Romain Fontugne

2023 Fall Semester

Information Network Systems

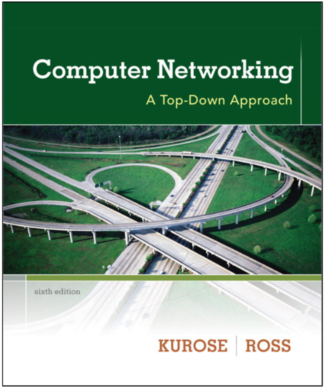
Introduction to the Internet

Course objectives:

Understand how the Internet (and other networks) **works!**

- Get the standard network terminology: router, switch, links, AS, ...
- Understand the principal protocols: Ethernet, TCP, IP, DNS, HTTP...
- Tackle more advanced topics: security, management, ...

Textbook



Computer Networking: A Top-Down Approach

Jim Kurose, Keith Ross

6th edition, 2012, Pearson

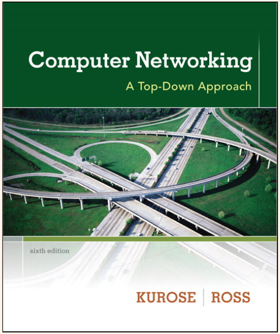
ISBN: 978-0-13-285620-1

Online Ressources

- Website of the textbook: **http://kuroseross.com/**
- Lecture's slides available on Waseda portal
- Questions? **romain@iij.ad.jp**

Today's Lecture: Introduction to the Internet

1. What is the Internet?
2. Internet Infrastructure
   - The Network Edge
   - Physical media
   - The Network Core
3. Packet Switching: Delay, Loss, and Throughput
4. Internet Protocols
5. Internet History



- Chapter 1

What is the Internet?

Definition from Wikipedia:

The Internet is a global system of interconnected computer networks that use the standard Internet protocol suite to serve several billion users worldwide. [plus a few hundreds lines...]

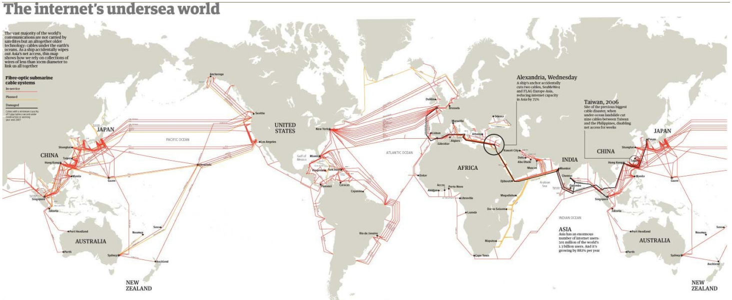
Two points of view:

- Internet Infrastructure
- Internet Services

Internet Infrastructure

The largest engineered system ever created by mankind

- Spans the globe
- 8.7 billion connected devices [Forbes/Cisco,2013] (workstations, servers, laptops, smartphones, tablets, TV, gaming consoles, cars, sensor, security systems, coffee machine....)
- Plus the numerous communication cables and switches!
- Billions of users



Internet Infrastructure

Computing devices

- hosts = end systems
- e.g. PCs, smartphones, servers
- running distributed applications

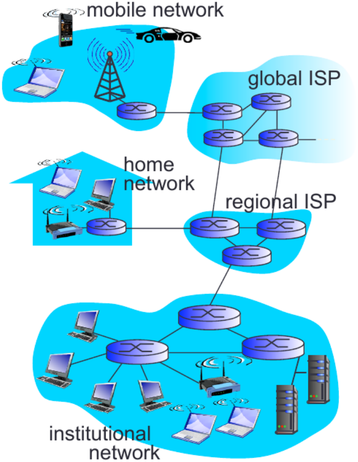
Communication links

- wired: optical fiber, copper wire
- wireless: radio, satellite

Packet switches

- forward packets (chunks of data)
- routers and switches

Internet is a "network of networks"



The Network Edge

Host

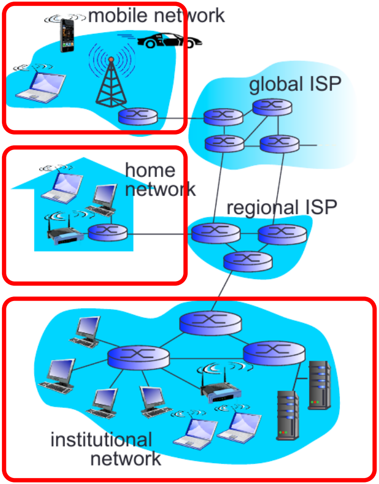
- PCs, smartphones, servers, ...

Home network

- Wired (e.g. Twisted pair)
- Wireless (e.g. Wifi)
- Switch, wireless access point

Internet access

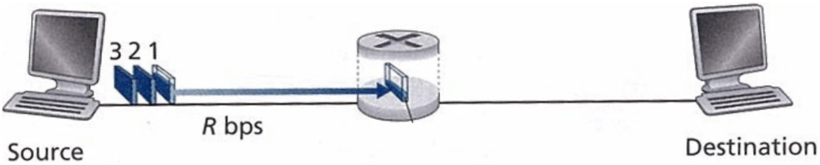
- Modem
- Wired (e.g. DSL, cable, dial-up)
- Wireless (e.g. 3G, satellite)



