COMP 4621
Computer Communications and Networks

Spring 2015

Computer Networking
A Top-Down Approach

KUROSE | ROSS

Lectures and Tutorials

❖ L-1
  ▪ Instructor: Qian Zhang, http://www.cse.ust.hk/~qianzh/
  ▪ Lectures: Tuesday and Thursday 3:00 pm – 4:20 pm
  ▪ Venue: G009A, CYT Bldg

❖ Web site: http://course.cs.ust.hk/comp4621/
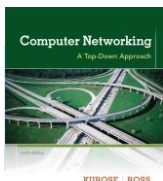
Lectures and Tutorials

❖ Lecture notes and tutorial materials
  ▪ Download course materials before class
  ▪ Homework and programming assignment will be put online

❖ Tutorial and Lab will start in week #5 (March 2, 2015)

  ▪ Lab 1 Monday   9:30 am - 10:20 am   Room 4214
  ▪ Lab 2 Monday   12:00 pm - 12:50 pm   Room 4214

Textbook

Computer Networking: A Top-Down Approach
James Kurose and Keith Ross, Pearson (6th Ed.)

| Application |
| Transport |
| Network |
| Link |
| Physical |

❖ **Bottom Up**: Start with physical (e.g., wires) layer and move up to applications (e.g., mail, web browsers) layer explaining how functions are implemented

❖ **Top Down** : Start with Application layer and move down to Physical layer, explaining what expectations from applications, and how such services are implemented
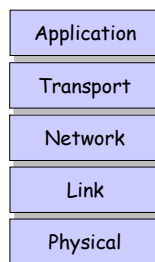
Grading Scheme

❖ Grading is based on

  ▪ Four Written Assignments – 20% (5% each)

  ▪ One Programming Assignment – 15%

  ▪ Midterm Examination - 25%

  ▪ Final Examination - 40%

Conduct in the Classroom

1. Please try to be on time to class
2. Please do not talk while in class except to raise questions
3. No eating/drinking
4. Phone – Silent mode

Plagiarism Policy

- There are differences between collaborations or discussions and copy!

- 1st Time: all involved get ZERO marks, and reported to ARR
- 2nd Time: need to terminate (Fail grade)
- Midterm or Final exam: an automatic FAIL

Course Prerequisite

- COMP 3511 (COMP 252) OS or equivalent
  - Process and thread, DMA and interrupt, inter-process communications

- Programming
  - UNIX environment
  - C/C++ programming

Lecture Format

- Lectures:
  - Lecture notes are available before class
  - It is important to attend the lectures (because not all materials and concepts are covered in slides)
  - Learn from textbook

- Tutorials
  - Supplement the lectures with more examples
  - Socket programming and project

- Reading the corresponding materials in the textbook
  - Slides do not cover everything

- Chapter or sub-chapter summary
  - These will be put online at the end of each chapter

Assignments

- Written assignments
  - Due by time specified
  - Re-grade requests will only be entertained within **one** week after the homework are handed back
  - Late policy: 15% reduction, only one day delay is allowed.

- Programming assignment
  - Individual projects
  - Due by time specified
  - Run on Unix and submit it using CASS
  - Re-grade policy will be announced
  - Late policy : 15% reduction, only one day delay is allowed.

Examination Schedules

- Midterm Exam
  - March 17, 2015 7:00-9:00pm
  - Room: 3007
- Final Exam
  - To be determined (TBD)
- All exams are closed-book

- No make-up exams will be given except
  - under very special circumstances, e.g., sickness, with letters of proof.
  - Instructors MUST BE informed beforehand

Tips for Learning

- Attend lectures
  - Download lecture notes prior to lectures
  - Important concepts are explained
- Complete homework alone
  - This is an exercise to test your knowledge
- Spend 30 minutes each week to review the content
  - Chapter summary can help to review
  - You can not expect to learn everything two days before exams no matter how smart you are ☺
  - Knowledge is accumulated incrementally
- Start your project earlier
  - Have a plan for the project
- Raise questions!
  - Do not delay your questions until prior to exams

Course Outline

❖ Introduction (1.5 weeks)
  ▪ Internet, network edge and core, performance, history
❖ Application Layer (3 weeks)
  ▪ Web, HTTP, E-mail and SMTP
  ▪ DNS, Peer-to-Peer applications, socket programming
❖ Transport Layer (3 weeks)
  ▪ UDP and TCP
  ▪ Principles of congestion control, TCP congestion control
❖ Network Layer (3 weeks)
  ▪ Virtual circuit, datagram, router, IP
  ▪ Routing protocols
❖ Link Layer and Local Area Network (2 weeks)
  ▪ MAC, addressing
  ▪ Ethernet and link-layer switch

Chapter 1: introduction

our goal:

