Lecture 17: Network Layer Addressing, Control Plane, and Routing

COMP 332, Spring 2018

Victoria Manfredi

Acknowledgements: materials adapted from Computer Networking: A Top Down Approach 7th edition: ©1996-2016, J.F Kurose and K.W. Ross, All Rights Reserved as well as from slides by Abraham Matta at Boston University, and some material from Computer Networks by Tannenbaum and Wetherall.

Today

1. Announcements
   - homework 6 due Wed. by 11:59p

1. Addressing
   - usage in routing
   - how to get an IP address
   - IPv6 addressing
   - Dynamic Host Configuration Protocol (DHCP)
   - Network Address Translation (NAT)

2. Control plane
   - overview
   - link state routing

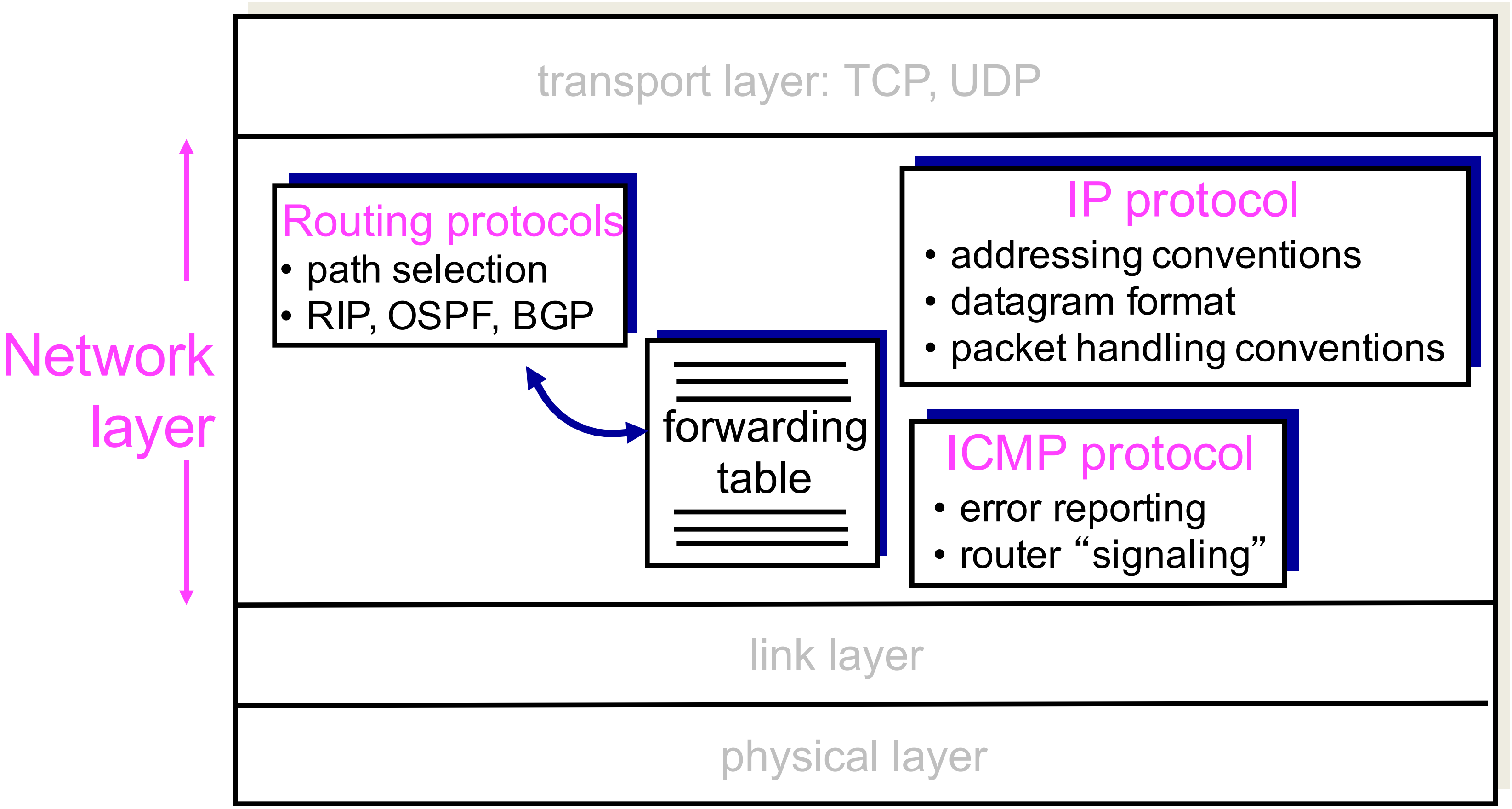
Addressing

USAGE IN ROUTING

Internet's network layer

Network layer functions on hosts and routers

- control plane vs. data plane

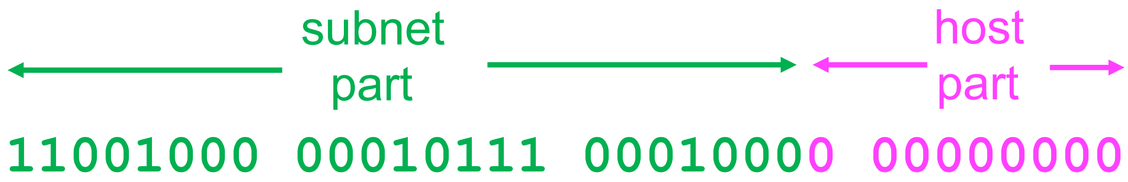


vumanfredi@wesleyan.edu

CIDR: Classless InterDomain Routing

Address format

- a.b.c.d/x, where x is # bits in subnet portion of address
- 200.23.16.0/23



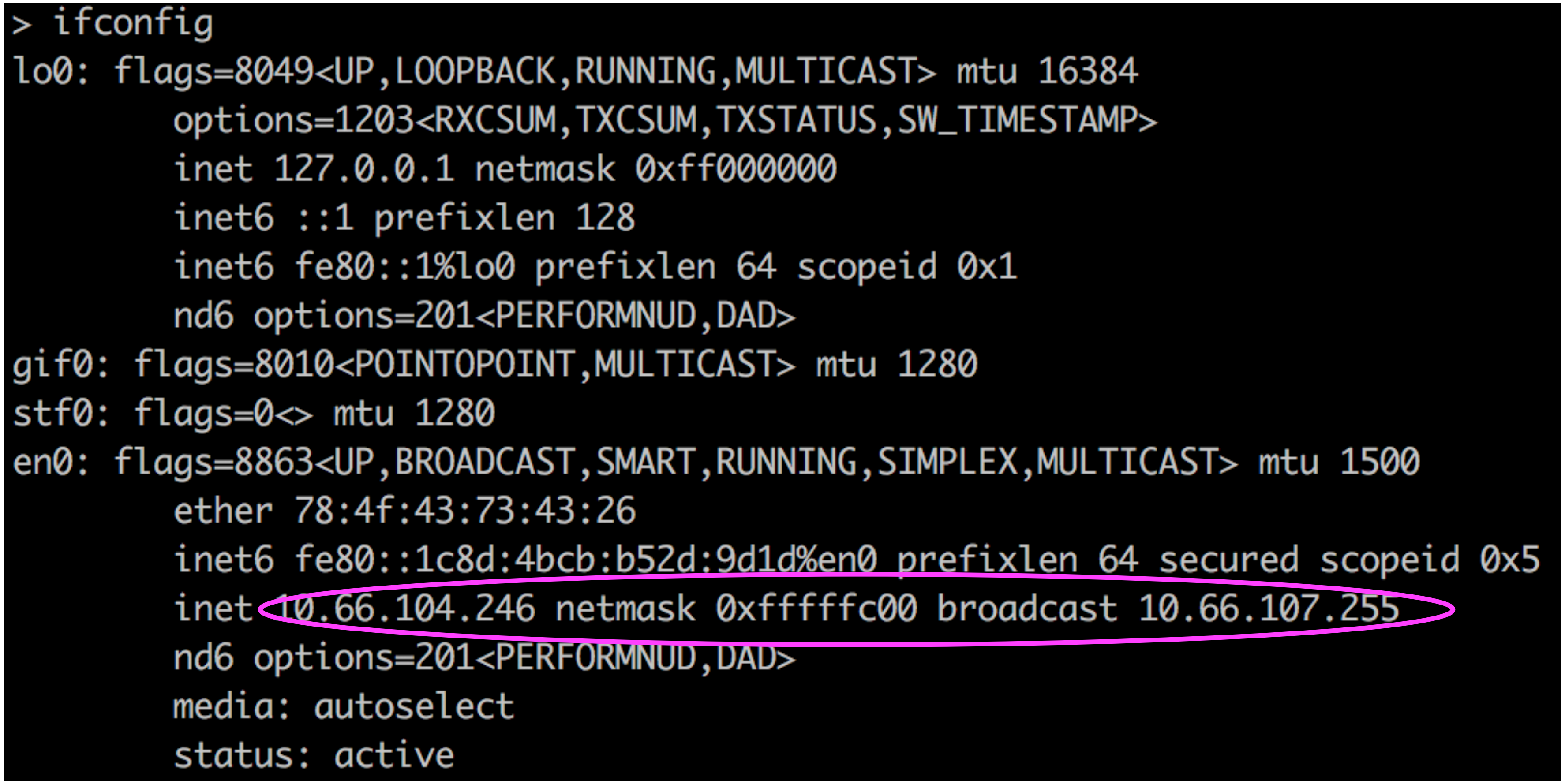
`11001000 00010111 00010000 00000000`

Subnet masks

- define variable partition of host part of addresses
  - /23 subnet mask zeroes out host part of address
    - `11111111 11111111 11111110 00000000`
  - take logical "and" of subnet mask with address to get subnet part
    - 1 AND 1 → 1
    - 1 AND 0 → 0
    - 0 AND 1 → 0
    - 0 AND 0 → 0

