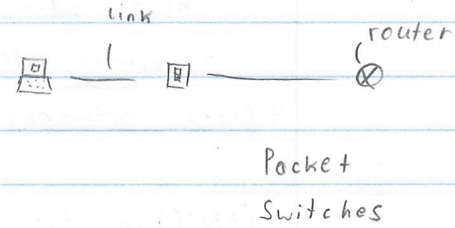
# Computer Network Protocols (Lecture 1)

## What's the internet: "nuts and bolts" view

- hosts = end systems
- running network apps




- Communication Links
  - Fiber, copper, radio, satellite
  - transmission rate: Bandwidth
- Packet Switches: forward packets (chuncks of data)
  - Routers and switches

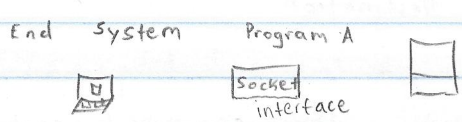
- Protocols define format, order of messages sent and recieved among network entities, and actions taken on message transmission, reciept

## What's a protocol

- Human Protocols
  - What's the time
  - "I have a question"
  - Introductions
- Network Protocols
  - Machines rather than humans
  - All comunications activitity in the internet

## What's the internet: Service View

- Infrastructure that provides service to applications:
  - Web, VOIP, email, games, e-commerce
- Provides programming interface to apps
  - Hooks that allow sending and recieving app programs to "connect" to the internet
  - Provides service options analgous to postal service

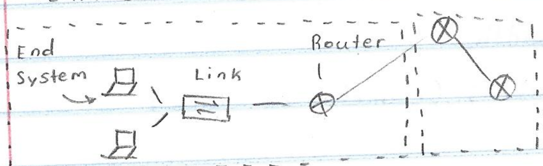



- In order to send data you must follow some loop (defined by socket interface

## A closer look at network Structure

- Network Edge
  - hosts: clients and servers
  - Servers often in data centers
- Access networks, physical media: wired wireless

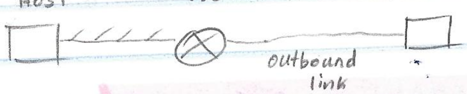
LAN (Local Area Net)




Access Network Network Core

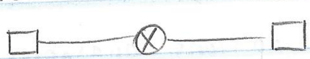
## The network Core

- Mesh of interconnected routers
- Packet - Switching: hosts break applications - layer messages into packets
  - Each packet transmitted at full link capacity
  - Forward packets from one router to the next, across links on path from source to destination




- Host sends all the information to router, once all the information is gathered, it then sends to outbound link to the destination

Example: 3 Packets



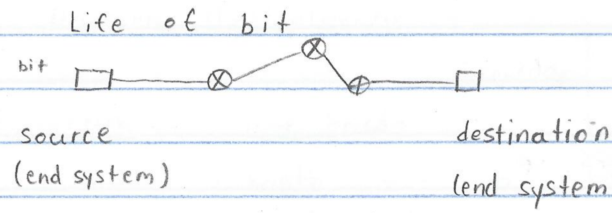

| | | Source | Router | Destination |
|---|---|---|---|---|
| 1 | At Time 0: | 1 | | |
| 2 | At Time $L/R$: | 2 | 1 | |
| 3 | At Time $2L/R$: | 3 | 2 | 1 |
| | At Time $3L/R$: | | | 2 |
| 4 | At Time $4L/R$: | | | 3 |

## Host: Sends Packet of Data

- Host sending function
  - Takes application message
  - Breaks into smaller chunks known as packets of length L bits
  - Transmits packet code into access networks at transmission

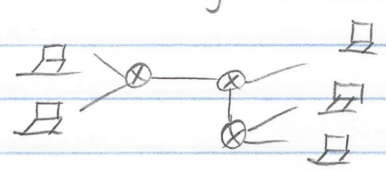
## Physical Media

- Bit: Propogates between transmitter / reciever pairs; sent by propogating electromagnetic waves or pulses across a physical medium




- Physical link: What lies between a transmitter & Reciever
- Guided Media: signal propogate in solid media: Copper, fiber, coax
- Unguided Media: Signals propogate freely (Radio)

