Introduction

What is a computer network?

Components of a computer network:

- hosts (PCs, laptops, handhelds)
- routers & switches (IP router, Ethernet switch)
- links (wired, wireless)
- protocols (IP, TCP, CSMA/CD, CSMA/CA)
- applications (network services)
- humans and service agents

Hosts, routers & links form the *hardware* side.

Protocols & applications form the *software* side.

Protocols can be viewed as the "glue" that binds everything else together.

A physical network:

Protocol example: low to high

- NIC (network interface card): hardware

  → e.g., Ethernet card, WLAN card
- device driver: part of OS
- ARP, RARP: OS
- IP: OS
- TCP, UDP: OS
- OSPF, BGP, HTTP: application
- web browser, ssh: application

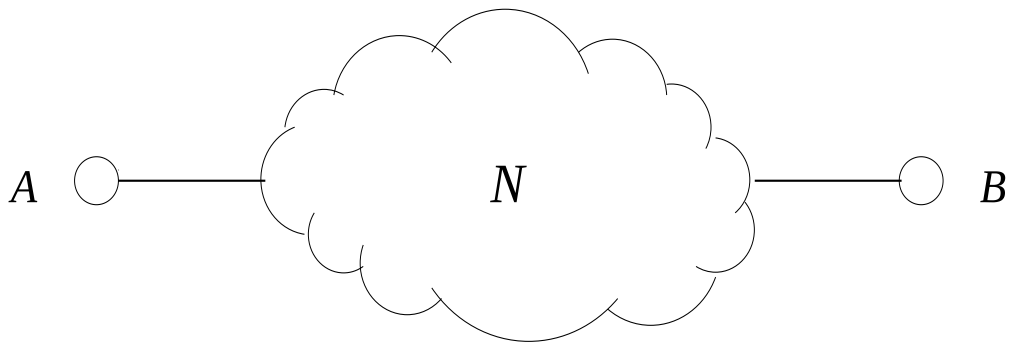
$\longrightarrow$ multi-layered glue

What is the role of protocols?

$\longrightarrow$ facilitate communication or networking

Simplest instance of networking problem:

> Given two hosts A, B interconnected by some network N, facilitate communication of information between A & B.



Information abstraction

- objects (e.g., files)
- bytes & bits

  $\longrightarrow$ digital form
- signals over physical media (e.g., electromagnetic waves)

  $\longrightarrow$ analog form

Minimal functionality required of A, B

- encoding of information
- decoding of information

$\longrightarrow$ data representation & a form of translation

Additional functionalities may be required depending on properties of network N

- information corruption

  $\rightarrow 10^{-9}$ for fiber optic cable

  $\rightarrow 10^{-3}$ or higher for wireless

- information loss: packet drop
- information delay: think of airport
- information security