❖ get "feel" and terminology
❖ more depth, detail *later* in course
❖ approach:
  ▪ use Internet as example

overview:

❖ what's the Internet?
❖ what's a protocol?
❖ network edge; hosts, access net, physical media
❖ network core: packet/circuit switching, Internet structure
❖ performance: loss, delay, throughput
❖ security
❖ protocol layers, service models
❖ history

Chapter 1: roadmap

1.1 what *is* the Internet?
1.2 network edge
  ▪ end systems, access networks, links
1.3 network core
  ▪ packet switching, circuit switching, network structure
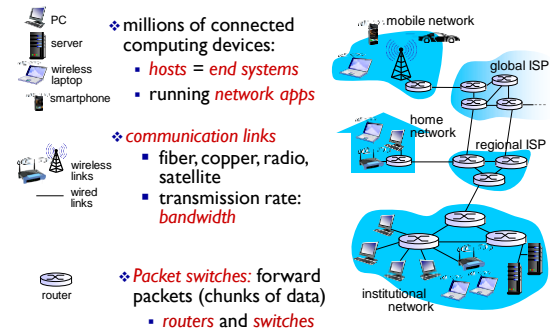1.4 delay, loss, throughput in networks
1.5 protocol layers, service models
1.6 networks under attack: security
1.7 history

What's the Internet: "nuts and bolts" view

PC
server
wireless laptop
smartphone

❖ millions of connected computing devices:
  ▪ *hosts = end systems*
  ▪ running *network apps*

❖ *communication links*
  ▪ fiber, copper, radio, satellite
  ▪ transmission rate: *bandwidth*

wireless links
wired links

router

❖ *Packet switches:* forward packets (chunks of data)
  ▪ *routers* and *switches*

mobile network
global ISP
home network
regional ISP
institutional network

"Fun" internet appliances

IP picture frame
http://www.ceiva.com/

Web-enabled toaster + weather forecaster

Tweet-a-watt: monitor energy use

Internet refrigerator

Slingbox: watch, control cable TV remotely

Internet phones

What's the Internet: "nuts and bolts" view

❖ *Internet:* "network of networks"
  ▪ Interconnected ISPs
❖ *protocols* control sending, receiving of msgs
  ▪ e.g., TCP, IP, HTTP, Skype, 802.11
❖ *Internet standards*
  ▪ RFC: Request for comments
  ▪ IETF: Internet Engineering Task Force

mobile network
global ISP
home network
regional ISP
institutional network
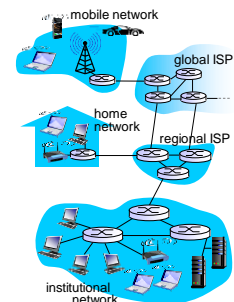
What's the Internet: a service view

❖ *Infrastructure that provides services to applications:*
  ▪ Web, VoIP, email, games, e-commerce, social nets, ...
❖ *provides programming interface to apps*
  ▪ hooks that allow sending and receiving app programs to "connect" to Internet
  ▪ provides service options, analogous to postal service
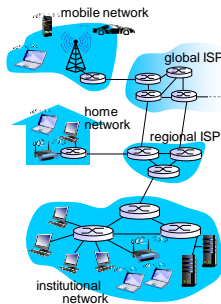
What's a protocol?

human protocols:
❖ "what's the time?"
❖ "I have a question"
❖ introductions

... specific msgs sent
... specific actions taken when msgs received, or other events

network protocols:
❖ machines rather than humans
❖ all communication activity in Internet governed by protocols

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

What's a protocol?

a human protocol and a computer network protocol:
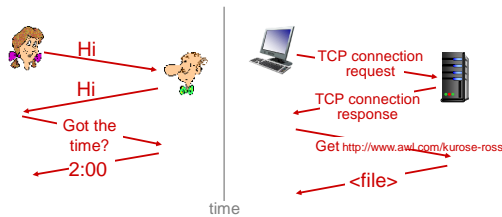
Hi
Hi
Got the time?
2:00

time

TCP connection request
TCP connection response
Get http://www.awl.com/kurose-ross
<file>

Q: other human protocols?

