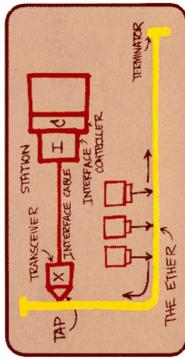


Link Layer

Ethernet

- 5.1 Introduction and services
- 5.6 Link-layer switches
- 5.7 PPP
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet

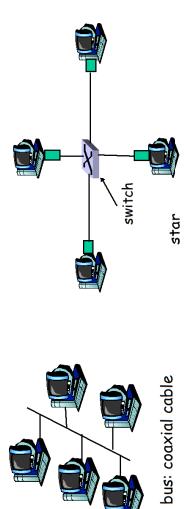
- “dominant” wired LAN technology:
- cheap \$20 for NIC
- first widely used LAN technology
- Simpler and cheaper
- kept up with speed race: 10 Mbps - 10 Gbps



5: DataLink Layer 5:1

Star topology

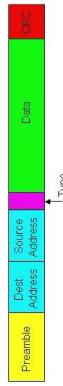
- bus topology popular through mid 90s
 - all nodes in same collision domain (can collide with each other)
- today: star topology prevails
 - active *switch* in center



5: DataLink Layer 5:3

Ethernet Frame Structure

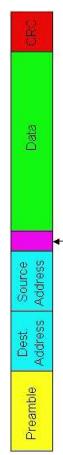
- Sending adapter encapsulates IP datagram (or other network layer protocol packet) in **Ethernet frame**



5: DataLink Layer 5:2

Ethernet Frame Structure (more)

- **Addresses:** 6 bytes
 - if adapter receives frame with matching destination address, or with broadcast address (eg 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)
- **CRC:** checked at receiver, if error is detected, frame is dropped



5: DataLink Layer 5:5

Ethernet: Unreliable, connectionless

- **connectionless:** **No handshaking** between sending and receiving NICs
- **unreliable:** receiving NIC doesn't send acks or nacks to sending NIC
 - stream of datagrams passed to network layer can have gaps (missing datagrams)
 - gaps will be filled if app is using TCP
- **Reliability by TCP responsible**
- otherwise, app will see gaps
- Ethernet's MAC protocol: unslotted CSMA/CD

5: DataLink Layer 5:6

Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission. If NIC senses channel busy, waits until channel idle, then transmits
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame!
4. If **NIC detects another transmission while transmitting, aborts and sends jam signal**
 - **Goal:** adapt retransmission attempts to estimated current load
 - **If heavy load: random wait will be longer**
 - first collision: choose K from {0,1}; delay is K·512 bit transmission times
 - after second collision: choose K from {0,1,2,3}...
 - after ten collisions, choose K from {0,1,2,3,4,...,1023}

**类似于全双工：一边发送，一边可以侦听
退避机制**

Ethernet's CSMA/CD (more)

- 1. **Jam Signal:** make sure all other transmitters are aware of collision: 48 bits
- 2. Bit time: 1 microsec for 10 Mbps Ethernet ; for K=1023, wait time is about 50 msec
- 3. Ethernet's MAC protocol: unslotted CSMA/CD
- 4. **Exponential Backoff:**
 - **Goal:** adapt retransmission attempts to estimated current load
 - **If heavy load: random wait will be longer**
 - first collision: choose K from {0,1}; delay is K·512 bit transmission times
 - after second collision: choose K from {0,1,2,3}...
 - after ten collisions, choose K from {0,1,2,3,4,...,1023}

5: DataLink Layer 5:7

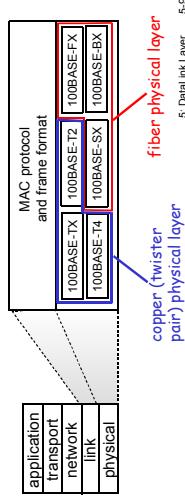
5: DataLink Layer 5:8

802.3 Ethernet Standards: Link & Physical Layers

Link Layer

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

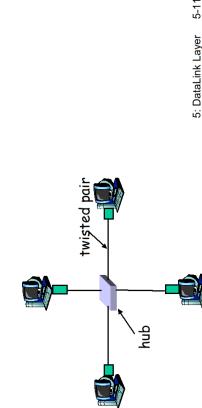
Common link layer, different PHY layers



Hubs

... physical-layer ("dumb") repeaters (**简单转发**):

- bits coming in one link go out **all** other links at same rate
- all nodes connected to hub can collide with one another
- no frame buffering
- no CSMA/CD at hub: host NICs detect collisions



Switch: allows *multiple simultaneous* transmissions

- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on **each** incoming link, but no collisions: full duplex
 - **each link is its own domain**
- **switching:** A-to-A' and B-to-B simultaneously, without collisions
 - not possible with dumb hub

Switch (选择性转发, 基于 MAC 地址, 和路由器区别)

- **link-layer device: smarter than hubs, take 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 are unaware of presence of switches
 - **plug-and-play, self-learning**
 - switches do not need to be configured

5: DataLink Layer 5-12

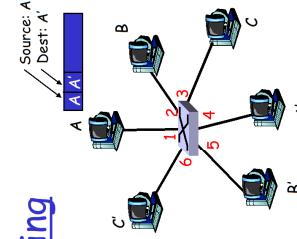
Switch Table 记录其 MAC 地址和到达的接口

- **Q:** how does switch know that 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?
- **switch with six interfaces (1,2,3,4,5,6)**

5: DataLink Layer 5-14

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	Switch table (initially empty)
A	1	60	

如果表里没有 **dest** 接口, 则群发
forward on all but the interface on which the frame arrived

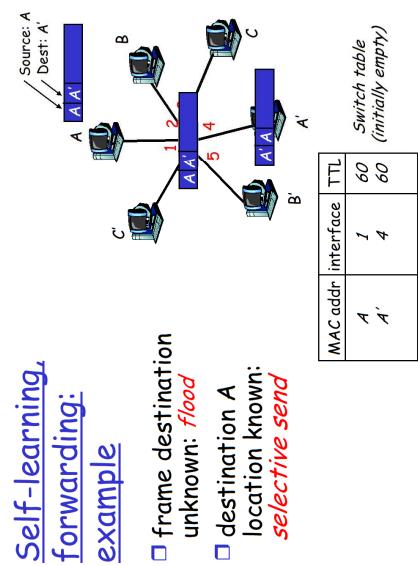
5: DataLink Layer 5-16

Switch: frame filtering/forwarding

- When frame received:
 1. record link associated with sending host
 2. index switch table using MAC dest address
 3. if entry found for destination
 - if dest on segment from which frame arrived
 - then drop the frame
 - else forward the frame on interface indicated
 - else **forward**

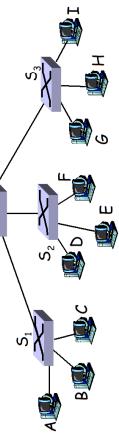
5: DataLink Layer 5-14

Self-learning, forwarding example



Interconnecting switches

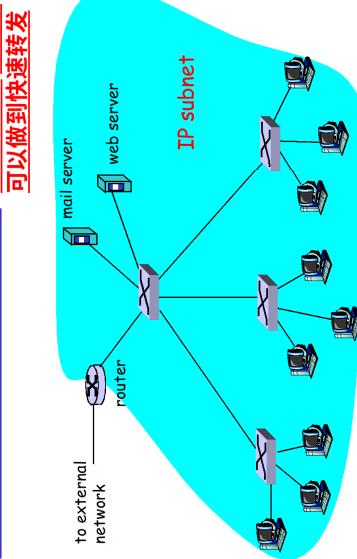
- switches can be connected together



- 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!)

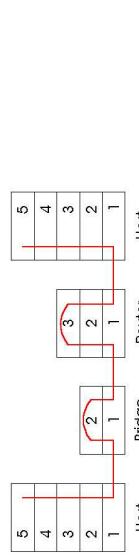
5: DataLink Layer 5:18

Institutional network



Switches vs. Routers

- both **store-and-forward** devices
 - routers: network layer devices (examine network layer headers)
 - switches are link layer devices
- routers maintain routing tables, implement **routing** algorithms
- switches maintain switch tables, implement **filtering, learning** algorithms



Link Layer

- 5.1 Introduction and services
- 5.6 Hubs and switches
- 5.7 PPP
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet

5: DataLink Layer 5:21

Point to Point Data Link Control

- one sender, one receiver, **one link: easier than broadcast link:**
 - no Media Access Control
 - no need for explicit MAC addressing
 - e.g., **dialup** link, ISDN line
- popular point-to-point DLC protocols:
 - PPP (point-to-point protocol)
 - HDLC: High level data link control (Data link used to be considered "high layer" in protocol stack!)

例: **拨号上网 link**

5: DataLink Layer 5:22

PPP Design Requirements [RFC 1557]

- **packet framing:** encapsulation of network-layer datagram in data link frame
 - carry network layer data of **any network layer** protocol (**not just IP, at same time**)
- **bit transparency:** must carry **any bit pattern in the data field**
 - error detection (no correction)
 - connection liveness: detect, signal link failure to network layer
 - **network layer address negotiation:** endpoint can learn/configure each other's network address
- Error recovery, flow control, data re-ordering
 - all relegated to higher layers!

5: DataLink Layer 5:23

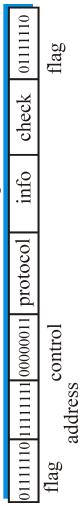
5: DataLink Layer 5:24

PPP Data Frame

PPP Data Frame

- **Flag:** delimiter (framing)
 - **Address:** does nothing (only one option)
 - **Control:** does nothing; in the future possible multiple control fields
 - **Protocol:** upper layer protocol to which frame delivered (eg, PPP-LCP, IP, IPCP, etc)
- | | | | | | | |
|------|---------|---------|----------|-----------------|--------|----------|
| 1 | 1 | 1 | 1 or 2 | variable length | 2 or 4 | 1 |
| flag | address | control | protocol | info | check | 01111110 |

5: DataLink Layer 5:25



5: DataLink Layer 5:26

Chapter 5: Link layer

- principles behind data link layer services:
 - error detection, correction
 - sharing a broadcast channel: multiple access
 - link layer addressing
- instantiation and implementation of various link layer technologies
 - Ethernet
 - switched LANs
 - PPP

5: DataLink Layer 5:27

Chapter 5: let's take a breath

- journey down protocol stack **complete**
- solid understanding of networking principles, practice
- could stop here but **lots** of interesting topics!
 - wireless
 - multimedia
 - security
 - network management

5: DataLink Layer 5:28

Chapter 5: Link layer

our goals:

- understand principles behind link layer services:
 - error detection, correction
 - sharing a broadcast channel: multiple access
 - link layer addressing
 - local area networks: Ethernet
- instantiation, implementation of various link layer technologies

Link layer: introduction

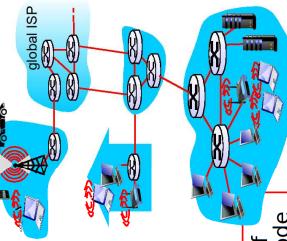
terminology:

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

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

Link Layer 5:1

5: DataLink Layer 5:2



Link layer services

(more)

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

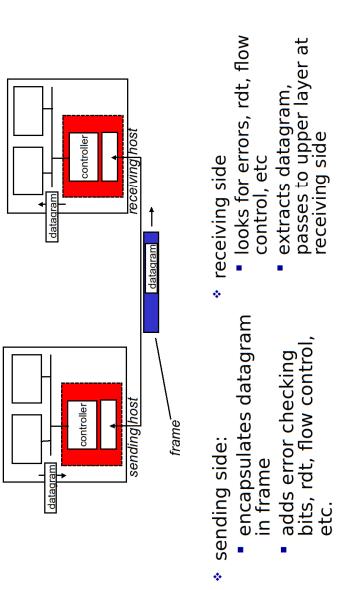
Link Layer 5:3

Link Layer 5:4

Where is the link layer implemented?

- in each and every host
- link layer implemented in "adapter" (aka **network interface card** NIC) or on a chip
 - Ethernet card, 802.11 card; Ethernet chipset
 - implements link, physical layer
 - attaches into host's system buses
 - combination of hardware, software, firmware

Adapters communicating



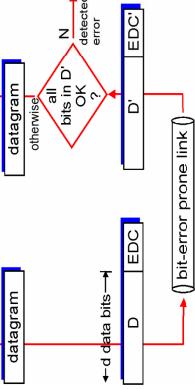
Error detection

EDC = Error Detection and Correction bits (redundancy)

D = Data protected by error checking, may include header fields

Error detection not 100% reliable!

