed only from that address + port



Filtering Behavior

- Endpoint independent: any packets allowed back
- Address dependent: external hosts can return packets
- Address and port dependent
  - Packets sent to an address + port → incoming packets allowed only from that address + port

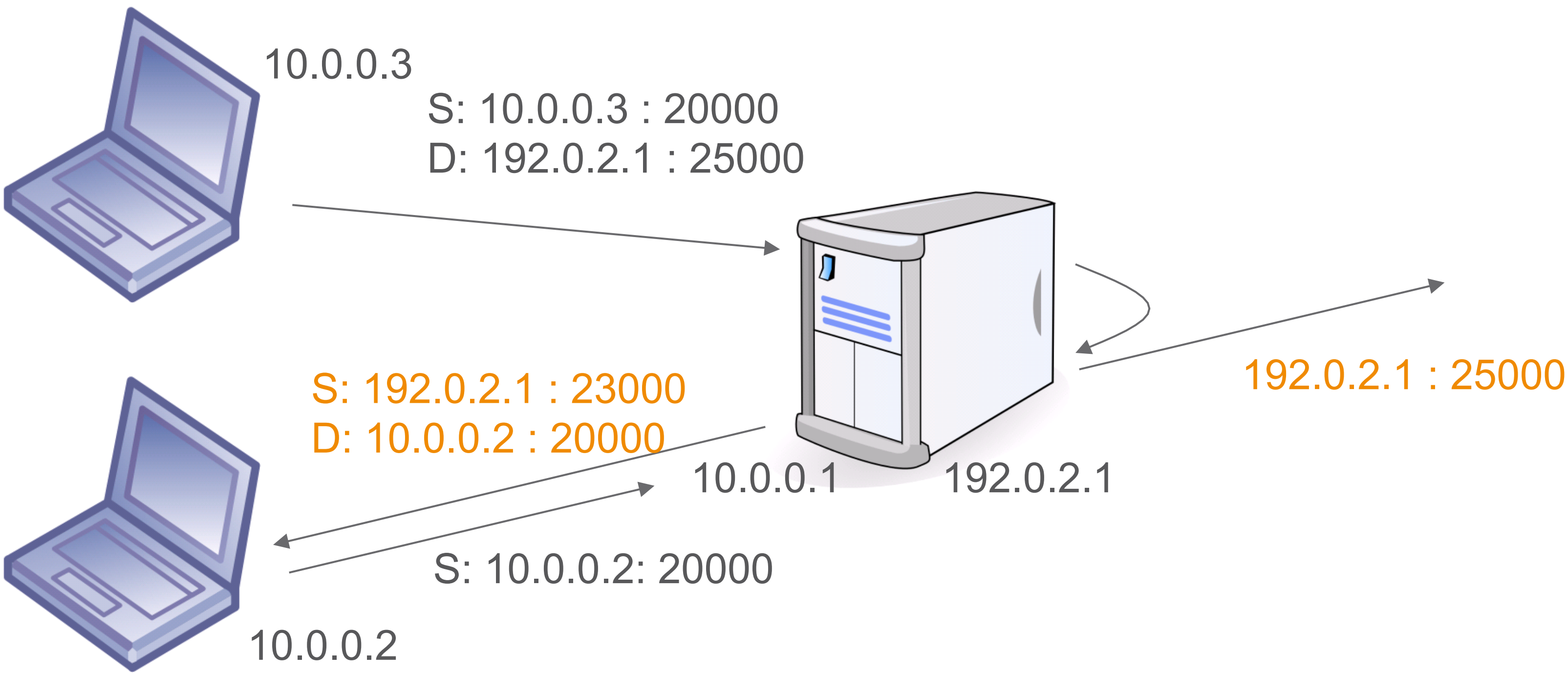


Filtering Behavior

- Endpoint independent filtering is RECOMMENDED
  - However, opens up ports for attackers
- If more strict filtering is required, address dependent filtering is RECOMMENDED
- Address and port dependent filtering complicates NAT traversal (more on this later)

Hairpinning

- Internal hosts communicate using external addresses
  - MUST be supported



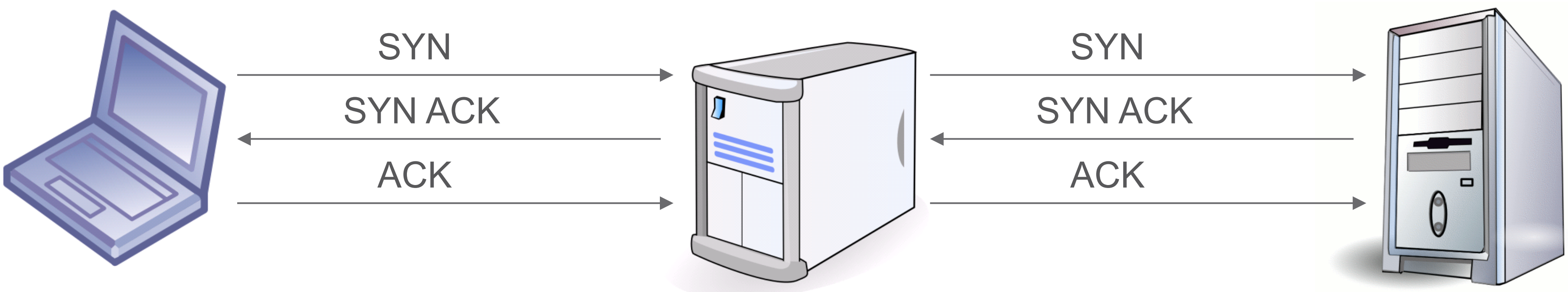
Outline

- Introduction to NATs
- NAT Behavior
  - UDP
  - TCP
- NAT Traversal
  - STUN
  - TURN
  - ICE
  - Others
- NAT64

TCP Connection Establishment

- Three-way handshake
  - MUST be supported
- Simultaneous open
  - MUST be supported

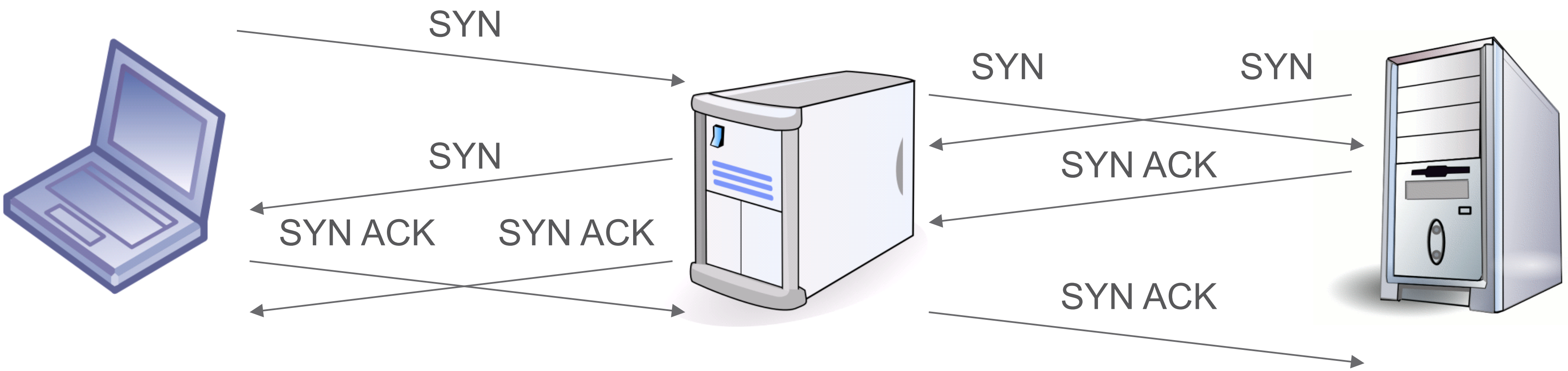
Three-way Handshake



TCP Connection Establishment

- Three-way handshake
  - MUST be supported
- Simultaneous open
  - MUST be supported

Simultaneous Open



NAT TCP Session Timeout

- Established connections
  - MUST NOT be less than 2 hours and 4 minutes
  - By default TCP keepalives are sent every 2 hours
- Partially opened or partially closed connections
  - MUST NOT be less than 4 minutes
- TIME_WAIT timeout not specified

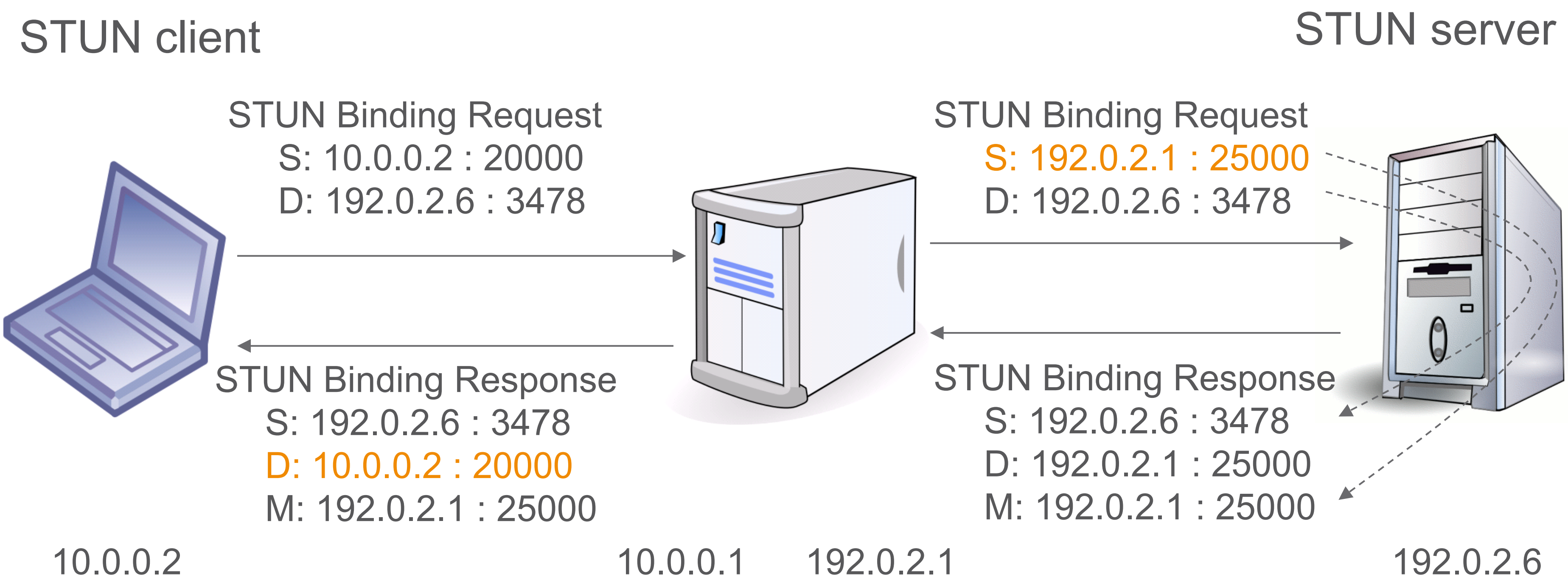
Outline

- Introduction to NATs
- NAT Behavior
  - UDP
  - TCP
- NAT Traversal
  - STUN
  - TURN
  - ICE
  - Others
- NAT64

STUN

- Session Traversal Utilities for NAT (RFC 5389)
  - Binding discovery using STUN servers
  - NAT keepalives
  - Authentication and integrity (short-term and long term credentials)
- Type-Length-Value (TLV) encoded, extensible protocol
- Can run on UDP, TCP, or TLS/TCP
- STUN server discovered using DNS SRV
- Transactions
  - Request/response
  - Indications (not delivered reliably)
- Can be multiplexed with other protocols
  - Two first bits are zeros (unlike with RTP)
  - Magic cookie
  - FIGERPRINT attribute

Binding Discovery



M: STUN (XOR-)MAPPED-ADDRES TLV

XOR-MAPPED-ADDRESS

- Some NATs inspect packets and translate IP addresses known to them
  - Try to be smart and “fix” the application layer protocol
- The mapped address is obfuscated in the response so that NAT does not recognize it
  - Simple XOR operation

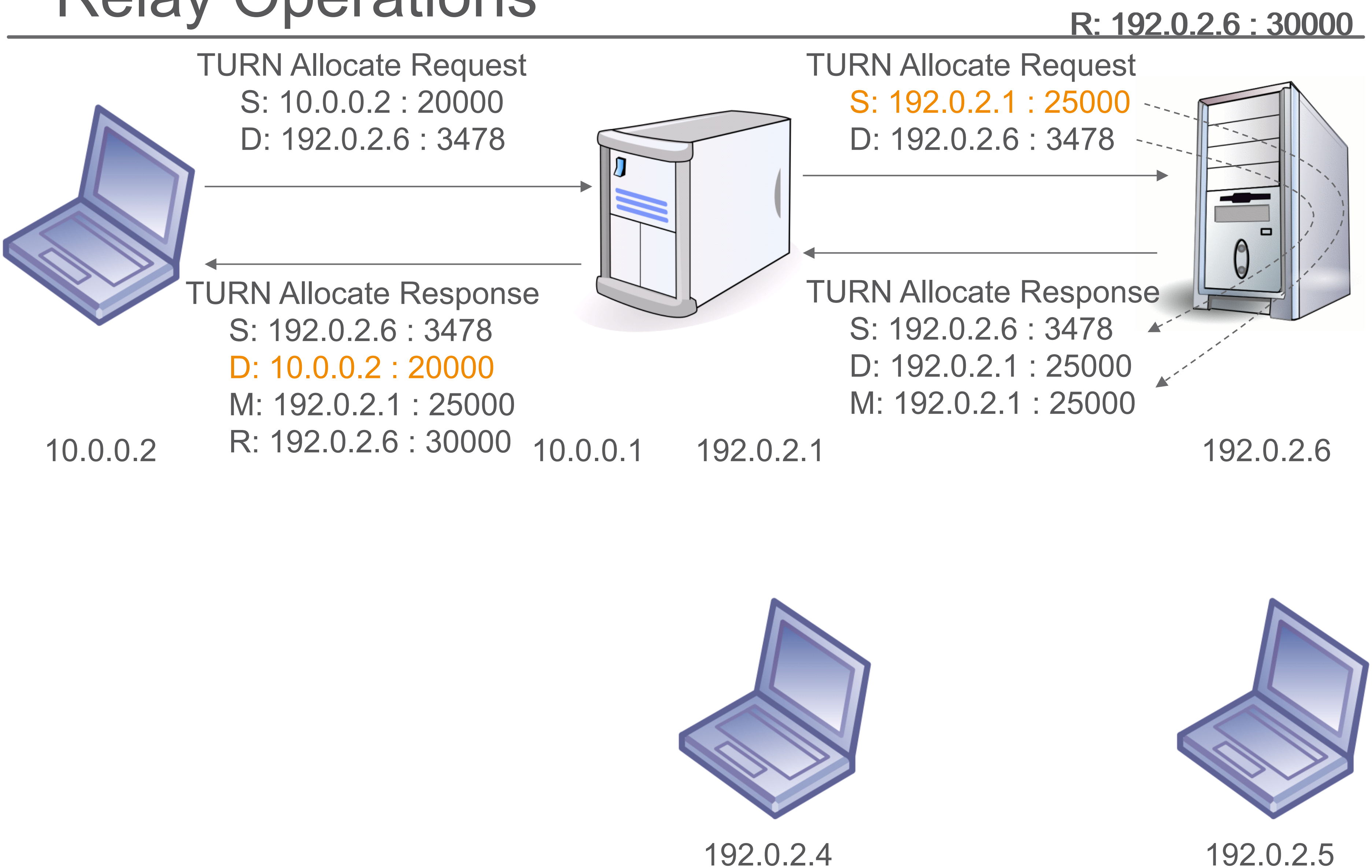
Outline

- Introduction to NATs
- NAT Behavior
  - UDP
  - TCP
- NAT Traversal
  - STUN
  - TURN
  - ICE
  - Others
- NAT64

