Inter Domain Structure
Autonomous System (AS): Set of routers that have a single policy running under a single administration
Looks like a single entity from outside, w/ AS#
Routing info exchanged by Border Gateway Protocol (BGP)

Network Layers

Application
Presentation
Session
Transport
Network
Data Link
Physical

Transporting protocols have 2 functions
① Manipulate data: moving data to/from net, error detection, buffering, encryption, presentation formatting
② Control transfer process: flow control, congestion control, multiplexing, time stamping, detect ntwk transmission problems.

More Specifically
① Manipulate data
- Data buffering inside OS.
- Presentation Formatting: x86 to Sun via NetCDF (common data format)
- Error detection: can be carried out by Ntw HW & OS.

② Control transfer process
- Mtplexing among apps.
- Flow ctrl: receiver buffer full, send msg to pause sending
- Congestion Ctrl: pause tx if NTW infrastructure full

COMP 535
PH 1

Other transporting protocol Issues:
③ Addressing: Mostly done by NTW, port addressing for Mplexing apps.
④ Connection establishment: Tx Rx Sync, state establishment
⑤ Connection Release

Drawbacks of Layered Implementation
① Efficiency: could lead to multiple copying
② Framing: Data can be divided/joined as it goes through layered stack

TCP Concerns
① Prevent buffer overRun
② Recover from NTW data loss
③ Detect & Correct corrupted data
④ Deal w/ out of order data reception

Sliding window protocols. (3 types)
① One bit sliding window
② Go back N
③ Selective Repeat

[1] One bit sliding window.
Send pkt, wait for timeout or recieve ACK

Performance.
RTT = 2xeTe prop delay
$T_{transmit}$ = Packet length / BW
$\text{Utilization}_{sender} = \frac{T}{T+RTT}$

Generic forms of pipelined protocols.
② Back N
③ Select Repeat

COMP535
P 12

Network Science: characterize ntws.
- Degree distribution
- Node Clustering
- Hierarchical Structure

Node degree distribution
- Power law distrib: few large degree nodes & many small (Internet)
- Peaked distrib: Average distrib & bell shape curved.

Autonomous Systems (AS): Independently managed Ntw. (Backbone companies for Internet)

Switching & routing
- Circuit switching: Establish virtual circuit & use/hog channel
- Packet switching: Hop by Hop till dest. Channel multiplexed = better utilization.

Ntw Fundamentals
① Addresses: most humanmade Ntw explicitly assign add to nodes.
② Forwarding: Use route table to determine on which interface to forward packet
③ Routing: Aggregates topological info to create routing tables used by forwarding

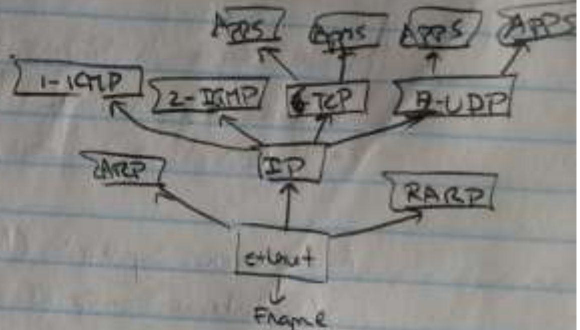
Hop by hop protocols: IP, Ethernet
End-to-End protocols: TCP, FTP

IP forwarding
① Src & dest in same NTW: launch ARP, Forward to MAC
② Src & dest not in same NTW: lookup routing table, Forward Frame to next hop Router.

COMP535
P01

Classes		Range	Default Mask
A	0...	0-127	255.0.0.0
B	10...	128-191	FF.FF.0.0
C	110...	192-223	FF.FF.FF.0
D	1110....	224-239	

IP Header contains protocol field.
1- ICMP
2- IGMP
6- TCP
17- UDP.



Ethernet Frame Header Protocols.
IP, ARP, RARP.

