CS3103 Computer Networks Practice

IP-options, ICMP, NAT-Hole Punching

IPv4 – Options Processing
Internet Control Message Protocol (ICMP)
Ping and Traceroute
NAT Traversal (Port Forwarding and Hole Punching)

(Things to know for Assignment 3b and OSPF lab)

Dr Anand Bhojan
COM3-02-49, School of Computing
banand@comp.nus.edu.sg ph: 651-67351

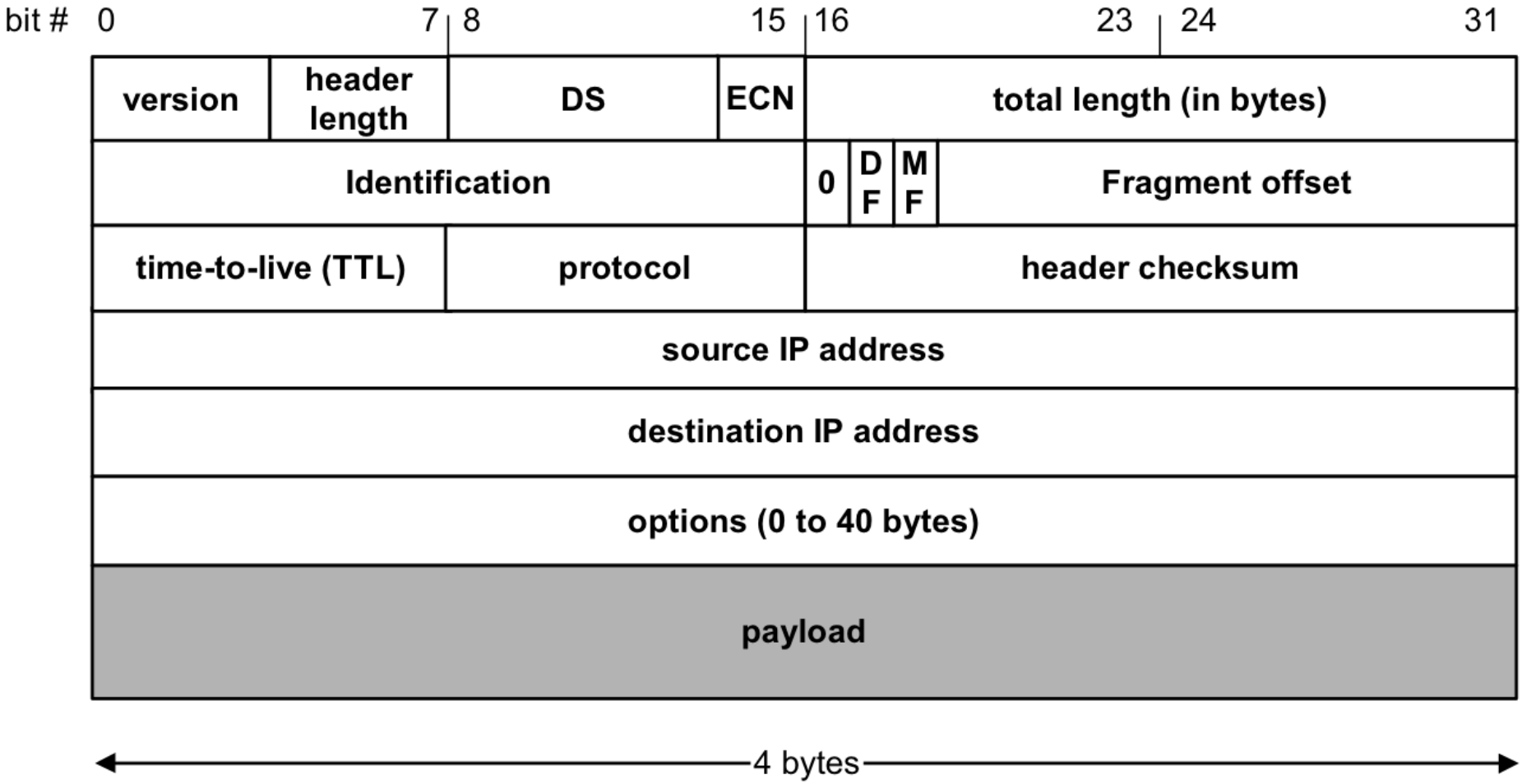
Learning Objectives

- Bridge-In (Recap) – Recap on IP Datagram Format & IP Fragmentation
- Understand IPv4 Options Processing – Record route and Source route
- Understand ICMP protocol and its applications
- Understand NAT traversal methods

Related Assessments: Assignment, Labs

FRORP – For Your Own Reading Pleasure. May not be discussed in class.

IP Datagram Format. - For your reference.



- 20 bytes (min header) ≤ Header Size < 2^4 x 4 bytes = 64 bytes
- 20 bytes (min header) ≤ Total Length < 2^{16} bytes = 65536 bytes

IP Datagram Format

FRORP

- **Question:** In which order are the bytes of an IP datagram transmitted?
- **Answer:**
  - Transmission is row by row
  - For each row:
    1. First transmit bits 0-7
    2. Then transmit bits 8-15
    3. Then transmit bits 16-23
    4. Then transmit bits 24-31
- This is called **network byte** order or **big endian** byte ordering.
- **Note:** Many computers (incl. Intel processors, Apple M1 processors) store 32-bit words in little endian format..

Big endian vs. small endian

FRORP

- Conventions to store a multi-byte word
- Example: a 4 byte Long Integer **`Byte3 Byte2 Byte1 Byte0`**

Little Endian

- Stores the low-order byte at the lowest address and the highest order byte in the highest address.

```
Base Address+0 Byte0
Base Address+1 Byte1
Base Address+2 Byte2
Base Address+3 Byte3
```

- Intel processors use this order

Big Endian

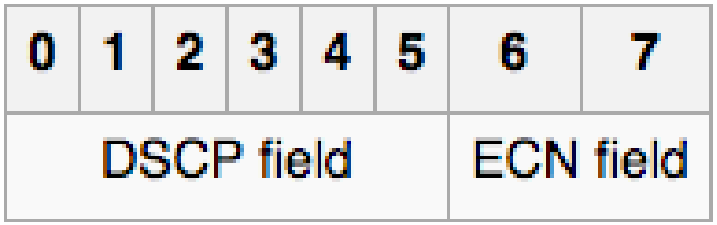
- Stores the high-order byte at the lowest address, and the low-order byte at the highest address.

```
Base Address+0 Byte3
Base Address+1 Byte2
Base Address+2 Byte1
Base Address+3 Byte0
```

LEON, Motorola processors use big endian.

Differentiated Services

- **DS/ECN field (1 byte)**
  - This field was previously called as Type-of-Service (TOS) field. The role of this field has been re-defined, but is "backwards compatible" to TOS interpretation
  - Differentiated Service (DS) (6 bits):
    - Used to specify service level (currently not supported in the Internet). Under DiffServ, **DSCP (Differentiated Services Code Point)** is defined.
  - Explicit Congestion Notification (ECN) (2 bits):
    - New feedback mechanism used by TCP (Most server support ECN today)

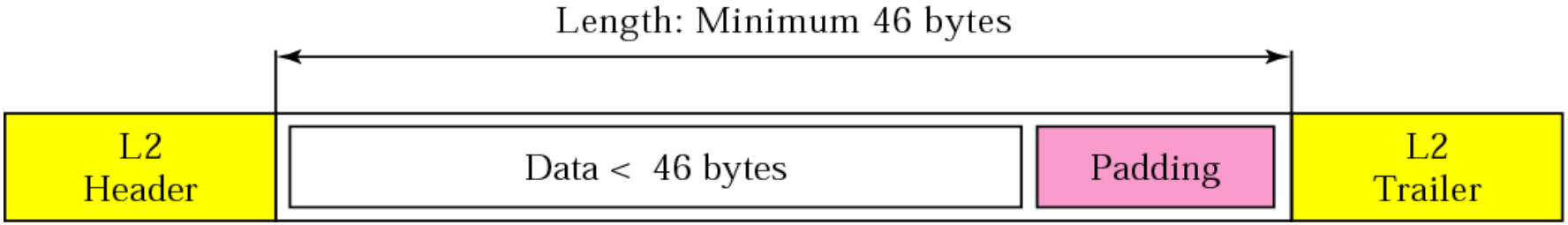


ECN

- 00 – Non ECN-Capable Transport, Non-ECT
- 10 – ECN Capable Transport, ECT(0)
- 01 – ECN Capable Transport, ECT(1)
- 11 – Congestion Encountered, CE.

Encapsulation of a small datagram in an Ethernet frame

SMALL DATAGRAM



Q: Why there is a minimum length requirement?

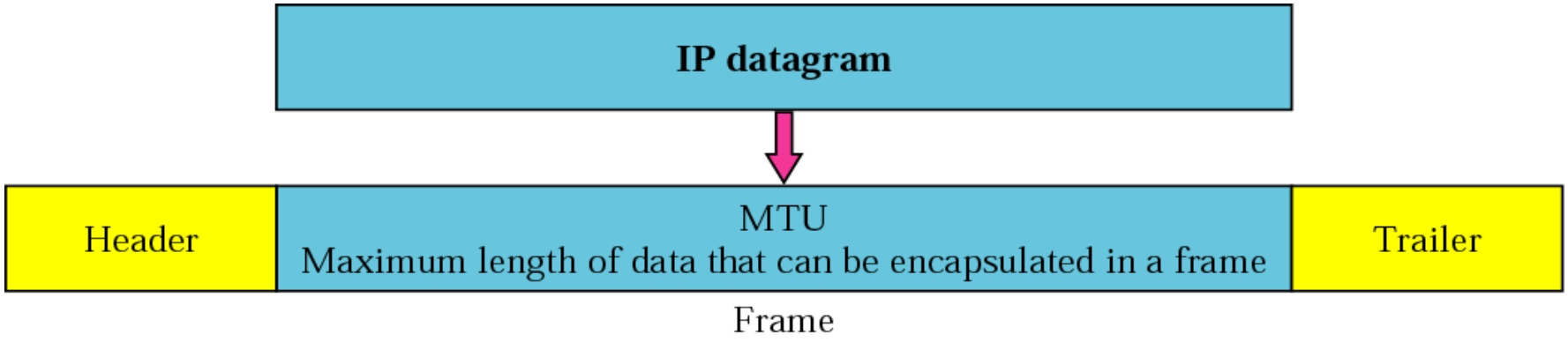
Make sure you LOGIN using your NUSNET ID.

https://pollev.com/banand

Encapsulation of a big datagram in an Ethernet frame

MTU – Maximum Transmission Unit

BIG DATAGRAM

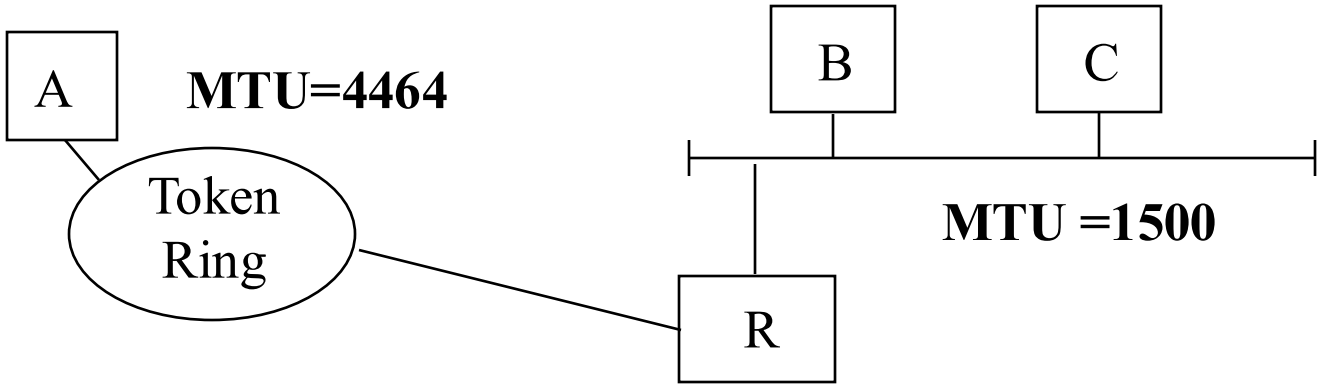


Q: Why there is a maximum length requirement for Ethernet?

Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

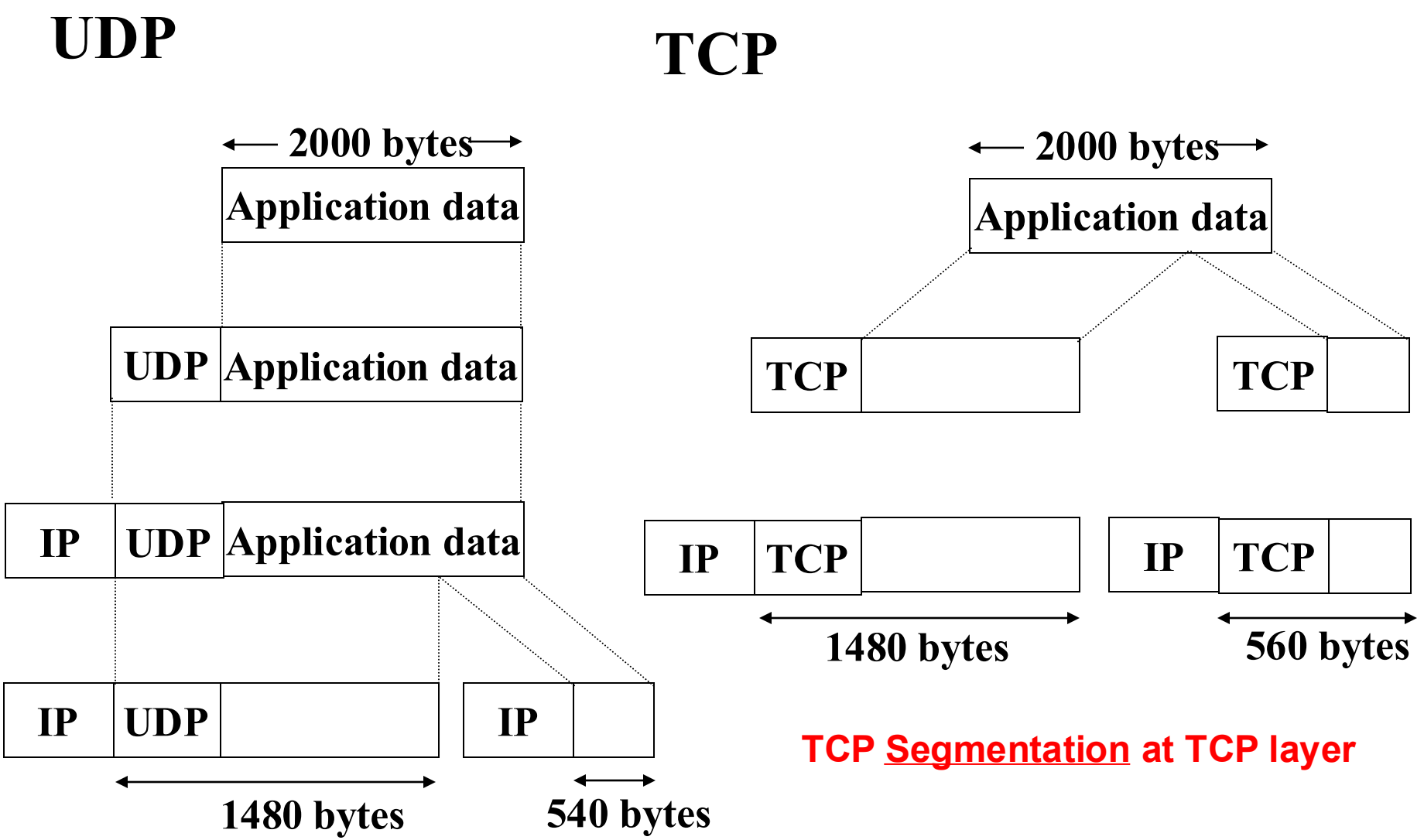
IP Fragmentation

fragmentation total length = 16 bits, offset = 13 bits the 3 bits is for 0, reserved for future use, bit 1 for dont frag, bit 2 - more frag means that there are more frag packets



- IP Fragmentation takes place because of physical level limitation - MTU.
- Fragmentation takes place at a **Router** or **Original Sending Host**.
- Fragments are reassembled at the **destination host.**

Segmentation and Fragmentation



Example

- An IP datagram with 636 bytes of data (IP payload) arrives at a router and is destined to a network with a MTU of 256 bytes. Assume 20 bytes IP header.

- IP Header (recap) - 3-bit Flag:-

D: Do not fragment
M: More fragments

Example

636 bytes of data to be transferred. MTU is 256 bytes.