Allows more fine-grained division of blocks of addresses than classes

Ifconfig example



```
> ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 78:4f:43:73:43:26
        inet6 fe80::1c8d:4bcb:b52d:9d1d%en0 prefixlen 64 secured scopeid 0x5
        inet 10.66.104.246 netmask 0xfffffc00 broadcast 10.66.107.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

Hex is [0:15] where A=10, B=11, C=12, D=13, E=14, F=15

1111 1111 1111 1111 1111 1100 0000 0000
f f f f f c 0 0

Q: Why is broadcast addr 10.66.107.255?

Routers forward traffic to networks not hosts

Forwarding table

- does not contain row for every dest IP address
- instead computes routes between subnets (blocks of addresses)

Destination Address Range	Link Interface

`11001000 00010111 00010000 00000000`
through
`11001000 00010111 00010111 11111111`
	0

`11001000 00010111 00011000 00000000`
through
`11001000 00010111 00011000 11111111`
	1

`11001000 00010111 00011001 00000000`
through
`11001000 00010111 00011111 11111111`
	2

otherwise	3

What if address ranges don’t divide up nicely?

Longest prefix matching

- use longest address prefix that matches destination address

Destination Address Range	**Link interface**
`11001000 00010111 00010*** *********`	0
`11001000 00010111 00011000 *********`	1
`11001000 00010111 00011*** *********`	2
otherwise	3

Question

DA: 11001000 00010111 00010110 10100001 which interface?

DA: 11001000 00010111 00011000 10101010 which interface?

How big is a routing table for a core router?

From http://www.cidr-report.org/as2.0/

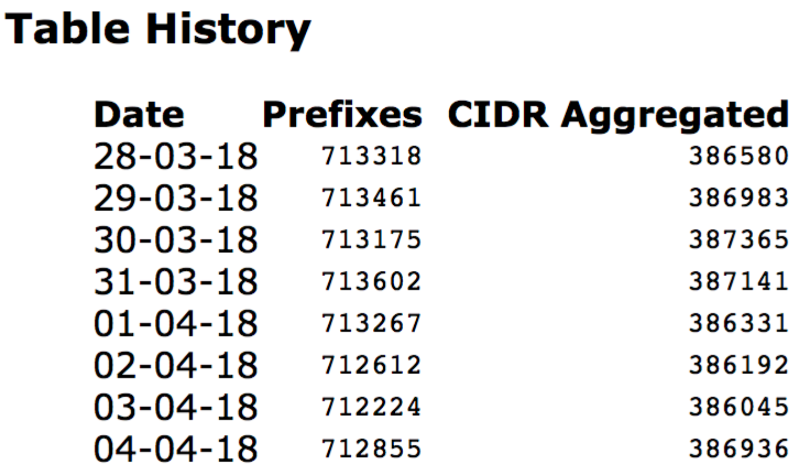
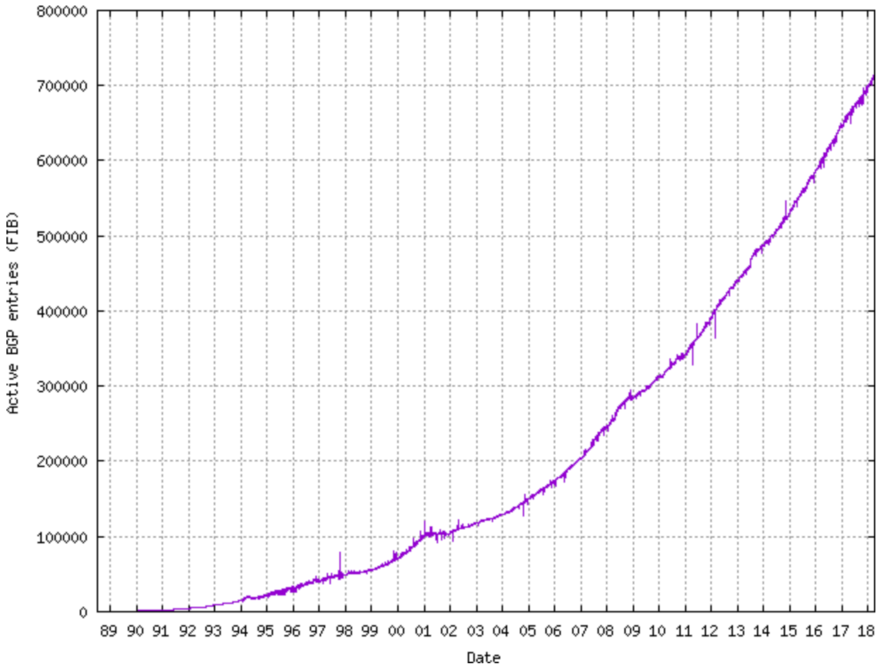
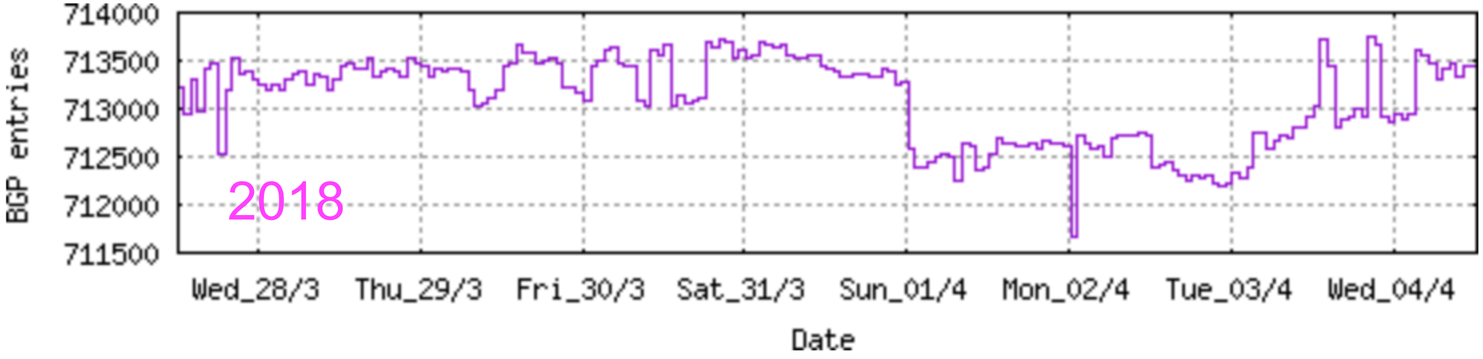


Table History

Date	Prefixes	CIDR Aggregated
28-03-18	713318	386580
29-03-18	713461	386983
30-03-18	713175	387365
31-03-18	713602	387141
01-04-18	713267	386331
02-04-18	712612	386192
03-04-18	712224	386045
04-04-18	712855	386936





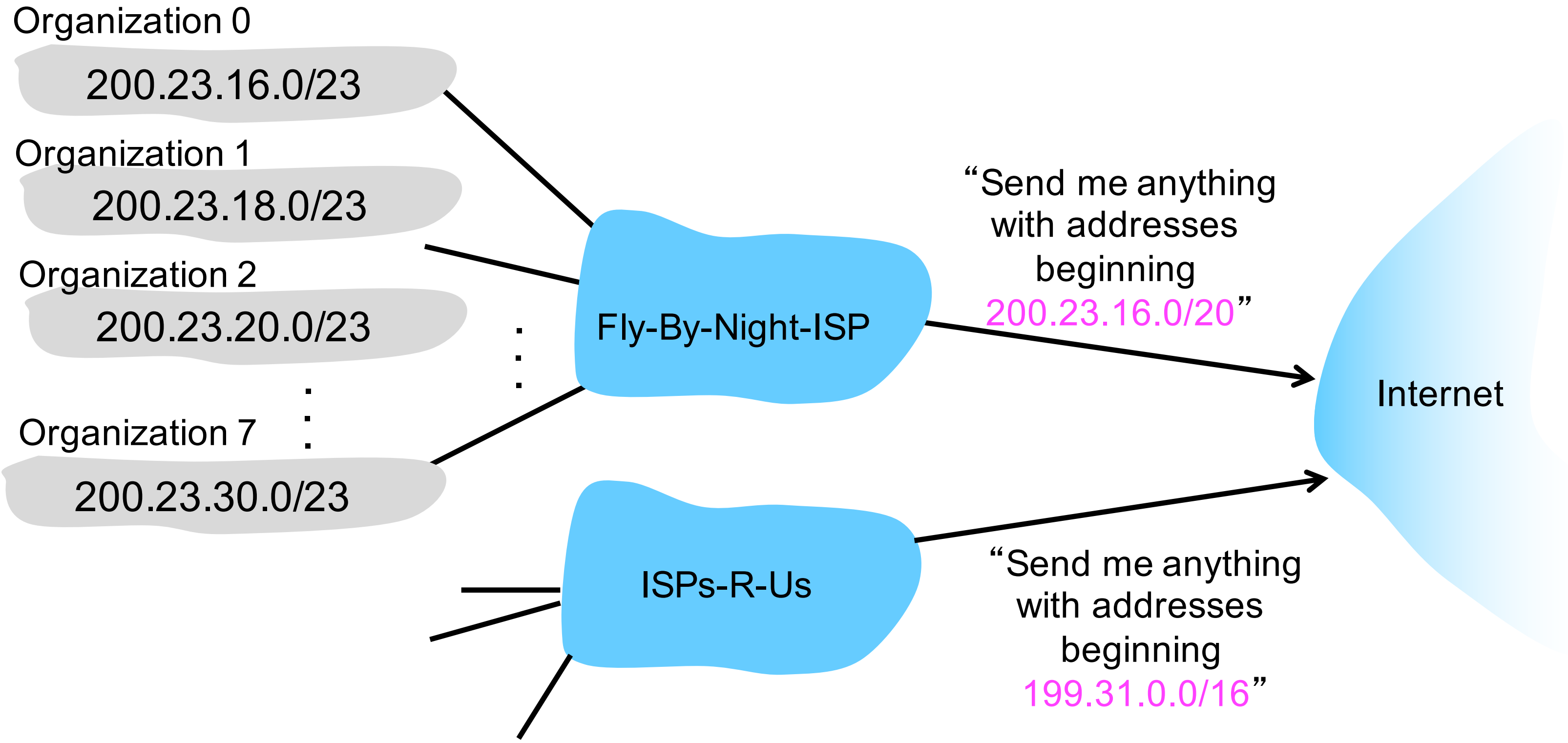
2018

Q: If a core router processes 1million pkts+ per second, how fast does it need to be able to search table?

Hierarchical addressing

Route aggregation

- combine multiple small prefixes into a single larger prefix
- allows efficient advertisement of routing information



Organization 0

200.23.16.0/23

Organization 1

200.23.18.0/23

Organization 2

200.23.20.0/23

