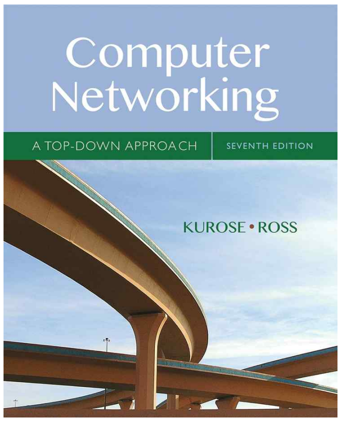
Chapter 1 Introduction



Seongwook Youn
Department of Software
Korea National University of Transportation

Fall 2019

Computer Networking: A Top Down Approach
7th edition
Jim Kurose, Keith Ross
Addison-Wesley

Chapter 1: roadmap

1.1 what *is* the Internet?
1.2 network edge
- end systems, access networks, links

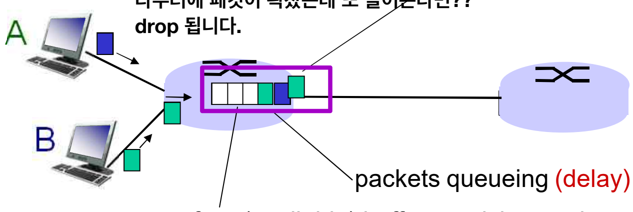
1.3 network core
- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks
1.5 protocol layers, service models
1.6 networks under attack: security
1.7 history

How do loss and delay occur?

packets *queue* in router buffers

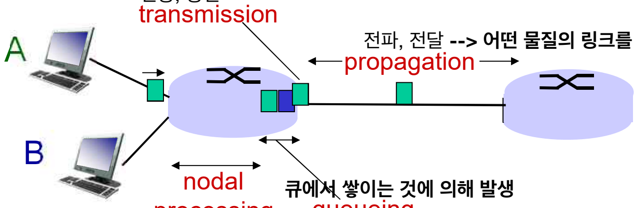
- packet arrival rate to link (temporarily) exceeds output link capacity 패킷 도착하는 속도 > 링크를 나가는 속도
- packets queue, wait for turn



Introduction 1-3

패킷이 delay되는 4가지 이유

Four sources of packet delay



$$\boxed{d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}}$$

하나의 노드에서 발생하는 모든 딜레이: processing, queueing, transmission, propagation 딜레이

d_{proc}: nodal processing
- check bit errors
- determine output link
- typically < msec

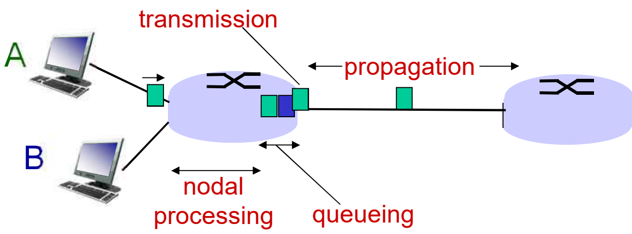
1) 들어오는 비트들에 에러가 없는지
2) 여러 개의 output link 중 어디로 보낼지 계산하는 시간

d_{queue}: queueing delay
- time waiting at output link for transmission
- depends on congestion level of router

Introduction 1-4

[queue에 쌓이는 delay]
output link에 전송되기 위해 기다리는 시간
--> 라우터 복잡(혼잡) 정도에 달려있음
하나도 안 쌓여 있다면 바로 나갈거니깐요.

Four sources of packet delay



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{trans}: transmission delay:

- L: packet length (bits)
- R: link *bandwidth (bps)*
- $d_{trans} = L/R$

d_{prop}: propagation delay: 매체의 길이 매체의 속도

- d: length of physical link
- s: propagation speed in medium ($\sim 2 \times 10^8$ m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop} *very* different

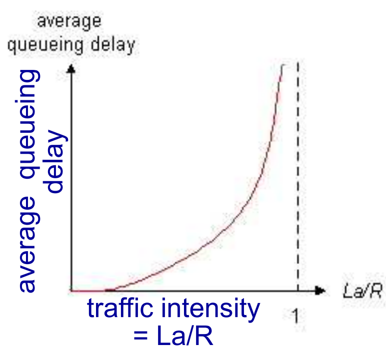
transmisson delay가 측정을 통해 알 수 있다면
propagation delay는 어떤 재료로 링크를 사용 하는지에 다름

내 생각에는 이거 계산하는 문제가 나올 것 같음!!