The Network Edge: Host

Source host sending function

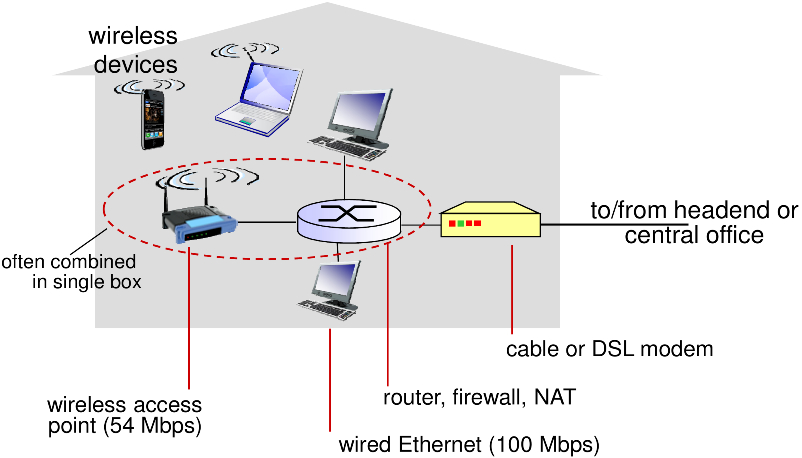
- Takes the application message
- Breaks it into smaller chunks, known as **packets**, of L bits
- **Packet**: help to improve communication performance and reliability



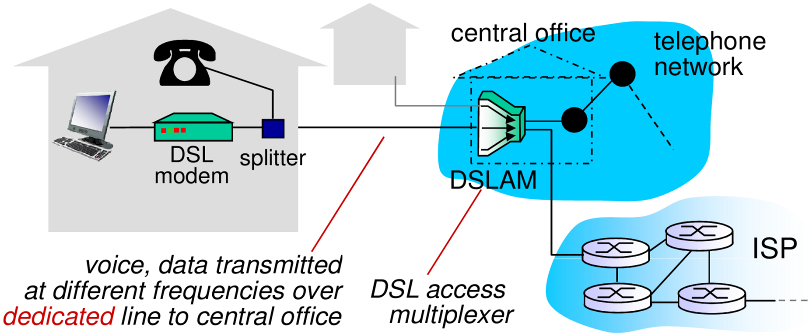
Destination host receiving function

- Aggregate the received packets
- Give the message to the application

The Network Edge: Home network

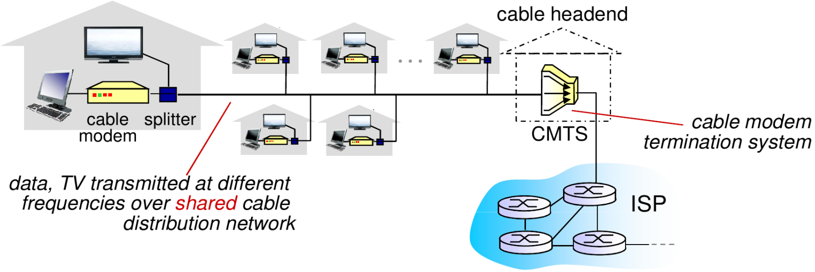


Internet access: Digital Subscriber Line (DSL)



Digital Subscriber Line (DSL)

- Use existing telephone line to central office DSLAM
- $<$ 2.5 Mbps upstream transmission rate (typically $<$ 1 Mbps)
- $<$ 24 Mbps downstream transmission rate (typically $<$ 10 Mbps)



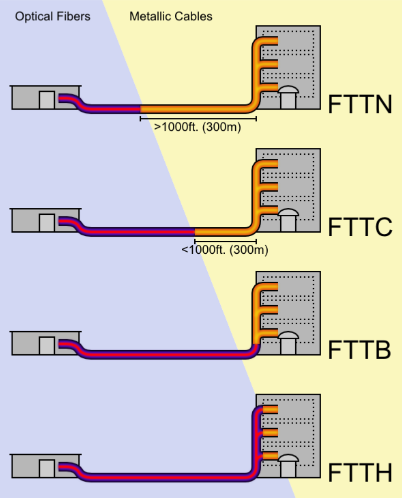
HFC: Hybrid Fiber Coax

- Network of cable and fiber attaches homes to ISP
  → Homes **share access network** to cable headend (unlike DSL)
- **Frequency division multiplexing**: different channels transmitted in different frequency bands
- $<$ 30 Mbps downstream and $<$ 2 Mbps upstream transmission rate

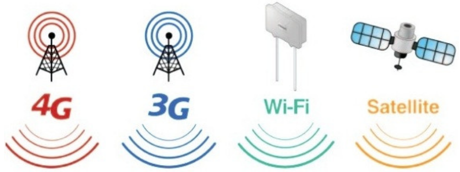
Internet access: Fiber to the Home (FTTH)

Fiber to the Home (FTTH)

- Optical fiber between the central office and the subscriber
- < 1 Gbps downstream
  < 1 Gbps upstream
- Transmission rate are mainly limited by the length of the copper wire
- Deployed mainly in urban areas



Wireless Internet access



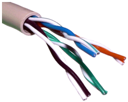
Shared wireless access network

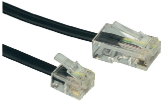
- Terrestrial Radio Channel
  - **Wireless LAN:**
    - within building (tens of meters)
    - e.g. 802.11b/g (WiFi): 11, 54 Mbps transmission rate
  - **Wide-area wireless access:**
    - used by mobile devices (tens of kilometers)
    - e.g. 3G, 4G: 1 to 10 Mbps transmission rate
- Satellite Radio Channel
  - Used in areas where DSL and cable access is unavailable
  - good transmission rate (< 40Mbps) but bad latency (ping $> 1/4$ sec.)

Physical media: Electric signals

Twisted pair (TP) coper wire

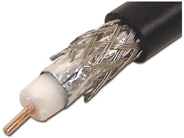
- Least expensive / most deployed (telephone network)
- Transmission rate mainly depends on the cable length
- Category 6 cable can achieve 10Gbps for up to a hundred meters



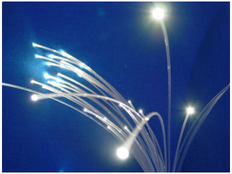


Coaxial cable

- Common in cable television systems
- Shared medium
- Robust to electromagnetic interference



Physical media: others



Fiber optic cable

- Conducts pulse of light
- Immune to electromagnetic interference
- Low signal attenuation ($>$ 100km)
- Good transmission rate (10 to 40Gbps)
- Expensive devices...

