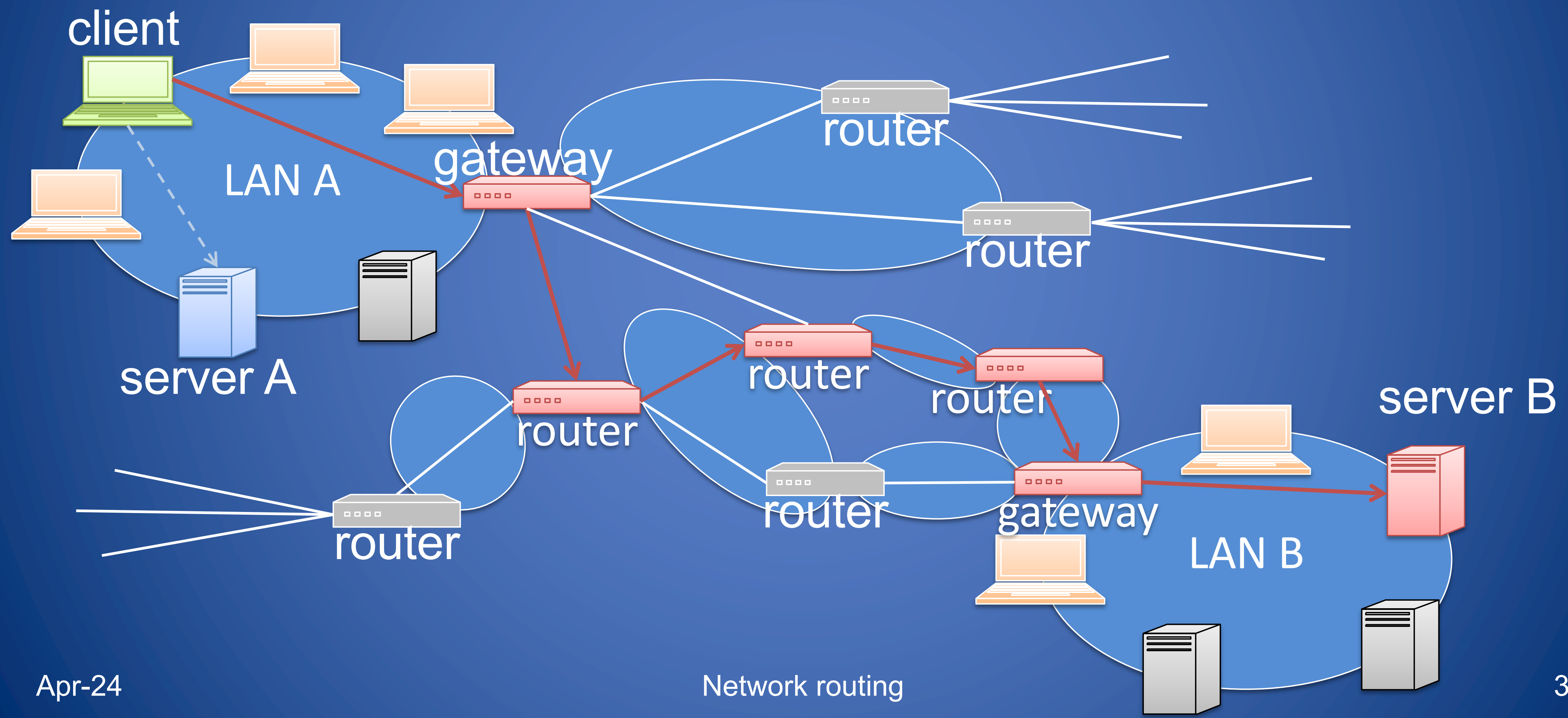
Networks III
DoS, Routing, Transport Layer

CS 1660: Introduction to Computer Systems Security

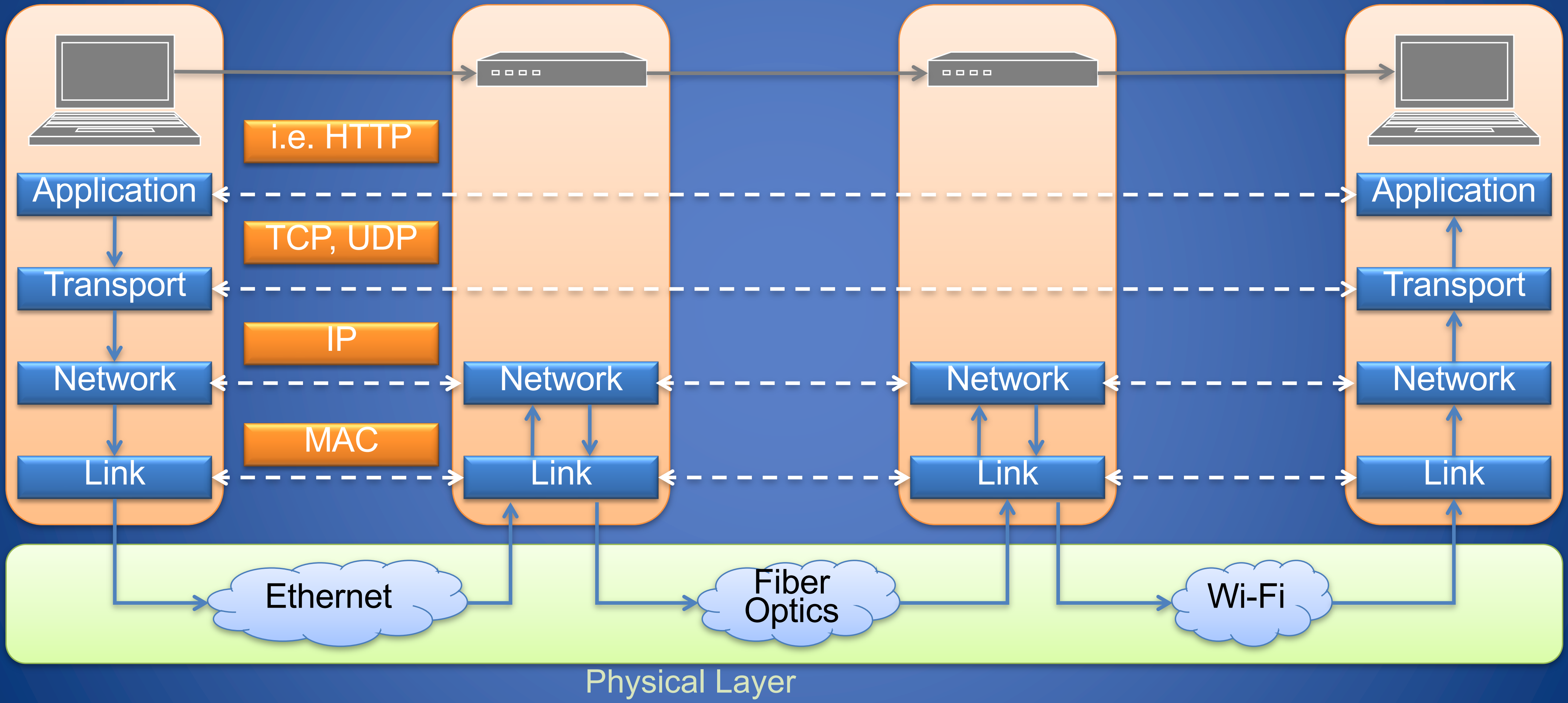
Routing
How does internet actually work?

Why Routing?

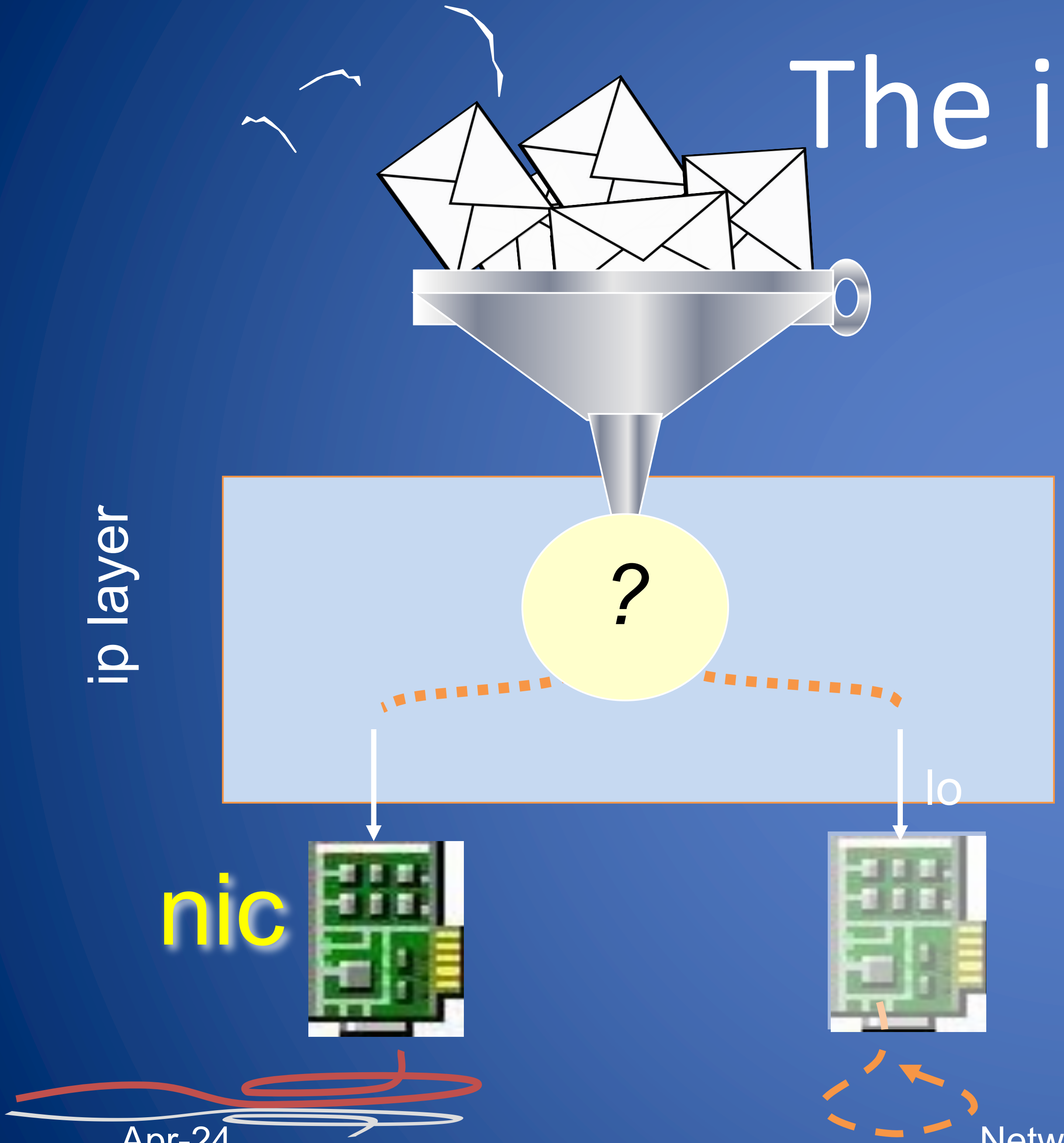
- Reaching a host within a network is a routing problem



