THE NETWORK LAYER – FORWARDING AND HELPING

COMP 30650: NETWORKS AND INTERNET SYSTEMS

Dr. Gavin McArdle
Email: gavin.mcardle@ucd.ie
Office: A1.09 Computer Science

RECAP

Network Layer

- Datagram Service
- Internet Protocol
- Virtual Packet Switching
- Internetworking
- IP Address format
- Forwarding

TODAY'S PLAN

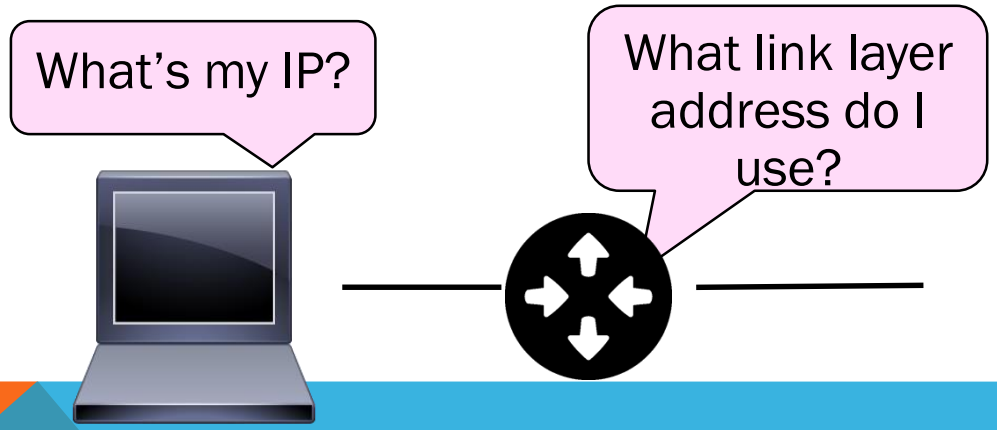
Helper Protocols for Network Layer

- ARP and DHCP

DHCP & ARP

Filling in the gaps we need to make for IP forwarding work in practice

- Getting IP addresses (DHCP)
- Mapping IP to link addresses (ARP)



GETTING IP ADDRESSES

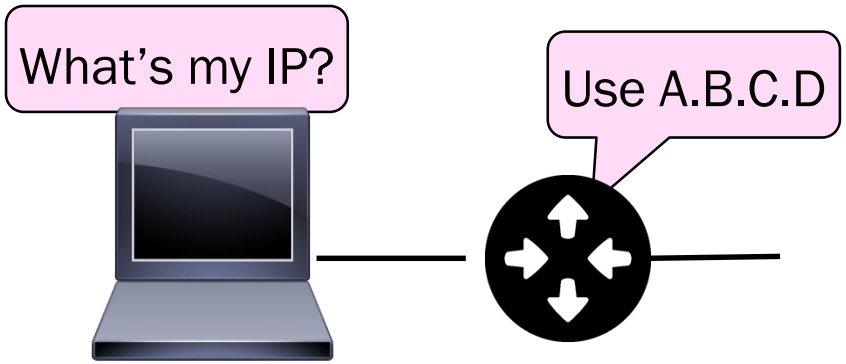
Problem:

- A node wakes up for the first time …
  - What is its IP address?
  - What's the IP address of its router?
  - What's the subnet mask?

Want this to just work without human interaction.

GETTING IP ADDRESSES

1. Manual configuration (old days)
   - Can't be factory set, depends on use
   - As seen in the practical sessions!
2. A protocol for automatically configuring addresses (DHCP)



DHCP

DHCP (Dynamic Host Configuration Protocol)

It leases IP addresses to nodes

Provides other parameters too

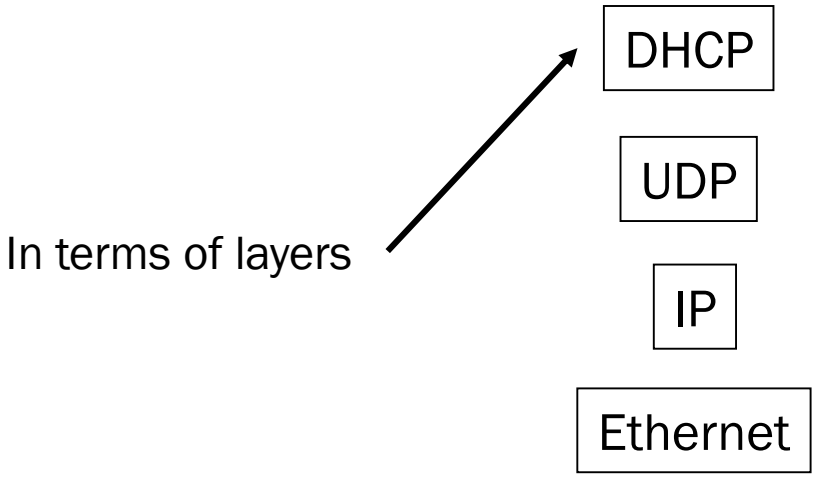
- Network prefix
- Address of local router
- DNS server, time server, etc.

  More on **D**ynamic **N**ame **S**ervice later

DHCP PROTOCOL STACK

DHCP is a client-server application

- Uses UDP
  - User Datagram Packets



DHCP ADDRESSING

Bootstrap issue:

- How does node send a message to DHCP server before it is configured?

DHCP ADDRESSING

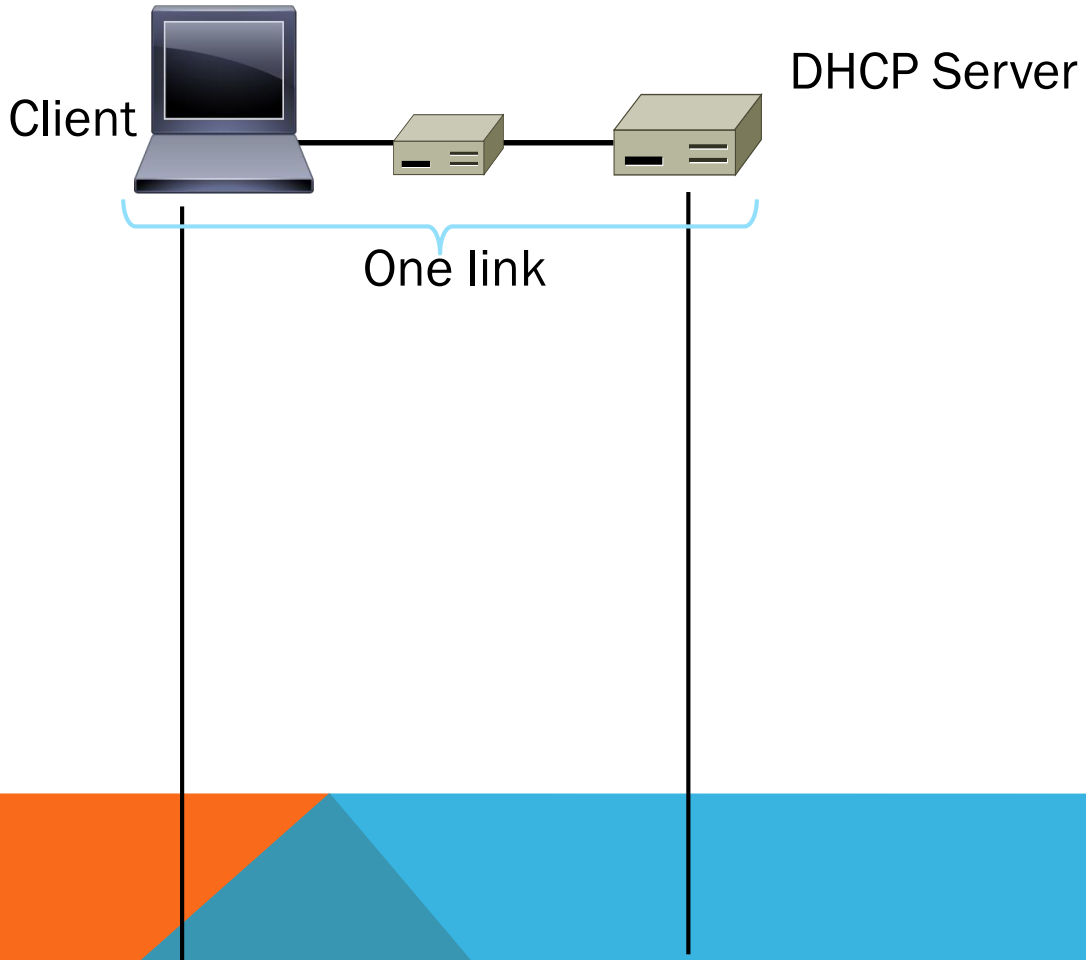
Bootstrap issue:

- How does node send a message to DHCP server before it is configured?

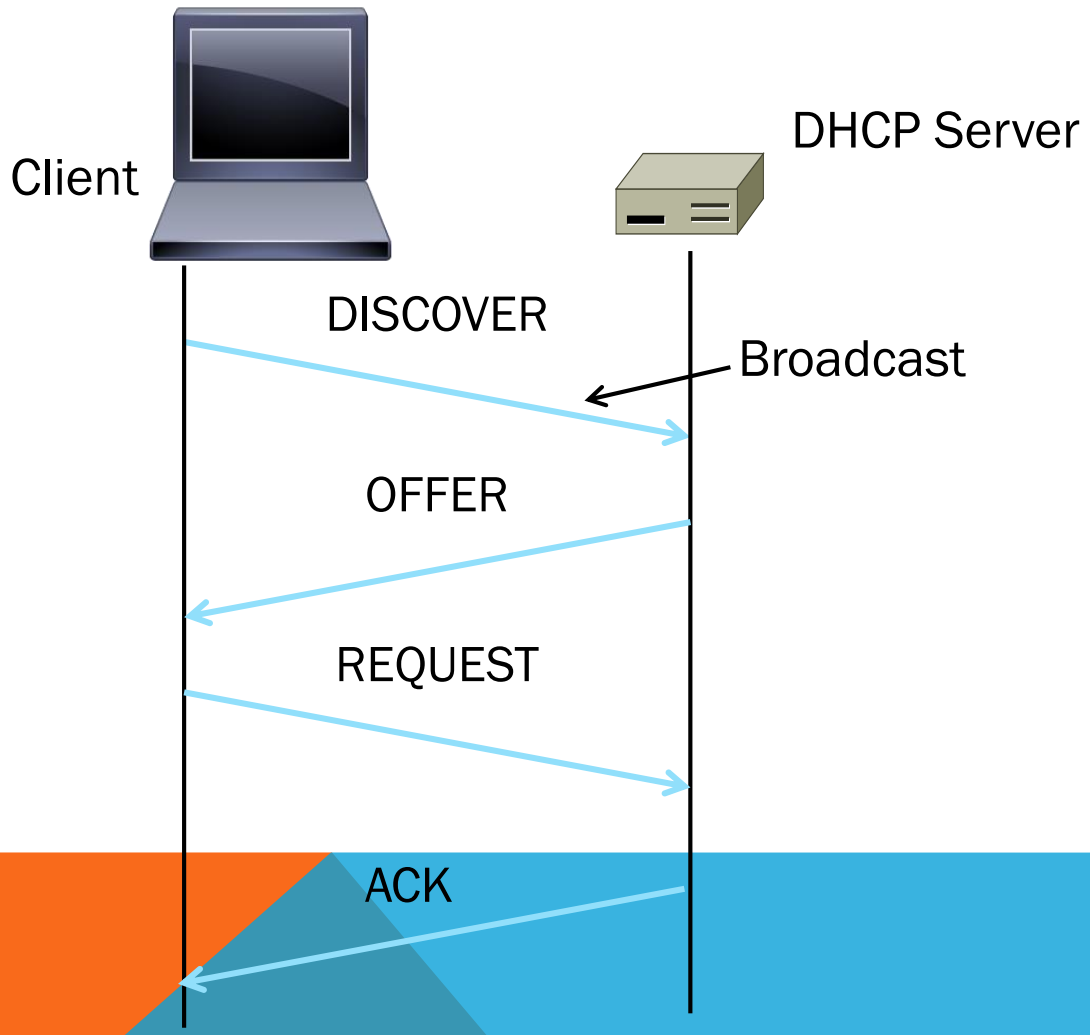
Answer:

- Node sends broadcast messages that deliver to all nodes on the network
- Broadcast address is all 1s
- IP (32 bit): 255.255.255.255
- Ethernet (48 bit): ff:ff:ff:ff:ff:ff

DHCP MESSAGES



DHCP MESSAGES



DHCP MESSAGES

To renew an existing lease, an abbreviated sequence is used:

