

**UCLA CS 118 Winter 2021**

Instructor: Giovanni Pau  
TAs:  
Hunter Dellaverson  
Eric Newberry

This chapter slide deck draws from different sources 7<sup>th</sup> and 8<sup>th</sup> edition of the textbook.

Transport Layer: 3-1

1

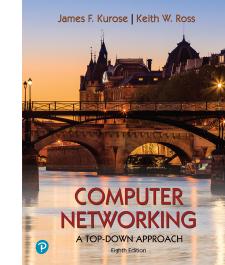
## Chapter 4 Network Layer: Data Plane

A note on the use of these PowerPoint slides:  
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

For a revision history, see the slide note for this page.

Thanks and enjoy! J.F.K./K.W.R.  
All material © 2019, 2020  
J.F. Kurose and K.W. Ross, All Rights Reserved



**Computer Networking: A Top-Down Approach**  
8<sup>th</sup> edition  
Jim Kurose, Keith Ross  
Pearson, 2020

2

<https://powcoder.com>

### Network layer: our goals

- understand principles behind network layer services, focusing on data plane:
  - network layer service models
  - forwarding versus routing
  - how a router works
  - addressing
  - generalized forwarding
  - Internet architecture
- instantiation, implementation in the Internet
  - IP protocol
  - NAT, middleboxes

Network Layer: 4-3

3

### Add WeChat powcoder

### Network layer: "data plane" roadmap

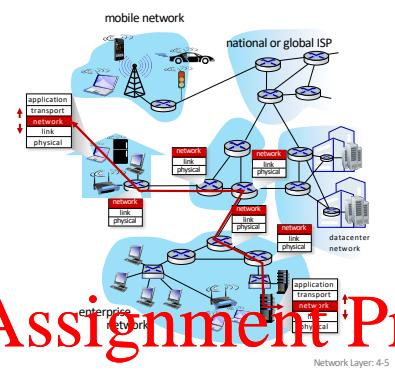
- Network layer: overview**
  - data plane
  - control plane
- What's inside a router**
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol**
  - datagram format
  - addressing
  - network address translation
  - IPv6
- Generalized Forwarding, SDN**
  - Match+action
  - OpenFlow: match+action in action
- Middleboxes**

Network Layer: 4-4

4

## Network-layer services and protocols

- 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
- routers:**
  - examines header fields in all IP datagrams passing through it
  - moves datagrams from input ports to output ports to transfer datagrams along end-end path



5

# Assignment Project Exam Help

## Two key network-layer functions

### network-layer functions:

- forwarding:** move packets from a router's input link to appropriate router output link
- routing:** determine route taken by packets from source to destination
  - routing algorithms*

### analogy: taking a trip

- forwarding:** process of getting through single interchange
- routing:** process of planning trip from source to destination



Network Layer: 4-6

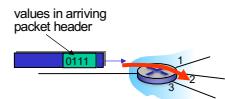
6

# <https://powcoder.com>

## Network layer: data plane, control plane

### Data plane:

- local**, per-router function
- determines how datagram arriving on router input port is forwarded to router output port



### Control plane

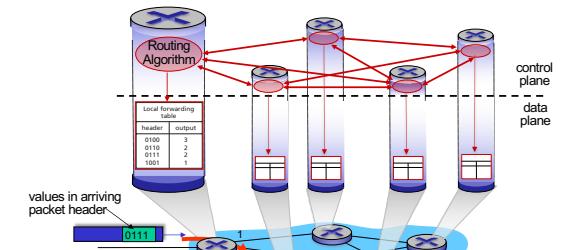
- network-wide** logic
- determines how datagram is routed among routers along end-end path from source host to destination host
- two control-plane approaches:
  - traditional routing algorithms*: implemented in routers
  - software-defined networking (SDN)*: implemented in (remote) servers

Network Layer: 4-7

7

## Add WeChat powcoder Per-router control plane

Individual routing algorithm components *in each and every router* interact in the control plane

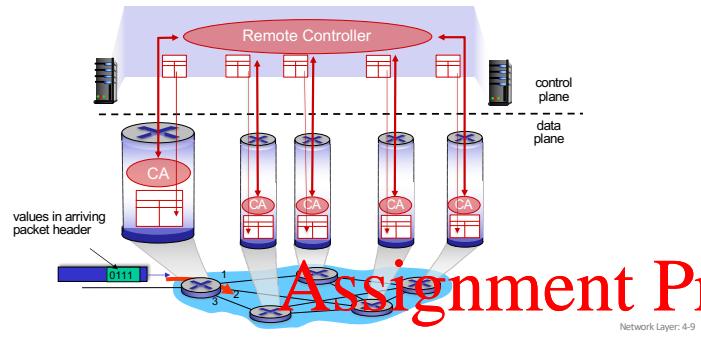


Network Layer: 4-8

8

## Software-Defined Networking (SDN) control plane

Remote controller computes, installs forwarding tables in routers



9

## Network service model

*Q:* What *service model* for “channel” transporting datagrams from sender to receiver?

example services for *individual* datagrams:

- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

example services for a *flow* of datagrams:

- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing

Network Layer: 4-10

10

<https://powcoder.com>

## Network-layer service model

Add WeChat powcoder

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

Network Layer: 4-11

11

## Network-layer service model

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

Network Layer: 4-12

12

**Reflections on best-effort service:**

- 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” for “most of the time”
- replicated, application-layer distributed services (datacenters, content distribution networks) connecting close to clients’ networks, allow services to be provided from multiple locations
- congestion control of “elastic” services helps

*It's hard to argue with success* of best-effort service!

Network Layer: 4-13

13

**Network layer: “data plane” roadmap**

- Network layer: overview
  - data plane
  - control plane
- What’s inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6
- Generalized Forwarding, SDN
  - Match+action
  - OpenFlow: match+action in action
- Middleboxes



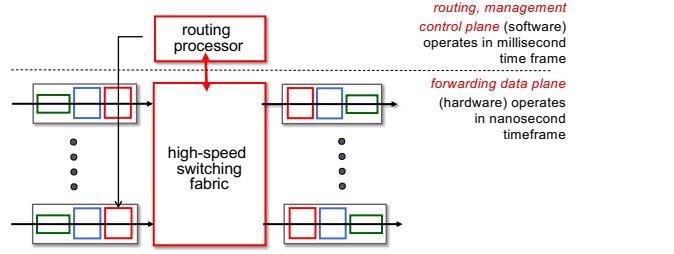
Network Layer: 4-14

14

<https://powcoder.com>

**Router architecture overview**

high-level view of generic router architecture:

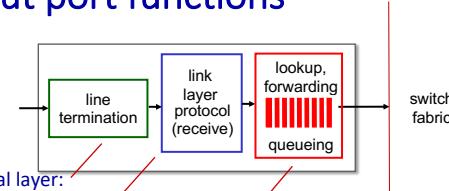


Network Layer: 4-15

15

**Add WeChat powcoder**

**Input port functions**



**physical layer:** bit-level reception

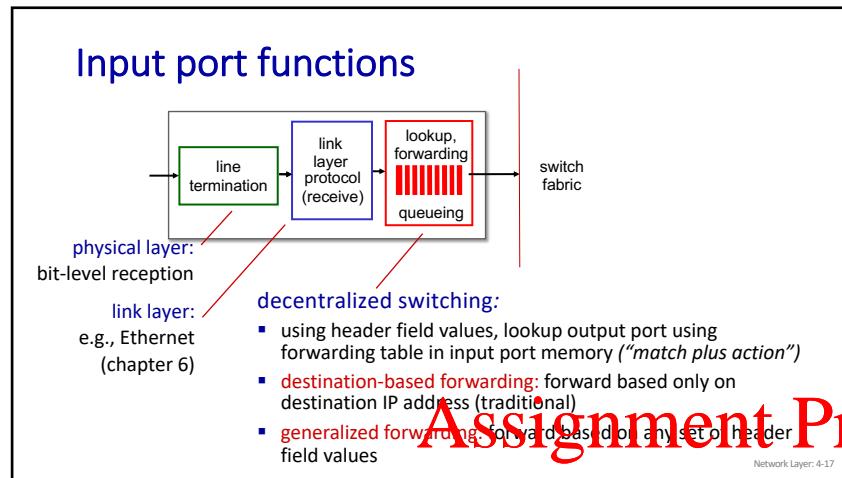
**link layer:** e.g., Ethernet (chapter 6)

**decentralized switching:**

- using header field values, lookup output port using forwarding table in input port memory (“*match plus action*”)
- goal: complete input port processing at ‘line speed’
- **input port queuing:** if datagrams arrive faster than forwarding rate into switch fabric

Network Layer: 4-16

16



17

**Destination-based forwarding**

*forwarding table*

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through	0
11001000 00010111 00010000 00000100 through	3
11001000 00010111 00010000 00000111	
11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000 through	2
11001000 00010111 00011111 11111111	
otherwise	3

Q: but what happens if ranges don't divide up so nicely?