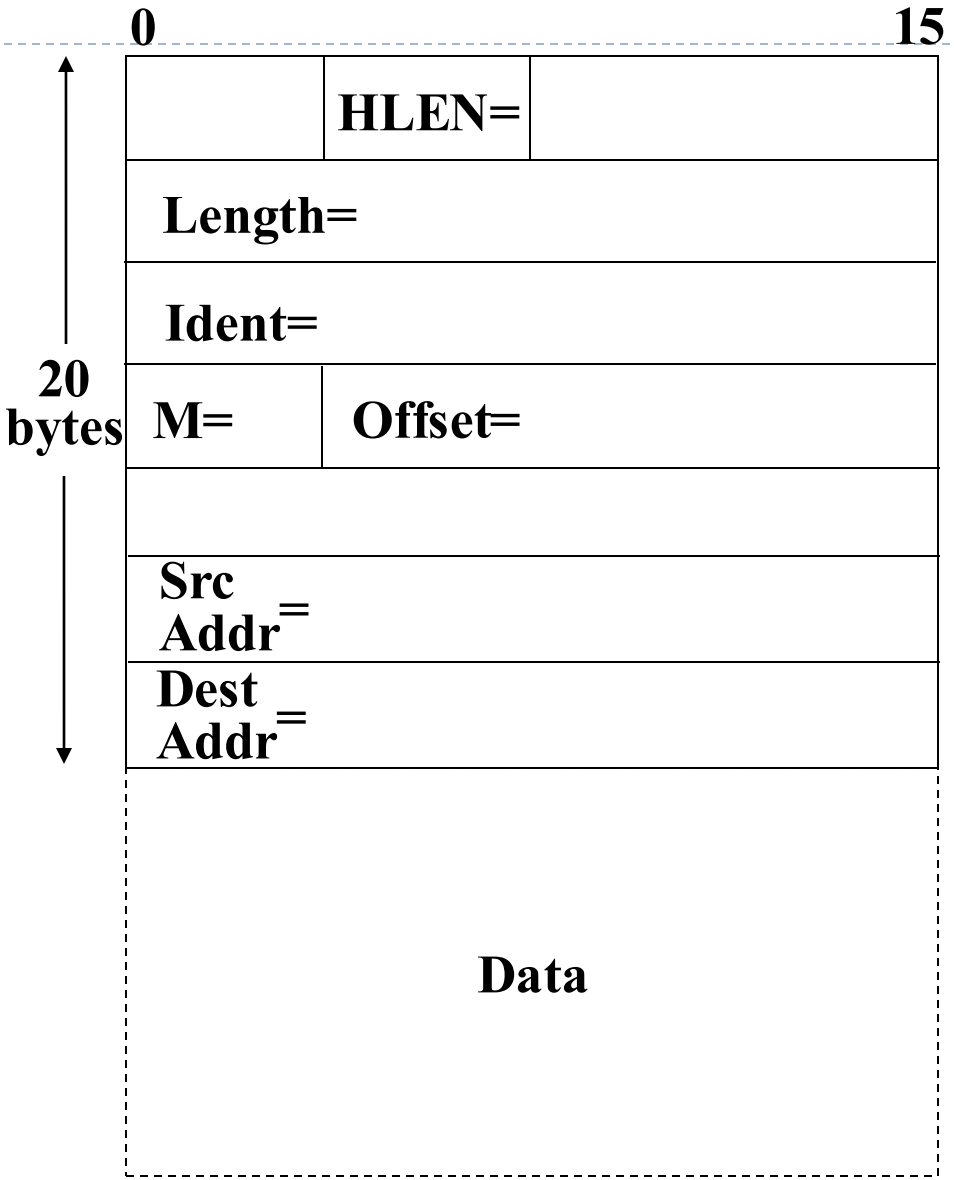
- Maximum possible data length in each fragment
  256 -20 = 236 bytes
- Min number of fragments required = 636/236 =3
- But length must be divisible by 8, except for last fragment
- Revised length of each fragment is 232, 232, 172

Question to ponder: Why the length must be divisible by 8?

Example (cont)

- First Fragment:

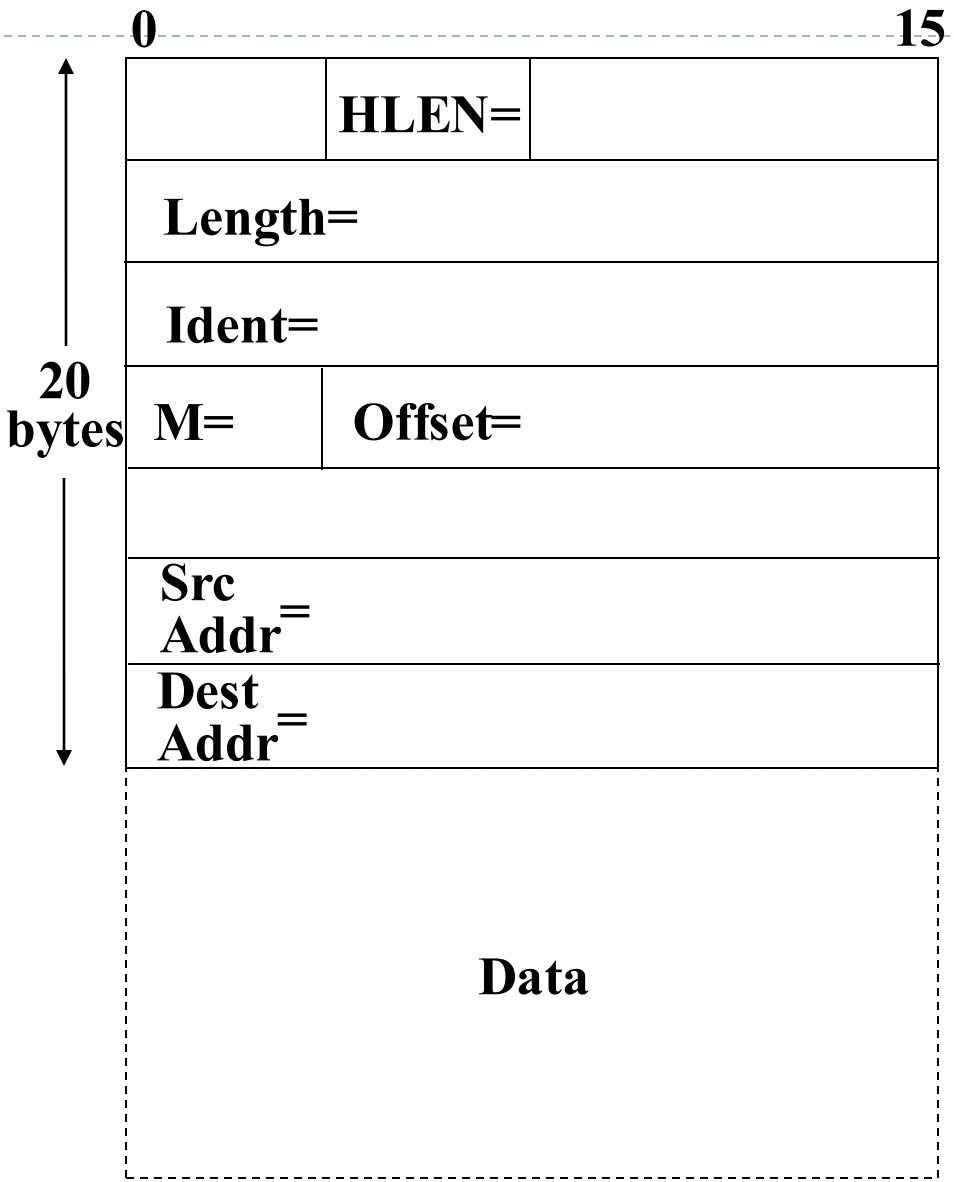
HLEN=5;
Fragment length = 232
Total Length =232+20
=252
Fragment Offset=0
More =1



Example (cont)

- Second Fragment:

HLEN=5;
Fragment length = 232
Total Length=252
Fragment Offset=29
More=1



Question to ponder: Why fragment offset is 29 instead of 232?
Why can't we have it as 232? save space?

Example (cont)

- Third Fragment

HLEN=5;

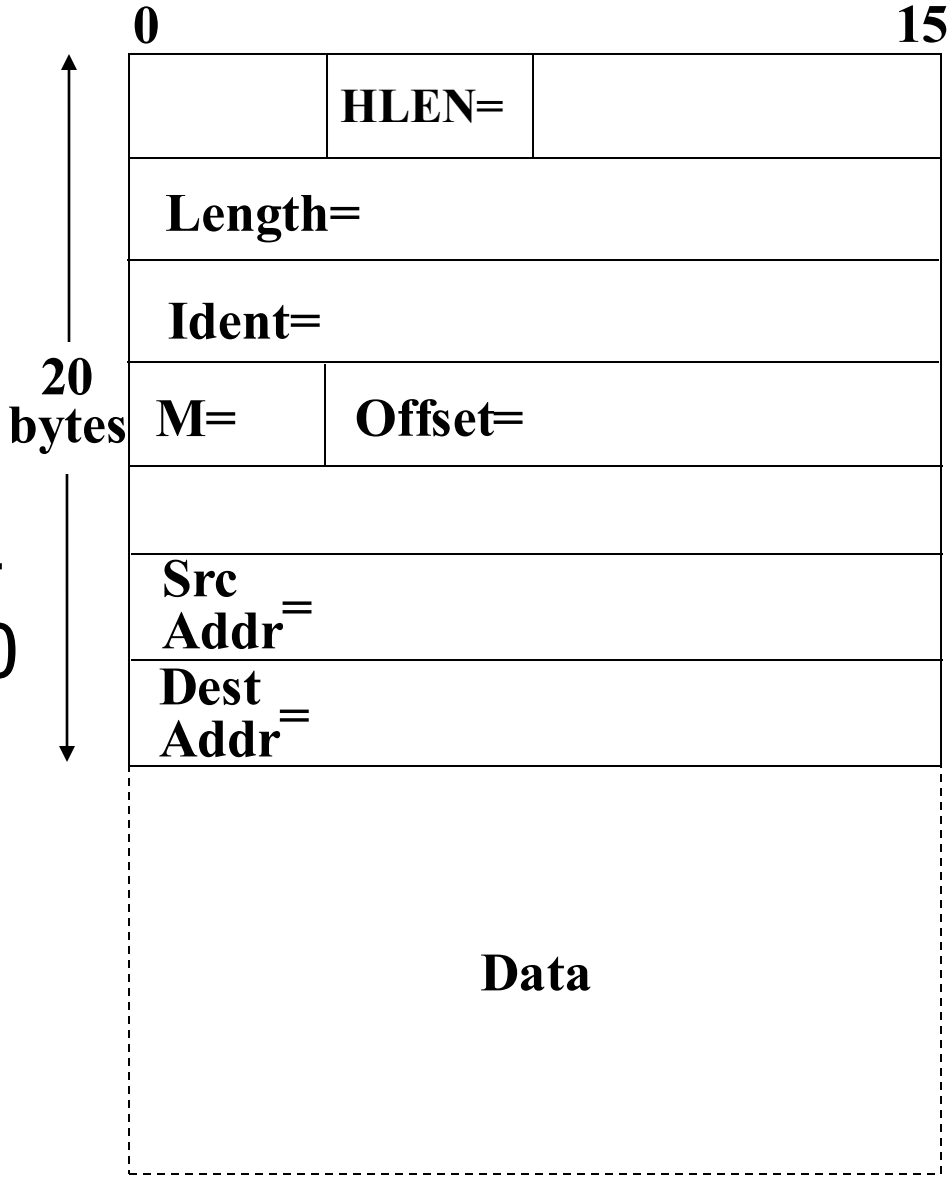
Fragment length=172

Total Length =172+20

=192

Fragment offset=58

More=0



EXAMPLES

1) A packet has arrived with an M bit value of 1 and a fragmentation offset value of zero. Is this the first fragment, the last fragment, or a middle fragment?.

2) A packet has arrived in which the offset value is 100, the value of HLEN is 5 and the value of the total length field is 100. What is the number of the first byte and the last byte of the IP payload?

FRORP

IP-options, ICMP, NAT-Hole Punching

IPv4 – Options Processing

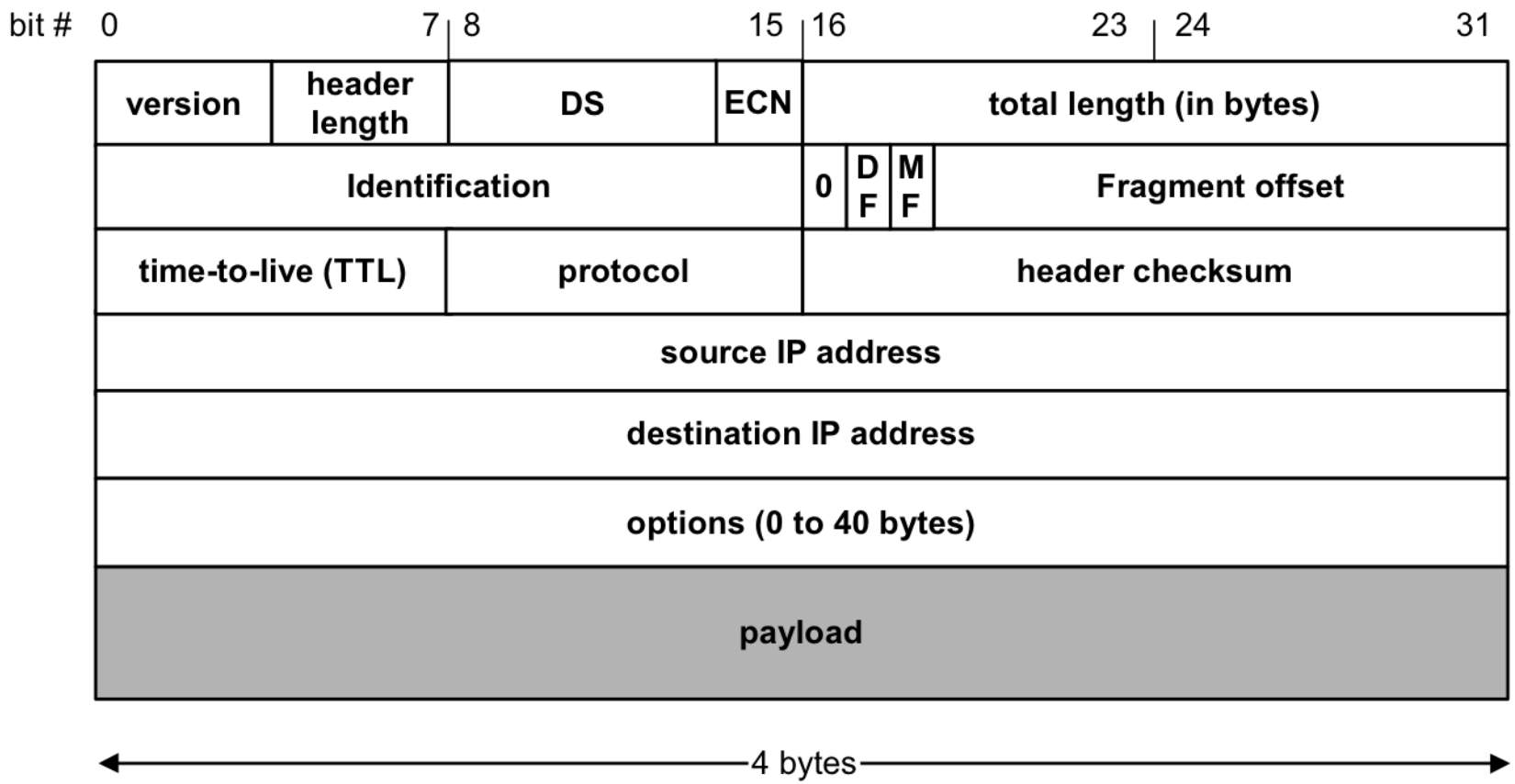
Internet Control Message Protocol (ICMP)

Ping and Traceroute

NAT Traversal (Port Forwarding and Hole Punching)

Dr Anand Bhojan
COM3-02-49, School of Computing
banand@comp.nus.edu.sg *ph: 651-67351*

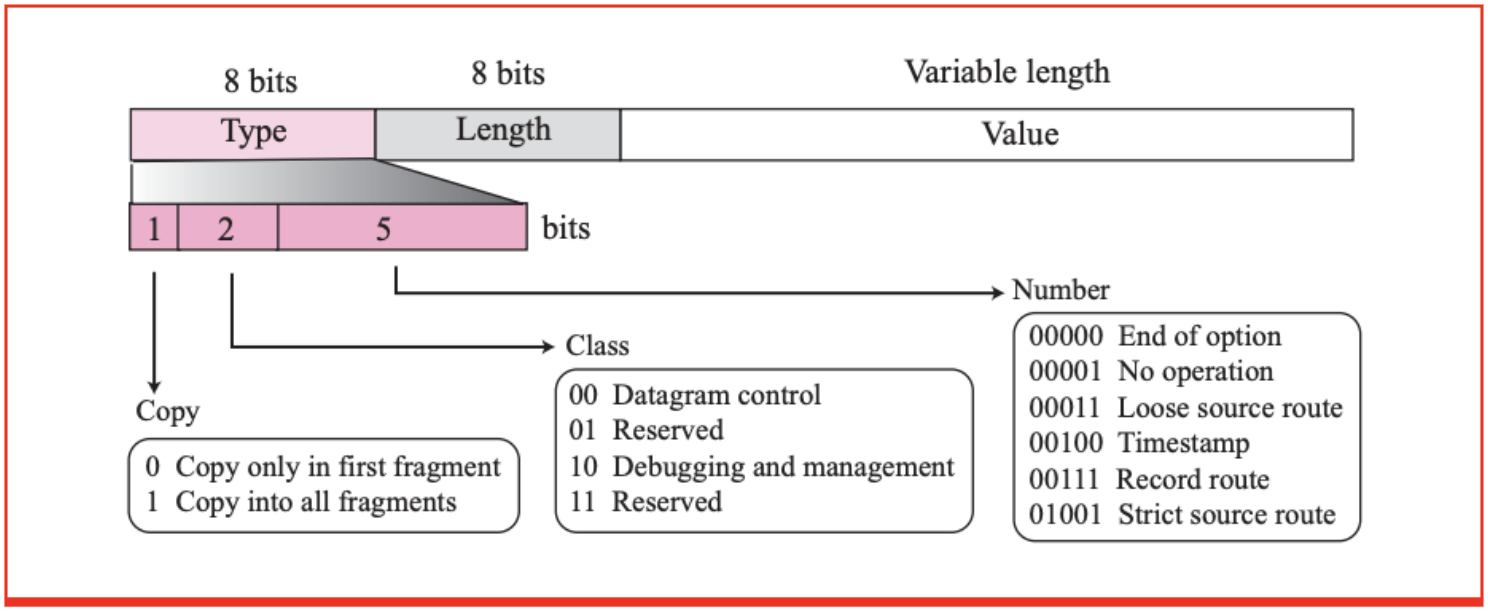
OPTIONS



Q: Given header-length is only 4 bits, how the header size is 60 bytes?

