Computer Networking

Slide Set 1

Andrew W. Moore

Andrew.Moore@cl.cam.ac.uk
Topic 1 Foundation

- Administrivia
- Networks
- Channels
- Multiplexing
- Performance: loss, delay, throughput
Course Administration

Commonly Available Texts

- Computer Networks: A Systems Approach
  Peterson and Davie
  https://book.systemsapproach.org

Other Selected Texts (non-representative)

- Computer Networking: A Top-Down Approach
  Kurose and Ross, (many editions), Addison-Wesley

- Internetworking with TCP/IP, vol. I + II
  Comer & Stevens, Prentice Hall

  Stevens, Fenner & Rudoff, Prentice Hall
What is a network?

• A system of “links” that interconnect “nodes” in order to move “information” between nodes

• Yes, this is all rather abstract
What is a network?

- We also talk about

or

or even

- Yes, abstract, vague, and under-defined….
There are *many* different types of networks

- Internet
- Telephone network
- Transportation networks
- Cellular networks
- Supervisory control and data acquisition networks
- Optical networks
- Sensor networks

We will focus almost exclusively on the Internet
The Internet has transformed everything

• The way we do business
  – E-commerce, advertising, cloud-computing
• The way we have relationships
  – Facebook friends, E-mail, IM, virtual worlds
• The way we learn
  – Wikipedia, search engines
• The way we govern and view law
  – E-voting, censorship, copyright, cyber-attacks
The Internet transforms everything

- The way we do business – E-commerce, advertising, cloud computing
- The way we have relationships – Facebook friends, E-mail, IM, virtual worlds
- The way we learn – Wikipedia, search engines
- The way we govern and view law – E-voting, censorship, copyright, cyber attacks

Taking the dissemination of information to the next level
The Internet is big business

• Many large and influential networking companies
  – Huawei, Broadcom, AT&T, Verizon, Akamai, Cisco, …
  – $132B+ industry (carrier and enterprise alone)

• Networking central to most technology companies
  – Apple, Google, Facebook, Intel, Amazon, VMware, …
But why is the Internet interesting?

“What’s your formal model for the Internet?” -- theorists

“Aren’t you just writing software for networks” – hackers

“You don’t have performance benchmarks???” – hardware folks

“Isn’t it just another network?” – old timers at BT

“What’s with all these TLA protocols?” – all

“But the Internet seems to be working…” – my mother
A few defining characteristics of the Internet
A federated system

- The Internet ties together different networks
  - >20,000 ISP networks (the definition is fuzzy)

Tied together by IP -- the “Internet Protocol” : a single common interface between users and the network and between networks
A federated system

- The Internet ties together different networks
  - >20,000 ISP networks

- A single, common interface is great for interoperability…
- …but tricky for business

- Why does this matter?
  - ease of interoperability is the Internet’s most important goal
  - practical realities of incentives, economics and real-world trust, drive topology, route selection and service evolution
Tremendous scale
(2020 numbers – so some ‘weird’)

- 4.57 Billion users (58% of world population)
- 1.8 Billion web sites
  - 34.5% of which are powered by the WordPress!
- 4.88 Billion smartphones (45.4% of population)
- 500 Million Tweets a day
- 100 Billion WhatsApp messages per day
- 1 Billion hours of YouTube video watched per day
- 500 hours of Youtube video added per minute
- 2+ billion TikTok installs
- 60% video streaming
  - 12.5% of the Internet traffic is native Netflix
Tremendous scale
(2020 numbers – so some ‘weird’)

- 4.57 Billion users (58% of world population)
- 1.8 Billion web sites
  - 34.5% of which are powered by WordPress!
- 4.88 Billion smartphones (45.4% of population)
- 500 Million Tweets a day
- 100 Billion WhatsApp messages per day
- 1 Billion hours of YouTube video watched per day
- 500 hours of Youtube video added per minute
- 2+ billion TikTok installs
- 60% video streaming
  - 12.5% of the Internet traffic is native Netflix
Enormous diversity and dynamic range

• Communication latency: microseconds to seconds ($10^6$)
• Bandwidth: 1Kbits/second to 400 Gigabits/second ($10^7$)
• Packet loss: 0 – 90%

• Technology: optical, wireless, satellite, copper

• Endpoint devices: from sensors and cell phones to datacenters and supercomputers
• Applications: social networking, file transfer, skype, live TV, gaming, remote medicine, backup, IM
• Users: the governing, governed, operators, malicious, naïve, savvy, embarrassed, paranoid, addicted, cheap …
Constant Evolution