IP Header
- Aligned per 32 bits.
- Version IPv4 o 6
- IHL: Internet header length in 32 bit words. (default 5 w/no options)
- Type of service: OSPF, etc...
- Total length: bytes in packet (including header)
- TTL : in hops. When Ø, pkt dropped & ICMP reported

Fragmentation. (ex of a packet bottlenecked @ 1.5K)

Incoming	ID	Length	Don't frag (DF)	More frag (MF)	offset	DATA
packet	X	4020	0	0	0	ABC
Frag 1	X	1520	0	1	0	A
Frag 2	X	1520	0	1	1500	B
Frag 3	X	1020	0	0	3000	C

Id + Source addy used to reconstruct original packet

COMP 535
p02

Alternative to Fragmentation:
Discover path's MTU
① Fire packet @ max length w/ DF=1
② If bottlenecked, dropped because of DF & ICMP error response sent
③ decrease probe size & restart

Routers optimized for No option headers - 5x32 bits.
Packets w/ options are treated w/ lower priority

Option alternative for Source Routing
· Instead of specifying option from A to B through C, fire up IP in IP.

to C from A	to B from A	DATA

when C receives it, unwraps packet & fire IP in IP

ICMP: Internet Control msg protocol
- Layered on top of IP (type 1)
- Answered when TTL=0 or destination unreachable

32 bit header

Type	code	checksum
unused		
IP Header + 64 bits of original packet...		

ICMP codes examples:
- Time exceeded ⇒ TTL=0
- Source quench = router congestion. Host should reduce sending
- Parameter problem: error in IP header encoding
- Ping: Replied by swapping IP src & dest, and ECHO by ECHO REPLY
- Traceroute: send packet w/ TTL=1
  increment TTL on subsequent packets & note IP of Killer routers until destination is reached
- Route discovery:
  ① Host sends "router solicitations"
  ② Router replies w/ "router advertisement" w/ list of neighbors + preference.

Network application: connect systems w/ distributed components
- Connect people
- Data access
- Remote processing
- Remote source control (mobile robots)
- Distributed processing: Grid computing

Classes of NTW apps.
- Symmetry
  - client/server apps
  - P2P apps
- Communicating
  - Dependent: distributed components constantly msging
  - Independent: pass few msgs (ex: SETI)
- Granularity
  - Fine grain apps: heavily communicating small loads
  - Coarse grain apps: Low intercommunication
- Time Sensitivity
  - Hard "Real Time": missing deadline = crash or system failure, etc.
  - Soft "Real Time": missing deadline degrades performance only
  - No "Real Time": No time based performance: want best effort

Network app Requirements
- Send msgs b/w end points (delay, Δdelay, bw)
- Reliability & QoS
- Resource discovery
  - Name based: ex DNS
  - Attribute based: ex Google

COMP535
P04

P2P Centralized

Pros

- Highly Scalable/upgradable
- Single info directory

Cons

- Single pt of failure (copyrights)
- Performance Bottleneck

P2P DeCentralized

- Each peer is/is assigned a groupleader
- Leader tracks content of child
- Peer queries G.L., who can query other GL

Pros:

- No single vulnerability
- More diff to shutdown

Cons:

- Bootstrap node needed to add new guy
- GL can get overloaded

P2P Gnutella

- No hierarchy
- Send join msg to bootstrap node to join
- Query neighbors, they forward

Pros

- Equal responsibility among Peers.
- Highly decentralized

Cons

- Excessive Query traffic
- Query radius may not have content
- Bootstrap node = point of failure
- High overhead for maintenance of overlay Ntw data

P2P Consistent Hashing

- Hashed over circular space
- Each node keeps complete table of node IDs & their IPs.
- When looking for data, search list for closest node
- Each node has a multilevel Routing table

ex 65a b31

0 1 2 3 4 5 6 7 8 9 a b c d e f

0 1 2 3 4 5 6 7 8 9 a b c d e f

Node knows about immediate data + neighbors.

Node (45) knows about data (678)

Node (112) knows about data (345)

COMP535

p 05.

DARPA design goals

1. Internet com. must continue despite loss of Ntw gateways
2. Must support multiple types of communication services
3. Must accomodate a variety of networks.
4. Must permit distributed Mgment of resources.
5. Cost effective
6. Permit host attachment w/ low level of effort
7. Resources used must be accountable

Jitter: variability of packet delays within same packet stream

Internet evolution possible for streaming

A) Integrated Service philosophy: fundamental changes to allow end-to-end BW reservation → Complex
B) Provide 1st & 2nd Class service → fewer changes.
C) Nothing - Content distribution = more BW when needed

Audio compression

speed = sample/sec + log samples

ex: 8 000 sample/sec than can take 256 values will require

$8000 \frac{\text{sample}}{\text{sec}} \times \frac{8\text{bits}}{\text{sample}} = 64\text{Kb/sec}$

Streaming audio

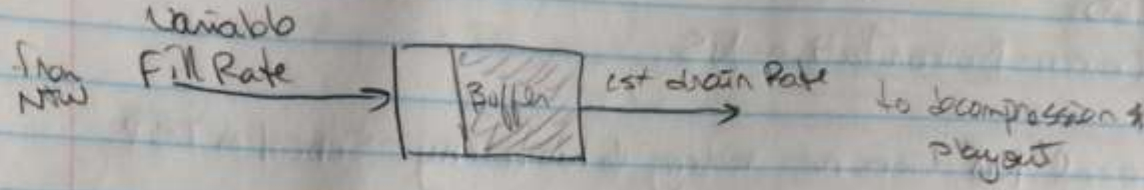
① HTTP Request / Response for metafile
② Browser launches player → passes metafile
③ Player contacts server.
④ Server Streams A/V to player.