TURN

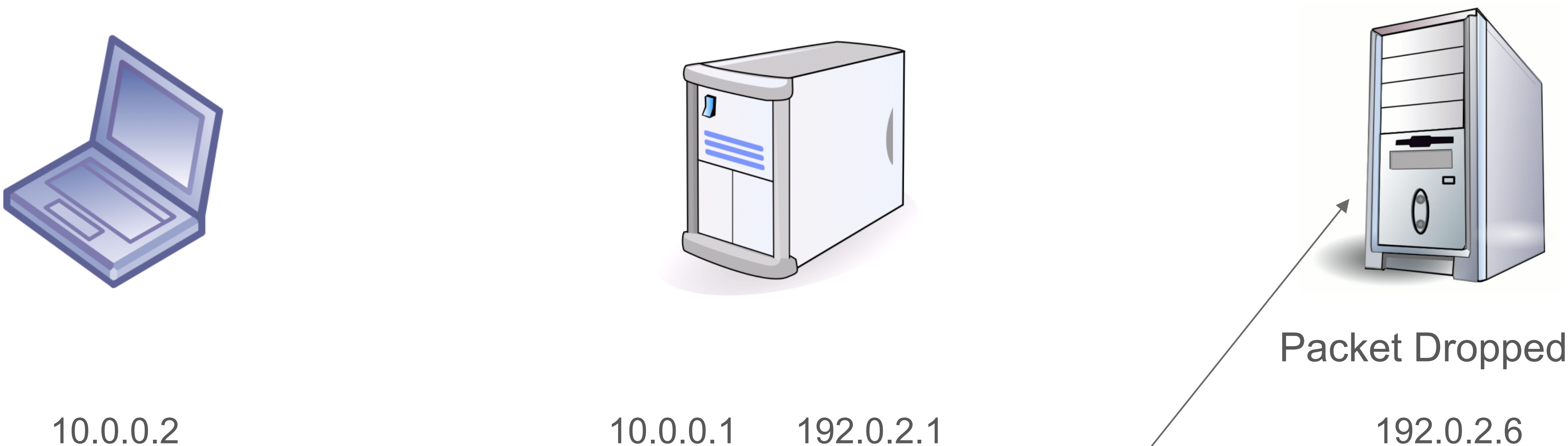
- Traversal Using Relays around NAT: Relay Extensions to Session Traversal Utilities for NAT (RFC 5766)
- Allocate request / response
  - Allocate an external “relayed” address at the relay
  - Responses carry the mapped and the relayed address
- Send and Data indication
  - STUN messages containing relayed data
  - Send data to a remote endpoint through the relay
  - Data received from remote endpoints through the relay
- Channels
  - Send and receive relayed data with minimalistic (32-bit) header
- Permissions

Relay Operations

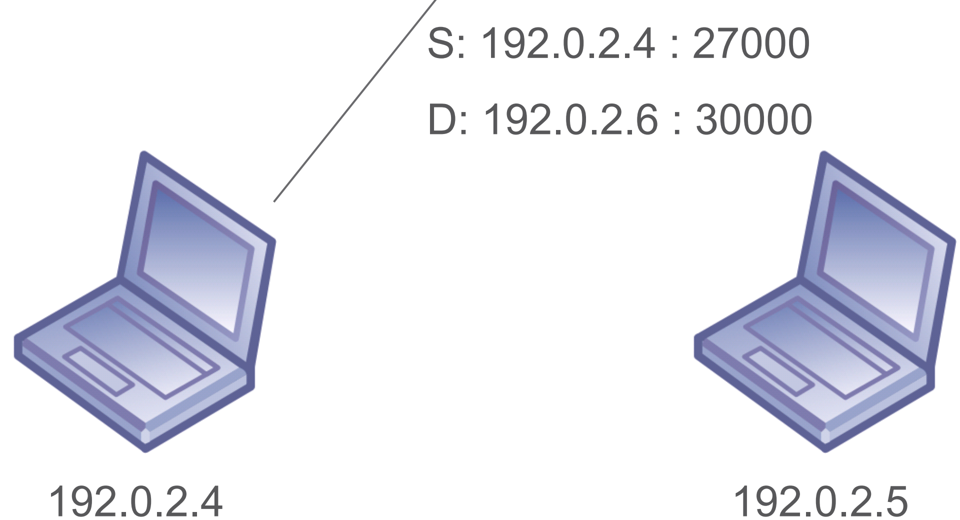


Relay Operations

R: 192.0.2.6 : 30000

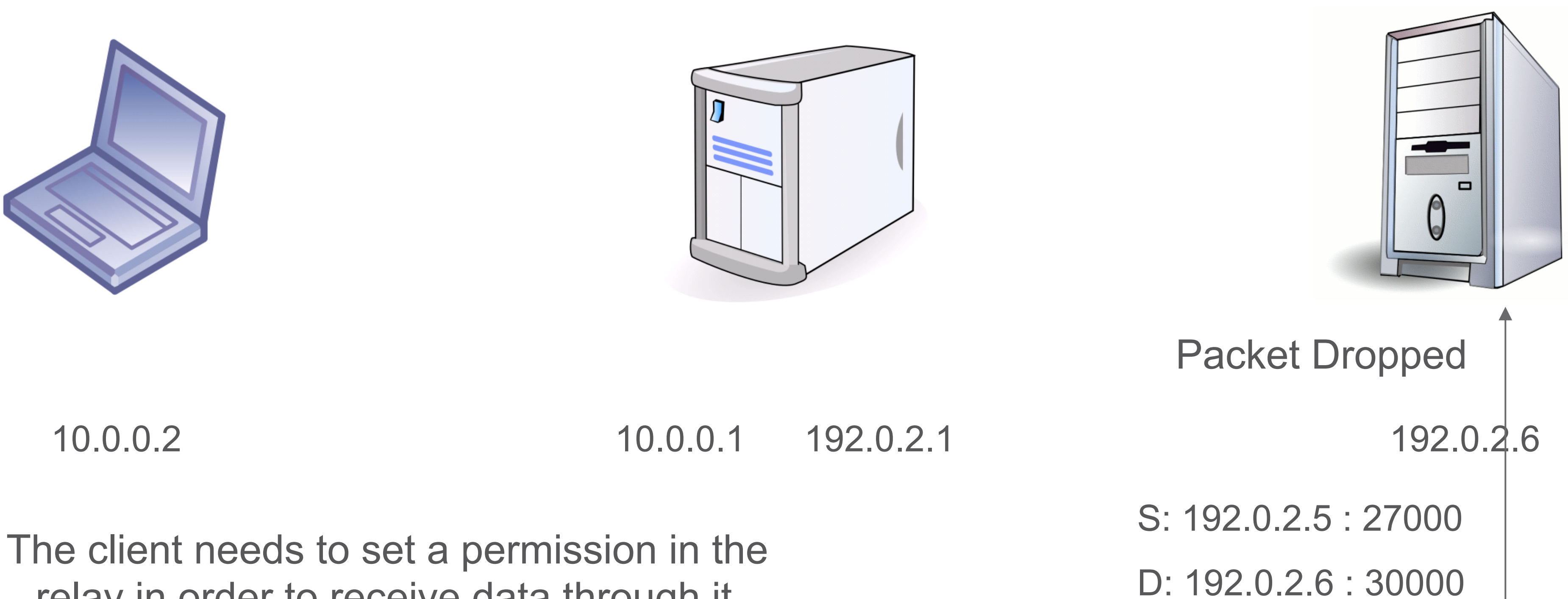


- The client needs to set a permission in the relay in order to receive data through it
- Equivalent to a NAT with:
  - Address dependent filtering policy
  - Endpoint independent mapping



Relay Operations

R: 192.0.2.6 : 30000



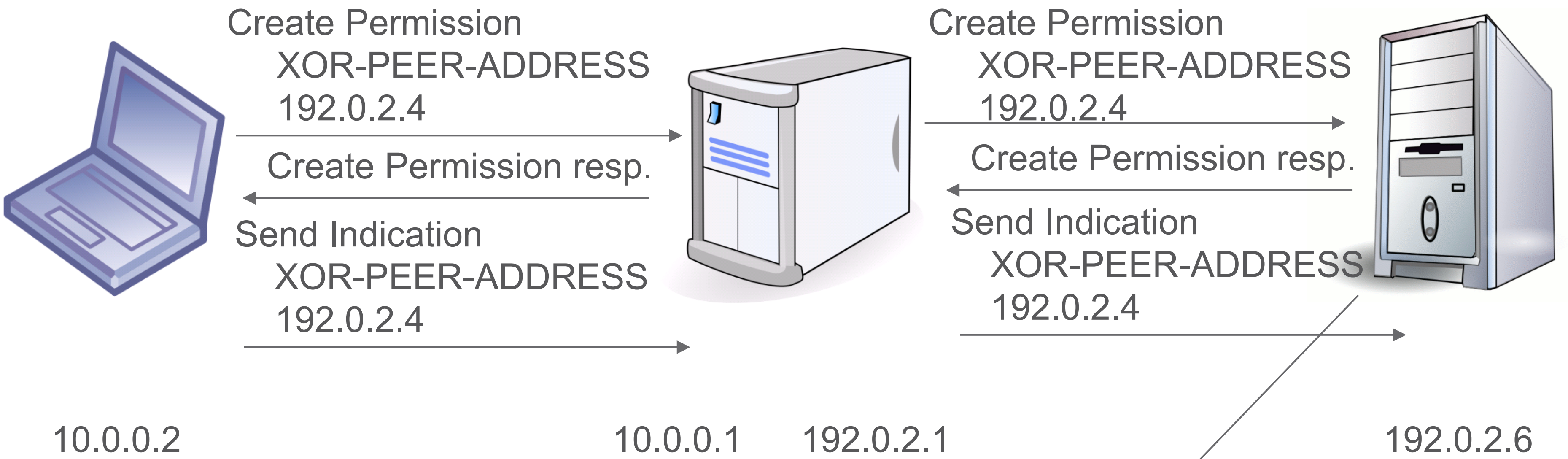
The client needs to set a permission in the relay in order to receive data through it

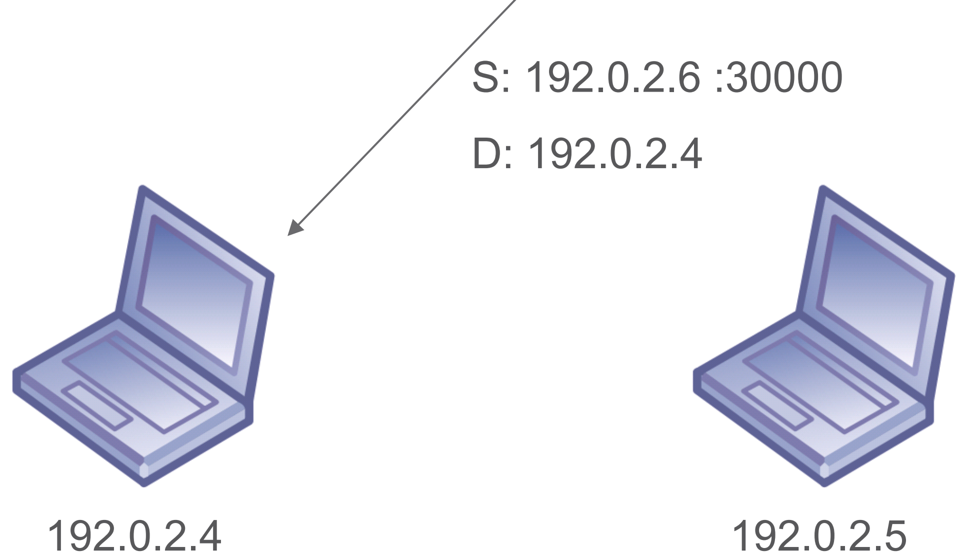
Equivalent to a NAT with:

- Address dependent filtering policy
- Endpoint independent mapping

Relay Operations

R: 192.0.2.6 : 30000



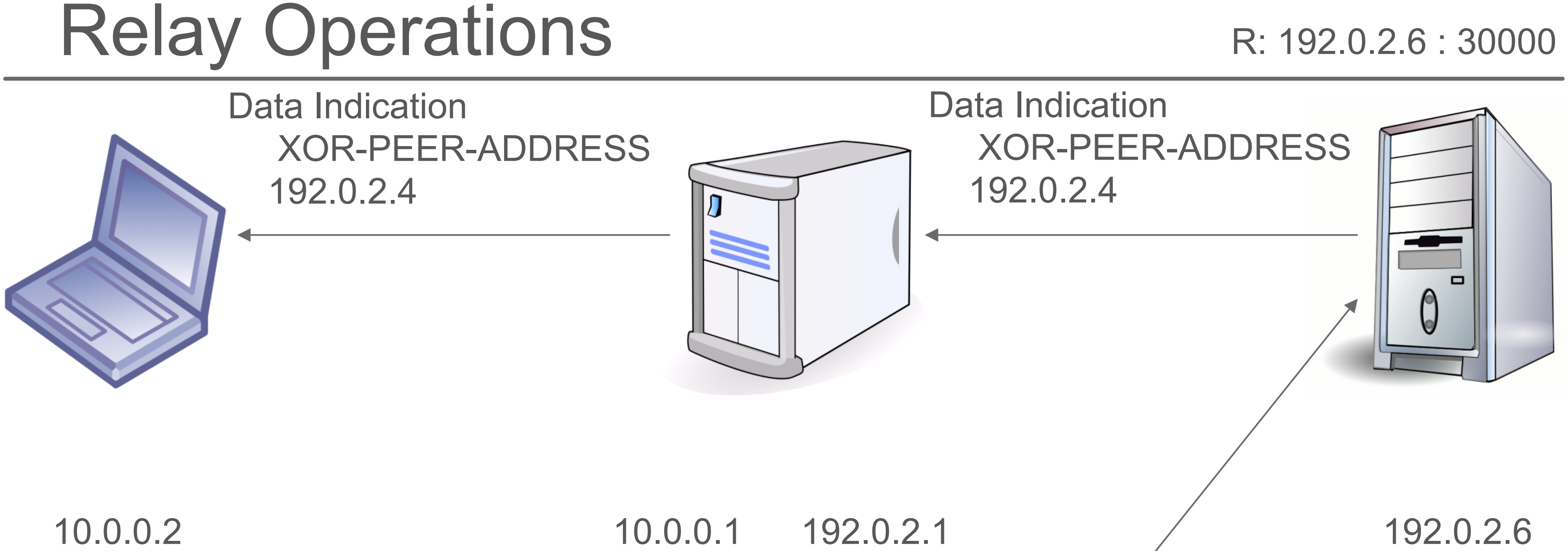


The client needs to set a permission in the relay in order to receive data through it

Equivalent to a NAT with:

- Address dependent filtering policy
- Endpoint independent mapping

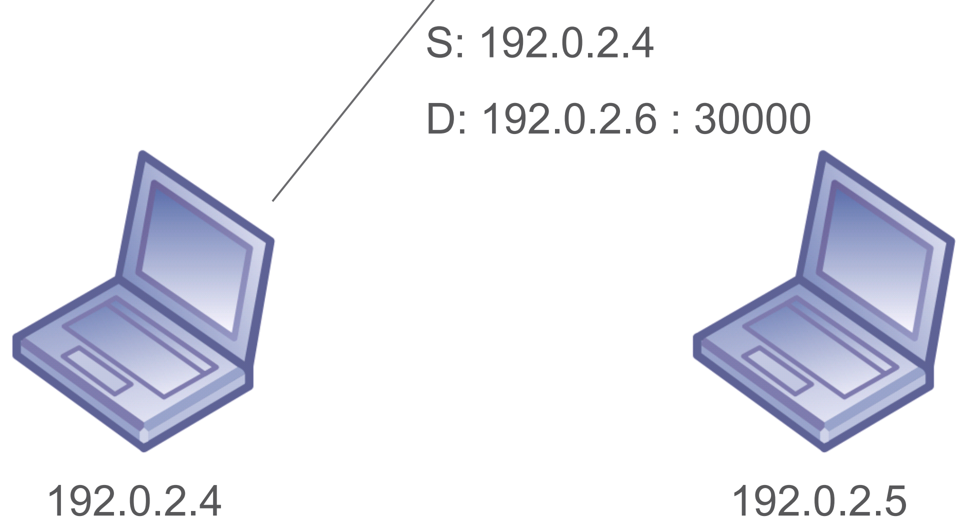
Relay Operations



The client needs to set a permission in the relay in order to receive data through it