Organization 7

200.23.30.0/23

Fly-By-Night-ISP

"Send me anything with addresses beginning 200.23.16.0/20"

Internet

ISPs-R-Us

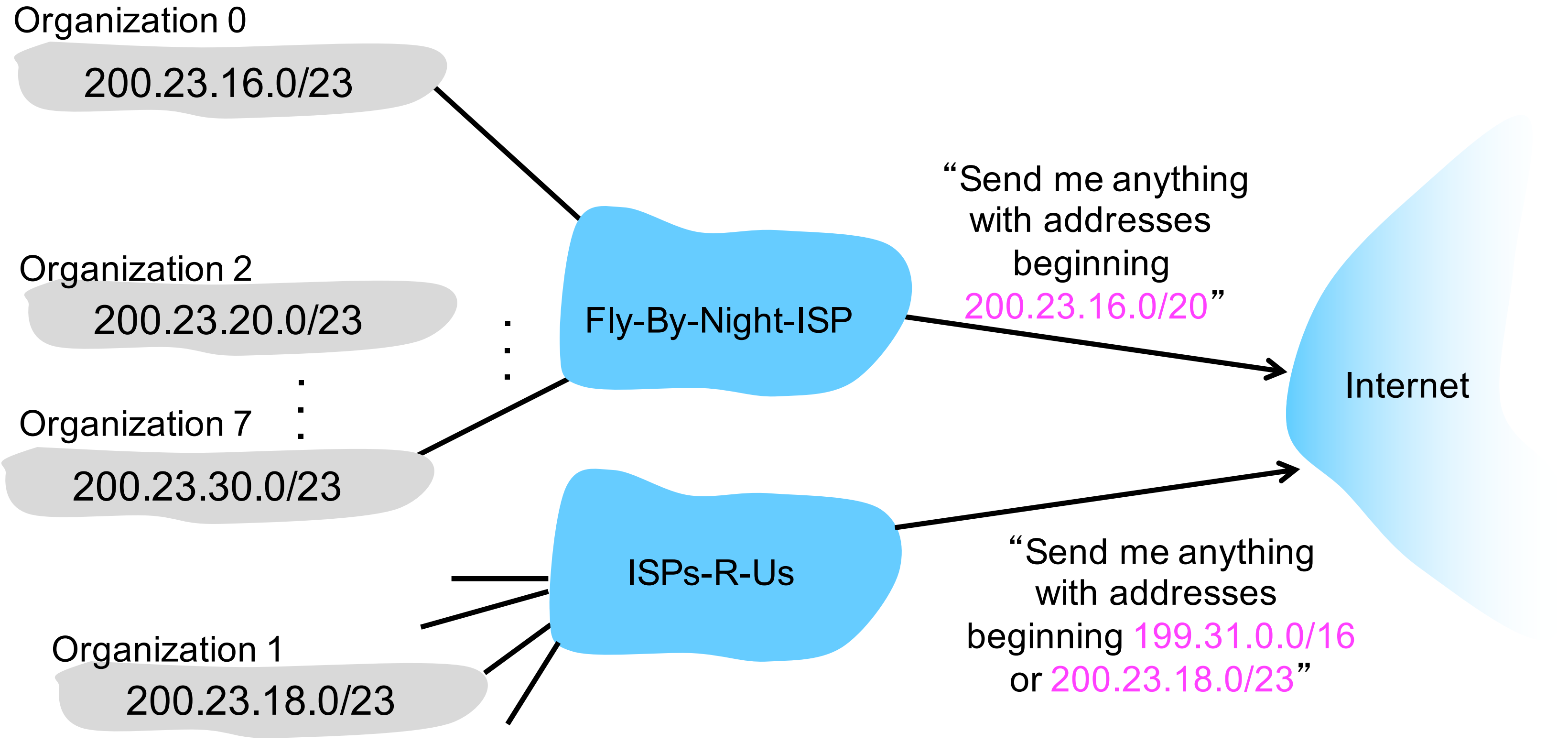
"Send me anything with addresses beginning 199.31.0.0/16"

vumanfredi@wesleyan.edu

Longest prefix matching

More specific routes

- ISPs-R-Us has a more specific route to Organization 1



vumanfredi@wesleyan.edu

Addressing

HOW TO GET AN IP ADDRESS?

vumanfredi@wesleyan.edu

How does ISP get block of addresses?

ICANN

- Internet Corporation for Assigned Names and Numbers
- http://www.icann.org/

ICANN functions

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes
- …

vumanfredi@wesleyan.edu

How does network get net id part of IP address?

Allocated portion of its provider ISP’s address space

ISP's block	11001000 00010111 00010000 00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000 00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010 00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100 00000000	200.23.20.0/23
...	….. ….	….
Organization 7	11001000 00010111 00011110 00000000	200.23.30.0/23

vumanfredi@wesleyan.edu

How does host get an IP address?

Option 1

- hard-coded by system admin in a file on your host

Option 2:

- dynamically get address from a server
  - DHCP: Dynamic Host Configuration Protocol

vumanfredi@wesleyan.edu

We’re running out of IPv4 addresses

Why?

- inefficient use of address space
  - from pre-CIDR use of address classes (A: /8, B: /16, C: /24)
- too many networks (and devices)
  - Internet comprises 100,000+ networks
  - routing tables and route propagation protocols do not scale

Q: how many IPv4 addresses are there?

- 2^{32}

Solutions

- IPv6 addresses
- DHCP: Dynamic Host Configuration Protocol
- NAT: Network Address Translation

vumanfredi@wesleyan.edu

Addressing

IPV6 ADDRESSES

vumanfredi@wesleyan.edu

IPv6 motivation

Initial motivation

- 32-bit address space soon to be completely allocated
- 128-bit IPv6 address: more than 1028x as many IPv4 address

Additional motivation

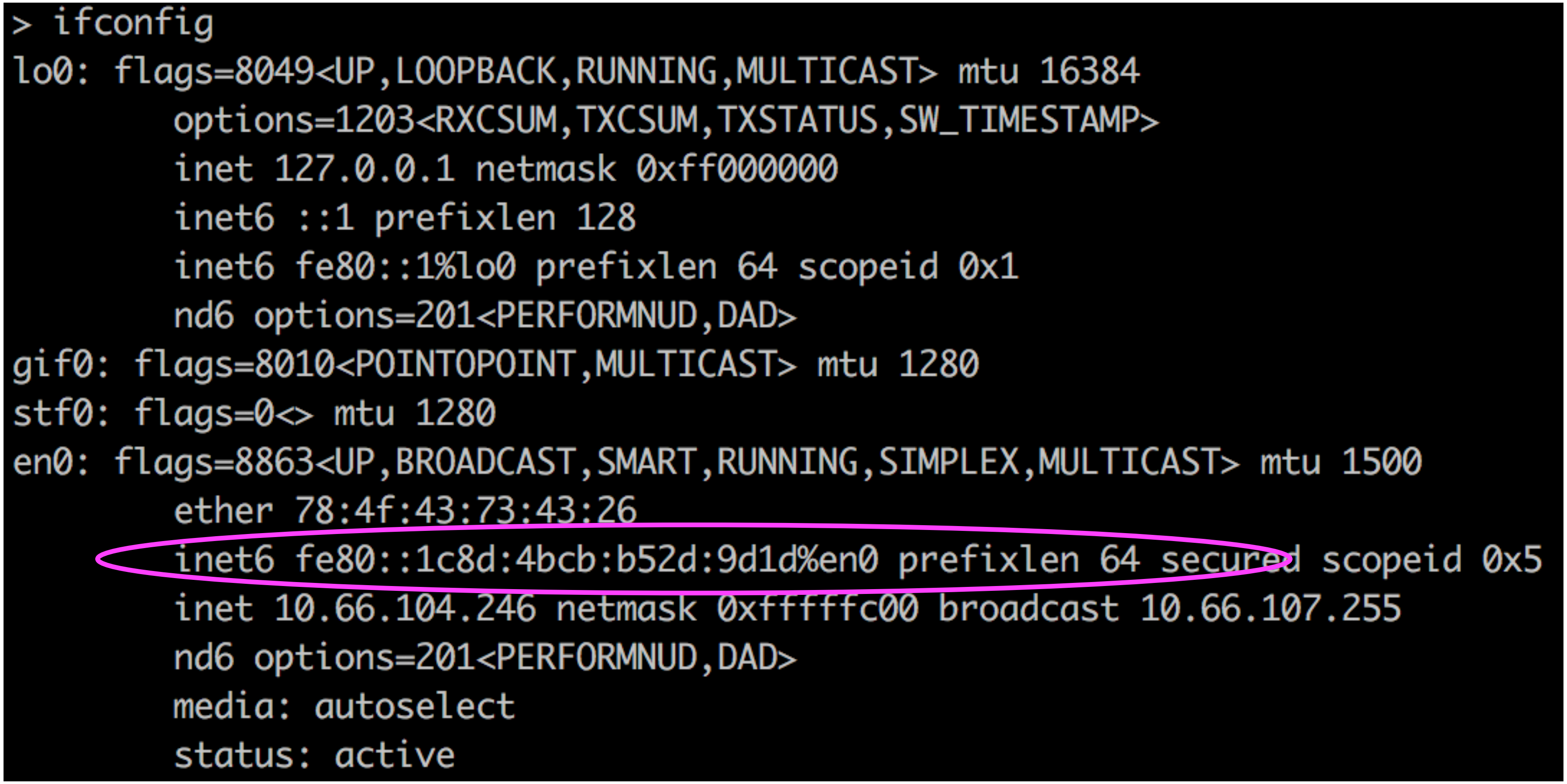
- header format helps speed processing/forwarding
- header changes to facilitate QoS

IPv6 packet format

- fixed-length 40 byte header
- no fragmentation allowed

vumanfredi@wesleyan.edu

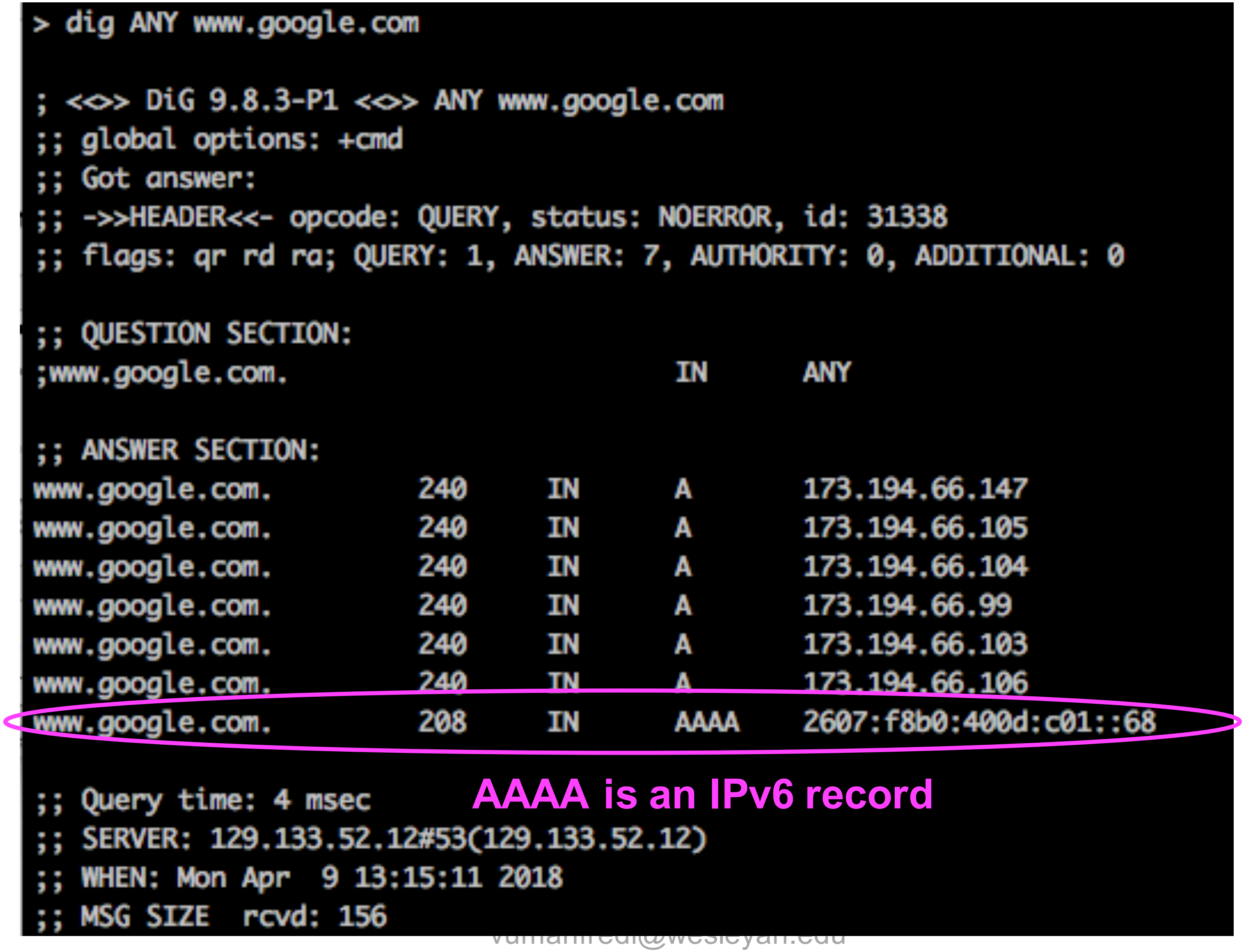
Ifconfig example



