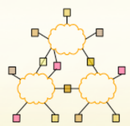
15-446 Distributed Systems Spring 2009



L-3 Networking 101

Today's Lecture

- Network Interface
- Link Layer
- Addressing/IP
- Routing
- TCP

Client-Server Paradigm

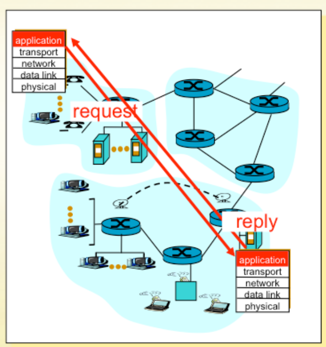
Typical network app has two pieces: *client* and *server*

Client:

- Initiates contact with server ("speaks first")
- Typically requests service from server,
- For Web, client is implemented in browser; for e-mail, in mail reader

Server:

- Provides requested service to client
- e.g., Web server sends requested Web page, mail server delivers e-mail



Transport Service Requirements of Common Apps

Application	Data loss	Bandwidth	Time Sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
web documents	no loss	elastic	no
real-time audio/ video	loss-tolerant	audio: 5Kb-1Mb video:10Kb-5Mb	yes, 100's msec
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few Kbps	yes, 100's msec
financial apps	no loss	elastic	yes and no

Other Requirements

- Network reliability
  - Network service must always be available
- Security: privacy, denial of service, authentication, …
- Scalability.
  - Scale to large numbers of users, traffic flows, …
- Manageability: monitoring, control, …

What Service Does an Application 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"

Bandwidth

- Some apps (e.g., multimedia) require minimum amount of bandwidth to be "effective"
- Other apps ("elastic apps") make use of whatever bandwidth they get

IP Service Model

- Low-level communication model provided by Internet
- Datagram
  - Each packet self-contained
    - All information needed to get to destination
    - No advance setup or connection maintenance
  - Analogous to letter or telegram

IPv4 Packet Format

0	4	8	12	16	19	24	28	31
version	HLen	TOS		Length				
Identifier				Flag	Offset			
TTL		Protocol		Checksum				
Source Address								
Destination Address								
Options (if any)								
Data								

Header (version through Options)

User Datagram Protocol(UDP): An Analogy

UDP	Postal Mail
• Single socket to receive messages • No guarantee of delivery • Not necessarily in-order delivery • Datagram – independent packets • Must address each packet	• Single mailbox to receive letters • Unreliable ☺ • Not necessarily in-order delivery • Letters sent independently • Must address each reply

Example UDP applications
Multimedia, voice over IP

Transmission Control Protocol (TCP): An Analogy

TCP	Telephone Call
• Reliable – guarantee delivery • Byte stream – in-order delivery • Connection-oriented – single socket per connection • Setup connection followed by data transfer	• Guaranteed delivery • In-order delivery • Connection-oriented • Setup connection followed by conversation

Example TCP applications
Web, Email, Telnet

9

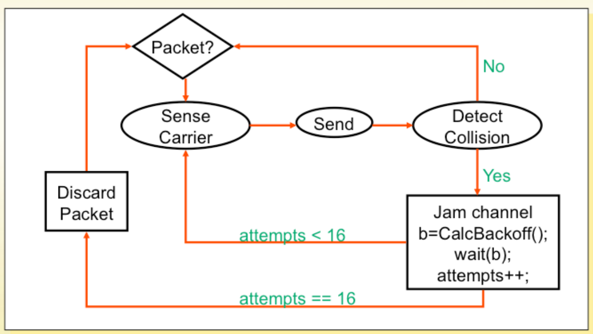
Today's Lecture

- Network Interface
- Link Layer
- Addressing/IP
- Routing
- TCP

10

Ethernet MAC (CSMA/CD)

