

Internet Technologies (COMP90007)

Assignment No. 1

Semester : August 2020 (Semester 2)

Student Id 1124298

Student Name - Sakshi Chandel

Question 1:

Solution :

Since its given ,video clip – 10 sec , 30 frame per sec, 1280 X 720 pixels per frame which needs 3 bytes per pixel

$$\begin{aligned} \text{Bits in 10 frame} &= 1280 \times 720 \times 24 \quad (\text{1 byte} = 8 \text{ bits}) \\ &= 22118400 \text{ bits} \end{aligned}$$

$$\begin{aligned} \text{Bits in 30 frames} &= 22118400 \times 30 \\ &= 663552000 \end{aligned}$$

$$\begin{aligned} \text{Bits in 10 second video clip} &= 663552000 \times 10 \\ &= 6635520000 \text{ bits} \end{aligned}$$

Latency = Time Delay + Propagation delay = Message in bits /Rate of transmission + length of channel/speed of signal .

Message in bits = 6635520000 bits

Transmission rate = 56kbps

Length of the channel = 10000000 m (distance between sender & receiver)

Speed of signal = 200000000 m/s

Part1)

$$\text{Latency} = 6635520 \text{ kbits} / 56 \text{ kbps} + 10000000 \text{ m} / (200000000 \text{ m/s}) = 118491.428 \text{ sec} + 0.05 \text{ sec}$$

Answer = **118491.478571 seconds**

Part 2)

$$\text{Latency} = 6635.520 \text{ Mbps} / 100 \text{ Mbps} + 10000000 \text{ m} / (200000000 \text{ m/s})$$

Answer = **66.4052 seconds**

Question 2 :

Solution 2:

Given here,

Channel bandwidth = 8 KHz

Max data rate = 128 kbps

Part 1) Channel is noisy,

According to Shannon's theorem relation between Max data rate with bandwidth and Sound to Noise ratio (S/N) is given as (channel is noisy) :

$$\text{Max. data rate} = B \times \log_2 (1 + S/N) \text{ bits/sec}$$

$$128 = 8 \times \log_2 (1 + S/N)$$

$$16 = \log_2 (1 + S/N)$$

$$S/N = 2^{16} - 1$$

$$S/N (\text{dB}) = 10 \times \log_{10} (S/N)$$

$$S/N (\text{dB}) = 10 \times \log_{10} (2^{16} - 1)$$

$$S/N (\text{dB}) = 10 \times \log_{10} (65535) \quad (2^{16} = 65536)$$

$$S/N (\text{dB}) = 10 \times 4.816 \quad (\log 65535 \text{ base } 10 = 4.816)$$

Answer : S/N (dB) = 48.16 dB

Part 2)

Channel is noiseless,

According to Nyquist's theorem , relation between Max data rate with bandwidth and signal level is given as (channel is noiseless) :

$$\text{Max. data rate} = 2 \times B \times \log_2 (V) \text{ bits/sec}$$

$$128 = 2 \times 8 \times \log_2 (V)$$

$$8 = \log_2 (V)$$

$$V = 2^8$$

Answer : We need a signal which has signal level of 256. That means 8 bits per pulse or sample .

Question 3:

Solution 3:

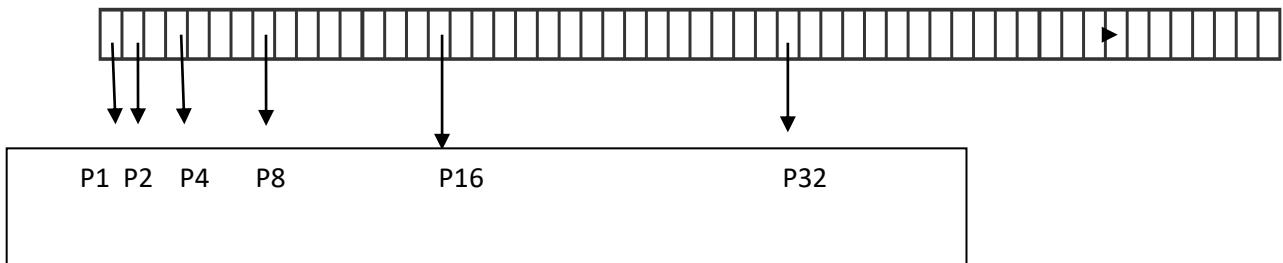
As given we are using Hamming code to correct errors. According to formula to calculate check bits ,

$$2^c \geq d+c-1 \text{ where } c = \text{check bits}, d = \text{data bits}$$

Given data bits d= 48

Answer : Minimum check bits required = 6

The redundant bits are placed at positions corresponding to power of 2- **1, 2, 4, 8, 16 and 32**



Question 4:

Solution 4 :

Benefits of layered structure in network are :

- 1) It prevents other layers to get affected from changes in one layer .
- 2) It allows different software and hardware to communicate with each other .

Question 5 :

Solution 5 :

Part (1):

Answer : I searched for site (<http://vornlocher.de/tower.html>)

The IP address of the source is **192.168.1.4**

The IP address of the destination is **88.217.240.50** (<http://vornlocher.de>)

To check confirm the IP address of the source one can run command “**ipconfig**” in cmd (Command Prompt) of the system.

To check confirm the IP address one can open the browser and paste the destination IP address and check .

Answer 2: Flow graph

The image below shows about the tcp stream when the connection is established between source and destination .The two service primitives that the graph depicts is “CONNECT” and “ACCEPT”.The graph shows how the source is sending the connection request as SYN and the destination is accepting the connection and sending back SYN ACK as an acknowledgement.

The starting green patch indicates the Connection and acceptance of the network between source and destination.

Time	192.168.1.4	vormlocher.vem-online.net	relay-a4ad563a.net.anydesk.com	Comment
0.000000	51538	SYN	80	Seq = 0
0.143247	51538	SYN, ACK	80	Seq = 0 Ack = 1
0.143325	51538	ACK	80	Seq = 1 Ack = 1
0.143842	51538	PSH, ACK - Len: 551	80	Seq = 1 Ack = 1
0.289006	51538	ACK	80	Seq = 1 Ack = 552
0.289848	51538	PSH, ACK - Len: 211	80	Seq = 1 Ack = 552
0.331267	51538	ACK	80	Seq = 552 Ack = 212
1.290820	51538	FIN, ACK	80	Seq = 212 Ack = 552
1.290904	51538	ACK	80	Seq = 552 Ack = 213
6.770808	51527	ACK	80	Seq = 1 Ack = 1
6.771047	51527	ACK	80	Seq = 1 Ack = 2
16.946097	51527	ACK	80	Seq = 1 Ack = 1
16.946177	51527	ACK	80	Seq = 1 Ack = 2
23.896550	51538	FIN, ACK	80	Seq = 552 Ack = 213
23.897741	51541	SYN	80	Seq = 0

Submitted by :

Name: Sakshi Chandel

Student Id : 1124298

COMP90007 Internet Technologies

Semester 2, 2020

Assignment 1

Due date: September 4th Friday 4:00 pm (GMT+10)

This assignment is worth 5% of the total marks for the subject. This assignment has 5 questions. The weighting of each question is shown beside the question. **Answers must be submitted as a PDF file via the COMP90007 Assignment 1 submission link in Canvas by the due date. Late submissions will attract a penalty of 10% per day (or part thereof).**

Please ensure your name and student ID are clearly presented on your submission. **Submission should only contain the question number and the answer (do not repeat the text of questions in your submission).** Please present all steps of the solutions for questions involving calculations and/or derivations, otherwise relevant penalties will be applied. Questions can be answered in a few sentences. Excessively long answers will not be accepted. Please type your answers and save as PDF. **Handwritten assignments that are scanned will not be accepted.**

All questions can be answered by studying the material covered. **All work presented should be your original individual effort/work.**

Question 1 (1 point)

Given a 10-second video clip with frame rate 30 fps (frames per second), each frame contains 1280 x 720 pixels, which needs 3 bytes/pixel. Assume the video clip is stored as an uncompressed simple video file. If the distance between the sender and the receiver is 10,000 km, what is the latency to send this video clip 1) over a 56kbps simple modem? 2) over a 100Mbps broadband link? (Assume the speed of the signal for both cases is 200,000 km/second).

Question 2 (1 point)

Given a channel with 8 kHz bandwidth,

- (1) if it's a noisy channel, what is the minimum signal-to-noise ratio (in dB) that can support a data rate of 128kbps?
- (2) if it's a noiseless channel, how should we send the data to support a data rate of 128kbps?

Question 3 (1 point)

We are transmitting 48-bit data, using Hamming code to correct errors. What is the minimum number of check bits needed to ensure that the receiver can correct a single-bit error? Where should we put these check bits? List the positions of all the check bits.

Question 4 (1 point)

What are the benefits of having layered structure in networks? List at least two benefits. (Please answer this question briefly in a few sentences. Excessively long answers will not be accepted.)

Question 5 (1 point)

Pick an **HTTP based website of your choice** and capture the packets exchanged using Wireshark, depicting the trace created when you access a website via a web browser/wget. (Note: Please review the steps and context that was covered in the Wireshark Lab exercise in Week 2 for this question.)

- (1) What are the IP addresses of the source (requesting the website) and destination (serving the website)? What steps should we undertake to validate that these IP addresses belong to the aforementioned source and destination and not some other server(s), etc?
- (2) We have introduced that there are six service primitives for implementing a simple connection-oriented service (as shown in **Table 1**, source: Ch 1.3.4 Tanenbaum & Wetherall, 2011). Select any two of the service primitives from **Table 1** and use the TCP stream captured in your Wireshark trace from the previous subsection, do the following - provide screenshots of your Flow graph diagram and explain what information in this graph corresponds to the two primitives you selected, and why? Please ensure that the graphs are legible in the PDF document, else relevant penalties will be applied.

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Table 1 Service Primitives

COMP90007 Internet Technologies
Semester 2 2020, Assignment 2
Due date: 19 October 2020, Monday, 5 PM (Melbourne Time)

This assignment is worth 5% of the total marks for the subject. The weighting of each question is shown beside the question. Answers must be submitted as a PDF file via the COMP90007 Assignment 2 submission link on Canvas. **As usual, late submissions will attract a penalty of 10% per day or part thereof.** Please ensure your name, username and student Id are clearly presented on the answer documents you submit on Canvas. Submissions should only contain the question number and the answer (*please do not repeat the text of questions in your submission*). **Answers should be typed and not handwritten.** Questions can be answered by studying the material covered. **All work should be your original individual effort/work.**

Question 1 (1 mark)

The shortest path routing is used on a network shown in Figure 1, with the weight of each edge marked in the label.

- (1) What are the weights of shortest paths from E to the other 5 nodes, respectively? Show all your calculation and the steps of using Dijkstra's algorithm (Please use a table like the one shown on Slide 50 of Network Layer).
- (2) Which links are in the network graph but not in the sink tree of node E?

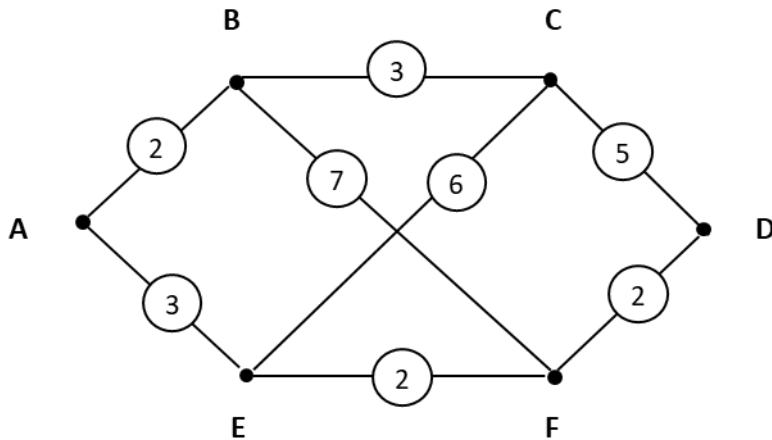


Figure 1 Network with nodes A-F

Question 2: (1 mark)

We have studied that four parameters define quality of service: Bandwidth, Delay, Jitter, and loss. The requirements for these parameters vary among applications and the quality of service perceived by users. Some applications require that high priority to be given to a parameter while other applications may require low priority for the same parameter. In the table below, please write 'High', 'Medium' or 'Low' against each parameter for all applications. Note, High means that a particular application require High need for quality for that parameter. For example, High in the Delay parameter column would mean that the particular application has high priority for delay to be minimized.

Application	Bandwidth	Delay	Jitter	Loss
ZOOM Meeting				
Online Shopping				
VoIP				
Bank Transactions				

Question 3: (1 mark)

Suppose an end-to-end network connection is using the TCP protocol on ethernet service. What is the maximum data size from an application that can be transmitted in one TCP segment, including overheads from TCP and IPv4 but excluding the overhead(s) imposed by the data link layer? How will your answer change if the transport layer protocol is UDP instead of TCP?

Question 4: (1 mark)

Congestion and buffer windows are used at sender and receiver ends to avoid network issues in TCP transport protocol. Describe:

- 1) What are the roles of these windows on sender and receiver sides?
- 2) What are the implications of using large vs. small window sizes on sender and receiver sides?

Question 5: (1 mark)

Suppose that the TCP Tahoe congestion window was at 39 KB when the time out occurred. How big will the window size be if the next four transmission bursts are all successful and threshold was set at 20 KB?

COMP90007 Internet Technologies

Network Analysis Project, Project 1

Semester 2, 2020

Due Date: Wednesday September 23, 4:00pm

1 Introduction

This project forms 10% of your final mark. The key output of this project is a report which has to follow a certain format (refer to Section 5 of this document). The project is about measuring bandwidth, delay, and jitter in networks. These tasks will be similar to those you have performed in the laboratory sessions held in the tutorials. It is recommended that you perform these tasks in a consistent networking environment to reduce the variance in your report.

Important Note: As evidence of your work, when you run the following commands, please remember to take screenshots of the results obtained and place it in the appendix of your report. Reports failing to do so will be penalized. All the plots needed to answer questions should be placed in the main body of your report where you will explain the observations being derived.

2 Measuring the hop count

In this section, we will be observing the number of intermediate hosts in the route taken to communicate with a remote server and its relation to the physical geographical distance.

To count the number of hops taken to reach a destination host, the command `tracert` will be used (or its corresponding equivalent, depending on your operating system). This utility should be pre-installed on your operating system.

The utility can be invoked by launching a command line terminal and typing in the command. An example output of the `traceroute` command (on OS X) and `tracert` command (on Windows) is as follows:

```
$ traceroute -nwl cis.unimelb.edu.au
traceroute to cis.unimelb.edu.au (128.250.37.164), 64 hops max, 52
byte packets
 1  10.0.0.254  533.676 ms  1.063 ms  0.940 ms
 2  58.96.2.205  27.872 ms  28.137 ms  28.293 ms
 3  58.96.2.129  28.647 ms  28.577 ms  28.085 ms
 4  218.100.78.33  28.299 ms  28.469 ms  28.332 ms
 5  202.158.200.9  29.626 ms  28.871 ms  29.841 ms
 6  202.158.210.26  31.320 ms  28.722 ms  29.135 ms
 7  202.158.200.250  29.668 ms  29.096 ms  28.660 ms
 8  *  *  *
 9  *  *  *
10  *  *  *
11  128.250.37.130  957.521 ms  33.475 ms  29.891 ms
12  128.250.37.164  29.940 ms  29.260 ms  30.020 ms
```

In this section of the project, you are interested in the number of hops it takes to reach the destination server. In the example above, the number of hops to reach `cis.unimelb.edu.au` is 12.

Based on the number of measurements you will be taking, there are some useful command line parameters you may wish to take advantage of, to speed up the time it takes to gather

results. The help documentation for the `traceroute` utility can be accessed by running `man traceroute` or `tracert /?` on Windows. You may also wish to investigate shell scripting to automate the collection of results, but this is not required for the project. Any scripts (Shell, Python, etc) you do choose to write, however, **must be included in the Appendix**.

2.1 Specific task description (2 marks)

Please include all raw measurements in the Appendix.

- 2.1 What do the command line parameters `-n w 1` (equivalently `-d -w 1` on Windows) mean in the example given above and what is the importance in using them?
- 2.2 Determine the hop count for the following hosts given in Table 1. It is recommended that students find one more public iperf server other than the ones listed here to gather their results.

Table 1: List of public iperf hosts

Host	Location
iperf.he.net	USA
bouygues.testdebit.info	France
iperf.comneonext.de	Germany
ikoula.testdebit.info	France
st2.nn.ertelecom.ru	Russia
iperf.biznetnetworks.com	Indonesia
iperf.scottlinux.com	USA
speedtest.serverius.net	Netherlands
iperf.volia.net	Ukraine

Determine the approximate geographical distance for the above hosts and plot the hop count versus the approximate geographical distance from the city you are currently in. Do you observe a correlation or not? Please explain your rationale with respect to networking concepts.

You may use any scientific computing package or spreadsheet software to do your plotting, **for example, Microsoft Excel**.

For finding out the physical geographical distance you may use any tool or application available online, for example, you may use a combination of: <https://db-ip.com> and <https://www.freemaptools.com/how-far-is-it-between.htm> or <https://www.site24x7.com/find-website-location.html> and <https://www.distancecalculator.net/> or anything of your choice. However, do make sure to **document it and provide the appropriate reference** to that application/ tool/ software used.

Note: The servers listed in Table 1 are public servers and are not maintained by the University of Melbourne, hence they are likely to go down at any point in time. Based on past experience, it would be advisable to **conduct your tests on these as soon as possible (ASAP) rather than leave it till the end** as there is a high probability that these servers might not be available and this cannot be used as an excuse for a late submission. If these servers stop responding then please visit the link: <https://iperf.fr/iperf-servers.php> and find your own servers (anything that is responding) or feel free to find any public iperf server from the internet. Some alternate strategies worth exploring also include changing port numbers and trying to get the iperf metrics.

3 Measuring delay and jitter

In this section, you will be measuring the delay and jitter of the hosts used in Section 2, located in different geographical locations.

We will be using the `ping` utility, to measure the round-trip delay of packets. The `ping` utility should be pre-installed on all major operating systems. The standard deviation of the round-trip delay time will be taken as the value for *jitter* for this project.

The standard deviation measures the variation in a set of data. It is defined as the square root of the variance and is expressed as follows:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

where \bar{x} is the mean of the set of data. Details of this simple statistical measure can be found in many sites online if you do not remember this from high school years.

A sample output of the `ping` utility is shown below, but this output will vary depending on your operating system.

```
$ ping unimelb.edu.au
PING unimelb.edu.au (172.22.44.10): 56 data bytes
64 bytes from 172.22.44.10: icmp_seq=0 ttl=124 time=3.364 ms
64 bytes from 172.22.44.10: icmp_seq=1 ttl=124 time=3.416 ms
64 bytes from 172.22.44.10: icmp_seq=2 ttl=124 time=3.730 ms
^C
--- unimelb.edu.au ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.364/3.503/3.730/0.162 ms
```

In the output above, various statistics, including the mean and standard deviation, were calculated for you by the utility. It is expected that you record all the values of delay and perform your own calculations to find the mean values and their standard deviation, to confirm the result.

3.1 Specific task description (2 marks)

For this section, you may want to consider the user-facing implications of high delay and high jitter in networking applications, especially for applications sensitive to the affect of high delay and high jitter. Please include all raw measurements in the Appendix.

- 3.1 Measure the round-trip delay for the following hosts. Make **three** delay measurements (run this command 3 times - not 1 command gathering 9-10 rows of ICMP responses) of each host and find the average round-trip delay and jitter by calculating the standard deviation, for all the hosts used in Section 2.

For each of the above hosts, plot the average round-trip delay versus the approximate physical geographical distance to the server. Do the same with the jitter (i.e. jitter vs geo distance).

- 3.2 From the two plots above, do you observe any correlation between delay and jitter as a function of distance? Why? Why not? Explain your results with reference to the network environment in which you were collecting your results (this includes metrics like your download/ upload speed, users sharing the network, load on network through other apps, etc) and how does your networking environment influence your results obtained (examples required)?

4 Measuring the bandwidth-delay product

In this section, we will be measuring the bandwidth of different hosts in order to determine the bandwidth-delay product, using the results from the previous sections.

The utility that will be used to perform bandwidth measurements will be the `iperf` utility. This command line utility is available for download for all operating systems from <https://iperf.fr>. Alternatively you may choose to use the package manager for your relevant operating system.

There are two modes of operation in `iperf`. The server mode will host a server which will listen to incoming requests from a client. An `iperf` instance running in client mode will connect to the server, and packets will be exchanged and timed between the two hosts to calculate the bandwidth. In this project, we will be running `iperf` in client mode.

A sample output of `iperf` in client mode is shown below, noting the `-c` flag to designate operating in client mode.

```
$ iperf -c iperf.eenet.ee
-----
Client connecting to iperf.eenet.ee, TCP port 5001
TCP window size: 129 KByte (default)
-----
[ 5] local 10.0.0.5 port 51878 connected with 193.40.100.7 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 5]  0.0-106.7 sec   128 KBytes  9.83 Kbits/sec
```

(Note: speedtest.serverius.net server) For speedtest.serverius.net in Table 1, we may need to use the port 5002, so the command can be: “`iperf3 -c speedtest.serverius.net -p 5002`” for this host. Also, **some iperf servers respond to iperf2, rest to iperf3** so please try to use both `iperf2` and `iperf3` to verify if the server is responsive to either version. If you have response from both `iperf2` and `iperf3` then you can pick `iperf2` amongst them.

4.1 Specific task description (6 marks)

Please include all raw measurements in the Appendix as usual.

- 4.1 What does the bandwidth-delay product tell us about the networks? Collect **three** set of measurements (run this command 3 times) measuring the bandwidth of the public `iperf` hosts in Section 2 and find the mean bandwidth for each host.
- 4.2 Take the mean bandwidth and calculate the bandwidth-delay product in kilobits. You may use the mean round-trip delay time from your `ping` experiments to use as the delay time. Plot a bar chart for each host showing your results. You may wish to use a logarithmic scale, if appropriate.
Explain your results making reference to your networking environment in which you performed your measurements. How do your results reflect upon your actual internet link speed and how does your network environment influence your results obtained (provide examples)? Are there outliers in your data? If yes, point out the outliers and explain why they are marked as outliers in your data?
- 4.3 Plot the bandwidth-delay product versus the hop count. Do you observe any correlation?
- 4.4 When running your tests for bandwidth, delay, and jitter, were there any variables which may have affected the accuracy or reliability of your results? How might you improve upon these (explain your rationale with examples from your experience)?

5 Project Administration

This project is to be performed individually and is worth 10% of your overall mark in the subject.

5.1 Getting help

If you have any questions, the Canvas discussion board will be a useful resource in resolving any issues. If your concern is a personal matter, then you should email the subject coordinator.

Any answers posted by the subject coordinator or the academic staff on the Canvas discussion board will be considered as part of the project specification. Any announcements made about the assignment in the lectures will be considered part of the project specification. In addition, please keep an eye on any Canvas announcements to any changes made to the project specification.

5.2 Report submission

The deadline for the final report submission is as specified at the start of this project document.

The report will consist of all relevant discussion, graphs, data and answers from the experimentation conducted in this project. You must place the raw data as screens or copy paste them to the Appendix of the document, however, the diagrams like charts, flow diagrams (if any) and so on relevant to the discussion will be placed in the main document and not the appendix. Every diagram and/or raw measure used for a specific question must be referred to (using a designated reference scheme) for us to verify the result. All plots and figures must be appropriately labelled. Any information obtained that is not of your own work must be cited.

The report must be submitted as a PDF file via Turnitin on Canvas (will be available for submission soon). Please include your name, student Id and login user name on the top of the first page. The report is to be formatted on A4 sized paper in 10 pt text, 1.5 line spacing, single column. It is highly recommended that students use the respective formatting scheme outlined by us in the **Format Guideline Document** and here as massive deviation from it may incur relevant penalties. The report should not exceed a maximum of 10 pages (excluding appendix) else relevant penalties will apply.

Late submissions will attract a penalty of 10% per day (or part thereof). No submissions will be allowed passed 5 days after the deadline.

Internet of Things : Survey

Submitted by : Sakshi Chandel

Student ID : 1124298

Canvas handle : schandel@student.unimelb.edu.au

Outline

- I. Introduction
- II. Related Work
- III. Comparison of Key Approaches
 - (Benefits and Disadvantages)
- IV. Conclusions and Future Directions
- V. Reference



Introduction

Internet of things is been a paradigm which is gaining popularity day by day in the current scenario of modern wireless telecommunications since the term was introduced in 1990s. The basic idea of “Internet of things” is, things such as Radio-Frequency Identification (RFID) tags, sensors, actuators, mobile phones, etc. –which through unique addressing schemes, are able to interact with each other. It allows ‘people and things to be connected Anytime, Anyplace, with Anything and Anyone. Sensors and actuators are devices, which help in interacting with the physical environment. The data collected by the sensors has to be stored and processed intelligently in order to derive useful inferences from it. IoT has a basic “[Three-Layer-Architecture](#)” which includes Perception layer, Network layer and Application layer. Perception layer takes data/information from the physical environment with the help of sensors and actuators such as temperature sensors etc. Network layer is responsible for connecting to other smart things, network devices, and servers. Its features are also used for transmitting and processing sensor data. Application layer is used to display desired output from these sensors as an application in a way better understand by the end-users like deploying this application on devices which support IoT such as smartphones .Such technology will help to create ‘a better world for human beings’, where objects around us know what we like, what we want, and what we need and act accordingly without explicit instructions. In this context, assisted living (smart AC, refrigerators), e-health, enhanced learning are only a few examples of possible application scenarios in which the new paradigm will play a leading role in the near future. Similarly, from the perspective of business users there are also other domains and environments in which the IoT can play a remarkable role and improve the quality of our lives. These applications include transportation, industrial automation, and emergency response to natural and man-made disasters where human decision making is difficult. According to one of the research papers of [IEEE, published in 2019](#) saying 23 billion devices were connected to the internet in 2019 , which will stretch to 30 billion devices by 2020 .Apart from advantages ,there are possible threats as mentioned in one research paper of [DIEE, University of Cagliari, Italy](#) concerning the threat due to information security and privacy .It also mentions the fact that this threat can cause harm more than that of internet has been today.

Related Works

There are different research papers published related to Internet of Things .One of them of which is published by [IEEE in 2020](#) discussed about Multimedia of IoT: A comprehensive Survey . The article focuses on giving a detailed survey of various M-IoT network architectures. The survey also discusses the various M-IoT applications i.e., traffic monitoring, habitat monitoring, surveillance for public safety, industrial monitoring, and health monitoring. It also comprehends the design for M-IoT communication by summarizing performance metrics for M-IoT architectures. The survey also tells the M-IoT computing paradigm comprising multimedia data compression, event processing, fog/edge computing, cloud computing, and Software Defined Networks (SDNs) for data computing. It also discuss various routing protocols in the context of multimedia data delivery in M-IoT. It also provide a survey on different physical MAC (PHYMAC) protocols for M-IoT. It discussed open issues, challenges, and future research directions involving M-IoT. This article does not shows the different architectures of IoT like other surveys mentioned below. This survey is basically a complete comprehensive survey about Internet of things. This survey give detailed survey on IoT's applications .

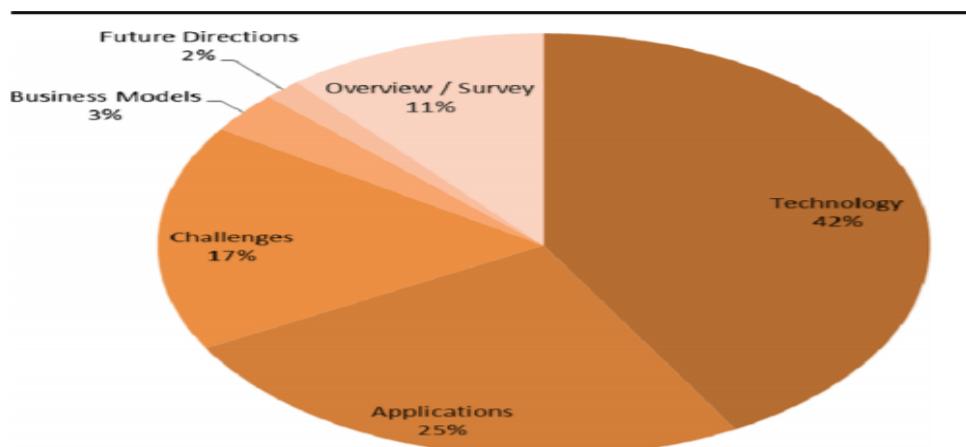
Another survey of [Hindawi as Journal of Electrical and Computer Engineering in 2017](#) discussed about different sensors of IoT like neural sensors , RFID , medical sensors , environmental and physical sensors .It also displays different layers of IoT as well like discussed in IEEE survey 2020.The survey states different applications of Internet of Things like Health care, Home Automation, smart cities. It also mentions different middlewares of IoT like OpenIoT, FiWare. It discussed about different architectures of IoT like Three-And-Five-Layer-Architecture, Cloud and fog based architecture .It relates different research papers on IoT . The survey also discusses the various M-IoT applications i.e., traffic monitoring, habitat monitoring, surveillance for public safety, industrial monitoring, and health monitoring. It also comprehends the design for IoT communication by summarizing performance metrics for IoT architectures. This survey also mentions of different applications of IoT like healthcare, and business models of IoT. This survey did not mention of different protocols used in IoT like mentioned in one survey of Ad. Hoc Network in 2015.

This survey mainly focusses about IoT architecture, protocols and applications .It also give a brief understanding and comparison of other surveys/articles about IoT.

Another survey paper by [Ad. Hoc Network published in 2015](#) discussed about Internet of Multimedia things and contributed about vision of the IoMT, whose potentialities are discussed with the help of specific use-cases. It also mentioned the distinct architectural design and characteristics of IoMT as compared to the existing multimedia systems are comprehensively discussed. The technical specifications and requirements posed by the IoMT systems are identified and discussed. The communication protocols designed for IoT are discussed and their feasibility for IoMT is analyzed. The potential multimedia processing technologies are presented that can facilitate efficient multimedia communication, specifically via wireless multimedia device. The solutions to the processing/computational issues are provided by introducing the notion of multimedia-aware cloud combined with multi-agent systems in IoMT architecture. The survey mainly focusses on the multimedia of Internet of things .This survey did not mention about different protocols of IoT but discusses about how Internet of things related to multimedia things. It focusses on how IoT can play a role in improving multimedia communications.

Another survey which was published by [Information Systems Frontiers in 2016](#) and mentioned about different classification/components of IoT like software and hardware . The survey also mentions about different applications like of healthcare sectors ,Social applications ,smart infrastructure .The survey discusses about challenges and security issues of Internet of Things. This survey mentions the facts about different hardware of IoT like RFID, sensors networks ,NFC, etc. The survey mentions the fact that how different software of IoT like middleware , searching and browsing .Different architectures of IoT are also explained ,some of them which are hardware or network architecture ,software architecture ,process architecture and general requirements . This article also compared different focusses of varieties of surveys written on IoT and it mentions the facts that about 42 percentage of IoT surveys focussed on Technology ,25 percentage of IoT surveys focussed on applications of IoT,11 percentage of IoT surveys focussed on overview/surveys of different surveys ,17 percentage of surveys focussed on challenges on Internet of things ,only 3 percentage of surveys focusses on business models, and only 2 percentage of surveys focussed on future directions.

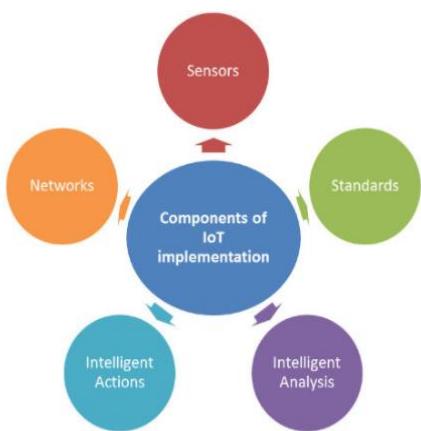
Another article which is written by [Shancang Li](#) published in 2014 ,which is a comprehensive survey about Internet of Things .This survey also mentions the different layers of Internet of Things which are namely Sensing Layer, Network Layer, Service Layer and Interface Layer. Sensing layer is integrated with available hardware objects to sense the statuses of things; & Network layer is the infrastructure to support over wireless or wired connections among things; & Service layer is to create and manage services required by users or applications; & Interfaces layer consists of the interaction methods with users or applications. This survey mainly provides an overview of the definitions, current research, standards, and future research of IoT. The survey mentions the current research on IoT system architecture is discussed. The article also discusses the enabling technologies of IoT are investigated. Last but not the least the applications of IoT are reviewed. Finally, some emerging research issues are identified and the future research directions are discussed. The survey mentions that how the lack of standards may decrease the competitiveness of IoT products .There have been many research which are going on related to IoT standards and policies .Some researches includes (1) designing policies and distributed architecture; (2) ensuring the privacy and protecting users; (3) realizing the trustiness, acceptability, and security of networks; (4) developing standards; (5) exploring new enabling technologies such as micro-electronicmechanical system. This survey also discussed various applications of IoT including Industrial applications , healthcare applications , social IoT etc Challenges are also mentioned in this survey like designing an SoA for IOT is a big challenge , in which service based things might suffer in terms of their performance .



A pie chart showing number of research articles that address/support a particular objective/feature.

Comparison of Key Approaches (Benefits & disadvantages)

There are different key approaches and benefits mentioned in various research papers. One research papers mentioned the fact the IoT can be implemented with various key approaches. One of the research papers mentioned protocols like Message Queuing Telemetry Transport (MQTT) , Zigbee, Bluetooth etc..One middle ware called FiWare used for IoT implementation which is very popular is being used in smart cities ,shop floor analytics. FiWare defines a set of SNMP APIs via which we can control the behavior of IoT devices and also configure them. Another middle ware called OpenIoT is also used which for IoT implementations . It collects data from IoT devices and also does some pre-processing of data. It has different APIs to interface with different types of physical nodes and get information from them. There are various ways by which IoT can be implemented .

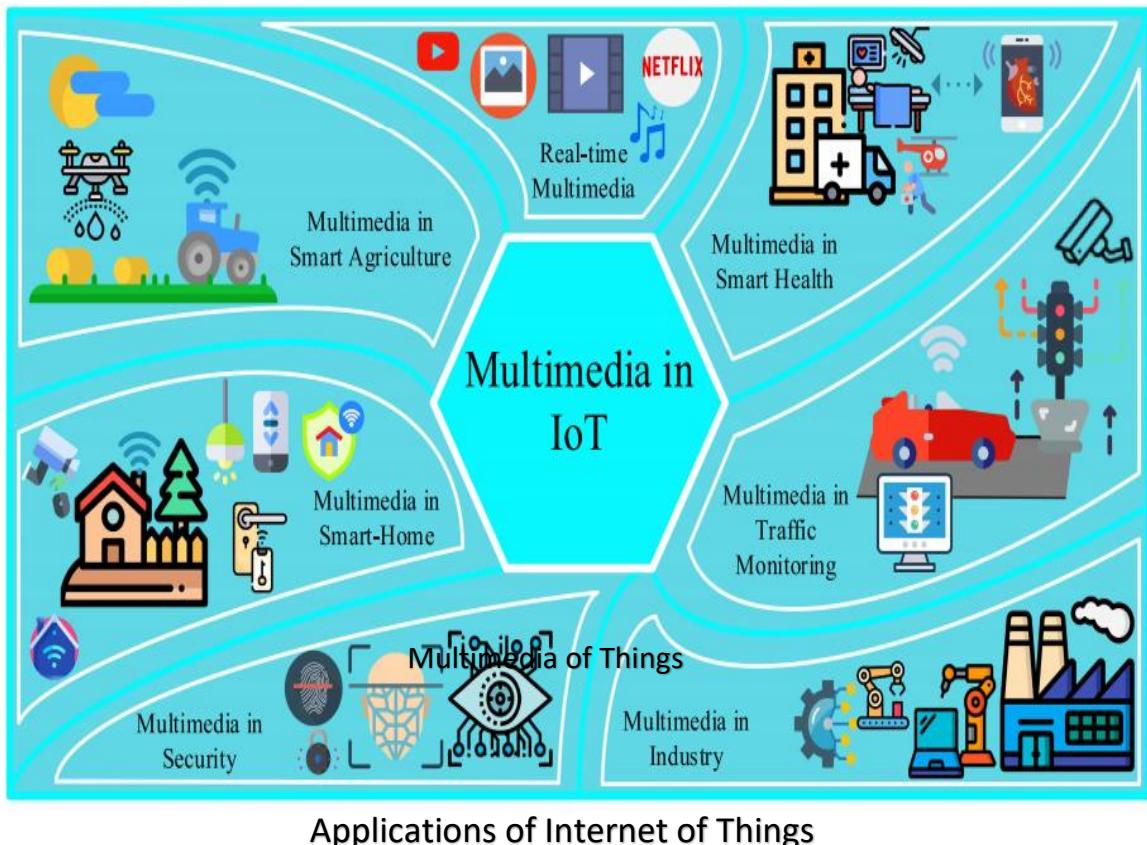
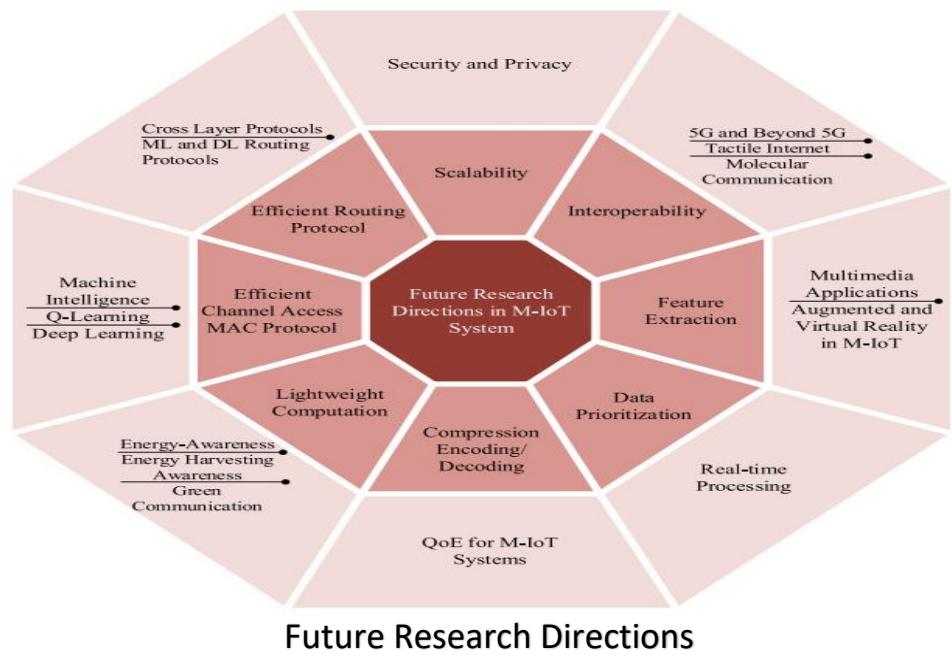


There are different components of IoT implementation including Networks, Sensors ,Intelligent Actions , Standards, Intelligent Analysis. First step of implementation includes sensors ,which according to IEEE is an electronic device that produces electrical, optical, or digital data derived from a physical condition or event.

There are many challenges faced by sensors like security , interoperability. The second implementation includes Network which is responsible for transmitting signals collected from sensors using different internet technologies including Wi-Fi, LTE ,Bluetooth. There are different challenges faced in this stage including security, power consumption etc. The third stage of implementation mentions about standards of IoT. There are many challenges faced including standard for handling unstructured data. Fourth step of implementation includes analysis from data being brought out. There are many challenges including legacy systems' ability to manage real- time data and more. Fifth step includes Intelligent actions which are taken after analysis is being done expressed as M2M interfaces using UI/UX. Challenges which are in this step includes security and privacy .

Conclusions and Future Directions

This survey of Internet of Things covers a comprehensive study of Internet of Things which discussed about different work done in this field. This survey gives an information about different architectures of Internet of Things. This survey focusses on different applications of Internet of things, its challenges, information security and more .The survey also gives a comparative study of different research papers and works already done in this field of IoT. There is no doubt that Internet of Things has a very good future in improving the quality of life by connecting smart devices around us .Some example of it will be a Smart Air conditioner which can be controlled via our smart phones automatically .Internet of Things has also played a role in improving the healthcare sector by connecting various health care devices like fit bit etc. Internet of Things has played an adequate role in improving the quality of life for people in the world. But there are many challenges which IoT should address as to successfully live up to its future expectations and these challenges cannot be ignored in a long run. One of the most important challenge is security as there were many instances where IoT devices were easily accessible by hackers and information was compromised. In order to take confidence of the people in business around the world , this challenge cannot be ignored .Other challenges includes legal accountability, privacy which also should be addressed. Since the IoT has not yet been realized, it might seem early to forecast the future directions of the IoT. One future vision for the IoT is the Web of Things. The Web of Things suggests the use of web standards to fully integrate smart objects into the World Wide Web. Using web technologies can make it easier for developers to build applications using smart objects and existing web protocols can more easily enable communication of different devices. Another future direction should be to make IoT devices more secure and maintain privacy. Future research should also focus of Quality of Experience (QoE) which is an important factor in making IoT a success in future. The IoT will create new legal challenges that must be addressed. Another future vision that involves integrating more devices to the IoT. This survey also mentions challenges and future directions that should be addressed. This survey focuses on comprehensive study of different survey papers in a brief way. This survey also gives a comparative study of advantages and disadvantages of this technology.



References

1. Ali Nauman ; Yazdan Ahmad Qadri ; Muhammad Amjad ; Yousaf Bin Zikria ; Muhammad Khalil Afzal ; Sung Won Kim "Multimedia Internet of Things: A Comprehensive Survey" published by IEEE in 15th Jan 2020
<https://ieeexplore.ieee.org/abstract/document/8950450>
2. Pallavi Sethi1 and Smruti R. Sarangi in Journal of Electrical and Computer Engineering "Internet of Things: Architectures, Protocols and Applications" /Hindawi/ 2017 ID 9324035 | <https://doi.org/10.1155/2017/9324035>
3. S. A. Alvi, B. Afzal, G. A. Shah, L. Atzori, and W. Mahmood, "Internet of Multimedia Things: Vision and challenges," Ad Hoc Network., vol. 33, pp. 87–111, Oct. 2015. Published by researchgate.net
4. A. Whitmore, A. Agarwal, and L. Da Xu, "The Internet of things—a survey of topics and trends," Information Systems Frontiers, vol. 17, no. 2, pp. 261–274, 2015.
5. Shancang Li & Li Da Xu & Shanshan Zhao," The internet of things: a survey" Published online: 26 April 2014. DOI 10.1007/s10796-014-9492-7
6. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Computer Network, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

Internet Technologies (COMP90007_2020_SM2)

Assignment 2

Name : Sakshi Chandel

LMS Canvas : schandel

Student Id : 1124298

Question 1:

Answer 1:

1)

Distance to E

n	A	B	C	D	E	F	
1	∞	∞	∞	∞	0	∞	{E}
2	3	∞	6	∞	--	2	{E,F}
3	3	9	6	4	--	--	{E,F,A}
4	--	5	6	4	--	--	{E,F,A,D}
5	--	5	6	--	--	--	{E,F,A,D,B}
6	--	--	6	--	--	--	{E,F,A,D,B,C}

Shortest distance from E->A = 3

Shortest distance from E->C = 6

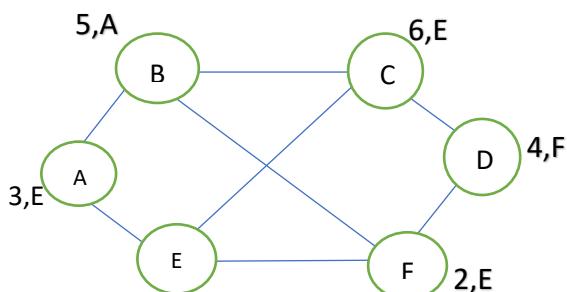
Shortest distance from E->F = 2

Shortest distance from E->B = 5

Shortest distance from E->D = 4

Shortest distance from E->E = 0

2) Sink tree of node E



The network link E->E = 0 is there in the table above but not in the sink tree of node E.

Question 2:

Answer 2:

Application	Bandwidth	Delay	Jitter	Loss
Zoom Meeting	High	High	High	Low
Online shopping	Medium	Medium	Low	Medium
VoIP	Low	Low	High	Low
Bank Transactions	Low	Medium	Medium	Medium

Question 3:

Answer 3:

TCP maximum data size = MTU (Maximum Transmission Unit) for Ethernet - **1500 bytes**

UDP safe maximum data size = **576 bytes** (including 8 bytes UDP header)

Question 4:

Answer 4:

1)

Flow control Congestion is accomplished by the receiver sending back a window to the sender. TCP window is the amount of un-acknowledged data a sender can send on a particular connection before it gets an acknowledgment back from the receiver.

Sender : -

The sending device sends all packets within the TCP window size as per TCP header without receiving an ACK,

and starts a timeout timer for each of them. It decrease the windows size by analyzing the traffic Congestion to reduce flow.

If a packet ACK not received at sending device, a timeout will occur and it will re transmit the lost Segment.

Receiver : -

Receiving device acknowledge each packet it received and have the sequence number of the last received packet.

After receiving the ACK from the receiving device, the sending device slides the window to right side.

2)

Window size is number of bytes determined by the receiving device which can vary.

Big windows are good for high bandwidth networks.

Window size keeps increasing as long as the receiver sends acknowledgments for all segments till max limit

and it get reduced when ACK not received (timeout).

In TCP slow start, the window size will initially grow exponentially window size doubles but once a packet is dropped, the window size will be reduced to one segment.

Question 5:

Answer 5:

When time out occurs TCP reacts strongly

1. It sets the value of threshold to one-half of the current window size
2. It sets congestion window to size of one segment
3. It starts the slow start phase again

Here when the congestion window size is 39KB when stime out occurs...Let's say the size of one segment is 1Kb so TCP set's congestion window to 1 Kb and threshold to 20Kb and starts slow start phase again.**(In slow start phase window size increases exponentially)...**

Here when congestion size is 1Kb 4 transmissions(rounds) are successful...after 1st round CNWD(congestion window) size will be 2...After second round CNWD size is 4Kb ...After third round CNWD size is 8Kb...after fourth round CNWD size is 16KB....Therefore the window size will be **16Kb** after 4 transmission bursts.

[Answer : 16 Kb](#)

COMP90007 Internet Technologies: Lab 1

Semester 2, 2020, Week 2

1 Question 1

Several ways to do this, but you can use ‘ipconfig’ on the command line (on Windows) to determine your IP address; or from Wireshark itself, determine that it is your computer sending the HTTP GET request to request a web page from the remote server.

The source IP address will depend on your computer. The destination IP address for the ‘unimelb.edu.au’ web server is 172.25.128.1 (when I tried within the University network).

2 Question 2

The amount of traffic captured will depend on the network conditions, whether some packets were lost and needed to be retransmitted, whether some background network traffic was also captured, etc.

3 Question 3

Again results may vary, but the overhead (for our purposes, which includes the headers from each of the packets and also the traffic involved with establishing and ending the connection) can be obtained by dividing the total traffic in bytes received from the previous question, by the file size of the retrieved HTML file (which is 44,220 bytes when I tried).

4 Question 4

The overheads/headers are used to allow each layer in the networking architecture perform their particular service. For example, the networking layer headers will have the source IP and destination IP addresses for that packet, which tells the intermediate hosts where the final destination of the packet is intended to be.

These headers are part of how data is transferred between the networking layers, what we call “encapsulation”, and keeps networking functions separate and modular between the layers.

Arguably, the size of the information contained in the headers can be reduced by having a “monolithic” architecture to networking, where there is only one “layer”, but then modularity and abstraction are sacrificed, which is bad from an engineering perspective.

5 Question 5

Apart from HTTP GET and HTTP 200 OK request and response packets, other packets of interest include the SYN and SYN ACK packets at the beginning which establishes a reliable

connection between two hosts (more on that when we get to TCP), and also the packets at the end when the connection is closed.

The packets in the middle are the data that make up the rest of the actual web page. Because the web page is quite large, the data will be split and transferred with multiple packets. It can also be observed that between every two or three packets that are received, our computer sends an ACK or acknowledgement packet to communicate to the remote host that the packets sent were successfully received. If the remote host fails to receive these packets, then it can be assumed that they were lost in transmission and need to be resent.

We will cover TCP handshaking and maintaining a reliable method of communication between two hosts in more detail as we progress in the semester.

6 Question 6

Similar to the last part in Question 4, a non-layered architecture may be more efficient, but at the cost of flexibility and modularity. Other disadvantages include a sizable overhead if the information to be transferred needs to be split across many packets.

However, the advantages of modularity and abstraction (also in engineering overall) outweigh the disadvantages. Advantages include, but aren't limited to, information hiding (i.e. if you were a network engineer, you would not need to know detailed information on the physical layer to work on improvements to another layer), and flexibility (the ability to change protocols in a certain layer, without affecting the operation of other layers).

There is a topic in research called "cross-layer optimisation" which breaks the barrier between the layers in an attempt to squeeze more performance, but this goes against the principle of abstraction.

7 Question 7 (Bonus Question)

For this question, you would have needed to plot a histogram, i.e. on the x-axis is a "bin" for a certain range of IATs, and on the y-axis the frequency or number of packets which fall into each particular bin. You should observe an exponential distribution. Feel free to use your favourite scientific computing/plotting package (Excel, MATLAB, Python, Mathematica, pen and paper, etc.).

COMP90007 Internet Technologies: Lab 1

Semester 2, 2020, Week 2

1 Objectives

- To examine the concept of encapsulation in networking and the overheads associated with the layered networking model.
- To learn how protocols and layering are represented in packets through the use of Wireshark.
- To examine IP (Internet Protocol) in detail. IP is the main network layer protocol used throughout the Internet.
- To examine TCP (Transmission Control Protocol) in detail. TCP is the main transport layer protocol used in the Internet.
- To examine HTTP (HyperText Transfer Protocol) in detail. HTTP is the main application layer protocol underlying the Web.

2 Requirements

There are two software requirements for this lab:

- *Wireshark*

Wireshark is a sniffing tool that allows the user to examine a packet trace, that is, a record of network traffic over time. It can be downloaded at <http://www.wireshark.org/download.html>.

- *wget*

‘wget’ is a tool to fetch web resources. A standalone Windows version can be found at <https://eternallybored.org/misc/wget/>. On Linux and OS X, ‘wget’ is usually pre-installed, or can be installed with a package manager. ‘curl’ is also a suitable alternative.

OS X users running newer versions of the operating system may need to download the *Development Release* of Wireshark.

3 Pre-lab

The tasks in this section are designed to familiarise yourself with the Wireshark software which will be used in this lab. This section should be done *before* your scheduled workshop session.

3.1 Getting started

- Close all browsers and other applications that use the network or the Internet. In Wireshark, go to Capture ⇒ Options.
- Select a network adapter to listen to, tick the box *resolve network-layer names* and untick the box *Use promiscuous mode on all interfaces*.
- Click on *Capture filter* and select *TCP only*.
- You are now ready to capture your first trace. Click *Start* and use your favourite web browser to fetch a URL. Once the web page has loaded, go back to Wireshark and stop the capture.

3.2 Inspecting the trace

You will now see a list of frames captured by Wireshark, including timestamps, protocol types and additional information. Note that we are going to use ‘packet’ as a general term for captured entities here. Strictly speaking, a unit of information at the link layer is called a **frame**. At the network layer it is called a **packet**, at the transport layer a **segment**, and at the application layer a **message**.

- Select a packet for which the protocol column is ‘HTTP’ and the info column says ‘GET’.

In the first window below the frame list you will see more detailed information about the packet and the headers of each layer (Have a look! Click on the ‘+’ to expand for more details). The second window below shows the raw hexadecimal data the packet is made up of and translations thereof.

The packet we selected is the packet that carries the web (HTTP) request sent from your computer to the server. The packet structure reflects the protocols that are in use. Since we are fetching a web page, we know that the protocol layers being used are. That is, HTTP is the application layer web protocol used to fetch URLs. Like many Internet applications, it runs on top of the TCP/IP transport and network layer protocols. The link and physical layer protocols depend on your network, but are typically combined in the form of Ethernet if your computer is wired, or 802.11 if your computer is wireless.

3.3 Pre-lab questions

Confirm the answer to these questions with your tutor.

1. What does *resolve network-layer names* do and why did we untick the box *Use promiscuous mode on all interfaces*. What does that option do?
2. Did you observe any other traffic other than the HTTP packets? What are several reasons for this additional traffic that is captured? Does this traffic serve any purpose?

4 Lab tasks

In this lab, we will measure the traffic generated in requesting a web page. We will compare the number of bytes sent and received through the network with the actual size of the web resource we request. ‘wget’ will be used to request and download a single web page while we examine the corresponding network traffic in Wireshark.

Step 1

Close all unnecessary background programs. This includes web browsers, e-mail clients and any other programs which may generate background network traffic, which can skew the measurements you take later on.

Step 2

Open up a Command Prompt and ensure that ‘wget.exe’ is in the current working directory. An easy way to open up a Command Prompt is to *Shift+Right-click* in the directory you have downloaded ‘wget’ to and click *Open command window here*.

In the command prompt, enter ‘`wget www.unimelb.edu.au`’, but do not press *Enter* yet. We need to prepare Wireshark to capture the traffic.

Step 3

Open Wireshark and prepare to start a new capture. Under *Capture options*, it will be useful to enter a capture filter to filter out background traffic irrelevant to our measurements. Enter ‘`tcp port http`’ in the capture filter text box. Make sure *Promiscuous mode* is unchecked. *Promiscuous mode* tells Wireshark to capture all packets that travel through our network interface—even those not intended for your computer.

If you have trouble capturing packets, first ensure you have the correct network interface selected—you may have to experiment a bit to see which interface is carrying the internet traffic on your computer. Alternatively, try clearing the capture filter and applying a *display* filter instead. The *display* filter syntax is a little different to the capture filter syntax. You will want to enter ‘`tcp.port eq 80`’ instead in the *Filter* text box in the Wireshark main window. If you are still unable to see packets being captured, try using port 8000 in the filters. The Unimelb proxy works on this port and may be causing problems.

Step 4

Once you have verified you can capture packets in Wireshark, start a new capture. Almost immediately after, hit *Enter* in the command prompt to download the Unimelb web page to your computer. Once the download has finished, stop the capture in Wireshark. The idea is to capture as little background network traffic as possible.

The packets you should have captured should start with a short TCP packet described as a SYN, which indicate the beginning of a connection. They will be followed by mostly longer packets in the middle (of roughly 1 to 1.5KB), of which the last one is a HTTP packet. This is the main portion of the download. And they will likely end with a short TCP packet that is part of ending the connection.

Step 5

In Wireshark, under the *File* menu, *export packet dissections* as a *CSV* file. Open your saved packet capture in your favourite spreadsheet software, and answer the questions below.

5 Questions

Once you have finished answering these questions, do you think you will have similar results each time? Can your dissimilarities, if any, be explained?

1. What is the source IP address of your computer? What is the destination IP address of the web page you requested?
2. How much traffic in bytes in total was received and transmitted in your request for the Unimelb web page? What is the percentage of traffic that originated from your computer, and what is the percentage of traffic that was sent by the remote host?
3. How do these sizes compare with the file size of the web page you downloaded? Estimate the download protocol overhead, or percentage of the download bytes taken up by protocol overhead.
4. What are the protocol overheads used for? Why are they useful? Aren't they a waste of space?
5. Are you able to identify other packets apart from the HTTP GET request and response packets in your capture? What might these packets be used for?
6. How do the protocol overheads relate to the networking layering architecture you have seen (or will see) in class? Is this architecture efficient? What are the advantages of this 'layered-cake' architecture? What are the disadvantages?
7. **Bonus:** Calculate the inter-arrival times (IAT) of the packets, i.e. the time between each packet arrival. Plot the number of packets against the IAT using a histogram. What sort of distribution do you observe? You may need to capture many more HTTP packets (this time using a web browser, for example) for a proper distribution to be seen.

COMP90007 Internet Technologies: Lab 1

Semester 2, 2020, Week 2

1 Objectives

- To examine the concept of encapsulation in networking and the overheads associated with the layered networking model.
- To learn how protocols and layering are represented in packets through the use of Wireshark.
- To examine IP (Internet Protocol) in detail. IP is the main network layer protocol used throughout the Internet.
- To examine TCP (Transmission Control Protocol) in detail. TCP is the main transport layer protocol used in the Internet.
- To examine HTTP (HyperText Transfer Protocol) in detail. HTTP is the main application layer protocol underlying the Web.

2 Requirements

There are two software requirements for this lab:

- *Wireshark*

Wireshark is a sniffing tool that allows the user to examine a packet trace, that is, a record of network traffic over time. It can be downloaded at <http://www.wireshark.org/download.html>.

- *wget*

‘wget’ is a tool to fetch web resources. A standalone Windows version can be found at <https://eternallybored.org/misc/wget/>. On Linux and OS X, ‘wget’ is usually pre-installed, or can be installed with a package manager. ‘curl’ is also a suitable alternative.

OS X users running newer versions of the operating system may need to download the *Development Release* of Wireshark.

3 Pre-lab

The tasks in this section are designed to familiarise yourself with the Wireshark software which will be used in this lab. This section should be done *before* your scheduled workshop session.

3.1 Getting started

- Close all browsers and other applications that use the network or the Internet. In Wireshark, go to Capture ⇒ Options.
- Select a network adapter to listen to, tick the box *resolve network-layer names* and untick the box *Use promiscuous mode on all interfaces*.
- Click on *Capture filter* and select *TCP only*.
- You are now ready to capture your first trace. Click *Start* and use your favourite web browser to fetch a URL. Once the web page has loaded, go back to Wireshark and stop the capture.

3.2 Inspecting the trace

You will now see a list of frames captured by Wireshark, including timestamps, protocol types and additional information. Note that we are going to use ‘packet’ as a general term for captured entities here. Strictly speaking, a unit of information at the link layer is called a **frame**. At the network layer it is called a **packet**, at the transport layer a **segment**, and at the application layer a **message**.

- Select a packet for which the protocol column is ‘HTTP’ and the info column says ‘GET’.

In the first window below the frame list you will see more detailed information about the packet and the headers of each layer (Have a look! Click on the ‘+’ to expand for more details). The second window below shows the raw hexadecimal data the packet is made up of and translations thereof.

The packet we selected is the packet that carries the web (HTTP) request sent from your computer to the server. The packet structure reflects the protocols that are in use. Since we are fetching a web page, we know that the protocol layers being used are. That is, HTTP is the application layer web protocol used to fetch URLs. Like many Internet applications, it runs on top of the TCP/IP transport and network layer protocols. The link and physical layer protocols depend on your network, but are typically combined in the form of Ethernet if your computer is wired, or 802.11 if your computer is wireless.

3.3 Pre-lab questions

Confirm the answer to these questions with your tutor.

1. What does *resolve network-layer names* do and why did we untick the box *Use promiscuous mode on all interfaces*. What does that option do?
2. Did you observe any other traffic other than the HTTP packets? What are several reasons for this additional traffic that is captured? Does this traffic serve any purpose?

4 Lab tasks

In this lab, we will measure the traffic generated in requesting a web page. We will compare the number of bytes sent and received through the network with the actual size of the web resource we request. ‘wget’ will be used to request and download a single web page while we examine the corresponding network traffic in Wireshark.

Step 1

Close all unnecessary background programs. This includes web browsers, e-mail clients and any other programs which may generate background network traffic, which can skew the measurements you take later on.

Step 2

Open up a Command Prompt and ensure that ‘wget.exe’ is in the current working directory. An easy way to open up a Command Prompt is to *Shift+Right-click* in the directory you have downloaded ‘wget’ to and click *Open command window here*.

In the command prompt, enter ‘`wget www.unimelb.edu.au`’, but do not press *Enter* yet. We need to prepare Wireshark to capture the traffic.

Step 3

Open Wireshark and prepare to start a new capture. Under *Capture options*, it will be useful to enter a capture filter to filter out background traffic irrelevant to our measurements. Enter ‘tcp port http’ in the capture filter text box. Make sure *Promiscuous mode* is unchecked. *Promiscuous mode* tells Wireshark to capture all packets that travel through our network interface—even those not intended for your computer.

If you have trouble capturing packets, first ensure you have the correct network interface selected—you may have to experiment a bit to see which interface is carrying the internet traffic on your computer. Alternatively, try clearing the capture filter and applying a *display* filter instead. The *display* filter syntax is a little different to the capture filter syntax. You will want to enter ‘`tcp.port eq 80`’ instead in the *Filter* text box in the Wireshark main window. If you are still unable to see packets being captured, try using port 8000 in the filters. The Unimelb proxy works on this port and may be causing problems.

Step 4

Once you have verified you can capture packets in Wireshark, start a new capture. Almost immediately after, hit *Enter* in the command prompt to download the Unimelb web page to your computer. Once the download has finished, stop the capture in Wireshark. The idea is to capture as little background network traffic as possible.

The packets you should have captured should start with a short TCP packet described as a SYN, which indicate the beginning of a connection. They will be followed by mostly longer packets in the middle (of roughly 1 to 1.5KB), of which the last one is a HTTP packet. This is the main portion of the download. And they will likely end with a short TCP packet that is part of ending the connection.

Step 5

In Wireshark, under the *File* menu, *export packet dissections* as a *CSV* file. Open your saved packet capture in your favourite spreadsheet software, and answer the questions below.

5 Questions

Once you have finished answering these questions, do you think you will have similar results each time? Can your dissimilarities, if any, be explained?