Chapter 1: roadmap

1.1 what *is* the Internet?
1.2 network edge
  ▪ end systems, access networks, links
1.3 network core
  ▪ packet switching, circuit switching, network structure
1.4 delay, loss, throughput in networks
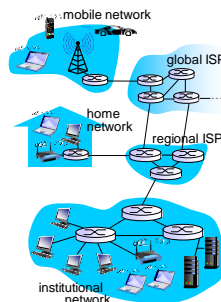1.5 protocol layers, service models
1.6 networks under attack: security
1.7 history

A closer look at network structure:

❖ *network edge:*
  ▪ hosts: clients and servers
  ▪ servers often in data centers
❖ *access networks, physical media:* wired, wireless communication links
❖ *network core:*
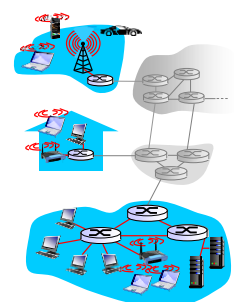 ▪ interconnected routers
  ▪ network of networks

Access networks and physical media

Q: How to connect end systems to edge router?
❖ residential access nets
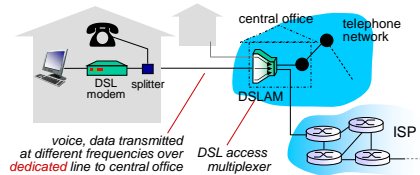❖ institutional access networks (school, company)
❖ mobile access networks

keep in mind:
❖ bandwidth (bits per second) of access network?
❖ shared or dedicated?

4

Access net: digital subscriber line (DSL)

voice, data transmitted at different frequencies over dedicated line to central office

- ❖ use *existing* telephone line to central office DSLAM
  - ▪ data over DSL phone line goes to Internet
  - ▪ voice over DSL phone line goes to telephone net
- ❖ < 2.5 Mbps upstream transmission rate (typically < 1 Mbps)
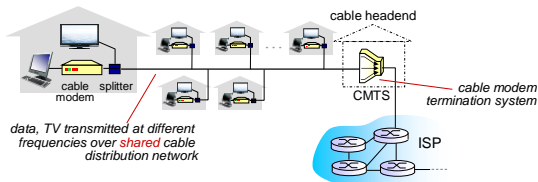- ❖ < 24 Mbps downstream transmission rate (typically < 10 Mbps)

Access net: cable network
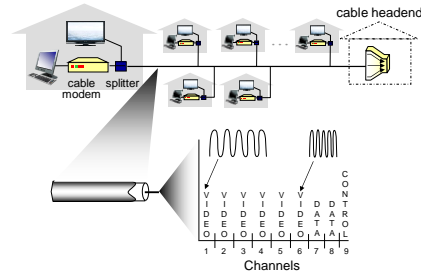
frequency division multiplexing: different channels transmitted in different frequency bands

Access net: cable network

data, TV transmitted at different frequencies over shared cable distribution network

- ❖ HFC: hybrid fiber coax
  - ▪ asymmetric: up to 30Mbps downstream transmission rate, 2 Mbps upstream transmission rate
- ❖ network of cable, fiber attaches homes to ISP router
  - ▪ homes *share access network* to cable headend
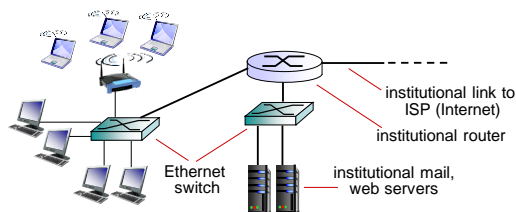 - ▪ unlike DSL, which has dedicated access to central office

Access net: home network
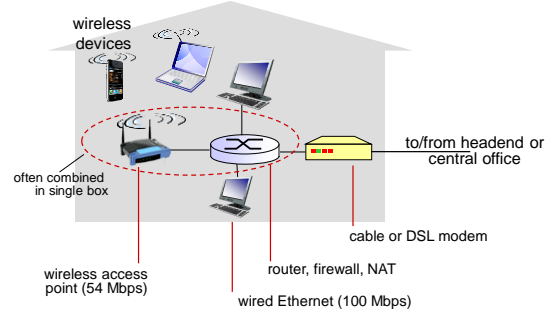
Enterprise access networks (Ethernet)

- ❖ typically used in companies, universities, etc
- ❖ 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
- ❖ today, end systems typically connect into Ethernet switch

Wireless access networks

- ❖ shared *wireless* access network connects end system to router
  - ▪ via base station aka "access point"

wireless LANs:
- ▪ within building (100 ft)
- ▪ 802.11b/g (WiFi): 11, 54 Mbps transmission rate

wide-area wireless access
- ▪ provided by telco (cellular) operator, 10's km
- ▪ between 1 and 10 Mbps
- ▪ 3G, 4G: LTE

to Internet

to Internet

Host: sends *packets* of data

host sending function:
- takes application message
- breaks into smaller chunks, known as *packets*, of length *L* bits
- transmits packet into access network at *transmission rate R*
  - link transmission rate, aka link *capacity, aka link bandwidth*

two packets, *L* bits each

2 1

host

R: link transmission rate
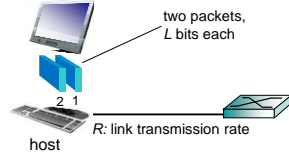
$$\text{packet transmission delay} = \text{time needed to transmit } L\text{-bit packet into link} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

