Chapter 1
Introduction

A note on the use of these PowerPoint slides:
We’re making these slides freely available to all (faculty, students, readers). They’re in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a lot of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we’d like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

For a revision history, see the slide note for this page.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2020
J.F Kurose and K.W. Ross, All Rights Reserved

The source PowerPoint slides are public available, provided by Authors (JFK/KWR). They are revised for CS536@Purdue.
Chapter 1: introduction

Chapter goal:
- Get “feel,” “big picture,” introduction to terminology
  - more depth, detail later in course

Overview/roadmap:
- What is the Internet? What is a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Protocol layers, service models
What is the Internet?
The Internet: a “nuts and bolts” view

Billions of connected computing devices:
- *hosts* = end systems
- running *network apps* at Internet’s “edge”

*Packet switches*: forward packets (chunks of data)
- routers, switches

*Communication links*
- fiber, copper, radio, satellite
- transmission rate: *bandwidth*

*Networks*
- collection of devices, routers, links: managed by an organization
The Internet: a “nuts and bolts” view

- **Internet**: “network of networks”
  - Interconnected ISPs

- **Protocols are everywhere**
  - control sending, receiving of messages
  - e.g., HTTP (Web), streaming video, Skype, TCP, IP, WiFi, 4G, Ethernet

- **Internet standards**
  - RFC: Request for Comments
  - IETF: Internet Engineering Task Force
The Internet: a “services” view

- **Infrastructure** that provides services to applications:
  - Web, streaming video, multimedia teleconferencing, email, games, e-commerce, social media, interconnected appliances, ...

- provides **programming interface** to distributed applications:
  - “hooks” allowing sending/receiving apps to “connect” to, use Internet transport service
  - provides service options, analogous to postal service
What’s a protocol?

**Human protocols:**
- “what’s the time?”
- “I have a question”
- introductions

**Network protocols:**
- computers (devices) rather than humans
- all communication activity in Internet governed by protocols

Rules for:
- specific messages sent
- specific actions taken when message received, or other events

*Protocols define the format, order of messages sent and received among network entities, and actions taken on message transmission, receipt*
What’s a protocol?

A human protocol and a computer network protocol:

**Q:** other human protocols?
Chapter 1: roadmap

- What is the Internet?
- What is a protocol?
- **Network edge**: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
A closer look at Internet structure

Network edge:
- hosts: clients and servers
- servers often in data centers
A closer look at Internet structure

Network edge:
- hosts: clients and servers
- servers often in data centers

Access networks, physical media:
- wired, wireless communication links
A closer look at Internet structure

Network edge:
- hosts: clients and servers
- servers often in data centers

Access networks, physical media:
- wired, wireless communication links

Network core:
- interconnected routers
- network of networks
Access networks and physical media

Q: How to connect end systems to edge router?

- residential access nets
- institutional access networks (school, company)
- mobile access networks (WiFi, 4G/5G)

Introduction: 1-13
Access networks: cable-based access

- **HFC: hybrid fiber coax**
  - asymmetric: up to 40 Mbps – 1.2 Gbps downstream transmission rate, 30-100 Mbps upstream transmission rate

- **network of cable, fiber attaches homes to ISP router**
  - homes *share access network* to cable headend

---

Introduction: 1-15
Wireless access networks

Shared *wireless* access network connects end system to router
- via base station aka “access point”

Wireless local area networks (WLANs)
- typically within or around building (~100 ft)
- 802.11b/g/n/ac/be (WiFi): up to 46 Gbps transmission rate
- Also called WiFi 5, 6, 6e and 7

Wide-area cellular access networks
- provided by mobile, cellular network operator (10’s km)
- 10 - 1000 Mbps
- 5G/4G cellular networks

Introduction: 1-18
Access networks: enterprise networks

- companies, universities, etc.
- mix of wired, wireless link technologies, connecting a mix of switches and routers (we’ll cover differences shortly)
  - Ethernet: wired access at 100Mbps, 1Gbps, 10Gbps
  - WiFi: wireless access points at 11, 54, 450 Mbps, 3.5Gbps, 46 Gbps
Access networks: data center networks

- high-bandwidth links (10s to 100s Gbps) connect hundreds to thousands of servers together, and to Internet

Courtesy: Massachusetts Green High Performance Computing Center (mghpcc.org)
Chapter 1: roadmap

- What is the Internet?
- What is a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History
The network core

- mesh of interconnected routers
- packet-switching: hosts break application-layer messages into packets
  - network forwards packets from one router to the next, across links on path from source to destination
Two key network-core functions

Forwarding:
- aka “switching”
- local action: move arriving packets from router’s input link to appropriate router output link

Routing:
- global action: determine source-destination paths taken by packets
- routing algorithms