1. What is the source IP address of your computer? What is the destination IP address of the web page you requested?
2. How much traffic in bytes in total was received and transmitted in your request for the Unimelb web page? What is the percentage of traffic that originated from your computer, and what is the percentage of traffic that was sent by the remote host?
3. How do these sizes compare with the file size of the web page you downloaded? Estimate the download protocol overhead, or percentage of the download bytes taken up by protocol overhead.
4. What are the protocol overheads used for? Why are they useful? Aren't they a waste of space?
5. Are you able to identify other packets apart from the HTTP GET request and response packets in your capture? What might these packets be used for?
6. How do the protocol overheads relate to the networking layering architecture you have seen (or will see) in class? Is this architecture efficient? What are the advantages of this 'layered-cake' architecture? What are the disadvantages?
7. **Bonus:** Calculate the inter-arrival times (IAT) of the packets, i.e. the time between each packet arrival. Plot the number of packets against the IAT using a histogram. What sort of distribution do you observe? You may need to capture many more HTTP packets (this time using a web browser, for example) for a proper distribution to be seen.

COMP90007 Internet Technologies

Semester 2, 2020

Mid-Semester Exam Solution

Question 1

The transmission media used in the physical layer can be wired or wireless. Compare wireless medium microwave with wired medium fibre optics:

- (1) List both the advantages and disadvantages of these two media.
 - (2) Describe one typical scenario when it is better to use fibre optics than wireless transmission.
- (1) The advantages and disadvantages of these two transmission media include:
- Microwave:
- + Naturally supports mobility
 - + Naturally supports broadcast
 - Transmissions interfere and must be managed
 - Signal strengths hence data rates vary greatly
- Fibre optics:
- + Easy to maintain a fixed data rate over point-to-point links
 - + Provide high data rates over long distances.
 - Can be expensive to deploy, esp. over distances or places hard to reach
 - Doesn't readily support mobility or broadcast
- (2) When it is better to use fibre optics: backbone links between ISP facilities

Question 2

Calculate the Internet checksum (with 4-bit word) using one's complement arithmetic for sending data 1001 0011 1110 0110.

- (1) Show your calculation.
 - (2) Briefly explain how the receiver can use this checksum to detect any errors.
 - (3) Can the receiver correct errors using the checksum? Why or why not?
- (1) $1001 + 0011 = 1100$
 $1100 + 1110 = 1010 + 1$ (move overflow to last bit) = 1011
 $1011 + 0110 = 0001 + 1$ (move overflow to last bit) = 0010
One's complement: 1101
Checksum: 1101
- (2) The receiver compute checksum for 1001 0011 1110 0110 1101 to see if the result is 1111. If it is not, there are errors.

(3) The checksum cannot help receiver to correct error, as it cannot locate the errors.

Another valid reason is that based on Hamming distance, the Hamming distance of checksum is 2, so it cannot even correct a single error.

Question 3

The link utilisation can be measured by the proportion of transmission time among total communication time. For a stop and wait protocol, given a link with bandwidth 10 Mbps, one-way propagation delay 10 ms.

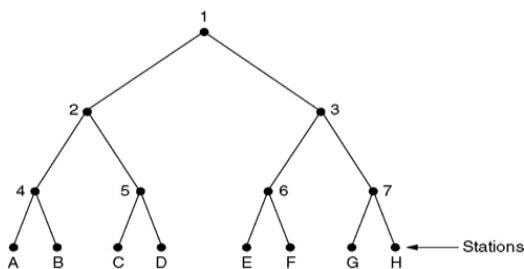
- (1) What is the link utilisation to transmit a frame of 1500 bytes? Show your calculation.
- (2) How can sliding window protocols improve the link utilisation? (using 2-3 sentences)

(1) Link utilisation = $L / (L + 2TpB) = 1500 * 8 / (1500 * 8 + 2 * 0.01 * 10^7) = 5.66\%$

(2) Sliding window protocols allow senders to keep transmitting the frames in the sliding window before receiving acknowledgements. The sender can increase the proportion of transmission time, which increases the link utilisation.

Question 4

We have an 8-station network with stations labelled as A, B, C..., H. All stations are competing to transmit using Adaptive Tree Walk protocol. Using the tree given below, show step by step what happens **in each timeslot** if A, C, D and G have data to send now.



- (1) Slot 1: A, C, D, G - collision
- (2) Slot 2: A, C, D - collision
- (3) Slot 3: A can send
- (4) Slot 4: C, D - collision
- (5) Slot 5: C can send
- (6) Slot 6: D can send
- (7) Slot 7: G can send

Question 5

Given a network A with prefix 128.18.3.0/25.

- (1) What is the maximum number of hosts that this network can represent?
- (2) Given an IP address 128.18.3.140, does this address belong to network A? Explain your reasoning.

- (1) The length of host portion is $32 - 25 = 7$, so the number of hosts are $2^7 = 128$
- (2) Network A starts from: 128.18.3.0 and the block ends at 128.18.3.127, this IP address 128.18.3.140 doesn't belong to network A.

Programming Abstractions in Cloud

Muhammed Tawfiqul Islam

*Cloud Computing and Distributed Systems (CLOUDS) Laboratory,
School of Computing and Information Systems
The University of Melbourne, Australia*



Outline

- Cloud Computing
 - Computing as a utility
- Programming networked computers
 - TCP ports and sockets
 - RPC
- Cloud Application Platforms
 - Platform as a Service (PaaS)
- Aneka: A Cloud Middle-ware Platform
- Summary

Cloud Computing

- Outsource IT facilities to cloud providers
- Avoid expensive up-front investments
- Computing as a utility
- On-demand
- Pay for what they use
- Virtualized resources

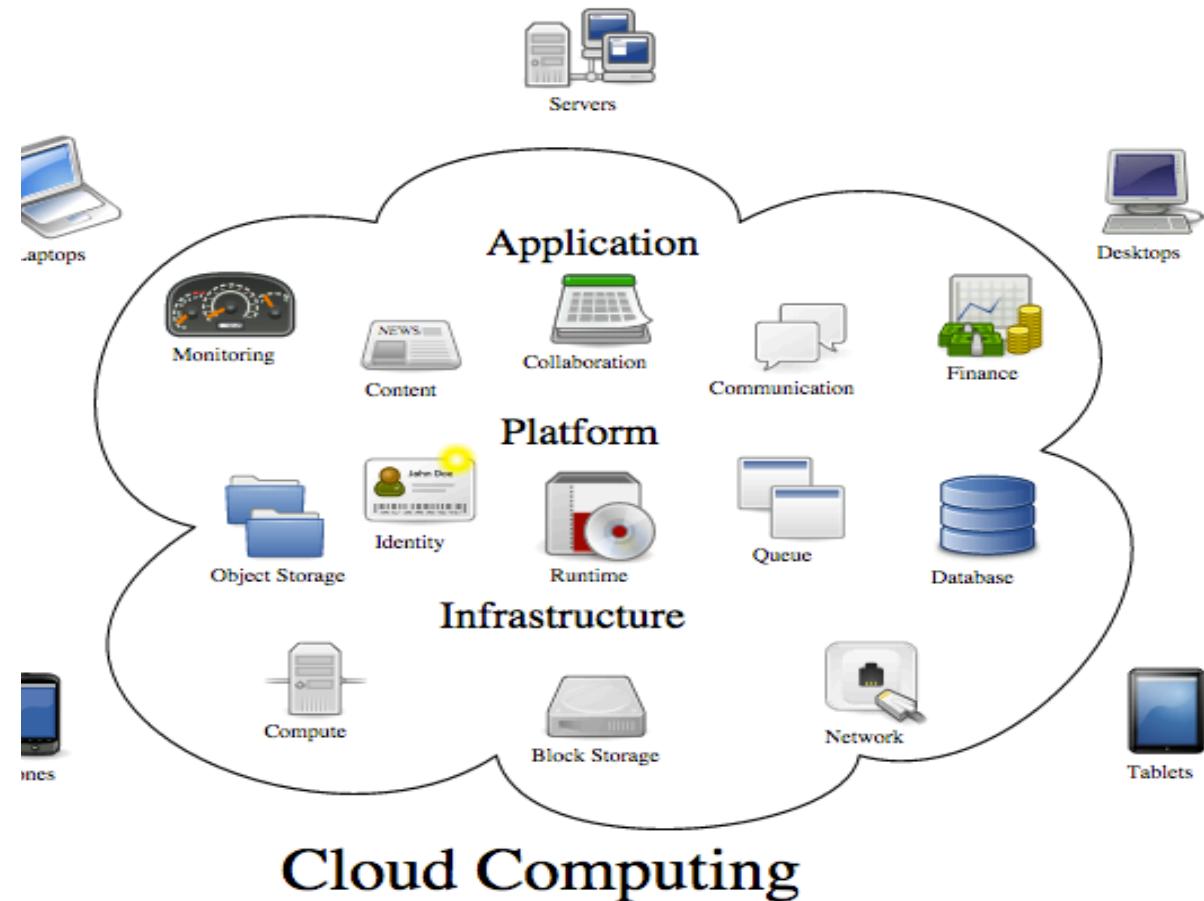


Figure: http://en.wikipedia.org/wiki/Cloud_computing

Programming Networked Computers

- Sockets
 - Most familiar communication way for networked applications
 - Bound to a local port
 - IP address + port = Socket ID/ address
 - Socket can be used for sending and receiving data.
 - Acts as a programming interface to application code and transport layer
 - Each socket is associated with a protocol (**UDP or TCP**)

Programming Networked Computers

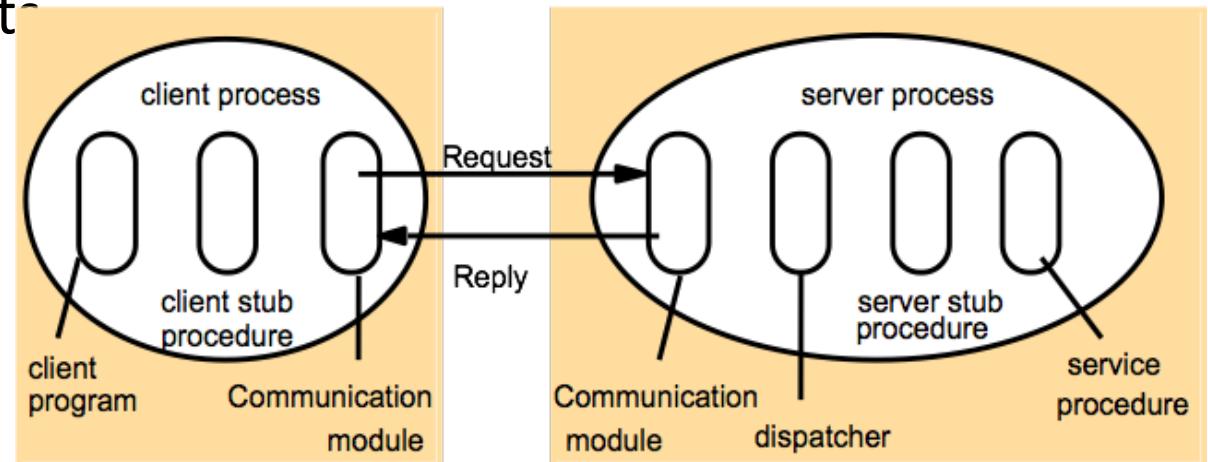
- Transmission Control Protocol (TCP)
 - Connection Oriented
 - Provides an abstraction for a two-way stream (called as packets).
 - Streams do not have message boundaries.
 - Stream provide the basis for producer/consumer communication.
 - Data sent by the producer are queued until the consumer is ready to receive them.
 - Example use cases – http, ftp
- TCP Socket Programming Demo

Programming Networked Computers

- Socket handling becomes complex when applications scale to large number of servers
 - Typical application in cloud spans from few hundred to thousands of servers
 - Individual port to each server
 - Read write buffers
 - Synchronization
 - Exception handling

Programming Networked Computers

- Remote Procedure Call (RPC)
 - RPCs enable clients to execute procedures in server processes based on a defined service interface (Procedural languages- C, Fortran, Go)
 - Remote Method Invocation (RMI) for Object Oriented languages (JAVA, C#)
 - Higher abstraction than sockets
- Key components of RPC:
 - Communication Module
 - Client Stub Procedure
 - Dispatcher
 - Server stub procedure



Programming Networked Computers

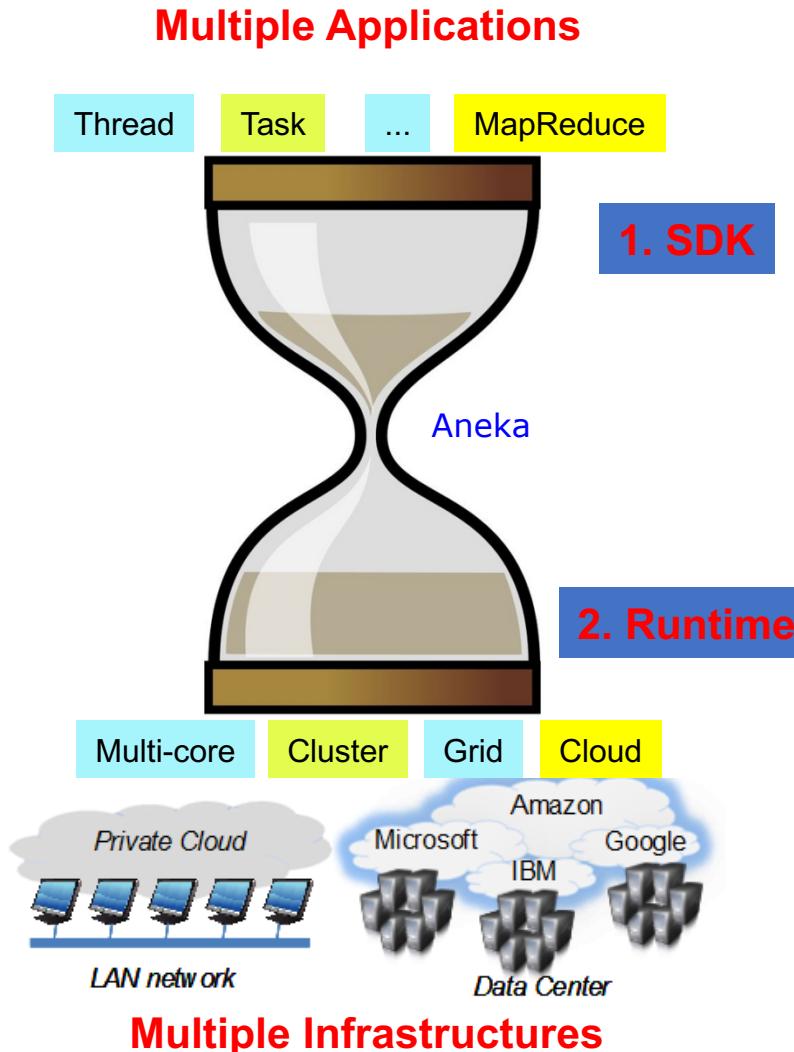
- RPC/ RMI is useful in simple applications where resources are usually static in nature.
- Cloud applications are dynamic that requires:
 - Elasticity – On demand resource provision and de-provision
 - Monitoring – Heartbeat, Budget Constraints, etc.
 - Robustness – Availability, Failure management
- Middleware that seamlessly manages dynamic resources and handles runtime network communication is necessary.

Cloud Application Platforms

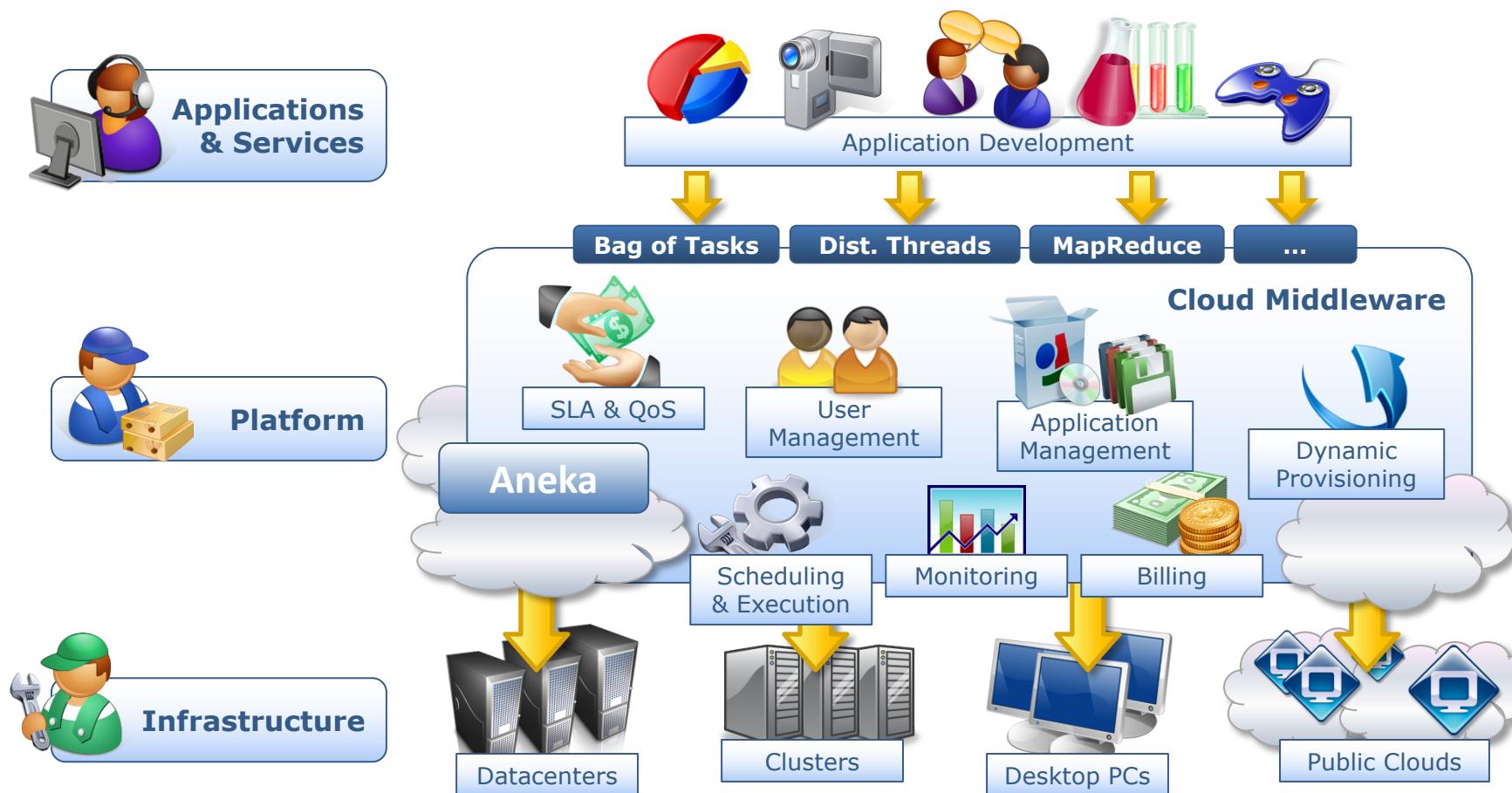
- Cloud Application Platforms (CAP) provide highest level of abstractions to build the applications
- Abstraction through SDKs, APIs and middle-ware platforms
- Less efforts on resource management and more focus on application
- Examples include
 - Google App Engine
 - Deploy web applications on the fly
 - Microsoft Azure PaaS platforms
 - Highly suitable platform for enterprise application developments
 - **Aneka (Manjrasoft, CLOUDS Lab, UniMelb)**
 - Cloud application platform with multiple programming models

Aneka: Cloud Application Platform(CAP) for Resource-Intensive/Elastic Apps

- Platform supporting multiple Cloud programming models (Task, Thread, MapReduce)
- **SDK (Software Development Kit)** containing APIs for multiple programming models and tools
- **Runtime** Environment for managing application execution on Clouds
- Suitable for
 - Development of Enterprise Cloud Applications
 - Cloud enabling legacy applications
- Portability for Customer Apps:
 - Enterprise ↔ Public Clouds
 - .NET/Win ↔ Mono/Linux



Aneka as a Cloud Application Platform



Aneka as a Cloud Application Platform

- Aneka Demo

Summary

- Cloud Platforms like Aneka simplifies the process to build the applications.
- Although middle-ware platforms provide abstraction to programmer, the underlying complex operations are still need to be implemented inside the platforms.
- Acquiring skills to build applications in such platforms is crucial.

Questions?

For any queries, please contact me:

tawfiqul.islam@unimelb.edu.au

COMP90007 Internet Technologies, Project 2

Semester 2, 2020

Due Date: October 29, 2020 (Thursday) 5:00 PM (Melbourne Time)

1. Introduction

This project forms 15% of your final mark. The project is about creating a short survey on a contemporary topic in networking that we give as below. This is a cornerstone activity that we would like you to learn about for your continuing development. In our sector, this is an activity you would need to do by yourselves regularly to keep up to date with developments.

The main outcome of this project is a short report. Detailed report formatting instructions are found at the end of this project description (in Project Administration Section).

For this project, you need to write a **brief survey** on: **Internet of Things**. This is a hot area in computer communications that will be good to get an overview of as well as practicing writing a survey report.

Project Requirements:

As expected from any survey, students are expected to not only list top papers in an area but also categorize these developments/approaches and compare/critique them. This is at the core of the survey. A list of papers with comments only is called an “annotated bibliography” and is not a survey and is not the purpose of this project.

As this is a simple survey, we do not expect you to learn every paper in detail and be overly comprehensive about the topic but rather cover the key papers, classifications/parameters.

Following report section-headings and length of each section are highly recommended:

- Introduction to the Topic (1 page)
- Related Work (3 pages)
- Comparison of Key Approaches (benefits and disadvantages) (1 page)
- Conclusions and Future Directions (1 page)
- References (1 page)

We expect that students should have looked at and cited at least 5 papers not including other surveys on this topic. Reading more papers are better but in general there are diminishing returns at a certain point... Given the topic, the number above is adequate!

General Guidelines:

You should initially start reading Wikipedia articles, news, and similar webpages to get high level idea for what Internet of Things is. Then you should use scholar.google.com or similar scholarly publication search engines for performing a more detailed background search and do further reading. It is important to note that if you login to our library with your student credentials, you will be able to access papers that are returned by these engines for free of charge in most cases.

The topic we have chosen is something you would already be partially familiar with some of the algorithms you have already seen earlier in the semester. The project is expected to be completed in 3-4 weeks in total. The stages of your project can be summarized as: Background search/reading selected papers (should not take more than 10 days and can be done in 1 week), organization of your report/drafting key points of your sections (1 week), finishing your report (1 week). After this exercise, students are expected to have a good idea in the topic.

Note: Using pure google.com is a good start but will most likely land you on more general news items again and again than recent research and developments and proper articles. You are encouraged to find other survey papers that already exist in these topics. Find one that is recent and relevant. Better, find many and you will see authors look at similar but not the same set of algorithms/protocols. They may also have different classifications. These should give you an idea on what common/popular methods exist and what key comparison parameters you can have between solutions. They are also a good example on how to write surveys. You cannot use other surveys or a book as a sole source for your survey and/or directly take their approach. You should also refer to individual key papers mentioned in the surveys and read them and make your own judgements and categorizations (although many of the categories you create could be similar to other survey papers in the area.) The number of citations a paper gets in scholar.google.com is an indicator about its leadership in the field, i.e., beyond the fact that it is cited in other surveys.

When reading the papers, please note that a technical paper is not read like a novel, i.e., not read from cover to cover sequentially, but is read in a manner that you can quickly grasp the key ideas, benefits/disadvantages. In particular to write a survey this is enough. At implementation time, technical papers could be read to the very extreme detail.

2. Project Administration

The deadline for the report is specified at the start of this document. **Late submissions will get a penalty of 10% per day**, similar to the previous assignments and projects. The report must be submitted as a PDF file on LMS. **Please include a proper title for the survey and your name, student id and login username as well, i.e., just before the introduction section starts.**

The report should be in A4 size paper in 12-point Times New Roman for the main text with 1.5 line spacing with 1-inch margins. It should be single column. The report **should not exceed 10 pages**, including all figures, appendices and references etc. The main text of the report, without figures etc., is expected to be not less than 4 pages as well, and thus putting many figures one after the other is not an acceptable report.

All explanations should be your own words and proper citations should be used when needed.
The project is an individual project. Students should work independently of each other on this project. Not sharing information about papers you found is also important, as finding is a part of this experience.

The marking criteria for this project is as follows:

- Format and structure of the report (2 points)
- Coverage of the survey regarding papers read (4 points)
- Description of the individual papers in report (4 points)
- Categorization and comparison of papers/approaches (3 points)
- Future directions and concluding discussions (2 points)

A couple of useful links for report writing are below as reference:

<https://students.unimelb.edu.au/academic-skills/explore-our-resources/report-writing/reviewing-the-literature>

<https://students.unimelb.edu.au/academic-skills/explore-our-resources/report-writing/technical-report-writing>

COMP90007 Internet Technologies

Project 1 – Network Analysis

Student Name: Sakshi Chandel

Login username : schandel@student.unimelb.edu.au

Student ID: 1124298

Question 2 :

Answer 2.1: “-d” in “tracert -d-wl” (in Windows) stands for do not resolve host addresses to host names.

“-w” stands for wait timeout milliseconds for each reply.

“-wl” stands for setting waiting time response to 1

Question 2.2

Answer 2.2: From Appendix Section 2

We can find the data of all the IP addresses

1) iperf.he.net (Section 2.a)

Distance between source and destination = 12752.384 km

Hop counts = 8

2) bouygues.testdebit.info (Section 2.b)

Distance between source and destination = 7208.271 km

Hop counts = 12

3) iperf.comneonext.de Section (2.c)

Distance between source and destination = 6854.0 km

Hop counts =11

4) ikoula.testdebit.info (Section 2.d)

Distance between source and destination = 7202.2 km

Hop counts = 9

5) st2.nn.ertelecom.ru (Section 2.e)

Distance between source and destination = 4699.0 km

Hop counts = 9

6) iperf.biznetnetworks.com (Section 2.f)

Distance between source and destination = 4378.03 km

Hop counts = 7

7) iperf.scottlinux.com (Section 2.g)

Distance between source and destination = 12820.9 km

Hop counts = 6

8) speedtest.serverius.net (Section 2.h)

Distance between source and destination = 6931.4 km

Hop counts 8

9) iperf.volia.net (Section 2.i)

Distance between source and destination = 5644.345 km

Hop counts = 8

10) iperf.eenet.ee (Section 2.j) (Not mentioned in project,searched by myself)

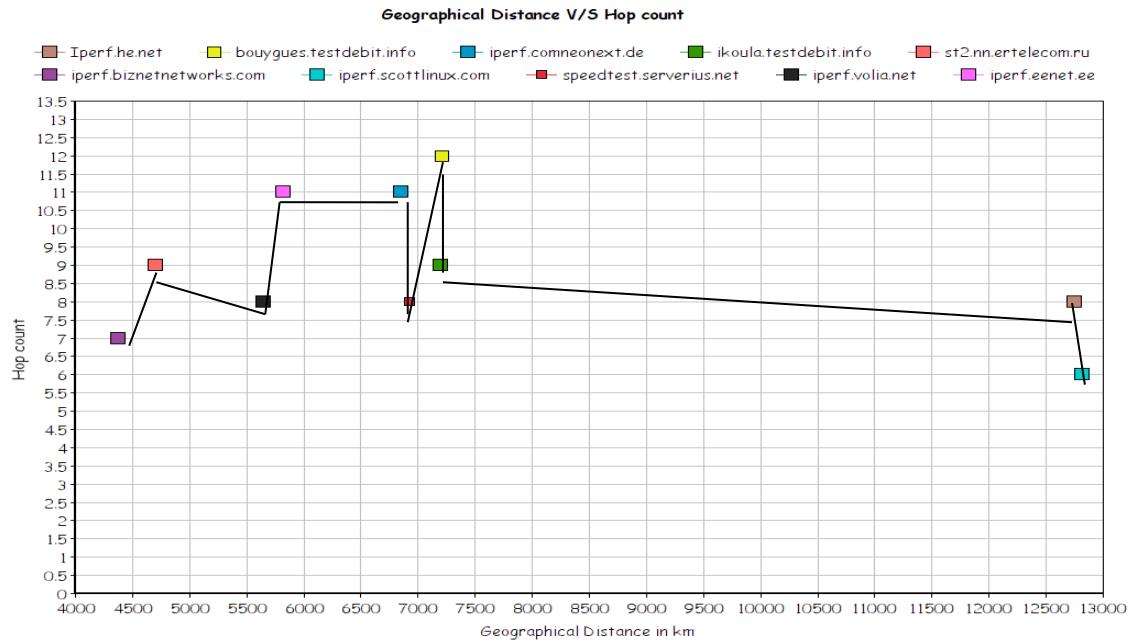
Distance between source and destination = 5819.501 km

Hop count = 11

The plot of graph shows distance between source and destination V/S Hop count below:

Analysis shows there is a loose relation between hop counts and distance b/w source destination

Generally thinking ,there should be a linear relation between both but it is not absolute as I can see in the graph.Because we can see two servers having same hop count but having a large difference in geographical distance .



Question 3.1 :

Answer 3.1: Delay time of all the hosts below: From Appendix ,Section 3

Reference for finding the distance : My city : "Jabalpur,Madhya Pradesh,India"

<https://www.freemaptools.com/how-far-is-it-between.html>

1) Iperf.he.net (Section 2.a)

Average = 338.3 ms, Jitter = 46.49 ms, Distance = 12752.3

2) bouygues.testdebit.info (Section 2.b)

Average= 215.6 ms, Jitter = 20.52 ms, Distance= 7208.27

3) iperf.comneonext.de (Section 2.c)

Average = 340 ms, Jitter : 188 ms ,Distance= 6854.00

4) ikoula.testdebit.info (Section 2.d)

Average = 240 ms, Jitter = 51.61 ms, Distance= 6932.49

5) st2.nn.ertelecom.ru (Section 2.e)

Average = 270.3 ms, Jitter = 85.73, Distance=4699.0

6) iperf.biznetnetworks.com (Section 2.f)

Average = 248.33 ms, Jitter = 8 ms, Distance = 4378.0

7) iperf.scottlinux.com (Section 2.g)

Average = 333ms ,Jitter = 45.53 ms, Distance=12820.98

8) speedtest.serverius.net (Section 2.h)

Average = 238 ms, Jitter = 41.40 ms ,Distance = 6932.49

9) iperf.volia.net (Section 2.i)

Average = 180 ms, Jitter = 4.53 ms, Distance=5644.34

10) iperf.eenet.ee

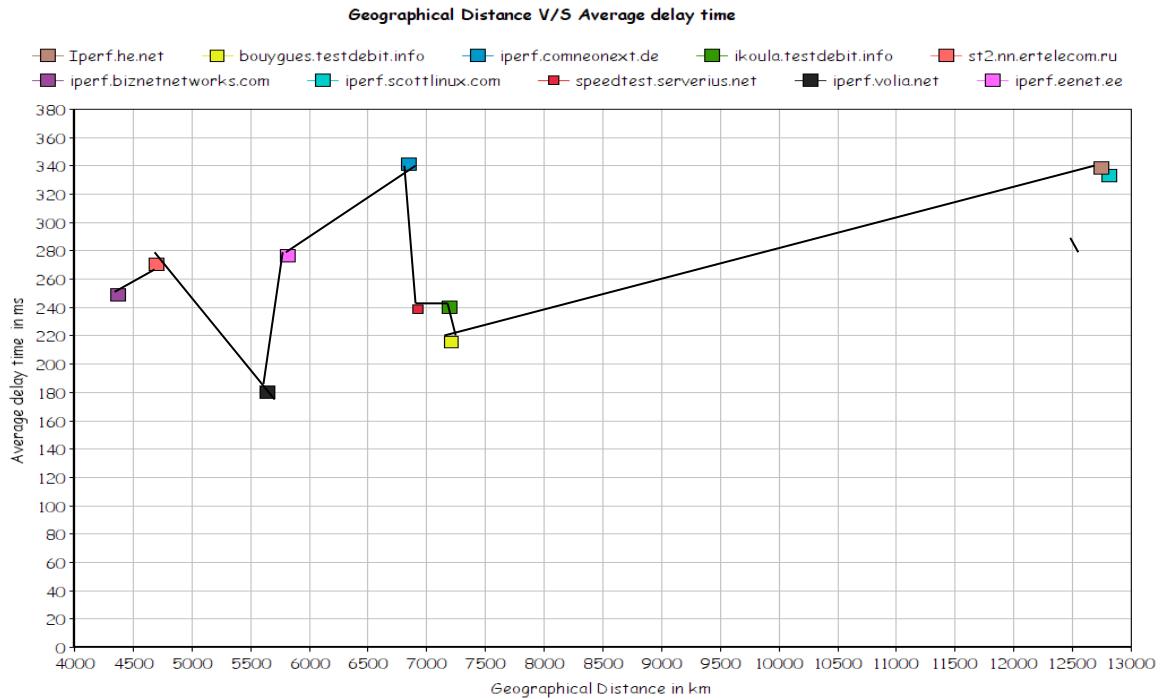
(Section 2.j)

Average = 276 ms ,Jitter = 0 ms ,Distane = 5819.501

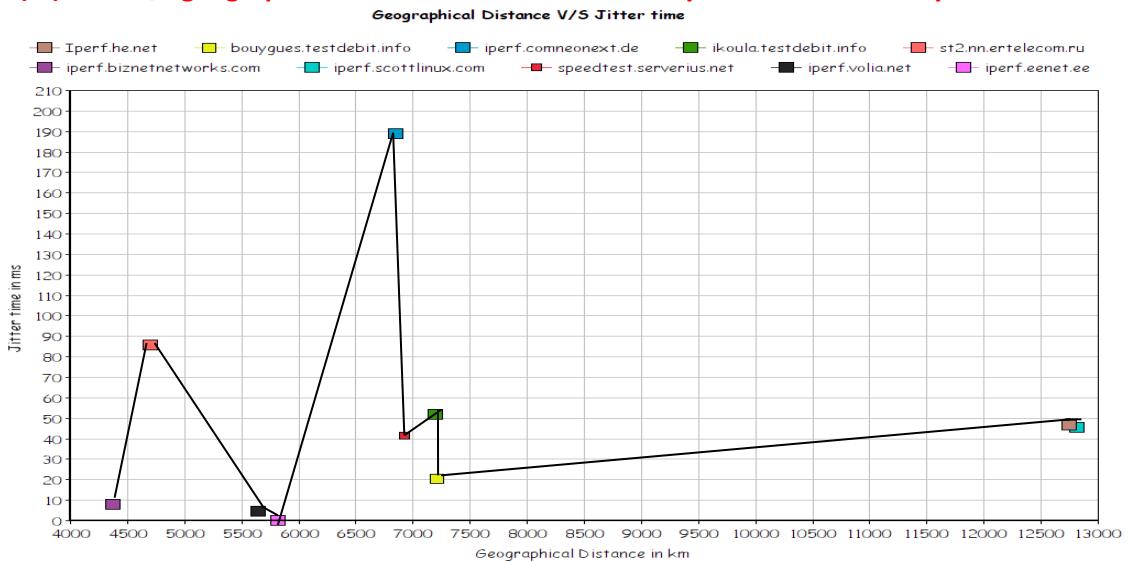
Below are the graphs for jitter and average delay time v/s geographical distance

GRAPHS

a)Average delay time v/s geographical distance between source city and destination city



Graph 2) b) Jitter v/s geographical distance between source city and destination city

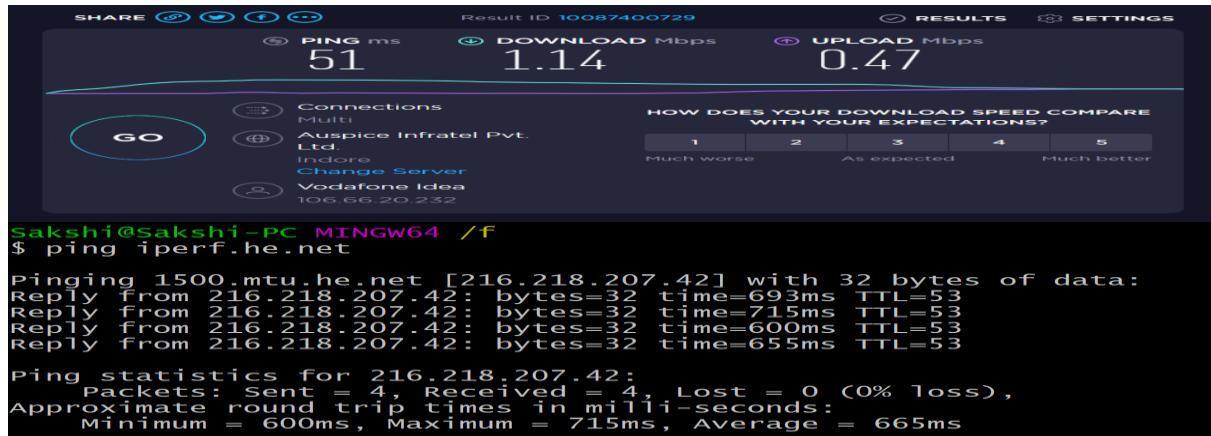


Question 3.2

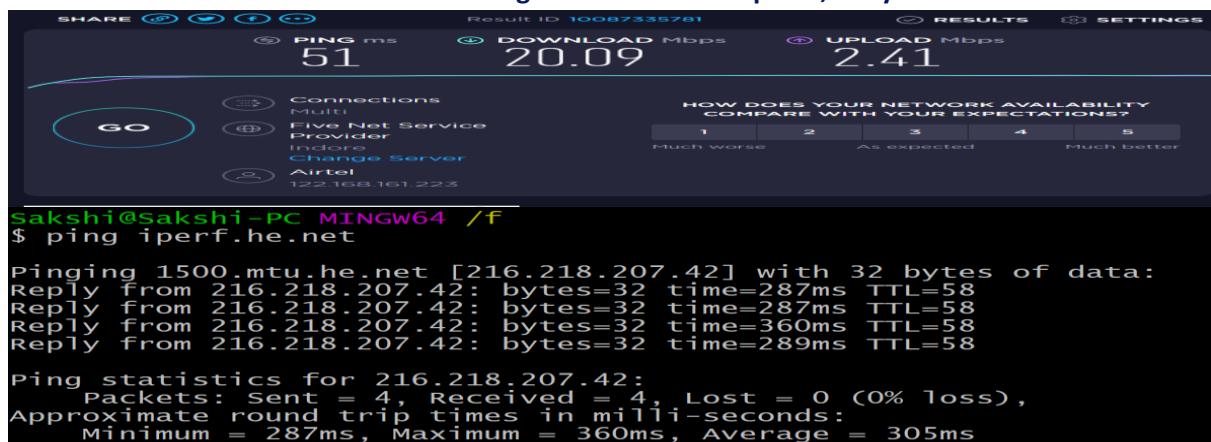
Answer 3.2) Jitter and average delay time is loosely dependent on geo graphical distance because even if distance is increasing jitter and average delay time is not increasing .But if my judgement is

wrong can be because of my ISP (Internet service provider) or due to network traffic and network speed. Average delay time and jitter is dependent on the network environment like network latency, change in routes, congestion. I have ran two three commands for each IP address when I used wifi (greater download speed), delay time was less as compared to when I used mobile data/hotspot (less download speed). Following are the details about it :

Network details when ran command with less download speed ,delay time was less.



Network details when ran command with greater download speed ,delay time was less.



Question 4.1

Answer 4.1 : From Appendix, section 4

Bandwidth delay product is a measurement of how many bits can fill up a network link. It gives the maximum amount of data that can be transmitted by the sender at a given time before waiting for acknowledgment.

1) iperf.he.net (Section 2.a)

Mean bandwidth = 1.99 Mbits/sec

2) bouygues.testdebit.info (Section 2.b)

Mean Bandwidth : 5.37 Mbits/sec

c) iperf.comneonext.de (Section 2.c)

Mean bandwidth = 4.9 Mbits/sec

d) ikoula.testdebit.info (Section 2.d)

Mean Bandwidth : 5.48 Mbits/sec

e) st2.nn.ertelecom.ru (Section 2.e)

Mean bandwidth: 2.41 Mbits/sec

- f) iperf.biznetnetworks.com (Section 2.f)**
 Mean bandwidth : 2.39 Mbits/sec
- g) iperf.scottlinux.com (Section 2.g)**
 Mean bandwidth = 981 Kbits/sec
- h) speedtest.serverius.net (Section 2.h)**
 Mean bandwidth : 70.3 Kbits/sec
- i) iperf.volia.net (Section 2.i)**
 Mean bandwidth : 59.8 Kbits/sec
- j) iperf.eenet.ee (Section 2.j)**
 Mean bandwidth : 1.68 Mbits/sec

Question 4.2)

Answer4.2: From Appendix,Section 4

Calculating Bandwidthdelay of all Ip addresses

a) iperf.he.net

Delay = 338 ms (from question3), Bandwidth = 1.99 Mbits/sec (from question 4.1)

Bandwidth delay product = $338 \times 1.99 = 672.6$ Kbits

b) bouygues.testdebit.info

Delay = 215.6ms (from question3), Bandwidth = 5.37 Mbits/sec (from question 4.1)

Bandwidth delay product = $215.6 \times 5.37 = 1157.772$ Kbits

(c) iperf.comneonext.de

Delay = 340.66 ms (from question3), Bandwidth = 4.9 Mbits/sec (from question 4.1)

Bandwidth delay product = $340.66 \times 4.9 = 1669.234$ Kbits

d) ikoula.testdebit.info

Delay = 240 ms (from question3), Bandwidth = 5.48 Mbits/sec (from question 4.1)

Bandwidth delay product = $240 \times 5.48 = 1315.2$ Kbits

e) st2.nn.ertelecom.ru

Delay = 270.3ms (from question3), Bandwidth = 2.41 Mbits/sec (from question 4.1)

Bandwidth delay product = $270.3 \times 2.41 = 651.423$ Kbits

f) iperf.biznetnetworks.com

Delay = 248.33ms (from question3), Bandwidth = 2.39 Mbits/sec (from question 4.1)

Bandwidth delay product = $248.33 \times 2.39 = 593.5$ Kbits

g) iperf.scottlinux.com

Delay = 333 ms (from question3), Bandwidth = 981 Kbits/sec (from question 4.1)

Bandwidth delay product = $333 \times 981 = 326.673$ Kbits

h) speedtest.serverius.net

Delay = 238.33 ms (from question3), Bandwidth = 70.3 Kbits/sec (from question 4.1)

Bandwidth delay product = $238.33 \times 70.3 = 16.754$ Kbits

i) iperf.volia.net

Delay = 180 ms (from question3), Bandwidth = 59.8 Kbits/sec (from question 4.1)

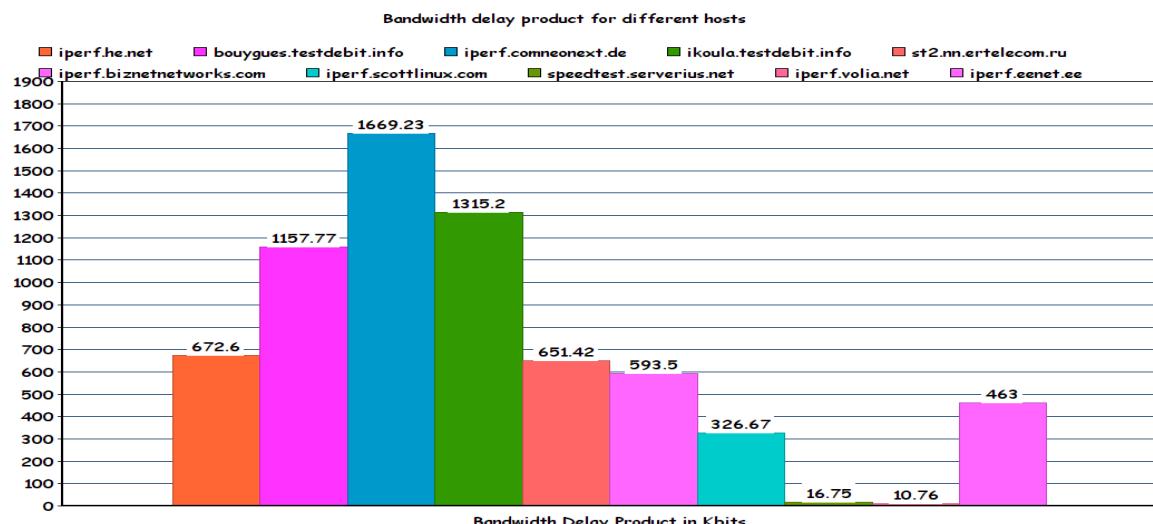
Bandwidth delay product = $180 \times 59.8 = 10.764$ Kbits

j) iperf.eenet.ee

Delay = 276 ms (from question3), Bandwidth = 1.68 Mbits/sec (from question 4.1)

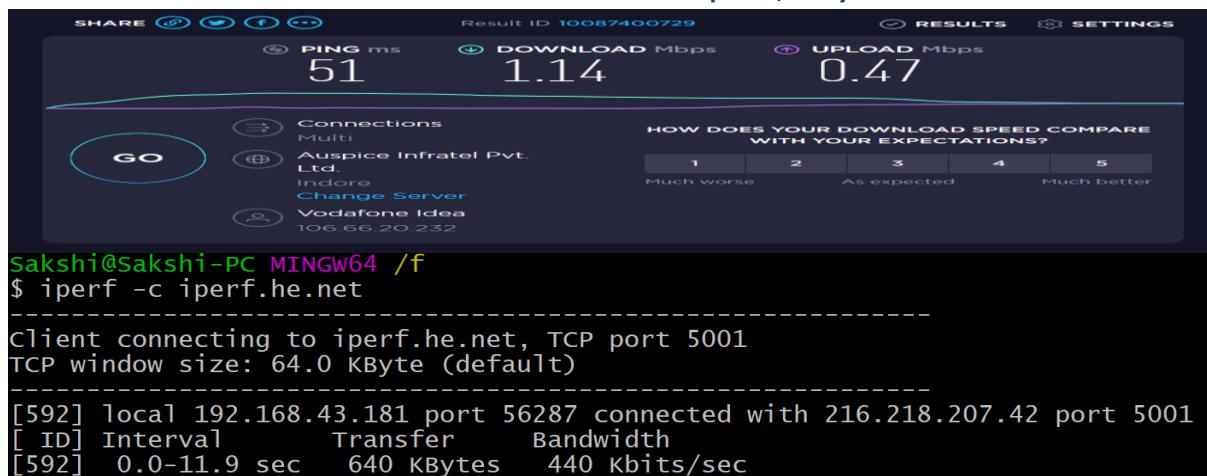
Bandwidth delay product = $276 * 1.68 = 463$ Kbits

Bar graph depicting Bandwidth delay product of all IP addresses.

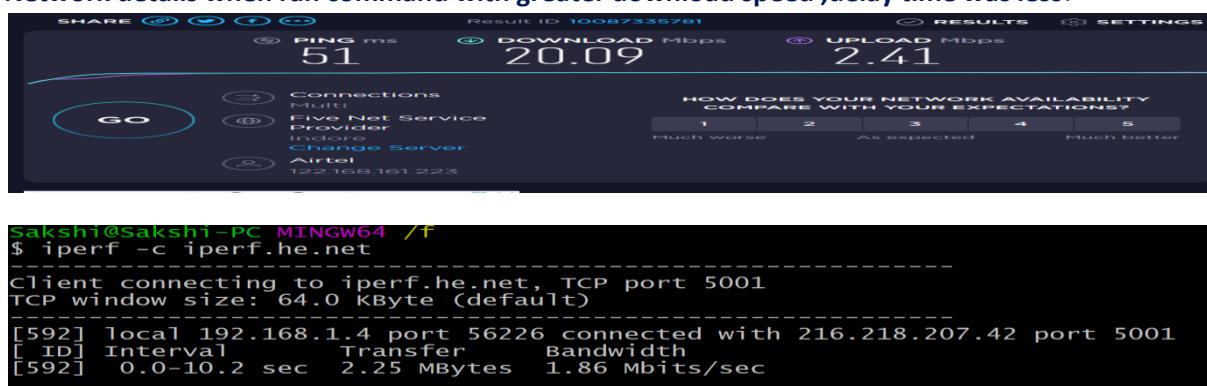


Bandwidth product delay depends on the network like traffic ,congestion in the network and network speed.

Network details when ran command with less download speed ,delay time was



Network details when ran command with greater download speed ,delay time was less.



Question 4.3

Answer 4.3 From Appendix Section 4

a) iperf.he.net (Section 2.a)

Bandwidth delay product = $338 * 1.99 = 672.6$ Kbits, Hop Count = 8

b) bouygues.testdebit.info (Section 2.b)

Bandwidth delay product = $470 * 5.37 = 2523.9$ Kbits , Hop Count = 12

c) iperf.comneonext.de (Section 2.c)

Bandwidth delay product = $442.6 * 4.9 = 2168.74$ Kbits , Hop Count = 11

d) ikoula.testdebit.info (Section 2.d)

Bandwidth delay product = $475.3 * 5.48 = 2604.6$ Kbits, Hop Count = 9

e) st2.nn.ertelecom.ru (Section 2.e)

Bandwidth delay product = $523 * 2.41 = 1260.43$ Kbits, Hop Count = 9

f) iperf.biznetnetworks.com (Section 2.f)

Bandwidth delay product = $574.6 * 2.39 = 1373.2$ Kbits, Hop Count=7

g) iperf.scottlinux.com (Section 2.g)

Bandwidth delay product = $615 * 981 = 603.315$ Kbits, Hop Count=6

h) speedtest.serverius.net (Section 2.h)

Bandwidth delay product = $1089 * 70.3 = 76.5$ Kbits, Hop Count =8

i) iperf.volia.net (Section 2.i)

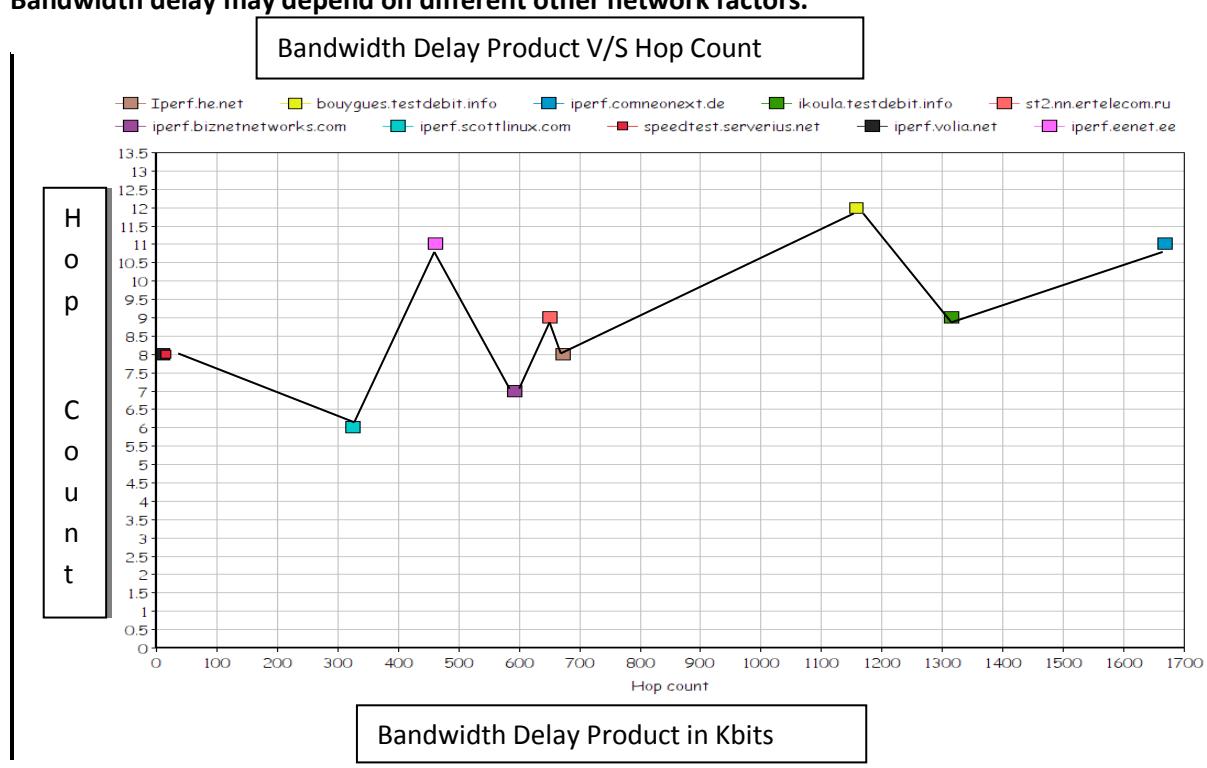
Bandwidth delay product = $178 * 59.8 = 10.6$ Kbits, Hop Count =8

j) iperf.eenet.ee (Section 2.j)

Bandwidth delay product = $276 * 1.68 = 463$ Kbits, Hop Count=11

Bandwidth delay product is loosely dependent on hop count. As we can see in the graph there is no linear relation between them. Hop count is not dependent on bandwidth delay product.

Bandwidth delay may depend on different other network factors.



Question 4.4

Answer 4.4 : After running tests for bandwidth, delay, and jitter the factor that depends on it are network like network speed and bandwidth .

To improve the result, better internet connection with good bandwidth and closing other applications running on system to reduce network traffic should be used.