- REQUEST, followed by ACK

Protocol also supports replicated servers for reliability

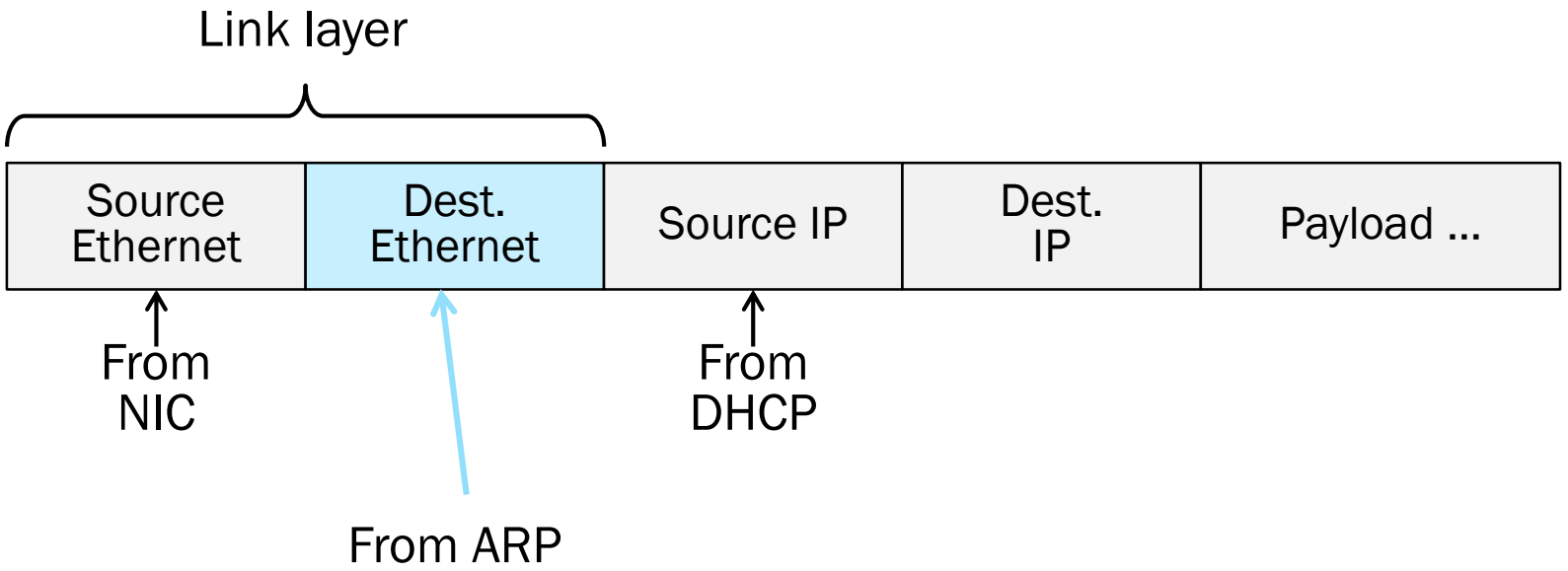
SENDING AN IP PACKET

Problem:

- A node needs **Link layer** addresses to send a frame over the local link
- How does it get the destination link address from a destination IP address?

ARP (ADDRESS RESOLUTION PROTOCOL)

Node uses it to map a local IP address to its Link layer addresses



NEXT PROBLEM

How do we connect networks with different maximum packet sizes?

- Need to split up packets *in transit*, **or** discover the largest size to use in advance of host sending packet

PACKET SIZE PROBLEM

Different networks have different maximum packet sizes or MTUs

- MTU = Maximum Transmission Unit
- E.g., Ethernet 1.5K, WiFi 2.3K

Prefer large packets for efficiency

- But what size is too large?
- Difficult because node does not know complete network path

PACKET SIZE SOLUTIONS

Fragmentation

- Split up large packets **in the network** if they are too big to send
  - Classic method

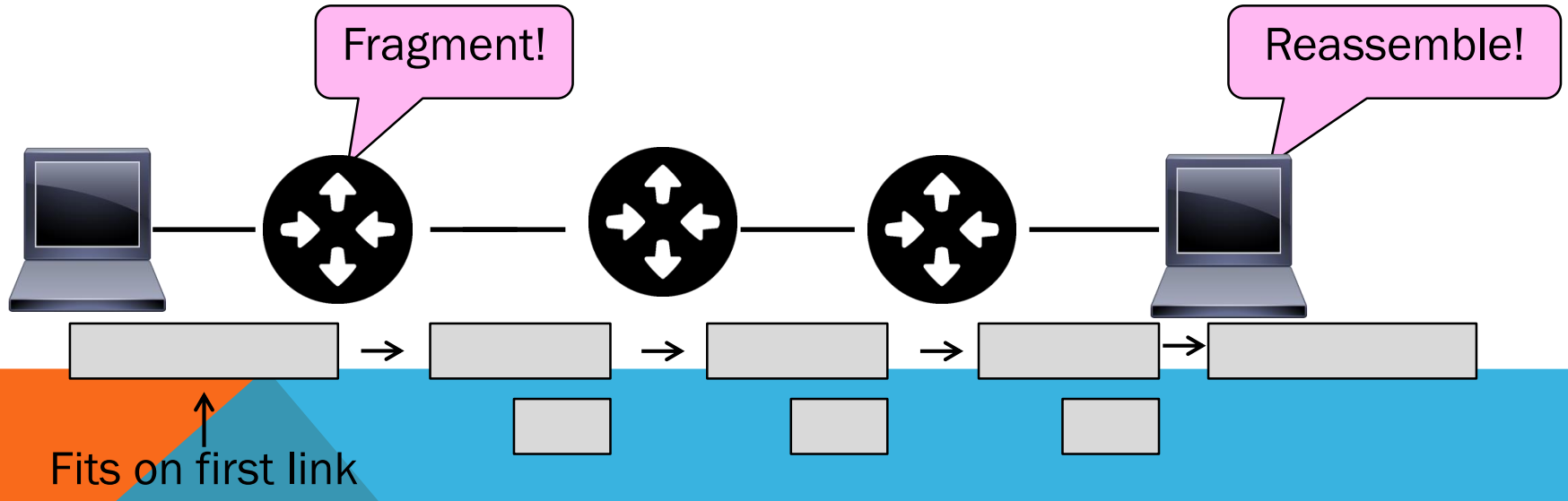
Discovery

- Find the largest packet that fits on the network path and use it
  - IP uses today instead of fragmentation

IPV4 FRAGMENTATION

Routers fragment packets that are too large to forward

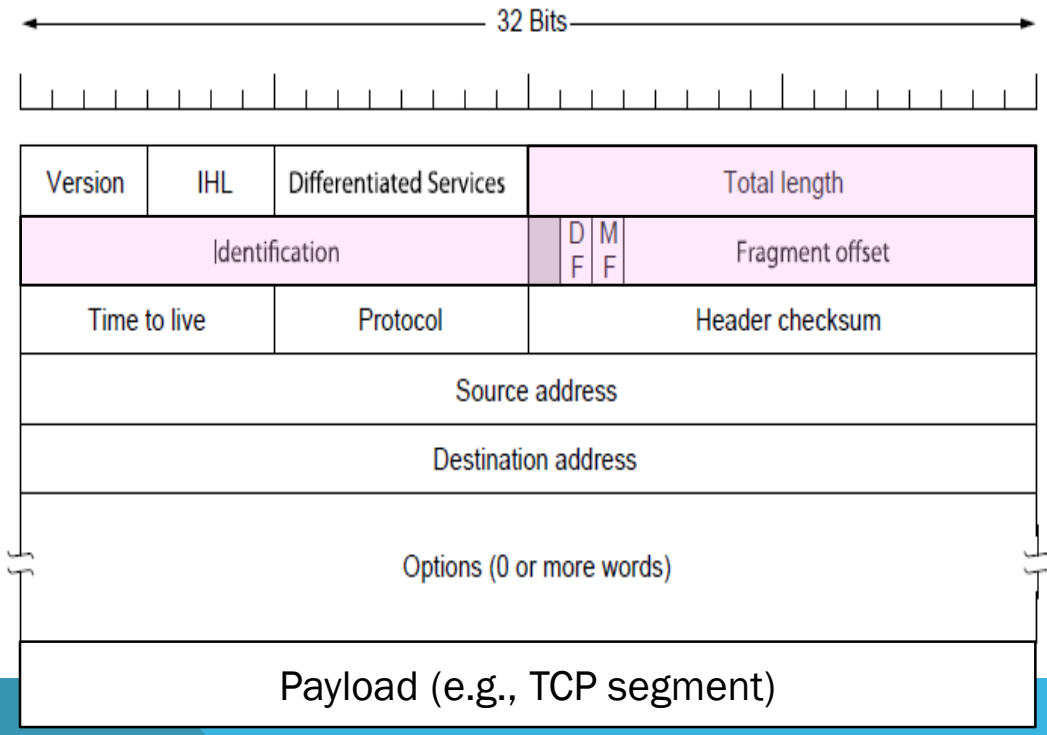
Receiving **host** reassembles to reduce load on routers



