Chair for Network Architectures and Services – Prof. Carle
Department of Computer Science
TU München

Master Course Computer Networks IN2097

Prof. Dr.-Ing. Georg Carle
Christian Grothoff, Ph.D.

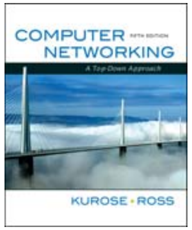
Chair for Network Architectures and Services
Department of Computer Science
Technische Universität München
http://www.net.in.tum.de

Course Outline (tentative)

- Part 1: Internet protocols
  1. Overview on Computer Networks
  2. Application Layer
  3. Transport Layer
  4. Network Layer
  5. Link Layer
- Part 2: Advanced Concepts
  6. Network Monitoring and Measurements
  7. Quality of Service
  8. Signalling and Internet Telephony Services
  9. Network design principles
     - *common themes:* signaling, indirection, virtualization, multiplexing, randomization, scalability
     - *implementation principles:* techniques
     - *network architecture:* the big picture, synthesis
     - *Future Internet* approaches

Acknowledgements

- *Significant parts of Part 1 of this lecture are based on the book*
  Computer Networking: A Top Down Approach ,
  5th edition.
  Jim Kurose, Keith Ross
  Addison-Wesley, April 2009.
- The lecture is based to a significant extent on slides by Jim Kurose and Keith Ross



Jim Kurose
University of Massachusetts, Amherst

Keith Ross
Polytechnic Institute of New York University

Course organization

- Lecture
  - Friday, 10:15-11.45, MI H2 weekly
  - Monday, 16:15-17.45, MI H2 first weekly, then typically bi-weekly
- Exercises
  - After start of exercises, typically bi-weekly Monday 16:15-17.45
- Students are requested to subscribe to lecture and exercises at
  http://www.net.in.tum.de/en/teaching/ws1011/lectures/masterkurs-rechnernetze/
  ⇨ Email list, svn access for subscribers of course
- TUMonline: required for exam registration
- Questions and Answers / Office hours
  - Prof. Dr. Georg Carle, carle@net.in.tum.de
    - After the course and upon appointment (typically Thursday 11-12)
  - Christian Grothoff, Ph.D., grothoff@net.in.tum.de
- Course Material
  - Slides are available online. Slides may be updated during the course.

Grading

- Exercises
  - prepare for the examination (but do not give a bonus)
- Practical assignments
  - Two practical assignments are planned
  - You have to succeed in at least one
  - They will be graded
- Our concept for grading
  (may be changed – rules will be fixed before registration for the exam)
  - Final examinations will be oral and give an individual grade.
    You must pass the oral exam for being successful in the course.
  - For overall grade, grade of one practical assignment gives 25% of final grade
  - If your grade for a second practical assignment is better than your examination grade, it is accounted for by another 25%

Questions

- Who studies what?
  - Diploma degree?
  - Master in Informatics?
  - Master in Informatics – English Track?
  - Master in Information Systems [Wirtschaftsinformatik]?
  - Other Master courses?
  - Bachelor in Informatics?
  - Other courses
- Who comes from where?
- Which previous relevant courses?
  - IN0010 - Grundlagen Rechnernetze und Verteilte Systeme?
  - What else?

Chair for Network Architectures and Services – Prof. Carle
Department of Computer Science
TU München

Overview

Chapter 1: Introduction

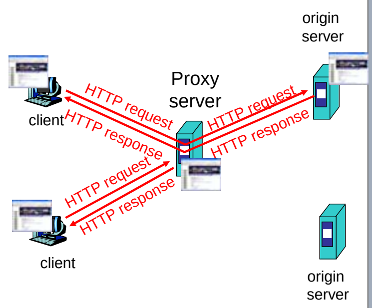
- what's the Internet?
- what's a protocol?
- network edge; hosts, access net, physical media
- network core: packet/circuit switching, Internet structure
- performance: loss, delay, throughput
- protocol layers, service models

Chapter 2: Application layer

- Principles of network applications
- Web and HTTP
- DNS
- P2P applications
- Socket programming with TCP
- Socket programming with UDP

Chapter 2: Web caches (proxy server)

