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

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

→ geographical proximity is secondary although often goes hand-in-hand

→ counter examples?

Naming: LAN and IP addresses are insufficient.

Often communicating entities are apps running as processes in a host/router operating systems (e.g., Linux, Windows, IOS).

$\rightarrow$ IP only specifies NIC of host/server/router

$\rightarrow$ device with with multiple NICs may have multiple IP addresses

$\rightarrow$ called multi-homed

To identify a process to whom a message is destined:

→ use 16-bit port number supported by operating systems

→ why not use process IDs?

→ typical address: (IP address, port number) pair

→ note: IP address must eventually be translated to LAN address

When is port number not needed?

Network performance

In networks, speed is at a premium:

$\rightarrow$ if slow, typically not used in practice

$\rightarrow$ e.g., cryptographic protocols tend to be turned off at routers

Network design approach:

$\rightarrow$ emphasis of lightweight network core

$\rightarrow$ push heavyweight stuff toward the edge (i.e., host/server)

$\rightarrow$ called end-to-end paradigm

$\rightarrow$ has guided Internet design and evolution

$\rightarrow$ other approaches have been tried and failed

Performance yardsticks:

- bandwidth in bps (bits-per-second)

  → physical bandwidth ignoring slow-down due to protocols

- throughput (bps): includes protocol overhead

  → protocol: firmware in NIC and device driver in OS

  → in practice: app and user space OS overhead lead to further slow-down

- latency in msec (millisecond)

  → signal propagation speed (roughly: speed of light)

  → processing and buffering delay (queueing)

- jitter: delay variation

  → average delay small but max delay large

  → bad for multimedia

Meaning of "high-speed" networks:

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

  → ∼186K miles/s (∼300K km/s)

  → optical fiber, copper: slower than SOL

- Ex.: latency: Purdue to West Coast

  → for 2000 miles: ∼10 msec (= 2000/186000)

  → lower bound

- Ex.: geostationary satellites at ∼22.2K miles

  → latency: ∼120 msec

  → end-to-end (one-way): ∼240 msec

  → round-trip (two-way): ∼480 msec

  → roughly: half a second

  → shows up in news channel interviews

Meaning of high-speed:

- a single bit cannot go faster

  → can only increase bandwidth (bps): bits packed into 1 second

  → analogous to widening highway, i.e., more lanes

  → we will discuss how this packing is done

  → also called broadband

- interpretation of "high-speed" ⇔ "many lanes"

  → what does it buy?

  → completion time of large files is faster

  → in this sense, "higher" speed

  → for small files: marginal benefit

Some units:

Tbps, Gbps, Mbps, Kbps:

