

Chapter 6 The Link Layer and LANs

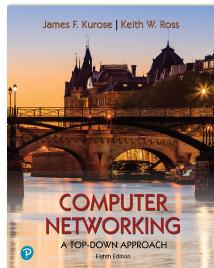
A note on the use of these PowerPoint slides:
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent 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! JFK/KVR

All material copyright 1996-2020
J.F Kurose and K.W. Ross, All Rights Reserved



*Computer Networking: A
Top-Down Approach*
8th edition
James F. Kurose, Keith W. Ross
Pearson, 2018

1

Link layer and LANs: our goals

▪ understand principles behind link layer services:

- error detection, correction
- sharing a broadcast channel: multiple access
- link layer addressing
- local area networks: Ethernet, VLANs

▪ instantiation, implementation of various link layer technologies



Link Layer: 6-2

Assignment Project Exam Help

2

<https://powcoder.com>

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
 - addressing, ARP
 - Ethernet
 - switches
 - VLANs
- link virtualization: MPLS
- data center networking



- a day in the life of a web request

Link Layer: 6-3

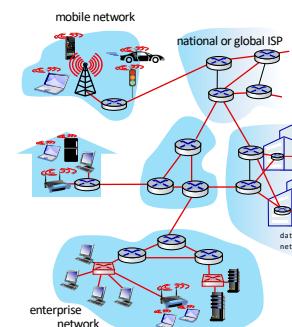
Add WeChat powcoder

Link layer: introduction

terminology:

- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
 - wired
 - wireless
 - LANs
- layer-2 packet: *frame*, encapsulates datagram

link layer has responsibility of transferring datagram from one node to physically adjacent node over a link



Link Layer: 6-4

3

4

Link layer: context

- datagram transferred by different link protocols over different links:
 - e.g., WiFi on first link, Ethernet on next link
- each link protocol provides different services
 - e.g., may or may not provide reliable data transfer over link

transportation analogy:

- trip from Princeton to Lausanne
 - limo: Princeton to JFK
 - plane: JFK to Geneva
 - train: Geneva to Lausanne
- tourist = datagram
- transport segment = communication link
- transportation mode = link-layer protocol
- travel agent = routing algorithm

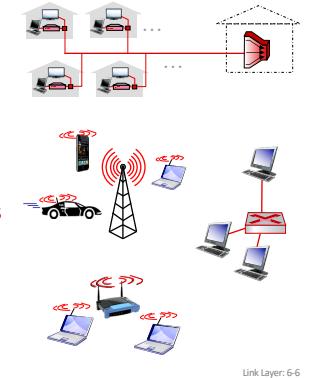
Link Layer: 6-5

Assignment Project Exam Help

5

Link layer: services

- framing, link access:**
 - encapsulate datagram into frame, adding header, trailer
 - channel access if shared medium
 - “MAC” addresses in frame headers identify source, destination (different from IP address!)
- reliable delivery between adjacent nodes**
 - we already know how to do this!
 - seldom used on low bit-error links
 - wireless links: high error rates
 - Q: why both link-level and end-end reliability?



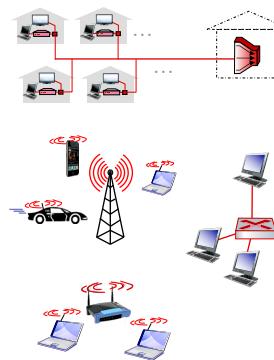
Link Layer: 6-6

6

<https://powcoder.com>

Link layer: services (more)

- flow control:**
 - pacing between adjacent sending and receiving nodes
- error detection:**
 - errors caused by signal attenuation, noise.
 - receiver detects errors, signals retransmission, or drops frame
- error correction:**
 - receiver identifies *and corrects* bit error(s) without retransmission
- half-duplex and full-duplex:**
 - with half duplex, nodes at both ends of link can transmit, but not at same time



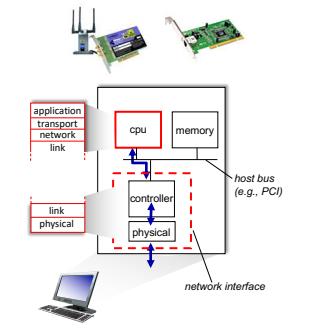
Link Layer: 6-7

7

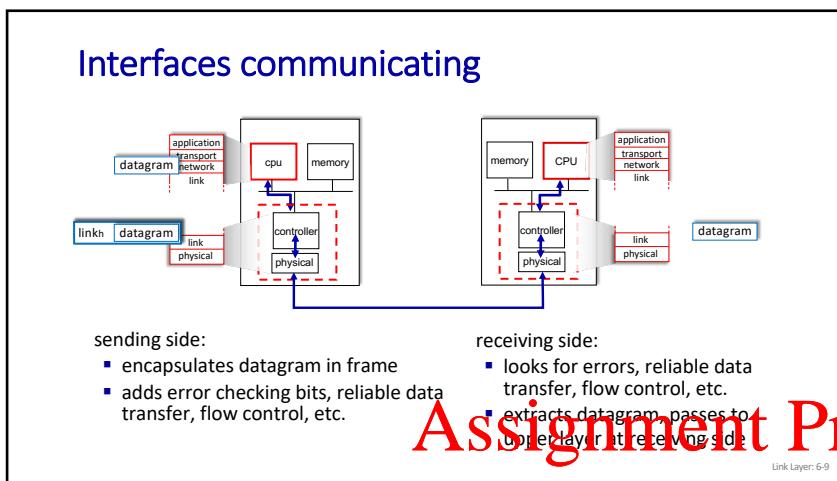
Add WeChat powcoder

Where is the link layer implemented?

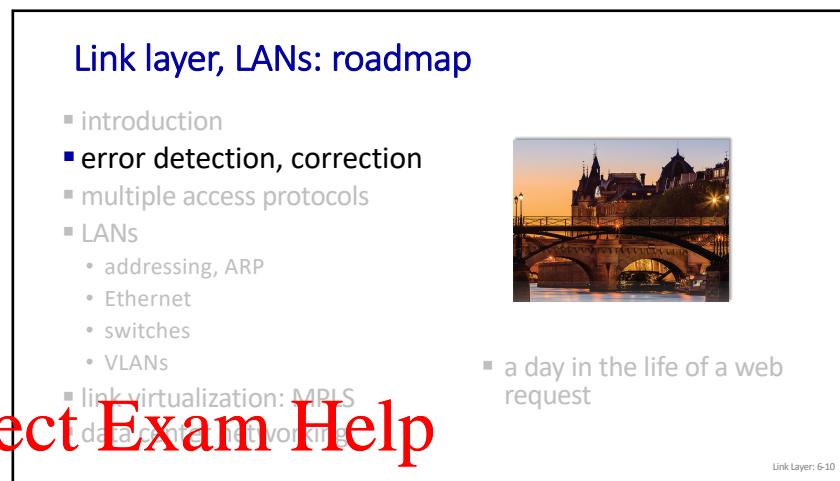
- in each-and-every host
- link layer implemented in *network interface card* (NIC) or on a chip
 - Ethernet, WiFi card or chip
 - implements link, physical layer
- attaches into host’s system buses
- combination of hardware, software, firmware



8



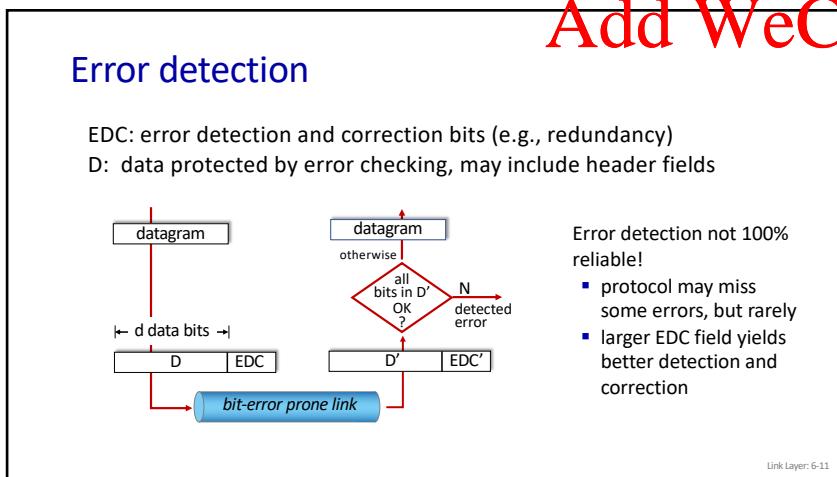
9



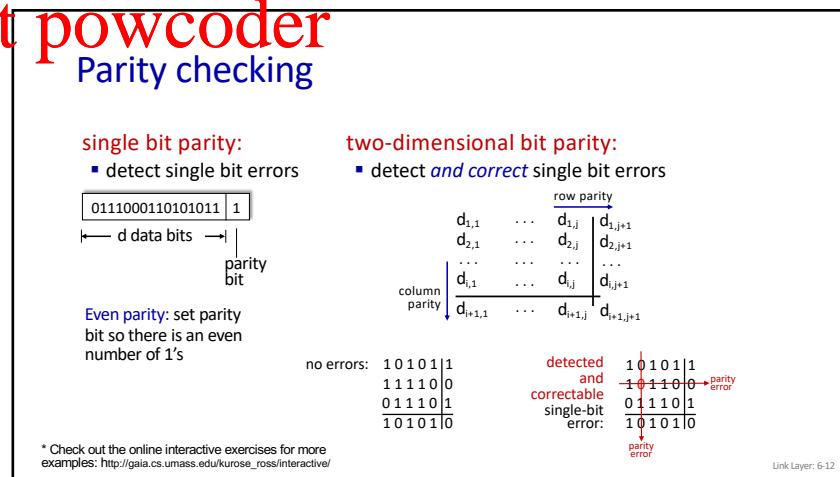
10

Assignment Project Exam Help

<https://powcoder.com>



11



12

Internet checksum (review)