Network Layer: 4-18

18

Assignment Project Exam Help

<https://powcoder.com>

**Longest prefix matching**

longest prefix match ——————

when looking for forwarding table entry for given destination address, 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

examples:

11001000 00010111 00010110 10100001	which interface?
11001000 00010111 00011000 10101010	which interface?

Network Layer: 4-19

19

**Add WeChat powcoder**

**Longest prefix matching**

longest prefix match ——————

when looking for forwarding table entry for given destination address, use **longest** address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010000 *** *****	0
11001000 00010111 00011000 *****	1
11001000 match! 1 00011*** *****	2
otherwise	3

examples:

11001000 00010111 00010110 10100001	which interface?
11001000 00010111 00011000 10101010	which interface?

Network Layer: 4-20

20

## Longest prefix matching

longest prefix match

when looking for forwarding table entry for given destination address, 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

examples:

11001000 00010111 00011000 which interface?  
11001000 00010111 00011\*\*\* which interface?

match!

21

## Longest prefix matching

longest prefix match

when looking for forwarding table entry for given destination address, 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

examples:

11001000 00010111 00011000 which interface?  
11001000 00010111 00011\*\*\* which interface?

Network Layer: 4-22

22

Assignment Project Exam Help

## Longest prefix matching

- we'll see *why* longest prefix matching is used shortly, when we study addressing
- longest prefix matching: often performed using ternary content addressable memories (TCAMs)
  - content addressable**: present address to TCAM: retrieve address in one clock cycle, regardless of table size
  - Cisco Catalyst: ~1M routing table entries in TCAM

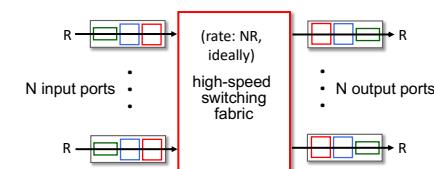
Network Layer: 4-23

23

Add WeChat powcoder

## Switching fabrics

- transfer packet from input link to appropriate output link
- switching rate**: rate at which packets can be transferred from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable

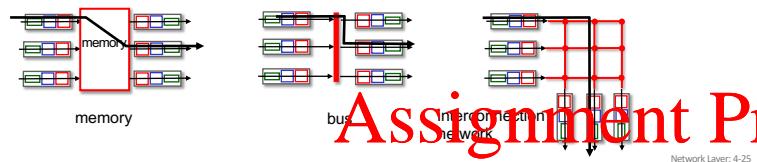


Network Layer: 4-24

24

## Switching fabrics

- transfer packet from input link to appropriate output link
- **switching rate:** rate at which packets can be transferred from inputs to outputs
  - often measured as multiple of input/output line rate
  - $N$  inputs: switching rate  $N$  times line rate desirable
- three major types of switching fabrics:

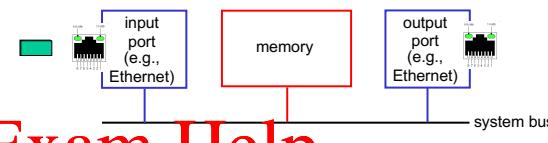


25

## Switching via memory

### first generation routers:

- traditional computers with switching under direct control of CPU
- packet copied to system's memory
- speed limited by memory bandwidth (2 bus crossings per datagram)

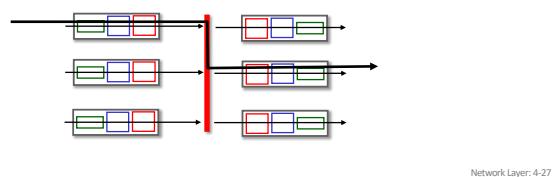


26

Assignment Project Exam Help  
<https://powcoder.com>

## Switching via a bus

- datagram from input port memory to output port memory via a shared bus
- **bus contention:** switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access routers

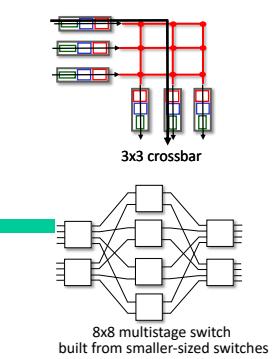


27

Add WeChat powcoder

## Switching via interconnection network

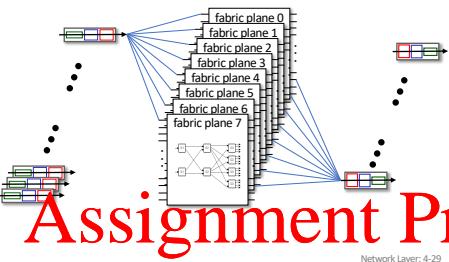
- Crossbar, Clos networks, other interconnection nets initially developed to connect processors in multiprocessor
- **multistage switch:**  $nxn$  switch from multiple stages of smaller switches
- **exploiting parallelism:**
  - fragment datagram into fixed length cells on entry
  - switch cells through the fabric, reassemble datagram at exit



28

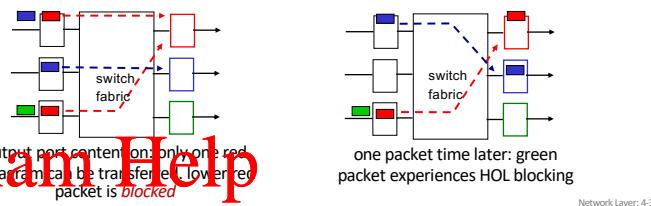
### Switching via interconnection network

- scaling, using multiple switching “planes” in parallel:
  - speedup, scaleup via parallelism
  
- Cisco CRS router:
  - basic unit: 8 switching planes
  - each plane: 3-stage interconnection network
  - up to 100's Tbps switching capacity



### Input port queuing

- If switch fabric slower than input ports combined -> queueing may occur at input queues
  - queueing delay and loss due to input buffer overflow!
- **Head-of-the-Line (HOL) blocking:** queued datagram at front of queue prevents others in queue from moving forward

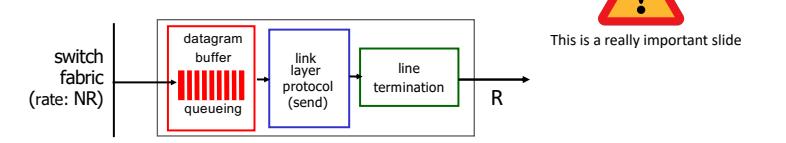


29

30

<https://powcoder.com>

### Output port queuing



- **Buffering** required when datagrams arrive from fabric faster than link transmission rate. **Drop policy:** which datagrams to drop if no free buffers?
- **Scheduling discipline** chooses among queued datagrams for transmission

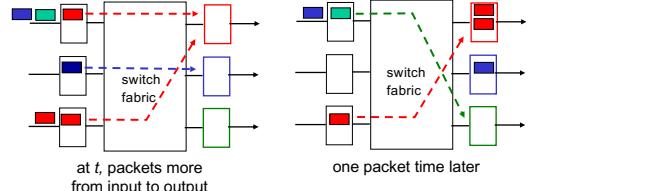
Datagrams can be lost due to congestion, lack of buffers

Priority scheduling – who gets best performance, network neutrality

Network Layer: 4-31

### Add WeChat powcoder

### Output port queuing



- buffering when arrival rate via switch exceeds output line speed
- **queueing (delay) and loss due to output port buffer overflow!**

at t, packets more from input to output

one packet time later

Network Layer: 4-32

31

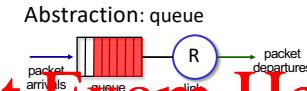
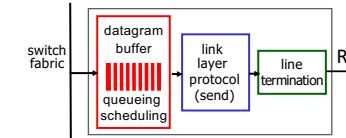
32

## How much buffering?

- RFC 3439 rule of thumb: average buffering equal to “typical” RTT (say 250 msec) times link capacity C
  - e.g., C = 10 Gbps link: 2.5 Gbit buffer
- more recent recommendation: with  $N$  flows, buffering equal to  $\frac{RTT \cdot C}{\sqrt{N}}$
- but too much buffering can increase delays (particularly in home routers)
  - long RTTs: poor performance for realtime apps, sluggish TCP response
  - recall delay-based congestion control: “keep bottleneck link just full enough (busy) but no fuller”

33

## Buffer Management



### buffer management:

- **drop:** which packet to add, drop when buffers are full
  - **tail drop:** drop arriving packet
  - **priority:** drop/remove on priority basis
- **marking:** which packets to mark to signal congestion (ECN, RED)

Network Layer: 4-34

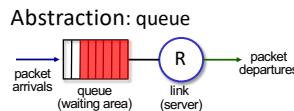
# Assignment Project Exam Help

34

<https://powcoder.com>

## Packet Scheduling: FCFS

- packet scheduling:** deciding which packet to send next on link
- first come, first served
  - priority
  - round robin
  - weighted fair queueing



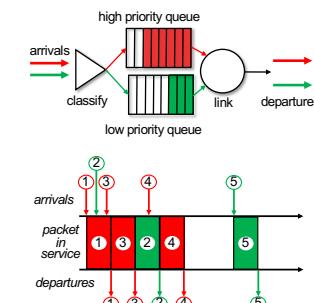
Network Layer: 4-35

## Add WeChat powcoder

## Scheduling policies: priority

### Priority scheduling:

- arriving traffic classified, queued by class
  - any header fields can be used for classification
- send packet from highest priority queue that has buffered packets
  - FCFS within priority class



