

ECE 356/COMPSI 356

Computer Network Architecture

Routing Wrap-Up.
Miscellaneous IP Topics.

Wednesday October 9th, 2019

Welcome Back from the Break!



Recap

- Last lecture:
 - Link state routing
 - Inter-domain routing
- Readings for this lecture: **PD 3.2.7, 3.2.9, 4.1.3, sidebar on pg. 335**

This Lecture

- Routing wrap-up and review
 - Textbook material: **PD 3.3.1-3.3.3**
- Finishing up a collection of disjoint but important IP-related topics
 - Dynamic Host Configuration Protocol (DHCP)
 - Network Address Translation (NAT)
 - IPv6
 - IP tunnels

Routing: Key Points to Remember

- Difference between routing and forwarding
 - Routing protocols establish forwarding tables at routers
- Underlying routing protocols are *distributed algorithms* for determining paths from a source to a destination
 - Connections to graph theory and foundational graph algorithms
 - Distributed decentralized operation is imperative
- Inter-domain and intra-domain routing
 - Internet as a collection of Autonomous Systems (ASs)
 - Each AS can run its own routing protocol. BGP runs in-between the ASs

Difference Between Distance-Vector and Link-State Routing

- In distance-vector routing:
 - Each node talks only to its directly connected neighbors
 - Tells them everything it has learned
 - Distances to all neighbors
- In link-state routing:
 - Each nodes talks to all other nodes
 - Tells them only what it knows for sure
 - State of its own links

Distance Vector Routing: Key Points to Remember (1/2)

- Routers talk *only to directly connected neighbors*
 - Messages do not get propagated across the entire network
- Advertise learned distances to all nodes in the network: *distance vectors*
- Algorithms to know: distance vector algorithm

Distance Vector Routing: Key Points to Remember (2/2)

- Distance vector routing suffers from a count-to-infinity problem
 - One solution to it is split-horizon advertisement
- Routing Information Protocol (RIP) is a straightforward implementation of Distance Vector Routing
 - Used in Lab 3

Link State Routing: Key Points to Remember

- With link state routing, all nodes form the full picture of network connectivity
- Link state information is flooded across the network, using reliable flooding algorithms
- Algorithms to know: Dijkstra, forward search algorithms
- OSPF is a commonly deployed link state routing protocol
 - Supports authentication (why is it needed?)
 - Supports hierarchy (why is it needed?)

Inter-Domain Routing: Key Points to Remember

- Inter-domain routing connects different ASs. It is subject to both technical and economics/policy constraints
- In inter-domain routing, we find *a* path to a destination, rather than the shortest path
- BGP is the one inter-domain routing protocol used in the Internet
- BGP advertises *complete paths to a destination*, as a sequence of ASs

Lecture Outline

- Routing wrap-up and review
- Finishing up a collection of disjoint but important IP-related topics
 - **Dynamic Host Configuration Protocol (DHCP)**
 - Network Address Translation (NAT)
 - IPv6
 - IP tunnels

Dynamic Host Configuration (DHCP)

- How your laptop and mobile phone get IP addresses on a campus network
 - Why you did not need to find out many of the details of IP and IP forwarding until this class
 - Why your IP address can change once in a while
 - Widely used
- A *network management* protocol
 - Another network management protocol we already studied: ICMP
- A client-server protocol

Dynamic Assignment of IP Addresses

- Avoid manual IP configuration
 - Inconvenient, error prone
 - *Note that this is one of the themes of this class: static approaches exist, but are not practical in reality*
- Dynamic assignment of IP addresses is desirable
- IP addresses are assigned on-demand
 - Examples to keep in mind: Duke visitors network, RDU WiFi

DHCP: An Introduction

- Designed in 1993
 - Precursor host configuration protocols existed since 1984
- DHCP uses a server and a series of *relays*
- DHCP client can acquire all IP configuration parameters
 - Default router, network mask, DNS resolver
- Supports temporary allocation (“leases”) of IP addresses
- Runs over UDP

Client Acquiring IP Configuration Parameters

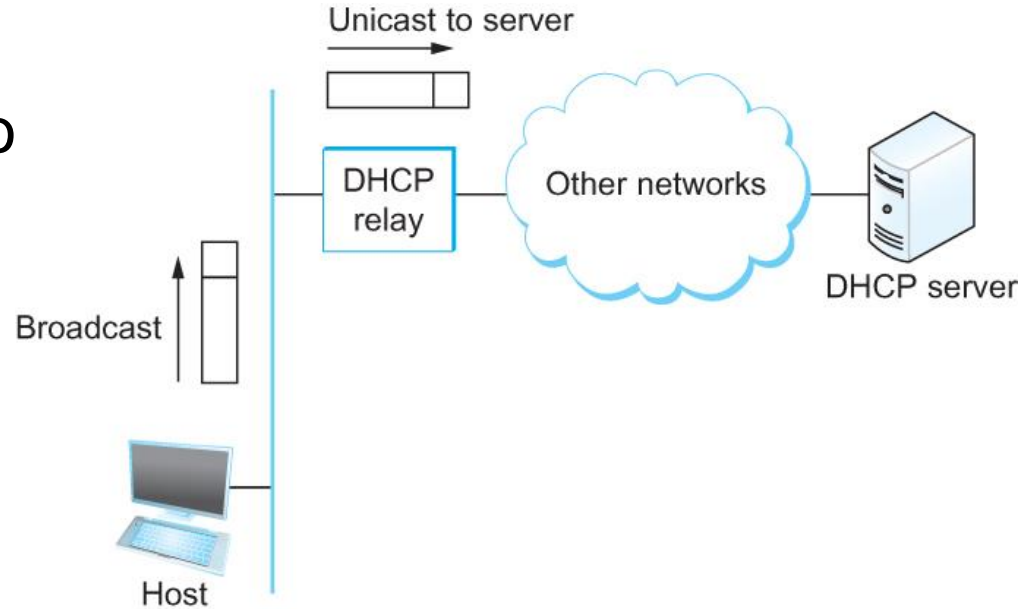
- *Host does not need to know the address of the DHCP server*
- Newly booted or attached host sends a DHCPDISCOVER message to an IP broadcast address, 255.255.255.255
- This information needs to reach a DHCP server

Need for DHCP Relays

- How many DHCP servers do we need?
 - Routers do not forward broadcast IP addresses
 - One per subnetwork! Too many
- Solution: *relay agents*
 - Configured with the DHCP server's IP address
 - One relay agent per subnetwork

Relay-to-Server Operations

- DHCP relay agent unicasts the message to DHCP server and waits for the response



IP Address Allocation in DHCP

- Dynamic
 - IP addresses are *leased* to hosts
 - Addresses returned to the common pool and reused when no longer needed by the host
- Automatic
 - Preferentially assign to hosts same IP addresses they previously had
- Manual (static) allocation
 - Mapping pre-defined by the administrator

DHCP: Associated Complexity

- Dynamic host IP addresses
 - MAC to IP mappings are not guaranteed to be constant
 - Users: cannot count on device IP staying the same
 - Administrators: network troubleshooting complexity
- In practice, ISPs and companies try to keep MAC-IP mappings constant if possible, for manageability