1970s:
• 56kilobits/second “backbone” links
• <100 computers, a handful of sites in the US (and one UK)
• Telnet and file transfer are the “killer” applications

Today
• 400+Gigabits/second backbone links
• 40B+ devices, all over the globe
  – 27B+ IoT devices alone
Asynchronous Operation

• Fundamental constraint: speed of light

• Consider:
  – How many cycles does your 3GHz CPU in Cambridge execute before it can possibly get a response from a message it sends to a server in Palo Alto?
    • Cambridge to Palo Alto: 8,609 km
    • Traveling at 300,000 km/s: 28.70 milliseconds
    • Then back to Cambridge: 2 x 28.70 = 57.39 milliseconds
    • 3,000,000,000 cycles/sec * 0.05739 = 172,179,999 cycles!

• Thus, communication feedback is always dated
Prone to Failure

• To send a message, all components along a path must function correctly
  – software, wireless access point, firewall, links, network interface cards, switches,…
  – Including human operators

• Consider: 50 components, that work correctly 99% of time → 39.5% chance communication will fail

• Plus, recall
  – scale → lots of components
  – asynchrony → takes a long time to hear (bad) news
  – federation (internet) → hard to identify fault or assign blame
Recap: The Internet is…

- A complex federation
- Of enormous scale
- Dynamic range
- Diversity
- Constantly evolving
- Asynchronous in operation
- Failure prone
- Constrained by what’s practical to engineer
- Too complex for (simple) theoretical models
- “Working code” doesn’t mean much
- Performance benchmarks are too narrow
An Engineered System

• Constrained by what technology is practical
  – Link bandwidths
  – Switch port counts
  – Bit error rates
  – Cost
  – …
Nodes and Links

Channels = Links
Peer entities = Nodes
Properties of Links (Channels)

- Bandwidth (capacity): “width” of the links
  - number of bits sent (or received) per unit time (bits/sec or bps)
- Latency (delay): “length” of the link
  - propagation time for data to travel along the link (seconds)
- Bandwidth-Delay Product (BDP): “volume” of the link
  - amount of data that can be “in flight” at any time
  - propagation delay × bits/time = total bits in link
Examples of Bandwidth-Delay

• Same city over a slow link:
  – BW\sim 100\text{Mbps}
  – Latency\sim 10\text{msec}
  – BDP \sim 10^{6}\text{bits} \sim 125\text{KBytes}

• Cross-Atlantic over fast link:
  – BW\sim 10\text{Gbps}
  – Latency\sim 100\text{msec}
  – BDP \sim 10^{9}\text{bits} \sim 125\text{MBytes}

• Intra Datacenter:
  – BW\sim 100\text{Gbps}
  – Latency\sim 30\text{usec}
  – BDP \sim 10^{6}\text{bits} \sim 375\text{KBytes}

• Intra Host:
  – BW\sim 100\text{Gbps}
  – Latency\sim 16\text{nsec}
  – BDP \sim 1600\text{bits} \sim 200\text{Bytes}
Packet Delay

Sending a 100B packet from A to B?

Time when that bit reaches B = 1/10^6 + 1/10^3s

The last bit reaches B at (800 x 1/10^6) + 1/10^3s = 1.8ms

Packet Delay = (Packet Size ÷ Link Bandwidth) + Link Latency

Time to transmit 800 bits = 800 x 1/10^6s

Time to transmit a 100B packet from A to B?
Sending a 100B packet from A to B?

The last bit in the file reaches B at
\[(10^7 \times 800 \times \frac{1}{10^9}) + \frac{1}{10^3}s\]
\[= 8001\text{ms}\]

The last bit reaches B at
\[(800 \times \frac{1}{10^9}) + \frac{1}{10^3}s\]
\[= 1.0008\text{ms}\]

The last bit reaches B at
\[(800 \times \frac{1}{10^6}) + \frac{1}{10^3}s\]
\[= 1.8\text{ms}\]
Packet Delay: The “pipe” view

Sending 100B packets from A to B?
Packet Delay: The “pipe” view

Sending 100B packets from A to B?

1Mbps, 10ms (BDP=10,000)

1Mbps, 5ms (BDP=5,000)

10Mbps, 1ms (BDP=10,000)
Packet Delay: The “pipe” view

Sending 100B packets from A to B?

What if we used 200Byte packets??
Recall Nodes and Links
What if we have more nodes?

One link for every node?