Network Layer: 4-36

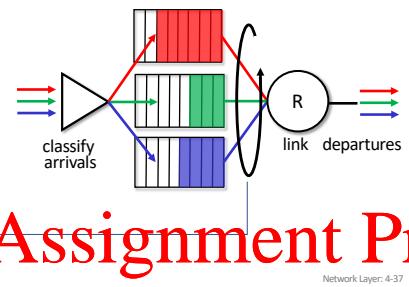
35

36

## Scheduling policies: round robin

### *Round Robin (RR) scheduling:*

- arriving traffic classified, queued by class
  - any header fields can be used for classification
- server cyclically, repeatedly scans class queues, sending one complete packet from each class (if available) in turn

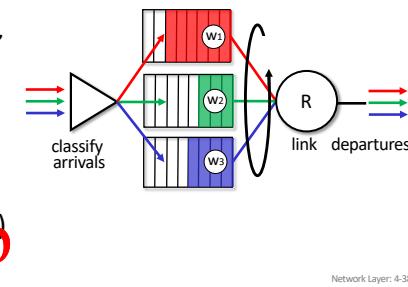


37

## Scheduling policies: weighted fair queueing

### *Weighted Fair Queueing (WFQ):*

- generalized Round Robin
  - each class,  $i$ , has weight,  $w_i$ , and gets weighted amount of service in each cycle:
- $$\frac{w_i}{\sum_j w_j}$$
- minimum bandwidth guarantee (per-traffic-class)



38

Assignment Project Exam Help

## Sidebar: Network Neutrality

### What is network neutrality?

- **technical:** how an ISP should share/allocation its resources
  - packet scheduling, buffer management are the *mechanisms*
- **social, economic** principles
  - protecting free speech
  - encouraging innovation, competition
- enforced **legal** rules and policies

*Different countries have different “takes” on network neutrality*

Network Layer: 4-39

39

Add WeChat powcoder

## Sidebar: Network Neutrality

2015 US FCC Order on Protecting and Promoting an Open Internet: three “clear, bright line” rules:

- **no blocking** ... “shall not block lawful content, applications, services, or non-harmful devices, subject to reasonable network management.”
- **no throttling** ... “shall not impair or degrade lawful Internet traffic on the basis of Internet content, application, or service, or use of a non-harmful device, subject to reasonable network management.”
- **no paid prioritization.** ... “shall not engage in paid prioritization”

Network Layer: 4-40

40

## ISP: telecommunications or information service?

Is an ISP a “telecommunications service” or an “information service” provider?

- the answer *really* matters from a regulatory standpoint!

US Telecommunication Act of 1934 and 1996:

- Title II:** imposes “common carrier duties” on **telecommunications services**: reasonable rates, non-discrimination and *requires regulation*
- Title I:** applies to **information services**:
  - no common carrier duties (*not regulated*)
  - but grants FCC authority “... as may be necessary in the execution of its functions”.

41

# Assignment Project Exam Help

Network Layer: 4-41

## Network layer: “data plane” roadmap

- Network layer: overview
  - data plane
  - control plane
- What’s inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol**
  - datagram format
  - addressing
  - network address translation
  - IPv6
- Generalized Forwarding, SDN
  - match+action
  - OpenFlow: match+action in action
- Middleboxes



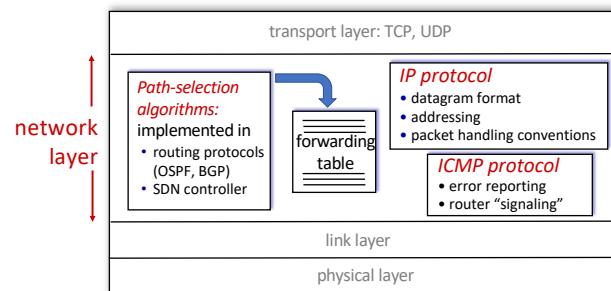
Network Layer: 4-42

42

<https://powcoder.com>

## Network Layer: Internet

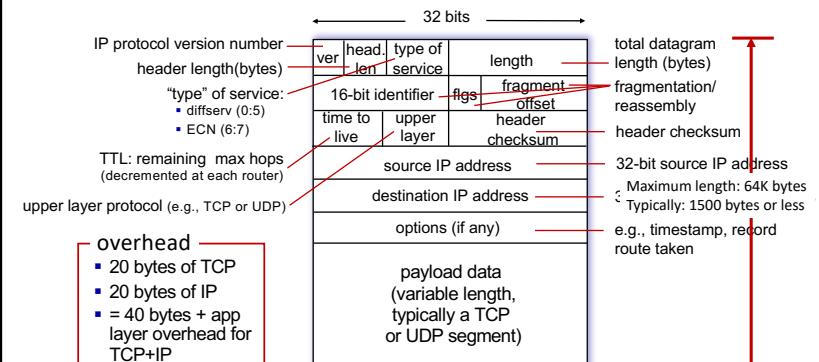
host, router network layer functions:



Network Layer: 4-43

## Add WeChat powcoder

### IP Datagram format



Network Layer: 4-44

43

44

## IP addressing: introduction

- **IP address:** 32-bit identifier associated with each host or router *interface*
- **interface:** connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)

dotted-decimal IP address notation:  
 $223.1.1.1 = 01011110\ 00000001\ 00000001\ 00000001$   
 Network Layer: 4-45

45

## IP addressing: introduction

- **IP address:** 32-bit identifier associated with each host or router *interface*
- **interface:** connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)

dotted-decimal IP address notation:  
 $223.1.1.1 = 11011111\ 00000001\ 00000001\ 00000001$   
 Network Layer: 4-46

46

Assignment Project Exam Help  
<https://powcoder.com>

## IP addressing: introduction

**Q:** how are interfaces actually connected?

**A:** we'll learn about that in chapters 6, 7

**A:** wired Ethernet interfaces connected by Ethernet switches

**For now:** don't need to worry about how one interface is connected to another (with no intervening router)

**A:** wireless WiFi interfaces connected by WiFi base station

Network Layer: 4-47

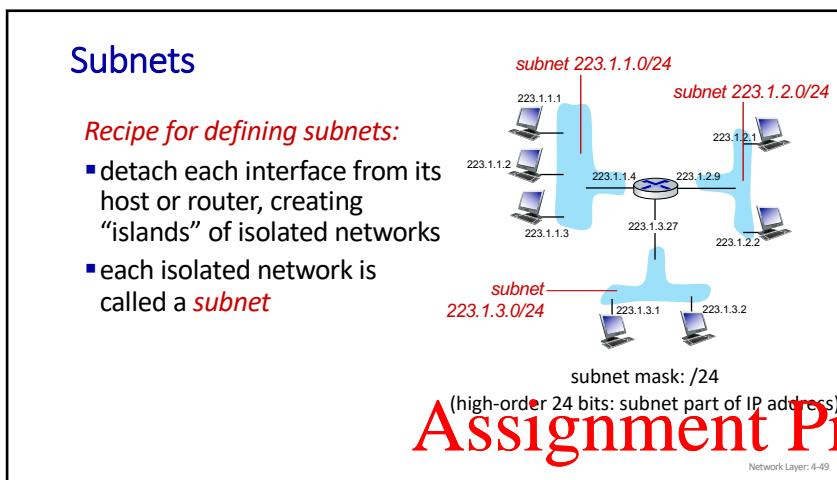
47

## Add WeChat powcoder Subnets

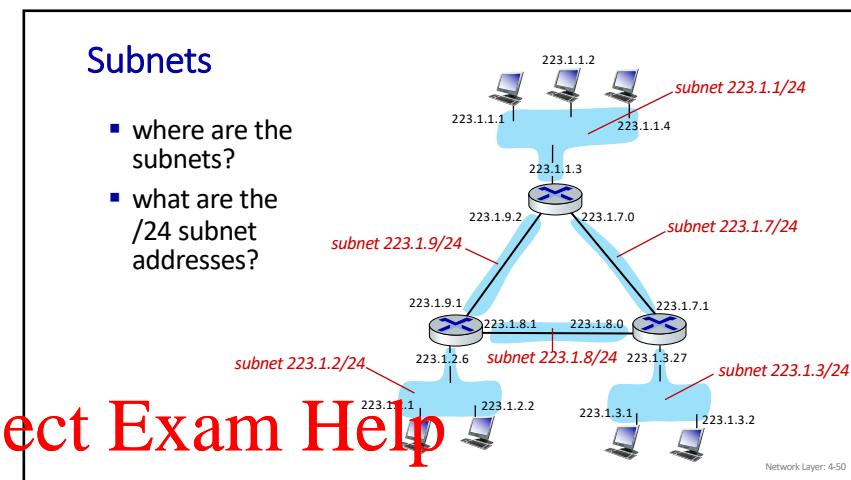
- **What's a subnet ?**
  - device interfaces that can physically reach each other **without passing through an intervening router**
- **IP addresses have structure:**
  - **subnet part:** devices in same subnet have common high order bits
  - **host part:** remaining low order bits