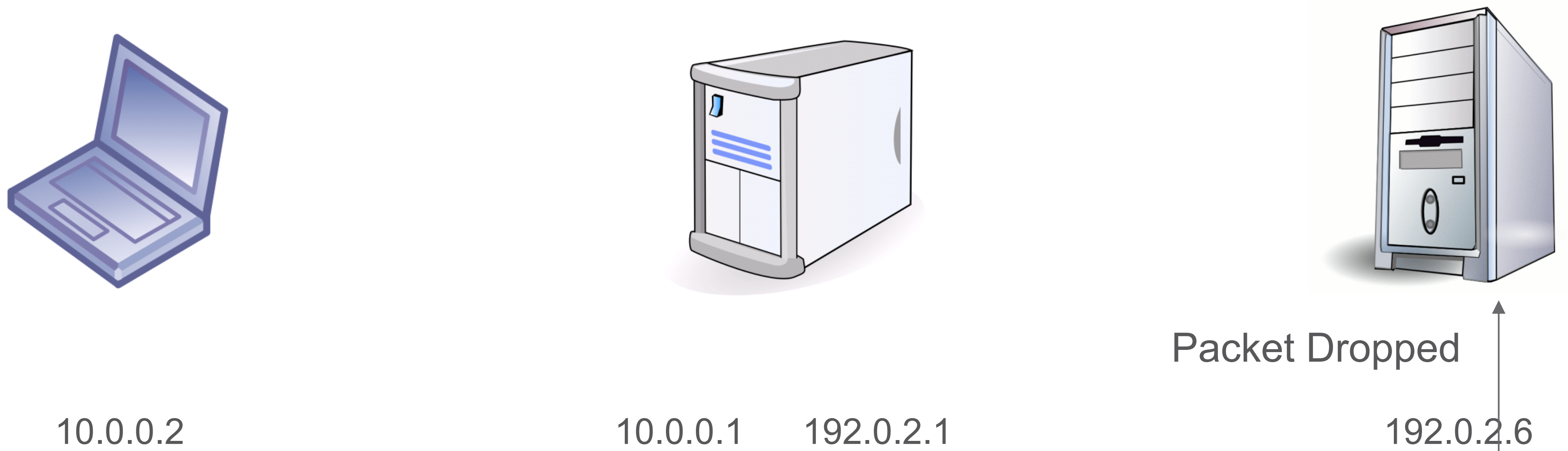
Equivalent to a NAT with:

- Address dependent filtering policy
- Endpoint independent mapping



Relay Operations

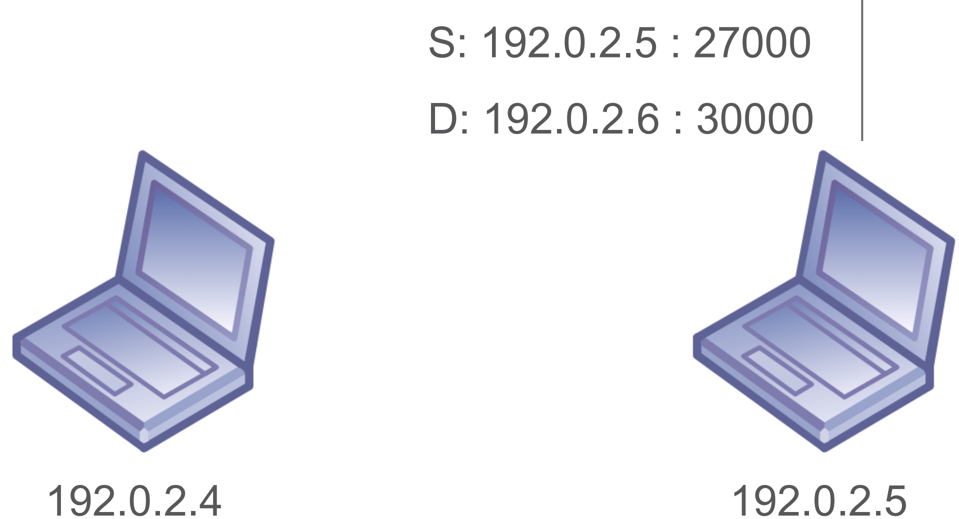
R: 192.0.2.6 : 30000



The client needs to set a permission in the relay in order to receive data through it

Equivalent to a NAT with:

- Address dependent filtering policy
- Endpoint independent mapping



Outline

- Introduction to NATs
- NAT Behavior
  - UDP
  - TCP
- NAT Traversal
  - STUN
  - TURN
  - ICE
  - Others
- NAT64

ICE

- Interactive Connectivity Establishment : A Protocol for Network Address Translator Traversal for Offer/Answer Protocols (RFC 5245)
- Uses and extends STUN and TURN protocols
- Overall procedure:
  - Endpoints gather all the addresses they can
    - Using e.g. STUN and/or TURN
  - Addresses (candidates) are exchanged with the peer
  - Connectivity checks are run between the candidates
  - The highest priority candidate pair that works is selected for use
- Session Initiation Protocol (SIP) for signaling
  - But other signaling protocols can be used too

Gathering Addresses

- Address types
  - Host candidates
  - Server-reflexive candidates
  - Relayed candidates
  - Peer-reflexive candidates
- Duplicated addresses are removed
- Foundation: used to freeze addresses (related to connectivity checks)
  - Same type
  - Bases with the same IP address
  - Same STUN server

Prioritizing Addresses

Priority = 2^{24} (type preference) +
2^{8} (local preference) +
2 (256 – component ID)

- Type preference [0-126]: preference for the type of candidate (e.g., server reflexive)
- Local preference [0-65535]: preference for the interface the candidate was obtained from (e.g., multihomed hosts)
- Component ID [1-256]: for media with multiple components (e.g., RTP and RTCP)

Connectivity Checks

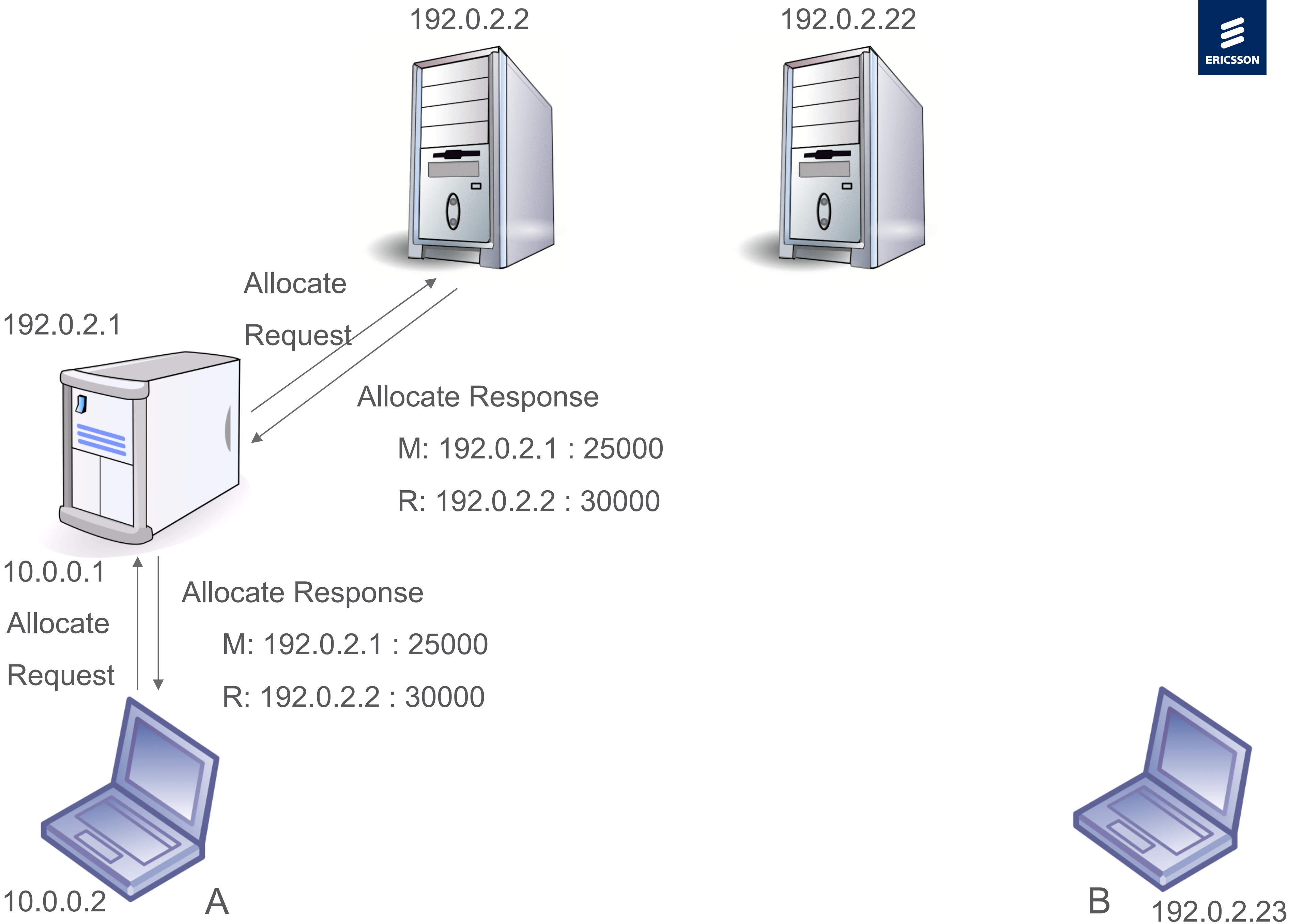
- Five states for a pair:
  - Waiting, in progress, succeeded, failed, frozen
- Periodic checks and triggered checks
  - Periodic checks performed in priority order
  - Incoming check may cause a triggered check
- Connectivity is checked with STUN Binding Requests
  - Using user names and passwords exchanged in the signaling channel (short term credentials)

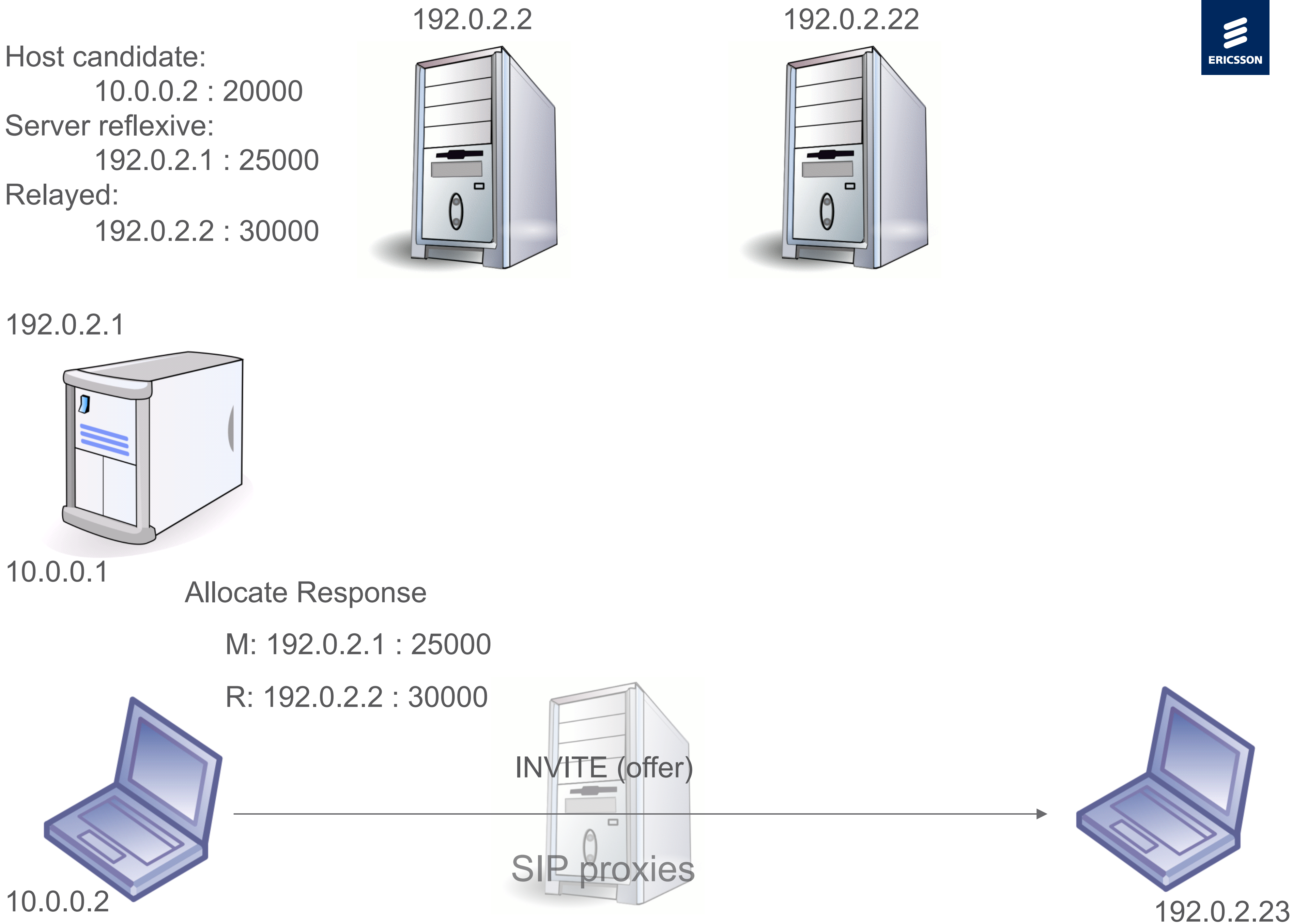
ICE Roles

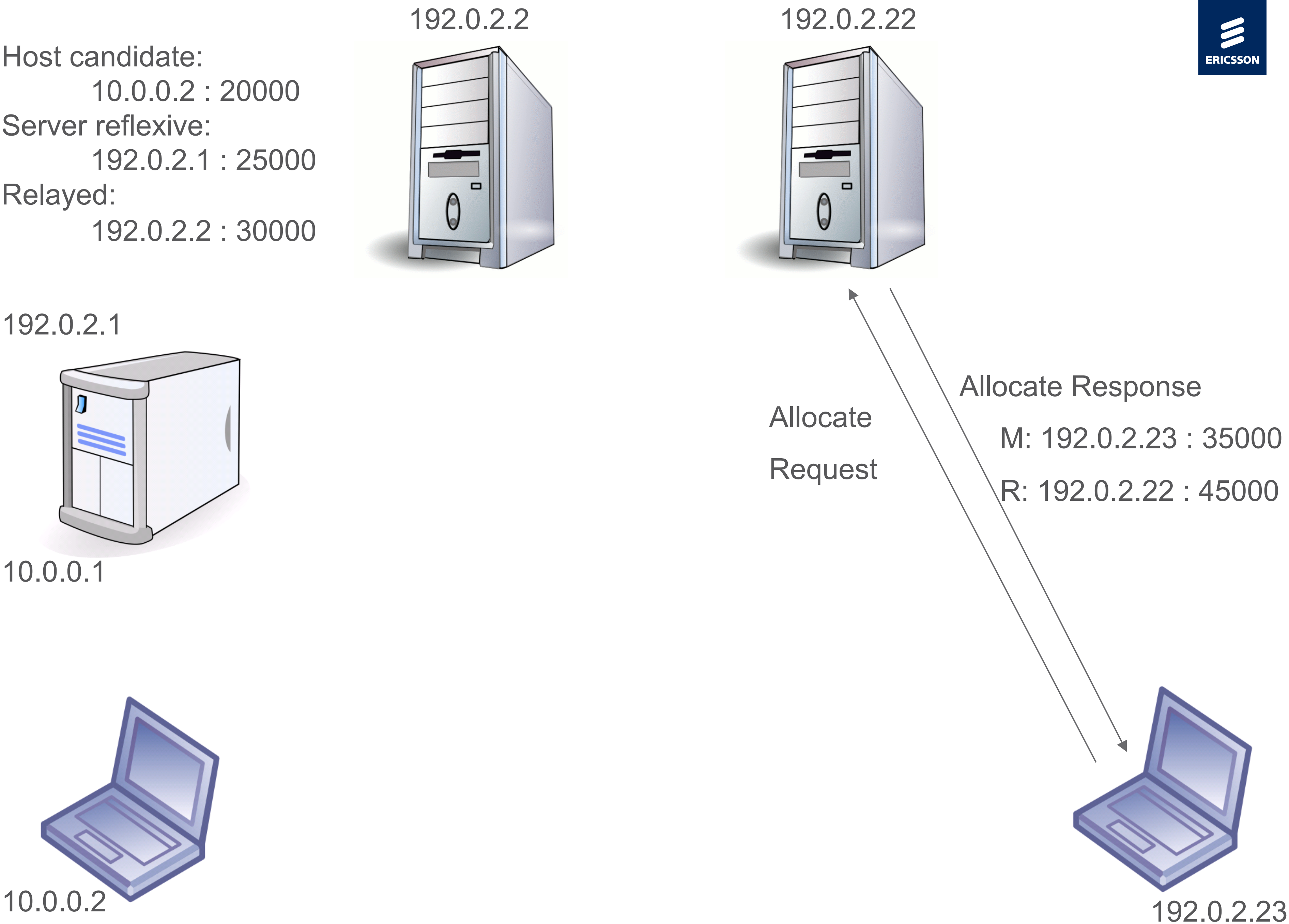
- Controlling agent
  - Agent that generates the initial offer
  - Selects which pair to eventually use
    - Implementation specific stopping criteria
    - USE-CANDIDATE attribute
- Controlled agent
  - Generates checks and responds to them like the controlling agent
  - Waits for the controlling agent to decide which candidate to use
- ICE lite agents
  - Know they are not behind a NAT
    - e.g., PSTN gateways, conferencing servers
  - Always in controlled role
  - Just respond to checks