Q: Given, these 40 bytes are optional, do every routers and end-hosts need to process the options?

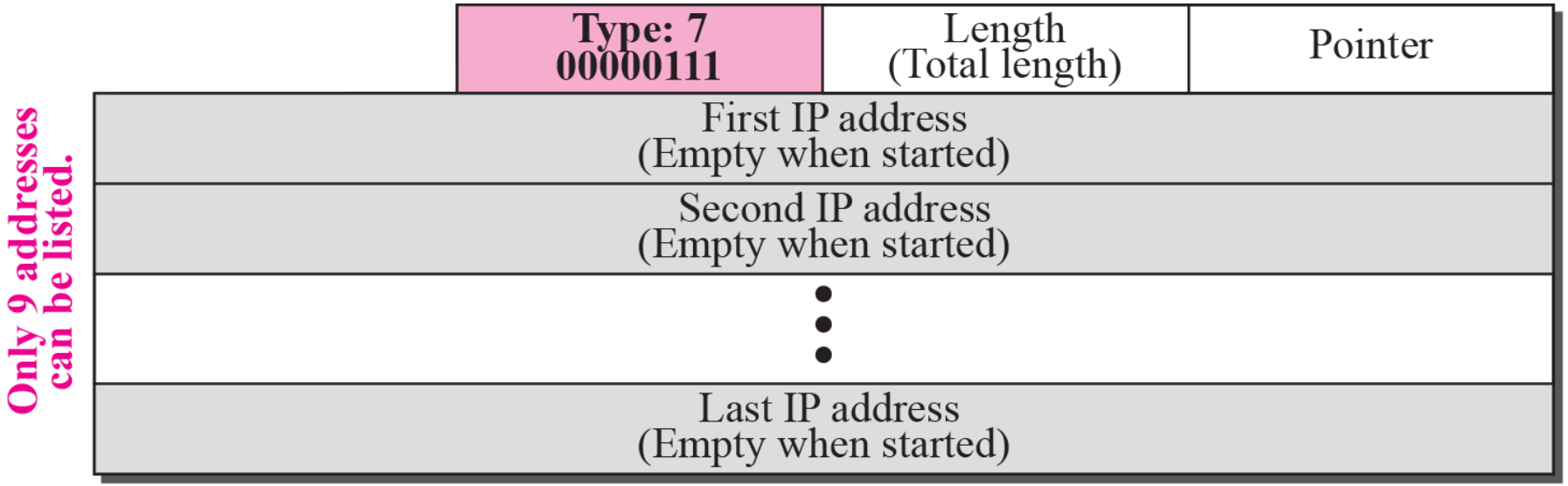
Option format



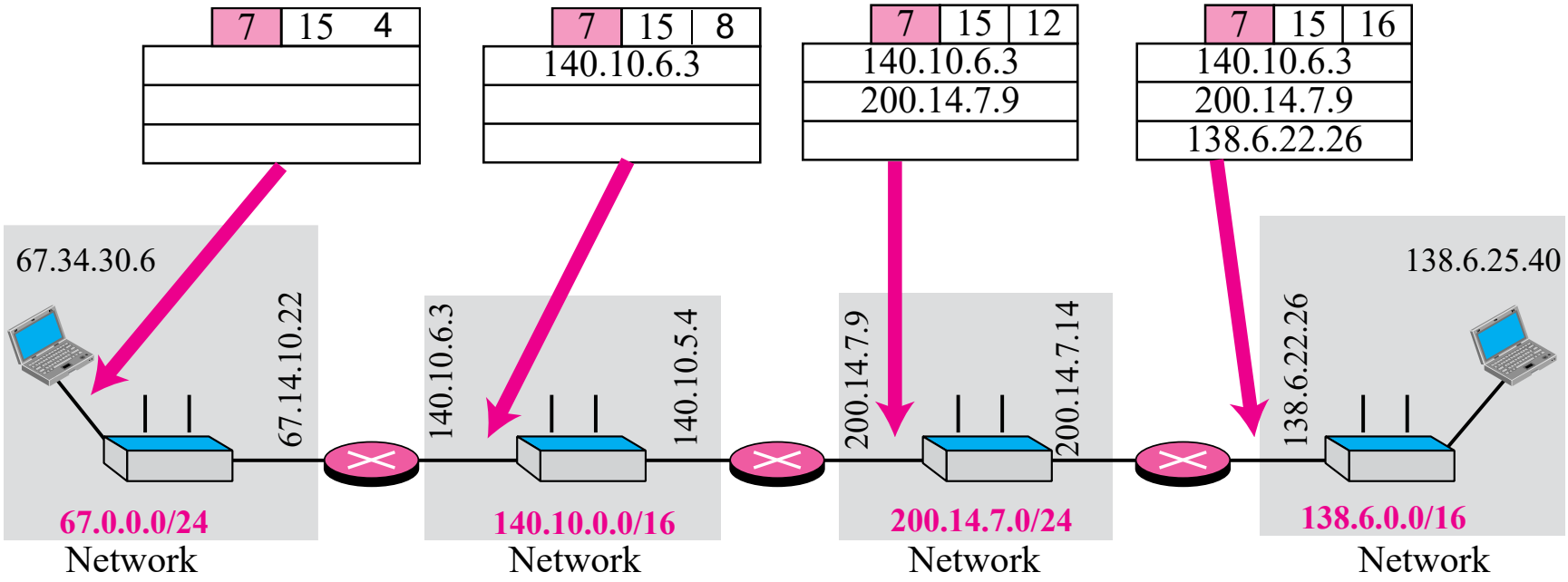
- First bit – 0 (Copy the option only in first fragment); 1 (Copy in all fragments)
- Second and Third Bits – General purpose of the option (option category). 00 (Datagram Control), 01 (Reserved), 10 (Debugging and Management), 11(Reserved)
- Remaining 5 bits gives 32 options. We have only 6 options defined so far.

[Length and Value are optional]
Length is total length including the Type, and Length field itself.

