CS118 Discussion Week 6: The Network Layer (Data Plane)

Questions?

- From this week or about the HW/Midterm

The Network Layer

- Transport segment from sending to receiving host
  - Sender: encapsulates segments into datagrams, passes to link layer
  - Receiver: delivers segments to transport layer protocol
- Network layer protocols in every Internet device: hosts, routers
- Two Key Features:
  - Forwarding (move packets from a router's input link to appropriate router output link)
  - Routing (determine route taken by packets from source to destination)

Control Plane

- Network-wide logic
- Determines how datagram is routed among routers along end-to-end path from source host to destination host
- Two control-plane approaches:
  - traditional routing algorithms: implemented in routers
  - software-defined networking (SDN): implemented in (remote) servers

Data Plane

- local, per-router function
- determines how datagrams arriving on router input ports are forwarded to router output ports
- What we'll be focusing on in this section!

Network Protocols

- Last section we went over two prominent Transport protocols – TCP and UDP.

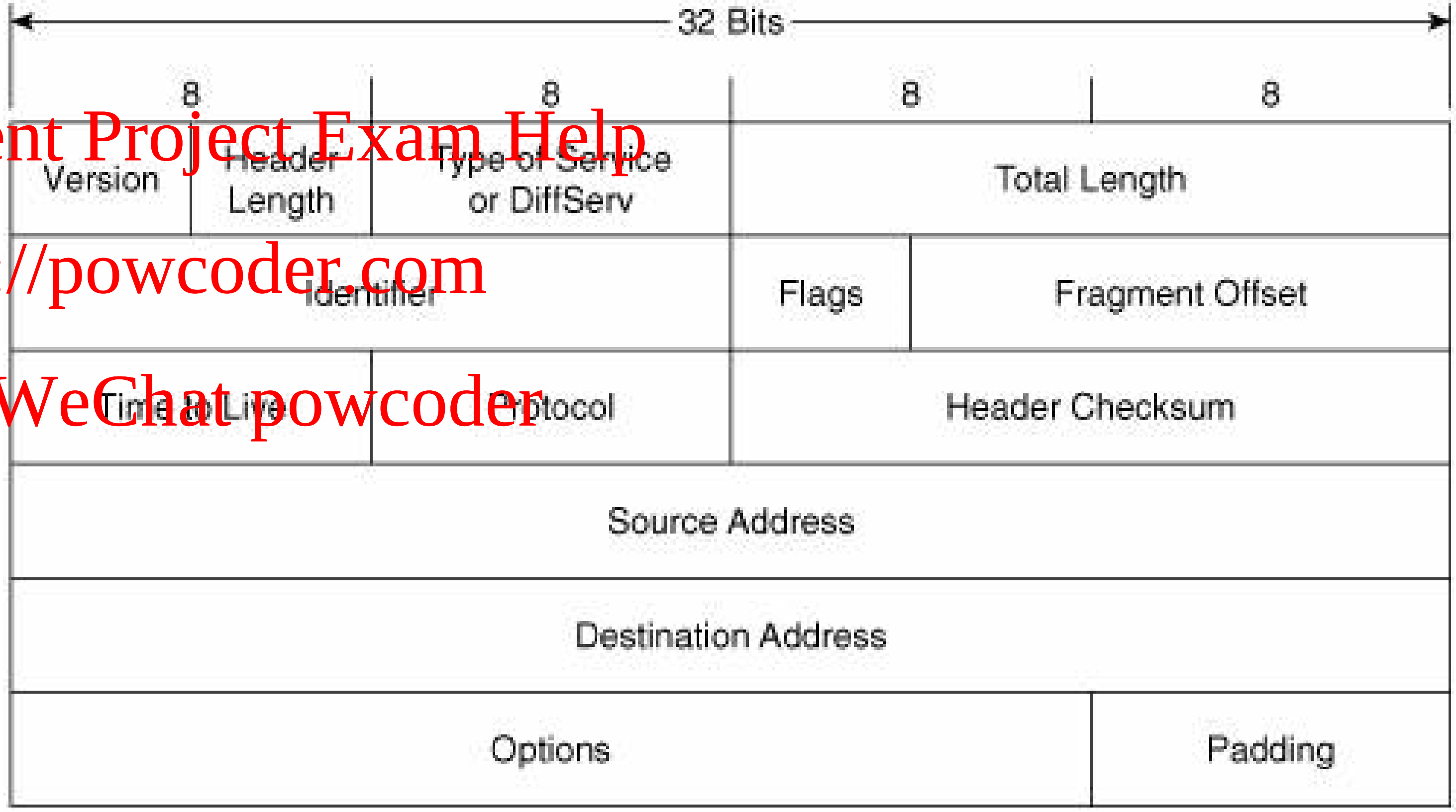
- For the network layer, there's (at least right now), only really one game in town.

