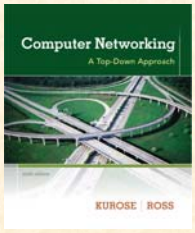
CMPUT 313: Computer Networks

Ehab S. Elmallah
elmallah@ualberta.ca

http://eclass.srv.ualberta.ca/



Outline:

- Assignments, exams, and final grades
- Curiosity questions we hope to answer
- **Concept 1**: circuit switching versus packet switching
- **Concept 2**: the layered architecture of the Internet
- **Concept 3**: delays in packet switching networks

Assignments, exams, and final grades

- Assignments (3) & Labs (3):
  - mix of homework questions and programming assignments in C (CNET simulation)
  - done individually (unless otherwise stated)
- Midterm: **See the course's WEB site**
- About labs
- Lab participation: 2%
  - for attending two presentations – to be announced
- Assignments: 37%, Midterm: 15%, Final: 46%
- Assigning final letter grades

Curiosity Questions We Hope to Answer

- How does the Internet work (especially routing)?
- What is the Internet "layered architecture" (e.g., the "**Internet 5-layer**" stack, or the "**ISO/OSI 7-layer**" stack)?
- What are TCP, IP, and TCP/IP suite of protocols?
- What is the "**Socket API**" ?
  - What networking concepts are needed to understand the API?
- To what extent does the Internet use "**plug-and-play**" design? How?

- What measures are suitable for describing the performance of a **networked application**?
  - How to define such measures?
  - How to analyze mathematically?
  - What to do if the mathematical analysis is difficult?
- What is "**discrete-time simulation**"?
  - What is CNET?
  - How to program, and use CNET to obtain results?

About Course Slides

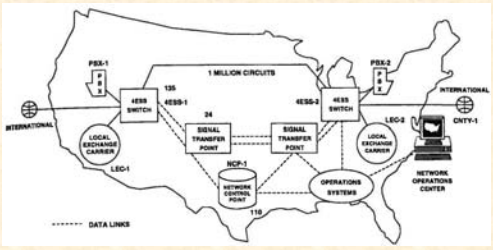
- **Not** intended to be a textbook replacement at all
- Expected to contain errors (please let me know about them)
- Use material (figures, sentences, key equations) from different sources:
  - the course textbook
  - A.S. Tanenbaum, *Computer Networks*, Prentice Hall

Introductory Concept 1:

The circuit switching aspect of the telephone networks, and the packet switching aspect of the Internet.

Circuit-switching vs Packet-switching

- Telephone Networks
  - Hierarchical: edge is simple (dumb?), core is sophisticated. e.g., the AT&T network has:



[Gerald Ash '97]

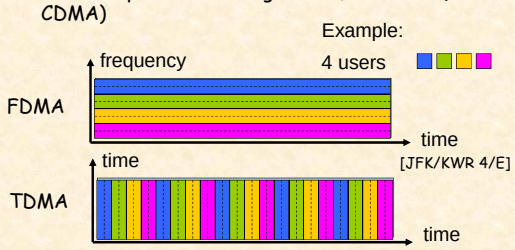
- A national network of
  - 135 large switching centers **(4ESS)**
  - 110 **Network Control Points (NCPs)** providing advanced database for service processing
  - 24 **Signal Transfer Points** that form a **Common Channel Signaling (CCS)** network
    - Note: the CCS network forms a logically independent network that uses packet switching

- A Network Operations Center in NJ for real time traffic management

[technobuffalo.com]

- Private Branch Exchanges (PBXs), and Local Branch Exchanges
- User equipment (connected by about one million trunks)

- How to share a link among several circuits?
  - can be implemented using TDMA, or FDMA (or CDMA)

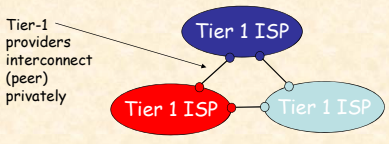


[JFK/KWR 4/E]

- Circuit switching:
  - provides connection oriented service
- Performance metrics: *call blocking probability*

- The Internet
  - As you probably know:
    - 1961-1972: early days of the ARPAnet
    - 1967: ARPAnet conceived by *Advanced Research Projects Agency*
    - 1969: first ARPAnet node (Interface Message Processor – IMP) operational
    - 1972: ARPAnet demonstrated publicly (15 nodes) at ICCC

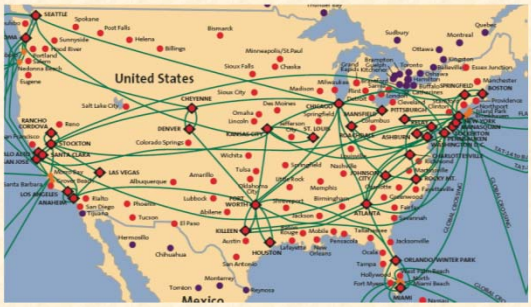
- Roughly hierarchical: edge is smart (and should be protected), core is intended to be rather simple, and may be prone to failure