References

Reference for finding the distance : My city : "Jabalpur, Madhya Pradesh, India"
<https://www.freemaptools.com/how-far-is-it-between.html>

Reference of finding standard deviation and mean :
<https://www.calculator.net/standard-deviation-calculator.html>

Appendix

Section 2: Details about all hosts

a) iperf.he.net

```
Sakshi@sakshi-PC MINGW64 ~
$ tracert -d-w1 iperf.he.net
Tracing route to 1500.mtu.he.net [216.218.207.42]
over a maximum of 30 hops:
 1   2 ms      1 ms      1 ms      192.168.1.1
 2   11 ms     12 ms     22 ms     125.21.18.205
 3   539 ms    1001 ms   527 ms    182.79.222.237
 4   873 ms    998 ms   998 ms    198.32.118.57
 5   770 ms    1182 ms   *        184.105.81.218
 6   2747 ms   294 ms   581 ms    72.52.92.117
 7   285 ms    282 ms   3855 ms   184.105.65.210
 8   292 ms    667 ms   292 ms    216.218.207.42
Trace complete.
```

b) bouygues.testdebit.info

```
Sakshi@sakshi-PC MINGW64 ~
$ tracert -d-w1 bouygues.testdebit.info
Tracing route to bouygues.testdebit.info [89.84.1.222]
over a maximum of 30 hops:
 1   1 ms      <1 ms      <1 ms      192.168.1.1
 2   11 ms     11 ms      11 ms      125.16.168.89
 3   157 ms    160 ms    157 ms     116.119.44.158
 4   161 ms    160 ms    161 ms     149.14.227.1
 5   151 ms    151 ms    150 ms     130.117.48.205
 6   150 ms    149 ms    149 ms     154.54.57.70
 7   158 ms    158 ms    158 ms     130.117.1.46
 8   152 ms    152 ms    152 ms     149.14.121.234
 9   151 ms    152 ms    151 ms     62.34.2.57
10   *         *         *         Request timed out.
11   150 ms    149 ms    149 ms     89.89.101.141
12   160 ms    160 ms    160 ms     89.84.1.222
Trace complete.
```

c) iperf.comneonext.de

```
Sakshi@sakshi-PC MINGW64 ~
$ tracert -d-w1 iperf.comneonext.de
Tracing route to iperf.comneonext.de [91.195.241.136]
over a maximum of 30 hops:
 1   74 ms      <1 ms      <1 ms      192.168.1.1
 2   11 ms      11 ms      11 ms      125.21.20.121
 3   148 ms     148 ms     149 ms     116.119.36.144
 4   149 ms     149 ms     147 ms     62.115.42.118
 5   166 ms     165 ms     165 ms     62.115.114.200
 6   166 ms     166 ms     168 ms     62.115.124.47
 7   166 ms     167 ms     166 ms     62.115.120.78
 8   189 ms     188 ms     197 ms     62.115.160.178
 9   166 ms     165 ms     166 ms     91.195.241.102
10   171 ms     169 ms     170 ms     91.195.241.106
11   165 ms     165 ms     165 ms     91.195.241.136
Trace complete.
```

d) ikoula.testdebit.info

```
Sakshi@sakshi-PC MINGW64 ~
$ tracert -d-wl ikoula.testdebit.info
Tracing route to ikoula.testdebit.info [213.246.63.45]
over a maximum of 30 hops:
1    73 ms      <1 ms      1 ms   192.168.1.1
2    11 ms      11 ms      12 ms   125.16.168.89
3    159 ms     157 ms     156 ms   116.119.36.142
4    162 ms     169 ms     165 ms   195.66.224.21
5    164 ms     183 ms     165 ms   184.105.223.254
6    152 ms     152 ms     160 ms   184.104.205.18
7    154 ms     157 ms     154 ms   213.246.50.193
8    *          155 ms     165 ms   213.246.50.182
9    154 ms     153 ms     153 ms   213.246.63.45

Trace complete.
```

e) st2.nn.ertelecom.ru

```
Sakshi@sakshi-PC MINGW64 ~
$ tracert -d-wl st2.nn.ertelecom.ru
Tracing route to st2.nn.ertelecom.ru [91.144.184.232]
over a maximum of 30 hops:
1    73 ms      <1 ms      <1 ms   192.168.1.1
2    10 ms      12 ms      10 ms   125.21.20.121
3    147 ms     147 ms     147 ms   116.119.36.144
4    192 ms     191 ms     192 ms   80.249.209.216
5    205 ms     208 ms     206 ms   87.245.233.246
6    205 ms     203 ms     201 ms   87.245.254.154
7    220 ms     218 ms     218 ms   109.194.232.26
8    218 ms     217 ms     217 ms   109.194.232.25
9    217 ms     222 ms     218 ms   91.144.184.232

Trace complete.
```

f) iperf.biznetnetworks.com

```
Sakshi@sakshi-PC MINGW64 ~
$ tracert -d-wl iperf.biznetnetworks.com
Tracing route to iperf.biznetnetworks.com [117.102.109.186]
over a maximum of 30 hops:
1    1 ms       1 ms       <1 ms   192.168.1.1
2    11 ms      10 ms      13 ms   125.21.20.121
3    147 ms     129 ms     124 ms   116.119.44.184
4    350 ms      *         150 ms   80.249.210.131
5    195 ms     196 ms     199 ms   202.169.34.177
6    203 ms     202 ms     206 ms   182.253.99.106
7    202 ms     201 ms     201 ms   117.102.109.186

Trace complete.
```

g) iperf.scottlinux.com

```
Sakshi@sakshi-PC MINGW64 ~
$ tracert -d-wl iperf.scottlinux.com
Tracing route to iperf.scottlinux.com [45.33.39.39]
over a maximum of 30 hops:
1    340 ms      <1 ms      1 ms   192.168.1.1
2    10 ms       10 ms      12 ms   125.16.168.89
3    253 ms     250 ms     244 ms   116.119.44.136
4    254 ms     254 ms     266 ms   206.72.211.198
5    254 ms     254 ms     254 ms   173.230.159.65
6    254 ms     254 ms     255 ms   45.33.39.39

Trace complete.
```

h) speedtest.serverius.net

```
Sakshi@sakshi-PC MINGW64 ~
$ tracert -d-w1 speedtest.serverius.net

Tracing route to speedtest.serverius.net [178.21.16.76]
over a maximum of 30 hops:

 1   76 ms      <1 ms      2 ms    192.168.1.1
 2   11 ms      11 ms      11 ms    125.16.168.69
 3   130 ms     131 ms     131 ms    116.119.44.184
 4   176 ms     176 ms     177 ms    80.249.209.216
 5   155 ms     155 ms     155 ms    87.245.232.44
 6   160 ms     210 ms     160 ms    87.245.246.61
 7   152 ms     152 ms     152 ms    185.8.179.33
 8   154 ms     154 ms     153 ms    178.21.16.76

Trace complete.
```

i) iperf.volia.net

```
Sakshi@sakshi-PC MINGW64 ~
$ tracert -d-w1 iperf.volia.net

Tracing route to speedtest.volia.net [77.120.3.236]
over a maximum of 30 hops:

 1   75 ms      1 ms      <1 ms    192.168.1.1
 2   11 ms      11 ms      11 ms    125.16.168.89
 3   135 ms     122 ms     122 ms    182.79.222.81
 4   177 ms     171 ms     171 ms    80.249.209.216
 5   177 ms     176 ms     176 ms    87.245.232.155
 6   176 ms     177 ms     176 ms    87.245.237.57
 7   175 ms     174 ms     176 ms    77.120.1.125
 8   178 ms     177 ms     178 ms    77.120.1.49
 9   172 ms     171 ms     172 ms    77.120.3.236

Trace complete.
```

Section 3: Delay time of all the hosts below:

Reference of finding standard deviation and mean :

<https://www.calculator.net/standard-deviation-calculator.html>

a)iperf.he.net

```
Sakshi@sakshi-PC MINGW64 ~
$ ping iperf.he.net

Pinging 1500.mtu.he.net [216.218.207.42] with 32 bytes of data:
Reply from 216.218.207.42: bytes=32 time=291ms TTL=58
Reply from 216.218.207.42: bytes=32 time=291ms TTL=58
Reply from 216.218.207.42: bytes=32 time=291ms TTL=58
Reply from 216.218.207.42: bytes=32 time=341ms TTL=58

Ping statistics for 216.218.207.42:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 291ms, Maximum = 341ms, Average = 303ms
```

```

Sakshi@sakshi-PC MINGW64 /f
$ ping iperf.he.net
Pinging 1500.mtu.he.net [216.218.207.42] with 32 bytes of data:
Reply from 216.218.207.42: bytes=32 time=587ms TTL=53
Reply from 216.218.207.42: bytes=32 time=338ms TTL=53
Reply from 216.218.207.42: bytes=32 time=320ms TTL=53
Reply from 216.218.207.42: bytes=32 time=320ms TTL=53
Ping statistics for 216.218.207.42:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 320ms, Maximum = 587ms, Average = 391ms

Sakshi@sakshi-PC MINGW64 /f
$ ping iperf.he.net
Pinging 1500.mtu.he.net [216.218.207.42] with 32 bytes of data:
Reply from 216.218.207.42: bytes=32 time=321ms TTL=53
Reply from 216.218.207.42: bytes=32 time=322ms TTL=53
Reply from 216.218.207.42: bytes=32 time=321ms TTL=53
Reply from 216.218.207.42: bytes=32 time=322ms TTL=53
Ping statistics for 216.218.207.42:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 321ms, Maximum = 322ms, Average = 321ms

```

Standard Deviation Calculator

Result

Standard Deviation, s: **46.490142324296**

Count, N: 3
 Sum, Σx : 1015
 Mean, \bar{x} : 338.333333333333
 Variance, s^2 : 2161.3333333333

Steps

$$\begin{aligned}
 s &= \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}, \\
 s^2 &= \frac{\sum (x_i - \bar{x})^2}{N-1} \\
 &= \frac{(303 - 338.3333333333)^2 + \dots + (321 - 338.3333333333)^2}{3-1} \\
 &= \frac{4322.66666666667}{2} \\
 &= 2161.3333333333 \\
 s &= \sqrt{2161.3333333333} \\
 &= 46.490142324296
 \end{aligned}$$

b) bouygues.testdebit.info

```

  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 159ms, Maximum = 378ms, Average = 233ms

Sakshi@sakshi-PC MINGW64 ~
$ ping bouygues.testdebit.info
Pinging bouygues.testdebit.info [89.84.1.222] with 32 bytes of data:
Reply from 89.84.1.222: bytes=32 time=266ms TTL=54
Reply from 89.84.1.222: bytes=32 time=184ms TTL=54
Reply from 89.84.1.222: bytes=32 time=206ms TTL=54
Reply from 89.84.1.222: bytes=32 time=229ms TTL=54
Ping statistics for 89.84.1.222:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 184ms, Maximum = 266ms, Average = 221ms

Sakshi@sakshi-PC MINGW64 ~
$ ping bouygues.testdebit.info
Pinging bouygues.testdebit.info [89.84.1.222] with 32 bytes of data:
Reply from 89.84.1.222: bytes=32 time=171ms TTL=54
Reply from 89.84.1.222: bytes=32 time=271ms TTL=54
Reply from 89.84.1.222: bytes=32 time=158ms TTL=54
Reply from 89.84.1.222: bytes=32 time=172ms TTL=54
Ping statistics for 89.84.1.222:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 158ms, Maximum = 271ms, Average = 193ms

```

Standard Deviation Calculator

Result

Standard Deviation, s: **20.526405757788**

Count, N: 3
Sum, Σx : 647
Mean, \bar{x} : 215.666666666667
Variance, s^2 : 421.333333333333

Steps

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2},$$
$$s^2 = \frac{\sum(x_i - \bar{x})^2}{N-1}$$
$$= \frac{(233 - 215.666666666667)^2 + \dots + (193 - 215.666666666667)^2}{3-1}$$
$$= \frac{842.666666666667}{2}$$
$$= 421.333333333333$$
$$s = \sqrt{421.333333333333}$$
$$= 20.526405757788$$

c) iperf.comneonext.de

Approximate round trip times in milli-seconds:
Minimum = 178ms, Maximum = 1691ms, Average = 558ms

Sakshi@sakshi-PC MINGW64 ~
\$ ping iperf.comneonext.de

Pinging iperf.comneonext.de [91.195.241.136] with 32 bytes of data:
Reply from 91.195.241.136: bytes=32 time=214ms TTL=55
Reply from 91.195.241.136: bytes=32 time=234ms TTL=55
Reply from 91.195.241.136: bytes=32 time=256ms TTL=55
Reply from 91.195.241.136: bytes=32 time=277ms TTL=55

Ping statistics for 91.195.241.136:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 214ms, Maximum = 277ms, Average = 245ms

Sakshi@sakshi-PC MINGW64 ~
\$ ping iperf.comneonext.de

Pinging iperf.comneonext.de [91.195.241.136] with 32 bytes of data:
Reply from 91.195.241.136: bytes=32 time=192ms TTL=55
Reply from 91.195.241.136: bytes=32 time=214ms TTL=55
Reply from 91.195.241.136: bytes=32 time=207ms TTL=55
Reply from 91.195.241.136: bytes=32 time=266ms TTL=55

Ping statistics for 91.195.241.136:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 192ms, Maximum = 266ms, Average = 219ms

Standard Deviation Calculator

Result

Standard Deviation, s: **188.66460540688**

Count, N: 3
Sum, Σx : 1022
Mean, \bar{x} : 340.666666666667
Variance, s^2 : 35594.3333333333

Steps

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2},$$
$$s^2 = \frac{\sum(x_i - \bar{x})^2}{N-1}$$
$$= \frac{(558 - 340.666666666667)^2 + \dots + (219 - 340.666666666667)^2}{3-1}$$
$$= \frac{71188.6666666667}{2}$$
$$= 35594.3333333333$$
$$s = \sqrt{35594.3333333333}$$
$$= 188.66460540688$$

d) ikoula.testdebit.info

```
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 169ms, Maximum = 579ms, Average = 297ms

Sakshi@sakshi-PC MINGW64 ~
$ ping ikoula.testdebit.info

Pinging ikoula.testdebit.info [213.246.63.45] with 32 bytes of data:
Reply from 213.246.63.45: bytes=32 time=280ms TTL=58
Reply from 213.246.63.45: bytes=32 time=168ms TTL=58
Reply from 213.246.63.45: bytes=32 time=169ms TTL=58
Reply from 213.246.63.45: bytes=32 time=168ms TTL=58

Ping statistics for 213.246.63.45:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 168ms, Maximum = 280ms, Average = 196ms

Sakshi@sakshi-PC MINGW64 ~
$ ping ikoula.testdebit.info

Pinging ikoula.testdebit.info [213.246.63.45] with 32 bytes of data:
Reply from 213.246.63.45: bytes=32 time=196ms TTL=58
Reply from 213.246.63.45: bytes=32 time=217ms TTL=58
Reply from 213.246.63.45: bytes=32 time=239ms TTL=58
Reply from 213.246.63.45: bytes=32 time=261ms TTL=58

Ping statistics for 213.246.63.45:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 196ms, Maximum = 261ms, Average = 228ms
```

Standard Deviation Calculator

Result

Standard Deviation, s: **51.617180602328**

Count, N: 3
Sum, Σx : 721
Mean, \bar{x} : 240.333333333333
Variance, s^2 : 2664.3333333333

Steps

$$\begin{aligned}s &= \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}, \\s^2 &= \frac{\sum (x_i - \bar{x})^2}{N-1} \\&= \frac{(297 - 240.3333333333)^2 + \dots + (228 - 240.3333333333)^2}{3-1} \\&= \frac{5328.6666666667}{2} \\&= 2664.3333333333 \\s &= \sqrt{2664.3333333333} \\&= 51.617180602328\end{aligned}$$

e) st2.nn.ertelecom.ru

```
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 196ms, Maximum = 261ms, Average = 228ms  
  
Sakshi@sakshi-PC MINGW64 ~  
$ ping st2.nn.ertelecom.ru  
  
Pinging st2.nn.ertelecom.ru [91.144.184.232] with 32 bytes of data:  
Reply from 91.144.184.232: bytes=32 time=695ms TTL=56  
Reply from 91.144.184.232: bytes=32 time=307ms TTL=56  
Reply from 91.144.184.232: bytes=32 time=227ms TTL=56  
Reply from 91.144.184.232: bytes=32 time=248ms TTL=56  
  
Ping statistics for 91.144.184.232:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 227ms, Maximum = 695ms, Average = 369ms  
  
Sakshi@sakshi-PC MINGW64 ~  
$ ping st2.nn.ertelecom.ru  
  
Pinging st2.nn.ertelecom.ru [91.144.184.232] with 32 bytes of data:  
Reply from 91.144.184.232: bytes=32 time=214ms TTL=56  
Reply from 91.144.184.232: bytes=32 time=195ms TTL=56  
Reply from 91.144.184.232: bytes=32 time=258ms TTL=56  
Reply from 91.144.184.232: bytes=32 time=190ms TTL=56  
  
Ping statistics for 91.144.184.232:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 190ms, Maximum = 258ms, Average = 214ms
```

Standard Deviation Calculator

Result

Standard Deviation, s: **85.734085014849**

Count, N: 3
Sum, Σx : 811
Mean, \bar{x} : 270.33333333333
Variance, s^2 : 7350.33333333333

Steps

$$\begin{aligned}s &= \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}, \\ s^2 &= \frac{\sum (x_i - \bar{x})^2}{N-1} \\ &= \frac{(228 - 270.3333333333)^2 + \dots + (214 - 270.3333333333)^2}{3-1} \\ &= \frac{14700.6666666667}{2} \\ &= 7350.3333333333 \\ s &= \sqrt{7350.3333333333} \\ &= 85.734085014849\end{aligned}$$

f) iperf.biznetnetworks.com

```
Approximate round trip times in milli-seconds:  
    Minimum = 217ms, Maximum = 306ms, Average = 239ms  
  
Sakshi@sakshi-PC MINGW64 ~  
$ ping iperf.biznetnetworks.com  
  
Pinging iperf.biznetnetworks.com [117.102.109.186] with 32 bytes of data:  
Reply from 117.102.109.186: bytes=32 time=220ms TTL=58  
Reply from 117.102.109.186: bytes=32 time=218ms TTL=58  
Reply from 117.102.109.186: bytes=32 time=329ms TTL=58  
Reply from 117.102.109.186: bytes=32 time=248ms TTL=58  
  
Ping statistics for 117.102.109.186:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 218ms, Maximum = 329ms, Average = 253ms  
  
Sakshi@sakshi-PC MINGW64 ~  
$ ping iperf.biznetnetworks.com  
  
Pinging iperf.biznetnetworks.com [117.102.109.186] with 32 bytes of data:  
Reply from 117.102.109.186: bytes=32 time=229ms TTL=58  
Reply from 117.102.109.186: bytes=32 time=272ms TTL=58  
Reply from 117.102.109.186: bytes=32 time=295ms TTL=58  
Reply from 117.102.109.186: bytes=32 time=218ms TTL=58  
  
Ping statistics for 117.102.109.186:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 218ms, Maximum = 295ms, Average = 253ms
```

Standard Deviation Calculator

Result

Standard Deviation, s: **8.0829037686548**

Count, N: 3

Sum, Σx : 745

Mean, \bar{x} : 248.33333333333

Variance, s^2 : 65.33333333333

Steps

$$\begin{aligned}s &= \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}, \\s^2 &= \frac{\sum (x_i - \bar{x})^2}{N-1} \\&= \frac{(239 - 248.3333333333)^2 + \dots + (253 - 248.3333333333)^2}{3-1} \\&= \frac{130.66666666667}{2} \\&= 65.33333333333 \\s &= \sqrt{65.33333333333} \\&= 8.0829037686548\end{aligned}$$

g)iperf.scottlinux.com

```
Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
Approximate round trip times in milli-seconds:
    Minimum = 378ms, Maximum = 387ms, Average = 382ms

Sakshi@sakshi-PC MINGW64 ~
$ ping iperf.scottlinux.com

Pinging iperf.scottlinux.com [45.33.39.39] with 32 bytes of data:
Reply from 45.33.39.39: bytes=32 time=386ms TTL=55
Reply from 45.33.39.39: bytes=32 time=305ms TTL=55
Reply from 45.33.39.39: bytes=32 time=325ms TTL=55
Reply from 45.33.39.39: bytes=32 time=287ms TTL=55

Ping statistics for 45.33.39.39:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 287ms, Maximum = 386ms, Average = 325ms

Sakshi@sakshi-PC MINGW64 ~
$ ping iperf.scottlinux.com

Pinging iperf.scottlinux.com [45.33.39.39] with 32 bytes of data:
Reply from 45.33.39.39: bytes=32 time=287ms TTL=55
Reply from 45.33.39.39: bytes=32 time=287ms TTL=55
Reply from 45.33.39.39: bytes=32 time=290ms TTL=55
Reply from 45.33.39.39: bytes=32 time=304ms TTL=55

Ping statistics for 45.33.39.39:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 287ms, Maximum = 304ms, Average = 292ms
```

Standard Deviation Calculator

Result

Standard Deviation, s: **45.530209751329**

Count, N: 3
Sum, Σx : 999
Mean, \bar{x} : 333
Variance, s^2 : 2073

Steps

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2},$$
$$s^2 = \frac{\sum (x_i - \bar{x})^2}{N-1}$$
$$= \frac{(325 - 333)^2 + \dots + (382 - 333)^2}{3-1}$$
$$= \frac{4146}{2}$$
$$= 2073$$
$$s = \sqrt{2073}$$
$$= 45.530209751329$$

h) speedtest.serverius.net

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 160ms, Maximum = 560ms, Average = 285ms

Sakshi@sakshi-PC MINGW64 ~
$ ping speedtest.serverius.net

Pinging speedtest.serverius.net [178.21.16.76] with 32 bytes of data:
Reply from 178.21.16.76: bytes=32 time=239ms TTL=56
Reply from 178.21.16.76: bytes=32 time=229ms TTL=56
Reply from 178.21.16.76: bytes=32 time=158ms TTL=56
Reply from 178.21.16.76: bytes=32 time=200ms TTL=56

Ping statistics for 178.21.16.76:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 158ms, Maximum = 239ms, Average = 206ms

Sakshi@sakshi-PC MINGW64 ~
$ ping speedtest.serverius.net

Pinging speedtest.serverius.net [178.21.16.76] with 32 bytes of data:
Reply from 178.21.16.76: bytes=32 time=157ms TTL=56
Reply from 178.21.16.76: bytes=32 time=159ms TTL=56
Reply from 178.21.16.76: bytes=32 time=424ms TTL=56
Reply from 178.21.16.76: bytes=32 time=159ms TTL=56

Ping statistics for 178.21.16.76:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 157ms, Maximum = 424ms, Average = 224ms
```

Standard Deviation Calculator

Result

Standard Deviation, s: **41.404508611181**

Count, N: 3
Sum, Σx : 715
Mean, \bar{x} : 238.333333333333
Variance, s^2 : 1714.3333333333

Steps

$$\begin{aligned}s &= \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}, \\ s^2 &= \frac{\sum (x_i - \bar{x})^2}{N-1} \\ &= \frac{(285 - 238.3333333333)^2 + \dots + (224 - 238.3333333333)^2}{3-1} \\ &= \frac{3428.6666666667}{2} \\ &= 1714.3333333333 \\ s &= \sqrt{1714.3333333333} \\ &= 41.404508611181\end{aligned}$$

i) iperf.volia.net

```
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 175ms, Maximum = 242ms, Average = 191ms  
  
Sakshi@sakshi-PC MINGW64 ~  
$ ping iperf.volia.net  
  
Pinging speedtest.volia.net [77.120.3.236] with 32 bytes of data:  
Reply from 77.120.3.236: bytes=32 time=175ms TTL=56  
  
Ping statistics for 77.120.3.236:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 175ms, Maximum = 175ms, Average = 175ms  
  
Sakshi@sakshi-PC MINGW64 ~  
$ ping iperf.volia.net  
  
Pinging speedtest.volia.net [77.120.3.236] with 32 bytes of data:  
Reply from 77.120.3.236: bytes=32 time=175ms TTL=56  
Reply from 77.120.3.236: bytes=32 time=174ms TTL=56  
Reply from 77.120.3.236: bytes=32 time=175ms TTL=56  
Reply from 77.120.3.236: bytes=32 time=175ms TTL=56  
  
Ping statistics for 77.120.3.236:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 174ms, Maximum = 175ms, Average = 174ms
```

Standard Deviation Calculator

Result

Standard Deviation, s: **9.5393920141695**

Count, N: 3
Sum, Σx : 540
Mean, \bar{x} : 180
Variance, s^2 : 91

Steps

$$\begin{aligned}s &= \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}, \\ s^2 &= \frac{\sum (x_i - \bar{x})^2}{N-1} \\ &= \frac{(191 - 180)^2 + \dots + (175 - 180)^2}{3-1} \\ &= \frac{182}{2} \\ &= 91 \\ s &= \sqrt{91} \\ &= 9.5393920141695\end{aligned}$$

j) iperf.eenet.ee

```
Approximate round trip times in milli-seconds:  
    Minimum = 276ms, Maximum = 278ms, Average = 276ms  
  
Sakshi@sakshi-PC MINGW64 ~  
$ ping iperf.eenet.ee  
  
Pinging iperf.eenet.ee [193.40.55.7] with 32 bytes of data:  
Reply from 193.40.55.7: bytes=32 time=276ms TTL=53  
  
Ping statistics for 193.40.55.7:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 276ms, Maximum = 276ms, Average = 276ms  
  
Sakshi@sakshi-PC MINGW64 ~  
$ ping iperf.eenet.ee  
  
Pinging iperf.eenet.ee [193.40.55.7] with 32 bytes of data:  
Reply from 193.40.55.7: bytes=32 time=276ms TTL=53  
Reply from 193.40.55.7: bytes=32 time=276ms TTL=53  
Reply from 193.40.55.7: bytes=32 time=277ms TTL=53  
Reply from 193.40.55.7: bytes=32 time=276ms TTL=53  
  
Ping statistics for 193.40.55.7:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 276ms, Maximum = 277ms, Average = 276ms
```

Standard Deviation Calculator

Result

Standard Deviation, s: **0**

Count, N: 3
Sum, Σx : 828
Mean, \bar{x} : 276
Variance, s^2 : 0

Steps

$$\begin{aligned}s &= \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}, \\ s^2 &= \frac{\sum (x_i - \bar{x})^2}{N-1} \\ &= \frac{(276 - 276)^2 + \dots + (276 - 276)^2}{3-1} \\ &= \frac{0}{2} \\ &= 0 \\ s &= \sqrt{0} \\ &= 0\end{aligned}$$

Section 4:

a) iperf.he.net

```
Sakshi@sakshi-PC MINGW64 /f
$ iperf3 -c iperf.he.net 5002
Connecting to host iperf.he.net, port 5201
[ 4] local 192.168.1.4 port 50963 connected to 216.218.207.42 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-1.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  1.00-2.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  2.00-3.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  3.00-4.00   sec      0.00 Bytes  0.00 bits/sec
[ 4]  4.00-5.00   sec      0.00 Bytes  0.00 bits/sec
[ 4]  5.00-6.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  6.00-7.00   sec    384 KBytes  3.15 Mbits/sec
[ 4]  7.00-8.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  8.00-9.00   sec    384 KBytes  3.14 Mbits/sec
[ 4]  9.00-10.00  sec    384 KBytes  3.14 Mbits/sec
[ -----]
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-10.00  sec   2.38 MBytes  1.99 Mbits/sec
[ 4]  0.00-10.00  sec   2.29 MBytes  1.92 Mbits/sec
                                         sender
                                         receiver
iperf Done.

Sakshi@sakshi-PC MINGW64 /f
$ iperf3 -c iperf.he.net 5002
Connecting to host iperf.he.net, port 5201
[ 4] local 192.168.1.4 port 50963 connected to 216.218.207.42 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-1.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  1.00-2.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  2.00-3.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  3.00-4.00   sec      0.00 Bytes  0.00 bits/sec
[ 4]  4.00-5.00   sec      0.00 Bytes  0.00 bits/sec
[ 4]  5.00-6.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  6.00-7.00   sec    384 KBytes  3.15 Mbits/sec
[ 4]  7.00-8.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  8.00-9.00   sec    384 KBytes  3.14 Mbits/sec
[ 4]  9.00-10.00  sec    384 KBytes  3.14 Mbits/sec
[ -----]
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-10.00  sec   2.38 MBytes  1.99 Mbits/sec
[ 4]  0.00-10.00  sec   2.29 MBytes  1.92 Mbits/sec
                                         sender
                                         receiver
iperf Done.

Sakshi@sakshi-PC MINGW64 /f
$ iperf3 -c iperf.he.net 5002
Connecting to host iperf.he.net, port 5201
[ 4] local 192.168.1.4 port 50963 connected to 216.218.207.42 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-1.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  1.00-2.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  2.00-3.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  3.00-4.00   sec      0.00 Bytes  0.00 bits/sec
[ 4]  4.00-5.00   sec      0.00 Bytes  0.00 bits/sec
[ 4]  5.00-6.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  6.00-7.00   sec    384 KBytes  3.15 Mbits/sec
[ 4]  7.00-8.00   sec    256 KBytes  2.10 Mbits/sec
[ 4]  8.00-9.00   sec    384 KBytes  3.14 Mbits/sec
[ 4]  9.00-10.00  sec    384 KBytes  3.14 Mbits/sec
[ -----]
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-10.00  sec   2.38 MBytes  1.99 Mbits/sec
[ 4]  0.00-10.00  sec   2.29 MBytes  1.92 Mbits/sec
                                         sender
                                         receiver
iperf Done.
```

b) bouygues.testdebit.info

```
Sakshi@sakshi-PC MINGW64 /f/iperf-2.0.9-win64
$ iperf -c bouygues.testdebit.info
-----
Client connecting to bouygues.testdebit.info, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50767 connected with 89.84.1.222 port 5001
write failed: Broken pipe
[ ID] Interval           Transfer     Bandwidth
[ 3]  0.0- 0.4 sec    256 KBytes  5.87 Mbits/sec
```

```
sakshi@sakshi-PC MINGW64 /f/iperf-2.0.9-win64
$ iperf -c bouygues.testdebit.info
-----
client connecting to bouygues.testdebit.info, TCP port 5001
TCP window size: 208 KByte (default)
[ 3] local 192.168.1.4 port 50765 connected with 89.84.1.222 port 5001
write failed: Broken pipe
[ ID] Interval      Transfer     Bandwidth
[ 3] 0.0- 0.4 sec   256 KBytes   5.02 Mbits/sec

Sakshi@sakshi-PC MINGW64 /f/iperf-2.0.9-win64
$ iperf -c bouygues.testdebit.info
-----
Client connecting to bouygues.testdebit.info, TCP port 5001
TCP window size: 208 KByte (default)
[ 3] local 192.168.1.4 port 50768 connected with 89.84.1.222 port 5001
write failed: Broken pipe
[ ID] Interval      Transfer     Bandwidth
[ 3] 0.0- 0.4 sec   256 KBytes   5.24 Mbits/sec
```

c) iperf.comneonext.de

```
sakshi@sakshi-PC MINGW64 /f
$ iperf -c iperf.comneonext.de -p 80
-----
client connecting to iperf.comneonext.de, TCP port 80
TCP window size: 64.0 KByte (default)
[584] local 192.168.1.4 port 60962 connected with 91.195.241.136 port 80
[ ID] Interval      Transfer     Bandwidth
[584] 0.0- 0.4 sec   256 KBytes   5.11 Mbits/sec
write failed: Connection reset by peer

sakshi@sakshi-PC MINGW64 /f
$ iperf -c iperf.comneonext.de -p 80
-----
Client connecting to iperf.comneonext.de, TCP port 80
TCP window size: 64.0 KByte (default)
[584] local 192.168.1.4 port 60974 connected with 91.195.241.136 port 80
[ ID] Interval      Transfer     Bandwidth
[584] 0.0- 0.5 sec   256 KBytes   4.47 Mbits/sec
write failed: Connection reset by peer

sakshi@sakshi-PC MINGW64 /f
$ iperf -c iperf.comneonext.de -p 80
-----
Client connecting to iperf.comneonext.de, TCP port 80
TCP window size: 64.0 KByte (default)
[592] local 192.168.1.4 port 60975 connected with 91.195.241.136 port 80
[ ID] Interval      Transfer     Bandwidth
[592] 0.0- 0.4 sec   256 KBytes   5.12 Mbits/sec
```

d) ikoula.testdebit.info

```
Sakshi@Sakshi-PC MINGW64 /f
$ iperf -c ikoula.testdebit.info
-----
Client connecting to ikoula.testdebit.info, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50812 connected with 213.246.63.45 port 5001
write failed: Broken pipe
[ ID] Interval      Transfer     Bandwidth
[ 3]  0.0- 0.3 sec   256 KBytes   6.47 Mbits/sec
Sakshi@Sakshi-PC MINGW64 /f
$ iperf -c ikoula.testdebit.info
-----
Client connecting to ikoula.testdebit.info, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50809 connected with 213.246.63.45 port 5001
write failed: Broken pipe
[ ID] Interval      Transfer     Bandwidth
[ 3]  0.0- 0.4 sec   256 KBytes   5.12 Mbits/sec
Sakshi@Sakshi-PC MINGW64 /f
$ iperf -c ikoula.testdebit.info
-----
Client connecting to ikoula.testdebit.info, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50813 connected with 213.246.63.45 port 5001
write failed: Broken pipe
[ ID] Interval      Transfer     Bandwidth
[ 3]  0.0- 0.4 sec   256 KBytes   4.85 Mbits/sec
```

e) st2.nn.ertelecom.ru

```
Sakshi@Sakshi-PC MINGW64 /f
$ iperf -c st2.nn.ertelecom.ru
-----
Client connecting to st2.nn.ertelecom.ru, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50818 connected with 91.144.184.232 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 3]  0.0-10.1 sec  3.38 MBytes  2.80 Mbits/sec
Sakshi@Sakshi-PC MINGW64 /f
$ iperf -c st2.nn.ertelecom.ru
-----
Client connecting to st2.nn.ertelecom.ru, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50815 connected with 91.144.184.232 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 3]  0.0-10.2 sec  3.75 MBytes  3.10 Mbits/sec
Sakshi@Sakshi-PC MINGW64 /f
$ iperf -c st2.nn.ertelecom.ru
-----
Client connecting to st2.nn.ertelecom.ru, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50826 connected with 91.144.184.232 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 3]  0.0-10.3 sec  1.62 MBytes  1.33 Mbits/sec
```

f) iperf.biznetnetworks.com

```
Sakshi@sakshi-PC MINGW64 /f
$ iperf3 -c iperf.biznetnetworks.com
Connecting to host iperf.biznetnetworks.com, port 5201
[ 4] local 192.168.1.4 port 50851 connected to 117.102.109.186 port 5201
[ ID] Interval Transfer Bandwidth
[ 4] 0.00-1.00 sec 256 KBytes 2.10 Mbits/sec
[ 4] 1.00-2.00 sec 256 KBytes 2.09 Mbits/sec
[ 4] 2.00-3.00 sec 256 KBytes 2.10 Mbits/sec
[ 4] 3.00-4.00 sec 0.00 Bytes 0.00 bits/sec
[ 4] 4.00-5.00 sec 128 KBytes 1.05 Mbits/sec
[ 4] 5.00-6.00 sec 128 KBytes 1.05 Mbits/sec
[ 4] 6.00-7.00 sec 256 KBytes 2.10 Mbits/sec
[ 4] 7.00-8.00 sec 128 KBytes 1.05 Mbits/sec
[ 4] 8.00-9.00 sec 256 KBytes 2.09 Mbits/sec
[ 4] 9.00-10.00 sec 256 KBytes 2.10 Mbits/sec
- - - - -
[ ID] Interval Transfer Bandwidth
[ 4] 0.00-10.00 sec 1.88 MBytes 1.57 Mbits/sec
[ 4] 0.00-10.00 sec 1.77 MBytes 1.48 Mbits/sec
sender receiver

iperf Done.

Sakshi@sakshi-PC MINGW64 /f
$ iperf3 -c iperf.biznetnetworks.com
Connecting to host iperf.biznetnetworks.com, port 5201
[ 4] local 192.168.1.4 port 50837 connected to 117.102.109.186 port 5201
[ ID] Interval Transfer Bandwidth
[ 4] 0.00-1.00 sec 256 KBytes 2.09 Mbits/sec
[ 4] 1.00-2.00 sec 256 KBytes 2.10 Mbits/sec
[ 4] 2.00-3.00 sec 384 KBytes 3.14 Mbits/sec
[ 4] 3.00-4.00 sec 256 KBytes 2.10 Mbits/sec
[ 4] 4.00-5.00 sec 256 KBytes 2.10 Mbits/sec
[ 4] 5.00-6.00 sec 256 KBytes 2.10 Mbits/sec
[ 4] 6.00-7.00 sec 256 KBytes 2.10 Mbits/sec
[ 4] 7.00-8.00 sec 256 KBytes 2.10 Mbits/sec
[ 4] 8.00-9.00 sec 384 KBytes 3.15 Mbits/sec
[ 4] 9.00-10.00 sec 256 KBytes 2.10 Mbits/sec
- - - - -
[ ID] Interval Transfer Bandwidth
[ 4] 0.00-10.00 sec 2.75 MBytes 2.31 Mbits/sec
[ 4] 0.00-10.00 sec 2.69 MBytes 2.25 Mbits/sec
sender receiver

iperf Done.

SAKSHI@SAKSHI-PC MINGW64 /T
$ iperf3 -c iperf.biznetnetworks.com
Connecting to host iperf.biznetnetworks.com, port 5201
[ 4] local 192.168.1.4 port 50857 connected to 117.102.109.186 port 5201
[ ID] Interval Transfer Bandwidth
[ 4] 0.00-1.00 sec 256 KBytes 2.10 Mbits/sec
[ 4] 1.00-2.00 sec 128 KBytes 1.05 Mbits/sec
[ 4] 2.00-3.01 sec 128 KBytes 1.04 Mbits/sec
[ 4] 3.01-4.00 sec 256 KBytes 2.12 Mbits/sec
[ 4] 4.00-5.00 sec 256 KBytes 2.10 Mbits/sec
[ 4] 5.00-6.00 sec 256 KBytes 2.10 Mbits/sec
[ 4] 6.00-7.00 sec 384 KBytes 3.15 Mbits/sec
[ 4] 7.00-8.00 sec 384 KBytes 3.15 Mbits/sec
[ 4] 8.00-9.00 sec 384 KBytes 3.14 Mbits/sec
[ 4] 9.00-10.00 sec 384 KBytes 3.15 Mbits/sec
- - - - -
[ ID] Interval Transfer Bandwidth
[ 4] 0.00-10.00 sec 2.75 MBytes 2.31 Mbits/sec
[ 4] 0.00-10.00 sec 2.72 MBytes 2.28 Mbits/sec
sender receiver

iperf Done.
```

g) iperf.scottlinux.com

```
Sakshi@sakshi-PC MINGW64 /f
$ iperf -c iperf.scottlinux.com -p 80
-----
Client connecting to iperf.scottlinux.com, TCP port 80
TCP window size: 64.0 KByte (default)
-----
[584] local 192.168.1.4 port 60941 connected with 45.33.39.39 port 80
[ ID] Interval Transfer Bandwidth
[584] 0.0-12.1 sec 1.00 MBytes 694 Kbytes/sec

Sakshi@sakshi-PC MINGW64 /f
$ iperf -c iperf.scottlinux.com -p 80
-----
Client connecting to iperf.scottlinux.com, TCP port 80
TCP window size: 64.0 KByte (default)
-----
[588] local 192.168.1.4 port 60944 connected with 45.33.39.39 port 80
[ ID] Interval Transfer Bandwidth
[588] 0.0-10.3 sec 1.13 MBytes 920 Kbytes/sec

Sakshi@sakshi-PC MINGW64 /f
$ iperf -c iperf.scottlinux.com -p 80
-----
Client connecting to iperf.scottlinux.com, TCP port 80
TCP window size: 64.0 KByte (default)
-----
[584] local 192.168.1.4 port 60947 connected with 45.33.39.39 port 80
[ ID] Interval Transfer Bandwidth
[584] 0.0-10.2 sec 1.63 MBytes 1.33 Mbytes/sec
```

h) speedtest.serverius.net

```
Sakshi@sakshi-PC MINGW64 /f
$ iperf -c speedtest.serverius.net
-----
Client connecting to speedtest.serverius.net, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50898 connected with 178.21.16.76 port 5001
write failed: Connection reset by peer
[ ID] Interval Transfer Bandwidth
[ 3] 0.0-27.9 sec 256 KBytes 75.1 Kbytes/sec

Sakshi@sakshi-PC MINGW64 /f
$ iperf -c speedtest.serverius.net
-----
Client connecting to speedtest.serverius.net, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50879 connected with 178.21.16.76 port 5001
write failed: Connection reset by peer
[ ID] Interval Transfer Bandwidth
[ 3] 0.0-30.1 sec 256 KBytes 69.6 Kbytes/sec

Sakshi@sakshi-PC MINGW64 /f
$ iperf -c speedtest.serverius.net
-----
Client connecting to speedtest.serverius.net, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50901 connected with 178.21.16.76 port 5001
write failed: Connection reset by peer
[ ID] Interval Transfer Bandwidth
[ 3] 0.0-31.6 sec 256 KBytes 66.3 Kbytes/sec
```

i) iperf.volia.net

```
Sakshi@sakshi-PC MINGW64 /f/iperf-2.0.9-win64
$ iperf -c iperf.volia.net
-----
Client connecting to iperf.volia.net, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50729 connected with 77.120.3.236 port 5001
write failed: Connection reset by peer
[ ID] Interval      Transfer     Bandwidth
[ 3] 0.0-38.5 sec   256 KBytes   54.4 Kbits/sec

Sakshi@sakhi-PC MINGW64 /f/iperf-2.0.9-win64
$ iperf -c iperf.volia.net
-----
client connecting to iperf.volia.net, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50717 connected with 77.120.3.236 port 5001
write failed: Connection reset by peer
[ ID] Interval      Transfer     Bandwidth
[ 3] 0.0-34.0 sec   256 KBytes   61.7 Kbits/sec
Sakshi@sakhi-PC MINGW64 /f/iperf-2.0.9-win64
$ iperf -c iperf.volia.net
-----
client connecting to iperf.volia.net, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 192.168.1.4 port 50730 connected with 77.120.3.236 port 5001
write failed: Connection reset by peer
[ ID] Interval      Transfer     Bandwidth
[ 3] 0.0-33.0 sec   256 KBytes   63.5 Kbits/sec
```

j) iperf.eenet.ee

```
Sakshi@sakhi-PC MINGW64 /f
$ iperf -c iperf.eenet.ee -p 80
-----
client connecting to iperf.eenet.ee, TCP port 80
TCP window size: 64.0 KByte (default)
-----
[488] local 192.168.1.4 port 52540 connected with 193.40.55.7 port 80
[ ID] Interval      Transfer     Bandwidth
[488] 0.0-10.1 sec  1.63 MBytes  1.35 Mbits/sec

Sakshi@sakhi-PC MINGW64 /f
$ iperf -c iperf.eenet.ee -p 80
-----
client connecting to iperf.eenet.ee, TCP port 80
TCP window size: 64.0 KByte (default)
-----
[588] local 192.168.1.4 port 52541 connected with 193.40.55.7 port 80
[ ID] Interval      Transfer     Bandwidth
[588] 0.0-10.2 sec  3.13 MBytes  2.57 Mbits/sec

Sakshi@sakhi-PC MINGW64 /f
$ iperf -c iperf.eenet.ee -p 80
-----
client connecting to iperf.eenet.ee, TCP port 80
TCP window size: 64.0 KByte (default)
-----
[592] local 192.168.1.4 port 52542 connected with 193.40.55.7 port 80
[ ID] Interval      Transfer     Bandwidth
[592] 0.0-10.3 sec  1.38 MBytes  1.12 Mbits/sec
```


Week 7: Transport Layer

Internet Technologies COMP90007

Lecturer: Muhammad Usman

Semester 2, 2020

Introduction to Instructor

Dr Muhammad Usman

Since 2019: Senior Lecturer at School of Computing and Information Systems

Since 2014: at the University of Melbourne

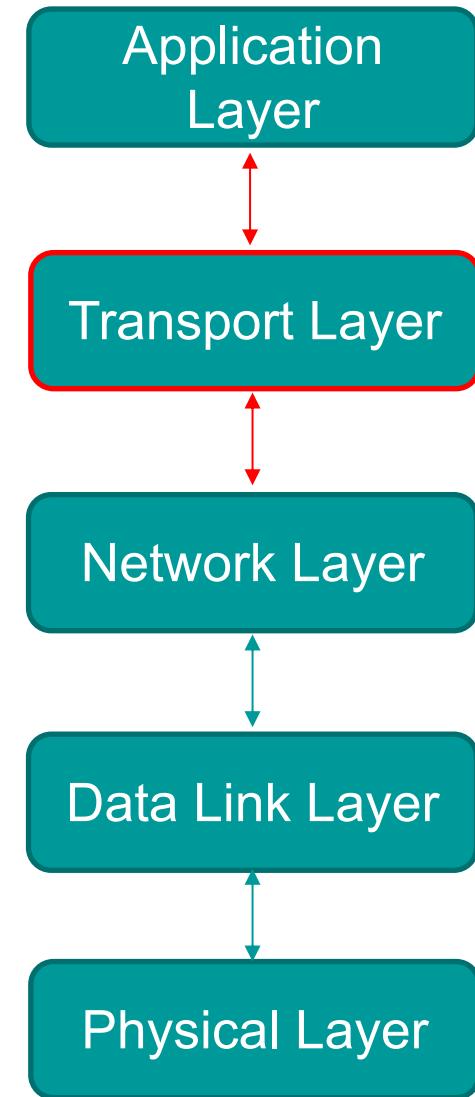
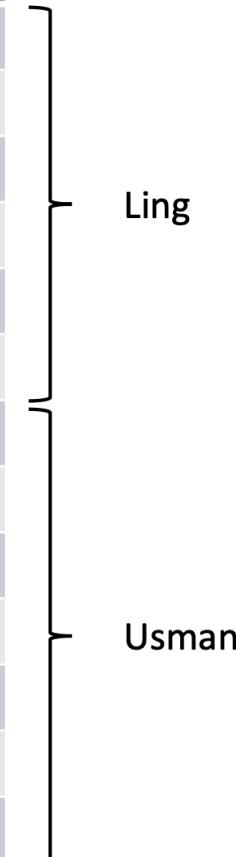
2010: PhD in Electrical Engineering from Purdue University USA

Areas of Interest: Quantum Computing, Nanoelectronics, Machine Learning

Layered Network

Tentative Schedule

Week	Topic
1	Introduction
2	Physical Layer
3	Data Link Layer
4	Medium Access Control
5	Network Layer
6	Network Layer
7	Transport Layer
8	Transport Layer
9	Application Layer
	Non-teaching period
10	Application Layer
11	Network Security
12	Review



Transport Layer Function

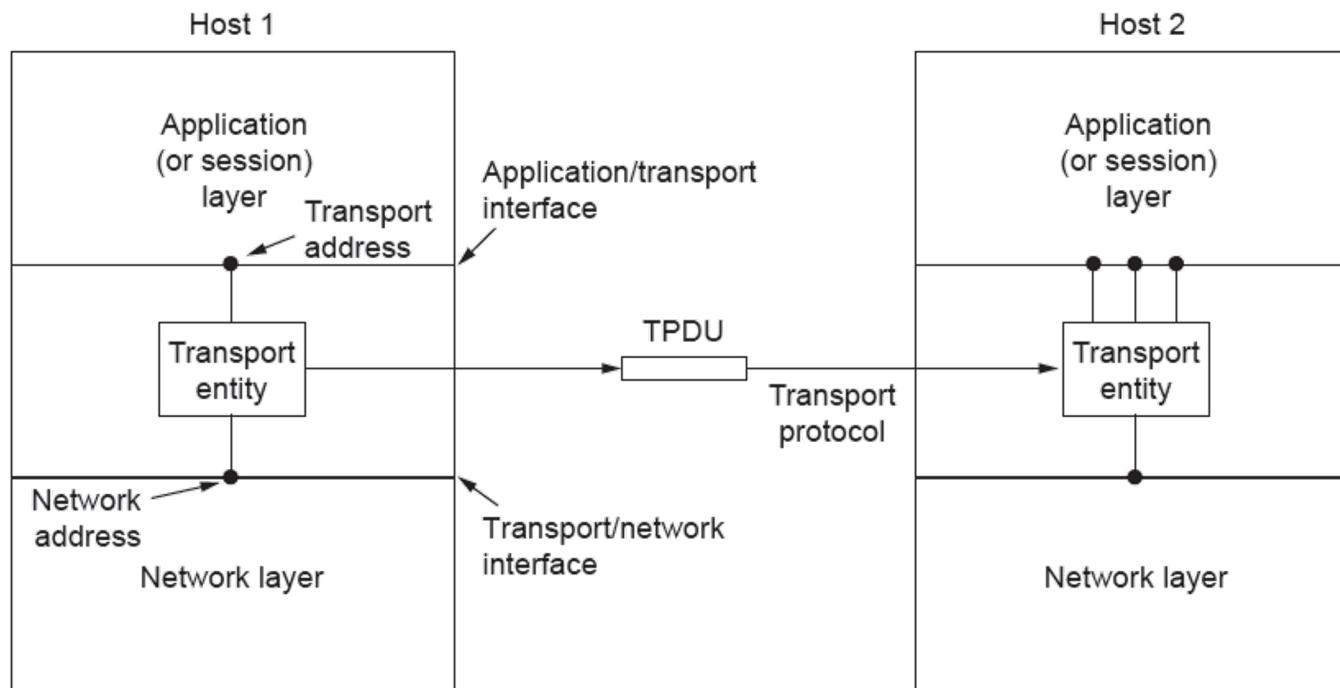
- Main function
 - provide efficient, reliable & cost-effective data transmission service to the processes in the application layer...independent of physical or data networks
- Recall: To Achieve this
 - It calls services provided by the network layer

Transport Layer Services

- Transport Layer **Services** provide interfaces between the Application Layer and the Network Layer
- Transport **Entities** (the hardware or software which actually does the work) can exist in multiple locations:
- **Where and where it should not be (but sometimes is)?**
 - OS kernel
 - System library (library package bound into network applications)
- Not so much...
 - User process
 - Network interface card

Services Contd.

- Transport layer adds reliability to the network layer
 - Offers connectionless (e.g., UDP) in addition to **connection-oriented** (e.g., TCP) services to applications
- Relationship between network, transport and application layers:



Transport Layer and Network Layer Services Compared

- If Transport and Network layers are so similar, why are there two layers?
- Transport layer code runs entirely on hosts, Network layer code runs almost entirely on routers.....
- *Users have no real control over the network* layer – Transport layer: we can improve QoS
- Transport layer ***fixes reliability problems*** caused by the Network layer (e.g., delayed, lost or duplicated packets)

Position of the Transport Layer

- The Transport Layer occupies a key position in the layer hierarchy because it clearly delineates
 - **providers** of data transmissions services
 - at the network, data link, and physical layers
 - **users** of reliable data transmission services
 - at the application layer
- In particular, **users commonly access connection-oriented transport services** for a reliable service on top of an unreliable network

Example:

Your First Network (Pseudo)Code

```
Socket A_Socket = createSocket("TCP");  
  
connect(A_Socket, 128.255.16.0, 80);  
  
send(A_socket, "My first message!");  
  
disconnect(A_socket);
```

*... there is also a server component for this client
that runs on another host...*

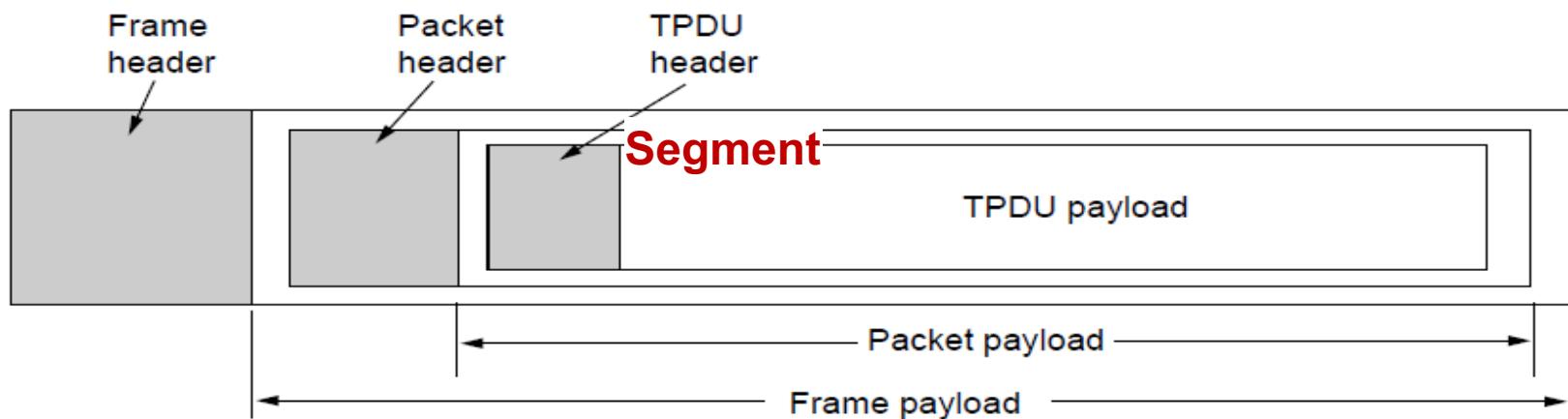
Features of a *Simple* Transport Layer

- Abstraction and primitives provide a **simpler API** for application developers independent of network layer

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Transport Layer Encapsulation

- Abstract representation of messages sent to and from transport entities
 - Transport Protocol Data Unit (TPDU)
- Encapsulation of **TPDUs** transport layer units to network layer units (to frames in data layer units)



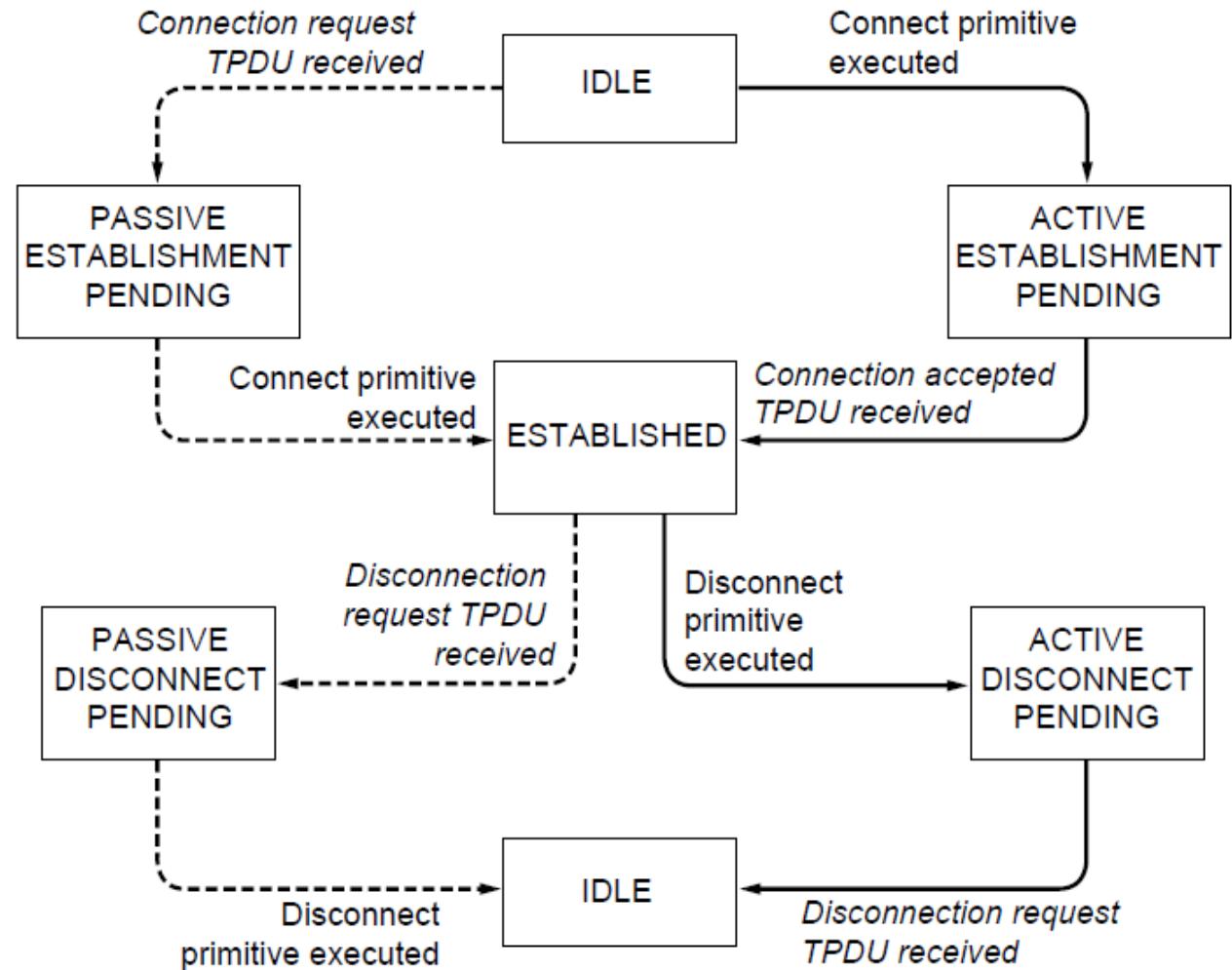
Transport Service Primitives/ Segments

- Primitives that applications might call to transport data for a simple connection-oriented service:
 - Server executes **LISTEN**
 - Client executes **CONNECT**
 - Sends CONNECTION REQUEST TPDU to Server
 - Receives CONNECTION ACCEPTED TPDU to Client
 - Data exchanged using **SEND** and **RECEIVE**
 - Either party executes **DISCONNECT**

Primitive	Segment sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

Simple Connection Illustrated

- Solid lines (right) show client state sequence
- Dashed lines (left) show server state sequence
- Transitions in italics are due to segment arrivals



Elements of Transport Protocols

- Connection establishment
- Connection release
- Addressing

Connection Establishment in the Real World

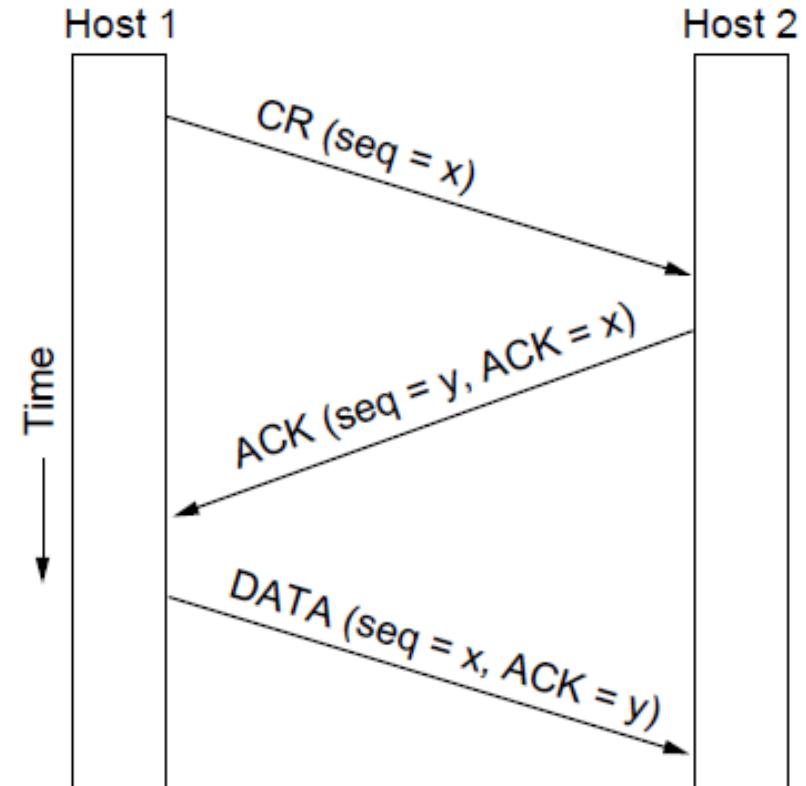
- When networks can **lose, store and duplicate** packets, connection establishment can be complicated
 - congested networks may delay acknowledgements
 - incurring repeated multiple transmissions
 - any of which may not arrive at all or out of sequence – delayed duplicates
 - applications degenerate with such congestion (eg. imagine duplication of bank withdrawals)

Reliable Connection Establishment

- Key challenge is to ensure reliability even though packets may be lost, corrupted, delayed, and duplicated
 - Don't treat an old or duplicate packet as new
 - (Use repeat requests and checksums for loss/corruption)
- Approach:
 - Don't reuse sequence numbers within maximum segment lifetime
 - Use a sequence number space large enough that it will not wrap, even when sending at full rate
 - Three-way handshake for establishing connection..

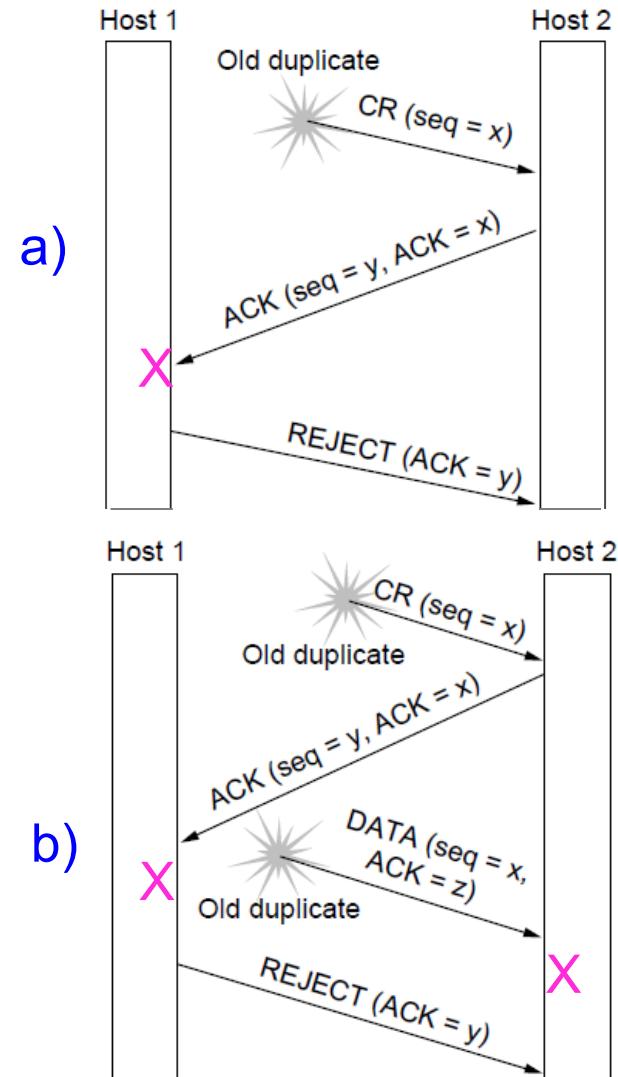
Three Way Handshake

- Three-way handshake used for initial packet
 - Since no state from previous connection
 - Both hosts contribute fresh seq. numbers
 - CR = Connect Request



Three Way Handshake Contd.

- Three-way handshake protects against odd cases:
 - Duplicate CR. Spurious ACK does not connect
 - Duplicate CR and DATA. Same plus DATA will be rejected (wrong ACK).



Connection Release

■ Asymmetric Disconnection

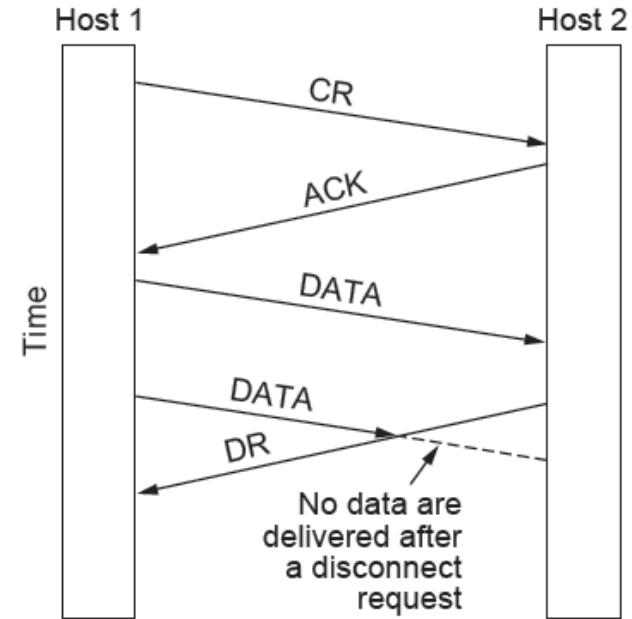
- Either party can issue a DISCONNECT, which results in DISCONNECT TPDU and transmission ends in both directions

■ Symmetric Disconnection

- Both parties issue DISCONNECT, closing only *one direction at a time* - allows flexibility to remain in receive mode

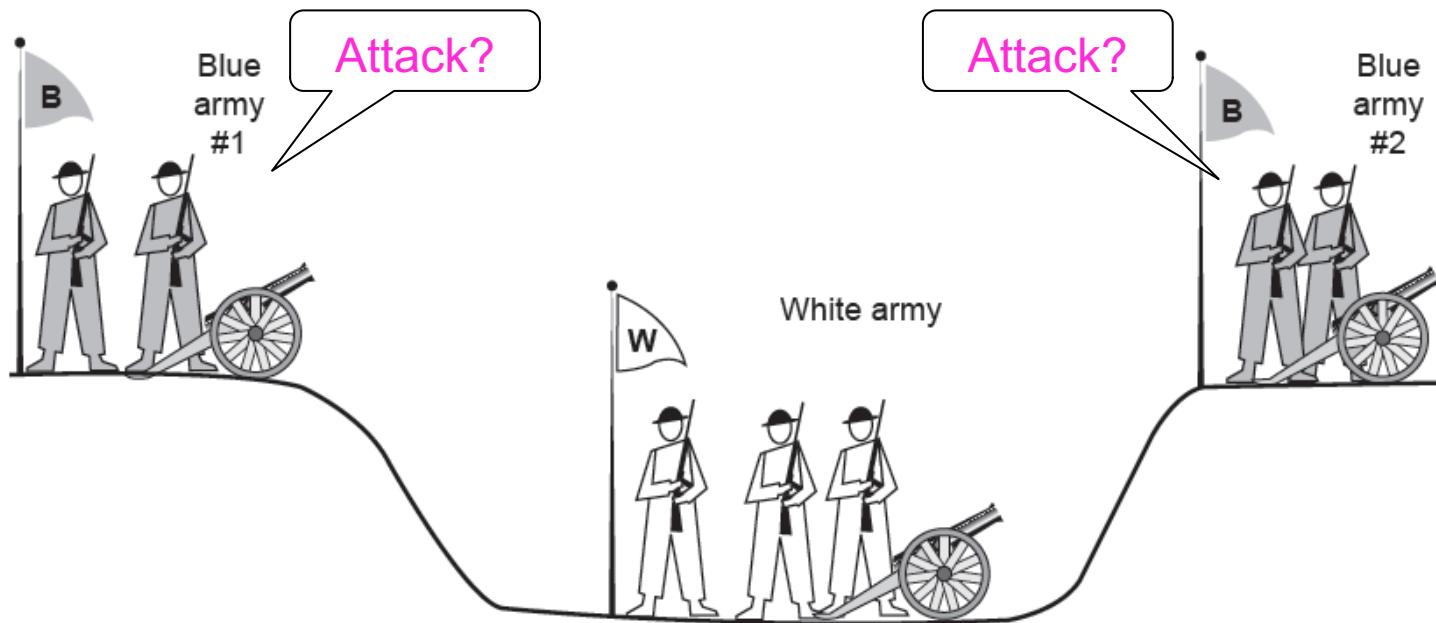
Connection Release (Cont.)

- Asymmetric vs Symmetric connection release types
- **Asymmetric** release may result in data loss hence symmetric release is more attractive
- **Symmetric** release works well where each process has a set amount of data to transmit and knows when it has been sent



Generalizing the Connection Release Problem

- How do we decide the importance of the last message? Is it essential or not?
- No protocol exists which can resolve this ambiguity - Two-army problem shows pitfall of agreement



Week 7: Transport Layer

Internet Technologies COMP90007

Lecturer: Muhammad Usman

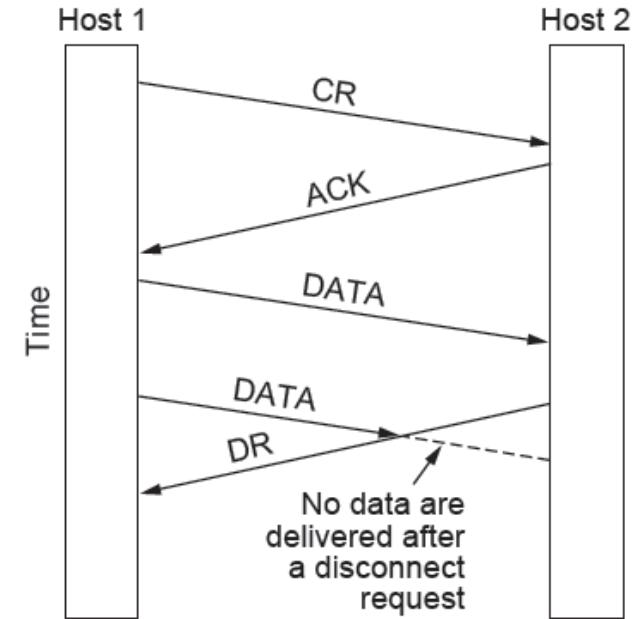
Semester 2, 2020

Connection Release

- **Asymmetric** Disconnection
 - Either party can issue a DISCONNECT, which results in DISCONNECT TPDU and transmission end in both directions
- **Symmetric** Disconnection
 - Both parties issue DISCONNECT, closing only one direction at a time – allows flexibility to remain in receive mode

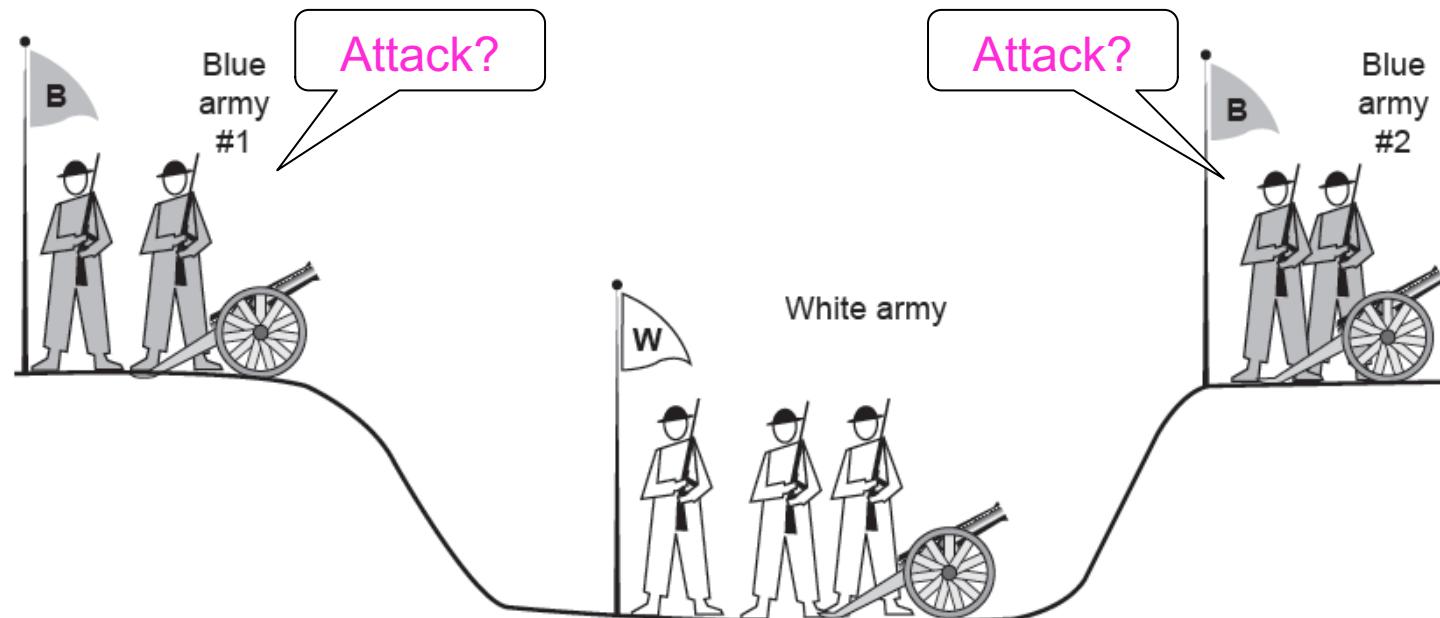
Connection Release (Cont.)

- Asymmetric vs Symmetric connection release types
- **Asymmetric** release may result in data loss hence symmetric release is more attractive
- **Symmetric** release works well where each process has a set amount of data to transmit and knows it has been sent



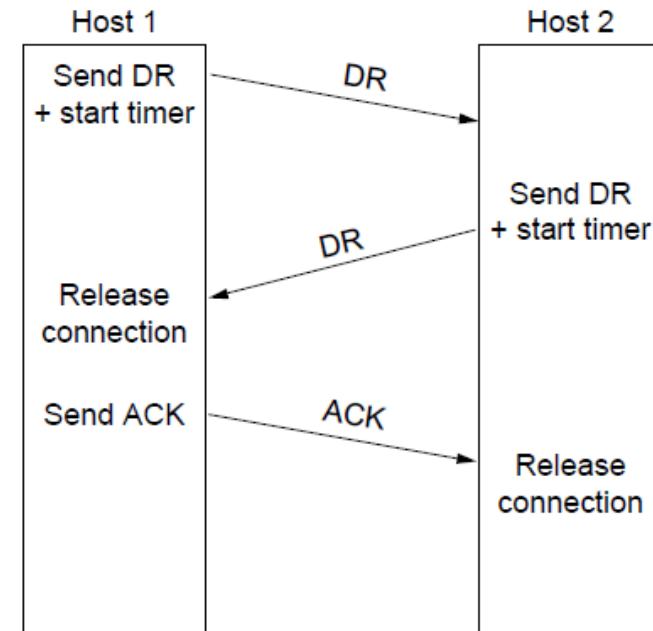
Generalizing the Connection Release Problem

- How do we decide the importance of the last message? Is it essential or not?
- No protocol exists which can resolve this ambiguity
 - Two-army problem shows pitfall of agreement



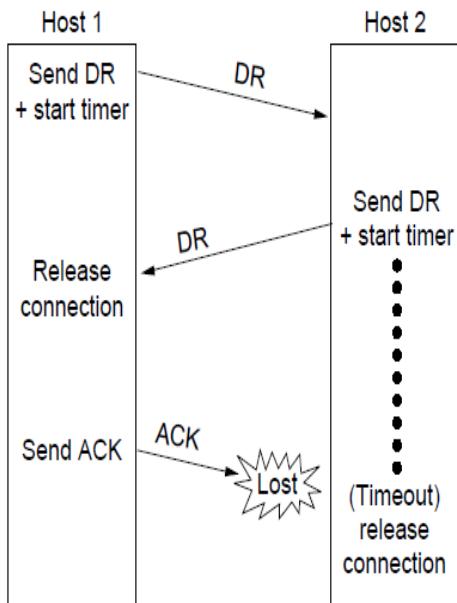
Strategies for Connection Release

- 3 way handshake
- Finite retry
- Timeouts
- Normal release sequence,
initiated by transport user on
Host 1
 - DR=Disconnect Request
 - Both DRs are ACKed by the
other side

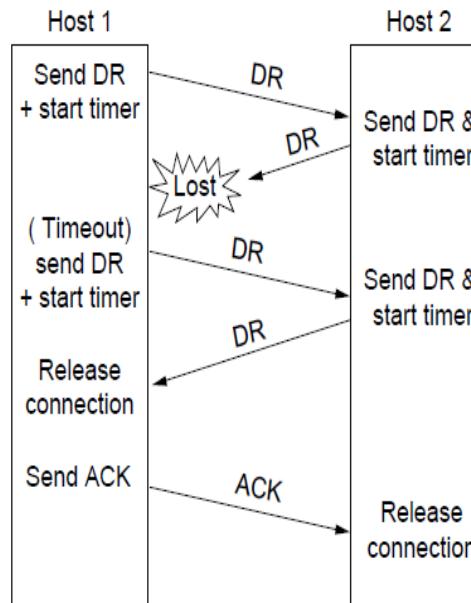


Connection Release (Error Cases)

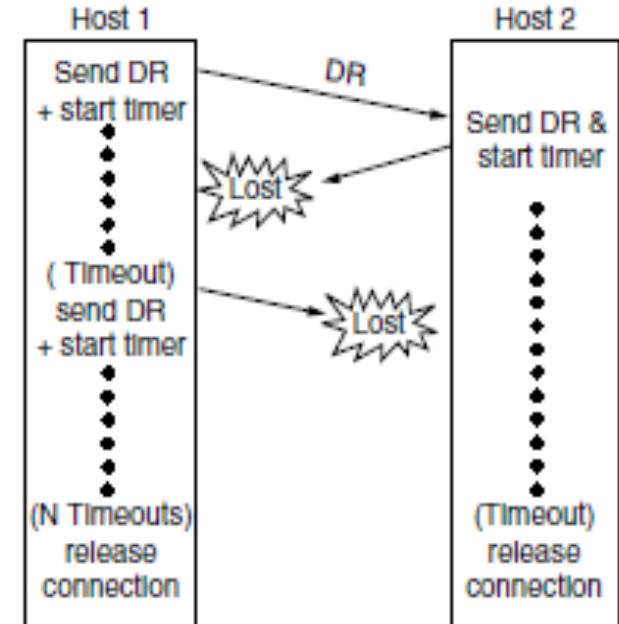
- Error cases are handled with timers and retransmission



Final ACK
lost, Host 2
times out



Lost DR causes
retransmissions

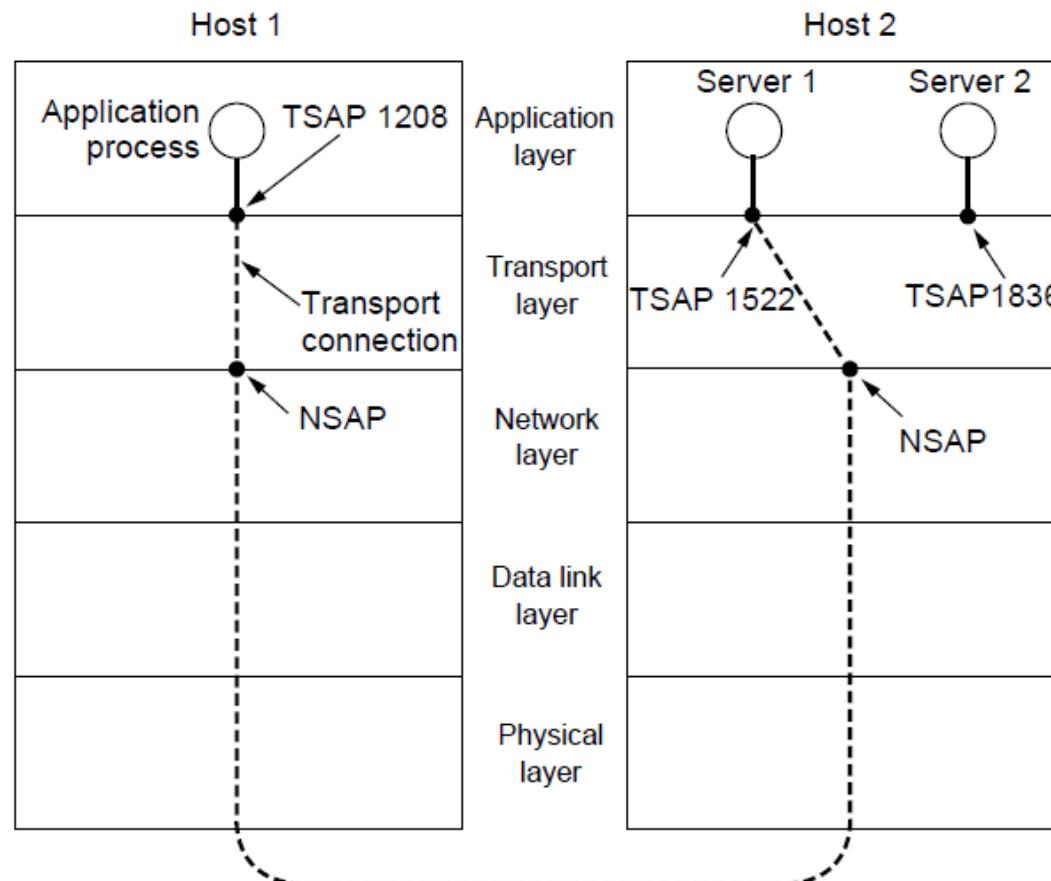


Extreme: Many
lost DRs cause
both hosts to
timeout

Addressing

- Specification of **remote process to connect to** is required at application and transport layers
- Addressing in transport layer is typically done using **Transport Service Access Points** (TSAPs)
 - on the Internet, a TSAP is commonly referred to as a port (e.g. **port** 80)
- Addressing in the network layer is typically done using **Network Service Access Points** (NSAPs)
 - on the Internet, the concept of an NSAP is commonly interpreted as simply an **IP address**

TSAPs, NSAPs and Transport Layer Connections Illustrated



Types of TSAP Allocation

1. Static

- ❑ Well known services have standard allocated TSAPs/ports, which are embedded in OS

2. Directory Assistance – Port-mapper

- ❑ A new service must register itself with the portmapper, giving both its service name and TSAP

3. Mediated

- ❑ A process server intercepts inbound connections and spawns requested server and attaches inbound connection
- ❑ cf. Unix /etc/(x)inetd

Programming using Sockets

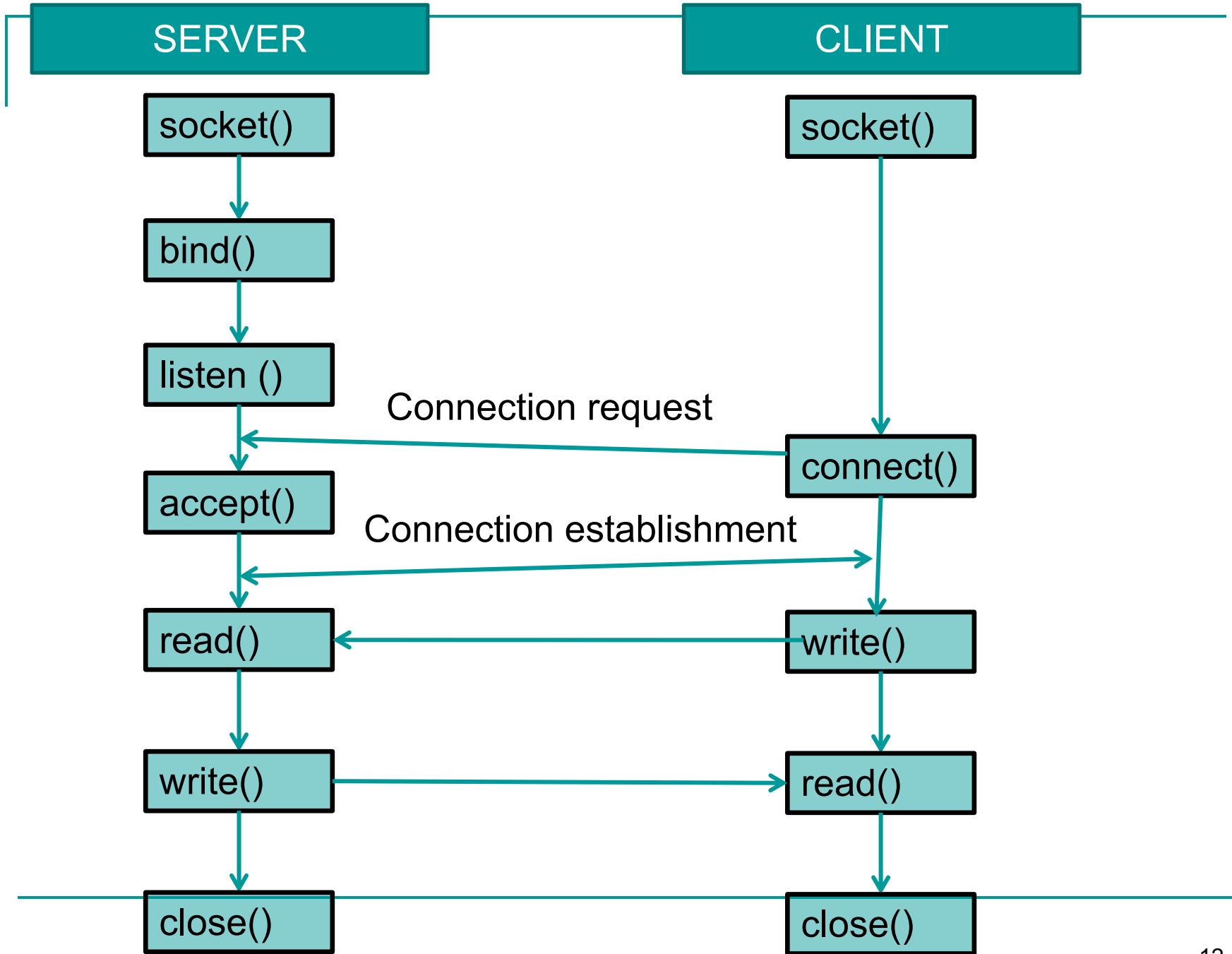
- Sockets widely used for interconnections
 - “Berkeley” sockets are predominant in internet applications
 - Notion of “sockets” as transport endpoints
 - Like the simple set plus SOCKET, BIND, and ACCEPT

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Associate a local address with a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Passively establish an incoming connection
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

Recall Example Pseudo Code