Internet Layers

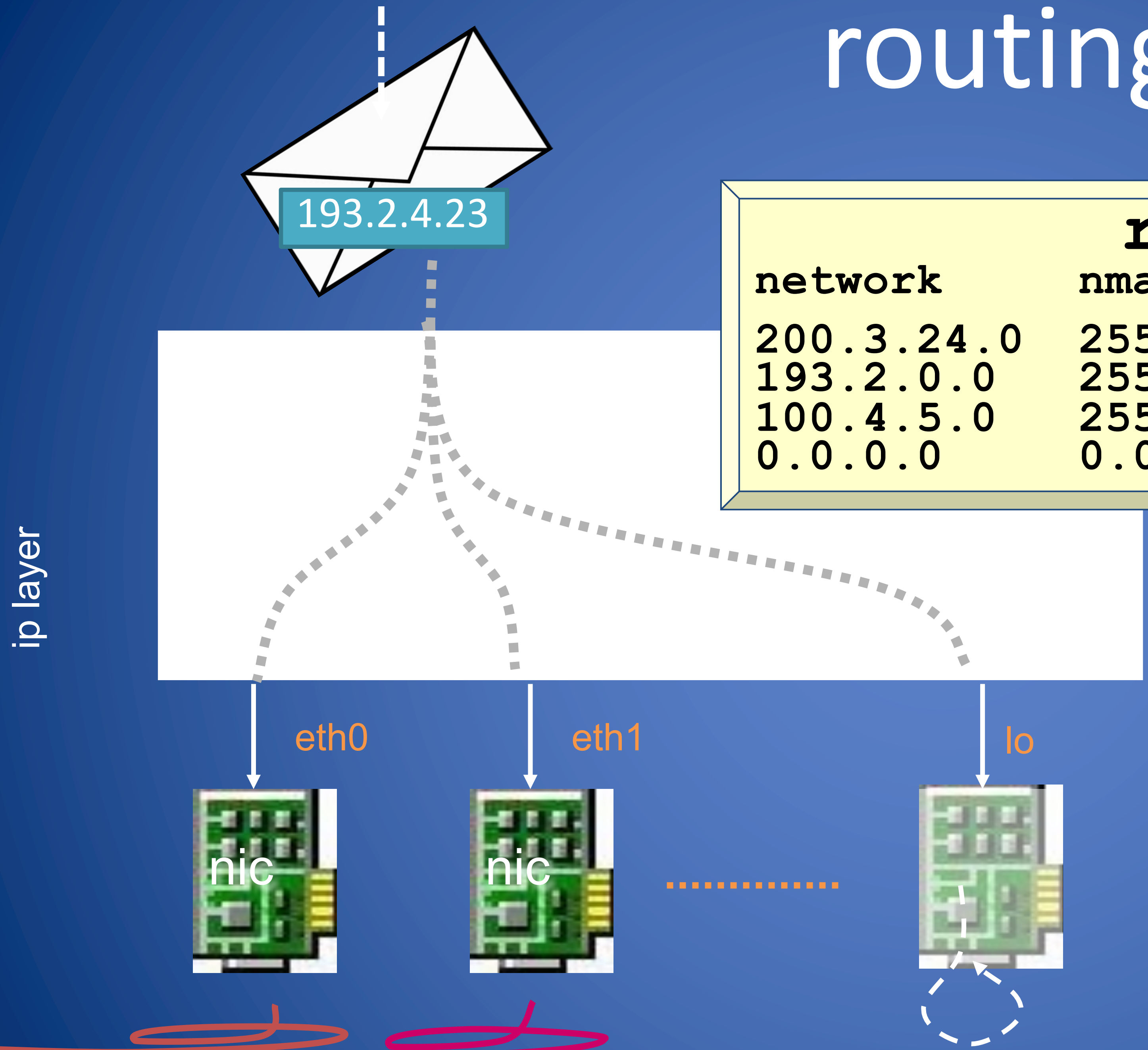


The ip layer



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

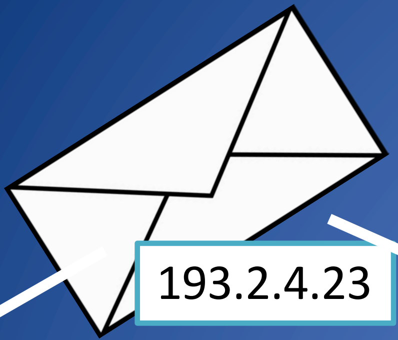
routing table



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

routing table usage



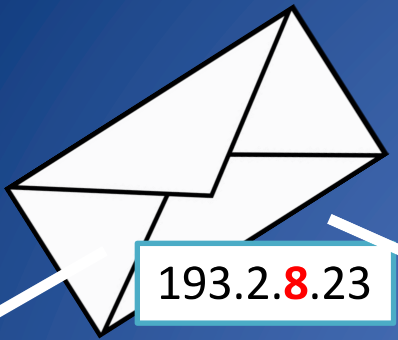
1100 0001.0000 0010.0000 0100.0001 0111

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



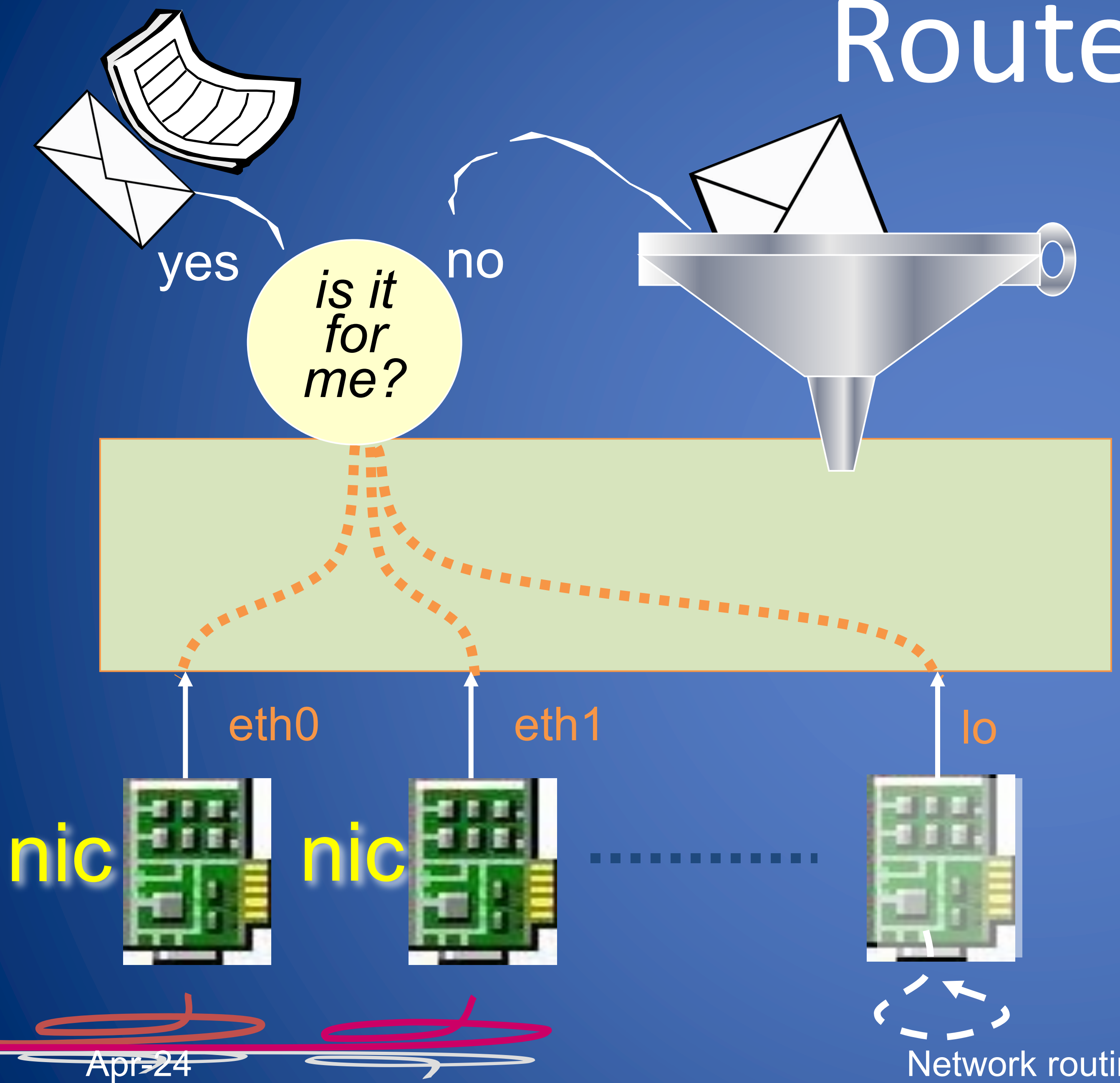
1100 0001.0000 0010.0000 1000.0001 0111

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*

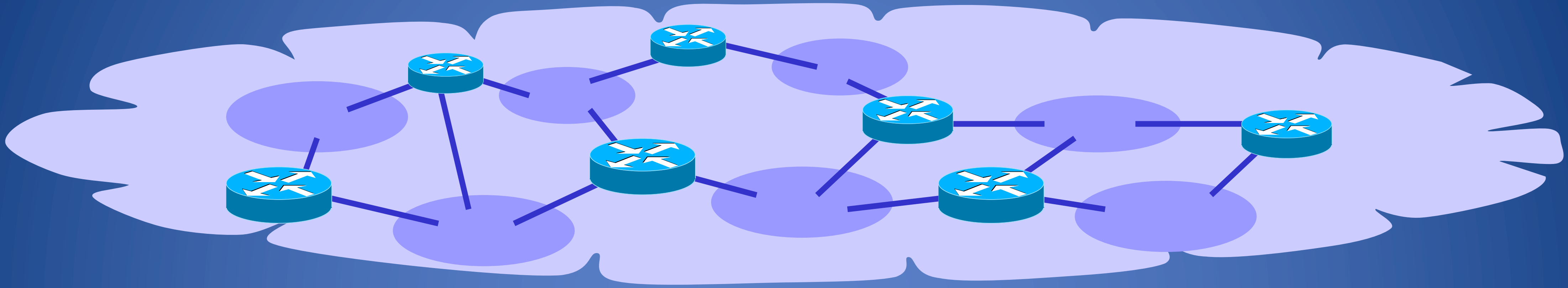
how to update the routing tables?