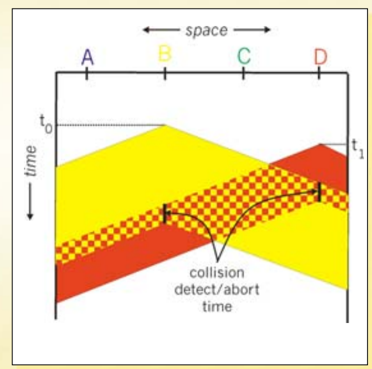
- Carrier Sense Multiple Access/Collision Detection



11

Minimum Packet Size

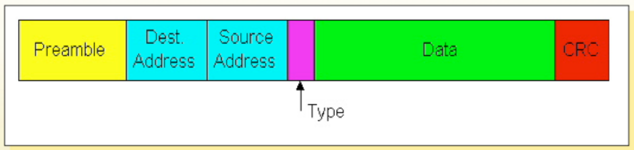
- What if two people sent really small packets
  - How do you find collision?



12

Ethernet Frame Structure

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



13

Ethernet Frame Structure (cont.)

- Addresses: 6 bytes
  - Each adapter is given a globally unique address at manufacturing time
    - Address space is allocated to manufacturers
      - 24 bits identify manufacturer
      - E.g., 0:0:15:* → 3com adapter
    - Frame is received by all adapters on a LAN and dropped if address does not match
  - Special addresses
    - Broadcast – FF:FF:FF:FF:FF:FF is "everybody"
    - Range of addresses allocated to multicast
      - Adapter maintains list of multicast groups node is interested in

14

Summary

- CSMA/CD → carrier sense multiple access with collision detection
  - Why do we need exponential backoff?
  - Why does collision happen?
  - Why do we need a minimum packet size?
    - How does this scale with speed?
- Ethernet
  - What is the purpose of different header fields?
  - What do Ethernet addresses look like?
- What are some alternatives to Ethernet design?

15

Today's Lecture

- Network Interface
- Link Layer
- Addressing/IP
- Routing
- TCP

16

Routing Techniques Comparison

	Source Routing	Global Addresses	Virtual Circuits
Header Size	Worst	OK – Large address	Best
Router Table Size	None	Number of hosts (prefixes)	Number of circuits
Forward Overhead	Best	Prefix matching (Worst)	Pretty Good
Setup Overhead	None	None	Connection Setup
Error Recovery	Tell all hosts	Tell all routers	Tell all routers and Tear down circuit and re-route

17

IP Addresses

- Fixed length: 32 bits
- Initial classful structure (1981) (not relevant now!!!)
- Total IP address size: 4 billion
  - Class A: 128 networks, 16M hosts
  - Class B: 16K networks, 64K hosts
  - Class C: 2M networks, 256 hosts

High Order Bits	Format	Class
0	7 bits of net, 24 bits of host	A
10	14 bits of net, 16 bits of host	B
110	21 bits of net, 8 bits of host	C

18

Subnet Addressing RFC917 (1984)

- Class A & B networks too big
  - Very few LANs have close to 64K hosts
  - For electrical/LAN limitations, performance or administrative reasons
- Need simple way to get multiple "networks"
  - Use bridging, multiple IP networks or split up single network address ranges (subnet)
- CMU case study in RFC
  - Chose not to adopt – concern that it would not be widely supported ☺

19

Classless Inter-Domain Routing (CIDR) – RFC1338

- Allows arbitrary split between network & host part of address
  - Do not use classes to determine network ID
  - Use common part of address as network number
  - E.g., addresses 192.4.16 - 192.4.31 have the first 20 bits in common. Thus, we use these 20 bits as the network number → 192.4.16/20
- Enables more efficient usage of address space (and router tables) → How?
  - Use single entry for range in forwarding tables
  - Combined forwarding entries when possible

20

IP Addresses: How to Get One?

Network (network portion):

- Get allocated portion of ISP's address space:

ISP's block 200.23.16.0/20	11001000 00010111 00010000	00000000
Organization 0 200.23.16.0/23	11001000 00010111 00010000	00000000
Organization 1 200.23.18.0/23	11001000 00010111 00010010	00000000
Organization 2 200.23.20.0/23	11001000 00010111 00010100	00000000
... .	.....	....
Organization 7 200.23.30.0/23	11001000 00010111 00011110	00000000

21

IP Addresses: How to Get One?