DHCP: Key Points

- A ubiquitously deployed helper protocol that:
 - Provides IP information to newly booted or connected hosts
 - Allows assigning IP addresses *dynamically*
- Hosts not needing to know DHCP server address is achieved via:
 - IP broadcasting
 - Use of relay agents
- Dynamic address allocation can be challenging for network administrators

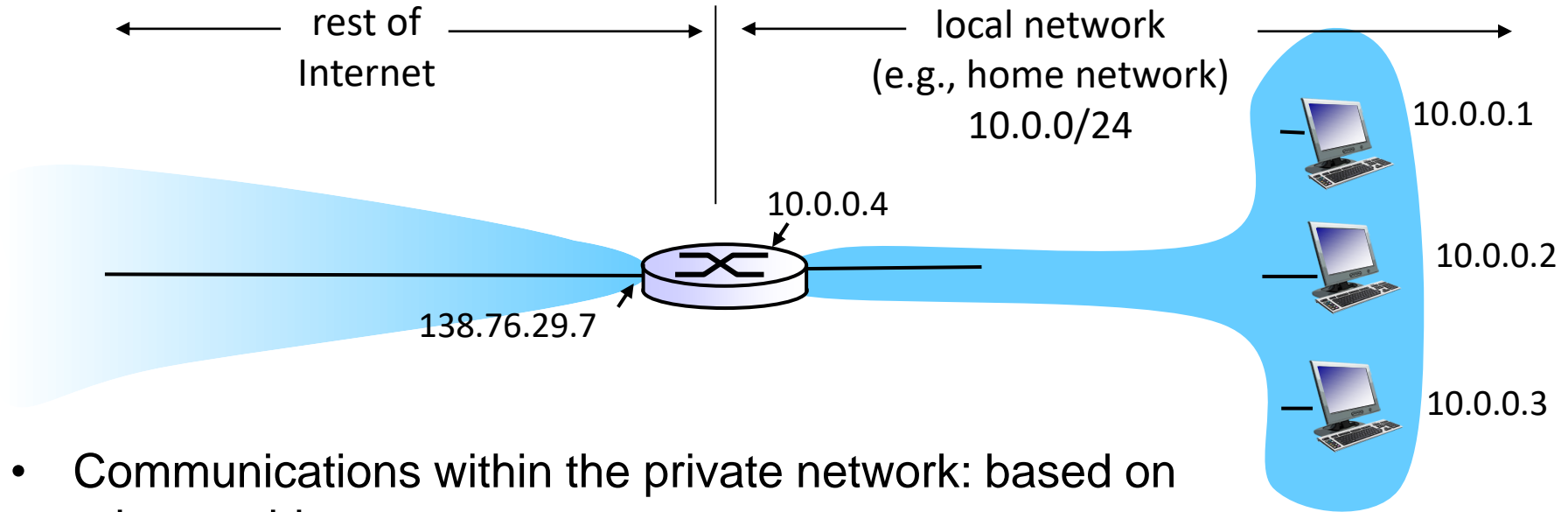
This Lecture

- Routing wrap-up and review
- Finishing up a collection of disjoint but important IP-related topics
 - Dynamic Host Configuration Protocol (DHCP)
 - **Network Address Translation (NAT)**
 - IPv6
 - IP tunnels

Network Address Translation

- We are running out of IP addresses (only 2^{32} addresses in total). How can we work around it, without creating a new variant of IP?
 - For address depletion, NAT is a work-around, not a fix
 - Widely deployed
 - Use every day on Duke network
- NAT is a router function where IP addresses (and possibly port numbers) of IP datagrams are replaced at the boundary of a private network

Network Address Translation: An Example



- Communications within the private network: based on private addresses
- Outside communications: based on a public address

See For Yourself: Your IP Address as Seen from the Outside

- Home network example: **ipchicken.com**



See For Yourself: Your IP Address as Seen by Your Computer

- Linux/MacOS:
ifconfig
- Windows:
ipconfig

```
C:\Users\Marina>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix . . : lan
    IPv6 Address. . . . .                : fd81:2822:e5b5::fa7
    IPv6 Address. . . . .                : fd81:2822:e5b5:0:30cd:290d:225d:4322
    Temporary IPv6 Address. . . . .     : fd81:2822:e5b5:0:24f8:1027:7d54:b779
    Temporary IPv6 Address. . . . .     : fd81:2822:e5b5:0:31b8:b542:5b05:f93a
    Temporary IPv6 Address. . . . .     : fd81:2822:e5b5:0:6cee:cf65:d68e:91ab
    Temporary IPv6 Address. . . . .     : fd81:2822:e5b5:0:a941:e817:7360:9ab8
    Link-local IPv6 Address . . . . .    : fe80::30cd:290d:225d:4322%7
    IPv4 Address. . . . .                : 192.168.0.227
    Subnet Mask . . . . .                : 255.255.255.0
    Default Gateway . . . . .            : 192.168.0.1
```



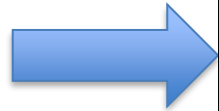
See For Yourself: Your IP Address as Seen from the Outside

- Duke network example: ipchicken.com



Duke Network Example: Your IP Address as Seen by Your Computer

- Linux/MacOS: ifconfig
- Windows: ipconfig



```
Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : wireless.duke.edu
Link-local IPv6 Address . . . . . : fe80::11e:c957:7de1:ea8e%3
IPv4 Address. . . . . : 10.197.4.49
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . : 10.197.0.1
```

Private Network (1/2)

- *Private IP* network is an IP network that is not directly connected to the Internet
- Public IP address are assigned via Internet registries
- IP addresses in a private network can be assigned arbitrarily.
 - Not registered and not guaranteed to be globally unique

Private Network (2/2)

- Generally, private networks use addresses from the following experimental address ranges (*non-routable addresses*):

- **10.0.0.0 – 10.255.255.255**
- 172.16.0.0 – 172.31.255.255
- **192.168.0.0 – 192.168.255.255**

```
Link-local IPv6 Address . . . . . : fe80::30cd:290d:2f
IPv4 Address. . . . . : 192.168.0.227
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.0.1
```

- Private addresses only have meaning within a private network

```
Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : wireless.duke.edu
Link-local IPv6 Address . . . . . : fe80::11e:e957:7de1:ea8e%3
IPv4 Address. . . . . : 10.197.4.49
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . : 10.197.0.1
```