network consisting of 3 subnets  
 Network Layer: 4-48

48

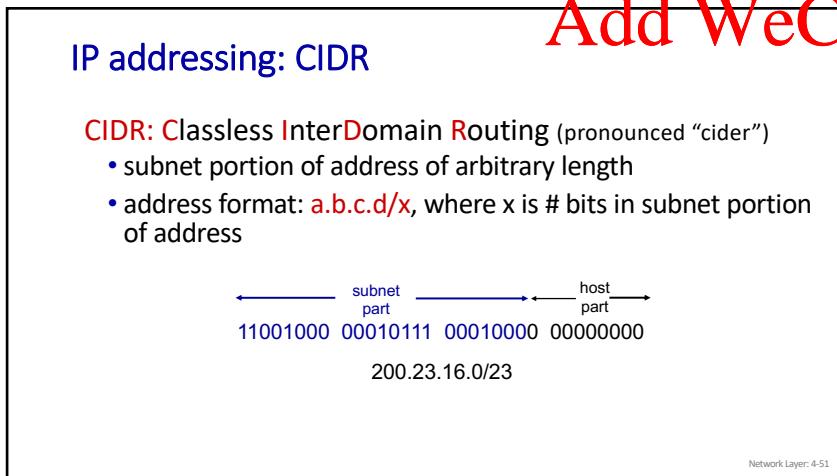


49

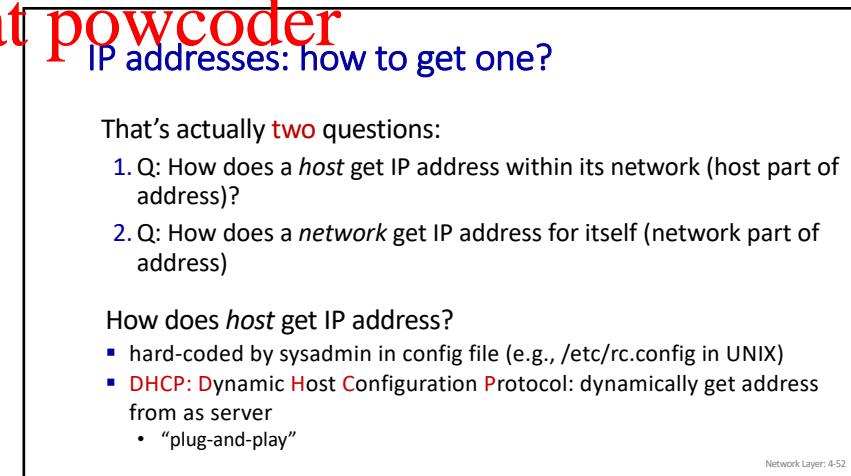


50

Assignment Project Exam Help  
<https://powcoder.com>



51



52

## DHCP: Dynamic Host Configuration Protocol

**goal:** host *dynamically* obtains IP address from network server when it "joins" network

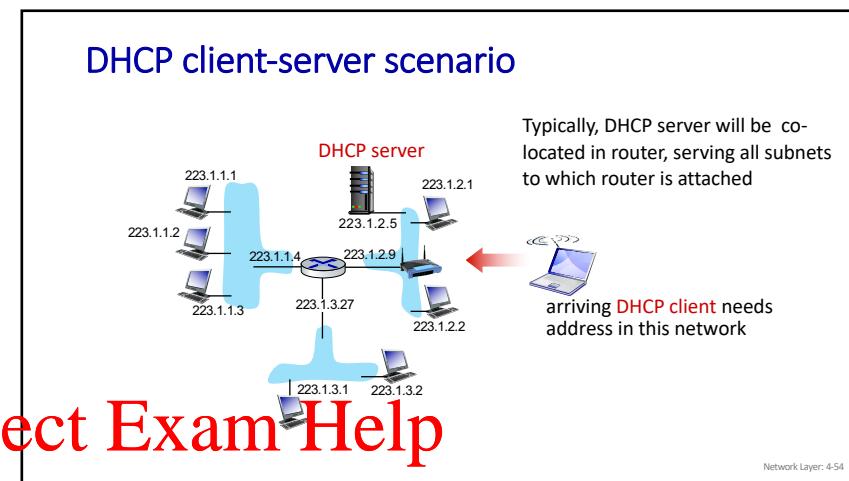
- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/on)
- support for mobile users who join/leave network

**DHCP overview:**

- host broadcasts **DHCP discover** msg [optional]
- DHCP server responds with **DHCP offer** msg [optional]
- host requests IP address: **DHCP request** msg
- DHCP server sends address: **DHCP ACK** msg

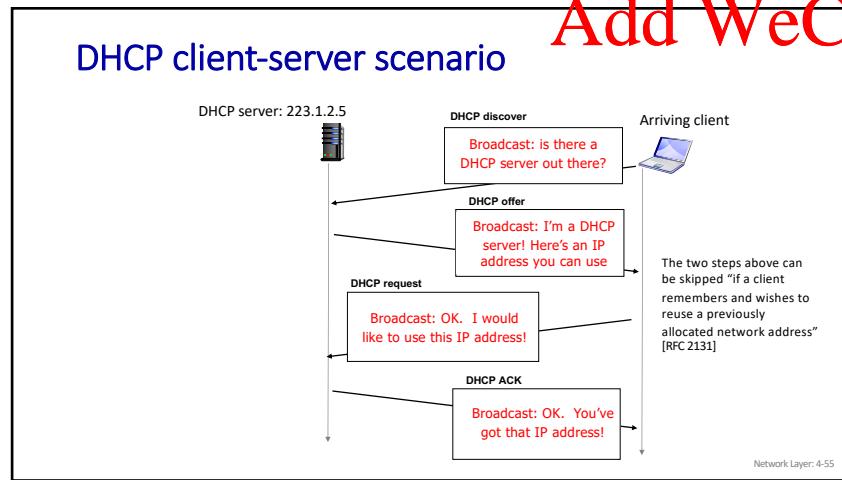
Assignment Project Exam Help  
Network Layer: 4-53

