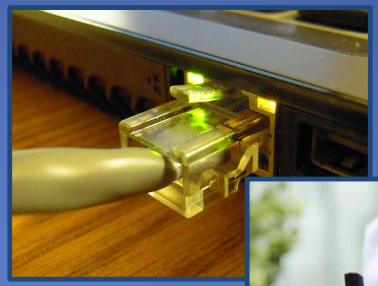
Networks III
Routing, BGP, Transport Layer, NMAP

CS 1660: Introduction to Computer Systems Security

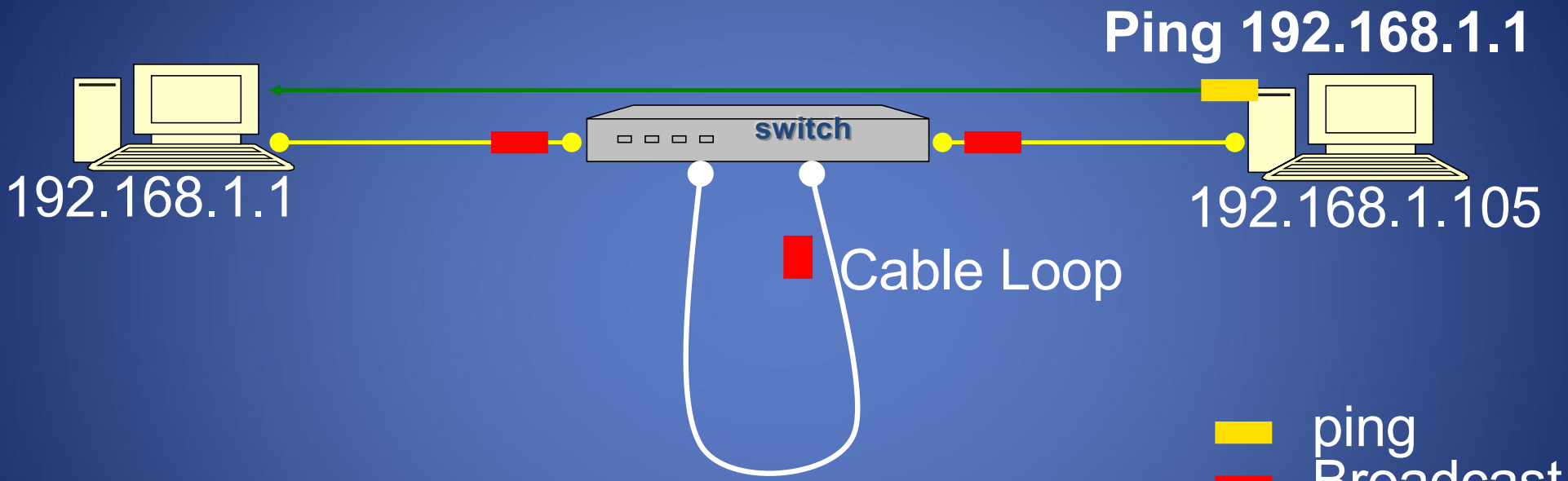
(recap) Where we are…



Local Area Network (LAN): "small" network: within a building, house, floor of office, etc.

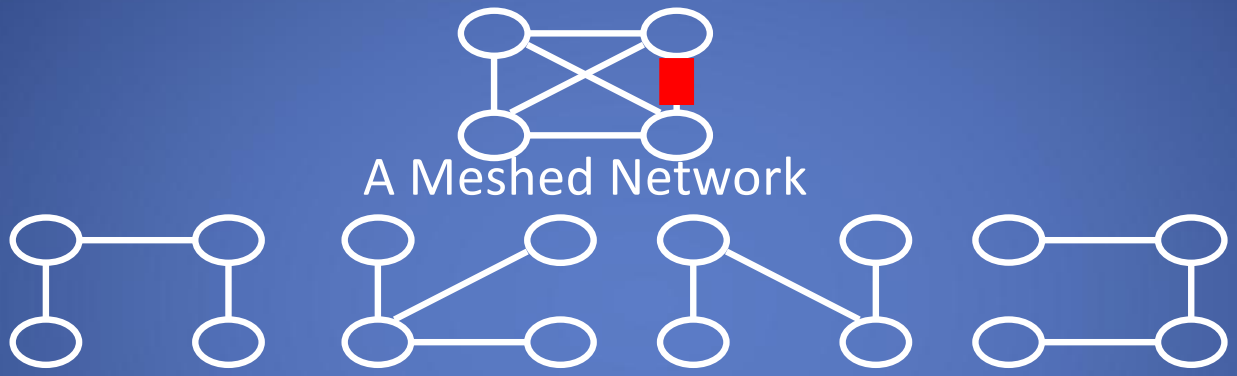
=> Security concerns before we even start talking to the wider Internet!

(recap) network DOS using Broadcat Requests



How can it be solved?

(recap) Spanning Tree Protocol (ISO 802.1D)



A Meshed Network

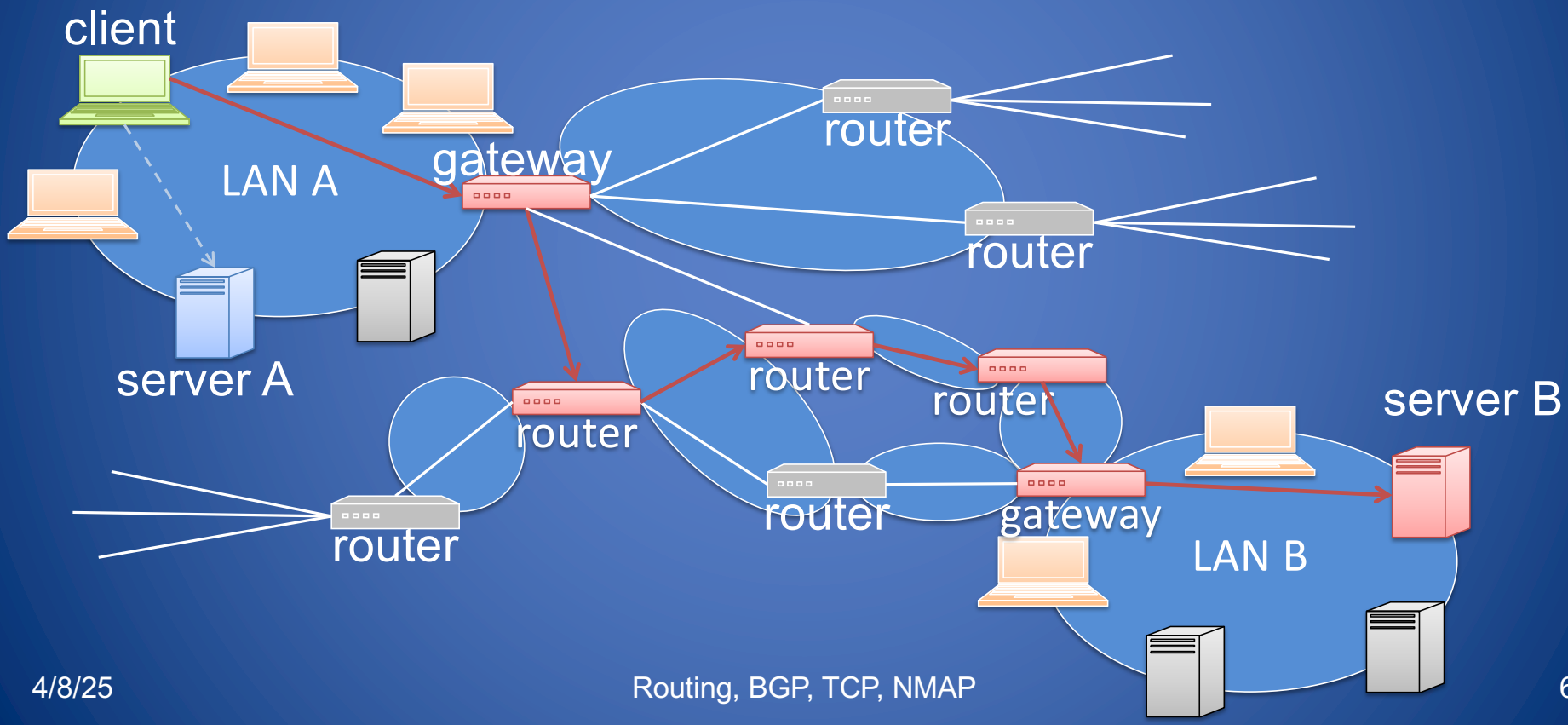
Four spanning trees of the Meshed Network

- Suppose you have a Meshed Network with bidirectional links that make loops/cycles…
- …then a spanning tree of the Meshed Network is the same network and no loops/cycles