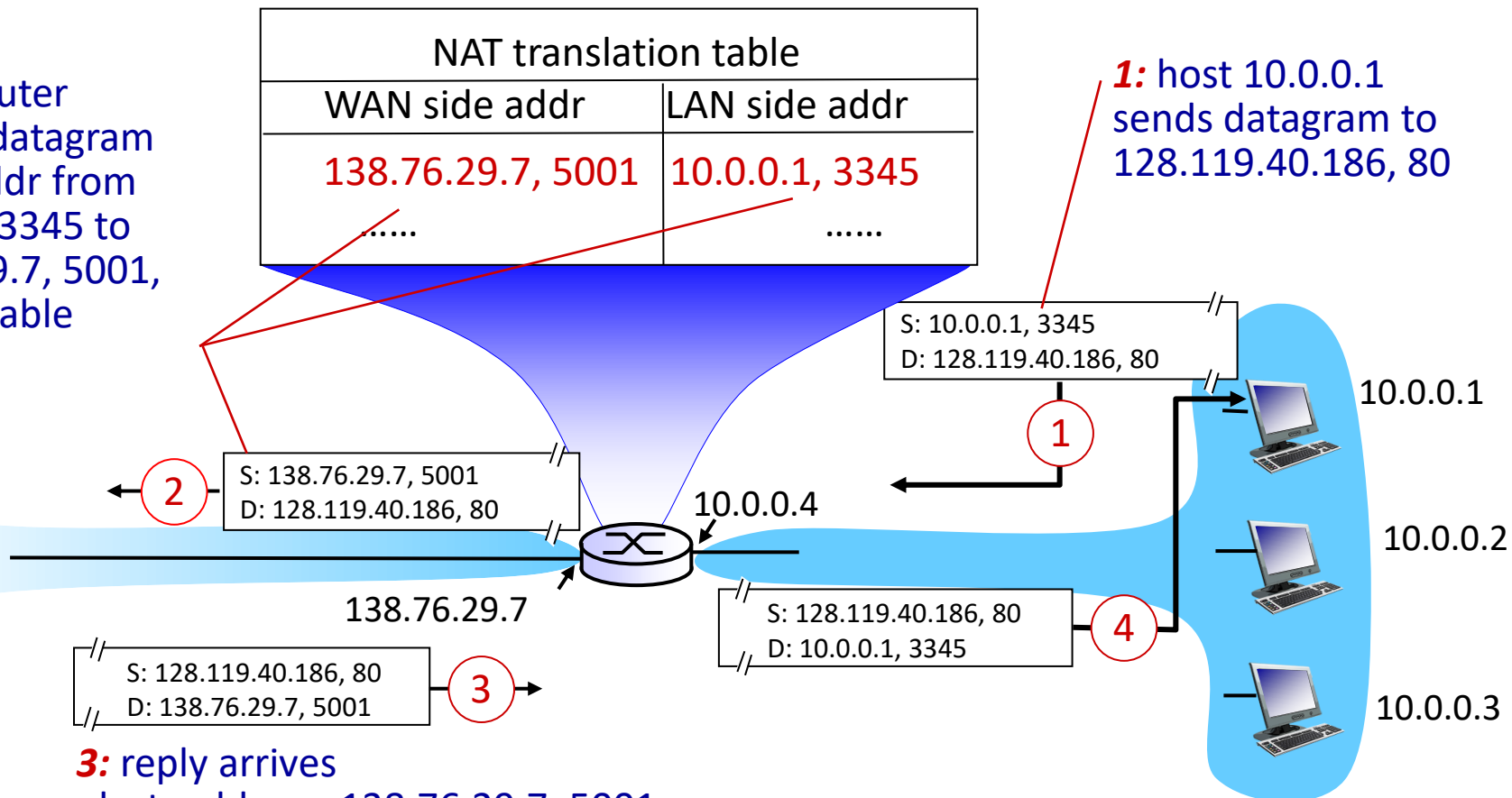
How It Works

- Clever use of port manipulations for mapping public and private addresses
 - Not how ports were intended to be used
- NAT-enabled routers create **NAT translation tables**

How It Works: NAT Translation Table

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80



3: reply arrives
dest. address: 138.76.29.7, 5001

Use of NAT: Pooling of IP Addresses

- Scenario: Corporate network has many hosts but only a small number of public IP addresses
- NAT solution:
 - Corporate network is managed with a private address space
 - NAT device manages a pool of public IP addresses

Use of NAT:

Load Balancing of Servers (1/2)

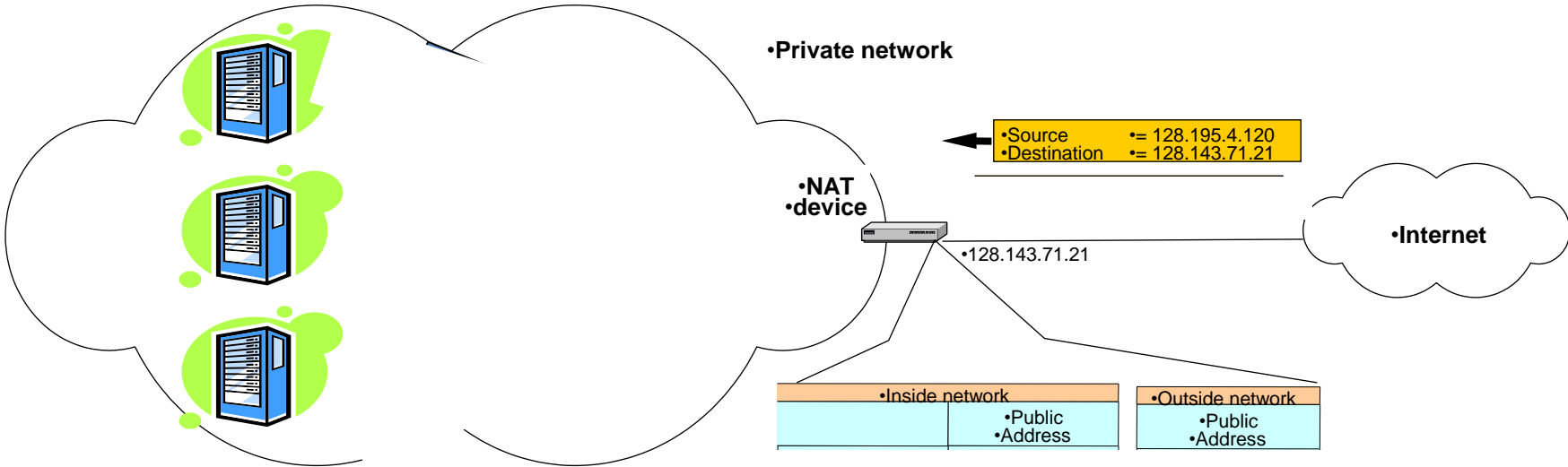
- **Scenario:** Balance the load on a set of identical servers, which are accessible from a single IP address
 - Used by distributed service providers such as Google
 - Commonly deployed

Use of NAT:

Load Balancing of Servers (2/2)

- The servers are assigned private addresses
- NAT device acts as a proxy for requests to the server from the public network
- The NAT device changes the destination IP address of arriving packets to one of the private server addresses
- Balancing the load of the servers:
 - E.g., assign the addresses of the servers in a round-robin fashion

Load Balancing of Servers



Use of NAT: Supporting Migration Between Network Service Providers

- **Scenario:** a corporate network changes its ISP
 - Change all IP addresses in the network?
- **NAT solution:**
 - Assign private addresses to the hosts of the corporate network
 - NAT device has address translation entries which bind the private address of a host to the public address
 - Migration to a new network service provider merely requires an update of the NAT device. The migration is not noticeable to the hosts on the network

Concerns about NAT: Performance

- Modifying the IP header by changing the IP address requires that NAT boxes recalculate the IP header checksum
- Modifying port number requires that NAT boxes recalculate TCP checksum
- *Theme in dealing with IP routers: want them to do as little work as possible*