- Goal: satisfy client request without involving origin server
- non-transparent web cache: user sets browser: Web accesses via cache
- browser sends all HTTP requests to cache
  - object in cache: cache returns object
  - else cache requests object from origin server, then returns object to client

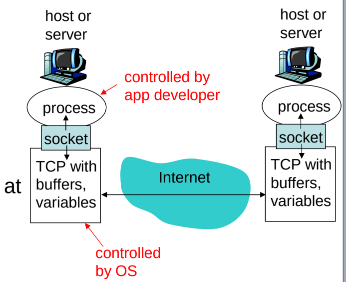


Chapter 3 Outline

- Transport-layer services
- Multiplexing and demultiplexing
- Connectionless transport: UDP
- Connection-oriented transport: TCP
  - segment structure
  - reliable data transfer
  - flow control
  - connection management
- TCP congestion control

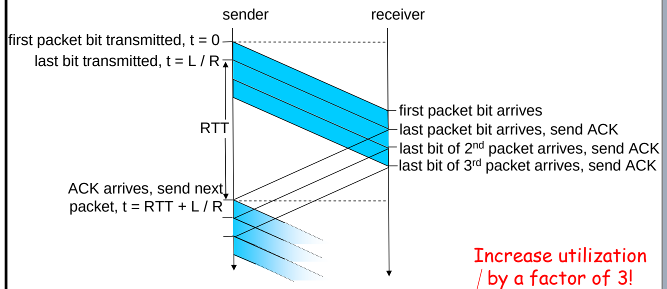
Chapter 3: Sockets

- process sends/receives messages to/from its socket
- socket analogous to door
  - sending process shoves message out door
  - sending process relies on transport infrastructure on other side of door which brings message to socket at receiving process



- API: (1) choice of transport protocol; (2) ability to fix a few parameters

Chapter 3: Pipelining: increased utilization



$$U_{sender} = \frac{3 * L / R}{RTT + L / R} = \frac{.024}{30.008} = 0.0008$$

Chapter 3: TCP Round Trip Time and Timeout

Setting the timeout

- **EstimtedRTT** plus "safety margin"
  - large variation in **EstimatedRTT ->** larger safety margin
- first estimate of how much SampleRTT deviates from EstimatedRTT:

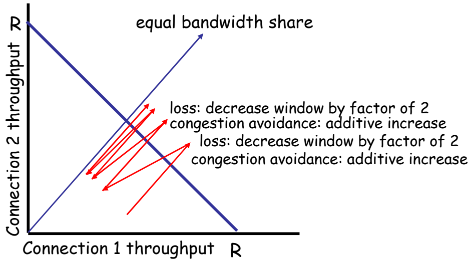
```
DevRTT = (1-β)*DevRTT +
             β*|SampleRTT-EstimatedRTT|

(typically, β = 0.25)
```

Then set timeout interval:

```
TimeoutInterval = EstimatedRTT + 4*DevRTT
```


Chapter 3: Why is TCP fair?

Two competing sessions:
- Additive increase gives slope of 1, as throughout increases
- multiplicative decrease decreases throughput proportionally



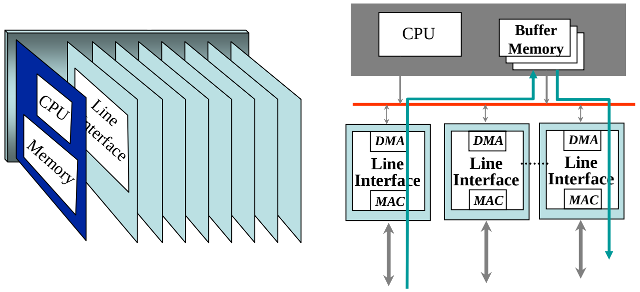
Chapter 4: Network Layer

Part 1
- Introduction
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP

Part 2
- IPv6
- NAT
- Virtual circuit and datagram networks
- What's inside a router

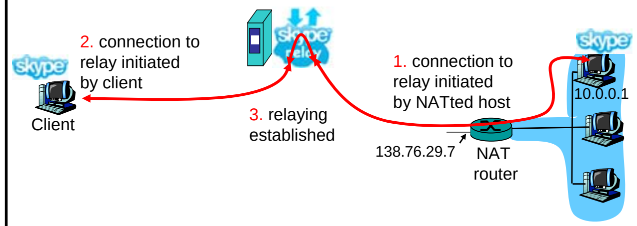
Part 3
- Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- Routing in the Internet
  - RIP
  - OSPF
  - BGP
- Broadcast and multicast routing

