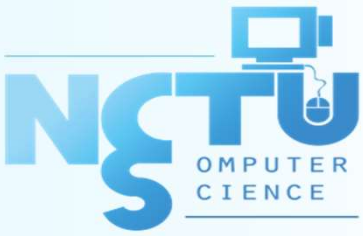


Network Address Translation, Port Forwarding and Hole Punching

Prof. Chien-Chao Tseng

曾 建 超 教授

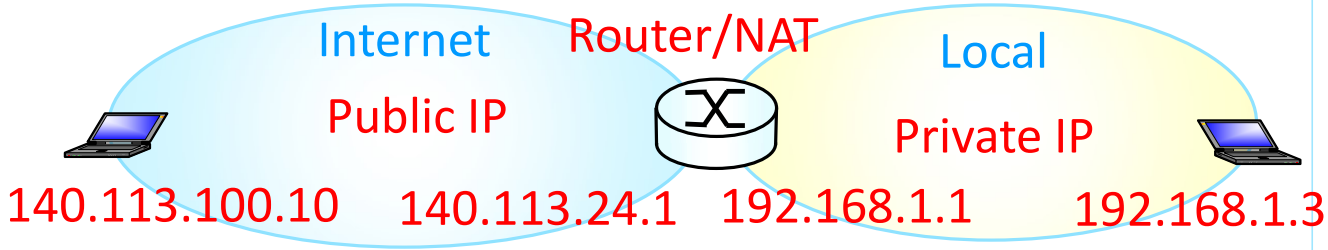
Department of Computer Science
National Yang Ming Chiao Tung University
cctseng@cs.nctu.edu.tw

References:

Depletion of IP Addresses

- Internet Protocol (IP) or IPv4 addresses: 32-bits long.
  - Total address space of IP: 0 ~ $2^{32} - 1$ = 4,294,967,295.
- Public IPs: globally unique, registered IP addresses
  - Administered by Internet Assigned Numbers Authority (IANA)
- Machine connects directly into Internet must have a public IP address

➢ Depletion of IP address



■ Solutions
  - Short term
    - Classless Inter-Domain Routing (CIDR) with subnetting
    - Private IP addresses with Network Address Translation (NAT)
  - Long term
    - IPv6 (with 128 bits address space)

Private IP Addresses

- **Three blocks of IP address space** for **private networks** (RFC 1918)
  - Reserved by Internet Assigned Numbers Authority (IANA).

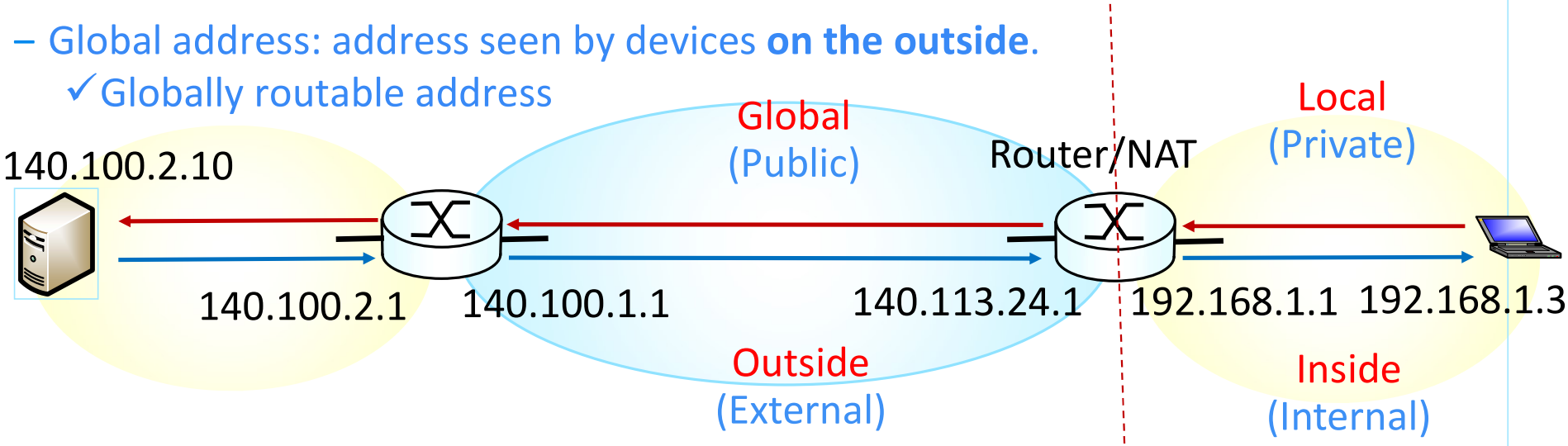
Block Size	CIDR Prefix	Private Address Space	Class	No. of NWs
24-bit Block	10.0.0.0/8	10.0.0.0 – 10.255.255.255	A	1
20-bit Block	172.16.0.0/12	172.16.0.0 – 172.31.255.255	B	16
16-bit Block	192.168.0.0/16	192.168.0.0 – 192.168.255.255	C	256

- Aim to delay IPv4 address exhaustion
- **Not Globally Delegated**
  - Used for intranets (private networks),
    - when globally routable addresses are not mandatory, or are not available
  - Cannot be transmitted onto the public Internet
  - **Network Address Translation** (NAT)

16 NWs:
172.00010000.0.0 ~ 172.00011111.255.255
172.11110000.0.0 ~ 172.11111111.255.255
Mask: 255.240.0.0

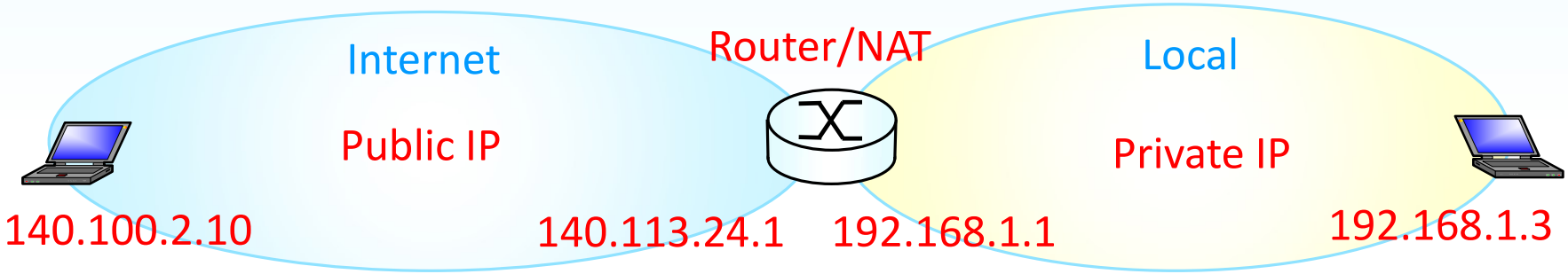
Anatomy of IP Addresses

- **NAT device** divides its universe into
  - Inside: Private network and devices connected to the network
  - Outside: Public Internet and devices reachable over Internet.
- Addresses could be classified as either ***local*** or ***global***.
  - Local address: address seen by devices **on the inside**
  - Global address: address seen by devices **on the outside**.
    - ✓Globally routable address



What is NAT?

- A method that maps IP addresses from one address realm to another

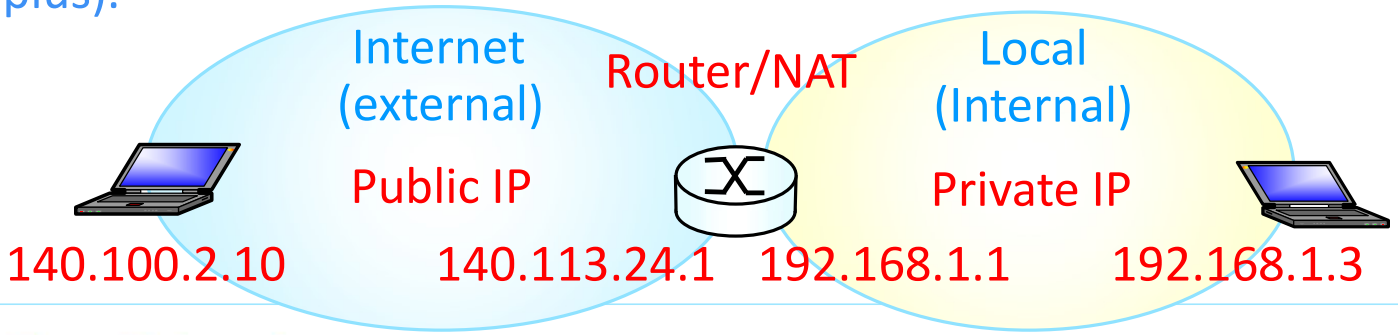


- NAT maps private addresses into globally routable ones and vice versus
  - Allows organization to use **private IP addresses** and yet **connect to the Internet**

✓ With NAT, **internal network** of an organization ***appear,*** from the outside, to be using a **different IP address space** (than what it is actually using.)

