CSE589: Modern Network Concepts

Notes

Jinghao Shi
jinghaos@buffalo.edu

December 12, 2013

This page intentionally left blank.

Contents

1 Computer Networks and the Internet

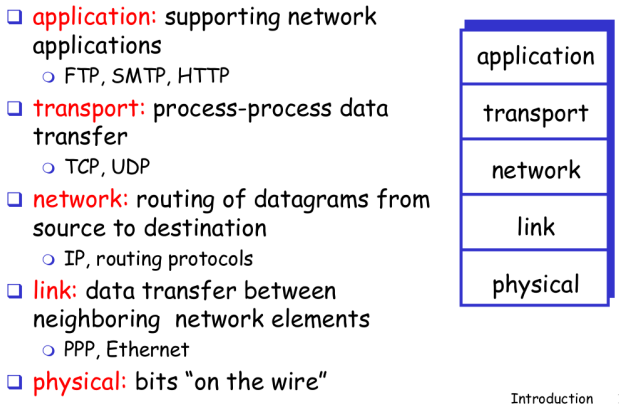
1.1 Network Edge

Network edge Applications and hosts

1.2 Network Core

Network core Routers, Network of networks

Internet protocol stack



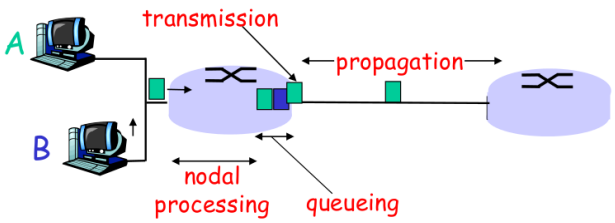
Apps Using TCP HTTP (Web), FTP (file transfer), Telnet (remote login), SMTP (email)

Apps using UDP streaming media, teleconferencing, **DNS**, Internet telephony

1.3 Delay & loss in packet-switched networks

Four sources of delay

1. Transmission delay d_{trans}
2. Propagation delay d_{prop}
3. Nodal processing d_{proc}
4. Queueing d_{queue}



Virtual Circuit With VC, two packets with the same destination can be assigned two different VC# and forced to take different paths.

2 Application Layer

2.1 Principles of network applications

TCP 4-tuple

- Source ⟨IP, Port⟩
- Dest ⟨IP, Port⟩

What transport service does an app need?

Data loss
- some apps (e.g., audio) can tolerate some loss
- other apps (e.g., file transfer, telnet) require 100% reliable data transfer

Timing
- some apps (e.g., Internet telephony, interactive games) require low delay to be "effective"

Throughput
- some apps (e.g., multimedia) require minimum amount of throughput to be "effective"
- other apps ("elastic apps") make use of whatever throughput they get

Security
- Encryption, data integrity, …

2.2 Web and HTTP

HTTP connections

Nonpersistent HTTP
- At most one object (e.g., a HTML file, or a jpeg image but not both!) is sent over a TCP connection.
- HTTP/1.0 uses nonpersistent HTTP

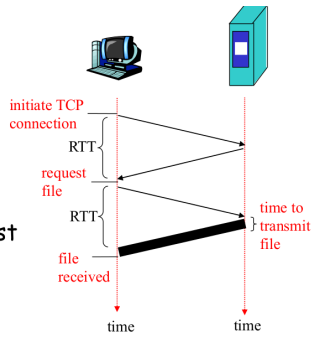
Persistent HTTP
- Multiple objects can be sent over single TCP connection between client and server.
- HTTP/1.1 uses persistent connections in default mode

Non-Persistent HTTP Response time modeling

Definition of RRT: time to send a small packet to travel from client to server and back.

Response time:
- one RTT to initiate TCP connection
- one RTT for HTTP request and first few bytes of response to return
- One file/one object transmission time

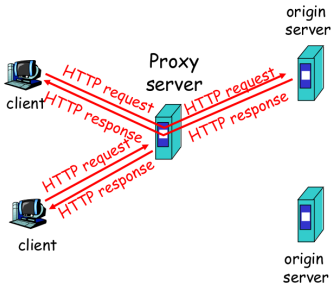
total = 2RTT+transmit time



Web caches (proxy server) and CDN

Goal: satisfy client request without involving origin server

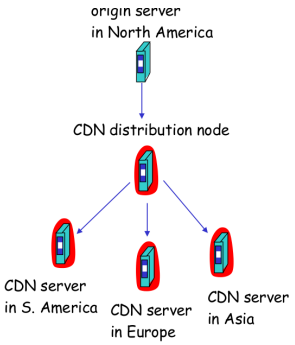
- user sets browser: Web accesses via cache
- browser sends all HTTP requests to cache
  - If object in cache: cache returns object
  - else cache requests object from origin server, then returns object to client



- The content providers (e.g., foxnews.com) own the original server

Content replication

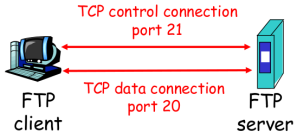
- CDN company (e.g., Akamai) installs hundreds of CDN servers throughout Internet
  - in lower-tier ISPs, close to users
- CDN replicates its customers' content in CDN servers. When provider updates content, CDN updates servers



2.3 FTP

FTP: separate control, data connections

- FTP client contacts FTP server at port 21, specifying TCP as transport protocol
- Client obtains authorization over control connection
- Client browses remote directory by sending commands over control connection.
- When server receives a command for a file transfer, the server opens a TCP data connection to client
- After transferring one file, server closes connection.



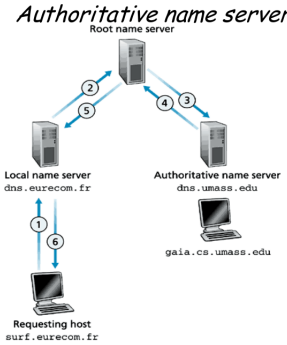
- Server opens a second TCP data connection to transfer another file.
- Control connection: "out of band"
- FTP server maintains "state": current directory, earlier authentication

2.4 DNS

DNS: Domain Name System

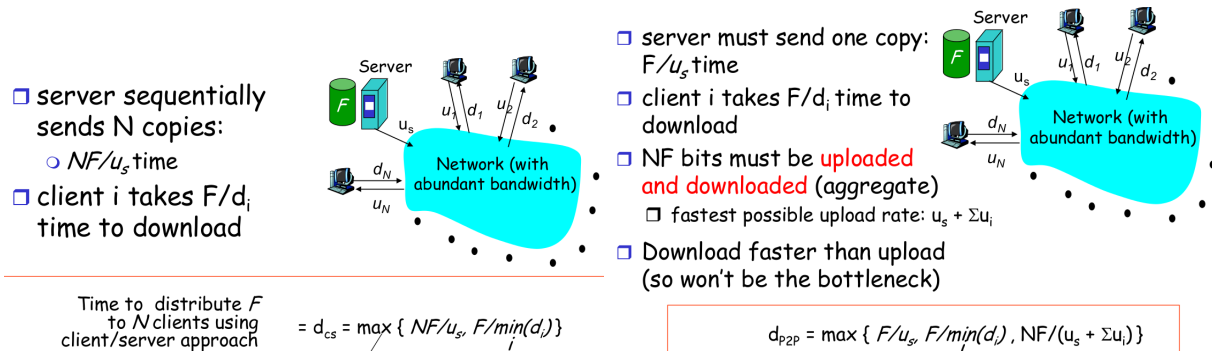
- Maps between a host's name and its IP address
- Other functions
  - host and mail server aliasing (with multiple names)
  - Load balancing with replicated web servers (one name maps to a set of IP addresses
- *distributed database* implemented with a hierarchy and caching

- *Local (default), Root and Authoritative name servers*



2.5 P2P applications

File distribution time: server-client vs. P2P



Optimistically Unchoke

- Alice sends chunks to four neighbors currently sending her chunks at the highest rate
- Re-evaluate top 4 every 10 secs
- Every 30 secs: randomly select another peer, starts sending chunks
- Newly chosen peer may join top 4

3 Transport Layer

3.1 Transport-layer services

Internet transport-layer protocols

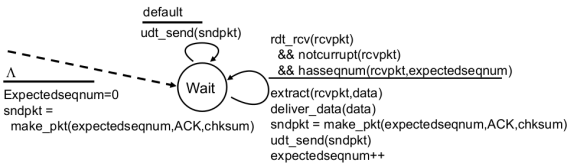
- Reliable, in-order delivery (TCP)
  - congestion control
  - flow control
  - connection setup
- Unreliable, unordered delivery (UDP)
  No-frills extension of "best-effort" IP

3.2 Principles of reliable data transfer

Go-Back-N Maximum window size is $N-1$

Sender:
- k-bit seq # in pkt header
- "window" of up to N, consecutive unack'ed pkts allowed