Goal: detect errors (i.e., flipped bits) in transmitted segment

sender:

- treat contents of UDP segment (including UDP header fields and IP addresses) as sequence of 16-bit integers
- **checksum:** addition (one's complement sum) of segment content
- checksum value put into UDP checksum field

receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
 - not equal - error detected
 - equal - no error detected. *But maybe errors nonetheless? More later*

Transport Layer: 3-13

13

Cyclic Redundancy Check (CRC)

- more powerful error-detection coding
- **D:** data bits (given, think of these as a binary number)
- **G:** bit pattern (generator), of $r+1$ bits (given)



goal: choose r CRC bits, R , such that $\langle D, R \rangle$ exactly divisible by G (mod 2)

- receiver knows G , divides $\langle D, R \rangle$ by G . If non-zero remainder: error detected!

can detect all burst errors less than $r+1$ bits

widely used in practice (Ethernet, 802.11 WiFi)

Link Layer: 6-14

14

Assignment Project Exam Help

<https://powcoder.com>

Link layer, LANs: roadmap

- introduction
- error detection, correction
- **multiple access protocols**
- LANs
 - addressing, ARP
 - Ethernet
 - switches
 - VLANs
- link virtualization: MPLS
- data center networking



- a day in the life of a web request

Link Layer: 6-16

16

Add WeChat powcoder

Multiple access links, protocols

two types of “links”:

- **point-to-point**
 - point-to-point link between Ethernet switch, host
 - PPP for dial-up access
- **broadcast (shared wire or medium)**
 - old-fashioned Ethernet
 - upstream HFC in cable-based access network
 - 802.11 wireless LAN, 4G/4G, satellite



Link Layer: 6-17

17

Multiple access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
 - *collision* if node receives two or more signals at the same time

multiple access protocol

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
 - no out-of-band channel for coordination

An ideal multiple access protocol

given: multiple access channel (MAC) of rate R bps

desiderata:

1. when one node wants to transmit, it can send at rate R .
2. when M nodes want to transmit, each can send at average rate R/M
3. fully decentralized:
 - no special node to coordinate transmissions
 - no synchronization of clocks, slots
4. simple

Assignment Project Exam Help

Link Layer: 6-18

18

19

<https://powcoder.com>

MAC protocols: taxonomy

three broad classes:

- **channel partitioning**
 - divide channel into smaller “pieces” (time slots, frequency, code)
 - allocate piece to node for exclusive use
- **random access**
 - channel not divided, allow collisions
 - “recover” from collisions
- **“taking turns”**
 - nodes take turns, but nodes with more to send can take longer turns

Link Layer: 6-20

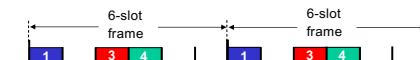
20

Add WeChat powcoder

Channel partitioning MAC protocols: TDMA

TDMA: time division multiple access

- access to channel in “rounds”
- each station gets fixed length slot (length = packet transmission time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle



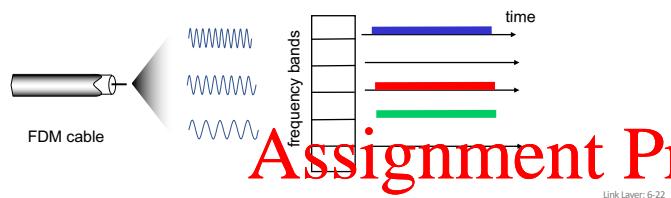
Link Layer: 6-21

21

Channel partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have packet to send, frequency bands 2,5,6 idle



22

Random access protocols

- when node has packet to send
 - transmit at full channel data rate R.
 - no *a priori* coordination among nodes
- two or more transmitting nodes: "collision"
- random access MAC protocol specifies:
 - how to detect collisions
 - how to recover from collisions (e.g., via delayed retransmissions)
- examples of random access MAC protocols:
 - ALOHA, slotted ALOHA
 - CSMA, CSMA/CD, CSMA/CA

Link Layer: 6-23

Assignment Project Exam Help

23

<https://powcoder.com>

Slotted ALOHA

assumptions:

- all frames same size
- time divided into equal size slots (time to transmit 1 frame)
- nodes start to transmit only slot beginning
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

operation:

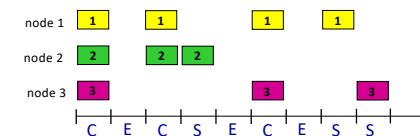
- when node obtains fresh frame, transmits in next slot
 - if no collision:* node can send new frame in next slot
 - if collision:* node retransmits frame in each subsequent slot with probability p until success

randomization – why?

Link Layer: 6-24

Add WeChat powcoder

Slotted ALOHA



Pros:

- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons:

- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet
- clock synchronization

Link Layer: 6-25

24

25

Slotted ALOHA: efficiency

efficiency: long-run fraction of successful slots (many nodes, all with many frames to send)

- **suppose:** N nodes with many frames to send, each transmits in slot with probability p
 - prob that given node has success in a slot = $p(1-p)^{N-1}$
 - prob that *any* node has a success = $Np(1-p)^{N-1}$
 - max efficiency: find p^* that maximizes $Np(1-p)^{N-1}$
 - for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:
max efficiency = $1/e = .37$
- **at best:** channel used for useful transmissions 37% of time!

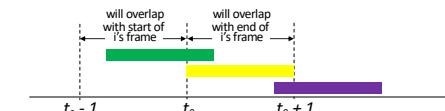
Assignment Project Exam Help

Link Layer: 6-26

26

Pure ALOHA

- unslotted Aloha: simpler, no synchronization
 - when frame first arrives: transmit immediately
- collision probability increases with no synchronization:
 - frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$



▪ pure Aloha efficiency: 18% !

Link Layer: 6-27

27

<https://powcoder.com>

CSMA (carrier sense multiple access)

simple CSMA: listen before transmit:

- if channel sensed idle: transmit entire frame
- if channel sensed busy: defer transmission
- human analogy: don't interrupt others!

CSMA/CD: CSMA with *collision detection*

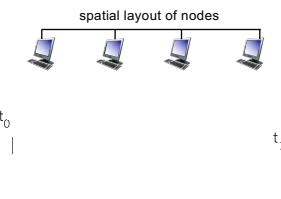
- collisions detected within short time
- colliding transmissions aborted, reducing channel wastage
- collision detection easy in wired, difficult with wireless
- human analogy: the polite conversationalist

Link Layer: 6-28

28

CSMA: collisions

- collisions can still occur with carrier sensing:
 - propagation delay means two nodes may not hear each other's just-started transmission
- **collision:** entire packet transmission time wasted
 - distance & propagation delay play role in determining collision probability

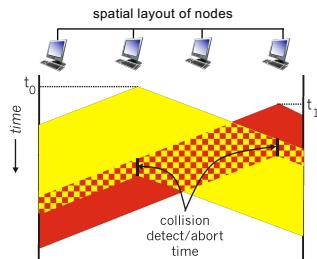


Link Layer: 6-29

29

CSMA/CD:

- CSMA/CS reduces the amount of time wasted in collisions
 - transmission aborted on collision detection



30

Assignment Project Exam Help

Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel:
 - if **idle**: start frame transmission.
 - if **busy**: wait until channel idle, then transmit
3. If NIC transmits entire frame without collision, NIC is done with frame !
4. If NIC detects another transmission while sending: abort, send jam signal
5. After aborting, NIC enters ***binary (exponential) backoff***:
 - after m th collision, NIC chooses K at random from $\{0,1,2, \dots, 2^m-1\}$. NIC waits $K \cdot 512$ bit times, returns to Step 2
 - more collisions: longer backoff interval

Link Layer: 6-31

31

<https://powcoder.com>

CSMA/CD efficiency

- T_{prop} = max prop delay between 2 nodes in LAN
 - t_{trans} = time to transmit max-size frame
- $$\text{efficiency} = \frac{1}{1 + 5t_{prop}/t_{trans}}$$
- efficiency goes to 1
 - as t_{prop} goes to 0
 - as t_{trans} goes to infinity
 - better performance than ALOHA: and simple, cheap, decentralized!