Chapter 4: First-Generation IP Routers



Chapter 4: NAT traversal

- One of several NAT traversal solutions: relaying (used in Skype)
  - NATed client establishes connection to relay
  - External client connects to relay
  - relay bridges packets between to connections



Chapter 5: Link Layer

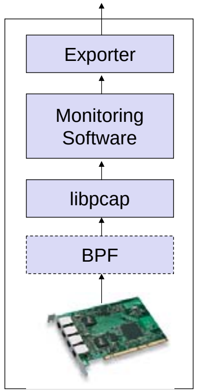
- Introduction and services
- Multiple access protocols
- Link-layer Addressing
- Ethernet
- Link-layer switches
- Link virtualization: ATM, MPLS

Chapter 6: Network Measurements

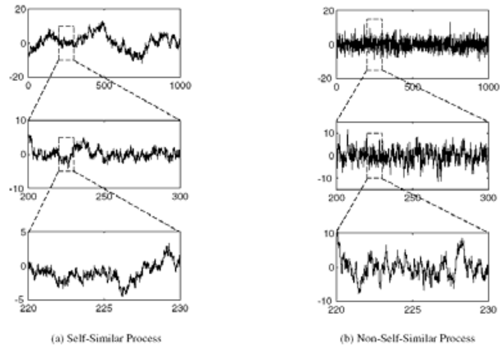
- Introduction
- Architecture & Mechanisms
- Protocols
  - IPFIX (Netflow Accounting)
  - PSAMP (Packet Sampling)
- Scenarios

Chapter 6: Monitoring Probe

- Standardized data export
- Monitoring Software
- HW adaptation, [filtering]
- OS dependent interface (BSD)
- Network interface



Chapter 6: Self-Similar Stochastic Process



Chapter 7 outline – Quality-of-Service Support

- Link virtualization: ATM
- Providing multiple classes of service
- Providing Quality-of-Service (QoS) guarantees
- Signalling for QoS

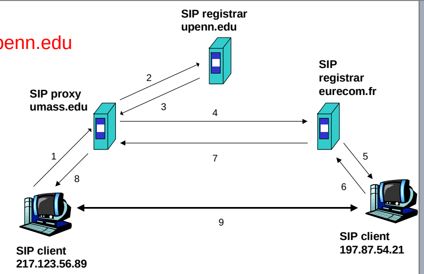
Chapter 8: Signaling

signaling: exchange of messages among network entities to enable (provide service) to connection/call

- before, during, after connection/call
  - call setup and teardown (state)
  - call maintenance (state)
  - measurement, billing (state)
- between
  - end-user <-> network
  - end-user <-> end-user
  - network element <-> network element
- examples
  - Q.921, SS7 (Signaling System no. 7): telephone network
  - Q.2931: ATM
  - RSVP (Resource Reservation Protocol)
  - H.323: Internet telephony
  - **SIP** (Session Initiation Protocol): Internet telephony

Chapter 9: Voice over IP Example

Caller jim@umass.edu
places a call to keith@upenn.edu
(1) Jim sends INVITE message to umass SIP proxy.
(2) Proxy forwards request to upenn registrar server.
(3) upenn server returns redirect response, indicating that it should try keith@eurecom.fr



(4) umass proxy sends INVITE to eurecom registrar.
(5) eurecom registrar forwards INVITE to 197.87.54.21, which is running keith's SIP client.
(6-8) SIP response sent back
(9) media sent directly between clients.
Note: SIP ack messages not shown.

Chapter 9: Network design principles

Network design principles

- *common themes:* signaling, indirection, virtualization, multiplexing, randomization, scalability
- *implementation principles:* techniques
- *network architecture:* the big picture, synthesis
- *Future Internet* approaches

Chapter 9: Design Principles and Future Internet

Internet Design Philosophy (Clark'88)
(In order of importance)

Different ordering of priorities would make a different architecture!