- IP (Internet Protocol, v4 and v6).

  - For the sake of simplicity we'll go over v4 first.
  - Concerns itself with sending information from one address (x.y.z.a) to another.

IPv4

- Quite similar to TCP (the two together are often known as TCP/IP)
- At its core, sends a packet through the network based solely on its Destination Address.
- Interesting fields:
  - Identifier primarily used for uniquely identifying fragments.
  - Flags is used to control and identify fragments.
  - The fragment offset field, measured in units of eight-byte blocks, is 13 bits long and specifies the offset of a particular fragment relative to the beginning of the original unfragmented IP datagram
  - TTL is the maximum amount of time a datagram can live in the internet.
  - Source/Dest address: what you'd expect.



Network-layer service model: IP

Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no

Internet "best effort" service model

No guarantees on:

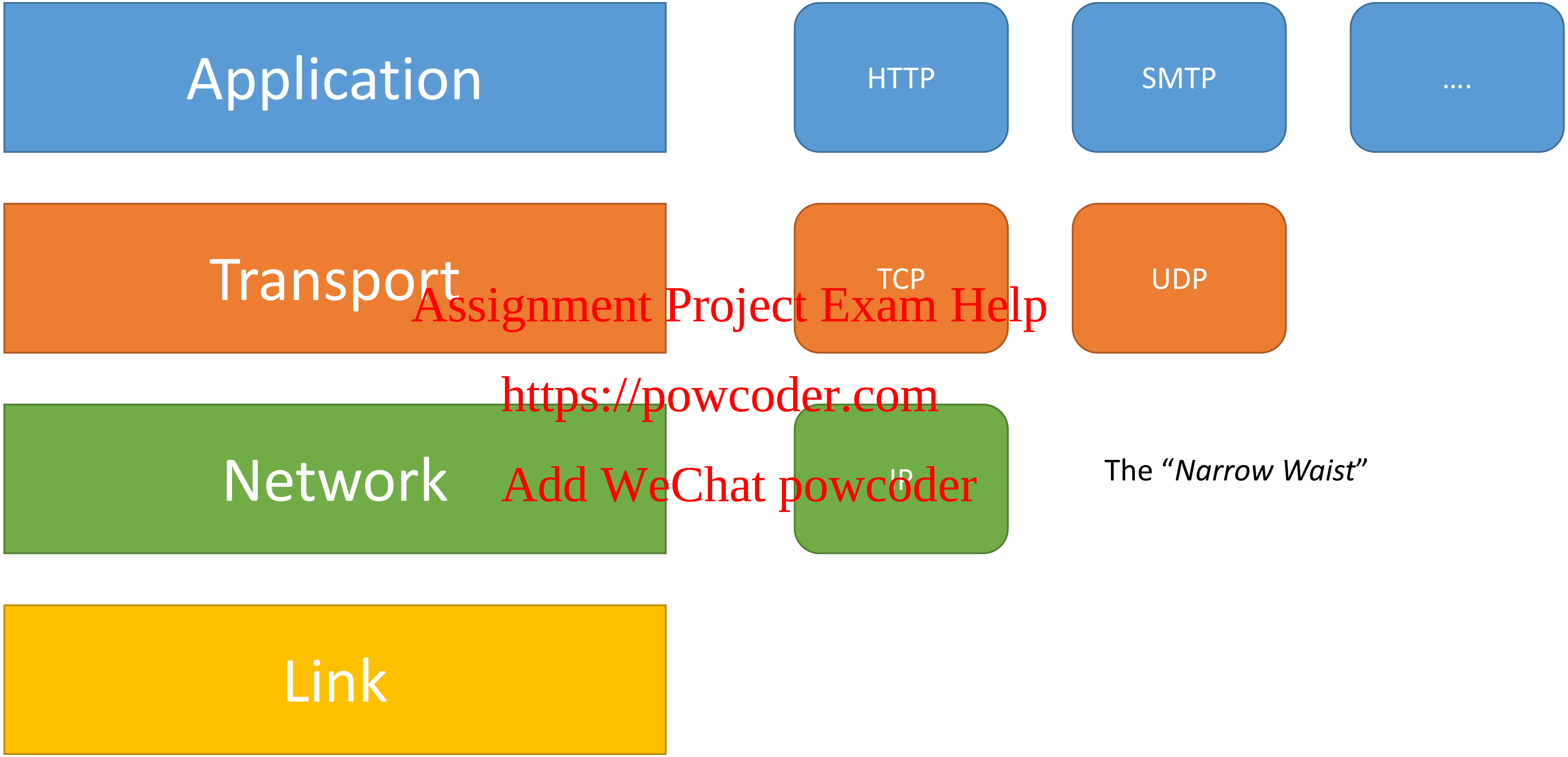
i. successful datagram delivery to destination
ii. timing or order of delivery
iii. bandwidth available to end-end flow