Link Layer: 6-32

32

Add WeChat powcoder

"Taking turns" MAC protocols

channel partitioning MAC protocols:

- share channel *efficiently* and *fairly* at high load
- inefficient at low load: delay in channel access, $1/N$ bandwidth allocated even if only 1 active node!

random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

"taking turns" protocols

- look for best of both worlds!

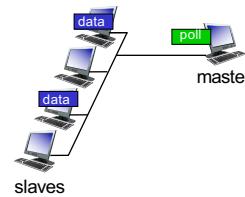
Link Layer: 6-33

33

"Taking turns" MAC protocols

polling:

- master node "invites" other nodes to transmit in turn
- typically used with "dumb" devices
- concerns:
 - polling overhead
 - latency
 - single point of failure (master)

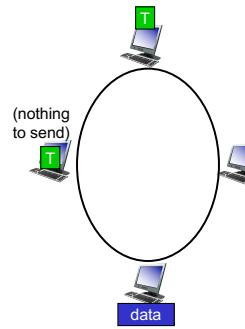


Link Layer: 6-34

"Taking turns" MAC protocols

token passing:

- control **token** passed from one node to next sequentially.
- token message
- concerns:
 - token overhead
 - latency
 - single point of failure (token)



Link Layer: 6-35

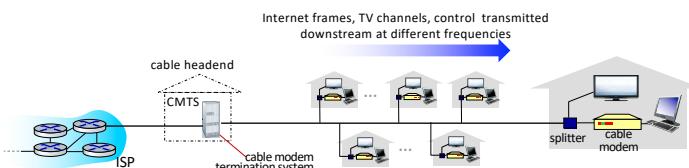
Assignment Project Exam Help

34

35

<https://powcoder.com>

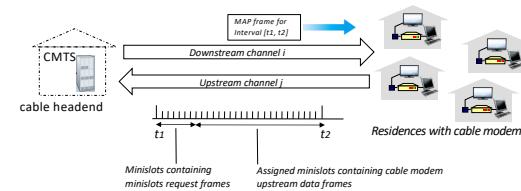
Cable access network: FDM, TDM and random access!



- **multiple** downstream (broadcast) FDM channels: up to 1.6 Gbps/channel
 - single CMTS transmits into channels
- **multiple** upstream channels (up to 1 Gbps/channel)
 - **multiple access:** all users contend (random access) for certain upstream channel time slots; others assigned TDM

Link Layer: 6-36

Cable access network:



DOCSIS: data over cable service interface specification

- FDM over upstream, downstream frequency channels
- TDM upstream: some slots assigned, some have contention
 - downstream MAP frame: assigns upstream slots
 - request for upstream slots (and data) transmitted random access (binary backoff) in selected slots

Link Layer: 6-37

36

37

Summary of MAC protocols

- **channel partitioning**, by time, frequency or code
 - Time Division, Frequency Division
- **random access** (dynamic),
 - ALOHA, S-ALOHA, CSMA, CSMA/CD
 - carrier sensing: easy in some technologies (wire), hard in others (wireless)
 - CSMA/CD used in Ethernet
 - CSMA/CA used in 802.11
- **taking turns**
 - polling from central site, token passing
 - Bluetooth, FDDI, token ring

Link Layer: 6-38

38

Assignment Project Exam Help

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- **LANs**
 - **addressing, ARP**
 - Ethernet
 - switches
 - VLANs
- link virtualization: MPLS
- data center networking



- a day in the life of a web request

Link Layer: 6-39

39

<https://powcoder.com>

MAC addresses

- 32-bit IP address:
 - *network-layer* address for interface
 - used for layer 3 (network layer) forwarding
 - e.g.: 128.119.40.136
- MAC (or LAN or physical or Ethernet) address:
 - function: used “locally” to get frame from one interface to another *physically-connected* interface (same subnet, in IP-addressing sense)
 - 48-bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
 - e.g.: 1A-2F-BB-76-09-AD
 - hexadecimal (base 16) notation
(each “numeral” represents 4 bits)

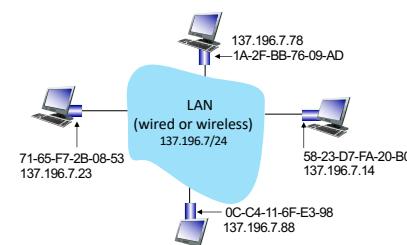
Link Layer: 6-40

40

Add WeChat powcoder

MAC addresses

- each interface on LAN
- has unique 48-bit **MAC address**
 - has a locally unique 32-bit IP address (as we've seen)



Link Layer: 6-41

41

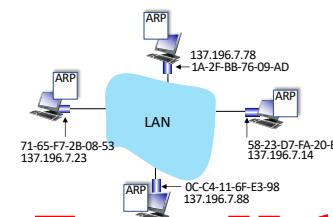
MAC addresses

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- analogy:
 - MAC address: like Social Security Number
 - IP address: like postal address
- MAC flat address: portability
 - can move interface from one LAN to another
 - recall IP address *not* portable: depends on IP subnet to which node is attached

42

ARP: address resolution protocol

Question: how to determine interface's MAC address, knowing its IP address?



ARP table: each IP node (host, router) on LAN has table

- IP/MAC address mappings for some LAN nodes:
< IP address; MAC address; TTL >
- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

Link Layer: 6-43

43

Assignment Project Exam Help

<https://powcoder.com>

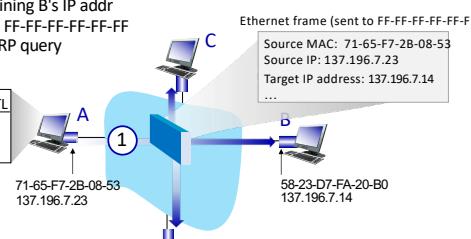
ARP protocol in action

example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

- A broadcasts ARP query, containing B's IP addr
 ① • destination MAC address = FF-FF-FF-FF-FF-FF
 • all nodes on LAN receive ARP query

ARP table in A		
IP addr	MAC addr	TTL



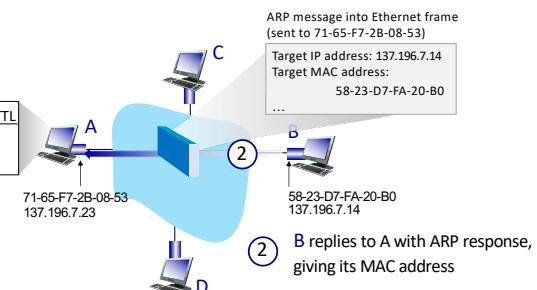
Link Layer: 6-44

Add WeChat powcoder

ARP protocol in action

example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address



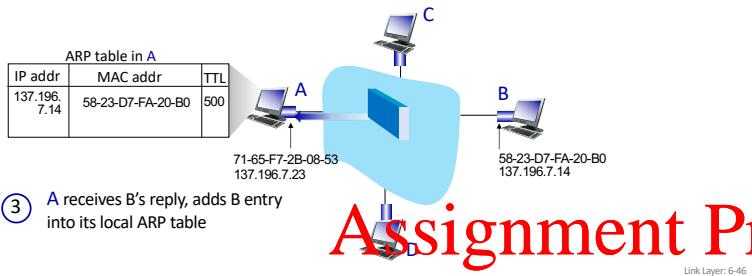
Link Layer: 6-45

44

45

ARP protocol in action

- example: A wants to send datagram to B
- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

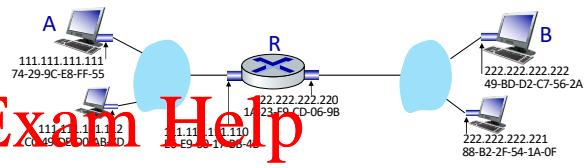


46

Routing to another subnet: addressing

walkthrough: sending a datagram from A to B via R

- focus on addressing – at IP (datagram) and MAC layer (frame) levels
- assume that:
 - A knows B's IP address
 - A knows IP address of first hop router, R (how?)
 - A knows R's MAC address (how?)

