"On the Internet, nobody knows you're a dog."

"Nurse, get on the internet, go to SURGERY.COM, scroll down and click on the 'Are you totally lost?' icon."

CS2505
Network Computing

Prof. Cormac J. Sreenan

Lecturer Details

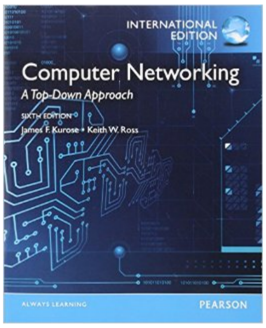
- Email
  - For Prof. Sreenan: cjs@cs.ucc.ie
  - Always put CS2505 in "Subject" line of message
  - Always send from ucc.ie to avoid being labelled as spam
- Meetings
  - Room 1-75; just email for an appointment

Course Information

- CS2505 is a 5-credit module
  - 24 lectures plus practical laboratory sessions
  - Two lectures per week (Period 2 only)
- Assessment
  - Summer Exam. 80%
  - Lab. assignments 20%
- Course lectures on Moodle
  - cs4.ucc.ie
  - Lecture notes added as the course progresses; also lab. details

Textbooks

- Required to purchase:
  - J. Kurose & K. Ross, "Computer Networking", Addison-Wesley Pub.
  - 6th is the latest International edition
- Other good books (in library):
  - L. Peterson and B. Davie. "Computer Networks: A Systems Approach". Morgan Kaufmann Pub.
  - A. Tanenbaum, "Computer Networks", Prentice Hall Pub.



Plagiarism

1. Plagiarism is presenting someone else's work as your own. It is a violation of UCC Policy and there are strict and severe penalties.
2. You must read and comply with the UCC Policy on Plagiarism www.ucc.ie/en/exams/procedures-regulations/
3. The Policy applies to *all* work submitted, including software.
4. You can expect that your work will be checked for evidence of plagiarism or collusion.
5. In some circumstances it may be acceptable to reuse a small amount of work by others, but *only* if you provide explicit acknowledgement and justification.
6. If in doubt ask your module lecturer *prior* to submission. Better safe than sorry!

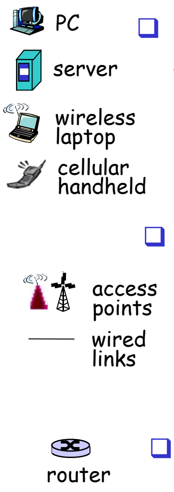
Course Overview

- Section 1: Networking Basics
- Section 2: Application layer
- Section 3: Transport layer
- Section 4: Network Management

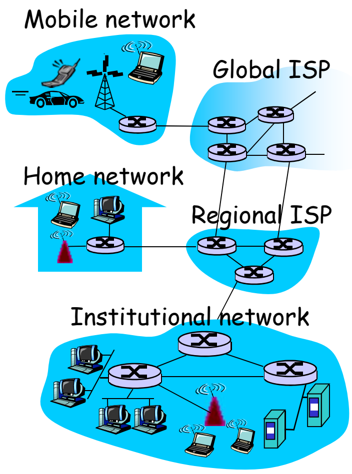
Section 1: roadmap

1.1 What *is* the Internet?
1.2 Network edge
1.3 Network core
1.4 Network performance metrics
1.5 Network protocol architecture
1.6 Networks under attack: security
1.7 History of computer networks

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



- millions of connected computing devices: *hosts = end systems*
  - running *network apps*
- *communication links*
  - fibre, copper, radio, satellite
  - transmission rate = *bandwidth*
- *routers:* forward packets (chunks of data)



"Cool" internet appliances


IP picture frame


Web-enabled toaster + weather forecaster

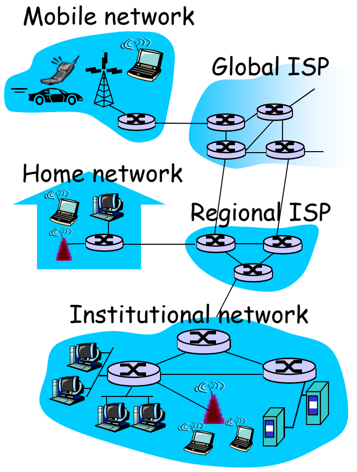

Internet of Things


Drones


Smart watches

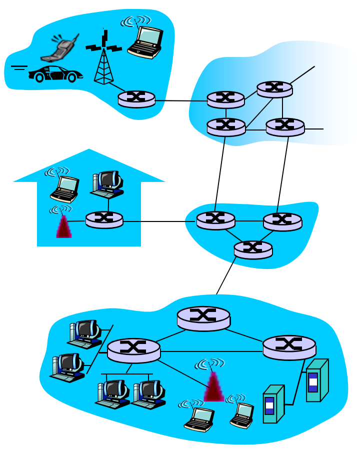
What's the Internet: "nuts and bolts" view

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



What's the Internet: a service view

- communication *infrastructure* enables distributed applications:
  - Web, VoIP, email, games, e-commerce, file sharing
- communication services provided to apps:
  - reliable data delivery from source to destination
  - "best effort" (unreliable) data delivery



What is a Protocol?

- The word originates in middle English where it referred to the fine details of an agreement

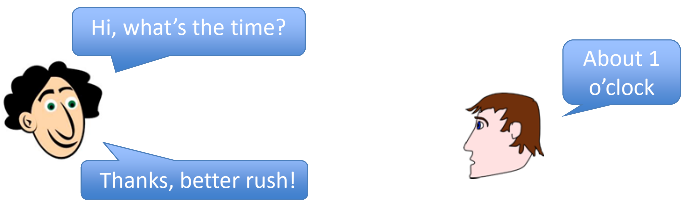
- For communication, it defines the "set of rules governing the exchange or transmission of data between devices." [Oxford English Dictionary]

A Human Protocol

Asking the time

A Human Protocol

Asking the time (a different way)



15

A Human Protocol

Asking the time (in French)

16

Computer Protocols

- People are really good at adapting to the nuances of speech and social interaction
  - And even to the use of different languages
- But computers work best when there is no ambiguity in the communication
  - So it must be specified as a set of rules

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

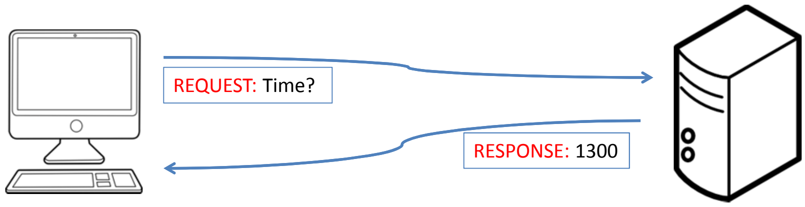
network protocols:

- 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

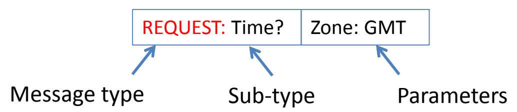
A Request-Response Protocol

- Most network protocols are called request-response
  - The client sends a message to a server requesting some information
  - The server responds by sending back a message to the client



Specifying Protocols

- Protocols are specified as follows:
  - The set of message types (eg request, response, error)
  - The format of each message



  - The action to be taken when a message is received, including what response to send