ICE Example (1)

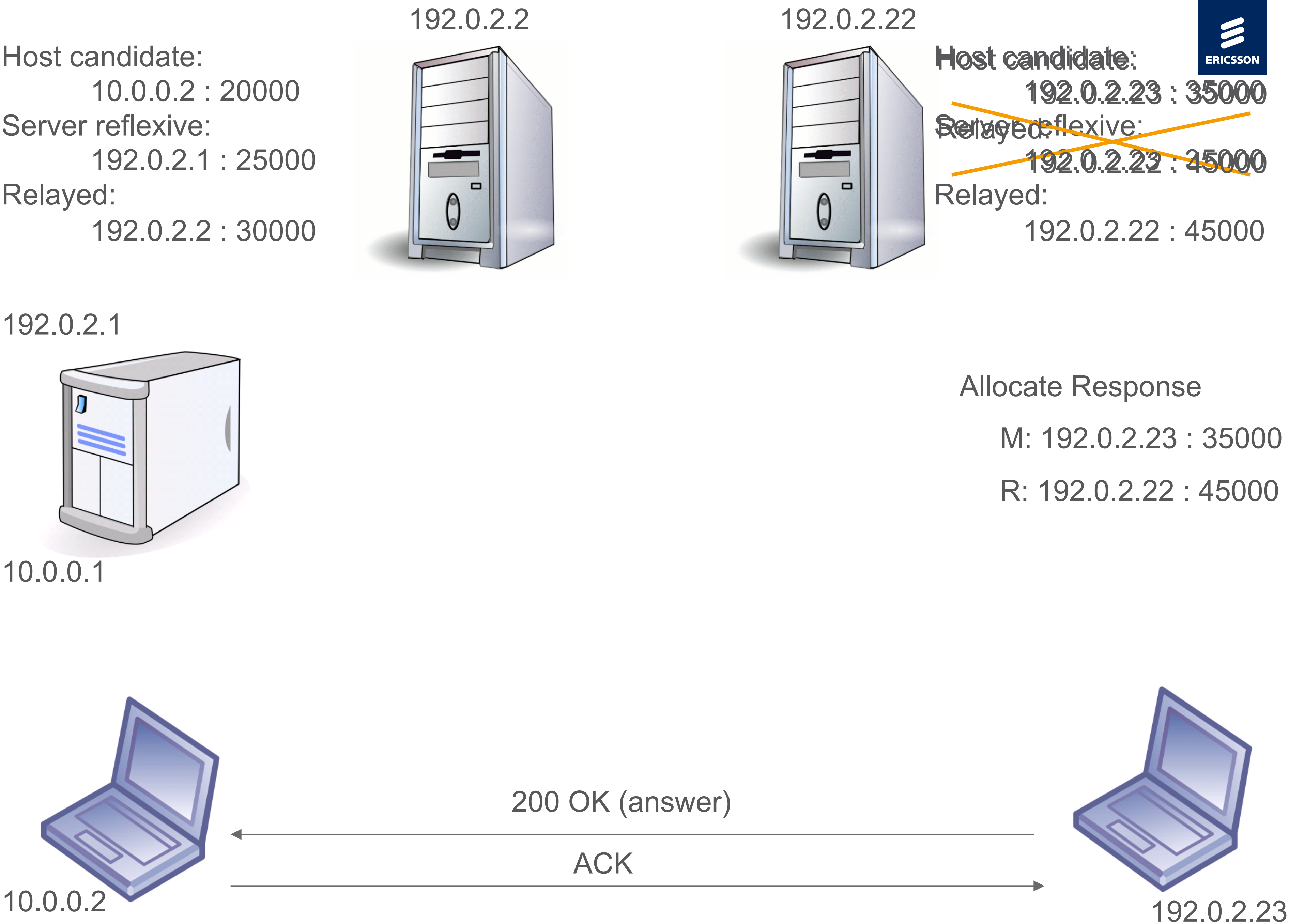
- One endpoint is behind a NAT
- Other endpoint has a public IP address
- Both endpoints use TURN servers
- SIP used for signaling between endpoints


192.0.2.2
192.0.2.22
Allocate
Request
192.0.2.1
Allocate Response
M: 192.0.2.1 : 25000
R: 192.0.2.2 : 30000
10.0.0.1
Allocate
Request
Allocate Response
M: 192.0.2.1 : 25000
R: 192.0.2.2 : 30000
10.0.0.2
A
B
192.0.2.23

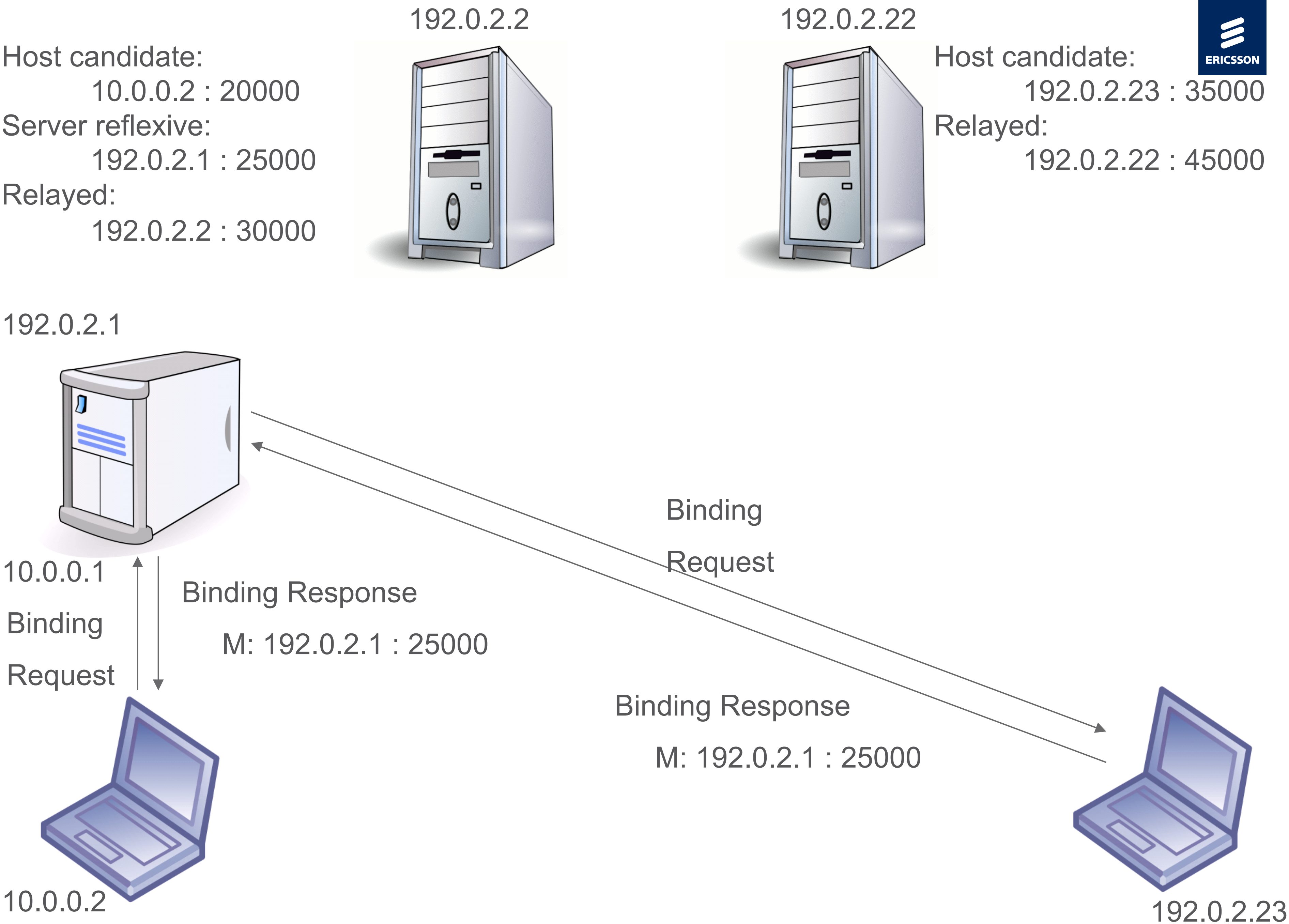

Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
192.0.2.2
192.0.2.22
192.0.2.1
10.0.0.1
Allocate Response
M: 192.0.2.1 : 25000
R: 192.0.2.2 : 30000
INVITE (offer)
SIP proxies
10.0.0.2
192.0.2.23

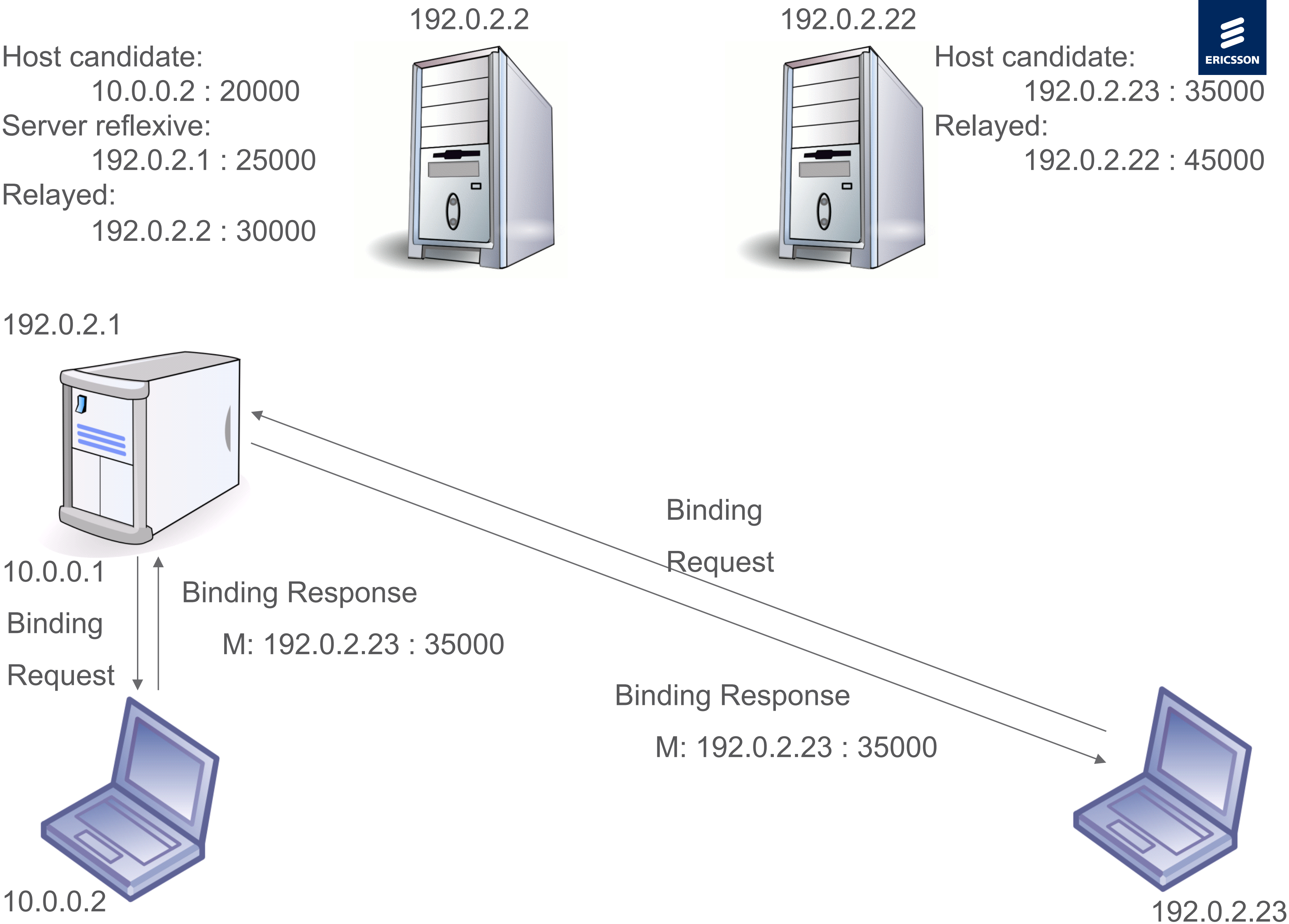


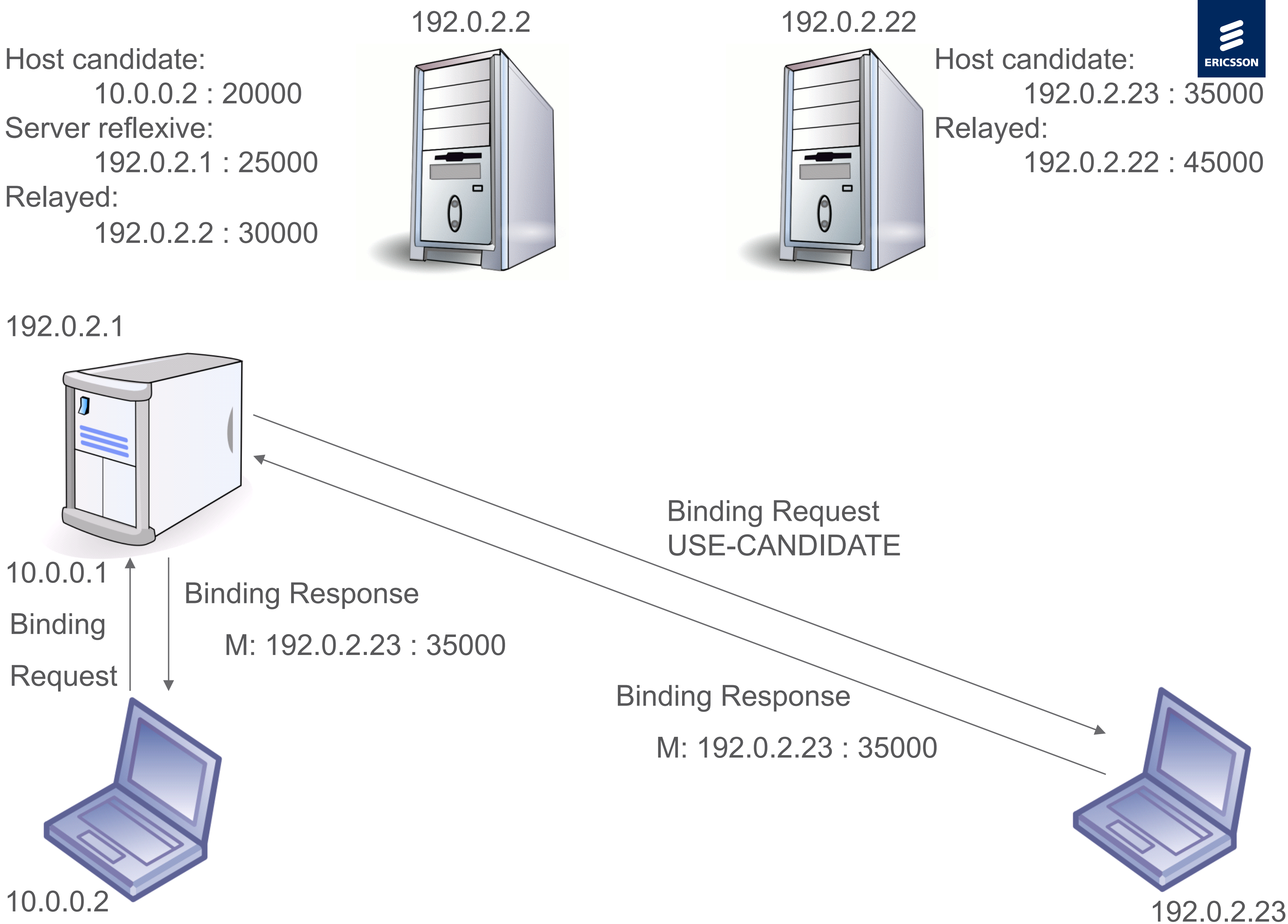
Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
192.0.2.2
192.0.2.22
192.0.2.1
10.0.0.1
Allocate
Request
Allocate Response
M: 192.0.2.23 : 35000
R: 192.0.2.22 : 45000
10.0.0.2
192.0.2.23