Radio

- Electromagnetic spectrum
- no physical "wire" / good for mobile devices
- Propagation issues: obstruction by objects, interference

The Network Core

End systems connect to Internet via access ISPs

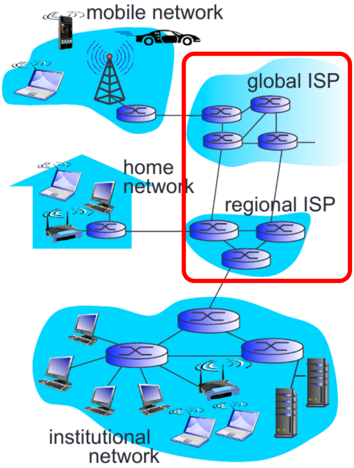
- Residential, company and university ISPs

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

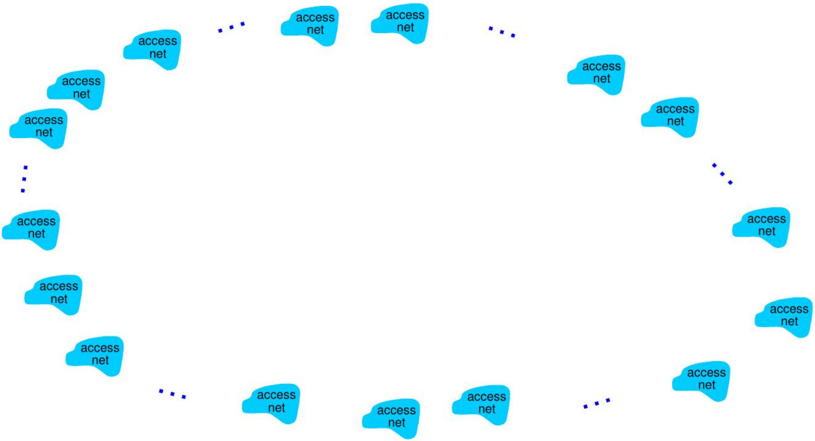
- Evolution was driven by economics and national policies



Internet structure: network of networks

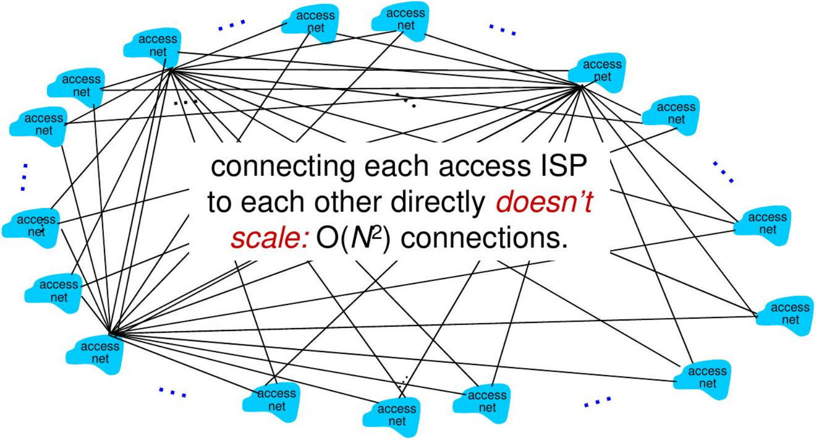
Question:

Given thousands of access ISPs, how to connect them together?



Option 1:

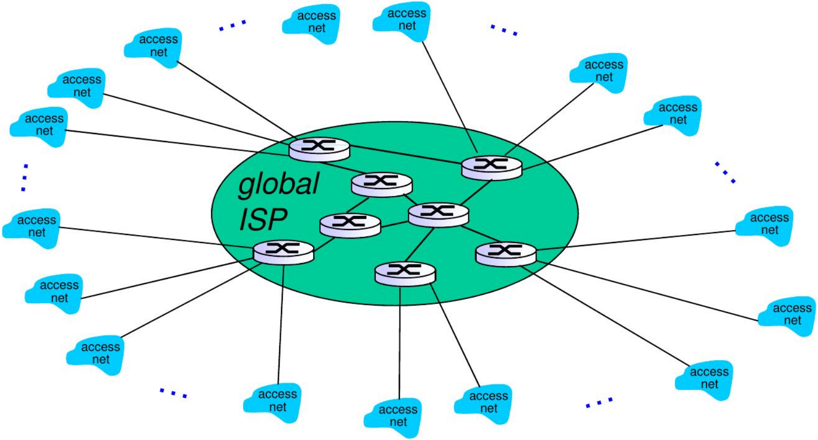
Connect each access ISP to every other access ISP?



Internet structure: network of networks

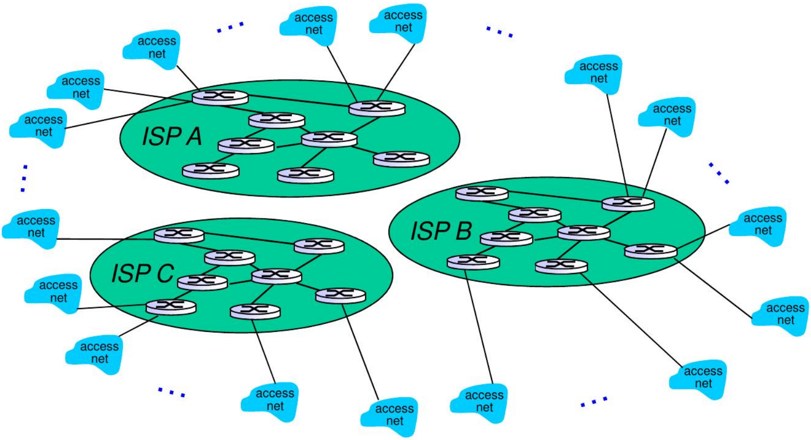
Option 2:

Connect each access ISP to a global transit ISP? Customer and provider ISPs have economic agreement.



