An overview

Chapter 1 from the Kurose and Ross textbook

Most of these slides are adapted from (or perhaps identical to) the authors' slides. All material copyright 1996-2012 J.F Kurose and K.W. Ross.

This presentation has been designed using resources from PoweredTemplate.com
What is the Internet?

- A nuts and bolts view vs. a service view

A closer look at the Internet

- **Network edge:**
  - Hosts are clients and servers
  - Servers are often in data centers

- **Access networks, physical media:**
  - Wired, wireless communication links

- **Network core:**
  - Interconnected routers
  - Network of networks
Network core

• Mesh of interconnected routers

• Packet-switching: hosts break application-layer messages into packets
  – Forward packets from one router to the next, across links on path from source to destination
  – Each packet transmitted at full link capacity

Remember Ma Bell?

• Well, there are two fundamental approaches to building a network core, and Ma Bell took the first one, circuit-switching.

• Circuit-switched networks, reserve resources along the length of the communication path for the duration of the communication session.

• Bandwidth is divided among the circuits.
Sharing bandwidth

FDMA

Example:
4 users

TDMA

Alternative core: circuit switching

- End-end resources allocated to, reserved for “call” between source & dest

- In diagram, each link has four circuits.
  - call gets 2nd circuit in top link and 1st circuit in right link.
- Dedicated resources: no sharing
- Circuit segment idle if not used by call (no sharing)
- Commonly used in traditional telephone networks
The Internet chose a different path

Packet switching

- Messages are broken into packets each of which travel from the source to destination through a maze of routers and links.

- Most routers are store-and-forward, meaning the switch must receive the entire packet before it can transmit it outbound link.
Why packet switching?

*Packet switching allows more users to use network!*

Example:
- 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 *

* Check out the online interactive exercises for more examples

---

Is packet switching a “slam dunk winner?”

- Great for bursty data
  - resource sharing
  - simpler, no call setup

- Excessive congestion possible: packet delay and loss
  - protocols needed for reliable data transfer, congestion control

- Q: How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - still an unsolved problem (chapter 7)

- Q: Human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?
The Internet as a Network of Networks

Internet structure: network of networks

- Question:
  - Given millions of access ISPs, how to connect them together?
Internet structure: network of networks

- **Option:**
  - connect each access ISP to every other access ISP?

  connecting each access ISP to each other directly doesn’t scale: \(O(N^2)\) connections.

Option: connect each access ISP to a global transit ISP?
Customer and provider ISPs have economic agreement.
Internet structure: network of networks

But if one global ISP is viable business, there will be competitors!

Internet structure: network of networks

But if one global ISP is viable business, there will be competitors! .... which must be interconnected

Internet exchange point

peering link
Internet structure: network of networks

... and regional networks may arise to connect access nets to ISPS.

Internet structure: network of networks

... and content provider networks (e.g., Google, Microsoft, Akamai) may run their own network, to bring services, content close to end users.
The Internet is a perfectly stable medium for sending data between endpoints
True or False?

FALSE!

- Everything is unreliable.
  - That is why networks research is so exciting and never ending!

- Due to the limitations of the physical infrastructure, throughput is limited.
  - This causes delays in communication and often loss of data

- Also, the distributed nature of the Internet makes it inherently unsecure.
Performance metrics in packet-switched networks

The story of the four delays …

Four sources of packet delay

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

\( d_{\text{proc}} \): nodal processing
- check bit errors
- determine output link
- typically < msec

\( d_{\text{queue}} \): queueing delay
- time waiting at output link for transmission
- depends on congestion level of router
Four sources of packet delay

\[ d_{\text{trans}} = \frac{L}{R} \]

Transmission delay:
- \( L \): packet length (bits)
- \( R \): link bandwidth (bps)
- \( d_{\text{trans}} = \frac{L}{R} \)

\[ d_{\text{prop}} = \frac{d}{s} \]

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

Queueing delay

- \( R \): link bandwidth (bps)
- \( L \): packet length (bits)
- \( a \): average packet arrival rate

\[ v = \frac{L a}{R} \]

- \( v = 0 \): avg. queueing delay small
- \( v > 1 \): avg. queueing delay large
- \( v > 1 \): more “work” arriving 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
  - router i will return packets to sender
  - sender times interval between transmission and reply.