## Packet Switching: Queueing, Delay, Loss

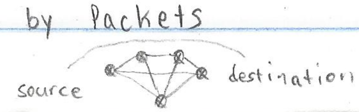


- Queuing and Loss

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

## TWO Key Network-Core Functions

Routing: determines source - destination route taken by Packets




Forwarding: Move Packets from router's input to appropriate router output

# Network Protocols (Lecture 2)

## Network Resources

| | |
|---|---|
| Packet switching | not reserved (sharing) |
| Circuit switching | reserved |

| Connections | Link Capacity |
|---|---|
| one | full |
| multiple | Partial |

## Circuit switching: FDM versus TDM

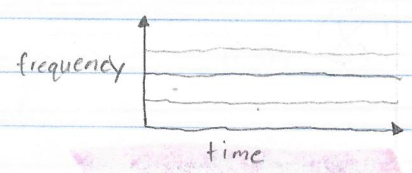
FDM - Frequency - Division Multiplexity

TDM - Time - Division Multiplexity

Example:

4 users

FDM



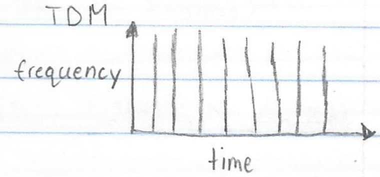

Frequency Spectrum

Four bands

Width

Bandwidth

TDM




## Packet switching versus Circuit switching

Example:

- 1 mb/s link
- Each user
  - 100 kb/s when "active"
- Circuit - Switching
  - 10 users
- Packet Switching
  - "The chance of >10 users is very little 0.0004"
  - Full link capacity
  - With 35 users, probability > 10 active at same time is less than 0.0004 *

Example:

1 user → 100 kb/s

1 MbPS

1 Mb/s

100 kb/s $\Rightarrow \frac{10^6}{10 \cdot 10^3} = \frac{10^6}{10^5} = 10$

## Packet switching vs Circuit switching

▫ Is packet switching a "slam dunk winner"

- Great for bursty data
  - resource sharing
  - Simpler, no call setup
- Excessive congestion possible: packet delay

Packet switch Problem:

A. $\frac{2\ Mb/s}{1\ Mb/s} = 2$

B. 2 < No queu delay
   2 > queu delay

C. 20%

D. $0.2 * 0.2 * 0.2 = 0.08$

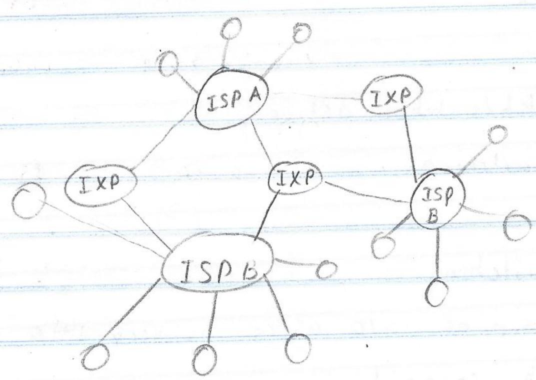
## Internet Structure

How do you connect Millions of ISP?

- Connect to all ISP's

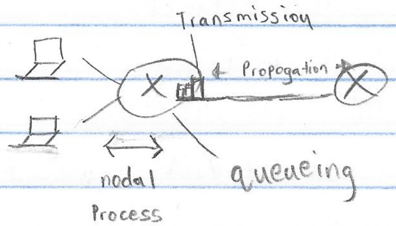
Network Core Edge - (Access network)

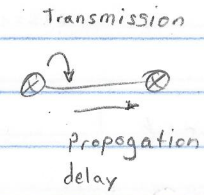
▫ IXP - Internet Exchange Point



- Packets que in router buffer
  - Packet arrival rate to link (temporarly) exceeds

## Four sources of Packet Delay

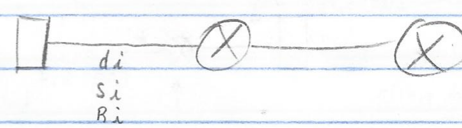



## Homework

Packet Size: L

Diagram




Delays:

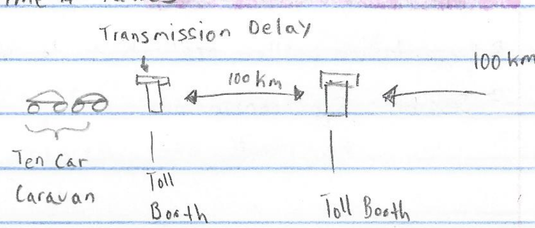
link
- Transmission Delay: $\frac{L}{R_1} + \frac{L}{R_2} + \frac{L}{R_3}$
- Propogation Delay: $\frac{d_1}{S_1} + \frac{d_2}{S_2} + \frac{d_3}{S_3}$

Switch
- Processing Delay: $d_{proc} + d_{proc}$
- No queueing Delay: 0

## Comparing Transmission and Propogation Delay

- 
- The propogation delay is the time it takes

## Caravan Analogy




Toll booth → Router

Segment → Link

Travel → Propogate

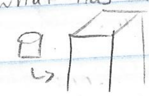
Ten Cars → Packet

One Car → bit

100 km/hr : 60 sec · 60 min

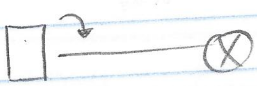
## ThroughPut

- $R_s < R_c$ What has the better . .