```
> ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 78:4f:43:73:43:26
        inet6 fe80::1c8d:4bcb:b52d:9d1d%en0 prefixlen 64 secured scopeid 0x5
        inet 10.66.104.246 netmask 0xfffffc00 broadcast 10.66.107.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

vumanfredi@wesleyan.edu

Dig www.google.com ANY



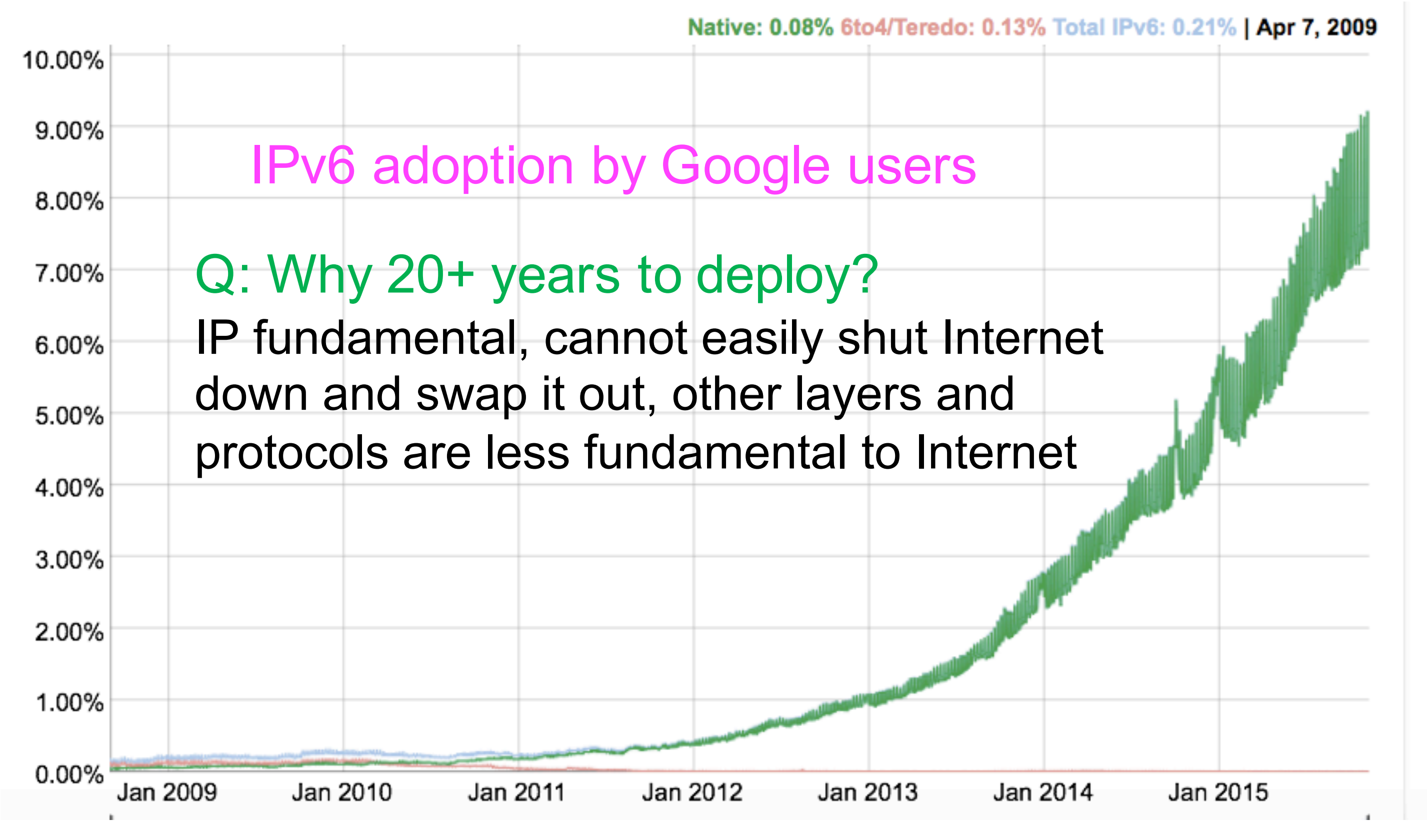
```
> dig ANY www.google.com

; <<>> DiG 9.8.3-P1 <<>> ANY www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31338
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.google.com.                        IN      ANY

;; ANSWER SECTION:
www.google.com.         240     IN      A       173.194.66.147
www.google.com.         240     IN      A       173.194.66.105
www.google.com.         240     IN      A       173.194.66.104
www.google.com.         240     IN      A       173.194.66.99
www.google.com.         240     IN      A       173.194.66.103
www.google.com.         240     IN      A       173.194.66.106
www.google.com.         208     IN      AAAA    2607:f8b0:400d:c01::68

;; Query time: 4 msec
;; SERVER: 129.133.52.12#53(129.133.52.12)
;; WHEN: Mon Apr  9 13:15:11 2018
;; MSG SIZE  rcvd: 156
```

AAAA is an IPv6 record

IPv6 deployment

Standardized ~1998

- 2008: IPv6 < 1% of Internet traffic
- 2011: IPv6 increasingly implemented in OS, mandated by governments and cell providers for new network devices, ….
- as recently as last year, Wesleyan did not support IPv6



IPv6 adoption by Google users

Q: Why 20+ years to deploy?