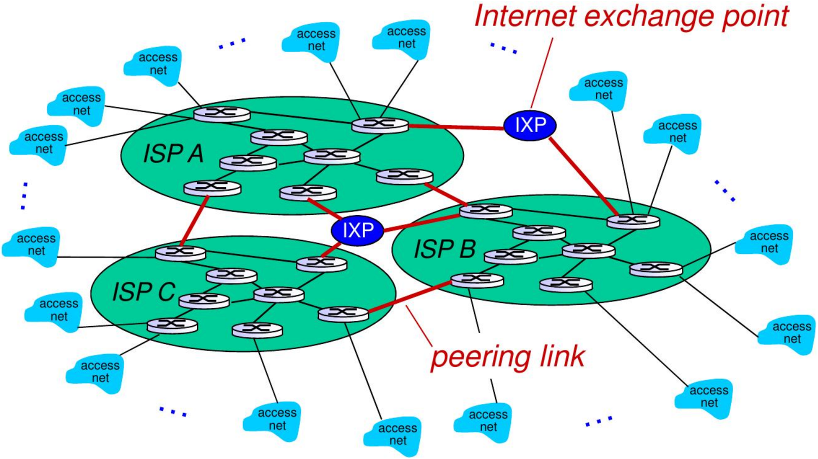
Internet structure: network of networks

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



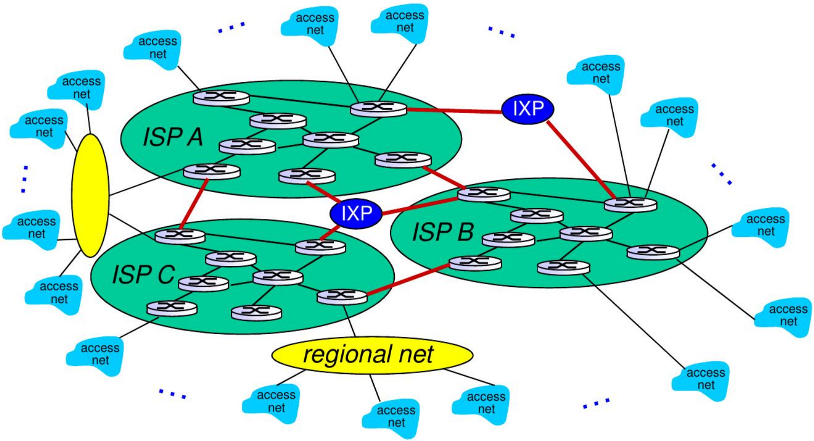
Internet structure: network of networks

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

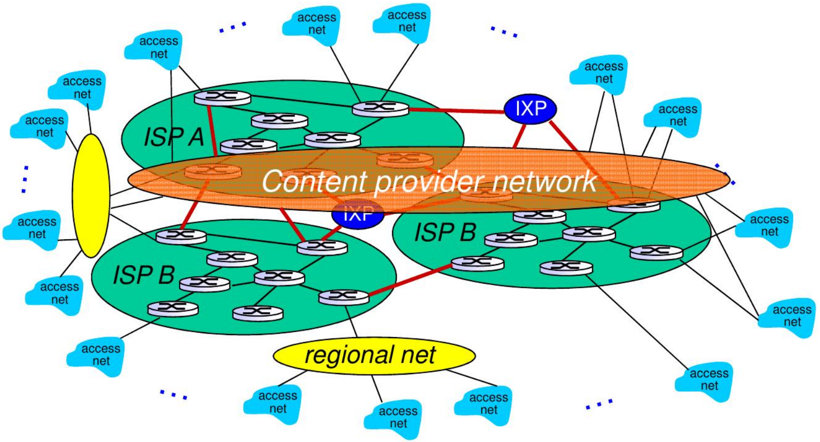


Internet structure: network of networks

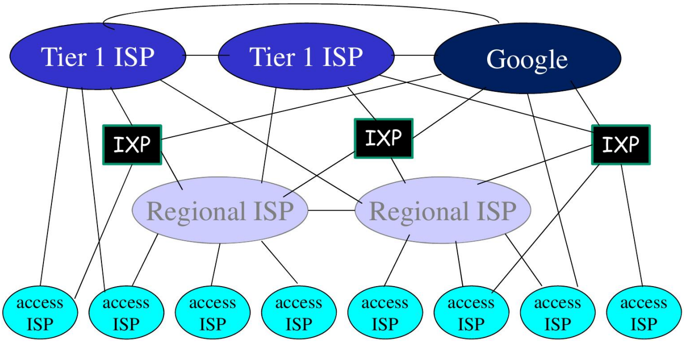
... and regional networks may arise to connect access networks to ISPs



... 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



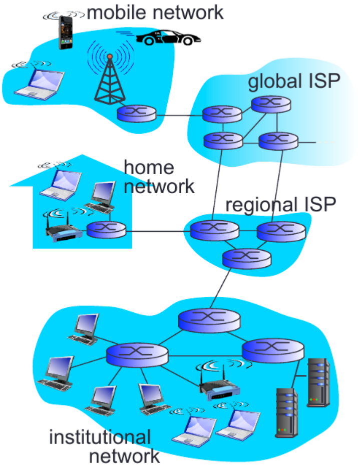
At center: small number of well-connected large networks

- "tier-1" commercial ISPs (e.g. Level 3, Sprint, AT&T, NTT) National and international coverage
- Content provider network (e.g. Google): private network that connects data centers to Internet, often bypassing tier-1, regional ISPs