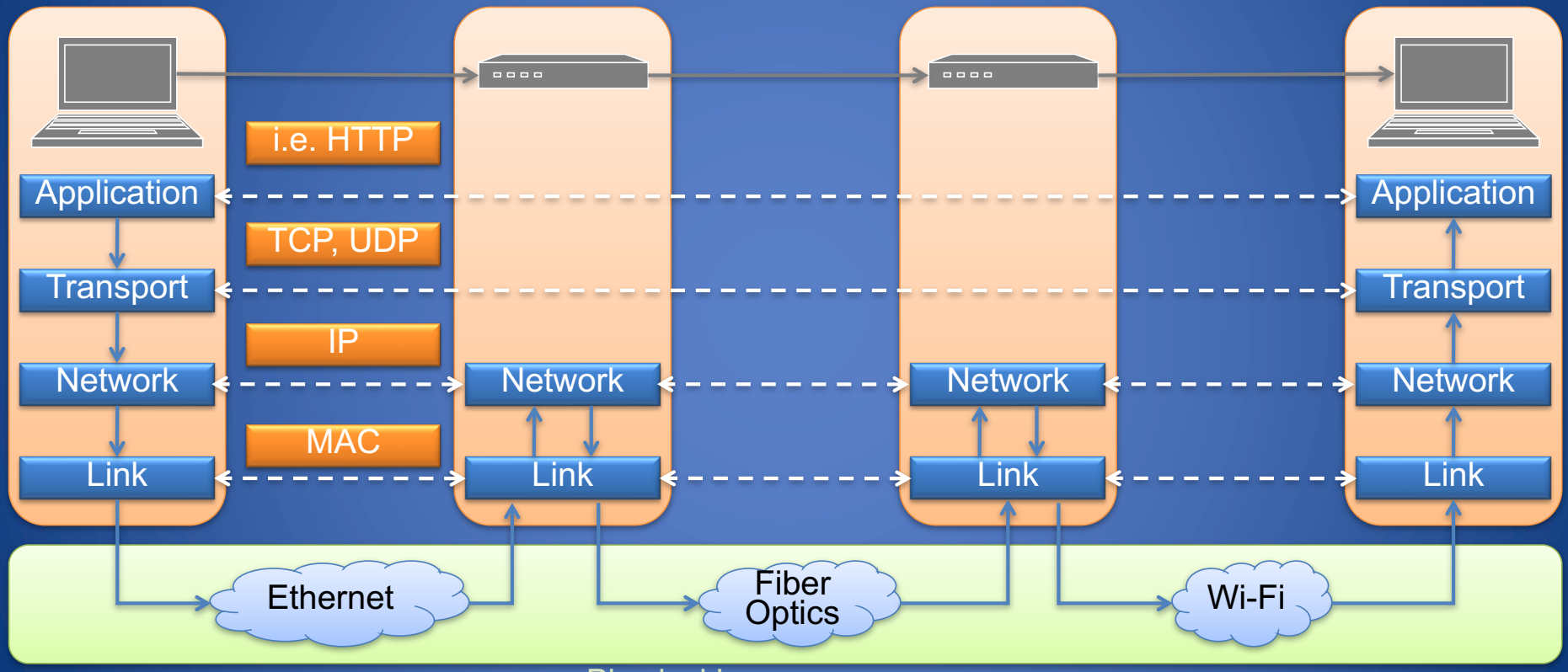
Routing
How does internet actually work?

Why Routing?

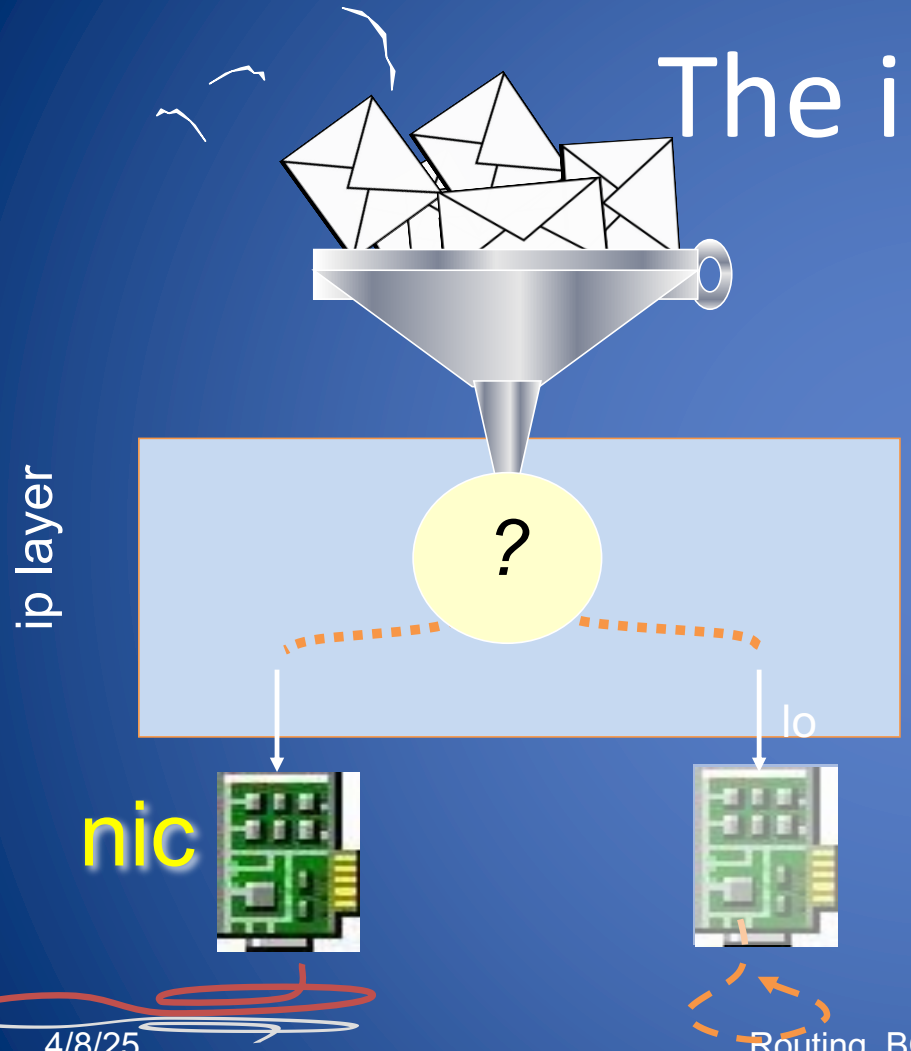
- Reaching a host within a network is a routing problem



Internet Layers

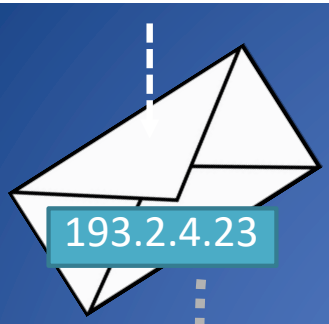


The ip layer



- the ip layer decides which interface an outgoing packet has to be forwarded to
  - regular hosts have at least two interfaces, nic and loopback

routing table



routing table

network	nmask	nexthop	int
200.3.24.0	255.255.255.0	12.0.0.4	eth1
193.2.0.0	255.255.248.0	11.0.0.2	eth0
100.4.5.0	255.240.0.0	11.0.0.3	eth0
0.0.0.0	0.0.0.0	11.0.0.2	eth0

ip layer

eth0 eth1 lo

nic nic

netstat -nr

routing table usage

1100 0001.0000 0010.0000 0100.0001 0111

routing table

network	nmask	nexthop	int
200.3.24.0	255.255.255.0	12.0.0.4	eth1
193.2.0.0	255.255.248.0	11.0.0.2	eth0
100.16.0.0	255.240.0.0	11.0.0.3	eth0
0.0.0.0	0.0.0.0	11.0.0.2	eth0

network	nmask
1100 1000.0000 0011.0001 1000.0000 0000	1111 1111.1111 1111.1111 1111.0000 0000
1100 0001.0000 0010.0000 0000.0000 0000	1111 1111.1111 1111.1111 1000.0000 0000
0110 0100.0001 0000.0000 0000.0000 0000	1111 1111.1111 0000.0000 0000.0000 0000
0000 0000.0000 0000.0000 0000.0000 0000	0000 0000.0000 0000.0000 0000.0000 0000

routing table usage

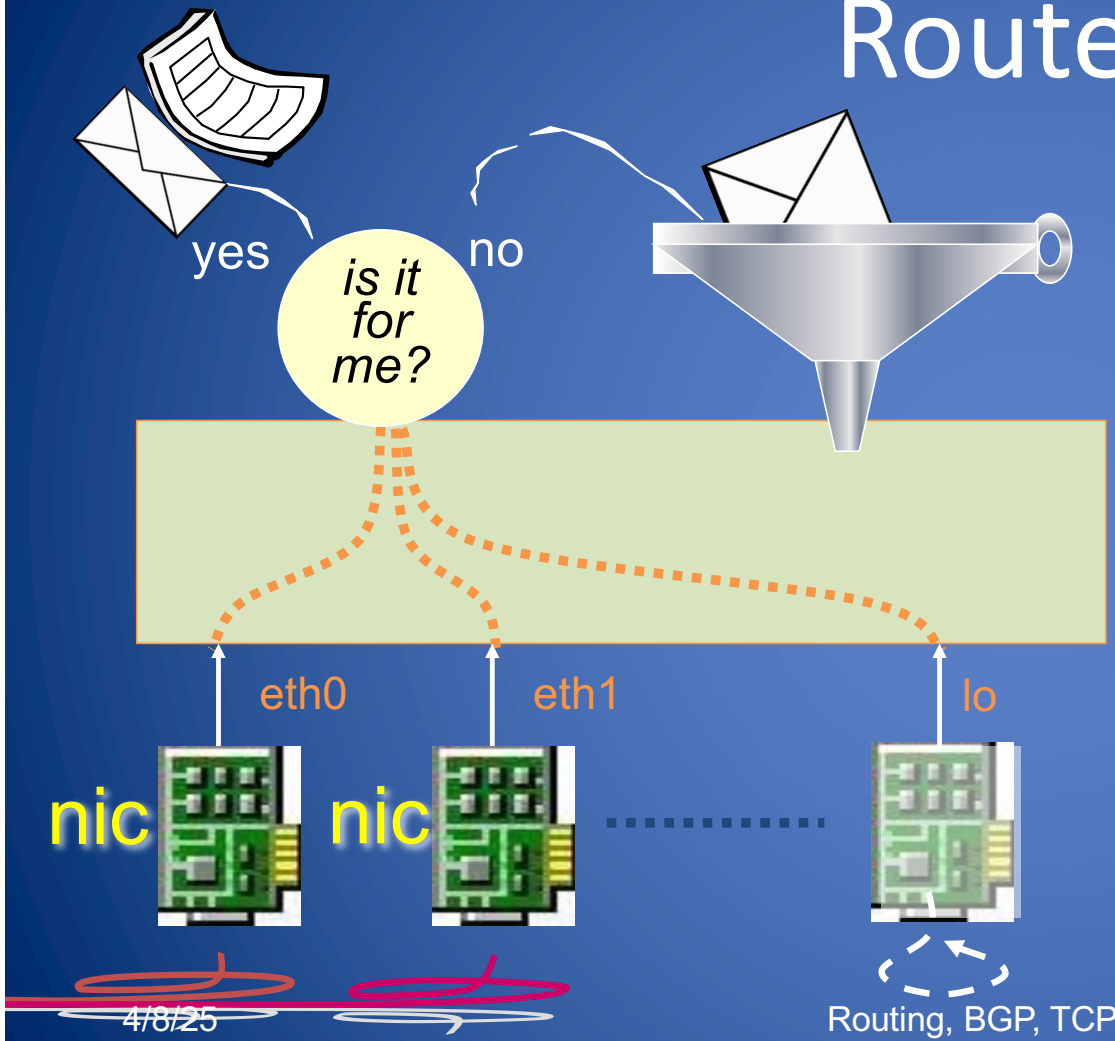
1100 0001.0000 0010.0000 1000.0001 0111

routing table

network	nmask	nexthop	int
200.3.24.0	255.255.255.0	12.0.0.4	eth1
193.2.0.0	255.255.248.0	11.0.0.2	eth0
100.16.0.0	255.240.0.0	11.0.0.3	eth0
0.0.0.0	0.0.0.0	11.0.0.2	eth0