IP fundamental, cannot easily shut Internet down and swap it out, other layers and protocols are less fundamental to Internet

Addressing

DYNAMIC HOST CONFIGURATION PROTOCOL

vumanfredi@wesleyan.edu

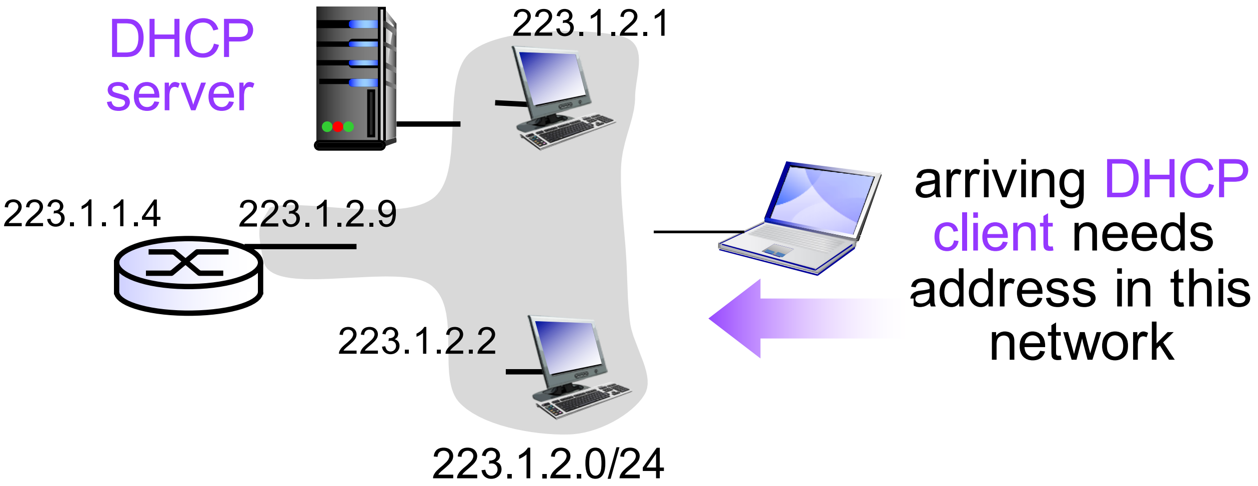
DHCP: Dynamic Host Configuration Protocol

Goal

- let host dynamically obtain IP addr from server when it joins network

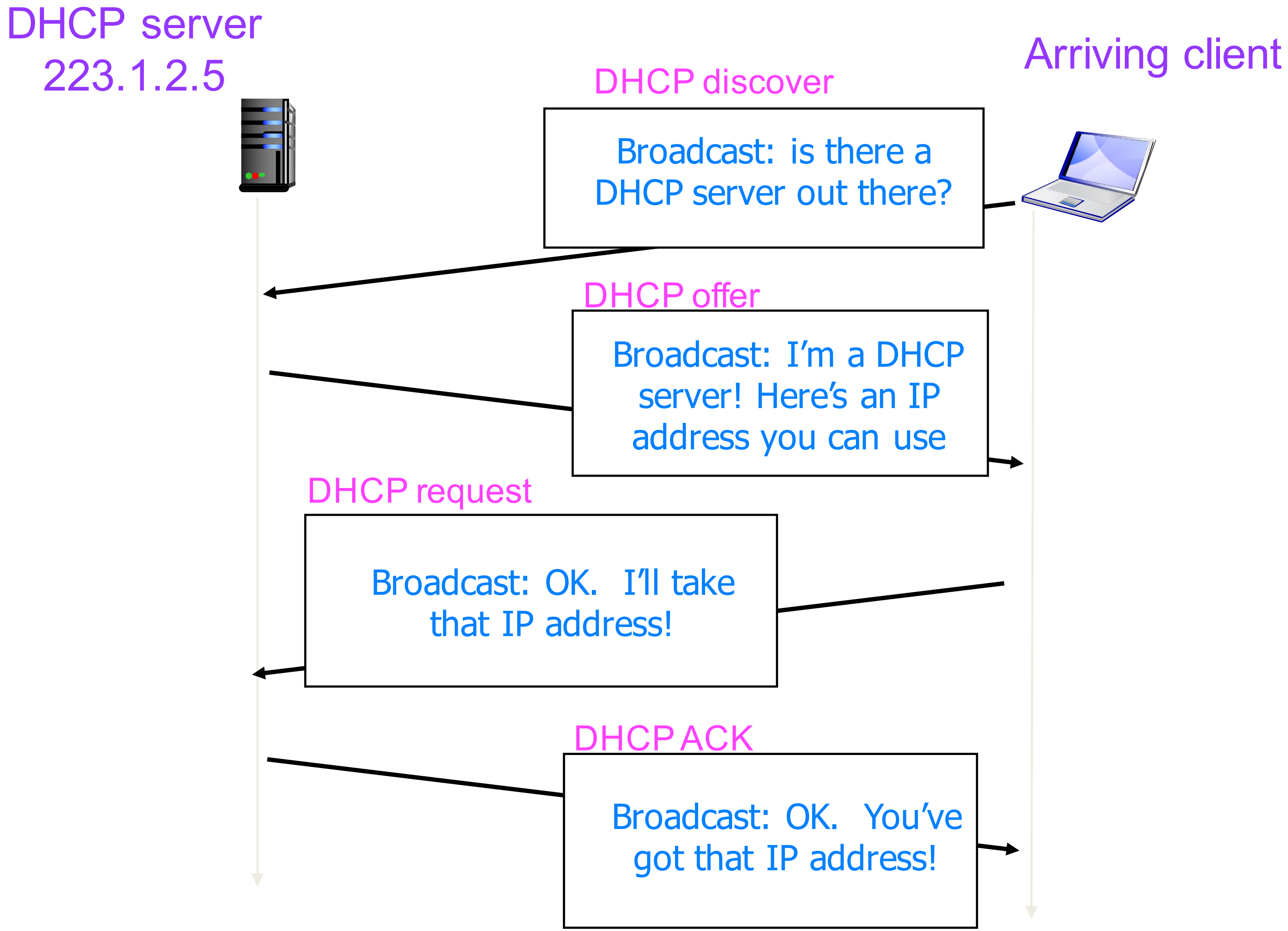
Benefits

- reuse of addresses by different hosts
  - only hold address while connected to network
  - host can renew its lease on address in use
- support for mobile users who want to join network



arriving DHCP client needs address in this network

Client-server scenario



DHCP server
223.1.2.5

Arriving client

DHCP discover

Broadcast: is there a DHCP server out there?

DHCP offer

Broadcast: I'm a DHCP server! Here's an IP address you can use

DHCP request

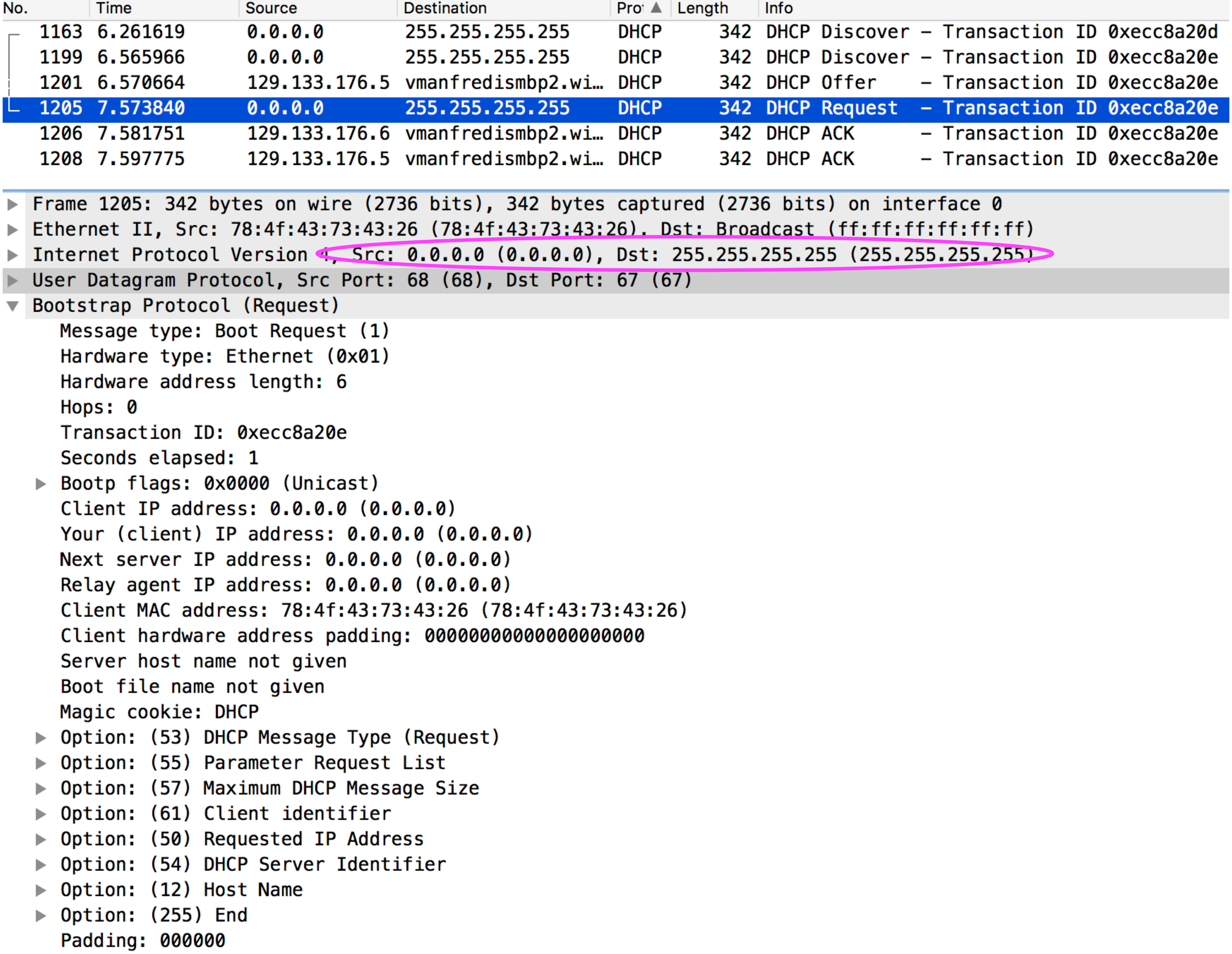
Broadcast: OK. I'll take that IP address!

DHCP ACK

Broadcast: OK. You've got that IP address!

Q: What layer is DHCP in?