Concerns about NAT

- Port numbers were not meant for addressing hosts
- Architectural concern: hosts should be talking directly to each other, without intermediaries modifying IP addresses and port numbers
 - NAT is an example of a *middlebox*
 - A host in the public Internet often cannot initiate communication to a host in a private network
 - The problem is worse, when two hosts that are in a private network need to communicate with each other.
 - Difficult to deploy peer-to-peer applications such as Skype

NAT: Key Points

- Ubiquitously deployed method for remapping IP addresses from one space to another
 - E.g., Duke network example
- How it works:
 - NAT translation tables at the NAT routers
 - Clever use of ports for address mapping
- Used for: using a small number of public IPs, server load balancing, corporate network migration
- Concerns: TO FILL IN

Lecture Outline

- Routing wrap-up and review
- Finishing up a collection of disjoint but important IP-related topics
 - Dynamic Host Configuration Protocol (DHCP)
 - Network Address Translation (NAT)
 - **IPv6**
 - IP tunnels

Next-Generation IP: IPv6

- Core difference: larger address space
 - IPv4 → IPv6: 32-bit address → 128-bit address
- All OSs support it
- Duke networks support it
 - To see for yourself:
 - Windows: **ipconfig**
 - Linux, MacOS: **ifconfig**

Professor's Example

```
C:\Users\Maria>ipconfig

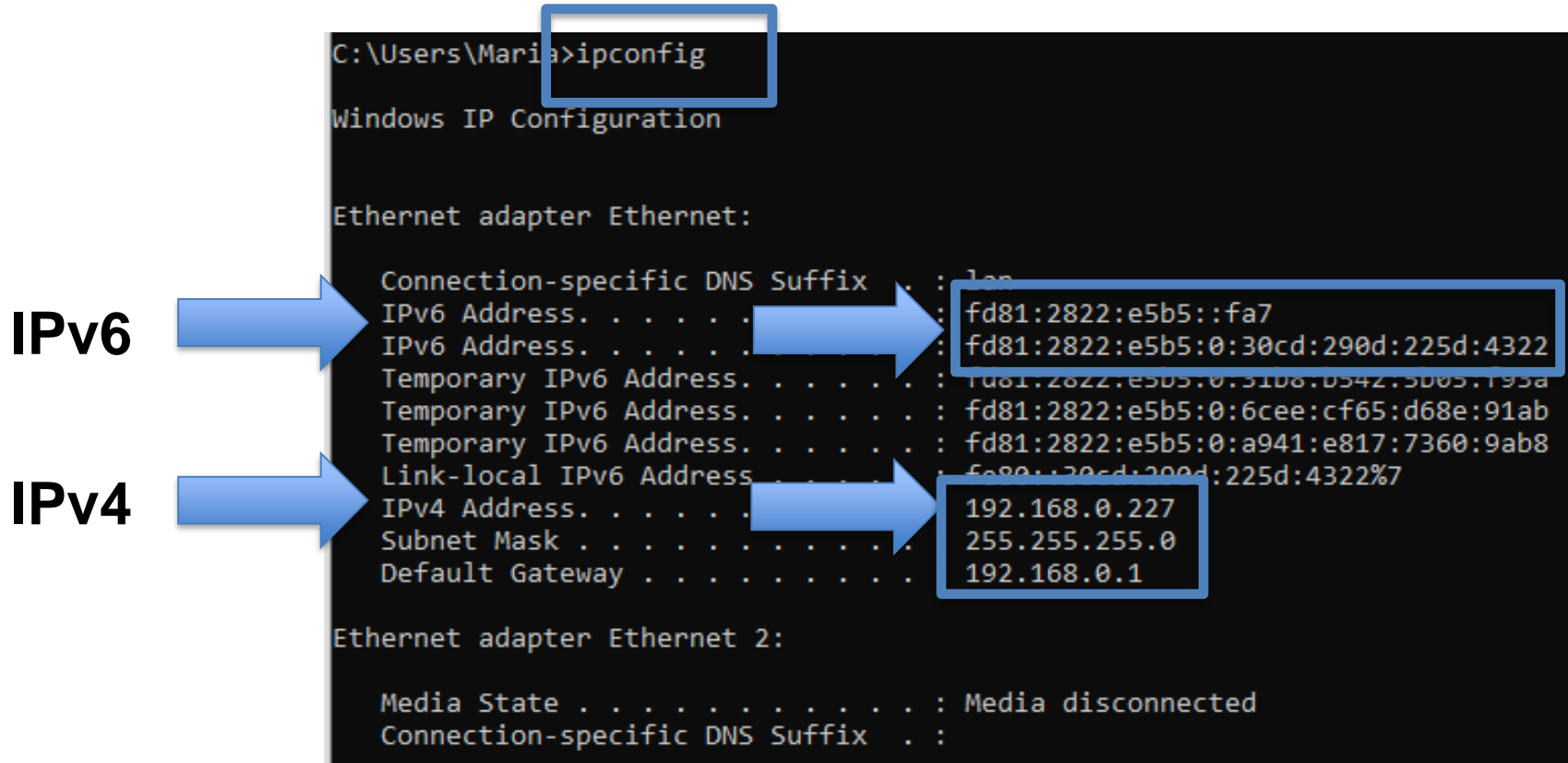
Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix . : lan
    IPv6 Address. . . . . : fd81:2822:e5b5::fa7
    IPv6 Address. . . . . : fd81:2822:e5b5:0:30cd:290d:225d:4322
    Temporary IPv6 Address. . . . . : fd81:2822:e5b5:0:5108:b342:5005:193a
    Temporary IPv6 Address. . . . . : fd81:2822:e5b5:0:6cee:cf65:d68e:91ab
    Temporary IPv6 Address. . . . . : fd81:2822:e5b5:0:a941:e817:7360:9ab8
    Link-local IPv6 Address . . . . . : fe80::30cd:290d:225d:4322%7
    IPv4 Address. . . . . : 192.168.0.227
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

Ethernet adapter Ethernet 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :
```



We Are Running Out of IP Addresses. Now What?



- There are more humans than IPv4 addresses
 - Humans on the planet: ~ 7 bln
 - $2^{32} = 4,294,967,296$ (~ 4.3 bln) addresses with IPv4
- Many machines need IP addresses, too
 - Internet of Things
- We increase the address space
 - IPv4 → IPv6: 32-bit address → 128-bit address
 - 340,282,366,920,938,463,463,374, 607,431,768,211,456 addresses with IPv6. *$5 \cdot 10^{28}$ addresses per every human*

“If the Earth were made entirely out of 1 cubic millimetre grains of sand, then you could give a unique [IPv6] address to each grain in 300 million planets the size of the Earth.”