network	nmask
1100 1000.0000 0011.0001 1000.0000 0000	1111 1111.1111 1111.1111 1111.0000 0000
1100 0001.0000 0010.0000 0000.0000 0000	1111 1111.1111 1111.1111 1000.0000 0000
0110 0100.0001 0000.0000 0000.0000 0000	1111 1111.1111 0000.0000 0000.0000 0000
0000 0000.0000 0000.0000 0000.0000 0000	0000 0000.0000 0000.0000 0000.0000 0000

Routers



- a router:
  - has more than one network interface card
  - feeds incoming ip packets (that are not for the router itself) back in the routing process
    - this operation is called *relaying* or *forwarding*
  - also called: *gateway, intermediate-system*

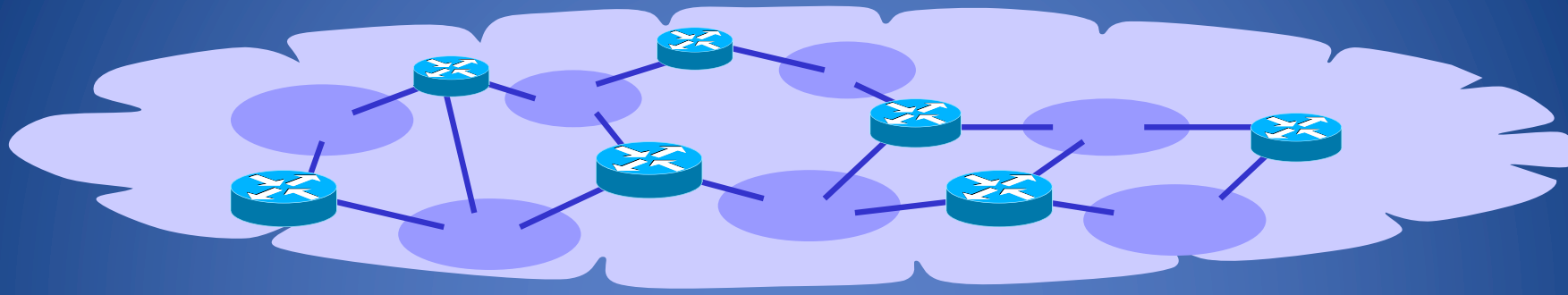
how to update the routing tables?

- Which are the main features that we need?
  1. Global reachability
  2. Dynamic & Automatic update
  3. Fast convergence time
- Different Routing protocols are available
  - Static and manual routing table update is possible but usually not practical

Routing protocols

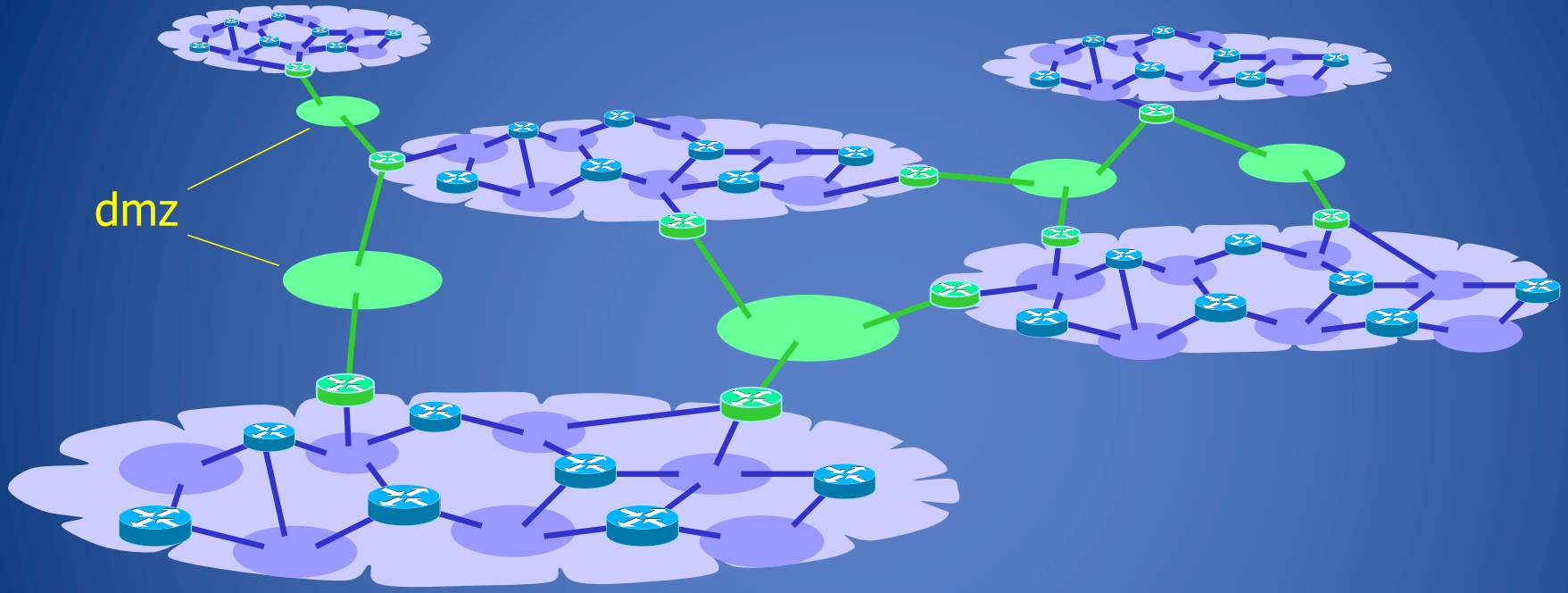
- They fall into two main cathegories:
  - link-state routing protocols
    - approach: talk about your neighbors to everyone
    - each router reconstructs the whole network graph and computes a shortest path tree to all destinations
    - examples: IS-IS, OSPF
  - distance-vector routing protocols
    - approach: talk about everyone with your neighbors
    - update your routing information based on what you hear
    - examples: RIP

Why interdomain routing?



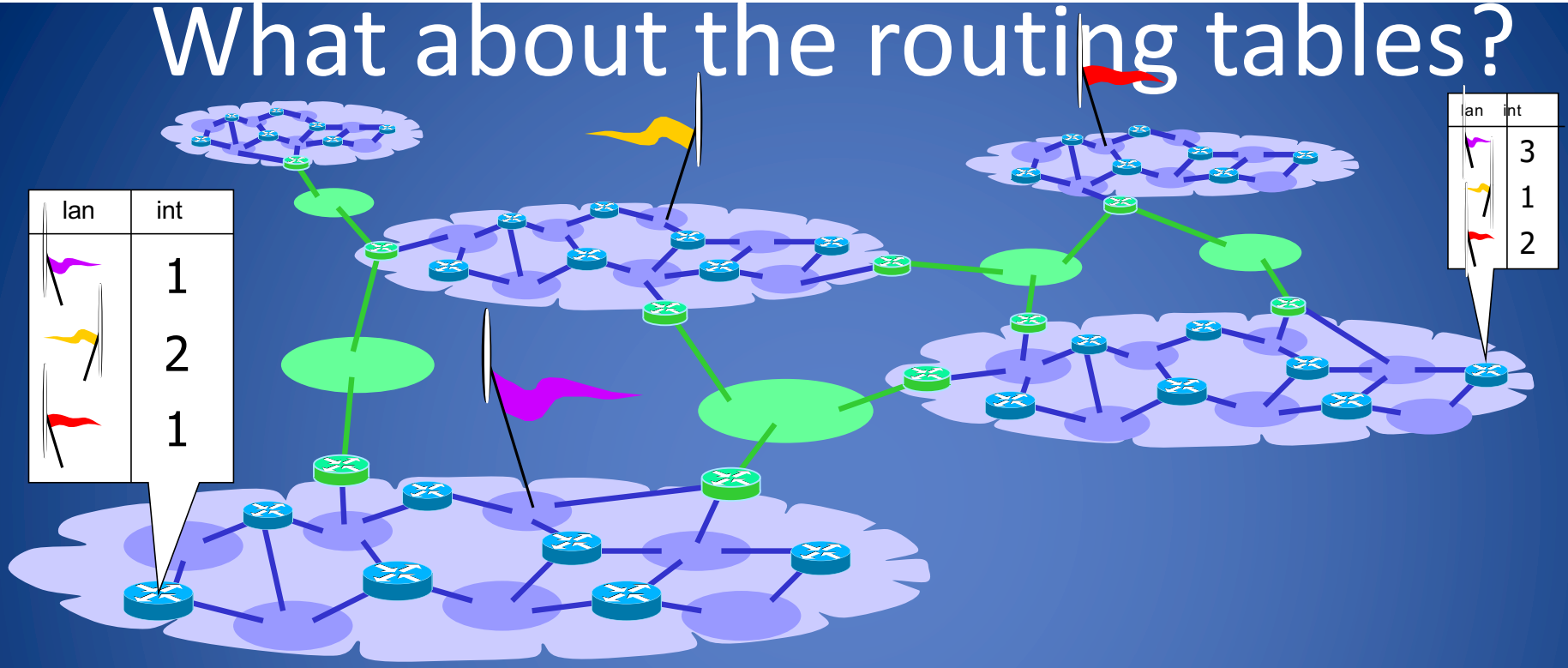
- Each organization is a collection of routers and Lan under a single administration
- A routing algorithm may be chosen to automatically update the routing tables

Why interdomain routing?