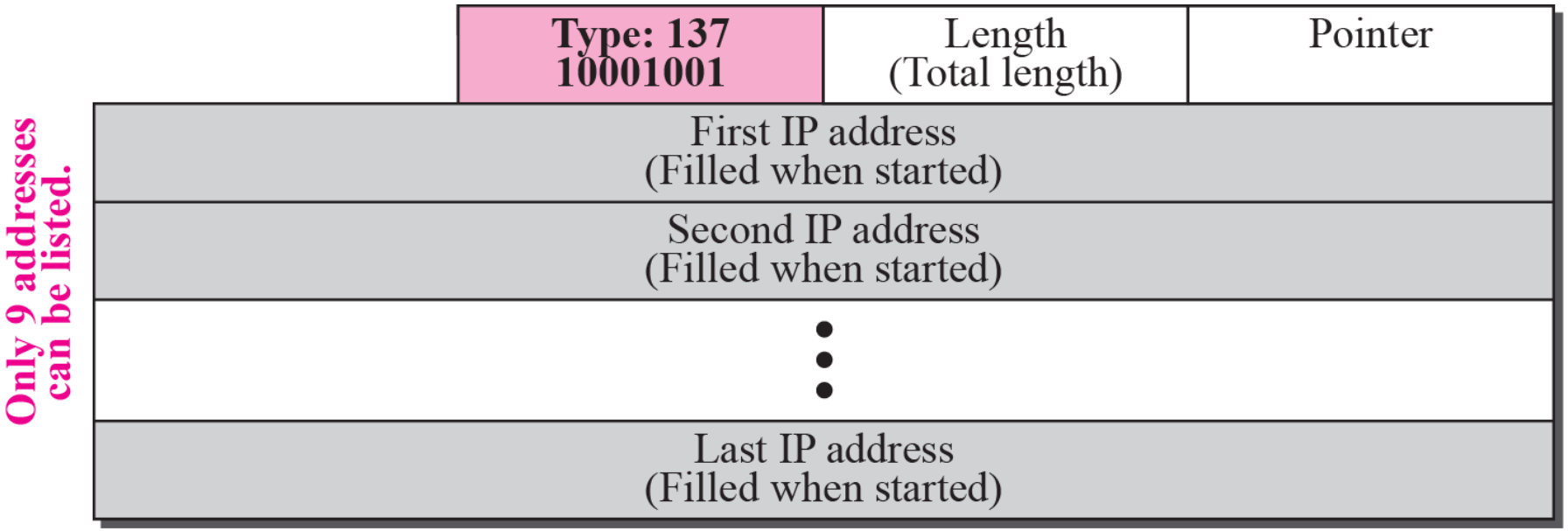
Record-route option



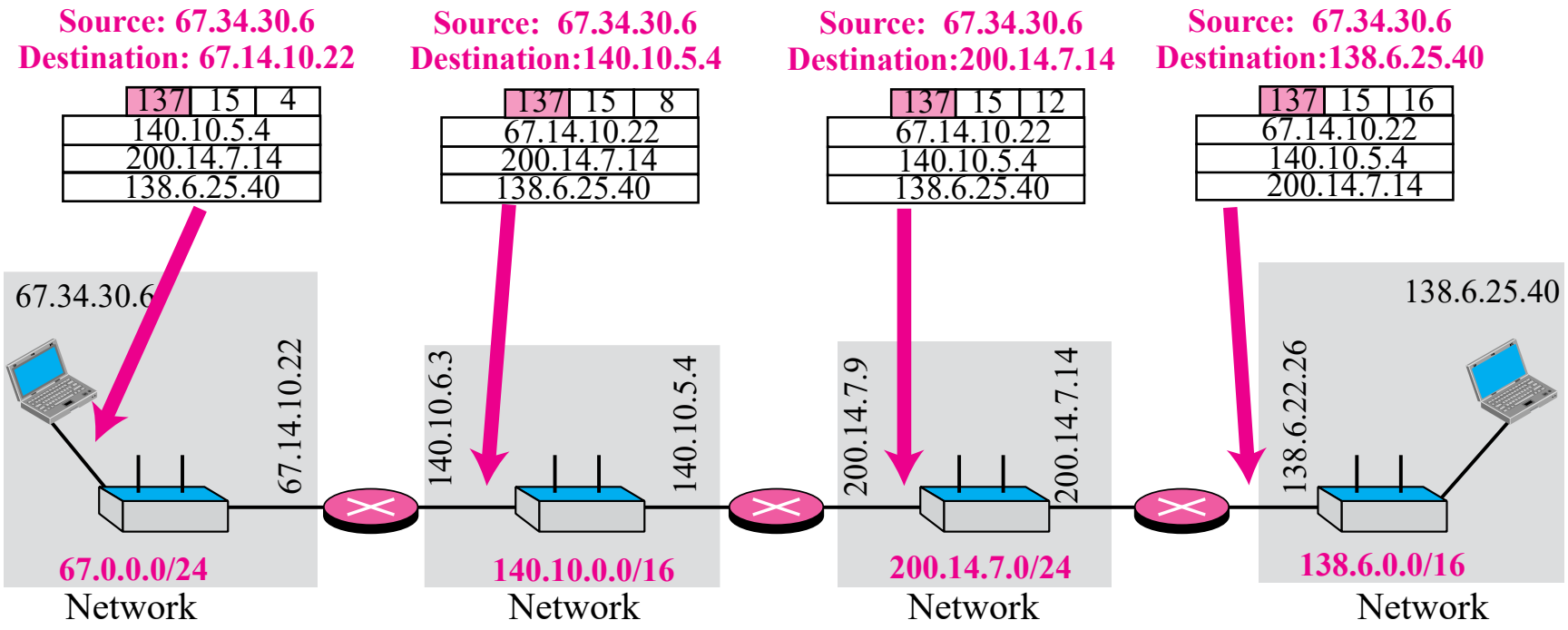
Record-route concept



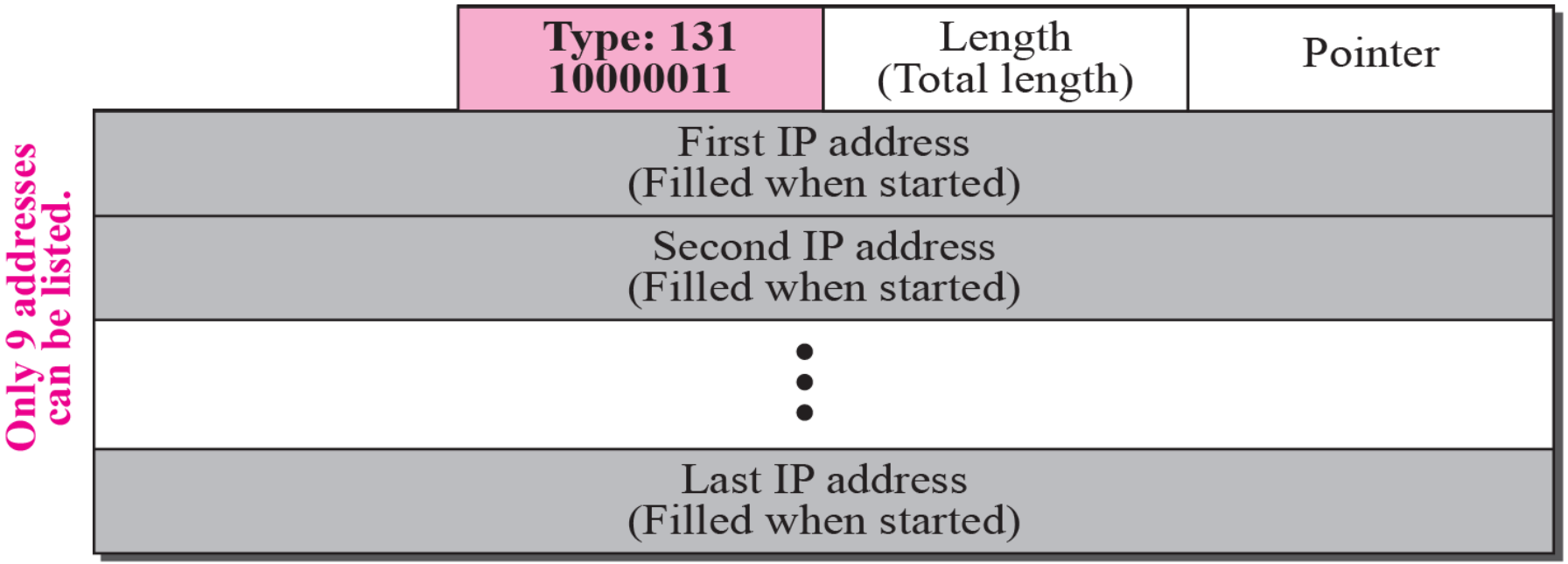
Strict-source-route option



Strict-source-route option

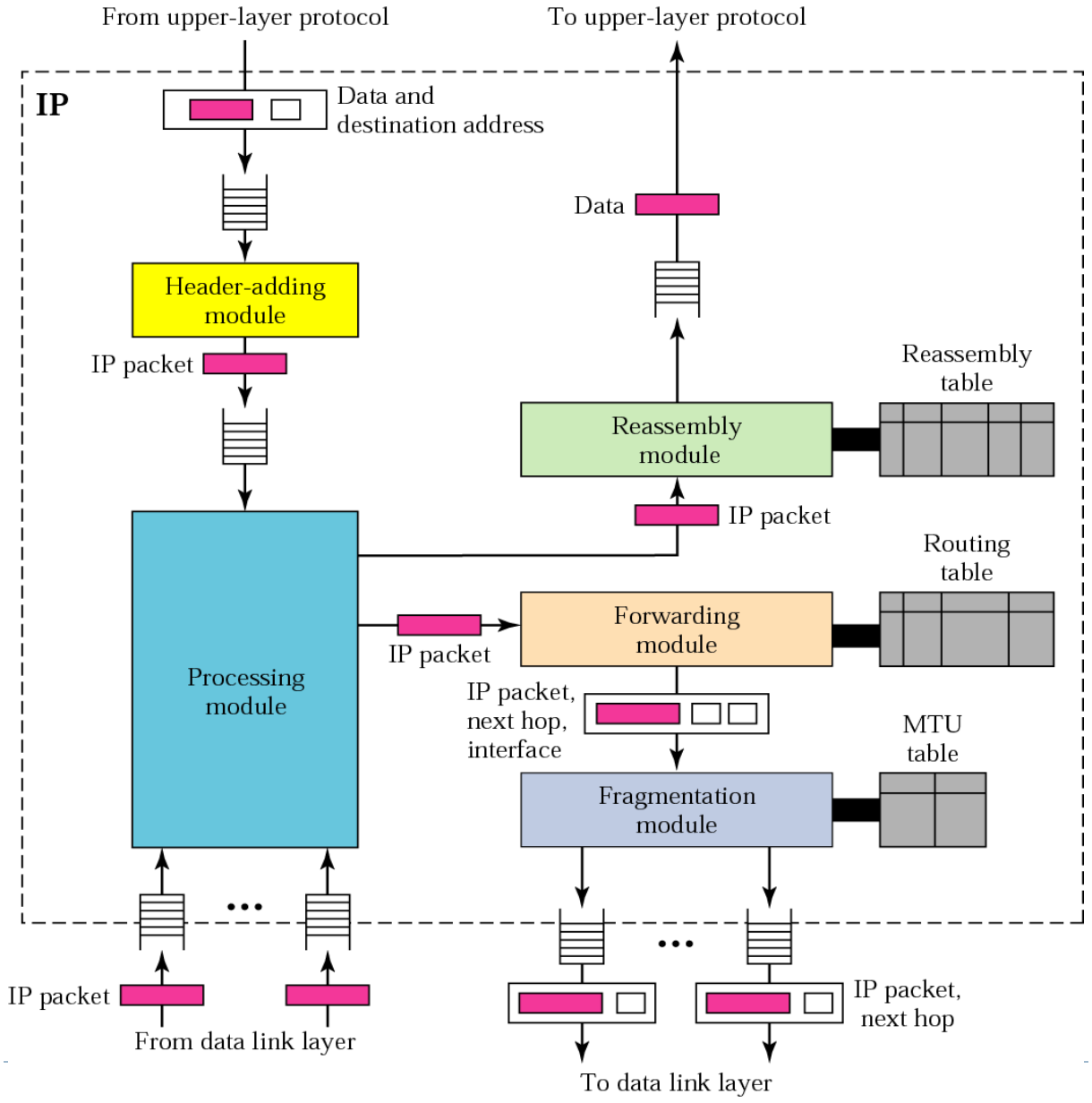


Loose-source-route option



IP components

FYORP



IP-options, ICMP, NAT-Hole Punching

IPv4 – Options Processing

Internet Control Message Protocol (ICMP)

Ping and Traceroute

NAT Traversal (Port Forwarding and Hole Punching)

Dr Anand Bhojan
COM3-02-49, School of Computing
banand@comp.nus.edu.sg ph: 651-67351

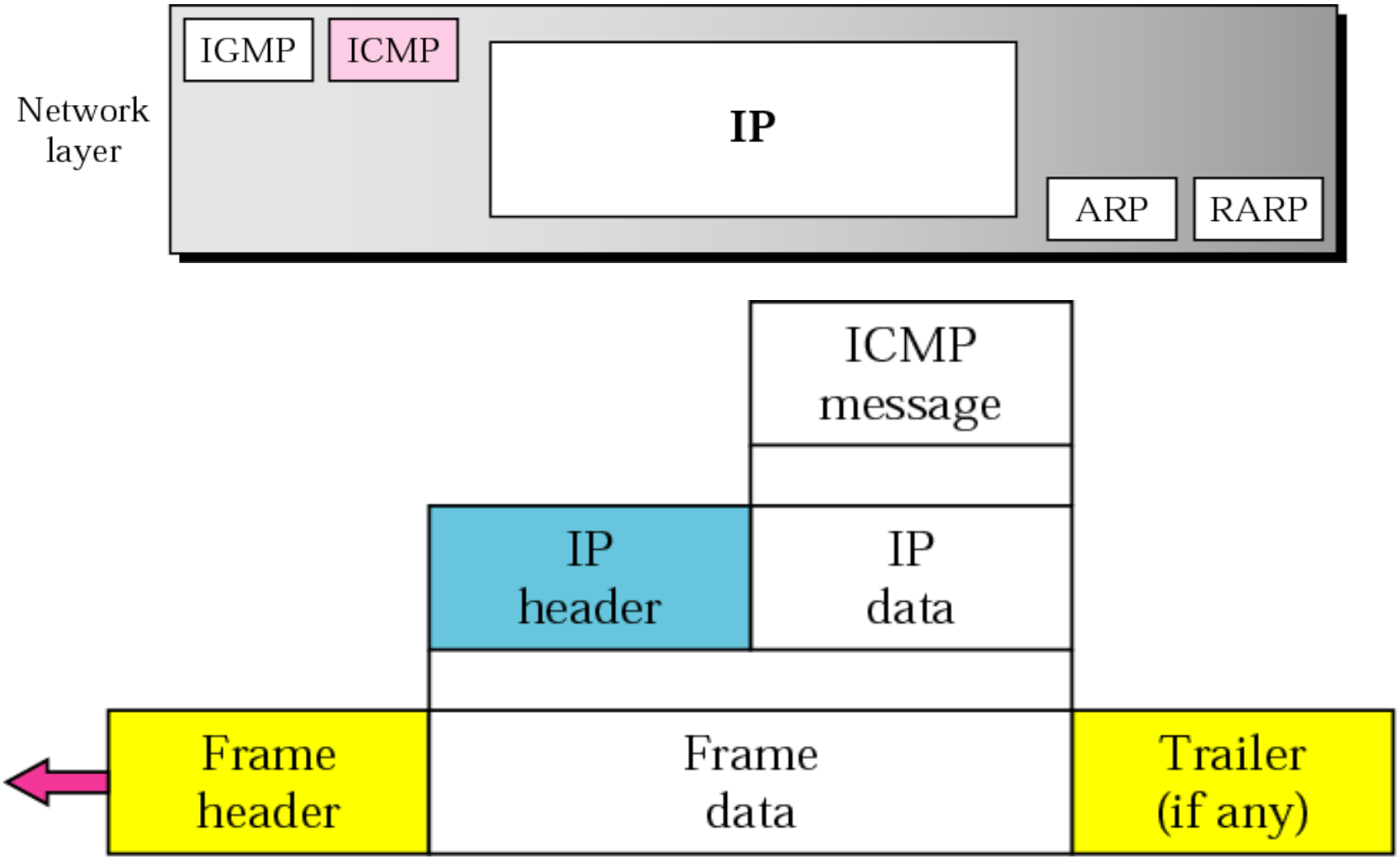
Why ICMP?

- *Q:What if a router cannot route or deliver a datagram?*
- *Q:What if a router experiences a congestion?*
- *Q:What if the TTL expires?*

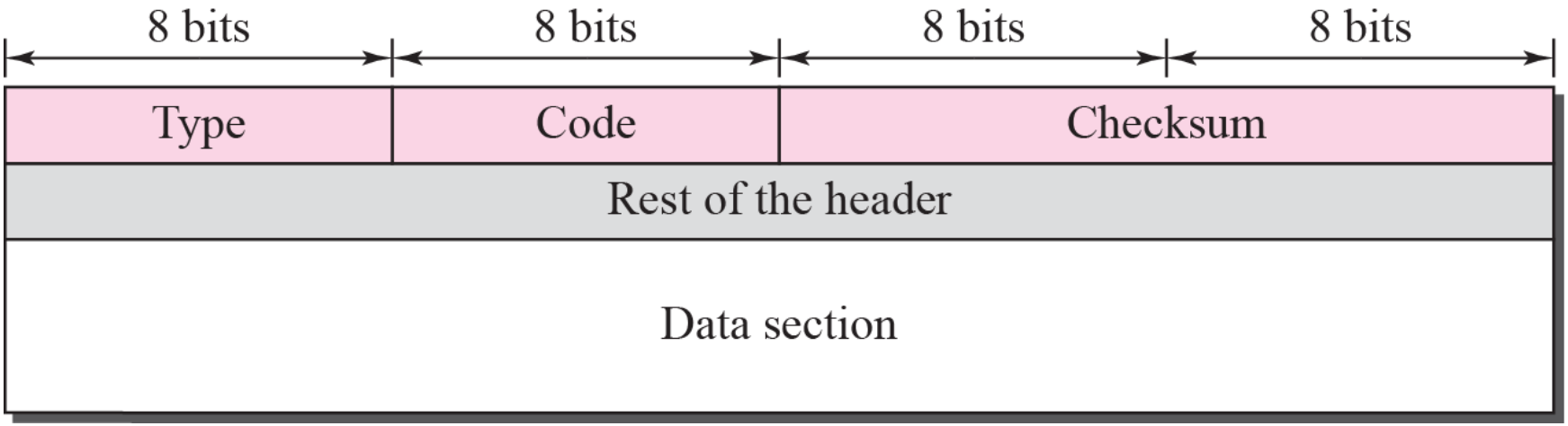
…

  - Router needs to inform source to take action to avoid or correct the problem
  - ICMP is an error reporting mechanism, and can only report error condition **back to the original source**
  - In addition, ICMP allows routers and hosts to send **information or control messages** to other routers or hosts.
  - ICMP is specified in **RFC 792**

Position of ICMP in the network layer & Encapsulation



General format of ICMP messages



- 8 byte header, variable size data section
- format for first 4 bytes of header is common to all ICMP packets
- Type – ICMP message type
- Code – reason for the message type generated

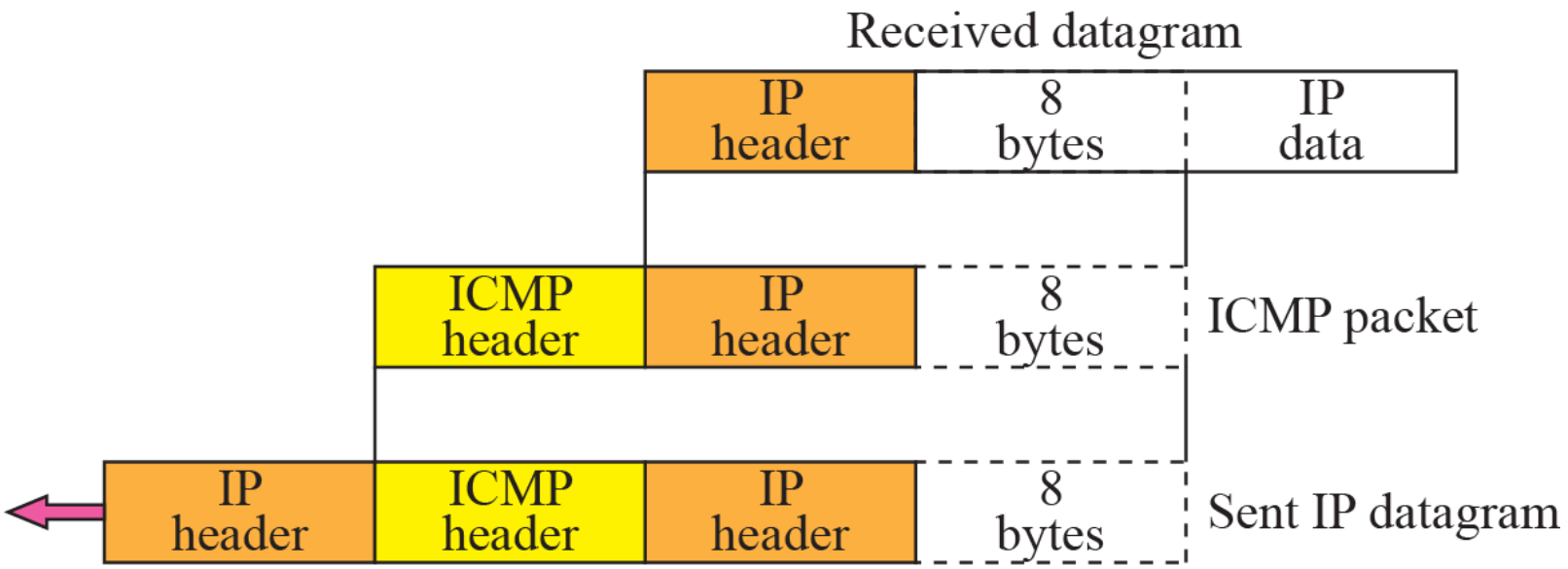
ICMP MESSAGES

Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.

The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host.

Contents of data field for the error message



Q: Why the IP header of the received datagram is included in the Error message?

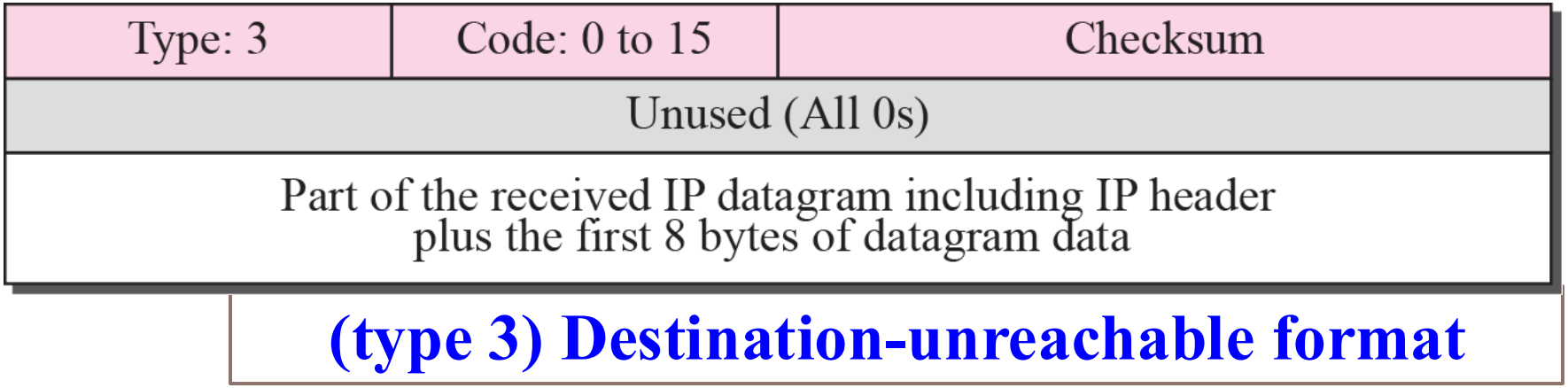
Q: Why the first 8-bytes of the received datagram is included in the Reply?

ERROR MESSAGES

ICMP error messages are used by routers and hosts to tell a device that sent a datagram about problems encountered in delivering it.

- Q1: What if a datagram carrying ICMP error message causes another error?
- Q2: Do we need ICMP error message for each fragment of a fragmented datagram that causes the error?
- Q3: ICMP error messages will not be generated for a datagram having a multicast or broadcast address in the destination address field, Why?
- Q4: ICMP error messages will not be for a datagram whose source address is not a single host, 0.0.0.0, 127.x.x.x, broadcast or multicast address, Why?

ICMP error messages - Examples

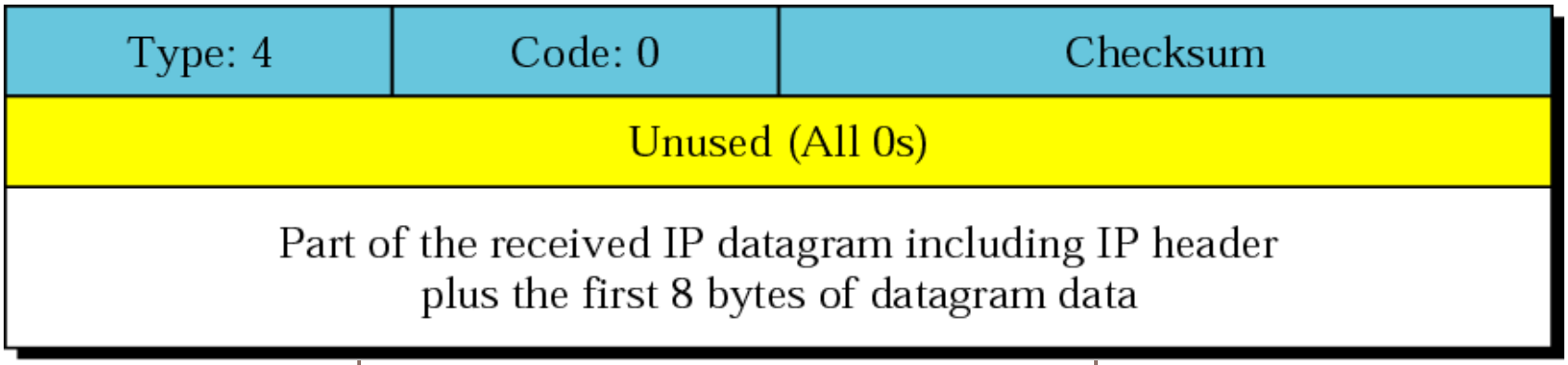


(type 3) Destination-unreachable format

Codes:
0: network unreachable
1: host unreachable
2: protocol unreachable
3: port unreachable
4: need fragmentation but DF flag is set
…….

Q: Who generates these ICMP error messages, hosts or routers?