* Allows for non HTTP protocols btw player & server (UDP)

MPS35 * Client side buffering → playout delay compensates

OG for ntw delay.

Playout delay: Time btw data starts to arrive & media starts to play (time to fill buffer)



UDP

- Server sends @ rate appropriate to client, oblivious of NTW congestion
  Send Rate = Encoding rate (cst)
  Fill Rate = Encoding rate (cst) - packet loss
- Only needs short playout delay for jitter

TCP

- Send data @ max rate under TCP
- Fill rate fluctuats w/ TCP congestion ctrl
- larger playout delays to smooth TCP delivery rate
- HTTP/TCP passes more easily through firewall

VOIP

- Loss tolerance ~1-10%
  - NTW loss: Router Buffer overflow = packet lost
  - Delay loss: Packet arrives too late ~400mS
- Fixed playout delay "q". Chunk @ time t plays between [t, t+q]
  - large q: less packet loss
  - Small q: better interaction

cont

DNS hierarchy
① Root Servers
② TLDs
③ Domain Authoritative NS.

Local NS: does not belong to hierarchy - School or ISP.
Acts as proxy, forwards queries

Queries
- Recursive: puts burden of name resolution on NS.
- Iterative: Contacted NS replies w/ name of next NS.

DNS DB - RRs.
format: [name, value, type, ttl]
Types
- SOA = Start of Authority = parameters for this zone
- A = [name = hostname], [value = IP]
- MX: mail server, [value = name of mailserver associated w/ name]
- NS: [name = domain] [value = IP of authoritative NS for this domain]
- CNAME canonical [name = alias of real name] [value = real name]
- HINFO: Host CPU & OS in ASCII
- TXT = uninterpreted ASCII

E-MAIL
- User agent mail reader/composer ex: Outlook
- Mail Server: mailbox (in), message Q (to be sent)
- SMTP: Simple Mail Transfer Protocol (TCP, port 25)
  used btw mail servers to transport mail
  3 phase transfer: Handshaking, transfer, closure

Planes

Management Plane

Controls Plane: Routing by: ① Dissimination: State info spread accross ntw
② Decision: Route computation/path selection

Data plane

① Dissimination:

consens

- Spreading time
- # of msgs
- Hotspot / bottleneck creation

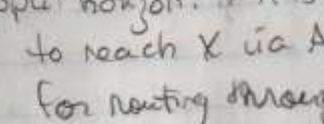
Types

- Flooding algorithm: Routers in fixed topology, communication along direct links. Dissimination time = ntw diameter, Msg complexity B (n-nodes)(m-edges)
- Swamping: neighbors grow as msg arrive from distant nodes. Dissimination time Θ(log n), message complexity n^3 +
- Gossiping: Probabilistic, each node selects K neighbors & exchange msgs. high probability of fast convergence

Distance Vector routing (ex RIP)

Split horizon: if A is routing to X via B, B should not try to reach X via A. Nodes ignores info from interface for routing through that interface.

Still issue w/ 2 links down



B to D via 4 cost 2

C to D via 5 Cost 2

E to D via 6 cost Inf

COMP535
P09

OSPF: Link State are based on a distributed map
Flooding
① Recieve Msg, lookup record in DB.
② If record absent, add it & broadcast msg
③ if record found & DB# lower, replace record w/ new msg & broadcast
④ if record found & DB# higher, transmit DB value through incoming interface.

Bringing up adjacencies: Sync 2 parties w/ most uptodate version of each record.

OSPF DB record protection
- Flooding procedures include hopbyhop ACKs.
- Link state has TTL, deleted if not refreshed on time
- Records have checksum
- Msgs can be authenticated by passwd.

Why is link state protocol better than distance vector protocol
- Fast, loopless
- Support for multiple metrics.
- Support for mltp paths to dest
- Seperate representation of external routes.

Best root metric:
- largest throughput (Bandwidth) → good for big FTP.
- lowest delay → VOIP
- lowest $ → hot potato routing
- best reliability → banking.