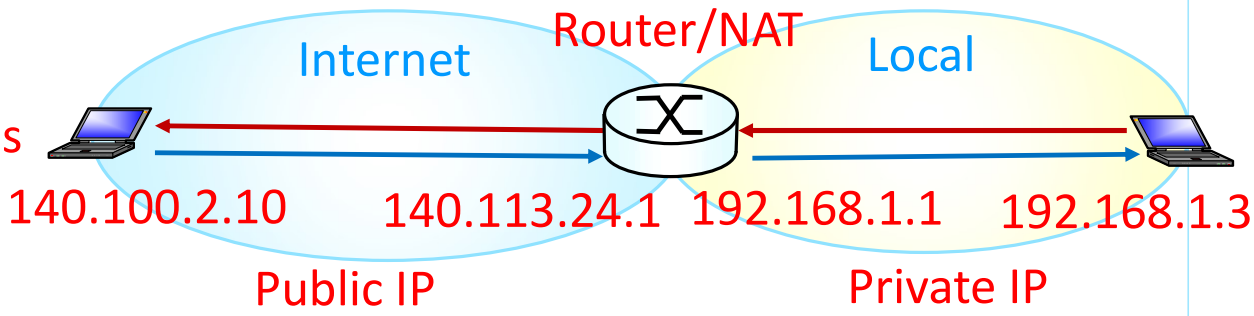
The Beauty of NAT

- Hosts on the internal network are not aware of the very existence of NAT.
- NAT configuration exists only on the router or a network device
  - typically at the boundary of the internal network.
- Local network does not need a range of addresses from ISP
  - May uses a single IP address for many devices
- Can change addresses of devices in local network without notifying outside world
  - Can change ISP without changing addresses of devices in local network
- Devices inside local network are not explicitly addressable, visible by outside world (a security plus).



Overview of NAT

- What does NAT do?
  - Re-write the source and/or destination addresses of IP packets when they pass through a router or firewall
  - What can be re-written?
    - Source/Destination IPs
    - Source/Destination Ports
- What can NAT do?
  - Solve the IPv4 address shortage. (Most common purpose)
  - Firewall (Security)
  - Load balancing (Scalability)
  - Fail over (High Availability)



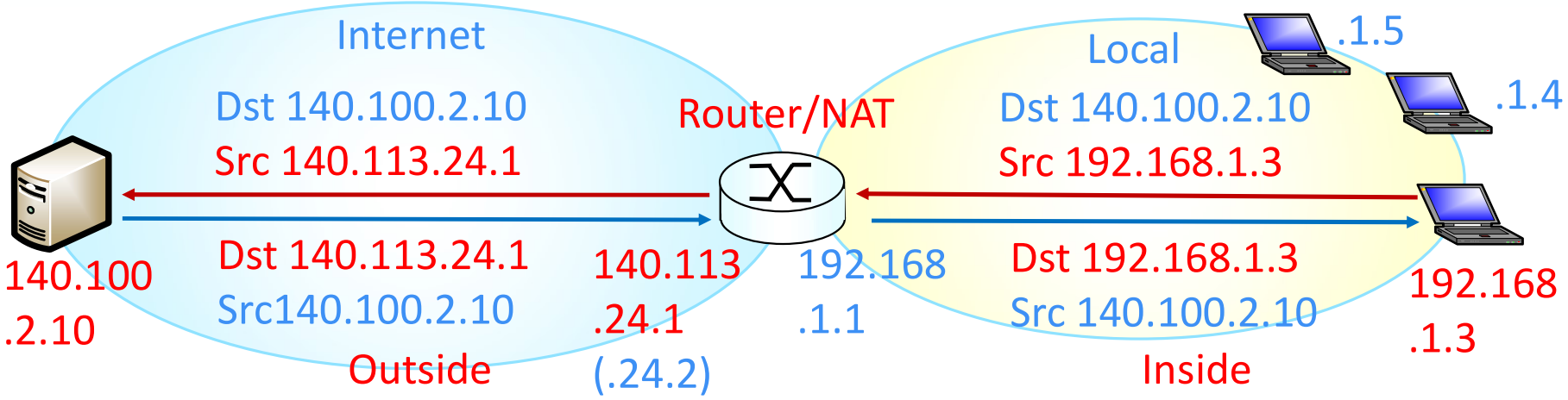
Basic NAT – First Attempt

- ✓ NAT translates inside local (private) addresses into inside global (public) addresses.
- ■ Basic NAT**:**
- ● **Static NAT:** Static address translation
  - – Statically configure **one-to-one mapping** between local and global addresses
  - – Needs **one registered public IP** address **for every host** on the inside network.
- ● **Dynamic NAT:** Dynamic address translation
  - – Dynamically map private IP addresses to registered public IP addresses
    - ▪ from a pool of available registered IP addresses
  - – Need to have many registered public IP addresses
    - ➢ **Does not help** to conserve **IP addresses**

- ➢ **NAT Overloading:** exactly what we need (later)

Basic NAT Operation

- Maps IP addresses from one group to another, transparent to end users
  - Re-writes IP addresses in the IP headers
  - keeps a record of these **re-writes** or **translations** in a **translation table**.



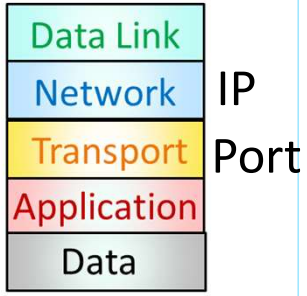
- NAT Translation Table:

External	Internal
140.113.24.1	192.168.1.3
140.113.24.2	192.168.1.5
. . .	. . .

NAT Overloading (Port Address Translation)

- A form of **Dynamic NAT**
- **Overloading: Many-to-one** mapping between **private** and **public** addresses
  - ➢ Mapping multiple **private IP** addresses to a single **registered public IP** address.
- **Network Address Port Translation** (NAPT)/**Port Address Translation** (PAT)
  - Not only track the **IP addresses**, but also the **protocol types** and **ports**
  - ➢ Translating both the **IP** address and the **port** number of a packet

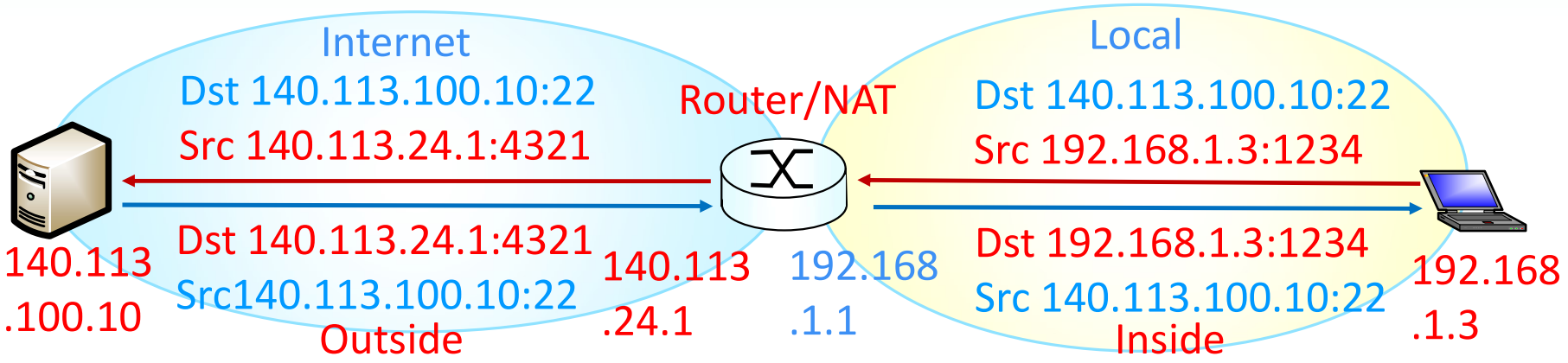
External	Internal
140.113.24.1:**4321**	192.168.1.3:**1234**
140.113.24.1:8765	192.168.1.5:5678
. . .	. . .



  - A single **inside global** address can map up to 65535 **inside local** addresses
- ➢ Many hosts can simultaneously connect to Internet using a single global IP address
- ✓ NAT Overloading helps with the issue of IP depletion problem.