Queueing delay (revisited)

- R: link bandwidth (bps)
- L: packet length (bits)
- a: average packet arrival rate

linear하게 증가하는 것이 아니라 ex.. 그렇게 증가한다.



La/R ~ 0

La/R -> 1

trace route
인터넷 상에서 특정 컴퓨터에 접속하려고 할 때,
실제 연결 경로가 어떻게 이루어지는지를 알려줍니다.

인터넷 상에서의 네트워크 연결은 매 순간 적절한 경로를 찾아가기 때
문에 접속할 때의 상황에 따라 경로가 다를 수 있습니다.

네트워크 매니저는 경로를 잘 관리함으로써 효율적으로 연결합니다.

=> 규범 표기가 trace route 입니다.

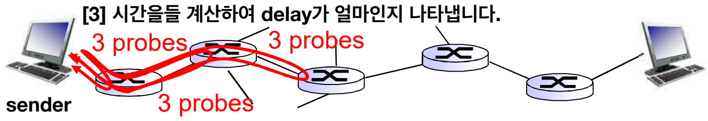
"Real" Internet delays and routes

- what do "real" Internet delay & loss look like?

traceroute는 리눅스 이름입니다. windows는 다릅니다. 직접 cmd에서 실험하는 것도 좋을 것 같습니다.

- `traceroute` program: provides delay measurement from source to router along end-end Internet path towards destination. For all *i:*
  - sends three packets that will reach router *i* on path towards destination
  - router *i* will return packets to sender
  - sender times interval between transmission and reply.

[1] 3개의 패킷을 보냅니다.
[2] 각 패킷이 돌아오는 시간을 확인합니다.
[3] 시간을들 계산하여 delay가 얼마인지 나타냅니다.



[예] sender

하나의 sender를 세 개의 packet에 보내고 돌아오는 각각의 시간을 계산하여 delay를 계산합니다.

"Real" Internet delays, routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