History

- In early 90s, IPv4 was running out of addresses
- Changing to a larger address space requires many changes
 - Core IP header changes → *a whole new version of the IP protocol*
- IETF solicited other desired features
- Chose one for IPv6 (**RFC 2460**) in 1998

IPv6: New Features

- Large address space (128-bit) 
- Hierarchical addressing and routing, autoconfiguration
- Built-in security, better support for QoS
- New protocols for neighboring node interactions
- Extensibility
- Simplified header format 

IPv6 Addresses

- 128 bits long
- Classless: similar to CIDR
- Inherently hierarchical
 - Parts represent the interface, parts represent the network

Recap: IPv4 Addresses

- Dotted decimal notation
- Each byte is identified by a decimal number in the range [0...255]:

10000000	10001111	10001001	10010000
----------	----------	----------	----------

1st Byte

2nd Byte

3rd Byte

4th Byte

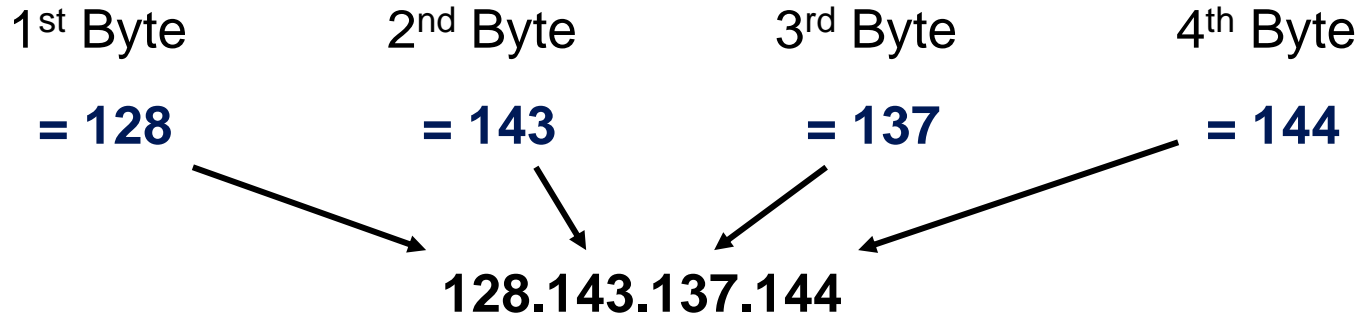
= 128

= 143

= 137

= 144

128.143.137.144



IPv6 Addresses: Semicolon-separated Hexadecimal Notation

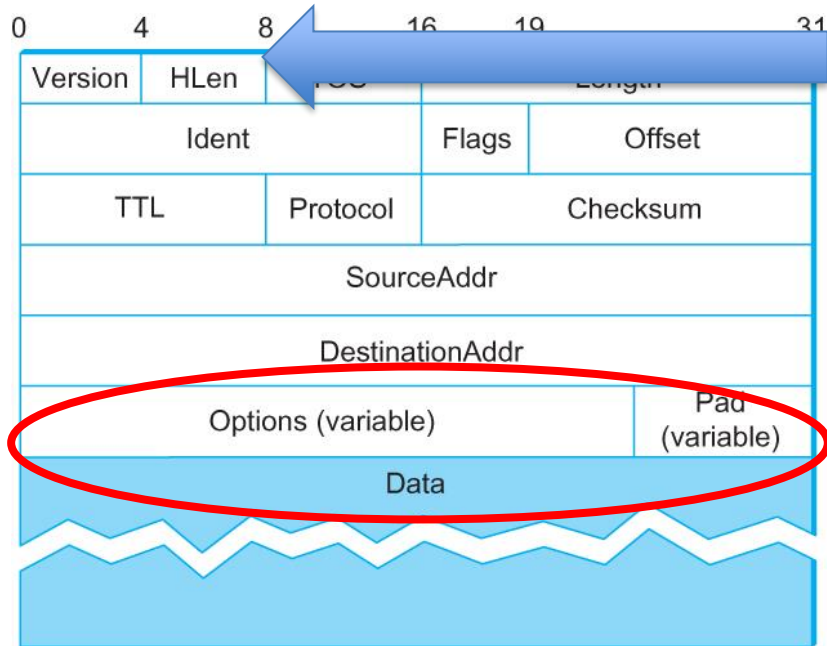
- **x:x:x:x:x:x:x:x**, where each x is a 16-bit hex number
 - E.g., **2001:0db8:85a3:0000:0000:8a2e:0370:7334**
- Compared to IPv4:
 - Twice more bits per separated item
 - Twice more items

IPv6 Addresses: Making Them Easier to Read

- Contiguous 0s are compressed
 - 2001:0db8:85a3:0000:0000:8a2e:0370:7334 becomes **2001:0db8:85a3::8a2e:0370:7334**
 - Can only be used for one set of consecutive 0s
- First or last zeros can be omitted
 - 2001:0DB8:AC10:FE01:0000:0000:0000:0000 becomes **2001:0DB8:AC10:FE01::**
 - Extreme example: loopback address
 - 0000:0000:0000:0000:0000:0000:0000:0001 becomes **::1**

IPv6 Headers: Longer but Simpler (1/3)

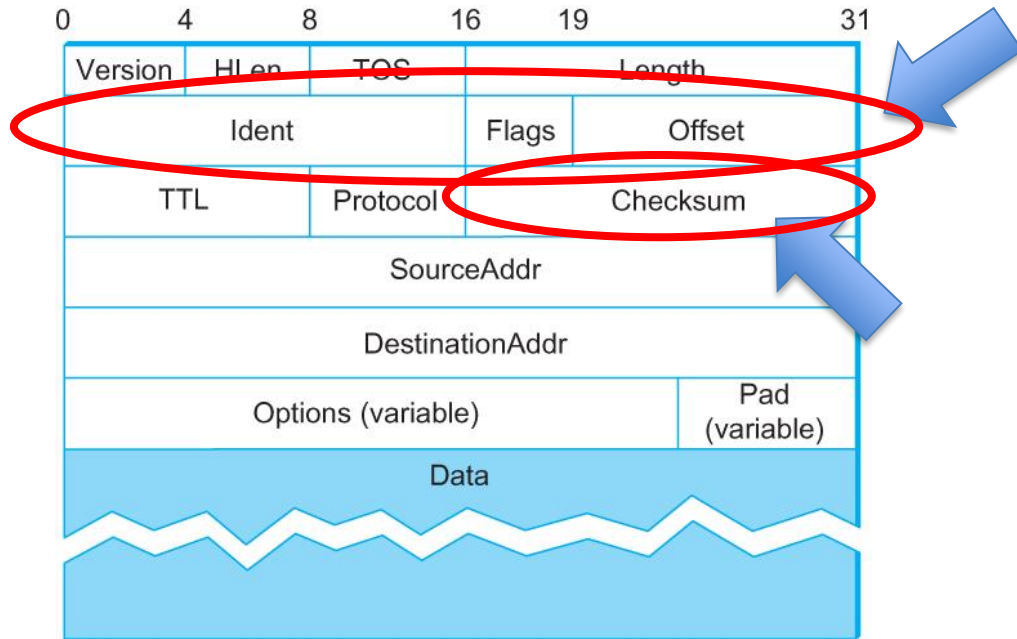
- Recall: IPv4 header



- Variable length
 - 20 bytes if there are no options
- Options: unordered collection of $\langle \text{type}, \text{length}, \text{value} \rangle$ tuples
 - In IPv6, *header extensions* in a pre-specified order
 - Routers can quickly determine relevant ones