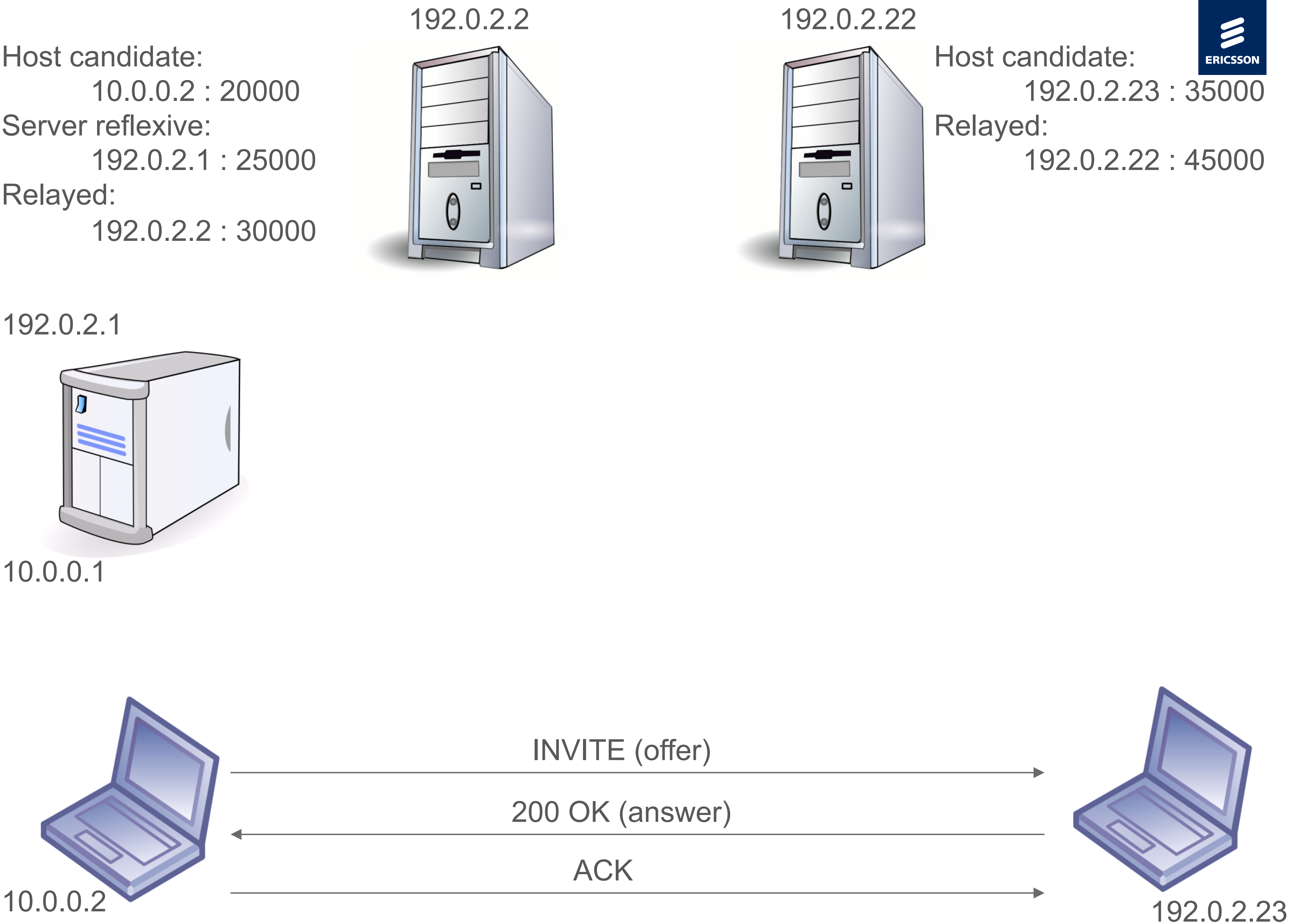


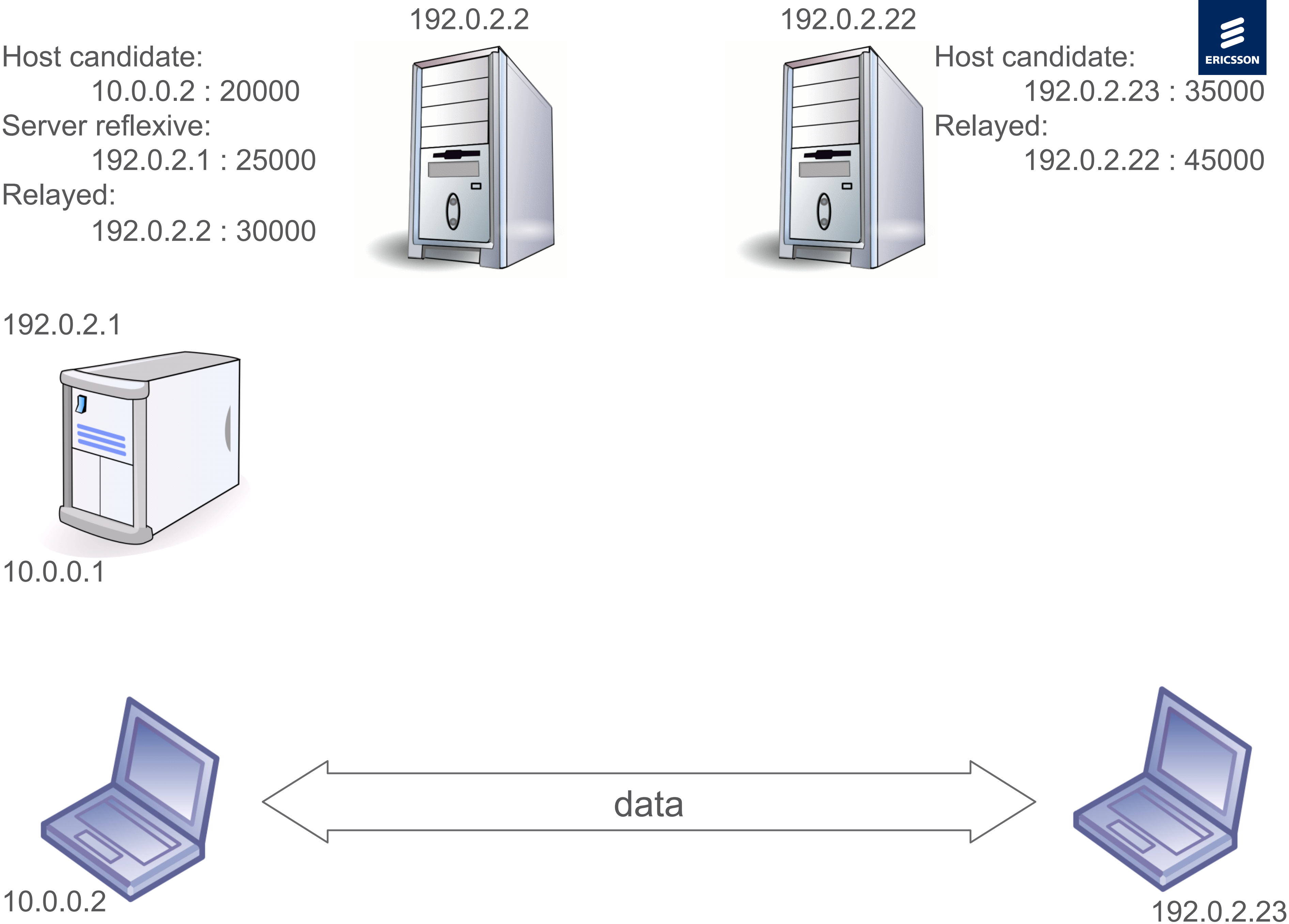
192.0.2.2
192.0.2.22
Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
Host candidate:
192.0.2.23 : 35000
Server reflexive:
192.0.2.23 : 35000
Relayed:
192.0.2.22 : 45000
192.0.2.1
10.0.0.1
Allocate Response
M: 192.0.2.23 : 35000
R: 192.0.2.22 : 45000
200 OK (answer)
ACK
10.0.0.2
192.0.2.23


Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
192.0.2.2
192.0.2.22
Host candidate:
192.0.2.23 : 35000
Relayed:
192.0.2.22 : 45000
192.0.2.1
10.0.0.1
Binding
Request
Binding Response
M: 192.0.2.1 : 25000
Binding
Request
Binding Response
M: 192.0.2.1 : 25000
10.0.0.2
192.0.2.23


192.0.2.2
192.0.2.22
Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
Host candidate:
192.0.2.23 : 35000
Relayed:
192.0.2.22 : 45000
192.0.2.1
10.0.0.1
Binding
Request
Binding Response
M: 192.0.2.23 : 35000
Binding
Request
Binding Response
M: 192.0.2.23 : 35000
10.0.0.2
192.0.2.23


192.0.2.2
192.0.2.22
Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
Host candidate:
192.0.2.23 : 35000
Relayed:
192.0.2.22 : 45000
192.0.2.1
10.0.0.1
Binding
Request
Binding Response
M: 192.0.2.23 : 35000
Binding Request
USE-CANDIDATE
Binding Response
M: 192.0.2.23 : 35000
10.0.0.2
192.0.2.23

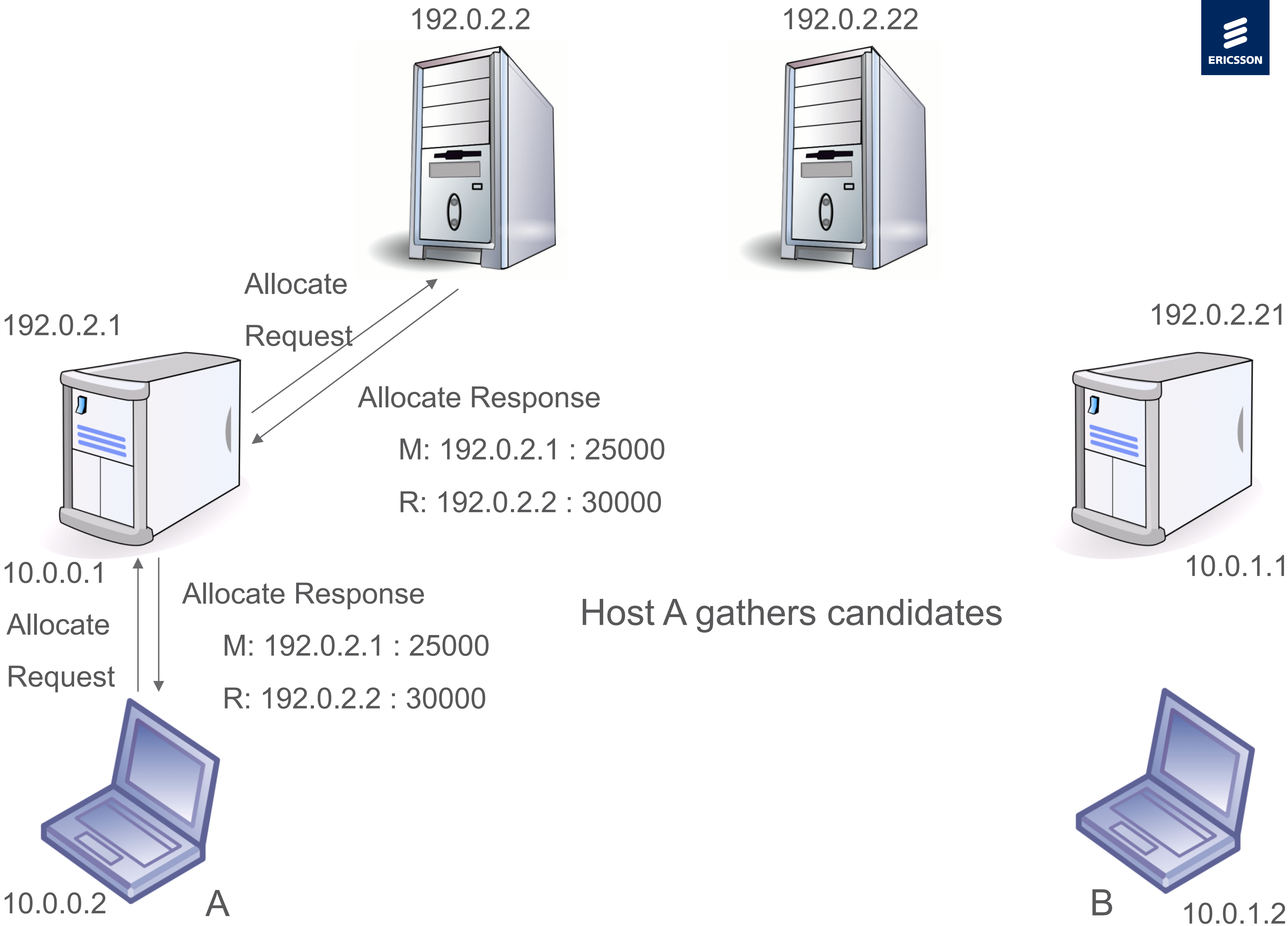

192.0.2.2
192.0.2.22
Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
Host candidate:
192.0.2.23 : 35000
Relayed:
192.0.2.22 : 45000
192.0.2.1
10.0.0.1
INVITE (offer)
200 OK (answer)
ACK
10.0.0.2
192.0.2.23

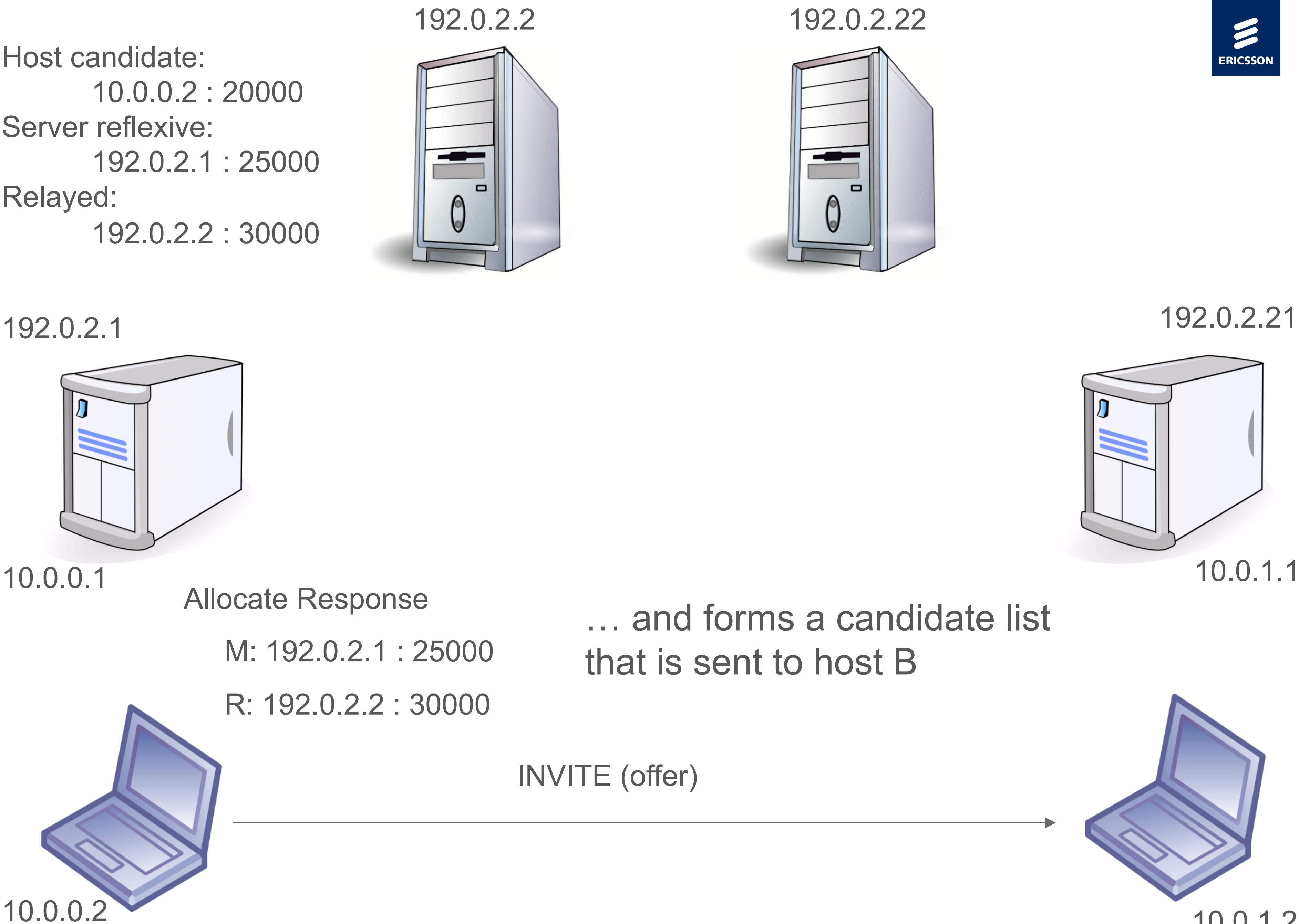

192.0.2.2
192.0.2.22
Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
Host candidate:
192.0.2.23 : 35000
Relayed:
192.0.2.22 : 45000
192.0.2.1
10.0.0.1
data
10.0.0.2
192.0.2.23