53

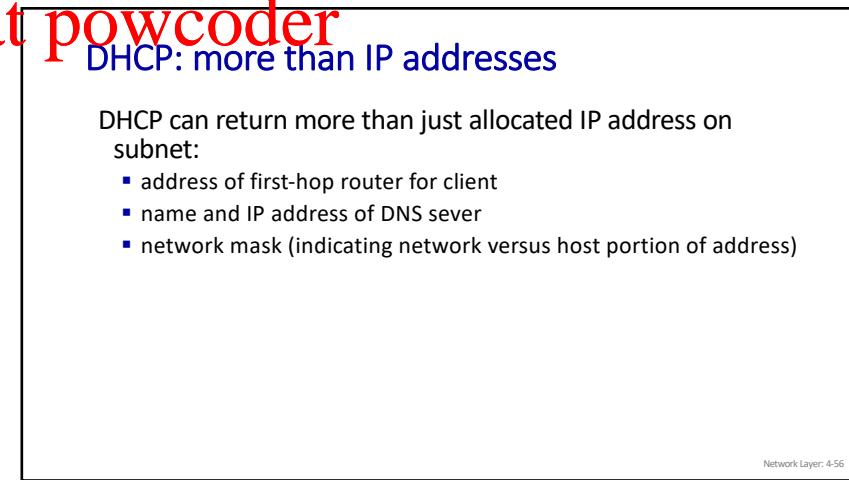


54

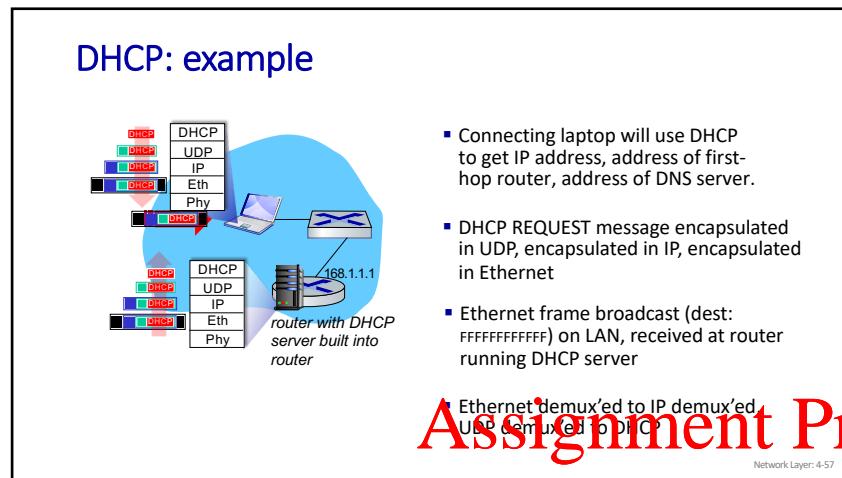
https://powcoder.com



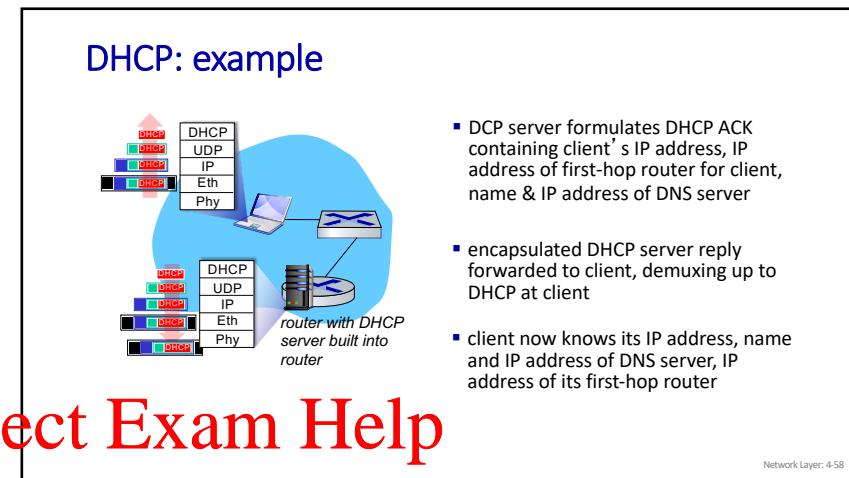
55



56

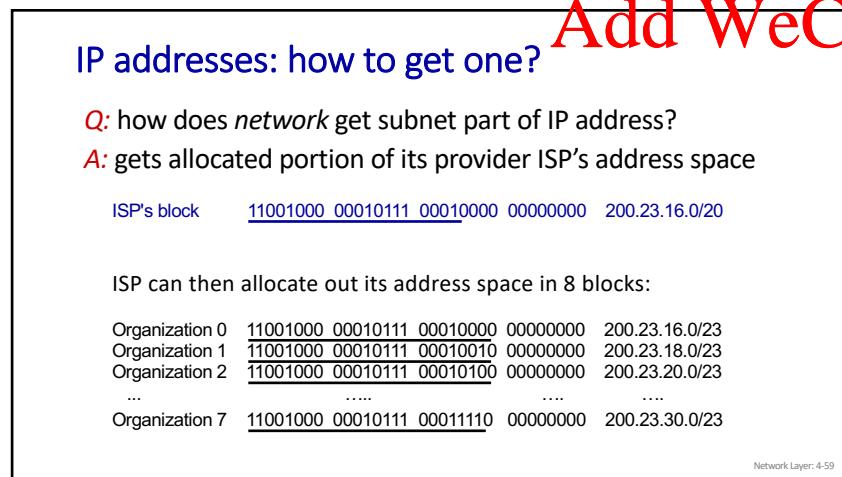


57

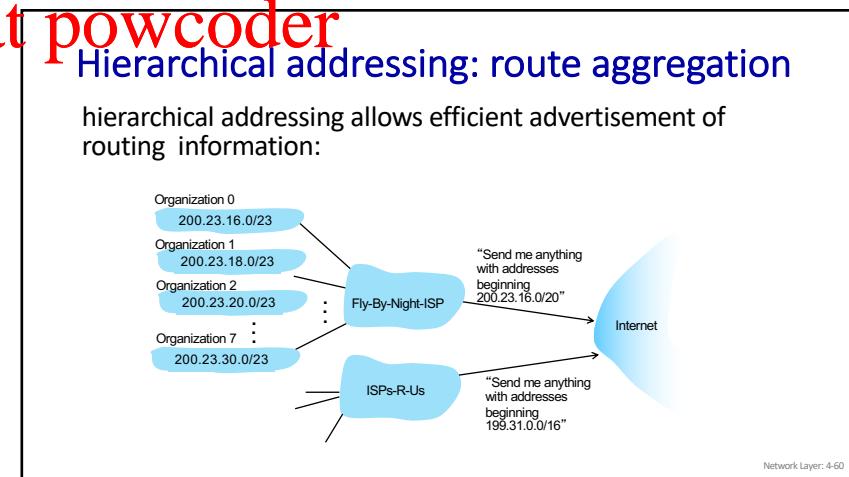


58

<https://powcoder.com>



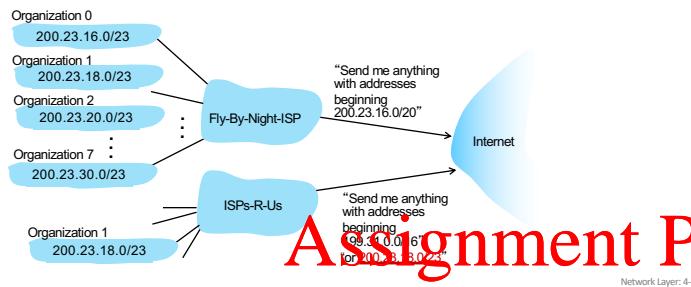
59



60

## Hierarchical addressing: more specific routes

- Organization 1 moves from Fly-By-Night-ISP to ISPs-R-Us
- ISPs-R-Us now advertises a more specific route to Organization 1



61

## Hierarchical addressing: more specific routes

- Organization 1 moves from Fly-By-Night-ISP to ISPs-R-Us
- ISPs-R-Us now advertises a more specific route to Organization 1



62

<https://powcoder.com>

## IP addressing: last words ...

**Q:** how does an ISP get block of addresses?

**A:** ICANN: Internet Corporation for Assigned Names and Numbers <http://www.icann.org/>

- allocates IP addresses, through 5 regional registries (RRs) (who may then allocate to local registries)
- manages DNS root zone, including delegation of individual TLD (.com, .edu, ...) management

**Q:** are there enough 32-bit IP addresses?

- ICANN allocated last chunk of IPv4 addresses to RRs in 2011
- NAT (next) helps IPv4 address space exhaustion
- IPv6 has 128-bit address space

"Who the hell knew how much address space we needed?" Vint Cerf (reflecting on decision to make IPv4 address 32 bits long)

Network Layer: 4-63

## Add WeChat powcoder

### Network layer: "data plane" roadmap

- Network layer: overview
  - data plane
  - control plane
- What's inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6
- Generalized Forwarding, SDN
  - match+action
  - OpenFlow: match+action in action
- Middleboxes



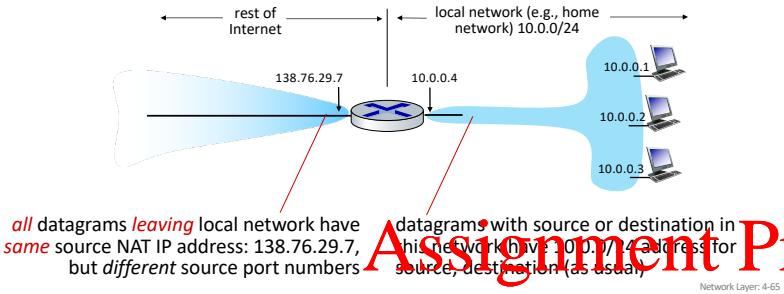
Network Layer: 4-64

63

64

## NAT: network address translation

**NAT:** all devices in local network share just **one** IPv4 address as far as outside world is concerned



65

## NAT: network address translation

- all devices in local network have 32-bit addresses in a “private” IP address space (10/8, 172.16/12, 192.168/16 prefixes) that can only be used in local network
- advantages:
  - just **one** IP address needed from provider ISP for **all** devices
  - can change addresses of host in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - security: devices inside local net not directly addressable, visible by outside world

Network Layer: 4-66

66

Assignment Project Exam Help

## NAT: network address translation

**implementation:** NAT router must (transparently):

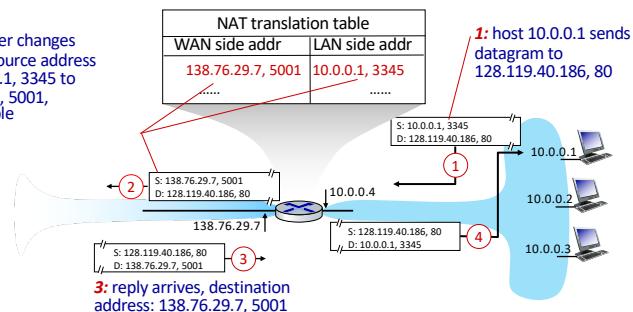
- **outgoing datograms:** **replace** (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  - remote clients/servers will respond using (NAT IP address, new port #) as destination address
- **remember (in NAT translation table)** every (source IP address, port #) to (NAT IP address, new port #) translation pair
- **incoming datograms:** **replace** (NAT IP address, new port #) in destination fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

Network Layer: 4-67

67

## NAT: network address translation

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



68

## NAT: network address translation

- NAT has been controversial:
  - routers “should” only process up to layer 3
  - address “shortage” should be solved by IPv6
  - violates end-to-end argument (port # manipulation by network-layer device)
  - NAT traversal: what if client wants to connect to server behind NAT?
- but NAT is here to stay:
  - extensively used in home and institutional nets, 4G/5G cellular nets