- protocol may miss some errors, but rarely
- larger EDC field yields better detection and correction

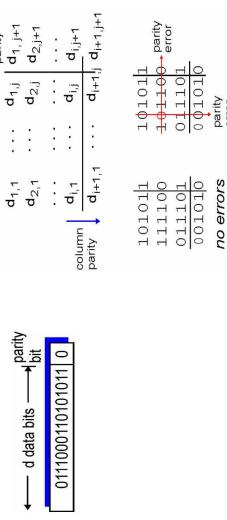


Internet checksum (review)

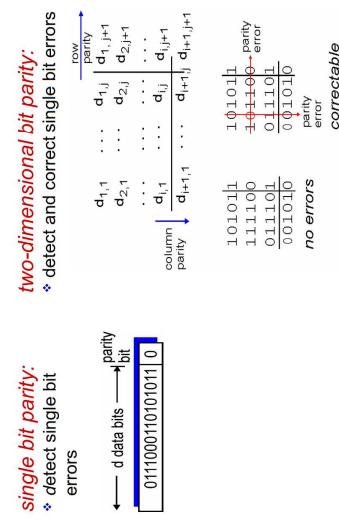
- goal:** detect "errors" (e.g., flipped bits) in transmitted packet (note: used at transport layer only)
- receiver:**
- compute checksum of received segment
 - check if computed checksum equals checksum field value:
 - NO - error detected
 - YES - no error detected. *But maybe errors nonetheless?*

Link Layer 5-9

- single bit parity:**
- detect single bit errors
 - two-dimensional bit parity:
 - detect and correct single bit errors



Parity checking



Cyclic redundancy check

- more powerful error-detection coding
- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that $\langle D, R \rangle$ exactly divisible by G (modulo 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, ATM)

Link Layer 5-10

$D * 2^r \text{ XOR } R$ mathematical formula

Link Layer 5-10

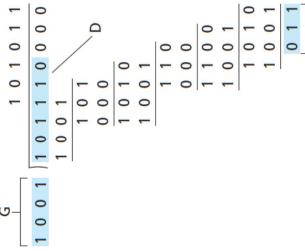
CRC example

want:
 $D2^r \text{ XOR } R = nG$

equivalently:
 $D2^r = nG \text{ XOR } R$

equivalently:
if we divide $D2^r$ by G, want remainder R, to satisfy:

$$R = \text{remainder} \left(\frac{D2^r}{G} \right)$$



Link Layer 5-11

Multiple access links, protocols

two types of "links":

- point-to-point
 - PPP for dial-up access
 - point-to-point link between Ethernet switch, host
- broadcast (shared wire or medium)
 - old-fashioned Ethernet
 - upstream HFC
 - 802.11 wireless LAN
 - shared wire (e.g., cabled Ethernet)
 - shared RF (e.g., 802.11 WiFi)
 - shared RF (satellite)
 - humans at a cocktail party (shared air, acoustical)

Link Layer 5-12

MAC protocols: taxonomy

Channel partitioning MAC protocols:

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

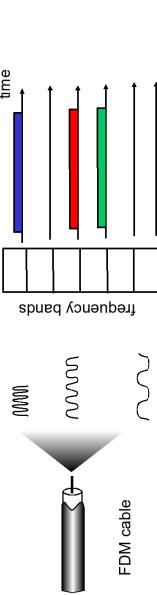
Link Layer 5-13

Channel partitioning MAC protocols: FDMA

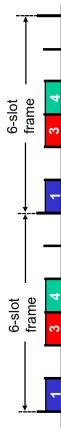
FDMA: frequency division multiple access

- ❖ 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 pkt, frequency bands 2,5,6 idle



Link Layer 5-15



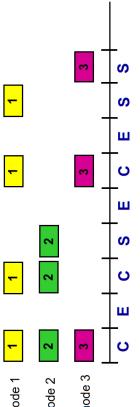
Link Layer 5-14

Random access protocols

- ❖ TDMA: time division multiple access
 - access to channel in "rounds"
 - each station gets fixed length slot (length = pkt trans time) in each round
 - unused slots go idle
 - example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle
- ❖ random access
 - 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:
 - slotted ALOHA
 - ALOHA
 - CSMA, CSMA/CD, CSMA/CA

Link Layer 5-16

Slotted ALOHA



Link Layer 5-17

Cons:

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

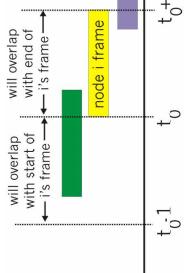
Pros:

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

Link Layer 5-18

Pure (unslotted) ALOHA

- ❖ unslotted Aloha: simpler, no synchronization
 - when frame first arrives
 - transmit immediately
 - collision probability increases:
 - frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$
 - will overlap with start of i 's frame
 - with end of i 's frame



Link Layer 5-19

CSMA (carrier sense multiple access)

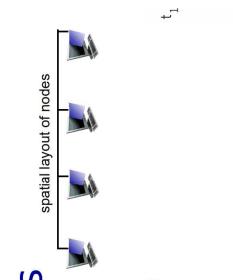
- ❖ CSMA: listen before transmit:
 - if channel sensed idle: transmit entire frame
 - if channel sensed busy, defer transmission

❖ human analogy: don't interrupt others!

CSMA collisions

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

Link Layer 5-19



Link Layer 5-20

CSMA/CD (collision detection)

"Taking turns" MAC protocols

CSMA/CD: carrier sensing, deferral as in CSMA

- collisions detected within short time
- colliding transmissions aborted, reducing channel wastage

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
- longer backoff interval with more collisions
- collision detection:

- easy in wired LANs: measure signal strengths, compare transmitted, received signals
- difficult in wireless LANs: received signal strength overwhelmed by local transmission strength

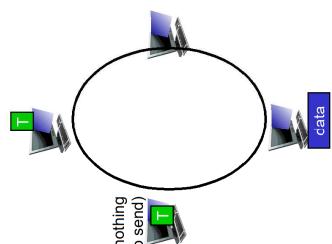
Link Layer 5:21

"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 5:23

Summary of MAC protocols

- channel partitioning**, by time, frequency or code
 - Time Division, Frequency Division
 - 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** (dynamic)
 - ALOHA, S-ALOHA, CSMA, CSMA/CD
 - carrier sensing; easy in some technologies (wire)
 - hard in others (wireless)
 - efficient at low load: single node can fully utilize channel
 - high load: collision overhead
 - taking turns**
 - polling from central site, token passing
 - bluetooth, FDDI, token ring

Link Layer 5:24

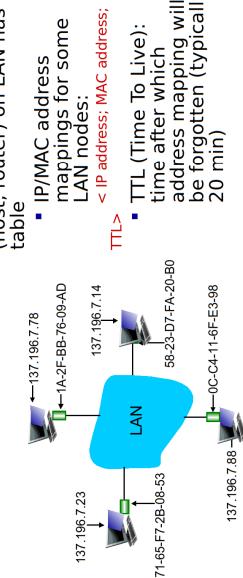
MAC addresses and ARP

- 32-bit IP address:
 - network-layer address for interface
 - used for layer 3 (network layer) forwarding
- MAC (or LAN or physical or Ethernet) address:
 - function: *used "locally" to get frame from **one** interface to another physically-connected interface (same network, in IP-addressing sense)*
 - 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable (hexadecimal base 16, octal 8, decimal 78, binary 9-AD (each nibble 4 bits))

Link Layer 5:24

ARP: address resolution protocol

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

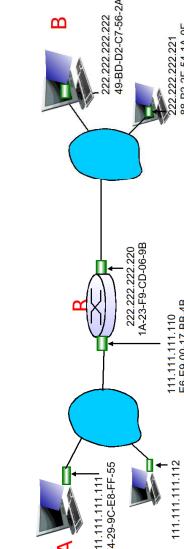


Link Layer 5:25

ARP protocol: same LAN

- A wants to send datagram to B
 - B's MAC address not in A's ARP table.
 - A **broadcasts** ARP query packet, containing B's IP address
 - dest MAC address = FF:FF:FF:FF:FF:FF
 - all nodes on LAN receive ARP query
 - B receives ARP packet, replies to A with its (B's) MAC address
 - frame sent to A's MAC address (unicast)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
 - soft state: information that times out (goes away) unless refreshed
 - ARP is "plug-and-play":
 - nodes create their ARP tables **without intervention from net administrator**

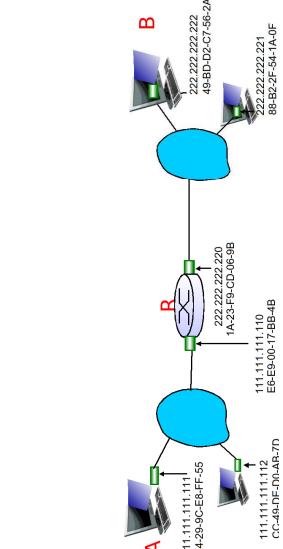
with rough end diagram from A to B via R
focus on addressing at IP (datagram) and MAC layer (frame)
assume A knows B's IP address.
< IP address: MAC address;
TTL > TTL (Time To Live):
time after which address mapping will be forgotten (typically 20 min)



Link Layer 5:27

Addressing: routing to another LAN

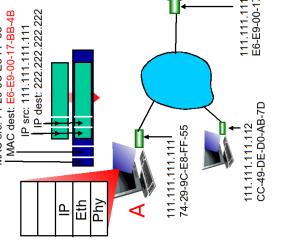
with rough end diagram from A to B via R
assume A knows B's IP address.
assume A knows IP address of first hop router, R (how?)
assume A knows R's MAC address (how?)



Link Layer 5:28

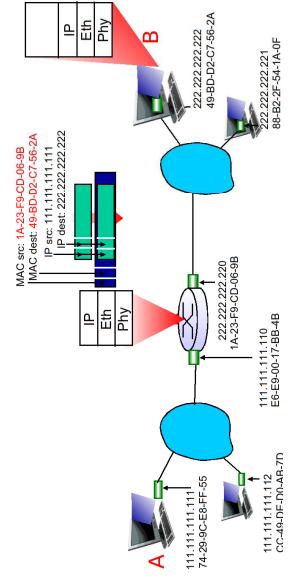
Addressing: routing to another LAN

- ❖ A creates IP datagram with IP source A, destination B
- ❖ R creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram



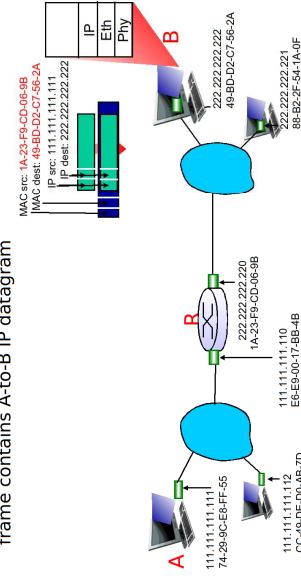
Addressing: routing to another LAN

- ❖ R forwards datagram with IP source A, destination B
- ❖ R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



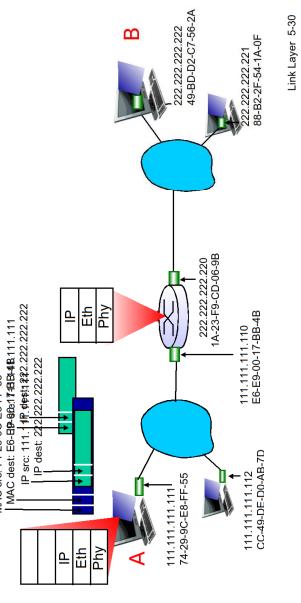
Addressing: routing to another LAN

- ❖ R forwards datagram with IP source A, destination B
- ❖ R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



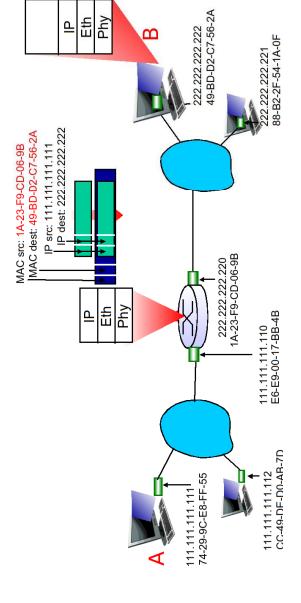
Addressing: routing to another LAN

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



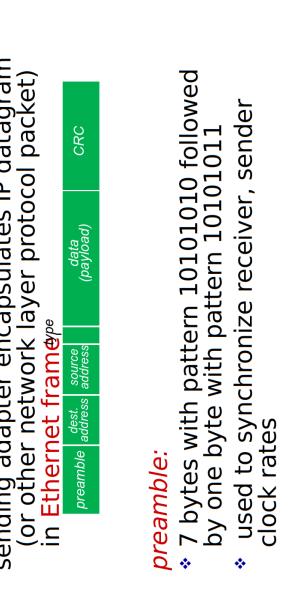
Addressing: routing to another LAN

- ❖ R forwards link-layer frame with IP source A, destination B
- ❖ R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