- 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

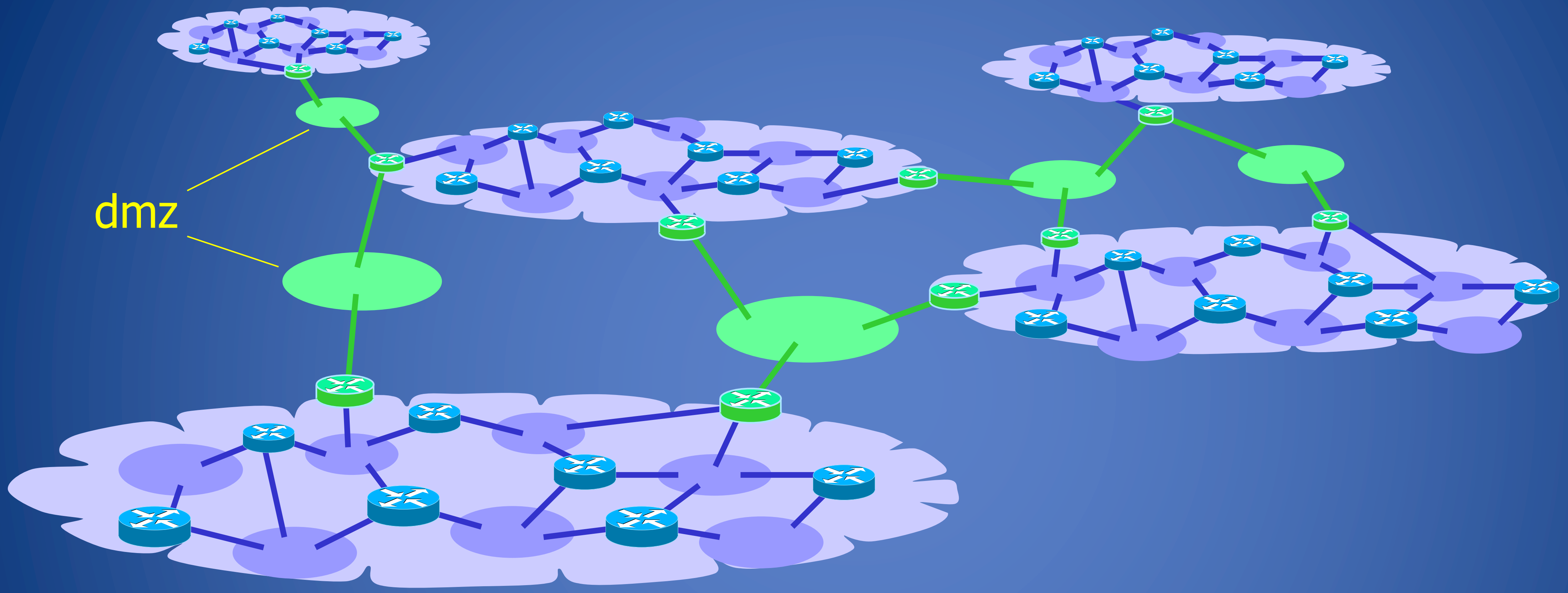
- They fall into two main cathegories:
  - link-state routing protocols
    - approach: talk about your neighbors to everyone
    - each router reconstructs the whole network graph and computes a shortest path tree to all destinations
    - examples: IS-IS, OSPF
  - distance-vector routing protocols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

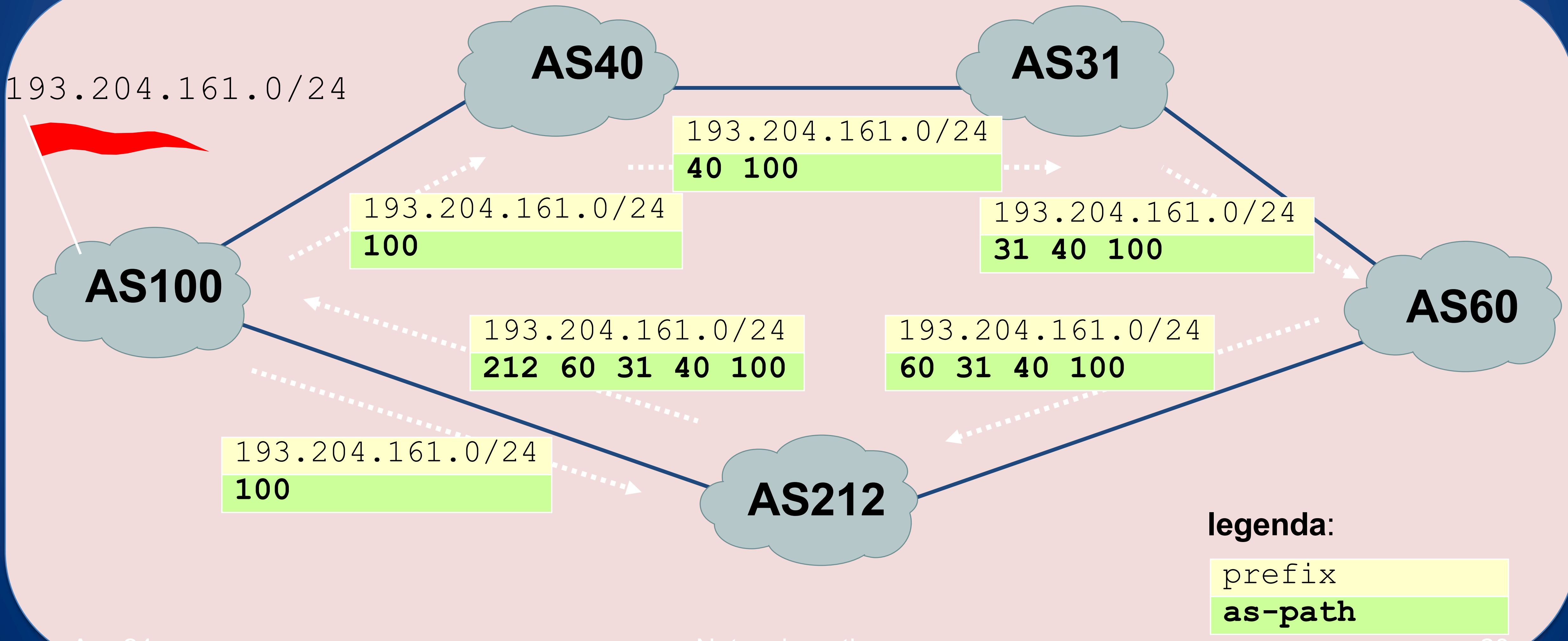
BGP peering

- 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
  - http://bgp.he.net - http://lg.he.net/
  - https://bgp.he.net/super-lg/#128.148.0.0/21