IP Vs Competitors

Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no
ATM	Constant Bit Rate	Constant rate	yes	yes	yes
ATM	Available Bit Rate	Guaranteed min	no	yes	no
Internet	Intserv Guaranteed (RFC 1633)	yes	yes	yes	yes
Internet	Diffserv (RFC 2475)	possible	possibly	possibly	no

Evaluation of IP

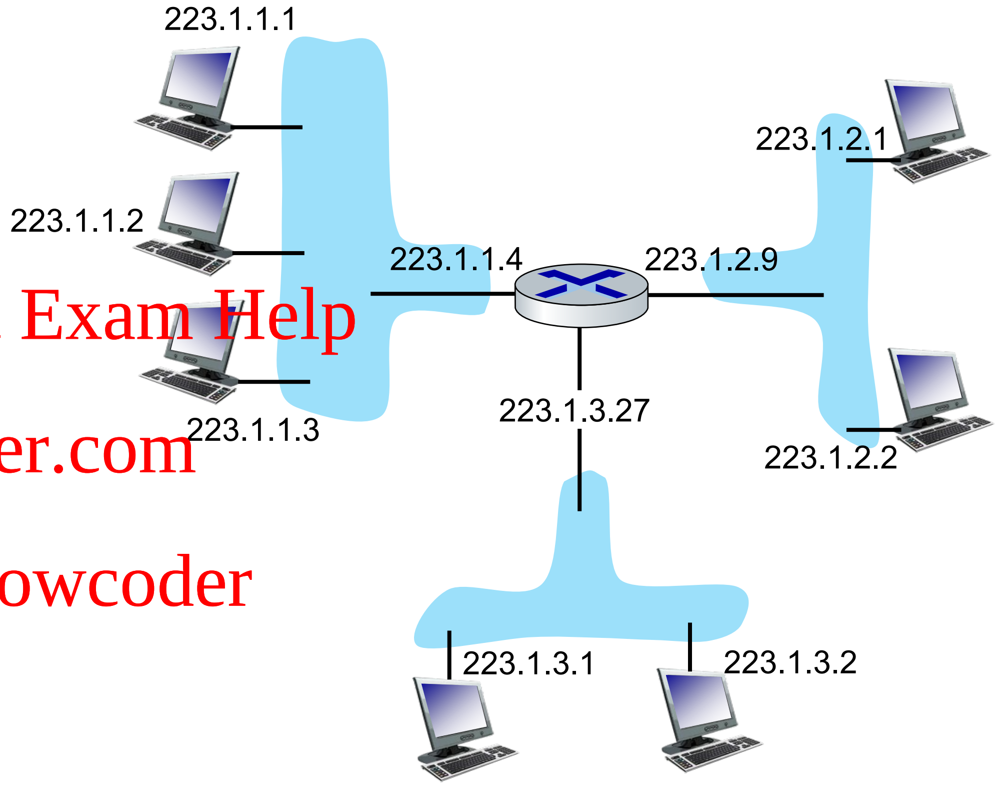
- Why has IP been so successful?
  - simplicity of mechanism has allowed Internet to be widely deployed adopted
  - sufficient provisioning of bandwidth allows performance of real-time applications (e.g., interactive voice, video) to be “good enough” “most of the time”
  - replicated, application-layer distributed services (datacenters, content distribution networks) connecting close to clients’ networks, allowing services to be provided from multiple locations
  - congestion control of “elastic” services helps



Application
HTTP
SMTP
....
Transport
TCP
UDP
Network
IP
The “*Narrow Waist*”
Link

IPv4 addressing

- **IPv4 address:** 32-bit identifier associated with each host or router *interface* – form x.y.z.a (where each of these is a number 0-255)
- **interface:** connection between host/router and physical link
  - routers typically have multiple interfaces
  - hosts typically have one or two interfaces (e.g., wired Ethernet, wireless 802.11)



dotted-decimal IP address notation:

223.1.1.1 = $\underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$

Subnetworking

- Computer networks are inherently “hierarchical”
  - E.g., UCLA’s network vs. Google’s network
- We split the IP address space up into “subnets”
  - In other words, logically group hosts by the first *n* bits of their addresses
- 192.168.0.0-192.168.0.255 can be a subnet
  - First 24 bits are network, last 8 bits are host in that network (2^8-2 hosts)
  - We can’t use the first or last address since they’re special
    - Network identifier and “broadcast” addresses
  - Left with 254 host addresses in this subnet

"Classful" Addressing

- In the old days, subnets could only be formed on byte boundaries
  - E.g., 10.0.0.0-10.255.255.255 or 192.168.54.0-192.168.54.255

- On which byte they were split was determined by their "class"

  - Class A: Split after first byte (2^{24}-2 hosts per subnet)
  - Class B: Split after second byte (2^{16}-2 hosts per subnet)

  - Class C: Split after third byte (2^{8}-2 hosts per subnet)
  - Other classes for special addresses (e.g., multicast)
- However, too coarse-grained to represent structure of Internet
- → Replaced with "Classless-Interdomain Routing" (CIDR)

CIDR

- Split network and byte portions of address after some number of bits
  - E.g., 27

128/25

24 | 25
0 | 0 . . .
0 | 1 . . .

- Network represented like 192.168.0.0/25
  - First IP address/number of bits in network portion
  - This represents network 192.168.0.0-192.168.0.127

- Greater flexibility, can split subnets up into smaller subnets
  - E.g., CS department subnet with UCLA subnet
  - And then, research group subnet with CS department subnet

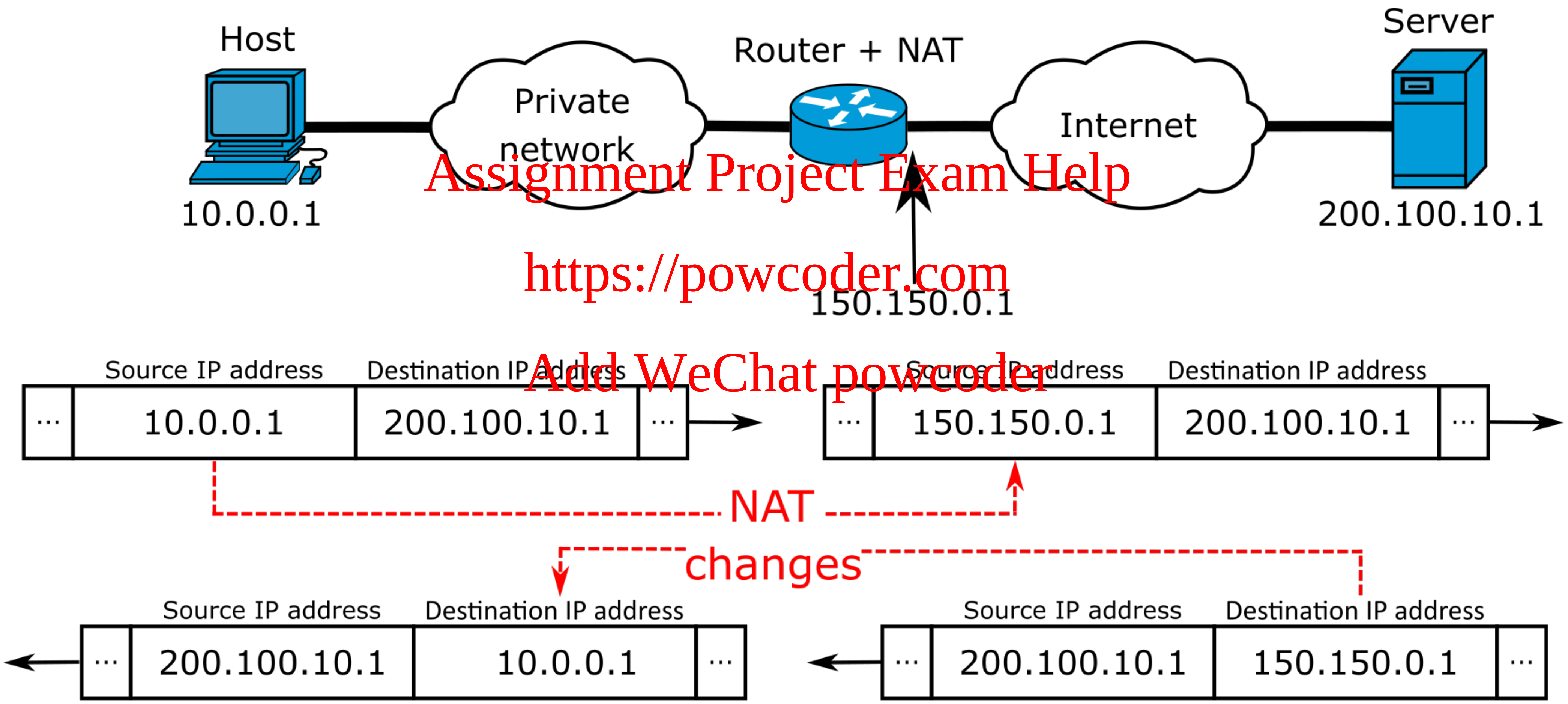
How do you get an IP Address, anyway?