```
Socket A_Socket = createSocket("TCP");  
  
connect(A_Socket, 128.255.16.0, 80);  
  
send(A_socket, "My first message!");  
  
disconnect(A_socket);
```

*... there is also a server component for this client
that runs on another host...*



Let's Look at the Code from the book (in a specific language)

Example from the book has more details but the essence is the same... This is the case in most languages...

```
s = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
if (s <0) fatal("socket");
memset(&channel, 0, sizeof(channel));
channel.sin_family= AF_INET;
memcpy(&channel.sin_addr.s_addr, h->h_addr, h->h_length);
channel.sin_port= htons(SERVER_PORT);
```

```
c = connect(s, (struct sockaddr *) &channel, sizeof(channel));
```

Socket Example – Server Side

Server code. . .

```
memset(&channel, 0, sizeof(channel));
channel.sin_family = AF_INET;
channel.sin_addr.s_addr = htonl(INADDR_ANY);
channel.sin_port = htons(SERVER_PORT);
```

```
s = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
if (s < 0) fatal("socket failed");
setsockopt(s, SOL_SOCKET, SO_REUSEADDR, (char *) &on, sizeof(on));
```

b = bind(s, (struct sockaddr *) &channel, sizeof(channel));
if (b < 0) fatal("bind failed");

```
I = listen(s, QUEUE_SIZE);
if (I < 0) fatal("listen failed");
```

... . . .

Assign address

Prepare for incoming connections

Server Code Contd

```
while (1) {
    sa = accept(s, 0, 0);
    if (sa < 0) fatal("accept failed");
    read(sa, buf, BUF_SIZE);
    /* Get and return the file. */
    fd = open(buf, O_RDONLY);
    if (fd < 0) fatal("open failed");
    ....
```

**Block waiting for
the next
connection**

**Read (receive)
request**

The server can also create a new thread to handle the connection on the new socket and go back to waiting for the next connection on the original socket...

An Example on Multi-Threading

```
ServerSocket serverSocket = new ServerSocket([parameters]);
```

```
While (true) {  
    Socket socket = serverSocket.accept();  
    MultiThreadMyServer server = new MultiThreadMyServer();  
    server.setMyService([some more parameters]);  
    server.setSocket(socket);  
    new Thread(server).start();  
    ....
```

- *(Code from OO Programming with Java; Chp. 14)*

More info on threads...

```
class MultiThreadMyServer extends Thread {  
    int somedata;  
    MultiThreadMyServer() {  
        this.somedata = ...;  
        ...  
    }  
}
```

... more methods here

```
public void run() {  
    ...  
}  
}
```

Looking under the hood for Transport Layer Services...

- The **most basic** is actually connectionless:
 - Called: User Datagram Protocol (UDP)
 - Does not add much to the Network Layer functionality
 - TCP we just does the real-deal for this layer, *reliability*...
 - For UDP: Just remove connection primitives to use it in a program
- **UDP good for:**
 - It is used for apps like video streaming/gaming regularly
- **The reliability issue is left to:**
 - the application layer... retransmission decisions as well as congestion control

New Code: UDP Client...

```
public static void main(String args[]) {  
    ....  
    DatagramSocket mySocket = new  
        DatagramSocket();  
    mySocket.send([data,address, etc  
        parameters]);  
    ....  
}
```

Server Side: UDP Example Contd