<table>
<thead>
<tr>
<th>header value</th>
<th>output link</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>3</td>
</tr>
<tr>
<td>0101</td>
<td>2</td>
</tr>
<tr>
<td>0111</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1</td>
</tr>
</tbody>
</table>
Packet-switching: store-and-forward

- **Packet transmission delay:** takes $L/R$ seconds to transmit (push out) $L$-bit packet into link at $R$ bps
- **Store and forward:** entire packet must arrive at router before it can be transmitted on next link

**One-hop numerical example:**
- $L = 10$ Kbits
- $R = 100$ Mbps
- one-hop transmission delay = 0.1 msec
Packet-switching: queueing

Queueing occurs when work arrives faster than it can be serviced:
Packet-switching: queueing

Packet queuing and loss: if arrival rate (in bps) to link exceeds transmission rate (bps) of link for some period of time:

- packets will queue, waiting to be transmitted on output link
- packets can be dropped (lost) if memory (buffer) in router fills up
Alternative to packet switching: circuit switching

end-end resources allocated to, reserved for “call” between source and destination

- in diagram, each link has four circuits.
  - call gets 2\textsuperscript{nd} circuit in top link and 1\textsuperscript{st} circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idle if not used by call (no sharing)
- commonly used in traditional telephone networks

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive
Packet switching versus circuit switching

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

Q: how many users can use this network under circuit-switching and packet switching?

- 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?
A: A math problem (for those with course in probability only)

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive
Packet switching versus circuit switching

Is packet switching a “slam dunk winner”? 

- great for “bursty” data – sometimes has data to send, but at other times not
  - resource sharing
  - simpler, no call setup
- excessive congestion possible: packet delay and loss due to buffer overflow
  - protocols needed for reliable data transfer, congestion control
- **Q:** How to provide circuit-like behavior with packet-switching? 
  - “It’s complicated.” We’ll study various techniques that try to make packet switching as “circuit-like” as possible.

**Q:** human analogies of reserved resources (circuit switching) versus on-demand allocation (packet switching)?
Chapter 1: roadmap

- What is the Internet?
- What is a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History
How do packet delay and loss occur?

- packets *queue* in router buffers, waiting for turn for transmission
  - queue length grows when arrival rate to link (temporarily) exceeds output link capacity
- packet *loss* occurs when memory to hold queued packets fills up

Packet delay and loss are illustrated in the diagram.

- Packet being transmitted: *transmission delay*
- Packets in buffers: *queueing delay*
- Free (available) buffers: arriving packets dropped (loss) if no free buffers

Introduction: 1-37
Packet delay: four sources

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

$d_{\text{proc}}$: nodal processing
- check bit errors
- determine output link
- typically < microsecs

$d_{\text{queue}}$: queueing delay
- time waiting at output link for transmission
- depends on congestion level of router
Packet delay: four sources

\[ d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}} \]

\( d_{\text{trans}} \): transmission delay:
- \( L \): packet length (bits)
- \( R \): link transmission rate (bps)
- \( d_{\text{trans}} = L/R \)

\( d_{\text{prop}} \): propagation delay:
- \( d \): length of physical link
- \( s \): propagation speed (~2x10^8 m/sec)
- \( d_{\text{prop}} = d/s \)

\( d_{\text{trans}} \) and \( d_{\text{prop}} \) very different
Caravan analogy

- car ~ bit; caravan ~ packet; toll service ~ link transmission
- toll booth takes 12 sec to service car (bit transmission time)
- “propagate” at 100 km/hr
- Q: How long until caravan is lined up before 2nd toll booth?

- time to “push” entire caravan through toll booth onto highway = 12*10 = 120 sec
- time for last car to propagate from 1st to 2nd toll booth: 100km/(100km/hr) = 1 hr
- A: 62 minutes
Caravan analogy

- suppose cars now “propagate” at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- **Q**: Will cars arrive to 2nd booth before all cars serviced at first booth?
  
  **A: Yes!** after 7 min, first car arrives at second booth; three cars still at first booth
Packet queueing delay (revisited)

- $a$: average packet arrival rate
- $L$: packet length (bits)
- $R$: link bandwidth (bit transmission rate)

\[
\frac{L \cdot a}{R} : \frac{\text{arrival rate of bits}}{\text{service rate of bits}} \quad \text{“traffic intensity”}
\]

- $La/R \sim 0$: avg. queueing delay small
- $La/R \to 1$: avg. queueing delay large
- $La/R > 1$: more “work” arriving is more than can be serviced - average delay infinite!
“Real” Internet delays and routes

- what do “real” Internet delay & loss look like?