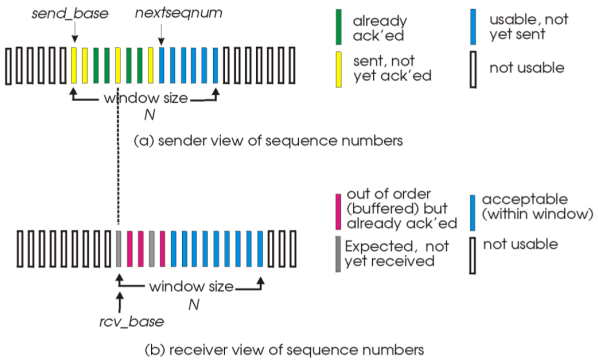
- ACK(n): ACKs all pkts up to, including seq # n - "cumulative ACK"
  - may receive duplicate ACKs (see receiver)
- timer for each in-flight pkt
- *timeout(n):* retransmit pkt n and all higher seq # pkts in window



ACK-only: always send ACK for correctly-received pkt with highest *in-order* seq #
- may generate duplicate ACKs
- need only remember **expectedseqnum**

out-of-order pkt:
- discard (don't buffer) -> only one pkt buffered (for app)!
- Re-ACK pkt with highest in-order seq #

Selective Repeat Maximum window size is $N/2$



sender

data from above :
- if next available seq # in window, send pkt

timeout(n):
- resend pkt n, restart timer

ACK(n) in [sendbase,sendbase+N]:
- mark pkt n as received
- if n is the smallest unACKed pkt, advance window base to next unACKed seq #

receiver

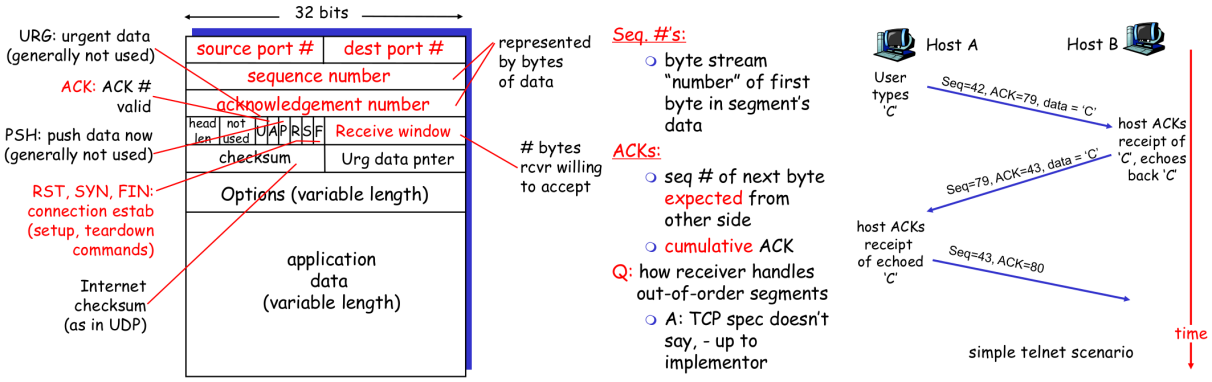
pkt n in [rcvbase, rcvbase+N-1]
- send ACK(n)
- out-of-order: buffer
- in-order: deliver (also deliver buffered, in-order pkts), advance window to next not-yet-received pkt

pkt n in [rcvbase-N,rcvbase-1]
- ACK(n): needed so the sender window can advance!

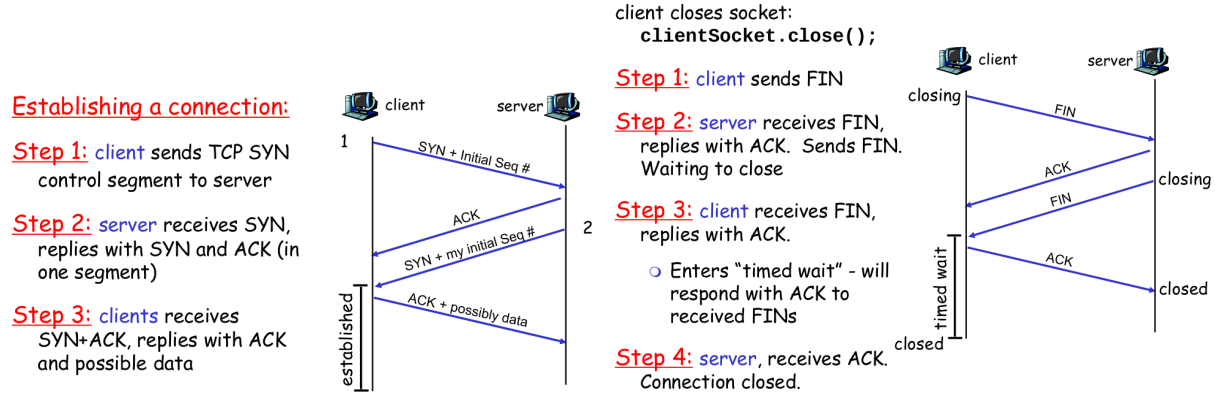
otherwise:
- ignore

3.3 Connection-oriented transport: TCP

TCP segment structure



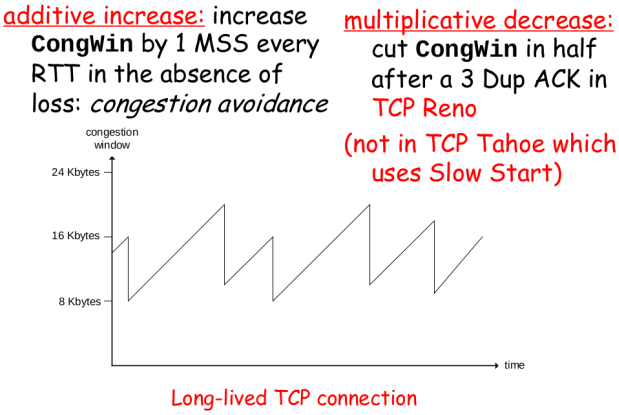
TCP Connection Establishment (3-way) and Close



3.4 TCP congestion control

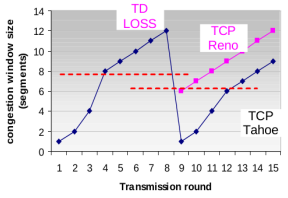
Congestion Too many sources sending too much data too fast for network to handle

TCP AIMD



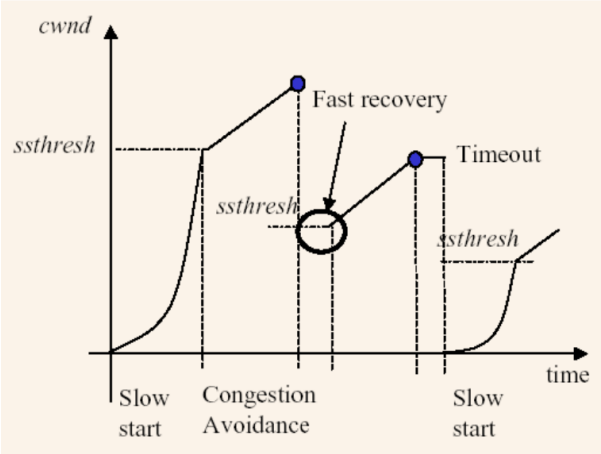
Fast Retransmit (Reno)

- After a TD loss:
  - **CongWin** cut in 1/2
  - window then grows linearly (congestion avoidance)
- But after a TO loss:
  - **CongWin** set to 1 MSS (slow start)
  - window then grows exponentially
  - to a new threshold, then grows linearly (as in TD)



Philosophy:

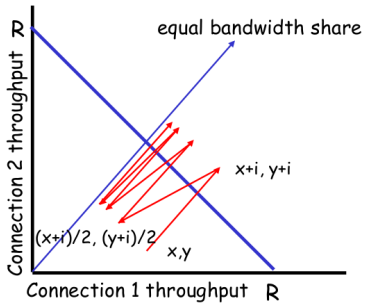
- 3 dup ACKs indicates network capable of delivering some segments
- timeout before 3 dup ACKs is "more alarming"



Why is TCP fair?