- How does an ISP get block of addresses?
  - From **Regional Internet Registries** (RIRs)
    - ARIN (North America, Southern Africa), APNIC (Asia-Pacific), RIPE (Europe, Northern Africa), LACNIC (South America)
- How about a single host?
  - Hard-coded by system admin in a file
  - DHCP: Dynamic Host Configuration Protocol: dynamically get address: "plug-and-play"
    - 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

22

Important Concepts

- Base-level protocol (IP) provides minimal service level
  - Allows highly decentralized implementation
  - Each step involves determining next hop
  - Most of the work at the endpoints
- ICMP provides low-level error reporting
- IP forwarding → global addressing, alternatives, lookup tables
- IP addressing → hierarchical, CIDR
- IP service → best effort, simplicity of routers
- IP packets → header fields, fragmentation, ICMP

23

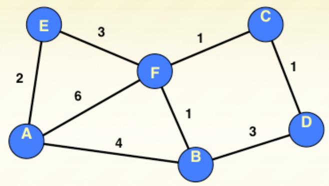
Today's Lecture

- Network Interface
- Link Layer
- Addressing/IP
- Routing
- TCP

24

Distance-Vector Method

Initial Table for A		
Dest	Cost	Next Hop
A	0	A
B	4	B
C	∞	–
D	∞	–
E	2	E
F	6	F



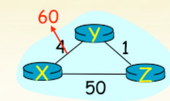
- Idea
  - At any time, have cost/next hop of best known path to destination
  - Use cost ∞ when no path known
- Initially
  - Only have entries for directly connected nodes

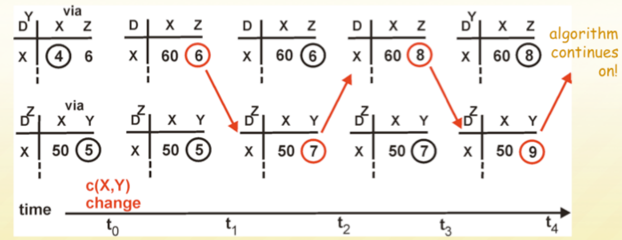
25

Distance Vector: Link Cost Changes

Link cost changes:

- Good news travels fast
- Bad news travels slow - "count to infinity" problem!



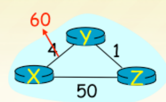


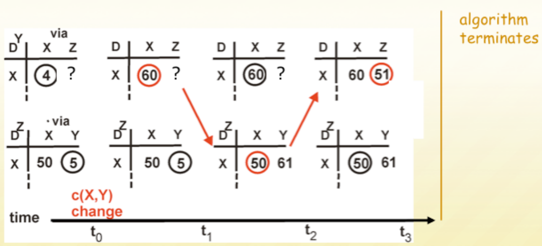
27

Distance Vector: Split Horizon

If Z routes through Y to get to X :

- Z does not advertise its route to X back to Y





28

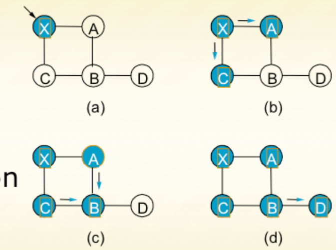
Link State Protocol Concept

- Every node gets complete copy of graph
  - Every node "floods" network with data about its outgoing links
- Every node computes routes to every other node
  - Using single-source, shortest-path algorithm
- Process performed whenever needed
  - When connections die / reappear

29

Sending Link States by Flooding

- X Wants to Send Information
  - Sends on all outgoing links
- When Node Y Receives Information from Z
  - Send on all links other than Z



Comparison of LS and DV Algorithms

Message complexity
- LS: with n nodes, E links, O(nE) messages
- DV: exchange between neighbors only O(E)