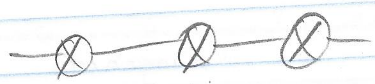


## Homework 1

1\. A.

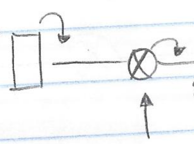


B.



5000 bits

C.



200 bit (Transmission delay reduced)

Cut Through

4,800 bits = 0

2\. g. min (bandwidth - delay product ; Packet size)

h. $\frac{\text{Length of link}}{\text{bandwidth-delay product}}$

## Layering of airline functionality

- Interent Protocol Stack (5 of them)

## ISO/OSI reference model

- Presentation : Allow applications to interpret meaning of data
- Session : Synchronization

A pair of programs

Client Program — executed ↓ — Client Prosses

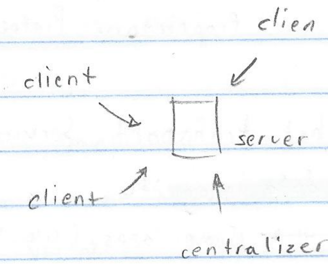
Server Program — ↓ executed — Server Process

socket interface

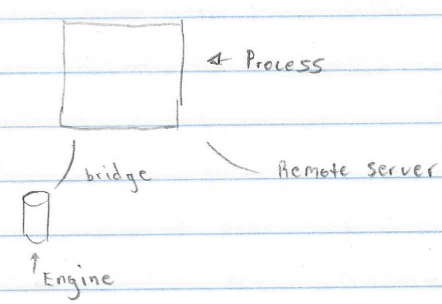
Network

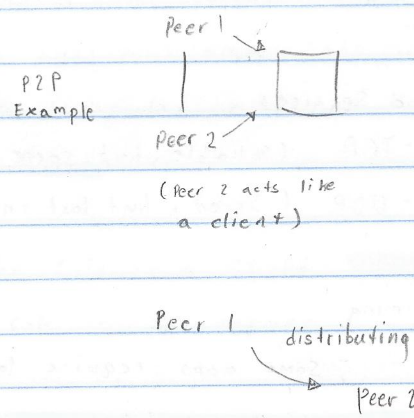
## Client Server Architecture

- Server
  - Always - on host
  - Permanent IP Address data centers for scalling
- Clients
  - Communicate with server
  - may be intermittently connected
  - May have dynamic IP Addresses




*important* Process : program running within a host




* Sockets best for changing messages
- Socket is between application process and transport

Addressing Process

- To recieve messages process must have identifier
- Q: does IP Adress of host on which process runs Suffice for identifying the process?

  A: Multiple Ports (Answer is NO); Use port numbers to identify a specific process

App - Layer Protocol Defines

- Type of messages exchanged
- message syntax
- message semantics
- rules
- Two types of Protocols
  - Open Protocols
  - Proprietary Protocols

What transport Service does an app need

- data integrity
  - Some apps (file transfer, etc.) require 100% reliable data transfer
  - Other apps can tolerate some loss (audio)

- Find Services
  - TCP (Reliable, but some delay)
  - UDP (Speed, but lost in data)

- Timing
  - Some apps require low delay to be effective
  - (internet telephony, interaactive games

- Throughput
  - Some apps require minimum amount of throughput to be effective
  - otherapps make use of whatever they get

- Security
  - Encryption, data integrity

Internet Transfer Protocols Services

TCP Service:

- reliable transport
- flow control
- Congestion Control
- Does not provide: timing, minimum throughput guarantee security
- Connection-Oriented

## P2P Architecture

- There is no minimal (or no) reliance on dedicated servers in data centers
- Exploits direct communication between pairs of hosts called peers
- P2P architectures have self-scalability
- Three major Challenges:
  - ISP Friendly: dimensioned for "assymmetrical" bandwidth usage
  - Security: highly distributed and open nature
  - Incentives: Convincing users to volunteer bandwidth
- Processes are used to communicate (A program running at the end system)
- Processes on two different end systems communicate with each other by exchanging messages across a computer network.
- There is a client and the Server
- P2P file sharing
  - The peer downloading the file is a Client.
  - The peer uploading the file is the server

## Interface Between the process and the computer Network

- A process send messages into and recieves messages from, the network through a software interface called a socket.
- Application Programming Interface (API) between the application and the network