IPv6 Headers: Longer but Simpler (3/3)

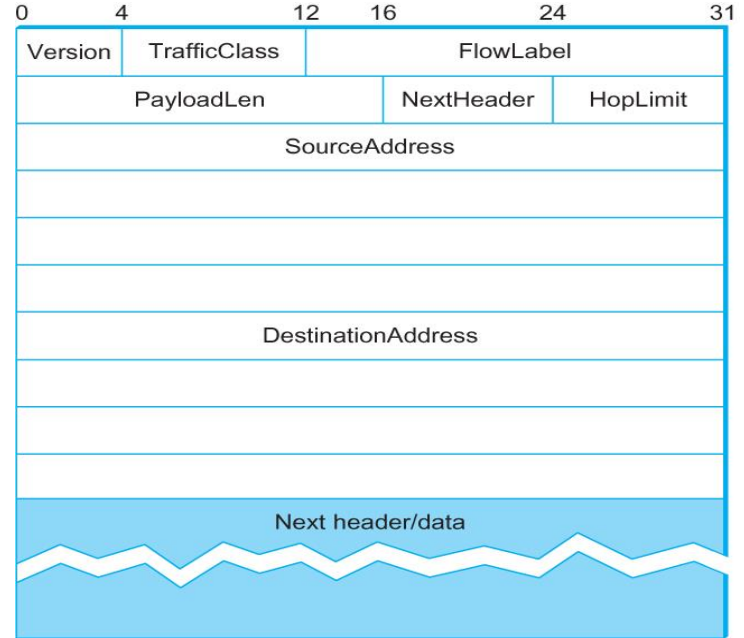
- Recall: IPv4 header



- Fragmentation
 - Not in IPv6
 - IPv6 forces MTU path discovery
- Checksum
 - Not in IPv6
 - Prioritizing faster processing by the routers

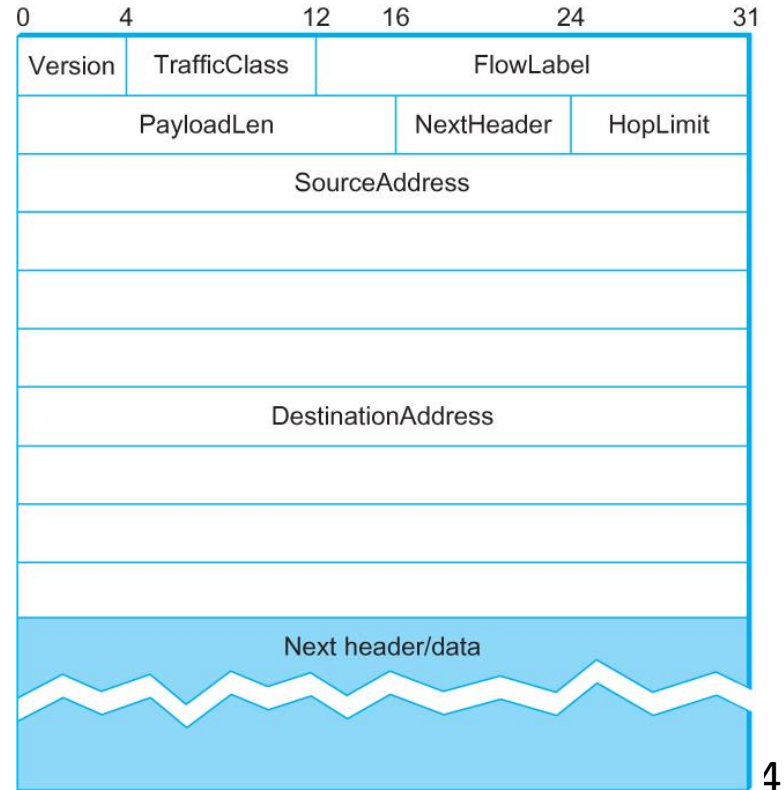
IPv6 Header: Longer but Simpler (3/3)

- IPv6: 40-byte “base” header
 - Longer: compared to 20 bytes
 - Largely taken up by the long addresses
- Extension headers (fixed order, mostly fixed length)



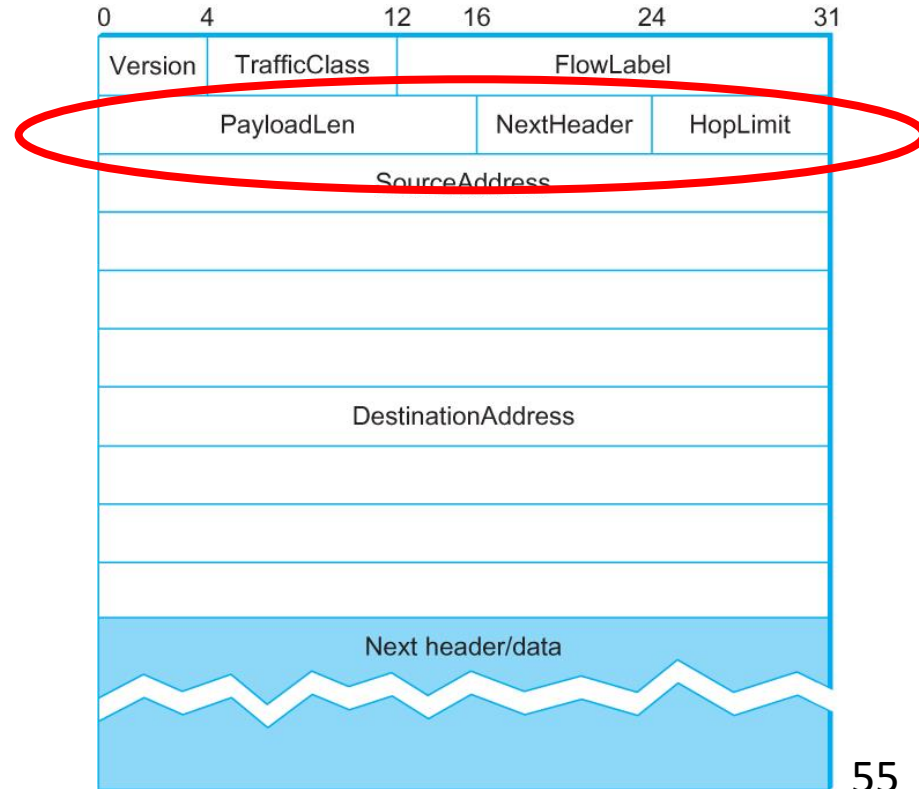
IPv6 Header: Fields (1/2)

- Version: 6
 - Putting “4” in this field does not make it an IPv4 packet
- Traffic class: similar to TOS field: can be used to give priority to certain datagrams
- Flow label: if want to treat e.g., audio, video as complete flows