Packet Switching

Packet-switching

- Source Host: Split application messages in packets
- Router/packet switch: Forward packets from one router to the next, in order to reach the destination host
  (Packets are transmitted at full link capacity)
- Destination Host: Re-construct the application message



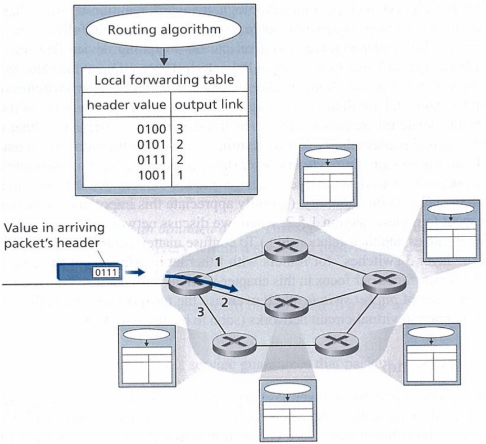
Packet-Switching: key functions

Routing algorithms

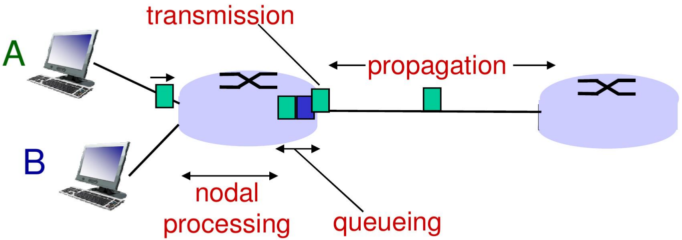
- Determines source-destination route taken by packets

Forwarding table

- Move packets from router's input to the appropriate router's output link



Packet-Switching: nodal delay

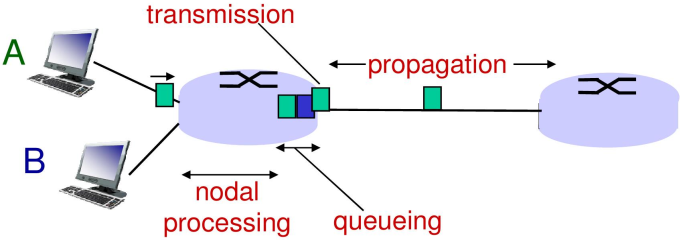


$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

At each node along a path packets suffer from several delays

- Processing delay
- Queuing delay
- Transmission delay
- Propagation delay

Packet-Switching: nodal delay

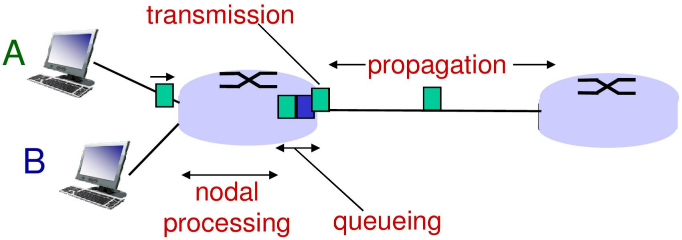


$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

Processing delay (d_{proc})

- Check bit error (meaningless to transmit corrupted packets)
- Examine the destination address and determine output link
- Typically on the order of μsec

Packet-Switching: nodal delay



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

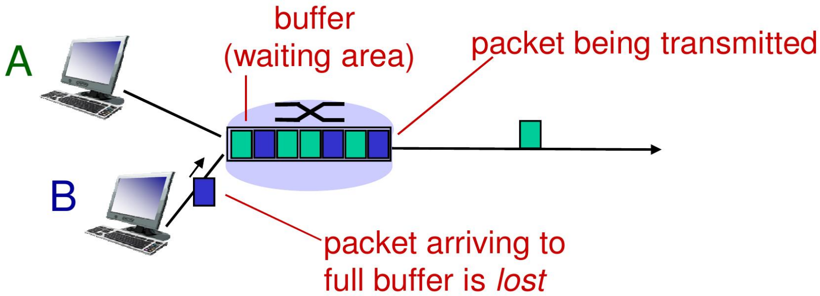
Queuing delay (d_{queue})

- Packet arrival rate (temporarily) exceeds output link capacity
- Packets queue, wait for turn
- Depends on congestion level of router
- Typically μsec to msec

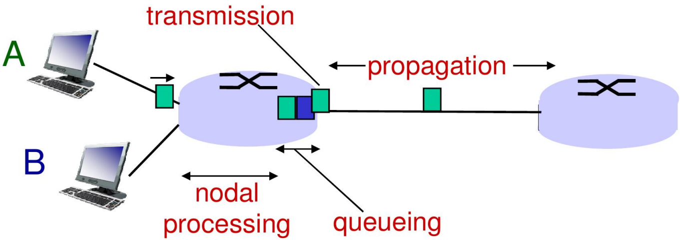
Packet-Switching: packet loss

When packet arrival rate exceeds output link capacity

- Packets queue, wait for turn
- But queue size is limited
- Packet arriving to full queue are dropped (aka lost)
- Lost packet may be retransmitted by previous node, by source end system, or not at all



Packet-Switching: nodal delay

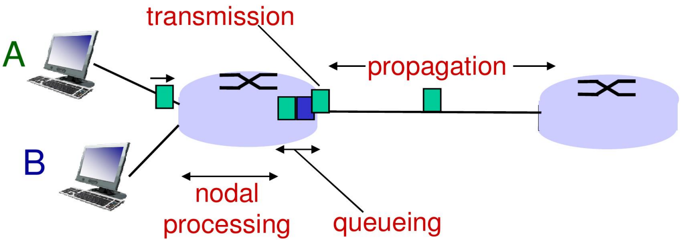


$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

Transmission delay (d_{trans})