Implementing Protocols

- Network protocols are usually implemented in software
- The software must faithfully implement the protocol specification
  - The choice of programming language and operating system does not matter
- This software must be installed at the client and the server computers
  - As long as the specification was adhered to, independent implementations *should* work together

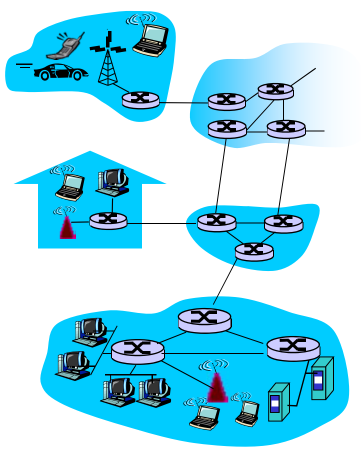
Types of Protocols

- Each protocol is designed for a specific purpose
- The most crucial protocols allow messages to be routed to the right destination and reliably delivered
  - Their operation is largely invisible to end-users
- Other protocols are more familiar, such as
  - HTTP which allows web browsers to send request to web servers
  - BitTorrent for peer-to-peer file sharing

Section 1: roadmap

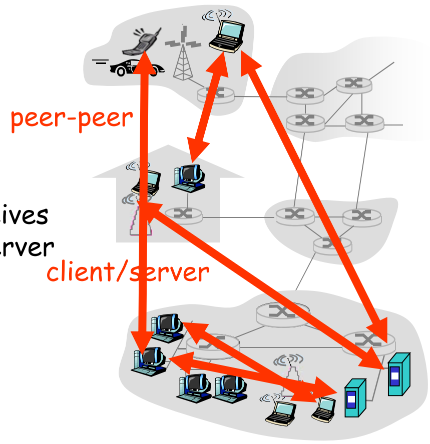
A closer look at network structure:

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



The network edge:

- end systems (hosts):
  - run application programs
  - e.g. Web, email
  - at "edge of network"
- client/server model
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server
- peer-peer model:
  - minimal (or no) use of dedicated servers
  - e.g. Skype, BitTorrent



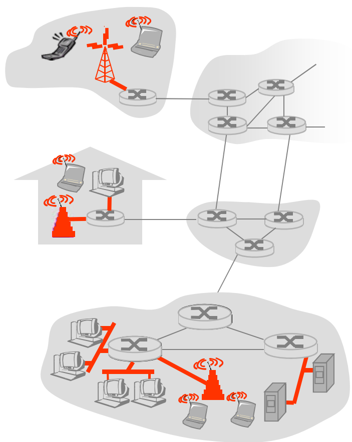
Access networks and physical media

Q: How to connect end systems to edge router?

- residential access nets
- institutional access networks (university, company)
- mobile access networks

Keep in mind:

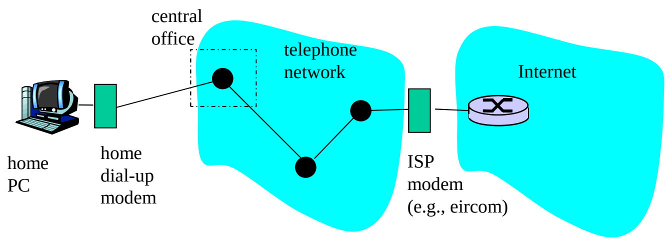
- bandwidth (bits per second) of access network?
- shared or dedicated?



Concept of Bandwidth

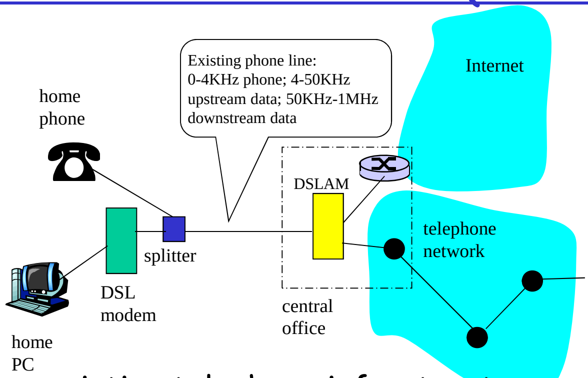
- Amount of data that can be transmitted per time unit
  - Example: 10 Mega bits per second (Mb/s or Mbps) or 100 Kilo bits per second (Kb/s)
  - Also called data rate or capacity
- Notation
  - distinguish between bits (b) and bytes (B)
  - One byte = 8 bits; bytes sometimes called octets
  - Kb/s = 10^3 bits per second; Mb/s = 10^6 bits per second

Dial-up Modem



- Old technology that uses existing telephony infrastructure
  - Home is connected to central office
- up to 56Kb/s direct access to router (often less)
- Can't surf and phone at same time: not "always on"

Digital Subscriber Line (DSL)



- Also uses existing telephone infrastructure
  - dedicated physical line to telephone central office
- Performance
  - up to 3.3 Mb/s upstream (today typically < 1 Mb/s)
  - up to 24 Mb/s downstream (today typically < 8 Mb/s)

Residential access: cable modems

- Does not use telephone infrastructure
  - Instead uses cable TV infrastructure
- HFC: hybrid fibre coax
  - asymmetric: typical per-home values of up to 30Mb/s downstream, 2 Mb/s upstream, but can be much higher
- network of cable and fibre attaches homes to ISP router (called a head-end)
  - homes share access to router
  - unlike DSL, which has dedicated access

Residential access: cable modems

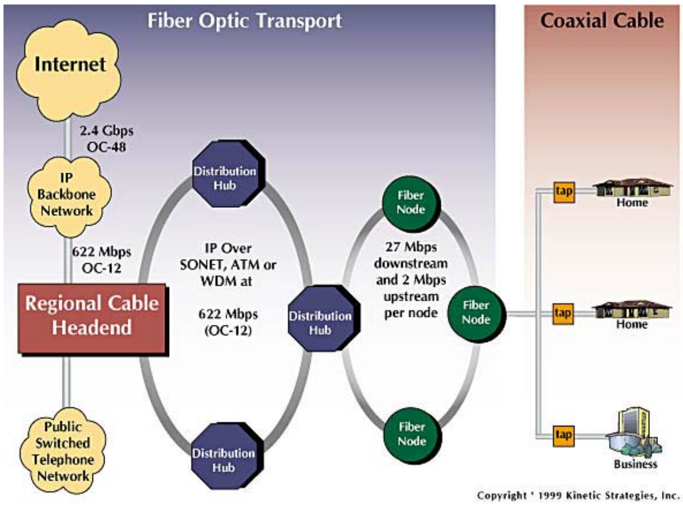
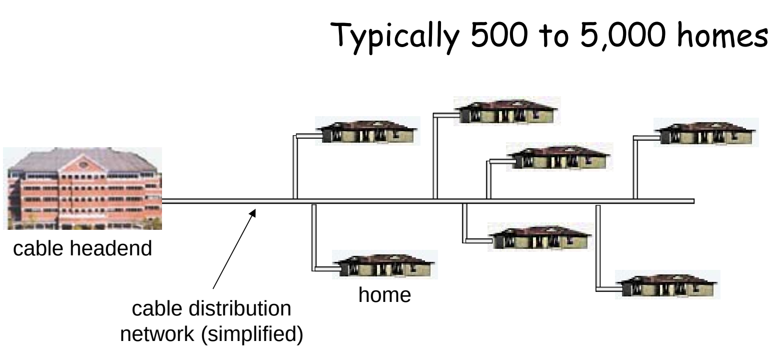
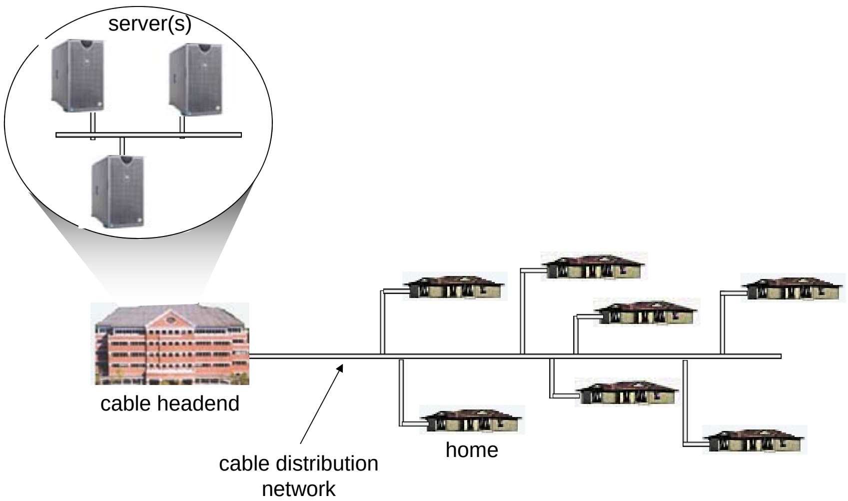