Addressing: routing to another LAN

- ❖ R forwards link-layer frame with IP source A, destination B
- ❖ R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



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)

- ❖ **CRC:** cyclic redundancy check at receiver
- ❖ error detected: frame is dropped



Link Layer 5:35

Ethernet: unreliable, connectionless

- ❖ **connectionless:** no handshaking between sending and receiving NICs
- ❖ **unreliable:** receiving NIC doesn't send acks or nacks to sending NIC
 - data in dropped frames recovered only if initial sender uses higher layer prot (e.g., TCP), otherwise dropped data lost

- ❖ Ethernet's MAC protocol: unslotted CSMA/CD with binary backoff

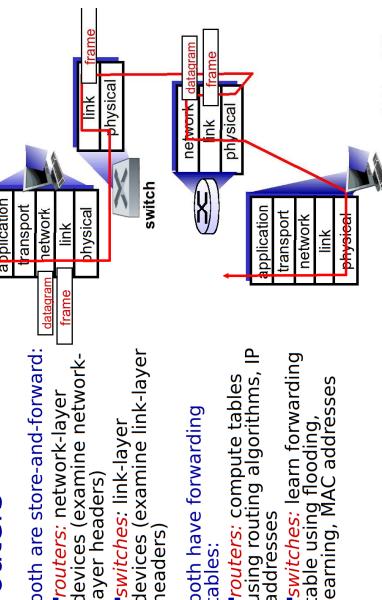
Link Layer 5:36

Ethernet switch

- ❖ **link-layer device:** takes an **active role**
 - store, forward Ethernet frames
 - 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 are unaware of presence of switches
 - **plug-and-play, self-learning**
 - switches do not need to be configured

Link Layer 5:37

Switches vs. routers



Data center networks

- ❖ 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

Link Layer 5:38

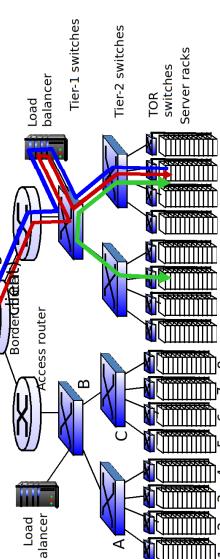


Link Layer 5:38

Data center networks

load balancer: application-layer routing

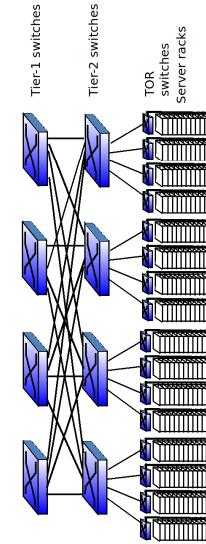
- receives external client requests
- directs workload within data center
- returns results to external client



Link Layer 5:41

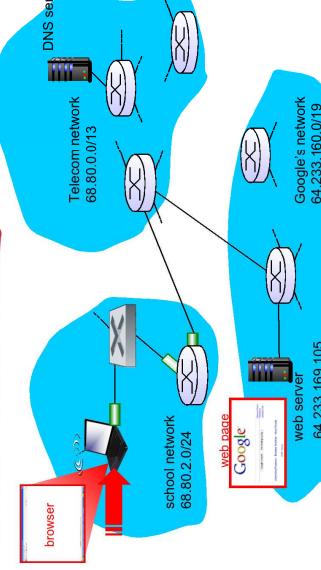
Data center networks

- ❖ rich interconnection among switches, racks:
 - increased throughput between racks (multiple routing paths possible)
 - increased reliability via redundancy



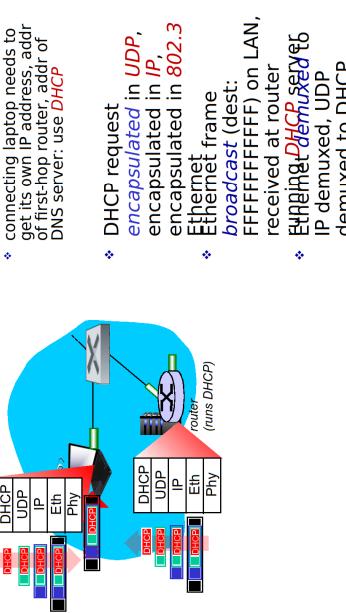
Link Layer 5:40

A day in the life: scenario



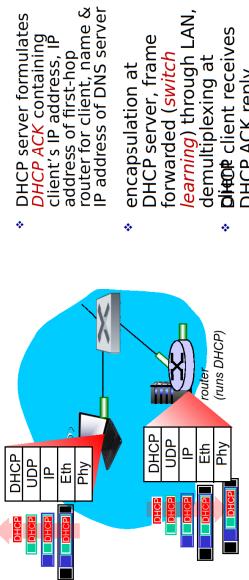
Link Layer 5:43

A day in the life... connecting to the internet

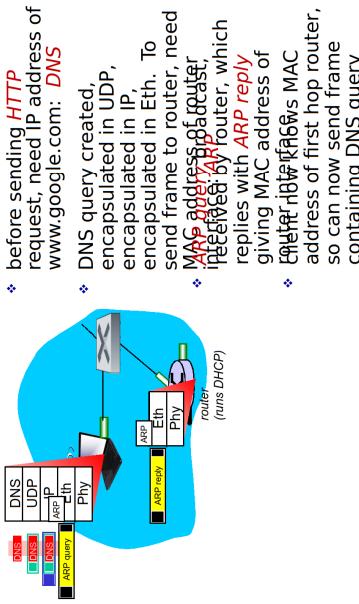


Link Layer 5:44

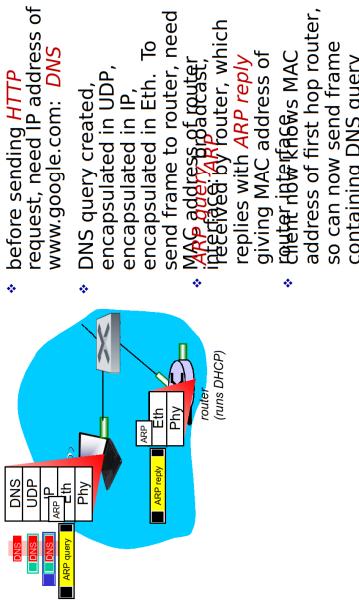
A day in the life... connecting to the Internet



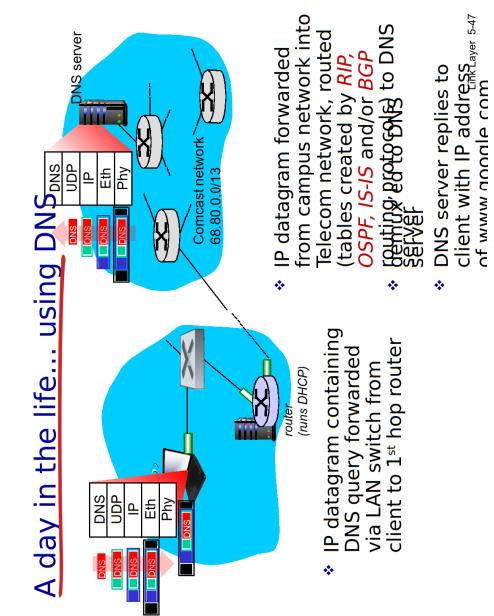
A day in the life... ARP (before HTTP)



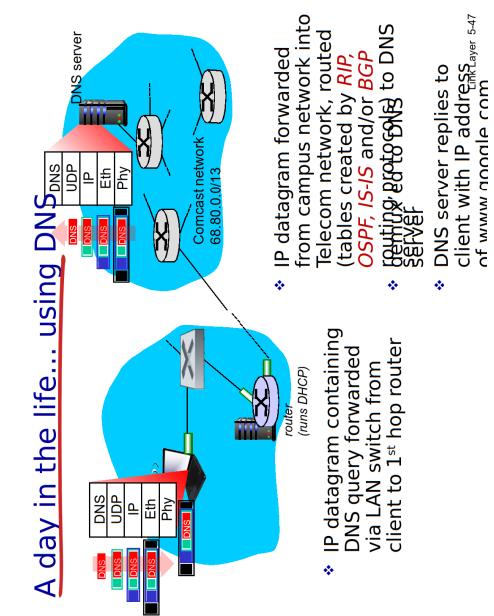
A day in the life... DNS (before HTTP)



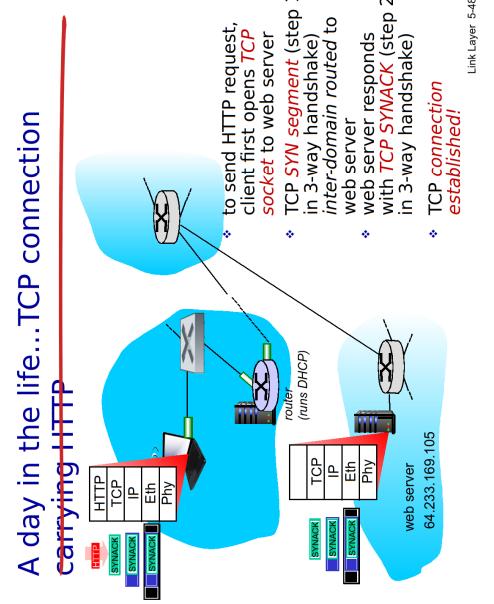
A day in the life... carrying HTTP



A day in the life... using DNS



A day in the life... TCP connection



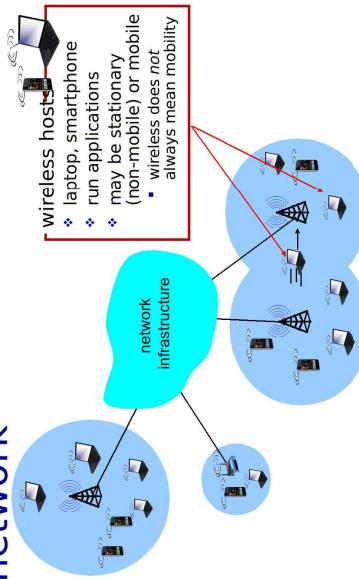
Ch. 6: Wireless and Mobile Networks

Background:

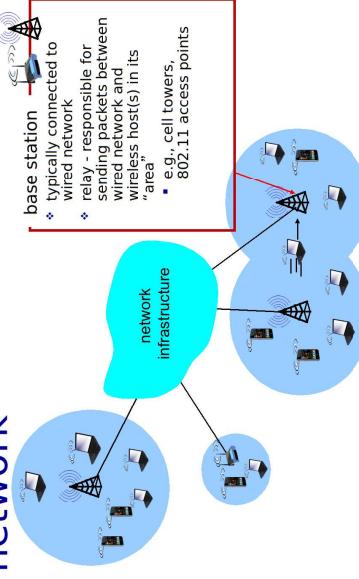
- # wireless (mobile) phone subscribers now exceeds # wired phone subscribers (5-to-1)!
- # wireless Internet-connected devices equals # wireless Internet-connected devices
 - laptops, Internet-enabled phones promise anytime untethered Internet access
- two important (but different) challenges
 - wireless:** communication over wireless link
 - mobility:** handling the mobile user who changes point of attachment to network