3 probes

Real Internet delays, routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

1 cs-gw (128.119.240.254) 1 ms 1 ms 2 ms
2 border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145) 1 ms 1 ms 2 ms
3 cht-vbns.gw.umass.edu (128.119.3.130) 6 ms 5 ms 5 ms
4 jn1-at1-0-0-19.wor.vbns.net (204.147.132.129) 16 ms 11 ms 13 ms
5 jn1-so7-0-0-0.wae.vbns.net (204.147.136.136) 21 ms 18 ms 18 ms
6 auburn-vbns.abilene.edu (198.32.11.9) 22 ms 18 ms 22 ms
7 nycm-wash.abilene.edu (198.32.8.46) 22 ms 22 ms 22 ms
8 62.40.103.253 (62.40.103.253) 104 ms 109 ms 106 ms
9 de2-1.de1.de.geant.net (62.40.96.129) 109 ms 102 ms 104 ms
10 de.fr1.geant.net (62.40.96.50) 113 ms 121 ms 114 ms
11 renater-gw.fr1.fr.geant.net (62.40.103.54) 112 ms 114 ms 112 ms
12 nio-n2.cssi.renater.fr (193.51.206.13) 111 ms 114 ms 116 ms
13 nice.csi.renater.fr (195.220.98.102) 123 ms 125 ms 124 ms
14 r3t2-nice.cssi.renater.fr (195.220.98.110) 126 ms 126 ms 124 ms
15 eurecom-valbonne.r3t2.ft.net (193.48.50.54) 135 ms 128 ms 133 ms
16 194.214.211.25 (194.214.211.25) 126 ms 128 ms 126 ms
17 * * *
18 * * *
19 fantasia.eurecom.fr (193.55.113.142) 132 ms 128 ms 136 ms

* means no response (probe lost, router not replying)

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

Throughput

- Throughput: rate (bits/time unit) at which bits transferred between sender/receiver
  - instantaneous: rate at given point in time
  - average: rate over longer period of time
Throughput (more)

- Rs < Rc What is average end-end throughput?

\[ R_s \text{ bits/sec} \rightarrow R_c \text{ bits/sec} \]

- \( R_s > R_c \) What is average end-end throughput?

\[ R_s \text{ bits/sec} \rightarrow R_c \text{ bits/sec} \]

bottleneck link

link on end-end path that constrains end-end throughput

Throughput: Internet 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
The Internet Layers

Managing complexity

• Is there any hope of understanding anything as complex as the Internet?

• Captain Abstraction thinks so.

• To reduce complexity, network hardware and software that implement protocols are designed in layers.
The Internet protocol stack

- **Application layer** supports network applications (HTTP, SMTP, FTP).
- **Transport layer** provides transport of messages between client and server (TCP and UDP).
- **Network layer** is responsible for routing datagrams from one host to another (IP).
- **Link layer** moves frames from node to node (Ethernet, PPP).
- **Physical layer** moves individual bits of within a frame from node to node.

Up and down the protocol stack*

*Encapsulation.
It is important to distinguish between network applications and application layer protocols.

An application-layer protocol is only one piece of a network application.

A Web application consists of many components, including HTML, browsers, servers, and HTTP.

An application-layer protocol defines:
- Type of message exchanged.
- The syntax of the various message types.
- The semantics of each message field.
- Rules for determining when and how a process sends messages and responds to messages.

Public domain application-layer protocols are specified in RFC (Request for Comments).

Proprietary protocols play their cards close to the vest.
Process communication

- To build a network application, we must understand how processes* running in multiple end systems communicate?
  - Inter-process communication is governed by the end system’s operating system (CS341).
  - Communication between end systems is accomplished by exchanging messages across a computer network (CS242).
  - Coordination of a computation across independent machines in a network (CS343).

*Roughly speaking a process is a program in execution within an end system.

Network Security
Network security

• Field of network security studies:
  – How bad guys can attack computer networks
  – How we can defend networks against attacks
  – How to design architectures that are immune to attacks

• Internet not originally designed with (much) security in mind
  – Original vision: “a group of mutually trusting users attached to a transparent network” 😊
  – Internet protocol designers playing “catch-up”
  – Security considerations in all layers!

Bad guys: put malware into hosts via Internet

• Malware can get in host from:
  – Virus: self-replicating infection by receiving/executing object (e.g., e-mail attachment)
  – Worm: self-replicating infection by passively receiving object that gets itself executed

• Spyware malware can record keystrokes, web sites visited, upload info to collection site

• Infected host can be enrolled in botnet, used for spam.
Bad guys: attack server, network infrastructure

Denial of Service (DoS): attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic.

1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts

Bad guys can sniff packets

- Media is broadcasted over shared ethernet and wireless connections.
- Promiscuous network interface reads/records all packets (e.g., including passwords!) passing by.

wireshark software used for end-of-chapter labs is a (free) packet-sniffer
Bad guys can use fake addresses

**IP spoofing:** send packet with false source address

… lots more on security (*Chapter 8 and CS241*)