NAT Port Address Translation

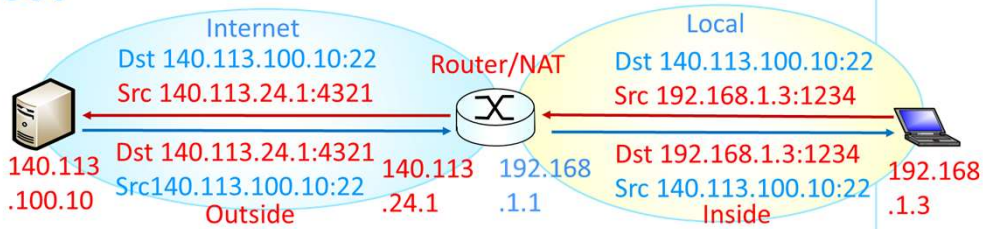
- Assigns transport identifiers (ports) for connections and
- Maps internal transport addresses (IP and port) to external transport addresses
- Records mapping and re-write transport addresses in the IP headers



External	Internal
140.113.24.1:4321	192.168.1.3:1234
140.113.24.1:8765	192.168.1.3:5678
140.113.24.1:9999	192.168.1.5:1234

How NAT/NAPT Works

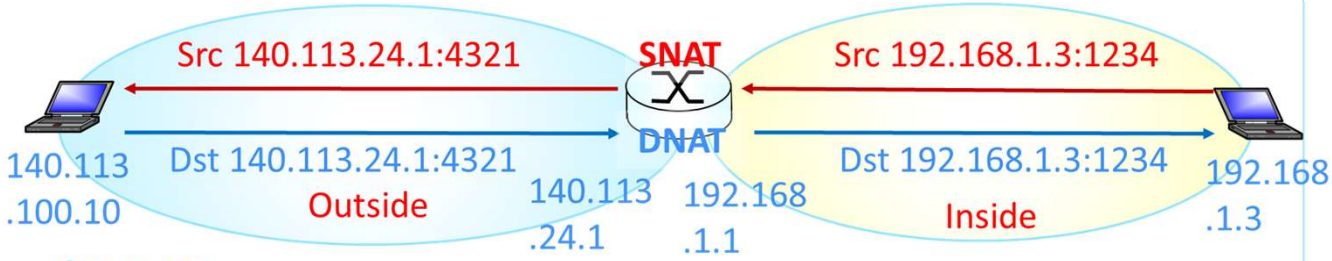
- *For outgoing datagrams:*
  - *Replaces* (Internal: IP address, Port #) in **Source Fields** with (External: NAT IP address, NAT Port #),
  - *Sends* replaced datagram to the remote host
    - Remote host responds using (External: NAT IP address, NAT Port #) as **Destination Transport address**



  - *Remember (in NAT translation table)* every (Internal: IP address, Port #) to (External: NAT IP address, NAT port #) translation pair
- *For incoming datagrams:*
  - *Replace* (External: NAT IP address, NAT Port #) in **Destination Fields** with corresponding (Internal: IP address, Port #) stored in NAT table

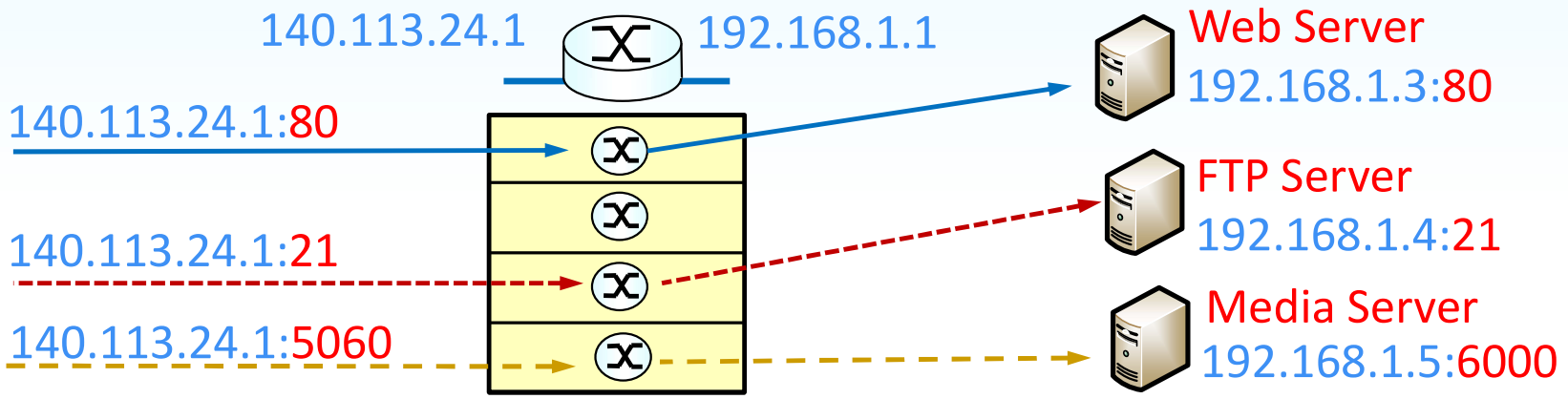
SNATs and DNATs

- **Source NAT (SNAT)**
  - Source IP address is changed but destination IP address is maintained.
  - Allows a host on the "**inside**" of the NAT with a private IP address to initiate a connection to a host on the "**outside**" of the NAT.
- **Destination NAT (DNAT):**
  - Source IP address is maintained but destination address is changed.
  - Allows a host on the "**outside**" with a public IP to connect to a host on the "**inside**" with a private IP.



- **Port forwarding**
  - A common application of DNAT
  - Redirects a packet from one transport address to another.
  - Commonly used to **publish** a service located in **a private network**

Illustration of Port Forwarding



- NAT Translation Table:

External	Internal
140.113.24.1:80	192.168.1.3:80
○ ○ ○	○ ○ ○
140.113.24.1:21	192.168.1.3:21
140.113.24.1:5060	192.168.1.5:6000

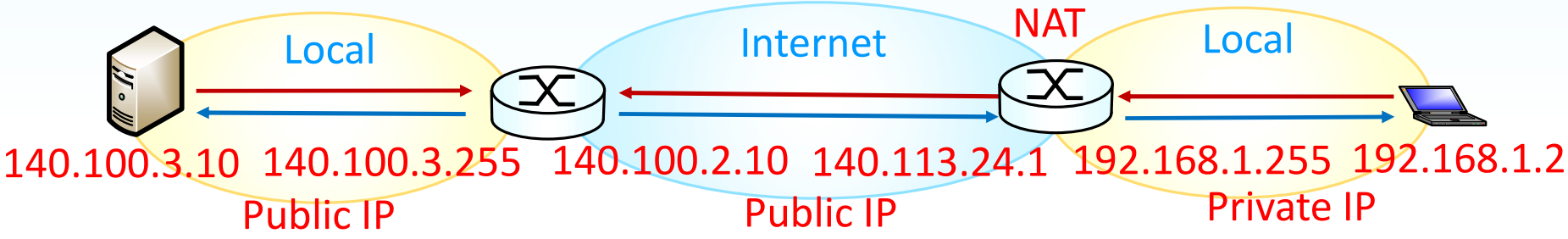
- **Notes:**
  - **SNAT** is performed after routing decision.
  - **DNAT** is performed before routing decision
- ✓ Recall: IP is destination-based routing

Issues of NAT

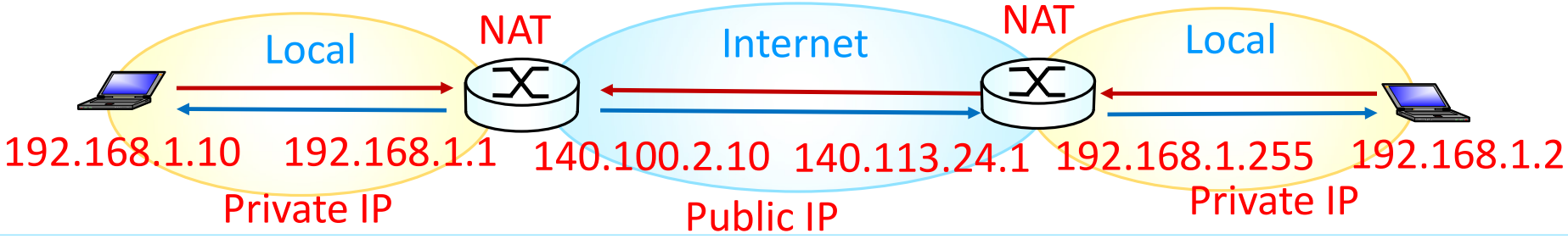
- Solve IPv4 address shortage problem: with 16-bit port-number field
  - 60,000 simultaneous connections with a single LAN-side address!
- Security, Load Balance, High Availability
- Controversial Issues:
  - Routers should only process up to layer 3?
  - Violates end-to-end argument
    - Host should be talking directly
      - without interfering nodes modifying IP addresses and port numbers
  - Address shortage should instead be solved by IPv6?
  - NAT Traversal Problem
    - **External hosts cannot initiate connection** to hosts (servers) within an NAT
      - Application designers must take NAT possibility into account,
        - » E.g., P2P applications

NAT Traversal Problem

- Client-Server Application



- NAT Traversal Problem: When peers behind NATs.
  - Cannot know the address of another peer behind an NAT
  - NAT may block unsolicited inbound packets