0 Connect existing networks
initially ARPANET, ARPA packet radio, packet satellite network
1. Survivability
   - ensure communication service even with network and router failures
2. Support multiple types of services
3. Must accommodate a variety of networks
4. Allow distributed management
5. Allow host attachment with a low level of effort
6. Be cost effective
7. Allow resource accountability

Chair for Network Architectures and Services – Prof. Carle
Department of Computer Science
TU München

Introduction

Chapter 1: Introduction

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
- protocol layers, service models

Our goal:

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

Chapter 1: roadmap

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

PC
server
wireless laptop
cellular handheld

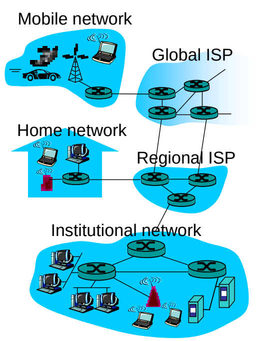
- millions of connected computing devices: *hosts = end systems*
  - running *network apps*

access points
wired links

- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate = *bandwidth*

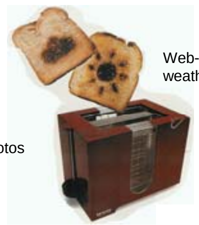
router

- *routers:* forward packets (chunks of data)



"Cool" internet appliances

IP picture frame
http://www.ceiva.com/
Free invitations for guests to send photos



Web-enabled toaster + weather forecaster

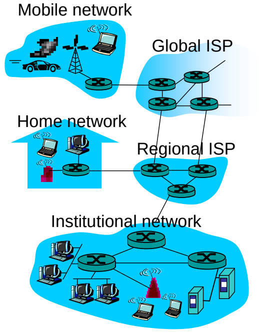
World's smallest web server in 1999

Internet phones

⇨ Who knows other cool internet appliances?

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

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



What's a protocol?

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

… specific msgs sent
… specific actions taken when messages received, or other events

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

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

Chapter 1: roadmap

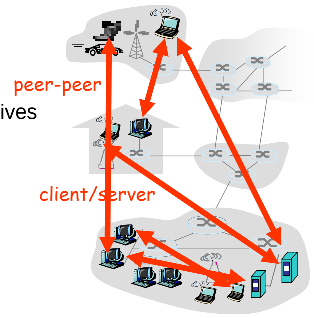
1.1 What *is* the Internet?
1.2 Network edge
- end systems, access networks, links

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

1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models

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



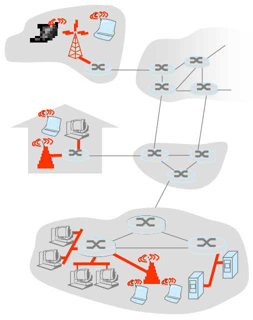
Access networks and physical media

Q: How to connect end systems to edge router?

- residential access networks
- institutional access networks (school, company)
- mobile access networks

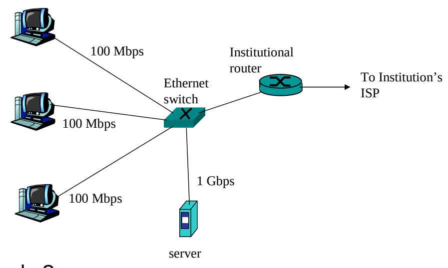
Relevant:

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



Ethernet Internet access

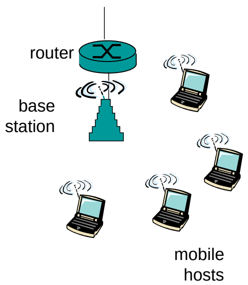
- Typically used in companies, universities, etc
  - 10 Mbs, 100Mbps, 1Gbps, 10Gbps Ethernet
  - Today, end systems typically connect into Ethernet switch



⇨ why?