- Dynamic Host Configuration Protocol (DHCP)
- Send a broadcast message to the local network asking for an address
- DHCP server will record address “lease” assignment and respond with IP address for host to use
- Response also includes information about which DNS servers to use and the local router’s IP address
- DHCP address “leases” are refreshed periodically
- Your home network uses DHCP, with the server running in your router

Modifications to IP: NAT

- Problem: IPv4 is running out of addresses
  - Who would have thought that there would be more than ~4B (2^32) hosts on the Internet?
    - “640K [of RAM] ought to be enough for anybody” –(allegedly) Bill Gates, 1981
- Stopgap Solution (“NAT”): Hide private networks behind gateways
  - Use a “private” IP address in one of the following ranges:
    - 192.168.0.0/16, 10.0.0.0/8, 172.16.0.0/12
  - Reuse these private addresses between networks
  - Map internal IP and port numbers to external IP and port numbers
  - And update packet headers as packets pass through router

Network Address Translation (NAT)



https://commons.wikimedia.org/wiki/File:NAT_Concept-en.svg

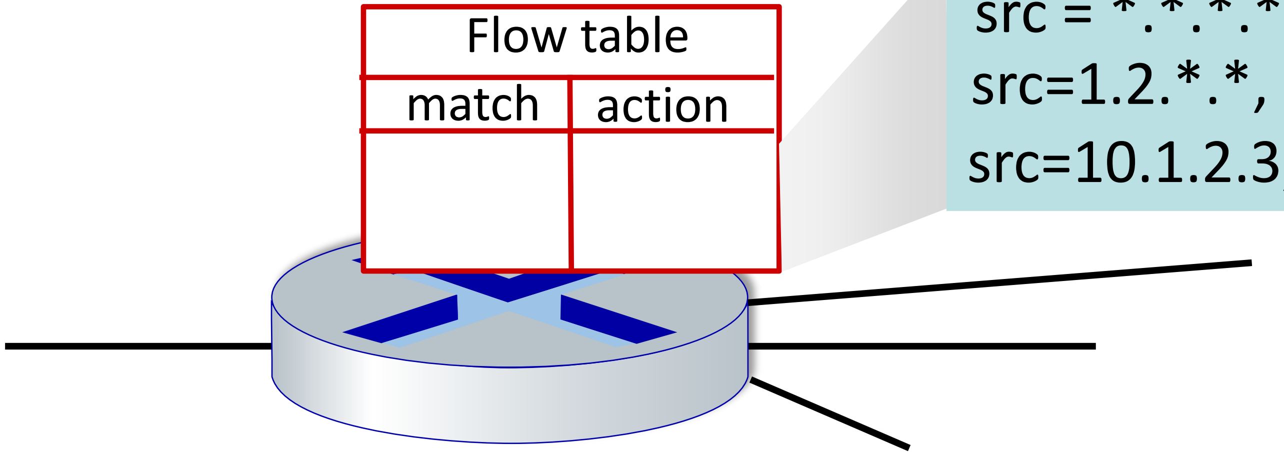
Modifications to IP: IPv6

- Better Solution: Expand IP address space from 32 to 128 bits!
  - 2^128 IP addresses!
  - Represented as 8 groups of 4 hex digits
  - 2001:0db8:85ae:0000:0000:~~0000~~:1234:5678:90ab
- However, greater IP address sizes mean packet format changes, which hampers adoption
- Therefore, IPv6 adoption has been very slow (30~34% as of Jan 2021)
  - A long time for a protocol devised in the 1990s
  - Instead, people still mostly use IPv4 with NAT
- But adoption is progressing
  - In late 2020, US government mandated that 80% of federal networks had to be *IPv6-only* by 2025

Working with the Data Plane: OpenFlow

- More abstractly, what is forwarding?
  - Match + Action sequence: packet comes in, match it with something in the forwarding table, take an action based on that.
  - Turn this into a flow table abstraction.