IPV4 FRAGMENTATION FIELDS

Header fields used to handle packet size differences

- Identification, Fragment offset, MF/DF control bits



IPV4 FRAGMENTATION PROCEDURE

Routers split a packet that is too large:

- Typically break into large pieces
- Copy IP header to pieces
- Adjust length on pieces
- Set offset to indicate position
- Set MF (More Fragments) on all pieces except last

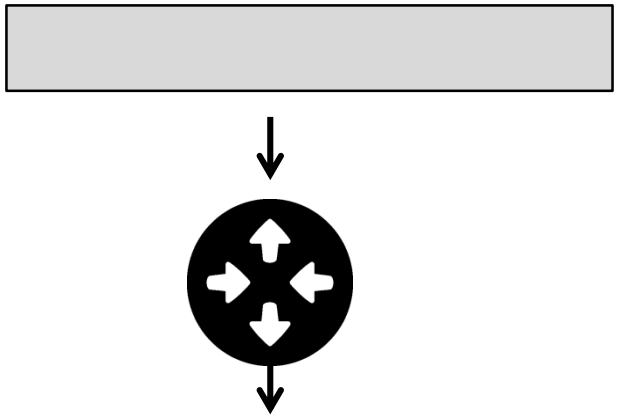
Receiving hosts reassembles pieces:

- Identification field links pieces together, MF tells receiver when it has all pieces

IPV4 FRAGMENTATION

Before
MTU = 2300

ID = 0x12ef
Data Len = 2300
Offset = 0
MF = 0



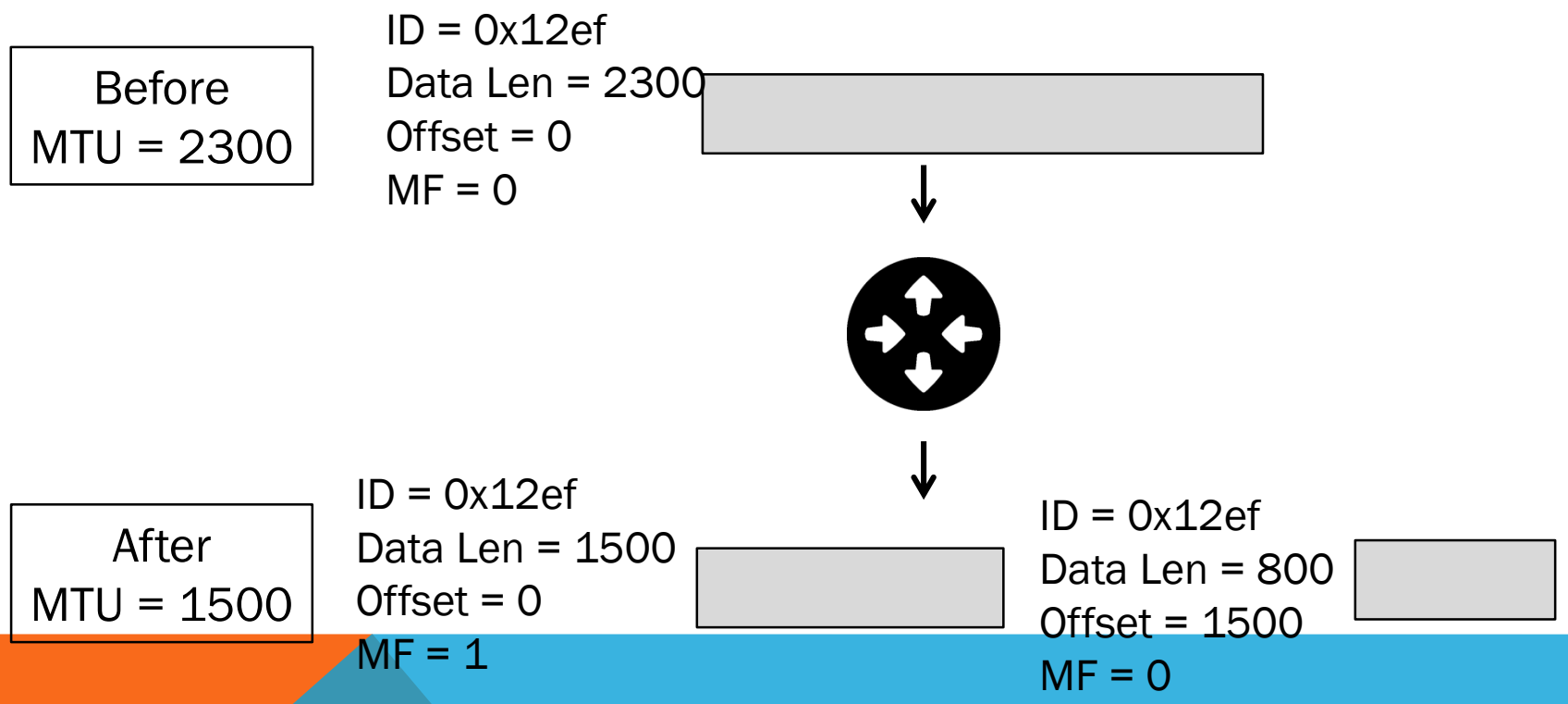
(Ignore length of headers)

After
MTU = 1500

ID =
Data Len =
Offset =
MF =

ID =
Data Len =
Offset =
MF =

IPV4 FRAGMENTATION (3)



IPV4 FRAGMENTATION

But fragmentation is undesirable

- More work for routers and hosts
- Tends to magnify loss rate
- Security vulnerabilities too

PATH MTU DISCOVERY

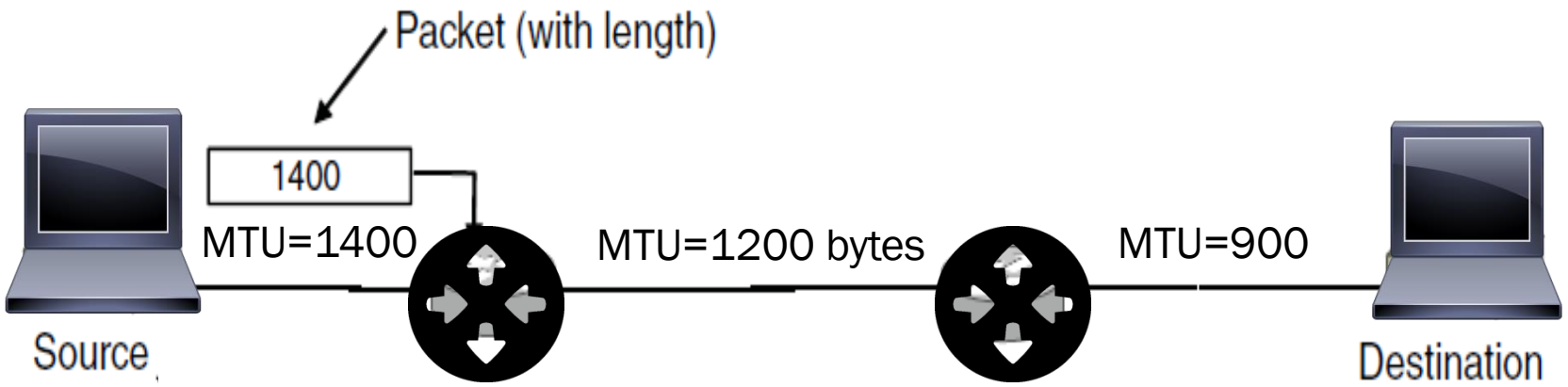
Discover the MTU that will fit

- Maximum Transmission Unit
- So we can avoid fragmentation
- The method in use today by IP