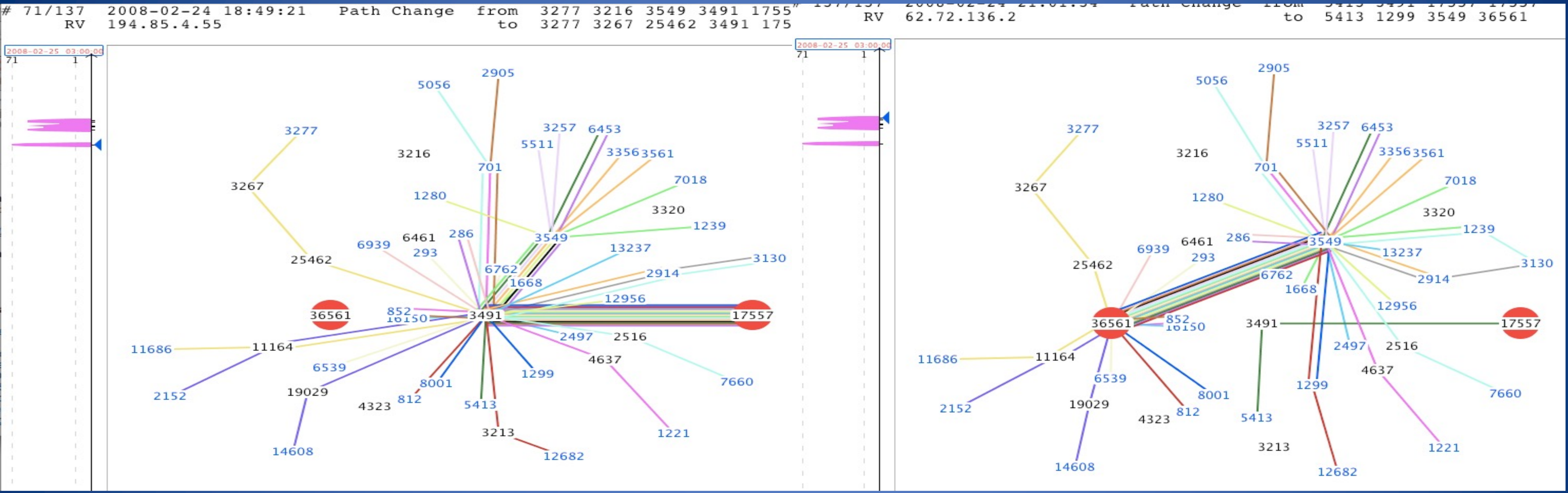
BGP Vulnerabilities

- 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



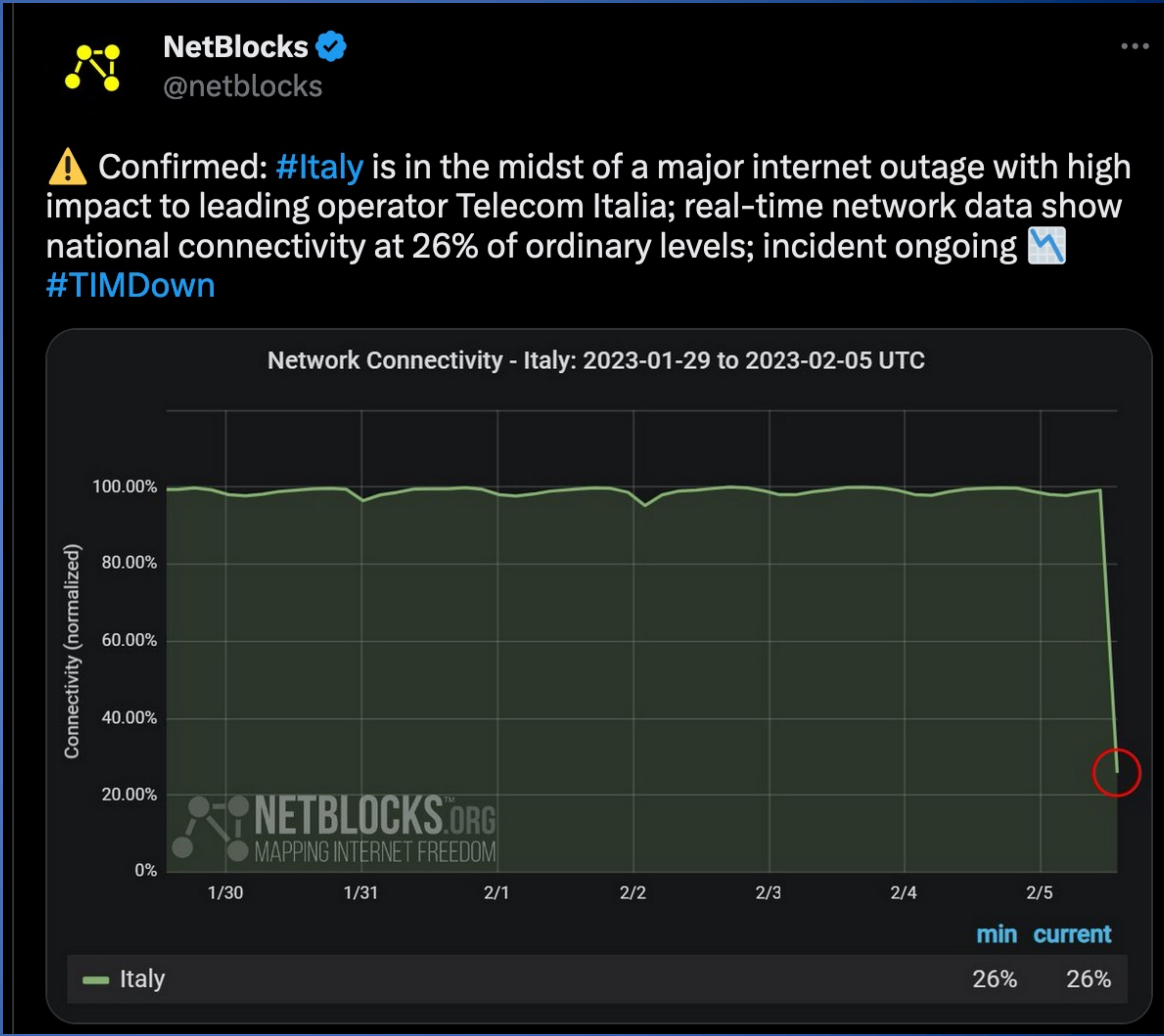
AS 17557 Pakistan, AS 36561 Youtube

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

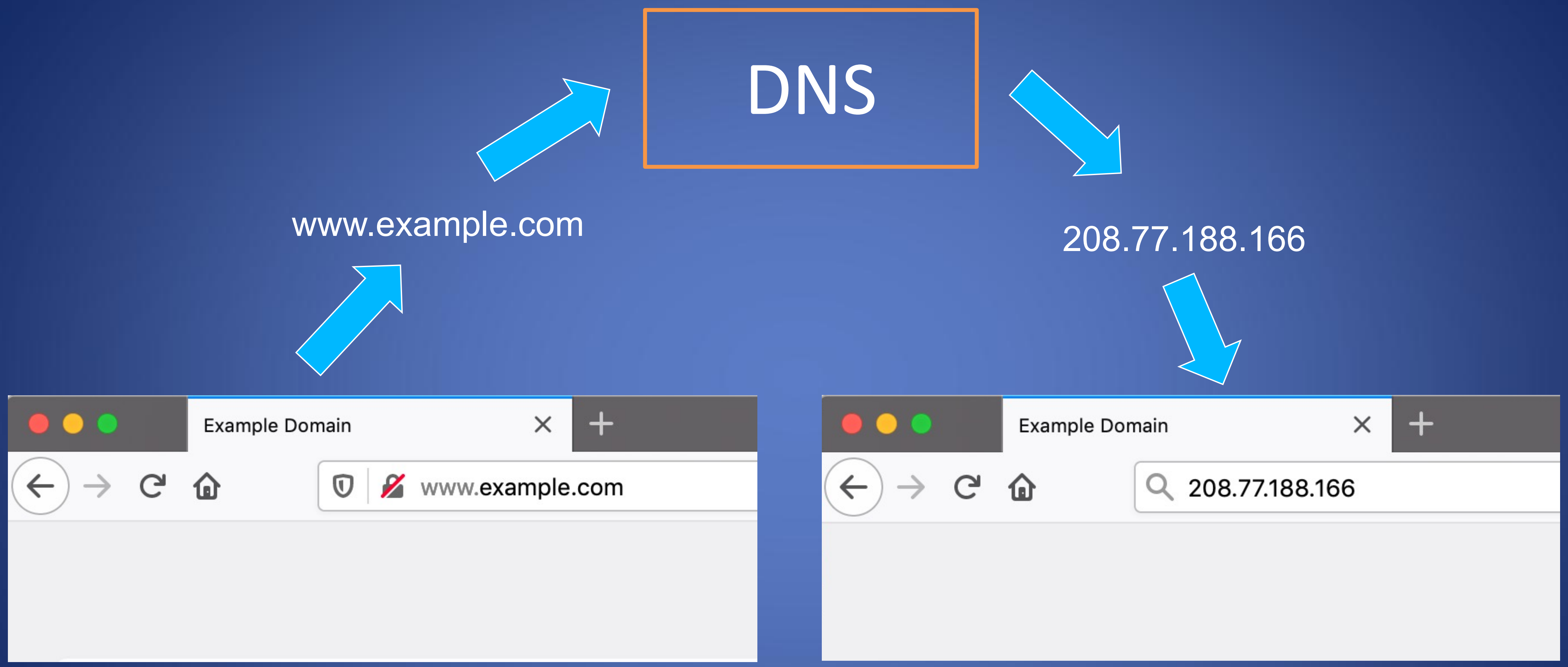
TIMDown

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

Probably a human error due to a bad DDOS configuration



DNS Domain Name System


DNS
www.example.com
208.77.188.166
Example Domain
www.example.com
Example Domain
208.77.188.166

Domain Name System

- The domain name system (DNS) is an application-layer protocol
- Basic function of DNS
  - Map domain names to IP addresses
  - The mapping is many to many
- Examples:
  - www.cs.brown.edu and cs.brown.edu map to 128.148.32.12
  - google.com maps to 198.7.237.251, 198.7.237.249, and other addresses
- More generally, DNS is a distributed database that stores resource records
  - Address (A) record: IP address associated with a host name
  - Mail exchange (MX) record: mail server of a domain
  - Name server (NS) record: authoritative server for a domain

Domains

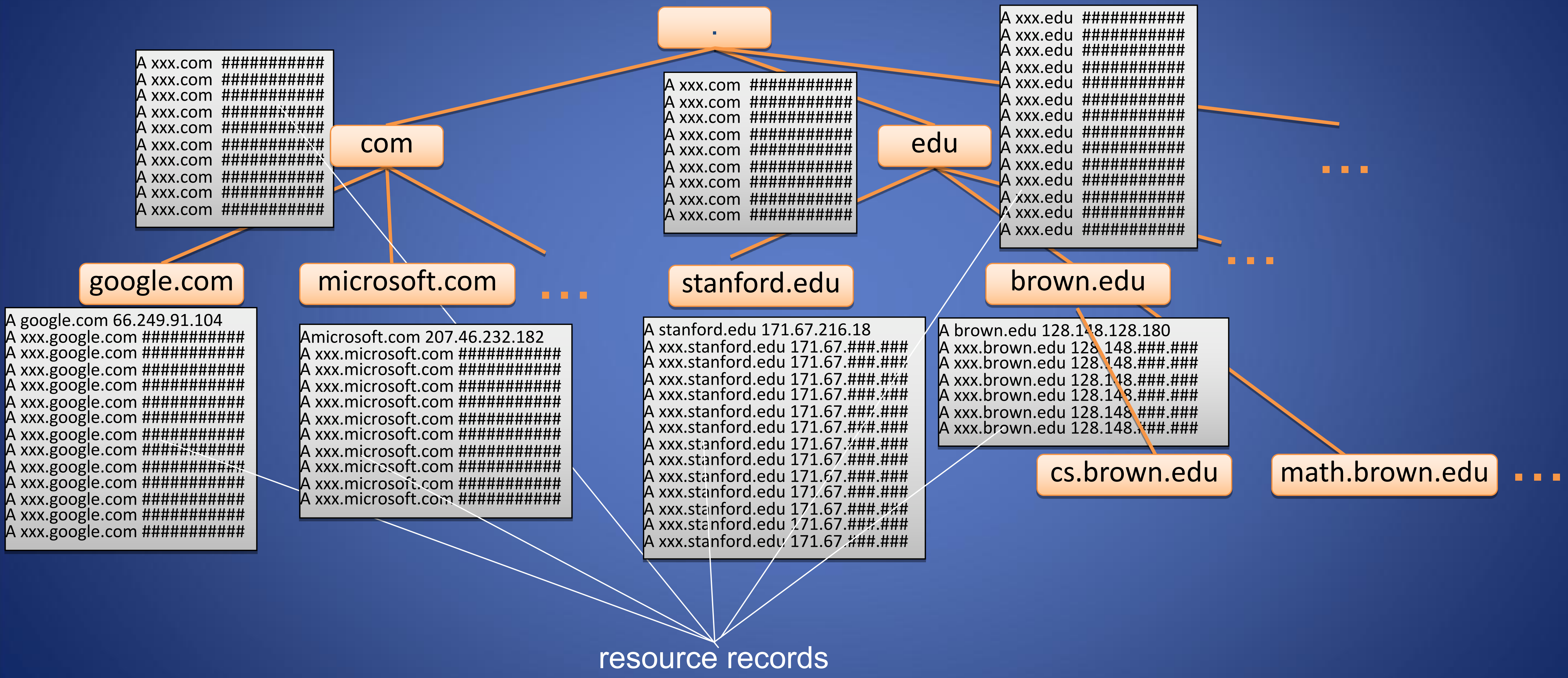
- FQDN (Fully Qualified Domain Name)
  - [Host name].[Domain].[TLD].[Root]
  - Two or more labels, separated by dots (e.g., cs.brown.edu)
- Root name server
  - It is a "." at the end of the FQDN
- Top-level domain (TLD)
  - Generic (gTLD), e.g., .com, .org, .net
  - Country-code (ccTLD), e.g., .ca, .it

- ICANN (Internet Corporation for Assigned Names and Numbers)

"One World. One Internet."

  - Keeps database of registered gTLDs (InterNIC)
  - Accredits registrars for gTLDs
- gTLDs
  - Managed by ICANN
- ccTLDs
  - Managed by government organizations

DNS Tree


.
com
edu
A xxx.com ###########
A xxx.com ###########
A xxx.com ###########
A xxx.com ###########
A xxx.com ###########
A xxx.com ###########
A xxx.com ###########
A xxx.com ###########
A xxx.com ###########
A xxx.com ###########
A xxx.edu ###########
A xxx.edu ###########
A xxx.edu ###########
A xxx.edu ###########
A xxx.edu ###########
A xxx.edu ###########
A xxx.edu ###########
A xxx.edu ###########
A xxx.edu ###########
A xxx.edu ###########
A xxx.edu ###########
A xxx.edu ###########
A xxx.edu ###########
A xxx.edu ###########
google.com
microsoft.com
stanford.edu
brown.edu
A google.com 66.249.91.104
A xxx.google.com ###########
Amicrosoft.com 207.46.232.182
A xxx.microsoft.com ###########
A stanford.edu 171.67.216.18
A xxx.stanford.edu 171.67.###.###
A brown.edu 128.148.128.180
A xxx.brown.edu 128.148.###.###
cs.brown.edu
math.brown.edu
. . .
resource records