```
public static void main(String args[]) {  
    ....  
    DatagramSocket server = new  
        DatagramSocket(port);  
    while (true) {  
        server.receive([parameters]);  
        ....  
    }  
}
```

Week 8: Transport Layer Contd

Internet Technologies COMP90007

Lecturer: Muhammad Usman

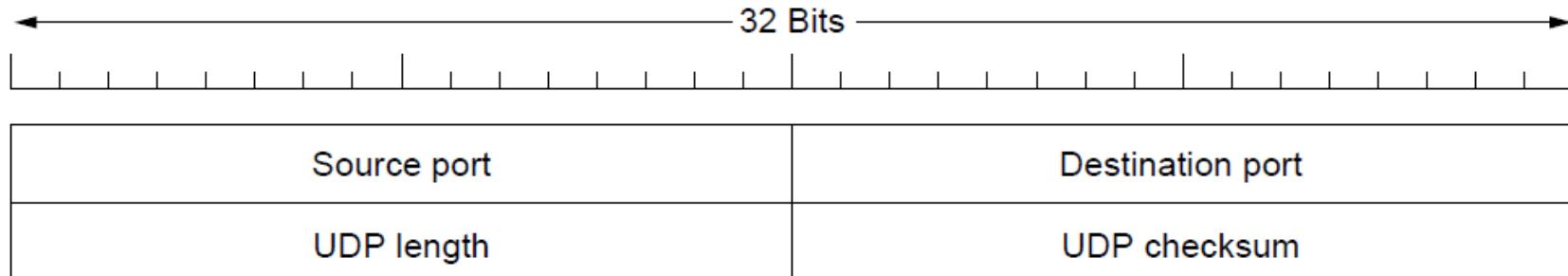
Semester 2, 2020

UDP

- Provides a protocol whereby applications can transmit encapsulated IP datagrams without a connection establishment
- UDP transmits in segments consisting of an 8-byte header followed by the payload
- UDP headers contain source and destination ports
- Payload is handed to the process which is attached to the particular port at destination

UDP Contd.

- Main **advantage** of using UDP over raw IP is:
 - the ability to specify ports for source and destination pairs, i.e., addressing for processes
- Both source and destination ports are required - destination allows for incoming segments, source allows reply for outgoing segments



Structure of UDP header: It has ports (TSAPs), length and checksum

Strengths and Weaknesses of UDP

- **Strengths:** provides an IP interface with multiplexing/demultiplexing capabilities and related transmission efficiencies
- **Weaknesses:** UDP does not include support for flow control, error control/retransmission of bad segments
- **Conclusion:** where applications require a precise level of control over packet flow/error/timing, UDP is a good choice as application layer can make choices
- **Domain Name System over the Internet is a famous user of UDP**

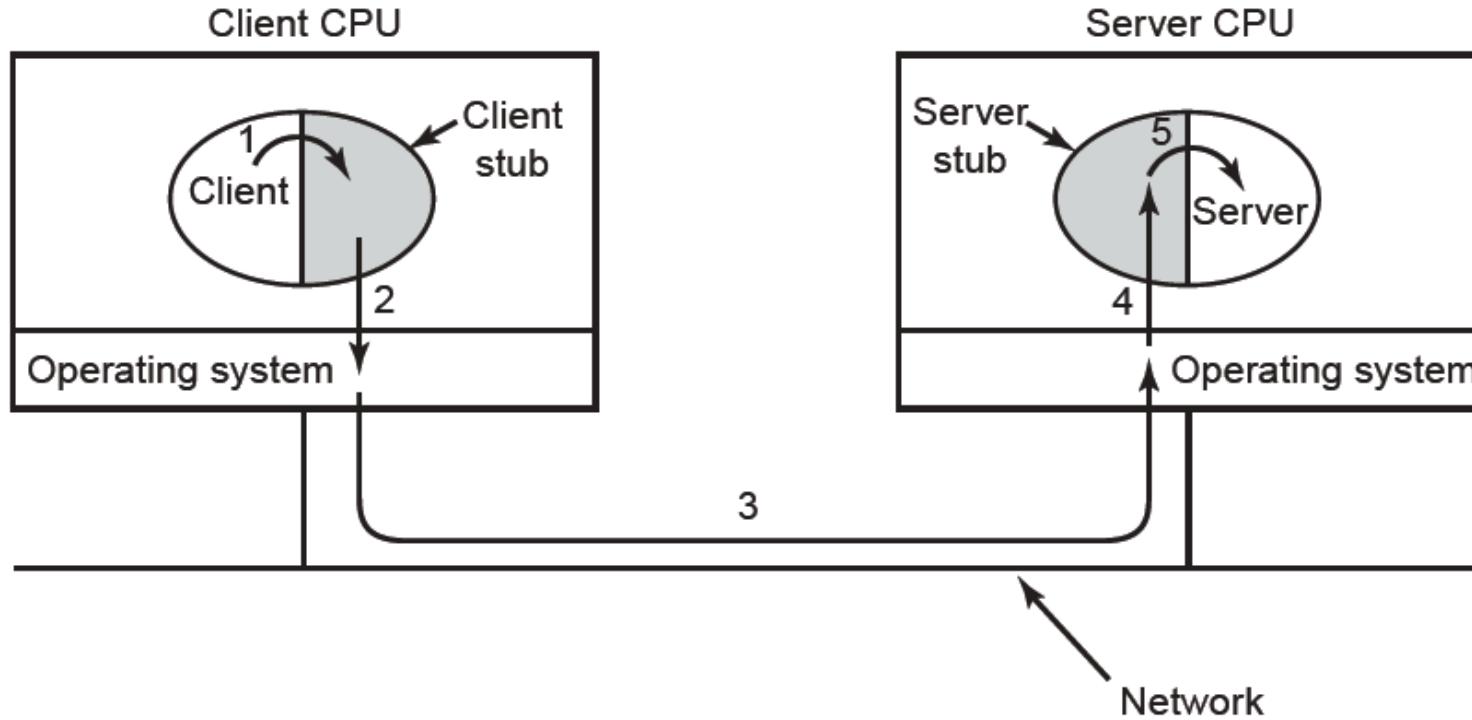
Another one with UDP: Remote Procedure Call (RPC)

- Sending a message and getting a reply back is analogous to making **a function call** in programming languages
- Birrell and Nelson modified this to allow programs to call procedures on remote hosts using UDP
 - **Remote Procedure Call (RPC)**

Remote Procedure Call (RPC)

- To call a remote procedure, the client is bound to a small library (the **client stub**) that represents the server procedure in the client's address space.
- Similarly the server is bound with a procedure called the **server stub**.
- These **stubs hide the fact that the procedure itself is not local.**

RPC Illustrated



Transmission Control Protocol (TCP)

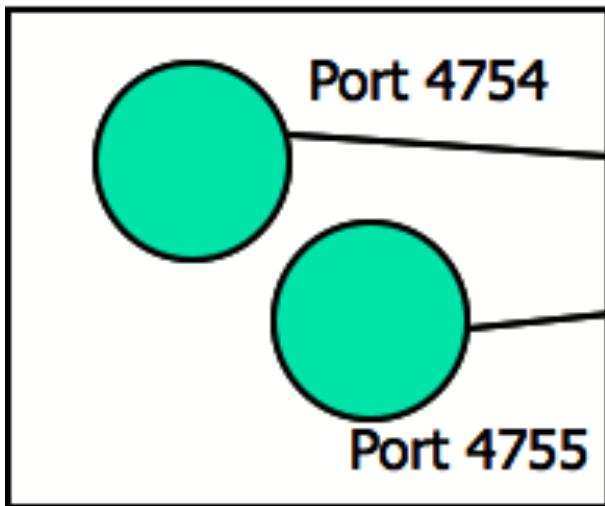
- Provides a protocol by which applications can transmit IP datagrams within a **connection-oriented** framework, thus increasing reliability
- TCP transport entity manages TCP streams and interfaces to the IP layer - can exist in numerous locations (kernel, library, user process)
- **TCP entity** accepts user data streams, and **segments them into pieces < 64KB** (often at a size in order so that the IP and TCP headers can fit into a single Ethernet frame), and sends each piece as a separate IP datagram
- Recipient TCP entities reconstruct the original byte streams from the encapsulation

The TCP Service Model

- Sender and receiver both create **sockets**, consisting of the IP address of the host and a port number as we saw earlier
- For TCP Service to be activated, **connections must be explicitly established between a socket at a sending host** (src-host, src-port) and a socket at a receiving host (dest-host, dest-port)
- Special one-way server sockets may be used for multiple connections simultaneously

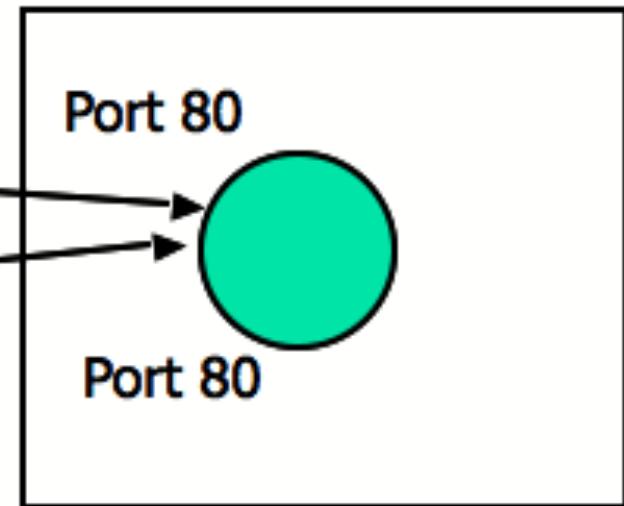
Example

Host 128.42.11.3



Web browser

Host 62.118.44.12



Web server

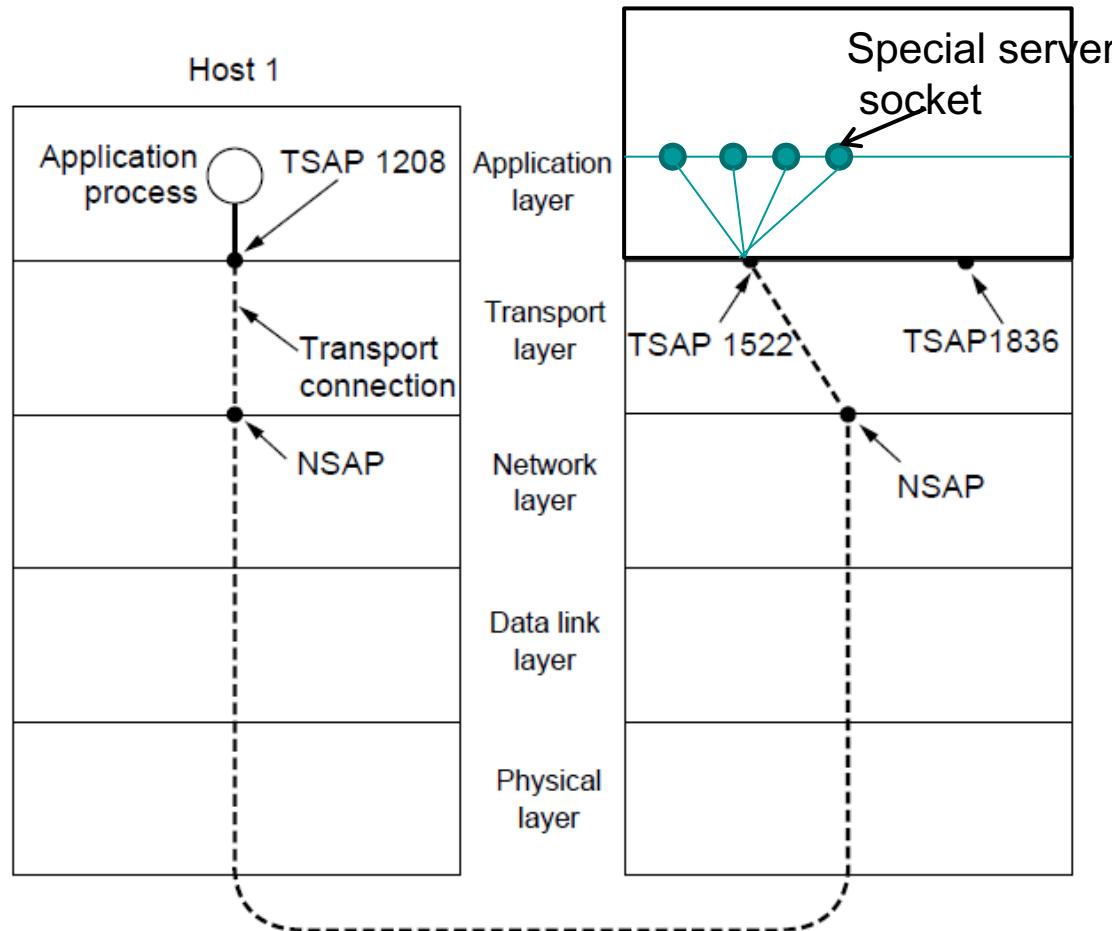
Port Allocations

- Recall TSAPs
- Port numbers can range from 0-65535
- Port numbers are regulated by IANA
(<http://www.iana.org/assignments/port-numbers>)
- Ports are classified into 3 segments:
 - Well Known Ports (0-1023)
 - Registered Ports (1024-49151)
 - Dynamic Ports (49152-65535)

Port	Protocol	Use
20, 21	FTP	File transfer
22	SSH	Remote login, replacement for Telnet
25	SMTP	Email
80	HTTP	World Wide Web
110	POP-3	Remote email access
143	IMAP	Remote email access
443	HTTPS	Secure Web (HTTP over SSL/TLS)
543	RTSP	Media player control
631	IPP	Printer sharing

Socket Library - Multiplexing

- Socket library provides a multiplexing tool on top of TSAPs to allow servers to service multiple clients
- It **simulate** the server using a different port to connect back to the client



Features of TCP Connections

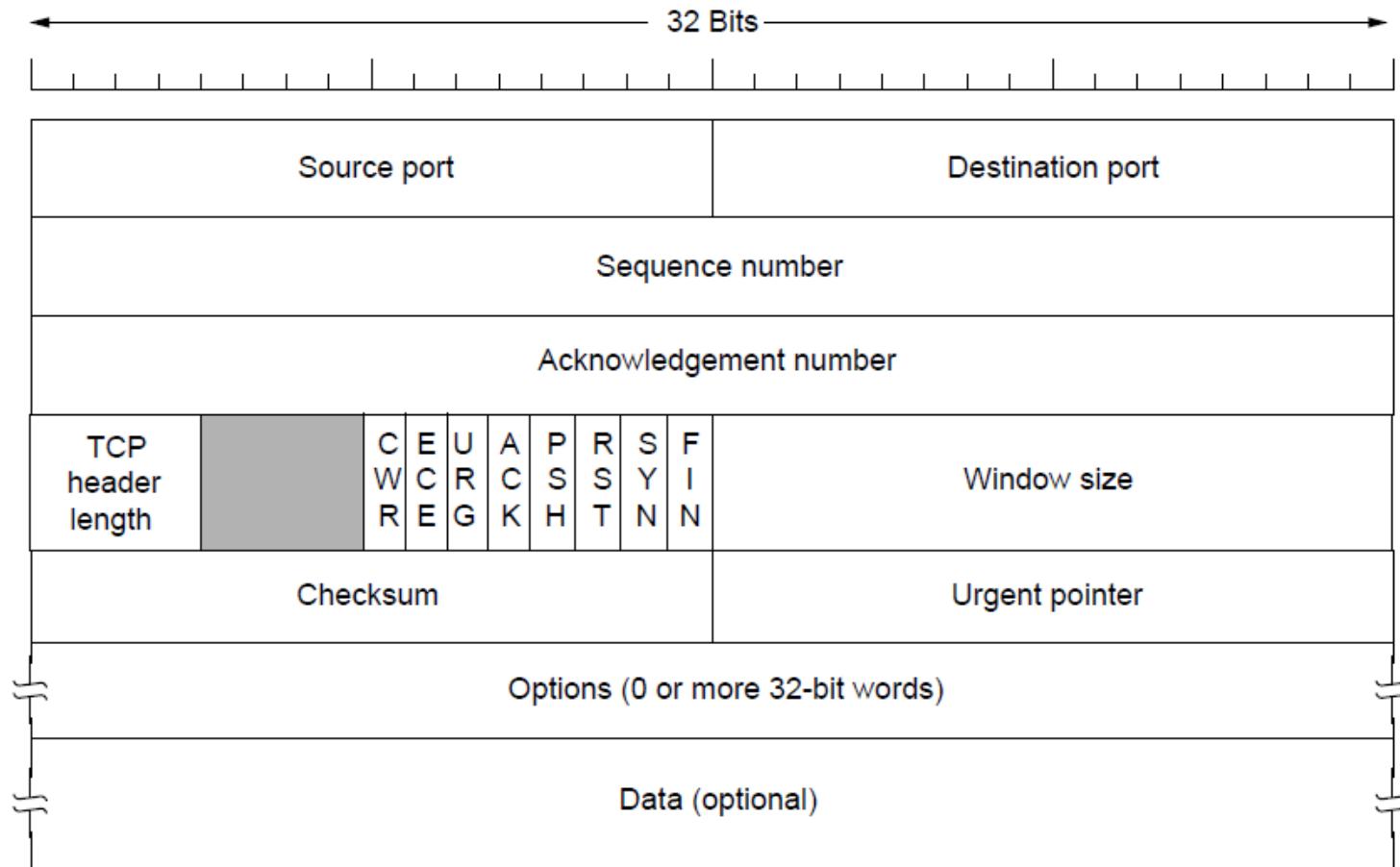
- TCP connections are:
- **Full duplex** - data in both directions simultaneously
- **Point to point** - exact pairs of senders and receivers
- **Byte streams**, not message streams - message boundaries are not preserved
- **Buffer options** - TCP entity can choose to buffer prior to sending or not depending on the context
 - **TCP_NODELAY in Java**
 - **Socket.setTcpNoDelay(boolean)**

TCP Contd

- Data sent between TCP entities in segments - segment has a 20 byte header plus zero or more data bytes
- TCP entities decide how large segments should be mainly with 2 constraints:
 - 65,515 byte IP payload
 - Ethernet unit size - generally 1500 bytes
- **Sliding window** - sender transmits and starts a timer
- Receiver sends back an acknowledgement which is the next sequence number expected - if sender's timer expires before acknowledgement, then the sender transmits the original segment again

The TCP Segment Header

- TCP header includes addressing (ports), sliding window (seq. / ack. number), flow control (window), error control (checksum) and more



The TCP Segment Header

- **Source port and Destination port** fields identify the local end points of the connection
- **Sequence number and Acknowledgement number** fields perform their usual functions
- **TCP header length** tells how many 32-bit words are contained in the TCP header
- **Window size** field tells how many bytes may be sent starting at the byte acknowledged
- **Checksum** is also provided for extra reliability. It checksums the header, the data
- **Options** field provides a way to add extra facilities not covered by the regular header
- **URG** is set to 1 if the *Urgent pointer* is in use. The Urgent pointer is used to indicate a byte offset from the current sequence number at which urgent data are to be found

The TCP Segment Header

- **CWR** and **ECE** are used to signal congestion when *ECN* (Explicit Congestion Notification) is used
- **ECE** is set to signal an ECN-Echo to a TCP sender to tell it to slow down when the TCP receiver gets a congestion indication from the network
- **CWR** is set to signal Congestion Window Reduced from the TCP sender to the TCP receiver so that it knows the sender has slowed down and can stop sending the ECN-Echo
- The **ACK** bit is set to 1 to indicate that the Acknowledgement number is valid. This is the case for nearly all packets. 0 means ignore ACK number field
- **PSH** bit indicates PUSHed data. The receiver is hereby kindly requested to deliver the data to the application upon arrival and not buffer it until a full buffer has been received

The TCP Segment Header

- The **RST** bit is used to abruptly reset a connection that has become confused due to a host crash or some other reason. It is also used to reject an invalid segment or refuse an attempt to open a connection
- The **SYN** bit is used to establish connections. The connection request has SYN = 1 and ACK = 0. The connection reply does bear an acknowledgement, so it has SYN = 1 and ACK = 1.
- In essence, the SYN bit is used to **denote both CONNECTION REQUEST and CONNECTION ACCEPTED**, with the ACK bit used to distinguish between those two possibilities.
- The **FIN** bit is used to release a connection. It specifies that the sender has no more data to transmit. However, after closing a connection, the closing process may continue to receive data.

TCP Connection Establishment and Release

- Connections established **using three-way handshake**
- Two **simultaneous connection attempts results in only one connection** (uniquely identified by end points)
- Connections released with **symmetric release**
- Timers used for lost connection releases

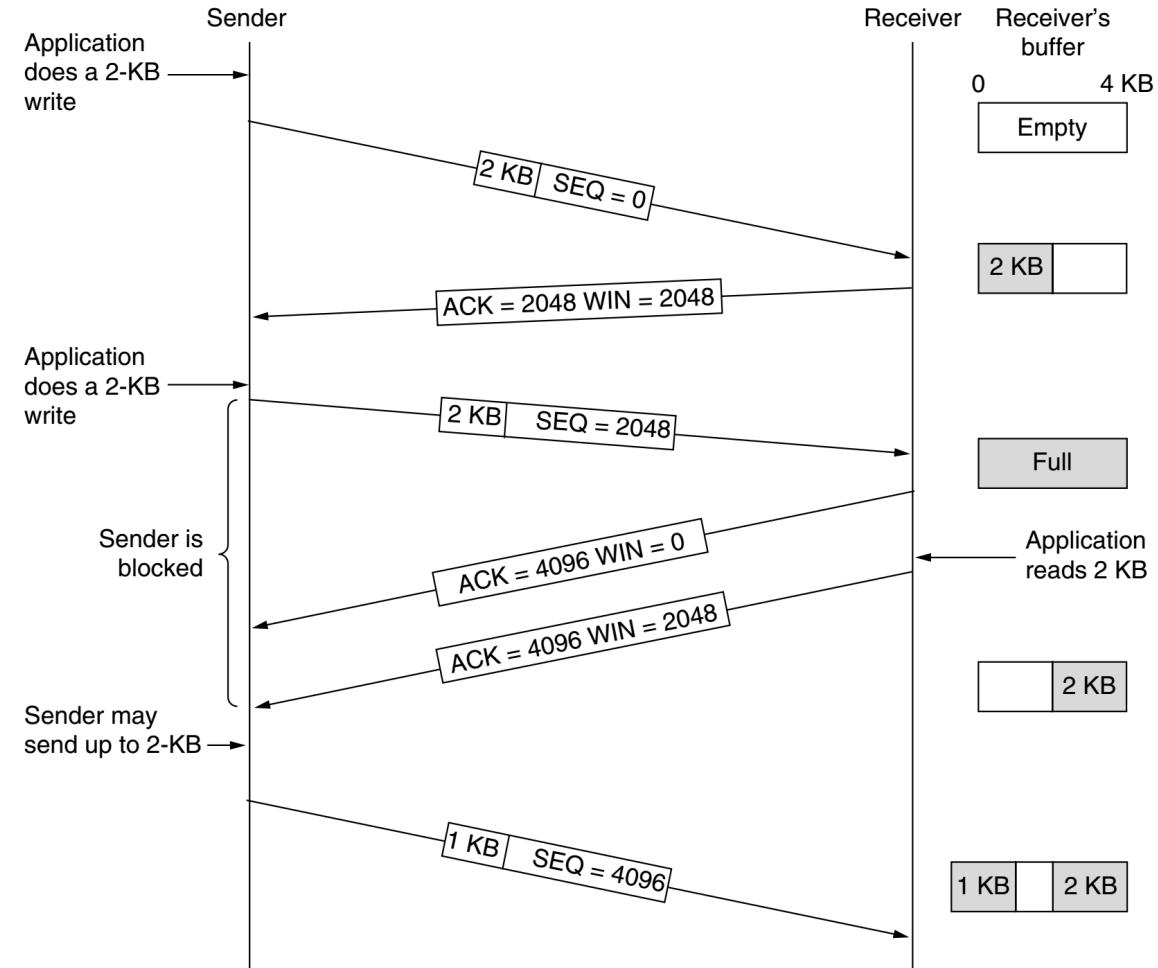
TCP Connection Management – Full Set of States

- The full TCP connection finite state machine has more states than the simple example from earlier.

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIME WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

TCP Transmission Policy

- TCP acknowledges bytes
- Receiver advertises window based on available buffer space



Week 8: Transport Layer Contd

Internet Technologies COMP90007

Lecturer: Muhammad Usman

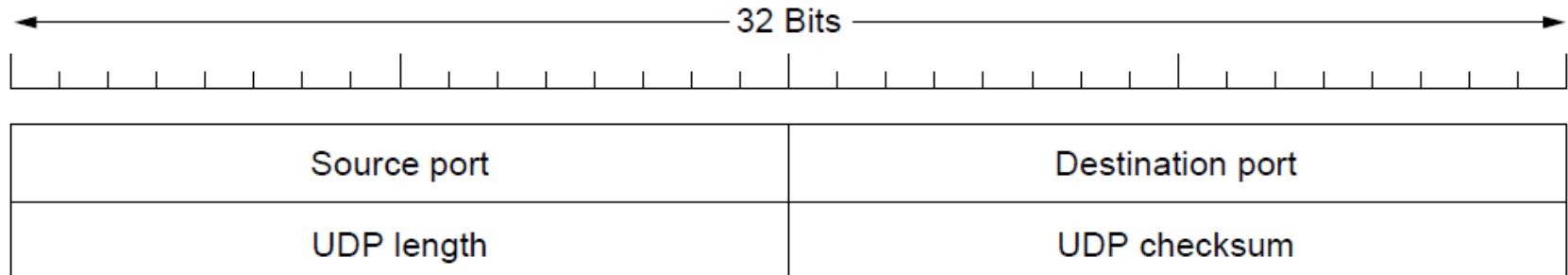
Semester 2, 2020

UDP

- Provides a protocol whereby applications can transmit encapsulated IP datagrams without a connection establishment
- UDP transmits in segments consisting of an 8-byte header followed by the payload
- UDP headers contain source and destination ports
- Payload is handed to the process which is attached to the particular port at destination

UDP Contd.

- Main **advantage** of using UDP over raw IP is:
 - the ability to specify ports for source and destination pairs, i.e., addressing for processes
- Both source and destination ports are required - destination allows for incoming segments, source allows reply for outgoing segments



Structure of UDP header: It has ports (TSAPs), length and checksum

Strengths and Weaknesses of UDP

- **Strengths:** provides an IP interface with multiplexing/demultiplexing capabilities and related transmission efficiencies
- **Weaknesses:** UDP does not include support for flow control, error control/retransmission of bad segments
- **Conclusion:** where applications require a precise level of control over packet flow/error/timing, UDP is a good choice as application layer can make choices
- **Domain Name System over the Internet is a famous user of UDP**

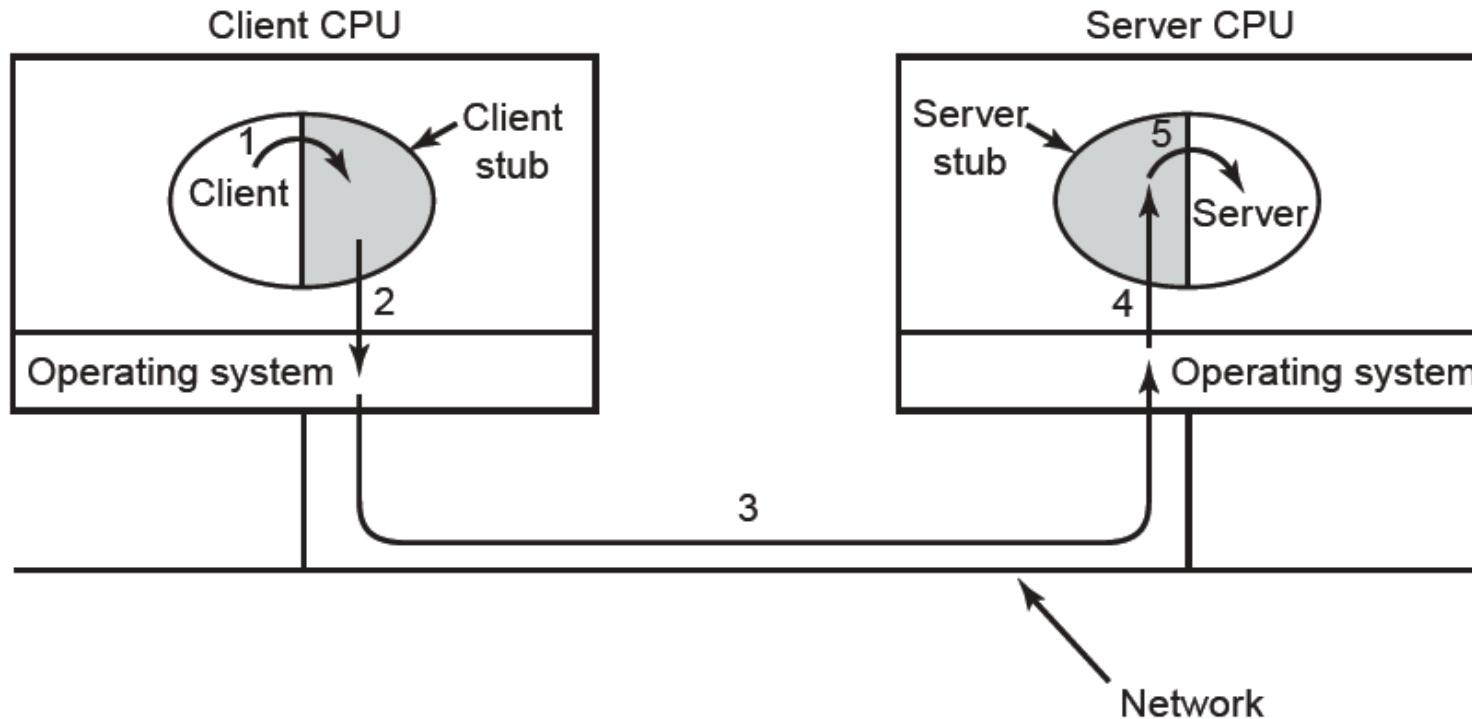
Another one with UDP: Remote Procedure Call (RPC)

- Sending a message and getting a reply back is analogous to making **a function call** in programming languages
- Birrell and Nelson modified this to allow programs to call procedures on remote hosts using UDP
 - **Remote Procedure Call (RPC)**

Remote Procedure Call (RPC)

- To call a remote procedure, the client is bound to a small library (the **client stub**) that represents the server procedure in the client's address space.
- Similarly the server is bound with a procedure called the **server stub**.
- These **stubs hide the fact that the procedure itself is not local.**

RPC Illustrated



Transmission Control Protocol (TCP)

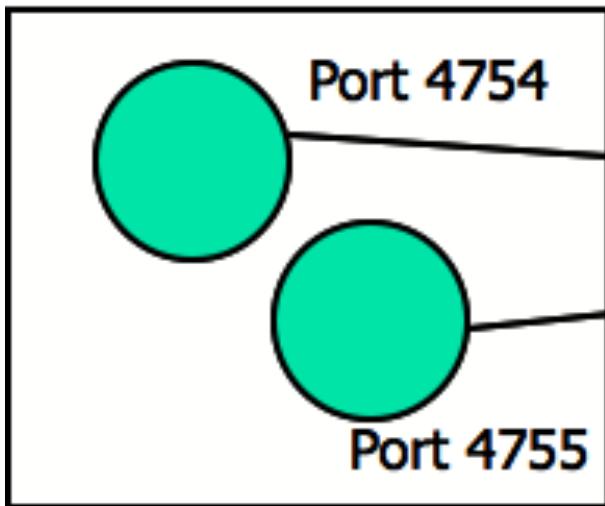
- Provides a protocol by which applications can transmit IP datagrams within a **connection-oriented** framework, thus increasing reliability
- TCP transport entity manages TCP streams and interfaces to the IP layer - can exist in numerous locations (kernel, library, user process)
- **TCP entity** accepts user data streams, and **segments them into pieces < 64KB** (often at a size in order so that the IP and TCP headers can fit into a single Ethernet frame), and sends each piece as a separate IP datagram
- Recipient TCP entities reconstruct the original byte streams from the encapsulation

The TCP Service Model

- Sender and receiver both create **sockets**, consisting of the IP address of the host and a port number as we saw earlier
- For TCP Service to be activated, **connections must be explicitly established between a socket at a sending host** (src-host, src-port) and a socket at a receiving host (dest-host, dest-port)
- Special one-way server sockets may be used for multiple connections simultaneously

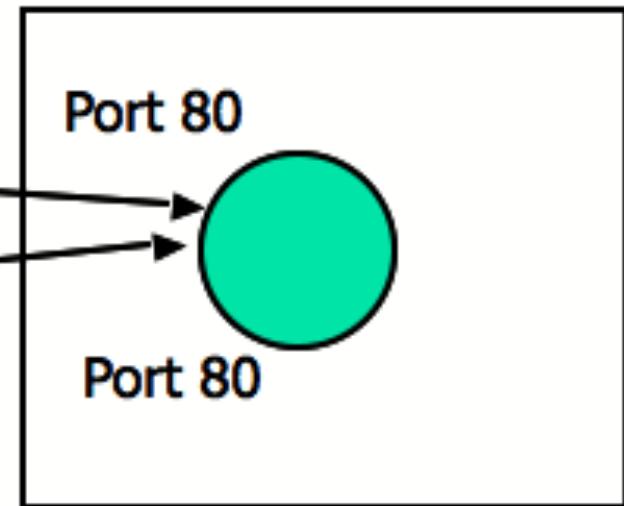
Example

Host 128.42.11.3



Web browser

Host 62.118.44.12



Web server

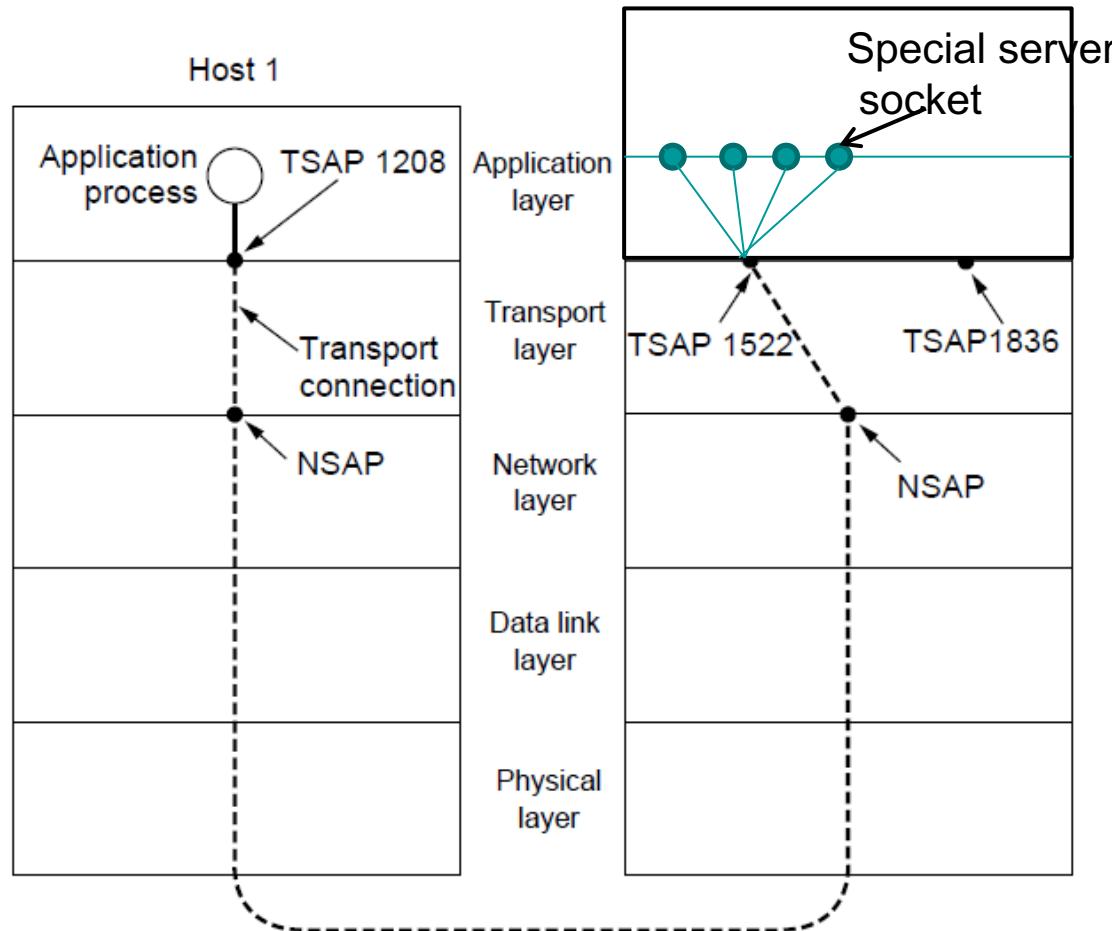
Port Allocations

- Recall TSAPs
- Port numbers can range from 0-65535
- Port numbers are regulated by IANA
(<http://www.iana.org/assignments/port-numbers>)
- Ports are classified into 3 segments:
 - Well Known Ports (0-1023)
 - Registered Ports (1024-49151)
 - Dynamic Ports (49152-65535)

Port	Protocol	Use
20, 21	FTP	File transfer
22	SSH	Remote login, replacement for Telnet
25	SMTP	Email
80	HTTP	World Wide Web
110	POP-3	Remote email access
143	IMAP	Remote email access
443	HTTPS	Secure Web (HTTP over SSL/TLS)
543	RTSP	Media player control
631	IPP	Printer sharing

Socket Library - Multiplexing

- Socket library provides a multiplexing tool on top of TSAPs to allow servers to service multiple clients
- It **simulate** the server using a different port to connect back to the client



Features of TCP Connections

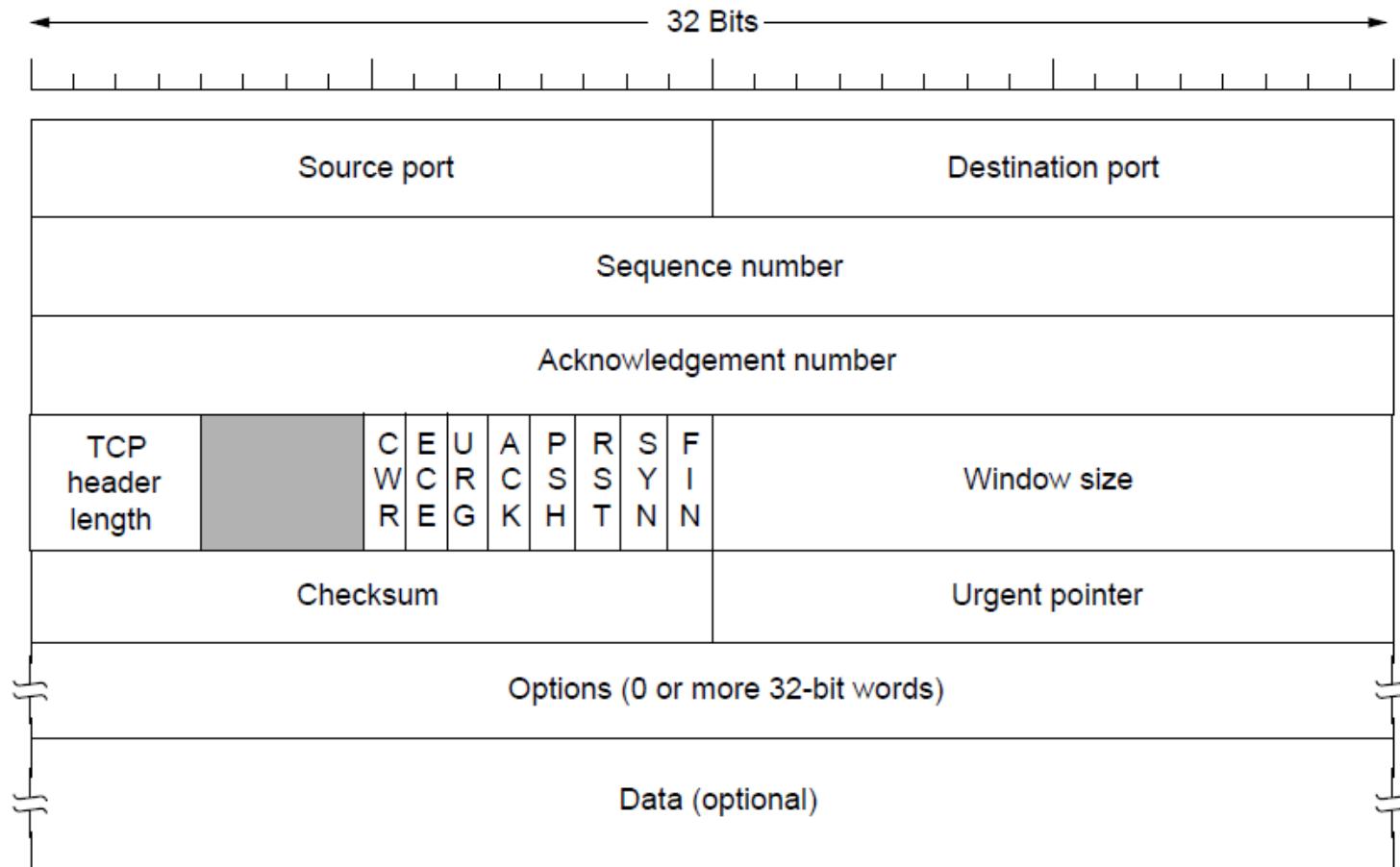
- TCP connections are:
- **Full duplex** - data in both directions simultaneously
- **Point to point** - exact pairs of senders and receivers
- **Byte streams**, not message streams - message boundaries are not preserved
- **Buffer options** - TCP entity can choose to buffer prior to sending or not depending on the context
 - **TCP_NODELAY in Java**
 - **Socket.setTcpNoDelay(boolean)**

TCP Contd

- Data sent between TCP entities in segments - segment has a 20 byte header plus zero or more data bytes
- TCP entities decide how large segments should be mainly with 2 constraints:
 - 65,515 byte IP payload
 - Ethernet unit size - generally 1500 bytes
- **Sliding window** - sender transmits and starts a timer
- Receiver sends back an acknowledgement which is the next sequence number expected - if sender's timer expires before acknowledgement, then the sender transmits the original segment again

The TCP Segment Header

- TCP header includes addressing (ports), sliding window (seq. / ack. number), flow control (window), error control (checksum) and more



The TCP Segment Header

- **Source port and Destination port** fields identify the local end points of the connection
- **Sequence number and Acknowledgement number** fields perform their usual functions
- **TCP header length** tells how many 32-bit words are contained in the TCP header
- **Window size** field tells how many bytes may be sent starting at the byte acknowledged
- **Checksum** is also provided for extra reliability. It checksums the header, the data
- **Options** field provides a way to add extra facilities not covered by the regular header
- **URG** is set to 1 if the *Urgent pointer* is in use. The Urgent pointer is used to indicate a byte offset from the current sequence number at which urgent data are to be found

The TCP Segment Header

- **CWR** and **ECE** are used to signal congestion when *ECN* (Explicit Congestion Notification) is used
- **ECE** is set to signal an ECN-Echo to a TCP sender to tell it to slow down when the TCP receiver gets a congestion indication from the network
- **CWR** is set to signal Congestion Window Reduced from the TCP sender to the TCP receiver so that it knows the sender has slowed down and can stop sending the ECN-Echo
- The **ACK** bit is set to 1 to indicate that the Acknowledgement number is valid. This is the case for nearly all packets. 0 means ignore ACK number field
- **PSH** bit indicates PUSHed data. The receiver is hereby kindly requested to deliver the data to the application upon arrival and not buffer it until a full buffer has been received

The TCP Segment Header

- The **RST** bit is used to abruptly reset a connection that has become confused due to a host crash or some other reason. It is also used to reject an invalid segment or refuse an attempt to open a connection
- The **SYN** bit is used to establish connections. The connection request has SYN = 1 and ACK = 0. The connection reply does bear an acknowledgement, so it has SYN = 1 and ACK = 1.
- In essence, the SYN bit is used to **denote both CONNECTION REQUEST and CONNECTION ACCEPTED**, with the ACK bit used to distinguish between those two possibilities.
- The **FIN** bit is used to release a connection. It specifies that the sender has no more data to transmit. However, after closing a connection, the closing process may continue to receive data.

TCP Connection Establishment and Release

- Connections established **using three-way handshake**
- Two **simultaneous connection attempts results in only one connection** (uniquely identified by end points)
- Connections released with **symmetric release**
- Timers used for lost connection releases

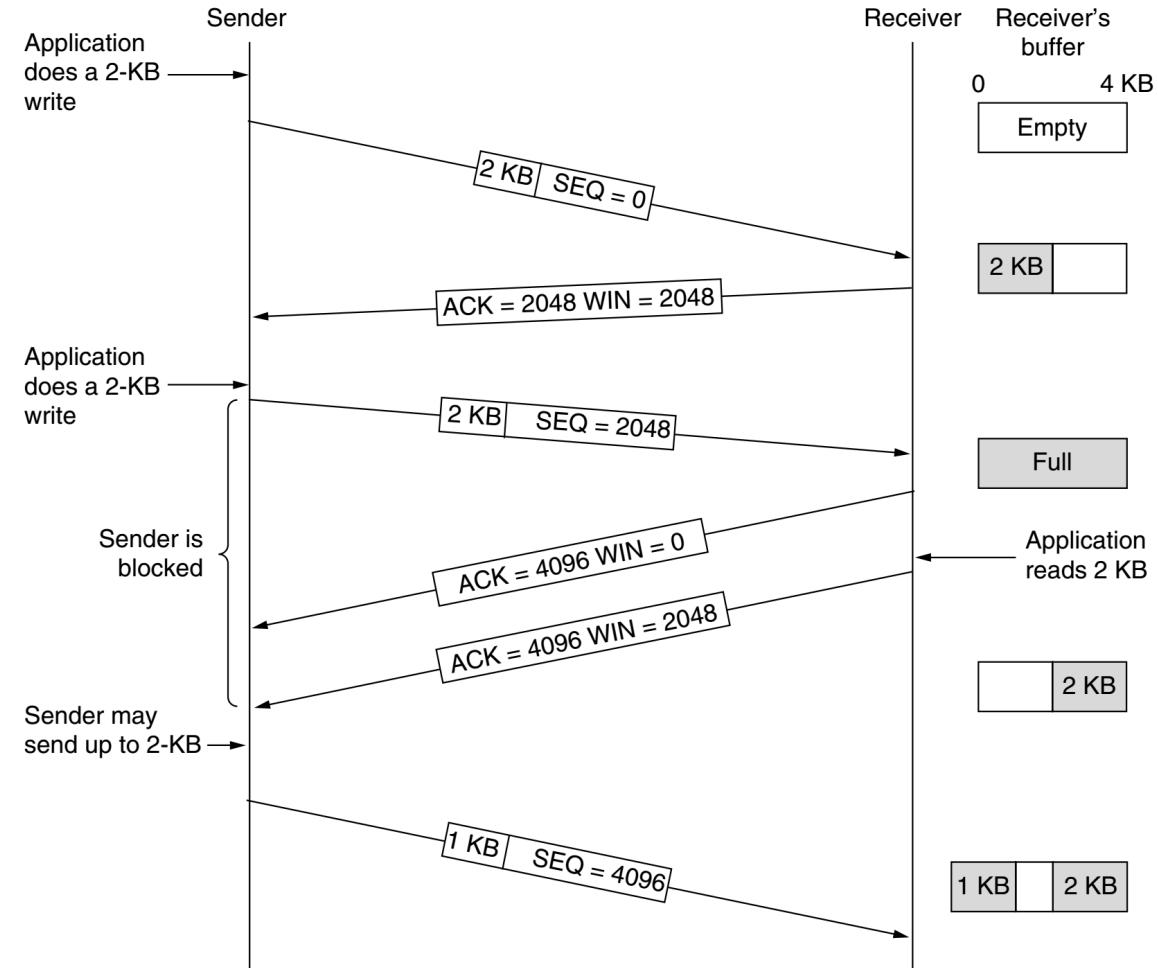
TCP Connection Management – Full Set of States

- The full TCP connection finite state machine has more states than the simple example from earlier.

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIME WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

TCP Transmission Policy

- TCP acknowledges bytes
- Receiver advertises window based on available buffer space



Week 8: Transport Layer Contd

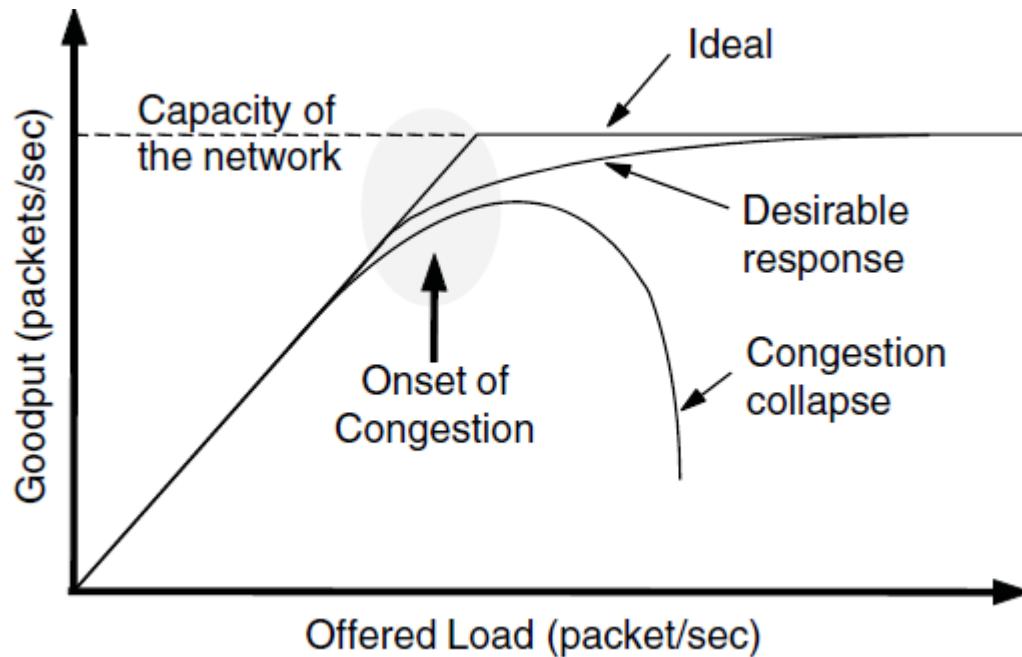
Internet Technologies COMP90007

Lecturer: Muhammad Usman

Semester 2, 2020

What happens when congested?

- Congestion results when too much traffic is offered; performance degrades due to loss/retransmissions
- Goodput (=useful packets) trials offered load



Congestion Control vs Flow Control

- **Flow control** is an issue for point to point traffic, primarily concerned with preventing sender transmitting data faster than receiver can receive it
- **Congestion control** is an issue affecting the ability of the subnet to actually carry the available traffic, in a global context

Load Shedding

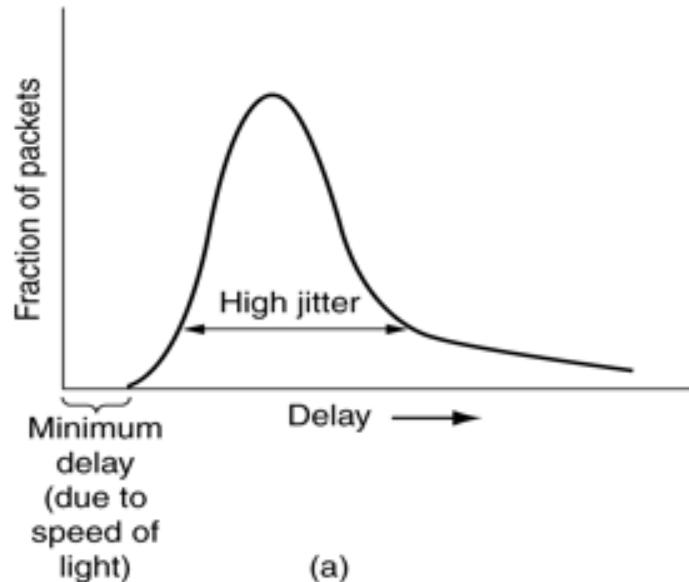
- When congestion control mechanisms fail, load shedding is the key remaining possibility
 - **drop packets**
- In order to ameliorate impact, applications can mark certain **packets as priority** to avoid discard policy

What is the key problem if network is not delivering properly:

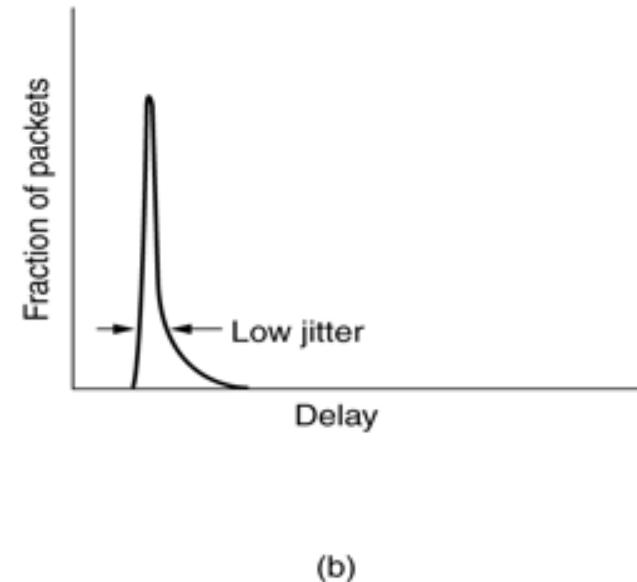
- Quality of Service becomes low
- Expected network performance is an important criterion for a wide range of network applications
- Some engineering techniques are available to guarantee QoS (Quality of Service)
- 4 things to watch out for:
bandwidth, reliability, delay, jitter

Jitter is Interesting/New for Us

- Jitter is the variation in packet arrival times
 - a) high jitter
 - b) low jitter



(a)



(b)

Mechanisms for Jitter Control

- Jitter is an issue for some applications
- Jitter can be contained by determining the expected transit time of a packet
- Packets can be shuffled at each hop in order to minimise jitter - slower packets sent first, faster packets wait in a queue
- For certain applications jitter control is extremely important as it mainly directly affects the quality perceived by the application user

QoS Requirements

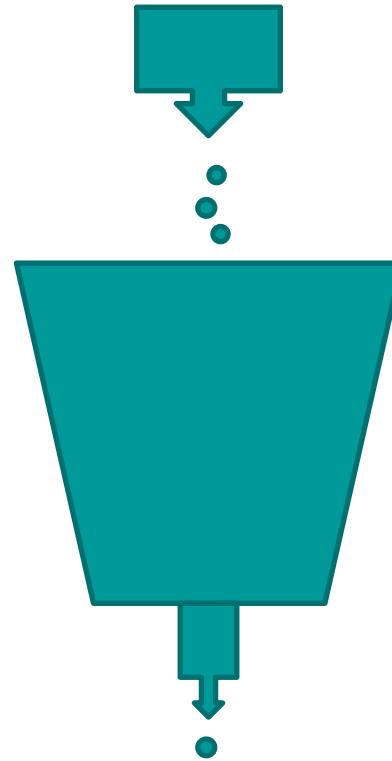
- Different applications care about different properties
 - We want all applications to get what they need
“High” means a demanding the requirement!

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

Techniques for Achieving QoS

- **Over-provisioning**
 - more than adequate buffer, router CPU, and bandwidth (expensive and not scalable ...)
- **Buffering**
 - buffer received flows before delivery - increases delay, but smoothes out jitter, no effect in reliability or bandwidth
- **Traffic Shaping**
 - regulate the average rate of transmission and burstiness of transmission
 - **leaky bucket**
 - **token bucket**

Leaky Bucket



Large **bursts** of traffic is buffered and smoothed while sending

e.g. can be done at host sending data

Techniques for Good QoS Contd

- **Resource reservation**

- reserve bandwidth, buffer space, CPU in advance

- **Admission control**

- routers can decide based on traffic patterns whether to accept new flows, or reject/**reroute** them

- **Proportional routing**

- traffic for same destination split across multiple routes

- **Packet scheduling**

- Create queue(s) based on priority etc
 - fair queuing, weighted fair queuing

TCP and Congestion Control

- When networks are overloaded, congestion occurs, potentially affecting all layers
- Although lower layers (data and network) attempt to ameliorate congestion, in reality **TCP impacts congestion most significantly** because TCP offers best methods to reduce the data rate, and hence reduce congestion itself

Congestion Control: Design

- Two different problems exist
 - network capacity and receiver capacity
 - these should be dealt with separately, but compatibly
- The sender maintains two windows actually
 - Window described by the receiver
 - Congestion window
- Each regulates the number of bytes the sender can transmit – the maximum transmission rate is the **minimum of the two windows**

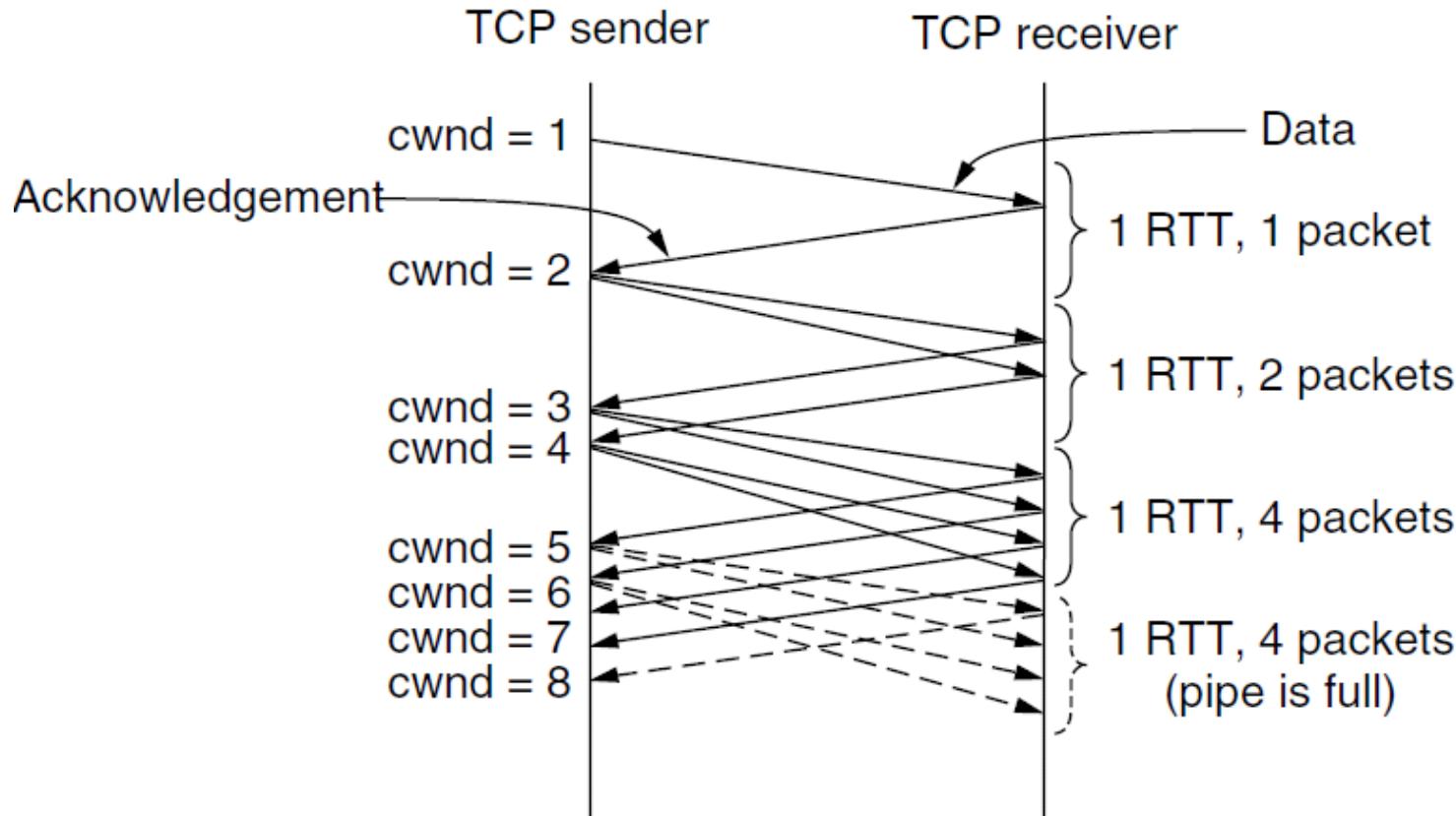
TCP and Congestion Control Contd

- TCP adopts a defensive stance:
 - At connection establishment, a suitable window size is chosen by the receiver based on its buffer size
 - If the sender is constrained to this size, then congestion problems will not occur due to buffer overflow at the receiver itself, but may still occur due to congestion within the network

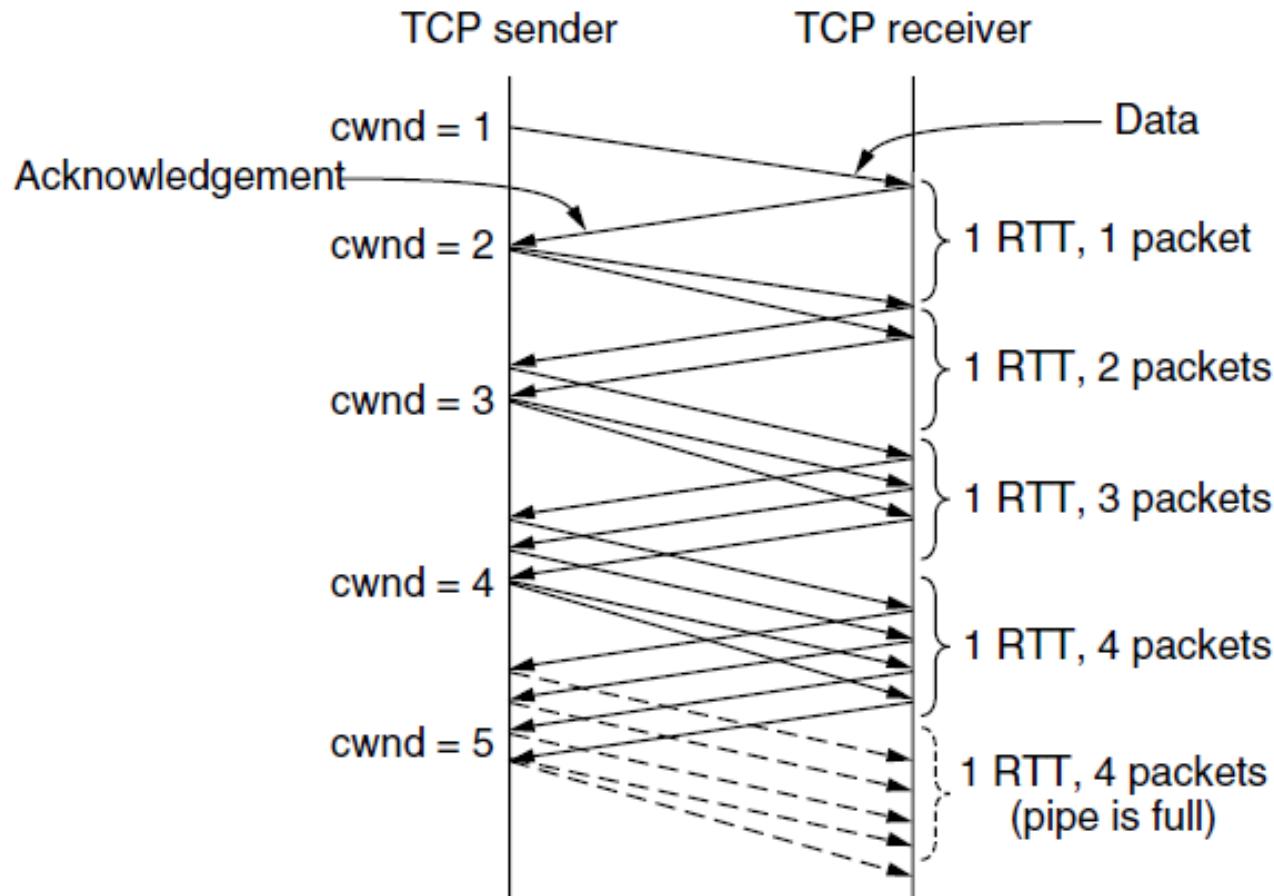
Incremental Congestion Control: Slow Start

- On connection establishment, the sender initializes the congestion window to a size, and transmits one segment
- If this segment is acknowledged before the timer expires, the sender adds another segment's worth of bytes to the congestion window, and transmits two segments
- As each new segment is acknowledged, the congestion window is increased by one more segment