Wireless, Mobile Networks 6-1

Elements of a wireless network

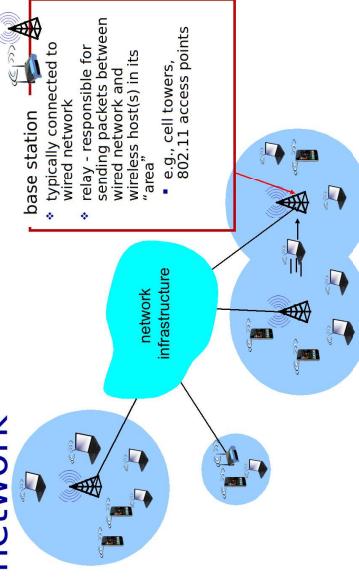


Elements of a wireless network



Wireless, Mobile Networks 6-2

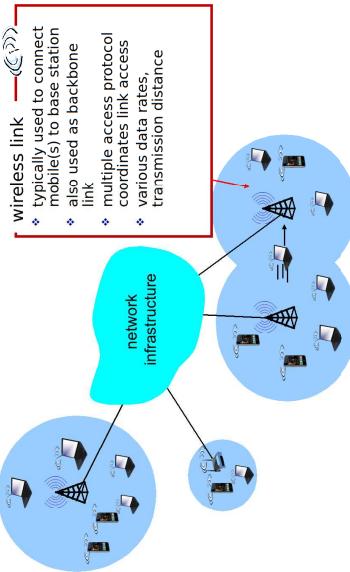
Elements of a wireless network



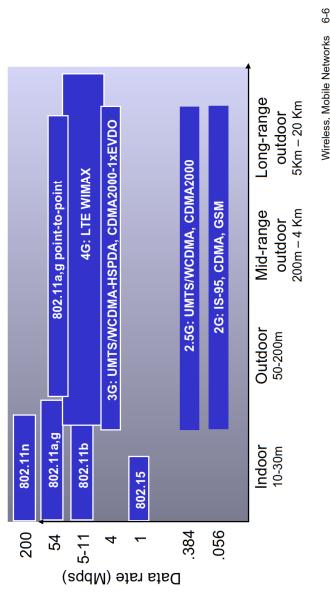
Wireless, Mobile Networks 6-3

Wireless, Mobile Networks 6-4

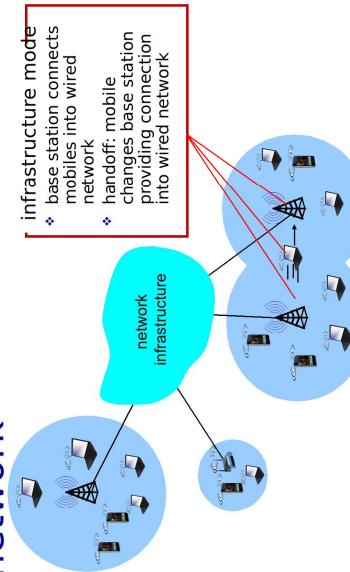
Elements of a wireless network



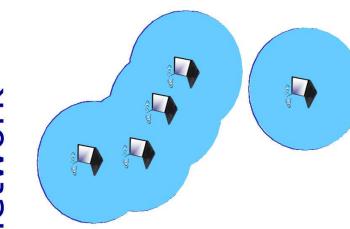
Characteristics of selected wireless links



Elements of a wireless network



Elements of a wireless network



Wireless Link Characteristics

important differences from wired link

- decreased signal strength: radio signal attenuates as it propagates through matter (path loss)
- interference from other sources: standardized wireless network frequencies (e.g., 2.4 GHz) shared by other devices (e.g., phone); devices (motors) interfere as well
- multipath propagation: radio signal reflects off objects/ground, arriving at destination at slightly different times

.... make communication across (even a point to point) wireless link much more "difficult"

Wireless, Mobile Networks 6-9

Wireless network characteristics

Multiple wireless senders and receivers create additional problems (beyond multiple access):



Hidden terminal problem

- B, A hear each other
- B, C hear each other
- A, C can not hear each other interfering at B

Wireless, Mobile Networks 6-10



Signal attenuation:

- B, A hear each other
- B, C hear each other
- A, C can not hear each other interfering at B

Code Division Multiple Access (CDMA)

- unique "code" assigned to each user;
i.e., code set partitioning
- all users share same frequency, but each user has own "chipping" sequence (i.e., code) to encode data
- allows multiple users to "coexist" and transmit simultaneously with minimal interference (if codes are "orthogonal")
- encoded signal = (original data) X (chipping sequence)
- decoding: inner-product of encoded signal and chipping sequence

Wireless, Mobile Networks 6-11

IEEE 802.11 Wireless LAN

- 802.11a
2.4-5 GHz unlicensed spectrum
up to 11 Mbps
- 802.11g
direct sequence spread spectrum (DSSS) in physical layer
all hosts use same chipping code
2.4-5 GHz range
up to 54 Mbps
- 802.11n: multiple antennae
2.4-5 GHz range
up to 200 Mbps

Wireless, Mobile Networks 6-12

- all use CSMA/CA for multiple access
- all have base-station and ad-hoc network versions

802.11: Channels, association

- 802.11b: 2.4GHz-2.485GHz spectrum divided into 11 channels at different frequencies
 - AP admin chooses frequency for AP
 - interference possible: channel can be same as that chosen by neighboring AP!
- host: must **associate** with an AP
 - scans channels, listening for **beacon frames** containing AP's name (SSID) and MAC address
 - selects AP to associate with
 - may perform authentication
 - will typically run DHCP to get IP address in AP's subnet

IEEE 802.11: multiple access

- avoid collisions: 2+ nodes transmitting at same time
 - 802.11: CSMA - sense before transmitting
 - don't collide with ongoing transmission by other node
 - 802.11: no collision detection!
 - difficult to receive (sense collisions) when transmitting due to weak received signals (fading)
 - can't sense all collisions in any case: hidden terminal, fading
- go **avoid collisions: CSMA/CS**

source

Wireless, Mobile Networks 6-13

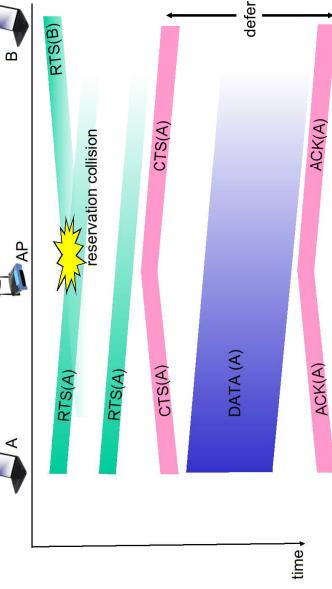
Avoiding collisions

- idea:** allow sender to "reserve" channel rather than random access of data frames; avoid collisions of long data frames
 - sender first transmits **small request-to-send (RTS)** packets to BS using CSMA
 - RTS may still collide with each other (but they're short)
 - BS broadcasts **clear-to-send CTS** in response to RTS
 - CTS heard by all nodes
 - sender transmits data frame
 - other stations defer transmissions

avoid data frame collisions completely using small reservation packets!

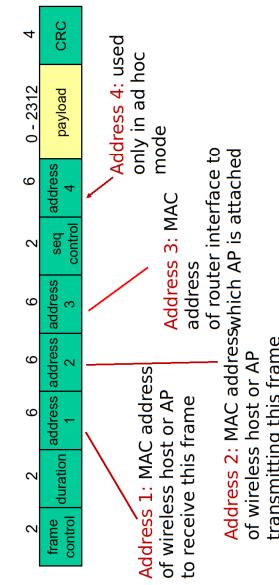
Wireless, Mobile Networks 6-16

Collision Avoidance: RTS-CTS exchange



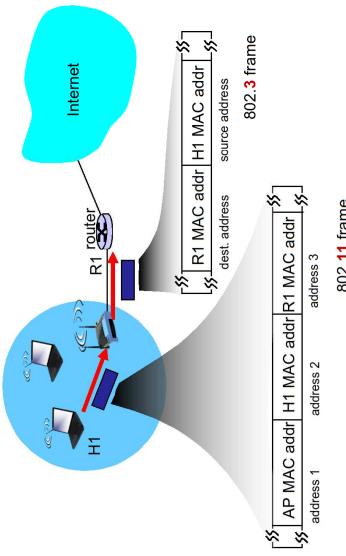
Wireless, Mobile Networks 6-17

802.11 frame: addressing



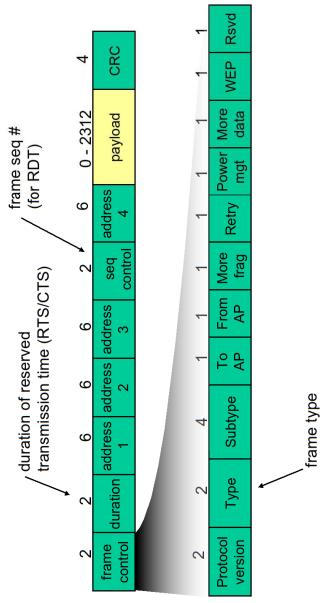
Wireless, Mobile Networks 6-18

802.11 frame: addressing



Wireless, Mobile Networks 6-19

802.11 frame: more



Wireless, Mobile Networks 6-20

802.11: passive/active scanning



Wireless, Mobile Networks 6-14

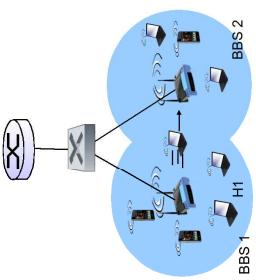
passive scanning:

- (1) beacon frames sent from APs
- (2) association Request frame sent: H1 to selected AP
- (3) association Response frame sent from selected AP to H1
- (4) Association Request frame sent from selected AP to H1

Wireless, Mobile Networks 6-15

802.11: mobility within same subnet

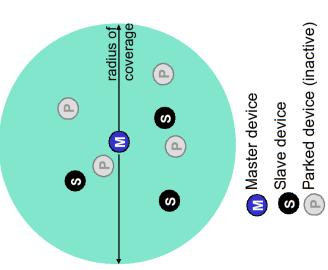
- ❖ H1 remains in same IP subnet: IP address can remain same
- ❖ switch: which AP is associated with H1?
- self-learning (Ch. 5): switch will see frame from H1 and "remember" which switch port can be used to reach H1



Wireless, Mobile Networks 6:21

802.15: personal area network

- ❖ less than 10 m diameter
- ❖ replacement for cables (mouse, keyboard, headphones)
- ❖ ad hoc: no infrastructure master/slaves:
 - slaves request permission to send (to master)
 - master grants requests
- ❖ 802.15; evolved from Bluetooth specification
 - 2.4-2.5 GHz radio band
 - up to 721 kbps



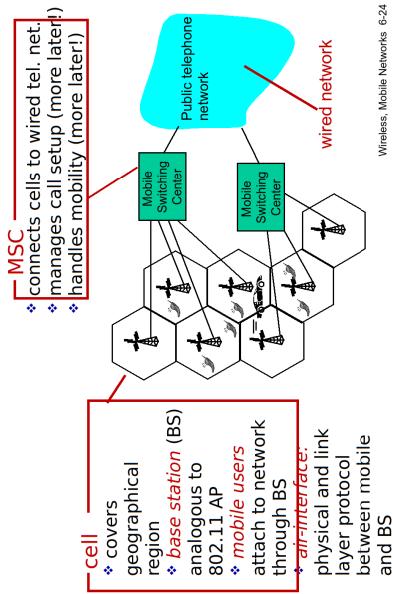
Wireless, Mobile Networks 6:23

802.11: advanced capabilities

- ❖ **power management**
 - node-to-AP: "I am going to sleep until next beacon frame"
 - AP knows not to transmit frames to this node
 - node wakes up before next beacon frame
 - beacon frame: contains list of mobiles with AP-to-mobile frames waiting to be sent
 - node will stay awake if AP-to-mobile frames to be sent; otherwise sleep again until next beacon frame

Wireless, Mobile Networks 6:22

Components of cellular network architecture

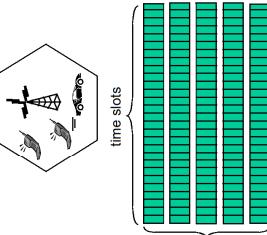


Wireless, Mobile Networks 6:24

Cellular networks: the first hop

Two techniques for sharing mobile-to-BS radio spectrum

- ❖ **combined FDMA/TDMA:** divide spectrum in frequency channels, divide each channel into time slots
- ❖ **CDMA:** code division multiple access



Wireless, Mobile Networks 6:25

How do you contact a mobile friend?

Consider friend frequently changing addresses, how do you find her?

- ❖ search all phone books?
- ❖ call her parents?
- ❖ expect her to let you know where he/she is?