## Addressing Process

- Two pieces of information needed
  1. Address of the host
  2. An identifier that specifies the recieving process in the destination host
- Port Number: A port number is a way to specify process to which an internet or other network messages is to be forwarded (Port 80 : Web server ; Port 25 : Mail Server

## Reliable Data Transfer

- Reliable Data Transfer, if a protocol provides such a delivery data transfer
- Loss-tolerant applications, when a transport-layer protocol doesn't provide reliable data transfer, some of the data sent by the process may never arrive at the recieving process

## Throughput

- Throughput, text of communication session between two processes along a network path
  - Requirement: Bandwidth-Sensitive applications
- Elastic Applications, can make use of as much or as little throughput as happens to be available

## Timing

- Transport-layer protocol can also provide timing guarentee
- Long delays in Internet telephony often resualt to unnatural pauses

## Transport Services Provided by the internet

- TCP:
  - TCP services model includes a connection-oriented service and a reliable data transfer service
    - Connection Oriented: TCP has the client and server exchange transport-layer control information with each other before the application-level messages begin to flow
      - Process:

        Exchange information → Handshaking Process (alerts the client and server, in order to prepare for onslaught of packets) → TCP Connection exists between sockets of the two processes → Connection is Complete
    - Reliable data transfer Services: All data is sent without error and in the proper order
    - Also contains congenstion-control mechanism

- UDP
  - It is connectionless, so there is no handshaking before the two processes start to communicate
  - Provides an unreliable data transfer service
  - Provides no guarantee that the message will reach the recieving process
  - May arrive out of order
  - Does not include a congestion-control mechanism (can pump at any rate)

## Overview of HTTP

- Hyper Text Transfer Protocols
  - Contains two programs: Client Program & Server Program
  - Both programs execute on different end systems
  - HTTP uses TCP as it's underlying transport protocol
  - Server sends requested files to clients without storing any state of information (HTTP is said to be a stateless protocol)

## File Transfer: FTP

- User is sitting in front of the host and wants to transfer to and from remote desktop
- FTP uses two Parallel TCP connections to transfer a file, a controll connection and a data connection

- Differences:
  - FTP is said to send its control information out of band
  - HTTP is said to send its information out of bound

Lecture 4

- Peer acts as both client and server

## Start The Program

- Name your program as "chat"
- Specify the port number that the process is running on

ifconfig - Linux command

ipconfig - Linux command

## Code Structure:

Select ( ) - descripte

STDIN - input from keyboard is ready to read

Socketcount

if (socketcount > 0)

for (int i = 0; i <

Select ( ) - multiple clients

HTTP Connections:

* non-Pesistant · HTTP
  - A most on object sent over TCP Connection
    · A connection then closed
  - Downloading multiple objects required multiple connections

Parallel TCP Connection

Parallel Propogation

* Alway 1 TCP Connection
* 1 RTT for all referenced object

Lecture 5

## HTTP request message

Connection: keep-Alive

will time out in a certain amount of time

- Last-modified
  - When the object was created or modified

400 Bad Request

404 Not Found

1.5 Mbps · 100k → 1.5 Mbps · $10^5$

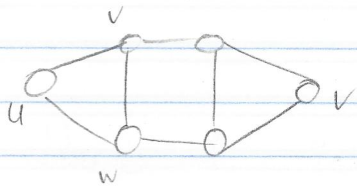
- SMTP needs 7 bit ASCII Pull Places all of the messages into one message
- HTTP uses 8 bit ASCII Push

$\frac{NF}{U_s}$ $\frac{NF}{U_s + E_n}$

Server P2P

October 31, 2017 Lecture

Topology




$D(v) = C(u,v) = \infty$

$D(v) = D(w) + C(w,v)$

Node X has 2 neighbors

$c(x,v) + d_v(y)$

$C(X,W)$

| | Y | Z | W |
|---|---|---|---|
| X | 2 | 1 | 3 |

Distance Vector

Each row distance vector

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 2 | 7 |
| Y | | | |
| Z | | | |

| | X | Y | Z |
|---|---|---|---|
| X | 0 | | |
| Y | 2 | 0 | 1 |
| Z | 7 | 1 | 0 |

Homework3: Node X

*Final Network Core - A mesh of routers

November 7 2017 Lecture # ?

- SDN can determine the routing

0 00000000 — 0011 1111 $2^6 = 64$
1 01000000 — 0101 1111 $2^5 = 32$
2 0110 0000 — 0111 1111 $2^5 = 32$ +
0100 0000 — 0100 0000 $2^6 : 64$ +
3. 1100 0000 — 1111 1111 ↳ 96
$2^6 = 64$

↳ * Final