Q: What transport layer protocol does DHCP run over?



```
No.    Time       Source          Destination          Prot  Length  Info
  1163 6.261619   0.0.0.0         255.255.255.255      DHCP     342  DHCP Discover - Transaction ID 0xecc8a20d
  1199 6.565966   0.0.0.0         255.255.255.255      DHCP     342  DHCP Discover - Transaction ID 0xecc8a20e
  1201 6.570664   129.133.176.5   vmanfredismbp2.wi…   DHCP     342  DHCP Offer    - Transaction ID 0xecc8a20e
  1205 7.573840   0.0.0.0         255.255.255.255      DHCP     342  DHCP Request  - Transaction ID 0xecc8a20e
  1206 7.581751   129.133.176.6   vmanfredismbp2.wi…   DHCP     342  DHCP ACK      - Transaction ID 0xecc8a20e
  1208 7.597775   129.133.176.5   vmanfredismbp2.wi…   DHCP     342  DHCP ACK      - Transaction ID 0xecc8a20e

▶ Frame 1205: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface 0
▶ Ethernet II, Src: 78:4f:43:73:43:26 (78:4f:43:73:43:26), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▶ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
▶ User Datagram Protocol, Src Port: 68 (68), Dst Port: 67 (67)
▼ Bootstrap Protocol (Request)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xecc8a20e
    Seconds elapsed: 1
  ▶ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: 78:4f:43:73:43:26 (78:4f:43:73:43:26)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▶ Option: (53) DHCP Message Type (Request)
  ▶ Option: (55) Parameter Request List
  ▶ Option: (57) Maximum DHCP Message Size
  ▶ Option: (61) Client identifier
  ▶ Option: (50) Requested IP Address
  ▶ Option: (54) DHCP Server Identifier
  ▶ Option: (12) Host Name
  ▶ Option: (255) End
    Padding: 000000
```

Addressing

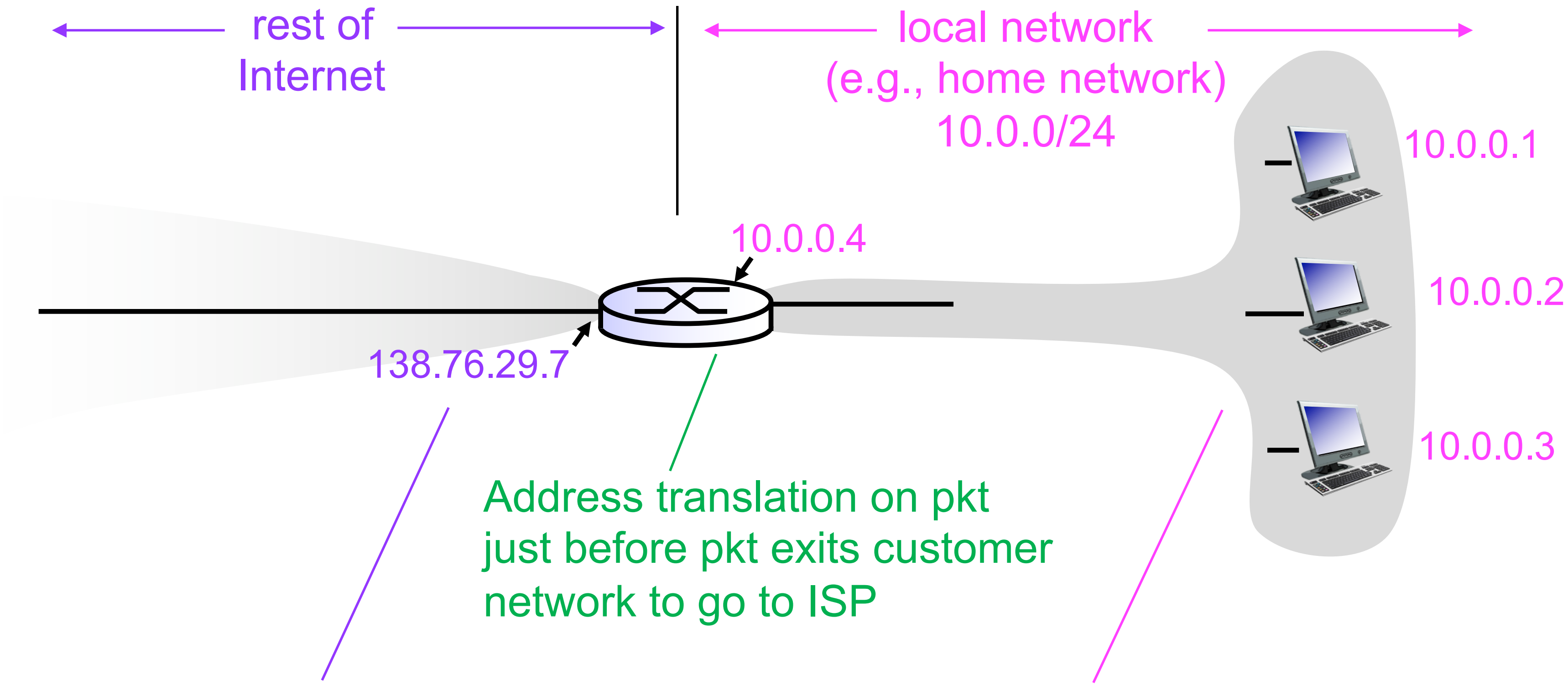
NETWORK ADDRESS TRANSLATION

vumanfredi@wesleyan.edu

Network Address Translation (NAT)

Motivation

– local network uses 1 IP address as far as outside world is concerned



Address translation on pkt just before pkt exits customer network to go to ISP

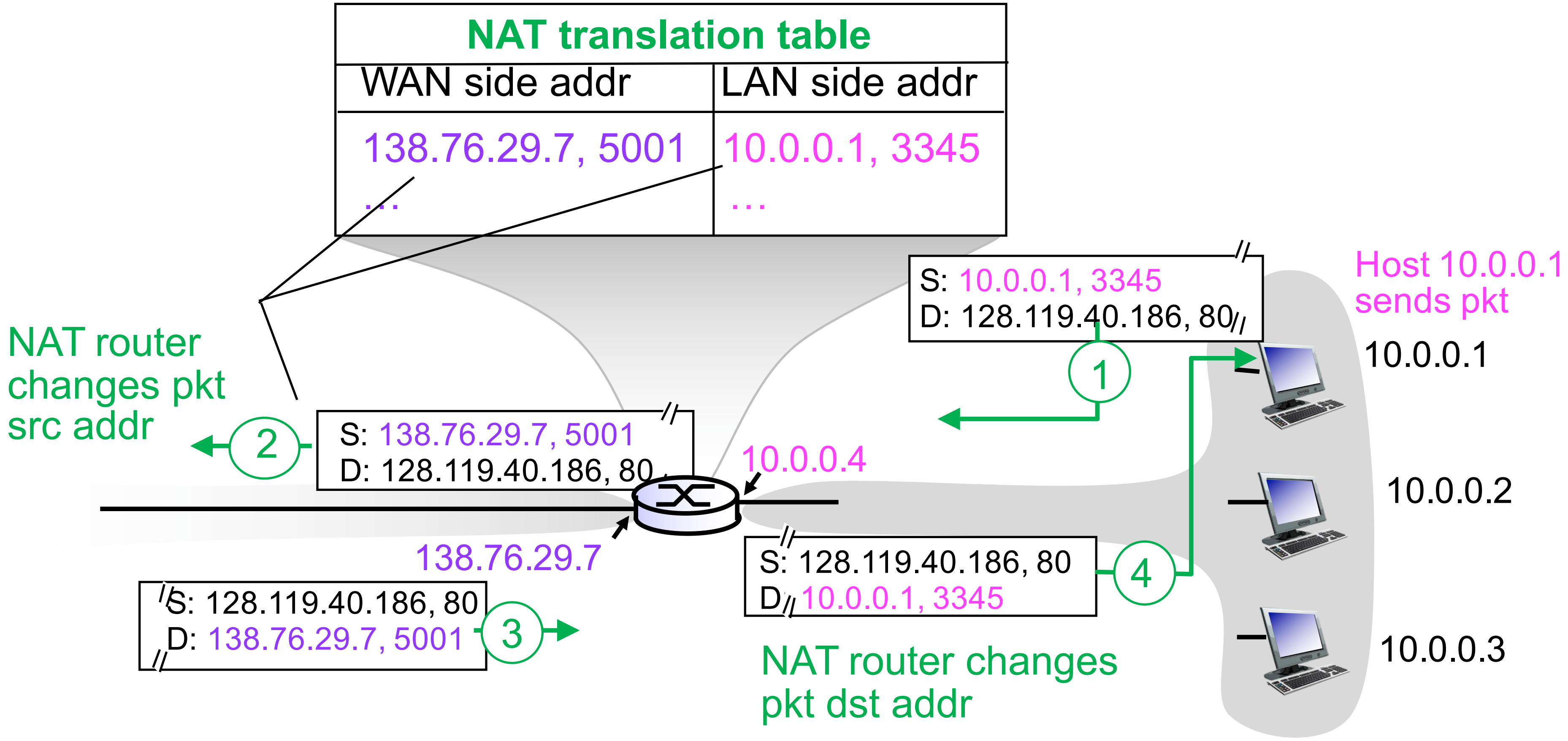
Externally: all packets leaving local network have same single source NAT IP address: 138.76.29.7, different source port #s

Internally: each host gets unique address from set of private subnet addresses, 10.0.0/24

NAT implementation on router

Outgoing packets

Replace (src IP addr, port #) to (NAT IP addr, new port #)



NAT translation table

WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
…	…

Incoming packets

Replace (NAT IP addr, new port #) in dst fields with corresponding (src IP addr, port #) in NAT table

Q: # of connections supported with 16-bit port #?

Q: Why was NAT was designed this way? Can ICMP traffic reach host behind NAT router?

Most traffic is TCP or UDP

NAT pros and cons

Pros

- don't need range of addresses from ISP
  - just one public IP address for all devices
- change private addresses of devices
  - without notifying outside world
- change ISP
  - without changing addresses of devices in local network
- security
  - devices inside local network not explicitly addressable or visible

Cons: NAT is controversial!