3 delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18 ms 18 ms
6  abilene-vbns.abilene.ucaid.edu (198.32.11.9)  22 ms  18 ms  22 ms
7  nycm-wash.abilene.ucaid.edu (198.32.8.46)  22 ms  22 ms  22 ms
8  62.40.103.253 (62.40.103.253)  104 ms 109 ms 106 ms
9  de2-1.de1.de.geant.net (62.40.96.129)  109 ms 102 ms 104 ms
10  de.fr1.fr.geant.net (62.40.96.50)  113 ms 121 ms 114 ms
11  renater-gw.fr1.fr.geant.net (62.40.103.54)  112 ms  114 ms  112 ms
12  nio-n2.cssi.renater.fr (193.51.206.13)  111 ms  114 ms  116 ms
13  nice.cssi.renater.fr (195.220.98.102)  123 ms  125 ms  124 ms
14  r3t2-nice.cssi.renater.fr (195.220.98.110)  126 ms  126 ms  124 ms
15  eurecom-valbonne.r3t2.ft.net (193.48.50.54)  135 ms  128 ms  133 ms
16  194.214.211.25 (194.214.211.25)  126 ms  128 ms  126 ms
17  * * *
18  * * *
19  fantasia.eurecom.fr (193.55.113.142)  132 ms  128 ms  136 ms
```

trans-oceanic link

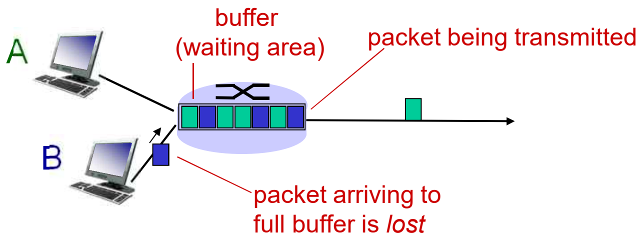
lose가 발생한 경우는 결과가 나오지 않습니다.

* means no response (probe lost, router not replying)

* Do some traceroutes from exotic countries at www.traceroute.org

Packet loss

* queue (aka buffer) preceding link in buffer has finite capacity
보통, 이전의, 앞선, 선행하는 / 한정된, 유한한
* packet arriving to full queue dropped (aka lost)
* lost packet may be retransmitted by previous node, by source end system, or not at all



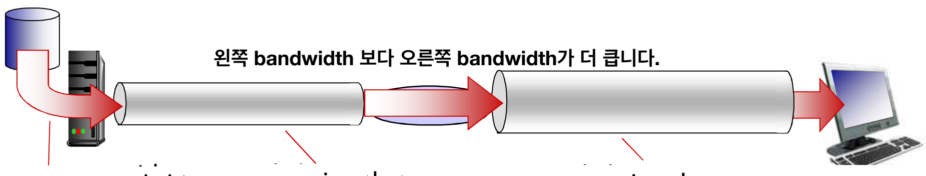
buffer에 쌓일 수 있는 패킷의 양이 정해져 있습니다.
이 양을 넘어서면(꽉찬 buffer에 새로운 packet이 들어오면) packet loss가 발생합니다.

Throughput 처리량

sender<->receiver 사이에 비트가 전송되는 속도를 말합니다.

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

왼쪽 bandwidth 보다 오른쪽 bandwidth가 더 큽니다.



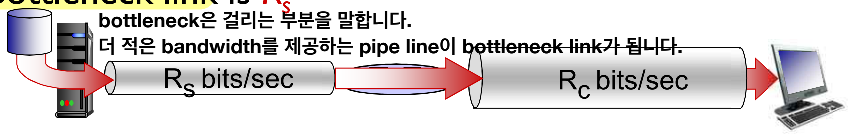
server가 bits를 파이프라인으로 보냅니다.

길이가 짧은 쪽을 S, 긴 쪽을 C라고 합니다.

Throughput (more)

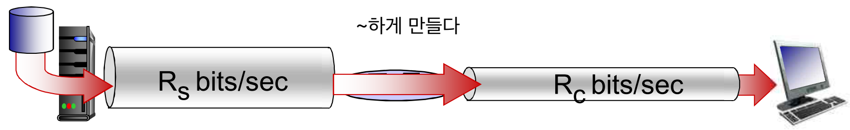
- $R_s < R_c$ What is average end-end throughput?
  병목
  Bottleneck link is R_s

bottleneck은 걸리는 부분을 말합니다.
더 적은 bandwidth를 제공하는 pipe line이 bottleneck link가 됩니다.



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

~하게 만들다



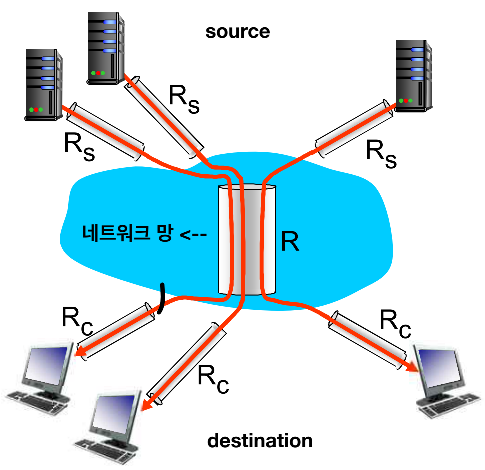
bottleneck link

link on end-end path that constrains end-end throughput

Throughput: Internet scenario

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

Rc 혹은 Rs가 bottleneck입니다.
더 속도가 작은 쪽이 되겠죠.



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

Chapter 1: roadmap

Protocol "layers"

Networks are complex, with many "pieces":

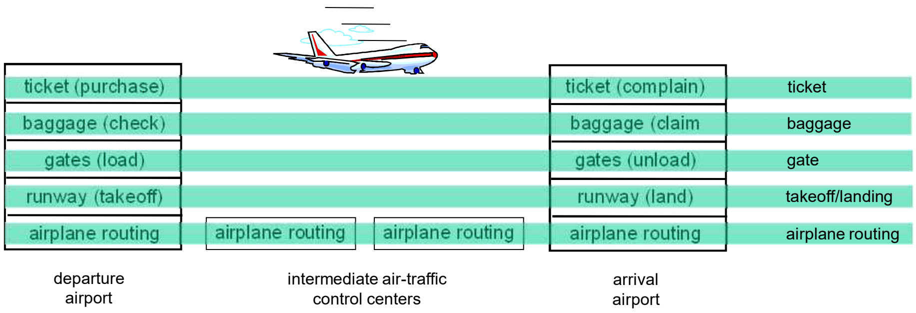
- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