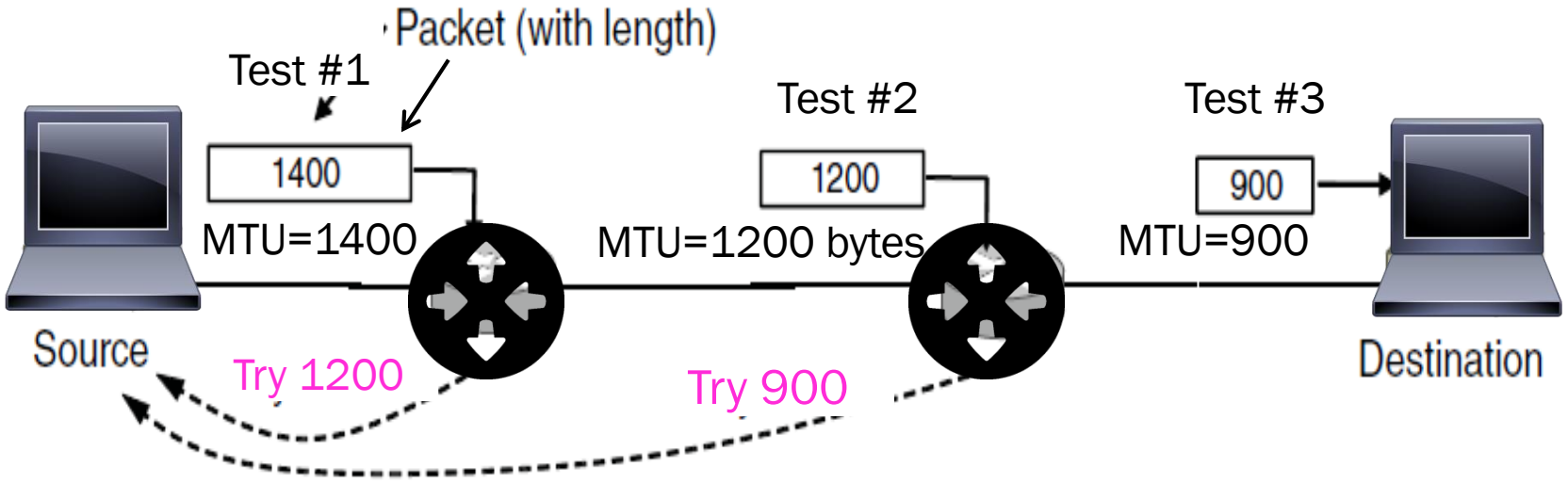
Host tests path with large packet

- Routers provide feedback if too large; they tell host what size would have fit

PATH MTU DISCOVERY (2)



PATH MTU DISCOVERY



PATH MTU DISCOVERY

Process may seem involved

- But usually quick to find right size

Path MTU depends on the path, so can change over time

- Search is ongoing

Implemented with ICMP

- Set DF (Don't Fragment) bit in IP header to get feedback messages

IP ERRORS - ICMP

What happens when something goes wrong during forwarding?

- Need to be able to find the problem

INTERNET CONTROL MESSAGE PROTOCOL

ICMP is a companion protocol to IP

- They are implemented together
- Sits on top of IP

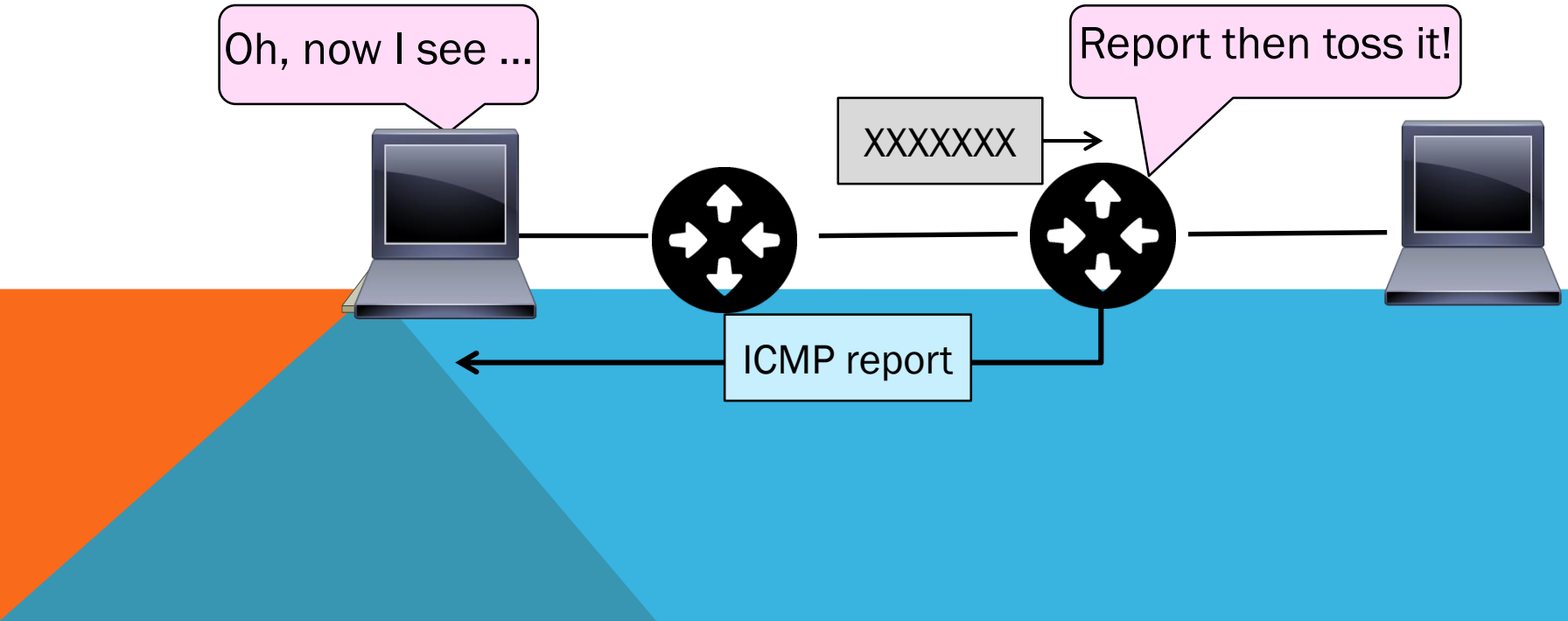
Provides error report and testing

- Error is at router while forwarding

ICMP ERRORS

When router encounters an error while forwarding:

- It sends an ICMP error report back to the IP source address
- It discards the problematic packet; host needs to rectify