Network N connecting two or more nodes can be

- point-to-point links
- multi-access links
- internetworks

$\longrightarrow$ logical topology point-of-view

$\longrightarrow$ may differ from physical topology

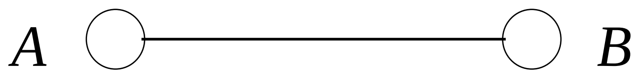
Network medium may be

- wired
- wireless

Node (e.g., hosts, routers) may be

- stationary
- mobile

Point-to-point links

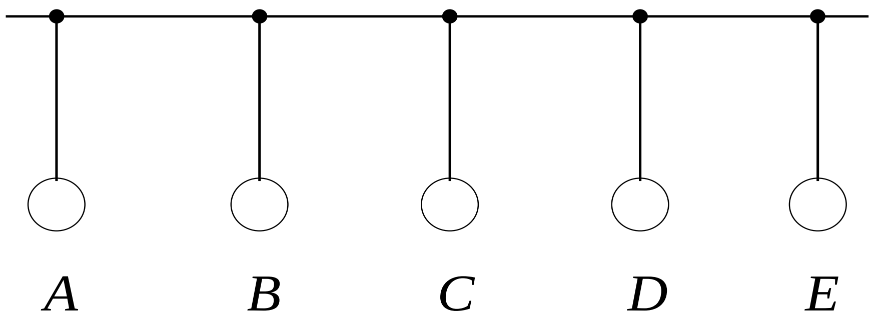


- various "cables"
- line of sight wireless communication

  → directional antennas
- no addressing necessary

  → special case

Multi-access links

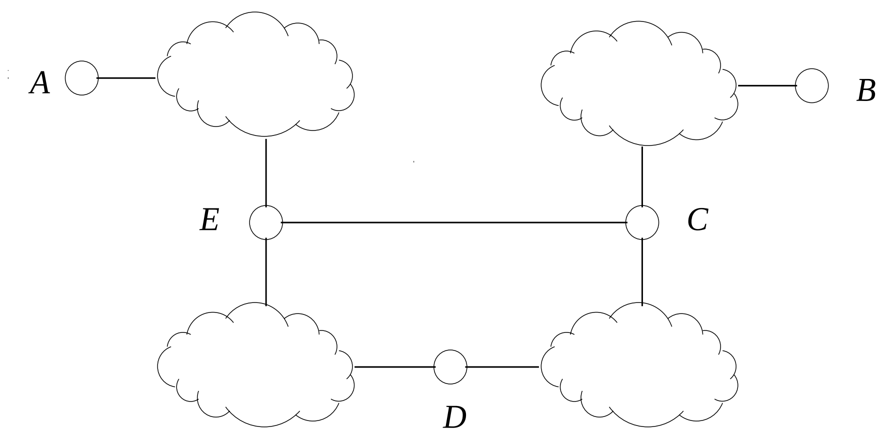


- bus (e.g., old Ethernet, others)
- wireless media

  → omni-directional antennas
- broadcast mode
- access control: i.e., bus arbitration

  → resolve contention and recover from interference
- addressing necessary

Internetwork



- recursive definition
- addressing necessary
- path selection between sender/receiver: routing
- protocol translation: internetworking
- location management

  → mobile IP

LAN (local area network) vs. WAN (wide area network) distinction:

- LAN: point-to-point, multi-access

  → when wireless: WLAN
- WAN: internetwork

⟶ geographical distinction is secondary

⟶ often go hand-in-hand

⟶ counter example?

Myriad of different LAN technologies co-existing in a WAN. For example:

- Fast Ethernet (100 Mbps)
- Gigabit Ethernet (1000 Mbps)
- FDDI (Fiber Distributed Data Interface)
- wireless Ethernet (11 Mbps, 54 Mbps)
- ATM
- SONET
- modem/DSL & PPP

Thus, a network (internetwork), at the base level, is a collection of LANs that are connected together.

Each LAN, in general, speaks a different language.

$\longrightarrow$ e.g., message format

Internetworking solves this problem by translating everything to IP . . .

$\longrightarrow$ technical definition of **I**nternet

But:

$\longrightarrow$ not necessary

$\longrightarrow$ layer 2 switches

Remark on addresses (or names):

Communicating entities are *processes* residing on nodes A and B running some operating system.

Thus an *address* must also identify which process a message is destined for on a host.

$\longrightarrow$ e.g., port numbers in UNIX

Key Issues

<u>Fault-tolerance</u>

The larger the network, the more things can go wrong.

E.g.: node/link failures, message corruption, lost messages, outdated messages.

In a network system with n components, assume a component fails with independent probability p

$\longrightarrow$ expected number of failures: $n \cdot p$

$\longrightarrow$ probability of no failures: $(1 - p)^n$

$\longrightarrow$ probability of k simulaneous failures: p^k

In reality, failures are not independent.

$\longrightarrow$ e.g., power outage, natural disasters

We have:

⟶ Murphy's Law

- issue of reliable communication
- reliable network services

  ⟶ main principle: redundancy

- For example:
  - routing of messages: alternate/back-up routes
  - domain name servers: duplication
  - transmission by space probes: forward error correction (FEC)

Network security

Features:

- confidentiality: encryption
- integrity: message has not been tampered
- authentication: sender really is who she claims to be

$\longrightarrow$ cryptography

$\longrightarrow$ end-to-end

Modern security vulnerabilities:

- denial of service (DoS) attack

  $\rightarrow$ e.g., SYN flooding

- distributed DoS (DDoS) attack

  $\rightarrow$ e.g., commercial, personal, infrastructure

- virus attacks: e.g., Code Red

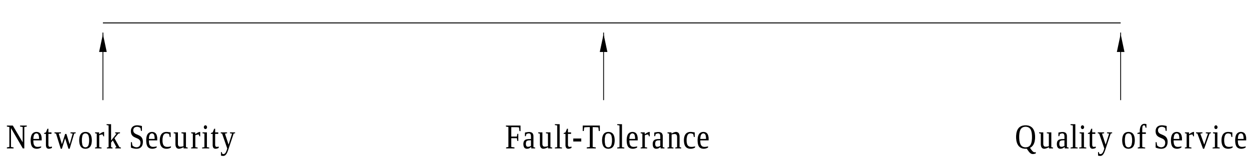
- along with fault-tolerance impacts QoS (quality of service)
- trade-off with overhead

    $\longrightarrow$ what is the desired operating point?

    $\longrightarrow$ too much $\Rightarrow$ too slow

    $\longrightarrow$ too little $\Rightarrow$ too vulnerable

Big picture:



Performance

Issues:

- excessive traffic can cause congestion (analogous to highways)

  → differences

- traffic volume exhibits large fluctuations

  → burstiness

- multimedia traffic is voluminous even for single user

- ubiquitous access

  → wired/wireless Internet

Thus, potential for bottleneck development.

⟶ similar consequences as failures

Different applications require different levels of service (fast, slow, accurate, etc.).

$\longrightarrow$ how to provide customized QoS

$\longrightarrow$ many users and applications: scalability

$\longrightarrow$ must interoperate with legacy Internet

$\longrightarrow$ incremental deployment

Current state:

- overprovisioning
- still no customized QoS
- not economic

Data networking & telephony convergence

Recent developments:

- VoIP (voice-over-IP): wired world

  → traditional TDM-based telephony system is entirely separate network

- cellular voice networks: 2G, 2.5G, 3G

  → what is 4G?

  → telcos/cellular providers are concerned

  → take-over by WLAN + IP?

6 million (or billion/trillion) $ question:

⟶ what will the wireless/wireline future hold?

Network performance

Three yardsticks or performance measures:

- throughput: bps or b/s (bits-per-second)
- latency: msec, ms (millisecond)

  → signal propagation speed
- delay: msec and second

  → includes software processing overhead
- jitter: delay variation (standard deviation)

Bandwidth vs. throughput:

bandwidth—maximum data transmission rate achievable at the hardware level; determined by signalling rate of physical link and NIC.

throughput—maximum data transmission rate achievable at the software level; overhead of network protocols inside OS is accounted for.

reliable throughput—maximum reliable data transmission rate achievable at the software level; effect of recovery from transmission errors and packet loss accounted for.

$\longrightarrow$ "true" measure of communication speed.

$\longrightarrow$ as opposed to raw throughput

$\longrightarrow$ point-to-point link: simple

$\longrightarrow$ multi-hop connection: more complicated

Trend on protocol implementation and overhead side:

migration of protocol software functionality into NICs; NIC is becoming a powerful, semi-autonomous device.

network processors: programmable NICs and more such as forwarding between NICs, i.e., router

$\longrightarrow$ as opposed to ASIC based devices

$\longrightarrow$ trade-off between hardware & software

$\longrightarrow$ boundary between hardware & software blurred

Meaning of "high-speed" networks:

- signal propagation speed is bounded by SOL (speed-of-light)

  $\longrightarrow$ ~300K km/s or ~186K miles/s

  $\longrightarrow$ optical fiber, copper

  $\longrightarrow$ coast-to-coast latency

  $\longrightarrow$ geostationary satellites: ~22.2K miles/s

  $\longrightarrow$ limitation of sending a single bit (e.g., as photon)

- can only increase "bandwidth"

  $\longrightarrow$ analogous to widening highway, i.e., more lanes

  $\longrightarrow$ simulatenous transmission

  $\longrightarrow$ a single bit does not travel faster

A key issue:

- $\longrightarrow$ fat & length pipes
- $\longrightarrow$ large *delay-bandwidth product*
- $\longrightarrow$ significant damage before recovery
- $\longrightarrow$ e.g., oil pipeline
- $\longrightarrow$ reactive cost
- $\longrightarrow$ characteristic feature of feedback systems

Some units:

Gbps (Gb/s), Mbps (Mb/s), kbps (kb/s):

10^9, 10^6, 10^3 bits per second; indicates data transmission rate; influenced by clock rate (MHz) of signalling hardware; soon Tbps.