Physical media

- bit: propagates between transmitter/receiver pairs
- physical link: what lies between transmitter & receiver
- guided media:
  - signals propagate in solid media: copper, fiber, coax
- unguided media:
  - signals propagate freely, e.g., radio

twisted pair (TP)
- two insulated copper wires
  - Category 5: 100 Mbps, 1 Gpbs Ethernet
  - Category 6: 10Gbps

Physical media: coax, fiber

coaxial cable:
- two concentric copper conductors
- bidirectional
- broadband:
  - multiple channels on cable
  - HFC

fiber optic cable:
- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (e.g., 10's-100's Gpbs transmission rate)
- low error rate:
  - repeaters spaced far apart
  - immune to electromagnetic noise

Physical media: radio

- signal carried in electromagnetic spectrum
- no physical "wire"
- bidirectional
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

radio link types:
- terrestrial microwave
  - e.g. up to 45 Mbps channels
- LAN (e.g., WiFi)
  - 11Mbps, 54 Mbps
- wide-area (e.g., cellular)
  - 3G cellular: ~ few Mbps
- satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
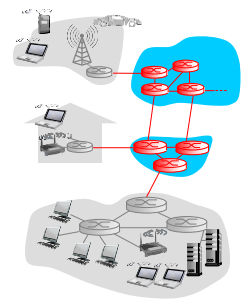 - 270 msec end-end delay
  - geosynchronous versus low altitude

Chapter 1: roadmap

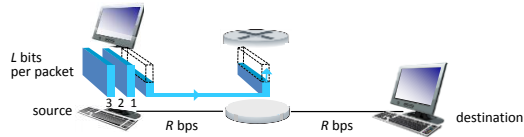## The network core

- 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

Packet-switching: store-and-forward

L bits per packet

source

3 2 1

R bps   R bps

destination

- 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
- end-end delay = $2L/R$ (assuming zero propagation delay)

one-hop numerical example:
- L = 7.5 Mbits
- R = 1.5 Mbps
- one-hop transmission delay = 5 sec

more on delay shortly ...

Packet Switching: queueing delay, loss

A R = 100 Mb/s   C

B

queue of packets waiting for output link

R = 1.5 Mb/s

D

E
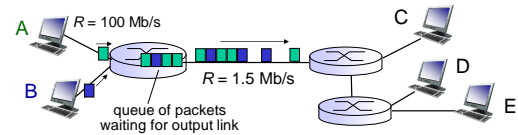
queuing and loss:
- If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
  - packets will queue, wait to be transmitted on link
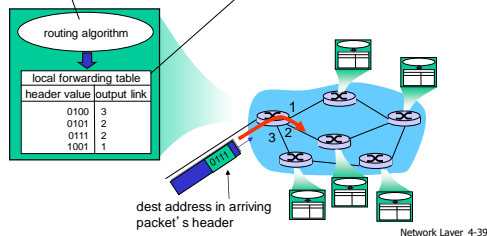 - packets can be dropped (lost) if memory (buffer) fills up

Two key network-core functions

routing: determines source-destination route taken by packets
- *routing algorithms*

forwarding: move packets from router's input to appropriate router output


routing algorithm

local forwarding table

header value	output link
0100	3
0101	2
0111	2
1001	1

1

2

3

dest address in arriving packet's header

# 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-like (guaranteed) performance
- circuit segment idle if not used by call *(no sharing)*
- Commonly used in traditional telephone networks
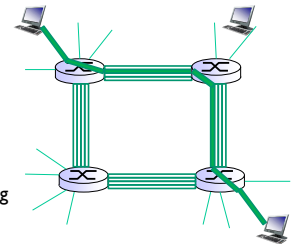
Circuit switching: FDM versus TDM

FDM

Example:
4 users

frequency

time

TDM

frequency
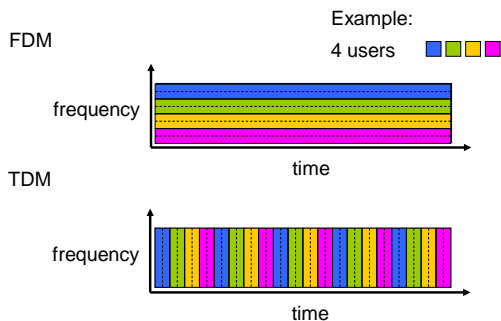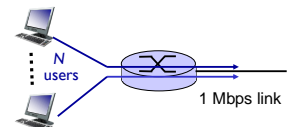
time

Packet switching versus circuit 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

N users

1 Mbps link

- *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?

Q: what happens if > 35 users ?

* Check out the online interactive exercises for more examples

Packet switching versus circuit switching

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)?