- when several organizations join to form the internet they have to set up links between them
  - the added lan are called “demarcation zones”

What about the routing tables?



- in order to have global connectivity:
  - each router must have a routing entry (possibly the default one) that matches the destination address of the packet
  - this should be true for packets to be delivered locally as well as for packets to be delivered to remote lans

Border Gateway Protocol (BGP)

- The routing protocol that makes the Internet work
  - A path vector protocol (similar to a distance vector)
- Used by:
  - customers connected to an Internet Service Provider (ISP) or several ISPs
  - transit providers
  - ISPs that exchange traffic at an Internet eXchange Point (IXP) or Neutral Access Point (NAP)
  - customers with very large networks

Autonomous System

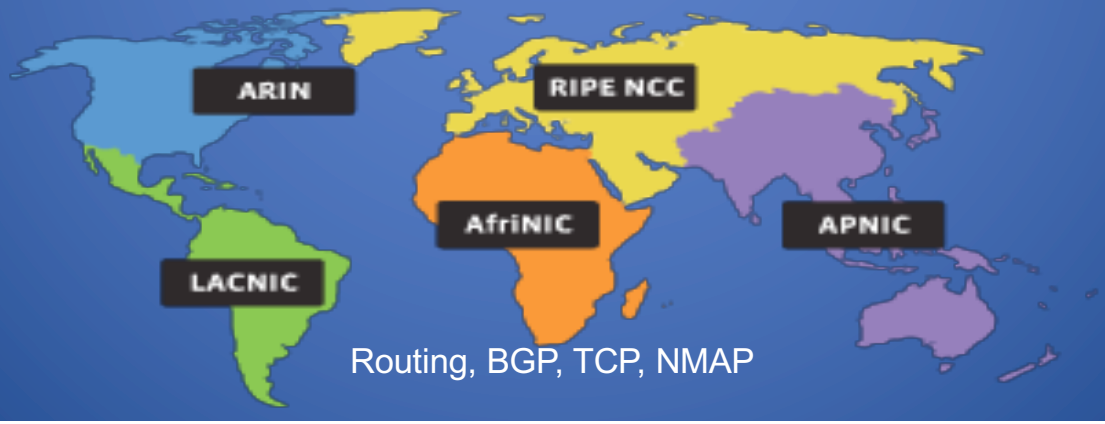
- autonomous systems (ASes) are the cornerstones of BGP
  - used to uniquely identify networks with a common routing policy
  - usually under single ownership, trust and administrative control
- each AS is identified by an *autonomous system number* (asn): 32 bit integer
- two ranges
  - 0-65535 (original 16-bit range)
  - 65536-4294967295 (32-bit range - RFC4893)

Autonomous System Number

- you may ask an asn to:
  - global asn - to your *regional internet registry* (rir): ripe, arin, apnic, etc.
  - private asn - to your upstream isp
- see also:

  www.iana.org/assignments/as-numbers

  bgpview.io/reports/countries

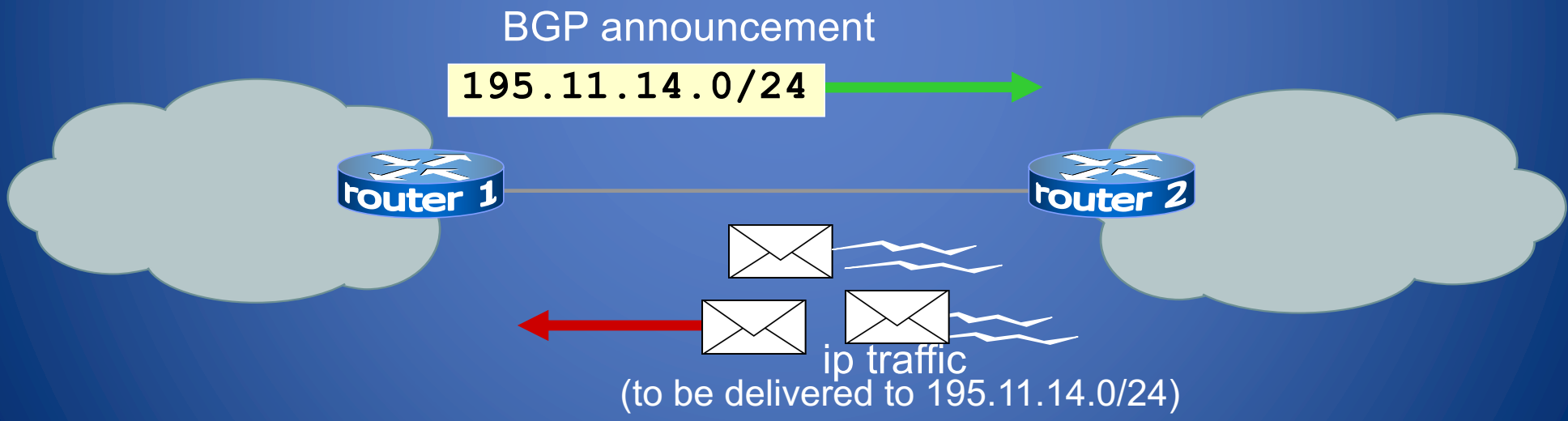


BGP peering

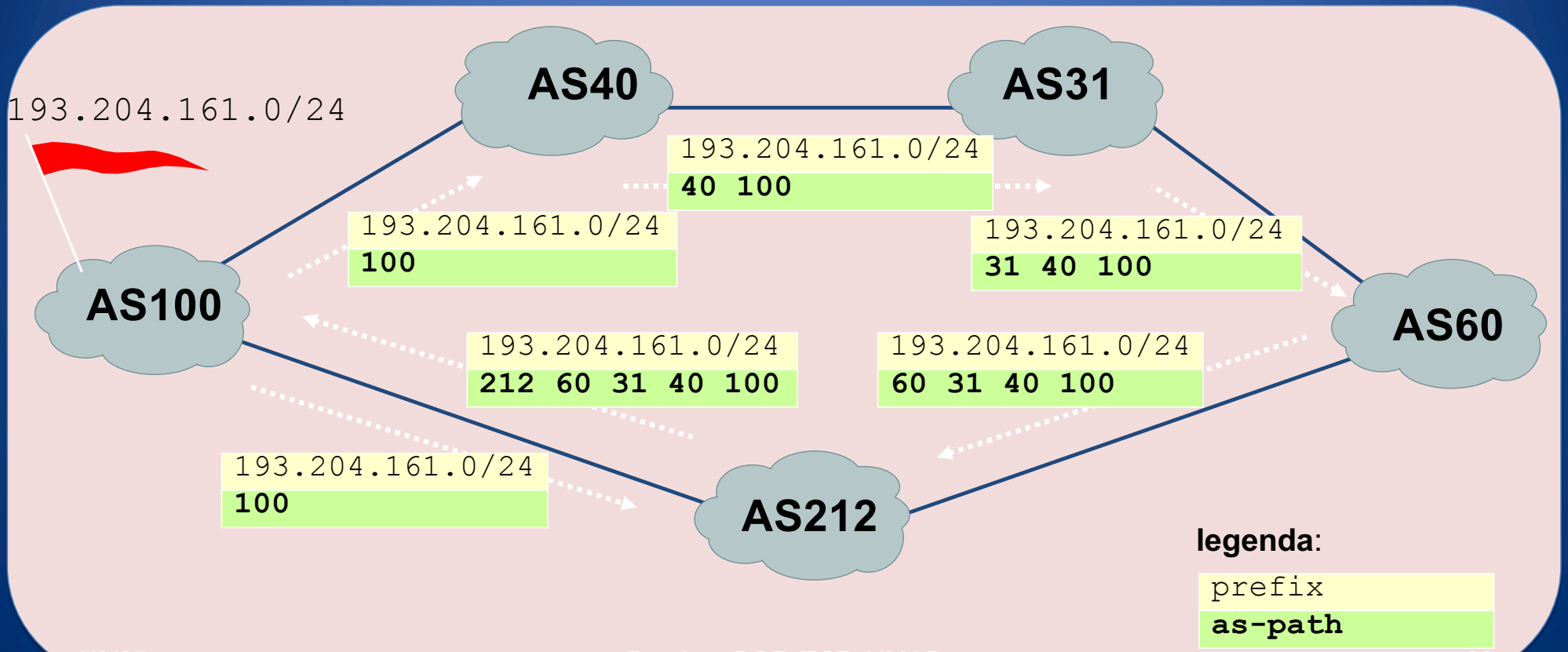
- BGP allows routers to exchange information only if a *peering* session is up
- a BGP peering is the tcp connection (port 179) over which routing information will be exchanged

Announcements and traffic flows

- BGP allows a router to offer connectivity to another router
- “offering connectivity” means “promising the delivery to a specific destination”