$\longrightarrow$ communication rate: factors of 1000

Common bit rates:

- 10 Mbps (10BaseT), 100 Mbps (100BaseT)
- 100 Mbps (FDDI)
- 64kb/s (digitized voice)
- 144kb/s (ISDN line 2B + D service)
- 1.544 Mbps (T1), 44.736 Mbps (T3)
- 155.52 Mbps (OC-3), 622.08 Mbps (OC-12)
- OC-24, OC-48

GB, MB, kB:

2^{30}, 2^{20}, 2^{10} bytes; size of data being shipped; influenced by the memory structure of computer; already TB.

$\longrightarrow$ data size: factors of 1024

$\longrightarrow$ byte over bit

Common data sizes:

- 512 B, 1 kB (TCP segment size)
- 64 kB (maximum IP packet size)
- 53 B (ATM cell)
- 810 B (SONET frame)

Packet, frame, cell, datagram, message, etc.

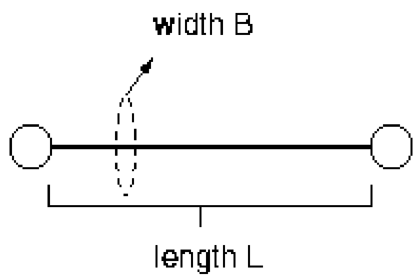
$\longrightarrow$ *packet* most generic term

Conventional usage

- frame: LAN-level
- datagram: IP packets
- cell: ATM packets
- packet: generic
- PDU (protocol data unit): generic
- message: high-level (e.g., e-mail)

Characteristics of message loss & delay:

(i) Point-to-point link



- Single bit:

  $\rightarrow \approx L/\text{SOL}$

  $\rightarrow$ latency

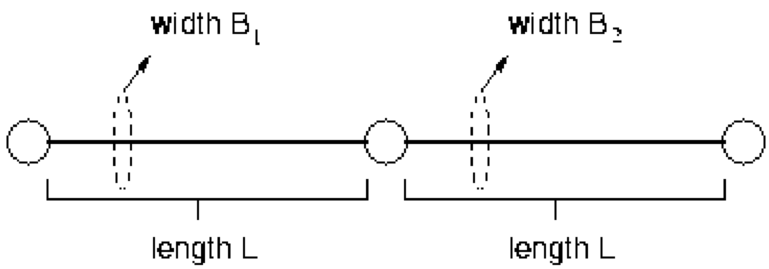
- Multiple, say S, bits:

  $\rightarrow \approx L/\text{SOL} + S/B$

  $\rightarrow$ latency + transmission time

... which dominates?

(ii) Multi-hop connection



- Case 1: $B_1 = B_2$

  $\longrightarrow = 2(L/\text{SOL} \ + \ S/B) \ + \ \varepsilon$

  $\longrightarrow$ ε: other processing overhead

- Case 2: $B_1 < B_2$
- Case 3: $B_1 > B_2$

  $\longrightarrow$ without memory, i.e., buffer: information loss

  $\longrightarrow$ loss rate $= 1 - (B_2/B_1)$ at full throttle

  $\longrightarrow$ with buffer: depends

  $\longrightarrow$ how much buffer space required for no loss?

Example:

- Suppose $B_1 = 2B_2$.
- Suppose transmitting at B_1 bps for 10 seconds.

  $\longrightarrow$ 5sec $\times$ B_1 bits
- Conservation argument:

  $\longrightarrow$ during 10s, 10sec $\times$ B_1 bits coming in

  $\longrightarrow$ during the same time, 10sec $\times$ B_2 bits going out

  $\longrightarrow$ since $B_2 = B_1/2$, excess 5sec $\times$ B_1 bits

  $\longrightarrow$ commensurate holding space for no loss

No loss comes at a cost:

- fast memory is not cheap
- management overhead
- packets have to wait in line for their turn

  → queueing delay

  → how long?

Depends on scheduling.

- FIFO (first-in-first-out) or FCFS
- round robin
- priority queue
- weighted fair queue

  ⟶ can use TOS field of IPv4 to encode priority

Is adding more and more buffer space a good solution?

When is it outright bad?

Is the speed mismatch problem inherent?

$\longrightarrow$ yes and no