Wireless access networks

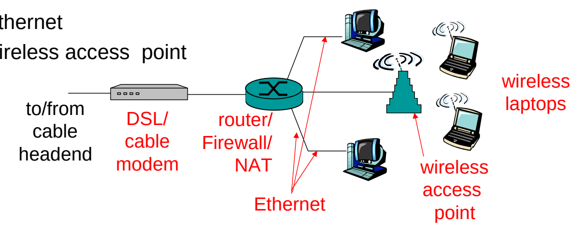
- shared *wireless* access network connects end system to router
  - via base station aka "access point"
- wireless LANs:
  - 802.11b/g (WiFi): 11 or 54 Mbps
- wider-area wireless access
  - provided by telco operator
  - ~1Mbps over cellular system (HSDPA)
  - next cellular network technology: LTE (10's Mbps) over wide area



Home networks

Typical home network components:

- DSL or cable modem
- router/firewall/NAT
- Ethernet
- wireless access point

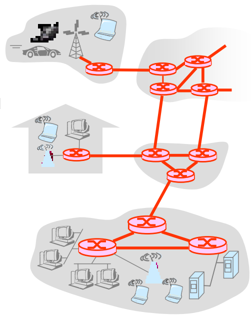


⇨ Our research project AutHoNe: targetting many innovations

Chapter 1: roadmap

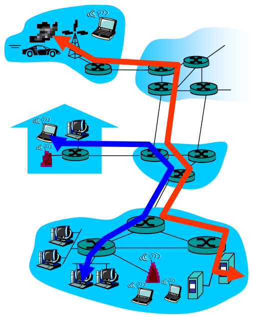
The Network Core

- 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
- 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
- Inefficient for bursty sources (⇨why?)
- Quality guarantee, but call blocking



Network Core: Packet Switching

each end-end data stream divided into *packets*

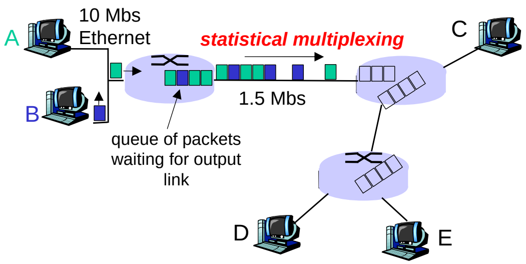
- user A, B packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*

resource contention:

- aggregate resource demand can exceed amount available
- congestion: packets queue, wait for link use
- store and forward: packets move one hop at a time
  - Node receives complete packet before forwarding

Bandwidth division into "pieces"
Dedicated allocation
Resource reservation

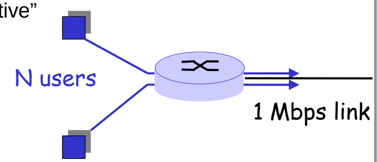
Packet Switching: Statistical Multiplexing



- Sequence of A & B packets does not have fixed pattern ➔ ***statistical multiplexing***.
- In TDM each host gets same slot in revolving TDM frame.

Packet switching versus circuit switching

- For bursty sources, Packet switching allows more users to use network! Example:
  - 1 Mbit link
  - each user:
    - 100 kbps when "active"
    - active 10% of time
  - circuit-switching:
    - 10 users
  - packet switching:
    - with 35 users, probability > 10 active less than .0004



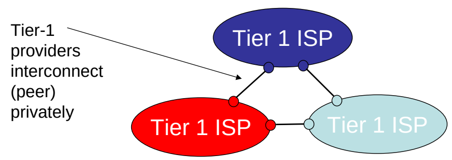
Packet switching versus circuit switching

Is packet switching obviously better than circuit switching?

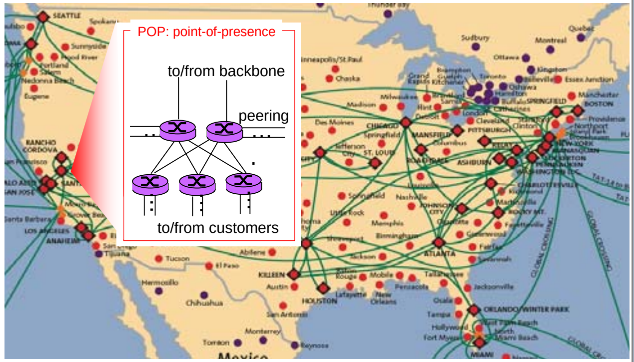
- packet switching is great for bursty data
  - resource sharing
  - simpler, no call setup
- possibility of excessive congestion: packet delay and loss
  - protocols needed for reliable data transfer, congestion control
- Q: How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - Internet-wide still an unsolved problem (⇨later)

Internet structure: network of networks

- roughly hierarchical
- at center: "tier-1" ISPs (AT&T, Global Crossing, Level 3, NTT, Qwest, Sprint, Tata, Verizon (UUNET), Savvis, TeliaSonera), national/international coverage
  - treat each other as equals
  - can reach every other network on the Internet without purchasing IP transit or paying settlements.