Need a **scalable** way to interconnect nodes
Solution: A switched network

Nodes *share* network link resources

How is this sharing implemented?
Two forms of switched networks

- Circuit switching (used in the POTS: Plain Old Telephone system)
- Packet switching (used in the Internet)
Circuit switching

Idea: source reserves network capacity along a path

1. Node A sends a reservation request
2. Interior switches establish a connection — i.e., “circuit”
3. A starts sending data
4. A sends a “teardown circuit” message
Sharing makes things efficient (cost less)
• One airplane/train for 100’s of people
• One telephone for many calls
• One lecture theatre for many classes
• One computer for many tasks
• One network for many computers
• One datacenter many applications
Multiplexing

Sharing makes things efficient (cost less)

• One airplane/train for 100’s of people
• One telephone for many calls
• One lecture theatre for many classes
• One computer for many tasks
• One network for many computers
• One datacenter many applications
Old Time Multiplexing
Circuit Switching: FDM and TDM

Frequency Division Multiplexing

Example:
4 users

Time Division Multiplexing

Radio Schedule
...,News, Sports, Weather, Local, News, Sports,...
Time-Division Multiplexing/Demultiplexing

- Time divided into frames; frames into slots
- Relative slot position inside a frame determines to which conversation data belongs
  - e.g., slot 0 belongs to orange conversation
- Slots are reserved (released) during circuit setup (teardown)
- If a conversation does not use its circuit capacity is lost!
Timing in Circuit Switching

Circuit Establishment

Transfer

Information

Circuit Tear-down
Circuit switching: pros and cons

• Pros
  – guaranteed performance
  – fast transfer (once circuit is established)

• Cons
Timing in Circuit Switching

- Circuit Establishment
- Transfer
- Circuit Tear-down
Circuit switching: pros and cons

• Pros
  – guaranteed performance
  – fast transfer (once circuit is established)

• Cons
  – wastes bandwidth if traffic is “bursty”
Timing in Circuit Switching

- **Circuit Establishment**
- **Transfer**
- **Circuit Tear-down**

Information
Timing in Circuit Switching

Circuit Establishment
Transfer
Circuit Tear-down
Circuit switching: pros and cons

- **Pros**
  - guaranteed performance
  - fast transfers (once circuit is established)

- **Cons**
  - wastes bandwidth if traffic is “bursty”
  - connection setup time is overhead
Circuit switching doesn’t “route around failure”
Circuit switching: pros and cons

• Pros
  – guaranteed performance
  – fast transfers (once circuit is established)

• Cons
  – wastes bandwidth if traffic is “bursty”
  – connection setup time is overhead
  – recovery from failure is slow
Numerical example

• How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  – All links are 1.536 Mbps
  – Each link uses TDM with 24 slots/sec
  – 500 msec to establish end-to-end circuit

Let’s work it out!

\[
\frac{1}{24} \times 1.536 \text{Mbps} = 64 \text{kb/s}
\]

\[
\frac{640,000}{64 \text{kb/s}} = 10 \text{s}
\]

\[
10 \text{s} + 500 \text{ms} = 10.5 \text{s}
\]
Two forms of switched networks

• Circuit switching (e.g., telephone network)
• Packet switching (e.g., Internet)
Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”*

* After Nick McKeown © 2006
Packet Switching

• Data is sent as chunks of formatted bits (Packets)
• Packets consist of a “header” and “payload”*
  – payload is the data being carried
  – header holds instructions to the network for how to handle packet (think of the header as an API)
    – In this example, the header has a destination address
    – More complex headers may include
      • How this traffic should be handled? (first class, second class, etc)
      • Who signed for it?
      • Were the contents ok?
Packet Switching

• Data is sent as chunks of formatted bits (Packets)
• Packets consist of a “header” and “payload”
• Switches “forward” packets based on their headers
Switches forward packets

Forwarding Table

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLASGOW</td>
<td>4</td>
</tr>
<tr>
<td>OXFORD</td>
<td>5</td>
</tr>
<tr>
<td>EDIN</td>
<td>2</td>
</tr>
<tr>
<td>UCL</td>
<td>3</td>
</tr>
</tbody>
</table>
What about the time to process the packet at the switch?

- We’ll assume it’s relatively negligible (mostly true)
Could the switch start transmitting as soon as it has processed the header?