Router's flow table define router's match+action rules



src = *.*.*.*, dest=3.4.*.*	forward(2)
src=1.2.*.*, dest=*.*.*.*	drop
src=10.1.2.3, dest=*.*.*.*	send to controller

* : wildcard

OpenFlow

- OpenFlow is a programmable network protocol that lets you input these match + action rules manually.
- Here are a few examples:

Firewall:

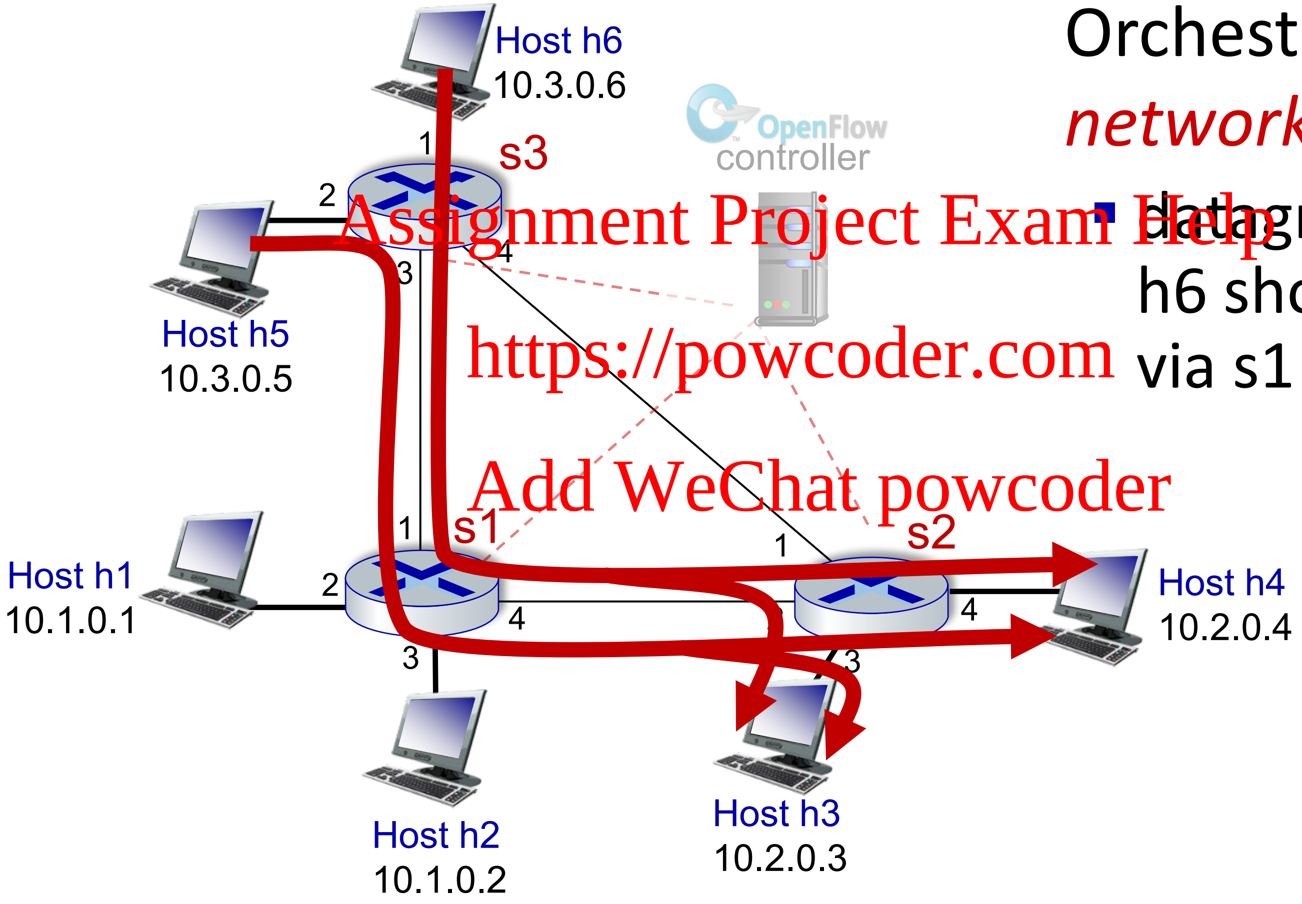
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	*	*	*	*	22	drop

Block (do not forward) all datagrams destined to TCP port 22 (ssh port #)