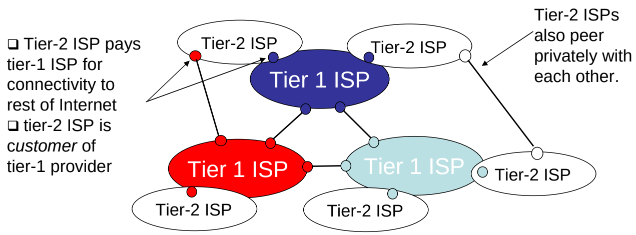


Tier-1 ISP: e.g., Sprint



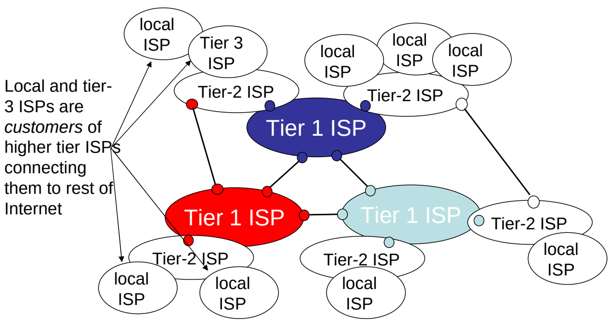
Internet structure: network of networks

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



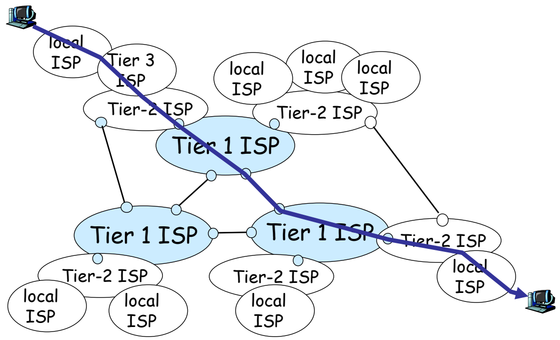
Internet structure: network of networks

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



Internet structure: network of networks

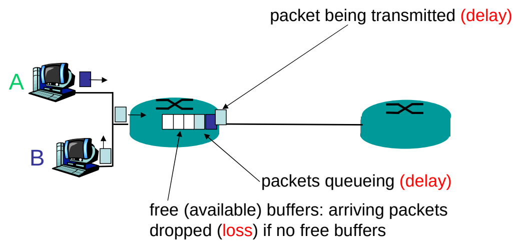
- a packet passes through many networks!



Chapter 1: roadmap

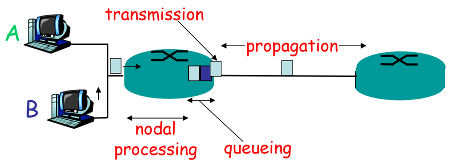
How do loss and delay occur?

packets *queue* in router buffers
- packet arrival rate to link exceeds output link capacity
- packets queue, wait for turn



Four sources of packet delay

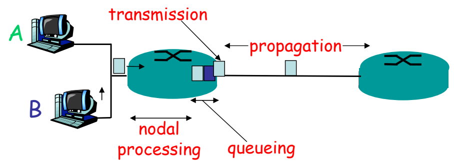
- 1. nodal processing:
  - check bit errors
  - determine output link
- 2. queueing
  - time waiting at output link for transmission
  - depends on congestion level of router



Delay in packet-switched networks

3. Transmission delay:
- R=link bandwidth (bps)
- 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)
- propagation delay = d/s

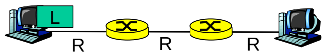


Nodal delay

- d_{proc} = processing delay
  - typically a few microsecs or less
- d_{queue} = queuing delay
  - depends on congestion
- d_{trans} = transmission delay
  - = L/R, significant for low-speed links
- d_{prop} = propagation delay
  - a few microsecs to hundreds of msecs