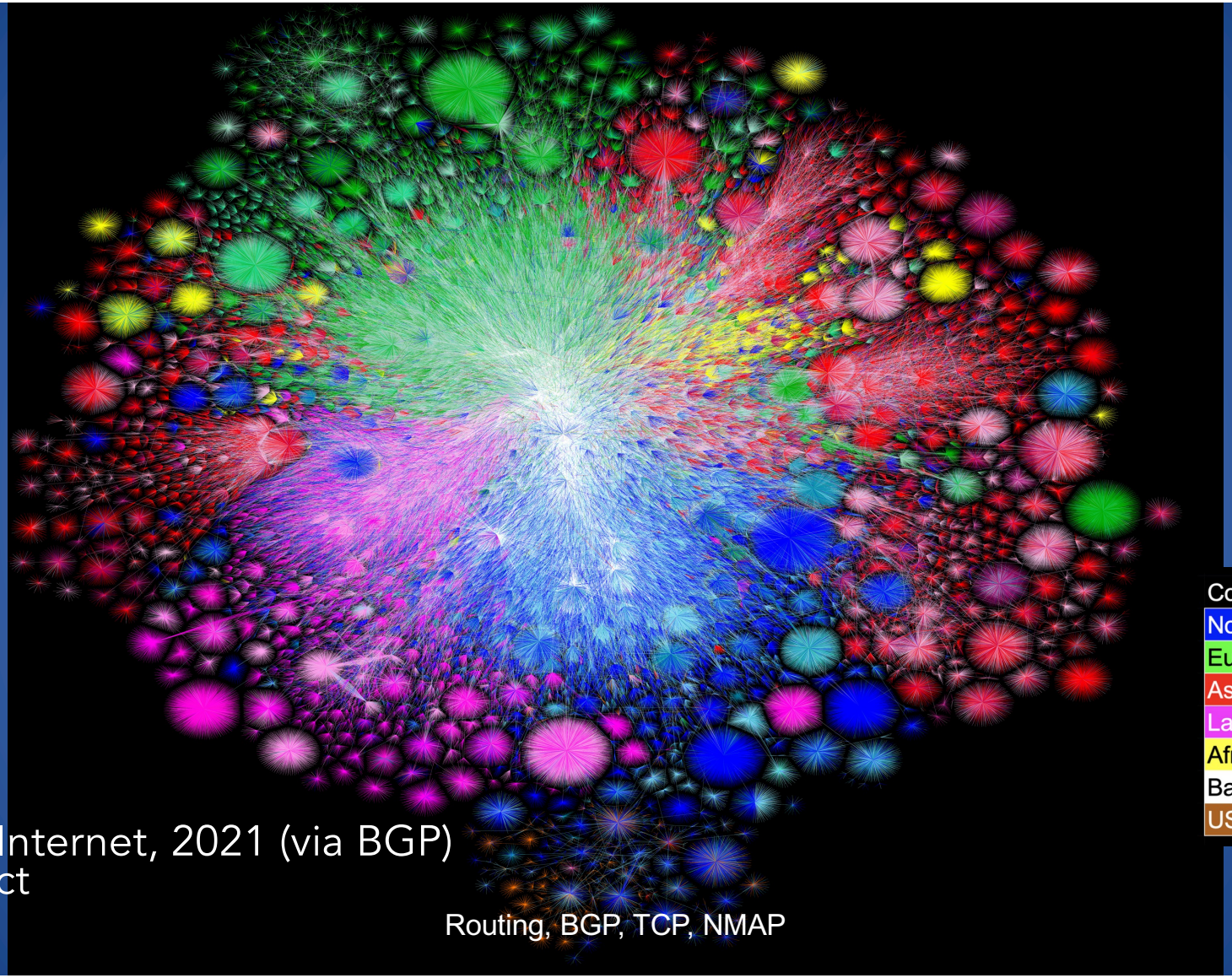


attributes: AS-path



Looking Glass Server (Demo)

- Provides backbone routing and network efficiency information
  - BGP, Traceroute, and Ping
    - tools that are possible to use with the same transparency that users on ISP network receive directly
- Demo: Hurricane Electric
  - https://bgp.he.net - https://lg.he.net/
  - https://bgp.he.net/super-lg/#128.148.0.0/21



Map of the Internet, 2021 (via BGP)
OPTE project

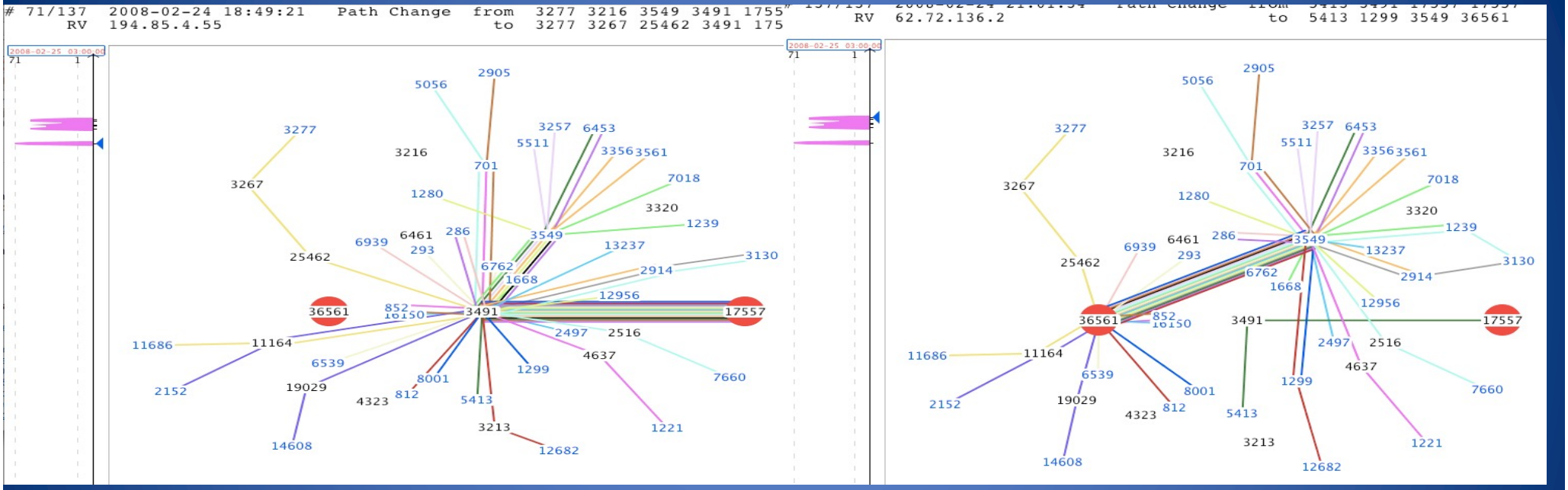
BGP Vulnerabilities

- In the original version BGP has no security mechanisms:
  - No encryption: Eavesdropping
  - No timestamp: Replaying
  - No signature: Hijacking
  - Selective dropping
- Possible attacks:
  - Injecting false information into the global routing database
  - Reroute traffic to perform a Man-in-the-Middle (MITM) attack
  - Trying to create a Denial of Service (DoS) like a black hole in the network

A big incident

- February 2008 Pakistan Telecom (PT) would like to block Youtube access from Pakistan
  - PT falsely informed that through this company there was the most directed way to reach Youtube
- Soon over 2/3 of the Internet was not able to reach Youtube for a couple of hours
- A Routing problem…

YouTube Internet Hijacking In Pakistan



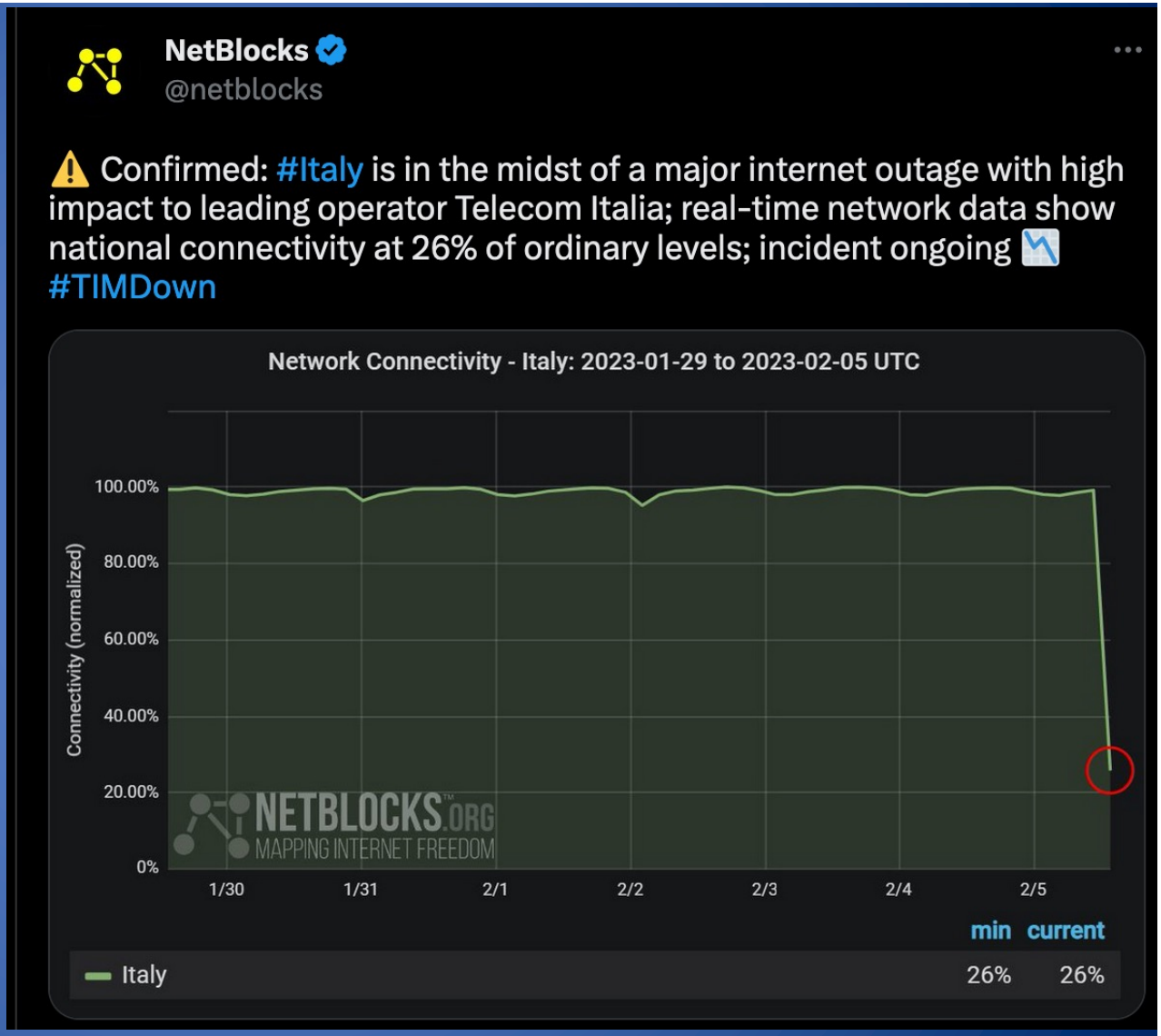
AS 17557 Pakistan, AS 36561 Youtube

[Ripe description using bgplay tool developed at Roma Tre University: https://www.youtube.com/watch?v=IzLPKuAOe50]