is there any hope of *organizing* structure of network?

Protocol: defines the format and the order of messages exchanged between two or more communication entities, as well as the actions taken on the transmission and/or receipt of a message or other event

Layering of airplane functionality



layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

Why layering?

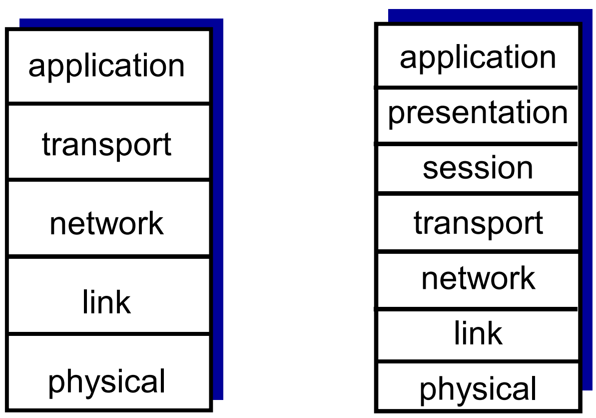
dealing with complex systems:

- explicit structure allows identification, relationship of complex system's pieces
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- For large and complex systems that are constantly being updated, the ability to change the implementation of a service without affecting other components of the system is another important advantage of layering

Protocol layering?

- Network designers organize protocols in layers
- Each protocol belongs to one of the layers
- Each layer provides its service
  - By performing certain actions within that layer
  - By using the services of the layer directly below it
- Potential drawbacks of layering
  - One layer may duplicate lower-layer functionality
    - Ex. Error recovery
  - Functionality at one layer may need information that is present in another layer
- Protocol stack
  - Protocols of the various layers
  - Consists of 5 layers: application, transport, network, link, and physical

Internet protocol stack and OSI reference model



Application layer

- Network applications and their application layer protocol reside
  - HTTP - provides for web document request and transfer
  - SMTP – provides for transfer of files between e-mail messages
  - FTP – provides for transfer of files between two end systems
  - DNS – translation of human-friendly names for Internet end systems
    - www.ietf.org to a 32-bit network address
- Message
  - A Packet of information at the application layer

Transport layer

- Transport application-layer messages between application endpoints
- Two transport protocols
  - TCP
    - Connection-oriented service
    - Guaranteed delivery and flow control
    - Congestion control
  - UDP
    - Connectionless
    - No-frills service
- Segment
  - A transport-layer packet

개념을 정확히 알아두세요.

Network layer

- Routing of network-layer packets (known as datagrams) from one host to another
- Contains IP protocol and numerous routing protocols
  - IP protocol
    - Defines the fields in the datagram as well as how the end systems and routers act on these fields
  - Routing protocol