- Time required to push all of the packet's bit into the link
- Takes $d_{trans} = L/R$ seconds to push L-bit packet into a link at R bps
- Proportional to the packet size not the distance between the nodes!
- Typically μsec to msec

Packet-Switching: nodal delay



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

Propagation delay (d_{prop})

- Time required to travel from the beginning to the end of the link
- $d_{prop} = d/s$: Proportional to d, length of the link, and s, wave propagation speed (2.8 to 3.8m/sec))
- Typically on the order of msec

Packet-Switching: end-to-end delay

Nodal delay

- From when last bit of packet arrives at this node to when last bit arrives at next node
- $d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$

End-to-end delay over N identical nodes/links from client c to server s

- From when last bit of packet leaves client to when last bit arrives at server
- $d_{C-S} = d_{prop} + Nd_{nodal}$

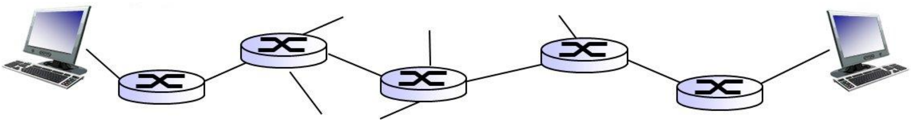
Round Trip Time (RTT)

- From when last bit of packet leaves client to when last bit arrives at server
- $RTT = d_{C-S} + d_{S-C} + t_{server}$
  where t_{server} is server processing time

Packet-Switching: real Internet delay and route

Easy way to check what Internet delay & route look like?

- traceroute program: measures delays along an end-to-end path
- For each router i on the path:
  - Send three packets to the router i
  - Router i returns packets to the sender
  - Sender measures times interval between transmission and reply



Traceroute

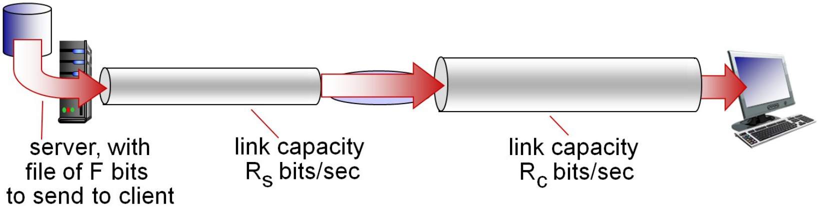
- Example from Tokyo to www.nytimes.com

```
traceroute to www.nytimes.com (170.149.168.130), 30 hops max, 60 byte packets
 1  ve-222.juniper1.hongo.wide.ad.jp (203.178.135.1)  1.309 ms  1.335 ms  1.360 ms
 2  ve-130.foundry4.nezu.wide.ad.jp (203.178.136.185)  0.214 ms  0.240 ms  0.268 ms
 3  ve-42.foundry6.otemachi.wide.ad.jp (203.178.136.65)  0.347 ms  0.369 ms  0.390 ms
 4  ve-51.cisco2.notemachi.wide.ad.jp (203.178.141.142)  0.606 ms  0.663 ms  0.740 ms
 5  ge-8-2.a15.tokyjp01.jp.ra.gin.ntt.net (203.105.72.17)  0.872 ms  0.936 ms  0.971 ms
 6  ae-5.r24.tokyjp05.jp.bb.gin.ntt.net (203.105.72.153)  5.657 ms  5.487 ms ae-5.r25.toky
 7  ae-13.r20.tokyjp01.jp.bb.gin.ntt.net (129.250.6.191)  0.607 ms  0.575 ms ae-12.r20.tok
 8  as-1.r20.sttlwa01.us.bb.gin.ntt.net (129.250.4.189)  98.351 ms  84.700 ms  93.908 ms
 9  ae-1.r05.sttlwa01.us.bb.gin.ntt.net (129.250.5.47)  90.392 ms  105.557 ms  90.339 ms
10  ae-0.level3.sttlwa01.us.bb.gin.ntt.net (129.250.8.74)  84.487 ms  98.076 ms  93.649 ms
11  ae-32-52.ebr2.Seattle1.Level3.net (4.69.147.182)  162.847 ms  171.983 ms  176.430 ms
12  ae-2-2.ebr2.Denver1.Level3.net (4.69.132.54)  198.330 ms  191.047 ms  202.615 ms
13  ae-3-3.ebr1.Chicago2.Level3.net (4.69.132.62)  177.894 ms  168.466 ms  182.681 ms
14  ae-6-6.ebr1.Chicago1.Level3.net (4.69.140.189)  203.643 ms  213.071 ms  200.345 ms
15  ae-2-2.ebr2.NewYork2.Level3.net (4.69.132.66)  166.730 ms  166.527 ms  181.364 ms
16  ae-1-100.ebr1.NewYork2.Level3.net (4.69.135.253)  181.125 ms  176.312 ms  173.774 ms
17  ae-4-4.ebr1.NewYork1.Level3.net (4.69.141.17)  207.655 ms 4.69.201.45 (4.69.201.45)  1
18  ae-2-2.ebr1.Newark1.Level3.net (4.69.132.98)  191.437 ms  179.563 ms  176.368 ms
19  ae-11-51.car1.Newark1.Level3.net (4.69.156.5)  177.557 ms  185.348 ms  189.754 ms
20  NEW-YORK-TI.car1.Newark1.Level3.net (4.30.129.234)  191.326 ms  190.797 ms  191.178 ms
21  170.149.168.130 (170.149.168.130)  169.056 ms  205.439 ms  218.156 ms
```

Packet-Switching: Throughput

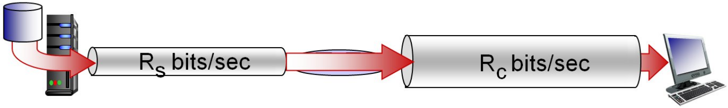
Throughput

- Rate (bits/time unit) at which bits are transferred between sender/receiver
- Instantaneous: rate at a given point in time
- Average: rate over longer period of time
- Analogy: pipe/volume/fluid