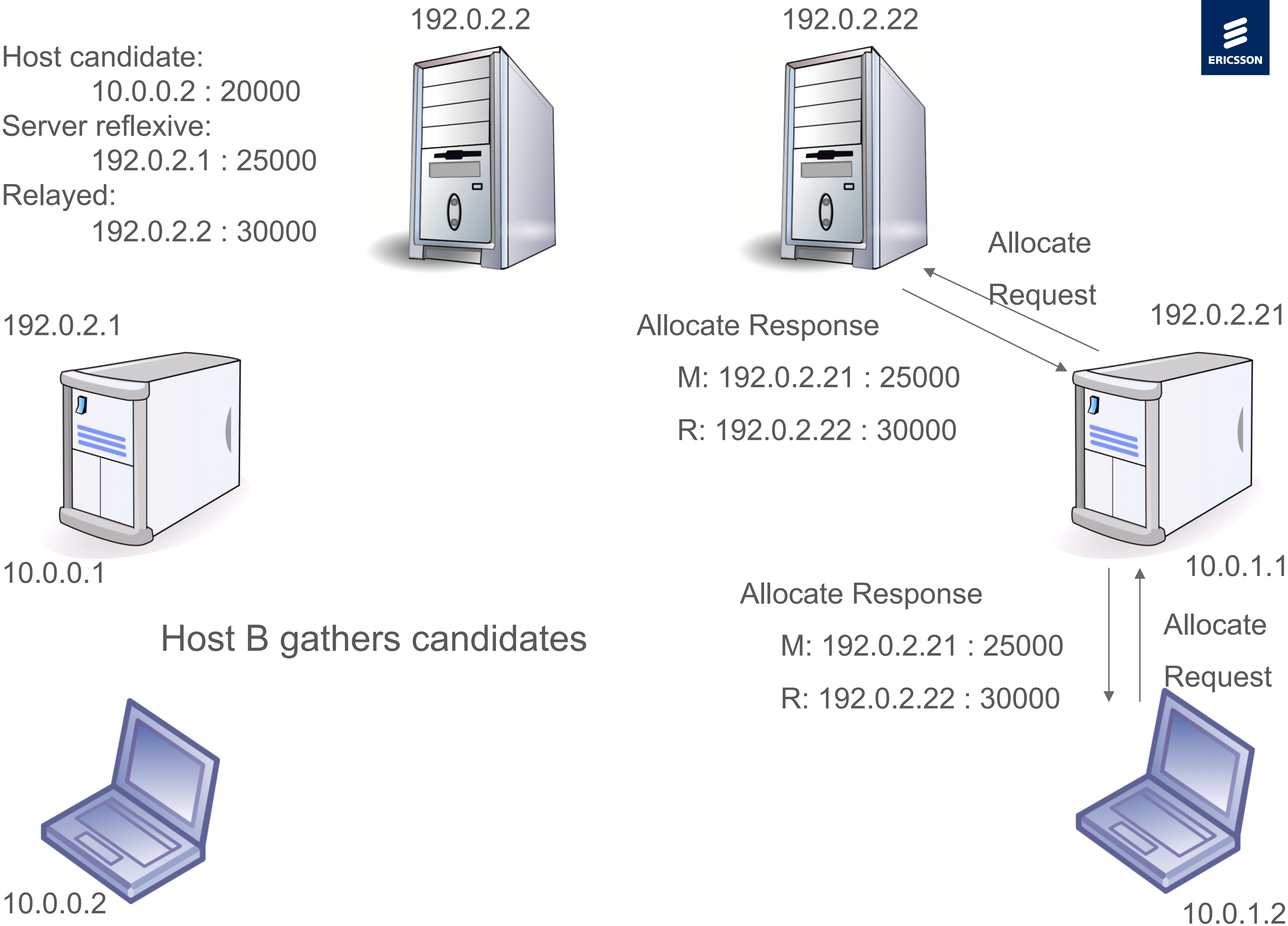
ICE Example (2)

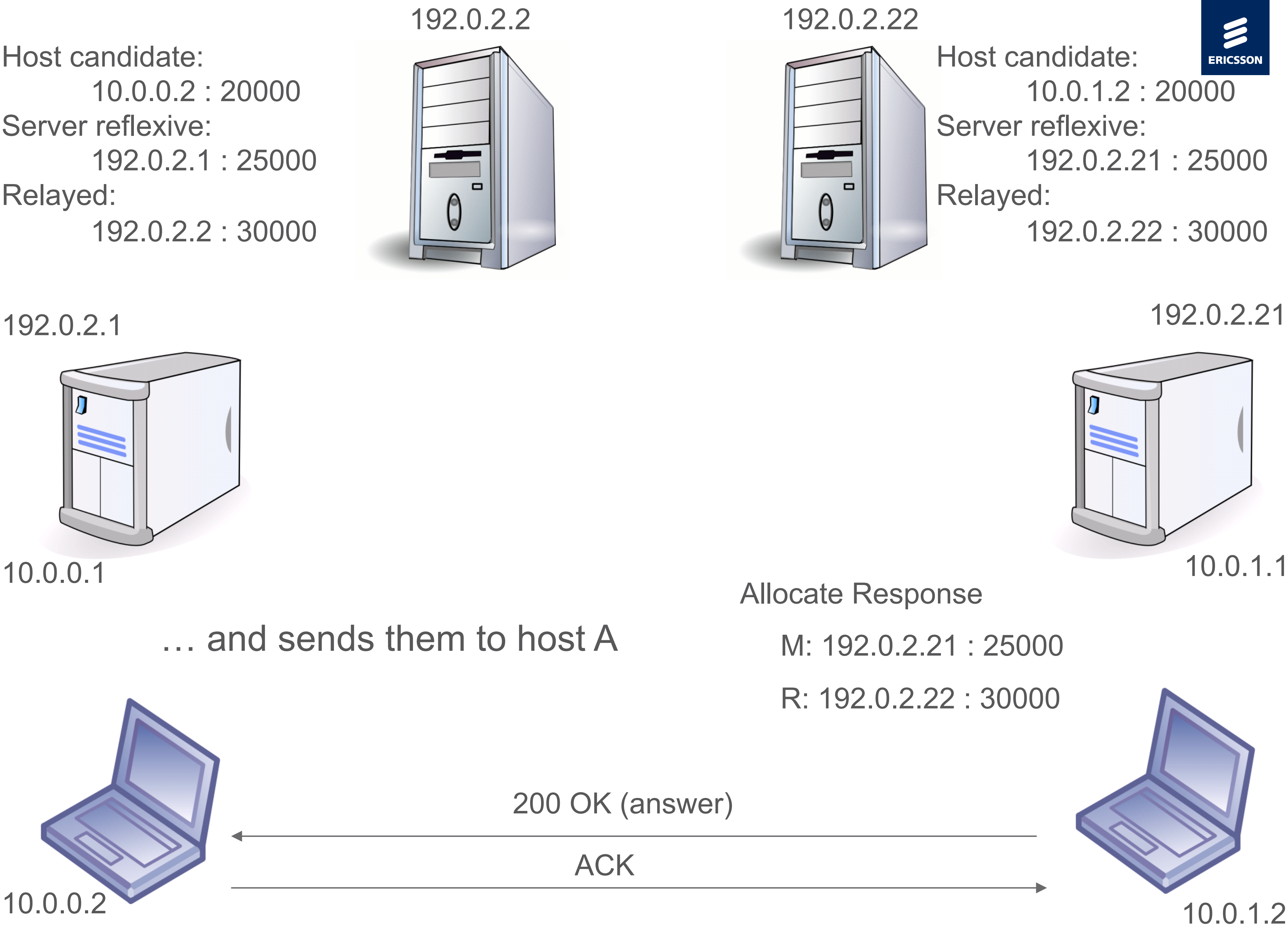
- Both endpoint are behind NATs
- Endpoints use TURN servers