# Assignment Project Exam Help

Network Layer: 4-69

69

## IPv6: motivation

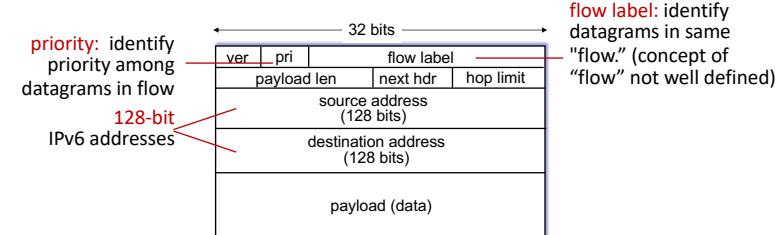
- **initial motivation:** 32-bit IPv4 address space would be completely allocated
- **additional motivation:**
  - speed processing/forwarding: 40-byte fixed length header
  - enable different network-layer treatment of “flows”

Network Layer: 4-70

70

<https://powcoder.com>

## IPv6 datagram format



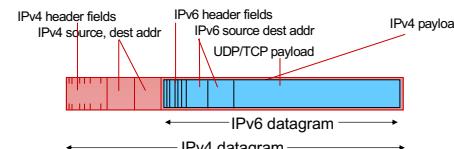
Network Layer: 4-71

71

# Add WeChat powcoder

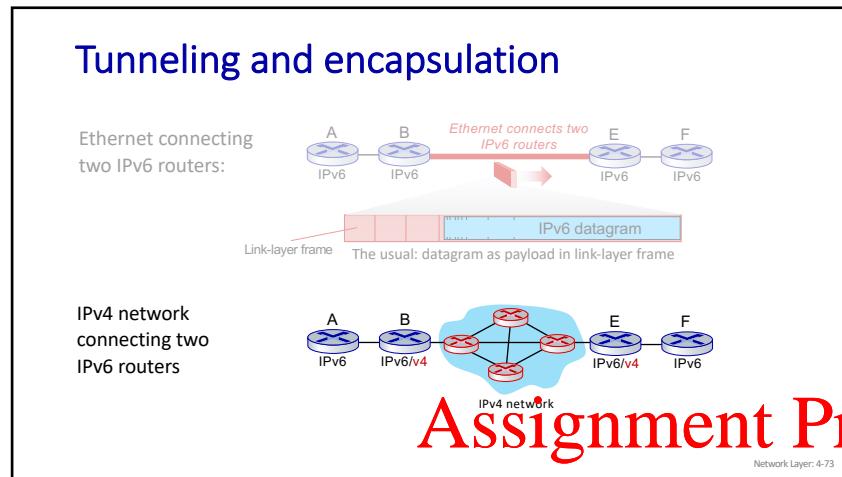
## Transition from IPv4 to IPv6

- not all routers can be upgraded simultaneously
  - no “flag days”
  - how will network operate with mixed IPv4 and IPv6 routers?
- **tunneling:** IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers (“packet within a packet”)
  - tunneling used extensively in other contexts (4G/5G)

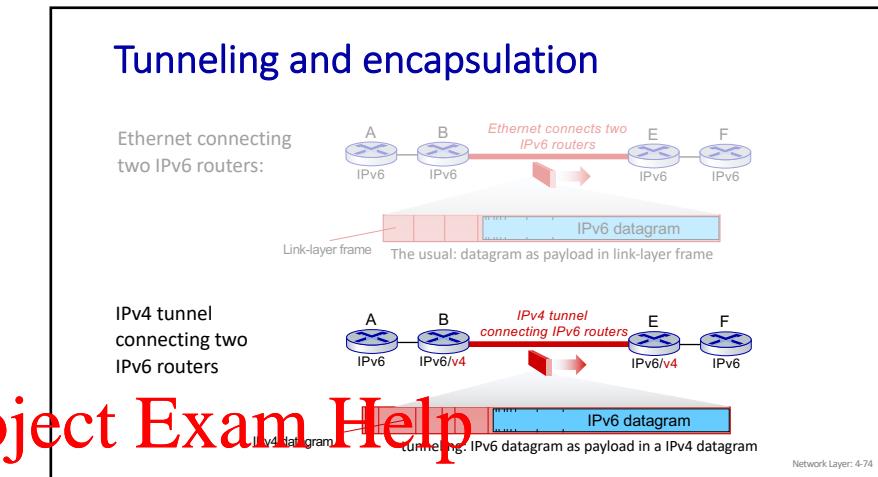


Network Layer: 4-72

72

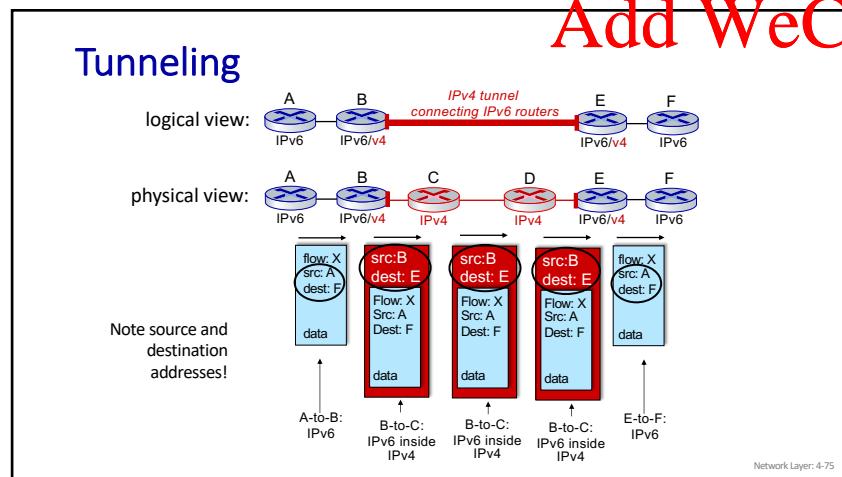


73

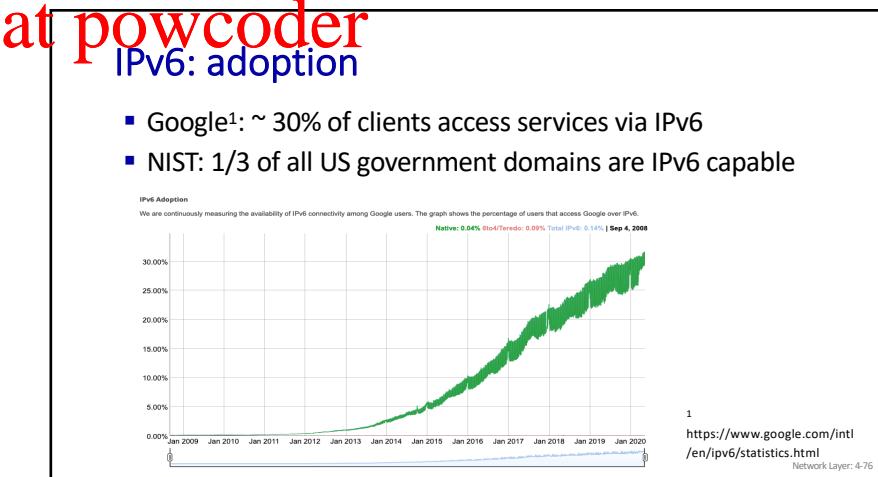


74

Assignment Project Exam Help  
<https://powcoder.com>



75



76

## IPv6: adoption

- Google<sup>1</sup>: ~ 30% of clients access services via IPv6
- NIST: 1/3 of all US government domains are IPv6 capable
- Long (long!) time for deployment, use
  - 25 years and counting!
  - think of application-level changes in last 25 years: WWW, social media, streaming media, gaming, telepresence, ...
  - *Why?*

<sup>1</sup> <https://www.google.com/intl/en/ipv6/statistics.html>

Network Layer: 4-77

77

# Assignment Project Exam Help

## Network layer: “data plane” roadmap

- Network layer: overview
  - data plane
  - control plane
- What’s inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - QoS
- Generalized Forwarding, SDN
  - Match+action
  - OpenFlow: match+action in action
- Middleboxes



Network Layer: 4-78

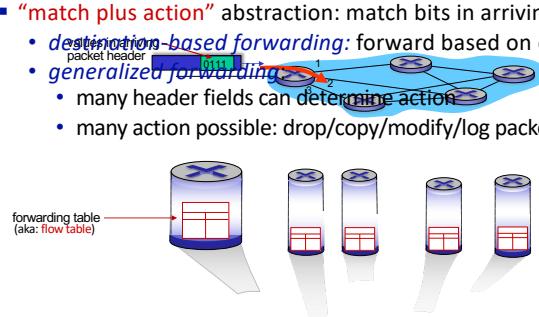
78

<https://powcoder.com>

## Generalized forwarding: match plus action

*Review:* each router contains a **forwarding table** (aka: **flow table**)

- “**match plus action**” abstraction: match bits in arriving packet, take action
  - **destination-based forwarding**: forward based on dest. IP address
  - **generalized forwarding**
    - many header fields can determine action
    - many actions possible: drop/copy/modify/log packet