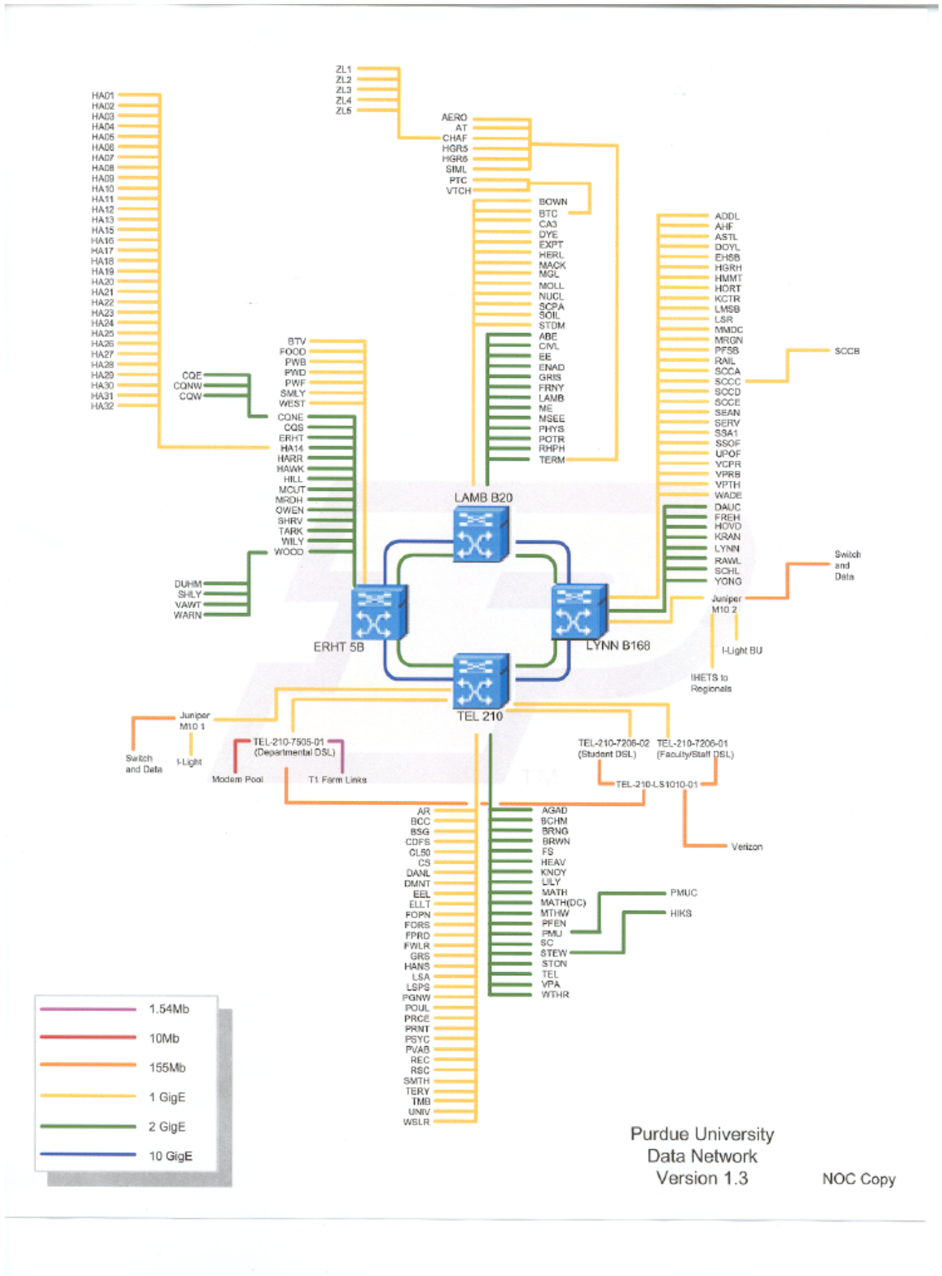
10^{12}, 10^9, 10^6, 10^3 bits per second; indicates data transmission rate; influenced by clock rate (THz/GHz/MHz/ of underlying hardware