ICMP MESSAGE FORMAT

Each ICMP message has a Type, Code, and Checksum

Often carry the start of the offending packet as payload

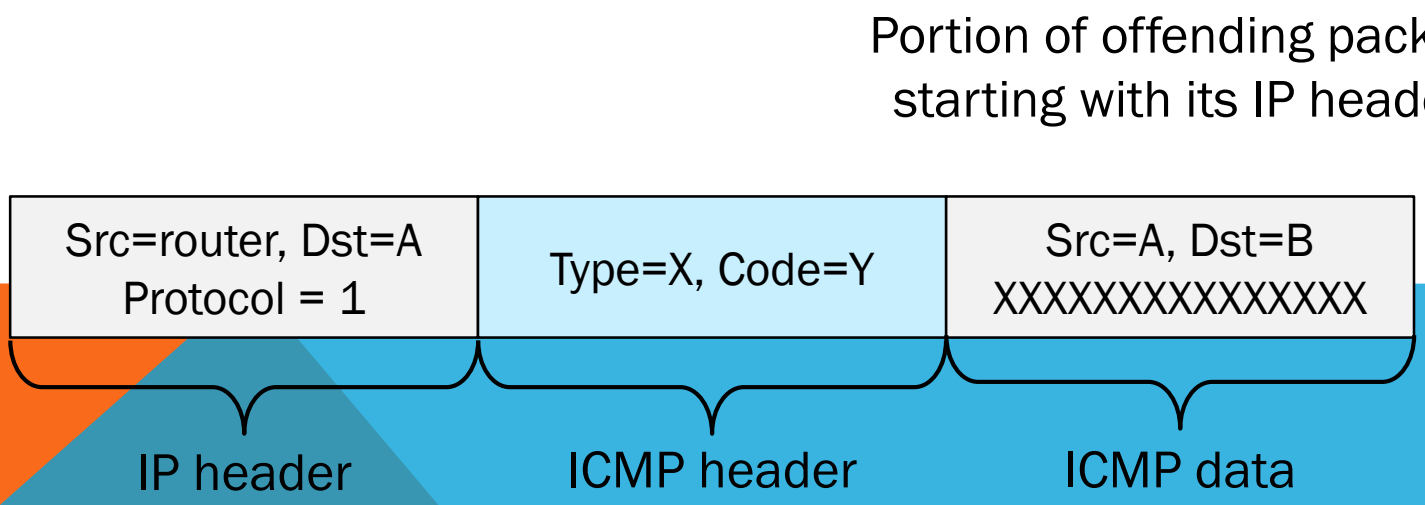
Each message is carried in an IP packet

ICMP MESSAGE FORMAT

Each ICMP message has a Type, Code, and Checksum

Often carry the start of the offending packet as payload

Each message is carried in an IP packet



EXAMPLE ICMP MESSAGES

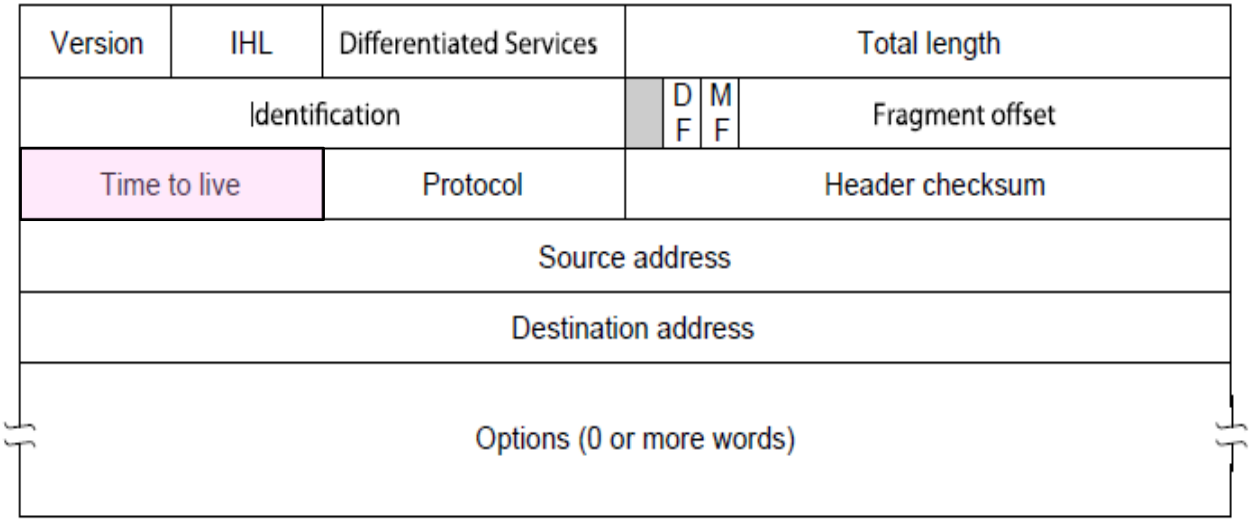
Name	Type / Code	Usage
Dest. Unreachable (Net or Host)	3 / 0 or 1	Lack of connectivity
Dest. Unreachable (Fragment)	3 / 4	Path MTU Discovery
Time Exceeded (Transit)	11 / 0	Traceroute
Echo Request or Reply	8 or 0 / 0	Ping

Testing, not a forwarding error: Host sends Echo Request, and destination responds with an Echo Reply

TRACEROUTE

IP header contains TTL (Time to live) field

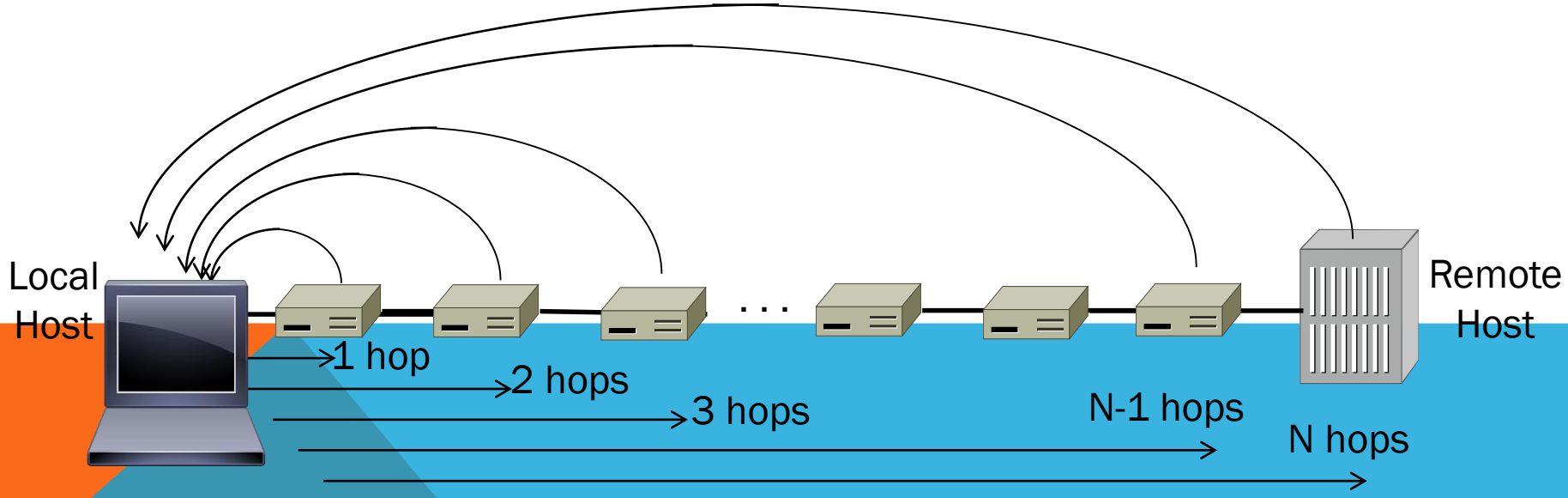
- Decremented every router hop, with ICMP error if it hits zero
- Protects against forwarding loops



TRACEROUTE

Traceroute repurposes TTL and ICMP functionality

- Sends probe packets increasing TTL starting from 1
- ICMP errors identify routers on the path



PUBLIC VERSUS PRIVATE IP ADDRESSES - NAT

What is NAT (Network Address Translation)? How does it work?