Tier 1 ISPs: e.g. UUNET, Sprint, AT&T

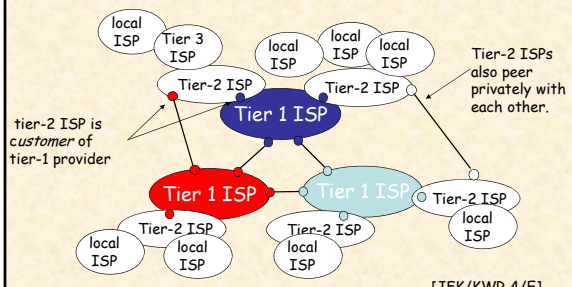
[JFK/KWR 4/E]

- Tier-1 ISP: e.g., Sprint



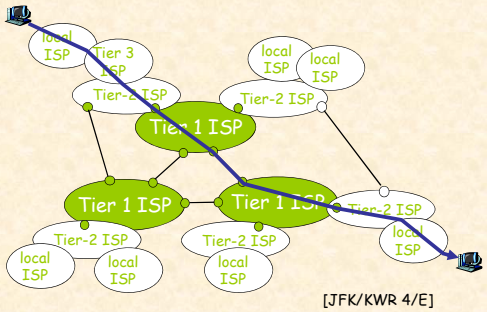
[JFK/KWR 4/E]

- Tier-2 and tier-3 ISPs



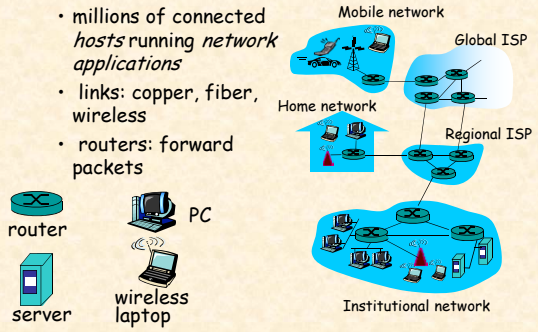
[JFK/KWR 4/E]

- A packet passes through heterogeneous networks



[JFK/KWR 4/E]

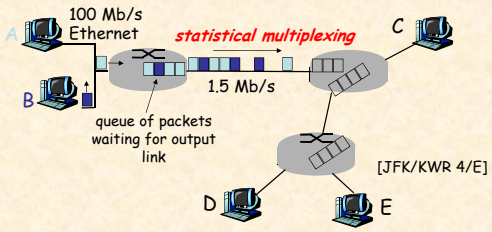
- The Internet "nuts-and-bolts" view
  - millions of connected *hosts* running *network applications*
  - links: copper, fiber, wireless
  - routers: forward packets



router
PC
server
wireless laptop

[JFK/KWR 4/E]

- Packet switching: statistical multiplexing



[JFK/KWR 4/E]

Sequence of A & B packets does not have fixed pattern

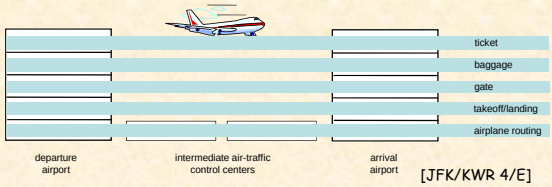
- Basic Services:
  - connection-oriented reliable service (TCP)
  - connectionless unreliable service (UDP)
- Performance: more issues to consider than in telephony.

Introductory Concept 2:

The layered architecture of the Internet

The Internet Layering Architecture

- Layering: key to understand complex systems

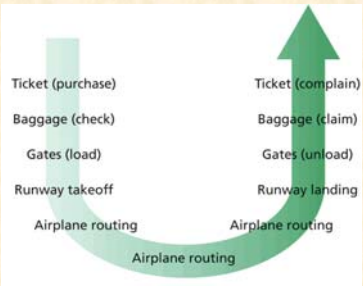


[JFK/KWR 4/E]

Layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

- Steps:



[JFK/KWR 4/E]

- Modular layered architecture in each node

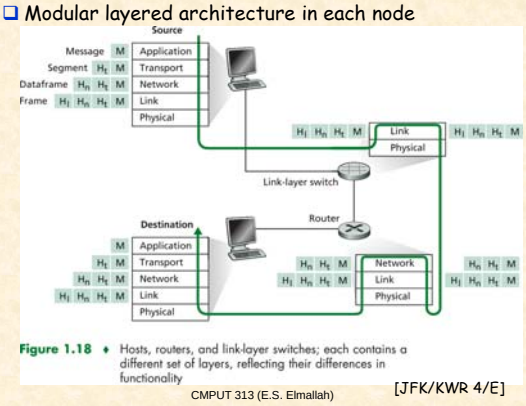


Figure 1.18 ♦ Hosts, routers, and link-layer switches; each contains a different set of layers, reflecting their differences in functionality

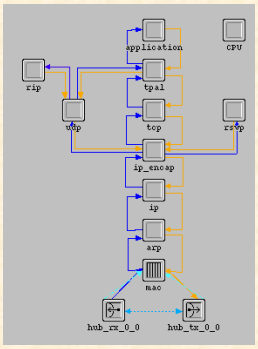
[JFK/KWR 4/E]