- In effect, each set of acknowledgements doubles the congestion window - which grows until either a timeout occurs or the receiver's specified window is reached

Slow Start



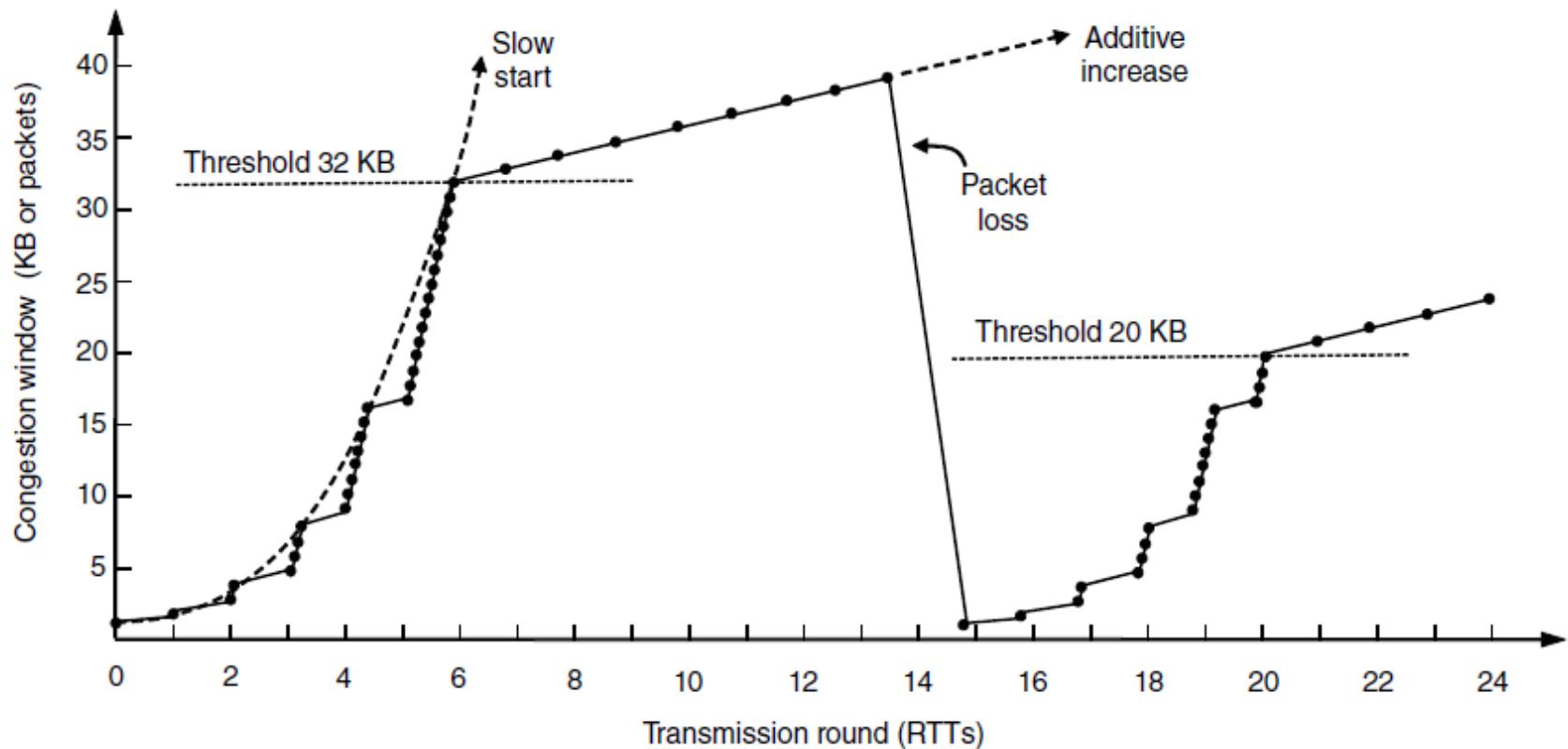
Additive increase



Internet Congestion Control

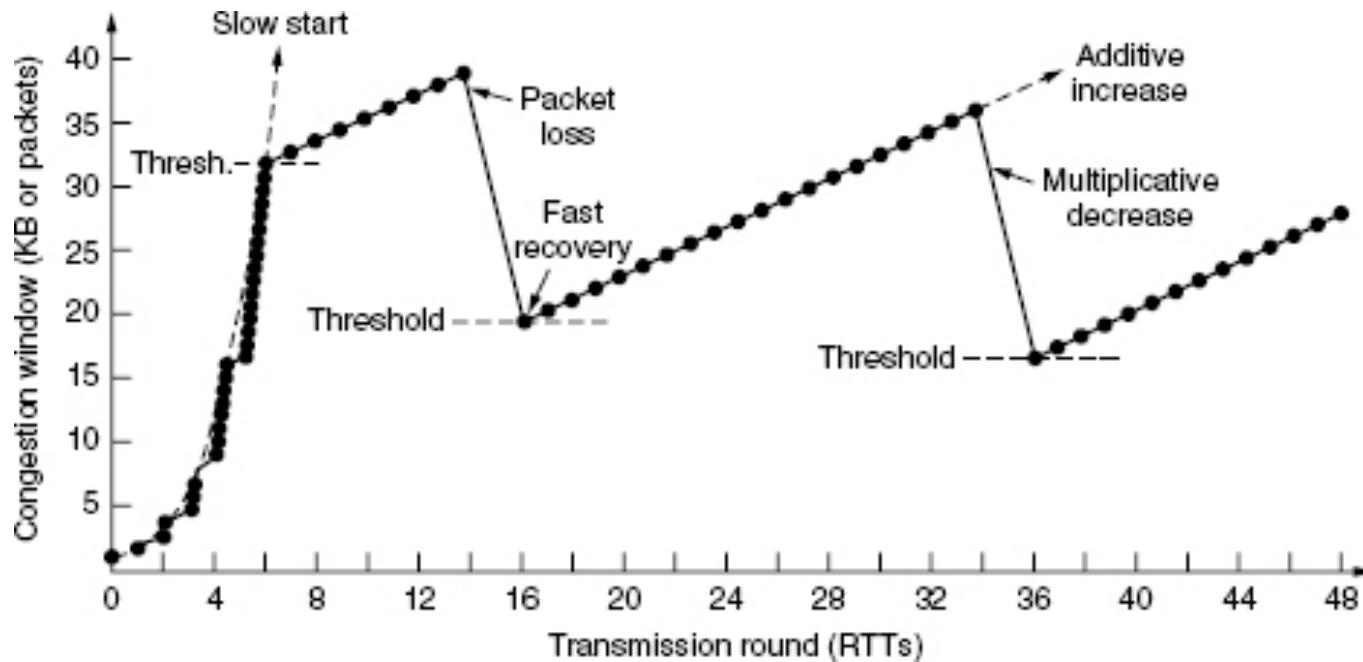
Slow start followed by additive increase (TCP Tahoe)

Threshold is half of previous



Internet Congestion Control Contd

Another one with TCP Reno



Congestion Control And Wireless

- Much harder to deal with
 - Things are increasingly wireless
 - Not everything is wireless, but parts of a path
 - So how does one know where wireless is
 - More variety on wireless links as well
 - SNR varies when people move
 - Delay is different if it is Wifi vs Satellite
 - This is a hot area of research...

TCP Timer

- A key worry is when timers go out
- Too early means too many resends
- Too late means reliability comes with more additional cost
- Solutions rely on dynamicity as network conditions change
- One needs to measure network performance and adapt timers

Week 8: Transport Layer Contd

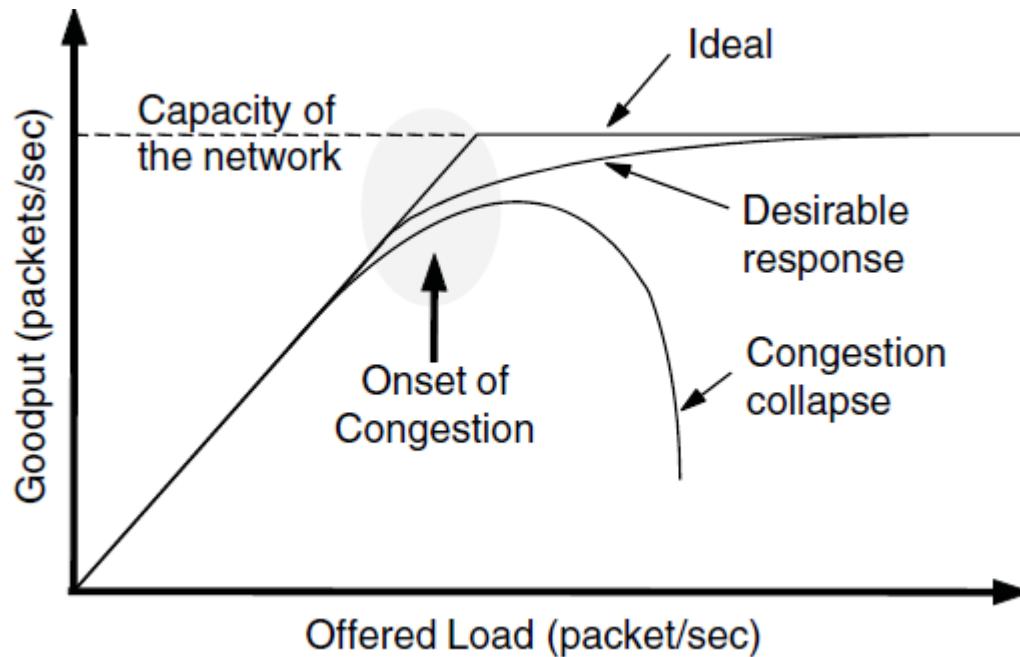
Internet Technologies COMP90007

Lecturer: Muhammad Usman

Semester 2, 2020

What happens when congested?

- Congestion results when too much traffic is offered; performance degrades due to loss/retransmissions
- Goodput (=useful packets) trials offered load



Congestion Control vs Flow Control

- **Flow control** is an issue for point to point traffic, primarily concerned with preventing sender transmitting data faster than receiver can receive it
- **Congestion control** is an issue affecting the ability of the subnet to actually carry the available traffic, in a global context

Load Shedding

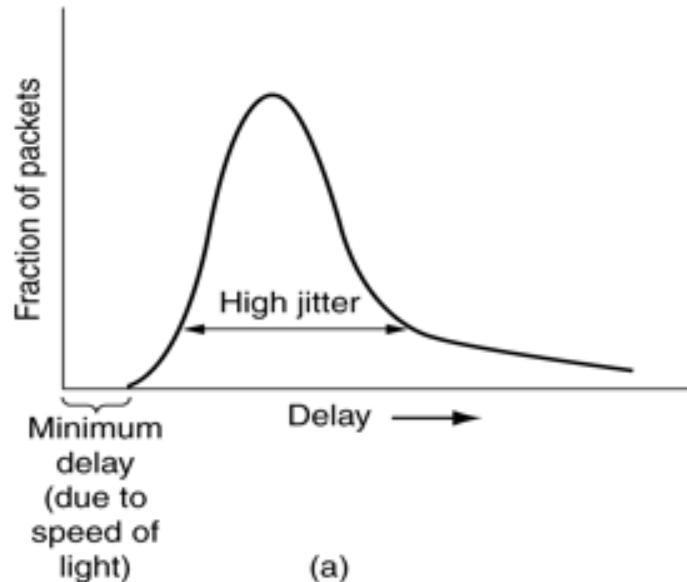
- When congestion control mechanisms fail, load shedding is the key remaining possibility
 - **drop packets**
- In order to ameliorate impact, applications can mark certain **packets as priority** to avoid discard policy

What is the key problem if network is not delivering properly:

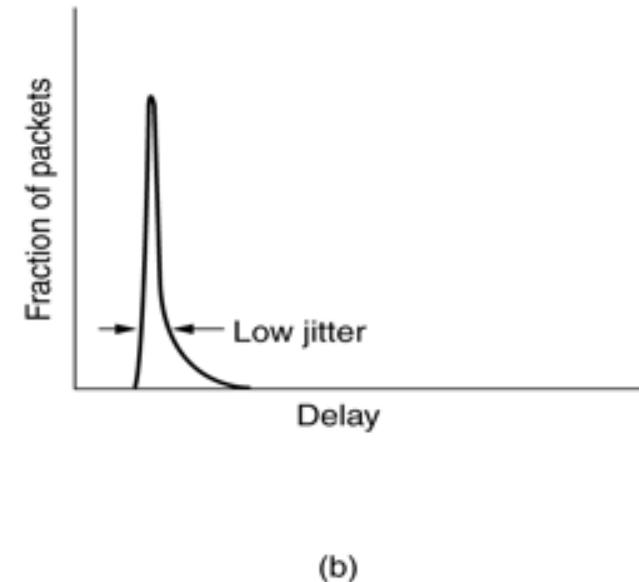
- Quality of Service becomes low
- Expected network performance is an important criterion for a wide range of network applications
- Some engineering techniques are available to guarantee QoS (Quality of Service)
- 4 things to watch out for:
bandwidth, reliability, delay, jitter

Jitter is Interesting/New for Us

- Jitter is the variation in packet arrival times
 - a) high jitter
 - b) low jitter



(a)



(b)

Mechanisms for Jitter Control

- Jitter is an issue for some applications
- Jitter can be contained by determining the expected transit time of a packet
- Packets can be shuffled at each hop in order to minimise jitter - slower packets sent first, faster packets wait in a queue
- For certain applications jitter control is extremely important as it mainly directly affects the quality perceived by the application user

QoS Requirements

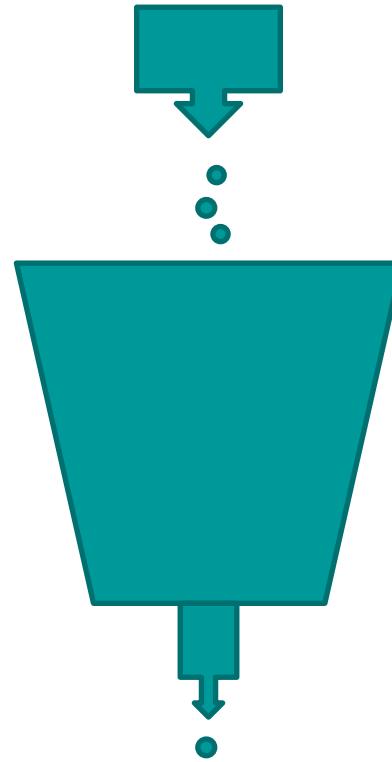
- Different applications care about different properties
 - We want all applications to get what they need
“High” means a demanding the requirement!

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

Techniques for Achieving QoS

- **Over-provisioning**
 - more than adequate buffer, router CPU, and bandwidth (expensive and not scalable ...)
- **Buffering**
 - buffer received flows before delivery - increases delay, but smoothes out jitter, no effect in reliability or bandwidth
- **Traffic Shaping**
 - regulate the average rate of transmission and burstiness of transmission
 - **leaky bucket**
 - **token bucket**

Leaky Bucket



Large **bursts** of traffic is buffered and smoothed while sending

e.g. can be done at host sending data

Techniques for Good QoS Contd

- **Resource reservation**

- reserve bandwidth, buffer space, CPU in advance

- **Admission control**

- routers can decide based on traffic patterns whether to accept new flows, or reject/**reroute** them

- **Proportional routing**

- traffic for same destination split across multiple routes

- **Packet scheduling**

- Create queue(s) based on priority etc
 - fair queuing, weighted fair queuing

TCP and Congestion Control

- When networks are overloaded, congestion occurs, potentially affecting all layers
- Although lower layers (data and network) attempt to ameliorate congestion, in reality **TCP impacts congestion most significantly** because TCP offers best methods to reduce the data rate, and hence reduce congestion itself

Congestion Control: Design

- Two different problems exist
 - network capacity and receiver capacity
 - these should be dealt with separately, but compatibly
- The sender maintains two windows actually
 - Window described by the receiver
 - Congestion window
- Each regulates the number of bytes the sender can transmit – the maximum transmission rate is the **minimum of the two windows**

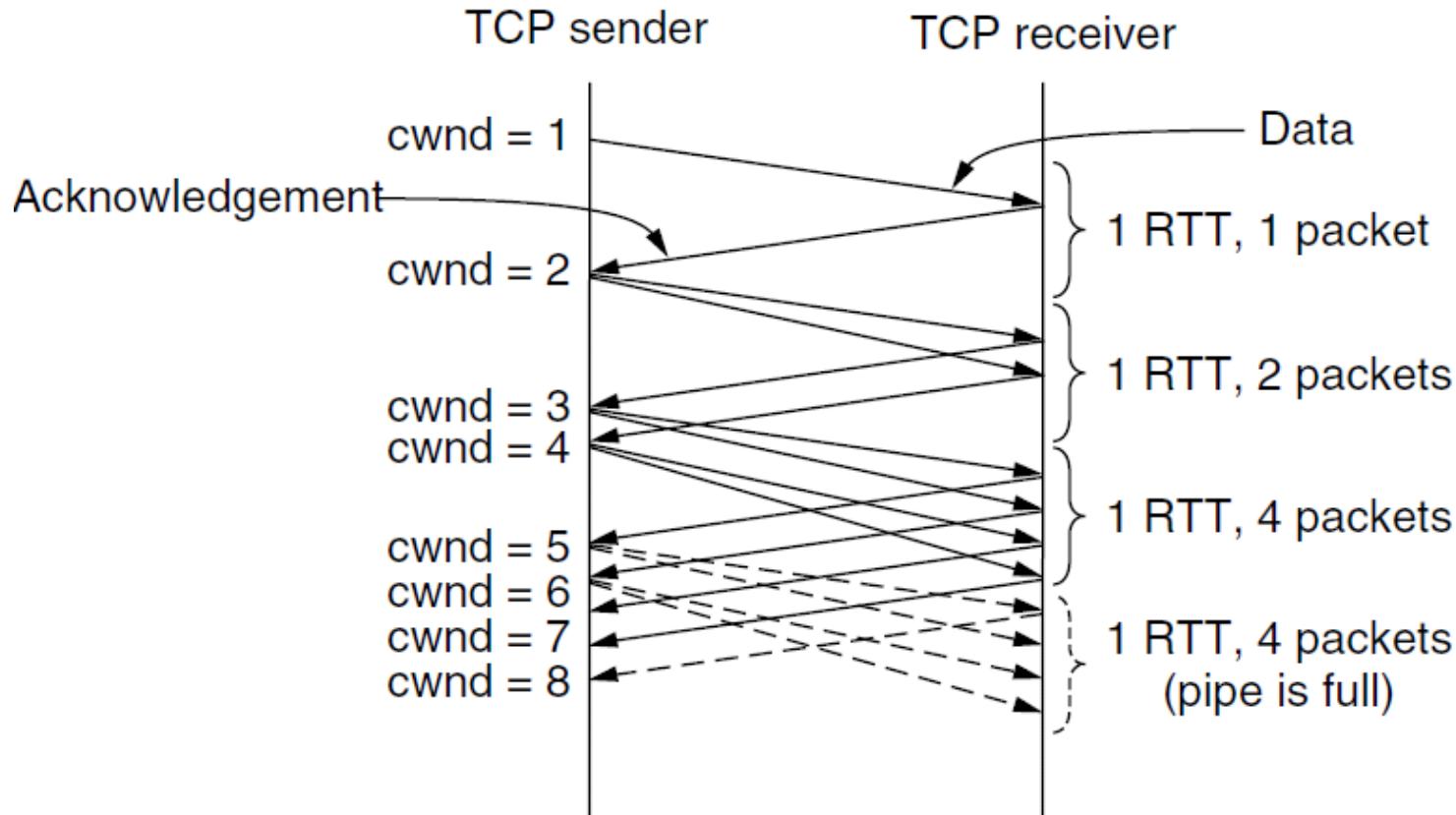
TCP and Congestion Control Contd

- TCP adopts a defensive stance:
 - At connection establishment, a suitable window size is chosen by the receiver based on its buffer size
 - If the sender is constrained to this size, then congestion problems will not occur due to buffer overflow at the receiver itself, but may still occur due to congestion within the network

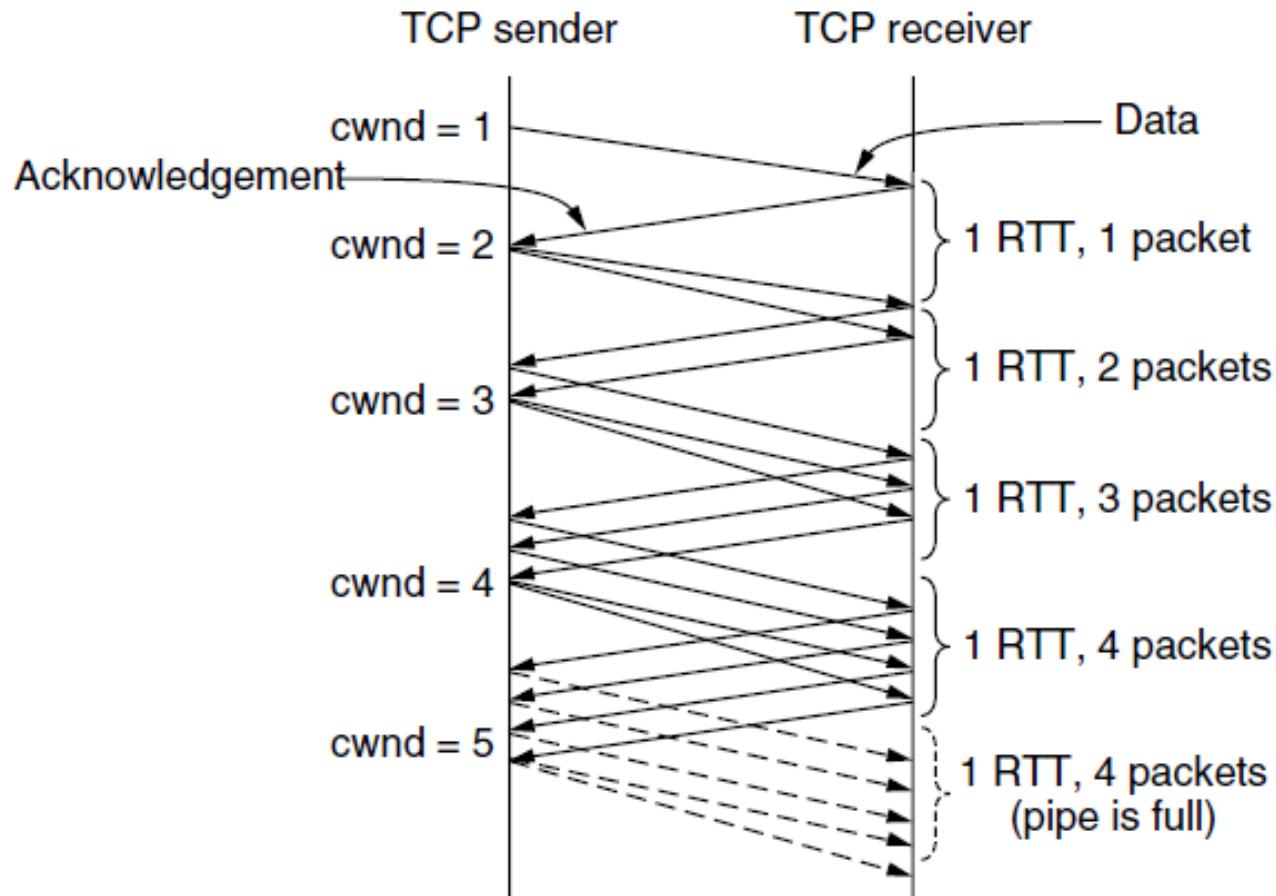
Incremental Congestion Control: Slow Start

- On connection establishment, the sender initializes the congestion window to a size, and transmits one segment
- If this segment is acknowledged before the timer expires, the sender adds another segment's worth of bytes to the congestion window, and transmits two segments
- As each new segment is acknowledged, the congestion window is increased by one more segment
- In effect, each set of acknowledgements doubles the congestion window - which grows until either a timeout occurs or the receiver's specified window is reached

Slow Start

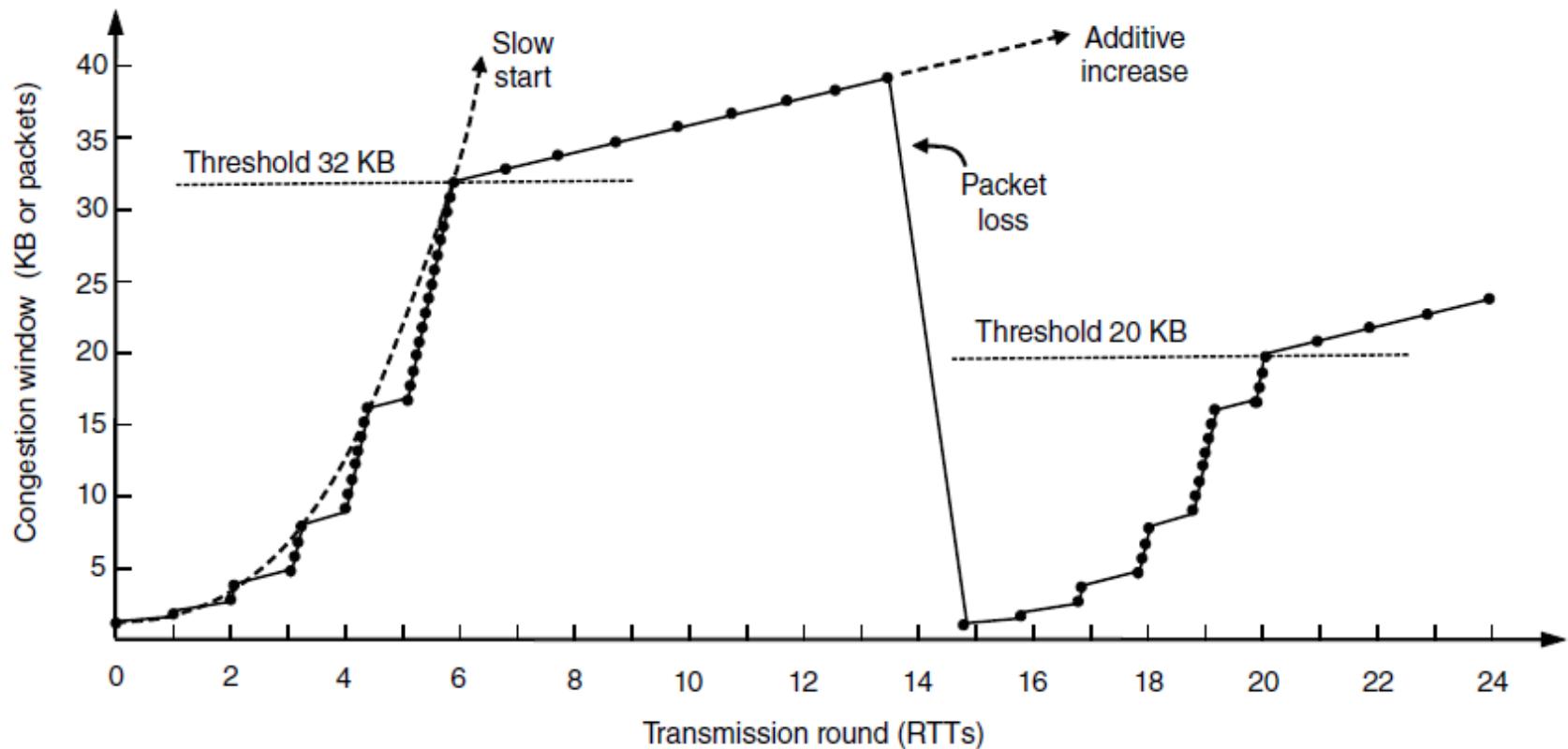


Additive increase



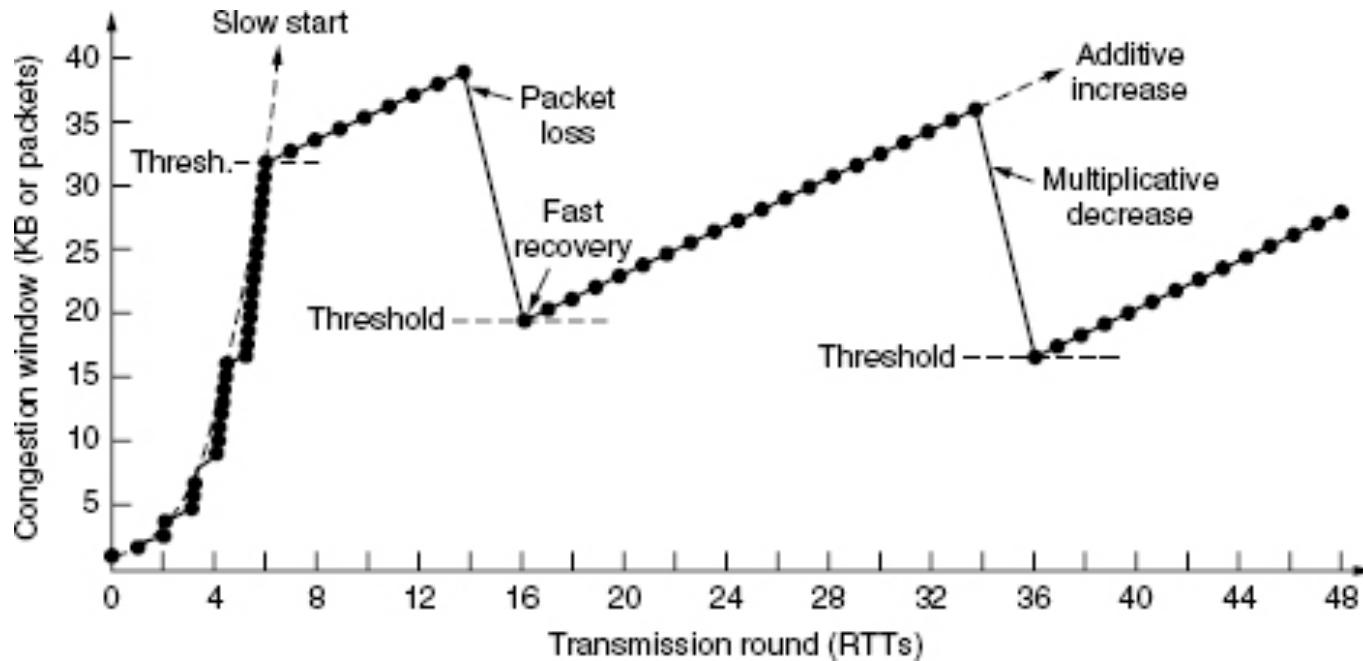
Internet Congestion Control

Slow start followed by additive increase (TCP Tahoe)
Threshold is half of previous



Internet Congestion Control Contd

Another one with TCP Reno



Congestion Control And Wireless

- Much harder to deal with
 - Things are increasingly wireless
 - Not everything is wireless, but parts of a path
 - So how does one know where wireless is
 - More variety on wireless links as well
 - SNR varies when people move
 - Delay is different if it is Wifi vs Satellite
 - This is a hot area of research...

TCP Timer

- A key worry is when timers go out
- Too early means too many resends
- Too late means reliability comes with more additional cost
- Solutions rely on dynamicity as network conditions change
- One needs to measure network performance and adapt timers



COMP90007

Internet Technologies

Ling Luo
Semester 2, 2020





Lecturers

Dr. Ling Luo

- Lecturer at School of Computing and Information Systems
- Main research interests are machine learning, data mining, behaviour analytics
- ling.luo@unimelb.edu.au
- More information: <https://findanexpert.unimelb.edu.au/profile/849504-ling-luo>

Dr. Muhammad Usman

- Senior Lecturer at School of Computing and Information Systems
- Main research interests are quantum computing, machine learning, quantum devices.
- muhammad.usman@unimelb.edu.au
- More information: <https://findanexpert.unimelb.edu.au/profile/643365-m.-usman>

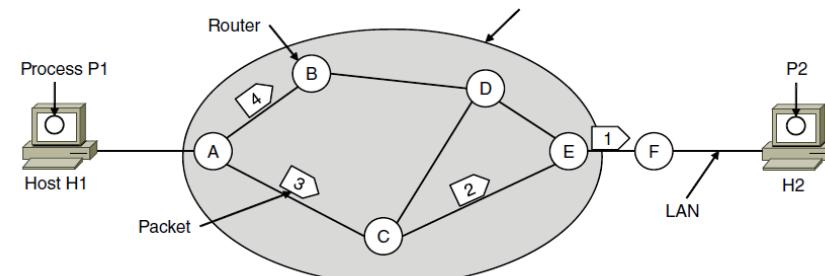
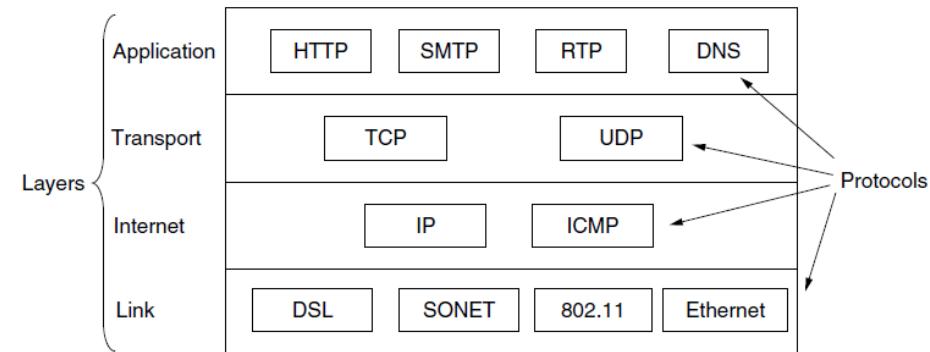
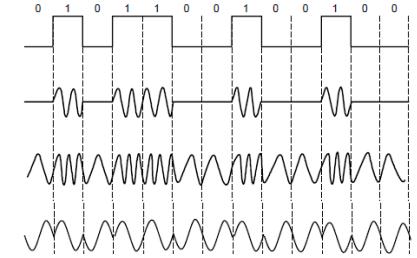
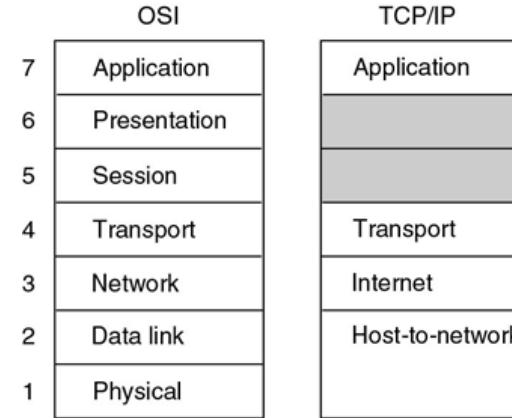


More about you?



Overview of Topics

- **Basics of computer networks** through a study of layered models of computer networks and applications.
- **Main Topics:** Introduction to Internet, reference model layers, protocols and services, data transmission basics, interface standards, network topologies, data link protocols, message routing, LANs, WANs, TCP/IP suite, detailed study of common network applications...



Lectures

- **3 lectures per week for 12 weeks**
 - Online lectures, the recordings will be available after class on Canvas
 - Mondays 1:00 pm – 2:00 pm
 - Tuesdays 3:15 pm – 4:15 pm
 - Fridays 1:00 pm – 2:00 pm

Tentative Schedule

Week	Topic
1	Introduction
2	Physical Layer
3	Data Link Layer
4	Medium Access Control
5	Network Layer
6	Network Layer
7	Transport Layer
8	Transport Layer
9	Application Layer
	Non-teaching period
10	Application Layer
11	Network Security
12	Review

Ling

Usman



Tutorials

- **1-hour tutorial per week for 11 weeks**
 - Starting from Week 2, online via Zoom
 - Tutorials are the key place to solve questions interactively, measure and test things and get help for projects
- Tutors
 - Rahul Sharma
 - Shashikant Ilager
 - Muhammed Tawfiqul Islam
 - Yifei Wang
- Each tutor will set their own mode of contact and consultation method, please meet them in your tutorials next week

Time	Tutor
Tue	2:15 pm
Tue	4:15 pm
Wed	11:00 am
Wed	3:15 pm
Wed	4:15 pm
Thu	11:00 am
Thu	1:00 pm
Thu	2:15 pm
Thu	5:15 pm

Each student is expected to attend the same tutorial through out the semester for their tutor to follow the progress properly



Subject Material

- **Canvas LMS** is the primary portal for the subject: <https://canvas.lms.unimelb.edu.au/>
 - Announcements
 - Lecture and tutorial materials
 - Assessments
 - Grades
 - Discussion forum
 - Other subject information: handbook, academic integrity, guides etc.



Communication

- **Announcements on LMS**
- **General enquiries: Discussion forum on LMS**
 - Check discussion forum regularly
 - We encourage all students to join in discussions – answering other students' questions is one of the best ways to improve your own understanding
 - Please do not post sections of your assignments publicly!
- **Personal/private concerns: Email the instructors**
 - Please include “COMP90007” and your student ID in email subject
 - If you email us about a general enquiry, we may ask you to re-post your question in the forum



Assessments (1)

- **2 Assignments**, 5% of total mark for each
 - Similar to tutorial questions
 - Good preparation for exams
 - One for each half of the semester, they will be due around week 6 and 12 respectively
- **2 Projects**
 - **Project 1:** hands-on networking experience/measurements, 10%
This project will cover the first half of the semester in terms of your practical work and will be due around Week 8. (There will be a lecture related to this soon)
 - **Project 2:** written report on a networking related topic, 15%
You will do some research on an emerging topic in networking. This will cover the second half of the semester and will be due around Week 12.



Assessments (2)

- **Midterm exam, 5%**
 - A 45 minute test, during class time via LMS around Week 7
 - Covers first half of the semester
 - A good chance to test yourself and learn about types of questions you may get in the final
- **Final exam, 60%**
 - Centrally timetabled
 - Questions are similar to other assessments that you will work on during the semester

NOTE: Reading the book in the last minute will not help as there will be just too much material to cover



Assessments (3)

- All assessments are individual work, no team projects in this semester.
- Hurdle on assessments, i.e., 50% per assessment except the midterm
 - 50% overall
 - 50% in the homework assignments
 - 50% in the hands-on project and technical report-based project
 - 50% in the end-of-semester written examination

This means just doing the final exam well is not enough to pass the subject

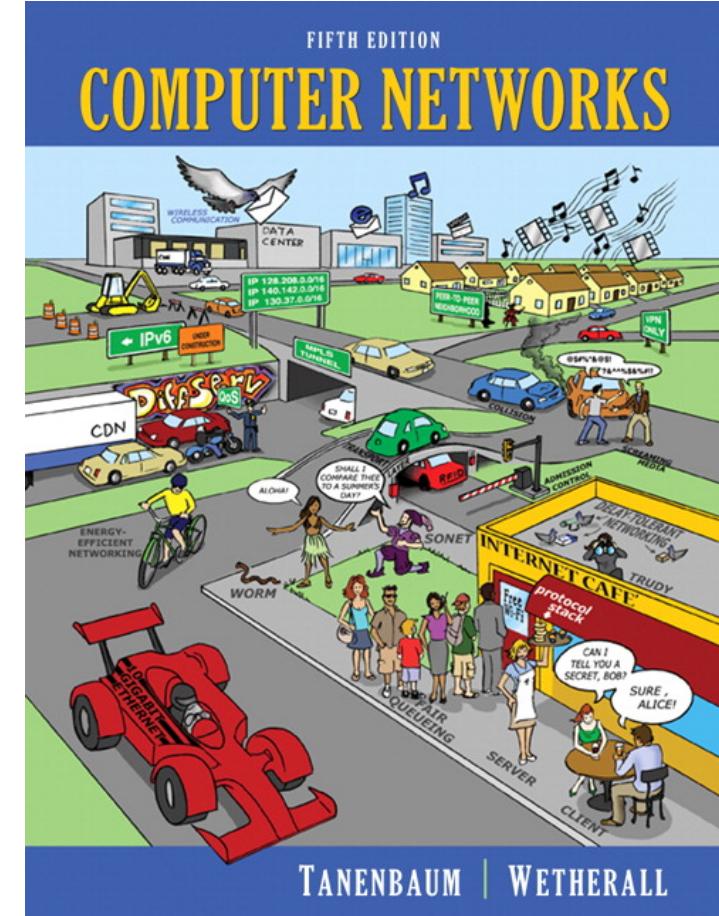


Academic Integrity

- More information: <https://academicintegrity.unimelb.edu.au/>
 - Plagiarism: Presenting the work of another person as your own
 - Self-plagiarism: You cannot re-use any part of your work that has already been submitted for assessment without proper citation.

Textbook

- **Computer Networks, 5th Edition** By: Andrew S. Tanenbaum; David J. Wetherall, Publisher: Pearson Library has online version (link on LMS)
Suggested readings will be posted on LMS each week





FAQ

- Will I have to program extensively in this subject? **No, but you need to know 1 programming language to comprehend some concepts**
- What if I have some background in networking? **Consider applying for credit now**
- Will there be team projects? **Not in this semester**
- What would the final exam be like? **Nothing surprising if you attended the subject with a genuine effort on all fronts**
- What is examinable in the exams: **Everything, you will know how much you need to know about each bit once you listen to the lectures/tutorials**



QUESTIONS?

Week 1 – Introduction to Networking

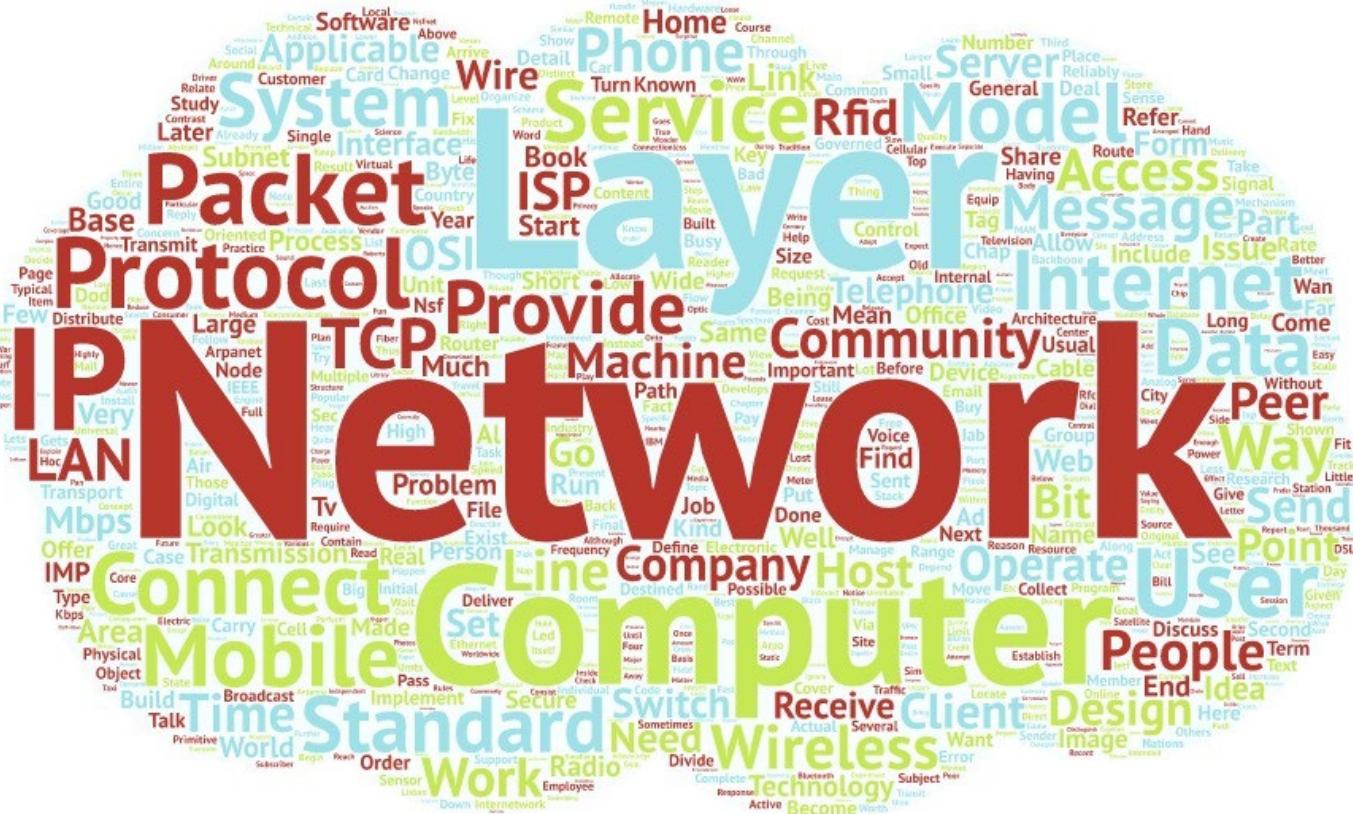
COMP90007 Internet Technologies

Lecturer: Ling Luo

Semester 2, 2020

Outline

- Computer Networks
- Network Types
- Protocols, Layers and Services



Terminologies

- A **network device**: eg. PC, Router, Switch, Phone
- **Server**: Provider of a service. Accept requests from clients
- **Client**: A network device connecting to a server and requesting a service
- **Computer Network**: A collection of autonomous computers interconnected by a single technology

Terminologies

- **Packet**: A message sent between two network devices
- **IP address**: A unique number identifying a network device

Network vs Computer Network

■ Network:

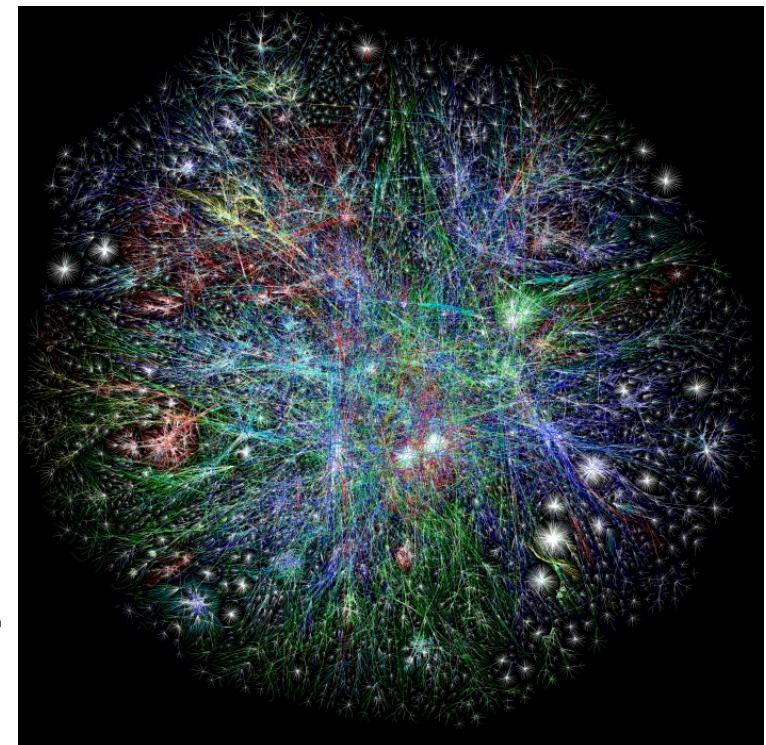
- An intricately connected system of things or people
- An interconnected or intersecting configuration or system of components

■ Computer Network:

- A data network with computers at one or more of the nodes [Oxford Dictionary of Computing]
- A collection of autonomous computers interconnected by a single technology

What are the Internet and the World Wide Web?

- Neither the Internet nor the WWW is a computer network!
- Simple answers:
 - The **Internet** is not a single network but a **network of networks**!
 - The **WWW** is a distributed system that **runs on top of the Internet**



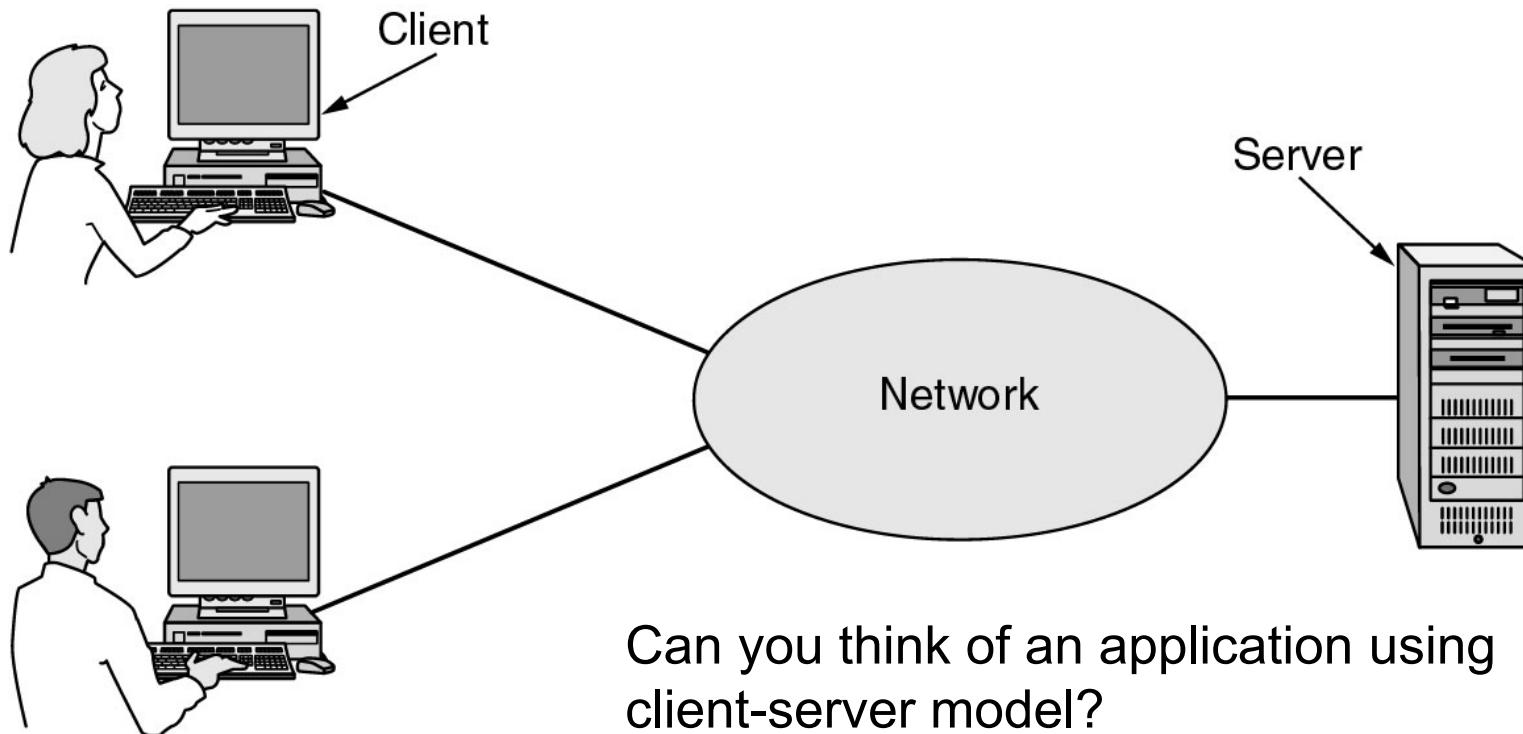
<https://mountpeaks.wordpress.com/>

Uses of Computer Networks

- Business Applications
 - Resource sharing (e.g., printer, scanner, files)
- Home Applications
 - Access to remote information
 - Interactive entertainment
 - E-commerce
 - Social Interactions
- Mobile Users
- Internet-of-things
 - parking, smart-meter, vending machines

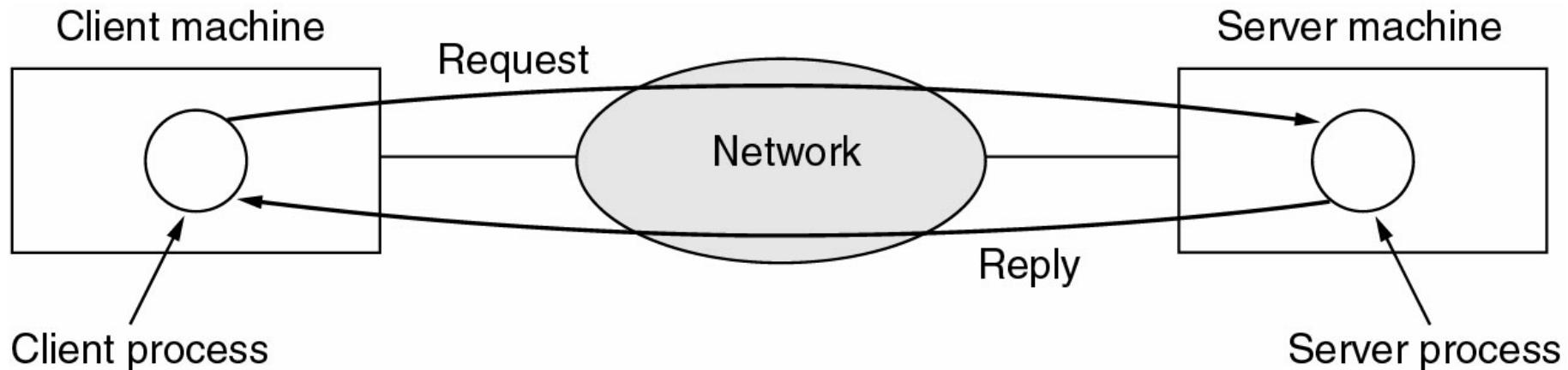
A Core Application Domain: Business Applications of Networks

- Origins: Simple Client-Server Network
- A network with two clients and one server

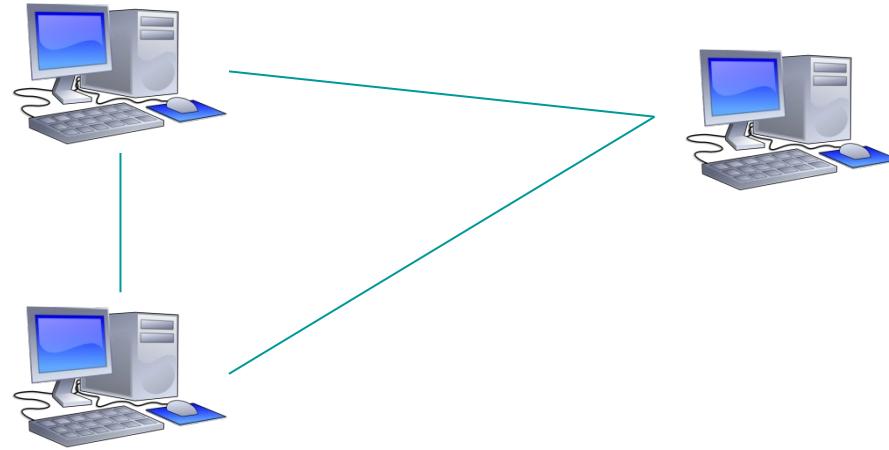


Business Applications of Networks (2)

- The client-server model involves requests and replies



Computer Networks



How does it scale to billions of devices?
What about distances?

Differentiating Factors of Networks

- Types of transmission technology

- Broadcast link

- Broadcast networks have a single communication channel shared by all machines on a network. Packets sent by any machine are received by all others, an address field in the packet specifies the intended recipient. Intended recipients process the packet contents, others simply ignore it.
 - Broadcasting is a mode of operation which allows a packet to be transmitted that every machine in the network must process.

Differentiating Factors of Networks

■ Types of transmission technology

□ Point-to-point links

- Data from sender machine is not seen and processed by other machines
- Point to point networks consist of many connections between individual pairs of machines. Packets travelling from source to destination must visit intermediate machines to determine a route.
- Unicasting is the term used where point-to-point networks with a single sender and receiver pair can exchange data

□ Multicasting

- Transmission to a subset of the machines

Differentiating Factors of Networks

■ By Scale

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	
100 m	Building	
1 km	Campus	Local area network
10 km	City	
100 km	Country	Metropolitan area network
1000 km	Continent	
10,000 km	Planet	

Differentiating Factors of Networks

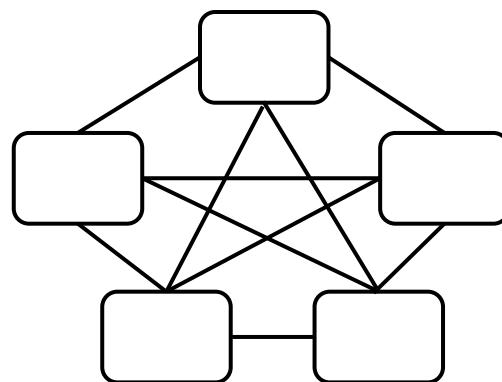
■ By Topology

□ Mesh

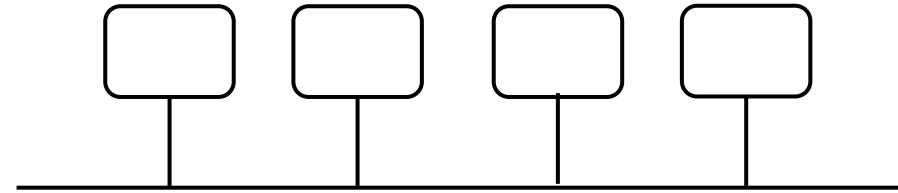
- Fully mesh: each device has a dedicated point-to-point link to every other device.

□ Bus

- All devices are attached to a shared medium.
- Only a single device on the network can transmit at any point in time.
Requires a negotiation mechanism to resolve transmission conflicts.
- e.g. Ethernet is the most common bus network.



(a) mesh

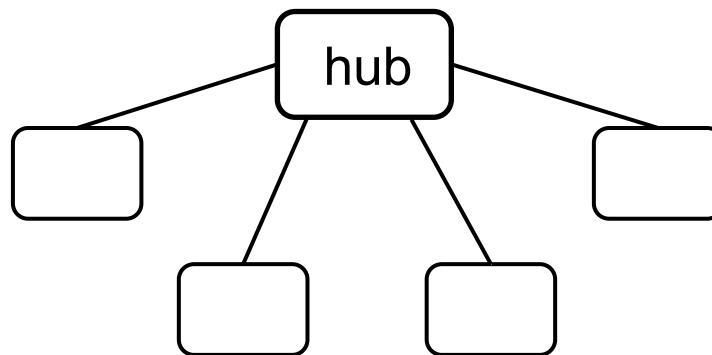


(b) bus

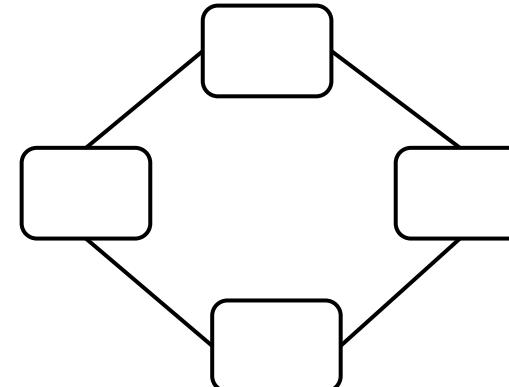
Differentiating Factors of Networks

■ By Topology

- Star
 - All devices are attached to a central device (hub).
- Ring
 - Each device on the ring receives the data from the previous device and forwards it to the next device.
 - Requires access control to resolve propagation queuing.
 - e.g., Token ring.



(c) star

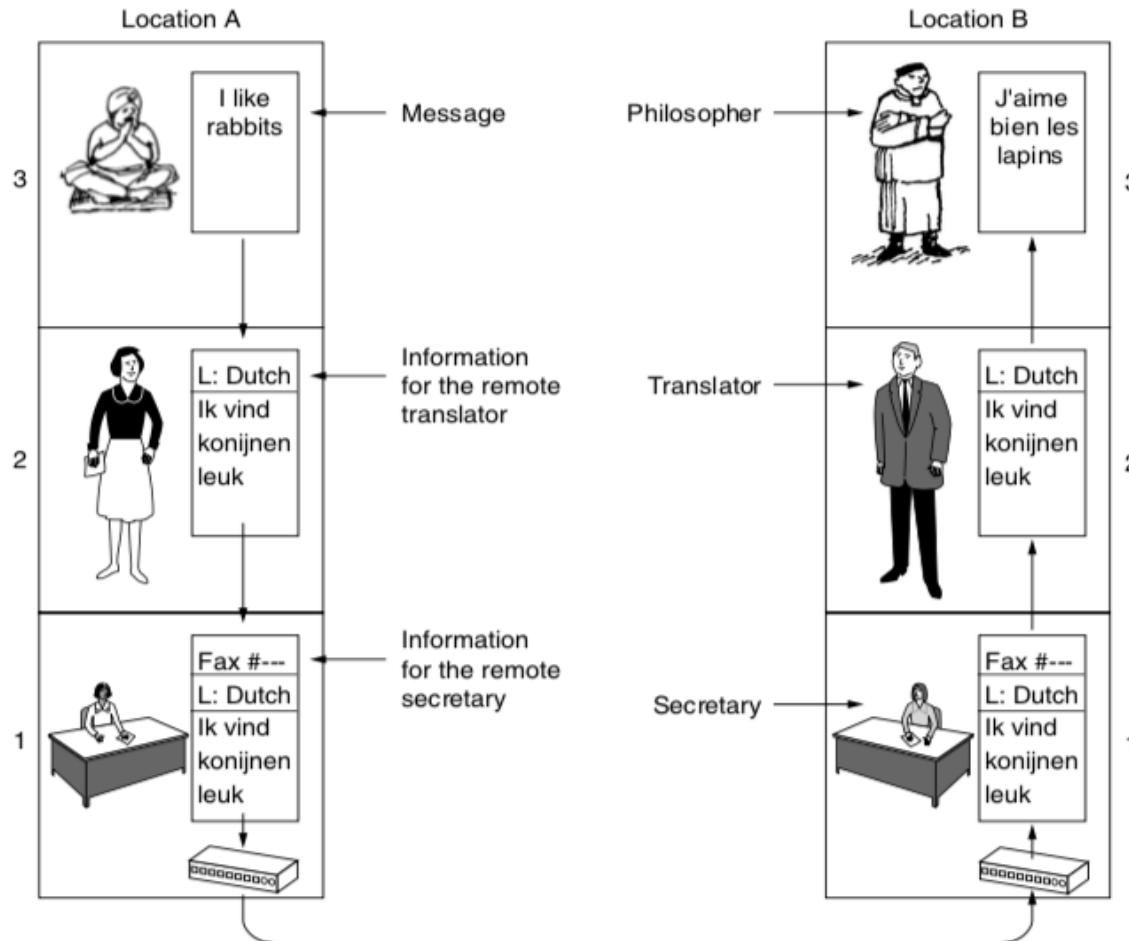


(d) ring

What Makes the Internet Work

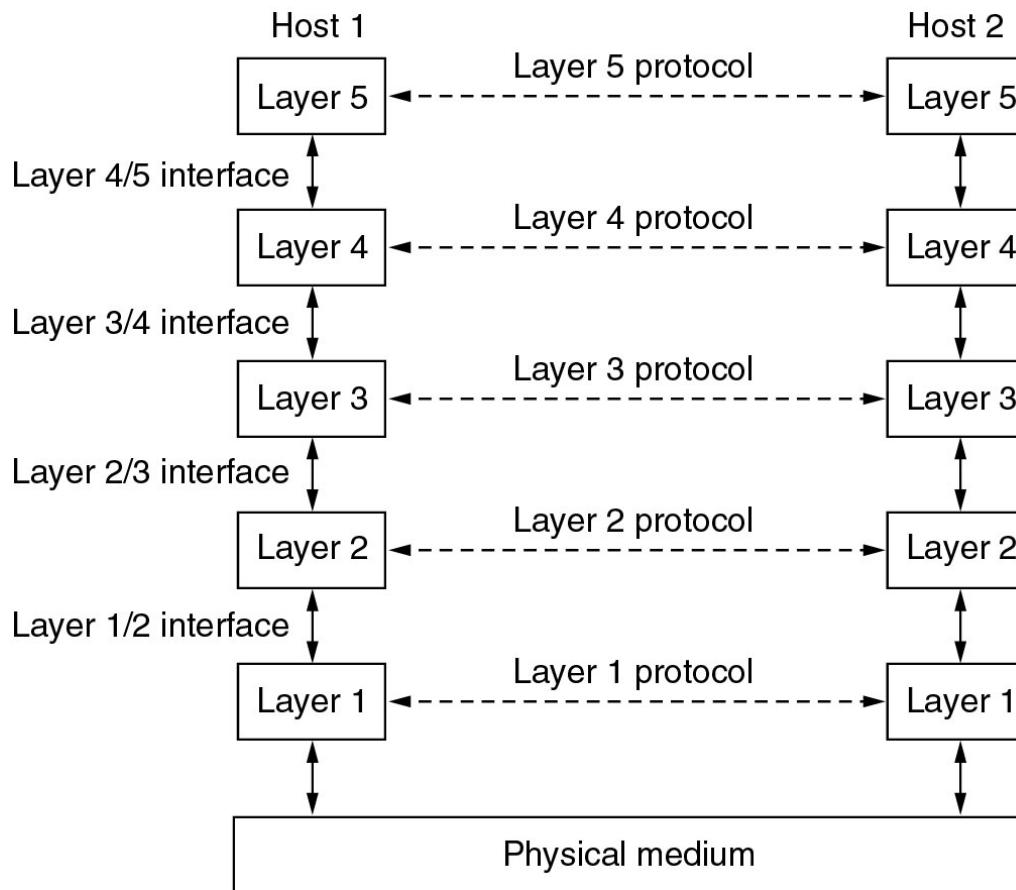
- Protocols, Layers and Services
 - Protocol Hierarchies
 - Design of Layer Models
 - Connection-Oriented and Connectionless Services
 - Services Primitives
 - Services and Protocols
- Network Reference Models
 - Open Systems Interconnect
 - TCP/IP
- Network Standards

The Philosopher-translator-secretary Architecture



Network Software: Protocol Hierarchies (1)

- Layers, protocols and interfaces



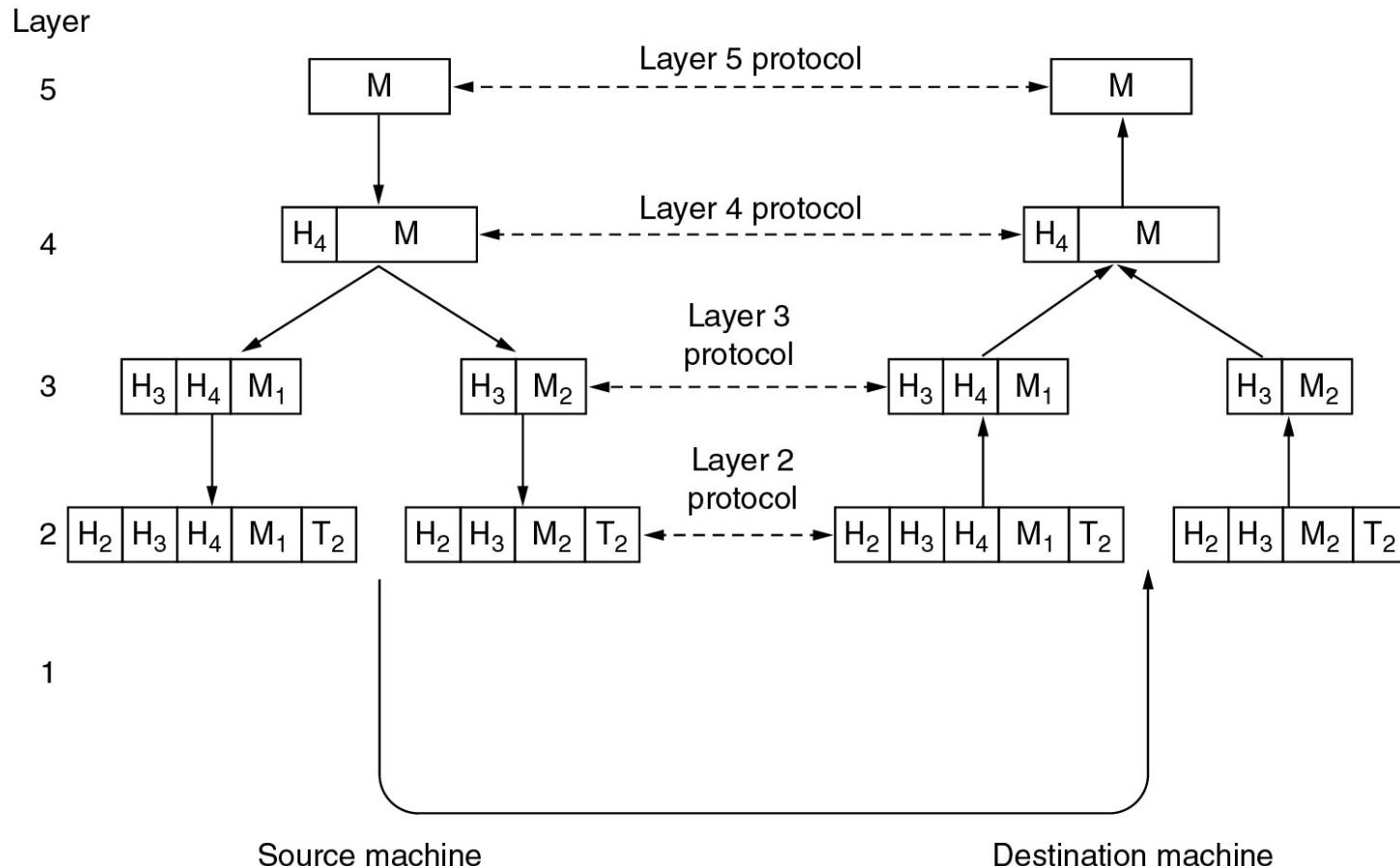
Consider the network as a stack of **layers**

Each layer offers **services** to layers above it through **interface**

Protocol is an agreement between the communicating parties on how communication is to proceed

Network Software: Protocol Hierarchies (2)

- Information flow supporting virtual communication in layer 5



Services

- Choice of service type has a corresponding impact on the reliability and quality of the service
- Connection-Oriented vs. Connectionless
 - Connection-Oriented: connect, use, disconnect (similar to telephone service). Negotiation inherent in connection setup
 - Connectionless: just send (similar to postal service)

Connection-Oriented and Connectionless

■ Six different types of services

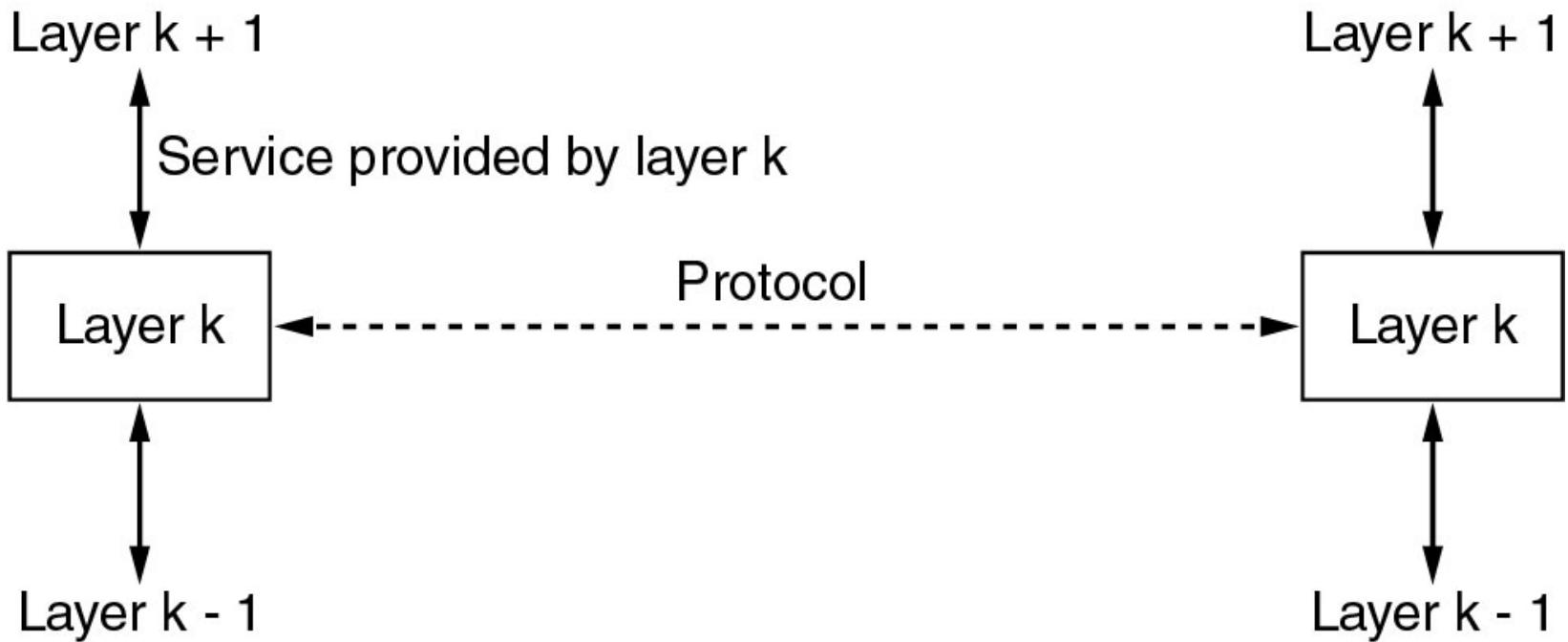
	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Remote login
	Unreliable connection	Digitized voice
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Registered mail
	Request-reply	Database query

Service Primitives

- Primitives are a formal set of operations for services
- The number and type of primitives in any particular context depends on the nature of service - in general more complex services require more service primitives
- Six service primitives for implementing a simple connection-oriented service

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Relationship of Services and Protocols



Relationship of Services and Protocols

- **Service = set of primitives that a layer provides to a layer above it**
 - Provided through the interfaces between layers (service provider vs service users)
 - Defines what operations the layer is prepared to perform on behalf of its users
 - It says nothing about how these operations are implemented
- **Protocol = a set of rules governing the format and meaning of packets that are exchanged by peers within a layer**
 - Packets sent between peer entities

Reference Models

- The OSI Reference Model
- The TCP/IP Reference Model
- A Comparison of OSI and TCP/IP
- A Critique of the OSI Model and Protocols
- A Critique of the TCP/IP Reference Model

Why do we need a reference model?

- A reference model provides a **common baseline for the development** of many services and protocols by independent parties
- Since networks are very complex systems, a reference model can serve to **simplify the design process**
- It's engineering *best practice* to have an **"abstract" reference model**, and corresponding implementations are always required for validation purposes

Week 1 – Introduction to Networking Continued

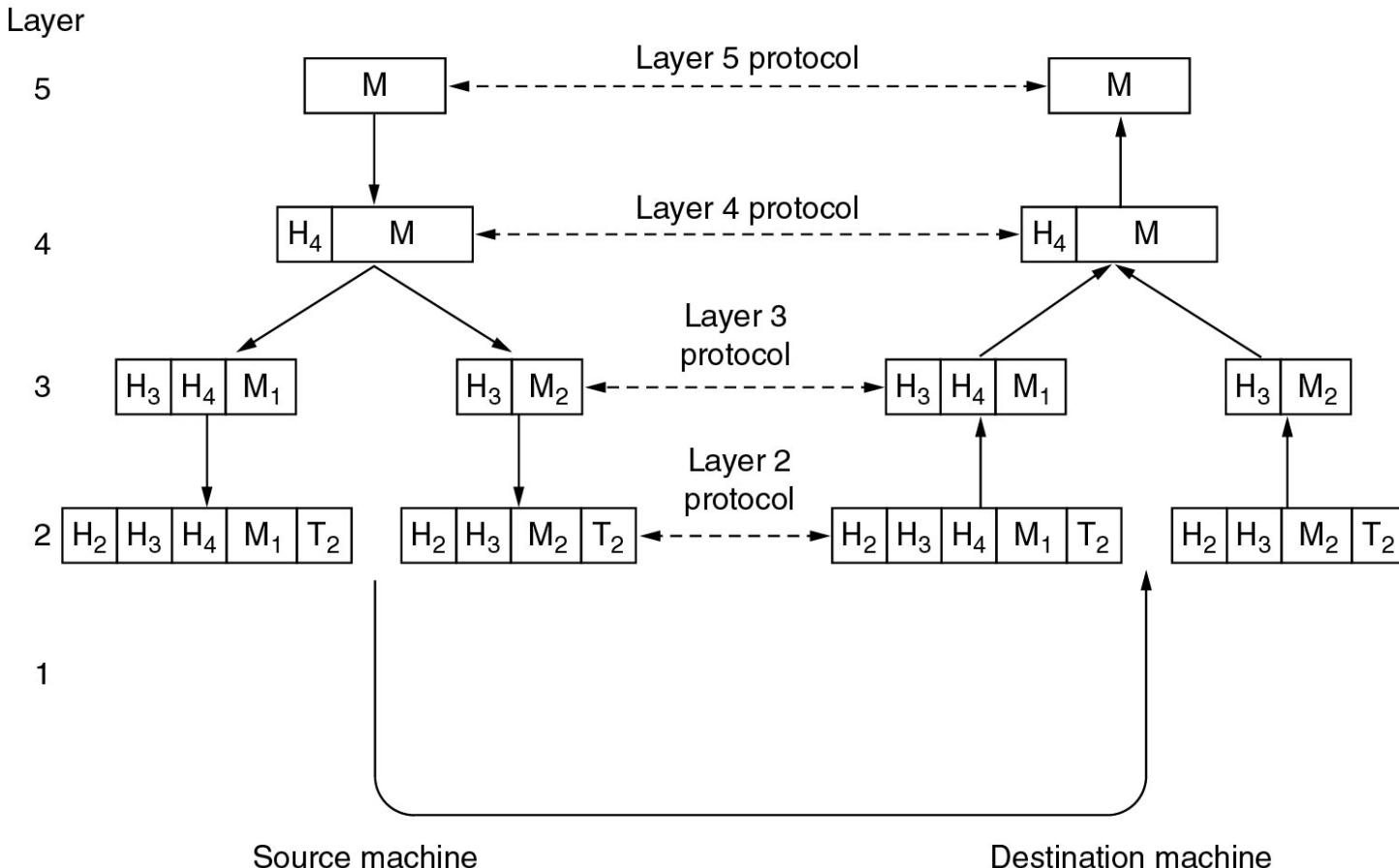
COMP90007 Internet Technologies

Lecturer: Ling Luo

Semester 2, 2020

Recap: Protocol Hierarchies

- Example information flow supporting virtual communication in layer 5



Services

- Choice of service type has a corresponding impact on the reliability and quality of the service
- Connection-Oriented vs. Connectionless
 - Connection-Oriented: connect, use, disconnect (similar to telephone service). Negotiation inherent in connection setup
 - Connectionless: just send (similar to postal service)

Connection-Oriented and Connectionless

■ Six different types of services

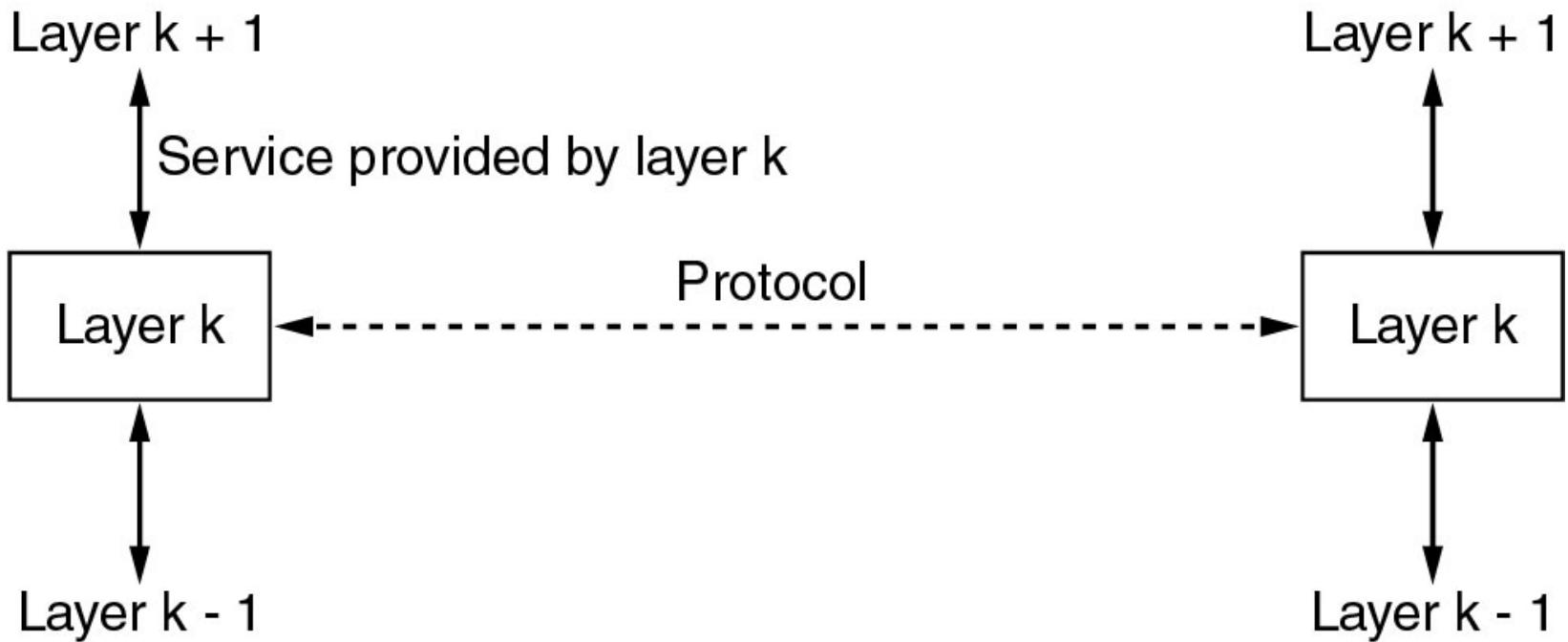
	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Remote login
	Unreliable connection	Digitized voice
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Registered mail
	Request-reply	Database query

Service Primitives

- Primitives are a formal set of operations for services
- The number and type of primitives in any particular context depends on the nature of service - in general more complex services require more service primitives
- Six service primitives for implementing a simple connection-oriented service

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Relationship of Services and Protocols



Relationship of Services and Protocols

Object-Oriented Programming