Diagram: http://www.cabledatacomnews.com/cmic/diagram.html

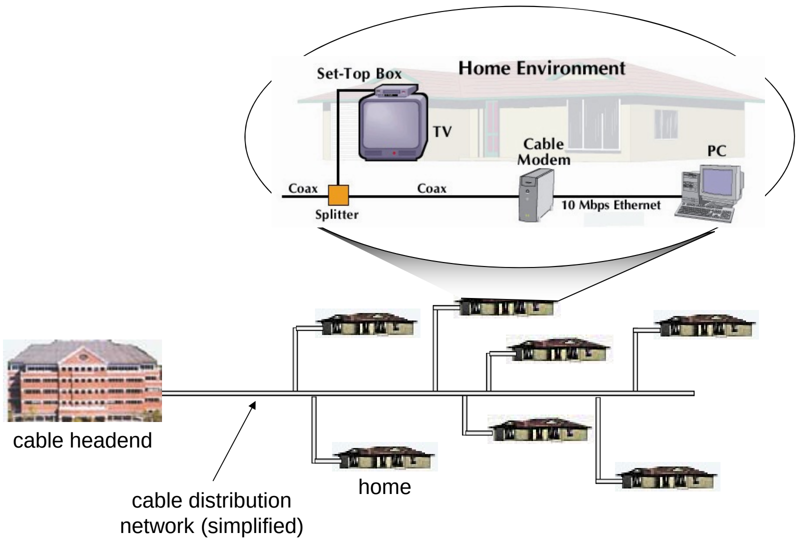
Cable Network Architecture: Overview



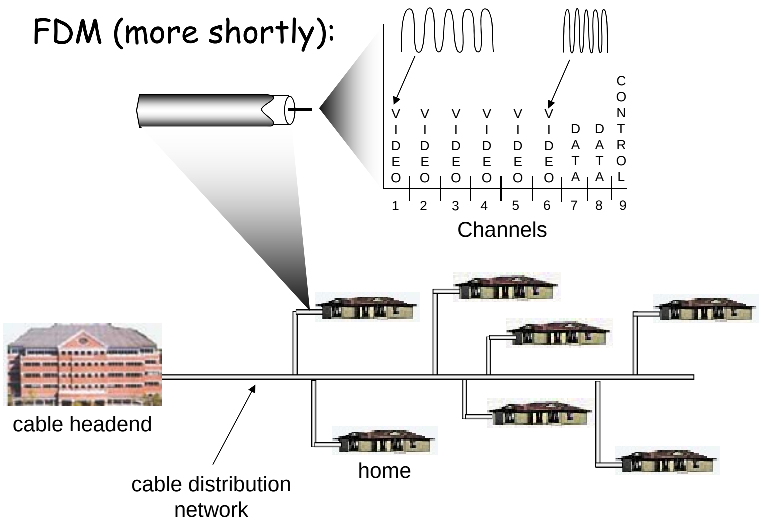
Cable Network Architecture: Overview



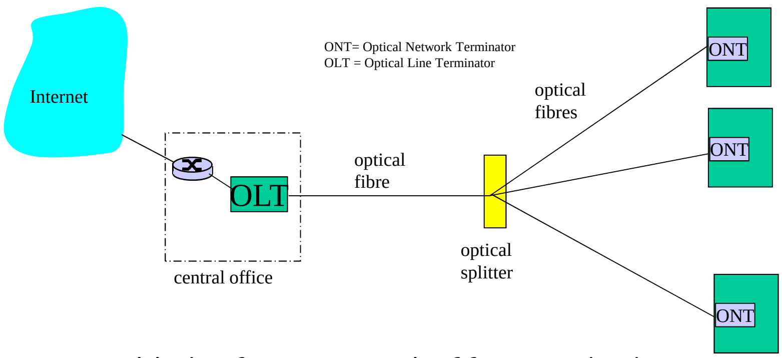
Cable Network Architecture: Overview



Cable Network Architecture: Overview

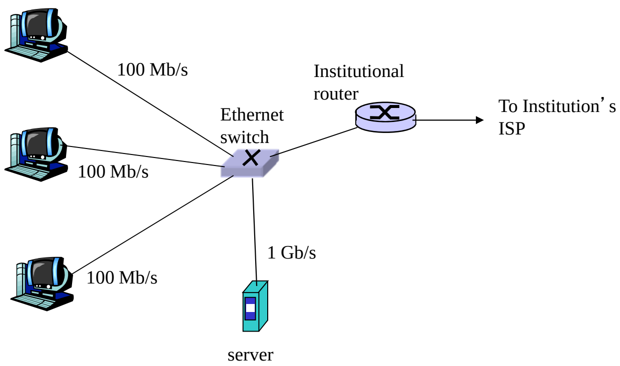


Fibre to the Home



- Optical links from central office to the home
- Two competing optical technologies:
  - Passive Optical network (PON)
  - Active Optical Network (PAN)
- Much higher Internet rates; fibre also carries television and phone services

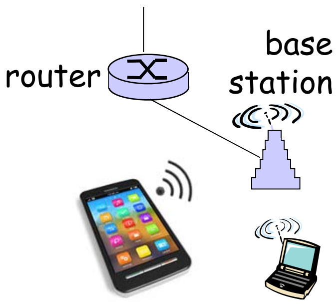
Ethernet Internet access

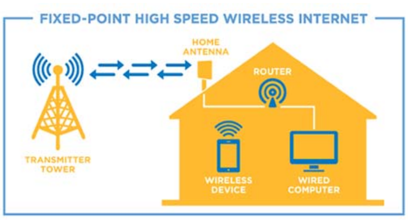


- Typically used in companies, universities, etc
- 10 Mb/s, 100Mb/s, 1Gb/s, 10Gb/s Ethernet
- Today, end systems typically connect into Ethernet switch