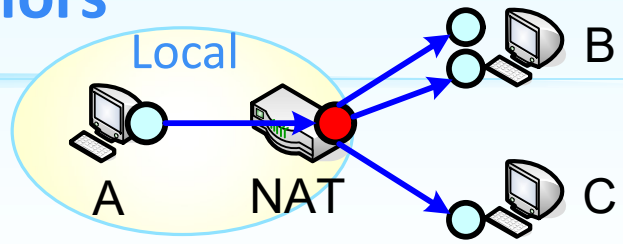


Problems of NAT Traversal

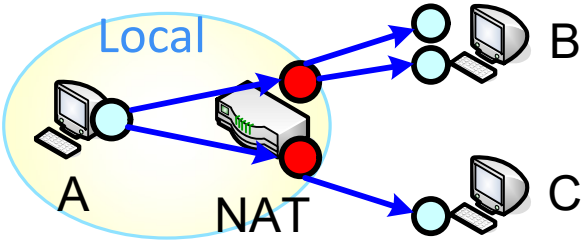
- **Three Characteristics of NAT behavior** affect TCP NAT traversal:
  1) **NAT Mapping Rules**
     - Determine how **mapped-addresses** allocated for **outbound packets** (sessions).
  2) **NAT Filtering Rules**
     - Determine which **inbound packets** can **traverse across NAT** via **existing mapped-address**.
  - Many P2P applications use TCP to transport packets
    - TCP 3-way handshake for connection establishment
  3) **NAT TCP State Tracking**
     - May block unexpected TCP packet sequence
     - Many NATs implement some sort of TCP state tracking mechanism
     - ➢ More difficult to establish a TCP connection between two host behind NATs

NAT Mapping Behaviors

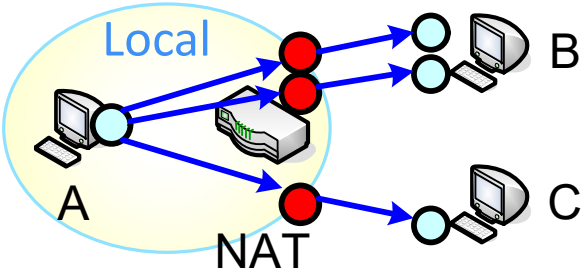
- **Mapping Behaviors:**
  Determine how **mapped-addresses** are allocated for **outbound packets** (sessions).
- **Three Types** of **NAT Mapping Behaviors**
  1) **Endpoint-Independent**
     Reuse mapped-address for subsequent sessions (independent of the destination).
  2) **Address-dependent**
     Same mapped-address for subsequent sessions to same destination IP.
  3) **Address and Port-dependent**
     A unique mapped-address for each outgoing session to a specific transport address.



(a) Independent



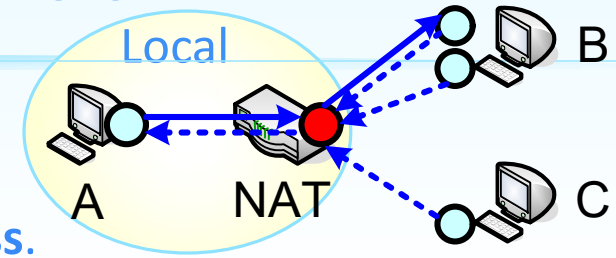
(b) Address Dependent



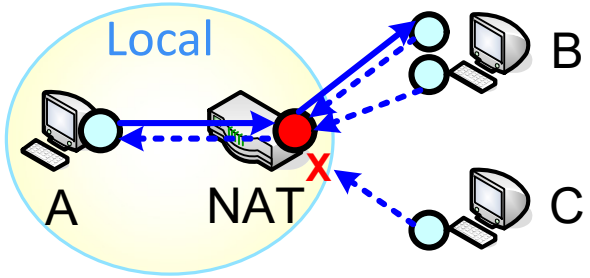
(c) Address & Port Dependent

NAT Filtering Behaviors

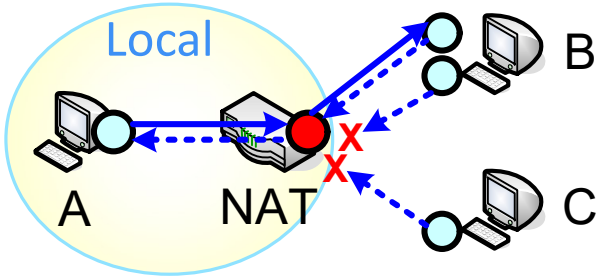
- **Filtering Behaviors**:
  Determine which inbound packets
  can **traverse across NAT** via **existing mapped-address**.
- **Three Types** of NAT Filtering Behaviors:
  1) **Endpoint-Independent**
     - Allow inbound packets from
       any transport addresses.
  2) **Address-dependent**
     - Allow inbound packets from
       the **IP address** ever sent to.
  3) **Address and Port-dependent**
     Allow only the inbound packets from
     the **transport address** (IP and Port) ever sent to.



(a) Independent



(b) Address Dependent



(c) Address & Port Dependent

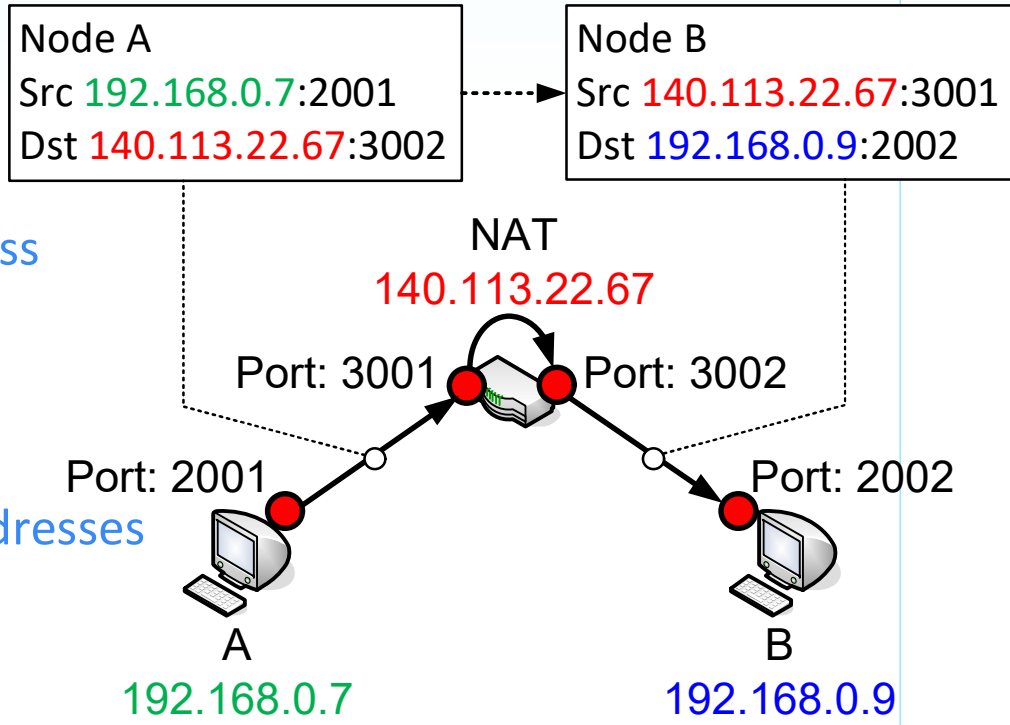
Types of NATs

- **Mapping behaviors**
  classify NATs into **Cone** and **Symmetric** NAT.
- **Filtering behaviors**
  further classify **Core NATs** into **three types of Cone NATs**.
- **Cone NAT**:
  Same mapped-address for each outgoing session (Independent mapping).
  - **Full Cone** (FC): Independent filtering.
  - **Address-Restricted** Cone (AR): Address-dependent filtering.
  - **Address and Port-Restricted** cone (PR): Address and port-dependent filtering.
- **Symmetric NAT** (SY):
  A unique mapped-address for each outgoing session to a different transport addr.
  - Normally with **Address and Port-restricted** filtering.

Hairpin (Loopback) Translation

- NAT translation that allows two hosts behind the same NAT exchange packets via their external mapped-addresses.