- Each layer takes data from above
  - adds header information to create new data unit
  - passes new data unit to layer below

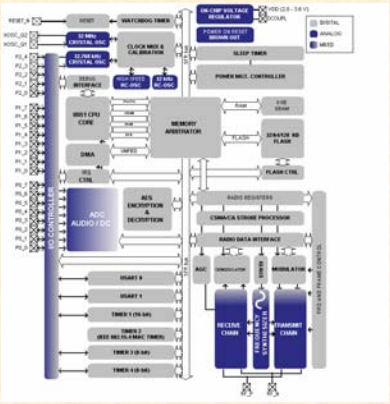
- The Internet Protocol Stack
  a hybrid stack of the ISO/OSI 7-layer stack, and the original TCP/IP stack
  - application: FTP, HTTP
  - transport: process-to-process data transfer (TCP, UDP)
  - network: host-to-host data transfer (IP)
  - link: data transfer between neighbouring network elements
  - physical: "bits on the wire"

	PDUs
application	Message
transport	Segment
network	Datagram
link	Frame
physical	1-PDU

- A typical network simulator:



An example product: TI CC2430

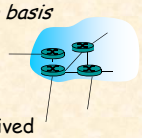


Introductory Concept 3:

Sources of delays in store-and-forward networks

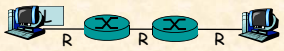
A first look at delays in store-and-forward networks

- End-to-end routing is done on a *per hop basis*



- Each router:
  - waits until a complete packet is received
  - checks the packet for error (using some error detecting/correcting code)
  - examines the destination address
  - moves the packet to a suitable output queue (inside a router)
  - the packet is queued for transmission
  - each packet is transmitted at full speed

- So, in store-and-forward networks, the end-to-end transmission delay depends on how many routers lie on the used route

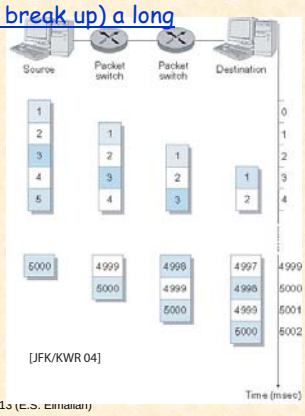


[JFK/KWR 4/E]

- For example,
  - if L= 7.5 Mbits is a message size,
  - R = 1.5 Mbps is each link speed, then
  - it takes L/R= 5 sec to transmit packet on each link
  - the entire packet must arrive at router before it can be transmitted on next link, and
  - total delay= 3 $\times$ 5= 15 sec
- Note: the above remark (dependence on the number of routers) is in contrast with circuit-switching (why?)

To break up (or not to break up) a long message?

- Segmenting a long *message* into shorter *packets* reduces the total end-to-end transmission delay (why?)
  - Suppose, we break up the 7.5 Mbit message into 5000 packets
  - Each packet has 1500 bits $\rightarrow$ 1 msec to transmit packet on one link
  - Delay reduces from 15 sec to 5.002 sec.

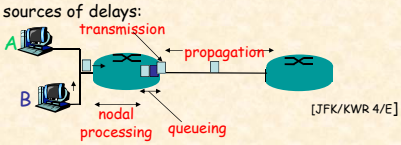


[JFK/KWR 04]

- Other tradeoff factors:
  - the shorter the packet length, the more *header overhead* (e.g., payload length, destination tag, checksum sequence, packet sequence number) we have
  - In case of error, the use of shorter packets is helpful.

Other sources of packet delays

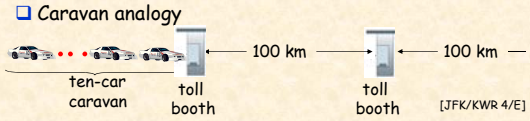
Four basic sources of delays:



- d_{proc}: *nodal processing delay*
  - time to check bit errors, determine output link
  - order: microseconds in high speed routers
- d_{queue}: *queueing delay*
  - time spent by a packet waiting in a queue for transmission
  - order: microseconds to milliseconds

- d_{trans}: *transmission delay*
  - time to send (push out) all packet's bits into the link
  - if L= packet length, and R= link speed, then $d_{trans} = L/R$
  - order: microseconds to milliseconds
- d_{prog}: *propagation delay*
  - time required for one bit to travel a distance d (between two routers)
  - if d = length of physical link, and
  - S= propagation speed in medium (e.g., 3×10^8 m/sec), then
  - $d_{prog} = d/S$
  - order: a few microseconds to hundreds of msecs (e.g., two routers interconnected by a geostationary satellite link)

❑ Caravan analogy



- caravan ⇒ packet , car ⇒ bit , toll booth ⇒ router
- transmission delay: assume booth takes 1 minute to service a car
- propagation delay: assume cars travel at 1000 km/hr (⇒ d_{prog} = 1/10 hr = 6 minutes)
- Note: after 7 minutes the first car lines up at the second booth waiting for 3 cars to finish from the first booth.

❑ traceroute provides delay measurements