Wireless access networks

- shared *wireless* access network connects end system to router
  - via base station aka "access point"
- WiFi wireless LANs:
  - 802.11g: up to 54 Mb/s
  - 802.11n up to 150 Mb/s
- wider-area wireless access
  - "terrestrial" fixed wireless usually < 8 Mb/s and asymmetric
  - Mobile, up to 2Mb/s over 3G and 1 Gb/s in 4G cellular

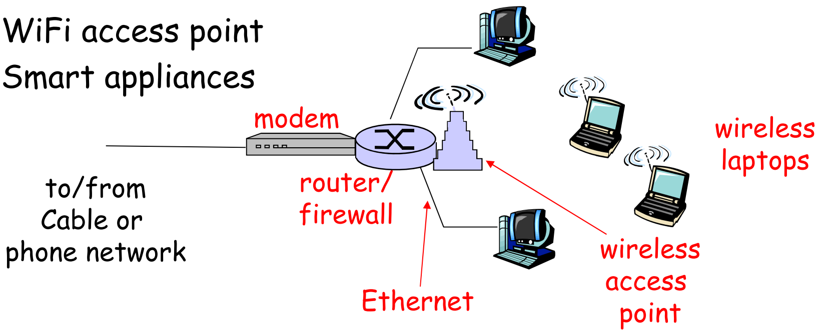




Home networks

Typical home network components:

- DSL or cable modem
- router/firewall/NAT
- Ethernet
- WiFi access point
- Smart appliances



Physical Media

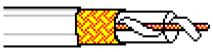
- Bit: propagates between transmitter/rcvr pairs
- physical link: what lies between transmitter & receiver
- guided media:
  - signals propagate in solid media: copper, fibre, coax
- unguided media:
  - signals propagate freely, e.g., radio

Twisted Pair (TP)

- two insulated copper wires
  - Category 3: traditional phone wires, 10 Mb/s Ethernet
  - Category 5: 100Mb/s Ethernet
  - CAT 6: Gigabit Ethernet

Physical Media: coax, fibre

Coaxial cable:
- two concentric copper conductors
- bidirectional
- baseband:
  - single channel on cable
  - legacy Ethernet
- broadband:
  - multiple channels on cable
  - HFC



fibre optic cable:
- glass fibre carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (Tera b/s)
- low error rate: repeaters spaced far apart ; immune to electromagnetic noise

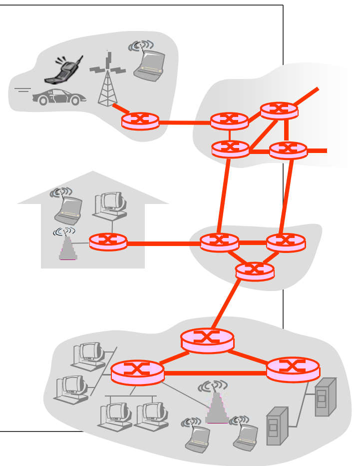
Physical media: radio

- signal carried in electromagnetic spectrum
- no physical "wire"
- Bidirectional, but possibly asymmetric
- propagation environment effects:
  - reflection
  - obstruction by objects
  - Interference
- Radio links types
  - Highly heterogeneous range and performance
  - Terrestrial microwave, WiFI LAN, cellular WAN, satellite

Section 1: roadmap

The Network Core

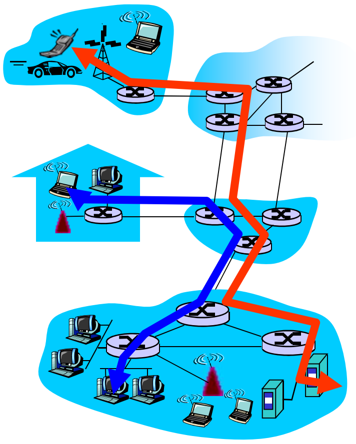
- mesh of interconnected routers
- *the* fundamental question: how is data transferred through net?
  - circuit switching: dedicated circuit per call: telephone net
  - packet-switching: data sent thru net in discrete "chunks"



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



University College Cork CS2505

45

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

University College Cork CS2505

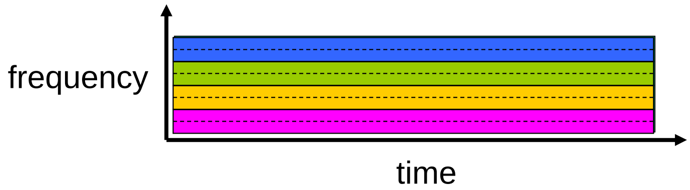
46

Circuit Switching: FDM and TDM

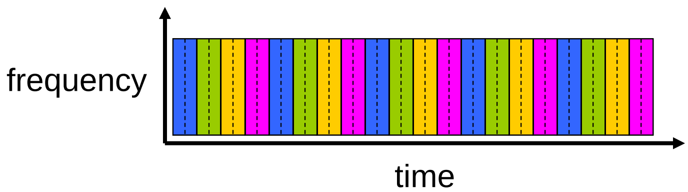
Example:

4 users

FDM



TDM



Numerical example

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - All links are 1.536 Mb/s
  - Each link uses TDM with 24 slots/sec
  - 500 msec to establish end-to-end circuit

Let's work it out!

Network Core: Packet Switching

each end-end data stream divided into *packets*

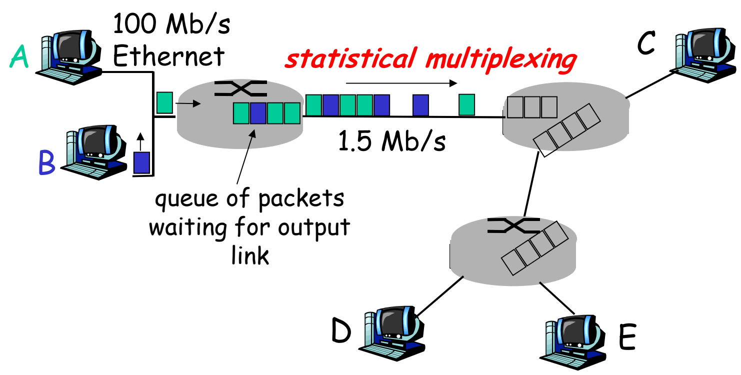
- user A, B packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*

Bandwidth division into "pieces"
Dedicated allocation
Resource reservation

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
  - Node receives complete packet before forwarding

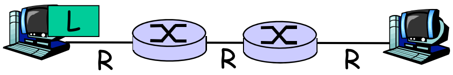
Packet Switching: Statistical Multiplexing



Sequence of A & B packets does not have fixed pattern, bandwidth shared on demand ➔ ***statistical multiplexing***.

TDM: each host gets same slot in revolving TDM frame.

Packet-switching: store-and-forward



- takes L/R seconds to transmit (push out) packet of L bits on to link at R b/s
- *store and forward:* entire packet must arrive at router before it can be transmitted on next link
- delay = 3 L/R (assuming zero propagation delay)

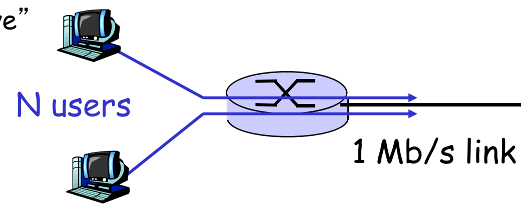
more on delay shortly …

Example:
- L = 7.5 Mbits
- R = 1.5 Mb/s
- transmission delay = 15 sec

Packet switching versus circuit switching

Packet switching allows more users to use network!

- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time
- *circuit-switching:*
  - 10 users
- *packet switching:*
  - At least 10; depends on probability of users being active at same time



Packet switching versus circuit switching

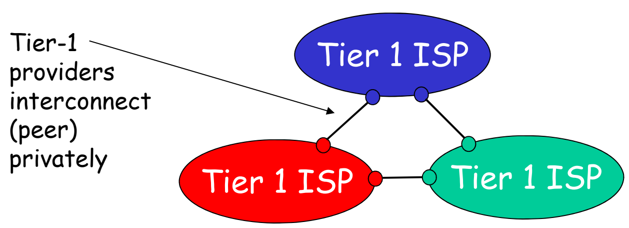
Is packet switching a clear winner?

- 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

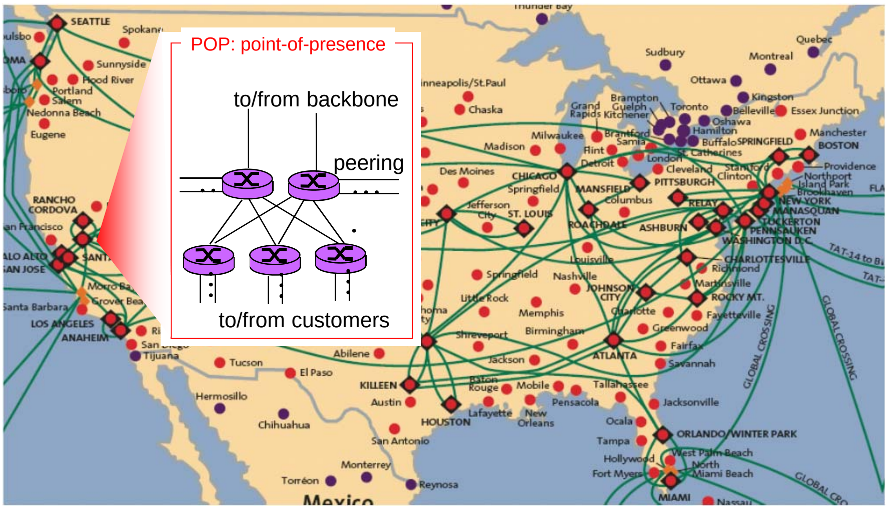
Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?

Internet structure: network of networks

- roughly hierarchical
- at center: "tier-1" ISPs (e.g., Global Crossing, Level 3, Sprint, AT&T), national/international coverage
  - treat each other as equals

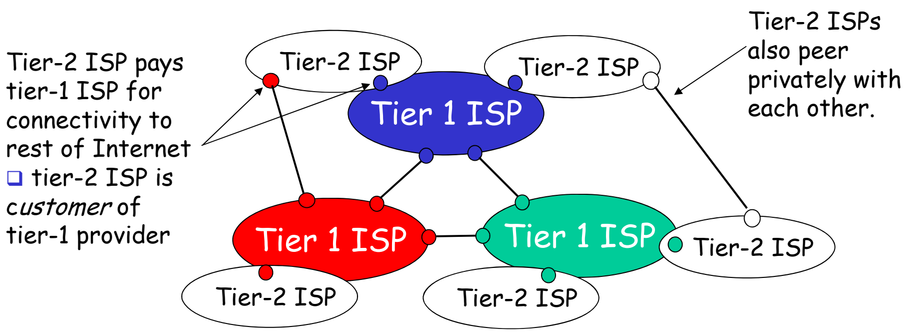


Tier-1 ISP: e.g., Sprint



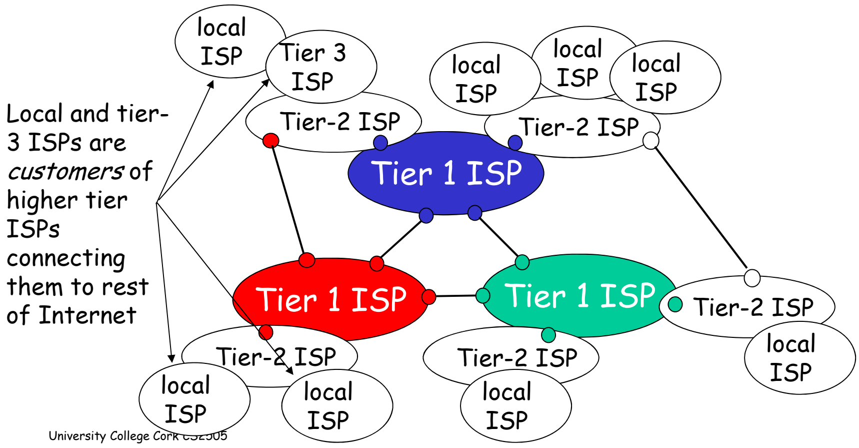
Internet structure: network of networks

- "Tier-2" ISPs: smaller (often regional) ISPs
    - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs



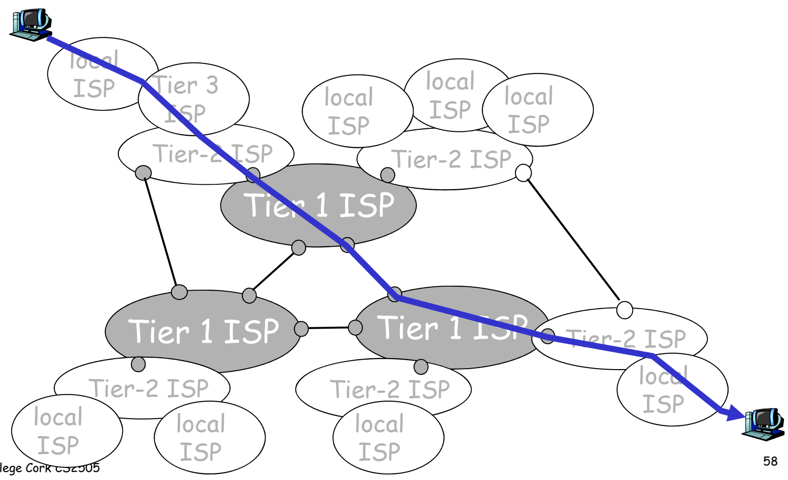
Internet structure: network of networks

- "Tier-3" ISPs and local ISPs
  - last hop ("access") network (closest to end systems)



Internet structure: network of networks

- a packet passes through many networks!

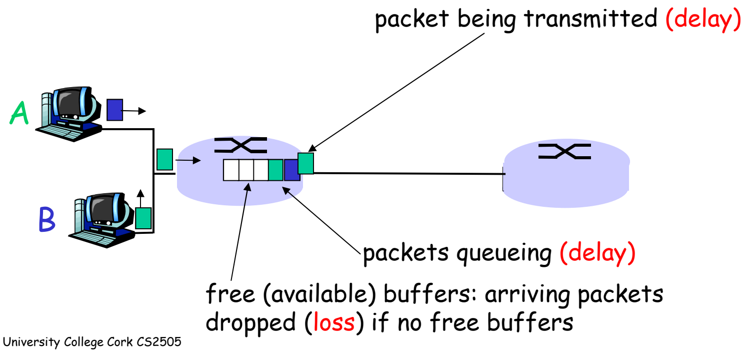


Section 1: roadmap

How do loss and delay occur?