Name Servers

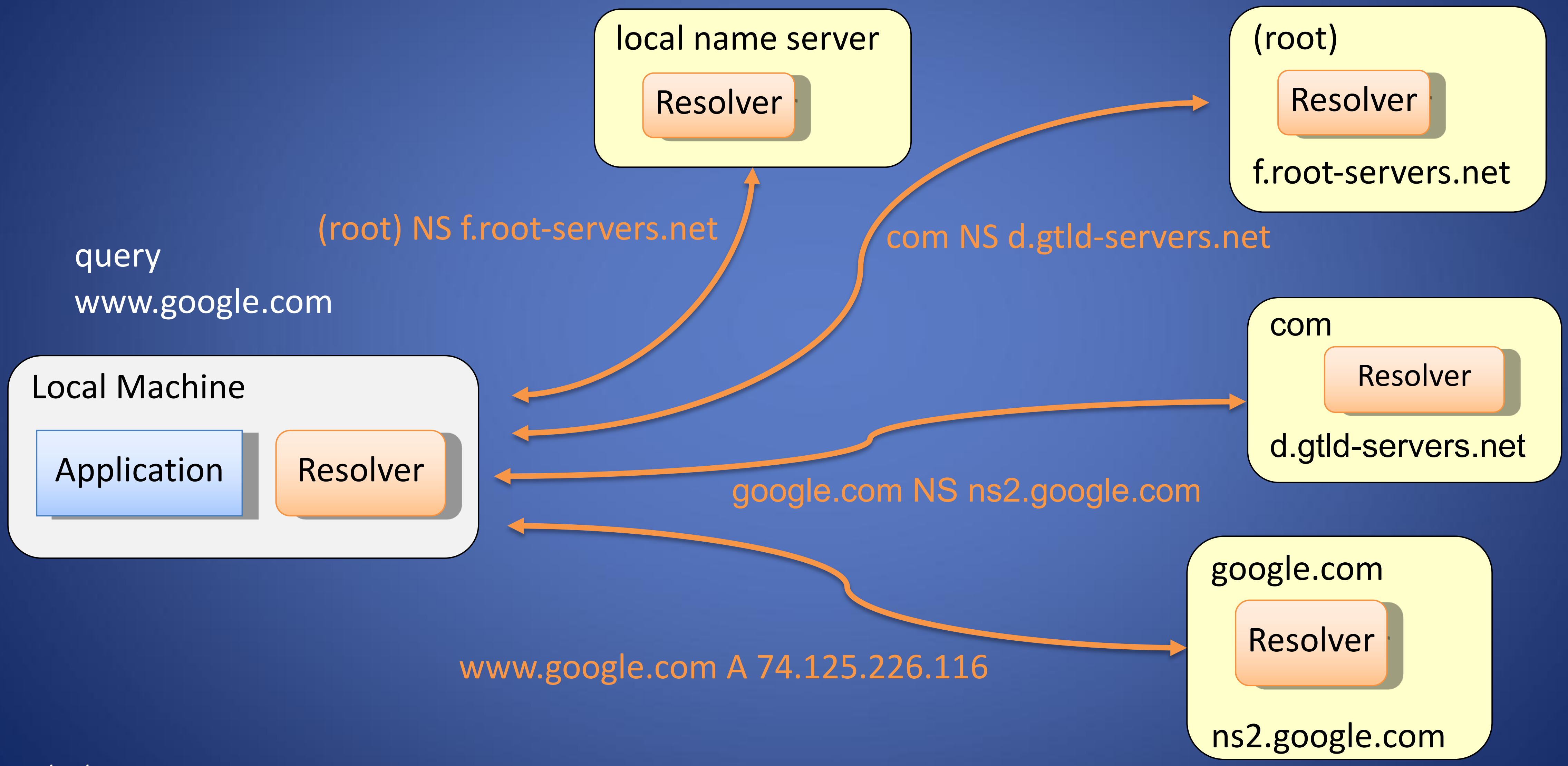
- Name server
  - Keeps local database of DNS records
  - Answers DNS queries
  - Can ask other name servers if record not in local database
- Authoritative name server
  - Stores reference version of DNS records for a zone (partial tree)
- Examples
  - dns.cs.brown.edu is authoritative for cs.brown.edu
  - bru-ns2.brown.edu is authoritative for brown.edu
- Root servers
  - Authoritative for the root zone (TLDs)
  - [a-m].root-servers.net (ICANN)
- Command tools for checking DNS
  - dig, nslookup, host
    (similar results with some differences)

Name Resolution

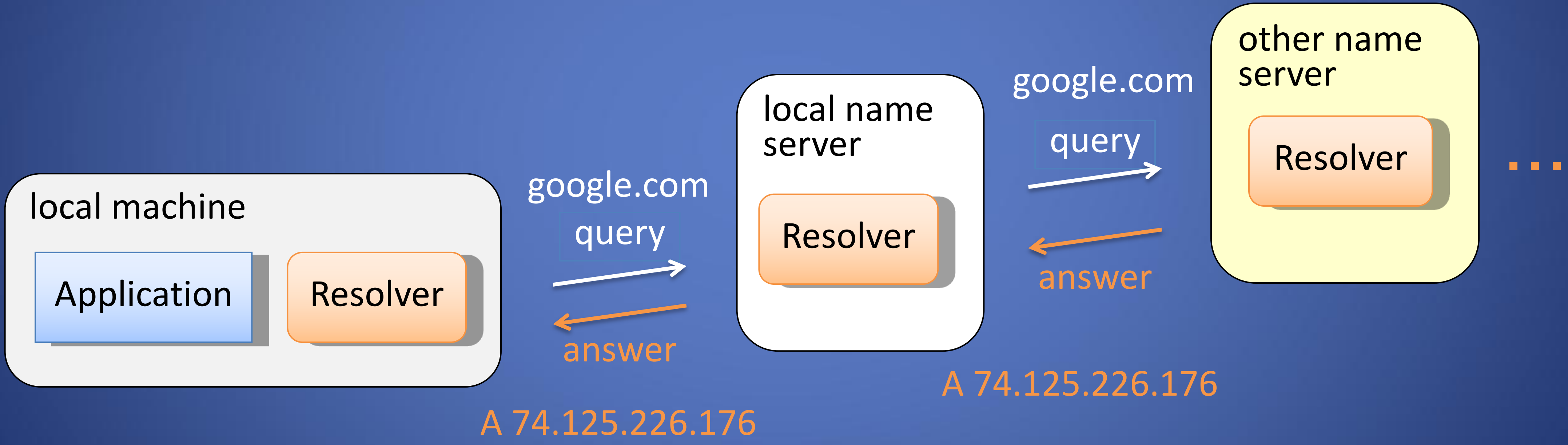
Name Resolution

- Resolver
  - Program that retrieves DNS records
  - Caches records received
  - Connects to a name server (default, root, or given)
- Iterative resolution
  - Name server refers client to authoritative server (e.g., a TLD server) via an NS record
  - Repeat
- Recursive resolution
  - Name server queries another server and forwards the final answer (e.g., A record) to client

Iterative Name Resolution



Recursive Name Resolution



Glue Records

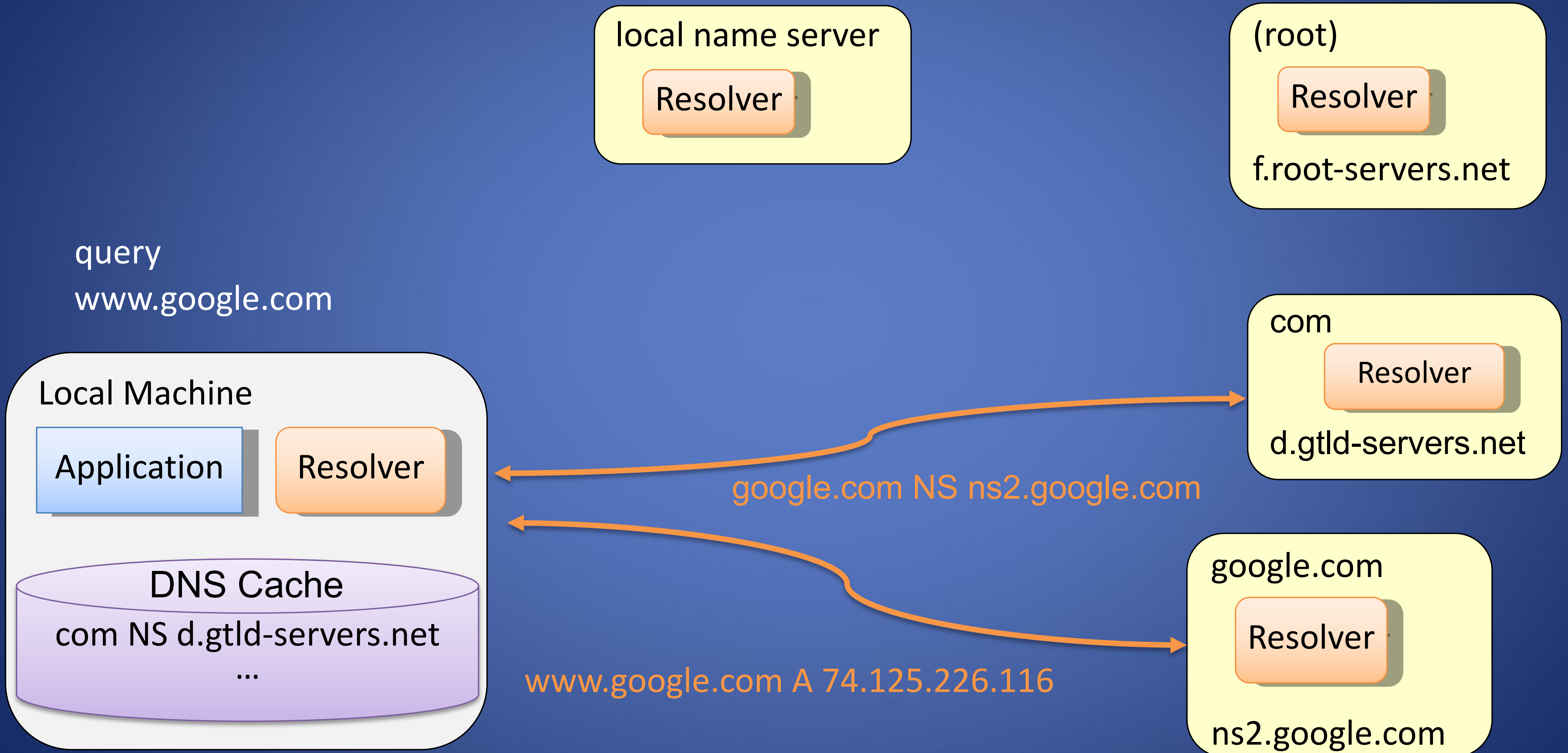
- Circular references
  - The authoritative name server for a domain may be within the same domain
  - E.g., dns.cs.brown.edu is authoritative for cs.brown.edu
- Glue record
  - Record of type A (IP address) for a name server referred to NS record
  - Essential to break circular references
- Example
  - brown.edu. NS bru-ns1.brown.edu.
  - bru-ns1.brown.edu. A 128.148.248.11 [glue record]