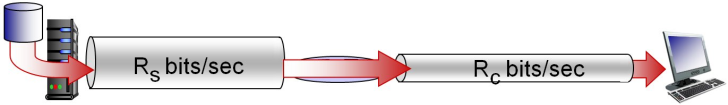


Packet-Switching: bottleneck

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



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

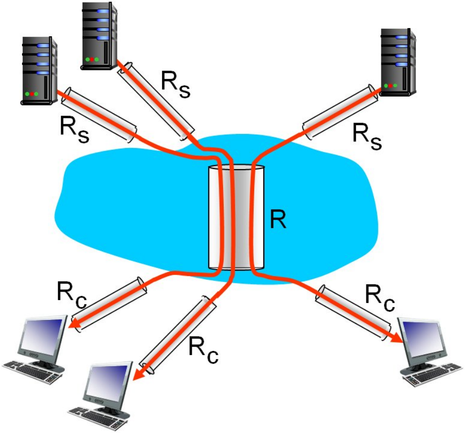


→ Bottleneck link: link on the path that constrains end-to-end throughput

Packet-Switching: Internet scenario

Shared infrastructure

- With 10 clients
- Throughput for each connection: $min(R_C, R_S, R/10)$
- In practice: R_C is often bottleneck
- But bottleneck can be high throughput link shared by many



Packet Switching: Alternatives?

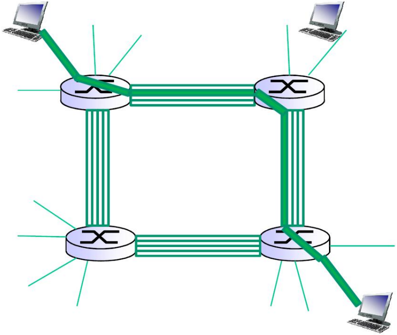
So far we've seen that packet switching has severe drawbacks:

- packet delay,
- packet loss,
- variable throughput...

What about other communication methods?

- Do we really need packets?
- Can't we avoid these delays
- and ensure a certain throughput?

Alternative to Packet Switching: Circuit Switching



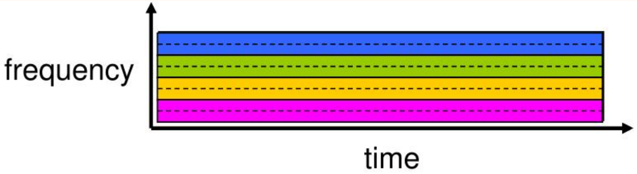
Circuit Switching

- Commonly used in traditional telephone network
- Call: Allocate resources between source & destination
- Guarantee circuit performance
- No sharing: Circuit segment idle if not used by call
- Limited number of users

Multiplexing in Circuit Switching

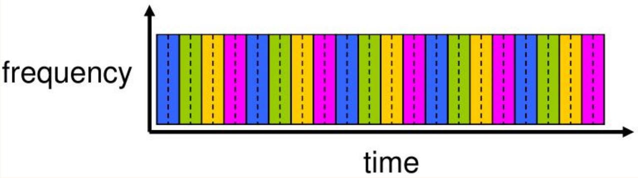
Frequency Division Multiplexing (FDM)

Example with 4 users:



Time Division Multiplexing (TDM)

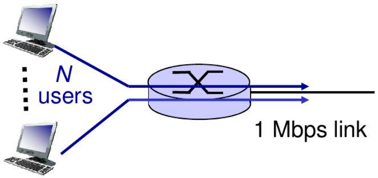
Example with 4 users:



Packet Switching vs. Circuit Switching

Example 1

- 1 Mbps link
- Each user:
  - 100 kbps when "active"
  - active 10% time



Circuit Switching

- $N = 1Mbps/100kbps$
  $= 10$ users

Packet Switching

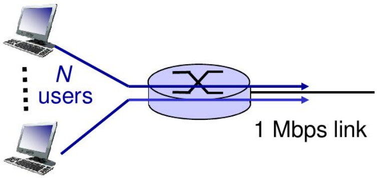
- Let say $N = 35$ users
- probability of > 10 active at the same time is less than .0004

→ Packet switching allows more users to share the network resources!

Packet Switching vs. Circuit Switching

Example 2

- 1 Mbps link
- 9 idle users
- 1 active user sends one thousands 1000-bit packets



Circuit Switching

- It takes 10 seconds for the active user to send all the packets

Packet Switching

- It takes 1 second for the active user to send all the packets

→**Packet switching takes full advantage of network resources!**

Packet Switching vs. Circuit Switching

Is packet switching always better?

✓ Great for bursty data (Internet browsing $\neq$ telephone call)
- resource sharing
- simpler, no call setup

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

- How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - still an unsolved problem!

What is the Internet?

Definition from Wikipedia:

The Internet is a global system of interconnected computer networks that use the standard Internet protocol suite to serve several billion users worldwide. [plus a few hundreds lines...]

Two points of view:

- Internet Infrastructure
- Internet Services

Internet Services

Internet enables computers to exchange data in various ways

- different types of data: file transfer, video/audio calls, emails,...
- different properties: Secured connections, reliable data delivery,...

How does it work...??

- with such complex network?
- and congestion/packet loss?
- and untrustworthy intermediate networks?
  (and other problems you'll see in next lectures...)

→ The Internet protocol suite!