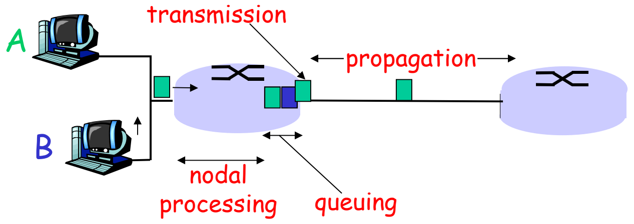
packets *queue* in router buffers

- packet arrival rate to link exceeds output link capacity
- packets queue, wait for turn



Four sources of packet delay

- 1. nodal processing:
  - check bit errors
  - determine output link
- 2. queueing
  - time waiting at output link for transmission
  - depends on congestion level of router

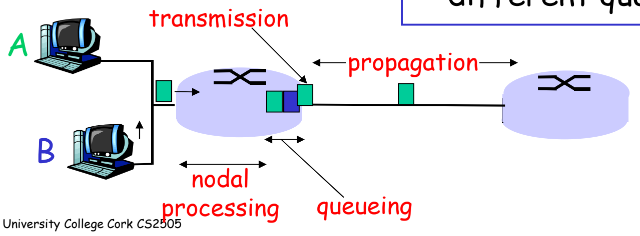


Delay in packet-switched networks

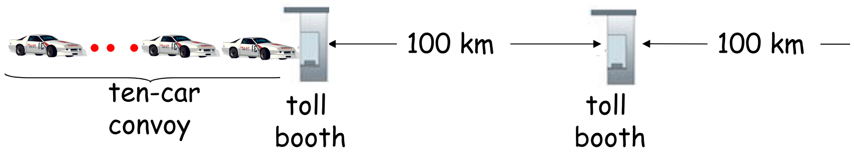
3. Transmission delay:
- R=link bandwidth (b/s)
- L=packet length (bits)
- time to send bits into link = L/R

4. Propagation delay:
- d = length of physical link
- s = propagation speed in medium (~2x10^8 m/sec for copper)
- propagation delay = d/s

Note: s and R are *very* different quantities!

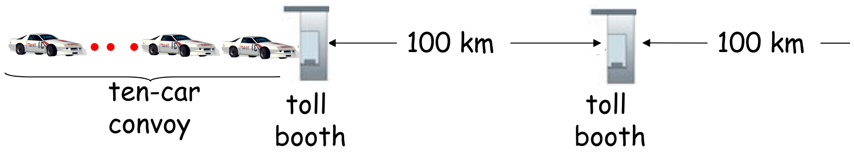


Vehicle analogy



- cars "propagate" at 100 km/hr
- toll booth takes 12 sec to service car (transmission time)
- car~bit; convoy~packet
- Q: How long until cars are lined up before 2nd toll booth?
- Time to "push" all cars 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

Vehicle analogy (more)



- Cars now "propagate" at 1000 km/hr
- Toll booth now takes 1 min to service a car
- Q: Will cars arrive to 2nd booth before all cars serviced at 1st booth?
- Yes! After 7 min, 1st car at 2nd booth and 3 cars still at 1st booth.
- 1st bit of packet can arrive at 2nd router before packet is fully transmitted at 1st router!

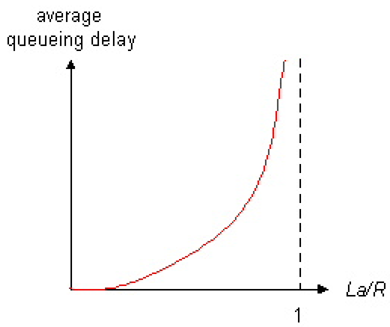
Nodal delay

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

- d_{proc} = processing delay
  - typically a few microsecs or less
- d_{queue} = queuing delay
  - depends on number of hops (routers) and traffic
- d_{trans} = transmission delay
  - = L/R, significant for low-speed links
- d_{prop} = propagation delay
  - a few microsecs to hundreds of millisecs (msecs)

Queueing delay (revisited)

- R=link bandwidth (b/s)
- L=packet length (bits)
- a=average packet arrival rate

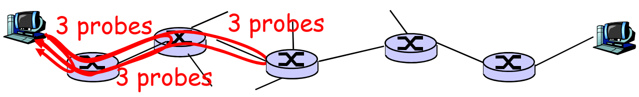


traffic intensity = La/R
where La is arrival rate

- La/R ~ 0: average queueing delay small
- La/R -> 1: delays become large
- La/R > 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.