Internet structure: network of networks

- ❖ End systems connect to Internet via access ISPs (Internet Service Providers)
  - ▪ Residential, company and university ISPs
- ❖ Access ISPs in turn must be interconnected.
  - ❖ So that any two hosts can send packets to each other
- ❖ Resulting network of networks is very complex
  - ❖ Evolution was driven by economics and national policies
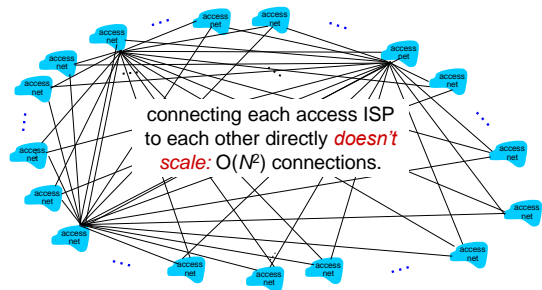- ❖ Let's take a stepwise approach to describe current Internet structure

Internet structure: network of networks

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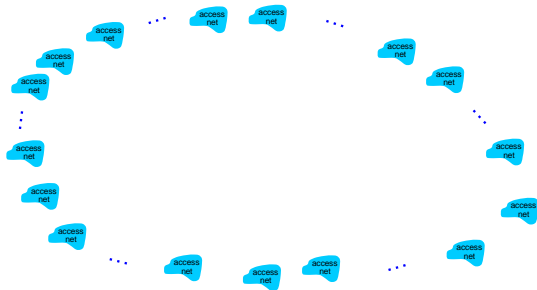
Internet structure: network of networks

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

connecting each access ISP to each other directly *doesn't scale:* O(*N²*) connections.

Internet structure: network of networks

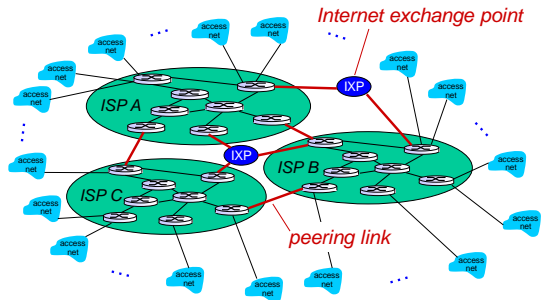*Option: connect each access ISP to a global transit ISP? Customer and provider ISPs have economic agreement.*
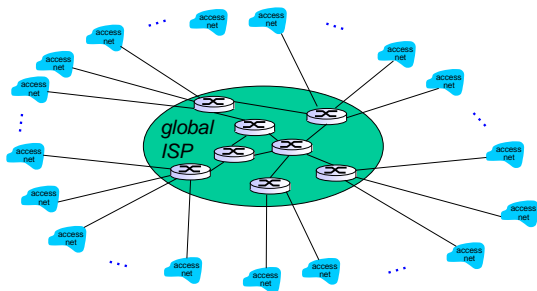
Internet structure: network of networks

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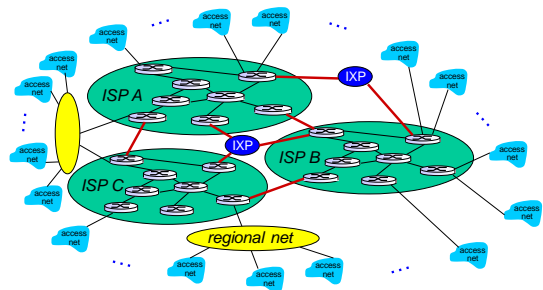

8
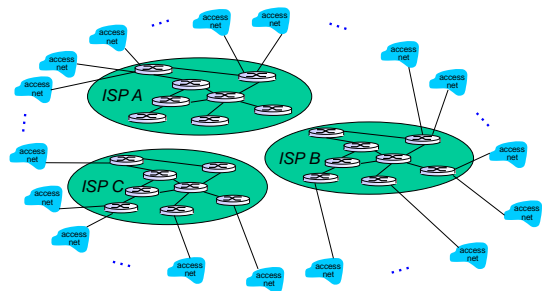
Internet structure: network of networks

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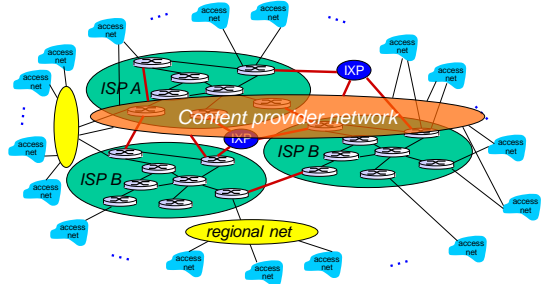…. which must be interconnected