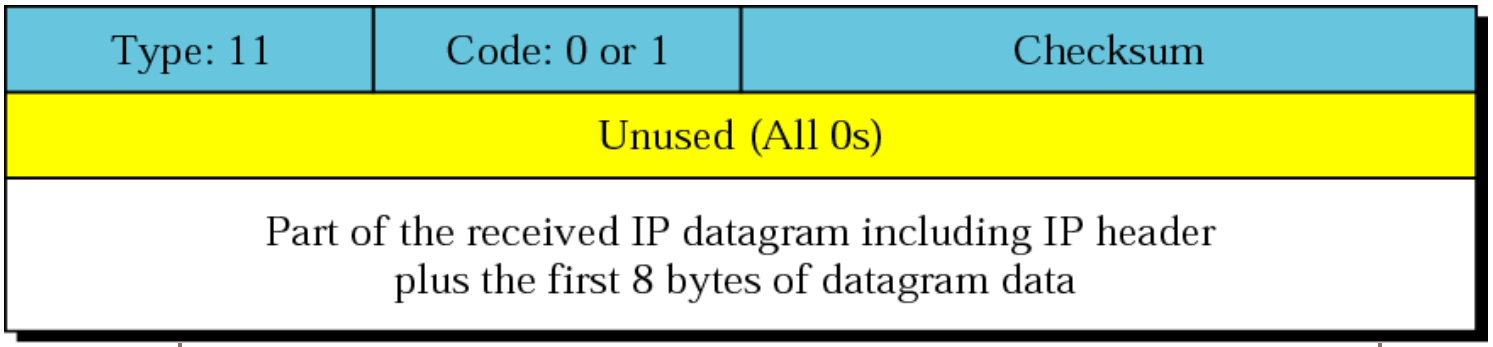
ICMP error messages - Examples



Source-quench format

- ***A source-quench message informs the source that a datagram has been discarded due to congestion in a router or the destination host.***
- ***The source must slow down the sending of datagrams until the congestion is relieved.***
- ***One message sent for each datagram discarded.***

ICMP error messages - Examples



Time-exceeded message format

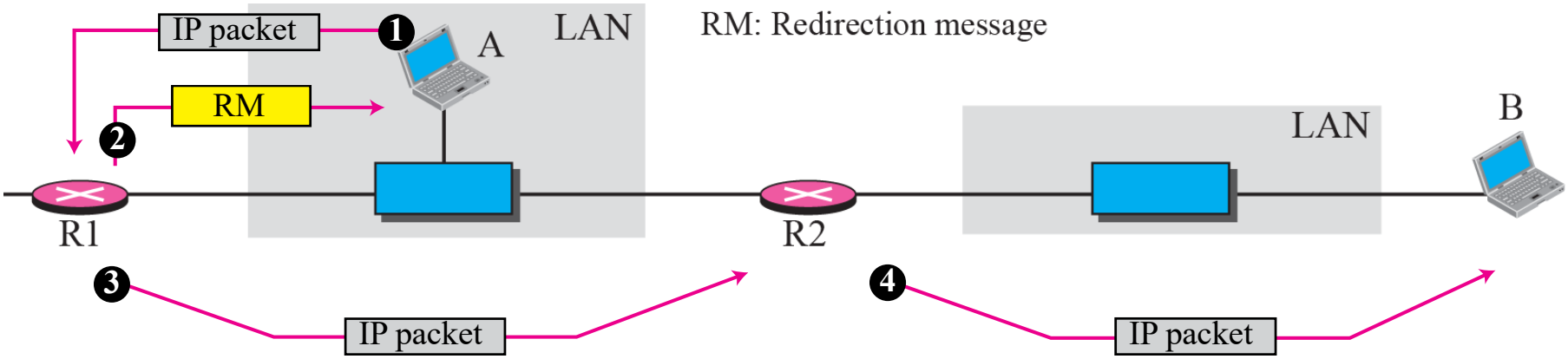
Two possible cases:

- ***Whenever a router decrements a datagram with a time-to-live value to zero, it discards the datagram and sends a time-exceeded message to the original source.***
- ***When the final destination does not receive all of the fragments in a set time, it discards the received fragments and sends a time-exceeded message to the original source.***

ICMP error messages - Examples

A host usually starts with a small (statically defined) routing table that is gradually augmented and updated. One of the tools to accomplish this is the ICMP redirection message.

ICMP error messages - Examples



Redirection concept

Type: 5	Code: 0 to 3	Checksum
IP address of the target router		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

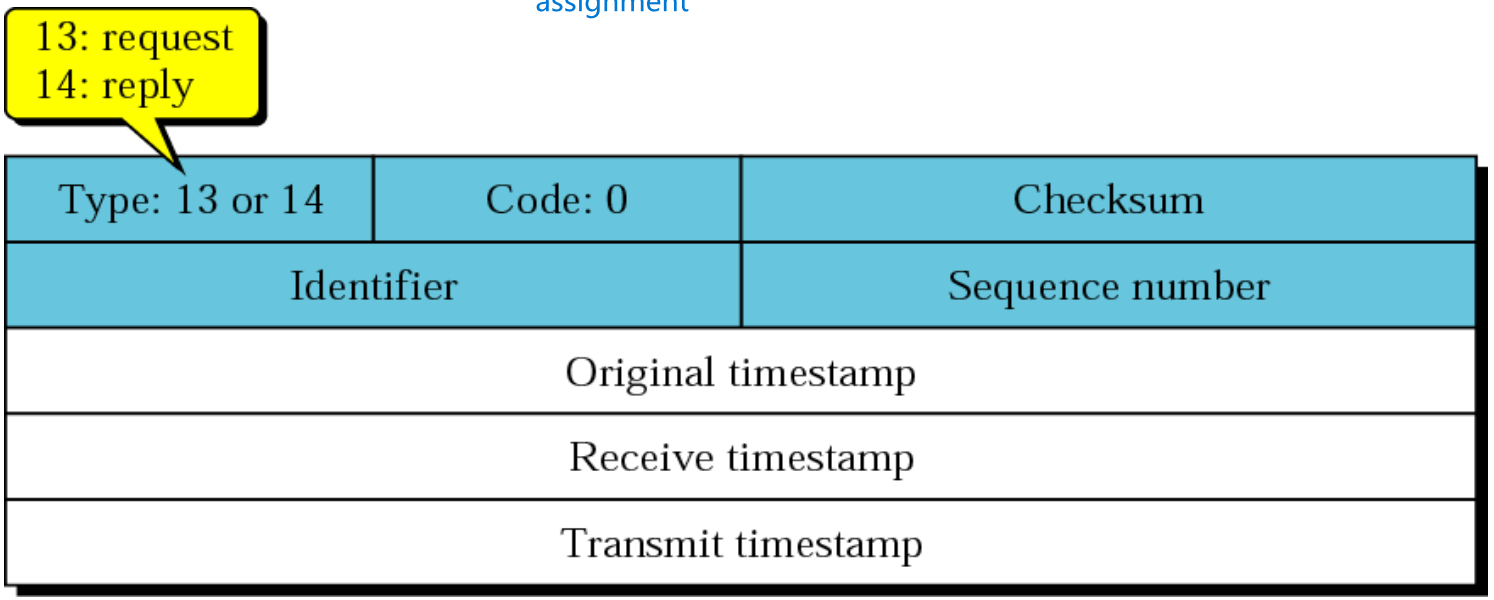
QUERY

ICMP can also diagnose some network problems through the query messages, a group of four different pairs of messages. In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node.

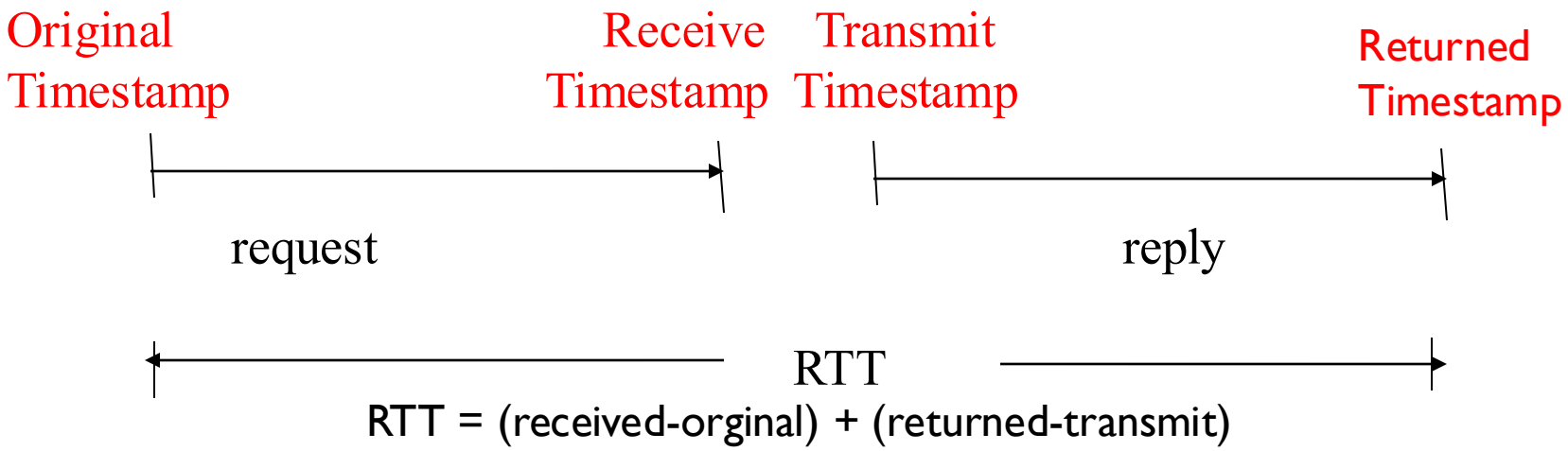
ICMP query messages - Examples

The timestamp-request and timestamp-reply messages can be used to synchronize two clocks in two machines if the exact one-way time duration is known.

assignment

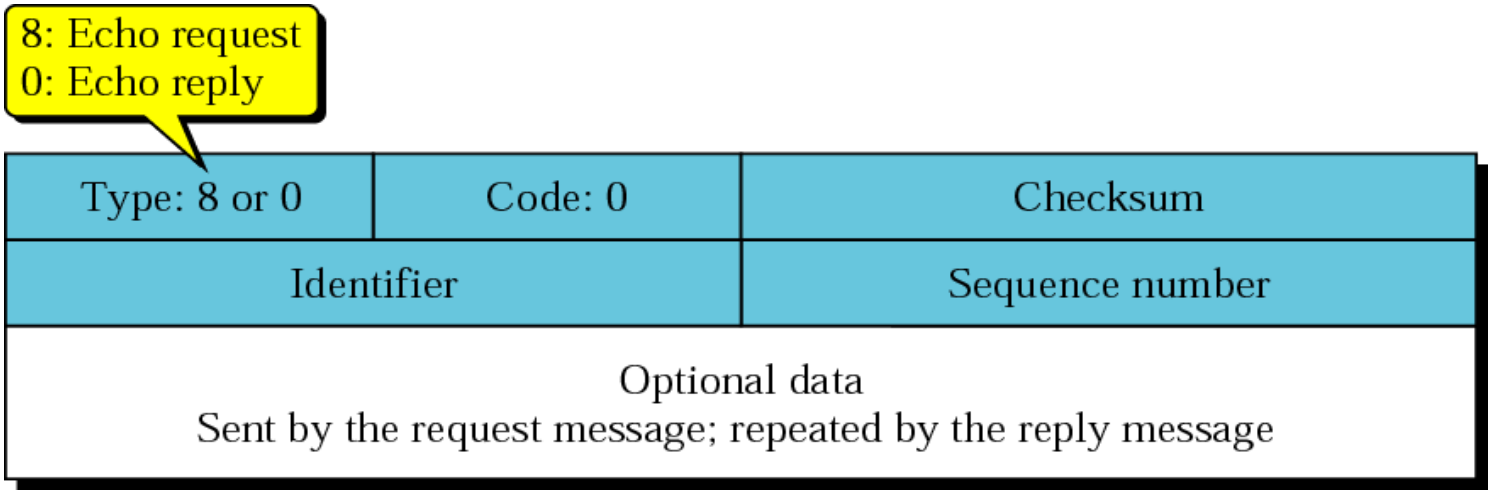


Timestamp-request and timestamp-reply message format



Q: If the two clocks are NOT synchronised, will the RTT be correct?

Echo-request and echo-reply messages



Echo-request and echo-reply messages can test the reachability of a host. This is usually done by invoking the ping command.

EXAMPLE 1

We use the ping program to test the server fhda.edu. The result is shown below:

```
$ ping fhda.edu
PING fhda.edu (153.18.8.1) 56 (84) bytes of data.
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=0 ttl=62 time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=1 ttl=62 time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2 ttl=62 time=1.90 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=3 ttl=62 time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=4 ttl=62 time=1.93 ms
```

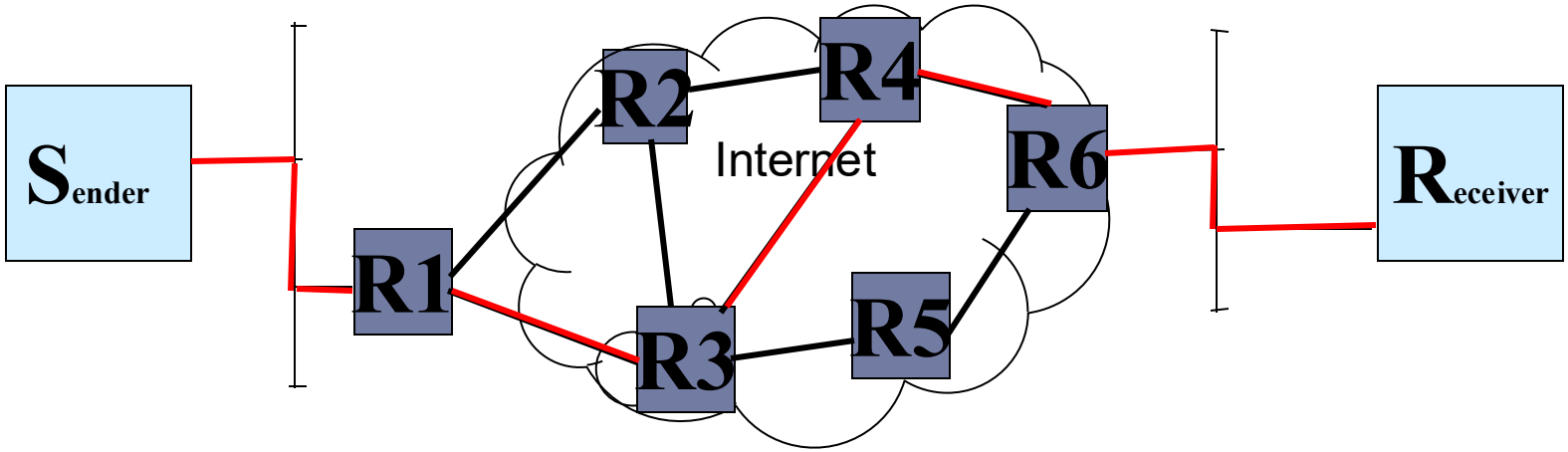
See Next Slide

EXAMPLE (CONTINUED)

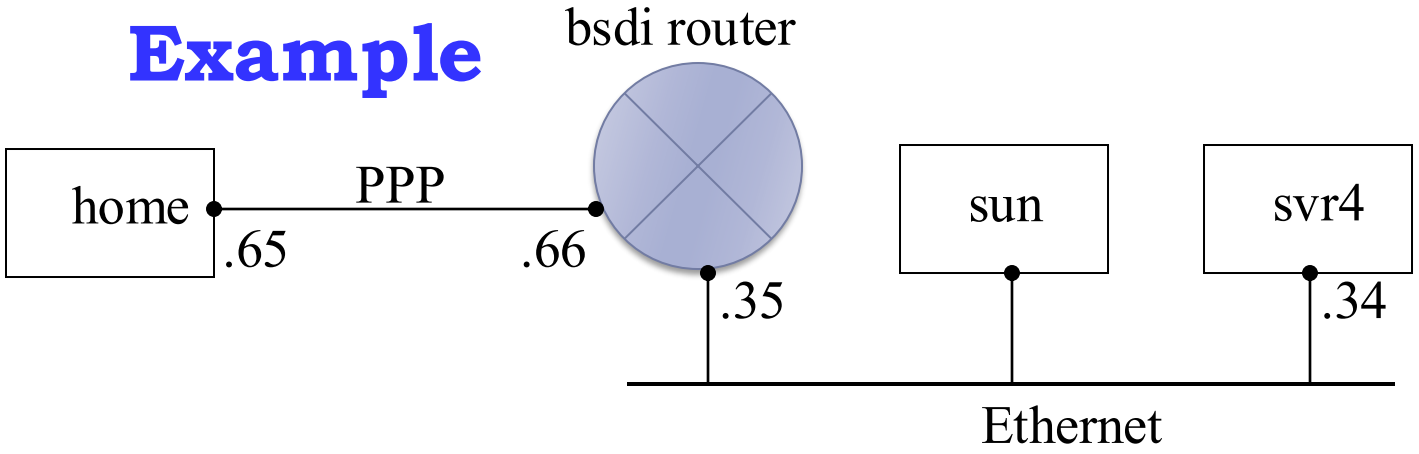
```
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=5 ttl=62 time=2.00 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=6 ttl=62 time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=7 ttl=62 time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=8 ttl=62 time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=9 ttl=62 time=1.89 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=10 ttl=62 time=1.98 ms

--- fhda.edu ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10103ms

rtt min/avg/max = 1.899/1.955/2.041 ms
```

Traceroute Program *(cont)*

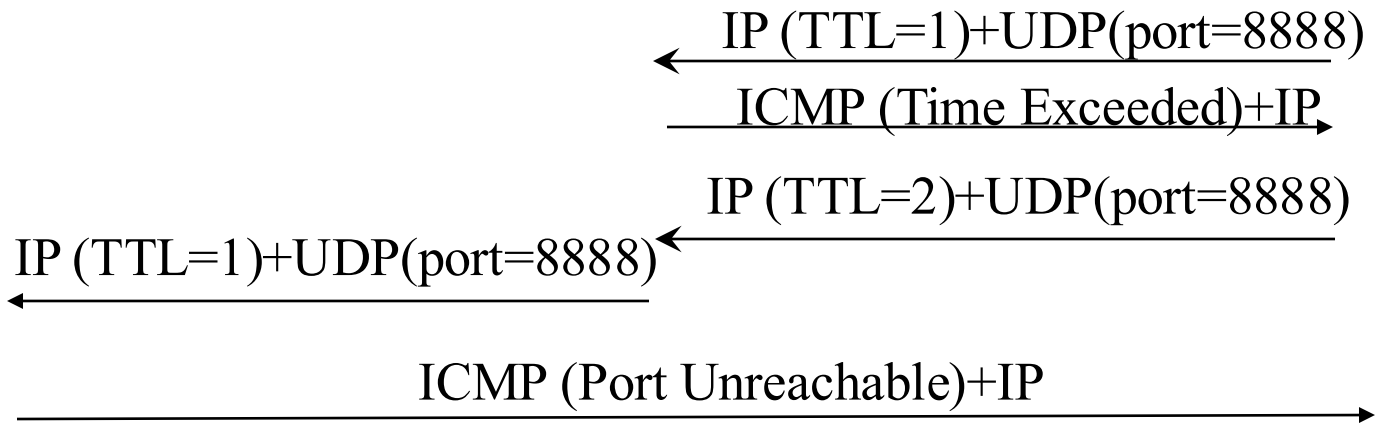
Allows us to see the route that IP datagrams follow from one host to another.