"Real" Internet delays and routes

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

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

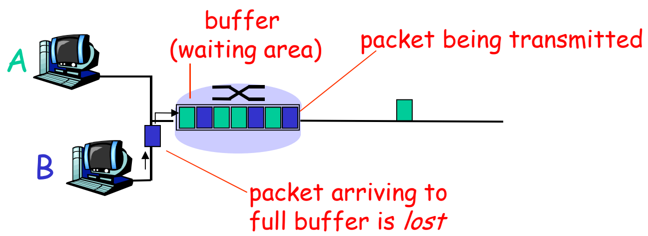
```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18 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)

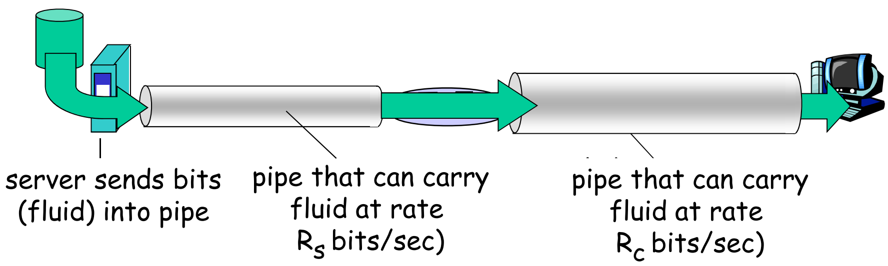
Packet loss

- queue (aka buffer) preceding link 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



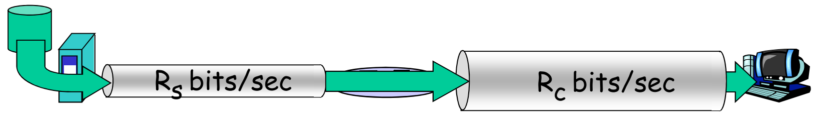
Throughput

- *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

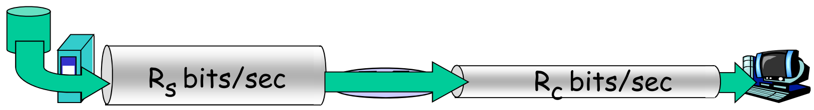


Throughput (more)

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



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

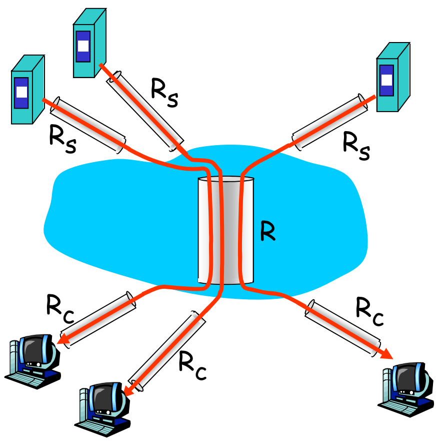


bottleneck link

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

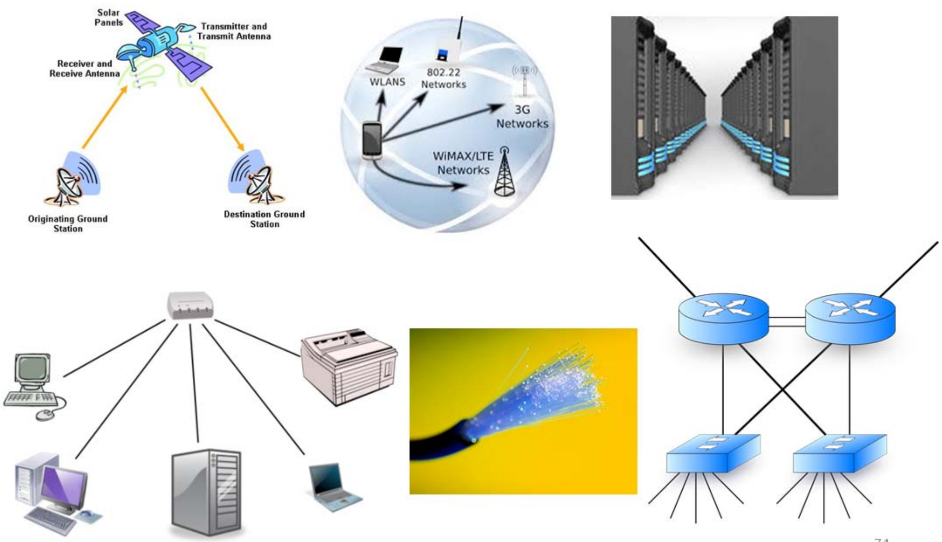


10 connections (fairly) share backbone bottleneck link R bits/sec

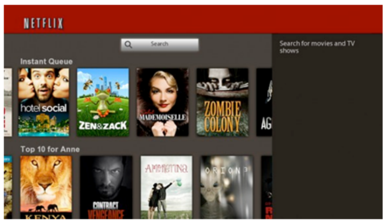
Section 1: roadmap

1.1 What *is* the Internet?
1.2 Network edge
1.3 Network core
1.4 Network performance metrics
1.5 Network protocol architecture
1.6 Networks under attack: security
1.7 History of computer networks

Today's Networks are Complex



…With Varied Applications



75

Dealing with Complexity

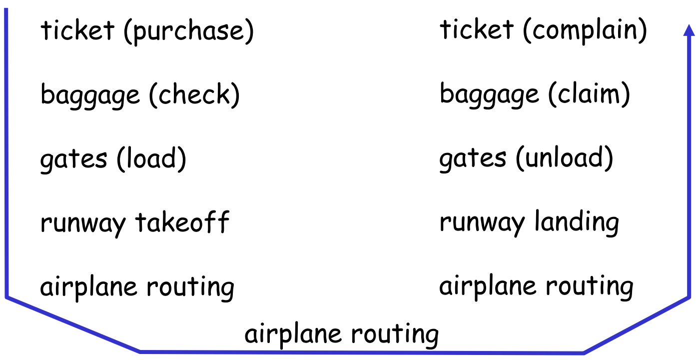
- Need to cope with heterogeneity of function and requirements, including for example
  - Network speed, errors, latency
  - Application security, reliability, quality of service
- Need to facilitate evolution of network elements and applications
  - Upgrade to better technology
  - Adapt to changing needs

76

Layered Architectures

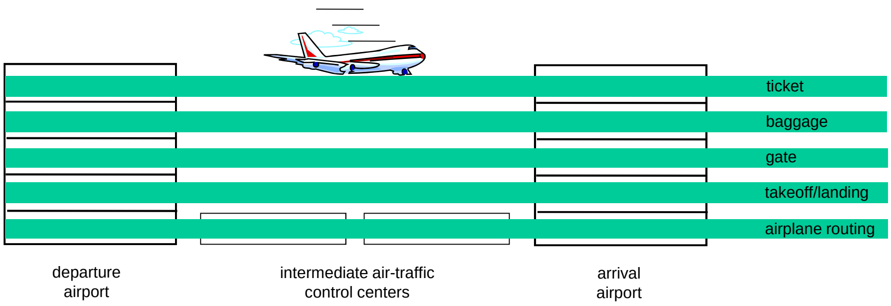
- ❑ Similar network functions are grouped together into distinct layers
- ❑ Layers are organised in a stack, representing increasing functionality
  - ❖ And the dependency of a layer on the layer below
- ❑ Layering is an abstraction that enforces modularity
  - ❖ Makes it easier to change the network functions

Organization of air travel



- ❑ a series of steps

Layering of airline functionality



Layers: each layer implements a service

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

Internet Protocol Architecture

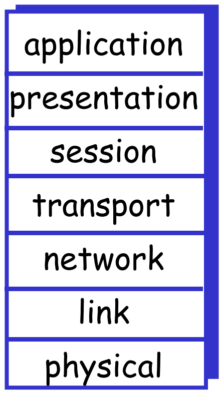
Message *Created by application*

5	Message Application layer	Application-specific functions, e.g. HTTP, E-mail, BitTorrent, DNS
4	Message Transport layer	Common end-to-end delivery functions like reliability, security, e.g. TCP
3	Message Network layer	Core network functions for delivery of a packet across multiple links to the destination, e.g. Internet Protocol (IP)
2	Message Link layer	Functions to deliver a packet across a single link, such as for error detection, e.g. WiFi
1	Message Physical layer	Defines the operation of a physical communication medium e.g. optical fiber

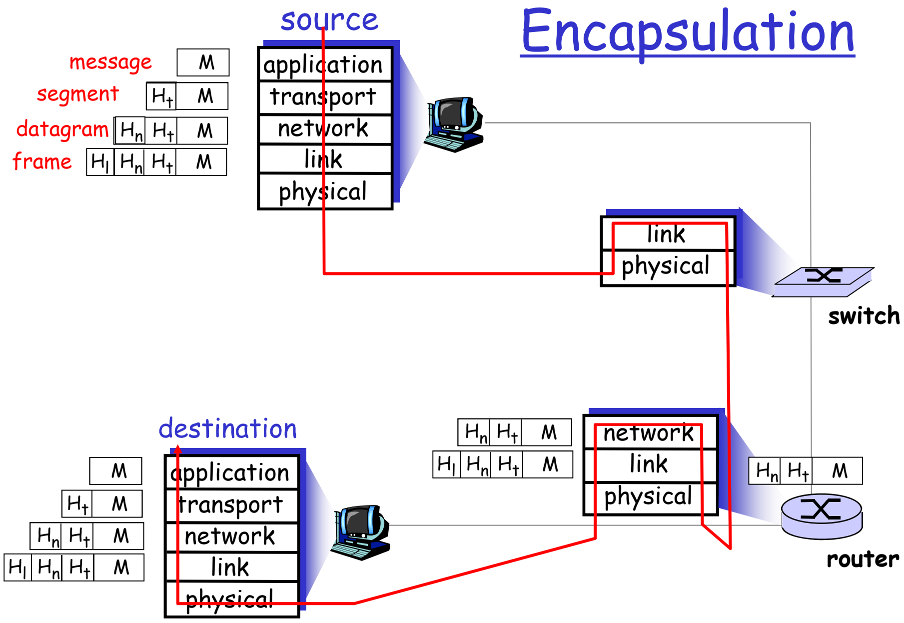
Message sent out on the link!

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?



Encapsulation



Section 1: roadmap

Network Security

- The field of network security is about:
  - 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!