Two competing sessions:

- Additive increase in throughput from (x,y) with slope of 1
- multiplicative decrease in throughput to ({x+i}/2, {y+i}/2)

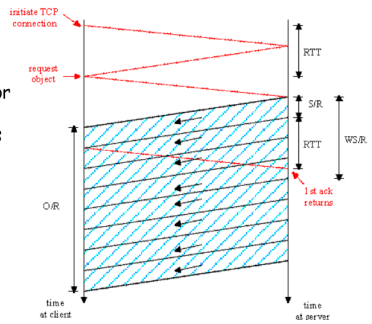


TCP Delay R link rate, S MSS size, O object/file size, W window size

First case:

WS/R > RTT + S/R: ACK for first segment in window returns before window's worth of data sent

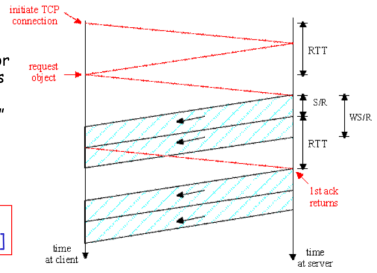
delay = 2RTT + O/R



Second case:

- WS/R < RTT + S/R: wait for ACK after sending window's worth of data sent
- "gap" between two "rounds" is S/R+RTT - WS/R
- Let K = O / WS be the number of rounds
- There are K-1 gaps

delay = 2RTT + O/R + (K-1)[S/R + RTT - WS/R]



4 Network Layer

4.1 Introduction

Key Network-Layer Functions

- *routing:* determine route taken by packets from source to dest.
  - *Routing algorithms*
- *connection setup:* in some (VC) networks only
- *forwarding:* move packets from router's input to appropriate router output

analogy:

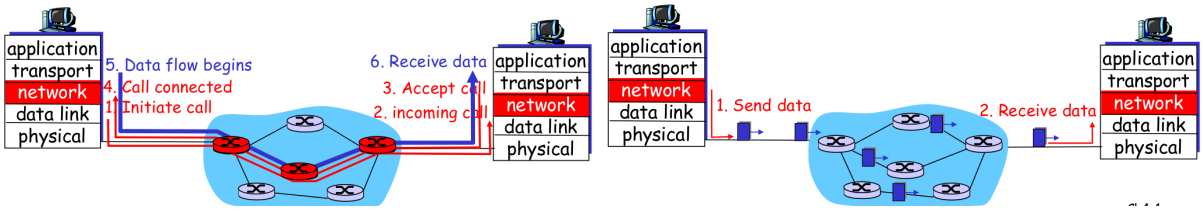
- routing: process of planning trip from source to dest
- connection setup: process of having police setting up road for parades
- forwarding: process of getting through single interchange

4.2 Virtual circuit and datagram networks

Virtual Circuits vs. Datagram Networks

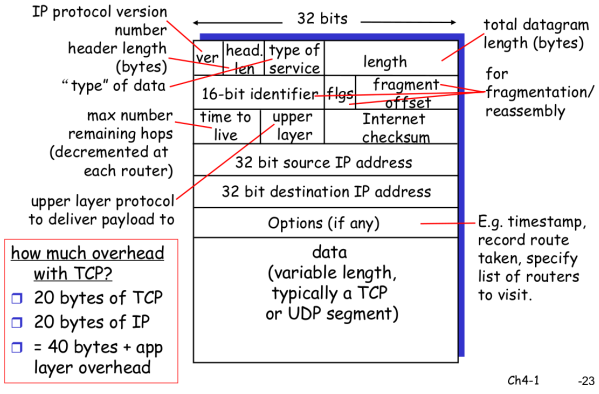
- used to setup, maintain teardown VC
- not used in original IP, but
- supported by Generalized Multi-Protocol Label Switching (G-MPLS)
  - label switched paths (LSPs) = VCs

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets forwarded using destination host address
  - packets between same source-dest pair may take different paths

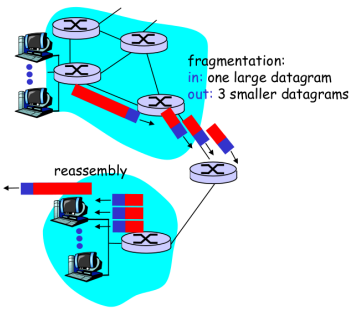


4.3 IP: Internet Protocol

IP datagram format



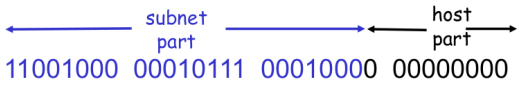
- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify,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



IP addressing: CIDR

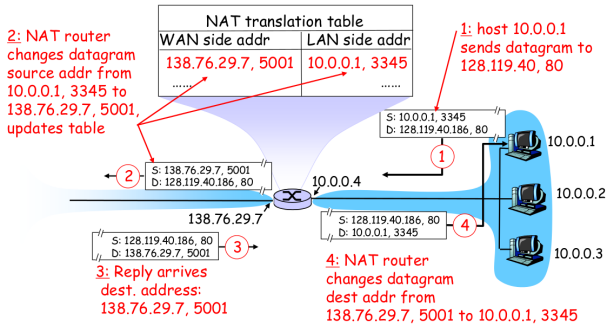
Now: CIDR: Classless InterDomain Routing

- subnet portion of address of arbitrary length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address



200.23.16.0/23

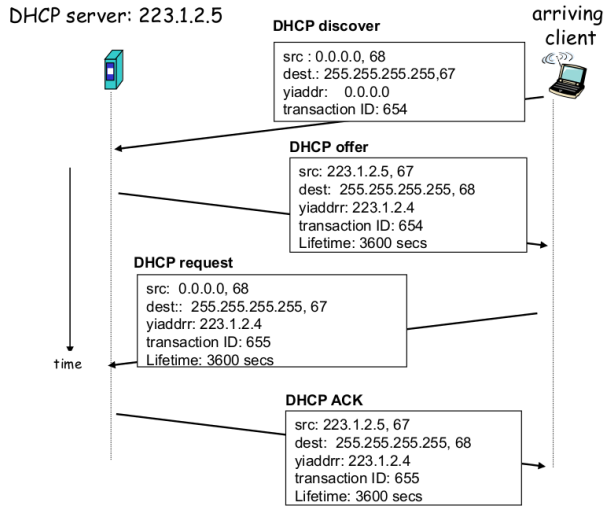
NAT: Network Address Translation



Implementation: NAT router must:

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

DHCP client-server scenario



Goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- Can renew its lease on address in use
- Allows reuse of addresses (only hold address while connected an "on"
- Support for mobile users who want to join network (more shortly)

DHCP overview:

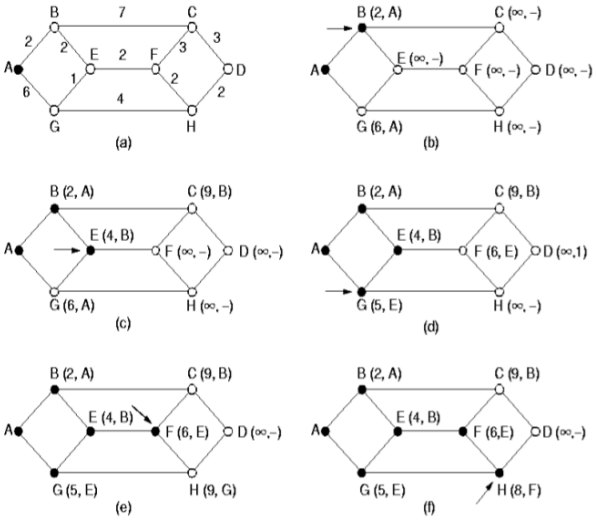
- host broadcasts "DHCP discover" msg
- DHCP server responds with "DHCP offer" msg
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

Client has to send DHCP request because it probably receive multiple DHCP offers from different DHCP servers, so that other DHCP servers can know which offer the client is taking, and can withdraw their offers.

4.4 Routing algorithms

Dijkstra algorithm

```
1  Initialization:
2   N' = {A}
3   for all nodes v
4     if v adjacent to A
5       then D(v) = c(A,v)
6       else D(v) = infinity          All neighboring nodes of those in N '
7
8  Loop
9    find w not in N' such that D(w) is a minimum
10   add w to N'
11   update D(v) for all v adjacent to w and not in N':
12     D(v) = min( D(v), D(w) + c(w,v) )
13   /* new cost to v is either old cost to v or the known
14    shortest path cost to w plus the cost from w to v */
15  until all nodes are in N'
```



Distance Vector

- $D_x(y)$ = *estimate* of least cost from x to y
- Distance vector: $D_x = [D_x(y): y \in N]$
- Node x knows cost to each neighbor v: c(x,v)
- Node x maintains $D_x = [D_x(y): y \in N]$
- Node x also receives its neighbors' distance vectors
  - For each neighbor v, x maintains $D_v = [D_v(y): y \in N]$

iterative:
- continues until no changes done, and no info. exchange
- *self-terminating*: no "signal" to stop

asynchronous:
- nodes need *not* exchange info, or iterate in lock step!

distributed:
- each node communicates its DV Changes *only* with directly-attached neighbors
  - neighbors then notify their neighbors if necessary

Each node:

wait for (change in local link cost of msg from neighbor)
↓
recompute estimates
↓
if DV to any dest has changed, *notify* neighbors