Speed of Convergence
- LS: Complex computation
  - But…can forward before computation
  - may have oscillations
- DV: convergence time varies
  - may be routing loops
  - count-to-infinity problem
  - (faster with triggered updates)

Space requirements:
- LS maintains entire topology
- DV maintains only neighbor state

Comparison of LS and DV Algorithms

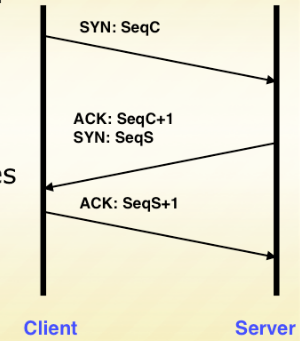
- Robustness: what happens if router malfunctions?
- LS:
  - node can advertise incorrect *link* cost
  - each node computes only its *own* table
- DV:
  - DV node can advertise incorrect *path* cost
  - each node's table used by others
    - errors propagate thru network
- Other tradeoffs
  - Making LSP flood reliable

Today's Lecture

- Network Interface
- Link Layer
- Addressing/IP
- Routing
- TCP
  - Connection establishment, flow control, reliability, congestion control

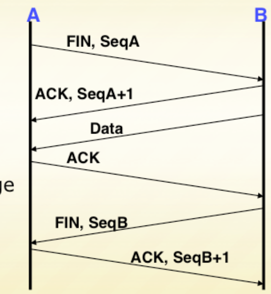
Establishing Connection: Three-Way handshake

- Each side notifies other of starting sequence number it will use for sending
  - Why not simply chose 0?
    - Must avoid overlap with earlier incarnation
    - Security issues
- Each side acknowledges other's sequence number
  - SYN-ACK: Acknowledge sequence number + 1
- Can combine second SYN with first ACK

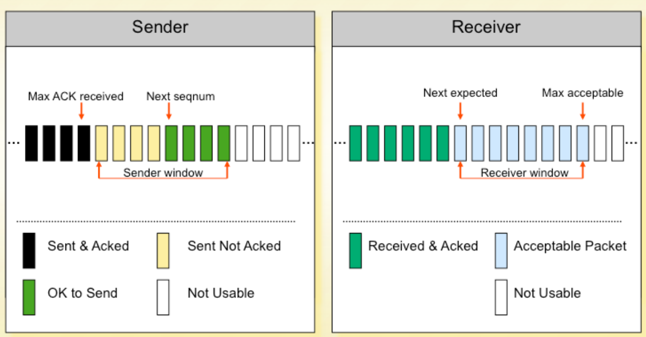


Tearing Down Connection

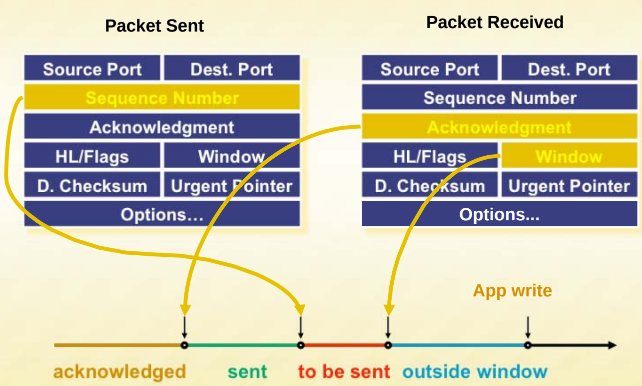
- Either side can initiate tear down
  - Send FIN signal
  - "I'm not going to send any more data"
- Other side can continue sending data
  - Half open connection
  - Must continue to acknowledge
- Acknowledging FIN
  - Acknowledge last sequence number + 1