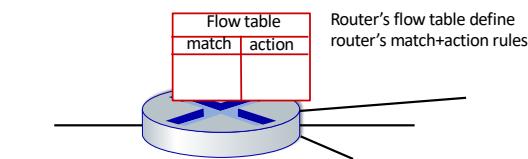
Network Layer: 4-79

79

## Add WeChat powcoder

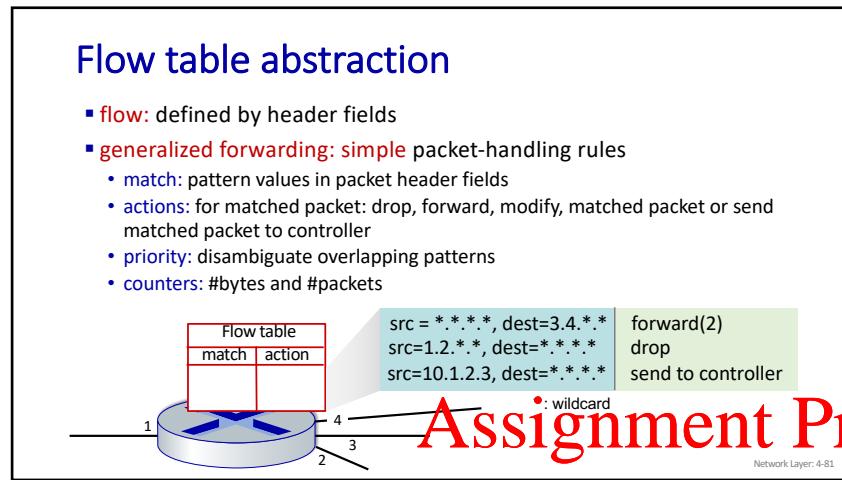
## Flow table abstraction

- **flow**: defined by header field values (in link-, network-, transport-layer fields)
- **generalized forwarding**: simple packet-handling rules
  - **match**: pattern values in packet header fields
  - **actions**: for matched packet: drop, forward, modify, matched packet or send matched packet to controller
  - **priority**: disambiguate overlapping patterns
  - **counters**: #bytes and #packets

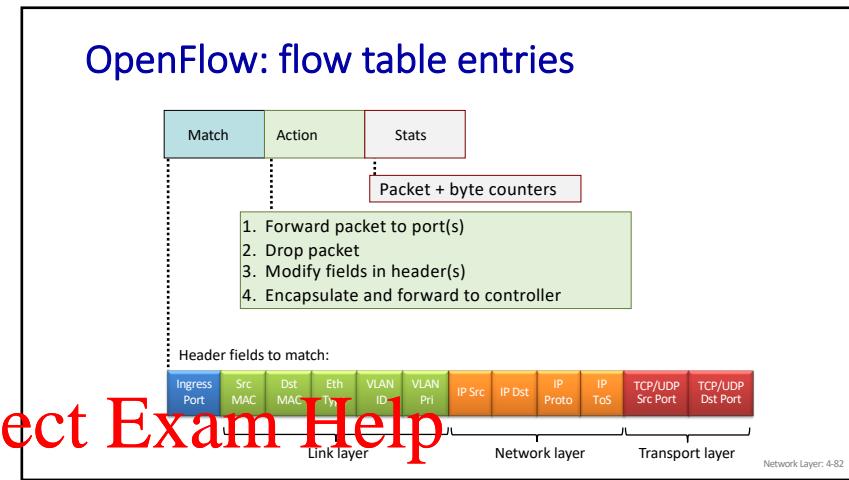


Network Layer: 4-80

80



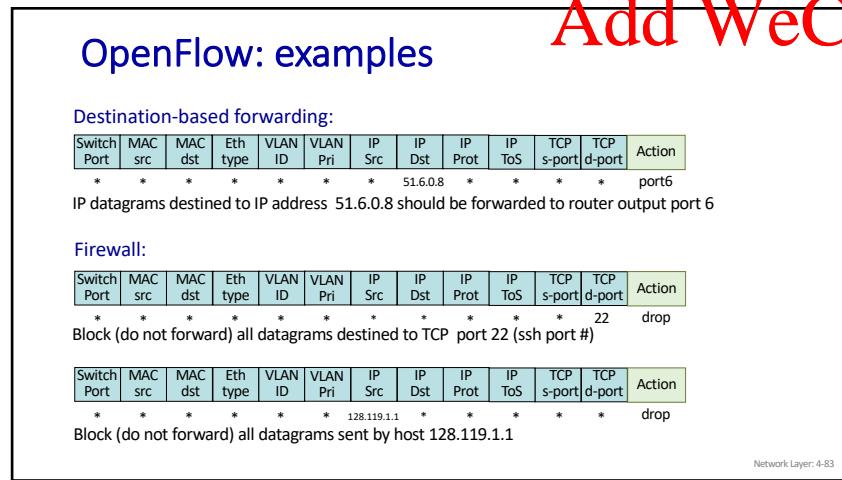
81



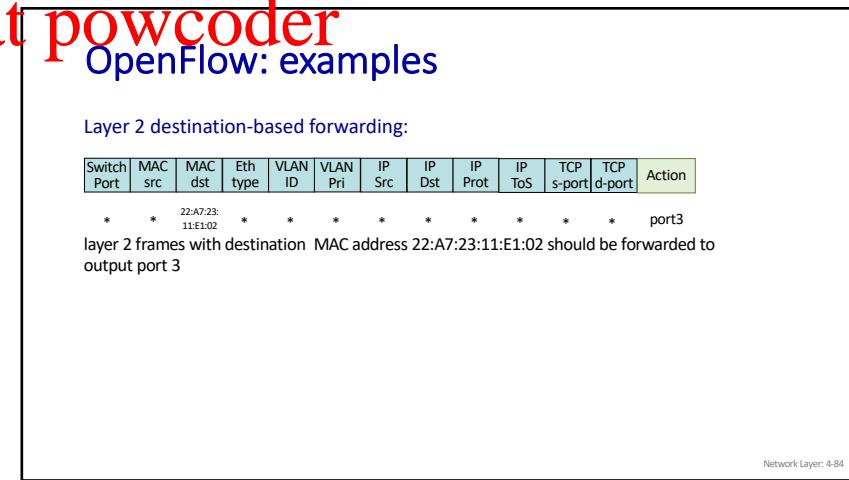
82

Assignment Project Exam Help

<https://powcoder.com>



83



84

## OpenFlow abstraction

- match+action: abstraction unifies different kinds of devices

<b>Router</b>	<b>Firewall</b>
<ul style="list-style-type: none"> <li>match: longest destination IP prefix</li> <li>action: forward out a link</li> </ul>	<ul style="list-style-type: none"> <li>match: IP addresses and TCP/UDP port numbers</li> <li>action: permit or deny</li> </ul>
<b>Switch</b>	<b>NAT</b>
<ul style="list-style-type: none"> <li>match: destination MAC address</li> <li>action: forward or flood</li> </ul>	<ul style="list-style-type: none"> <li>match: IP address and port</li> <li>action: rewrite address and port</li> </ul>

Network Layer: 4-85

85

## 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

Network Layer: 4-86

86

Assignment Project Exam Help

## 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

<b>match</b>	<b>action</b>
IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(3)

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

<b>match</b>	<b>action</b>
ingress port = 2 IP Dst = 10.2.0.3 ingress port = 2 IP Dst = 10.2.0.4	forward(3) forward(4)

Network Layer: 4-87

87

Add WeChat powcoder

Generalized forwarding: summary

- "match plus action" abstraction: match bits in arriving packet header(s) in any layers, take action
  - matching over many fields (link-, network-, transport-layer)
  - local actions: drop, forward, modify, or send matched packet to controller
  - "program" network-wide behaviors
- simple form of "network programmability"
  - programmable, per-packet "processing"
  - historical roots: active networking
  - today: more generalized programming: P4 (see p4.org).