    - Determine the routes that datagrams take between sources and destinations
  - Referred to as the IP layer

Link Layer에서 결정되는 것들을 이해합시다.

Link layer

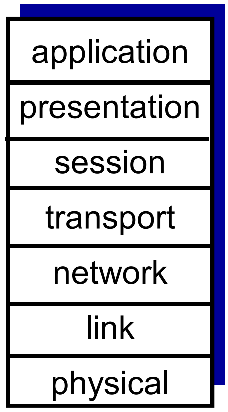
- Delivers the datagram to the next node along the route
- Ethernet, 802.11 (WiFi), PPP
- Frame
  - A link layer packet

Physical layer

- Move the individual bits within the frame from one node to the next
- Twisted-pair copper wire, single-mode fiber optics, etc.

ISO/OSI reference model

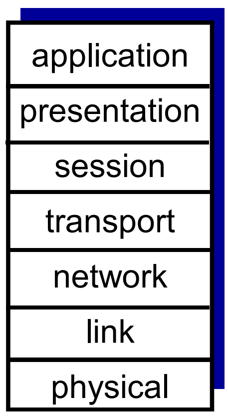
- International Organization for Standardization (ISO) proposed in the late 1970s
  - Computer networks is organized around seven layers
  - Called the Open System Interconnection (OSI) model
  - In fact, the inventors of the original OSI model probably did not have the Internet in mind when creating it



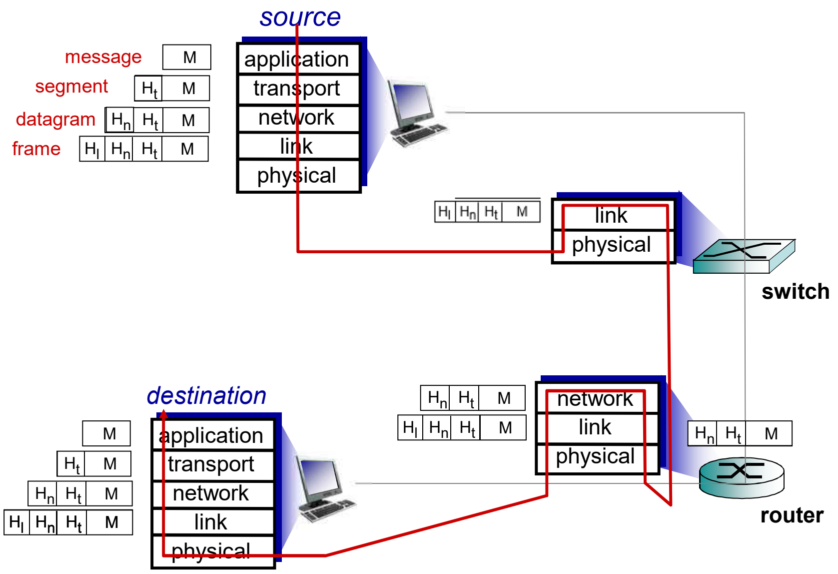
presentation, session 의 기능을 알아둡시다.

ISO/OSI reference model (Cont'd)

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



Encapsulation



Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

1.7 history

Network security

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

Bad guys: put malware into hosts via Internet

- malware can get in host from:
  - 사람이 무언가를 눌렀을 때 자가복제 합니다.
    virus: self-replicating infection by receiving/executing object (e.g., e-mail attachment)
  - 사람이 무언가를 하지 않아도 자가복제 합니다.
    worm: self-replicating infection by passively receiving object that gets itself executed. Standalone software and do not require a host program or human help to propagate
- Spyware malware는 키보드를 친 것 혹은 웹사이트 방분 기록을 보냅니다.
  spyware malware can record keystrokes, web sites visited, upload info to collection site
- botnet에 등록이 되고 DDcS에 공격이 될 수 있다.
  infected host can be enrolled in botnet, used for spam. DDoS attacks

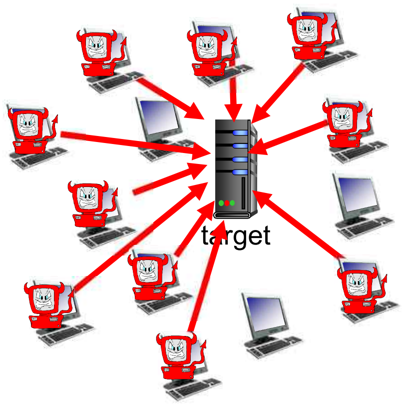
Robot+Network
중앙통제 장치로부터 지시를 받음

Bad guys: attack server, network infrastructure

DDoS는 앞에 분산이라는 뜻이 추가된 것임

Denial of Service (DoS): attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic

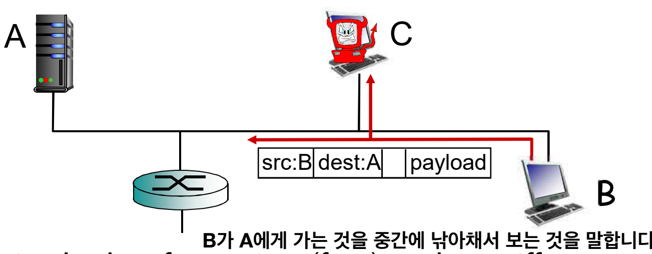
1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts



Bad guys can sniff packets

packet "sniffing": (킁킁거리다)

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

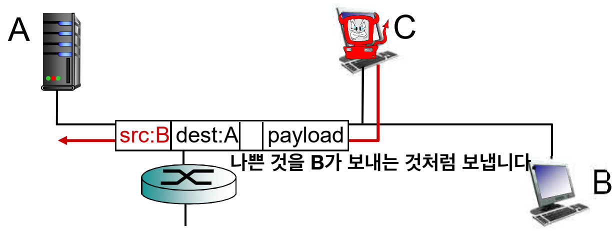


B가 A에게 가는 것을 중간에 낚아채서 보는 것을 말합니다.

- wireshark software is a (free) packet-sniffer

Bad guys can use fake addresses

IP spoofing: send packet with false source address



Chapter 1: roadmap

1.1 what *is* the Internet?
1.2 network edge