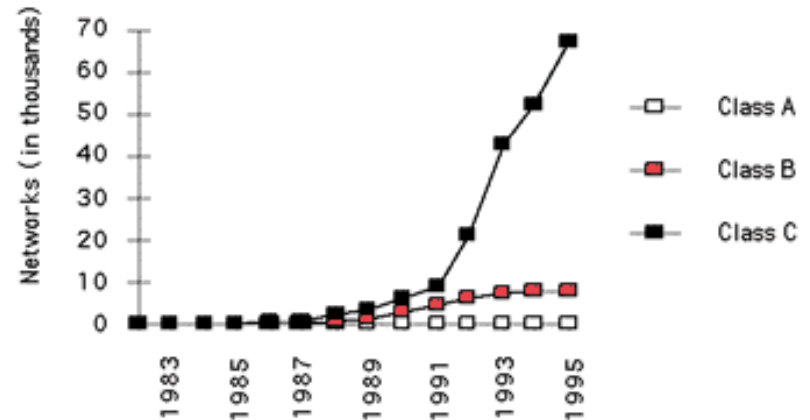
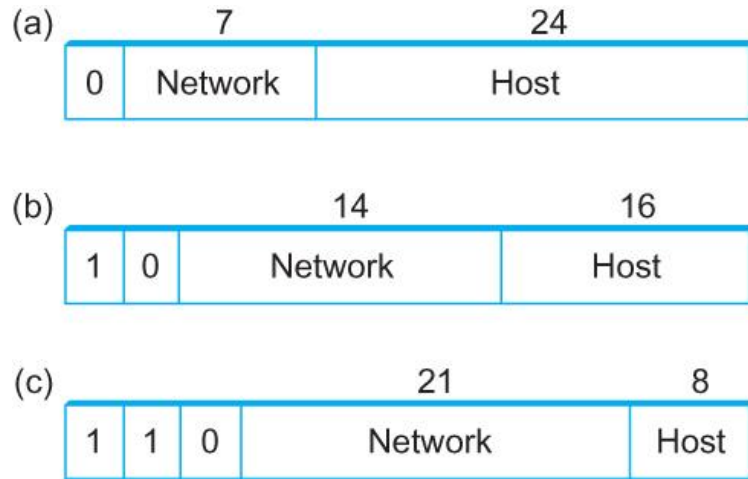
IPv6 Header: Fields (2/2)

- Similar to IPv4:
 - Payload length: size of the payload. Same as “length” in IPv4
 - Next header: e.g., UDP, TCP. Same as “protocol” in IPv4 header
 - Hop limit: decremented on forwarding. Same as TTL



IPv6 Adoption: Reasons for a Delay (1/2)

- Started to be discussed in early 1990s
 - When IP addresses were *classful*
 - 1993: Introduction of CIDR



IPv6 Adoption: Reasons for a Delay (2/2)

- With CIDR, IP address exhaustion became a less urgent issue
- Proliferation of NAT also helped delay address exhaustion
- IPv6 has been somewhat slow in gaining adoption, but it has picked up in late 2000s

IPv6 Usage in Most Popular Websites: Alexa Internet

- *Alexa Internet* list of most popular websites
 - www.alexa.com/topsites
 - Subsidiary of Amazon
 - Not related to Amazon Alexa
- Finds most popular websites based on a combination of page views and unique site visits
- As of June 2019
 - 26% of Alexa Top 1000 web servers support IPv6
 - 29% of users reach Google services with IPv6

IPv6 Usage by Major Companies

- Verizon Wireless an important pioneer
- As of 2018, 80% of the traffic from Verizon Wireless used IPv6
- Since 2018, Facebook has been eliminating IPv4 in datacenters

IPv6: Key Points

- IPv6 offers a much larger address space than IPv4
- Sufficiently different from IPv4
 - Has new features
 - Rethinks some of the design decisions
- Has been slower to be adopted than originally expected, but is in sufficiently wide use already

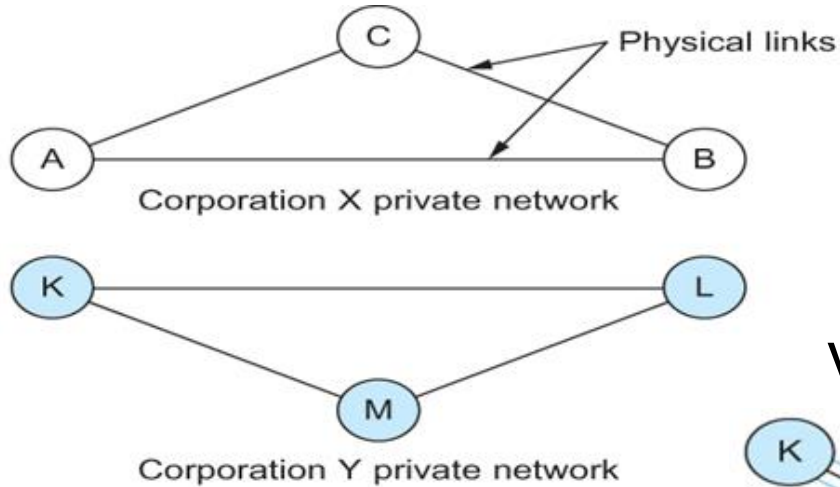
Lecture Outline

- Routing wrap-up and review
- Finishing up a collection of disjoint but important IP-related topics
 - Dynamic Host Configuration Protocol (DHCP)
 - Network Address Translation (NAT)
 - IPv6
 - **IP tunnels**

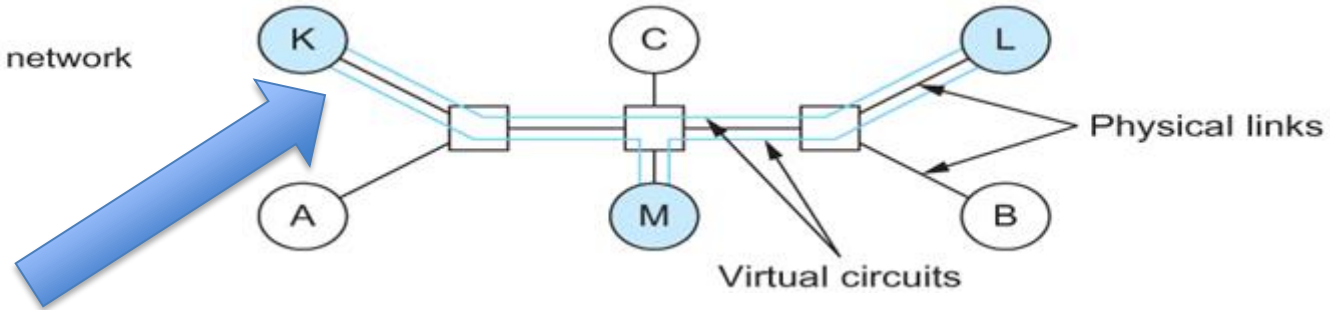
IP Tunnels

- How you “VPN” into Duke network
- A technique used in many scenarios
 - Virtual private networks (VPNs), IPv4-v6 transition, Mobile IP, Multicast, Non-IP forwarding, IPsec