Hierarchical Routing

- aggregate routers into regions, "autonomous systems" (AS)
- routers in same AS run same routing protocol
  - "intra-AS" routing protocol
  - routers in different AS can run different intra-AS routing protocol

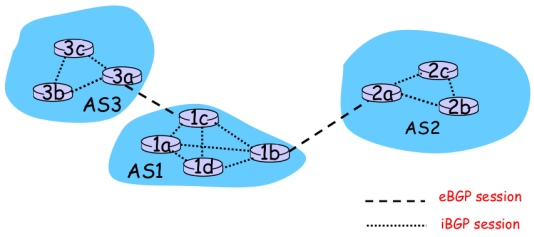
gateway routers
- special routers in AS
- run intra-AS routing protocol with all other routers in AS
- *also* responsible for routing to destinations outside AS
  - run *inter-AS routing* protocol with other gateway routers

4.5 Routing in the Internet

Intra-AS and inter-AS Routing

- Also known as Interior Gateway Protocols (IGP)
- Most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol (DV-based)
  - OSPF: Open Shortest Path First (LSA-based)
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)

- Pairs of routers (BGP peers) exchange routing info over semi-permanent TCP connections: BGP sessions
- Note that BGP sessions do not correspond to physical links.
- When AS2 advertises a prefix to AS1, AS2 is *promising* it will forward any datagrams destined to that prefix towards the prefix.
  - AS2 can aggregate prefixes in its advertisement



Why different Intra- and Inter-AS routing ?

Policy:
- Inter-AS: admin wants control over how its traffic routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

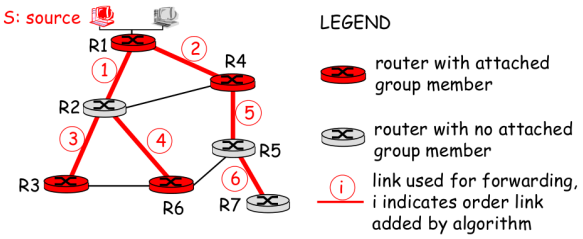
Scale:
- hierarchical routing saves table size, reduced update traffic

Performance:
- Intra-AS: can focus on performance
- Inter-AS: policy may dominate over performance

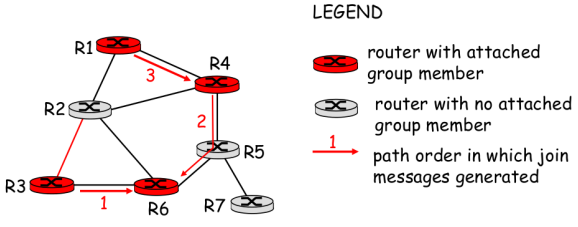
4.6 Broadcast and multicast routing

Multicast Routing

- mcast forwarding tree: tree of shortest path routes from source to all receivers
  - Dijkstra's algorithm



Suppose R6 chosen as center:



5 Link Layer

5.1 Introduction and services

Link Layer Services

- Framing, link access:
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, dest
    - different from IP address!
- Reliable delivery between adjacent nodes
  - we learned how to do this already (chapter 3)!
  - seldomly used on low bit error link (fiber, some twisted pair)
  - wireless links: high error rates
    - Q: why both link-level and end-end reliability?
- *Flow Control:* (also discussed earlier)
  - 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

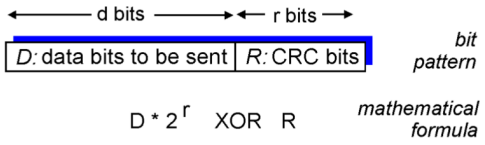
5.2 Error detection and correction

Hamming distance The (min) number of bits that differ, that is, need to be flipped/inverted to change from one codeword to the other.

If the minimum Hamming distance between any two valid codewords is d , then detect any error up to $d-1$ bits, and can correct any error up to $(d-1)/2$ (if d is odd), or $d/2-1$ (if d is even) bits.

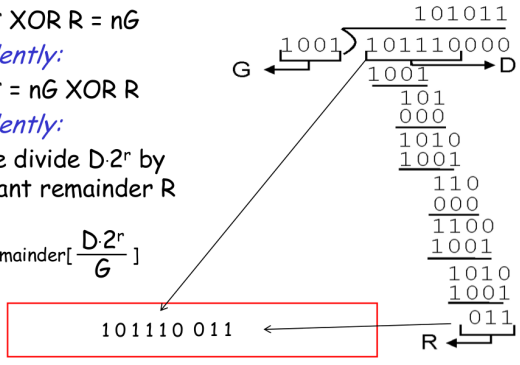
Checksumming: Cyclic Redundancy Check

- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
  - can detect all burst errors less than r+1 bits
- widely used in practice



Want:
$D \cdot 2^r$ XOR R = nG
equivalently:
$D \cdot 2^r = nG$ XOR R
equivalently:
if we divide $D \cdot 2^r$ by G, want remainder R

$R = \text{remainder}\left[\frac{D \cdot 2^r}{G}\right]$



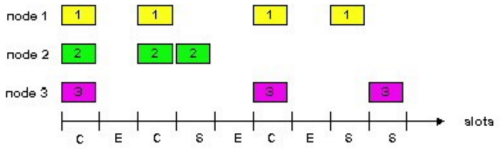
5.3 Multiple access protocols

MAC Protocols: a 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

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

Efficiency is the long-run fraction of successful slots when there are many nodes, each with many frames to send

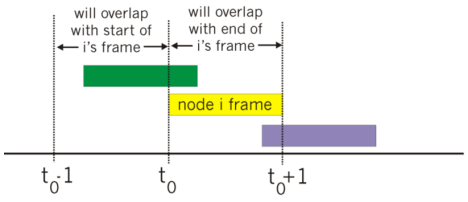
- Suppose N nodes with many frames to send, each transmits in slot with probability p
- prob that node 1 has success in a slot = $p(1-p)^{N-1}$
- prob that any node has a success = $Np(1-p)^{N-1}$

- For max efficiency with N nodes, 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 $1/e = .37$

At best: channel used for useful transmissions 37% of time!

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 $[t_0-1,t_0+1]$



P(success by given node) = P(node transmits) ·
P(no other node transmits in $(t_0-1,t_0]$ ·
P(no other node transmits in $[t_0,t_0+1]$
$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$
$= p \cdot (1-p)^{2(N-1)}$

... choosing optimum p and then letting n -> infty ...

Even worse ! $= 1/(2e) = .18$

CSMA (Carrier Sense Multiple Access)

- *1*-persistent:
  - continuously senses the channel until it's idle.
  - transmits whenever idle.
  - may still have collisions.
- nonpersistent:
  - if the channel is busy, waits a random period time before sensing it again
  - transmits if idle
- *p*-persistent: for *slotted* channels
  - if busy initially, senses the channel again for the next slot (like *1*-persistent).
  - if idle, transmits with probability *p* (< 1) (at the beginning of the next slot) or defers with probability *q = 1-p*.
  - if it didn't transmit, senses the channel in the next slot; if idle again, repeat the process (either transmits or defers); if busy, waits a random period time before sensing it again (like non-persistent)

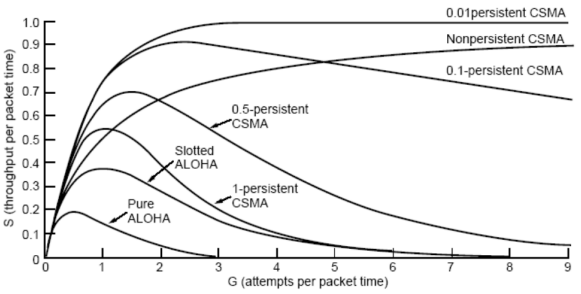


Fig. 4-4. Comparison of the channel utilization versus load for various random access protocols.

CSMA/CD Collision Detection Delay

CSMA/CD: carrier sensing, deferral as in CSMA
- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: receiver shut off while transmitting
- human analogy: the polite conversationalist

- One way propagation delay = τ

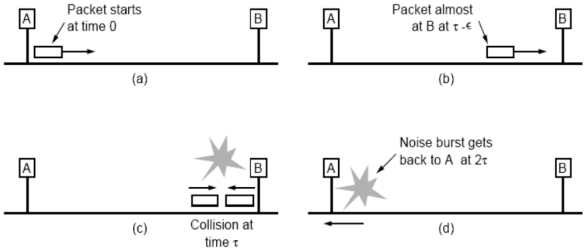
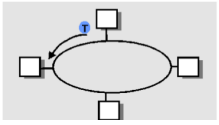


Fig. 4-22. Collision detection can take as long as 2τ.