- Yes! This would be called a “cut through” switch
We will always assume a switch processes/forwards a packet after it has received it entirely. This is called "store and forward" switching.
Packet Switching

• Data is sent as chunks of formatted bits (Packets)
• Packets consist of a “header” and “payload”
• Switches “forward” packets based on their headers
Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”
- Switches “forward” packets based on their headers
- Each packet travels independently
  – no notion of packets belonging to a “circuit”
Packet Switching

• Data is sent as chunks of formatted bits (Packets)
• Packets consist of a “header” and “payload”
• Switches “forward” packets based on their headers
• Each packet travels independently
• No link resources are reserved in advance. Instead packet switching leverages statistical multiplexing (stat muxing)
Multiplexing

Sharing makes things efficient (cost less)

• One airplane/train for 100’s of people
• One telephone for many calls
• One lecture theatre for many classes
• One computer for many tasks
• One network for many computers
• One datacenter many applications
Three Flows with Bursty Traffic

Data Rate 1

Data Rate 2

Data Rate 3

Capacity
When Each Flow Gets $1/3^{rd}$ of Capacity

Frequent Overloading
When Flows Share Total Capacity

Statistical multiplexing relies on the assumption that not all flows burst at the same time.

Very similar to insurance, and has same failure case.
Three Flows with Bursty Traffic

Data Rate 1

Data Rate 2

Data Rate 3

Capacity
Three Flows with Bursty Traffic

Data Rate 1

Data Rate 2

Data Rate 3

Capacity
Three Flows with Bursty Traffic

Data Rate 1+2+3 >> Capacity

What do we do under overload?
Sorry we don’t carry https here....
Statistical multiplexing: pipe view
Statistical multiplexing: pipe view
Statistical multiplexing: pipe view

No Overload
Statistical multiplexing: pipe view

Queue overload into Buffer

Transient Overload
Not such a rare event
Statistical multiplexing: pipe view

Queue overload into Buffer

Transient Overload
Not such a rare event
Statistical multiplexing: pipe view

Transient Overload
Not such a rare event
Statistical multiplexing: pipe view

Transient Overload
Not such a rare event

Queue overload into Buffer
Statistical multiplexing: pipe view

Queue overload into Buffer

Transient Overload
Not such a rare event
Statistical multiplexing: pipe view

Queue overload into Buffer

Buffer absorbs transient bursts
But NOT additional capacity
Statistical multiplexing: pipe view

Queue overload into Buffer

What about persistent overload?
Will eventually drop packets
Queues introduce queuing delays

- Recall,

  \[ \text{packet delay} = \text{transmission delay} + \text{propagation delay} \] (*

- With queues (statistical multiplexing)

  \[ \text{packet delay} = \text{transmission delay} + \text{propagation delay} + \text{queuing delay} \] (*

- Queuing delay caused by “packet interference”

- Made worse at high load
  - less “idle time” to absorb bursts
  - think about traffic jams at rush hour
    or rail network failure

  (* plus per-hop \textit{processing} delay that we define as negligible)
Queuing delay extremes

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

Traffic intensity $= \frac{La}{R}$

- $\frac{La}{R} \sim 0$: average queuing delay small
- $\frac{La}{R} \rightarrow 1$: delays become large
- $\frac{La}{R} > 1$: more “work” arriving than can be serviced, average delay infinite – or data is lost (dropped).
Recall the Internet *federation*

- The Internet ties together different networks  
  - >20,000 ISP networks

We can see (hints) of the nodes and links using traceroute...
"Real" Internet delays and routes

**traceroute**: rio.cl.cam.ac.uk to munnari.oz.au

(tracepath on windows is similar)

traceroute munnari.oz.au
traceroute to munnari.oz.au (202.29.151.3), 30 hops max, 60 byte packets