DNS Caching

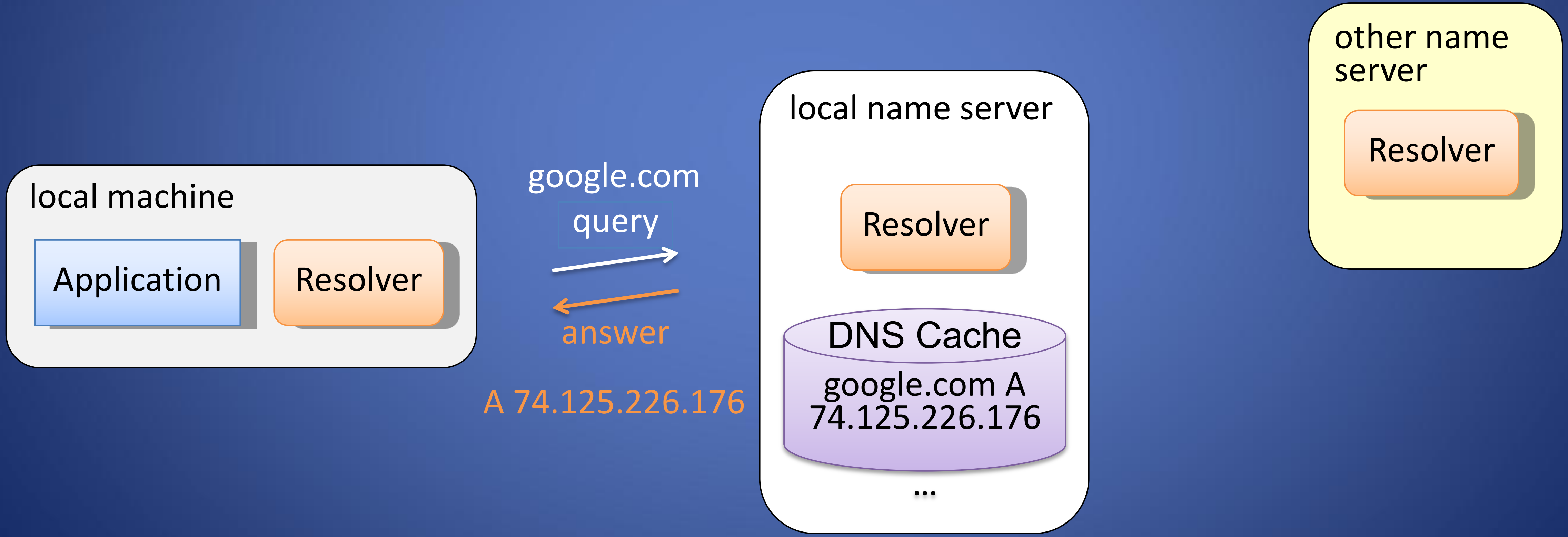
DNS Caching

- There would be too much network traffic if a path in the DNS tree would be traversed for each query
  - Root servers and TLD servers would be rapidly overloaded
- DNS servers cache records that are results of queries for a specified amount of time
  - Time-to-live field
- DNS queries with caching
  - First, resolver looks in cache for A record of query domain
  - Next , resolver looks in cache for NS record of longest suffix of query domain

Iterative Name Resolution with Caching



Recursive Name Resolution with Caching



Local DNS Cache

- Operating system and some applications (e.g., browsers) maintain local DNS cache
- Operating system DNS cache
  - On some versions, DNS cache is shared among all running processes
  - Can be displayed by all users
  - View DNS cache in Windows with command ipconfig /displaydns
  - Clear DNS cache in Windows with command ipconfig /flushdns
- Privacy issues with shared operating system DNS cache
  - Browsing by other users can be monitored
  - Note that private/incognito browsing does not clear DNS cache

Clicker Question (1)

Imagine the following name resolution protocol: the resolver looks in the cache of the local name server, finds a record of the query domain, and returns it to the client. What category does this fall under?

A. Iterative Name Resolution

B. Iterative Name Resolution with caching

C. Recursive Name Resolution

D. Recursive Name Resolution with caching

E. Both B and D

Clicker Question (1)

Imagine the following name resolution protocol: the resolver looks in the cache of the local name server, finds a record of the query domain, and returns it to the client. What category does this fall under?

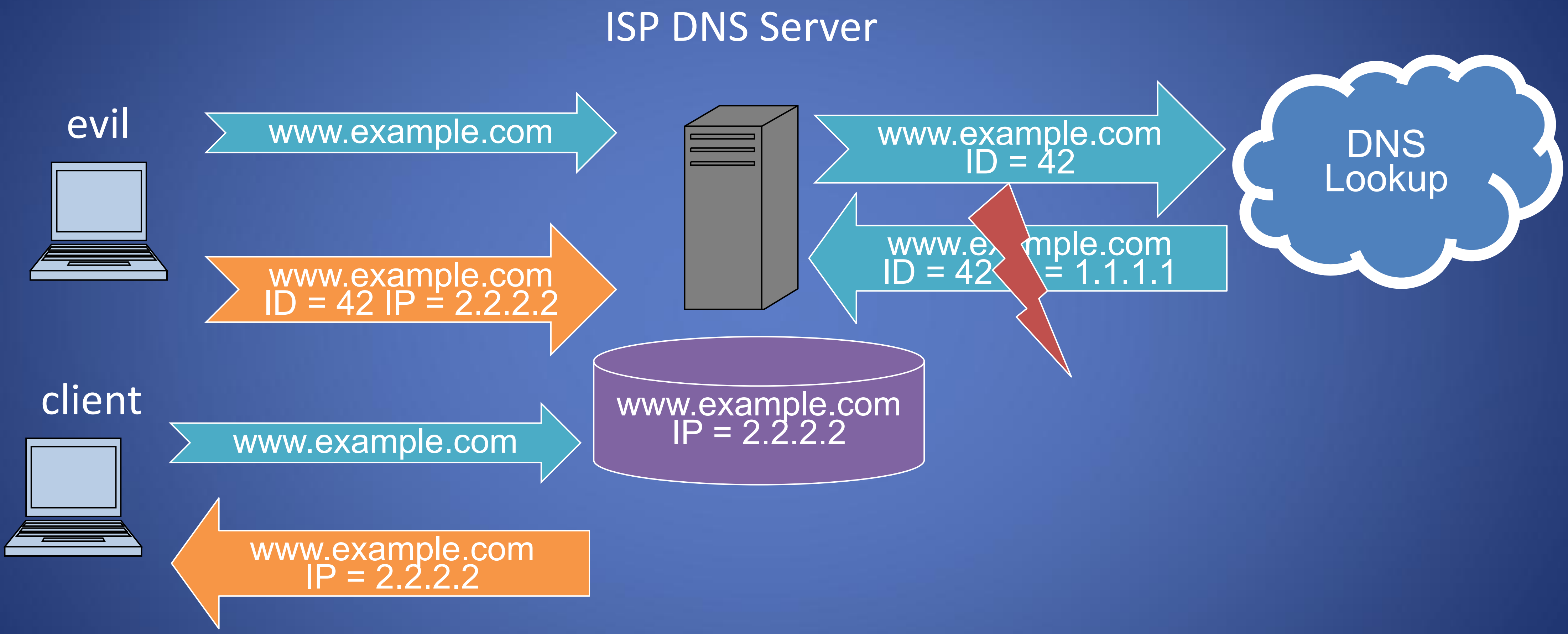
A. Iterative Name Resolution
B. Iterative Name Resolution with caching
C. Recursive Name Resolution
D. Recursive Name Resolution with caching
E. Both B and D

DNS Attacks

DNS Cache Poisoning

- Basic idea
  - Give a DNS server a false address record and get it cached
- DNS query mechanism
  - Queries issued over UDP on port 53
  - 16-bit request identifier in payload to match answers with queries
  - No authentication
  - No encryption

- Cache may be poisoned when a resolver
  - Disregards identifiers
  - Has predictable identifiers and return ports
  - Accepts unsolicited DNS records


ISP DNS Server
evil
www.example.com
www.example.com
ID = 42 IP = 2.2.2.2
www.example.com
ID = 42
DNS
Lookup
ID = 42
= 1.1.1.1
www.example.com
IP = 2.2.2.2
client
www.example.com
www.example.com
IP = 2.2.2.2

DNS Cache Poisoning Defenses

- Query randomization
  - Random request identifier (16 bits)
- Probability of guessing request ID or return port

$$1 / 2^{16} = 0.0015\%$$

- Check request identifier
- Use signed records
  - DNSSEC

Clicker Question

An attacker with a rogue machine on your local area network is sniffing traffic and wants to poison your DNS cache. Your DNS resolver uses both query ID and return port randomization. Is the poisoning attack going to succeed?

A. Not at all
B. Only with small probability $1 / 2^{16}$
C. Only with very small probability $1 / 2^{32}$
D. Likely

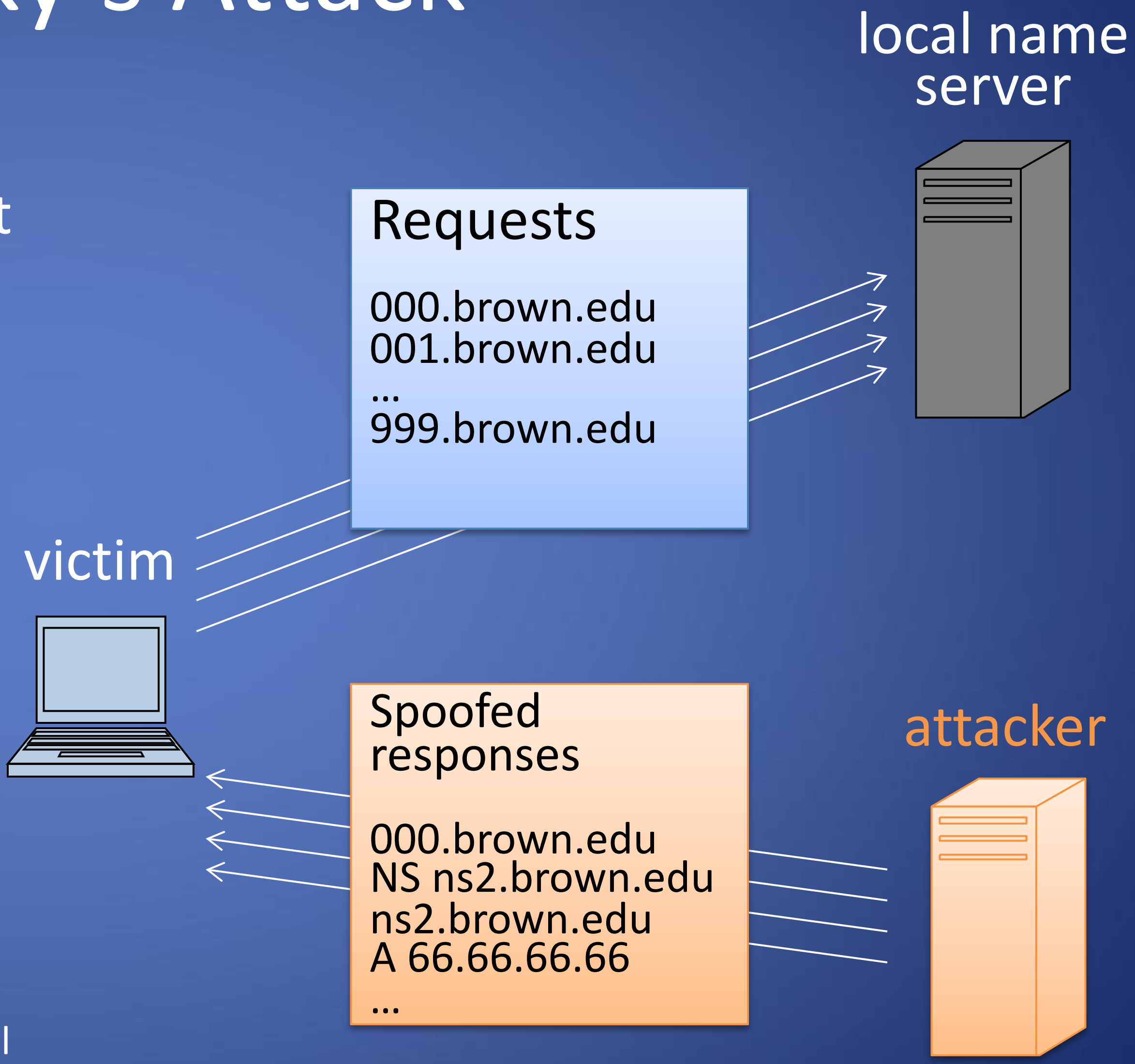
Clicker Question - Answer

An attacker with a rogue machine on your local area network is sniffing traffic and wants to poison your DNS cache. Your DNS resolver uses both query ID and return port randomization. Is the poisoning attack going to succeed?