Taking Turns MAC protocols

Polling:
- master node "invites" slave nodes to transmit in turn
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)

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



5: DataLink Layer 8-

Slot 0 for all stations under node *1*. Repeat if no collisions;
- Otherwise, *Slot 1* for those under node *2* (left subtree).
- If successful, *Slot 2* for those under node *3* (right subtree);
- Else, *Slot 2* for those under node *4* (the left sub-subtree).

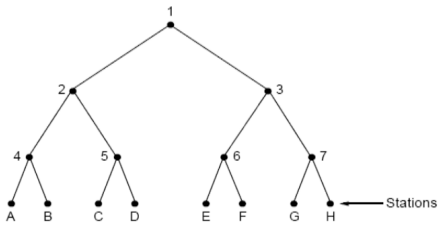


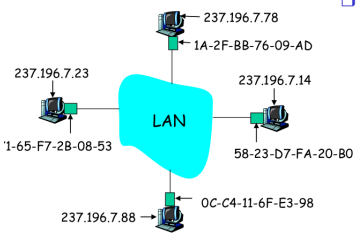
Fig. 4-9. The tree for eight stations.

5.4 Link-Layer Addressing

MAC (or LAN or physical or Ethernet) address used to get datagram from one interface to another physically-connected interface (same network)

ARP: Address Resolution Protocol

Question: how to determine MAC address of a host B knowing B's IP address?



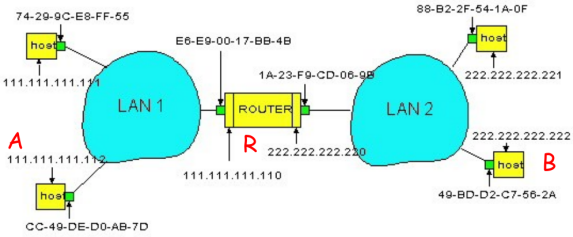
- Each IP node (Host, Router) on LAN has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes
  < IP address; MAC address; TTL>
  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

- A wants to send datagram to B, and 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 machines 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

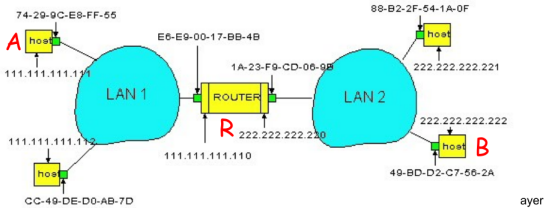
Routing to another LAN

walkthrough: send **datagram** from A to B via R
assume A know's B IP address



- In routing table at source Host A, find router R 111.111.111.110
- In ARP table at source, find R's MAC address E6-E9-00-17-BB-4B, etc..

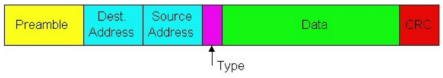
- A creates datagram with source A, destination B
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram
- A's adapter sends frame
- R's adapter receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B's MAC address
- R creates frame containing A-to-B IP datagram sends to B



5.5 Ethernet

Ethernet Frame Structure

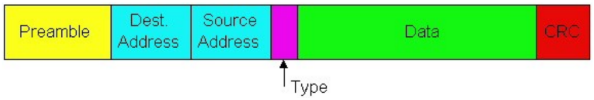
Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame



Preamble:

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

- Addresses: 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to net-layer protocol
  - otherwise, adapter discards frame
- Type: indicates the higher layer protocol (mostly IP but others may be supported such as Novell IPX and AppleTalk)
- CRC: checked at receiver, if error cannot be recovered, the frame is simply dropped



Ethernet CSMA/CD algorithm

1. Adaptor receives datagram from net layer & creates frame
2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame !
4. If adapter detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, adapter enters **exponential backoff**: after the *m*-th collision, adapter chooses a K at random from {0,1,2,…,2^m-1}. Adapter waits K·512 bit times and returns to Step 2

5: DataLink Layer 8-58

Jam Signal: make sure all other transmitters are aware of collision; 48 bits

Bit time: .1 microsec for 10 Mbps Ethernet ; for K=1023, wait time is about 50 msec

See/interact with Java applet on AWL Web site: highly recommended !

Exponential Backoff:

- *Goal*: adapt retransmission attempts to estimated current load
  - 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}

CSMA/CD efficiency

- T_{prop} = max prop between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame (with max data)

$$\text{efficiency} = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- Efficiency goes to 1 as t_{prop} goes to 0
- Goes to 1 as t_{trans} goes to infinity
- Much better than ALOHA, but still decentralized, simple, and cheap

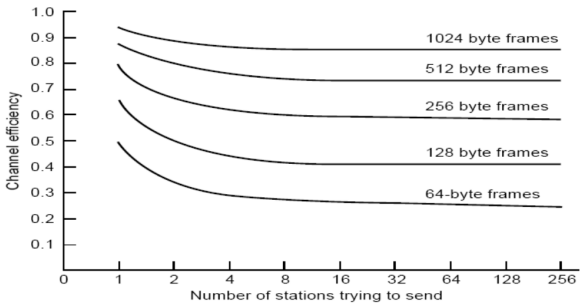
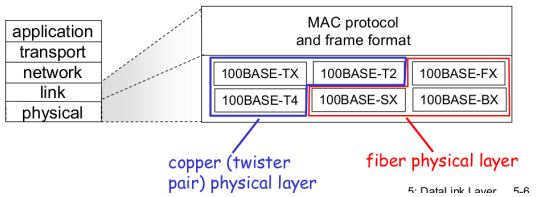


Fig. 4-23. Efficiency of 802.3 at 10 Mbps with 512-bit slot times.

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, 10G bps
  - different physical layer media: fiber, cable



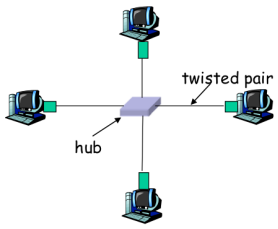
5: DataLink Layer 5-6

5.6 Interconnections: Hubs and switches

Hubs and Switches

Hubs are essentially physical-layer repeaters:

- bits coming from one link go out all other links
- at the same rate
- no frame buffering
- no CSMA/CD at hub: adapters detect collisions
- provides net management functionality



- 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

Switch: frame filtering/forwarding

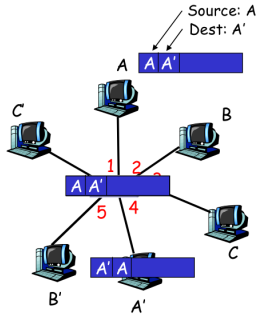
When frame received:

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

forward on all but the interface on which the frame arrived

Self-learning, forwarding: example

- frame destination A' unknown:
  Flood but learns about A
- destination A location now known:
  selective send
  learns about A'

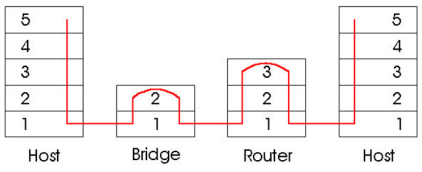


MAC addr	interface	TTL
A	1	60
A'	4	60

Switch table (initially empty)

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



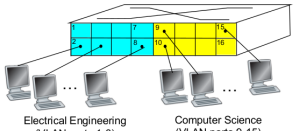
VLANs

VLANs

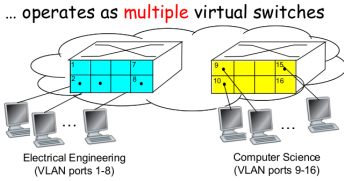
Virtual Local Area Network

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

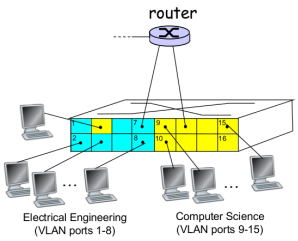
Port-based VLAN: switch ports grouped (by switch management software) so that *single* physical switch .....



... operates as multiple virtual switches



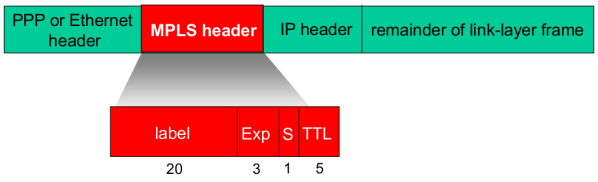
- *traffic isolation:* frames to/from ports 1-8 can *only* reach ports 1-8
  - can also define VLAN based on MAC addresses of endpoints, rather than switch port
- dynamic membership: ports can be dynamically assigned among VLANs
- forwarding between VLANS: done via routing (just as with separate switches)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



5.7 MPLS

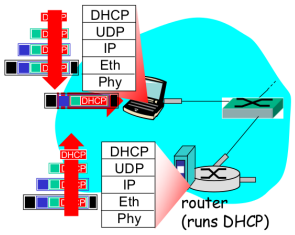
Multiprotocol label switching (MPLS)

- initial goal: speed up IP forwarding by using fixed length label (instead of IP address) to do forwarding
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!