1  gatwick.net.cl.cam.ac.uk (128.232.32.2) 0.416 ms 0.384 ms 0.427 ms
2  cl-sby.route-nwest.net.cam.ac.uk (193.60.89.9) 0.393 ms 0.440 ms 0.494 ms
3  route-nwest.route-mill.net.cam.ac.uk (192.84.5.137) 0.407 ms 0.448 ms 0.501 ms
4  route-mill.route-enet.net.cam.ac.uk (192.84.5.94) 1.006 ms 1.091 ms 1.163 ms
5  xe-11-3-0.camb-rbr1.eastern.ja.net (146.97.130.1) 0.300 ms 0.313 ms 0.350 ms
6  ae24.lowdss-sbr1.ja.net (146.97.37.185) 2.679 ms 2.664 ms 2.712 ms
7  ae28.londhx-sbr1.ja.net (146.97.33.17) 5.955 ms 5.953 ms 5.901 ms
8  janet.mx1.lon.uk.geant.net (62.40.124.197) 6.059 ms 6.066 ms 6.052 ms
9  ae0.mx1.par.fr.geant.net (62.40.98.77) 11.742 ms 11.779 ms 11.724 ms
10  ae1.mx1.mad.es.geant.net (62.40.98.64) 27.751 ms 27.734 ms 27.704 ms
11  mb-so-02-v4.bb.tein3.net (202.179.249.117) 138.296 ms 138.314 ms 138.282 ms
12  sg-so-04-v4.bb.tein3.net (202.179.249.53) 196.303 ms 196.293 ms 196.264 ms
13  th-pr-v4.bb.tein3.net (202.179.249.66) 225.153 ms 225.178 ms 225.196 ms
14  pyt-thairen-to-02-bdr-pyt.uni.net.th (202.29.12.10) 225.163 ms 223.343 ms 223.363 ms
15  202.28.227.126 (202.28.227.126) 241.038 ms 240.941 ms 240.834 ms
16  202.28.221.46 (202.28.221.46) 287.252 ms 287.306 ms 287.282 ms
17  ***
18  ***  * means no response (probe or reply lost, router not replying)
19  ***
20  coe-gw.psu.ac.th (202.29.149.70) 241.681 ms 241.715 ms 241.680 ms
21  munnari.OZ.AU (202.29.151.3) 241.610 ms 241.636 ms 241.537 ms
Internet structure: network of networks

- a packet passes through many networks!
Internet structure: network of networks

- “Tier-3” ISPs and local ISPs
  - last hop (“access”) network (closest to end systems)

Local and tier-3 ISPs are customers of higher tier ISPs connecting them to rest of Internet
Internet structure: network of networks

• “Tier-2” ISPs: smaller (often regional) ISPs
  – Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs

Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet

- tier-2 ISP is *customer* of tier-1 provider
Internet structure: network of networks

• roughly hierarchical
• at center: “tier-1” ISPs (e.g., Verizon, Sprint, AT&T, Cable and Wireless), national/international coverage
  – treat each other as equals

![Diagram of Internet structure]

Tier-1 providers interconnect (peer) privately
Tier-1 ISP: e.g., Sprint
Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”
- Switches “forward” packets based on their headers
- Each packet travels independently
- No link resources are reserved in advance. Instead, packet switching leverages *statistical multiplexing*
  - allows efficient use of resources
  - but introduces queues and queuing delays
Packet switching versus circuit switching

Packet switching may (does!) allow more users to use network

- 1 Mb/s link
- each user:
  - 100 kb/s when “active”
  - active 10% of time

- circuit-switching:
  - 10 users

- packet switching:
  - with 35 users, probability > 10 active at same time is less than .0004

Q: how did we get value 0.0004?
Packet switching versus circuit switching

Q: how did we get value 0.0004?

- 1 Mb/s link
- each user:
  - 100 kb/s when “active”
  - active 10% of time

- circuit-switching:
  - 10 users

- packet switching:
  - with 35 users, probability > 10 active at same time is less than .0004

Let U be number of users active
N the total users
P is 0.1 in our example to get 0.0004
\[
P(U = k) = \binom{n}{k} p^k (1 - p)^{n-k}
\]

\[
\therefore P(U \leq k) = \sum_{k=0}^{K} \binom{n}{k} p^k (1 - p)^{n-k} \quad \left[ P(U > K) = 1 - \sum_{k=0}^{K} \binom{n}{k} p^k (1 - p)^{n-k} \right]
\]

For \( n = 35, \; K = 10 \)

\[
P(U \leq 10) = \sum_{k=0}^{10} \binom{35}{k} p^k (1 - p)^{35-k}
\]

Where \( p = 0.1 \):

\[
P(U \leq 10) = 0.99958
\]

\[
P(U > 10) = 0.00042
\]
Circuit switching: pros and cons

• Pros
  – guaranteed performance
  – fast transfers (once circuit is established)

• Cons
  – wastes bandwidth if traffic is “bursty”
  – connection setup adds delay
  – recovery from failure is slow
Packet switching: pros and cons

• Cons
  – no guaranteed performance
  – header overhead per packet
  – queues and queuing delays

• Pros
  – efficient use of bandwidth (stat. muxing)
  – no overhead due to connection setup
  – resilient -- can `route around trouble’
Summary

• A sense of how the basic `plumbing’ works
  – links and switches
  – packet delays = transmission + propagation + queuing + (negligible) per-switch processing
  – statistical multiplexing and queues
  – circuit vs. packet switching