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 (with time-to-live field value of $i$)
- router $i$ will return packets to sender
- sender measures time interval between transmission and reply
“Real” Internet delays, routes

- traceroute: www.google.com

1 192.168.0.1 (192.168.0.1) 2.079 ms 1.292 ms 1.299 ms
2 96.120.112.229 (96.120.112.229) 10.446 ms 10.082 ms 9.130 ms
3 96.110.168.189 (96.110.168.189) 9.258 ms 8.858 ms 9.012 ms
4 be-22-ar01.indianapolis.in.indiana.comcast.net (68.86.188.97) 17.090 ms
   16.266 ms 16.521 ms
5 be-3-ar01.area4.il.chicago.comcast.net (68.86.188.181) 29.714 ms 28.072 ms
   28.933 ms
6 be-33491-cr02.350ecermak.il.ibone.comcast.net (68.86.91.165) 30.148 ms
   28.974 ms 29.939 ms
7 be-10577-pe03.350ecermak.il.ibone.comcast.net (68.86.86.2) 28.834 ms 28.609
   ms 30.490 ms
8 173.167.56.22 (173.167.56.22) 27.929 ms 29.434 ms 28.429 ms
9  **

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

1. traceroute www.google.com
2. traceroute europa.eu

- What are differences you can see?
- Can you see the link across the ocean?
- Why?
Networking tools: Packet sniffer and analyzer

- **tcpdump** (command)
  - `> tcpdump -i en0`
  - `> tcpdump -i en0 -c 10 -w test.cap`
  - `> tcpdump -r test.cap`

- **wireshark** (UI)
Packet loss

- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all

* Check out the Java applet for an interactive animation (on publisher’s website) of queuing and loss
Throughput

- **throughput**: rate (bits/time unit) at which bits are being sent from sender to receiver
  - *instantaneous*: rate at given point in time
  - *average*: rate over longer period of time

![Diagram showing server sending bits (fluid) into a pipe that can carry fluid at rate $R_s$ bits/sec, followed by another pipe that can carry fluid at rate $R_c$ bits/sec to a client.]
**Throughput**

*What is average end-end throughput?*

**Rs < Rc**

(link representation)

**Rs > Rc**

(link representation)

*bottleneck link*

(link representation)

link on end-end path that constrains end-end throughput
Throughput: network scenario

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

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

* Check out the online interactive exercises for more examples: [http://gaia.cs.umass.edu/kurose_ross/](http://gaia.cs.umass.edu/kurose_ross/)
Chapter 1: roadmap

- What is the Internet?
- What is a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History
Protocol “layers” and reference models

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?
- and/or our discussion of networks?
Why layering?

Approach to designing/discussing complex systems:

- explicit structure allows identification, relationship of system’s pieces
  - layered *reference model* for discussion

- modularization eases maintenance, updating of system
  - change in layer's service *implementation*: transparent to rest of system
  - e.g., change in gate procedure doesn’t affect rest of system
Layered Internet protocol stack

- **application**: supporting network applications
  - HTTP, IMAP, SMTP, DNS

- **transport**: process-process data transfer
  - TCP, UDP

- **network**: routing of datagrams from source to destination
  - IP, routing protocols

- **link**: data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi), PPP

- **physical**: bits “on the wire”
Application exchanges messages to implement some application service using services of transport layer

Transport-layer protocol transfers M (e.g., reliably) from one process to another, using services of network layer

- transport-layer protocol encapsulates application-layer message, M, with transport layer-layer header $H_t$ to create a transport-layer segment
  - $H_t$ used by transport layer protocol to implement its service
Services, Layering and Encapsulation

Transport-layer protocol transfers M (e.g., reliably) from one process to another, using services of network layer

Network-layer protocol transfers transport-layer segment \[H_t | M\] from one host to another, using link layer services

- network-layer protocol encapsulates transport-layer segment \[H_t | M\] with network layer-layer-layer header \(H_n\) to create a network-layer datagram
  - \(H_n\) used by network layer protocol to implement its service
Services, Layering and Encapsulation

- **Network-layer** protocol transfers transport-layer segment \([H_t \mid M]\) from one host to another, using link layer services.
- **Link-layer** protocol transfers datagram \([H_n \mid H_t \mid M]\) from host to neighboring host, using network-layer services.

- Link-layer protocol **encapsulates** network datagram \([H_n \mid H_t \mid M]\), with link-layer header \(H_t\) to create a link-layer frame.
Services, Layering and Encapsulation

source

application
transport
network
link
physical

message
segment
datagram
frame

H_t | M

H_n | H_t | M

H_l | H_n | H_t | M

destination

application
transport
network
link
physical

H_t | M

H_n | H_t | M

H_l | H_n | H_t | M

Introduction: 1-59
Encapsulation: an end-end view

Introduction: 1-60
Part I: Layering in Internet protocol stack

Applications
... built on ...

Reliable (or unreliable) transport
... built on ...

Best-effort global packet delivery
... built on ...

Best-effort local packet delivery
... built on ...

Physical transfer of bits

Source: Scott Shenker (UC Berkeley): slide 7 at The Future of Networking, and the Past of Protocols
https://www.youtube.com/watch?v=YHeyuD89n1Y&t=111s