Wireless, Mobile Networks 6:27

Mobility: approaches

- ❖ **let routing handle it:** routers advertise permanent address of mobile-nodes-in-residence via usual routing table exchange.
 - routing tables indicate where each mobile located
 - no changes to end-systems
- ❖ **let end-systems handle it:**
 - **indirect routing:** communication from correspondent to mobile goes through home agent, then forwarded to remote
 - **direct routing:** correspondent gets foreign address of mobile, sends directly to mobile

Wireless, Mobile Networks 6:26

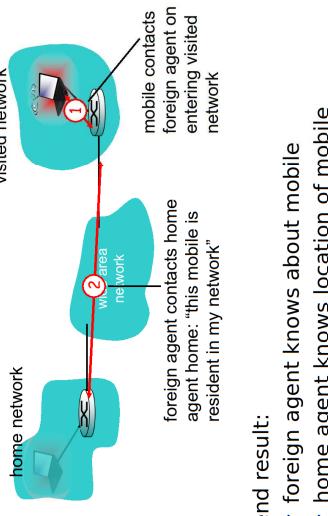
Wireless, Mobile Networks 6:28

Mobility: approaches

- ❖ let routing handle it:
 - address of mobile not scalable
 - routing table entries to millions of mobiles
 - no changes to systems
- ❖ let end-systems handle it:
 - **indirect routing:** communication from correspondent to mobile goes through home agent, then forwarded to remote
 - **direct routing:** correspondent gets foreign address of mobile, sends directly to mobile

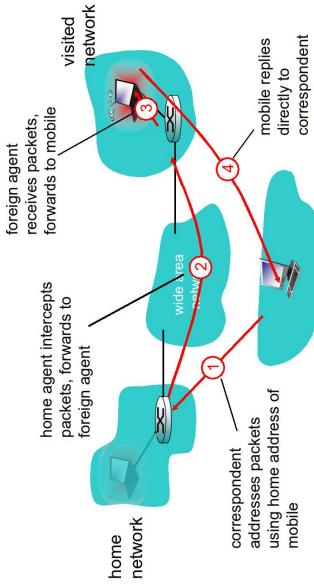
Wireless, Mobile Networks 6-29

Mobility: registration



Wireless, Mobile Networks 6-30

Mobility via indirect routing



Wireless, Mobile Networks 6-31

Indirect routing: moving between networks

- ❖ suppose mobile user moves to another network
 - registers with new foreign agent
 - new foreign agent registers with home agent
 - home agent update care-of-address for mobile
 - packets continue to be forwarded to mobile (but with new care-of-address)
- ❖ mobility, changing foreign networks transparent: **on going connections can be maintained!**

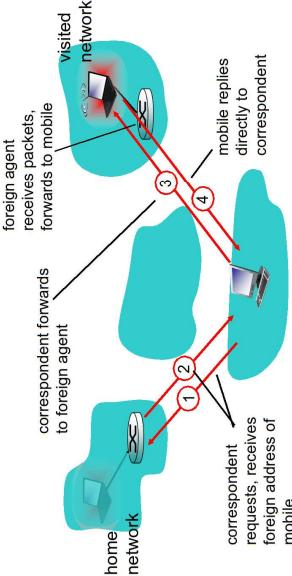
Wireless, Mobile Networks 6-33

Indirect Routing: comments

- ❖ mobile uses two addresses:
 - permanent address: used by correspondent (hence mobile location is **transparent** to correspondent)
 - care-of-address: used by home agent to forward datagrams to mobile
- ❖ foreign agent functions may be done by mobile itself
- ❖ triangle routing: correspondent-home-network-mobile
 - inefficient when correspondent, mobile are in same network

Wireless, Mobile Networks 6-32

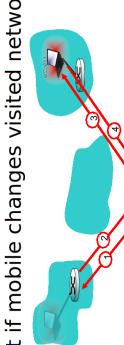
Mobility via direct routing



Wireless, Mobile Networks 6-34

Mobility via direct routing: comments

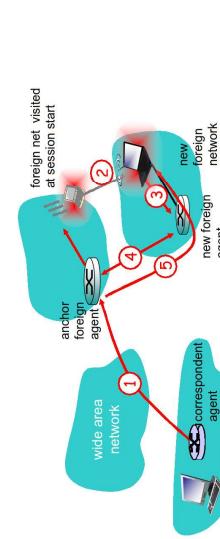
- ❖ overcome triangle routing problem
- ❖ **non-transparent to correspondent:** correspondent must get care-of-address from home agent
 - what if mobile changes visited network?



Wireless, Mobile Networks 6-35

Accommodating mobility with direct routing

- ❖ anchor foreign agent: FA in first visited network
- ❖ data always routed first to anchor FA
- ❖ when mobile moves: new FA arranges to have data forwarded from old FA (chaining)

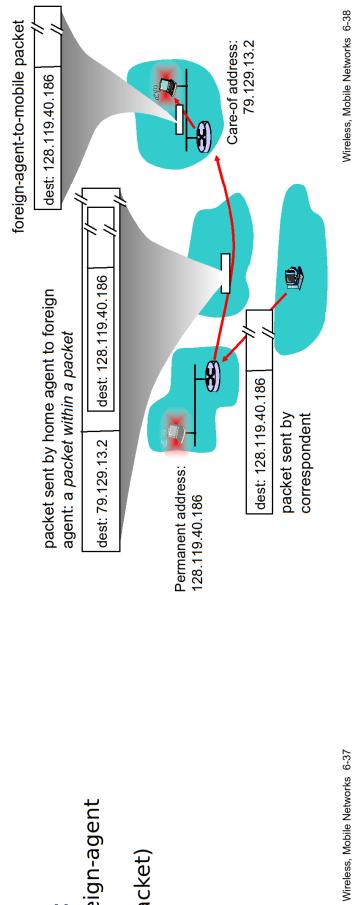


Wireless, Mobile Networks 6-36

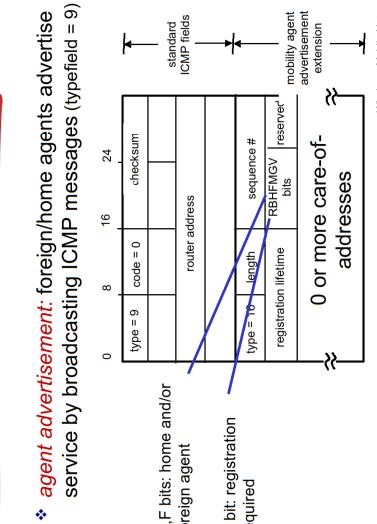
Mobile IP

Mobile IP: indirect routing

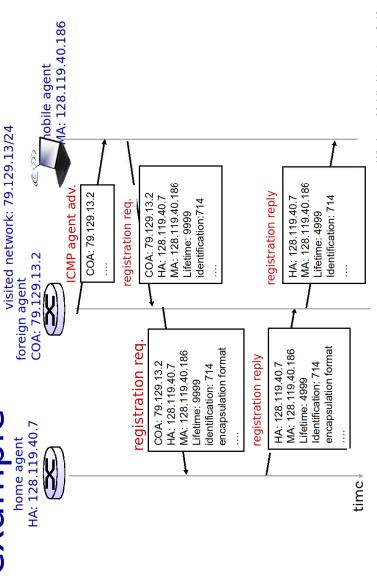
- ❖ RFC 3344
- ❖ has many features we've seen:
 - home agents, foreign agents, foreign-agent registration, care-of-addresses, encapsulation (packet-within-a-packet)
 - three components to standard:
 - indirect routing of datagrams
 - agent discovery
 - registration with home agent



Mobile IP: agent discovery



Mobile IP: registration example



Chapter 1: introduction

our goal:

- ❖ get "feel" and terminology
- ❖ more depth, detail later in course
- ❖ approach:
 - use Internet as example
 - protocol layers, service models
 - history
- ❖ what's the Internet?
- ❖ what's a protocol?
- ❖ network edge; hosts, access net, physical media
- ❖ network core: packet/circuit switching, Internet structure
- ❖ performance: loss, delay, throughput
- ❖ security
- ❖ specific actions taken when msgs received, or other events

What's the Internet: "nuts and bolts" view

Overview:

- ❖ millions of connected computing devices:
 - hosts = end systems
 - running network apps
- ❖ communication links
 - fiber, copper, radio, satellite transmission
 - transmission rate: **bandwidth**
- ❖ Packet switches:
 - forward packets (chunks of data)
 - routers and switches

Introduction 1-1

What's the Internet: "nuts and bolts" view

Internet: "network of networks"

- Interconnected ISPs
- protocols control sending, receiving of msgs
 - e.g., TCP, IP, HTTP, Skype, 802.11
- **Internet standards**
 - RFC: Request for comments
 - IETF: Internet Engineering Task Force

network protocols:

- ❖ "what's the time?"
- ❖ "I have a question"
- ❖ introductions
- ... specific msgs sent
- ... specific actions taken when msgs received, or other events
- ❖ machines rather than humans
- ❖ all communication activity in Internet governed by protocols
- ❖ **protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt**

Introduction 1-2

What's a protocol?

human protocols:

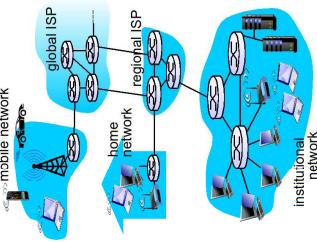
- ❖ "what's the time?"
- ❖ "I have a question"
- ❖ introductions
- ... specific msgs sent
- ... specific actions taken when msgs received, or other events
- ❖ machines rather than humans
- ❖ all communication activity in Internet governed by protocols
- ❖ **protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt**

Introduction 1-4

Introduction 1-3

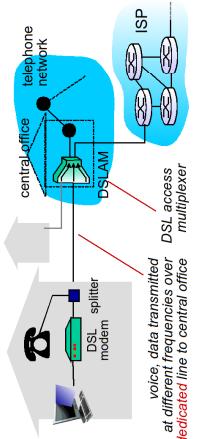
A closer look at network structure.

- ❖ **network edge:**
 - hosts: clients and servers
 - servers often in data centers
- ❖ **access networks, physical media:** wired, wireless communication links
- ❖ **network core:**
 - interconnected routers
 - network of networks



Introduction 1.5

Access net: digital subscriber line (DSL)

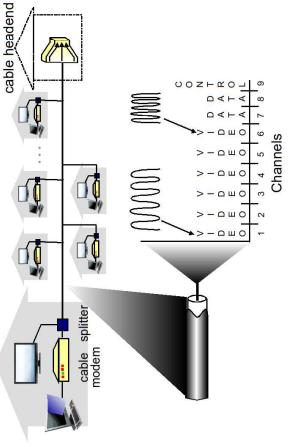


- ❖ use **existing** telephone line to central office DSLAM
 - data over DSL phone line goes to Internet
 - voice over DSL phone line goes to telephone net
 - < 2.5 Mbps upstream transmission rate (typically < 1 Mbps)
 - < 24 Mbps downstream transmission rate (typically > 10 Mbps)

- ❖ **Q: How to connect end systems to edge router?**
 - residential access nets
 - institutional access networks (school, company)
 - mobile access networks
- ❖ **Keep in mind:**
 - bandwidth (bits per second) of access network?
 - shared or dedicated?

Introduction 1.6

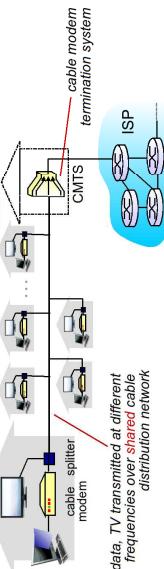
Access net: cable network



❖ **frequency division multiplexing:** different channels transmitted over different frequency bands

Introduction 1.8

Access net: cable network

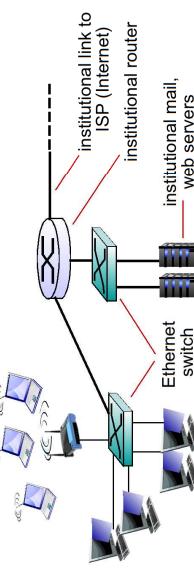


HFC: hybrid fiber coax

- ❖ asymmetric: up to 30Mbps downstream transmission rate, 2 Mbps upstream transmission rate
- ❖ network of cable, fiber attaches homes to ISP router
 - homes **share** **access network** to cable headend
 - unlike DSL, which has dedicated access to central office

Introduction 1.9

Enterprise access networks (Ethernet)



- ❖ typically used in companies, universities, etc
- ❖ 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
- ❖ today, end systems typically connect into Ethernet switch

Introduction 1.11

Access networks and physical media

- ❖ **Q: How to connect end systems to edge router?**
 - residential access nets
 - institutional access networks (school, company)
 - mobile access networks
- ❖ **Keep in mind:**
 - bandwidth (bits per second) of access network?
 - shared or dedicated?