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	128.119.1.1	*	*	*	*	*	drop

Block (do not forward) all datagrams sent by host 128.119.1.1

OpenFlow example



Orchestrated tables can create *network-wide* behavior, e.g.,:

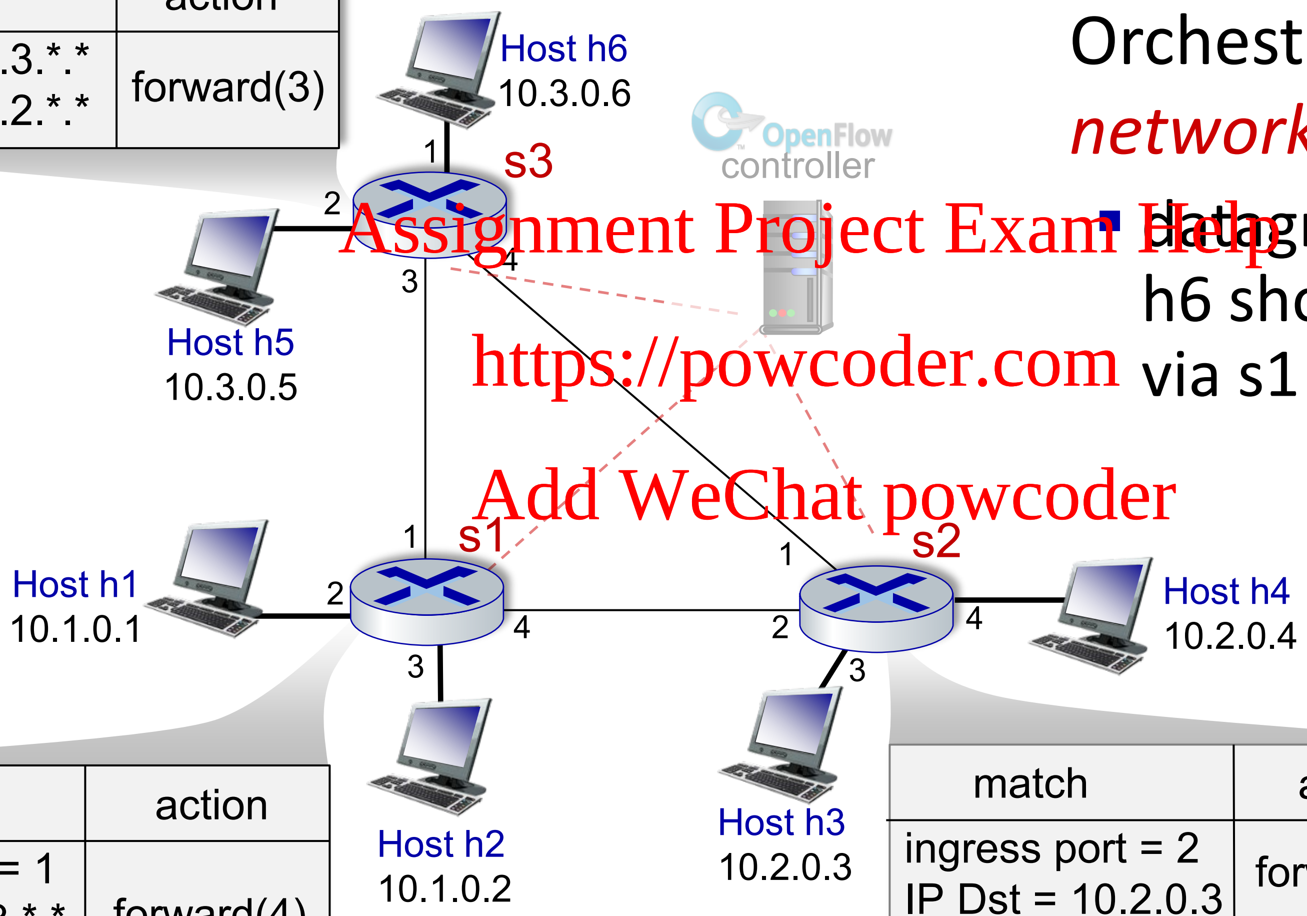
- datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2

OpenFlow example

match	action
IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(3)

Orchestrated tables can create *network-wide* behavior, e.g.,:

- datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2



match	action
ingress port = 1 IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(4)

match	action
ingress port = 2 IP Dst = 10.2.0.3	forward(3)
ingress port = 2 IP Dst = 10.2.0.4	forward(4)