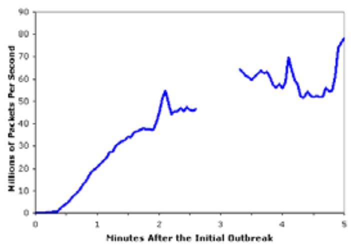
Bad guys can put malware into hosts via Internet

- Malware can get in host from a virus, worm, or trojan horse.
- Spyware malware can record keystrokes, web sites visited, upload info to collection site.
- Infected host can be enrolled in a botnet, used for spam and DDoS attacks.
- Malware is often self-replicating: from an infected host, seeks entry into other hosts

Bad guys can put malware into hosts via Internet

- Trojan horse
  - Hidden part of some otherwise useful software
  - Today often on a Web page (Active-X, plugin)
- Virus
  - infection by receiving object (e.g., e-mail attachment), actively executing
  - self-replicating: propagate itself to other hosts, users
- Worm:
  - infection by passively receiving object that gets itself executed
  - self- replicating: propagates to other hosts, users

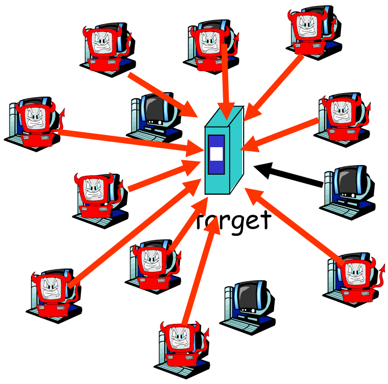
Sapphire Worm: aggregate scans/sec
in first 5 minutes of outbreak (CAIDA, UWisc data)



Bad guys can attack servers and network infrastructure

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

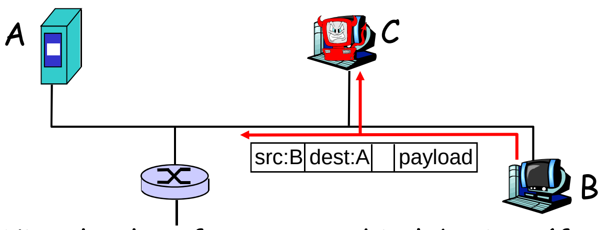
1. select target
2. break into hosts around the network (see botnet)
3. send packets toward target from compromised hosts



The bad guys can sniff packets

Packet sniffing:

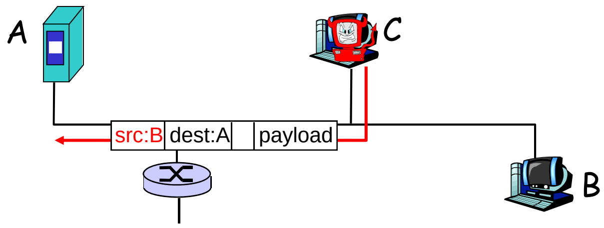
- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



- Wireshark software used in labs is a (free) packet-sniffer

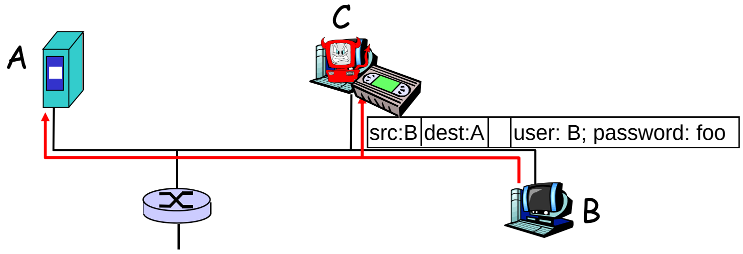
The bad guys can use false source addresses

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



The bad guys can record and playback

- *record-and-playback*: sniff sensitive info (e.g., password), and use later
  - password holder *is* that user from system point of view



Section 1: roadmap

1.1 What *is* the Internet?
1.2 Network edge
1.3 Network core
1.4 Network performance metrics
1.5 Network protocol architecture
1.6 Networks under attack: security
1.7 History of computer networks

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 demonstration
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes

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

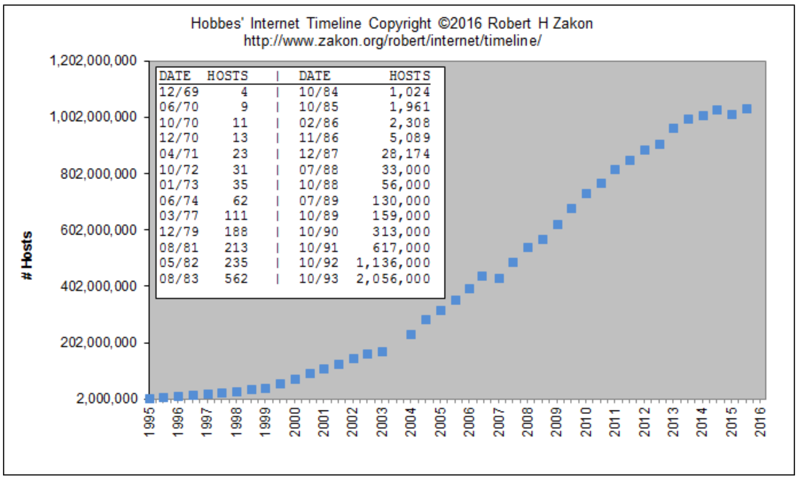
1990, 2000's: commercialization, the Web, new apps

- 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:

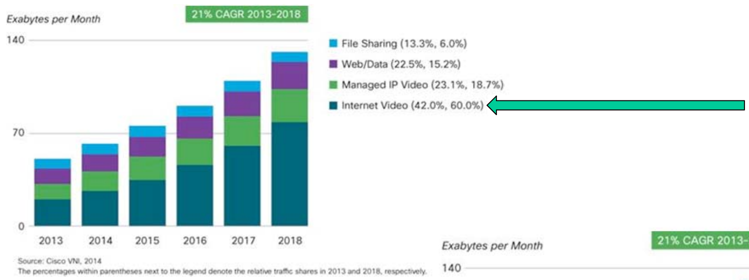
- more killer apps: instant messaging, P2P file sharing
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gb/s

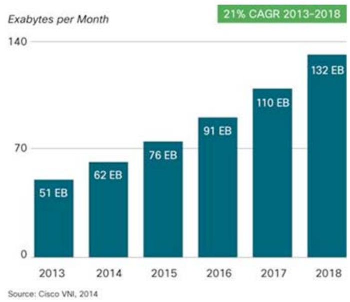
Internet Statistics



DATE	HOSTS	DATE	HOSTS
12/69	4	10/84	1,024
06/70	9	10/85	1,961
10/70	11	02/86	2,308
12/70	13	11/86	5,089
04/71	23	12/87	28,174
10/72	31	07/88	33,000
01/73	35	10/88	56,000
06/74	62	07/89	130,000
03/77	111	10/89	159,000
12/79	188	10/90	313,000
08/81	213	10/91	617,000
05/82	235	10/92	1,136,000
08/83	562	10/93	2,056,000

Global IP Traffic Predictions





- Source: Cisco® Visual Networking Index
- Exabyte = 10^{18} bytes (= 1 billion GB)
- CAGR = Compound Annual Growth Rate

Section 1: 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 *to follow!*