→ communication speed: factors of 1000

→ data size: 1 KB means 1024 bytes

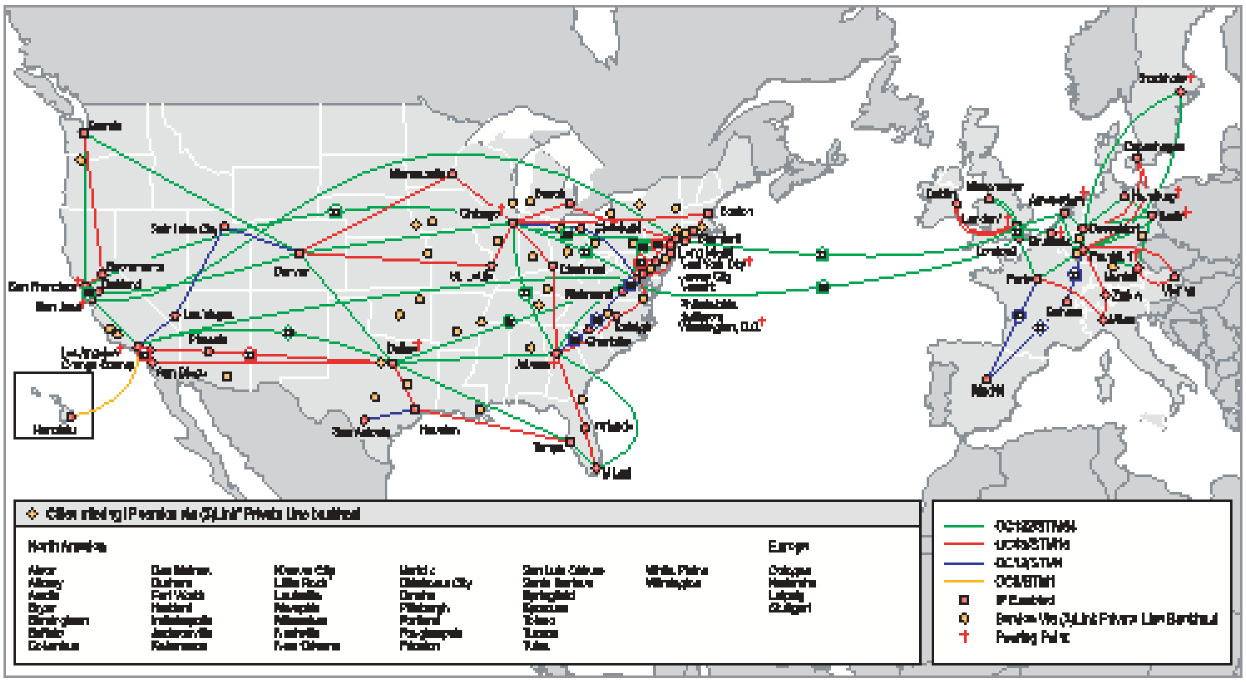
→ ballpark the same: exercise care when doing exact calculations

Example network pics: Purdue's backbone network



Level3 backbone network: www.level3.com

LEVEL 3 IP BACKBONE



→ 10 Gbps backbone (green): same speed as Purdue

→ outdated pic: faster backbone speeds now

What is traveling on the wires?

Mixture of:

bulk data (data, image, video, audio files), voice, streaming video/audio, real-time interactive data (e.g., games and some social media, etc.

→ around 90% of Internet traffic has been TCP file traffic

→ primarily a giant client/server system

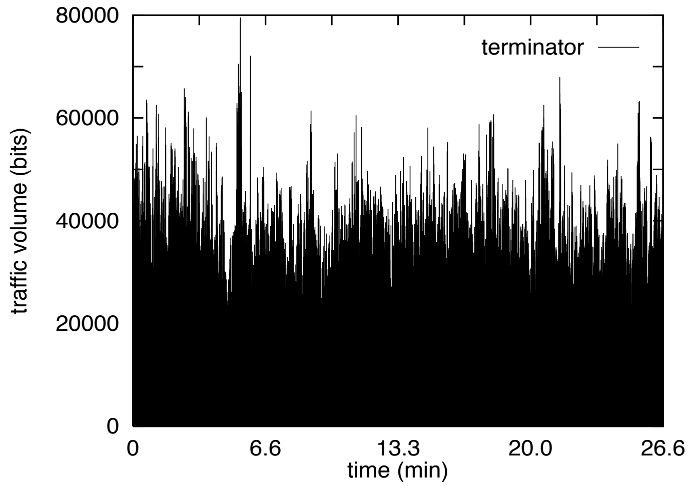
Multimedia (video/audio) streaming: rapid rise

→ streaming video: e.g., youtube, netflix

→ real-time: e.g., VoIP, video conferencing, games

→ target of traffic delimiting and shaping (e.g., fine print of "unlimited" data plans)

Internet traffic is "bursty": MPEG compressed real-time video



Reason:

- video compression

  → utilize inter-frame compression

- burstiness is not good for networks

  → why?

How to make sense of all this?

We will investigate three aspects:

- architecture

  → system design, real-world manifestation

- algorithms

  → how do the components work

- implementation

  → how are they actually implemented

A key concern and common thread: performance

→ slow means not being used in practice

→ performance heavily influences architecture, algorithm, implementation