Middleboxes

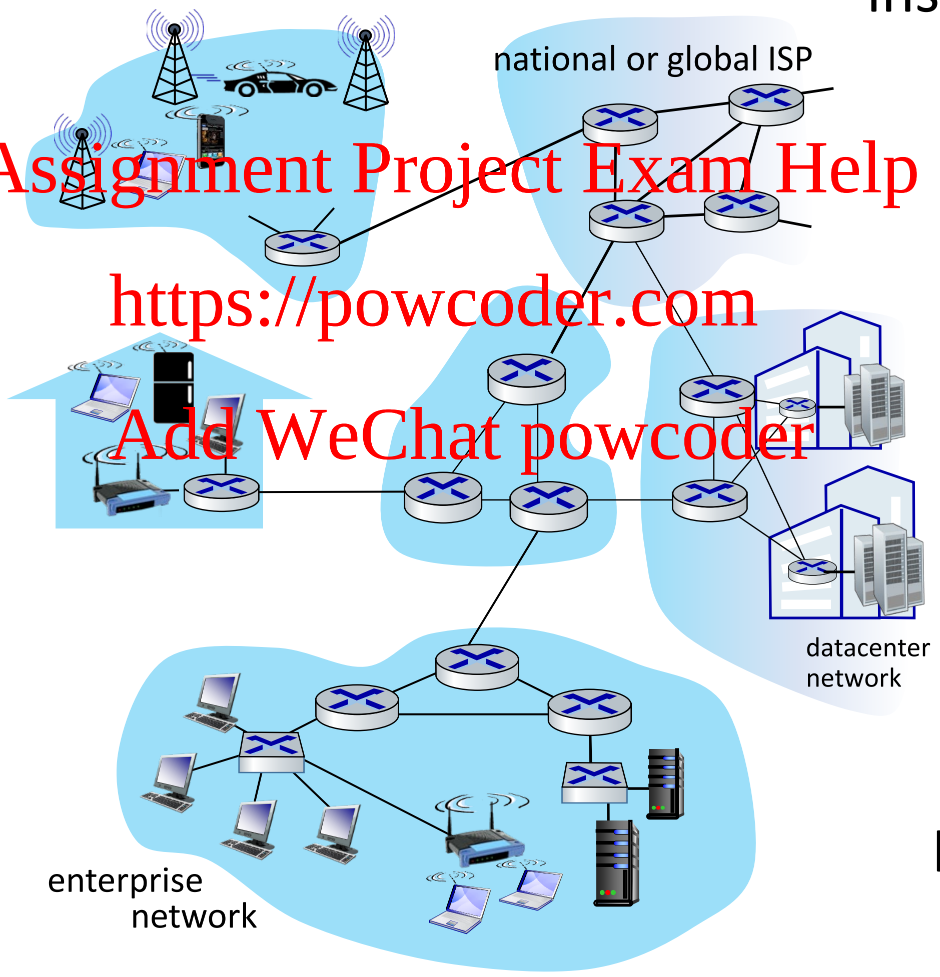
Middlebox (RFC 3234)

"any intermediary box performing functions apart from normal, standard functions of an IP router on the data path between a source host and destination host"

Middleboxes are everywhere!

Firewalls, IDS: corporate, institutional, service providers, ISPs

NAT: home, cellular, institutional



Load balancers: corporate, service provider, data center, mobile nets

Application-specific: service providers, institutional, CDN

Caches: service provider, mobile, CDNs

Middleboxes

- initially: proprietary (closed) hardware solutions
- move towards "whitebox" hardware implementing open APIs
  - move away from proprietary hardware solutions
  - programmable local actions via match+action
  - move towards innovation/differentiation in software
- SDN: (logically) centralized control and configuration management often in private/public cloud
- network functions virtualization (NFV): programmable services over white box networking, computation, storage

Router Algorithms

- How do we figure out where to send IP packets next in a router?
- Find the "longest-prefix matching" route!
- What does this mean?
  - "Routes" in a router take the form: Network/Length → Send on Port X
    - Or alternatively "Length" can be a bit mask, e.g., 24 → 255.255.255.0
  - Find the route with the greatest length "L" that matches the first "L" bits of the IP packet's destination address
  - Then, send the packet on the port listed in the longest matching route

Final Reminders

- Homework 2 is due Monday, February 15
- Project 1 is due Tuesday, February 16 (extended deadline!)
- Project 2 will be released soon and will deal with routers themselves
  - We think this project will be a lot of fun and will give you hands on experience with the inner workings of routers (they're comparatively simple actually!)
  - Most of the router is already implemented, but we leave some specific functionality for you to implement
  - You will be allowed to work in groups of up to 2 people