∗ E.g., Node A sends a packet to Node B.
  – NAT need to direct packets destined for its external address to its local address



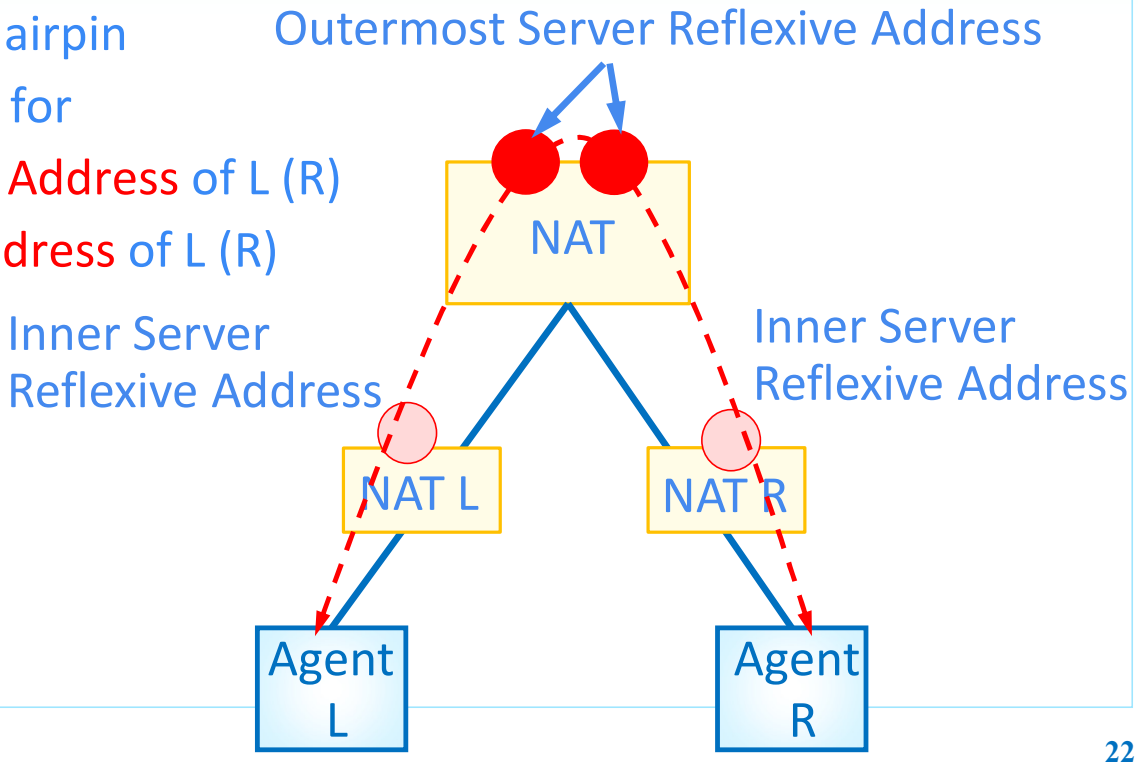
- Normally, two hosts behind same NAT establish a direct connection with local addresses
- How about in a Multi-level NAT Network?

Direct Connection in Multi-level NAT Environment

- Hairpin Translation needed in a multi-level NAT environment
  - Both nodes A and B acquire same outmost NAT address as server reflexive addresses
- Outmost NAT need to support hairpin
- i.e., Can direct packets destined for

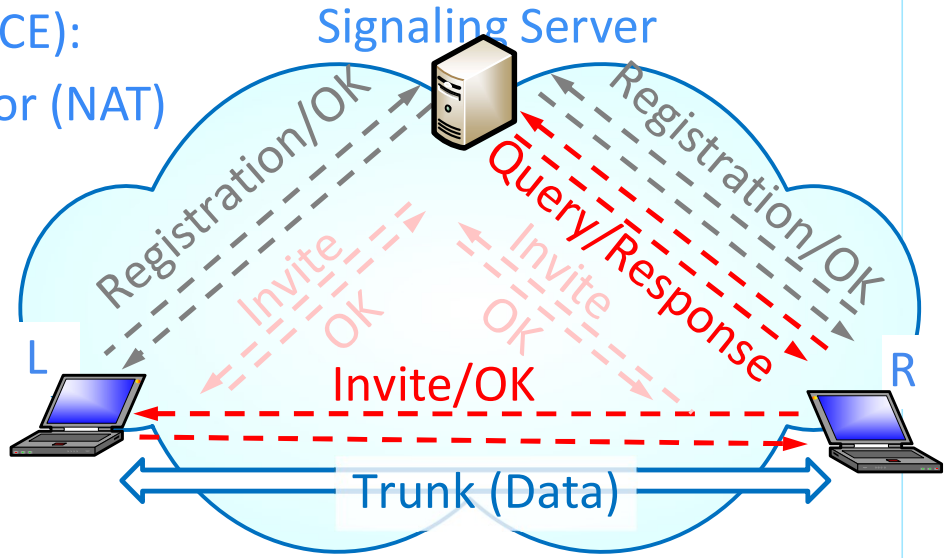
  Outermost Server Reflexive Address of L (R)

  to inner Server Reflexive Address of L (R)



NAT Traversal Methods for P2P Communication

- Also known as “Hole Punching”
- Standards for NAT “Hole Punching”
  - Session Traversal Utilities for NAT (STUN,) RFC 8489
  - Traversal Using Relays around NAT (TURN,) RFC ~~5766, 6156~~, 8656
  - Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal, RFC ~~5245~~, 8445

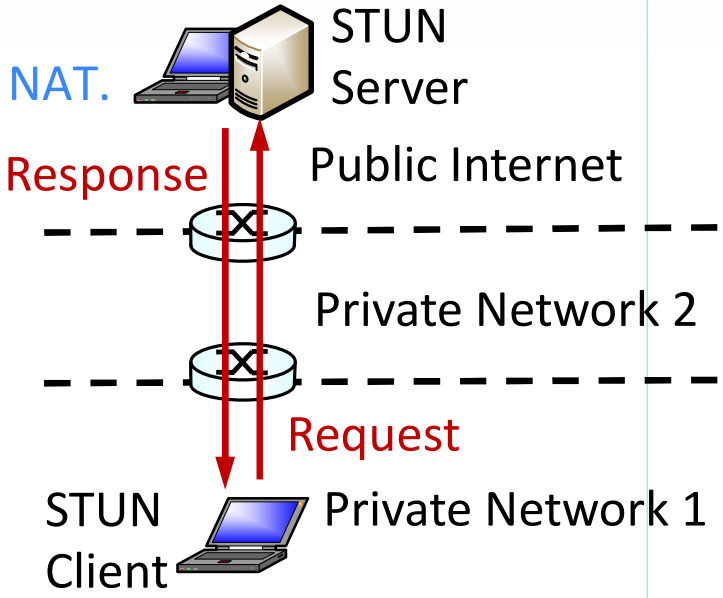


- Network Address Translator (NAT)-Friendly Application Design Guidelines, RFC 3235

Session Traversal Utilities for NAT (STUN)

- RFC ~~5389~~, 8489

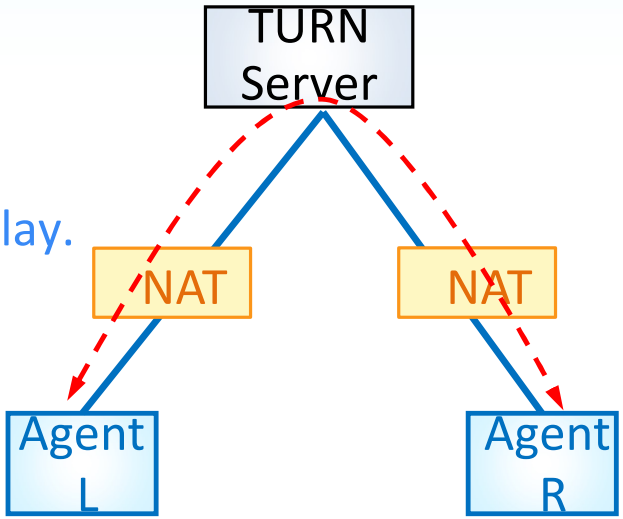
  A protocol that serves as a tool for other protocols in dealing with NAT traversal.
- Can be used by an endpoint
  - to **determine** the **IP address and port** allocated by a NAT.
  - to **check connectivity** between two endpoints and
  - as a keep-alive protocol to maintain NAT bindings.
- Does not require any special behavior from NAT.
- Not a NAT traversal solution by itself.
- **STUN Messages**: Binding request/response
- **Server Reflexive Transport Address**
  - Source transport address of binding request received by server
  - Public IP address and port created by the NAT closest to server.