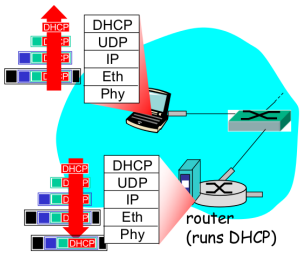


- a.k.a. label-switched router
- forwards packets to outgoing interface based only on label value (don't inspect IP address)
  - MPLS forwarding table distinct from IP forwarding tables
- signaling protocol needed to set up forwarding
  - RSVP-TE
  - forwarding possible along paths that IP alone would not allow (e.g., source-specific routing) !!
  - use MPLS for traffic engineering
- must co-exist with IP-only routers

5.8 A day in the life



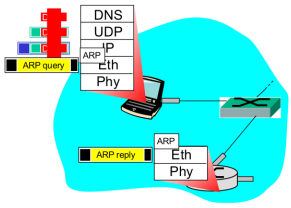
- connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use *DHCP*
- DHCP request **encapsulated** in **UDP**, encapsulated in **IP**, encapsulated in **802.1** Ethernet
- Ethernet frame **broadcast** (dest: FFFFFFFFFFFF) on LAN, received at router running **DHCP** server
- Ethernet **demux'ed** to IP demux'ed, UDP demux'ed to DHCP



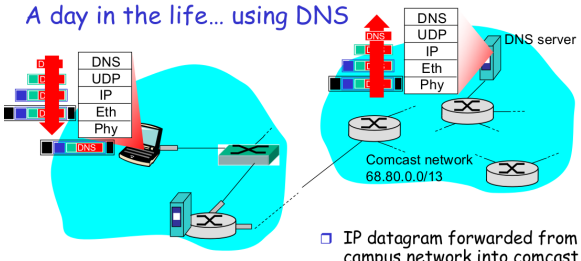
- DHCP server formulates *DHCP ACK* containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulation at DHCP server, frame forwarded (**switch learning**) through LAN, demultiplexing at client
- DHCP client receives DHCP ACK reply

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

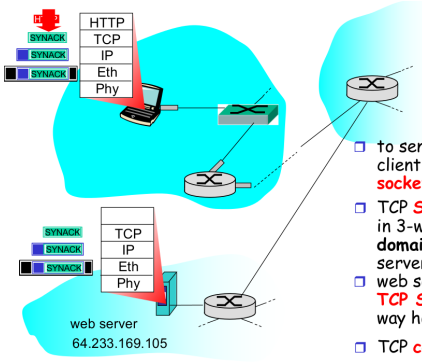
A day in the life... using DNS



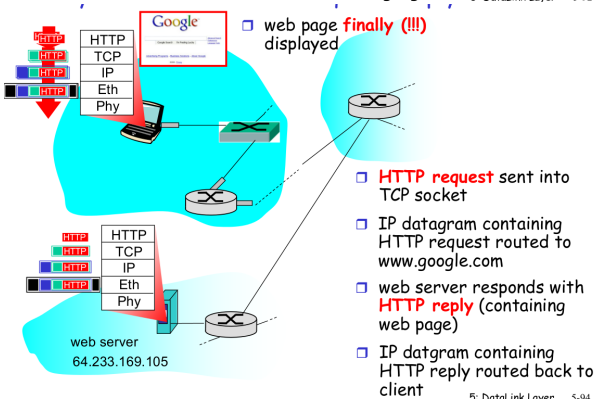
- before sending *HTTP* request, need IP address of www.google.com: *DNS*
- DNS query created, encapsulated in UDP, encapsulated in IP, encasulated in Eth. In order to send frame to router, need MAC address of router interface: **ARP**
- **ARP query** broadcast, received by router, which replies with **ARP reply** giving MAC address of router interface
- client now knows MAC address of first hop router, so can now send frame containing DNS query



- IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router
- IP datagram forwarded from campus network into comcast network, routed (tables created by **RIP, OSPF, IS-IS** and/or **BGP** routing protocols) to DNS server
- demux'ed to DNS server
- DNS server replies to client with IP address of www.google.com



- to send HTTP request, client first opens **TCP socket** to web server
- TCP **SYN segment** (step 1 in 3-way handshake) **inter-domain routed** to web server
- web server responds with **TCP SYNACK** (step 2 in 3-way handshake)
- TCP **connection established!**



- web page **finally (!!!)** displayed
- **HTTP request** sent into TCP socket
- IP datagram containing HTTP request routed to www.google.com
- web server responds with **HTTP reply** (containing web page)
- IP datgram containing HTTP reply routed back to client

6 Wireless and Mobile Networks

6.1 Introduction

6.2 Wireless Links Characteristics

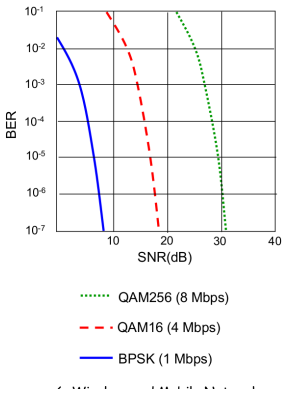
Wireless Link Characteristics

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 ad destination at slightly different times

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

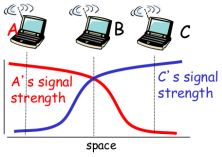
- SNR: signal-to-noise ratio
  - larger SNR - easier to extract signal from noise (a "good thing")
- *SNR versus BER tradeoffs*
  - *given physical layer:* increase power -> increase SNR->decrease BER
  - *given SNR:* choose physical layer that meets BER requirement, giving highest thruput
    - SNR may change with mobility: dynamically adapt physical layer (modulation technique, rate)



6.3 IEEE 802.11 Wireless LANs (wifi)

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
  - goal: *avoid collisions:* CSMA/C(ollision)A(voidance)

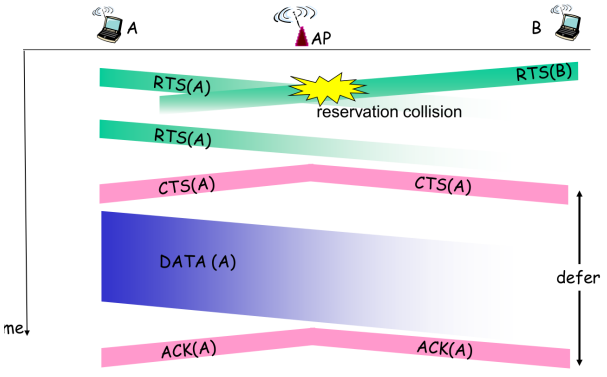


Collision Avoidance: RTS-CTS exchange

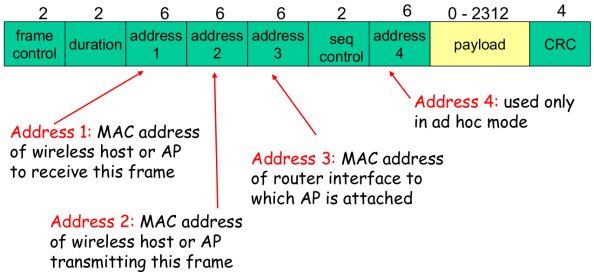
idea: allow sender to "reserve" channel rather than random access of data frames: avoid collisions of long data frames

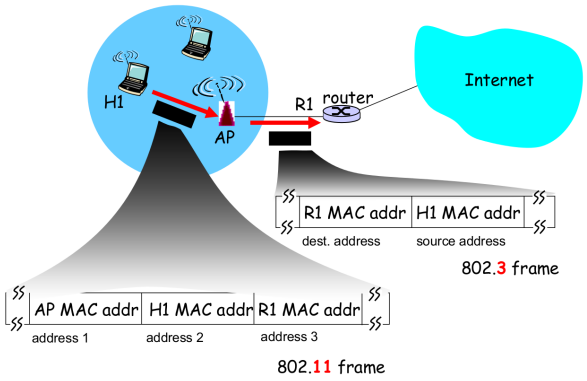
- sender first transmits *small* request-to-send (RTS) packets to BS using CSMA
  - RTSs may still collide with each other (but they're short)
- BS broadcasts clear-to-send CTS in response to RTS
- CTS heard by all nodes
  - sender transmits data frame
  - other stations defer transmissions

avoid data frame collisions completely using small reservation packets!