- NAT is widely used at the edges of the network, e.g., homes

NAT (NETWORK ADDRESS TRANSLATION) BOX

NAT box connects an internal network to an external network

- Many internal hosts are connected using few external addresses
- Middlebox that “translates addresses”

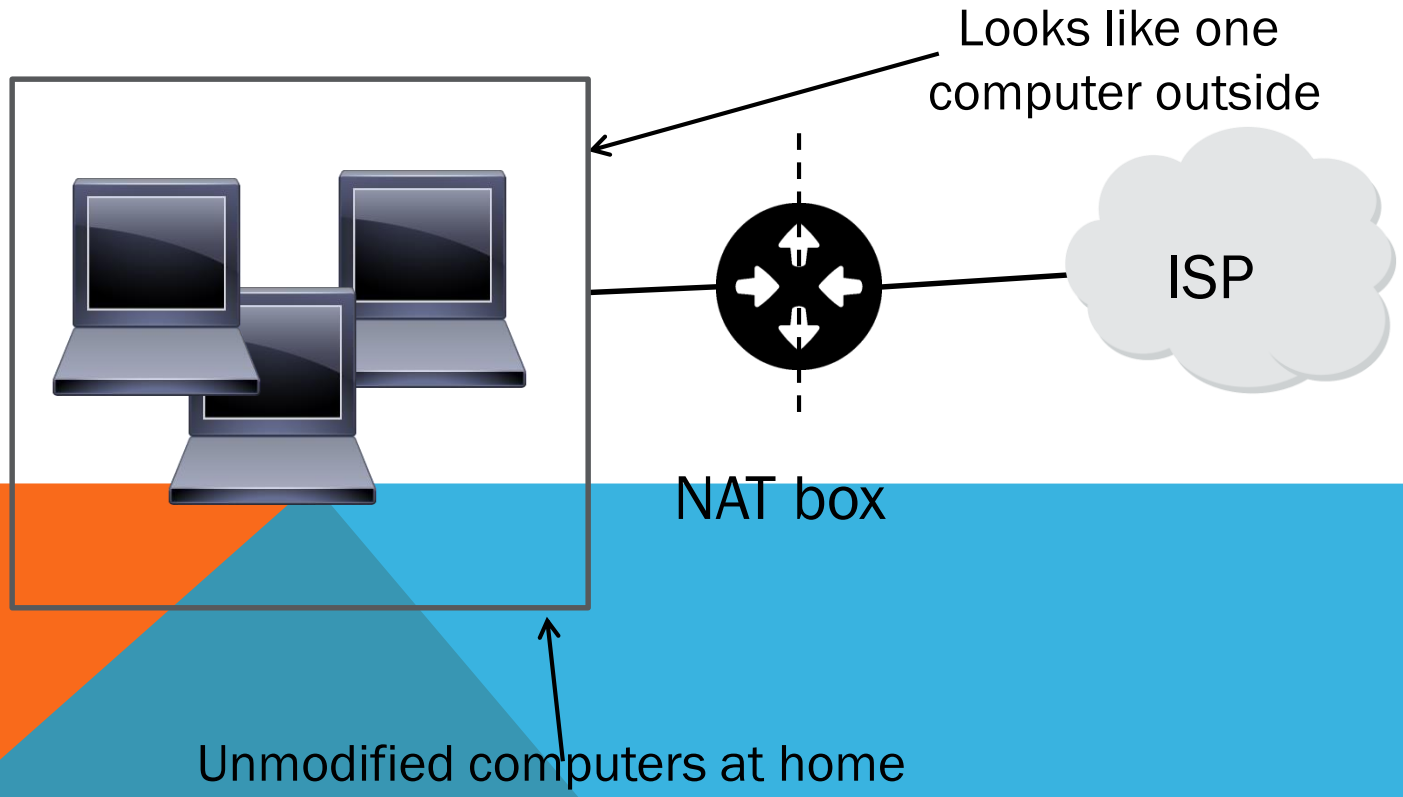
Motivated by IP address scarcity

- Controversial at first, now accepted

NAT

Common scenario:

- Home computers use “private” IP addresses
- NAT (in AP/firewall) connects home to ISP using a single external IP address



HOW NAT WORKS

Keeps an internal/external table

- Typically uses IP address + TCP port
- This is address and port translation
- Need ports to make mapping 1-1 since there are fewer external IPs

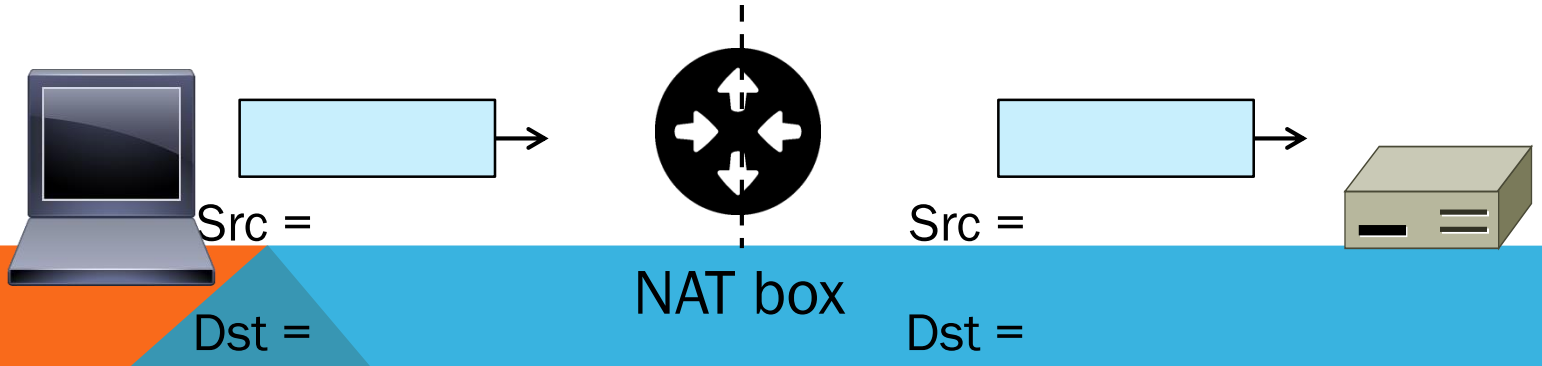
What host thinks	What ISP thinks
Internal IP:port	**External IP : port**
192.168.1.12 : 5523	44.25.80.3 : 1500
192.168.1.13 : 1234	44.25.80.3 : 1501
192.168.2.20 : 1234	44.25.80.3 : 1502

HOW NAT WORKS

Internal → External:

- Look up and rewrite Source IP/port

	Internal IP:port	External IP : port	
Internal source	192.168.1.12 : 5523	44.25.80.3 : 1500	External destination IP=X, port=Y