- end systems, access networks, links

1.3 network core
- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks
1.5 protocol layers, service models
1.6 networks under attack: security
1.7 history

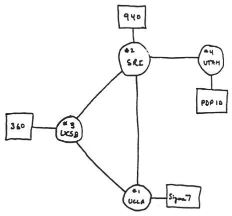
간단하게 읽어만 보세요~

Internet history

1961-1972: Early packet-switching principles

- 1961: Kleinrock - 패킷 스위칭의 효율성 queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational

- 1972:
  - ARPAnet public demo
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes



THE ARPA NETWORK

Internet history

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- late70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:
- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today's Internet architecture

Internet history

1980-1990: new protocols, a proliferation of networks

- 1983: deployment of TCP/IP
- 1982: smtp e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: ftp protocol defined
- 1988: TCP congestion control

- new national networks: Csnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

Internet history

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

- early 1990's: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990's: commercialization of the Web

late 1990's – 2000's:

- more killer apps: instant messaging, P2P file sharing
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gbps

Internet history

2005-present

- ~750 million hosts
  - Smartphones and tablets
- Aggressive deployment of broadband access
- Increasing ubiquity of high-speed wireless access
- Emergence of online social networks:
  - Facebook: soon one billion users
- Service providers (Google, Microsoft) create their own networks
  - Bypass Internet, providing "instantaneous" access to search, emai, etc.
- E-commerce, universities, enterprises running their services in "cloud" (eg, Amazon EC2)

Introduction 1-37

Introduction: summary

covered a "ton" of material!

- Internet overview
- what's a protocol?
- network edge, core, access network
  - packet-switching versus circuit-switching
  - Internet structure
- performance: loss, delay, throughput
- layering, service models
- security
- history

you now have:

- context, overview, "feel" of networking
- more depth, detail *to follow!*

Introduction 1-38

Leonard Kleinrock

- Distinguished professor of UCLA
- Creation of packet-switching principles in 1961
- First node of the Internet
  - First host-to-host message from UCLA to Stanford SRI ("Lo")
- B.E.E from CUNY
- MS and Ph.D. from MIT