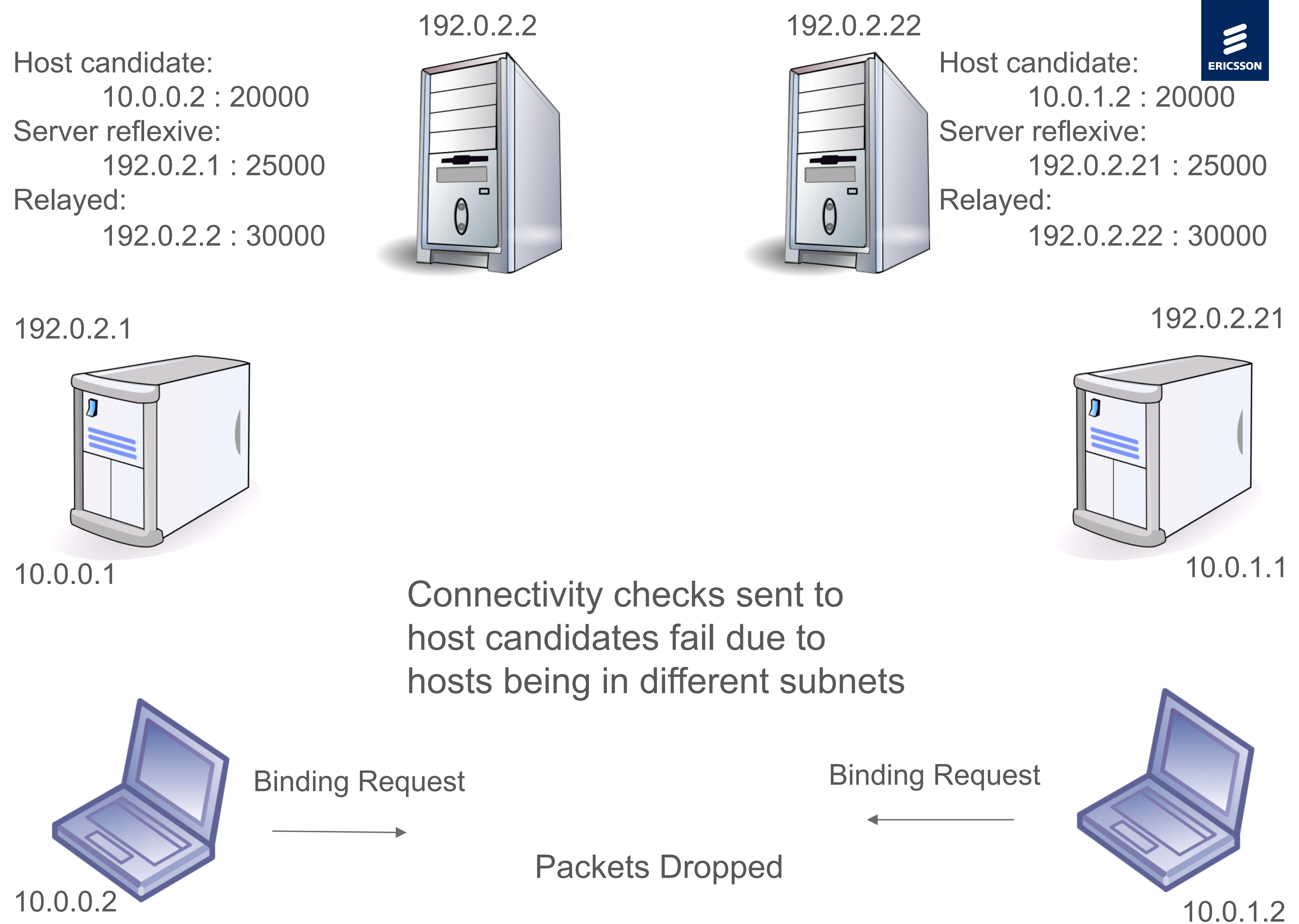
Host A gathers candidates

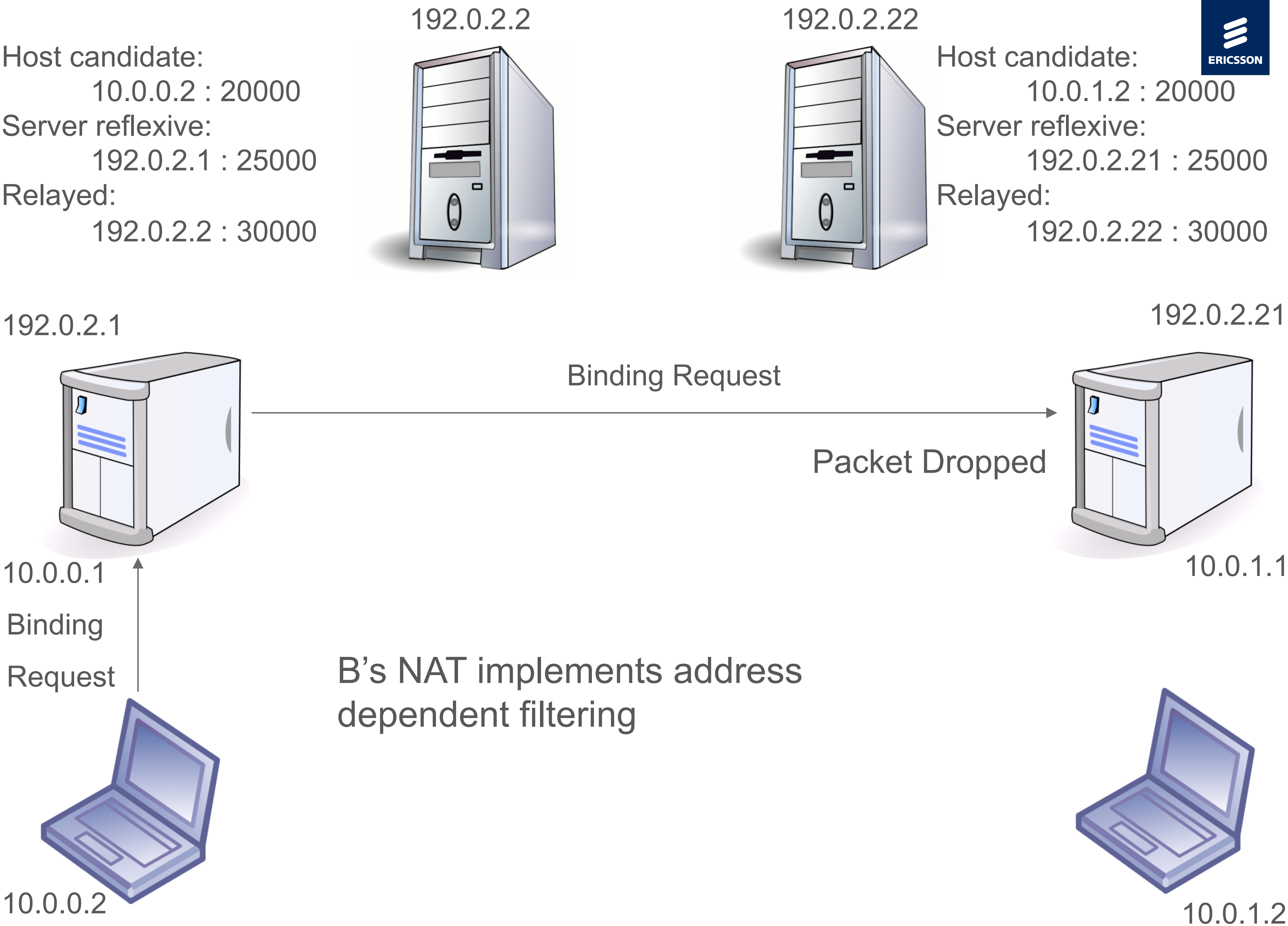


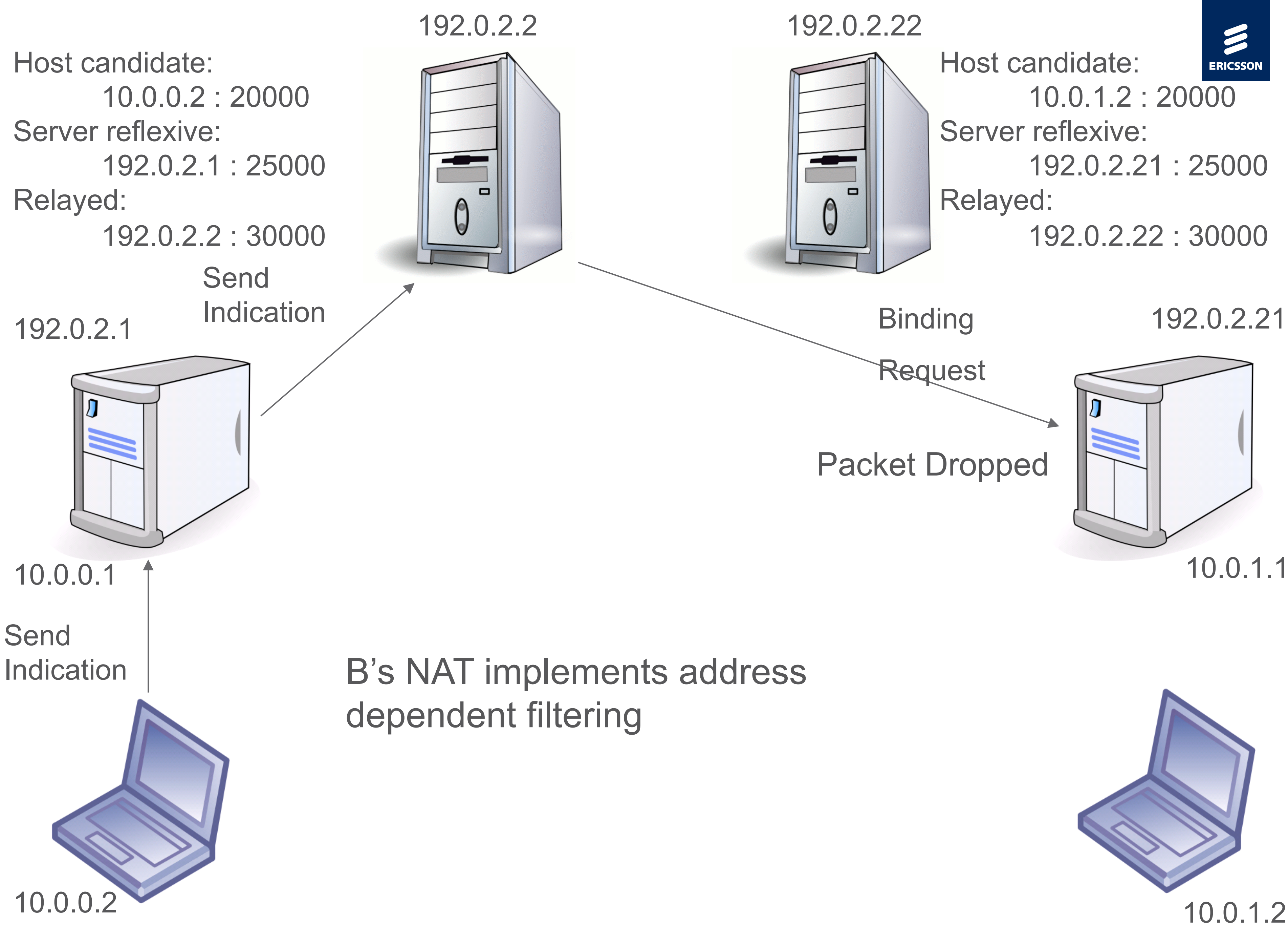

192.0.2.2
192.0.2.22
Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
192.0.2.1
192.0.2.21
10.0.0.1
10.0.1.1
Allocate Response
M: 192.0.2.1 : 25000
R: 192.0.2.2 : 30000
… and forms a candidate list that is sent to host B
INVITE (offer)
10.0.0.2
10.0.1.2

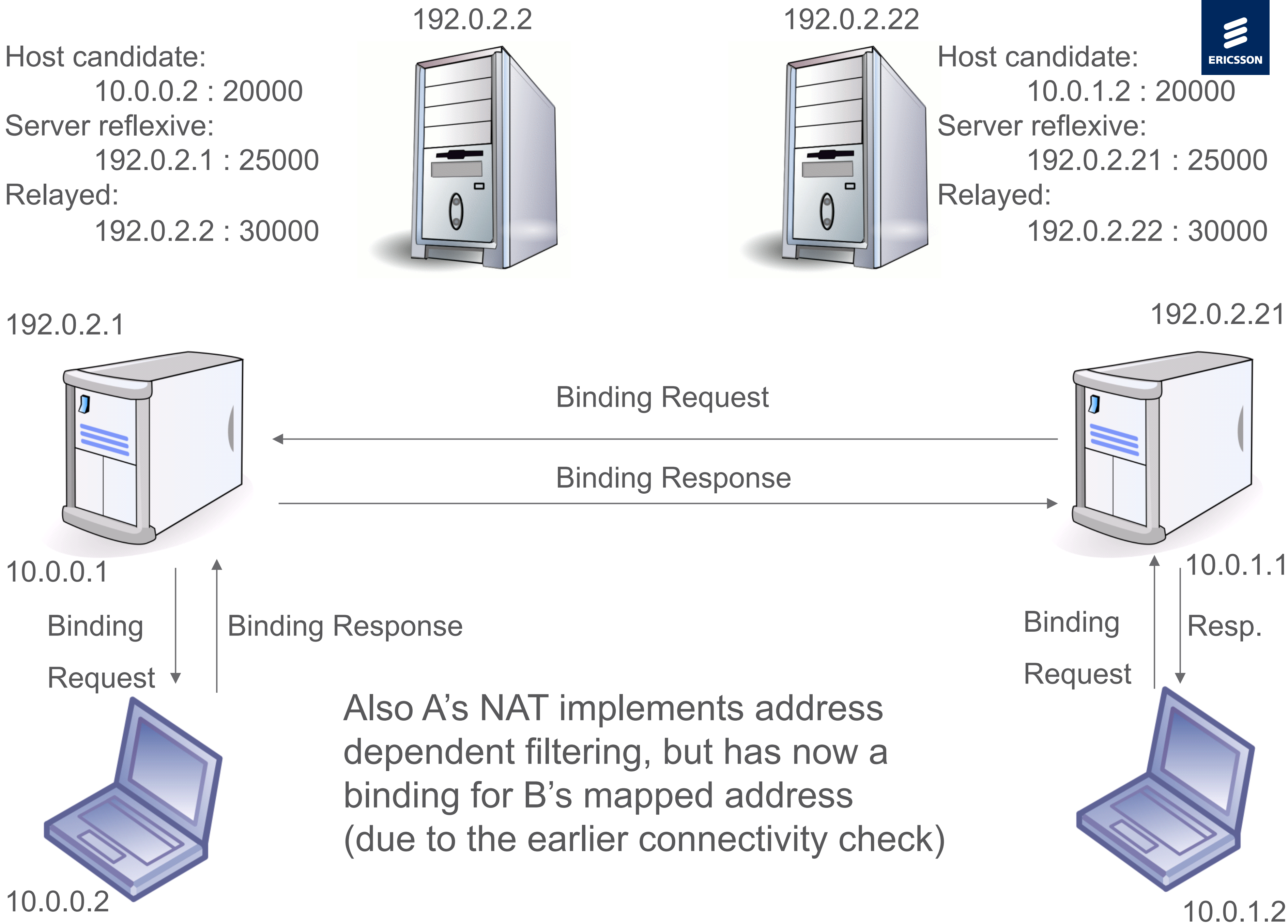

Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
192.0.2.2
192.0.2.22
Allocate
Request
Allocate Response
M: 192.0.2.21 : 25000
R: 192.0.2.22 : 30000
192.0.2.1
192.0.2.21
10.0.0.1
10.0.1.1
Allocate Response
M: 192.0.2.21 : 25000
R: 192.0.2.22 : 30000
Allocate
Request
Host B gathers candidates
10.0.0.2
10.0.1.2
ERICSSON


192.0.2.2
192.0.2.22
Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
Host candidate:
10.0.1.2 : 20000
Server reflexive:
192.0.2.21 : 25000
Relayed:
192.0.2.22 : 30000
192.0.2.1
192.0.2.21
10.0.0.1
10.0.1.1
Allocate Response
… and sends them to host A
M: 192.0.2.21 : 25000
R: 192.0.2.22 : 30000
200 OK (answer)
ACK
10.0.0.2
10.0.1.2

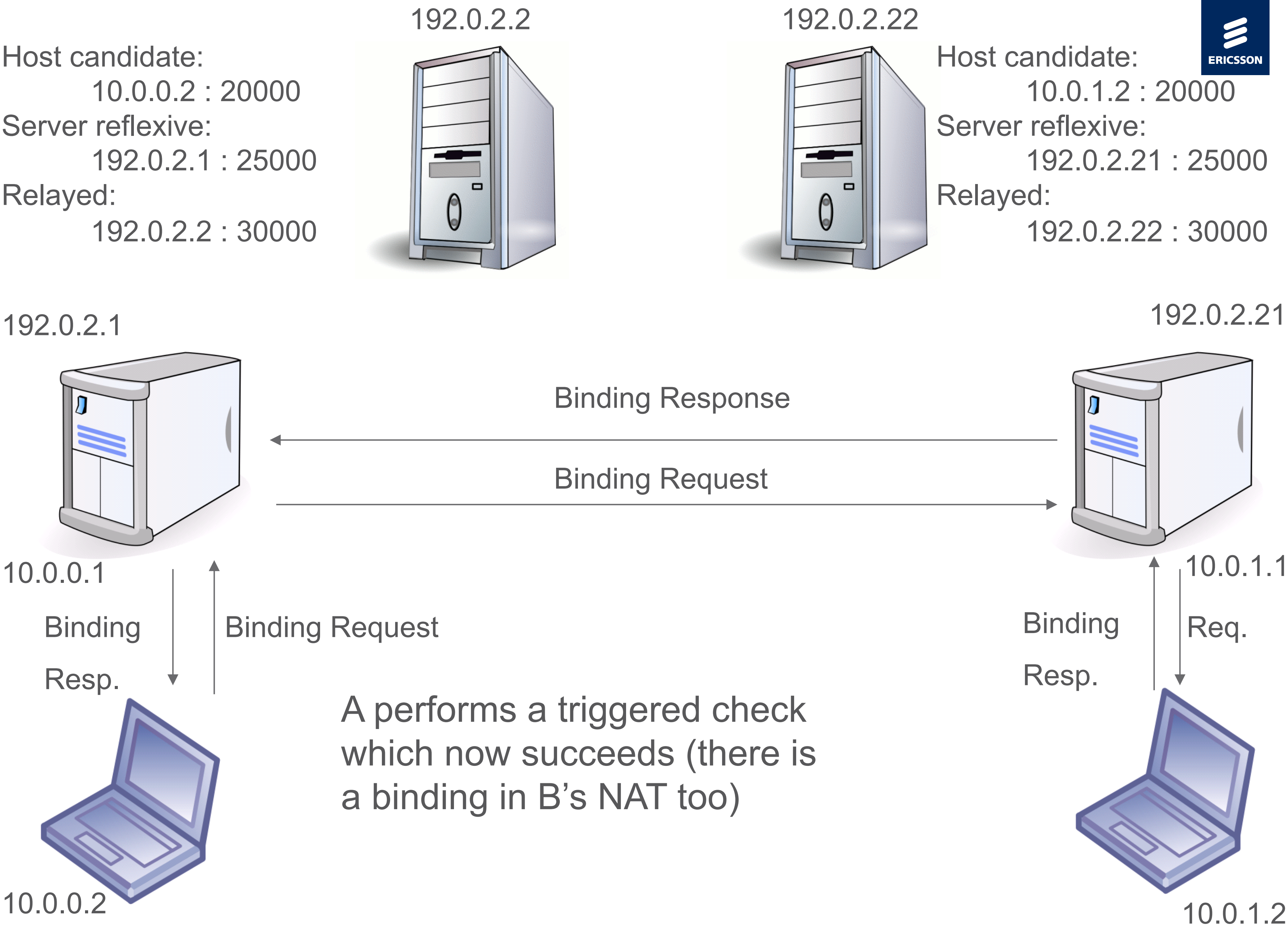

192.0.2.2
192.0.2.22
Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
Host candidate:
10.0.1.2 : 20000
Server reflexive:
192.0.2.21 : 25000
Relayed:
192.0.2.22 : 30000
192.0.2.1
10.0.0.1
192.0.2.21
10.0.1.1
Connectivity checks sent to host candidates fail due to hosts being in different subnets
Binding Request
Binding Request
Packets Dropped
10.0.0.2
10.0.1.2


192.0.2.2
192.0.2.22
Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
Host candidate:
10.0.1.2 : 20000
Server reflexive:
192.0.2.21 : 25000
Relayed:
192.0.2.22 : 30000
192.0.2.1
192.0.2.21
Binding Request
Packet Dropped
10.0.0.1
10.0.1.1
Binding
Request
B’s NAT implements address dependent filtering
10.0.0.2
10.0.1.2

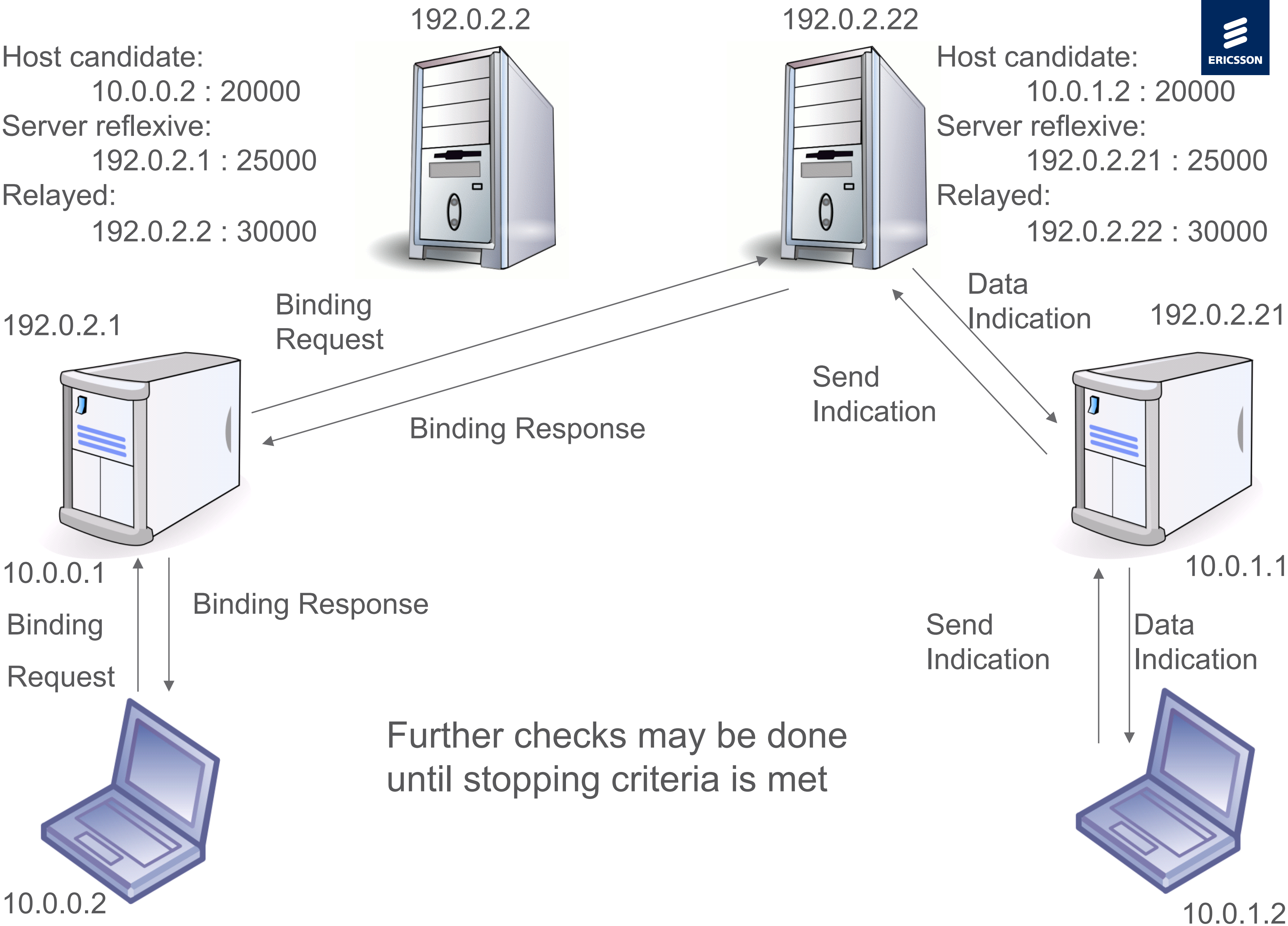

192.0.2.2
192.0.2.22
Host candidate:
10.0.0.2 : 20000
Server reflexive:
192.0.2.1 : 25000
Relayed:
192.0.2.2 : 30000
Host candidate:
10.0.1.2 : 20000
Server reflexive:
192.0.2.21 : 25000
Relayed:
192.0.2.22 : 30000
Send Indication
192.0.2.1
Binding Request
192.0.2.21
Packet Dropped
10.0.0.1
10.0.1.1
Send Indication
B's NAT implements address dependent filtering
10.0.0.2
10.0.1.2