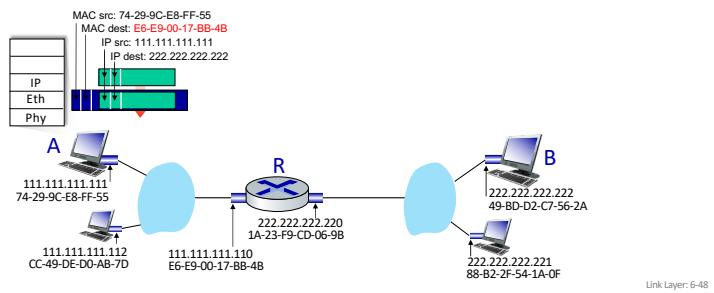


47

<https://powcoder.com>

Routing to another subnet: addressing

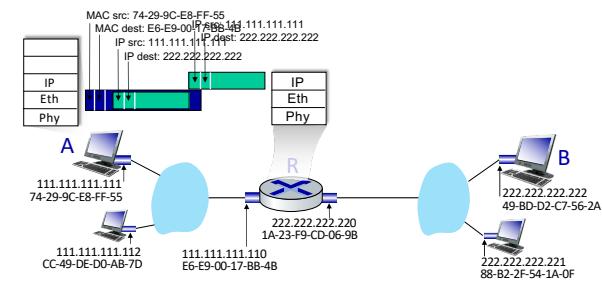
- A creates IP datagram with IP source A, destination B
- A creates link-layer frame containing A-to-B IP datagram
 - R's MAC address is frame's destination



48

Routing to another subnet: addressing

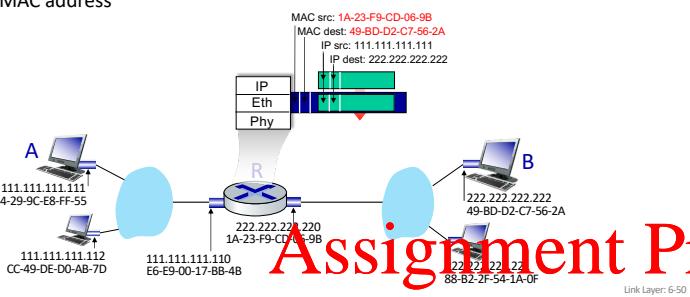
- frame sent from A to R
- frame received at R, datagram removed, passed up to IP



49

Routing to another subnet: addressing

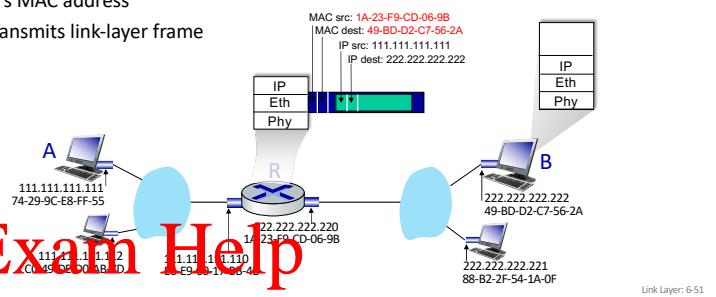
- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address



50

Routing to another subnet: addressing

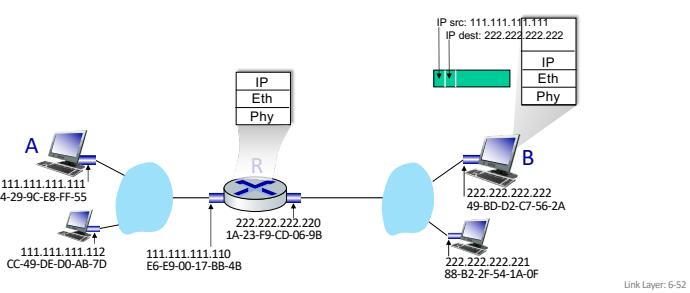
- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address
- transmits link-layer frame



<https://powcoder.com>

Routing to another subnet: addressing

- B receives frame, extracts IP datagram destination B
- B passes datagram up protocol stack to IP



52

Add WeChat powcoder

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs**
 - addressing, ARP
 - Ethernet**
 - switches
 - VLANs
- link virtualization: MPLS
- data center networking



- a day in the life of a web request

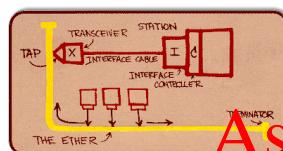
Link Layer: 6-53

53

Ethernet

“dominant” wired LAN technology:

- first widely used LAN technology
- simpler, cheap
- kept up with speed race: 10 Mbps – 400 Gbps
- single chip, multiple speeds (e.g., Broadcom BCM5761)



Metcalfe's Ethernet sketch

<https://www.uspto.gov/learning-and-resources/journeys-innovation/audio-stories/cutting-doubters>

Link Layer: 6-54

Ethernet: physical topology

- **bus:** popular through mid 90s
 - all nodes in same collision domain (can collide with each other)
- **switched:** prevails today
 - active link-layer 2 **switch** in center
 - each “spoke” runs a (separate) Ethernet protocol (nodes do not collide with each other)



Link Layer: 6-55

54

55

<https://powcoder.com>

Ethernet frame structure

sending interface encapsulates IP datagram (or other network layer protocol packet) in **Ethernet frame**

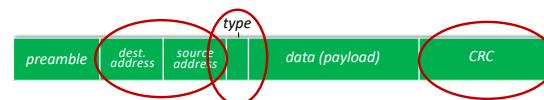


preamble:

- used to synchronize receiver, sender clock rates
- 7 bytes of 10101010 followed by one byte of 10101011

Link Layer: 6-56

Ethernet frame structure (more)



- **addresses:** 6 byte source, destination MAC addresses
 - if adapter receives frame with matching destination address, or with broadcast address (e.g., ARP packet), it passes data in frame to network layer protocol
 - otherwise, adapter discards frame
- **type:** indicates higher layer protocol
 - mostly IP but others possible, e.g., Novell IPX, AppleTalk
 - used to demultiplex up at receiver
- **CRC:** cyclic redundancy check at receiver
 - error detected: frame is dropped

Link Layer: 6-57

56

57

Ethernet: unreliable, connectionless

- **connectionless:** no handshaking between sending and receiving NICs
- **unreliable:** receiving NIC doesn't send ACKs or NAKs to sending NIC
 - data in dropped frames recovered only if initial sender uses higher layer rdt (e.g., TCP), otherwise dropped data lost
- Ethernet's MAC protocol: unslotted **CSMA/CD with binary backoff**

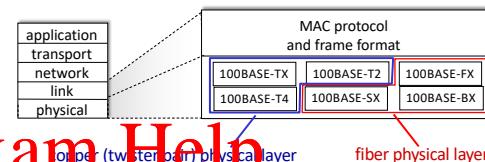
58

Assignment Project Exam Help

Link Layer: 6-58

802.3 Ethernet standards: link & physical layers

- **many** different Ethernet standards
 - common MAC protocol and frame format
 - different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10 Gbps, 40 Gbps
 - different physical layer media: fiber, cable



Link Layer: 6-59

59

<https://powcoder.com>

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- **LANs**
 - addressing, ARP
 - Ethernet
 - **switches**
 - VLANs
- link virtualization: MPLS
- data center networking



- a day in the life of a web request

Link Layer: 6-60

60

Add WeChat powcoder

Ethernet switch

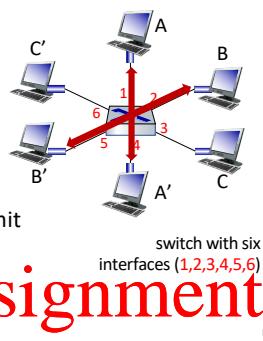
- Switch is a **link-layer** device: takes an **active** role
 - store, forward Ethernet frames
 - examine incoming frame's MAC address, **selectively** forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- **transparent:** hosts *unaware* of presence of switches
- **plug-and-play, self-learning**
 - switches do not need to be configured

Link Layer: 6-61

61

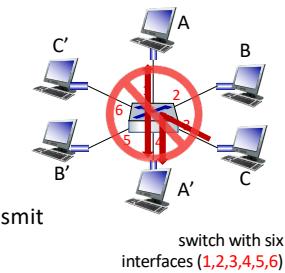
Switch: multiple simultaneous transmissions

- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
 - no collisions; full duplex
 - each link is its own collision domain
- switching:** A-to-A' and B-to-B' can transmit simultaneously, without collisions



Switch: multiple simultaneous transmissions

- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
 - no collisions; full duplex
 - each link is its own collision domain
- switching:** A-to-A' and B-to-B' can transmit simultaneously, without collisions
but A-to-A' and C to A' can *not* happen simultaneously



62

63

Assignment Project Exam Help

<https://powcoder.com>

Switch forwarding table

Q: how does switch know A' reachable via interface 4, B' reachable via interface 5?

A: each switch has a **switch table**, each entry:

- (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!

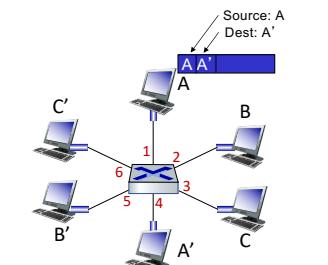
Q: how are entries created, maintained in switch table?

- something like a routing protocol?