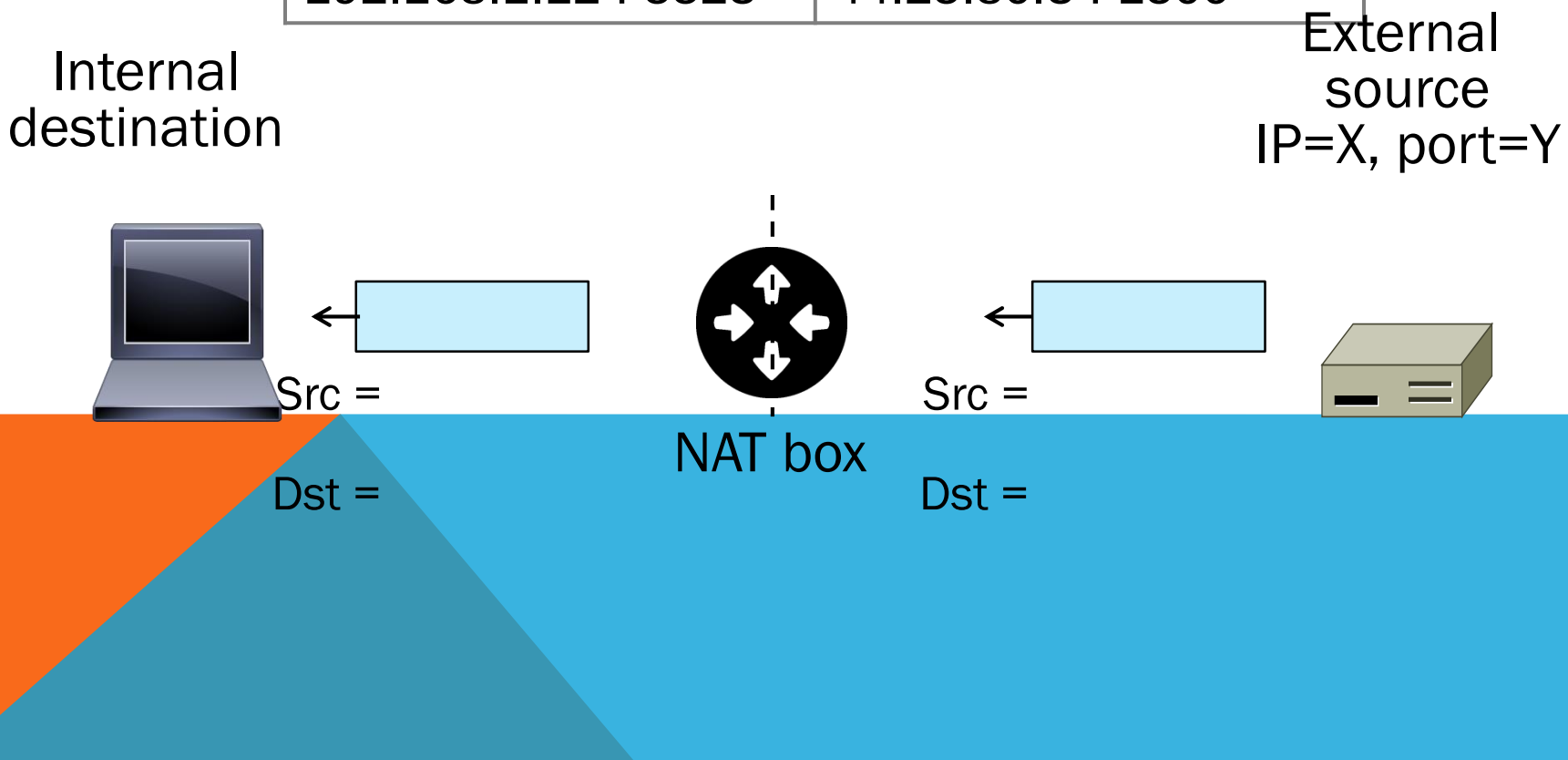


HOW NAT WORKS

External → Internal

- Look up and rewrite Destination IP/port

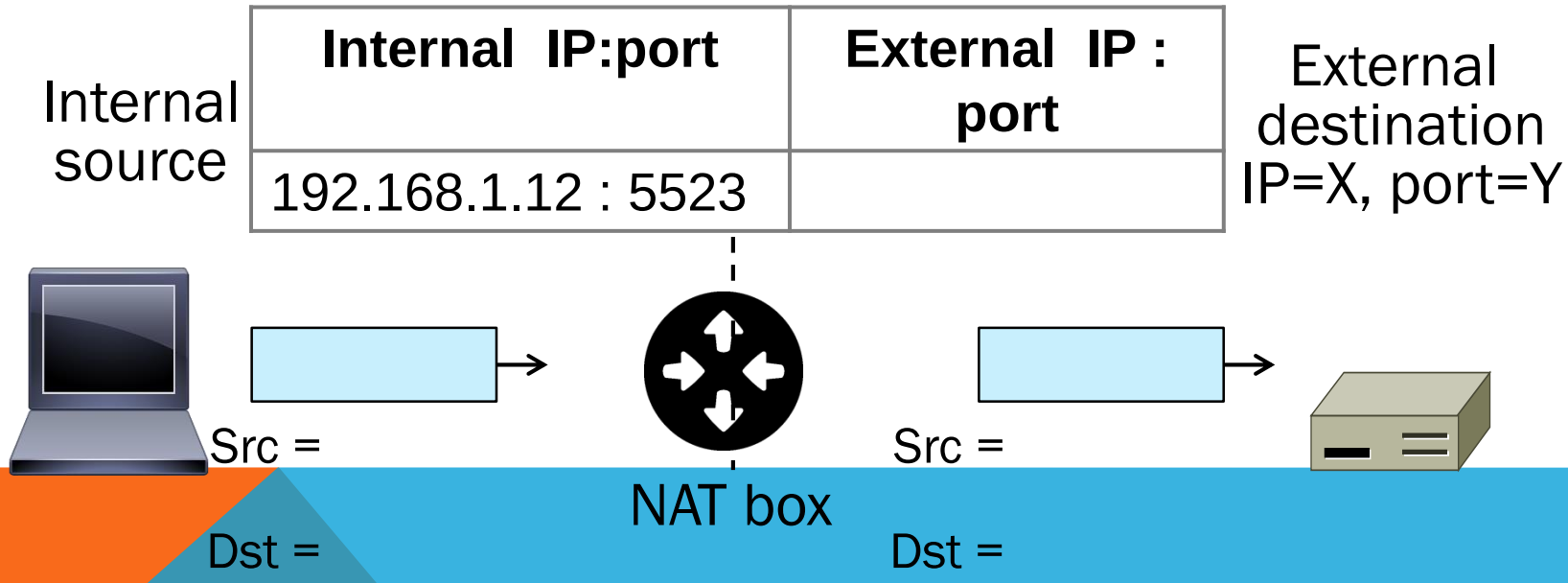
Internal IP:port	External IP : port
192.168.1.12 : 5523	44.25.80.3 : 1500



HOW NAT WORKS

Need to enter translations in the table for it to work

- Create external name when host makes a TCP connection



NAT DOWNSIDES

Connectivity has been broken!

- Can only send incoming packets after an outgoing connection is set up
- Difficult to run servers or peer-to-peer apps at home

Breaks apps that unwisely expose their IP addresses (FTP)

NAT UPSIDES

Relieves much IP address pressure

- Many home hosts behind NATs

Easy to deploy

- Rapidly, and by you alone

Useful functionality

- Firewall, helps with privacy