TIMDown

Stopped the communication for 6 hours on 2/5/23

Probably a human error due to a bad DDOS configuration



Transport Layer (Ports, TCP, UDP)

The Transport Layer

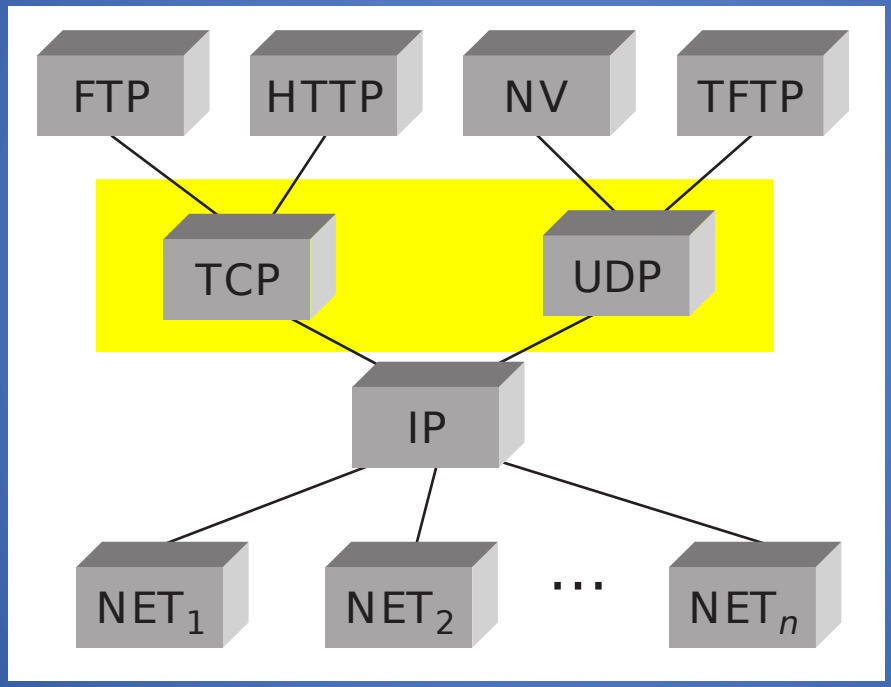
Network layer: moving data between hosts

Transport layer: Abstraction for getting data data to different *applications* on a host

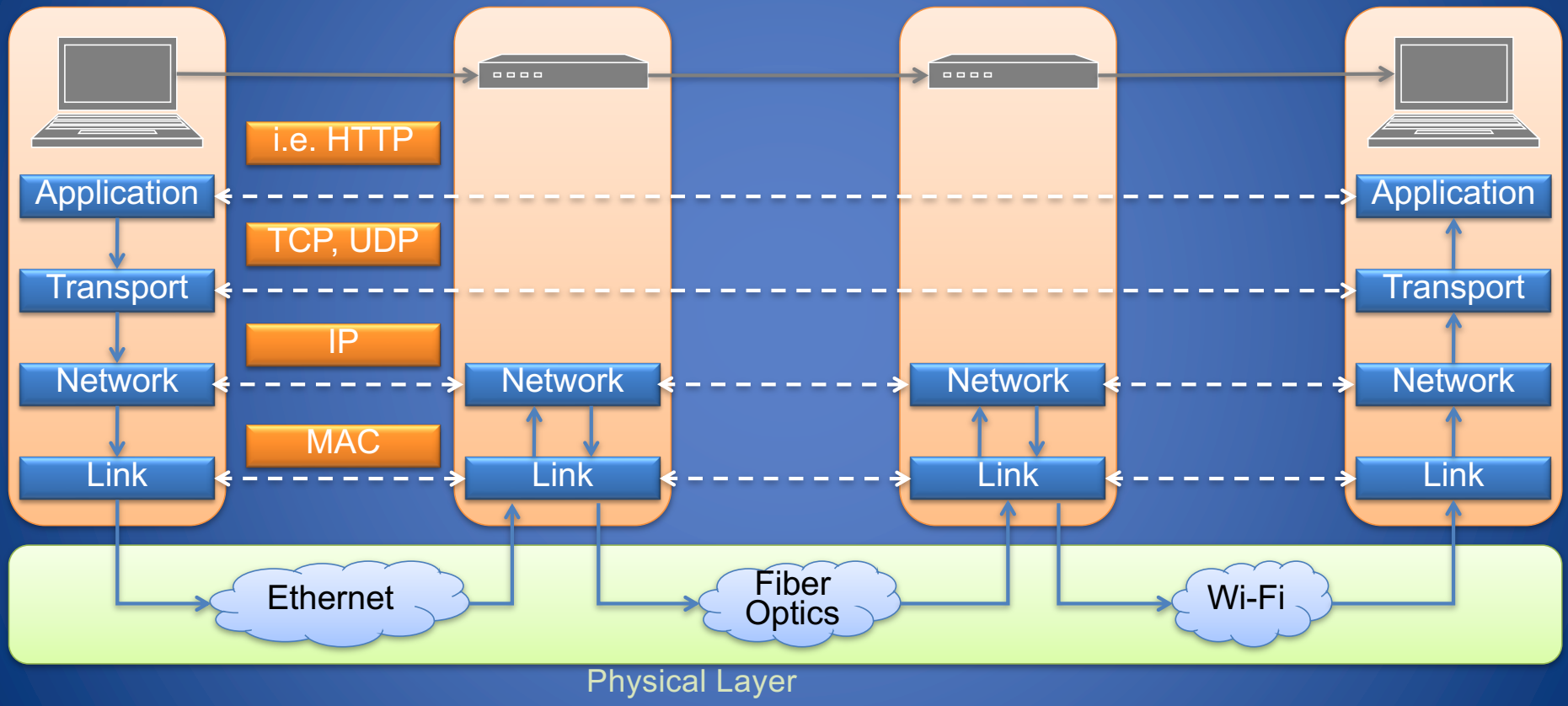
- Multiplexing multiple connections at the same IP with port numbers
- Series of packets => stream of data/messages
- May provide: reliable data delivery

Two key protocols: TCP, UDP

Transport Layer



Internet Layers



What's a port number?

- 16-bit unsigned number, 0-65535
- Ports define a communication *endpoint*, usually a process/service on a host
- OS keeps track of which ports map to which applications

Port numbering

- port < 1024: "Well known port numbers"
- port >= 20000: "ephemeral ports", for general app. use

Some common ports

Port	Service
20, 21	File Transfer Protocol (FTP)
22	Secure Shell (SSH)
23	Telnet (pre-SSH remote login)
25	SMTP (Email)
53	Domain Name System (DNS)
67, 68	DHCP
80	HTTP (Web traffic)
443	HTTPS (Secure HTTP over TLS)

How ports work

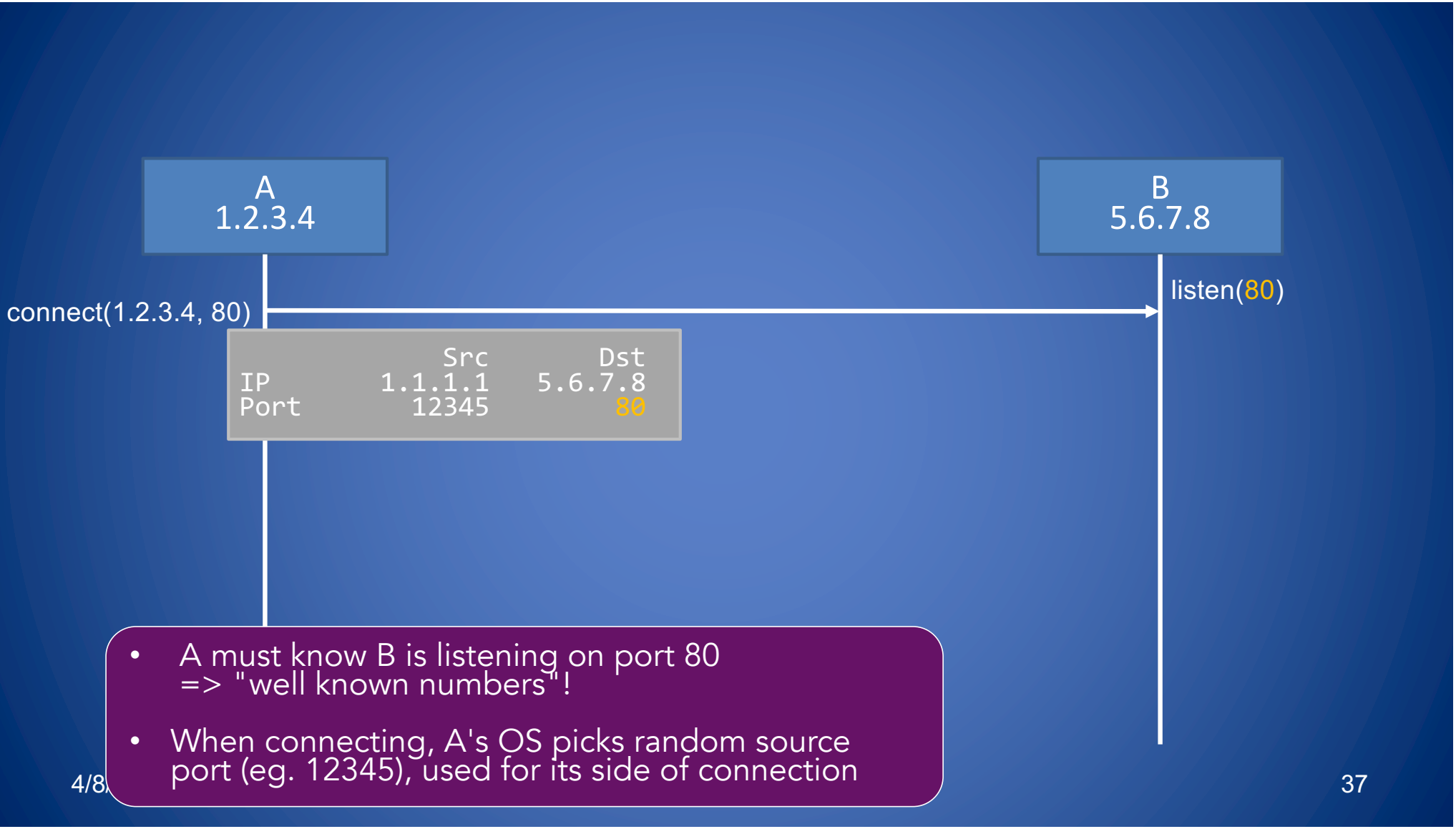
Two modes:

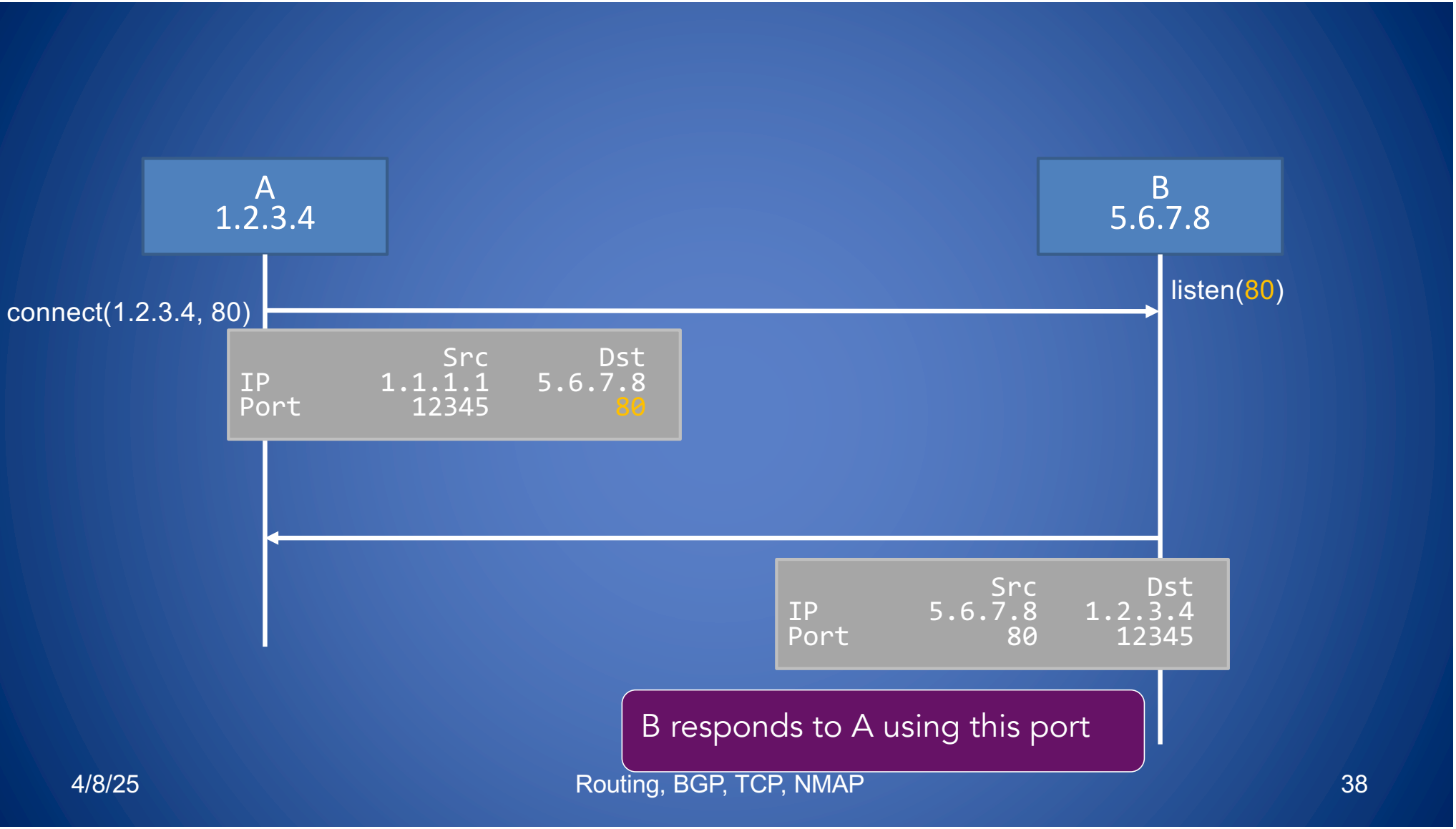
- Applications "listen on" or "bind to" a port to wait for new connections

  => Example: webserver listens on port 80 or 443

- Hosts make connections to a particular IP and port

  => Example: client connects to <webserver IP>, port 80 or 443 (eg. 1.2.3.4:80)


A
1.2.3.4
B
5.6.7.8
listen(80)
connect(1.2.3.4, 80)
Src Dst
IP 1.1.1.1 5.6.7.8
Port 12345 80
• A must know B is listening on port 80
=> "well known numbers"!
• When connecting, A's OS picks random source port (eg. 12345), used for its side of connection


A
1.2.3.4
B
5.6.7.8
listen(80)
connect(1.2.3.4, 80)
Src Dst
IP 1.1.1.1 5.6.7.8
Port 12345 80
Src Dst
IP 5.6.7.8 1.2.3.4
Port 80 12345
B responds to A using this port

Sockets

OS keeps track of which application uses which port

Two types:

- Listening ports
- Connections between two hosts (src/dst port)

Socket: OS abstraction for a network connection, like a file descriptor

Table maps: port => socket

Want to know more? Take CS1680!

Netstat

```
deemer@vesta ~/Development % netstat -an
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  10.3.146.161.51094     104.16.248.249.443     ESTABLISHED
tcp4       0      0  10.3.146.161.51076     172.66.43.67.443       ESTABLISHED
tcp6       0      0  2620:6e:6000:900.51074 2606:4700:3108::.443   ESTABLISHED
tcp4       0      0  10.3.146.161.51065     35.82.230.35.443       ESTABLISHED
tcp4       0      0  10.3.146.161.51055     162.159.136.234.443    ESTABLISHED
tcp4       0      0  10.3.146.161.51038     17.57.147.5.5223       ESTABLISHED
tcp6       0      0  *.22                   *.*                    LISTEN
tcp4       0      0  *.51036                *.*                    LISTEN
tcp4       0      0  127.0.0.1.9999         *.*                    LISTEN
```

`netstat -an`: Show all connections

`netstat -lnp`: Show listening ports + applications using them (as root)

Macosx: lsof -nP -iTCP -sTCP:LISTEN

Why do we care?

Ports define what services are exposed to the network

- Open port: can send data to application (reconnaissance, attacks, ...)

- OS and network hardware can monitor port numbers
  - Make decisions on how to filter/monitor traffic

Transport Layer

- The transport layer supports one or more of the following features
  A. Reliable data transfer (resending of dropped packets)
  B. In-order delivery of segments of file or media stream
  C. Congestion control (request longer/shorter segments)
  D. Ability to distinguish multiple applications on same host via ports (16-bit numbers)
- The main transport layer protocols are
  - UDP (supports B, D)
  - TCP (supports A, B, C, D)

User Datagram Protocol (UDP)

- Stateless, unreliable transport-layer protocol
- Can distinguish multiple concurrent applications on a single host
- No delivery guarantees or acknowledgments
  - Efficient
  - Suitable for audio/video streaming and voice calls
  - Unsuitable for file transmission and text messaging

Transmission Control Protocol (TCP)

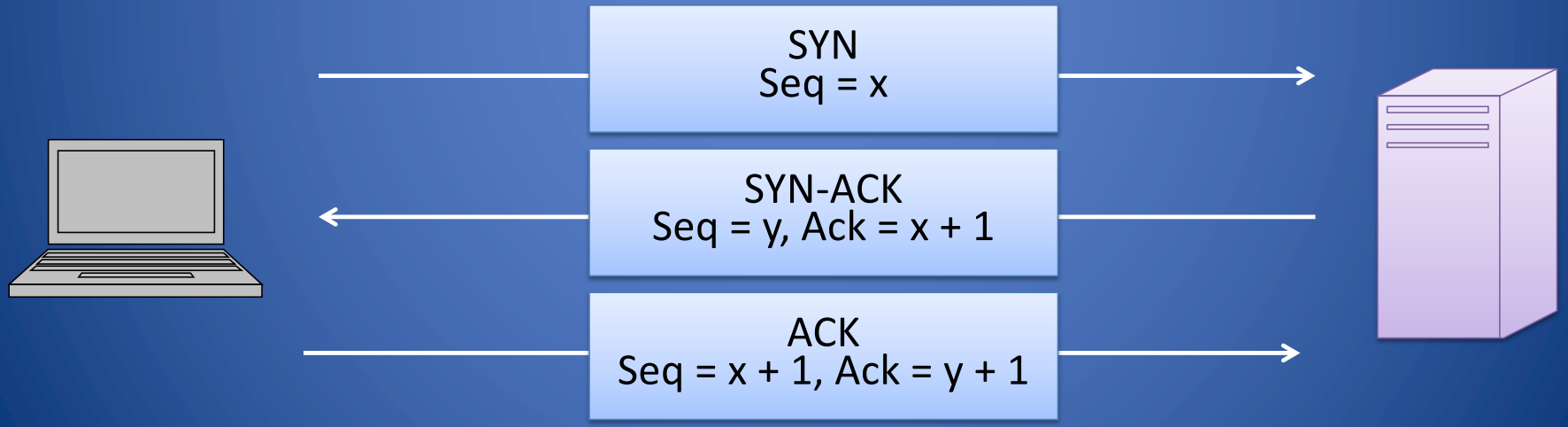
- Stateful protocol for reliable data transfer, in-order delivery of messages and ability to distinguish multiple applications on same host
  - HTTP and SSH are built on top of TCP
- TCP packages a data stream it into segments transported by IP
  - Order maintained by marking each packet with sequence number
  - Every time TCP receives a packet, it sends out an ACK to indicate successful receipt of the packet
- TCP generally checks data transmitted by comparing a checksum of the data with a checksum encoded in the packet

