Computer Networking: TL; DR

CS 352, Spring 2020

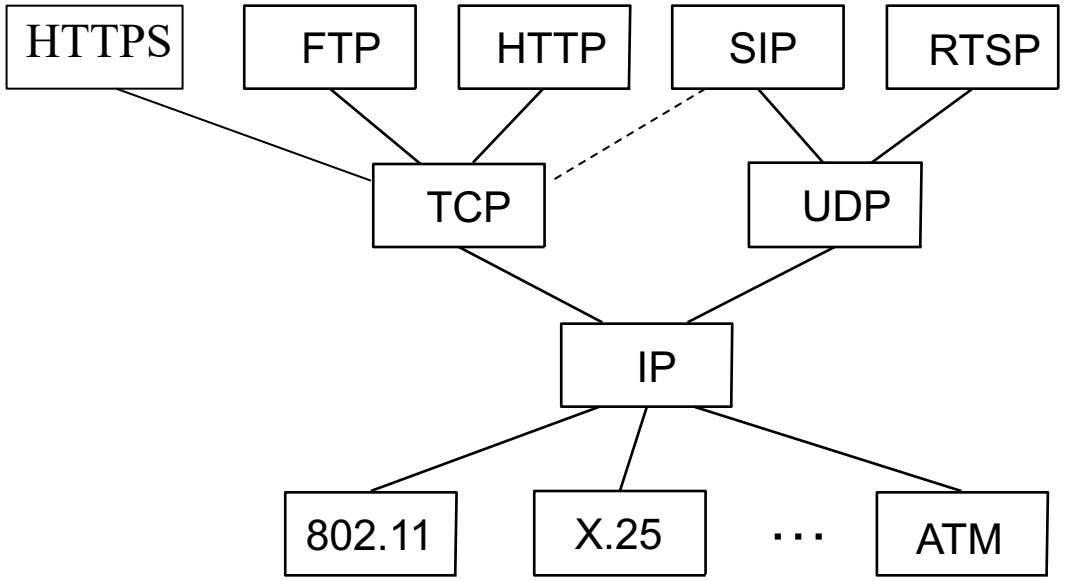
http://www.cs.rutgers.edu/~sn624/352

Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

The protocols of the Internet

- **Layering** and **Hourglass Design**



Protocol layers: Application layer

- Apps closest to the user: HTTP, SMTP, multimedia
- Helper protocols: DNS

- Deal with human concerns
- Readability of host names to reach certain services
- Often, protocols are in human readable plain text too
  - HTTP, SMTP, DNS
- Optimized for human-perceivable performance
  - The web, VoIP, mail, apps have different “optimizations” built into them

Protocol layers: Transport layer

- Providing guarantees for applications over a best-effort network
  - Transport layer runs on hosts
- Providing connectivity between applications
  - Multiplex data from multiple apps going out of a given machine
  - Demultiplex data coming into a machine to different apps
- UDP: main function is de/multiplexing
  - Also, error detection using checksums
  - Simple and lightweight
- TCP: reliable, in-order delivery
  - Much more heavyweight

Protocol layers: The transport layer

- TCP reliable delivery mechanisms
  - ACKnowledgments
  - Timeouts
  - Retransmission strategies
- TCP ordered delivery mechanisms
  - Sliding window
  - Flow control
- TCP efficiency and fairness
  - Congestion control: slow start, additive increase/multiplicative decrease

Protocol layers: The network layer

- Providing connectivity between machines across the Internet
  - Data plane: Moving data from point to point
  - Control plane: compute how data plane should move data

- Network layer runs on every host and every router

- Main issues: (1) Addressing
  - Machine addresses aren't necessarily human friendly (ex: IPv4/v6)
  - Addresses associated with network interfaces, not hosts

Protocol layers: The network layer

- Main issues: (2) Router design
  - High performance data movement between different network interfaces
  - High-speed destination-based forwarding
  - Longest-prefix-based matching

- Main issues: (3) Routing
  - Link-state, distance-vector, path-vector routing
  - Must try to avoid suboptimal paths and routing loops
  - Try to compute and converge fast

- Main issues: (4) QoS
  - Resolve contention at router queues
  - Priority queueing, fair queueing, leaky buckets, token buckets

Protocol Layers: The link layer

- Provide connectivity between machines over the physical medium
  - A co-ax cable, optic fiber, the air inside a room
- Main issues: (1) Data encoding
  - Must translate bits 0-1 from software into physical signals
- Main issues: (2) Error detection
  - the media are not without fault!
  - Parity bits

Protocol layers: The link layer

- Main issues: (3) Multiple access
  - Partitioning the medium's resources
  - Random access protocols: exponential back-off
  - Taking turns

- Main issues: (4) Handling nuances of wireless media
  - Fading, hidden terminals, half-duplex
  - Link-layer reliability
  - Waiting for fixed periods of time to transmit despite idle medium
  - Explicit reservation (RTS/CTS), resulting in "taking turns"

Multimedia transfers

- Streaming (ex: Netflix) and conversational (ex: Skype) media
- Peculiar characteristics:
  - Delay-sensitivity, loss tolerance, varying quality levels for same data
- Application-level adaptations:
  - Client-side buffering
  - Adaptive playout
- System-level adaptations:
  - Relay-based routing

The big picture

- Computer networks are a stack of layers
  - Built that way for modularity
  - Each layer does one set of functions very well
  - Each layer depends on the layers beneath it
  - But modularity can sometimes result in inefficiency

- Many general and useful principles
  - Borrowed from real life (ex: listen before you speak)
  - Borrowed from systems in general (ex: use indirection for flexibility)
  - Applicability goes the other way as well (ex: how to meter freeway ramps)