Example



assignment



svr4 % *traceroute home*

traceroute to home (140.252.13.65), 30 hops max, 40 byte packets

1	bsdi	(140.252.13.35)	20 ms	10 ms	10 ms
2	home	(140.252.13.65)	120 ms	120 ms	120 ms

Traceroute Operation

- Sends a UDP datagram to destination with TTL field in IP header set to value 1
  - datagram = 40bytes
    (8 bytes UDP + 20 bytes IP + 12 bytes data)
  - 12 bytes data = seq no, time sent
- Causes router to generate **"time exceeded"** ICMP error
- Increment TTL progressively until final destination is reached
- How to terminate?
  - UDP port chosen is a non-existence port
  - Destination sends ICMP **"port unreachable"** error
- For each TTL value, 3 datagrams (probes) are sent. WHY?

Q: Some firewalls disable UDP messages. Is there any alternate method?
Q: Is there any difference between the route info shown by Traceroute and IP-record route?

IP-options, ICMP, NAT-Hole Punching

IPv4 – Options Processing

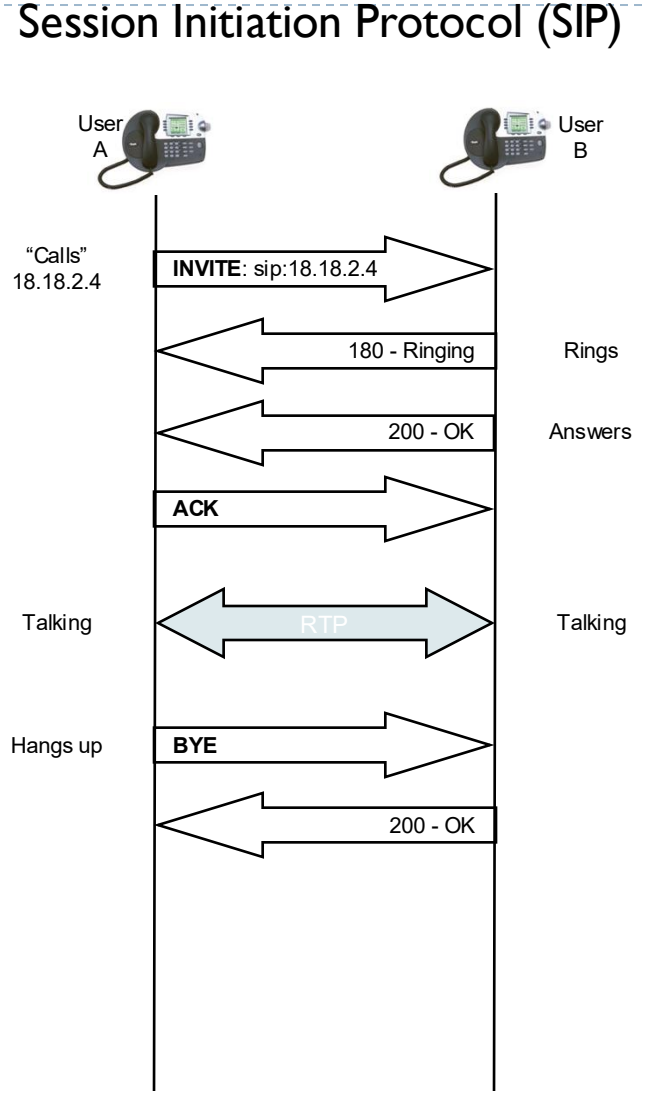
Internet Control Message Protocol (ICMP)

Ping and Traceroute

NAT Traversal (Port Forwarding and Hole Punching)

Dr Anand Bhojan
COM3-02-49, School of Computing
banand@comp.nus.edu.sg ph: 651-67351

A Simple SIP Flow



What if one or more peers are behind NAT (using private IP)?

NAT Traversal

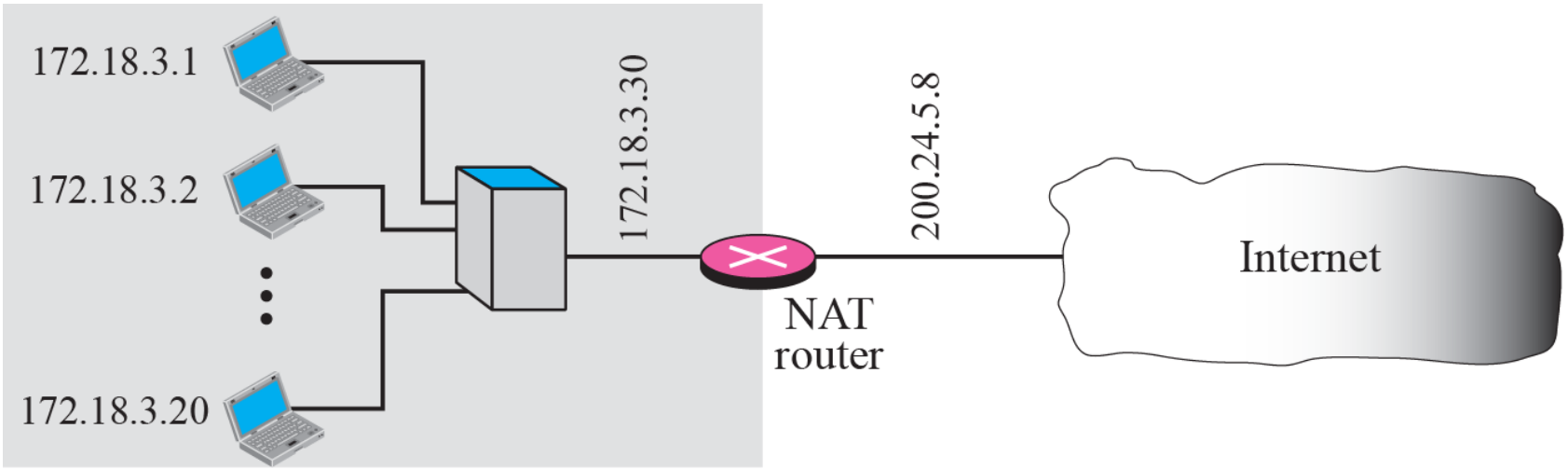
NAT traversal & why do we need it?

Network Address Translation (NAT)

- Remember NAT?
  - Private IP addresses
  - NAT gateway (usually on a gateway router)
    - **Translates** between internal addresses/ports & external ones

Network Address Translation (NAT)

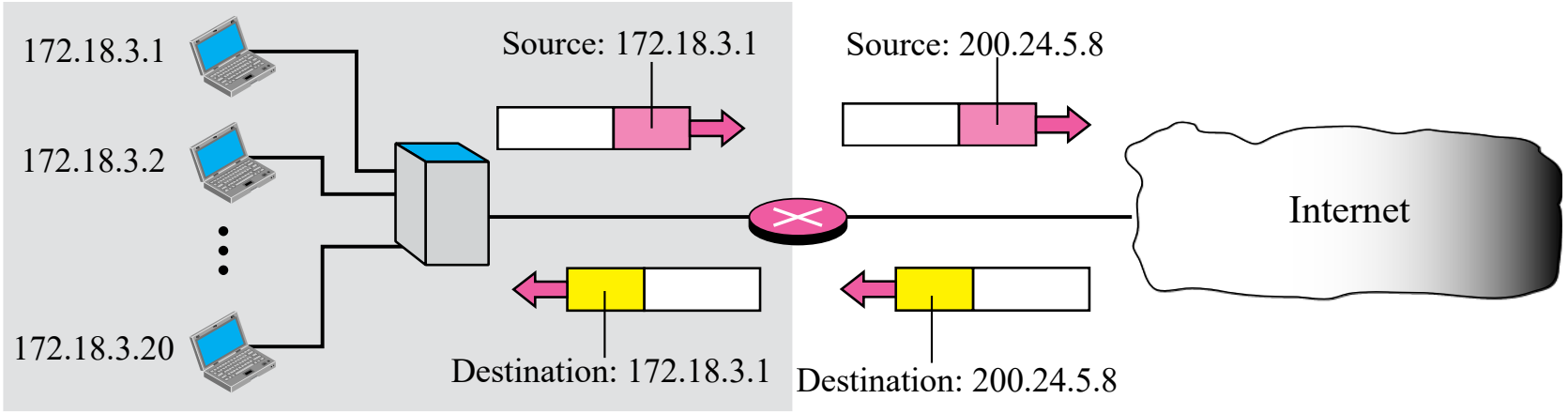
Revision ….



Site using private addresses

Network Address Translation (NAT)

Revision ….

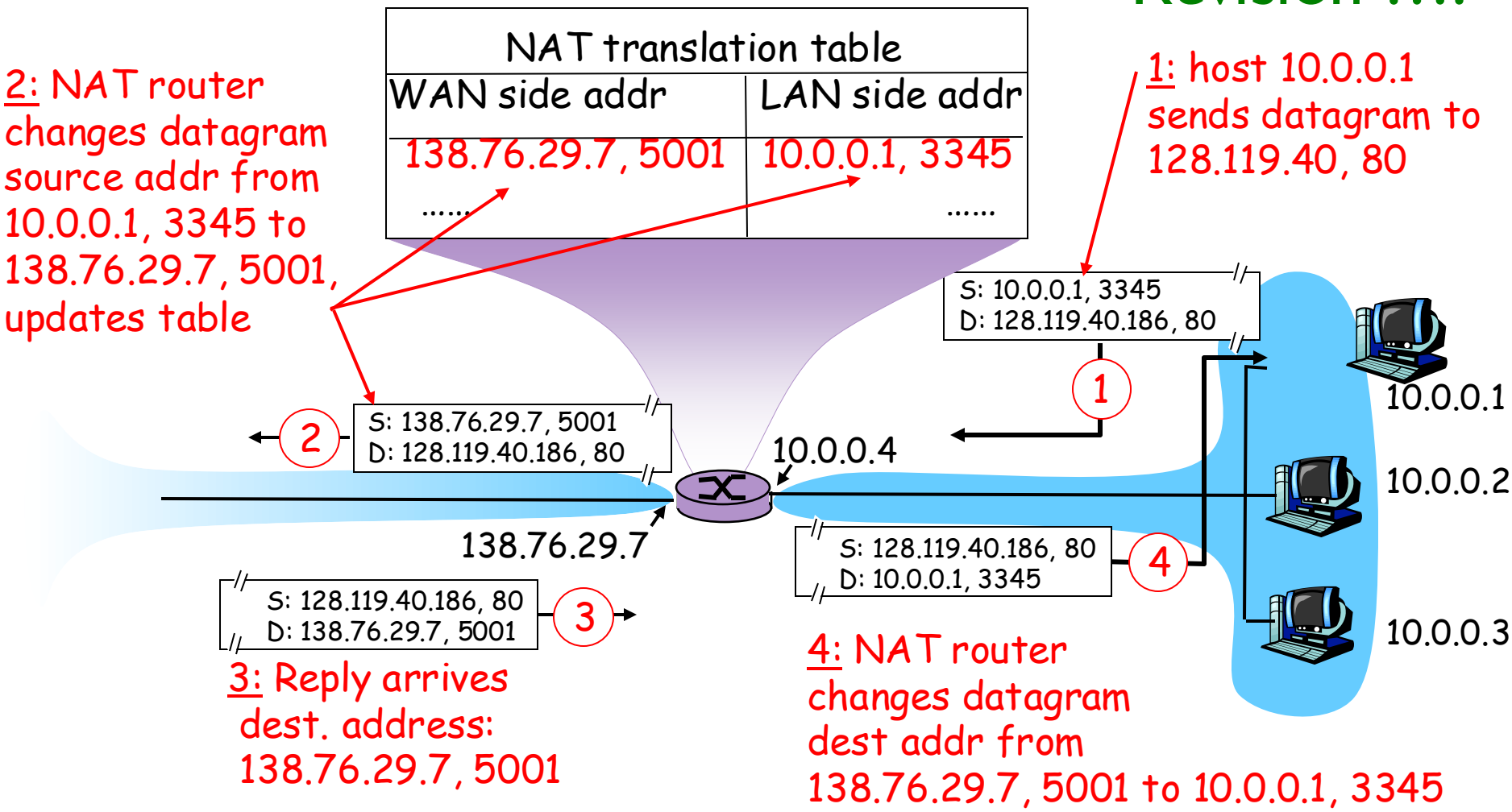


Site using private addresses

it simply maps the public ip to a private ip address alongside a port in the table.
so all the requests go through the same public ip address

NAT: Network Address Translation

Revision ....



NAT Table

Revision ….

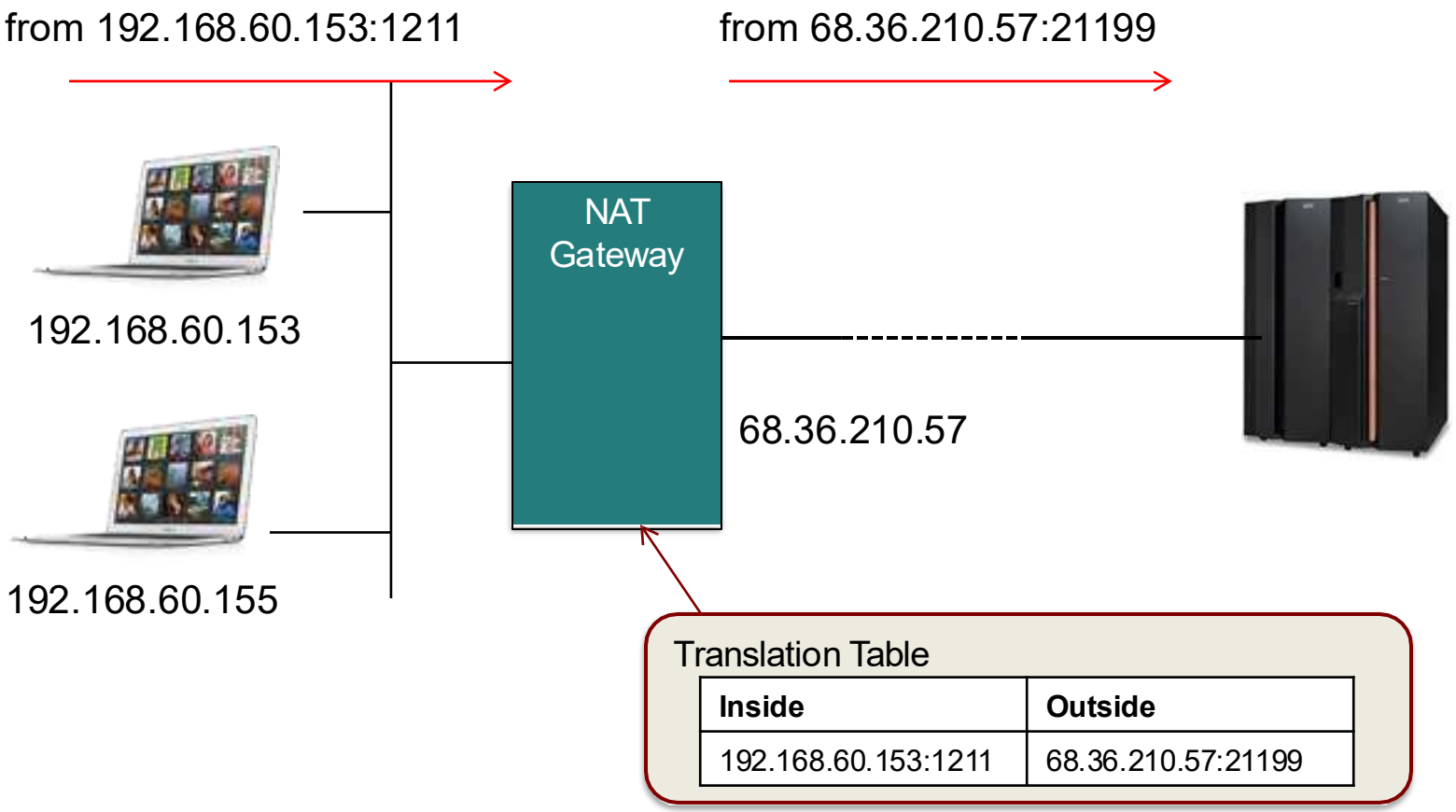
Five-column translation table

Private Address	*Private Port*	*External Address*	*External Port*	*Transport Protocol*
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
…	…	…	…	…

NAT traversal & why do we need it?

