Ken Q. Yang, ECE, URI

ELE 437
Computer Network

Presented by
Ken Q. Yang
Dept. Of ECE
URI

Ken Q. Yang

What Do You Learn?

1. Network Concepts and Architectures
2. Communication Protocols
   (e.g. TCP/IP)
3. Doing Useful Work on a Network

How?

- •Lectures,
- •Reading and programming
- •Project (20% of total grade),
- •Presentations (10% grade), and
- •Exams (40% for 2 Midtermexams + 30% for final).

Ken Q. Yang

Main Focus of This Semester

1. TCP/IP Architecture
   - Internet
2. Ethernet LAN
3. Wireless Communication
   - ZigBee or Bluetooth
4. Application Protocols
5. Performance Evaluation

Lecture I: Introduction

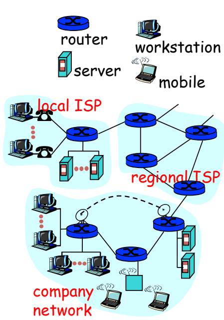
Goals:
- get context, overview, feel of networking
- more depth, detail *later* in course
- approach:
  - descriptive
  - use Internet as example

Overview:
- what is the Internet
- what is a protocol?
- network edge
- network core
- access net, physical media
- performance: loss, delay
- protocol layers, service models

What is the Internet: nuts and bolts view

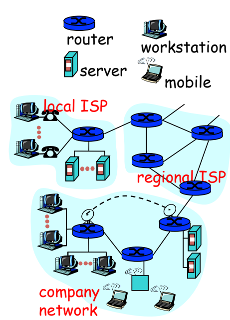
- millions of connected computing devices: *hosts, end-systems*
  - pc workstations, servers
  - PDA, phones, toasters

  running *network apps*
- *communication links*
  - fiber, copper, radio, satellite
- *routers:* forward packets (chunks) of data thru network



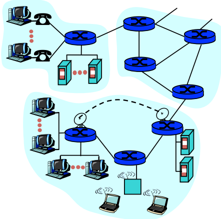
What is the Internet: nuts and bolts view

- *protocols*: control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, FTP, PPP
- *Internet:* network of networks
  - loosely hierarchical
  - public Internet versus private intranet
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



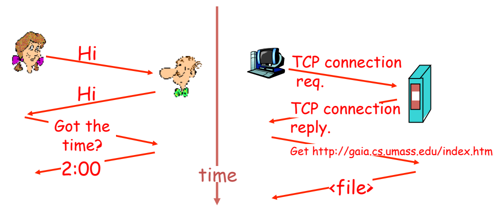
What is the Internet: a service view

- communication *infrastructure* enables distributed applications:
  - WWW, email, games, e-commerce, database., voting,
  - more?
- communication services provided:
  - connectionless
  - connection-oriented



What is a protocol?

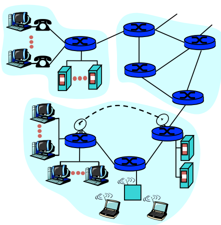
a human protocol and a computer network protocol:



protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt

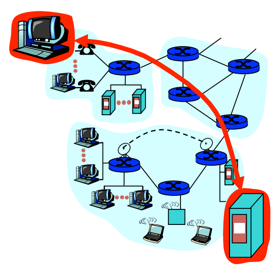
A closer look at network structure:

- network edge: applications and hosts
- network core:
  - routers
  - network of networks
- access networks, physical media: communication links



The network edge:

- end systems (hosts):
  - run application programs
  - e.g., WWW, email
  - at edge of network
- client/server model
  - client host requests, receives service from server
  - e.g., WWW client (browser)/ server; email client/server
- peer-peer model:
  - host interaction symmetric
  - e.g.: teleconferencing



Network edge: connection-oriented service

Goal: data transfer between end sys.

- *handshaking:* setup (prepare for) data transfer ahead of time
  - Hello, hello back human protocol
  - *set up state* in two communicating hosts
- TCP - Transmission Control Protocol
  - Internet's connection-oriented service

TCP service [RFC 793]

- *reliable, in-order* byte-stream data transfer
  - loss: acknowledgements and retransmissions
- *flow control:*
  - sender will not overwhelm receiver
- *congestion control:*
  - senders slow down sending rate when network congested

Network edge: connectionless service

Goal: data transfer between end systems

- same as before!
- UDP - User Datagram Protocol [RFC 768]: Internet connectionless service
  - unreliable data transfer
  - no flow control
  - no congestion control

App using TCP:

- HTTP (WWW), FTP (file transfer), Telnet (remote login), SMTP (email)

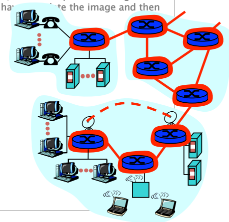
App using UDP:

- streaming media, teleconferencing, Internet telephony

The Network Core

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may ha te the image and then insert it again.

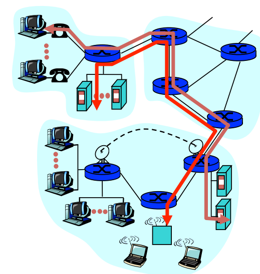
- mesh of interconnected routers
- *the* fundamental question: how is data transferred through net?
  - circuit switching: dedicated circuit per call: telephone net
  - packet-switching: data sent thru net in discrete chunks



Network Core: Circuit Switching

End-end resources reserved for call

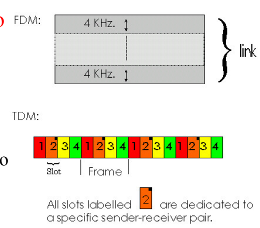
- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required



Network Core: Circuit Switching

network resources (e.g., bandwidth) divided into pieces

- pieces allocated to calls
- resource piece *idle* if not used by owning call *(no sharing)*
- dividing link bandwidth into pieces
  - frequency division
  - time division



Network Core: Packet Switching

each end-end data stream divided into *packets*

- user A, B packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*,

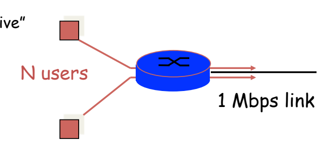
resource contention:

- aggregate resource demand can exceed amount available
- congestion: packets queue, wait for link use
- store and forward: packets move one hop at a time
  - transmit over link
  - wait turn at next link

Packet switching versus circuit switching

Packet switching allows more users to use network!

- 1 Mbit link
- each user:
  - 100 kbps when "active"
  - active 10% of time
- circuit-switching:
  - 10 users
- packet switching:
  - with 35 users, probability > 10 active less than .0004

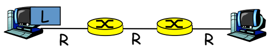


Packet switching versus circuit switching

Is packet switching a "slam dunk winner?"

- Great for bursty data
  - resource sharing
  - simpler, no call setup
- Excessive congestion: 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 6)

Packet-switching: store-and-forward



- Takes L/R seconds to transmit (push out) packet of L bits on to link or R bps
- Entire packet must arrive at router before it can be transmitted on next link: *store and forward*
- delay = 3L/R

Example:
- L = 7.5 Mbits
- R = 1.5 Mbps
- delay = 15 sec