Internet structure: network of networks

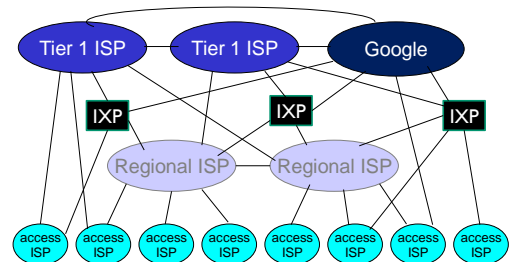… and regional networks may arise to connect access nets to ISPS

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
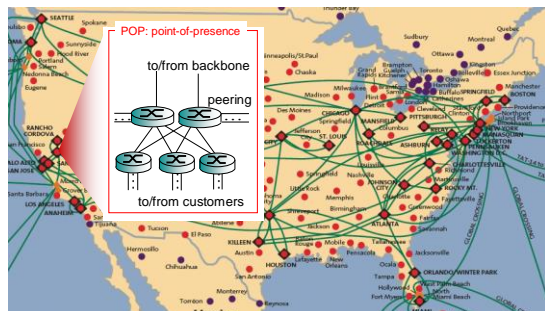

Internet structure: network of networks

- ❖ at center: small # of well-connected large networks
  - ▪ "tier-1" commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
  - ▪ content provider network (e.g, Google): private network that connects it data centers to Internet, often bypassing tier-1, regional ISPs Introduction 1-52

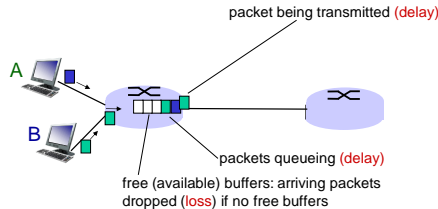## Tier-1 ISP: e.g., Sprint

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- ▪ end systems, access networks, links

1.3 network core

- ▪ packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

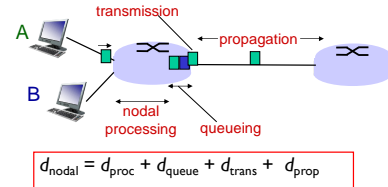1.6 networks under attack: security

1.7 history

How do loss and delay occur?

packets *queue* in router buffers

❖ packet arrival rate to link (temporarily) exceeds output link capacity
❖ packets queue, wait for turn

packet being transmitted (delay)

A

B

packets queueing (delay)

free (available) buffers: arriving packets
dropped (loss) if no free buffers

Four sources of packet delay

transmission

A

propagation

B

nodal
processing   queueing

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

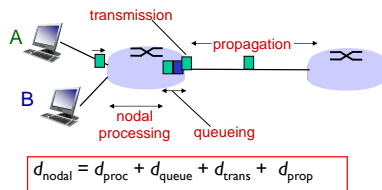d_{proc}: nodal processing
- check bit errors
- determine output link
- typically < msec

d_{queue}: queueing delay
- time waiting at output link for transmission
- depends on congestion level of router

Four sources of packet delay

transmission

A

propagation

B

nodal
processing   queueing

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

d_{trans}: transmission delay:
- L: packet length (bits)
- R: link *bandwidth (bps)*
- $d_{trans} = L/R$

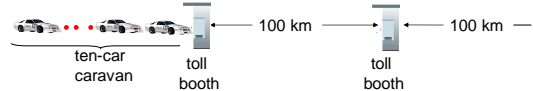d_{prop}: propagation delay:
- d: length of physical link
- s: propagation speed in medium ($\sim 2 \times 10^8$ m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop}
very different

# Caravan analogy

ten-car
caravan

toll
booth

100 km

toll
booth

100 km

❖ cars "propagate" at 100 km/hr
❖ toll booth takes 12 sec to service car (bit transmission time)
❖ car~bit; caravan ~ packet
❖ *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 both: 100km/(100km/hr)= 1 hr
- *A:* 62 minutes

Caravan analogy (more)

ten-car
caravan

toll
booth

100 km

toll
booth

100 km
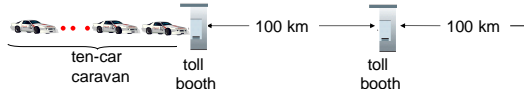
❖ suppose cars now "propagate" at 1000 km/hr
❖ and suppose toll booth now takes one min to service a car
❖ *Q:* Will cars arrive to 2nd booth before all cars serviced at first booth?

- *A: Yes!* after 7 min, 1st car arrives at second booth; three cars still at 1st booth.