Add WeChat powcoder

Switch: self-learning

- switch **learns** which hosts can be reached through which interfaces
 - when frame received, switch “learns” location of sender: incoming LAN segment
 - records sender/location pair in switch table



MAC addr	interface	TTL
A	1	60

Switch table (initially empty)

Link Layer: 6-65

64

65

Switch: frame filtering/forwarding

when frame received at switch:

1. record incoming link, MAC address of sending host
2. index switch table using MAC destination address
3. if entry found for destination
 - then {
 - if destination on segment from which frame arrived
 - then drop frame
 - else forward frame on interface indicated by entry
- else flood /* forward on all interfaces except arriving interface */

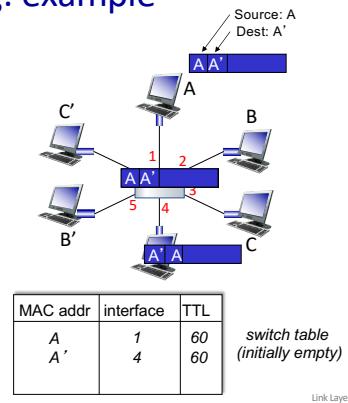
Assignment Project Exam Help

Link Layer: 6-66

66

Self-learning, forwarding: example

- frame destination, A', location unknown: **flood**
- destination A location known: **selectively send on just one link**



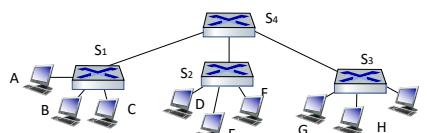
Link Layer: 6-67

67

<https://powcoder.com>

Interconnecting switches

self-learning switches can be connected together:



Q: sending from A to G - how does S₁ know to forward frame destined to G via S₄ and S₃?

- **A:** self learning! (works exactly the same as in single-switch case!)

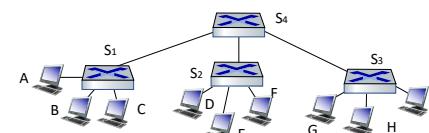
Link Layer: 6-68

68

Add WeChat powcoder

Self-learning multi-switch example

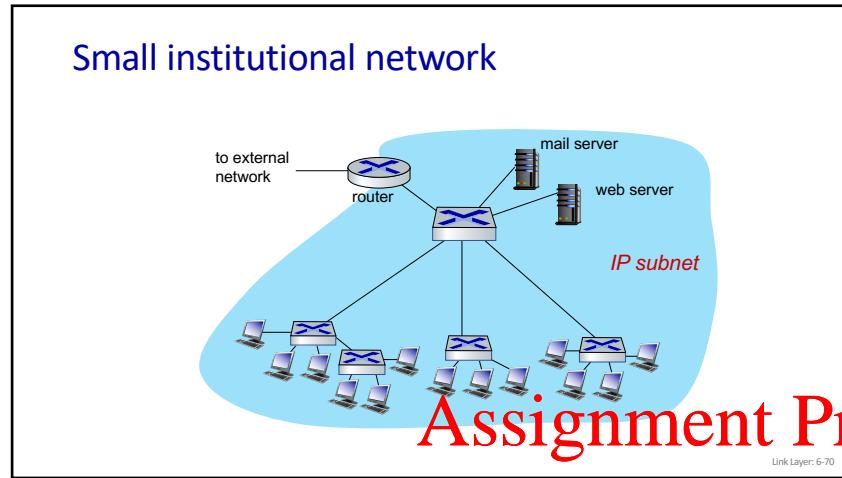
Suppose C sends frame to I, I responds to C



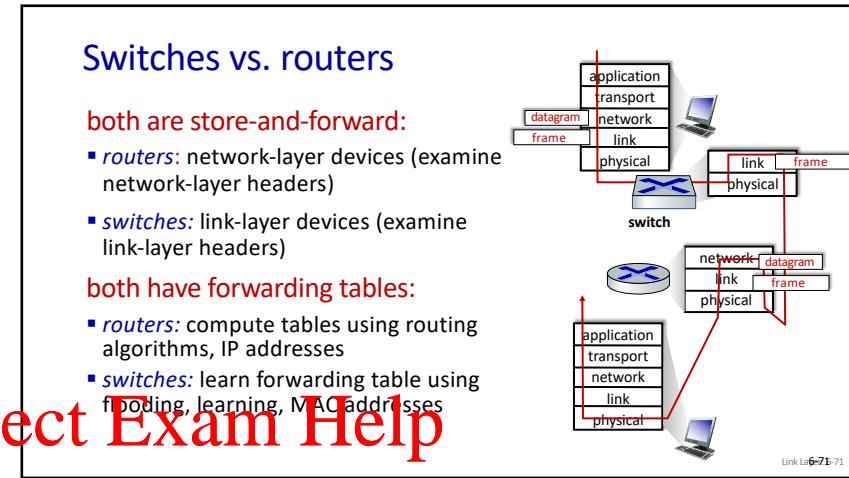
Q: show switch tables and packet forwarding in S₁, S₂, S₃, S₄

Link Layer: 6-69

69

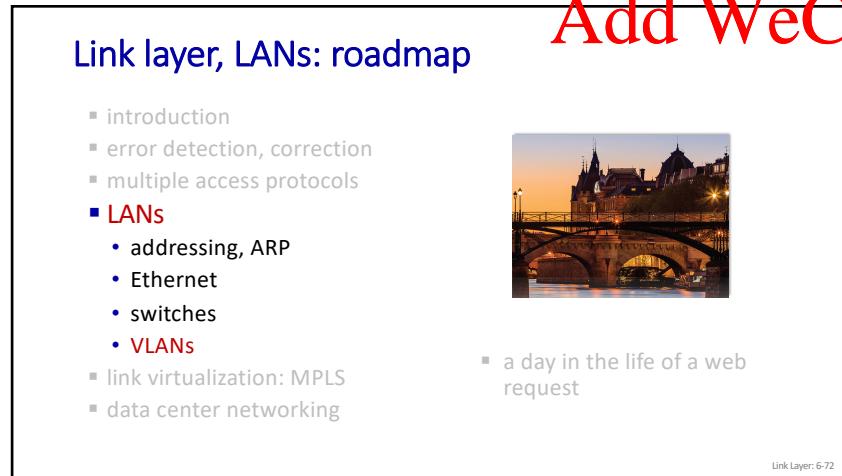


70

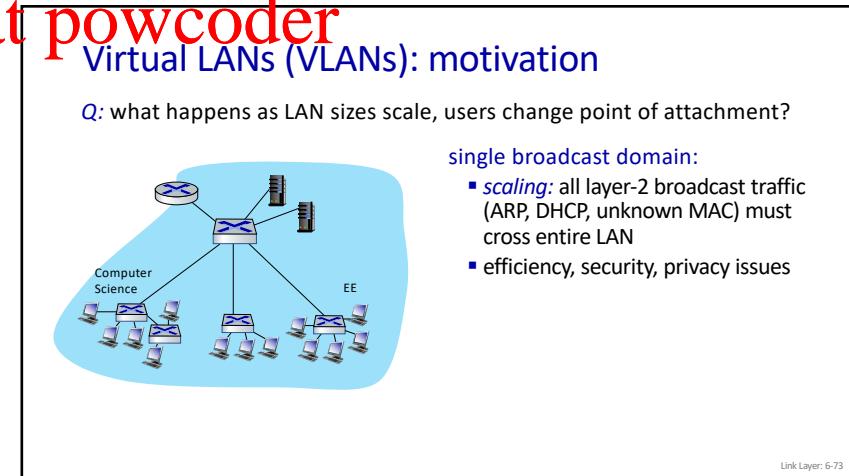


71

<https://powcoder.com>



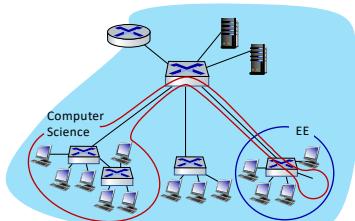
72



73

Virtual LANs (VLANs): motivation

Q: what happens as LAN sizes scale, users change point of attachment?



single broadcast domain:

- **scaling:** all layer-2 broadcast traffic (ARP, DHCP, unknown MAC) must cross entire LAN
- efficiency, security, privacy, efficiency issues
- administrative issues:**
- CS user moves office to EE - *physically* attached to EE switch, but wants to remain *logically* attached to CS switch

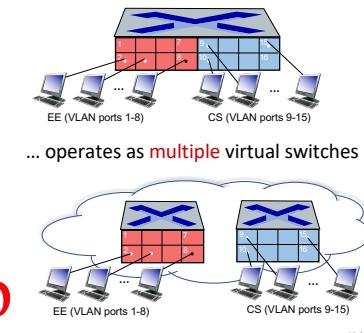
Assignment Project Exam Help

Port-based VLANs

Virtual Local Area Network (VLAN)

switch(es) supporting VLAN capabilities can be configured to define multiple *virtual LANS* over single physical LAN infrastructure.