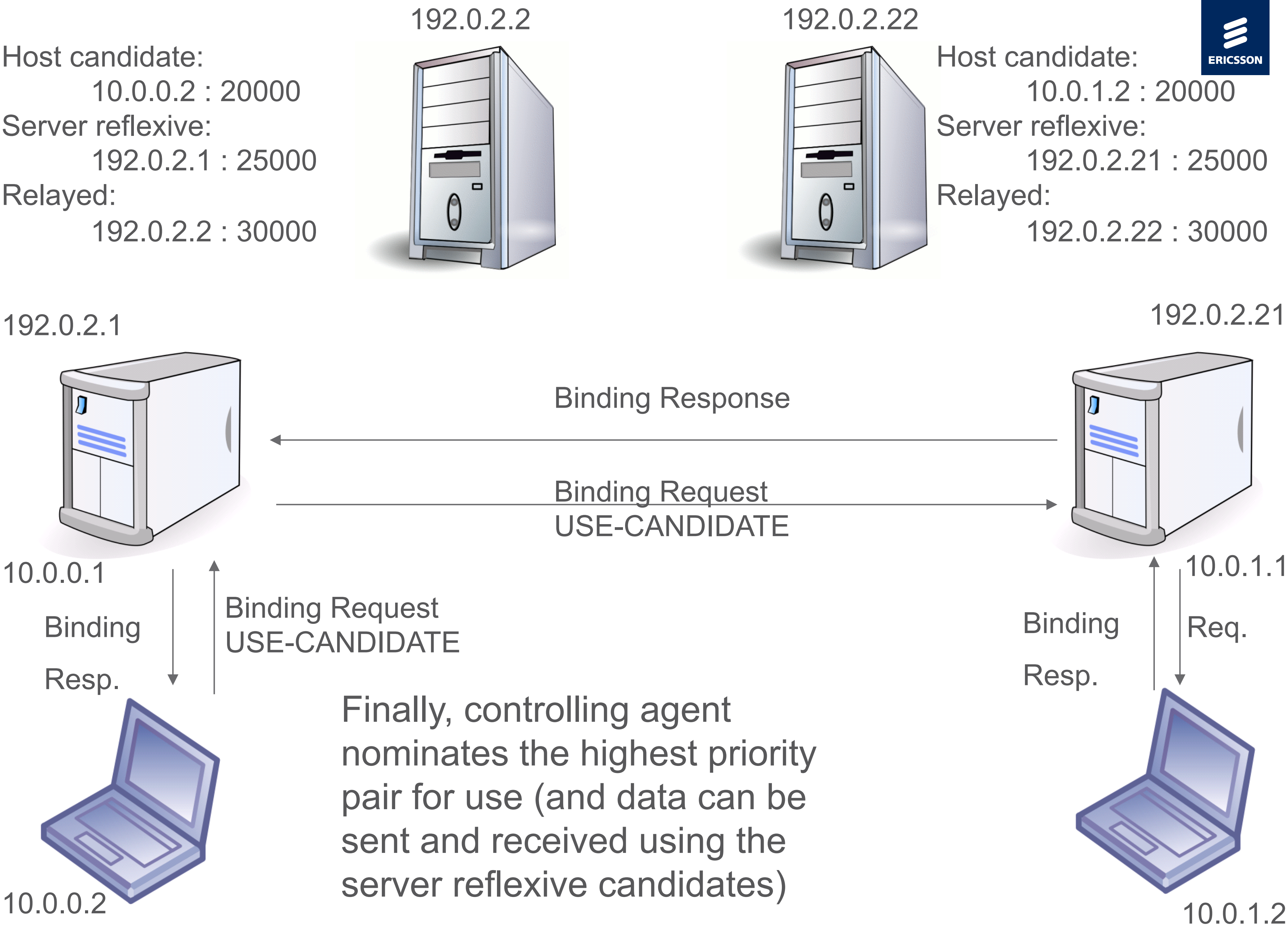
Also A's NAT implements address dependent filtering, but has now a binding for B's mapped address (due to the earlier connectivity check)



A performs a triggered check which now succeeds (there is a binding in B's NAT too)



Further checks may be done until stopping criteria is met



Finally, controlling agent nominates the highest priority pair for use (and data can be sent and received using the server reflexive candidates)

Other NAT Traversal Methods

- Middle box communications
  - Signaling with NATs to create proper state in them
  - UPnP, PCP, SOCKS, MIDCOM, etc.
- UDP/TCP hole punching
  - Number of variations for creating NAT bindings by sending packets to different addresses
  - One of the techniques used by ICE
- Transparently for applications
  - Teredo (own variant of UDP hole punching and IPv6 over UDP)
  - Host Identity Protocol (uses ICE and UDP encapsulation)
- …

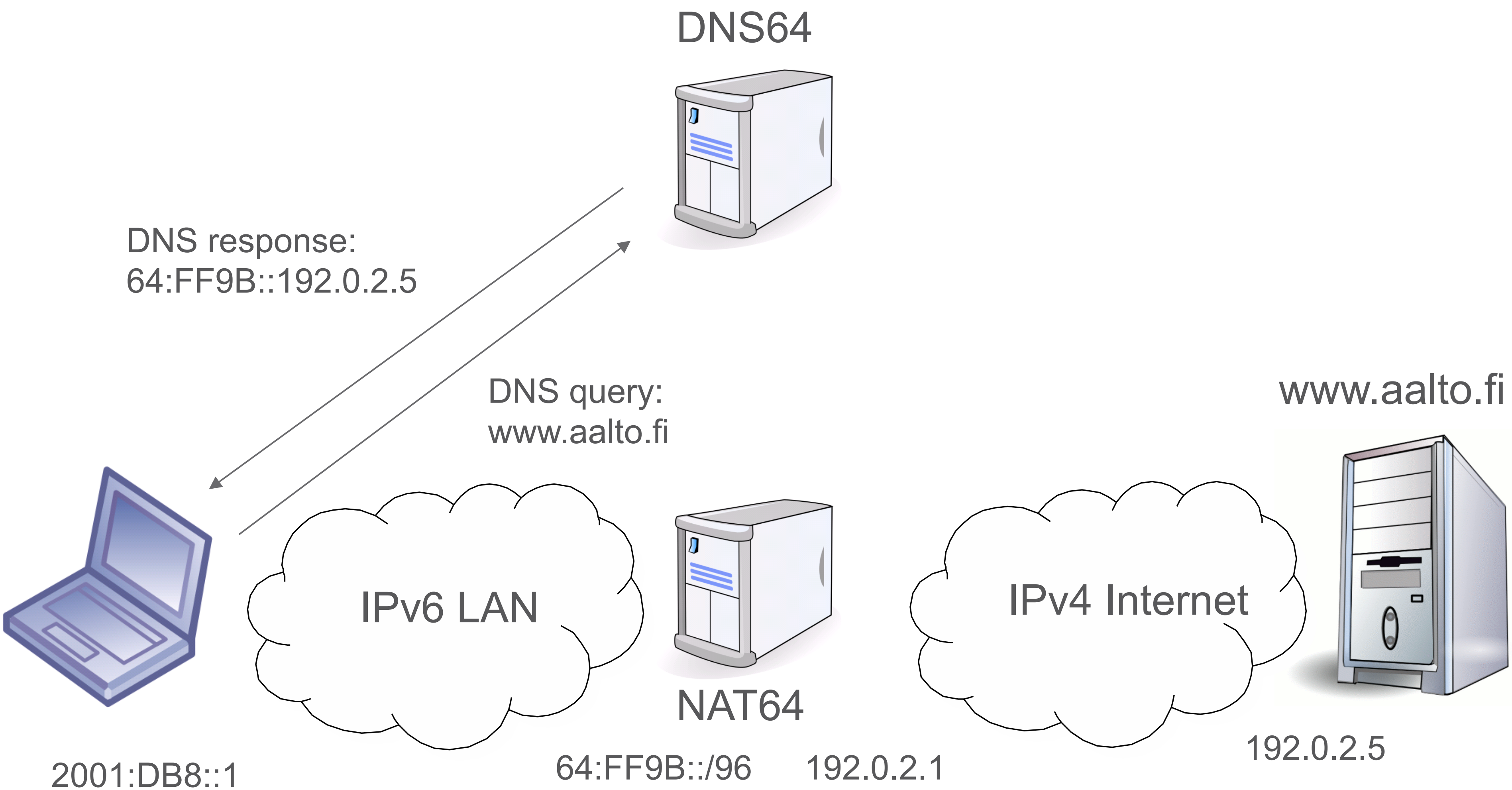
Comparing NAT Traversal Mechanisms

- ICE
  - Very effective for UDP
  - TCP more problematic (see RFC 6544)
- HIP
  - Uses ICE for creating a ”UDP tunnel” through which any (IP) protocol can be run
  - “As effective as ICE but for any protocol”
- Teredo
  - Similar UDP tunnel as with HIP
  - First version (RFC 4380) had fairly limited success
  - With extensions (RFC 6081) supports more NAT types; but still lower success probability than with ICE

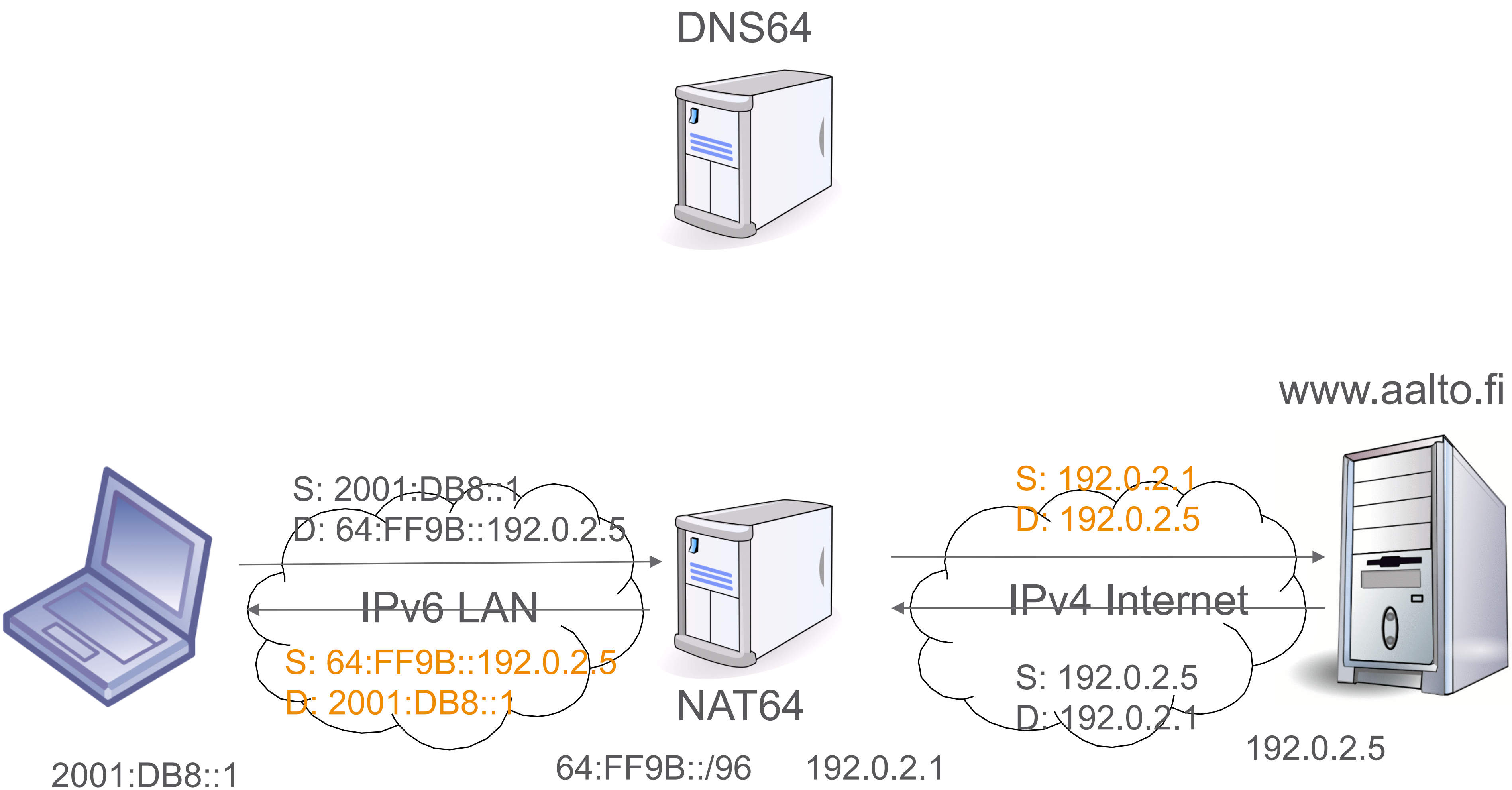
NAT64 and DNS64

- A client in IPv6-only network may need to communicate with a server in the IPv4-Internet
- NAT64 (RFC 6146) translates packets between IPv6 and IPv4
- DNS64 generates IPv6 addresses for servers that do not have one
  - Uses specific IPv6-prefix for routing traffic via the NAT64
  - Problems with hosts without a DNS entry

DNS64


DNS64
DNS response:
64:FF9B::192.0.2.5
DNS query:
www.aalto.fi
www.aalto.fi
IPv6 LAN
IPv4 Internet
NAT64
2001:DB8::1
64:FF9B::/96
192.0.2.1
192.0.2.5

NAT64


DNS64
www.aalto.fi
S: 2001:DB8::1
D: 64:FF9B::192.0.2.5
IPv6 LAN
S: 64:FF9B::192.0.2.5
D: 2001:DB8::1
S: 192.0.2.1
D: 192.0.2.5
IPv4 Internet
S: 192.0.2.5
D: 192.0.2.1
NAT64
2001:DB8::1
64:FF9B::/96
192.0.2.1
192.0.2.5

Summary

- NA(P)Ts originally invented to save IPv4 addresses
  - Can serve a whole subnet with a single IP address
  - Works (fairly well) for client-server, but breaks P2P connectivity
- NATs have different (and often un-deterministic) behavior
  - Endpoint-(in)dependent mapping and/or filtering
  - IP address and port assignment, timeouts, etc.
- NAT traversal developed to fix connectivity
  - STUN and TURN for server-reflexive and relayed addresses
  - ICE uses STUN and TURN for gathering candidates and running connectivity checks between them; tries various possible combinations and selects the best
- NAT64 provides IPv4 connectivity when network only provides IPv6

Questions?

ERICSSON