```
public class Car
{
    public int fuel;
    public int speed;
    public String plate_num;
```

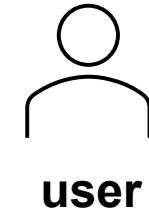
services

protocols

```
        public void accelerate(int num)
        {
            ...
        }
        public void decelerate (int num)
        {
            ...
        }
    }
```

Interface:

- Comments
- Headings of methods



user

Relationship of Services and Protocols

- **Service = set of primitives that a layer provides to a layer above it**
 - Provided through the interfaces between layers (service provider vs service users)
 - Defines what operations the layer is prepared to perform on behalf of its users
 - It says nothing about how these operations are implemented
- **Protocol = a set of rules governing the format and meaning of packets that are exchanged by peers within a layer**
 - Packets sent between peer entities

Reference Models

- The OSI Reference Model
- The TCP/IP Reference Model
- A Comparison of OSI and TCP/IP
- A Critique of the OSI Model and Protocols
- A Critique of the TCP/IP Reference Model

Why do we need a reference model?

- A reference model provides a **common baseline for the development** of many services and protocols by independent parties
- Since networks are very complex systems, a reference model can serve to **simplify the design process**
- It's engineering *best practice* to have an **"abstract" reference model**, and corresponding implementations are always required for validation purposes

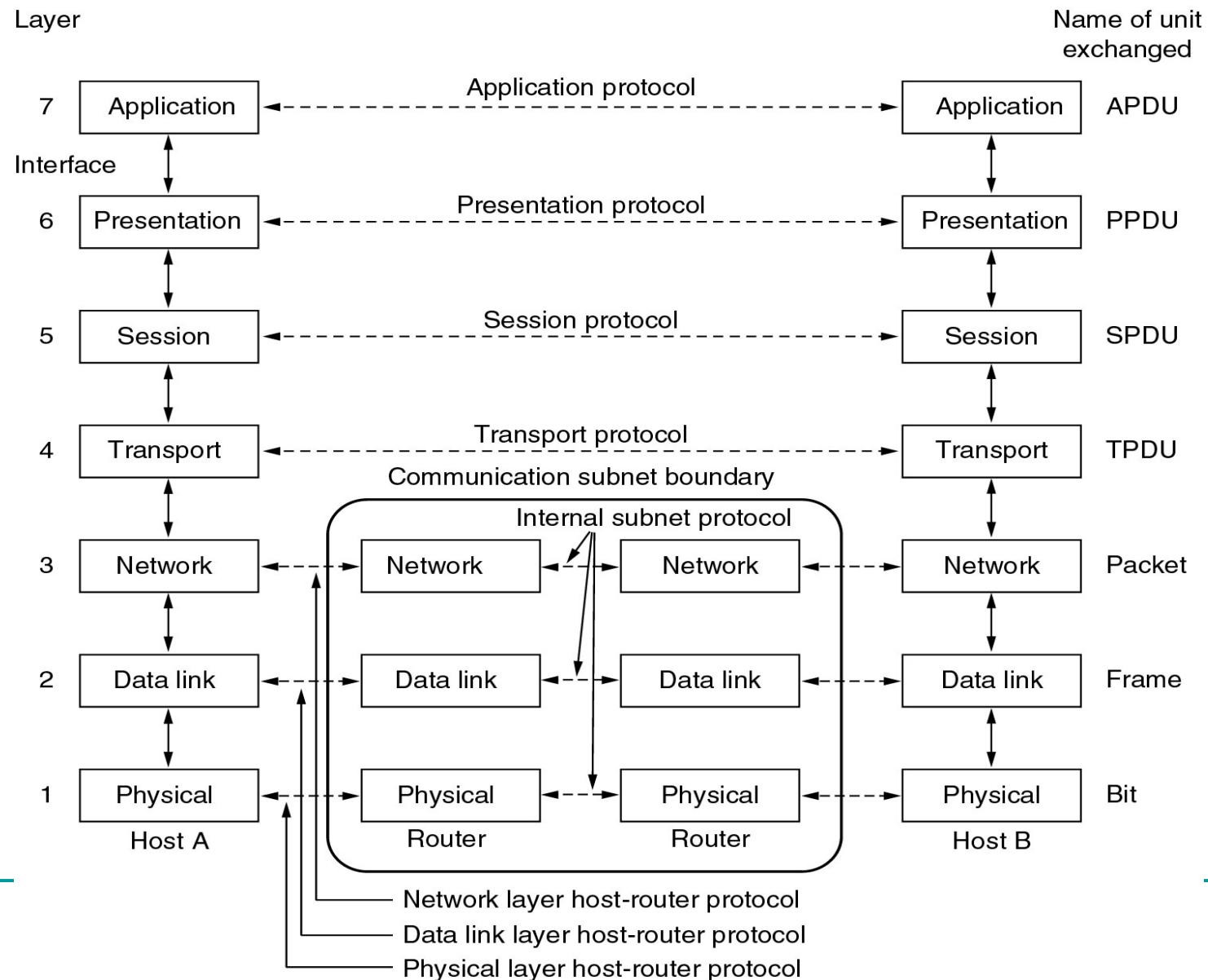
OSI Reference Model

- Open Systems Interconnection (OSI)
- ISO, John Day (revised 1995)
- 7 Layers
- Layer divisions based on principled decisions

OSI Layer Division Principles

1. A layer should be created where a different **abstraction** is needed.
2. Each layer should **perform a well defined function.**
3. The layer boundaries should be chosen to **minimise the information flow across the interfaces.**
4. The number of layers should be **large enough that** distinct functions need not to be thrown together in the same layer out of necessity; and **small enough that** the architecture does not become unwieldy.
5. The function of each layer should be chosen with a view toward defining **internationally standardised protocols.**

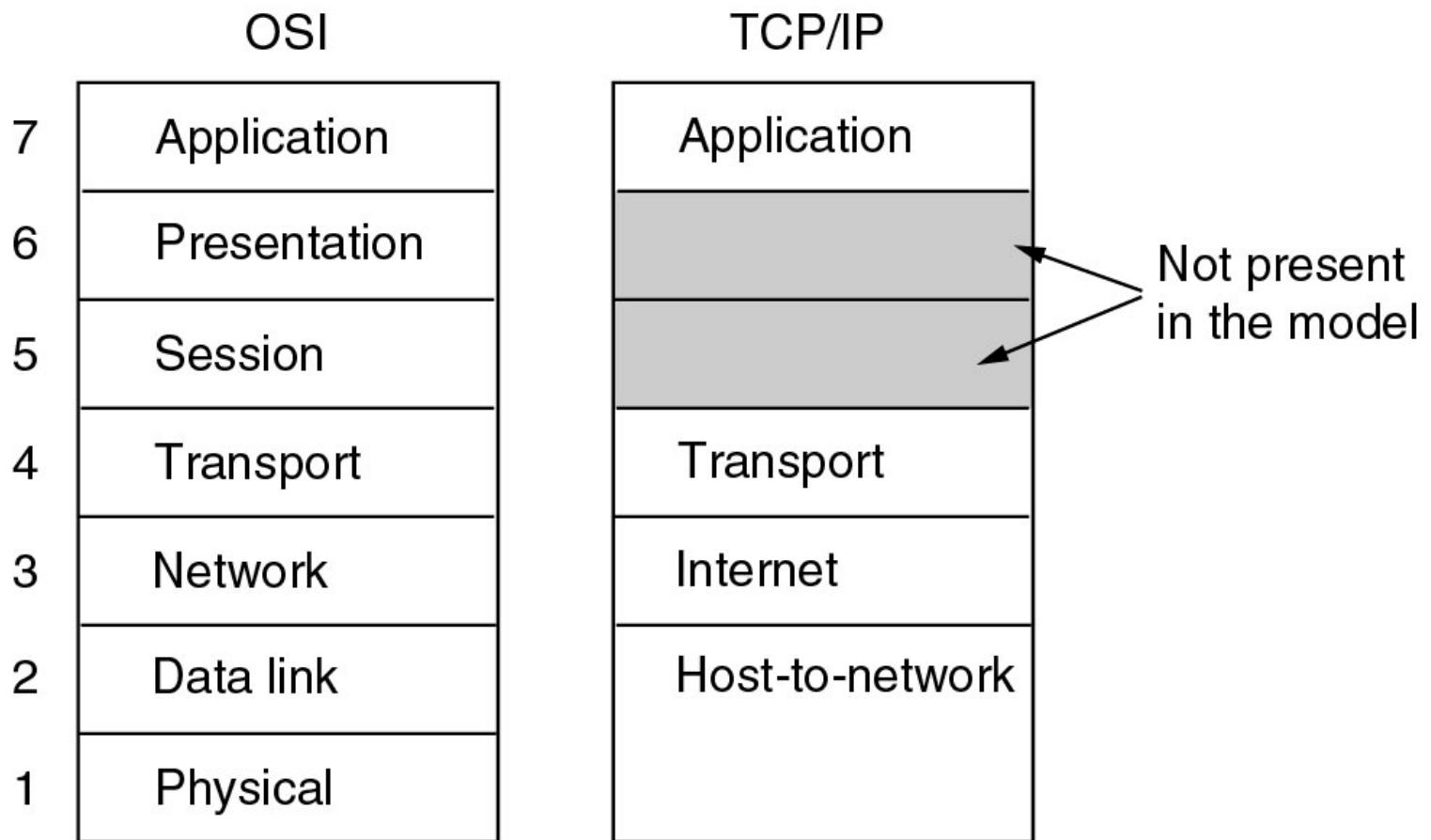
OSI Reference Model



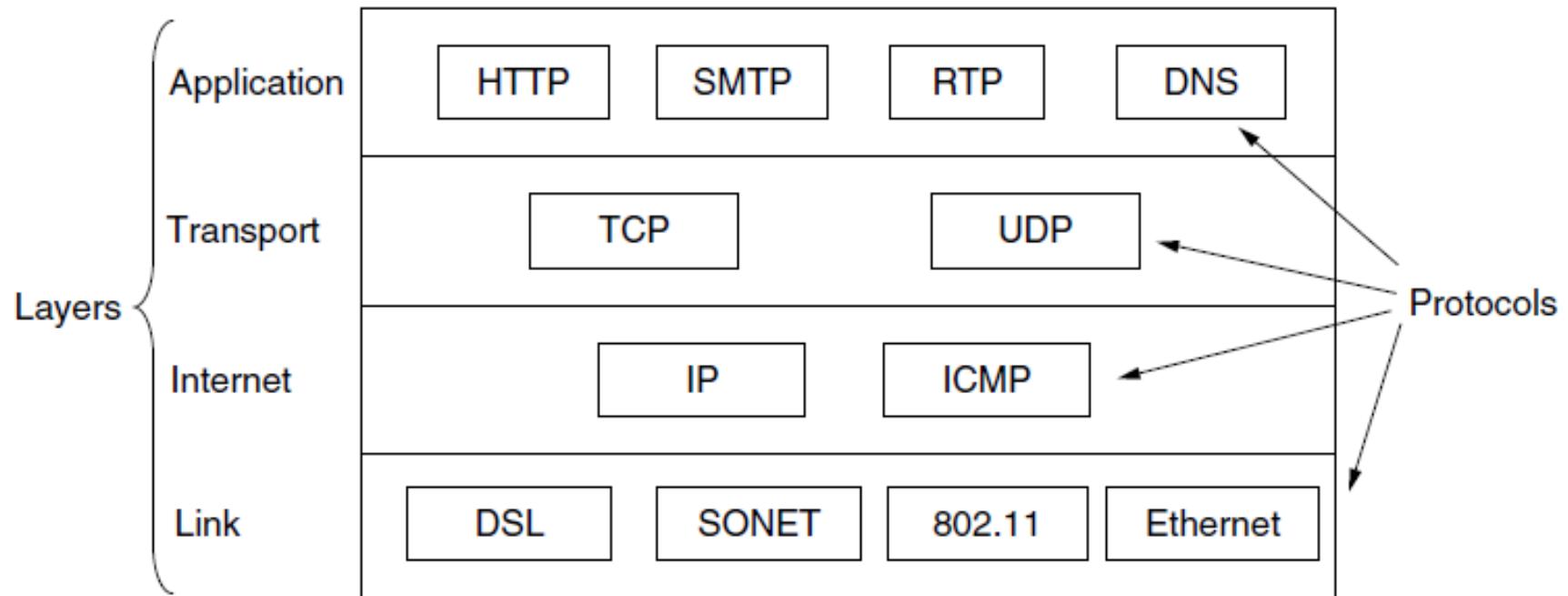
TCP/IP Reference Model

- Transmission Control Protocol/Internet Protocol
- Vint Cerf & Bob Kahn (1974)
- 4 layers

TCP/IP Reference Model (2)



TCP/IP Reference Model (3)



Comparing OSI and TCP/IP Models

- Different numbers of layers
- OSI distinguishes the following three concepts explicitly
 - Services
 - Interfaces
 - Protocols
- TCP/IP has successful protocols

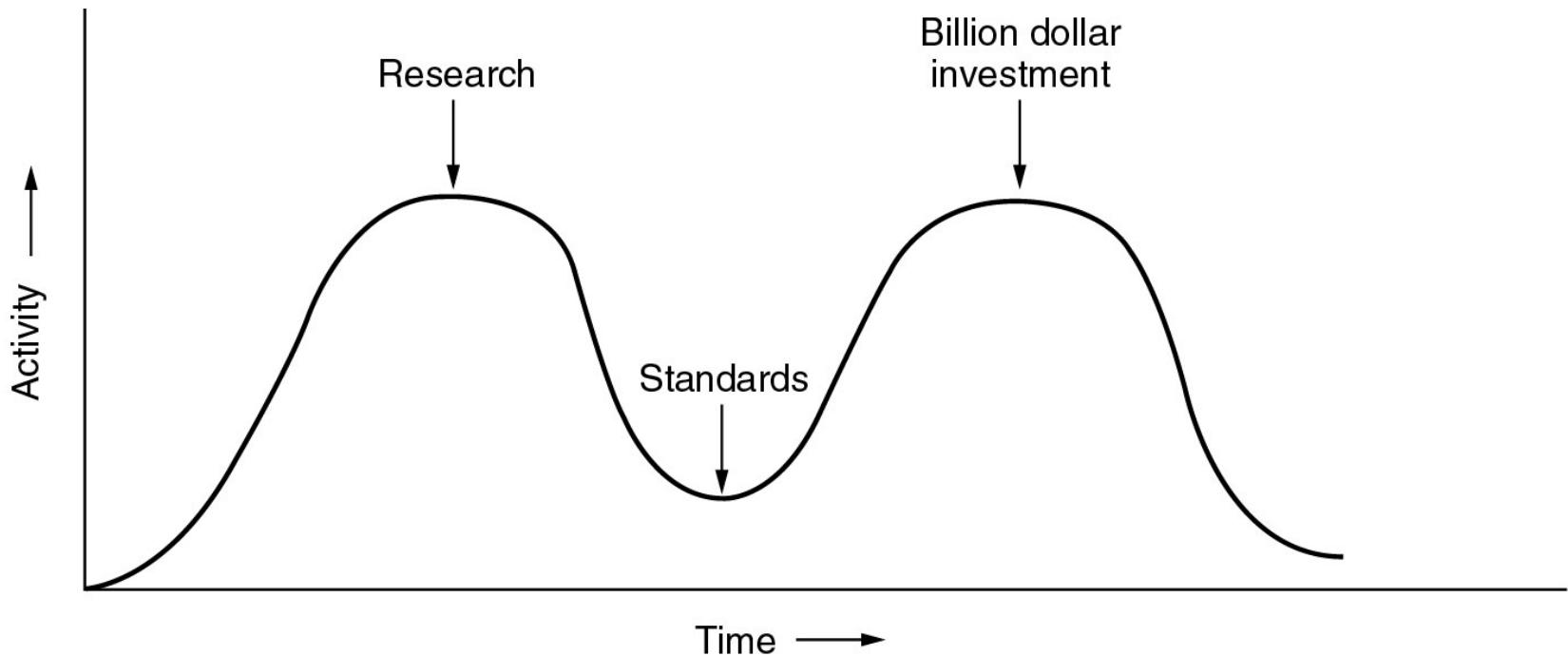
A Critique of the OSI Model

Why OSI did not take over the world?

- Bad timing
- Bad technology
- Bad implementations
- Bad politics

A Critique of the OSI Model: Bad Timing

- When is good timing for a standard?



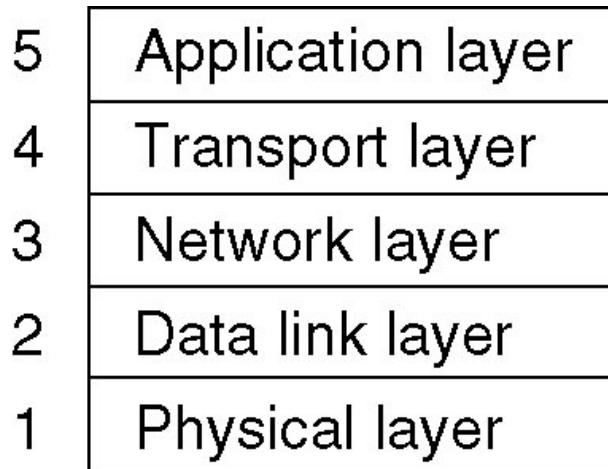
A Critique of the TCP/IP Model

Problems:

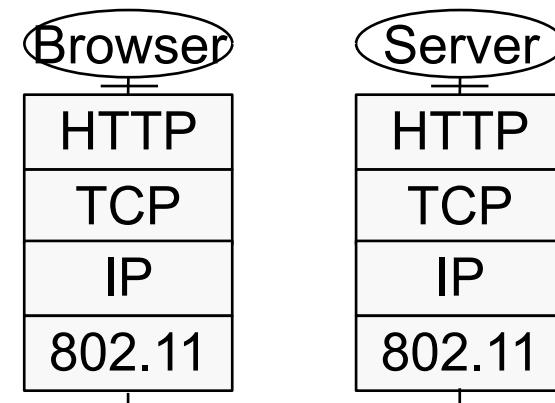
- Not a general model
- Service, interface, and protocol not distinguished
- Host-to-network “layer” not really a layer – interface between network and data link layers
- No mention of physical and data link layers
- Minor protocols deeply entrenched, hard to replace

Hybrid Model

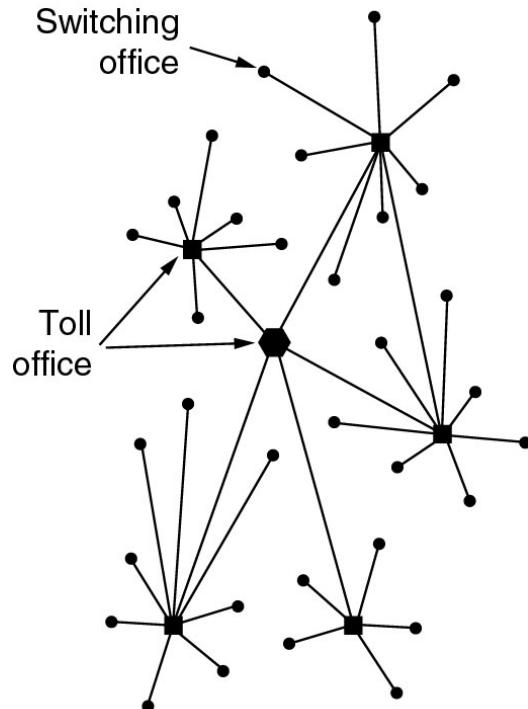
- The hybrid reference model to be used in this semester



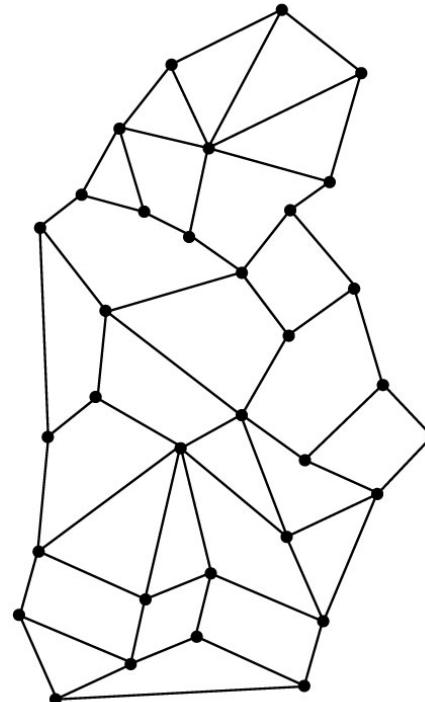
A typical network scenario



Origins of Internet: The ARPANET



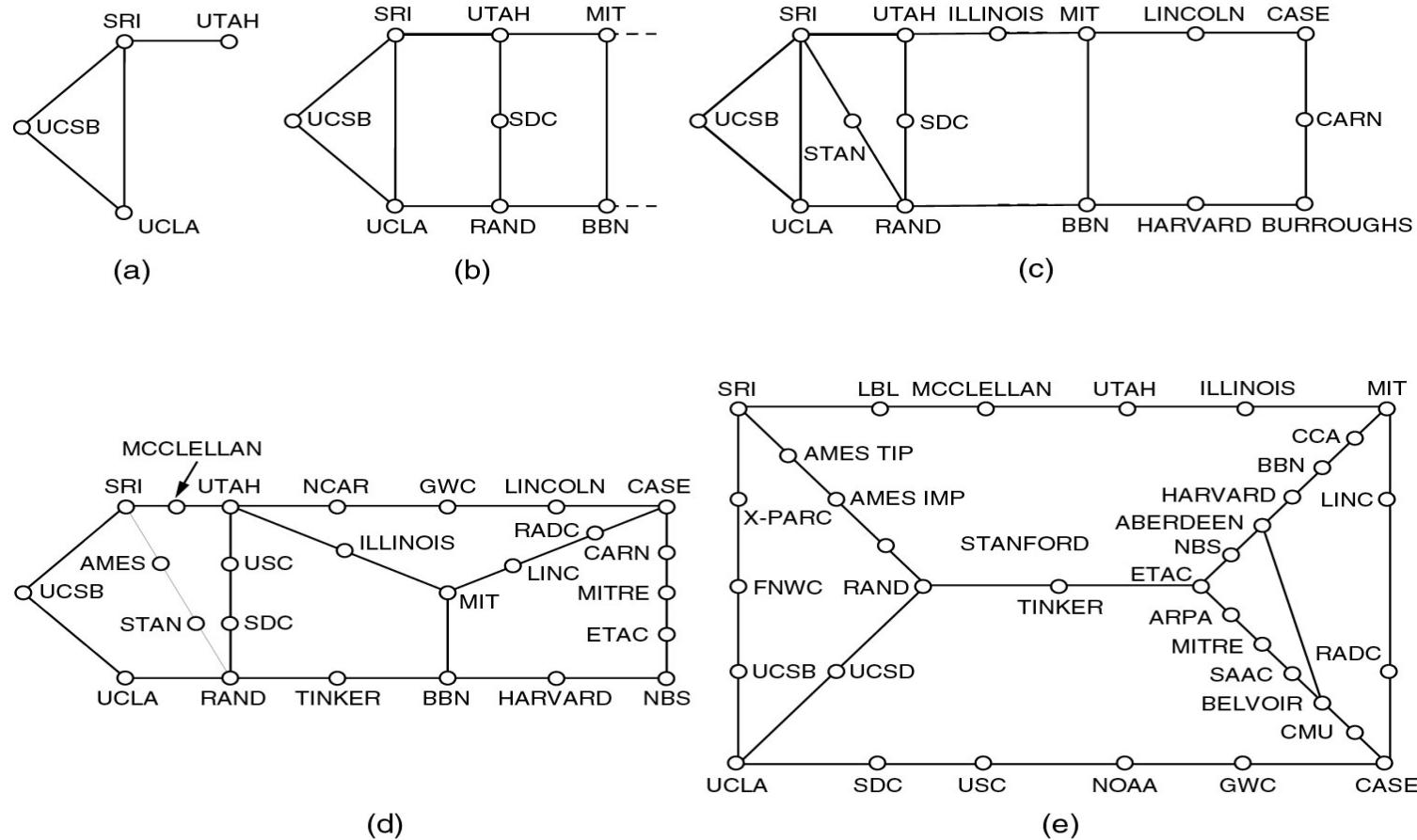
(a)



(b)

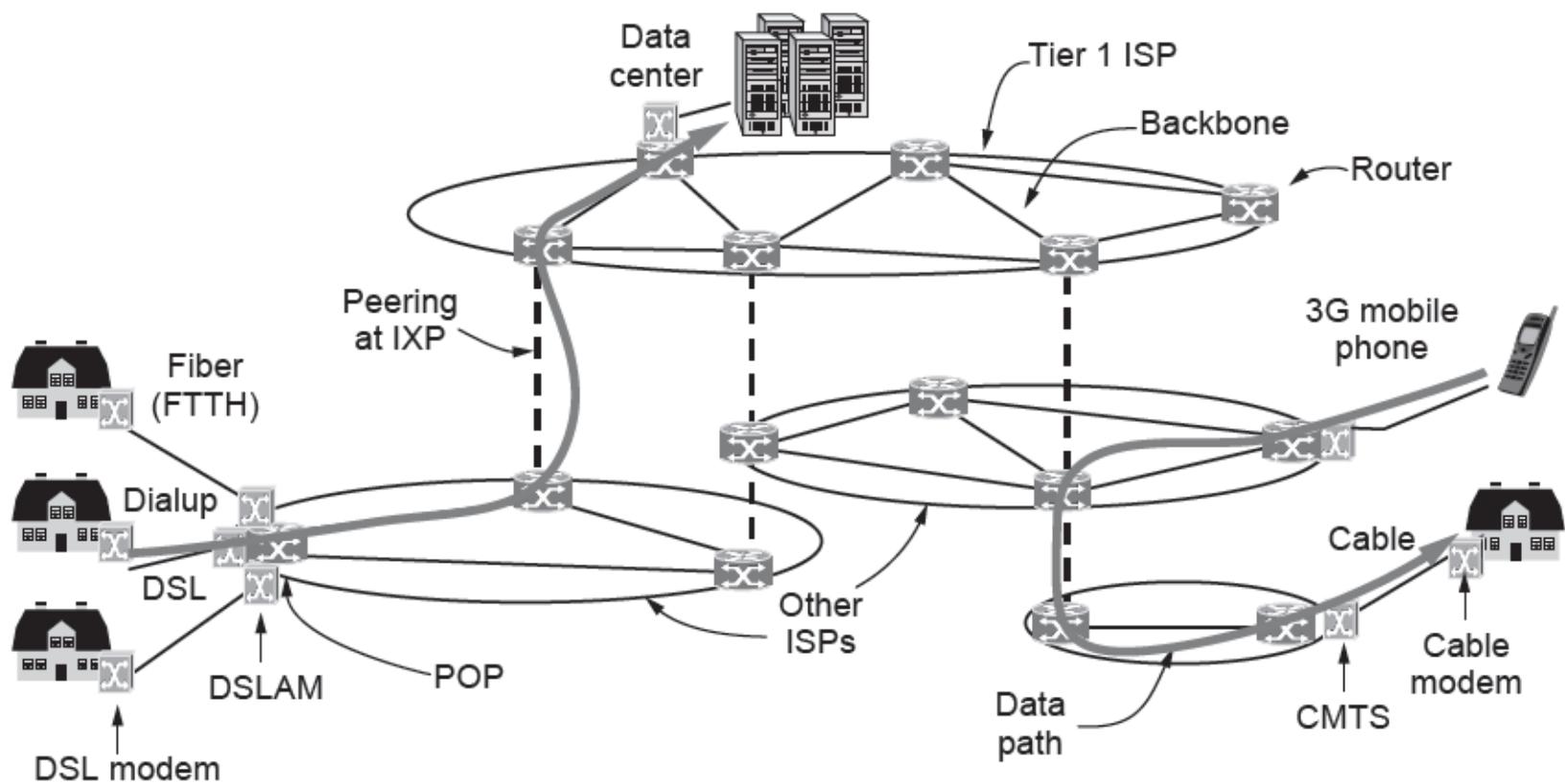
- (a) Structure of the telephone system.
- (b) Baran's proposed distributed switching system.

The ARPANET



- Growth of the ARPANET (a) December 1969. (b) July 1970.
 - (c) March 1971. (d) April 1972. (e) September 1972.

Architecture of the Internet



Network Standardisation

Body	Area	Examples
ITU (International Telecommunication Union)	Telecommunications	ADSL PON MPEG4
IEEE (Institute of Electrical and Electronics Engineers)	Communications	Ethernet WiFi
IETF (Internet Engineering Task Force)	Internet	HTTP/1.1 DNS
W3C (The World Wide Web Consortium)	Web	HTML5 standard

Week 2 – Physical Layer

COMP90007 Internet Technologies

Lecturer: Ling Luo

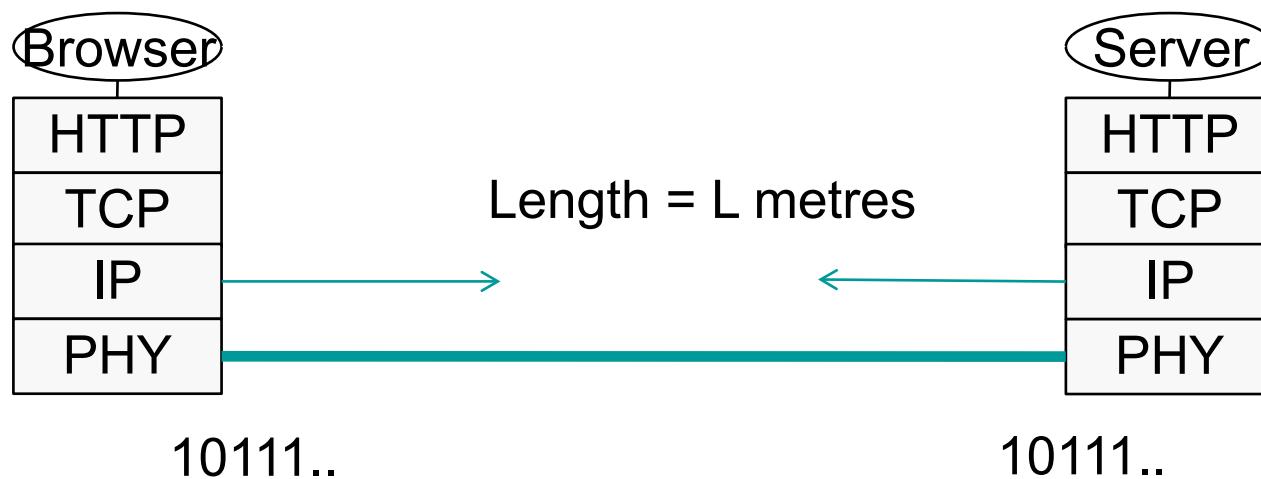
Semester 2, 2020

What is the Physical Layer ?

- Recall the layer hierarchy from network reference models
 - The physical layer is the lowest Layer in OSI model
 - The physical layer's properties in TCP/IP model are in the “host-to-network” division.
- The physical layer is concerned with the electrical, timing and mechanical interfaces of the network
 - Electrical: voltage levels, signal strength ...
 - Timing: data rate ...
 - Mechanical: material, cable length ...

Link Model

- We can abstract the physical channel as a link
- Simplified Link Model: Consider the network as a connected link between computers



Link Model

- **Bandwidth** is usually treated as rate of transmission in bits/second.
- **Delay** is the time required for the first bit to travel from computer A to computer B.

Example

- We need about 1 kbit/sec to transmit voice.
- Bandwidth of single mode fibre can reach 1 Tbit/sec.
- How many voice calls can be transmitted through a Fibre Optic Cable?

Message Latency

- Latency is the time delay associated with sending a message over a link
- This is made of up two parts
 - **Transmission delay**
 - $T\text{-delay} = \text{Message in bits} / \text{Rate of transmission}$
 - $= M/R$ seconds
 - **Propagation delay**
 - $P\text{-delay} = \text{length of the channel} / \text{speed of signals}$
 - $\text{Length} / \text{Speed of signal}$ ($2/3$ of speed of light for wire)
 - **Latency = L = M/R + P-delay**

Example-1

- A home computer is connected to an ISP server through 56 K bps modem. Assuming a frame size of 5600 bits, compute P-Delay and T-Delay for the link. Assume speed of signal = $2/3 C$ and length of the link is 5 K metres.

Example-2

- Now for the previous question, assume a countrywide optical broadband link of length 1000 kms of bandwidth 100 M bits/sec. Assuming a frame size of 5600 bits, compute P-Delay and T-Delay for the link. Assume speed of signal = $C = 300000$ km/sec.

The Bandwidth Revolution?

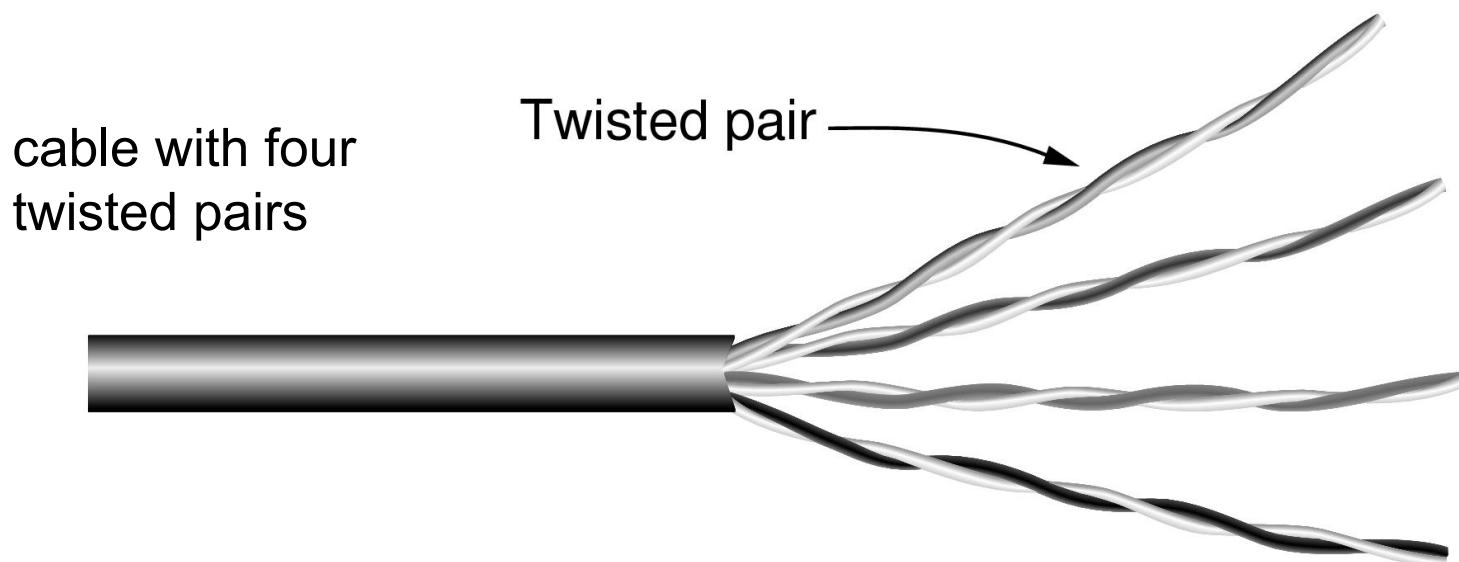
- CPU speeds increase by a factor of ~20 per decade
 - 1981: PC 4.77MHz vs 2001: PC 2 GHz
- Current CPU speed now approaching physical limits - constrained by physical properties pertaining to granularity of engraving on silicon
- Evolutionary steps in available bandwidth:
Bandwidth increases by a factor of ~125 per decade
(1981: Modem 56kbps vs 2001: Net 1Gbps)
- Current bandwidth available up to 65 Tbps - vastly exceeding the rate at which we can convert electrical impulses to optical pulses

Transmission Media

- How many different types of physical media can you think of?
- Various physical media can be used to transmit data, but all of them are affected by physical properties
- How far and how much data a medium can carry has a lot to do with signal attenuation.
- Attenuation: the loss or reduction in the amplitude (strength) of a signal as it passes through a medium.

Wires – Twisted Pair

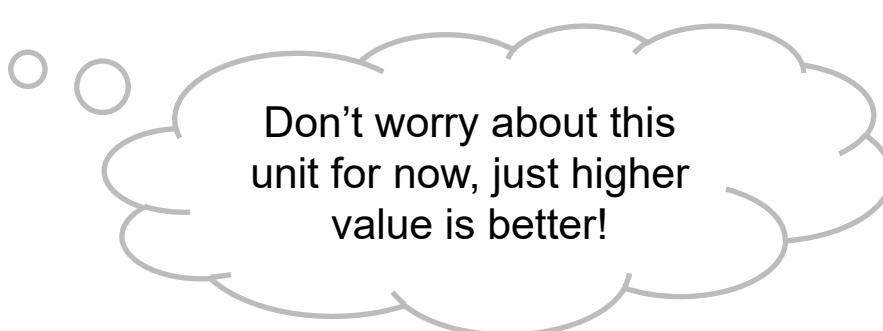
- ❑ Two insulated copper wires, twisted in helical (DNA) form.
- ❑ Twisting reduces interference: canceling out electromagnetic interference from external sources
- ❑ Distance up to <5km, repeaters can extend this distance



cable with four
twisted pairs

Properties and Types of Twisted Pair

- Bandwidth depends on distance, wire quality/ density
- Cat 3 - 2 wires, 4 pairs in sheath, 16MHz
- Cat 5 - 2 wires, 4 pair in sheath, more twists = less interference, higher quality over longer distance, 100 MHz
- Cat 8 – 2000 MHz



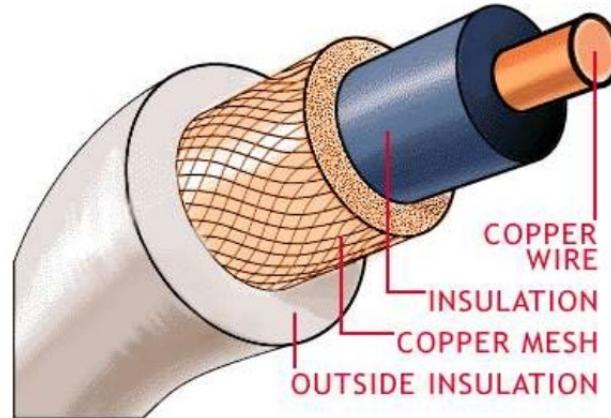
A thought bubble graphic with a light gray outline and a slightly irregular shape. It contains the text "Don't worry about this unit for now, just higher value is better!".

Don't worry about this unit for now, just higher value is better!

Coaxial Cable (Co-ax)

- ❑ Copper core with insulation, mesh, and sheath
- ❑ Better shielding than twisted pair = higher speeds over greater distances
- ❑ Bandwidth approaches 1GHz
- ❑ Still widely used for cable TV/Internet

A diagram of a coaxial cable

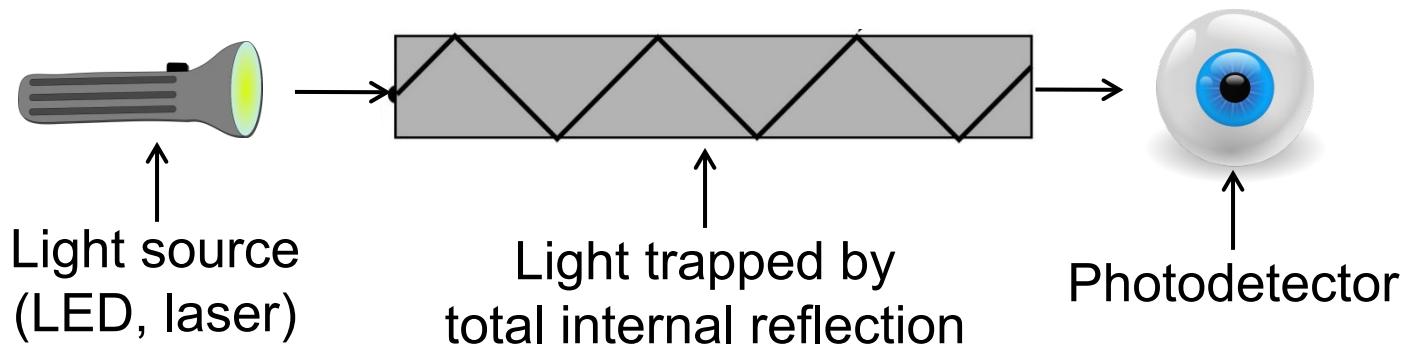


Fibre Optics

- Fibre has enormous bandwidth (THz) and tiny signal loss
- Data transmission over a fibre of glass
- Common for high rates and long distances
 - e.g. backbone links between ISP facilities, Fibre-to-the-Home (FTTH)

Transmission of Light Through Fibre

- 3 components: light source, transmission medium, detector
- Semantics: light = 1, no light = 0 (basic binary system)
- Signalling using LED's or semiconductor lasers
- A detector generates electrical pulse when light hits it
- Refraction between air/silica boundary is compensated for by design - total internal reflection



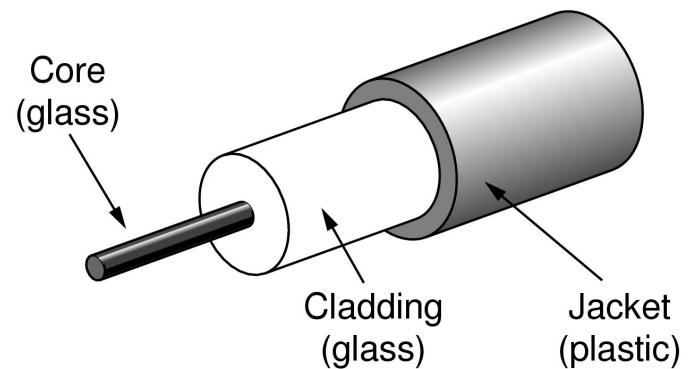
Fibre Optic Cables

Single-mode

- ❑ Narrow core (10um), light can't even bounce around
- ❑ Used with lasers for long distances, e.g., 100km

Multi-mode

- ❑ 50um core, light can bounce
- ❑ Used with LEDs for cheaper, shorter distance links



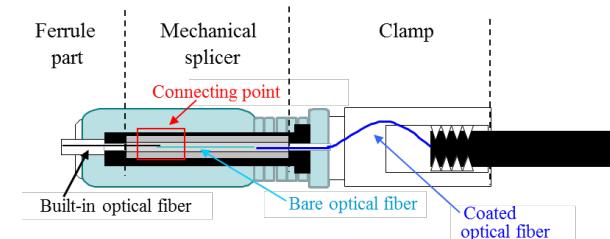
Fibre Optic Connections

- Connectors and Fibre Sockets (10-20% loss)
- Mechanical Splice (10% loss)
- Fusion (<1% loss)



Download from
Dreamstime.com
This watermark comp image is for previewing purposes only.

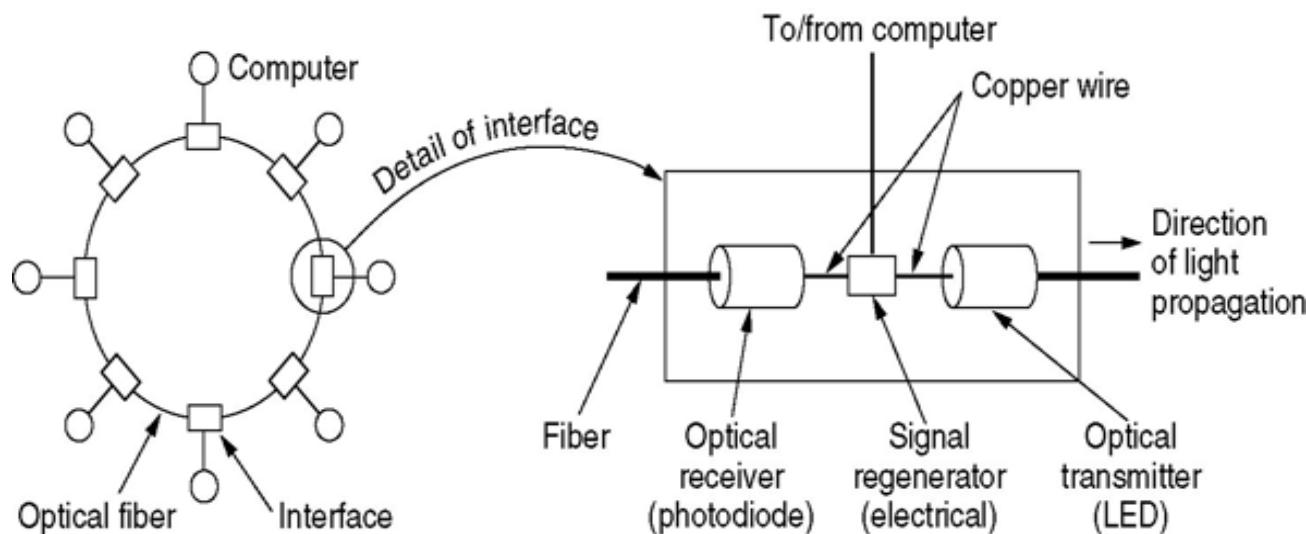
ID 36362829
Spettacolare | Dreamstime.com



Examples: mechanical splice

Fibre Optic Networks

- Fibre optic cable is a scalable network media - LAN, WAN, long distances
- Fibre optic cable networks can be organised either as a ring or as a bus network (series of point to point connections)



Fibre Optic Ring

Comparison: Wires and Fibre

Comparison of the properties of wires and fibre:

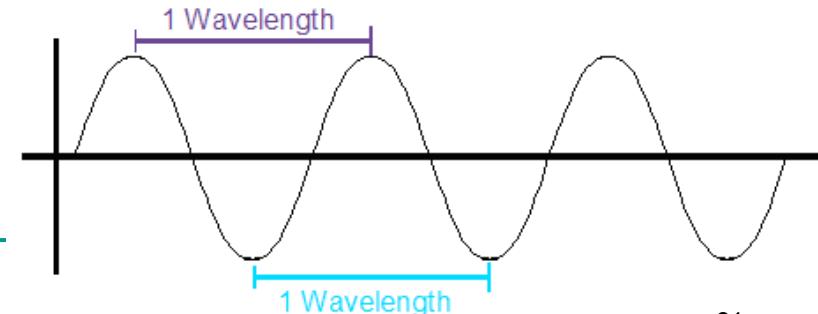
Property	Wires	Fibre
Distance	Short (100s of m)	Long (tens of km)
Bandwidth	Moderate	Very High
Security	Easy to tap	Hard to tap
Cost	Inexpensive	More Expensive
Convenience	Easy to use	Harder to use

Wireless Transmission

- Mobile users requires a mobility enabled network - contrast with the wired networks
- Wireless networks can provide advantages even in fixed location environments
- Wireless data transmission networks have a common basis - electromagnetic wave propagation
 - Unlike previous media, wireless signals are broadcasted over a region
 - Potential signal collisions – Need regulations

Basics of Electromagnetic Waves

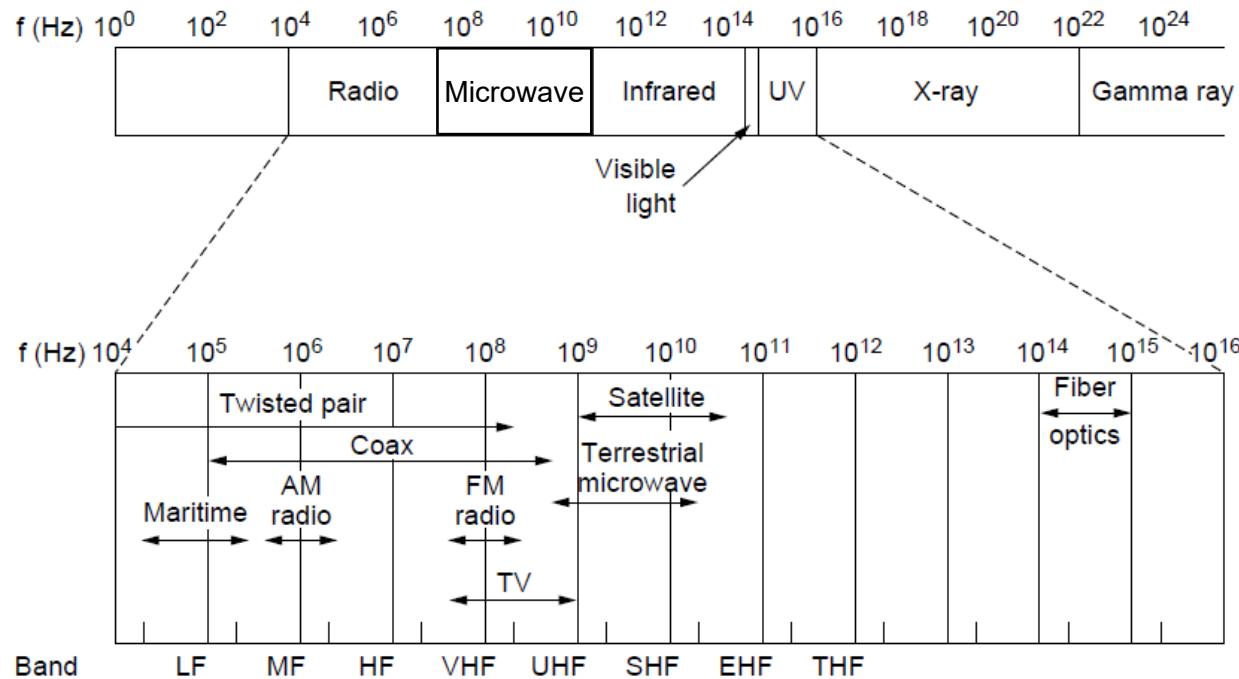
- **Frequency:** Number of oscillations per second of a wave, measured in Hertz (Hz).
- **Wavelength:** Distance between two consecutive minima or maxima.
- **Speed:** All EM waves travel at the same speed - the speed of light $\sim 3 \times 10^8$ m/s
- Fundamental relationship:
 - Wavelength \times Frequency = Speed of Light
 - Units: $(\text{m}) \times (\text{1/s}) = (\text{m/s})$



Electromagnetic Spectrum

Different bands have different uses:

- Radio: wide-area broadcast
- Microwave: LANs and 3G/4G
- Infrared/Light: line-of-sight



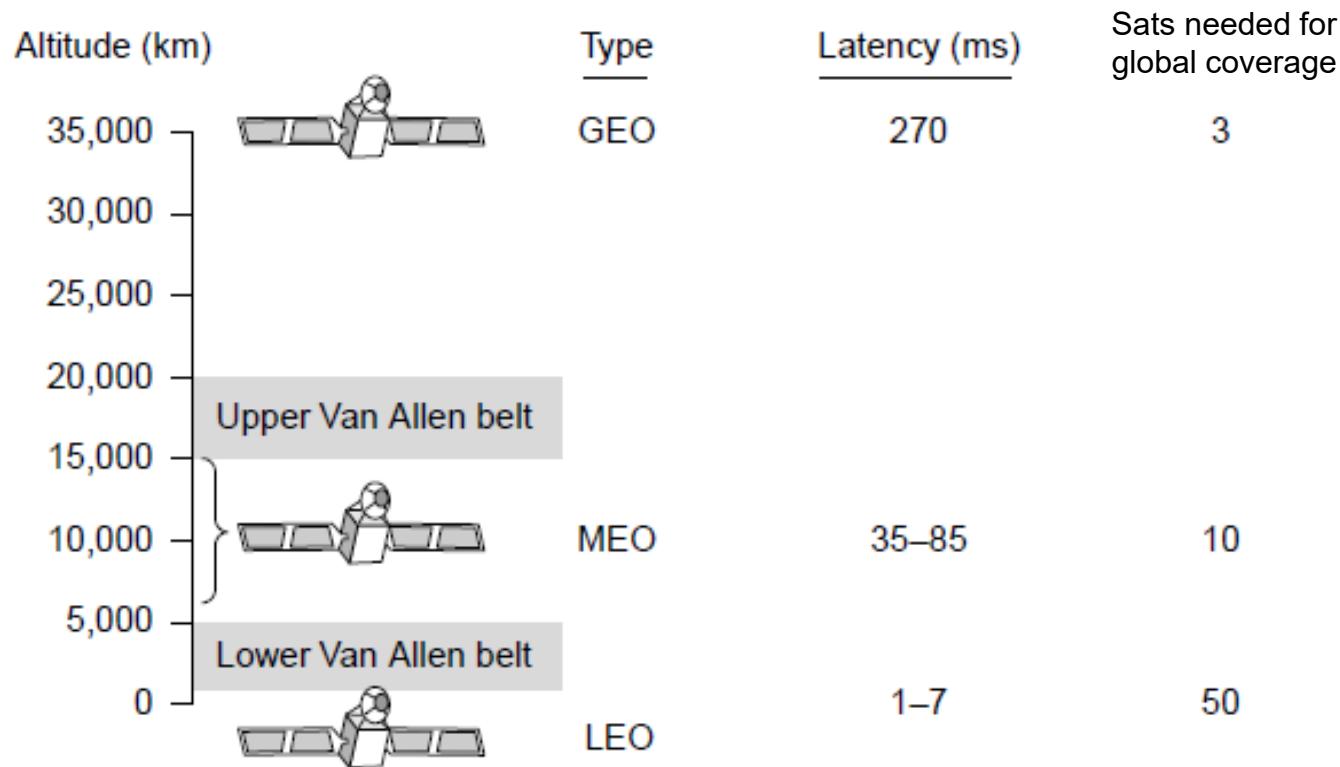
Communication Satellites

Satellites are effective for broadcast distribution and anywhere/anytime communications

- Types of satellites:
 - Geostationary (GEO) Satellites
 - Medium-Earth Orbit (MEO) Satellites
 - Low-Earth Orbit (LEO) Satellites

Types of Satellites

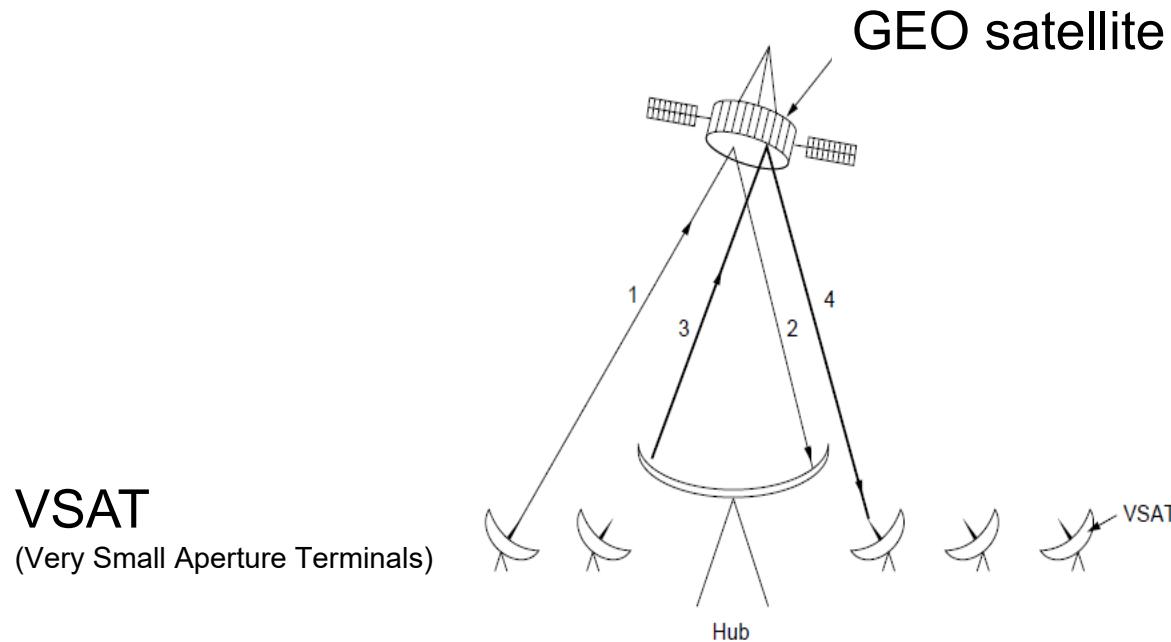
Satellites and their properties vary by altitude



Geostationary Satellites

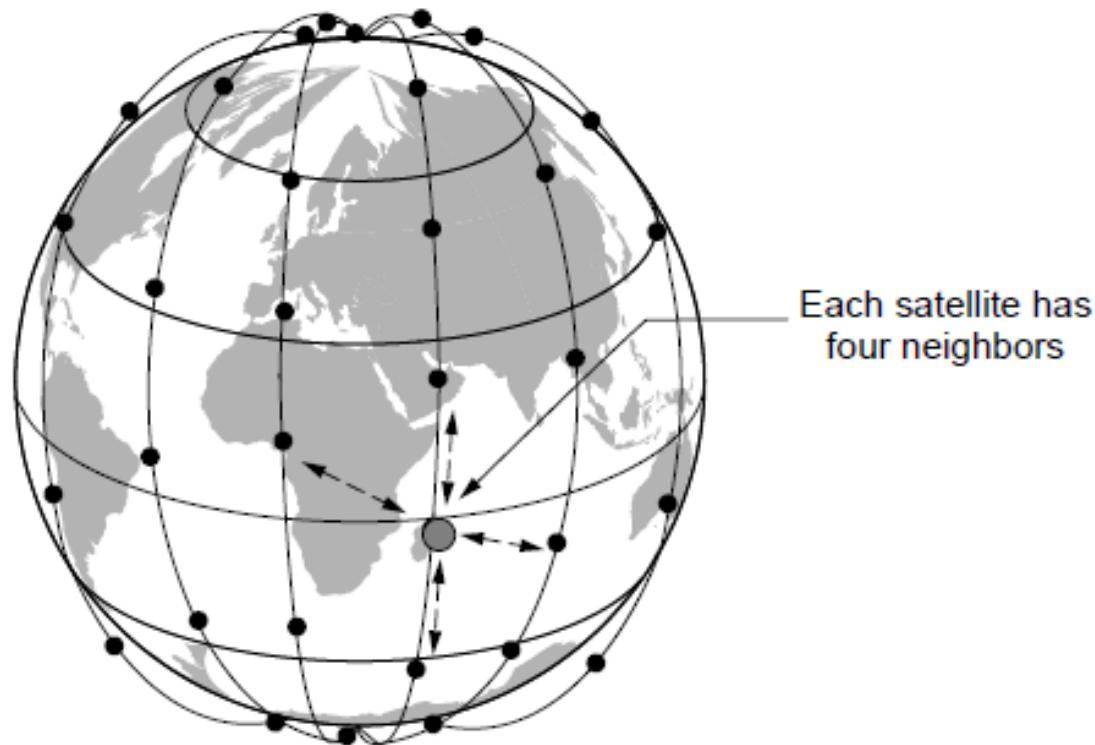
GEO satellites orbit 35,000 km above a fixed location

- VSAT (computers) can communicate with the help of a hub
- Different bands (L, S, C, Ku, Ka) in the GHz are in use but may be crowded or susceptible to rain.



Low-Earth Orbit Satellites

Systems such as Iridium use many low-latency satellites for coverage and route communications via them



Satellite vs. Fibre

Satellite:

- + Can rapidly set up anywhere/anytime communications (after satellites have been launched)
- + Can broadcast to large regions
- Limited bandwidth and interference to manage

Fibre:

- + Enormous bandwidth over long distances
- Installation can be difficult in rural areas

Wireless vs. Wires/Fibre

Wireless:

- + Naturally supports mobility
- + Naturally supports broadcast
- + Easy and inexpensive to deploy
- Transmissions interfere and must be managed
- Signal strengths and data rates vary greatly

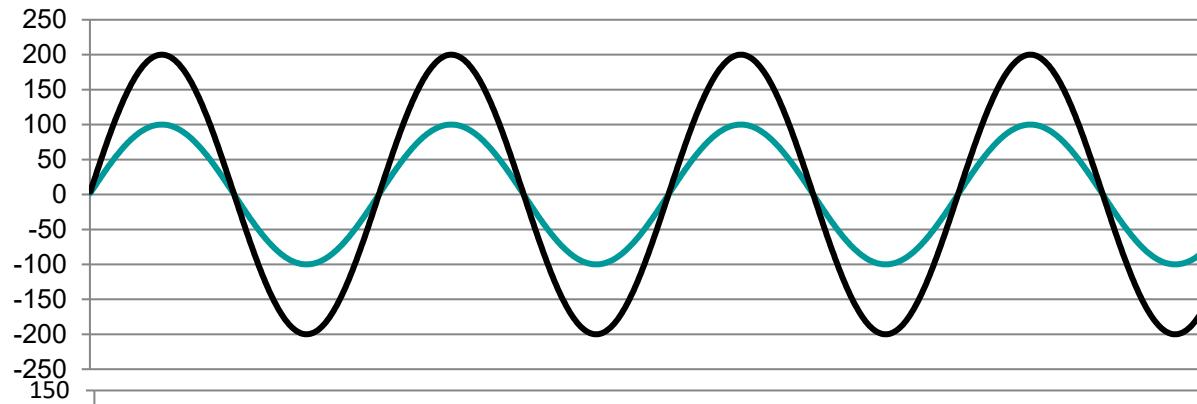
Wires/Fibre:

- + Easy to engineer a fixed data rate over point-to-point links
- Can be expensive to deploy, esp. over distances
- Doesn't readily support mobility or broadcast

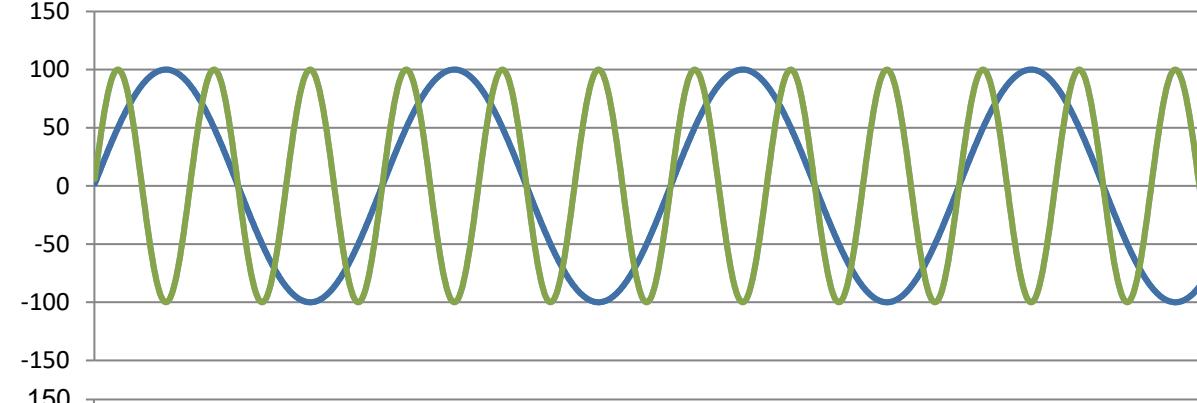
Data Communication using Signals (1)

- Information is transmitted by varying a physical property e.g. voltage, current
- For a sinewave :
function: $c * \sin(a * t + b)$
c: amplitude, $a/(2\pi)$:frequency and b:phase
can change the behaviour of the function.

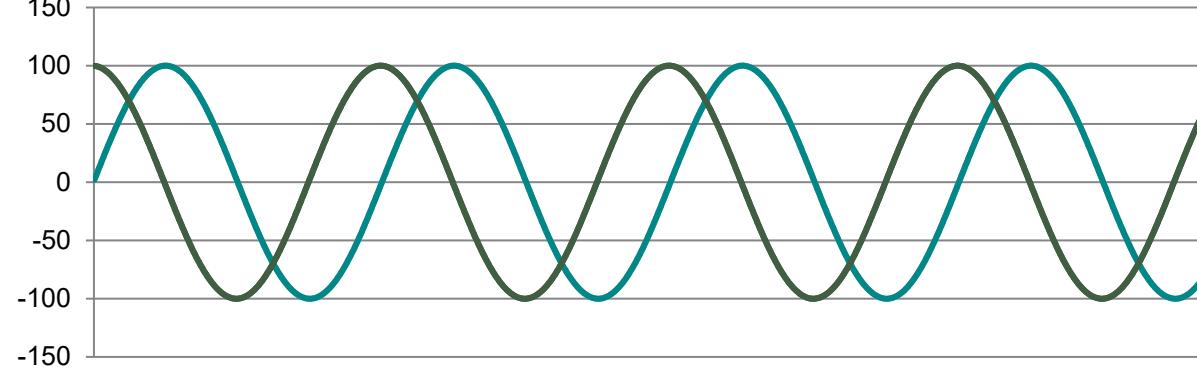
Data Communication using Signals (2)



Change in
Amplitude



Change in
Frequency



Change in
Phase

Digital Modulation

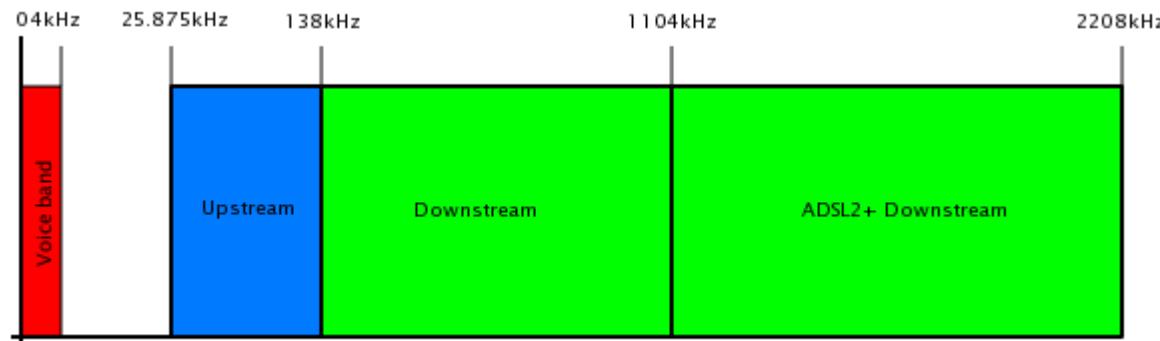
Modulation schemes send bits as signals

Baseband Transmission

- Signal that run from 0 up to a maximum frequency
- E.g., Telephone system: $0 \sim 4\text{kHz}$

Passband Transmission

- Signals that are shifted to occupy a higher range of frequencies



Example: ADSL

Modulation Types

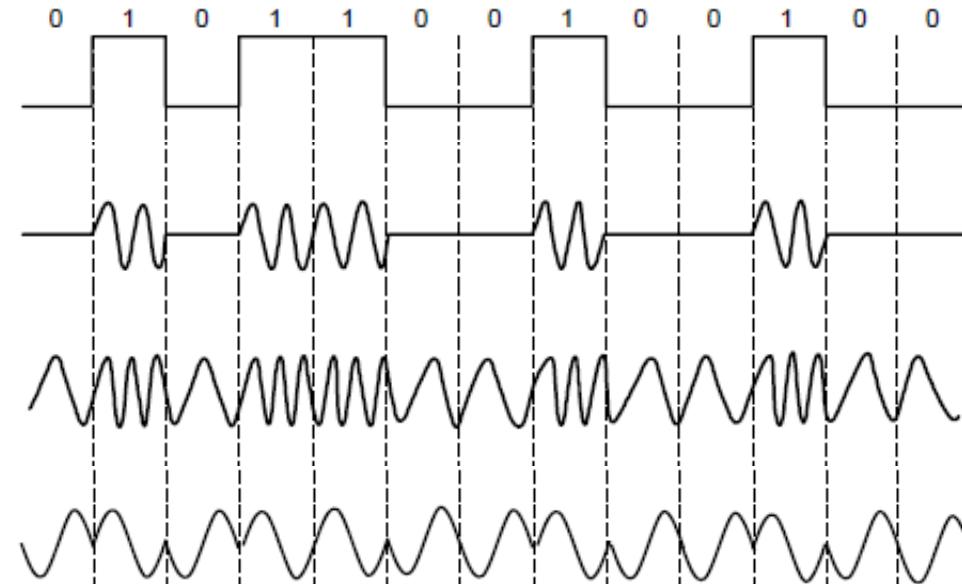
- Modulating the amplitude, frequency/phase of a carrier signal sends bits in a (non-zero) frequency range

NRZ signal of bits

Amplitude shift keying

Frequency shift keying

Phase shift keying



Data Communication using Signals (3)

- How would the receiver handle the signal to understand its meaning?
- How many different types of signals are there in each example?

Symbol Rate

- One symbol (signal element) can represent multiple bits (data elements)
- Symbol Rate (Baud Rate): number of signal changes per second

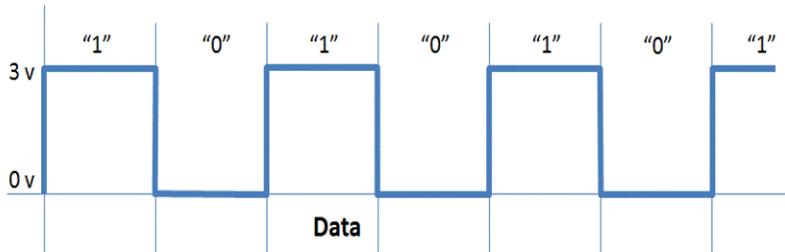


Figure 1. Data bits where logical “0” and “1” are represented by 0 volts and 3 volts respectively

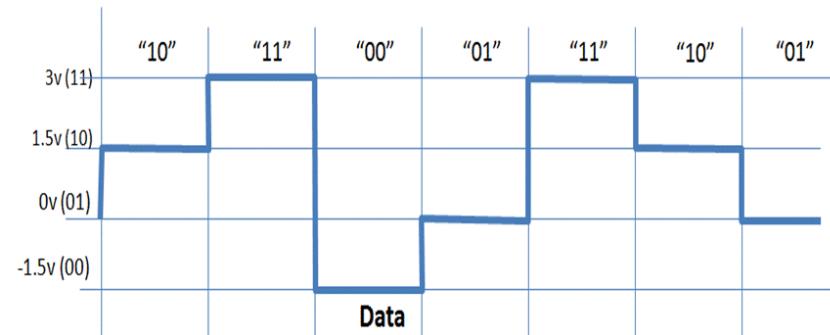


Figure 2. Four signaling levels per clock cycle can represent two data bits.

How much can we put on a link?



- Harry Nyquist
- Early theoretical work on determining fundamental limits for the bandwidth required for communication-heralded digital revolution

Maximum Data Rate of a Channel

- Nyquist's theorem relates the data rate to the bandwidth (B) and number of signal levels (V) (channel **without noise**):

$$\text{Max. data rate} = 2B \log_2 V \text{ bits/sec}$$

- Increase the bandwidth B can increase the data rate.
- If signal has V levels, each symbol can represent $\log_2 V$ bits.

What if there is noise?



- Claude Shannon - Father of Information theory.
- Contributed to information theory and cryptography.

Maximum Data Rate of a Channel

- Shannon's theorem relates the data rate to the bandwidth (B) and signal strength (S) relative to the **noise** (N):

$$\text{Max. data rate} = B \log_2(1 + S/N) \text{ bits/sec}$$

↑ ↑
How fast signal How many levels
can change can be seen

Example 1

Q: Given the signal-to-noise ratio (SNR) of 20 dB, and the bandwidth of 4kHz (telephone communications), what is the maximum data rate according to Shannon's theorem?

Example 2

Q: If a binary signal is sent over a 3-kHz channel whose signal-to-noise ratio is 20 dB, what is the maximum achievable data rate?

Link Terminology

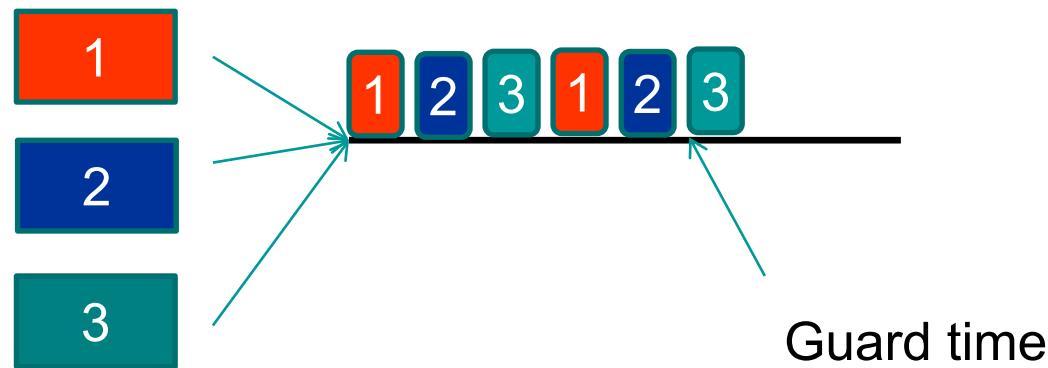
- Full-duplex link
 - Used for transmission in both directions at once
 - e.g., use different twisted pairs for each direction
- Half-duplex link
 - Both directions, but not at the same time
 - e.g., senders take turns on a wireless channel
- Simplex link
 - Only one fixed direction at all times; not common

Multiplexing

- When multiple sources want to access the medium
 - Time Division Multiplexing
 - Frequency Division Multiplexing
 - Statistical Multiplexing
 - Code Division Multiple Access

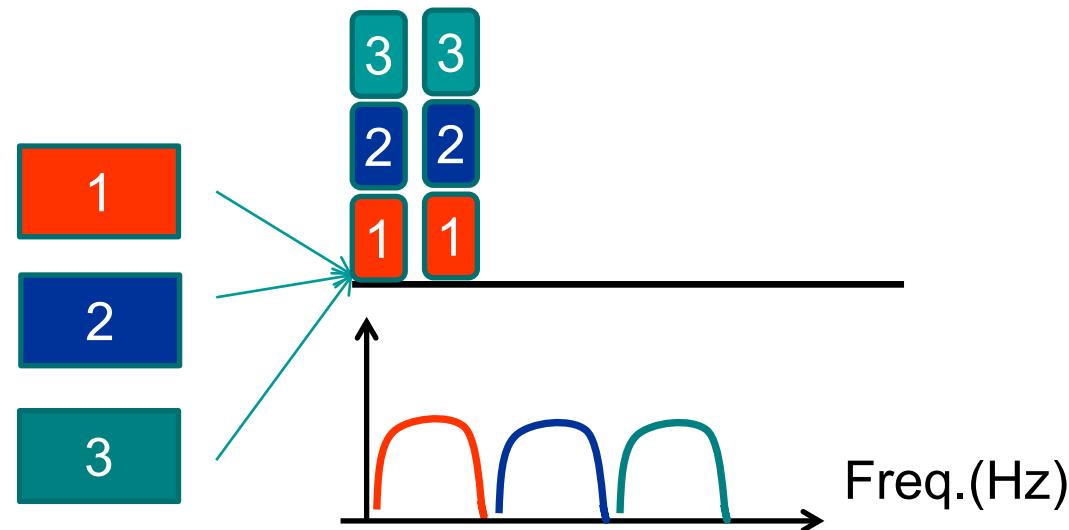
Time Division Multiplexing

- Users can send according to a fixed schedule
- Slotted access to the full speed of the media



Frequency Division Multiplexing

- Users can only use specific frequencies to send their data
- Continuous access with lower speed



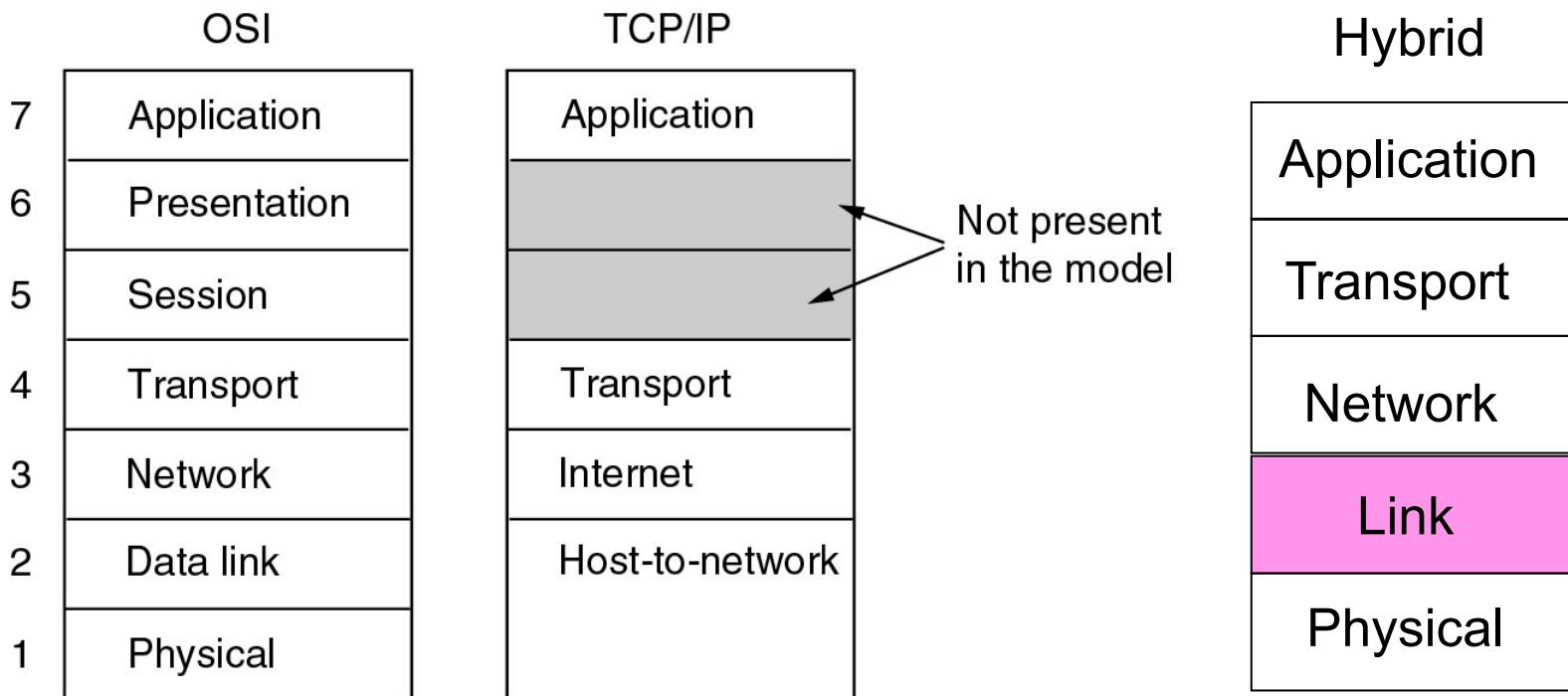
Week 3 – Data Link Layer

COMP90007 Internet Technologies

Lecturer: Ling Luo

Semester 2, 2020

The Data Link Layer in OSI and TCP/IP



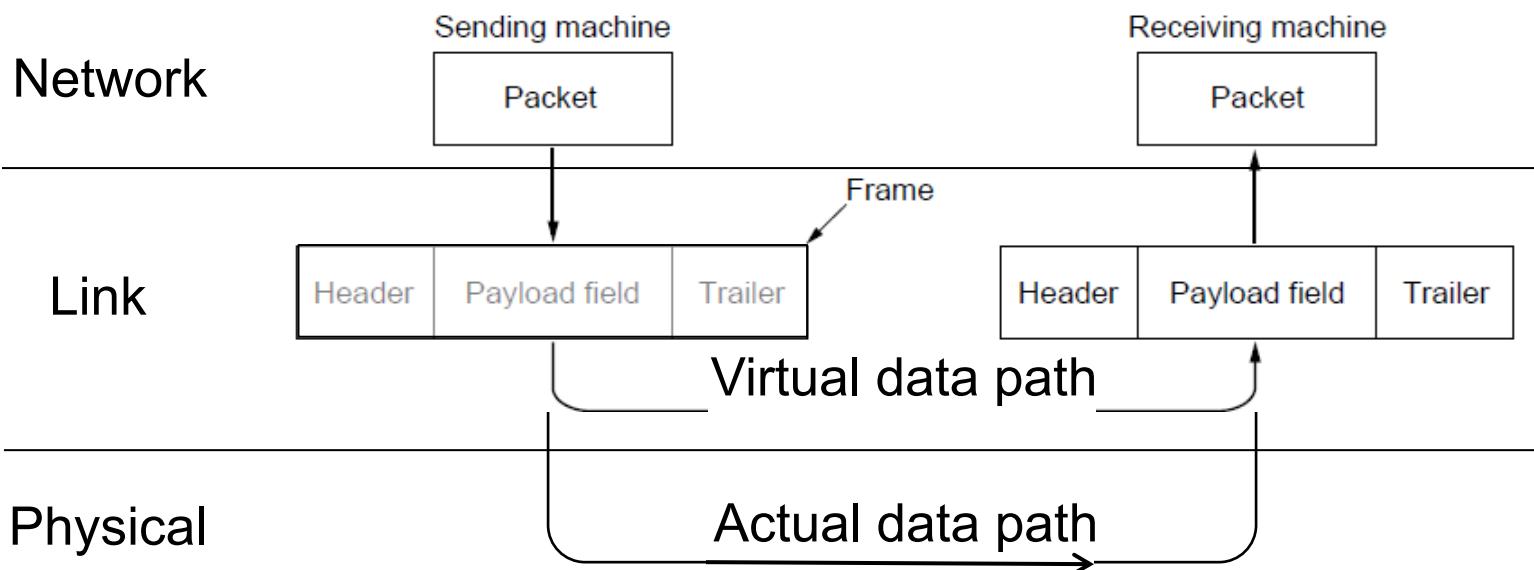
- **Reliable, efficient** communication of “**frames**” between two adjacent machines.
- Handles transmission errors and flow control.

Functions of the Data Link Layer

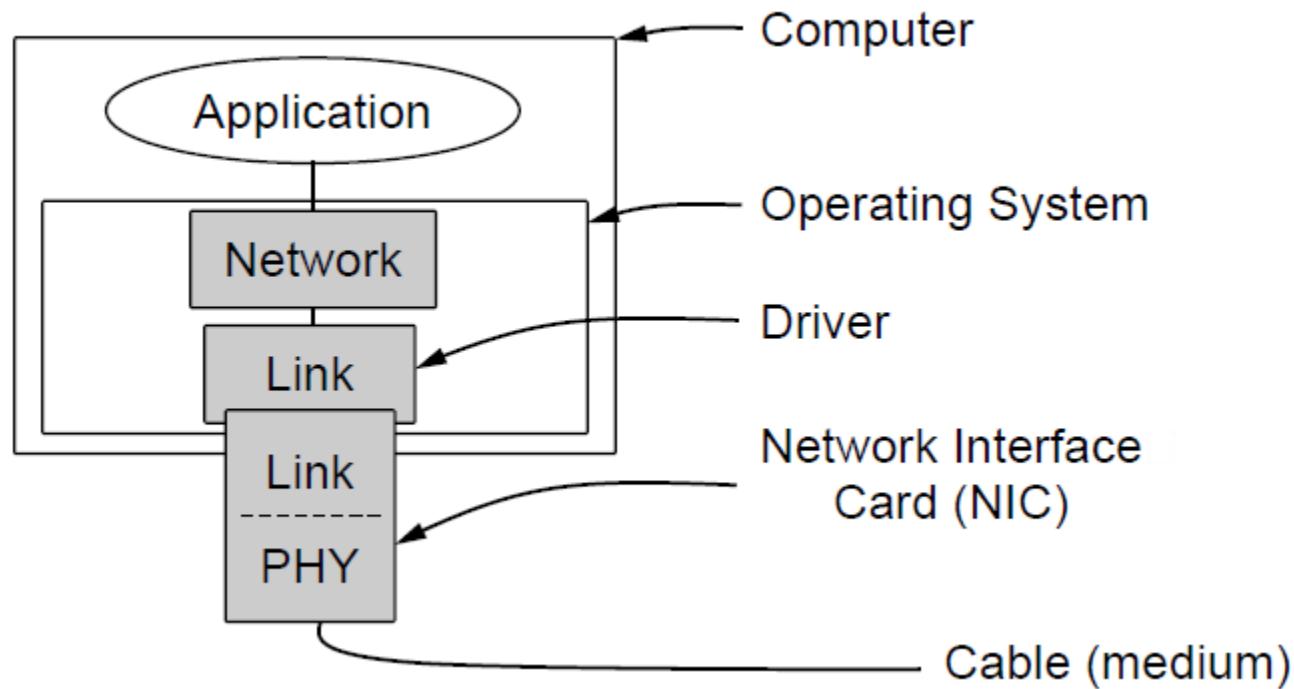
- Functions of the data link layer:
 1. Provide a well-defined service interface to network layer