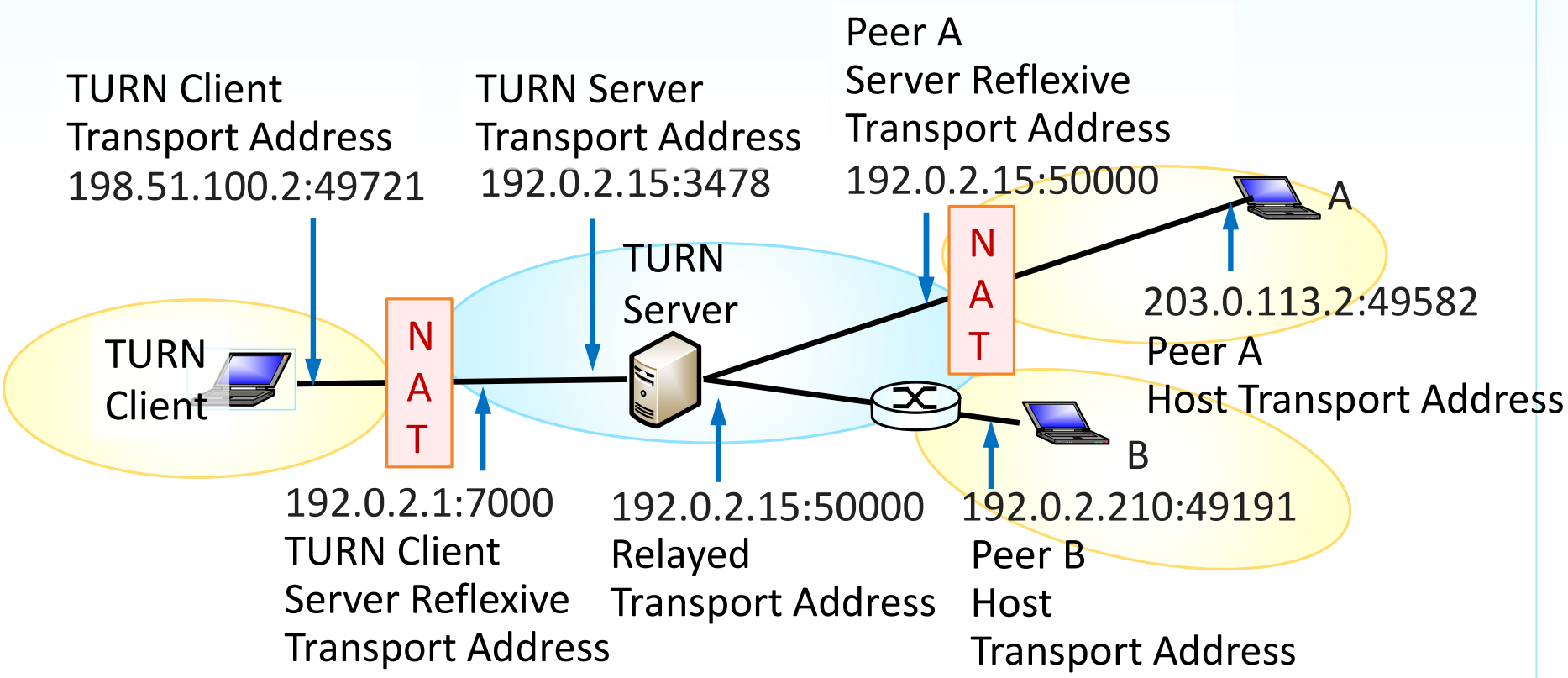
Traversal Using Relays around NAT (TURN)

- RFC ~~5766, 6156~~, 8656
  - A protocol that allows the host to **control** the operation of the relay and to exchange packets with its peers using the relay.
  - Used when direct communication is impossible
- **TURN (Relay) Server**

  An intermediate node that acts as a communication relay.
- Designed to be used as **Part of** the **Interactive Connectivity Establishment** (ICE) approach to NAT traversal,
  - Can also be used without ICE.
- Allows **a client** to communicate with **multiple peers** using a **single relay address**.

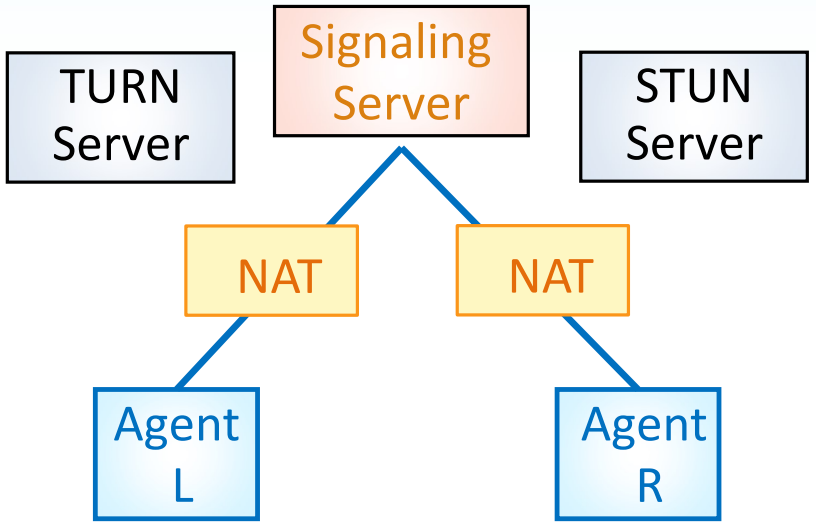


A Typical TURN Deployment



Interactive Connectivity Establishment (ICE)

- RFC 5245, 8445, Interactive Connectivity Establishment (ICE):
  - A protocol for Network Address Translator (NAT) traversal for UDP-based communication.
  - Makes use of STUN protocol and TURN
- Assumes agents can establish a signaling connection with each other
  - Ex., Via SIP Signaling Server

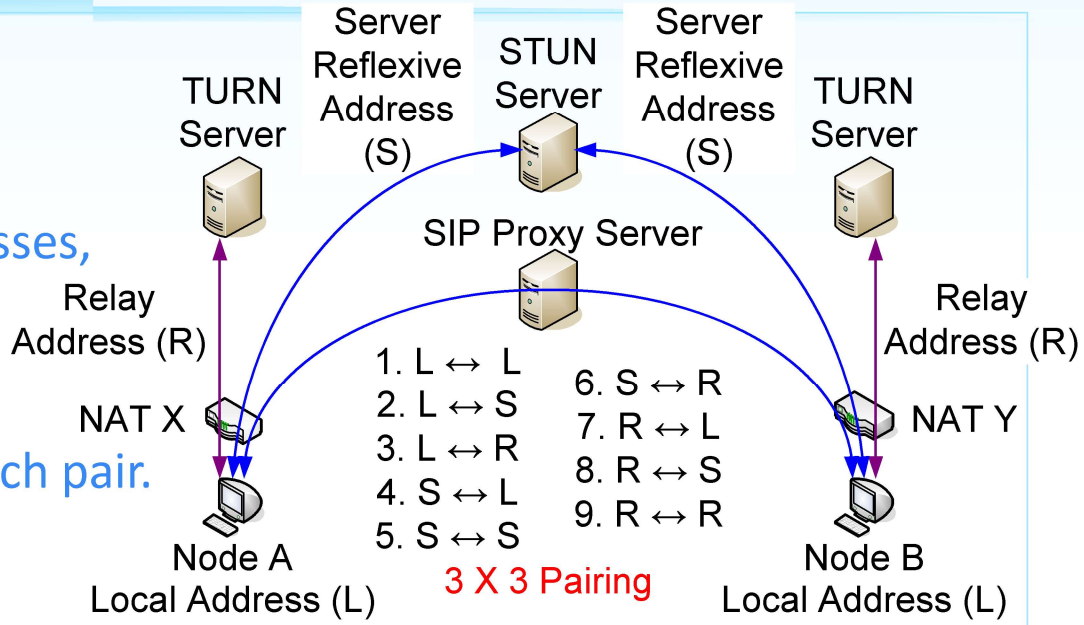


- RFC 6544, TCP Candidates with Interactive Connectivity Establishment (ICE)
  - extends ICE to TCP-based media, including the
  - ability to offer a mix of TCP and UDP-based candidates for a single stream.

Overview of ICE Procedure

1. Gathering Candidates

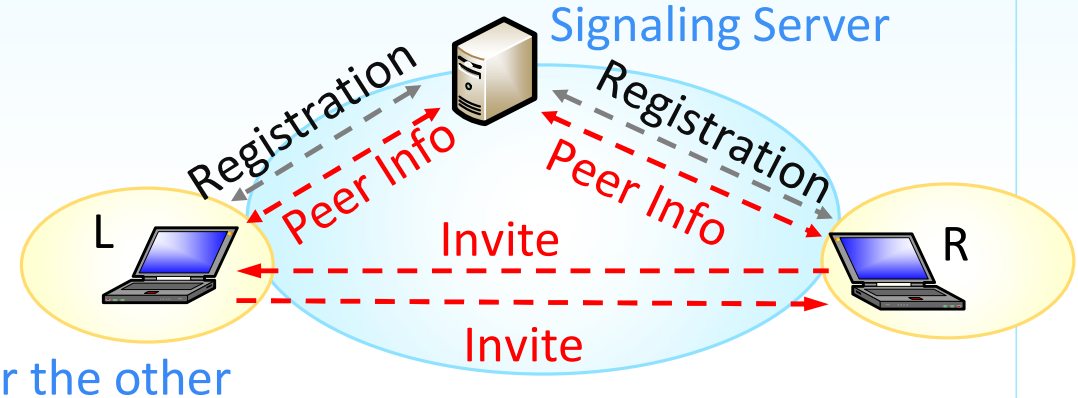
2. Connectivity Checks
   - Hosts exchange 3 candidate addresses, via signaling server
     - Makes 9 candidate pairs.
   - Performs connectivity check for each pair.