TCP Packet Format

Bit Offset	0-3	4-7	8-15	16-18	19-31
0	**Source Port**			**Destination Port**	
32	**Sequence Number**				
64	**Acknowledgment Number**				
96	Offset	Reserved	Flags	**Window Size**	
128	Checksum			Urgent Pointer	
160	Options				
>= 160	Payload				

Establishing TCP Connections

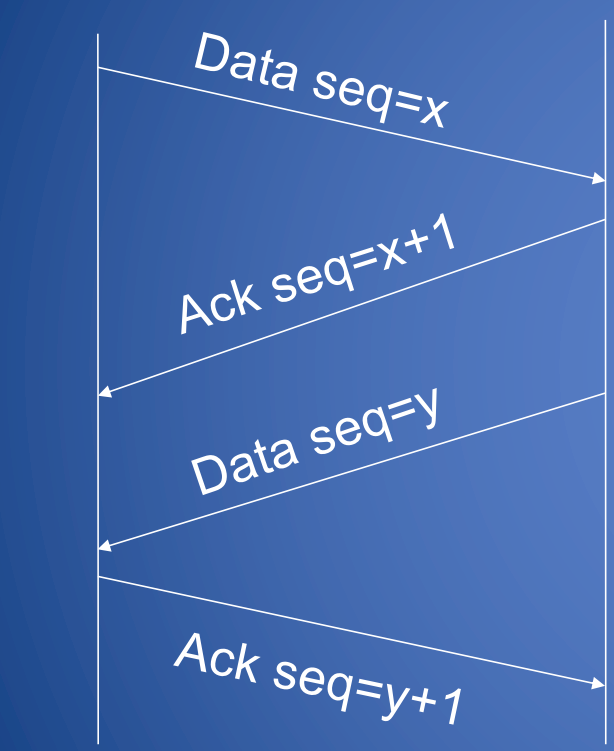
- TCP connections are established through a three-way handshake
- The server generally is a passive listener, waiting for a connection request
- The client requests a connection by sending out a SYN packet
- The server responds by sending a SYN/ACK packet, acknowledging the connection
- The client responds by sending an ACK to the server, thus establishing connection

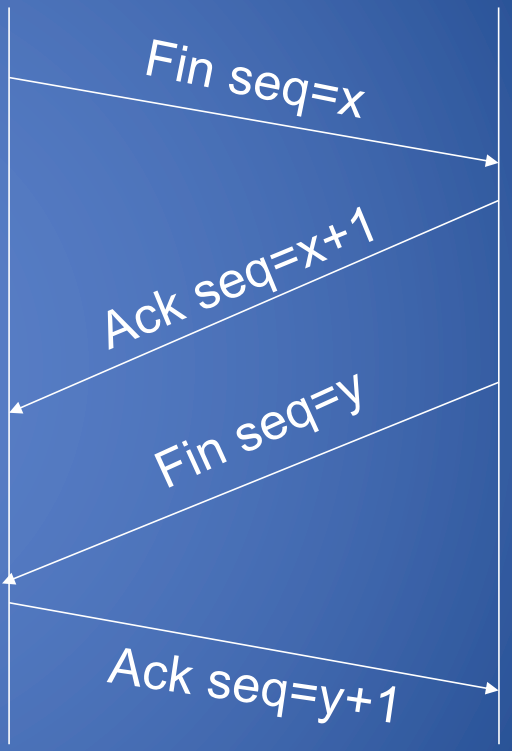


TCP Data Transfer

- The three way handshake initializes sequence numbers for the request and response data streams
- The TCP header includes a 16 bit checksum of the payload and parts of the header, including source and destination
- Acknowledgment or lack thereof is used by TCP to keep track of network congestion and control flow
- TCP connections are cleanly terminated with a 4-way handshake
  - The client which wishes to terminate the connection sends a FIN message to the other client
  - The other client responds by sending an ACK
  - The other client sends a FIN
  - The original client now sends an ACK, and the connection is terminated

TCP Data Transfer and Teardown





Why do we care?

```
deemer@vesta ~/Development % netstat -an
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp6       0      0  *.22                   *.*                    LISTEN
                                  . . .
```

If a listening port is open, you can send data to an application

=> Defines attack surface on network!

Implications for:

- How to find vulnerable hosts/services
- How we protect them

Port scanning

What can we learn if we just start connecting to well-known ports?

- Applications have common port numbers
- Network protocols use well-defined patterns

```
deemer@vesta ~/Development % nc <IP addr> 22
SSH-2.0-OpenSSH_9.1
```

netcat or nc: is a command-line utility that reads and writes data across network connections using the TCP or UDP protocols

Port scanners: try to connect to lots of ports, determine available services, find vulnerable services...

nmap

nmap: Widely-used network scanning tool

- Scan ranges of IPs, look for specific open ports
- Scan many ports on specific hosts, learn about available services
- Lots of extensions/scripts...

```
$ nmap -sV -A 172.17.48.44
Nmap scan report for 172.17.48.25
Host is up (0.00065s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE      VERSION
22/tcp   open  ssh          OpenSSH 6.2 (protocol 2.0)
88/tcp   open  kerberos-sec Heimdal Kerberos (server time: 2023-04-25 15:04:20Z)
5900/tcp open  vnc          Apple remote desktop vnc
Service Info: OS: Mac OS X; CPE: cpe:/o:apple:mac_os_x
```

OS/Service discovery

Different OSes use different defaults in packet headers

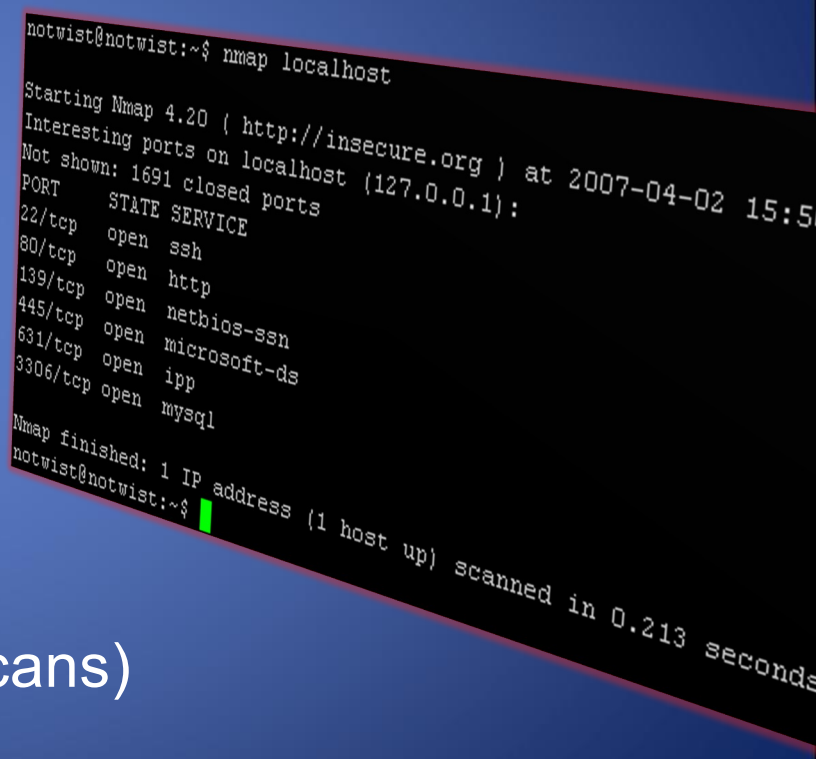
=> Can use for detection!

	linux 2.4	linux 2.6	openbsd	MACOS X	windows	
ttl	64	64	64	64	128	
packet length	60	60	64	64	48	
initial windows	5840	5840	16384	9000	16384	
mss	512	512	1460	1460	1460	
ip id	0	random	random	random	increment	
enabled tcp opt	MNNTNW	MNNTNW	M	M	MNW	
timestamp inc.	100hz	1000hz	unsupported	unsupported	100Hz	
sack	OK	OK	OK	OK	OK	
SYN attempts	5	5	4	3	3	

Enumeration with Nmap (Network Mapper)

Port Division
- open, closed, filtered, unfiltered, open|filtered and closed|filtered

Scanning techniques
-sP (Skip port scan – like a Ping)
-sS (TCP SYN scan)
-sT (TCP connect() scan)
-sU (UDP scans)
-sA (TCP ACK scan)
-sW (TCP Window scan)
-sM (TCP Maimon scan)
--scanflags (Custom TCP scan)
-sI <zombie host[:probeport]> (Idlescan)
-sO (IP protocol scan)
-sN; -sF; -sX (TCP Null, FIN, and Xmas scans)
-b <ftp relay host> (FTP bounce scan)



Large-scale port scanning

Can reveal lots of open/insecure systems!

Examples:

- shodan.io
  - Indirect scanning: the target does not know that you are scanning
- Open webcam viewers...
- …

Disclaimer

- Network scanning is often very easy to detect
- Unless you are the owner of the network, it's seen as malicious activity
- If you scan the whole Internet, the whole Internet will get mad at you (unless done very politely)
- Do NOT try this on the Brown network. We warned you.