- NAT is awesome!
  - Maps multiple local private addresses to a single public IP address. You **don't need many public addresses**!
  - **External hosts only see a single IP device** and do not have visibility into a network's internal/private structure
- **But** it breaks end-to-end connectivity!
  - Host behind NAT box can access services in the internet, but **hosts in the internet cannot (easily) reach hosts behind NAT box**.
  - Consider **two VoIP clients behind NAT** that want to communicate

NAT: Accessing external public IP server - Normal Operation - This is easy



SIMPLE, on the fly **address mapping**!

"Normal" operations in NAT box

how does nat work with socket programming?

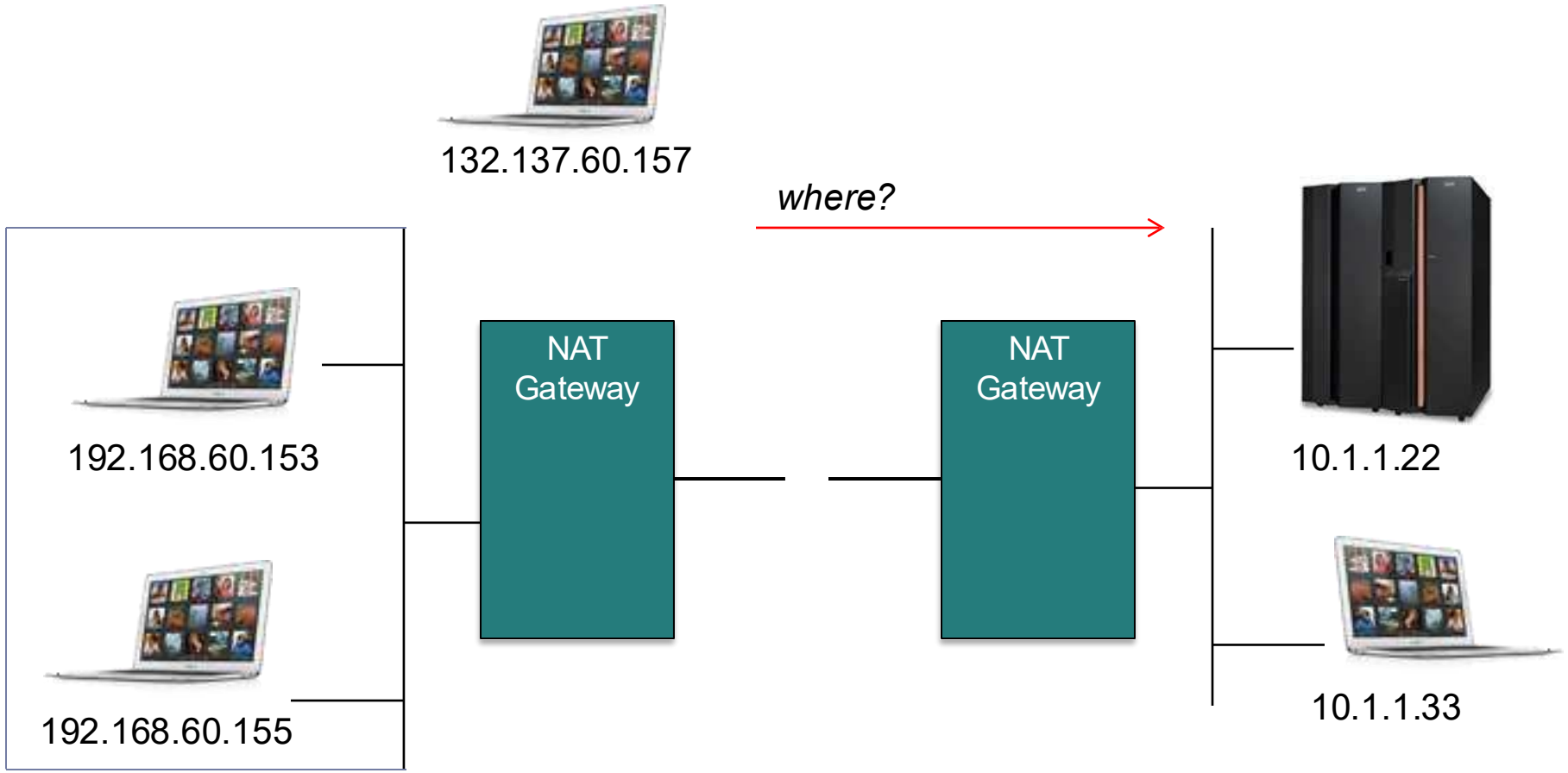
- Mapping is "reactive",
  - X (SIP:SP=192.168.31.3:5555, DIP:DP=34.56.78.9:80)
  - NAT entry is setup by the first packet from X to outside world

    NAT (SIP:SP=192.168.31.3:5555 > 132.137.3.13:2345, DIP:DP=34.56.78.9:80)
  - Expects a return packet with the source and destination information reversed.

    NAT Entry (SIP:SP=34.56.78.9:80, DIP:DP=132.137.3.13:2345 > 192.168.31.3:5555)

NAT: Accessing a server behind NAT- This is tricky

why is this so? isnt it just like sending from host to host? whats the issue?



NAT ISSUES….

so the issue is the inbound ?

- **Issue 1:** You need to access a server behind the NAT gateway from the Internet
  - Solution 1: Port Forwarding - Define a specific mapping for incoming packets (in advance) - See lab 7
  - Solution 2: Connection Reversal

- **Issue 2:** Both hosts are behind NAT boxes
  - Solution 1: Relay all with a Proxy
  - Solution 2: Hole Punching. - See lab 7

NAT Traversal Techniques

Port Forwarding (See lab 7)

Hole Punching (See lab 7)

Port Forwarding. - See lab 7

1. Define Mapping in advance (Port Forwarding):
   - X (192.168.31.3) runs a TCP server listening at port 5555
     - Known to the outside world as 132.137.3.13:443
     - We set up this translation in the NAT box in advance.
   - NAT (SIP:SP=*:*, DIP:IP=132.137.3.13:443 -> DIP:DP=192.168.31.3:5555)
   - **What's next?** Anything to do with DNS?
     - Buy a domain and insert a 'A' record with 132.137.3.13:443.
2. Another Solution: Connection Reversal

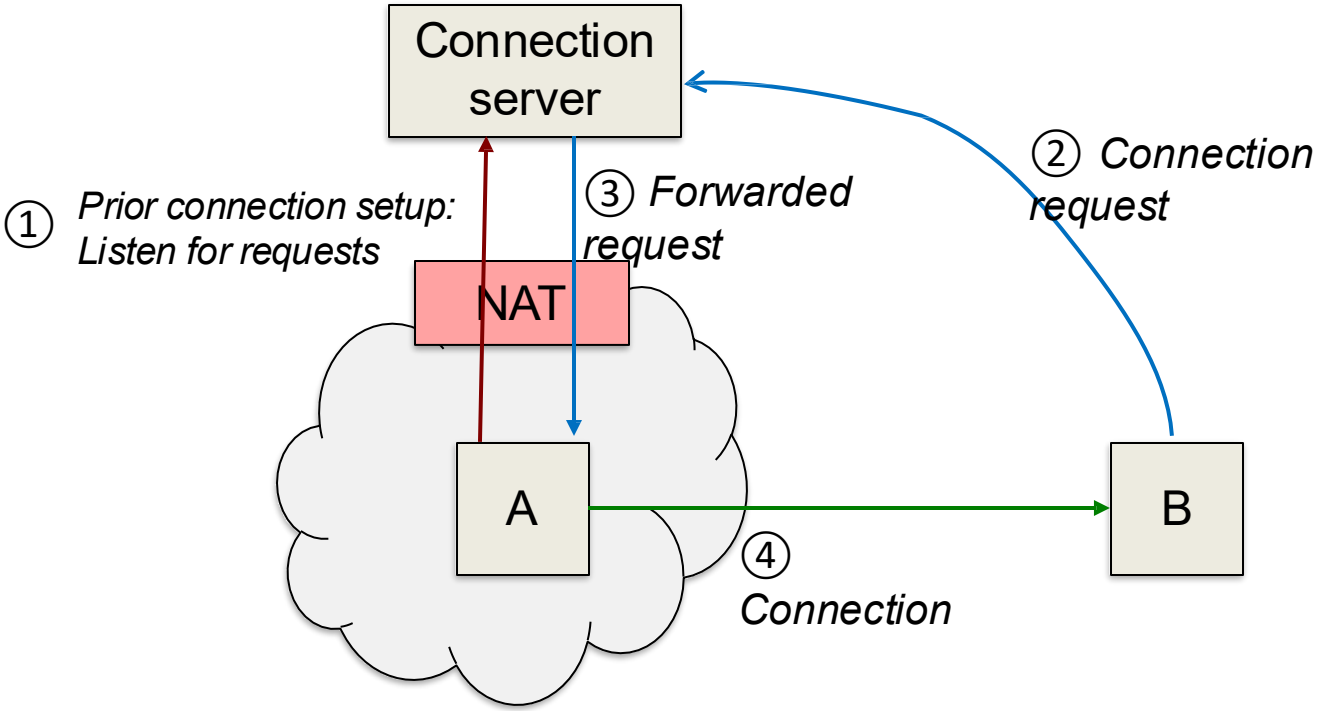
Connection reversal

- B wants to connect to A
  - But A is behind a NAT
- Somehow get B to send a message to A,
  - Ask A to open a connection to B
- Two approaches
  - Relay the request via a server (but A must be connected to the server)
  - Assume an existing connection between A & B and ask for a new one

so does this mean for example, there is no current NAT entry on the router with mappings to A from the public IP, hence a connection server is used, once the server receives a connection request from B, it tells to A as it is listening, then A will send a request to B and map on the NAT table as well to receive the connection from B? So first, why doesnt A directly create the mapping on the NAT and instead requires a connection server. And what does B receive?

Connection reversal

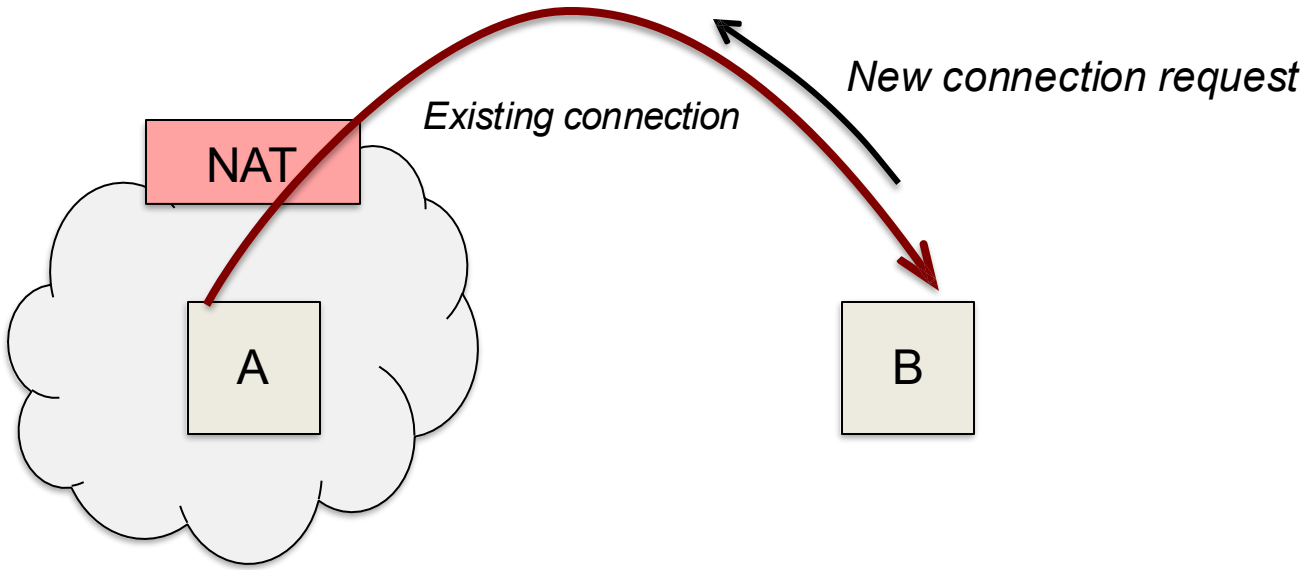
Use a server for sending only connection requests



Connection reversal

B wants to talk to A

There is an existing connection between A & B (set up by B previously)



Hole Punching - See lab 8

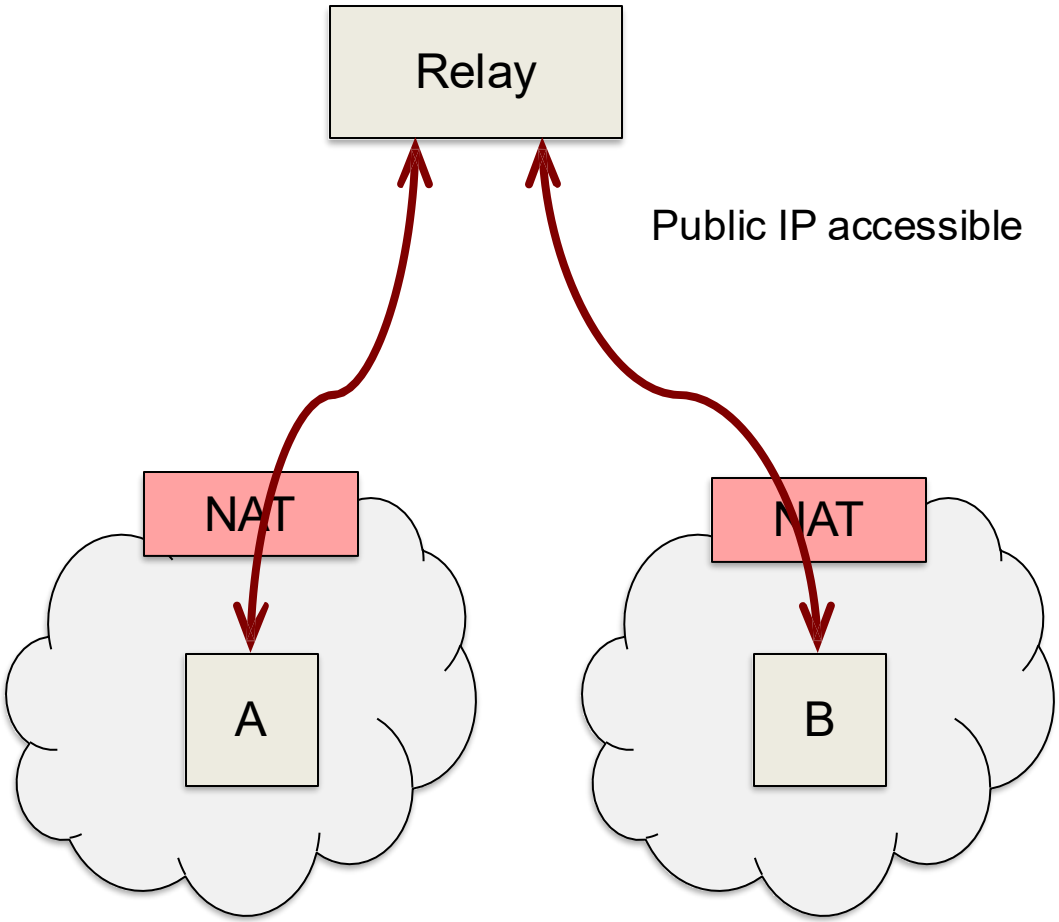
- Devices A and B are behind NAT gateways
  - Normal and port forwarding do not work!