Queueing delay (revisited)

❖ *R:* link bandwidth (bps)
❖ *L:* packet length (bits)
❖ a: average packet arrival rate

average queueing delay

traffic intensity
= La/R

La/R

1
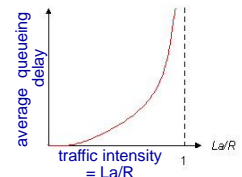
❖ $La/R \sim 0$: avg. queueing delay small
❖ $La/R \to 1$: avg. queueing delay large
❖ $La/R > 1$: more "work" arriving than can be serviced, average delay infinite!

La/R ~ 0

La/R -> 1

"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

3 probes

3 probes
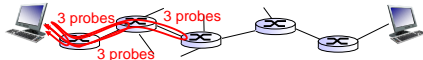
"Real" Internet delays, routes

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

3 delay measurements from
gaia.cs.umass.edu to cs-gw.cs.umass.edu

```
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  abilene-vbns.abilene.ucaid.edu (198.32.11.9)  22 ms  18 ms  22 ms
7  nycm-wash.abilene.ucaid.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.fr.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.cssi.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
```

trans-oceanic link

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

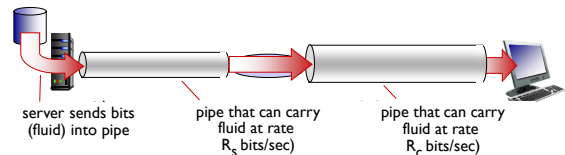* Do some traceroutes from exotic countries at www.traceroute.org

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

A

B

buffer (waiting area)

packet being transmitted

packet arriving to full buffer is *lost*
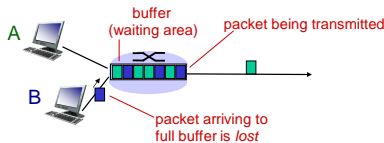
Throughput

- ❖ *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
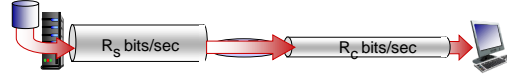

server sends bits (fluid) into pipe

pipe that can carry fluid at rate R_s bits/sec

pipe that can carry fluid at rate R_c bits/sec

Throughput (more)

- ❖ $R_s < R_c$ What is average end-end throughput?

R_s bits/sec R_c bits/sec

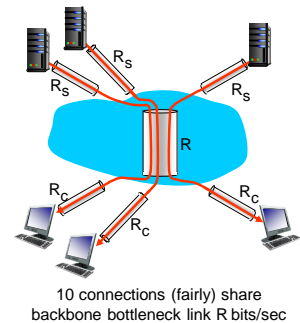- ❖ $R_s > R_c$ What is average end-end throughput?

R_s bits/sec R_c bits/sec
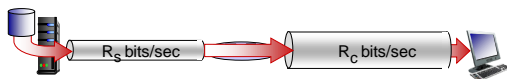
bottleneck link

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

R_s

R_s

R_s

R_s

R

R_c

R_c

R_c

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

Chapter 1: roadmap

Protocol "layers"

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?
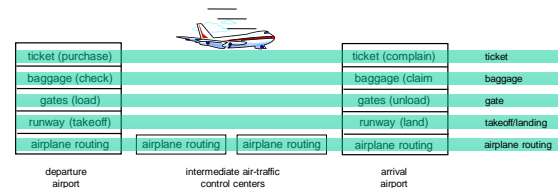
…. or at least our discussion of networks?

Organization of air travel

ticket (purchase)	ticket (complain)
baggage (check)	baggage (claim)
gates (load)	gates (unload)
runway takeoff	runway landing
airplane routing	airplane routing
airplane routing	

❖ a series of steps

Layering of airline functionality

ticket (purchase)			ticket (complain)	ticket
baggage (check)			baggage (claim)	baggage
gates (load)			gates (unload)	gate
runway (takeoff)			runway (land)	takeoff/landing
airplane routing	airplane routing	airplane routing	airplane routing	airplane routing

departure airport — intermediate air-traffic control centers — arrival airport

layers: each layer implements a service
- via its own internal-layer actions
- relying on services provided by layer below

Why layering?

dealing with complex systems:
- ❖ explicit structure allows identification, relationship of complex system's pieces
  - layered *reference model* for discussion
- ❖ modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- ❖ layering considered harmful?