802.11 frame: addressing

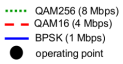


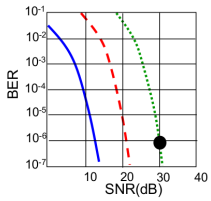


802.11: advanced capabilities

Rate Adaptation

- base station, mobile dynamically change transmission rate (physical layer modulation technique) as mobile moves, SNR varies





1. SNR decreases, BER increase as node moves away from base station
2. When BER becomes too high, switch to lower transmission rate but with lower BER

Power Management

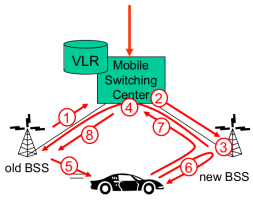
- node-to-AP: "I am going to sleep until next beacon frame"
  - AP knows not to transmit frames to this node
  - node wakes up before next beacon frame
- beacon frame: contains list of mobiles with AP-to-mobile frames waiting to be sent
  - node will stay awake if AP-to-mobile frames to be sent; otherwise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

6.4 Cellular Internet Access

6.5 Principles: addressing and routing to mobile users

6.6 Mobile IP

GSM: handoff with common MSC



1. old BSS informs MSC of impending handoff, provides list of 1+ new BSSs
2. MSC sets up path (allocates resources) to new BSS
3. new BSS allocates radio channel for use by mobile
4. new BSS signals MSC, old BSS: ready
5. old BSS tells mobile: perform handoff to new BSS
6. mobile, new BSS signal to activate new channel
7. mobile signals via new BSS to MSC: handoff complete. MSC reroutes call

8 MSC-old-BSS resources released

Mobility: GSM versus Mobile IP

GSM element	Comment on GSM element	Mobile IP element
Home system	Network to which mobile user's permanent phone number belongs	Home network
Gateway Mobile Switching Center, or "home MSC". Home Location Register (HLR)	Home MSC: point of contact to obtain routable address of mobile user. HLR: database in home system containing permanent phone number, profile information, current location of mobile user, subscription information	Home agent
Visited System	Network other than home system where mobile user is currently residing	Visited network
Visited Mobile services Switching Center. Visitor Location Record (VLR)	Visited MSC: responsible for setting up calls to/from mobile nodes in cells associated with MSC. VLR: temporary database entry in visited system, containing subscription information for each visiting mobile user	Foreign agent
Mobile Station Roaming Number (MSRN), or "roaming number"	Routable address for telephone call segment between home MSC and visited MSC, visible to neither the mobile nor the correspondent.	Care-of-address

7 Multimedia and Quality of Service

7.1 Multimedia Networking Applications

MM Networking Applications

Classes of MM *Streaming* applications:

1) stored streaming (e.g., Video on Demand)
2) live streaming (e.g., live concert, webinar)
3) interactive, real-time (e.g., video chat)

Jitter is the variability in *inter-packet* delays within the same stream

Fundamental characteristics:

- typically **delay sensitive**
  - end-to-end delay
  - delay *jitter*
- **loss tolerant**: infrequent losses cause minor glitches
- antithesis of data, which are loss *intolerant* but delay *tolerant*.

7.2 Streaming stored audio and video

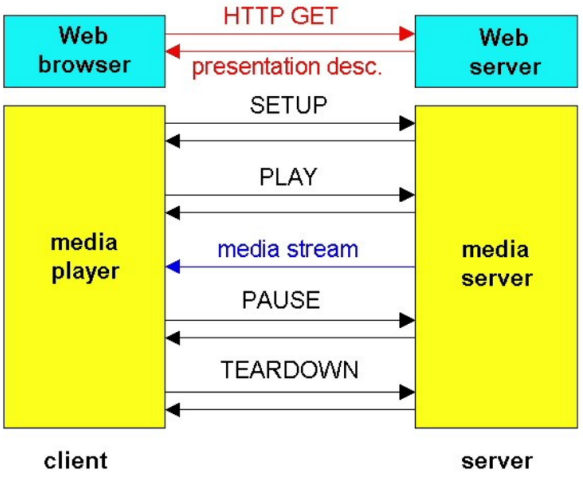
RTSP: out of band control

FTP uses an "out-of-band" control channel:

- file transferred over one TCP connection.
- control info (directory changes, file deletion, rename) sent over separate TCP connection
- "out-of-band", "in-band" channels use different port numbers

RTSP messages also sent out-of-band:

- RTSP control messages use different port numbers than media stream: out-of-band.
  - port 554
- media stream is considered "in-band".



Internet Phone: Packet Loss and Delay

- network loss: IP datagram lost due to network congestion (router buffer overflow)
- delay loss: IP datagram arrives too late for playout at receiver
  - delays: processing, queueing in network; end-system (sender, receiver) delays
  - typical maximum tolerable delay: 400 ms
- loss tolerance: depending on voice encoding, losses concealed, packet loss rates between 1% and 10% can be tolerated.

- receiver attempts to playout each chunk exactly q msecs after chunk was generated.
  - chunk has time stamp t: play out chunk at t+q .
  - chunk arrives after t+q: data arrives too late for playout, data discarded/"lost"
- tradeoff in choosing q:
  - *large q:* less packet loss
  - *small q:* better interactive experience
  - Need an *adaptive* playout schedule

Recovery from packet loss

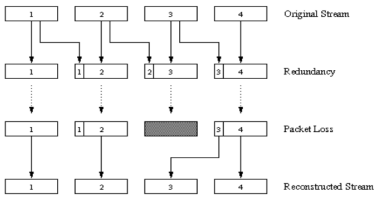
Forward Error Correction (FEC): simple scheme

- for every group of *n* chunks create redundant chunk by exclusive OR-ing *n* original chunks
- send out *n+1* chunks, increasing bandwidth by factor *1/n*.
- can reconstruct original *n* chunks if at most one lost chunk from *n+1* chunks

- playout delay: enough time to receive all n+1 packets
- tradeoff:
  - increase n, less bandwidth waste ☺
  - increase n, longer playout delay ☹
  - increase n, higher probability that 2 or more chunks will be lost ☹

2nd FEC scheme

- "piggyback lower quality stream"
- send lower resolution audio stream as redundant information
- e.g., nominal stream PCM at 64 kbps and redundant stream GSM at 13 kbps.



- whenever there is non-consecutive loss, receiver can conceal the loss.
- can also append (n-1)st and (n-2)nd low-bit rate chunk

7.3 Protocols for real-time interactive applications

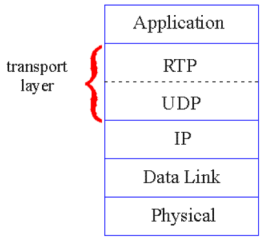
Real-Time Protocol (RTP)

- RTP specifies packet structure for packets carrying audio, video data
- RFC 3550
- RTP packet provides
  - payload type identification
  - packet sequence numbering
  - time stamping

- RTP runs in end systems
- RTP packets encapsulated in UDP segments
- interoperability: if two Internet phone applications run RTP, then they may be able to work together

RTP libraries provide transport-layer interface that extends UDP:

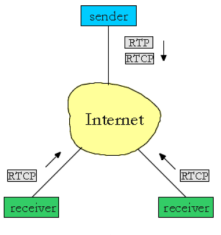
- port numbers, IP addresses
- payload type identification
- packet sequence numbering
- time-stamping



Real-Time Control Protocol (RTCP)

- works in conjunction with RTP.
- each participant in RTP session periodically transmits RTCP control packets to all other participants.
- each RTCP packet contains sender and/or receiver reports
  - report statistics useful to application: # packets sent, # packets lost, interarrival jitter, etc.

- feedback can be used to control performance
  - sender may modify its transmissions based on feedback

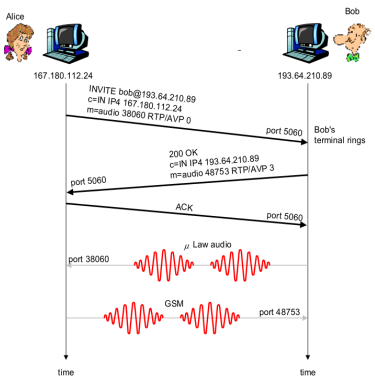


- RTCP can synchronize different media streams within a RTP session
- consider videoconferencing app for which each sender generates one RTP stream for video, one for audio.
- timestamps in RTP packets tied to the video, audio sampling clocks
  - *not* tied to a common wall-clock time

- each RTCP sender-report packet contains (for most recently generated packet in associated RTP stream):
  - timestamp of RTP packet
  - wall-clock time for when packet was created.
- receivers use association to synchronize playout of audio, video

SIP: Session Initiation Protocol

- Setting up a call, SIP provides mechanisms ..
  - for caller to let callee know she wants to establish a call
  - so caller, callee can agree on media type, encoding
  - to end call
- determine current IP address of callee:
  - maps mnemonic identifier to current IP address
- call management:
  - add new media streams during call
  - change encoding during call
  - invite others
  - transfer, hold calls

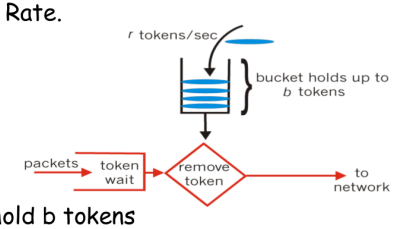


- Alice's SIP invite message indicates her port number, IP address, encoding she prefers to receive (PCM ulaw)
- Bob's 200 OK message indicates his port number, IP address, preferred encoding (GSM)
- SIP messages can be sent over TCP or UDP; here sent over RTP/UDP.
- default SIP port number is 5060.