Introduction 1.6

- ❖ **wide-area wireless access**
 - provided by telco (cellular) operator, 10's km between 1 and 10 Mbps
 - 3G, 4G: LTE
- ❖ **wireless LANs:**
 - within building (100 ft)
 - 802.11b/g WiFi: 11, 54 Mbps transmission rate
- ❖ **shared wireless access network** connects end system to router
 - via base station aka "access point"

Introduction 1.12

Host: sends *packets* of data

- host sending function:
 - takes application message
 - breaks it into smaller chunks, known as **packets**, of length **L** bits
 - transmits packet into access network at **transmission rate R**
 - link transmission rate, aka link **capacity**, aka **link bandwidth**

Physical media

- ❖ **bit:** propagates between transmitter/receiver pairs
 - ❖ **physical link:** what lies between transmitter & receiver
 - ❖ **guided media:**
 - signals propagate in solid media: copper, fiber, coax
 - ❖ **unguided media:**
 - signals propagate freely, e.g., radio

twisted pair (TP)

 - ❖ two insulated copper wires
 - Category 5: 100 Mbps, 1 Gbps
 - Ethernet
 - ~ 1 Gbps

$$\frac{\text{packet transmission delay}}{\text{time needed to transmit } L\text{-bit packet into link}} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

Introduction 1-14

Physical media: coax, fiber

- | | | |
|---------------------------|--|---|
| coaxial cable: | <ul style="list-style-type: none"> two concentric copper conductors bidirectional broadband: multiple channels on cable |  |
| fiber optic cable: | <ul style="list-style-type: none"> glass fiber carrying pulses, each pulse carries high-speed data high-speed point-to-point transmission (e.g., Gbps transmission) low error rate: repeaters spaced far apart immune to electro-magnetic noise |  |



Introduction

Physical media: radio

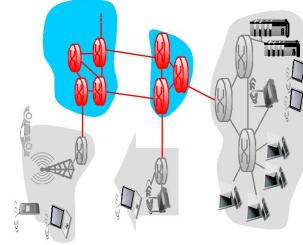
- ❖ signal carried in electromagnetic spectrum
 - ❖ no physical “wire”
 - ❖ bidirectional
 - ❖ propagation environment effects:
 - reflection
 - obstruction by objects
 - interference
 - ❖ terrestrial microwave
 - e.g. up to 45 Mbps channels
 - ❖ LAN (e.g. WiFi)
 - 11Mbps, 54 Mbps
 - ❖ wide-area (e.g., cellular)
 - 3G cellular: ~ few Mbps
 - ❖ satellite
 - Kbps to 45Mbps channel (or multiple smaller channels)
 - 270 msec end-end delay
 - geosynchronous versus low altitude



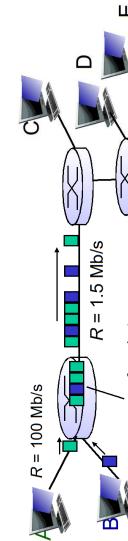
Introduction 1-16

The network core

- ❖ mesh of interconnected routers
 - ❖ packet-switching: hosts break application-layer messages into packets
 - forward packets from one router to the next, across links on path from source to destination
 - each packet transmitted at full link capacity



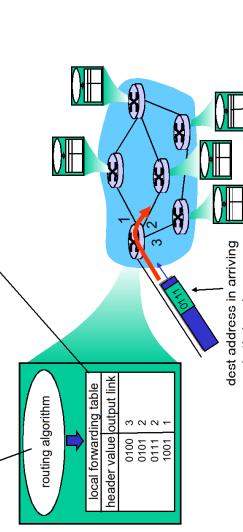
Packet Switching: queueing delay,



- ❖ If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
 - packets will queue, wait to be transmitted on link
 - packets can be dropped (lost) if memory (buffer) fills

Two key network-core

- forwarding:** move packets from router's input to appropriate output +
routing: determines source-destination route taken by packets
+
switching: move packets from one interface to another



1

...
כג

Packet-switching: store-and-forward

- one-hop numerical example:**

 - $L = 7.5 \text{ Mbit/s}$
 - $R = 1.5 \text{ Mbps}$
 - one-hop transmission delay = 5 sec

ignore on delay shortly ...

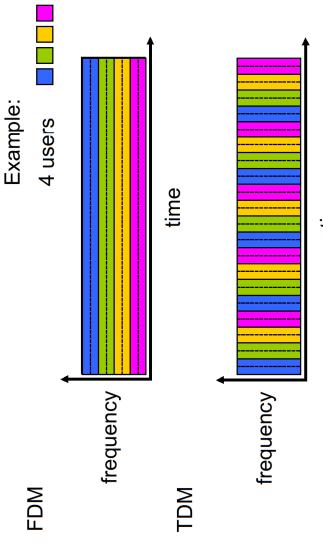
 - takes L/R seconds to transmit (push out) L -bit packet into link at R bps
 - store and forward:** entire packet must arrive at router before transmission continues

...
כג

Alternative core: circuit switching

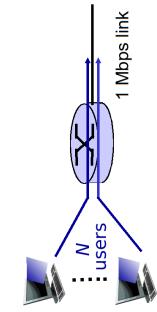
end-end resources allocated to, reserved for “call” between source & dest:
♦ In diagram, each link has four circuits.
▪ call gets 2nd circuit in top link and 1st circuit in right link.
♦ dedicated resources: no sharing
▪ circuit-like (guaranteed) performance
♦ circuit segment idle if not used by call (*no sharing*)
♦ Commonly used in traditional telephone networks

Circuit switching: FDM versus TDM



Packet switching versus circuit switching

packet switching allows more users to use network!



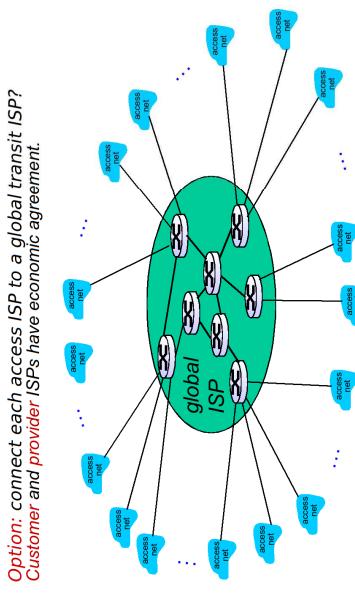
example:

- 1 Mb/s link
 - each user:
 - 100 kb/s when “active”
 - active 10% of time
- ❖ **circuit-switching:**
- 10 users
 - **packet switching:** Q: how did we get value 0.0004
 - with 35 users, probability > 10 active Q: what happens if > 35 users ? at same time is less than .0004 *

Introduction 1-21

Internet structure: network of networks

- ❖ End systems connect to Internet via **access ISPs** (Internet Service Providers)
 - Residential, company and university ISPs
 - Access ISPs in turn must be interconnected.
 - ❖ So that any two hosts can send packets to each other
 - Resulting network of networks is very complex
 - ❖ Evolution was driven by **economics** and **national policies**
 - ❖ Let's take a stepwise approach to describe current Internet structure

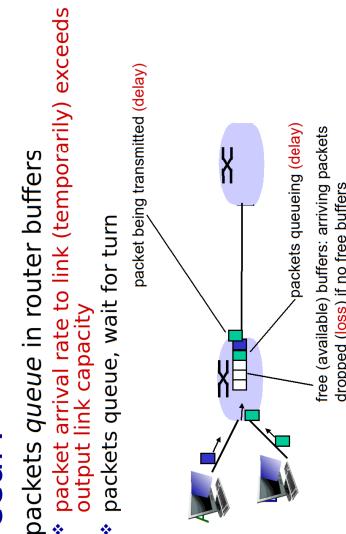


Introduction 1-24

Internet structure: network of networks

- ❖ at center: small # of well-connected large networks
 - “tier 1” **commercial ISPs** (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
 - **content provider network** (e.g., Google): private network that connects its data centers to Internet, often bypassing tier-1 regional ISPs
- ❖ at center: small # of well-connected large networks
 - “tier 1” **commercial ISPs** (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
 - **content provider network** (e.g., Google): private network that connects its data centers to Internet, often bypassing tier-1 regional ISPs

How do loss and delay occur?

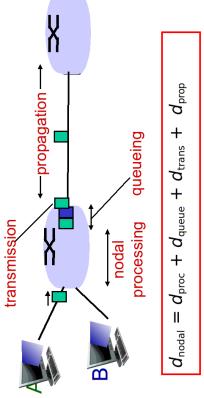


- ❖ packets queue in router buffers
 - ❖ **packet arrival rate to link (temporarily) exceeds output link capacity**
 - ❖ packets queue, wait for turn
- packet being transmitted (delay)
- free (available) buffers: arriving packets dropped (loss)

Introduction 1-27

Introduction 1-28

Four sources of packet delay

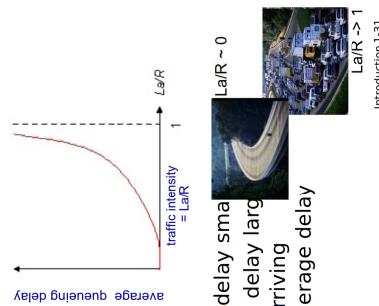


- d_{proc} : nodal processing**
- check bit errors
 - determine output link
 - typically < msec
- d_{queue} : queueing delay**
- time waiting at output link for transmission
 - depends on congestion level of router

Introduction 1.29

Queueing delay

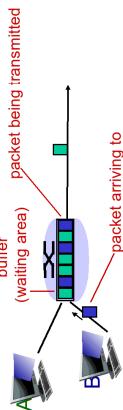
- ❖ R : link bandwidth (bps)
- ❖ L : packet length (bits)
- ❖ a : average packet arrival rate
- ❖ $La/R \sim 0$: avg. queueing delay small
- ❖ $La/R \rightarrow 1$: avg. queueing delay large
- ❖ $La/R > 1$: more “work” arriving than can be serviced, average delay infinite!



Introduction 1.31

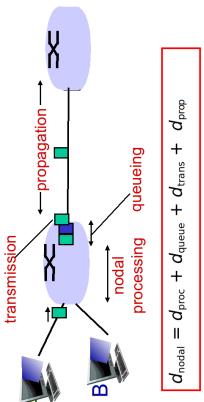
Packet loss

- d_{drop} : propagation delay:**
- queue (aka buffer) preceding link in buffer has finite capacity
 - packet arriving to full queue dropped (aka lost)
 - lost packet may be retransmitted by previous node, by source end system, or not at all



* Check out the Java applet for an interactive animation on trans vs prop delay

Introduction 1.30



- d_{trans} : transmission delay:**
- L : packet length (bits)
 - R : link bandwidth (bps)
 - $d_{trans} = L/R$
 - $d_{prop} = d_{trans}$ and $d_{prop} \neq d_{trans}$ very different

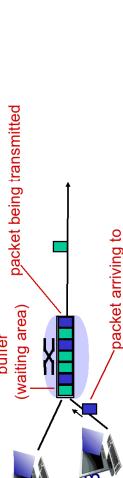
$d_{total} = d_{proc} + d_{queue} + d_{trans} + d_{drop}$

* Check out the Java applet for an interactive animation on trans vs prop delay

Introduction 1.32

Four sources of packet delay

- d_{drop} : propagation delay:**
- d : length of physical link
 - s : propagation speed in medium (~ 2×10^8 m/sec)
 - $d_{prop} = d/s$

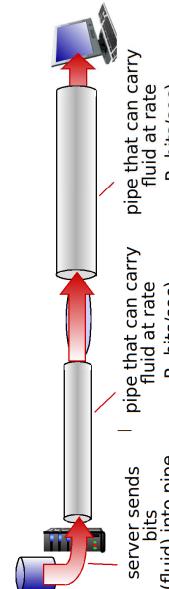


* Check out the Java applet for an interactive animation on trans vs prop delay

Introduction 1.30

Throughput

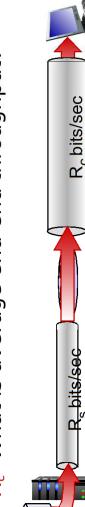
- ❖ **throughput**: rate (bps/time unit) at which bits transferred between sender/receiver
- **instantaneous**: rate at given point in time
 - **average**: rate over longer period of time



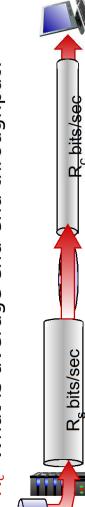
Introduction 1.33

Throughput (more)

- ❖ $R_s < R_c$ What is average end-end throughput?



- ❖ $R_s > R_c$ What is average end-end throughput?



bottleneck link throughput d path that constrains end-end throughput

Introduction 1.34

Throughput: Internet scenario

- ❖ per-connection end-end throughput: $\min(R_c, R_s, R/10)$
- ❖ in practice: R_c or R_s is often bottleneck

Networks are complex, with many “pieces”:

- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

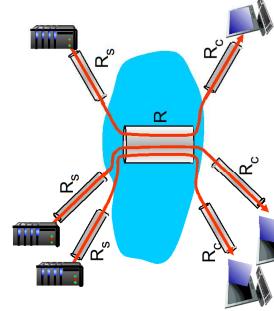
10 connections (fairly) share backbone bottleneck link R bits/sec

Introduction 1.35

Protocol “layers”

Question: ... is there any hope of organizing structure of network?

... or at least our discussion of networks?



Introduction 1.36

Why layering?

- dealing with complex systems:
 - explicit structure allows identification, relationship of complex system's pieces
 - layered **reference model** for discussion
 - modularization eases maintenance, updating of system
 - change of implementation of layer's service transparent to rest of system
 - e.g., change in gate, procedure doesn't affect rest of system
 - layering considered harmful?