A. Not at all
B. Only with small probability $1 / 2^{16}$
C. Only with very small probability $1 / 2^{32}$
D. Likely

Kaminsky's Attack

- Attacker causes victim to send
  - Many DNS requests for nonexistent subdomains of target domain
- Attacker sends victim
  - Forged NS responses for the requests
- Format of forged response
  - Random ID
  - Correct NS record
  - Spoofed glue record pointing to the attacker's name server IP
- http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html



DNS Cache Poisoning Defenses after Kaminsky's attack

- Query randomization
  - Random request identifier (16 bits)
  - **Random return port (16 bits)**
- Probability of guessing request ID or return port

  $1 / 2^{16} = 0.0015\%$
- **Probability of guessing request ID and return port is**

  $1 / 2^{32}$ (less than one in four billion)

- Check request identifier
- Use signed records
  - DNSSEC

DNS Hijacking/Redirecting

- Subvert the resolution of DNS queries
- **Malicious**: you type "bank.com" and attacker directs you to incorrect IP address
- **Censorship:** e.g. Great Firewall of China
- DoS: hacktivist can use to block dns resolution

- **ISPs:**
  - Display ads (instead of or in addition to existing ads),
  - Collect statistics about user traffic.
  - Block access to websites.

Transport Layer (Ports, TCP, UDP)

The Transport Layer

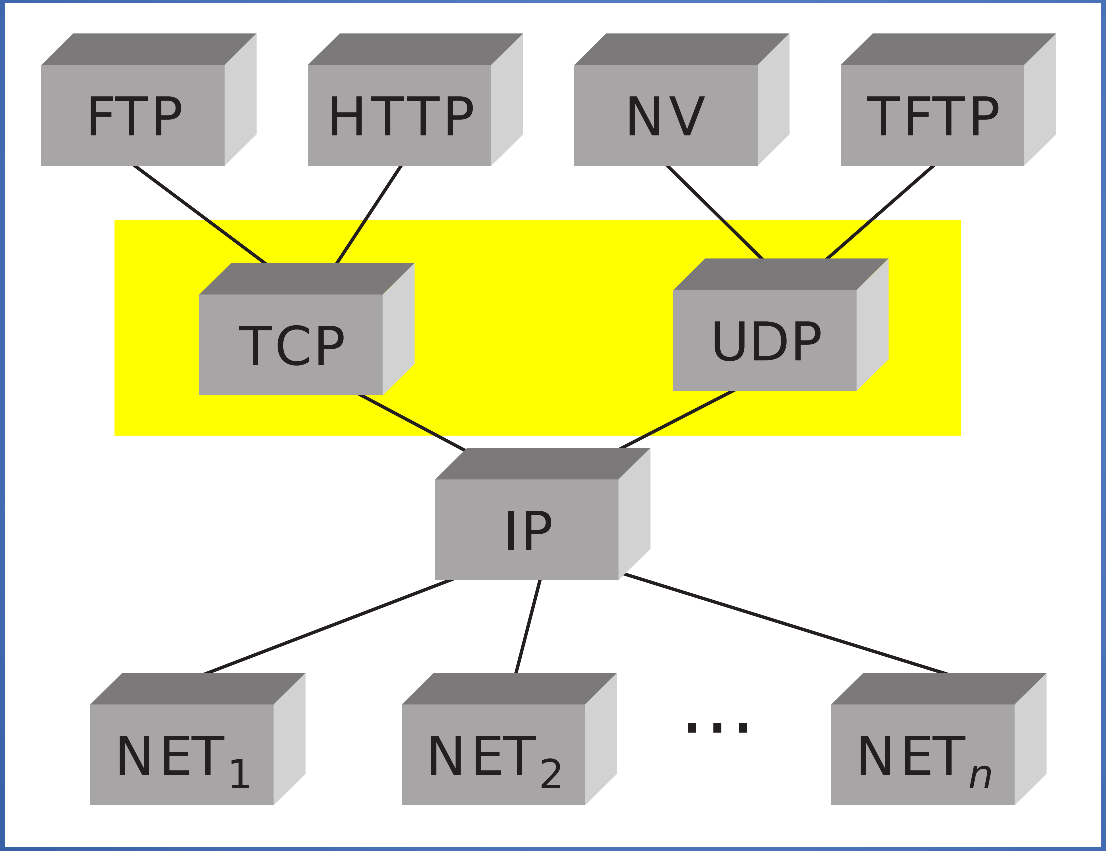
Network layer: moving data between hosts

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

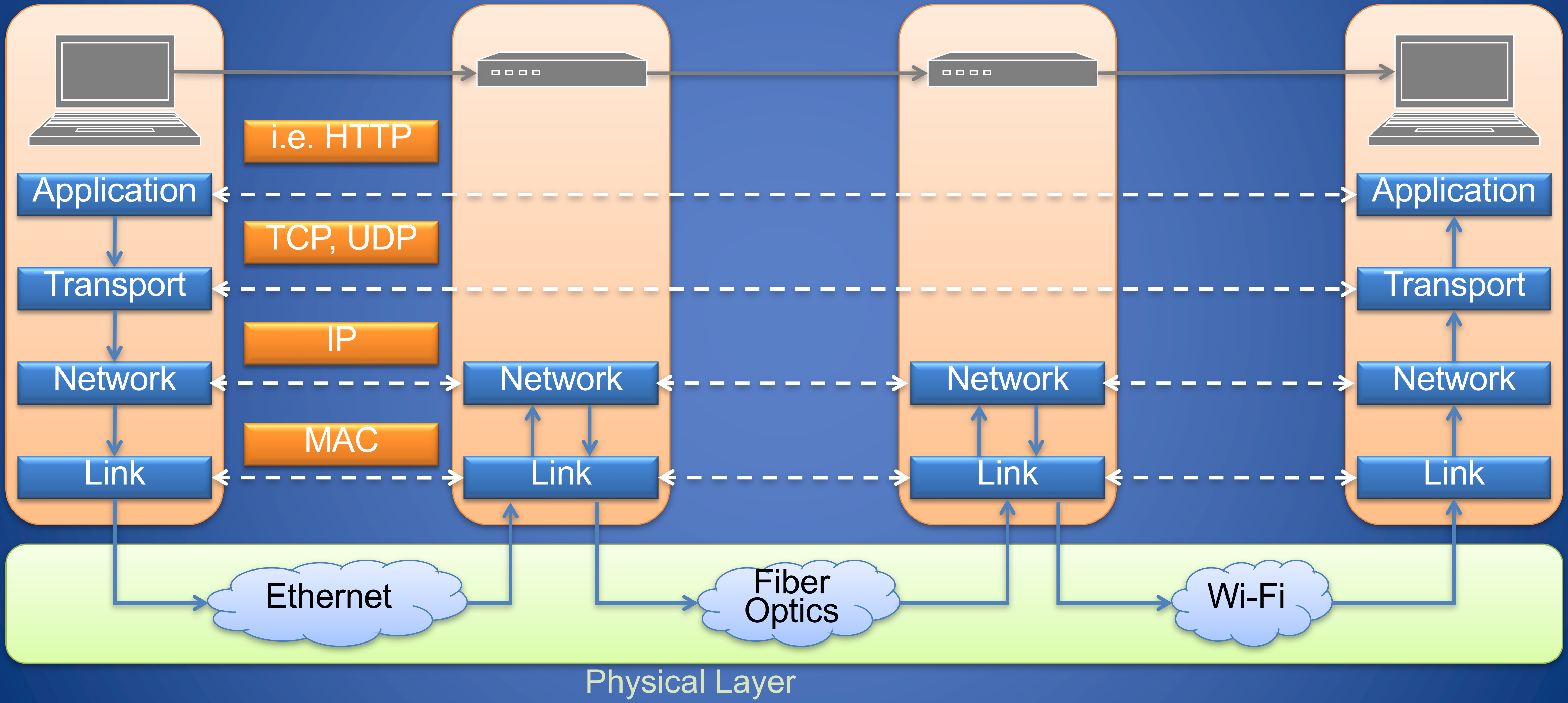
- Multiplexing multiple connections at the same IP with port numbers
- Series of packets => stream of data/messages
- May provide: reliable data delivery

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

<u>Port numbering</u>

- 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

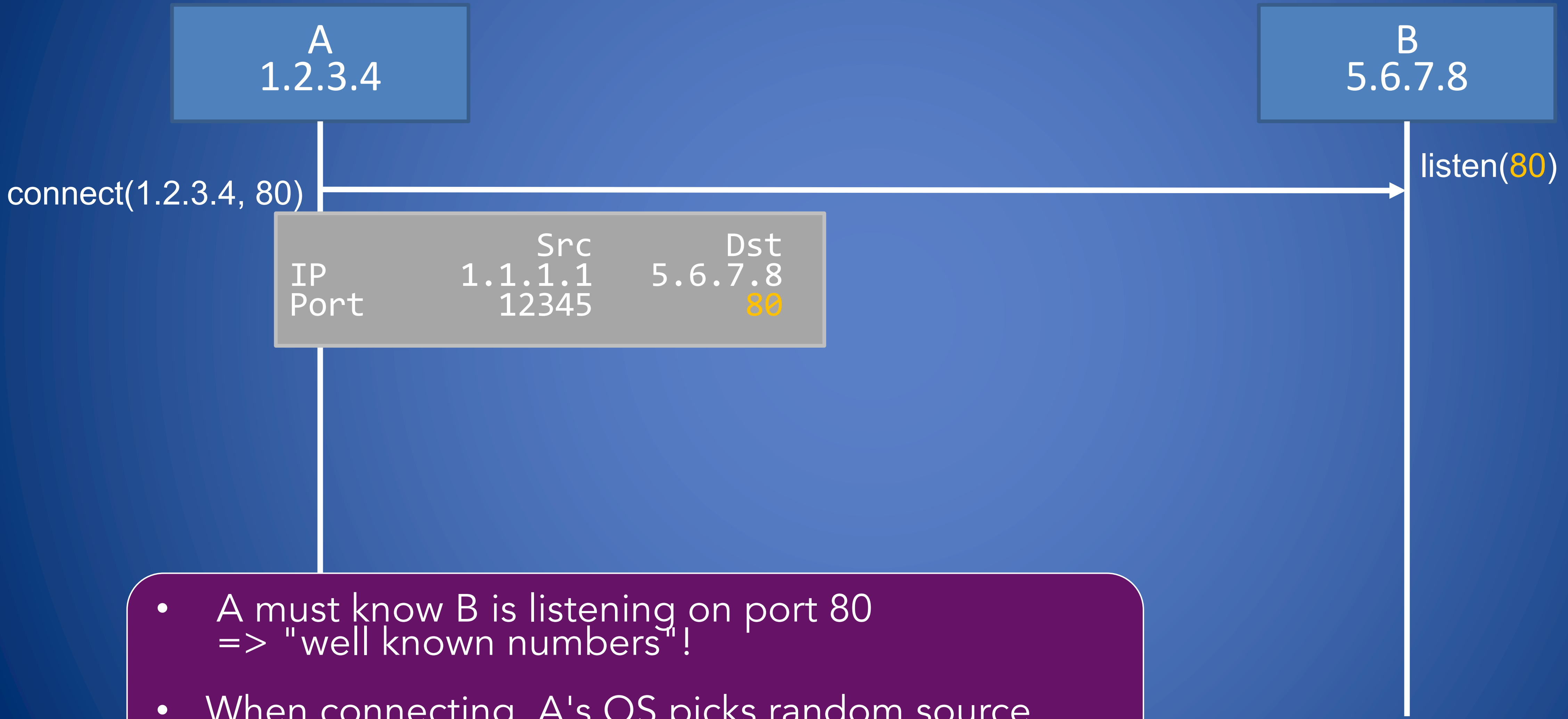
Two modes:

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

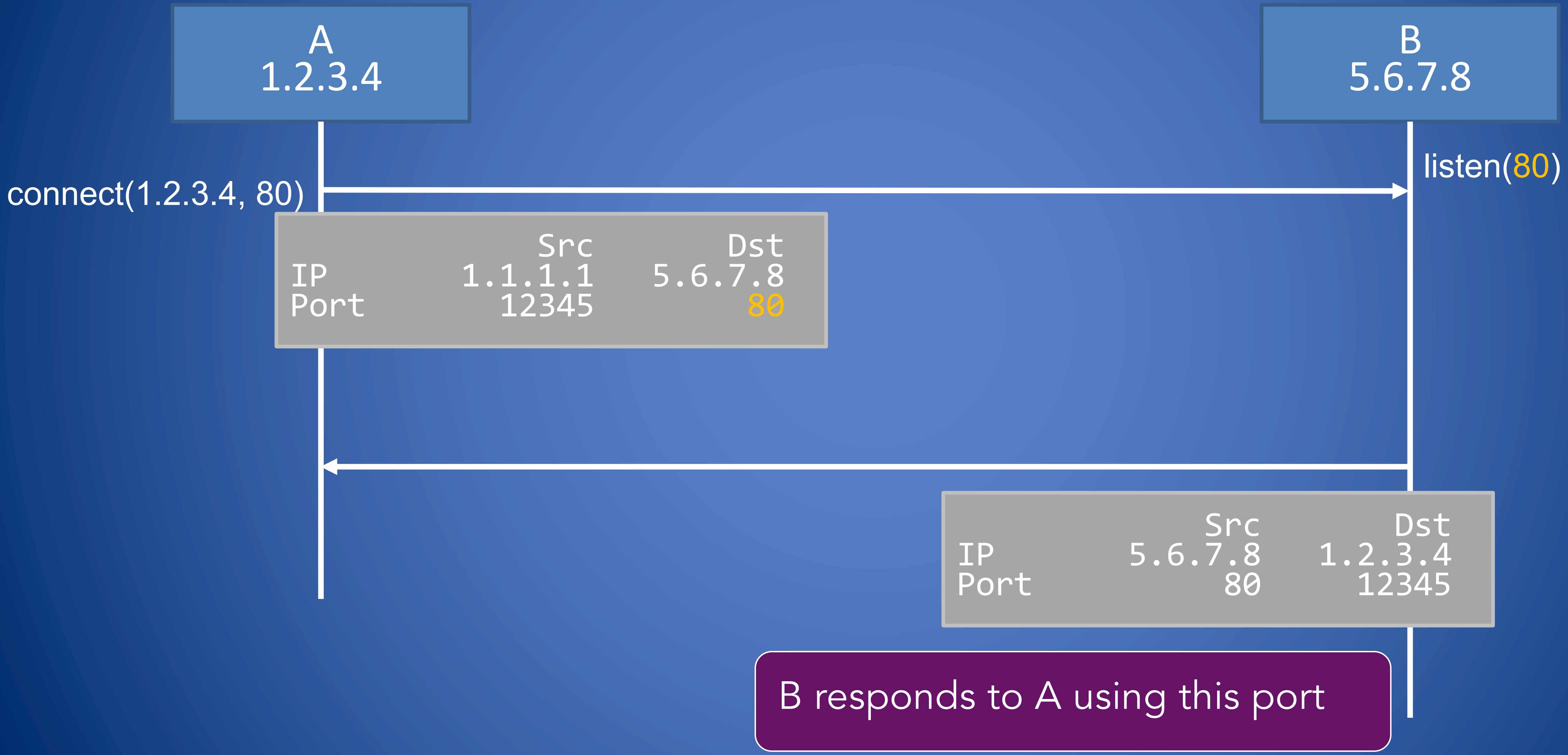
  => Example: webserver listens on port 80 or 443

- 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 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)

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)

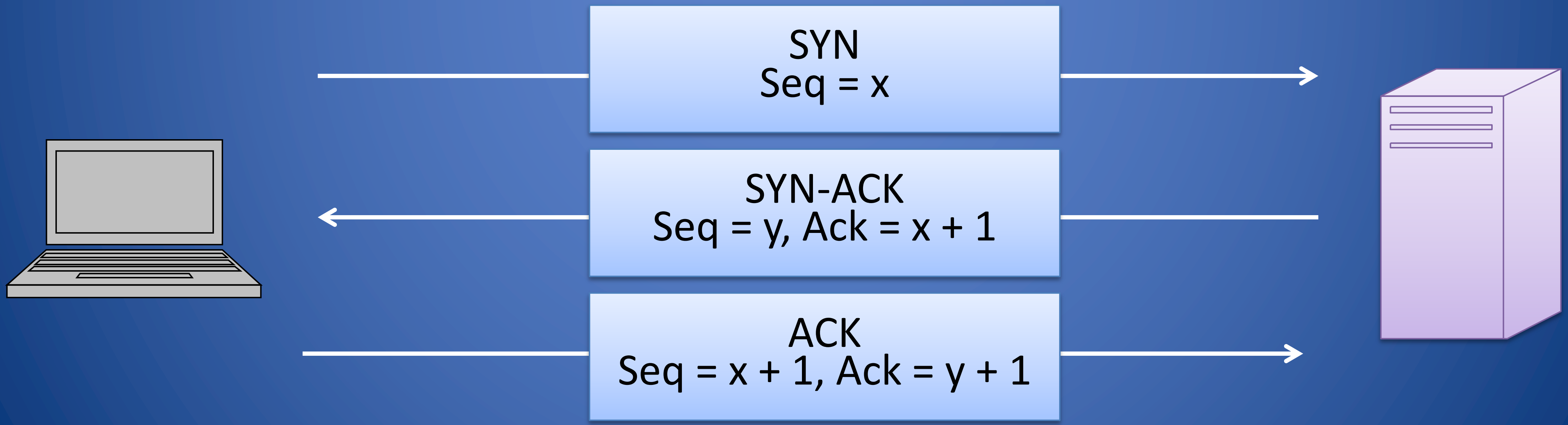
- Stateful protocol for reliable data transfer, in-order delivery of messages and ability to distinguish multiple applications on same host
  - HTTP and SSH are built on top of TCP
- TCP packages a data stream it into segments transported by IP
  - Order maintained by marking each packet with sequence 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

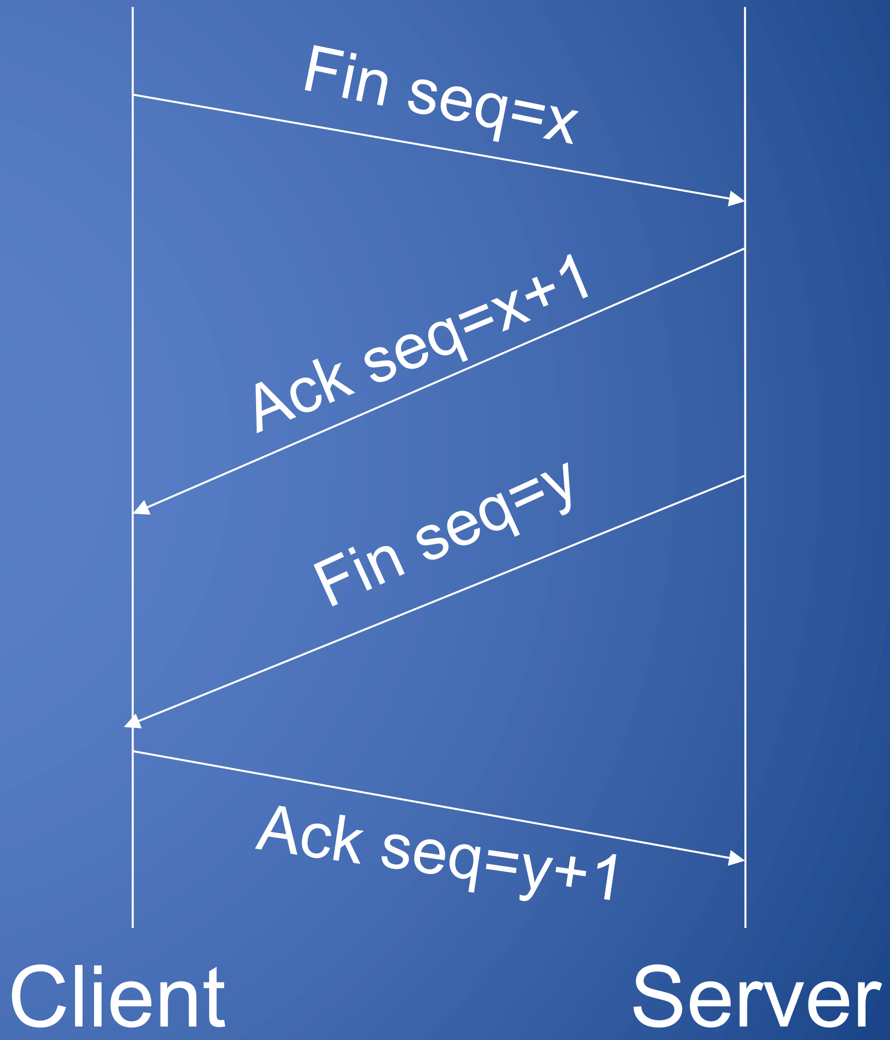
- TCP connections are established through a three-way handshake
- The server generally is a passive listener, waiting for a connection request
- The client requests a connection by sending out a SYN packet
- The server responds by sending a SYN/ACK packet, acknowledging the connection
- The client responds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



What We Have Learned

- IP address space allocation
- ARP protocol
- ARP poisoning attack
- Transport layer protocols
  - TCP for reliable transmission
  - UDP when packet loss/corruption is tolerated
- Lack of built-in security for link, network, and transport layer protocols
  - Security enhanced protocols have been developed for these layers
  - Alternate solution is to provide security at application layer