port-based VLAN: switch ports grouped (by switch management software) so that *single* physical switch

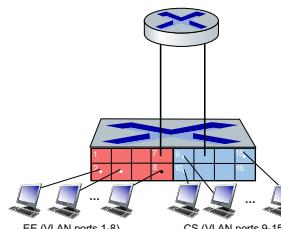


75

<https://powcoder.com>

Port-based VLANs

- **traffic isolation:** frames to/from ports 1-8 can *only* reach ports 1-8
 - can also define VLAN based on MAC addresses of endpoints, rather than switch port
- **dynamic membership:** ports can be dynamically assigned among VLANs
- **forwarding between VLANs:** done via routing (just as with separate switches)
 - in practice vendors sell combined switches plus routers

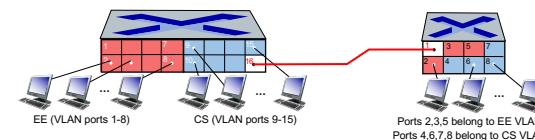


Link Layer: 6-76

76

Add WeChat powcoder

VLANs spanning multiple switches



Ports 2,3,5 belong to EE VLAN
Ports 4,6,7,8 belong to CS VLAN

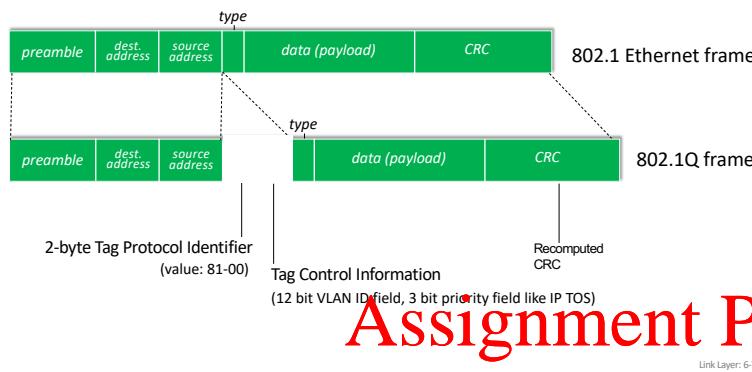
trunk port: carries frames between VLANs defined over multiple physical switches

- frames forwarded within VLAN between switches can't be vanilla 802.1 frames (must carry VLAN ID info)
- 802.1q protocol adds/removed additional header fields for frames forwarded between trunk ports

Link Layer: 6-77

77

802.1Q VLAN frame format



78

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
 - addressing, ARP
 - Ethernet
 - switches
 - VLANs
- link virtualization: MPLS
- a day in the life of a web request



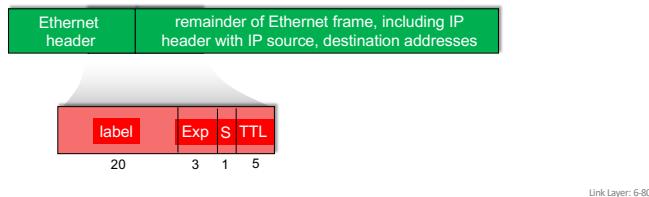
Link Layer: 6-79

79

<https://powcoder.com>

Multiprotocol label switching (MPLS)

- **goal:** high-speed IP forwarding among network of MPLS-capable routers, using fixed length label (instead of shortest prefix matching)
 - faster lookup using fixed length identifier
 - borrowing ideas from Virtual Circuit (VC) approach
 - but IP datagram still keeps IP address!



80

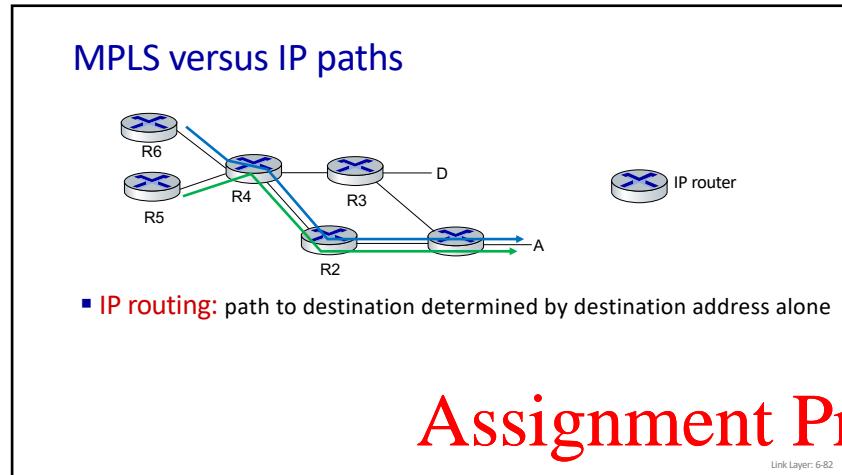
Add WeChat powcoder

MPLS capable routers

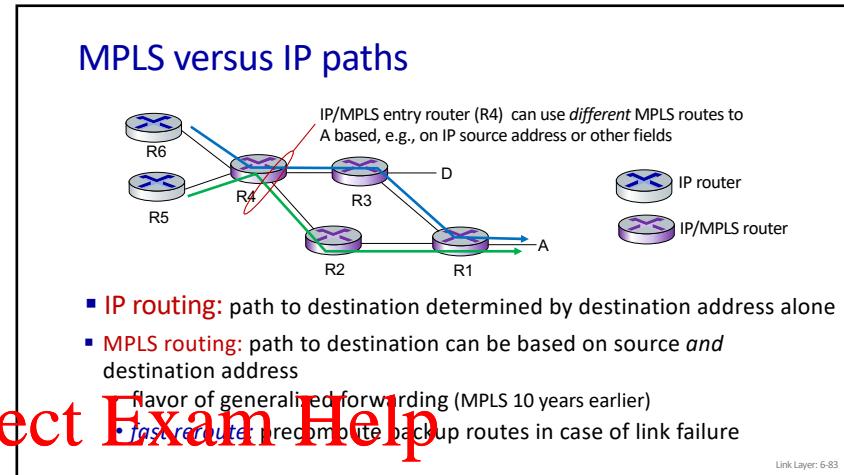
- a.k.a. label-switched router
- forward packets to outgoing interface based only on label value (*don't inspect IP address*)
 - MPLS forwarding table distinct from IP forwarding tables
- **flexibility:** MPLS forwarding decisions can *differ* from those of IP
 - use destination *and* source addresses to route flows to same destination differently (traffic engineering)
 - re-route flows quickly if link fails: pre-computed backup paths

Link Layer: 6-81

81

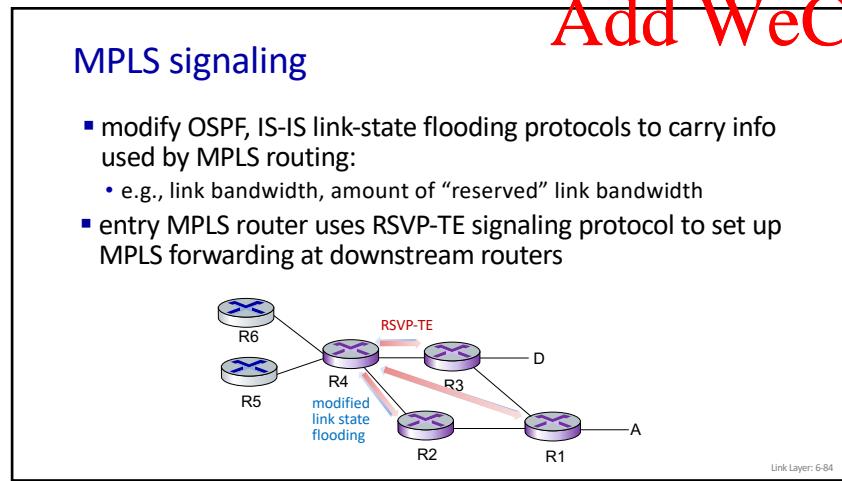


82

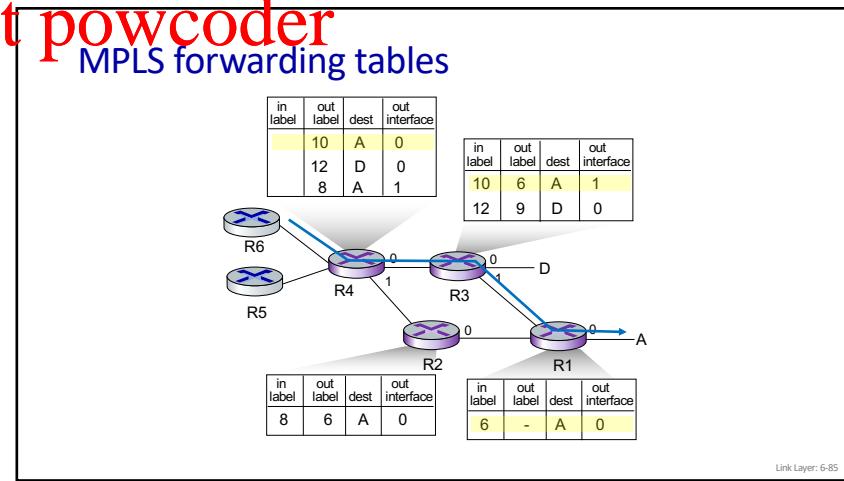


83

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



84



85

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
 - addressing, ARP
 - Ethernet
 - switches
 - VLANs
- link virtualization: MPLS
- data center networking