- routers should only process up to network layer
- address shortage should be solved by IPv6
- violates e2e argument
  - app designers (e.g., p2p) must account for NAT usage
- creates a strange kind of connection-oriented network
- NAT traversal
  - how to connect to server behind NAT? Problems for VOIP, FTP, …

Recall RFC 1958 architectural principles

1. Make sure it works: don’t finalize standard before implementing
2. Keep it simple: Occam’s razor
3. Make clear choices: choose one way to do it
4. Exploit modularity: e.g., protocol stack
5. Expect heterogeneity: different hardware, links, applications
6. Avoid static options and parameters: better to negotiate
7. Look for a good not necessarily perfect design: onus is on the designers with the outliers to work around design
8. Be strict when sending and tolerant when receiving
9. Think about scalability: no centralized databases, load evenly spread over resources
10. Consider performance and cost: if bad, no one will use network

vumanfredi@wesleyan.edu

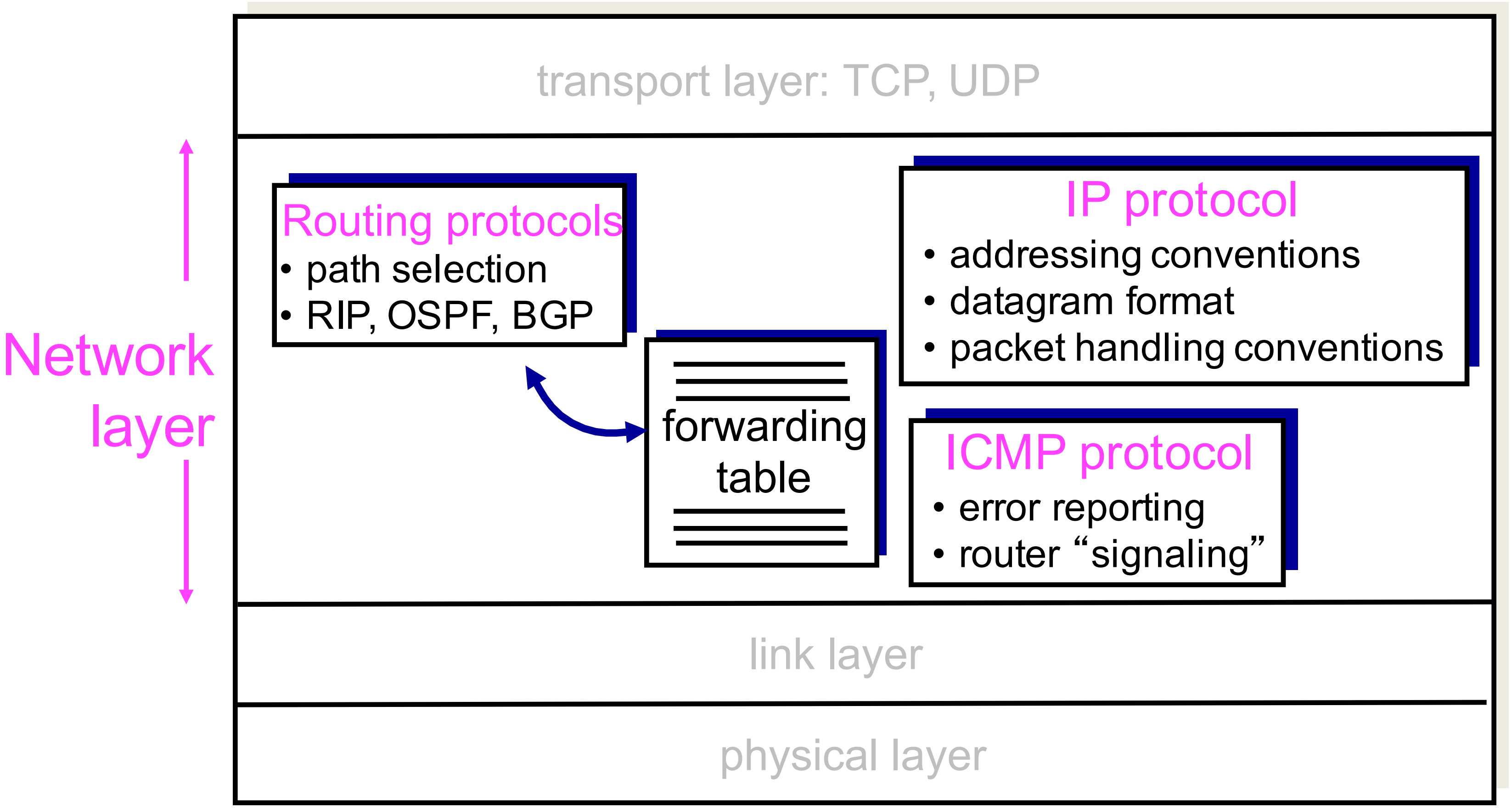
Control Plane
OVERVIEW

vumanfredi@wesleyan.edu

Internet’s network layer

Network layer functions on hosts and routers

- control plane vs. data plane



transport layer: TCP, UDP

Network layer

Routing protocols
- path selection
- RIP, OSPF, BGP

forwarding table

IP protocol
- addressing conventions
- datagram format
- packet handling conventions

ICMP protocol
- error reporting
- router “signaling”

link layer

physical layer

vumanfredi@wesleyan.edu

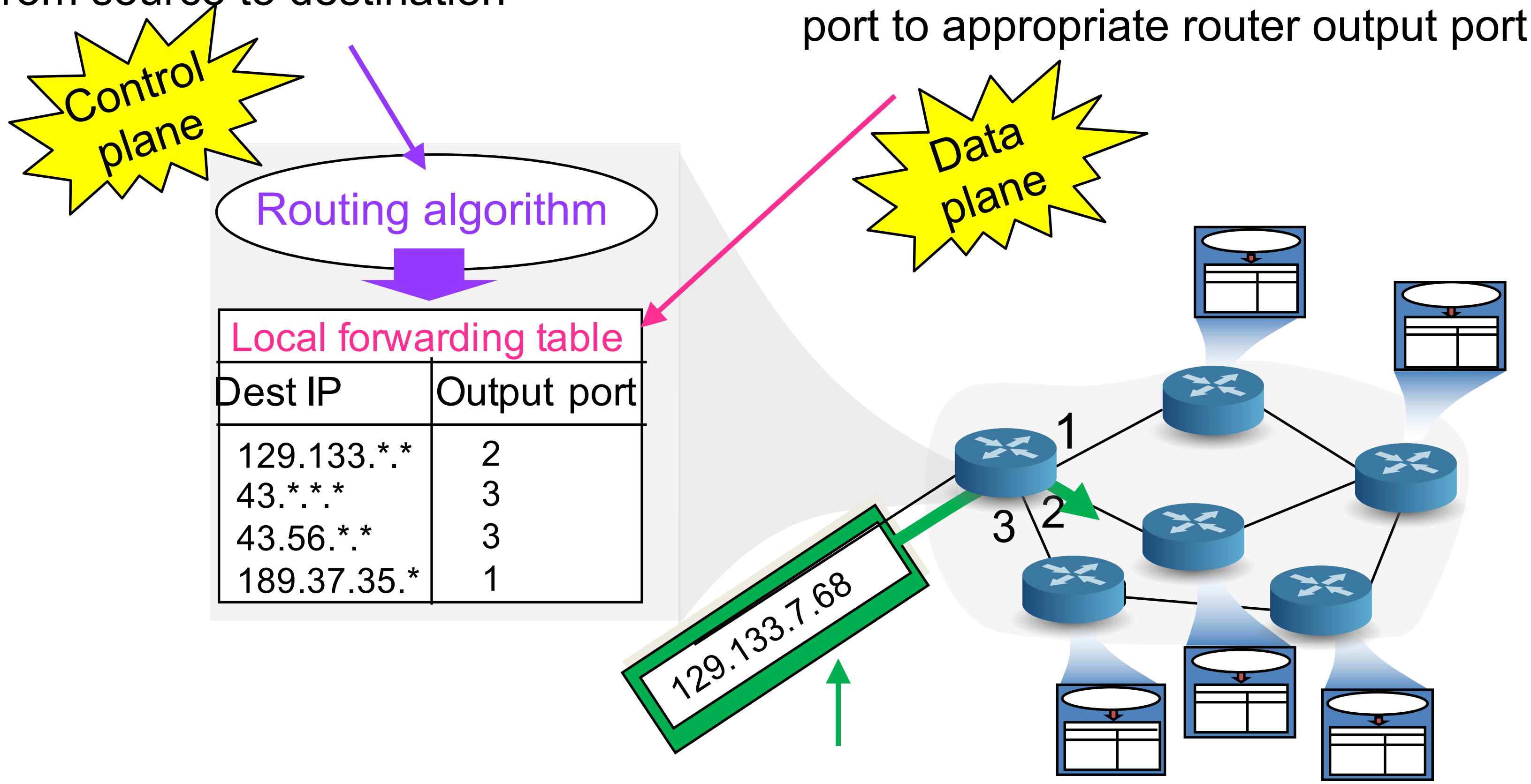
Control vs. data plane functions

Routing (slower time scale)

- determine route taken by packets from source to destination

Forwarding (faster time scale)