Internet protocol stack

- application:** supporting network applications
 - FTP, SMTP, HTTP
- transport:** process-process data transfer
 - TCP, UDP
- network:** routing of datagrams from source to destination
 - IP, routing protocols
- link:** data transfer between neighboring network elements
 - Ethernet, 802.11 (WiFi), PPP
- physical:** bits "on the wire"

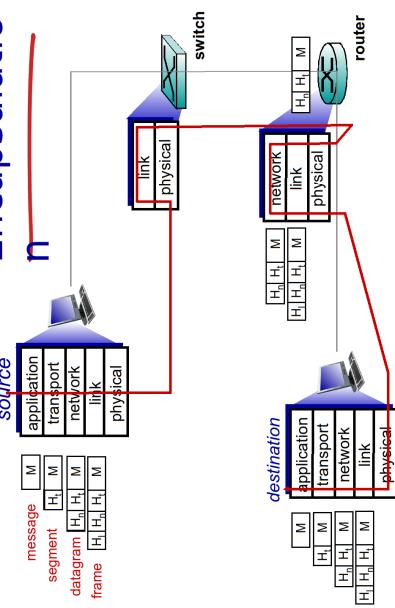
Introduction 1.37

Introduction 1.38

ISO/OSI reference model

- presentation:** allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- session:** synchronization, checkpointing, recovery of data exchange
- transport:** Internet stack "missing" these layers!
 - these services, *if needed*, must be implemented in application
 - needed?

Encapsulation



Introduction 1.39

Introduction 1.40

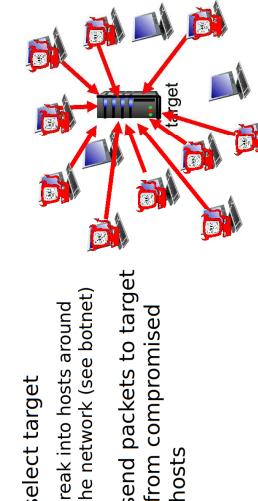
Network security

- field of network security:**
 - how bad guys can attack computer networks
 - how we can defend networks against attacks
 - how to design architectures that are immune to attacks
- Internet not originally designed with (much) security in mind**
 - original vision:* "a group of mutually trusting users attached to a transparent network" \odot
 - Internet protocol designers playing "catch-up"
 - security considerations in all layers!

Introduction 1.41

Bad guys: attack server, network infrastructure

- Denial of Service (DoS):** attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic
 - select target
 - break into hosts around the network (see botnet)
 - send packets to target from compromised hosts

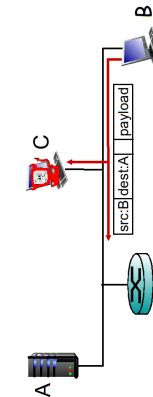


Introduction 1.43

Bad guys can sniff packets

packet "sniffing":

- broadcast media (shared ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by

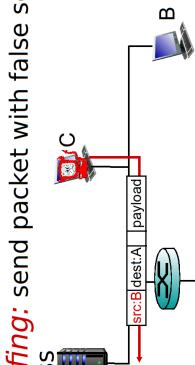


- wireshark software used for end-of-chapter labs is a (free) packet-sniffer

Introduction 1.44

Bad guys can use fake addresses

IP spoofing: send packet with false source address



The diagram shows three hosts labeled A, B, and C. Host A is on the left, host B is in the middle, and host C is on the right. A line connects A and B. A packet is shown traveling from A to B. The packet's header contains the source address 'src:B' and the destination address 'dest:A'. The payload of the packet is labeled 'payload'. A red arrow points from host C towards the connection between A and B, indicating that host C is spoofing its source address to appear as host B.

... lots more on security (throughout, Chapter 8)

Introduction 1.45

Internet history

1961-1972: Early packet-switching principles

- ❖ 1961: Kleinrock - queuing theory shows effectiveness of packet-switching
- ❖ 1964: Baran - packet-switching in military nets
- ❖ 1967: ARPAnet conceived by Advanced Research Projects Agency
- ❖ 1969: first ARPAnet node operational
- ❖ 1972: ARPA public demo NCP (Network Control Protocol) first host-host protocol
- ❖ 1972: first e-mail program
- ❖ 1972: ARPAnet has 15 nodes

THE ARP NETWORKS

Introduction 1.46

Internet history

1972-1980: Internetworking, new and proprietary nets

- ❖ 1970: ALOHAnet satellite network in Hawaii
 - ❖ 1974: Cerf and Kahn - architecture for interconnecting networks
 - ❖ 1976: Ethernet at Xerox PARC
 - ❖ late 70's: proprietary architectures: DECnet, SNA, XNA
 - ❖ late 70's: switching fixed length packets (ATM precursor)
 - ❖ 1979: ARPAnet has 200 nodes
- Cerf and Kahn's internetworking principles:
 - minimalism, autonomy - no internal changes required to interconnect networks
 - best effort service model
 - stateless routers
 - decentralized control
 - define today's Internet architecture

Introduction 1.47

Internet history

1990, 2000's: commercialization, the Web, new apps

- ❖ early 1990's: ARPAnet decommissioned
- ❖ 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- ❖ early 1990s: Web
 - hypertext [Bush 1945, Nelson 1960's]
 - HTML, HTTP: Berners-Lee
 - 1994: Mosaic, later Netscape
 - late 1990's: commercialization of the Web
- ❖ late 1990's - 2000's:
 - more killer apps; instant messaging, P2P file sharing
 - network security to forefront
 - est. 50 million hosts, 100 million+ users
 - backbone links running at Gbps

Introduction 1.49

Internet history

1980-1990: new protocols, a proliferation of networks

- ❖ 1983: deployment of TCP/IP
- ❖ 1982: smtp e-mail protocol defined
- ❖ 1983: DNS defined for name-to-IP-address translation
- ❖ 1985: ftp protocol defined
- ❖ 1988: TCP congestion control
- ❖ 1983: new national networks: Cernet, Bitnet, NSFnet, Minitel
- ❖ 1983: 100,000 hosts connected to confederation of networks

Introduction 1.50

Introduction 1.48

Chapter 7: Network Security

Friends and enemies: Alice, Bob, Trudy

- **confidentiality:** only sender, intended receiver should “understand” message contents
 - sender encrypts message
 - receiver decrypts message

authentication. sender, receiver want to confirm identity of each other

message integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

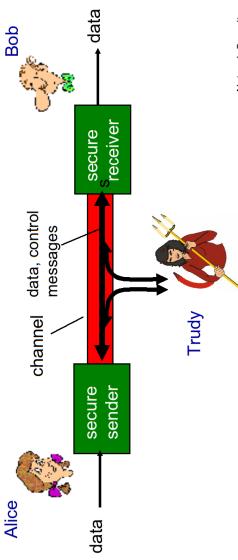
access and availability: services must be accessible and available to users

Network Security 8-3

Who might Bob, Alice be?

- ... well, *real-life* Bobs and Alices!
- Web browser/server for electronic transactions (e.g., on-line purchases)
- on-line banking client/server
- DNS servers
- routers exchanging routing table updates
- other examples?

Network Security 8-2

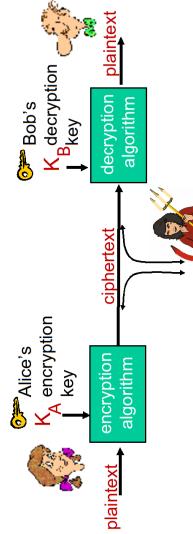


There are bad guys (and girls) out there!

- Q: What can a “bad guy” do?
A: A lot! See section 1.6
- **eavesdrop:** intercept messages
 - actively **insert** messages into connection
 - **impersonation:** can fake (spoof) source address in packet (or any field in packet)
 - **hijacking:** “take over” ongoing connection by removing sender or receiver, inserting himself in place
 - **denial of service:** prevent service from being used by others (e.g., by overloading resources)

Network Security 8-4

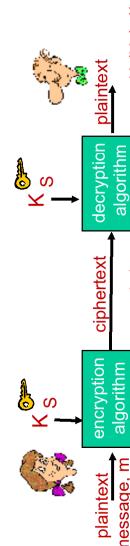
The language of cryptography



m plaintext message
 $m = K_A(K_B(m))$

Network Security 8-5

Symmetric key cryptography



symmetric key crypto: Bob and Alice share same (symmetric) key, K

- e.g., key is knowing substitution pattern in mono alphabetic substitution cipher
- how do Bob and Alice agree on key value?

Q: **Encryption key:** mapping from set of 26 letters to set of 26 letters

Network Security 8-7

Simple encryption scheme

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another

plaintext: abcdefghijklmnopqrstuvwxyz
ciphertext: mnbvcxzasdfghijklppoiuytrewq

e.g.: Plaintext: bob. i love you. alice
ciphertext: nkn. s gktc wky. mgsbc

Q: **Encryption key:** mapping from set of 26 letters to set of 26 letters

Network Security 8-8

Symmetric key crypto: DES

Symmetric key
crypto: DES

- ❖ US encryption standard [NIST 1993]

- ❖ 56-bit symmetric key, 64-bit plaintext input
 - ❖ how secure is DES?

- DES Challenge: 56-bit-key-encrypted phrase

decrypted (brute force) in less than

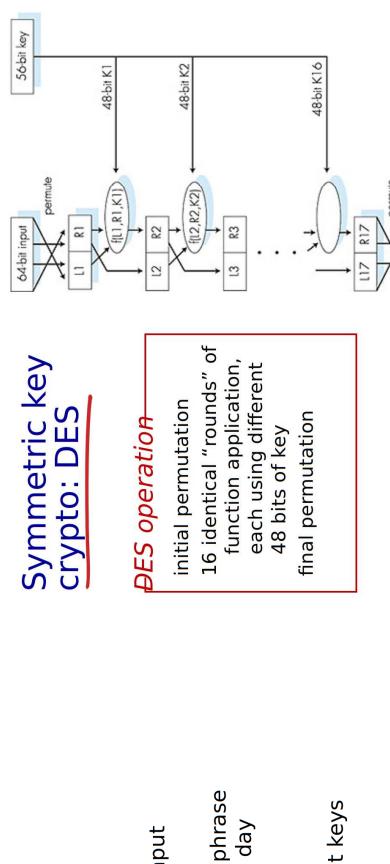
- no known good analytic attack

- ❖ making DES more secure:

- 3DES: encrypt 3 times with 3 different keys

AES: Advanced Encryption Standard

- ❖ symmetric-key NIST standard, replaced DES (Nov 2001)
 - ❖ processes data in 128 bit blocks
 - ❖ 128, 192, or 256 bit keys
 - ❖ brute force decryption (try each key)
taking 1 sec on DES, takes 149 trillion years for AES



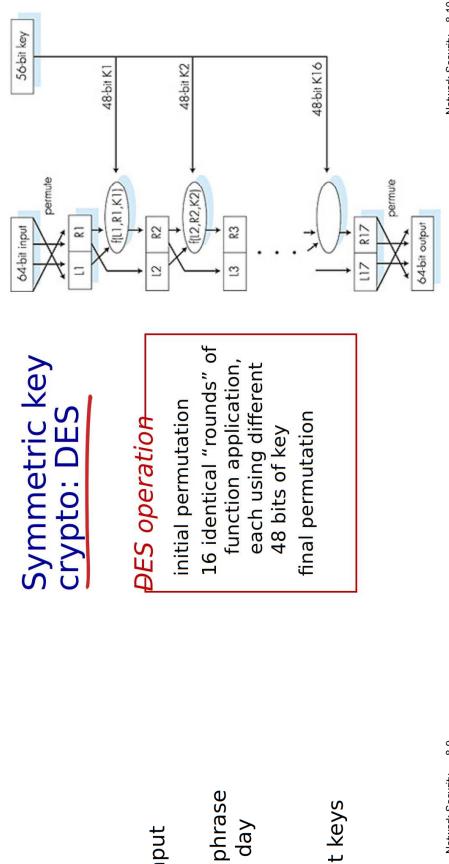
Public Key Cryptography

- | | |
|----------------------|---|
| DES operation | initial permutation
16 identical “rounds” of
function application,
each using different
48 bits of key
final permutation |
|----------------------|---|

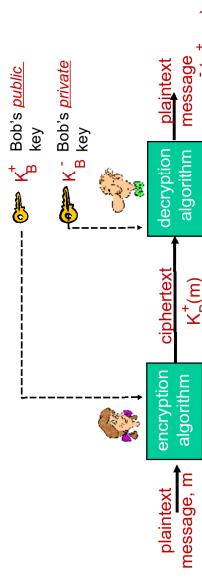


Public Key Cryptography

- | symmetric key crypto | public key crypto |
|--|--|
| <ul style="list-style-type: none"> requires sender, receiver do not share secret key public encryption key known to all private decryption key known only to receiver | <ul style="list-style-type: none"> sender, receiver do share secret key public encryption key known to all private decryption key known only to receiver |



Public key cryptography



Public key encryption
~~allows~~

- requirements:

- ① need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that

$$K_B^-(K_B^+(m)) = m$$
 - ② given public key K , it should be impossible to
 compute private key K_B

BSA: Biostat Shamsir Adelson algorithm

Prerequisite: modular arithmetic

- * $x \bmod n = \text{remainder of } x \text{ when divide by } n$
 - * facts:
 - $(a \bmod n) + (b \bmod n) \bmod n = (a+b) \bmod n$
 - $(a \bmod n) - (b \bmod n) \bmod n = (a-b) \bmod n$
 - $(a \bmod n) * (b \bmod n) \bmod n = (a*b) \bmod n$
 - * thus $(a \bmod n)^d \bmod n = a^d \bmod n$
 - * example: $x=14, n=10, d=2:$
 $(x \bmod n)^d \bmod n = 14^2 \bmod 10 = 6$
 $X^d = 14^2 = 196 \quad X^d \bmod 10 = 6$

RSA: getting ready

- ❖ message: just a bit pattern
 - ❖ bit pattern can be uniquely represented by an integer number
 - ❖ thus, encrypting a message is equivalent to encrypting a number.

example:

 - ❖ m = 101001 . This message is uniquely represented by the decimal number 145.
 - ❖ to encrypt m, we encrypt the corresponding number, which gives a new number (the ciphertext).