- a day in the life of a web request

Link Layer: 6-86

86

Datacenter networks

10's to 100's of thousands of hosts, often closely coupled, in close proximity:

- e-business (e.g. Amazon)
- content-servers (e.g., YouTube, Akamai, Apple, Microsoft)
- search engines, data mining (e.g., Google)

challenges:

- multiple applications, each serving massive numbers of clients
- reliability
- managing/balancing load, avoiding processing/networking/data bottlenecks



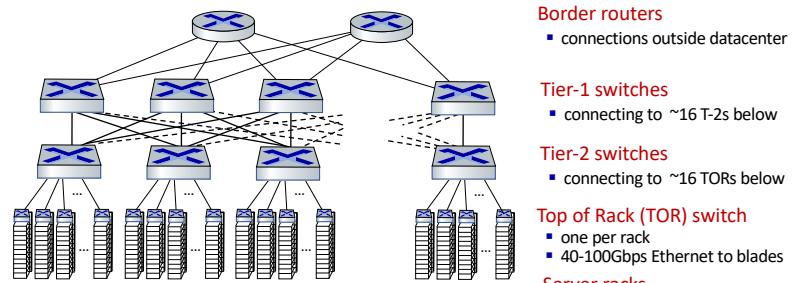
Inside a 40-ft Microsoft container, Chicago data center

Link Layer: 6-87

87

<https://powcoder.com>

Datacenter networks: network elements



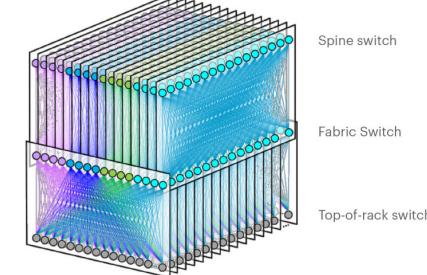
- Border routers**
 - connections outside datacenter
- Tier-1 switches**
 - connecting to ~16 T-2s below
- Tier-2 switches**
 - connecting to ~16 TORs below
- Top of Rack (TOR) switch**
 - one per rack
 - 40-100Gbps Ethernet to blades
- Server racks**
 - 20- 40 server blades: hosts

Link Layer: 6-88

88

Datacenter networks: network elements

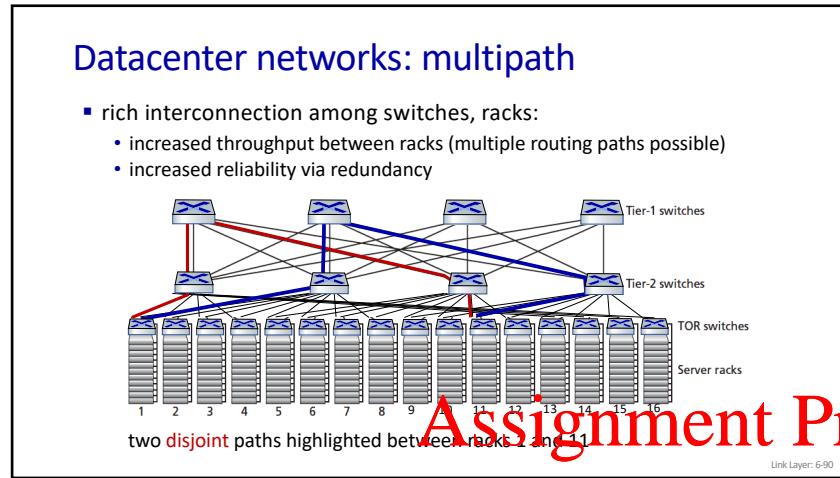
Facebook F16 data center network topology:



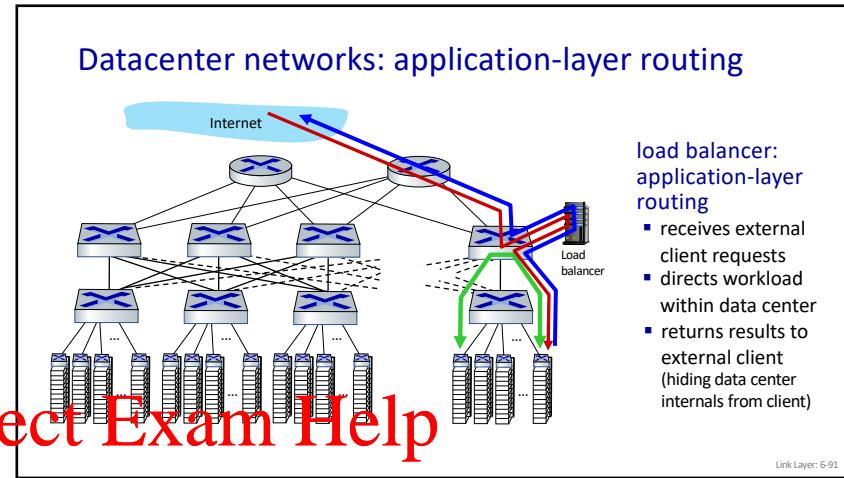
<https://engineering.fb.com/data-center-engineering/f16-minipack/> (posted 3/2019)

Link Layer: 6-89

89

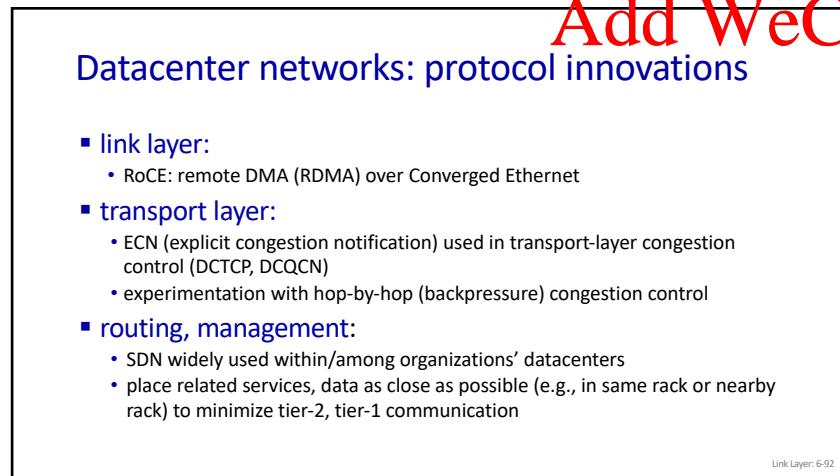


90

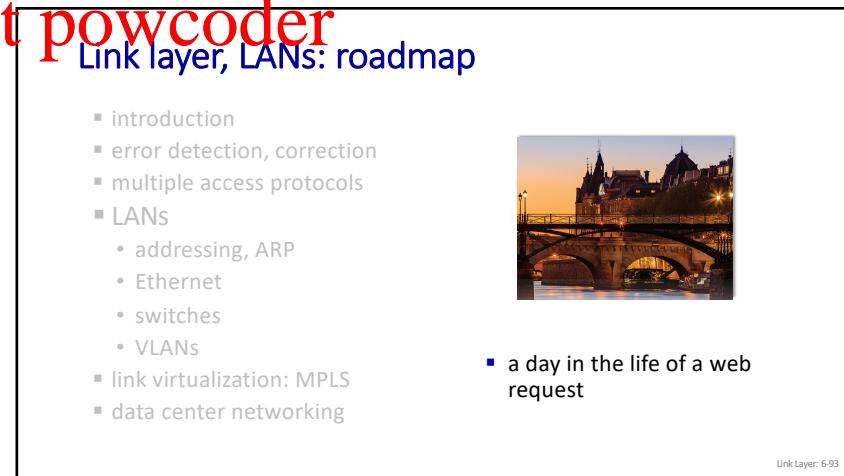


91

<https://powcoder.com>



92



93

Synthesis: a day in the life of a web request

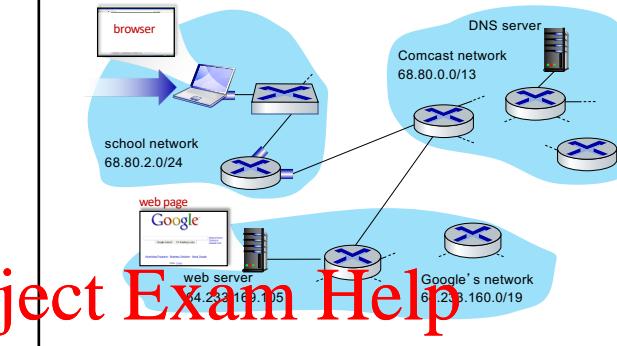
- our journey down the protocol stack is now complete!
 - application, transport, network, link
- putting-it-all-together: synthesis!
 - **goal:** identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
 - **scenario:** student attaches laptop to campus network, requests/receives www.google.com

Assignment Project Exam Help

Link Layer: 6-94

94

A day in the life: scenario



scenario:

- arriving mobile client attaches to network ...
- requests web page: www.google.com

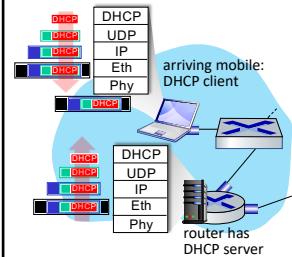
Sounds simple!