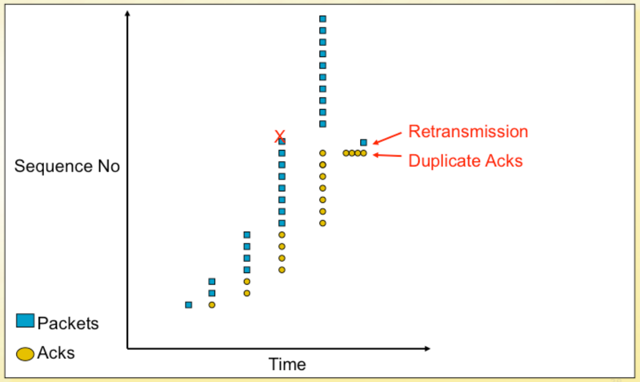
Sender/Receiver State



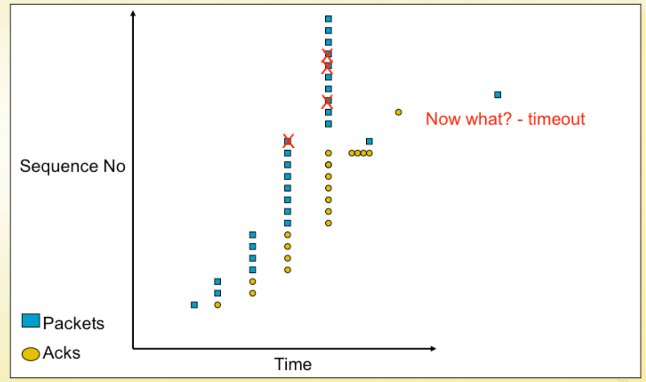
Window Flow Control: Send Side



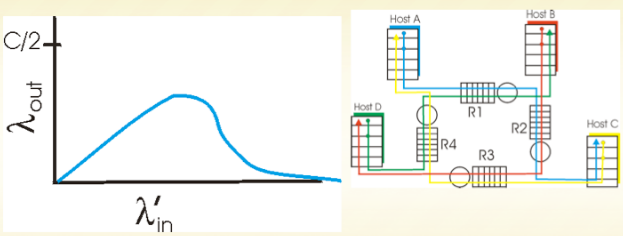
Fast Retransmit



TCP (Reno variant)



Causes & Costs of Congestion

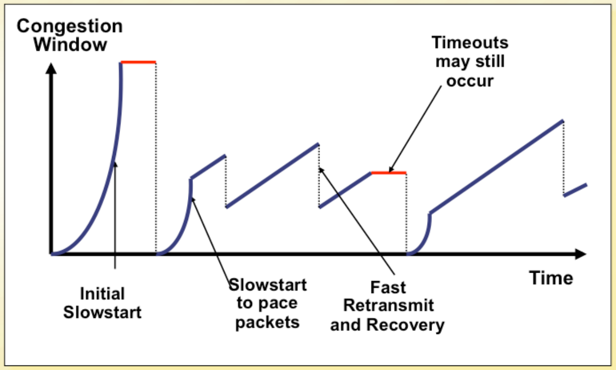


- When packet dropped, any "upstream transmission capacity used for that packet was wasted!

TCP Congestion Control

- Changes to TCP motivated by ARPANET congestion collapse
- Basic principles
  - AIMD
  - Packet conservation
  - Reaching steady state quickly
  - ACK clocking

TCP Saw Tooth Behavior



42

Important Lessons

- TCP state diagram → setup/teardown
  - Making sure both sides end up in same state
- TCP timeout calculation → how is RTT estimated
  - Good example of adapting to network performance
- Modern TCP loss recovery
  - Why are timeouts bad?
  - How to avoid them? → e.g. fast retransmit
  - Making the common case work well

43

Important Lessons

- Sliding window flow control
  - Addresses buffering issues and keeps link utilized
  - Need to ensure that distributed resources that are known about aren't overloaded
- Why is congestion control needed?
  - Need to share some resources without knowing their current state
- How to evaluate congestion control algorithms?
  - Why is AIMD the right choice for congestion control?
    - Results in stable and fair behavior

44

Next Lecture

- Android APIs (Dongsu)
- Reading
  - Project 1 handout

45