Network Security 8-15

Network Security 8-16

RSA: Creating public/private key pair

- choose two large prime numbers p, q . (e.g., 1024 bits each)
- compute $n = pq$, $z = (p-1)(q-1)$
- choose e (with $e < n$) that has no common factors with z (e, z are "relatively prime").
- choose d such that $ed-1$ is exactly divisible by z . (in other words: $ed \bmod z = 1$).
- public key is (n, e) , private key is K_B^+

Network Security 8:17

RSA: encryption, decryption

- given (n, e) and (n, d) as computed above
 - to encrypt message m ($< n$), compute $c \stackrel{e}{=} m \bmod n$
 - to decrypt received bit pattern, c , compute $m \stackrel{d}{=} c \bmod n$

Network Security 8:18

RSA example:

Bob chooses $p=5$, $q=7$. Then $n=35$, $z=24$.

$e=5$ (so e, z relatively prime).

$d=29$ (so $ed-1$ exactly divisible by z).

encrypting 8-bit messages.

$$\begin{array}{rcl} \text{encrypt: } & \underbrace{\text{letter}}_{\text{L}} & \underbrace{\text{m}}_{12} \xrightarrow{\text{m}^e} \underbrace{c}_{248832} \\ & & \xrightarrow{\text{c}^d} \underbrace{m}_{17} = \underbrace{c^d \bmod n}_{48168572106750915091411825223071687} \end{array}$$

Network Security 8:19

RSA: another important property

The following property will be **very** useful later:

$$K_B^{-1}(K_B^+(m)) = m = K_B^+(K_B^{-1}(m))$$

use public key first, followed by private key

result is the same!

Network Security 8:21

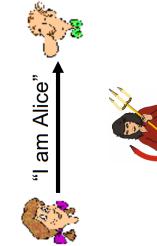
Why does RSA work?

- must show that $c^d \bmod n = m$ where $c = m^e \bmod n$
- fact: for any x and y : $x^y \bmod n = x^{y \bmod z} \bmod n$
 - where $n = pq$ and $z = (p-1)(q-1)$
- thus, $c^d \bmod n = (m^e \bmod n)^d \bmod n = m^{ed} \bmod n = m^{(ed) \bmod z} \bmod n = m^1 \bmod n = m$

Network Security 8:20

Authentication

Goal: Bob wants Alice to "prove" her identity to him
'rotocol ap1.0: Alice says "I am Alice"



Network Security 8:22

Authentication: another try

Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address



Network Security 8:23

Authentication

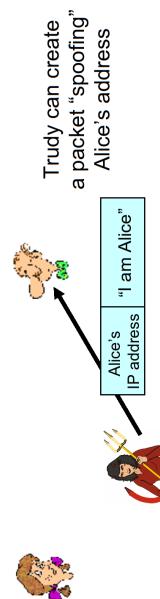
Goal: Bob wants Alice to "prove" her identity to him
'rotocol ap1.0: Alice says "I am Alice"



Network Security 8:24

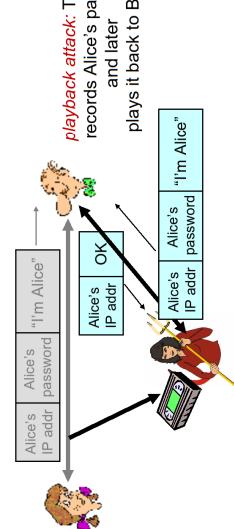
Authentication: another try

Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address



Authentication: another try

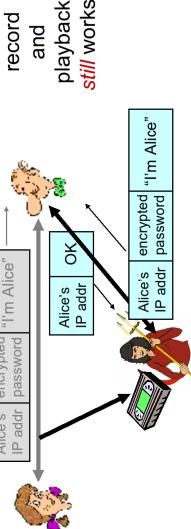
protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it



Network Security 8:27

Authentication: yet another try

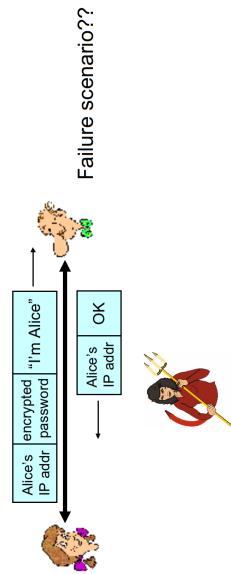
protocol ap3.1: Alice says "I am Alice" and sends her encrypted secret password to "prove" it



Network Security 8:29

Authentication: yet another try

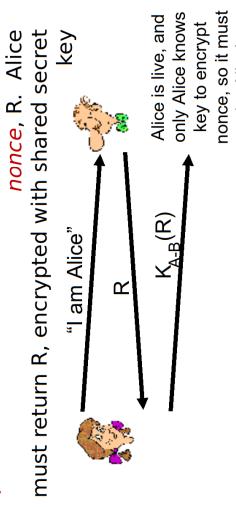
protocol ap3.0: Alice says "I am Alice" and sends her encrypted secret password to "prove" it



Network Security 8:26

Authentication: yet another try

protocol ap3.1: Alice says "I am Alice" and sends her encrypted secret password to "prove" it



Network Security 8:28

security hole

man (or woman) in the middle attack: Trudy poses as Alice (to Bob) and as Bob (to Alice)

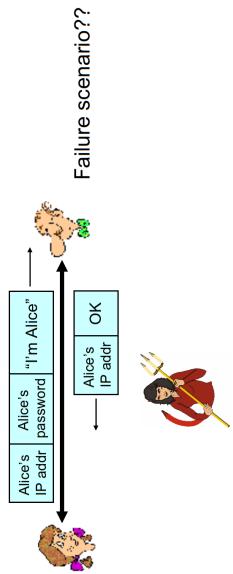


difficult to detect:

- Bob receives everything that Alice sends, and vice versa. (e.g., so Bob, Alice can meet one week later and recall conversations!)
- problem is that Trudy receives all messages as well!

Authentication: another try

protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it



Network Security 8:25

Digital signatures

cryptographic technique analogous to hand-written signatures:

- ❖ sender (Bob) digitally signs document, establishing he is document owner/creator.
- ❖ **verifiable, nonforgeable:** recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed document

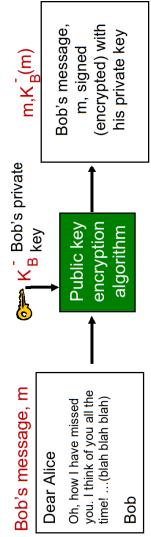
Network Security 8:31

Network Security 8:32

Digital signatures

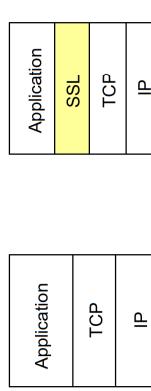
Simple digital signature for message m:

- Bob signs m by encrypting with his private key K_B , creating "signed" message, $K_B(m)$



Network Security 8-33

SSL and TCP/IP



normal application

- SSL provides application programming interface (API) to applications
- C and Java SSL libraries/classes readily available

Network Security 8-35

Toy SSL: a simple secure channel

- handshake:** Alice and Bob use their certificates, private keys to authenticate each other and exchange shared secret
- key derivation:** Alice and Bob use shared secret to derive set of keys
- data transfer:** data to be transferred is broken up into series of records
- connection closure:** special messages to securely close connection

Network Security 8-34

- widely deployed security protocol
 - supported by almost all browsers, web servers
 - https
- original goals:
 - Web e-commerce transactions
 - credit-card numbers
- mechanisms: [Woo 1994], implementation: Netscape variation -TLS: transport layer security, RFC 2246
 - provides
 - confidentiality
 - integrity
 - authentication
 - available to all TCP applications
 - secure socket interface

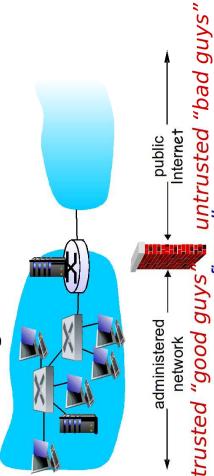
Network Security 8-34

SSL: Secure Sockets Layer

Firewalls

firewall

isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others



Network Security 8-37

Firewalls: why

prevent denial of service attacks:

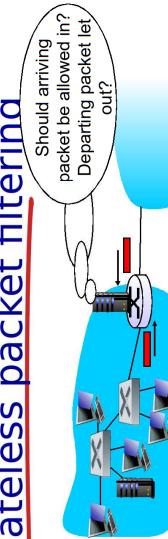
- SYN flooding: attacker establishes many bogus TCP connections, no resources left for "real" connections
- prevent illegal modification/access of internal data
 - e.g., attacker replaces homepage with something else
- allow only authorized access to inside network
 - set of authenticated users/hosts

three types of firewalls:

- stateless packet filters
- stateful packet filters
- application gateways

Network Security 8-38

Stateless packet filtering:



internal network connected to Internet via router

router filters packet-by-packet, decision to forward/drop packet based on:

- source IP address, destination IP address
- TCP/UDP source and destination port numbers
- ICMP message type
- TCP SYN and ACK bits

Network Security 8-39

Stateless packet filtering:

example

- example 1:** block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23
 - result:** all incoming, outgoing UDP flows and telnet connections are blocked
- example 2:** block inbound TCP segments with ACK=0.
 - result:** prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

Network Security 8-40

Stateless packet filtering: more examples

Policy	Firewall Setting
No outside Web access.	Drop all outgoing packets to any IP address, port 80
No incoming TCP connections, except those for institution's public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent Web-radios from eating up the available bandwidth.	Drop all incoming UDP packets - except DNS and router broadcasts.
Prevent your network from being used for a smurf DoS attack.	Drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255).
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

Network Security 8-41

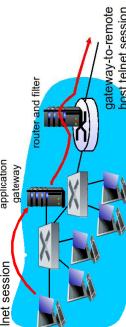
Stateful packet filtering

- ❖ **stateless packet filter:** heavy handed tool
 - admits packets that "make no sense," e.g., dest port = 80, ACK bit set, even though no TCP connection established:
- ❖ **stateful packet filter:** track status of every TCP connection
 - track connection setup (SYN), teardown (FIN): determine whether incoming, outgoing packets "makes sense"
 - timeout inactive connections at firewall: no longer admit packets

Network Security 8-42

Application gateways

- ❖ filter packets on application data as well as on IP/TCP/UDP fields.
- ❖ **example:** allow select internal users to telnet outside



1. require all telnet users to telnet through gateway.
2. for authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections
3. router filter blocks all telnet connections not originating from gateway.

Network Security 8-43

Limitations of firewalls, gateways

- ❖ **IP spoofing:** router can't know if data "really" comes from claimed source
- ❖ if multiple app's. need special treatment, each has own app. gateway client software must know how to contact gateway.
 - e.g., must set IP address of proxy in Web browser
- ❖ filters often use all or nothing policy for UDP
- ❖ **tradeoff:** degree of communication with outside world, level of security

Network Security 8-44