- move packets from router’s input port to appropriate router output port



Local forwarding table

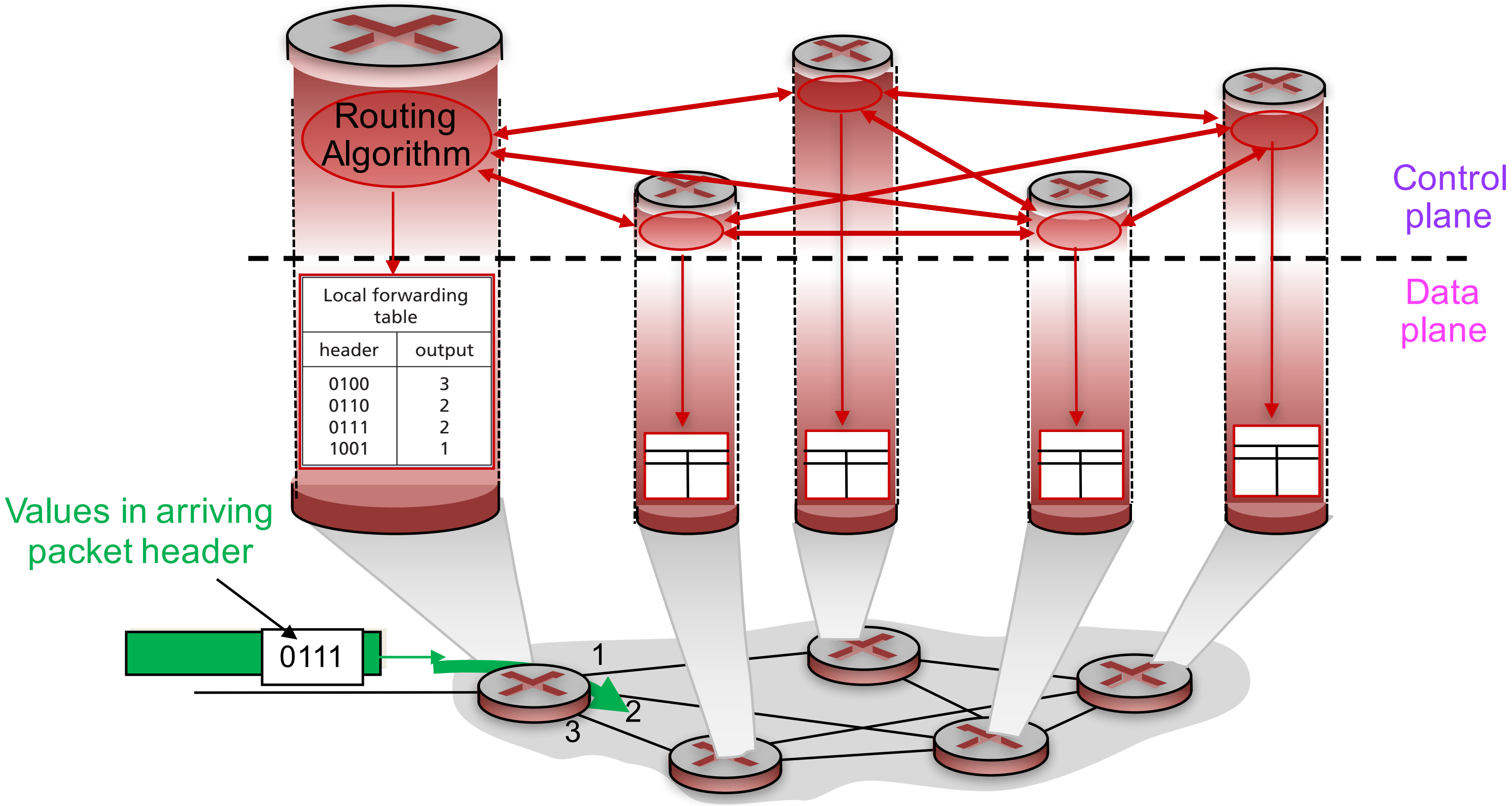
Dest IP	Output port
129.133.*.*	2
43.*.*.*	3
43.56.*.*	3
189.37.35.*	1

Dest IP addr in header of arriving packet

vumanfredi@wesleyan.edu

Approach 1: per-router control plane

Individual routing algorithm components in each and every router interact in the control plane to compute forwarding tables



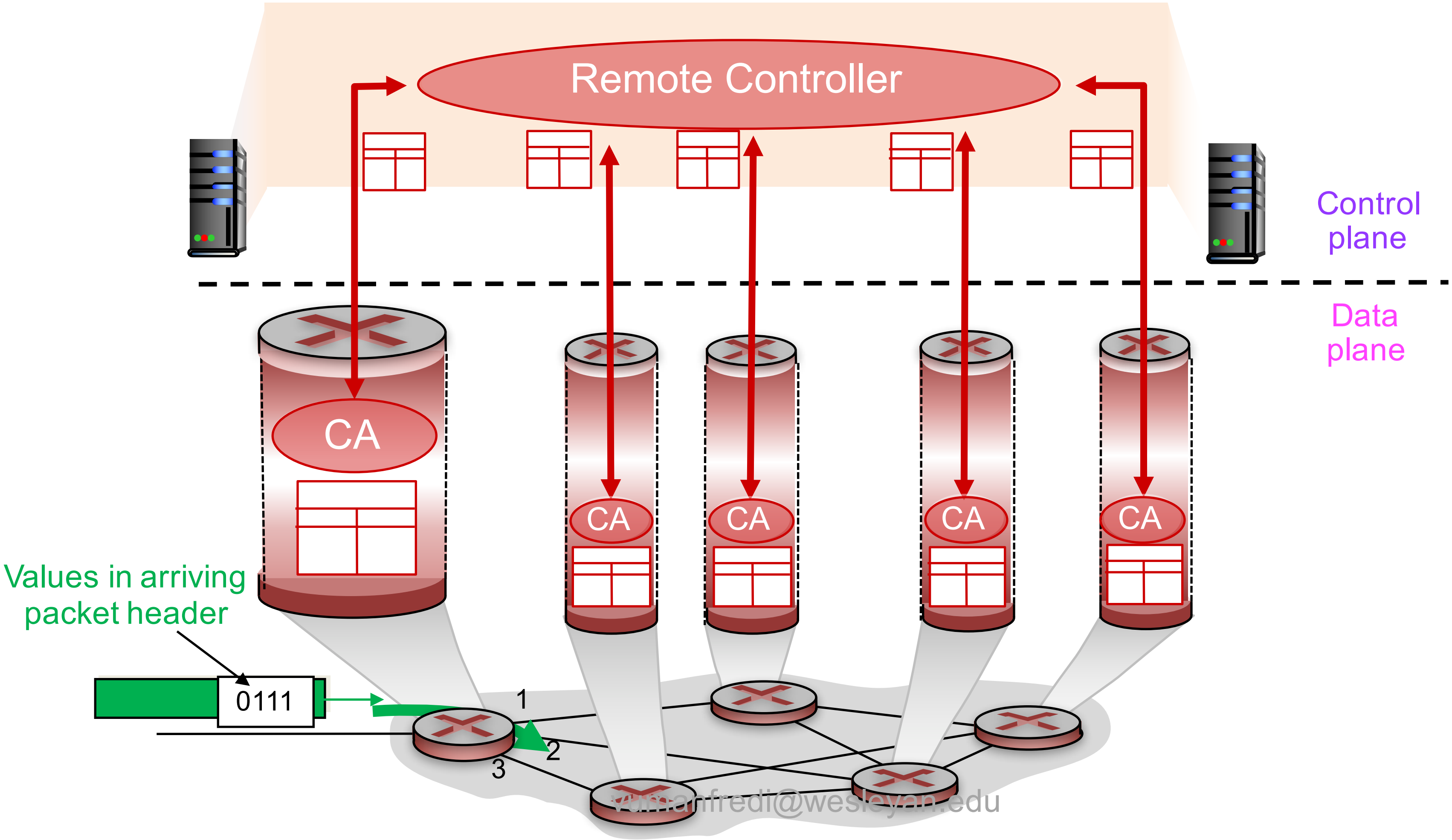
Local forwarding table

header	output
0100	3
0110	2
0111	2
1001	1

vumanfredi@wesleyan.edu

Approach 2: logically centralized control plane

A distinct (typically remote) controller interacts with local control agents (CAs) in routers to compute forwarding tables



vumanfredi@wesleyan.edu

Routing protocols

Goal

- determine “good” path from sending hosts to receiving host, through network of routers

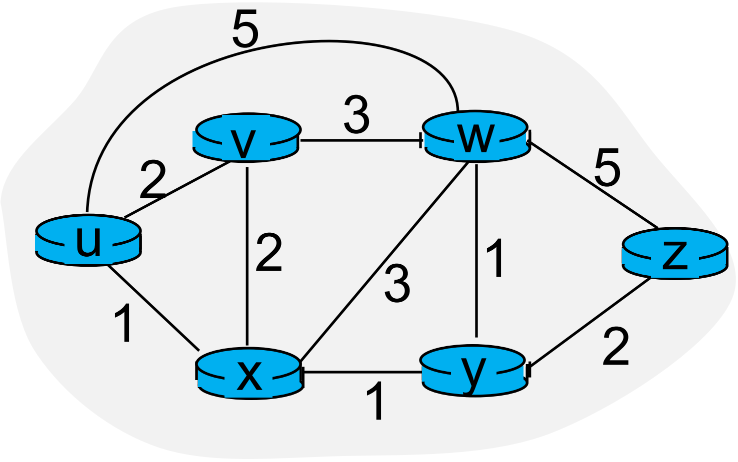
Path

- sequence of routers packets will traverse in going from given initial source host to given final destination host

“Good”

- least “cost”, “fastest”, “least congested”, …
- correctness constraints
  - no loops
  - no dead-ends

Abstract network as a graph

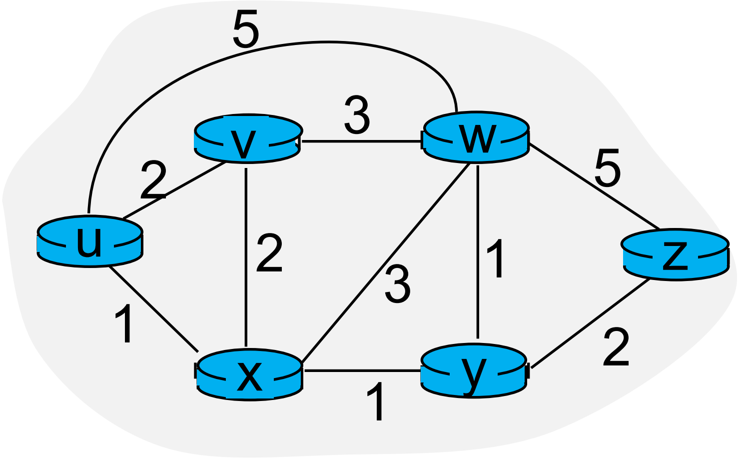


Graph: G = (N,E)

N = set of routers
= { u, v, w, x, y, z }

E = set of links
={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

Link costs



$c(x,x')$ = cost of link (x,x')
$c(w,z) = 5$

Q: how to set cost?
Cost could always be 1, inversely related to bandwidth, inversely related to congestion, …

Cost of path $(x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

Q: What's the least-cost path between u and z?

Routing algorithm: algorithm that finds least-cost path

Classifying routing algorithms

Global or decentralized information?

Global

- **link state** algorithms
- all routers have complete topology, link cost info

Decentralized

- **distance vector** algorithms
- router knows physically-connected neighbors, link costs to neighbors
- iterative computation
- exchange info with neighbors

Static or dynamic topology?

Static

- routes change slowly over time

Dynamic

- routes change more quickly
- periodic update in response to link cost changes

Both link state and distance vector algorithms used on Internet.

We'll first cover abstractly and then talk about them in context of specific Internet protocols (OSPF, BGP, RIP, …)