## Internet protocol stack

- ❖ *application:* supporting network applications
  - FTP, SMTP, HTTP
- ❖ *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"

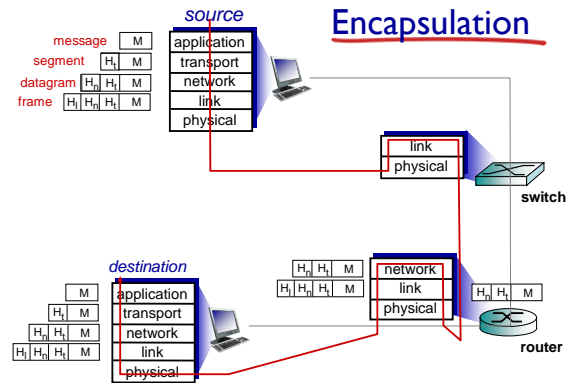| |
|---|
| application |
| transport |
| network |
| link |
| physical |
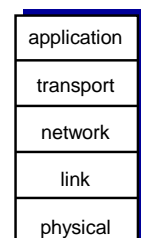
12

ISO/OSI reference model

- ❖ *presentation:* allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- ❖ *session:* synchronization, checkpointing, recovery of data exchange
- ❖ Internet stack "missing" these layers!
    - these services, *if needed,* must be implemented in application
    - needed?
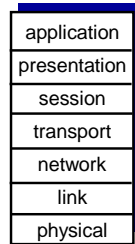
application
presentation
session
transport
network
link
physical

Encapsulation

Chapter 1: roadmap

Network security

- ❖ field of network security:
    - 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. DDoS attacks

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)
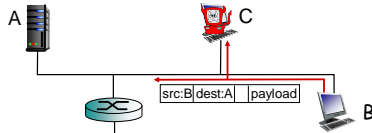3. send packets to target from compromised hosts
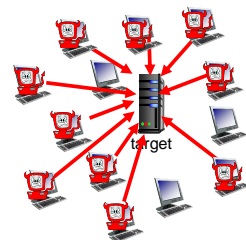
Bad guys can sniff packets

packet "sniffing":
- broadcast media (shared ethernet, wireless)
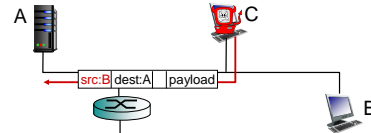- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by

A

C

src:B | dest:A | payload

B

❖ 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

A

C

src:B | dest:A | payload

B

… lots more on security (throughout, Chapter 8)

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge
- end systems, access networks, links

1.3 network core
- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

1.7 history

Internet history

1961-1972: Early packet-switching principles

❖ 1961: Kleinrock - queueing theory shows effectiveness of packet-switching

❖ 1964: Baran - packet-switching in military nets

❖ 1967: ARPAnet conceived by Advanced Research Projects Agency

❖ 1969: first ARPAnet node operational

❖ 1972:
- ARPAnet public demo
- NCP (Network Control Protocol) first host-host protocol
- first e-mail program
- ARPAnet has 15 nodes

THE ARPA NETWORK

Internet history

1972-1980: Internetworking, new and proprietary nets

❖ 1970: ALOHAnet satellite network in Hawaii

❖ 1974: Cerf and Kahn - architecture for interconnecting networks

❖ 1976: Ethernet at Xerox PARC

❖ late70's: proprietary architectures: DECnet, SNA, XNA

❖ late 70's: switching fixed length packets (ATM precursor)

❖ 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:
- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today's Internet architecture

Internet history

1980-1990: new protocols, a proliferation of networks

❖ 1983: deployment of TCP/IP

❖ 1982: smtp e-mail protocol defined

❖ 1983: DNS defined for name-to-IP-address translation

❖ 1985: ftp protocol defined

❖ 1988: TCP congestion control

❖ new national networks: Csnet, BITnet, NSFnet, Minitel

❖ 100,000 hosts connected to confederation of networks

Internet history

1990, 2000's: commercialization, the Web, new apps

❖ early 1990's: ARPAnet decommissioned
❖ 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
❖ early 1990s: Web
  ▪ hypertext [Bush 1945, Nelson 1960's]
  ▪ HTML, HTTP: Berners-Lee
  ▪ 1994: Mosaic, later Netscape
  ▪ late 1990's: commercialization of the Web

late 1990's – 2000's:
❖ more killer apps: instant messaging, P2P file sharing
❖ network security to forefront
❖ est. 50 million host, 100 million+ users
❖ backbone links running at Gbps

Internet history

2005-present

❖ ~750 million hosts
  ▪ Smartphones and tablets
❖ Aggressive deployment of broadband access
❖ Increasing ubiquity of high-speed wireless access
❖ Emergence of online social networks:
  ▪ Facebook: soon one billion users
❖ Service providers (Google, Microsoft) create their own networks
  ▪ Bypass Internet, providing "instantaneous" access to search, emai, etc.
❖ E-commerce, universities, enterprises running their services in "cloud" (eg, Amazon EC2)

Introduction: summary

covered a "ton" of material!
❖ Internet overview
❖ what's a protocol?
❖ network edge, core, access network
  ▪ packet-switching versus circuit-switching
  ▪ Internet structure
❖ performance: loss, delay, throughput
❖ layering, service models
❖ security
❖ history

you now have:
❖ context, overview, "feel" of networking
❖ more depth, detail *to follow!*