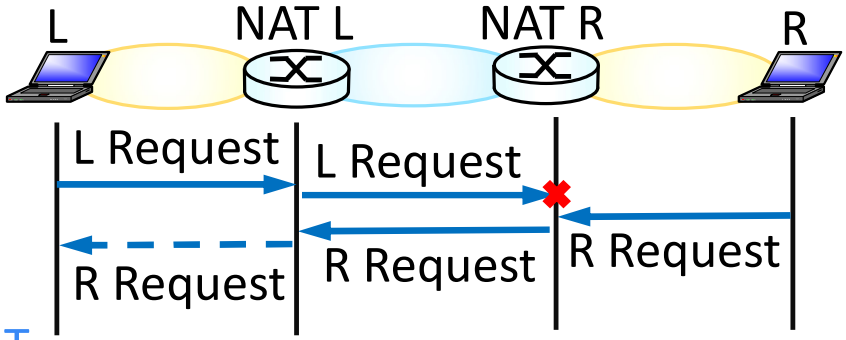
3. Nominating Candidate Pairs and Concluding ICE
   - TURN Server may provide server reflexive address too
   - Assigns one of the ICE agents as the controlling agent,
   - Controlling agent nominates a valid pair
   - Only selected pairs will be used for sending and receiving data

Illustration of Hole Punching Mechanism

- NAT Traversal with UDP
  - Host acquires reflexive address of peer from signaling server
  - Each host initiates a request to the peer's reflexive address
    - May punch a hole proactively for the other
  - ➢ Result depending on message sequence, and type of peer NATs

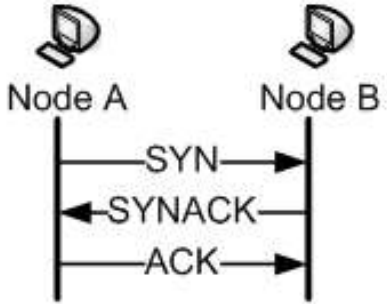


- Example:
  - NAT R does not have an addr. binding yet
  - NAT L already has an addr. Binding
  - But what if NAT L is
    - A Symmetric NAT or
    - An Address and Port-Restricted Core NAT

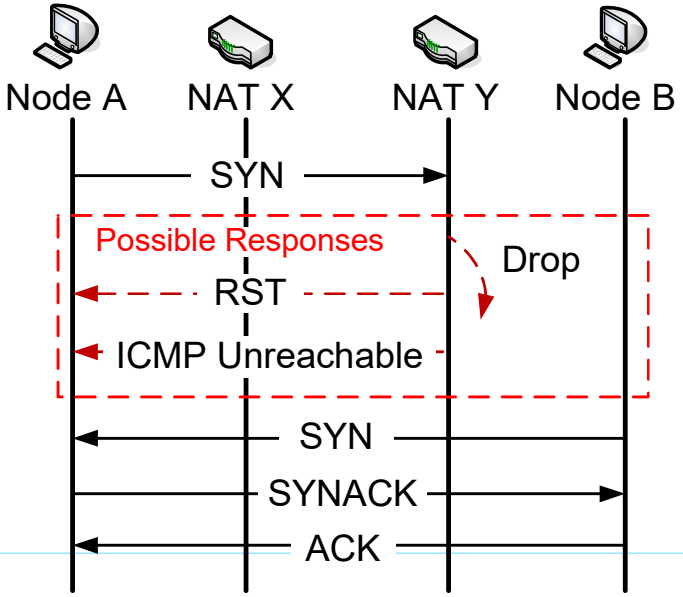


NAT TCP State Tracking

- TCP uses a 3-way handshaking procedure to establish a connection
- Some NAT implements **TCP State Tracking** mechanism to track TCP stages
  - Make **TCP NAT Traversal** much more difficult than **UDP NAT Traversal**
- Possible reaction of NAT with TCP state tracking for **unsolicited SYN**
  - Silent Drop
  - TCP Reset (RST)
  - ICMP Host Unreachable
    - TCP Three Way Handshaking



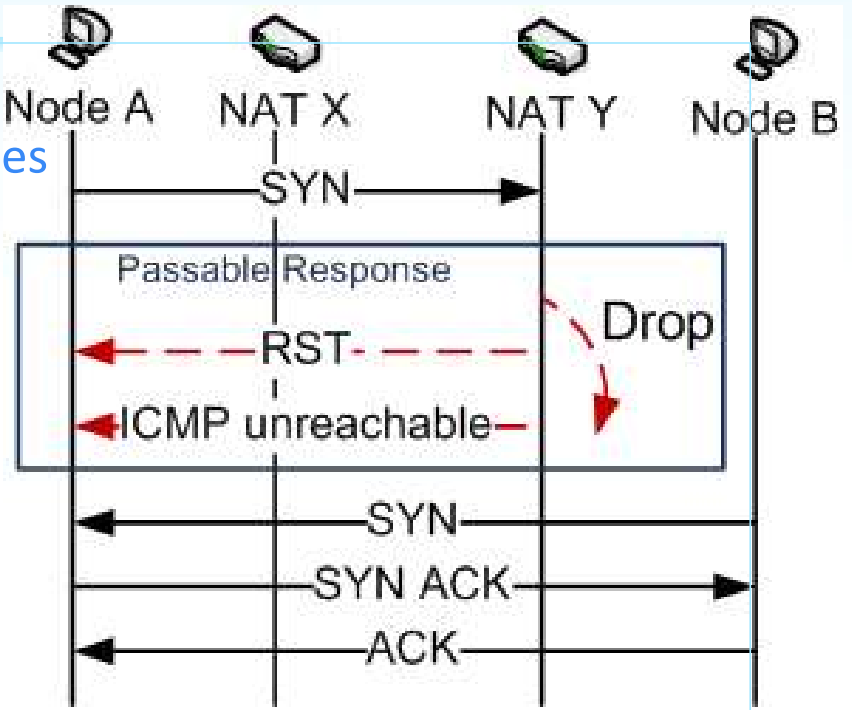
- Possible Reaction with TCP State Tracking



TCP NAT Traversal Methods

- Some TCP NAT Traversal Methods
  - Works for NATs that allow such packet sequences
    1. SYN with Normal TTL (SNT)
    2. SYN with Low TTL (SLT)
    3. Establish then SYN-in (ESi)

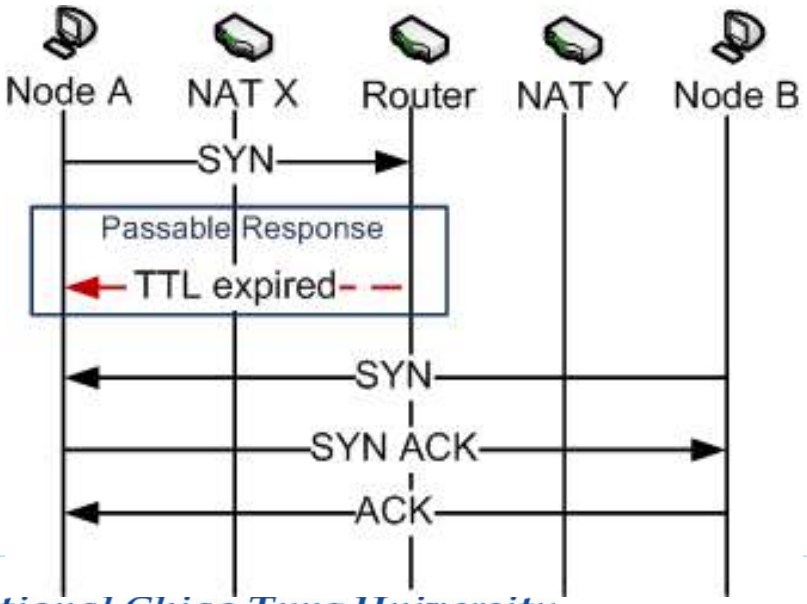
1. SYN with Normal TTL (SNT)
   - unsolicited SYN-in may cause an NAT to
     - Generate RST or ICMP messages, or even
     - Close the mapping
   - NAT X must support corresponding packet sequences
     - SYN-out → SYN-in,
     - SYN-out → RST-in → SYN-in, or
     - SYN-out → ICMP_Host_Unreachable-in →SYN-in



TCP NAT Traversal Methods (Cont.)