88

## Network layer: “data plane” roadmap

- Network layer: overview
- What's inside a router
- IP: the Internet Protocol
- Generalized Forwarding
- Middleboxes
  - middlebox functions
  - evolution, architectural principles of the Internet



Network Layer: 4-89

89

## 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”

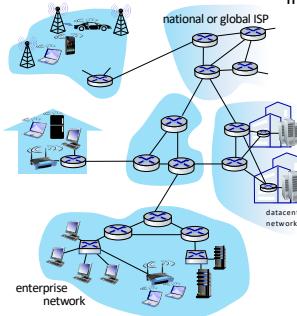
## Assignment Project Exam Help

90

<https://powcoder.com>

## Middleboxes everywhere!

**NAT:** home, cellular, institutional



**Application-specific:** service providers, institutional, CDN

## Add WeChat powcoder

### Middleboxes

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

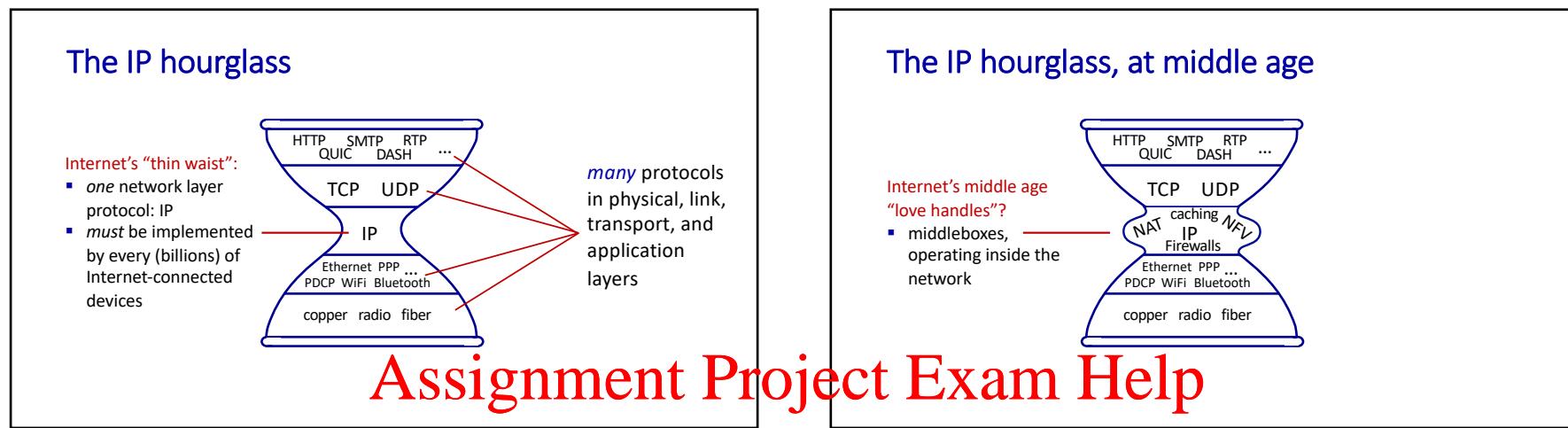
**Load balancers:** corporate, service provider, data center, mobile nets

**Caches:** service provider, mobile, CDNs

- initially: proprietary (closed) hardware solutions
- move towards “whitebox” hardware implementing open API
  - 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

91

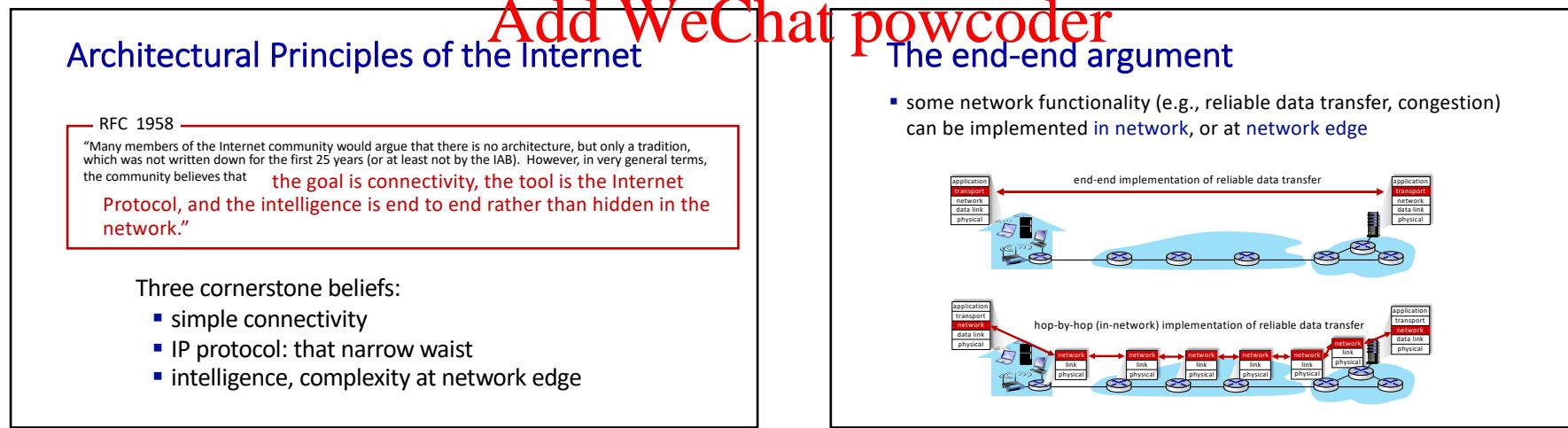
92



93

94

<https://powcoder.com>



95

96

## The end-end argument

- some network functionality (e.g., reliable data transfer, congestion) can be implemented **in network**, or at **network edge**

"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)

We call this line of reasoning against low-level function implementation the "end-to-end argument."

Saltzer, Reed, Clark, 1981

# Assignment Project Exam Help

## Where's the intelligence?



20<sup>th</sup> century phone net:

- intelligence/computing at network switches

Internet (pre-2005)

- intelligence, computing at edge

Internet (post-2005)

- programmable network devices
- intelligence, computing, massive application-level infrastructure at edge

97

98

<https://powcoder.com>

## Chapter 4: done!

- Network layer: overview
- What's inside a router
- IP: the Internet Protocol
- Generalized Forwarding, SDN
- Middleboxes



## Add WeChat powcoder

Additional Chapter 4 slides

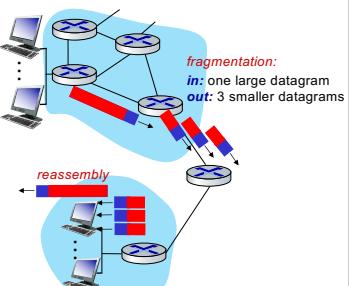
99

100

Network Layer: 4-100

### IP fragmentation/reassembly

- network links have MTU (max. transfer size) - largest possible link-level frame
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at destination
  - IP header bits used to identify, order related fragments



Network Layer: 4-101

### IP fragmentation/reassembly

**example:**

- 4000 byte datagram
- MTU = 1500 bytes

one large datagram becomes several smaller datagrams

length = 4000	ID = x	fragflag = 0	offset = 0	[ ]
---------------	--------	--------------	------------	-----

1480 bytes in data field

offset = 1480/8

length = 1500	ID = x	fragflag = 1	offset = 0	[ ]
---------------	--------	--------------	------------	-----

length = 1500	ID = x	fragflag = 1	offset = 185	[ ]
---------------	--------	--------------	--------------	-----

length = 1040	ID = x	fragflag = 0	offset = 370	[ ]
---------------	--------	--------------	--------------	-----

Network Layer: 4-102

101

102

Assignment Project Exam Help

Add WeChat powcoder

**DHCP: Wireshark output (home LAN)**

<b>request</b>	<b>reply</b>
----------------	--------------

Message type: **Boot Request (1)**  
 Hardware type: Ethernet  
 Hardware address length: 6  
 Hops: 0  
**Transaction ID: 0x6b3a11b7**  
 Seconds elapsed: 0  
 Boot 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: Wistron\_23:68:8a (00:16:d3:23:68:8a)**  
 Server host name not given  
 Boot file name not given  
 Magic cookie: (OK)  
 Option: (f=5,i=1) **DHCP Message Type = DHCP Request**  
 Option: (f=1,i=1) Client Identifier  
 Length: 7 Value: 010016D323688A;  
 Hardware type: Ethernet  
 Client MAC address: Wistron\_23:68:8a (00:16:d3:23:68:8a)  
 Option: (f=50,i=4) Requested IP Address = 192.168.1.101  
 Option: (f=12,i=5) Host Name = "nomad"  
**Option: (55) Parameter Request List**  
 Length: 11 Value: 010F03062C2E2F1F21F92B  
 1 = Subnet Mask; 15 = Domain Name  
 3 = Router; 6 = Domain Name Server  
 44 = NetBIOS over TCP/IP Name Server

Message type: **Boot Reply (2)**  
 Hardware type: Ethernet  
 Hardware address length: 6  
 Hops: 0  
**Transaction ID: 0x6b3a11b7**  
 Seconds elapsed: 0  
 Boot flags: 0x0000 (Unicast)  
**Client IP address: 192.168.1.101 (192.168.1.101)**  
 Your (client) IP address: 0.0.0.0 (0.0.0.0)  
**Next server IP address: 192.168.1.1 (192.168.1.1)**  
 Relay agent IP address: 0.0.0.0 (0.0.0.0)  
 Client MAC address: Wistron\_23:68:8a (00:16:d3:23:68:8a)  
 Server host name not given  
 Boot file name not given  
 Magic cookie: (OK)  
 Option: (f=53,i=1) **DHCP Message Type = DHCP ACK**  
 Option: (f=54,i=4) Server Identifier = 192.168.1.1  
 Option: (f=1,i=4) Subnet Mask = 255.255.255.0  
 Option: (f=3,i=4) Router = 192.168.1.1  
 Option: (6) Domain Name Server  
 Length: 12 Value: 445747E2445749F244574092;  
 IP Address: 68.87.71.226;  
 IP Address: 68.87.73.242;  
 IP Address: 68.87.64.146  
 Option: (f=15,i=20) Domain Name = "hsd1.ma.comcast.net."

Network Layer: 4-103

103