7.4 Providing multiple classes of service

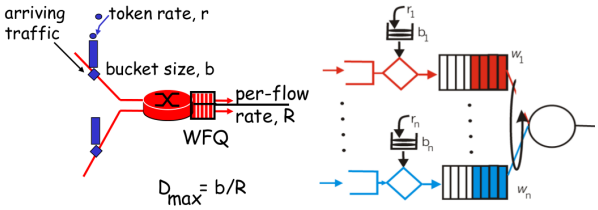
Policing Mechanisms

Token Bucket: limit input to specified Burst Size and Average Rate.



- bucket can hold b tokens
- tokens generated at rate *r token/sec* unless bucket full
- *over interval of length t: number of packets admitted less than or equal to (r t + b).*

- token bucket, WFQ combine to provide guaranteed upper bound on delay, i.e., *QoS guaranteed*



Call Admission

Arriving session must :

- declare its QOS requirement
  - R-spec: defines the QOS being requested
- characterize traffic it will send into network
  - T-spec: defines traffic characteristics
- signaling protocol: needed to carry R-spec and T-spec to routers (where reservation is required)
  - RSVP

connectionless (stateless) forwarding by IP routers + best effort service = no network signaling protocols in initial IP design

- New requirement: reserve resources along end-to-end path (end system, routers) for QoS for multimedia applications
- RSVP: Resource Reservation Protocol [RFC 2205]
  - " ... allow users to communicate requirements to network in robust and efficient way." i.e., signaling !
- earlier Internet Signaling protocol: ST-II [RFC 1819]
- Latest: Software defined networking and Openflow

8 Network Security

8.1 What is network security?

What is network security?

Confidentiality: only sender, intended receiver should "understand" message contents
- sender encrypts message
- receiver decrypts message

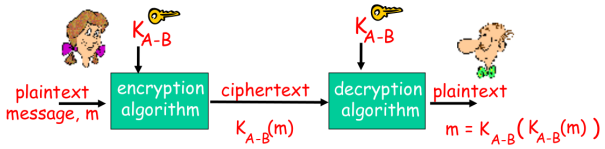
Authentication: sender, receiver want to confirm identity of each other

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

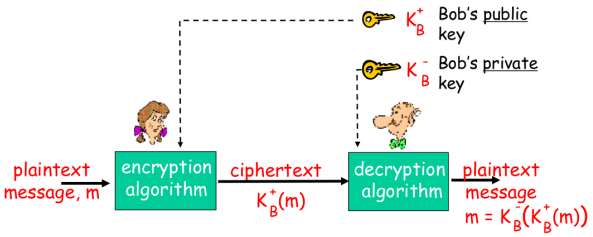
Access and availability: services must be accessible and available to users

8.2 Principles of cryptography

Symmetric vs. Public key cryptography



symmetric key crypto: Bob and Alice share know same (symmetric) key: K_{A-B}
- e.g., key is knowing substitution pattern in mono alphabetic substitution cipher
- Q: how do Bob and Alice agree on key value?



RSA: Choosing keys

1. Choose two large prime numbers p, q. (e.g., 1024 bits each)
2. Compute $n = pq$, $z = (p-1)(q-1)$
3. Choose e (with $e<n$) that has no common factors with z. (e, z are "relatively prime").
4. Choose d such that $ed-1$ is exactly divisible by z. (in other words: $ed \bmod z = 1$).
5. *Public* key is $\underbrace{(n,e)}_{K_B^+}$. *Private* key is $\underbrace{(n,d)}_{K_B^-}$.

0. Given (n,e) and (n,d) as computed above
1. To encrypt bit pattern, m, compute $c = m^e \bmod n$ (i.e., remainder when m^e is divided by n)
2. To decrypt received bit pattern, c, compute $m = c^d \bmod n$ (i.e., remainder when c^d is divided by n)

Magic happens! $m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$

RSA: another important property

The following property will be *very* useful later:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{use public key first, followed by private key}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{use private key first, followed by public key}}$$

Result is the same!

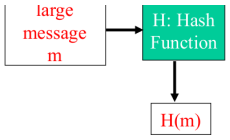
8.3 Message integrity

Message Digests

Computationally expensive to public-key-encrypt long messages

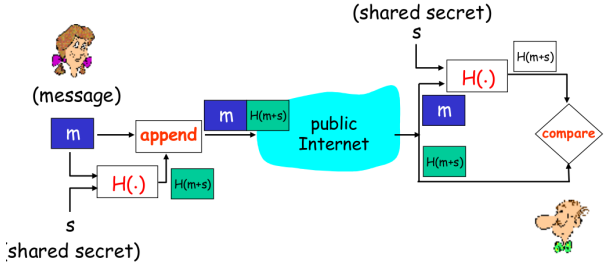
Goal: fixed-length, easy- to-compute digital "fingerprint"

- apply hash function H to *m*, get fixed size message digest, *H(m)*.



Hash function properties:

- many-to-1
- But given message digest x = H(m), it's infeasible to find m' that H(m') = H(m)
- Data integrity: cannot replace m with m'



Digital Signatures

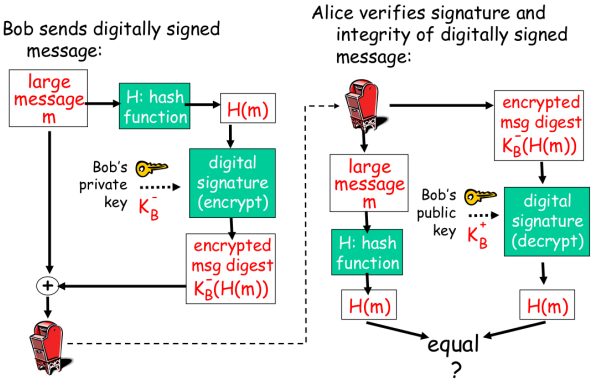
- suppose Alice receives msg m, digital signature $K_B^-(m)$
- Alice verifies m signed by Bob by applying Bob's public key K_B^+ to $K_B^-(m)$ then checks $K_B^+(K_B^-(m)) = m$.
- if $K_B^+(K_B^-(m)) = m$, whoever signed m must have used Bob's private key.

Alice thus verifies that:

- ✓ Bob signed m.
- ✓ No one else signed m.
- ✓ Bob signed m and not m'.

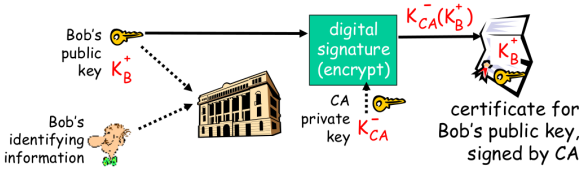
non-repudiation:

- ✓ Alice can take m, and signature $K_B^-(m)$ to court and prove that Bob signed m.

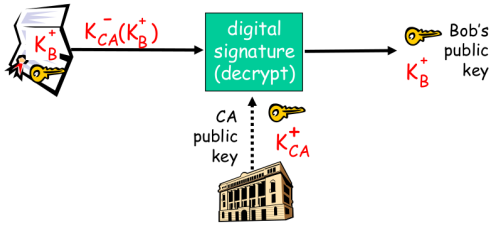


Certification Authorities

- Certification Authority (CA): binds public key to particular entity, E.
- E registers its public key with CA.
  - E provides "proof of identity" to CA.
  - CA creates certificate binding E to its public key.
  - certificate containing E's public key digitally signed by CA: CA says "This is E's public key."



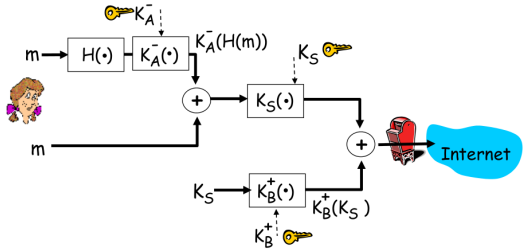
- when Alice wants Bob's public key:
  - gets Bob's certificate (Bob or elsewhere).
  - apply CA's public key to Bob's certificate, get Bob's public key



8.4 Securing e-mail

PGP

- Alice wants to provide secrecy, sender authentication, message integrity.



Alice uses three keys: her private key, Bob's public key, newly created symmetric key

Q: What does Bob have to do"?