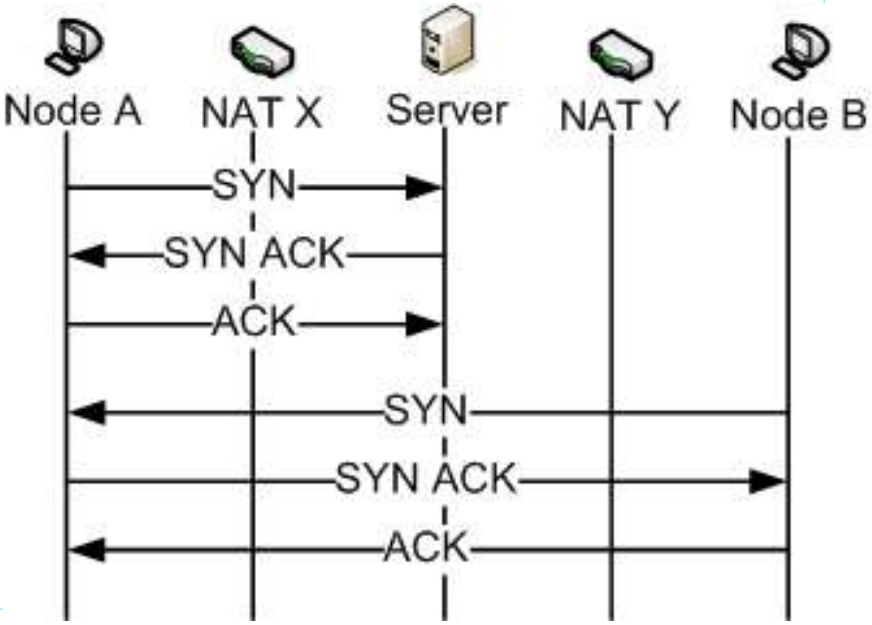
2. SYN with Low TTL (SLT)

- NAT X must support the following packet sequence
  - SYN-out → ICMP_TTL-in → SYN-in
- What value of TTL to set?



3. Establish then SYN-in (ESi)

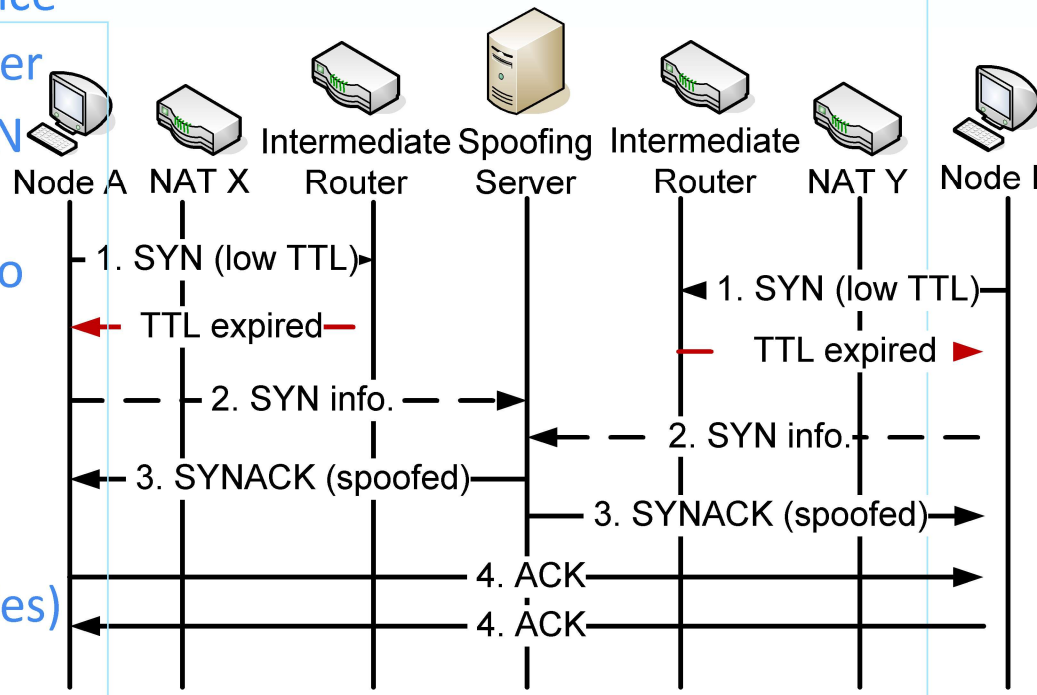
- Work for independent mapping and filtering rules
  - few NAT of such type



IP Spoofing (1/2)

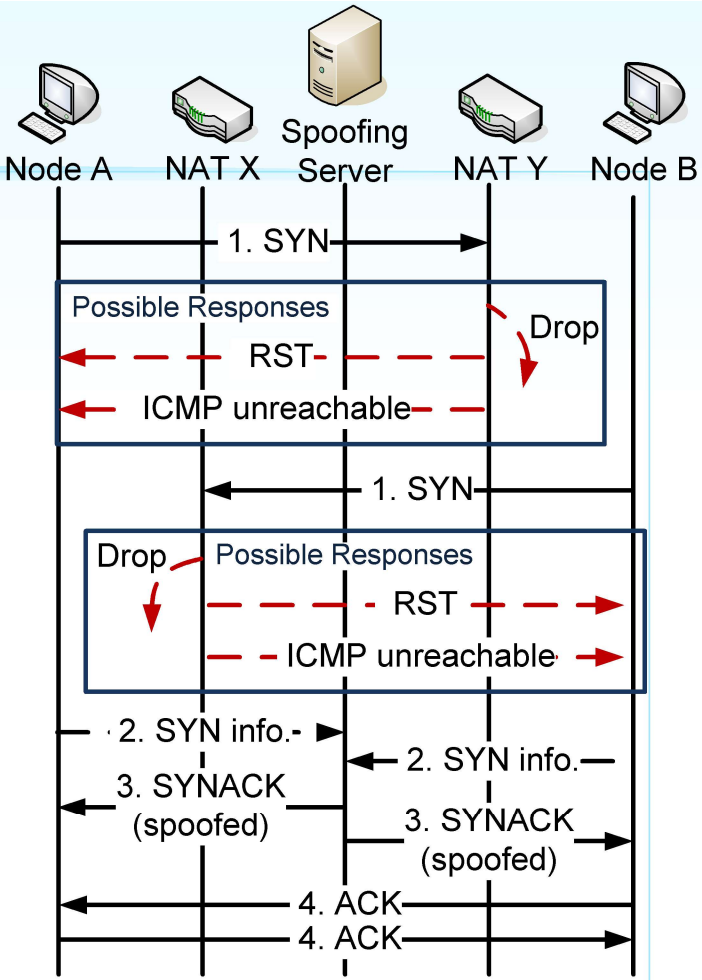
- IP spoofing
  - Some NATs do not accept "SYN-out RST-in SYNACK-in"
  - Set SYN with low TTL to avoid such sequence

1. Nodes send SYN with low TTL to each other
2. Some intermediate routers encounter SYN TTL expired.
   the routers send ICMP TTL-expired back to the sending nodes
3. Both nodes send SYN information to Spoofing server
4. Spoofing server spoofs SYNACK to the sending nodes (on behalf of the peer nodes)
5. Nodes sends ACK back to the peer nodes



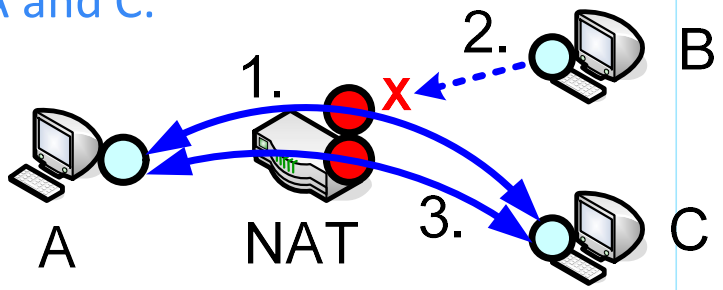
Problem with IP Spoofing

- If the TTL value of SYN is not low enough
  - Three possible responses at destination NAT:
    - Silent Drop
    - TCP Reset (RST)
    - ICMP Host Unreachable
- Possible sequences generated by IP spoofing
- TTL is low enough
  - SYN-out → ICMP_TTL-in → SYNACK-in
- TTL is too high
  - SYN-out → (Silent Drop by peer NAT) → SYNACK-in
  - SYN-out → RST-in (by peer NAT)→ SYNACK-in
  - SYN-out → ICMP_Host_Unreachable-in (by peer NAT) → SYNACK-in
- IP Spoofing may fail, if destination NAT responses RST



Connection Tracking (ConTrack)

- Keep track of all logical network connections or sessions.
- Allows NAT tracking sessions
  - Blocks unexpected session requests.
  - ➢ Leads to call-role sensitivity problem.
- E.g.,
  1. A and C created a session.
  2. B sends packets to mapped-address created by A and C.
     - ✓ NAT filters out B's packets.
     - ✓ NAT blocks the mapped-address.
  3. NAT assigns a new mapped-address for A and C.



- ➢ Initiator-change approach for ConTrack NAT.

Behavior-aware NAT Traversal

- NAT deployment is normally static.
  - Utilize network Contexts or NAT behaviors to assist NAT Traversal
- E.g., Initiator-change approach:
  1. Suppose A and C established a session.
  2. B informs A of its mapped-address.
  3. A initiates a session with node B to avoid connection tracking of NAT.