1. Simple Solution - **Relay** : Use a proxy server to relay messages
2. Better Solution - **Hole punching**: Use a rendezvous server (external controller) with known public **IPaddress:port** to inform both devices the NAT IP address and port to try

Relay all messages through a Proxy

- Hosts A & B want to communicate
- Have an Internet-accessible proxy, P
- A connects to P and waits for messages on the connection
- B talks to P; P relays messages to A
- Most reliable but not very efficient
  - Extra message relaying
  - Additional protocols needed (e.g., B needs to state what it wants)
  - Proxy can become a point of congestion (network links & CPU)

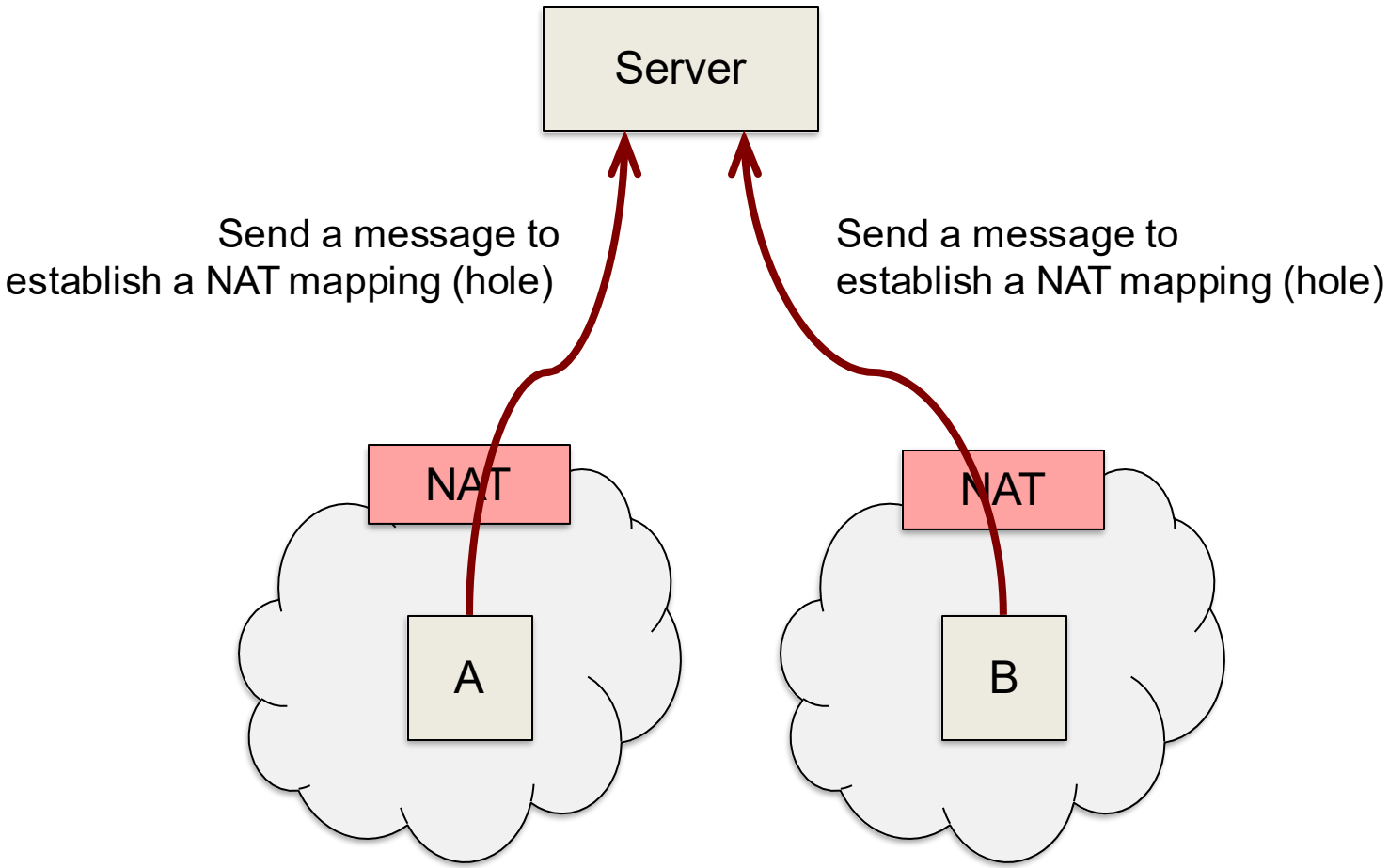
Relay all messages



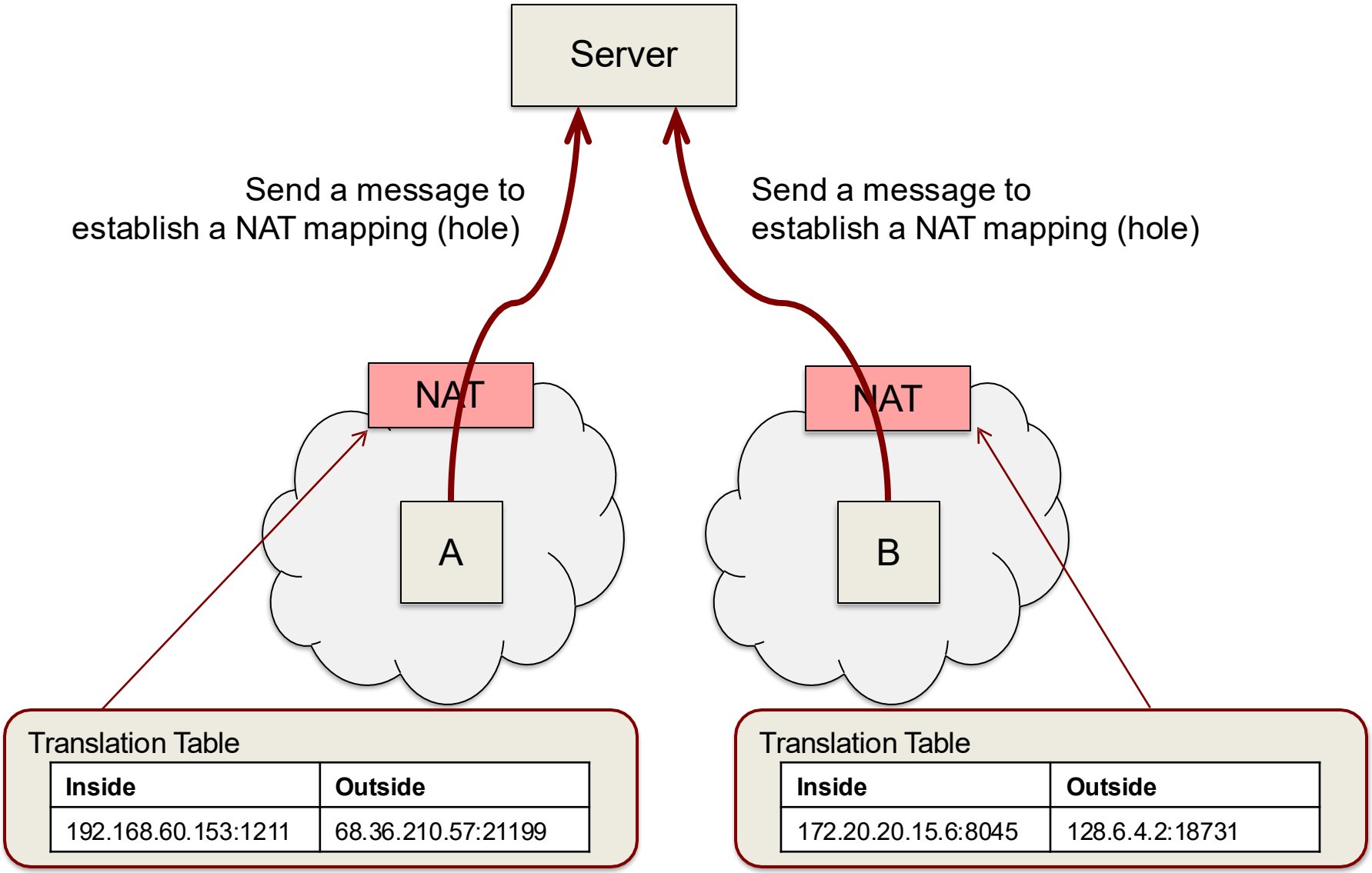
UDP hole punching

- Hosts A & B want to communicatee
- Have an Internet-accessible **rendezvous server** (external controller), S
- Host A sends a message to S
  - That sets up a NAT translation on A's NAT gateway
  - S now knows the external host & port
- Host B sends a message to S
  - That sets up a NAT translation on B's NAT gateway
  - S also knows the external host & port on B
- **S tells B:** *talk on A's IP address & port*
- **S tells A:** *talk to B's IP address & port*

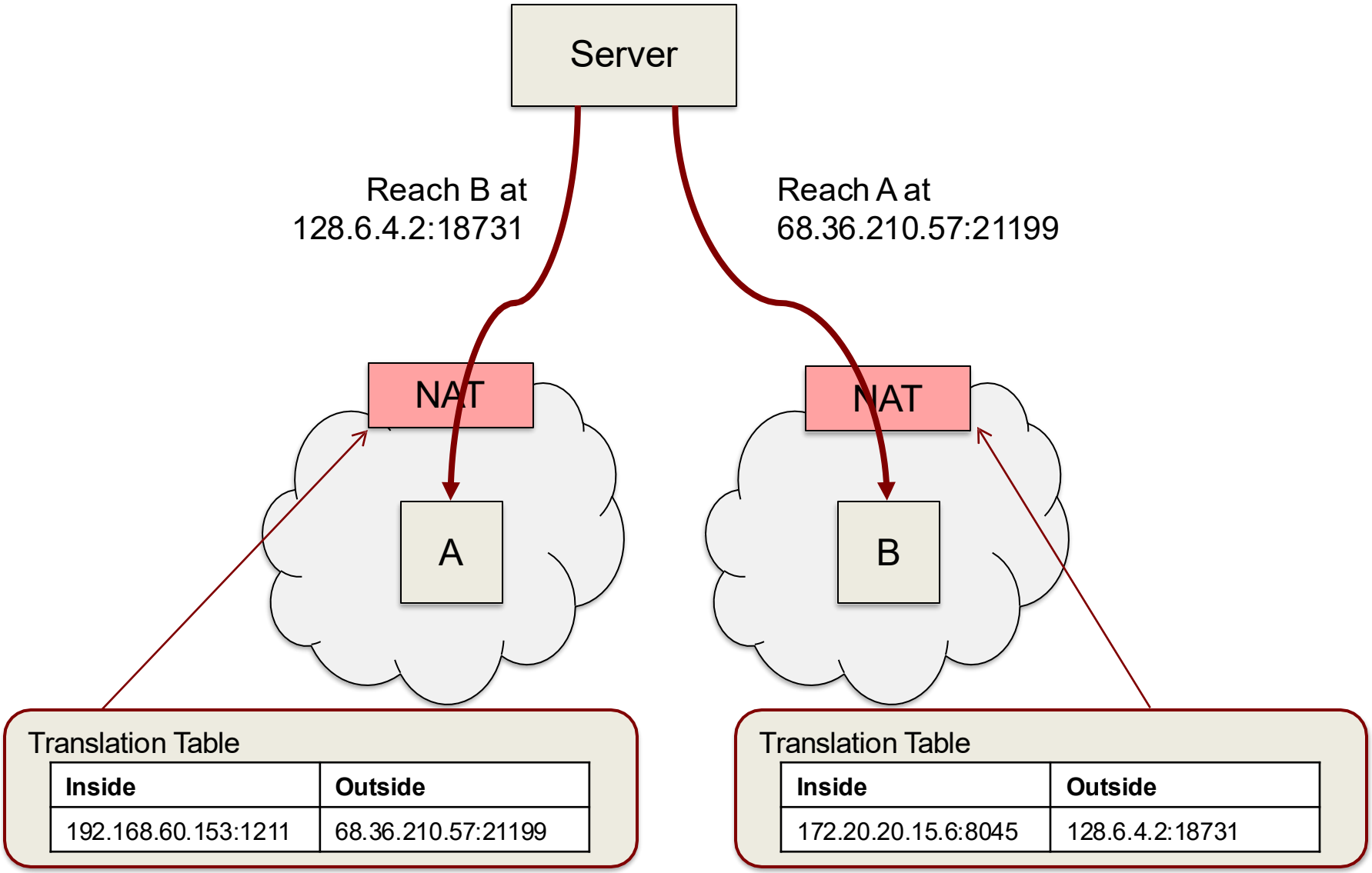
UDP hole punching



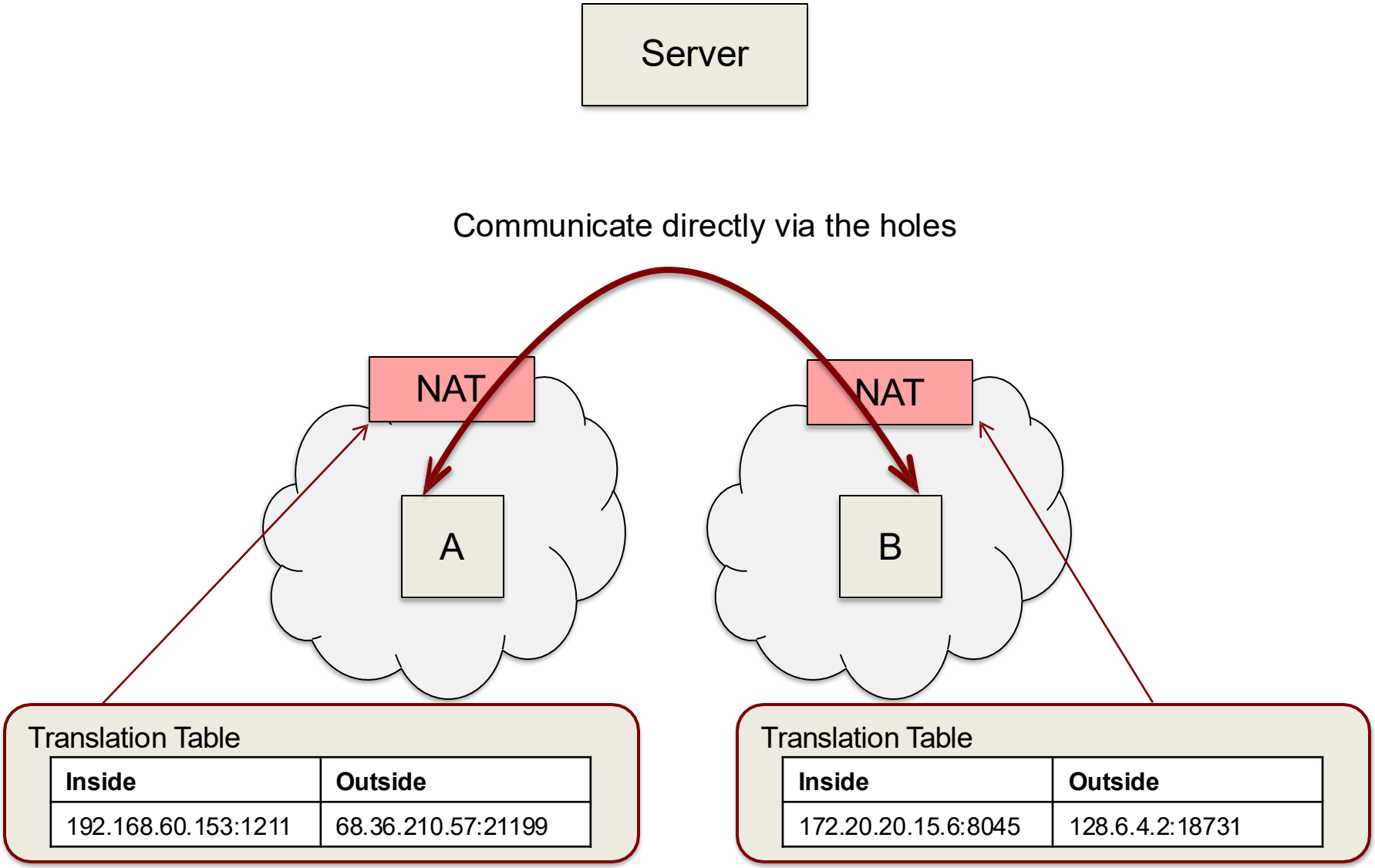
UDP hole punching



UDP hole punching



UDP hole punching



TCP hole punching

- Same principle (tell other host of your address:port) – BUT
  - Use TCP Simultaneous Open
    - Both hosts will try to connect to each other
    - Each NAT creates a translation rule
    - At least one of the SYN messages during connection set up will go through the NAT translation on the other side
      - The remote side will send a SYN-ACK
  - Need to re-use the same port # that the remote side knows about
    - Socket option to reuse an address:
      SO_REUSEADDR
  - Not guaranteed to work with all NAT system

FURTHER READING: https://bford.info/pub/net/p2pnat/

Questions to Ponder:

Q:Why simultaneous Open?

Q:When (which tye of NAT system) TCP hole punching will fail?

Summary

- IPv4 Format
- Fragmentation
- IP-Options: Record Route
- IP-Options: Strict/Loose Source Route
- ICMP – Error Message
  - Destination unreachable, Source-quench, time-exceeded, redirect.
- ICMP – Query Message
  - Echo req/rep **(ping)**, Time Stamp req/rep.
- Trace Route
- NAT Traversal -- Port Forwarding & Hole Punching

END