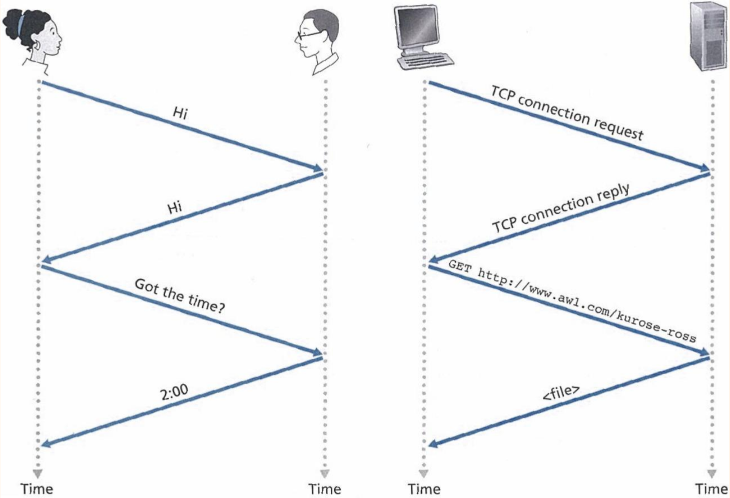
What is a protocol?

Protocol

- Set of rules governing the exchange or transmission of data between devices
- Define format, order of msgs sent and received between devices, and actions taken on msg transmission, receipt
- Internet protocol standards are defined by Internet Engineering Task Force (IETF)

Example of protocol

Easier to understand with human analogy:



Internet Protocol suite

All communication activity in Internet is governed by protocols:

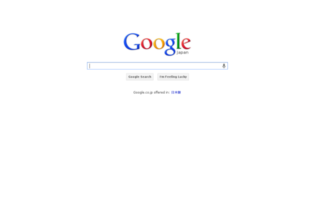
- You've probably seen many of them...
- ...but this is just the tip of the iceberg!

...since you use them everyday

- **World Wide Web: HTTP, HTTPS**
  (Firefox, Chrome, Internet Explorer, ...)
- **Electronic mail: SMTP, IMAP, POP**
  (Outlook, Thunderbird, Gmail, ...)
- **File transfer: FTP, SFTP, BitTorrent**
  (μTorrent, Firefox, Chrome, ...)
- **Messaging, voice and video calls: Skype, XMPP, SIP**
  (Google talk, skype, ...)

Internet Protocol suite

For example: A simple web request



- Involves more than 5 protocols

Each protocol

- completes a simple task
- it offers this service to applications/protocols
- uses services from other protocols

An analogy:

 Japan PM writes a letter →  US president reads the message

 His secretary prepare the envelope and address

 The secretary gives the letter to the president

 Postal service determines it must fly to Washington an put it in the corresponding mailbag

 The mails are sorted and sent to corresponding buildings

 At the airport mailbag is put in a container and loaded in airplane

 The container is opened and mailbag is passed to the postal service

 The plane flies to Washington

ist … en … PM i … t t … ag … itic … ng, unaware of the underlying mechanism … em … er … by plane? from

Layering

Divide complex systems in layers

- Each layer
  - implements a service
  - relies on services provided by layer below
- Modular design
  - change of implementation of layer's service transparent to rest of system
  - eases maintenance, updating of system

Internet Protocol Stack

Layers description

- Application: supporting network applications (e.g. FTP, SMTP, HTTP)
- Transport: process-process data transfer (e.g. TCP, UDP)
- Network: routing of datagrams from source to destination (e.g. IP, routing protocols)
- Link: data transfer between neighboring network elements (e.g. Ethernet, 802.11 (WiFi), PPP)
- Physical: bits "on the wire"

IP Stack

IP Stack
Application
Transport
Network
Link
Physical

ISO/OSI reference model

Two more layers

- Presentation: allow applications to interpret meaning of data, e.g., machine-specific conventions, encryption, compression
- Session: synchronization, checkpointing, recovery of data exchange
- Internet stack "missing" these layers!
  If needed these services must be implemented in application

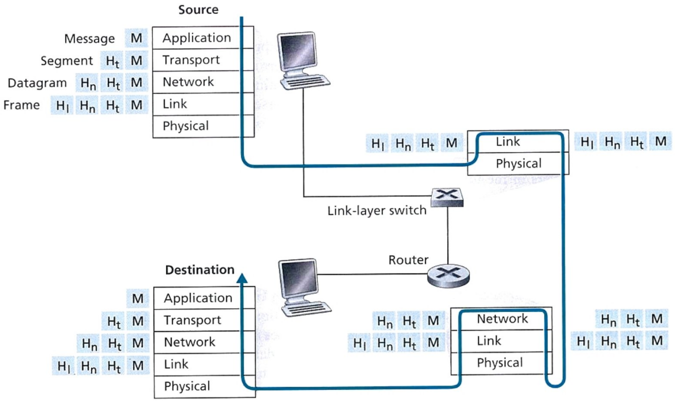
OSI model

Application
Presentation
Session
Transport
Network
Link
Physical

Encapsulation

Encapsulation at each layer: packet = header + payload

- Letter analogy: letter ⇄ envelope ⇄ mailbag ⇄ container
- The data is encapsulated accordingly and unpacked only as far as necessary by nodes passed



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 and four nodes network
- 1972:
  - ARPAnet public demo
  - 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
- 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
- $\rightarrow$ 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

- Aggressive deployment of broadband access
- Increasing ubiquity of high-speed wireless access
- 2011: Number of wireless devices outnumbered the number of wired devices
- 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, email, etc.
- E-commerce, universities, enterprises running their services in “cloud” (eg, Amazon EC2)

Introduction to the Internet: Summary

Today's lecture covered a "ton" of material!

- Internet overview
- network edge
- network core
- packet-switching
- performance: loss, delay, throughput
- protocol
- layering, service models
- history

In next lecture

Link layer: data transfer between neighboring network elements (e.g. Ethernet, 802.11 (WiFi), PPP)

IP Stack

Application

Transport

Network

Link

Physical

Today's important points

- Physical medias (optic fibre, wireless, coper wire,...)
- Nodal delay
- Throughput
- Protocol / Layering