Link Layer: 6-95

95

<https://powcoder.com>

A day in the life: connecting to the Internet

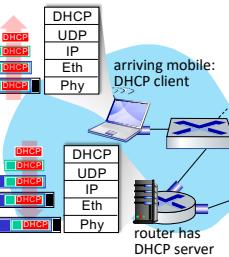


- connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use **DHCP**
- DHCP request **encapsulated in UDP**, encapsulated in **IP**, encapsulated in **802.3 Ethernet**
- Ethernet frame **broadcast** (dest: FFFFFFFFFFFF) on LAN, received at router running **DHCP server**
- Ethernet **demuxed** to IP demuxed, UDP demuxed to DHCP

Link Layer: 6-96

96

A day in the life: connecting to the Internet

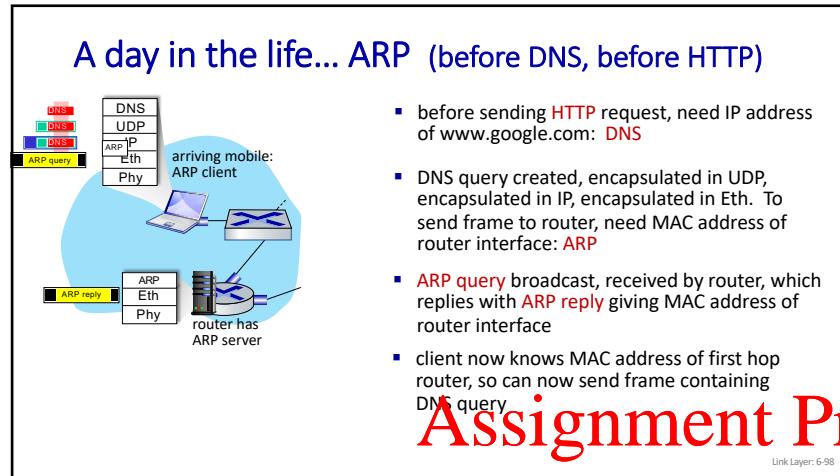


- DHCP server formulates **DHCP ACK** containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulation at DHCP server, frame forwarded (**switch learning**) through LAN, demultiplexing at client
- DHCP client receives DHCP ACK reply

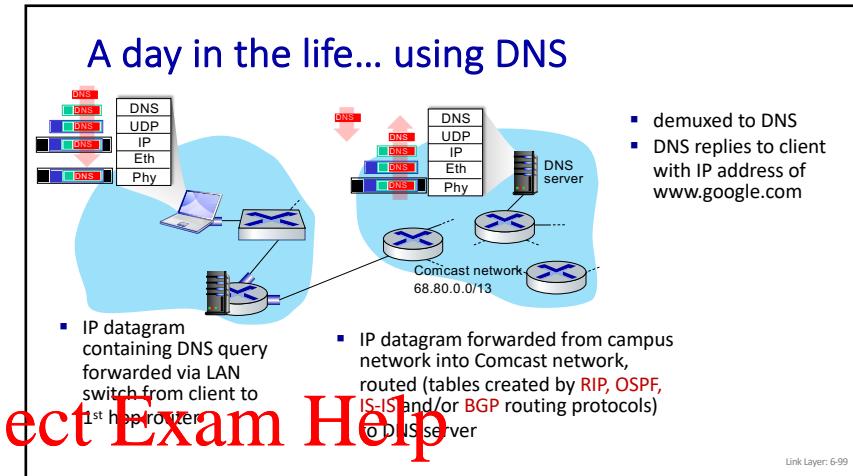
Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router

Link Layer: 6-97

97



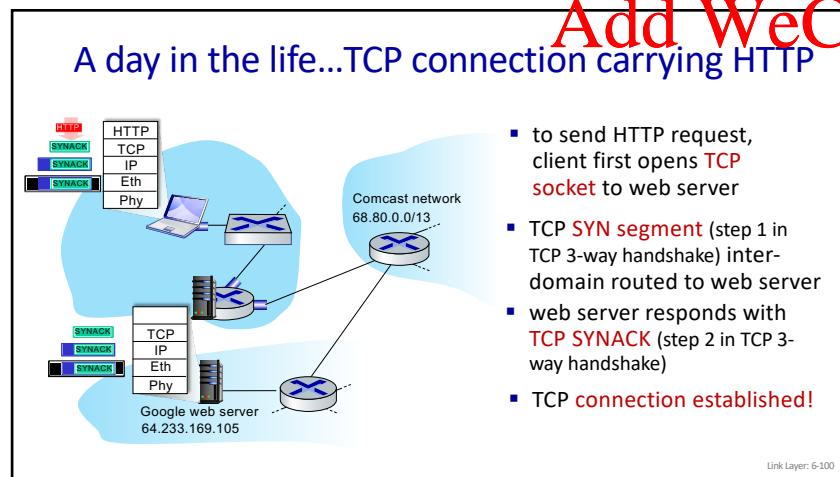
98



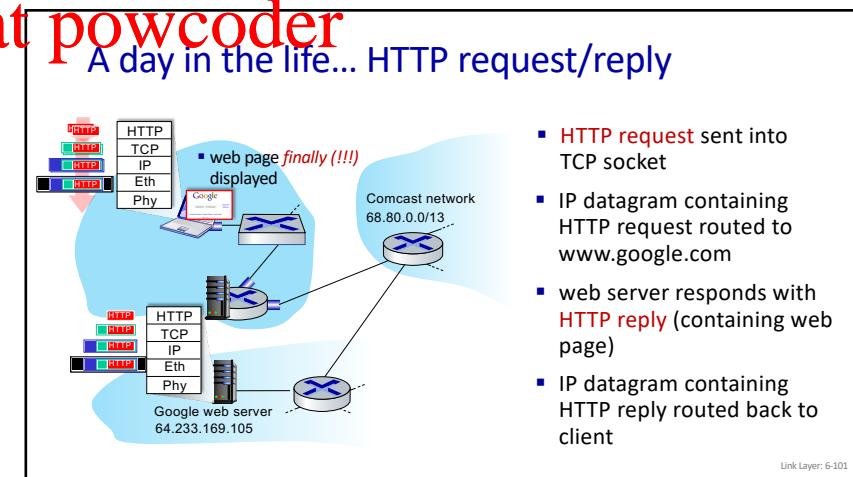
99

Assignment Project Exam Help

<https://powcoder.com>



100



101

Chapter 6: Summary

- principles behind data link layer services:
 - error detection, correction
 - sharing a broadcast channel: multiple access
 - link layer addressing
- instantiation, implementation of various link layer technologies
 - Ethernet
 - switched LANs, VLANs
 - virtualized networks as a link layer: MPLS
- synthesis: a day in the life of a web request

102

Link Layer: 6-102

Chapter 6: let's take a breath

- journey down protocol stack *complete* (except PHY)
- solid understanding of networking principles, practice!
- could stop here but *more* interesting topics!
 - wireless
 - security

103

Link Layer: 6-103

Assignment Project Exam Help

<https://powcoder.com>

104

Network Layer: 5-104

[Additional Chapter 6 slides](https://powcoder.com)

105

Link Layer: 6-105

Add WeChat powcoder

Pure ALOHA efficiency

$$\begin{aligned}
 P(\text{success by given node}) &= P(\text{node transmits}) * \\
 &\quad P(\text{no other node transmits in } [t_0-1, t_0] *) \\
 &\quad P(\text{no other node transmits in } [t_0-1, t_0]) \\
 &= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1} \\
 &= p \cdot (1-p)^{2(N-1)} \\
 \dots \text{ choosing optimum } p \text{ and then letting } n & \\
 &= 1/(2e) = .18 \rightarrow \infty
 \end{aligned}$$

even worse than slotted Aloha!