Virtual Private Networks

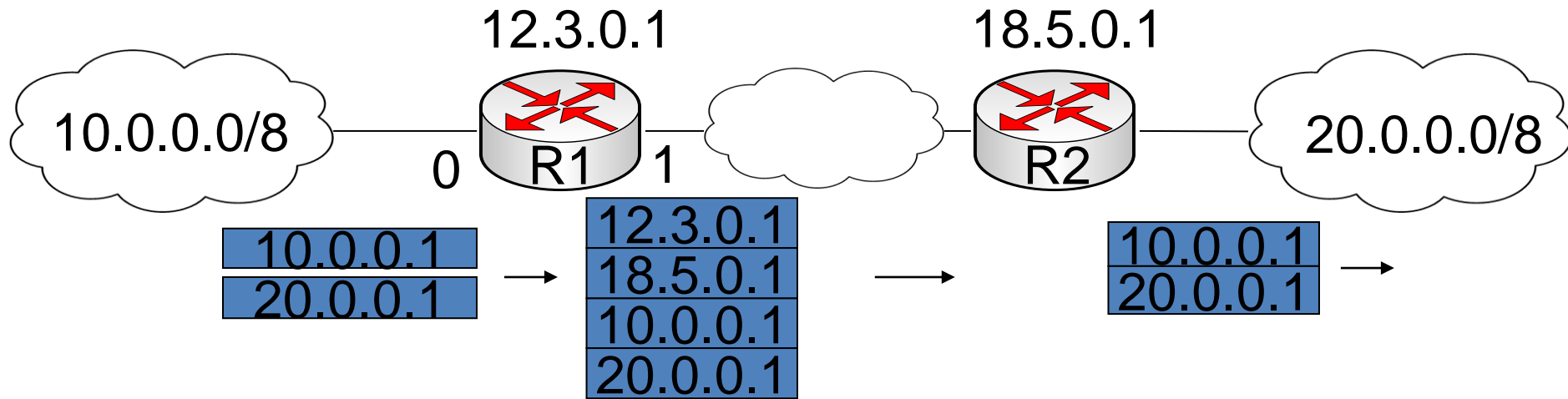


VPNs sharing common switches



- Need IP tunnels

What is a Tunnel?



- A “pseudowire”, or a virtual point-to-point link
 - Arbitrary number of networks in the middle
- The head router encapsulates a packet in an outer header destined to the tail router

Virtual Interface

- A router adds a tunnel header for packets sent to a virtual interface

- In this example:

- First add a header addressed to R2
- Then forward according to established forwarding rules
 - Here, send to the Default interface ether1



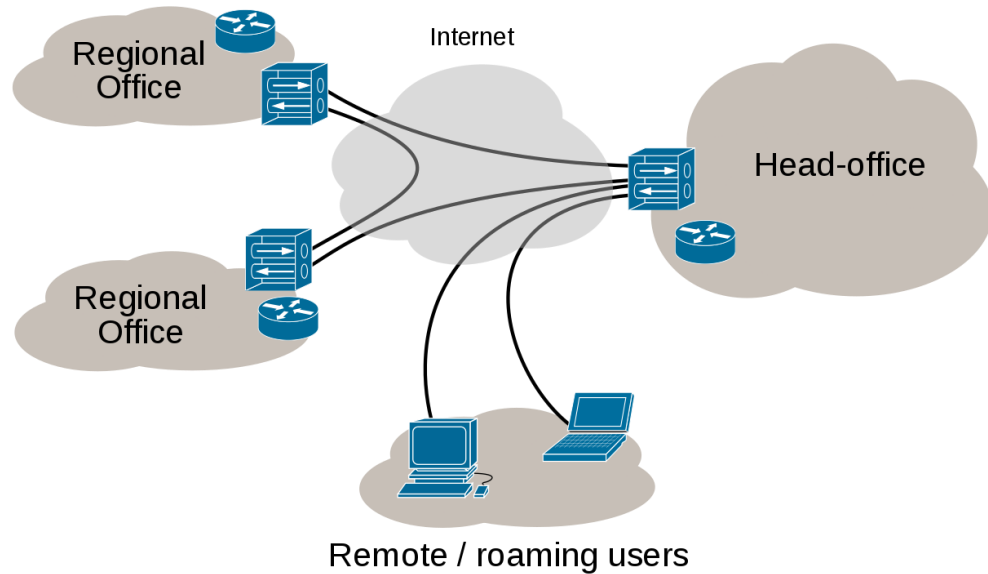
NetworkNum	nextHop
10.0.0.0/8	ether0
20.0.0.0/8	tun0
Default	ether1

- All other routers forward the packet like a regular packet destined for R2

Tunnel Applications

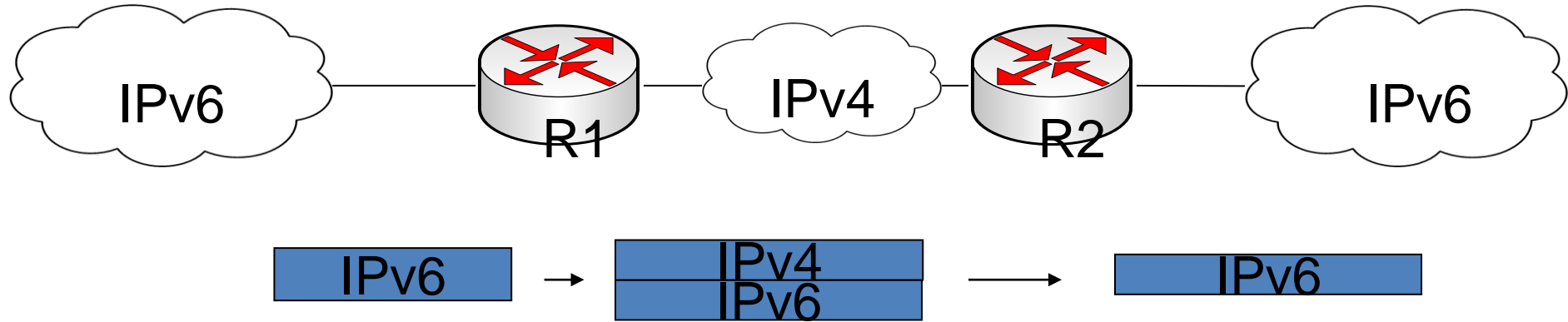
Internet VPN

- Security
 - E.g., VPNs



- Carrying non-IP traffic across IP networks

IP in IP Tunneling: IPv4-v6 Transition



IP Tunnel Performance Issues

- Tunnels need to be set up
- Increases the length of the packets
 - Extra headers
 - Particularly inefficient for short packets
- Tunnel entry and exit routers need to do more than simply forward packets
 - Potential for slow-down

IP Tunnels: Key Points

- If you've used Duke VPN, you've used IP tunnels
- Create virtual point-to-point connections across global networks
- To achieve it, we use:
 - Packet encapsulation
 - Tunnel-specific virtual interface at an entry router
- Pros: security
- Performance issues: higher overhead, slow-down on some routers

Lecture Summary

- Rounded off our discussion of IP
- Ubiquitously deployed “helpers”:
 - Dynamic Host Configuration Protocol (DHCP)
 - Network Address Translation (NAT)
- Next-generation version of IP
 - Sufficiently different from IPv4
 - Deployed, but has not yet overtaken IPv4
- Commonly used way of creating private networks on top of public ones

Next Lecture

- Transport control
 - Moving up the stack