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

Packet-switching: store-and-forward



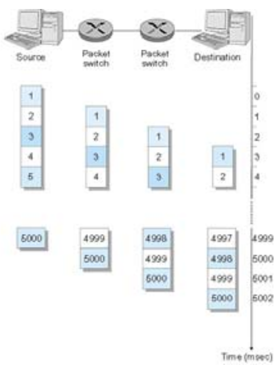
- Takes L/R seconds to transmit (push out) packet of L bits on to link or R bps
- Entire packet must arrive at router before it can be transmitted on next link: store and forward
- delay = 3L/R

Example:

Circuit Switching:
- L = 7.5 Mbits
- R = 1.5 Mbps
- Transmission delay = 5 sec

Packet Switching:
- L = 7.5 Mbits
- R = 1.5 Mbps
- Transmission delay = 15 sec

Packet Switching: Message Segmenting

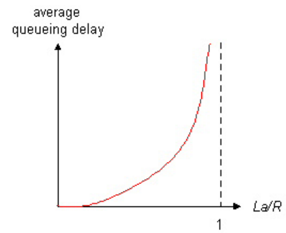


Now break up the message into 5000 packets
- Each packet 1,500 bits
- 1 msec to transmit packet on one link
- *pipelining:* each link works in parallel
- Delay reduced from 15 sec to 5.002 sec (as good as circuit switched)
- What did we achieve over circuit switching?
- Drawbacks (of packet vs. Message)

Queueing delay (revisited)

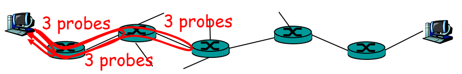
- R=link bandwidth (bit/s)
- L=packet length (bit)
- a=average packet arrival rate

traffic intensity = L·a/R
- L·a/R ~ 0: average queueing delay small
- L·a/R -> 1: delays become large
- L·a/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

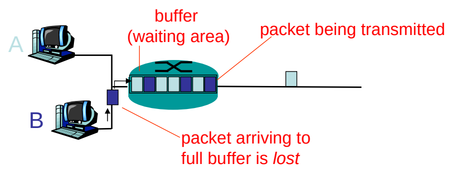
```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18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

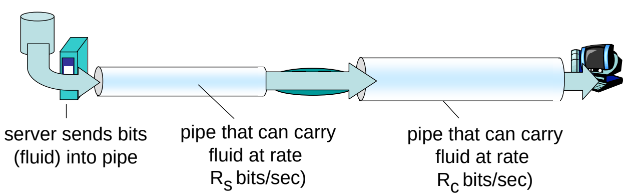
Packet loss

- queue (aka buffer) preceding link 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



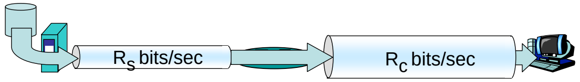
Throughput

- *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

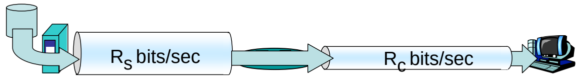


Throughput (more)

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



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



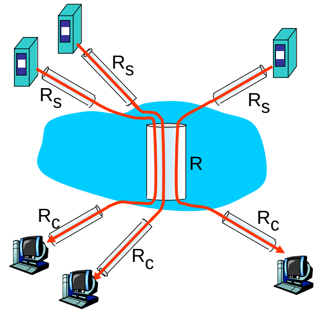
bottleneck link

link on end-end path that constrains end-end throughput

⇨ measurement challenge for networks with many nodes: identify bottleneck interfaces, e.g. with packet-pair measurements

Throughput: Internet scenario

- Example: 10 clients / servers share a bottleneck link
  - 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

Chapter 1: roadmap

1.1 What *is* the Internet?
1.2 Network edge
- end systems, access networks, links

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

1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models
1.6 Networks under attack: security
1.7 History

Protocol “Layers”

Networks are complex!

- 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?

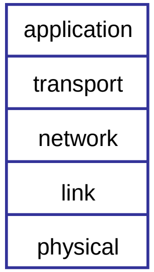
Why layering?

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
  - PPP, Ethernet
- physical: bits "on the wire"



Introduction: Summary

Covered a lot 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

You now have:

- context, overview, "feel" of networking
- more depth, detail *to follow!*