 2. Handling transmission errors
 3. Data flow regulation
- Primary process:
 - Take **packets** from network layer, and encapsulate them into **frames** (containing a header, a payload, a trailer)

Relation Between Packets and Frames

Link layer accepts **packets** from the network layer, and encapsulates them into **frames** that it sends using the physical layer; reception is the opposite process



Typical Implementation



Type of Services

- **Connection-Oriented vs Connectionless:**
Whether a connection is setup before sending a message
- **Acknowledged vs Unacknowledged:**
Whether the receiver gives the sender an acknowledgement upon receiving the message

Services Provided to Network Layer

- Transferring data from the network layer on source host to the network layer on destination host
- Services provided:
 - Unacknowledged connectionless service
 - Acknowledged connectionless service
 - Acknowledged connection-oriented service

Unacknowledged Connectionless Service

- Source host transmits independent frames to recipient host with no acknowledgement
- No logical connection establishment or release
- No lost frame recovery mechanism (or left to higher levels)
- Applications:
 - Ethernet LANs
 - Real-time traffic, e.g. voice

Acknowledged Connectionless Service

- Source host transmits independent frames to recipient host with acknowledgement
- No logical connection establishment or release
- Each frame is individually acknowledged, and retransmitted if lost or errors
- Application: Wireless – IEEE 802.11 WiFi

Acknowledged Connection-Oriented Service

- Source host transmits independent frames to recipient host after connection establishment and with acknowledgement
- Connection established and released (communicate rate and details of message)
- Frames numbered, counted, acknowledged with logical order enforced
- Application: Unreliable links such as satellite channel or long-distance telephone circuit

Framing (1)

- Framing: breaks raw bit stream into discrete units
- Physical layer provides no guarantee a raw stream of bits is error free
- The primary purpose of framing is to provide some level of reliability over the unreliable physical layer
- Checksums can be computed and embedded at the source, then computed and compared at the destination
 $\text{checksum} = f(\text{payload})$

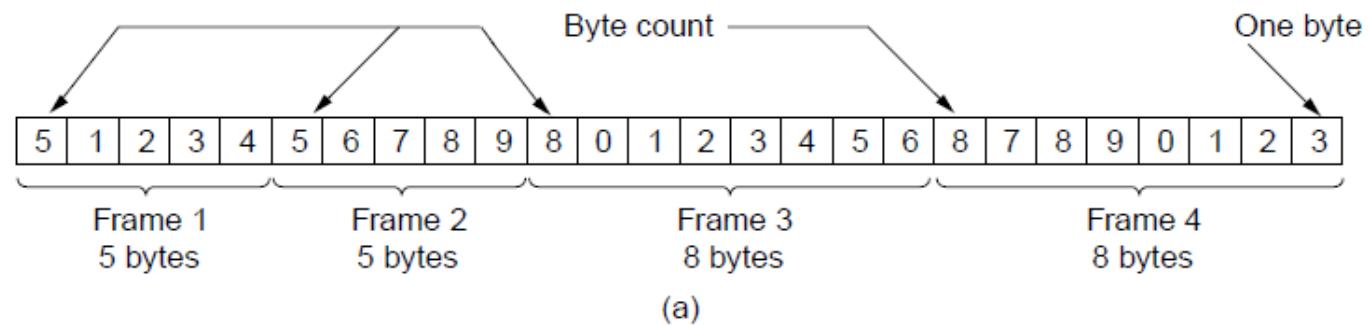
Framing (2)

- Methods:
 - Character (Byte) count
 - Flag bytes with byte stuffing
 - Start and end flags with bit stuffing
- Most data link protocols use a combination of character count and one other method

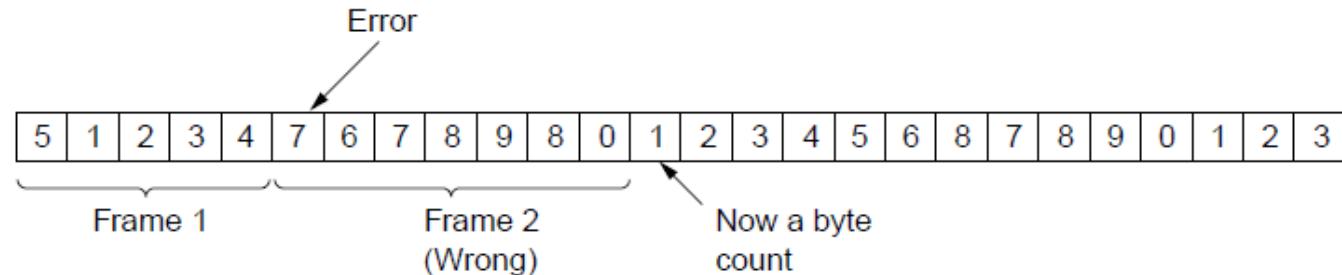
Character Counts

- Uses a field in the frame header to specify the number of characters in a frame

No error

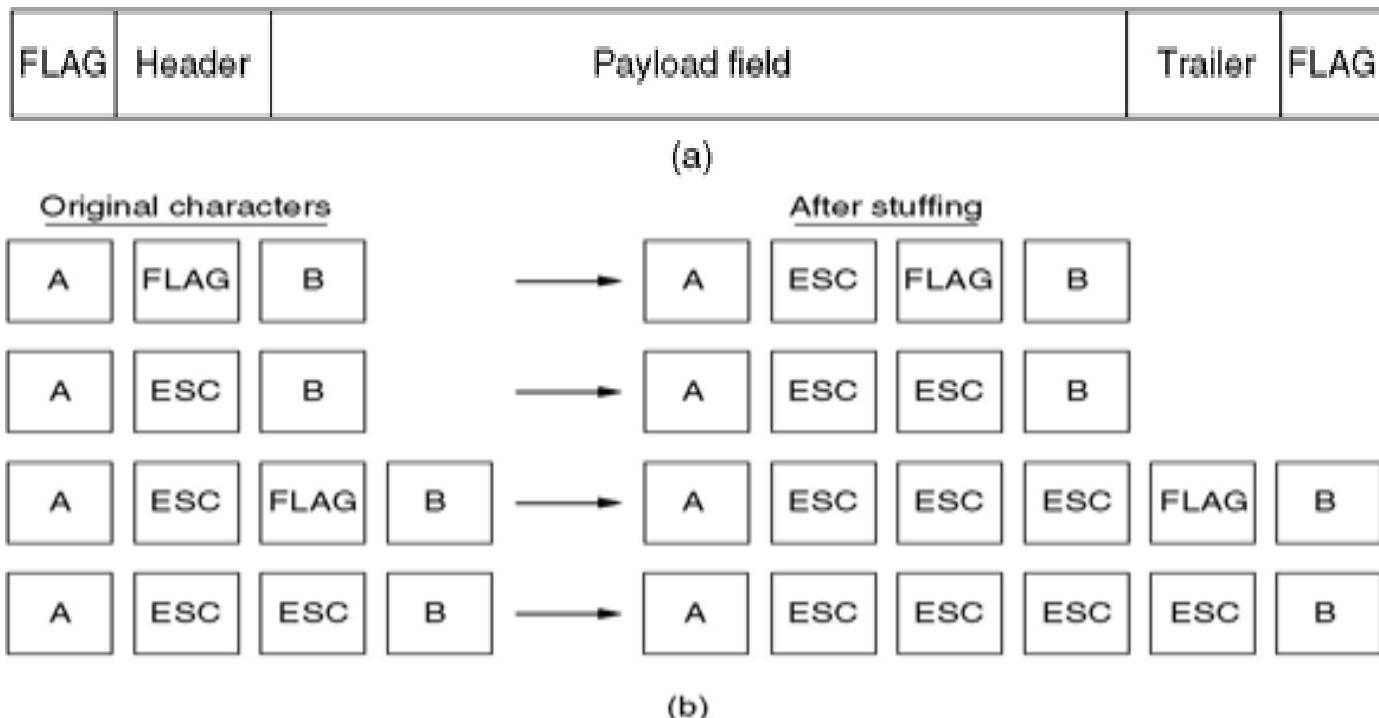


Case with
error



Flag Bytes with Byte Stuffing

- Each frame starts and ends with a special byte -“flag byte”



Start and End Flags with Bit Stuffing

- Frames contain an arbitrary number of bits
 - Each frame begins and ends with a special bit pattern
0111110

(a) 0110111111111111110010

The original data

(b) 01101111011111011111010010

Sent data

Stuffed bits

(c) 011011111111111111110010

Destuffing at receiver

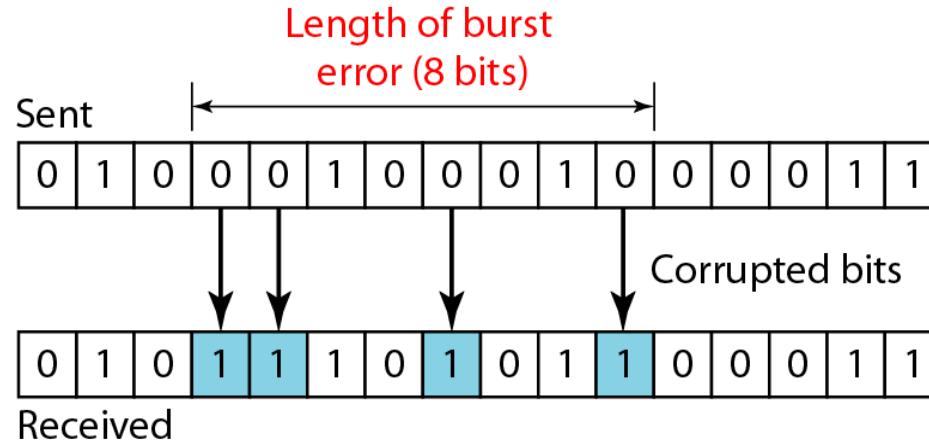
Insert 0 after five ones (11111)

Error Control

- Adding check bits to ensure that a garbled message by the physical layer is not considered as the original message by the receiver
- Error Control deals with
 - Detecting the error
 - Correcting the error
 - Re-transmitting lost frames

Error Detection and Correction (1)

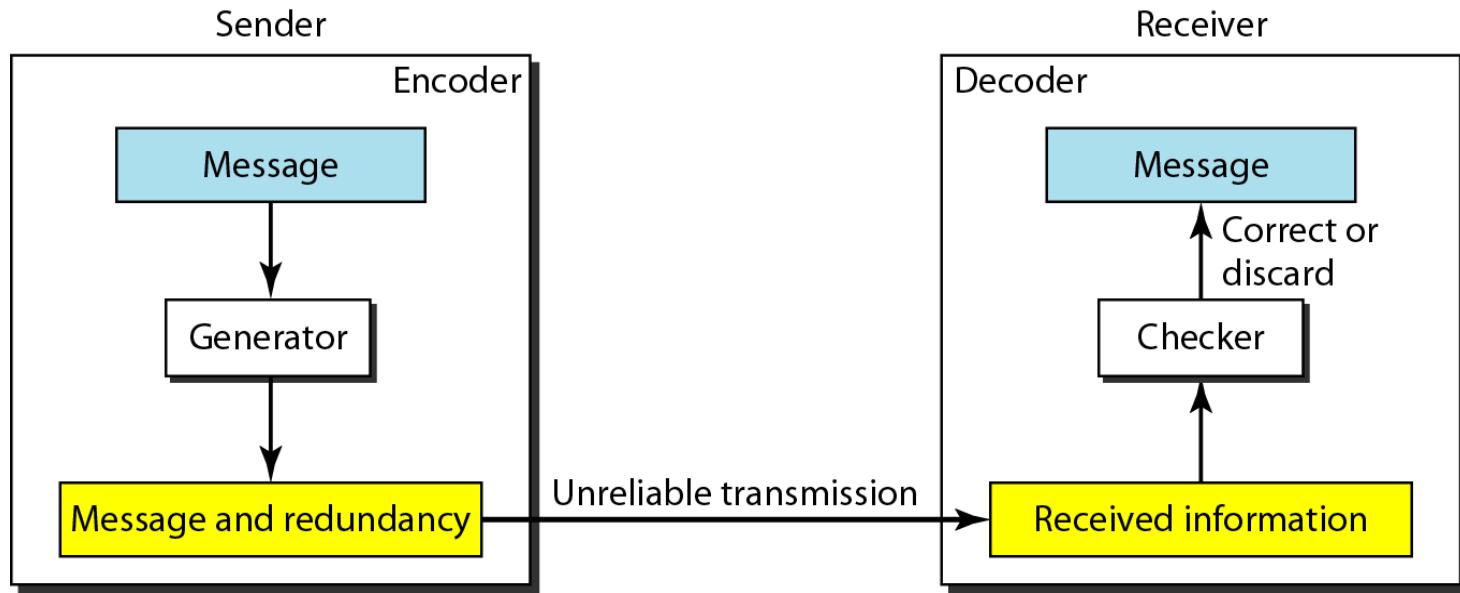
- Physical media may be subject to errors
- Errors may occur randomly or in bursts
 - Single-bit error
 - Burst error: two or more bits have changed



Bursts of errors are easier to detect but harder to resolve

Error Detection and Correction (2)

- Resolution needs to occur before handing data to network layer



- Key issues
 - Fast** mechanism and **low computational overhead**
 - Minimum amount of extra bits** send with the data
 - Detection of **different kinds of error**

Example

- Repeat the bits, if a copy is different than the other, there is an error
 - 01101 -> 000 111 111 000 111
- What is the overhead?
- How many errors can receiver detect?
- How many errors can receiver correct?
- What is the minimum number of errors that can fail the algorithm?

Error Bounds – Hamming distance

Error Bounds

Q: Why can a code with distance $2d+1$ correct up to d errors?

- Errors are corrected by **mapping** a received invalid codeword to the nearest valid codeword, i.e., the one that can be reached with the fewest bit flips
- If there are more than d bit flips, then the received codeword may be closer to another valid codeword than the codeword that was sent

Example: Sending 0000000000 with 2 flips might give 1100000000 which is closest to 0000000000, correcting the error.

But with 3 flips 1110000000 might be received, which is closest to 1111100000, which is still an error

Week 3 – Data Link Layer

COMP90007 Internet Technologies

Lecturer: Ling Luo

Semester 2, 2020

Hamming Code

- $n=2^k-k-1$ (n : number of data, k : check bits)

Example: Data: 0101 -> requires 3 check bits

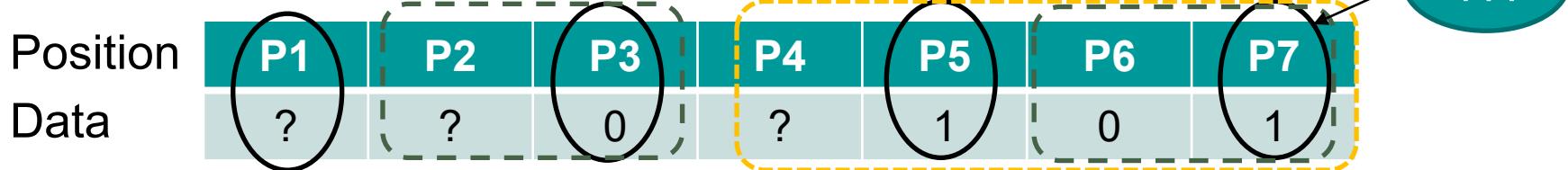
$$\downarrow \quad \searrow$$
$$4 = (2^3) - 3 - 1$$

- Put check bits in positions p that are power of 2, starting with position 1
- Check bit in position p is parity of positions with a p term in their value

Example

Put check bits in positions p that are power of 2, starting with position 1

- Data: 0101 → requires 3 check bits



1. Calculate the parity bits for P1, P2, P4 (rule: even parity)



$$P1 + P3 + P5 + P7 = ? + 0 + 1 + 1 \text{ (even)} \rightarrow P1 = 0$$



$$P2 + P3 + P6 + P7 = ? + 0 + 0 + 1 \text{ (odd)} \rightarrow P2 = 1$$



$$P4 + P5 + P6 + P7 = ? + 1 + 0 + 1 \text{ (even)} \rightarrow P4 = 0$$

Data sent: 0100101

error

error

Example 1: At the receiver: 0100100

$$\begin{cases} P1 + P3 + P5 + P7 = 0 + 0 + 1 + 0 = 1 \times \\ P2 + P3 + P6 + P7 = 1 + 0 + 0 + 0 = 1 \times \\ P4 + P5 + P6 + P7 = 0 + 1 + 0 + 0 = 1 \times \end{cases}$$

$$\text{Error bit} = P1 + P2 + P4 = P7$$

Example 2: At the receiver: 0000101

$$\begin{cases} P1 + P3 + P5 + P7 = 0 + 0 + 1 + 1 = 0 \\ P2 + P3 + P6 + P7 = 0 + 0 + 0 + 1 = 1 \times \\ P4 + P5 + P6 + P7 = 0 + 1 + 0 + 1 = 0 \end{cases}$$

$$\text{Error bit} = P2$$

Error Correcting Codes Key Points

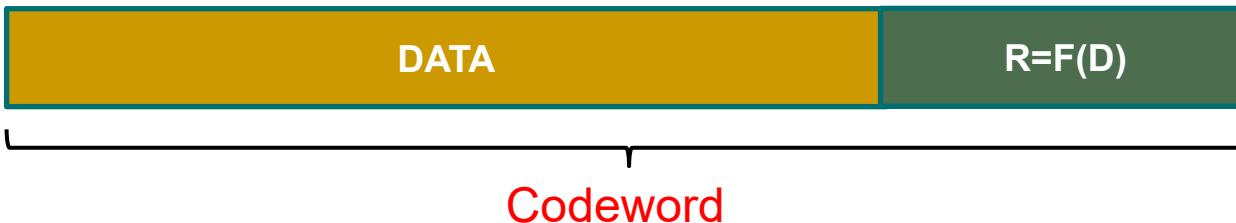
- More efficient in noisy transmission media e.g., wireless
- Challenge is that the error can be in the check bits
- Require assumption on a specific number of errors occurring in transmission

Error Detecting Codes

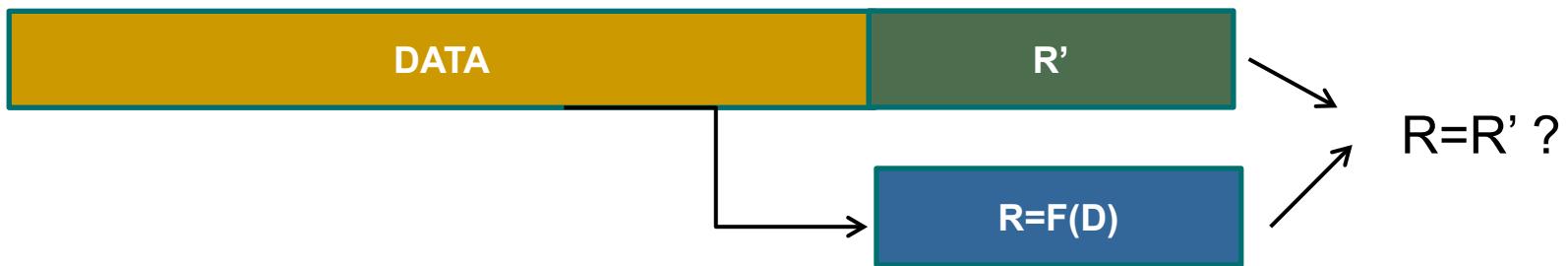
- More efficient in some transmission media – e.g. quality copper, where low error rates occur
- **Parity** (1 bit): (Hamming distance=2)
- **Checksum** (16 bits): (Hamming distance=2)
- **Cyclical Redundancy Check** (CRC) (Standard 32-bit CRC: Hamming distance=4)

How it works?

- Sender: calculates R check bits using a function of data bits:



- Receiver: receives the codeword and calculates the same function on the data and match the results with received check bits:



Parity Bit

Given data 10001110, count the number of 1s

Sender: Add parity bit → 100011100 (for even parity)

100011101 (for odd parity)

Receiver: Check the transferred data for errors on arrival.

Hamming distance is 2 for Parity Bit...

$2 - 1 = \underline{1 \text{ error bit can be detected}}$ and

$(2 - 1) / 2 = \frac{1}{2}$ not even 1 bit error can be corrected

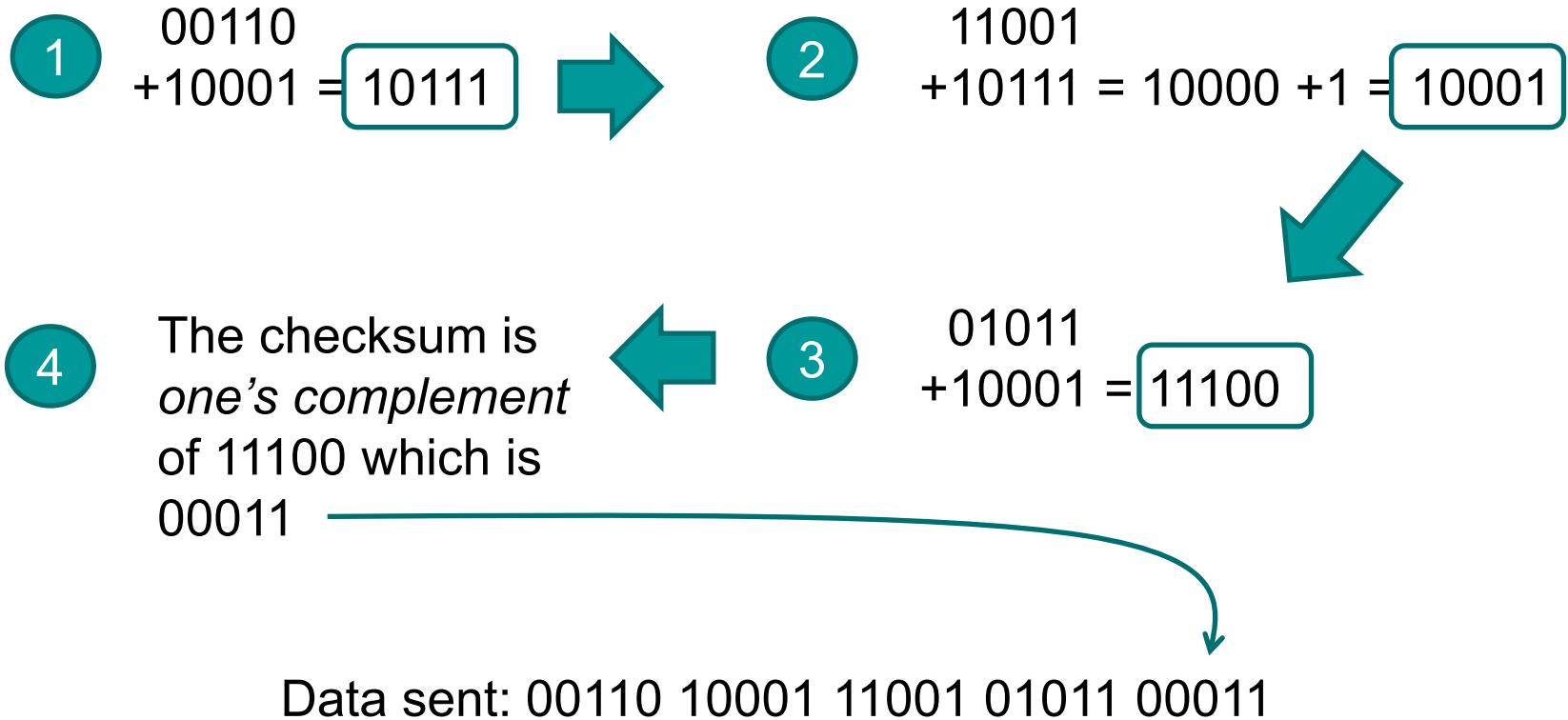
Internet Checksum

- There are different variations of checksum
- Internet Checksum (16-bit word):
Sum modulo 2^{16} and add any overflow of high order bits back into low-order bits

Example of Checksum

Calculate checksum (5-bit word) for data

00110 10001 11001 01011



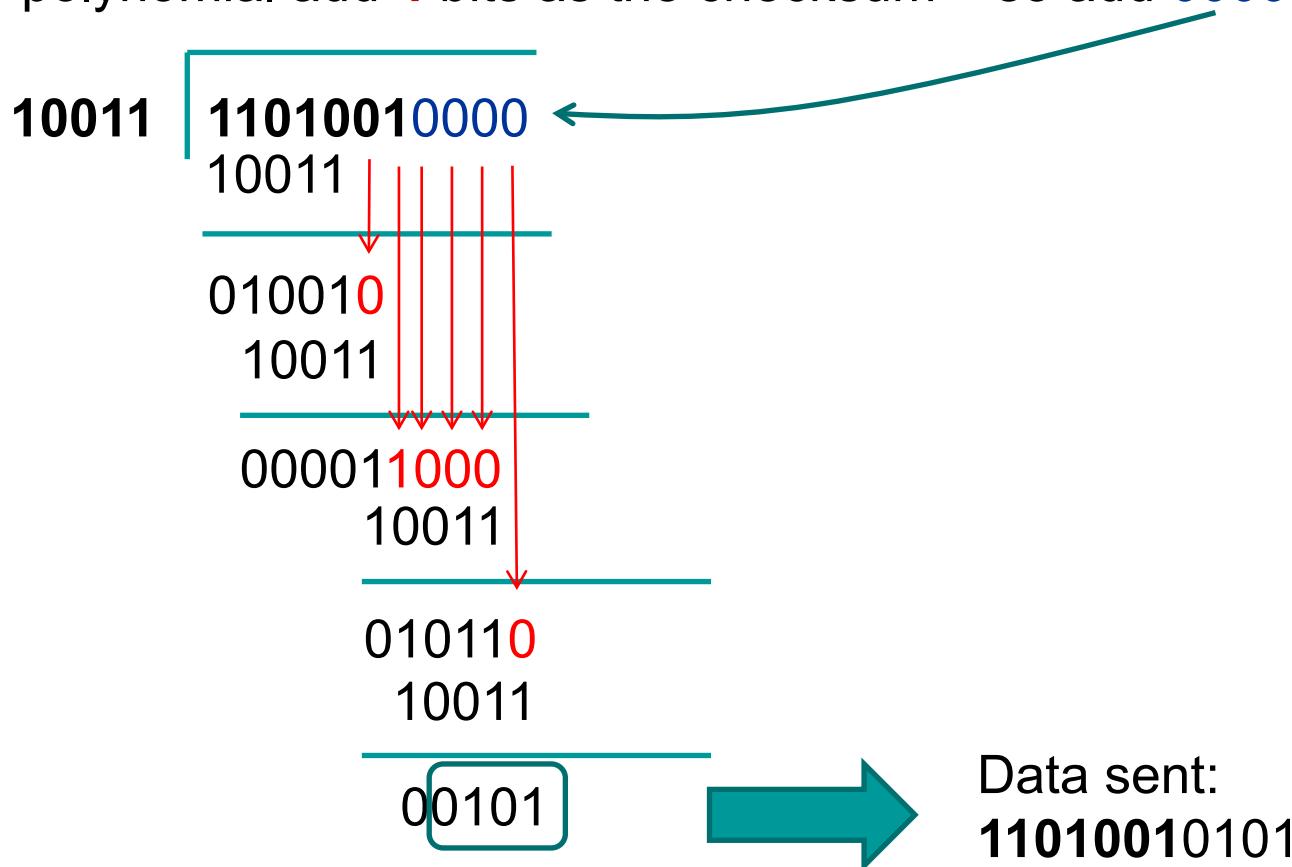
Cyclic Redundancy Check

- Based on a generator polynomial $G(x)$
 - e.g. $G(x) = x^4 + x + 1$ (10011)
 - Let r be the degree of $G(x)$ ($r=4$). **Append r zero bits to the low-order end of the frame** so it now contains $m + r$ bits and corresponds to the polynomial $x^r M(x)$.
 - **Divide the bit string corresponding to $G(x)$** into the bit string corresponding to $x^r M(x)$, using modulo 2 division.
 - **Subtract the remainder** (which is always r or fewer bits) from the bit string corresponding to $x^r M(x)$ using modulo 2 subtraction.
 - The result is the checksummed frame to be transmitted. Call its polynomial $T(x)$.

Example

Data: **1101001** and $G(x) = x^4+x+1$ (**10011**)

5 bits polynomial add **4** bits as the checksum – so add **0000**



Data sent:
11010010101

COMP90007 Internet Technologies

Week 3 Workshop

Semester 2, 2020

Question 1 (Layers)

- Identify 2 ways in which the OSI reference model and the TCP/IP reference model are the same.
- Identify 2 ways in which these models differ.
(NB: You can use the textbook to solve this question)

Question 2 (Delay and bandwidth)

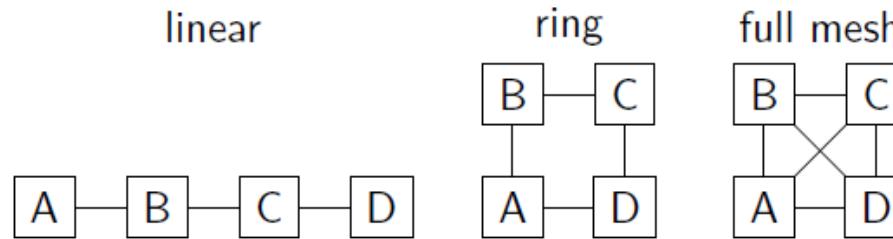
- Calculate the end-to-end transit time for a packet for
 - GEO (*Geostationary orbit*) (altitude: 35,800 km),
 - MEO (*Medium Earth orbit*) (altitude: 18,000 km) and
 - LEO (*Low Earth orbit*) (altitude: 750 km) satellites.

Question 3 (Delay and bandwidth)

- An image is 1600×1200 pixels with 3 bytes/pixel.
Assume the image is uncompressed.
 - How long does it take to transmit it over a 56-kbps modem channel, assuming zero propagation delay over the channel?
 - Over a 1-Mbps cable modem? Over a 10-Mbps Ethernet?
 - Over 100-Mbps Ethernet? Over gigabit Ethernet?

Question 4 (Topology)

- Consider the following 3 network topologies for connecting N nodes.
In the general case of an N node network:



- (a) How many links are there in each network?
- (b) What is the maximum delay between any pair of nodes, assuming each link has a delay of 10ms, and the shortest path is used between nodes?
- (c) What is the minimum number of links that need to be cut in order to isolate one or more nodes?
- (d) Which topology would you use to connect military command centres?

Question 5 (Topology)

- Is an oil pipe a simplex system, a half-duplex system, a full duplex system or none of the above? Under which conditions?

Question 6 (Topology)

- List two solutions that one can use for sharing a link between multiple senders and explain these solutions briefly.

COMP90007 Internet Technologies

Week 3 Workshop

Semester 2, 2020

Suggested solutions

Question 1 (Layers)

- Identify 2 ways in which the OSI reference model and the TCP/IP reference model are the same.
- Identify 2 ways in which these models differ.
(NB: You can use the textbook to solve this question)

Similarities:

- stacking of layered protocols
- similar functionality in each of the layers
- layers above transport layer relate to applications

Differences:

- TCP/IP does not distinguish between services, interfaces and protocols
- TCP/IP does not clearly separate physical and data link functions

Question 2 (Delay and bandwidth)

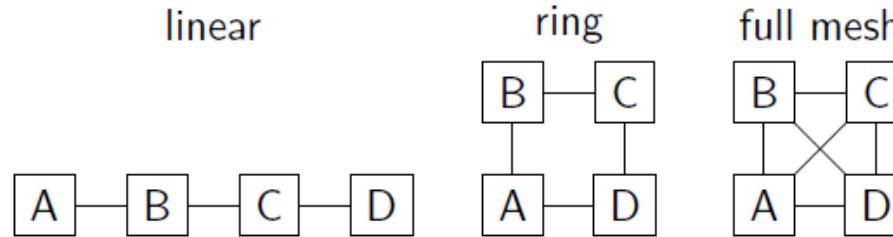
- Calculate the end-to-end transit time for a packet for
 - GEO (*Geostationary orbit*) (altitude: 35,800 km),
 - MEO (*Medium Earth orbit*) (altitude: 18,000 km) and
 - LEO (*Low Earth orbit*) (altitude: 750 km) satellites.
- *Transit time = 2 × distance / speed of light*, where $c = 3.0 \times 10^8$ m/s
- GEO: 239 ms
- MEO: 120 ms
- LEO: 5 ms

Question 3 (Delay and bandwidth)

- An image is 1600×1200 pixels with 3 bytes/pixel.
Assume the image is uncompressed.
 - How long does it take to transmit it over a 56-kbps modem channel, assuming zero propagation delay over the channel?
 - Over a 1-Mbps cable modem? Over a 10-Mbps Ethernet?
 - Over 100-Mbps Ethernet? Over gigabit Ethernet?
- Image size = $1600 \times 1200 \times 3 \times 8 = 46.08 \times 10^6$ bits
- | | |
|----------------------|---------|
| • 56 kbps modem: | 823 s |
| • 1 Mbps modem: | 46.1 s |
| • 10 Mbps Ethernet: | 4.61 s |
| • 100 Mbps Ethernet: | 0.46 s |
| • 1 Gbps Ethernet: | 0.046 s |

Question 4 (Topology)

- Consider the following 3 network topologies for connecting N nodes.
In the general case of an N node network:



- (a) How many links are there in each network?
Linear: $N - 1$ links Ring: N links Full mesh: $N(N - 1)/2$ links
- (b) What is the maximum delay between any pair of nodes, assuming each link has a delay of 10ms, and the shortest path is used between nodes?
Linear: $10(N - 1)$ ms Ring: $10*N/2$ ms Full mesh: 10 ms
- (c) What is the minimum number of links that need to be cut in order to isolate one or more nodes?
Linear: 1 link Ring: 2 links Full mesh: $N - 1$ links
- (d) Which topology would you use to connect military command centres?
Full mesh – cost not important, but reliability is essential

Question 5 (Topology)

- Is an oil pipe a simplex system, a half-duplex system, a full duplex system or none of the above? Under which conditions?
 - Oil can flow in either direction, but not both ways at once, therefore it **cannot** be *full duplex*.
 - Depending on the situation, at an oil refinery, for example, an oil pipe is *simplex*, as the oil only flows in one direction.
 - Theoretically oil can flow both ways, therefore it can be consider *half duplex*, similar to a single railroad track.

Question 6 (Topology)

- List two solutions that one can use for sharing a link between multiple senders and explain these solutions briefly.

Ans: Time division multiplexing and frequency division multiplexing. There are others but these are the key ones we saw in class in detail. The explanations are available from slides 42,43,44 of week 2.

Week 3 – Data Link Layer

COMP90007 Internet Technologies

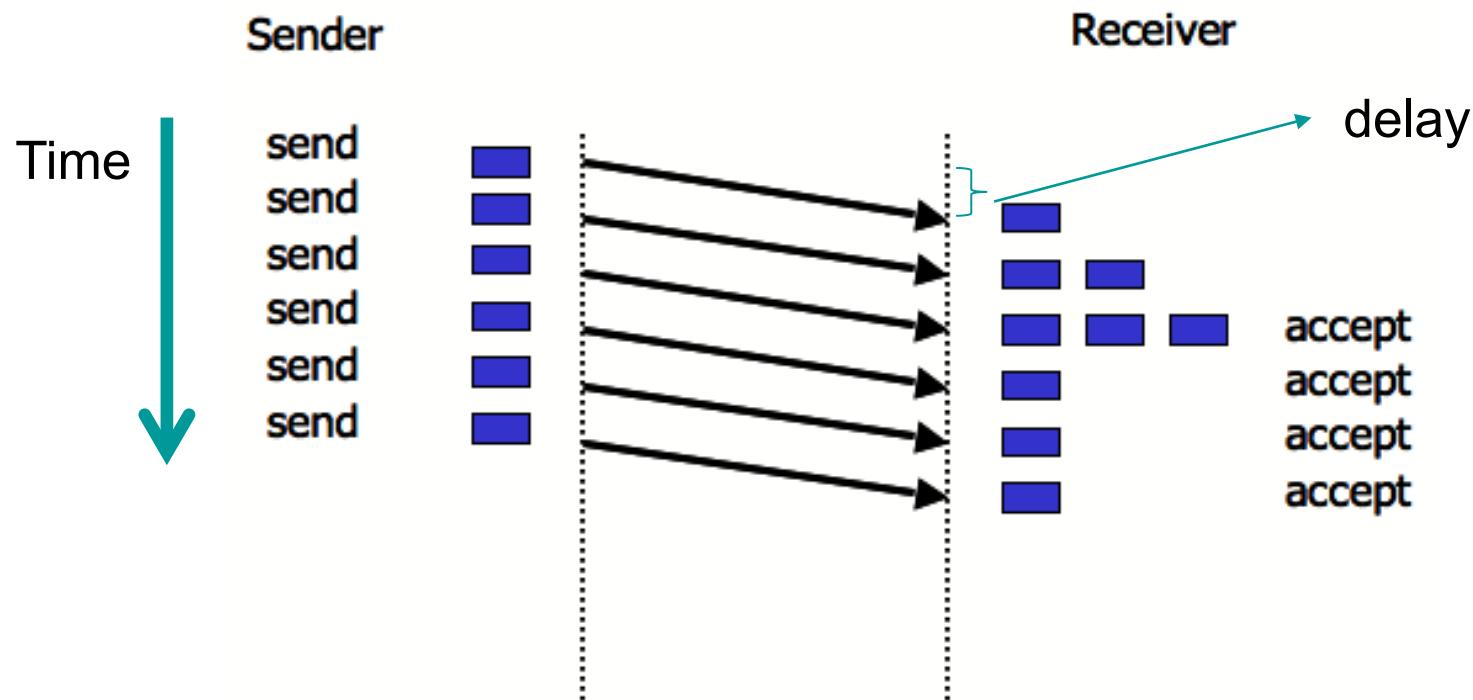
Lecturer: Ling Luo

Semester 2, 2020

Flow Control

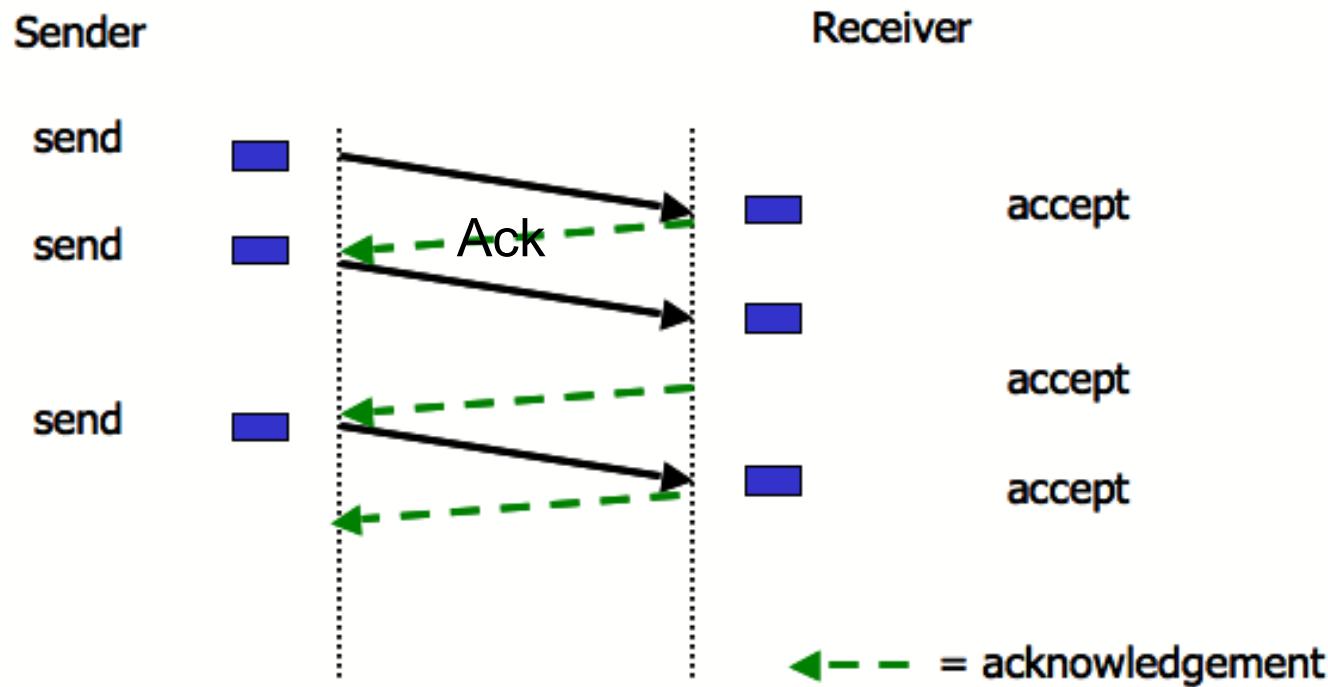
- So far we have discussed how to send single messages and now we will look at a series of messages
- The fast senders vs slow receivers problem requires a solution
- Principles to control when sender can send next frame
 - Feedback based flow control (usually used in Data Link layer)
 - Rate based flow control

A Very Simple Protocol



Acknowledged Transmission

Case: **fast sender / slow receiver**, the receiver's buffer space constrained

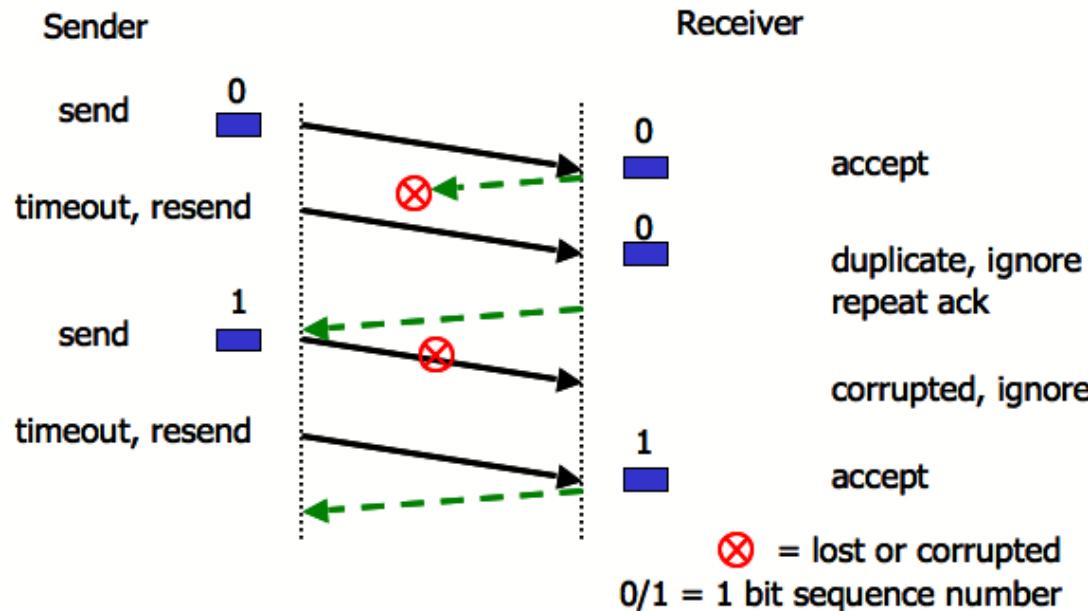


Noisy Channel Protocol

- Case: frames can be lost either entirely or partially
- Requires **distinction between frames already sent/received and those being retransmitted**
- Requires **timeout** function to determine arrival or non-arrival of complete frames

Stop and Wait Protocol

- ARQ (Automatic Repeat reQuest)
 - Ack and Timeout



Link Utilisation in Stop and Wait Protocols

Link Utilisation (U) measures efficiency in communication.

T_f = Transmission delay, time needed to transmit a frame of length L;

T_p = Propagation delay;

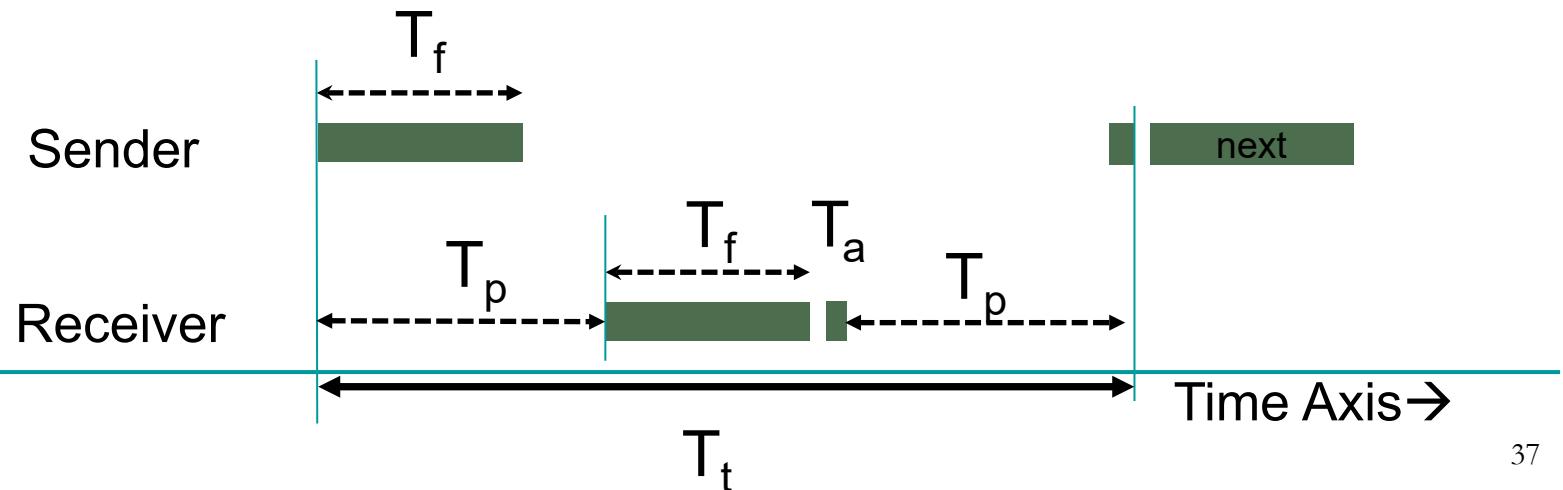
T_a = Time for transmitting an Ack, and we can assume $T_a = 0$.

$$T_t = T_f + 2T_p$$

$$U = (\text{Time of transmitting a frame}) / (\text{Total time for the transfer}) = T_f / T_t$$

Given bit rate B and $T_f = L/B$, we have

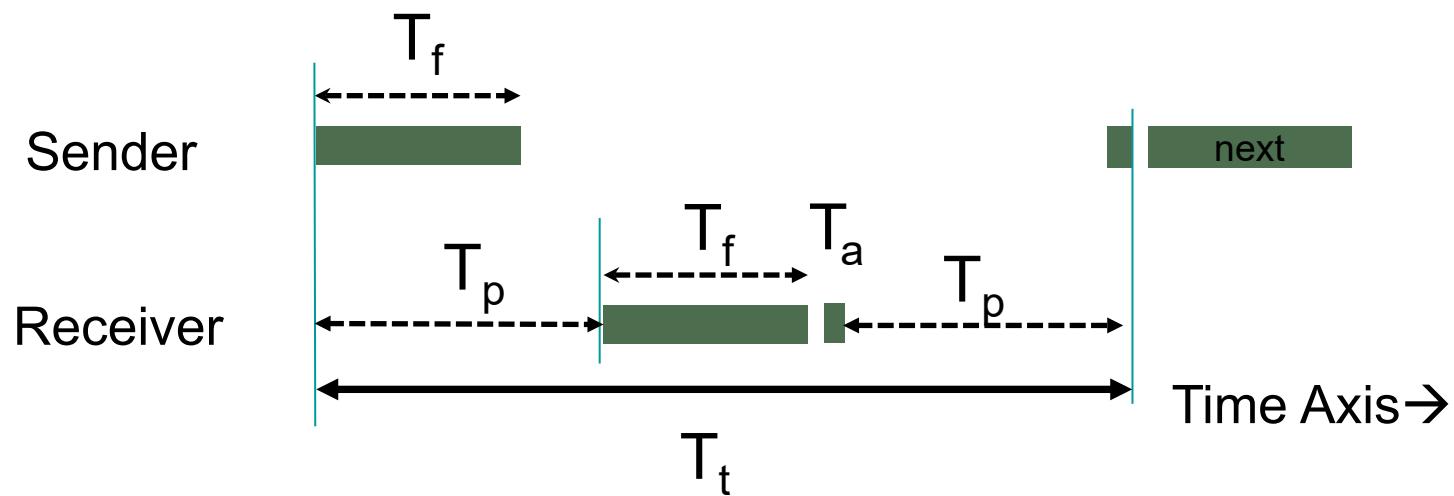
$$U = T_f / (T_f + 2T_p) = (L/B) / (L/B + 2T_p) = L / (L + 2T_p B).$$



Link Utilisation in Stop and Wait Protocols

For a link with $B=1$ Mbps, $T_p=50$ ms and frame size 10Kb, what is the link utilisation?

$$\begin{aligned} U &= L / (L + 2T_p B) \\ &= 10000 / (10000 + 2 * 0.05 * 10^6) = 1/11 \end{aligned}$$



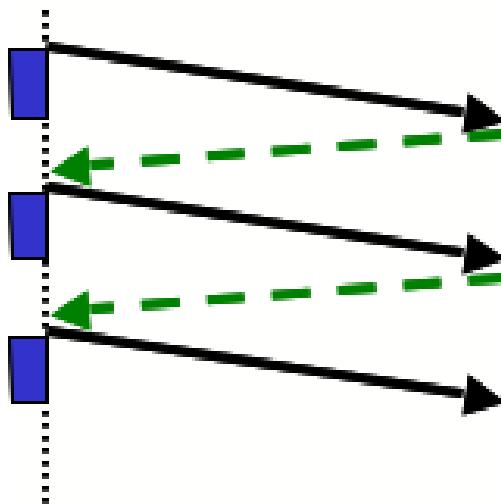
Sliding Window Protocols

- Sending window: Sender maintains a set of sequence numbers corresponding to frames allowed to send
- Receiving window: Receiver maintains a set of sequence numbers corresponding to frames allowed to accept
- What is the window size of Stop and Wait protocol?

Sliding Window Protocols

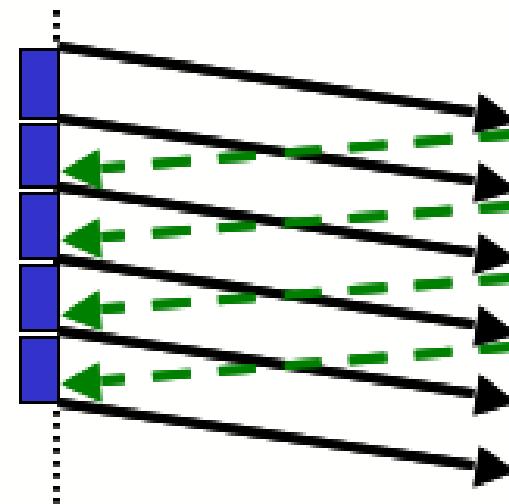
- Link Utilisation:

Stop and Wait



50% utilisation

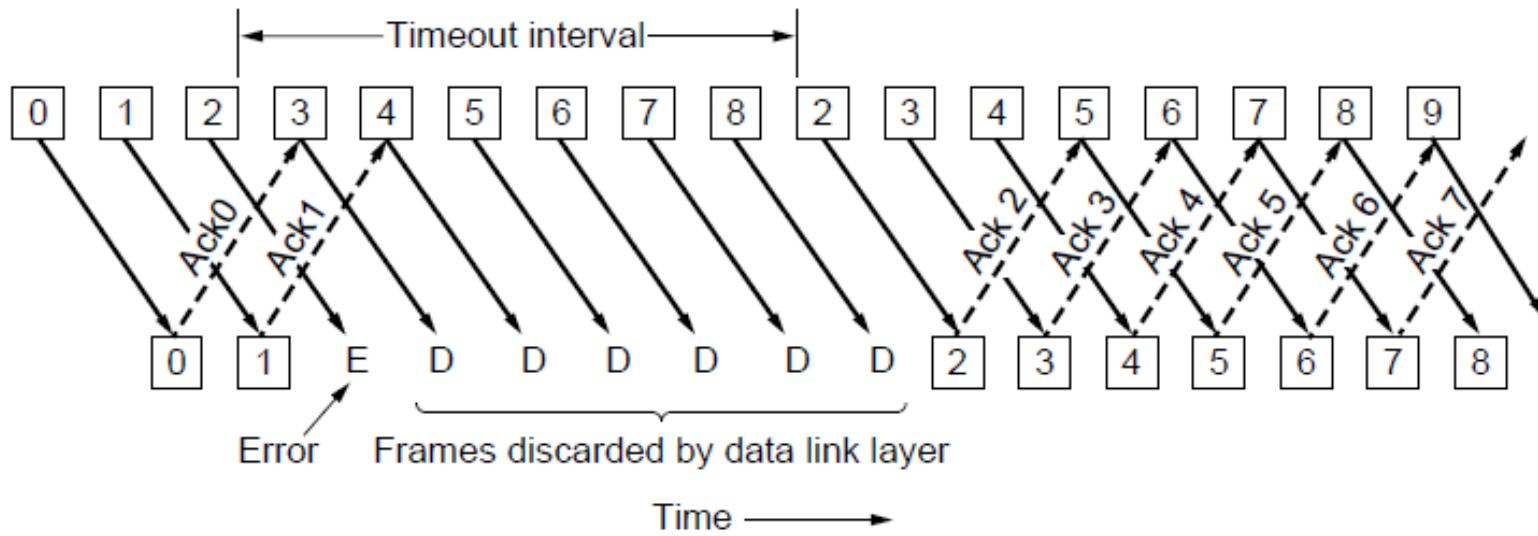
Sliding Window



100% utilisation

Go-Back-N

- Senders don't need to wait for acknowledgement for each frame before sending next frame

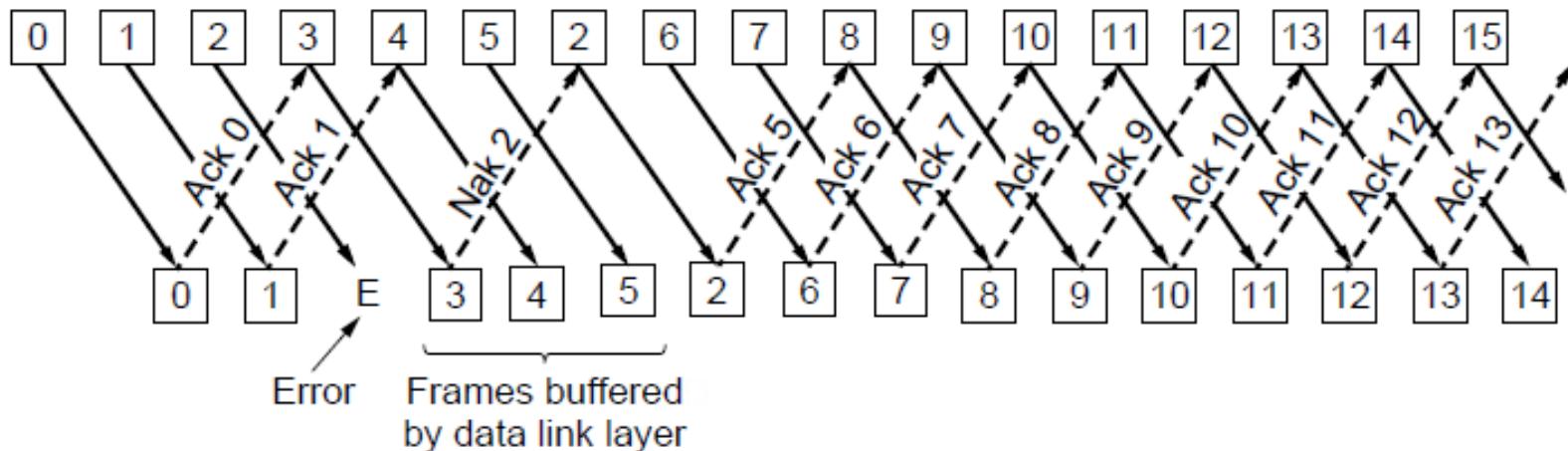


Receiver window size = 1, Sender window size is N

- Long transmission times need to be taken into account when programming timeouts e.g., low bandwidth or long distance

Selective Repeat

- Receiver accepts frames anywhere in receive window
 - NAK (negative ack) causes sender retransmission of a missing frame before a timeout resends window
 - Cumulative ack indicates highest in-order frame



Go-Back-N vs Selective Repeat

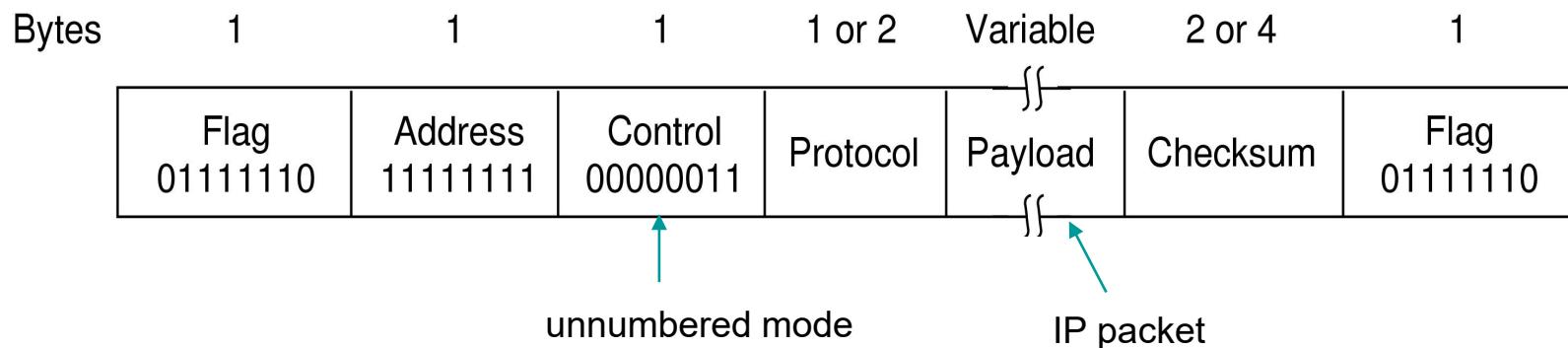
- Go-Back-N: receiver discards all subsequent frames from error point, sending no acknowledgement, until the next frame in sequence
- Selective Repeat: receiver buffers good frames after an error point, and relies on sender to resend oldest unacknowledged frames
- Trade-off between efficient use of bandwidth and data link layer buffer space

Examples of Data Link Protocols

- PPP (Point-to-Point Protocol)
- Packet over SONET
- PPP over ADSL

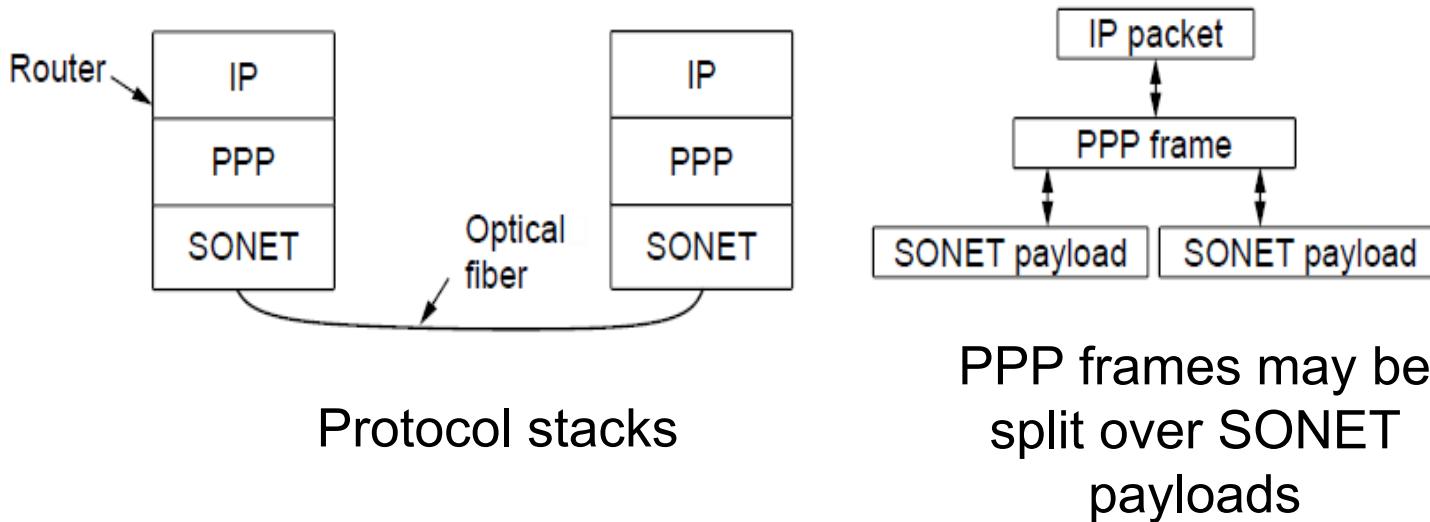
PPP

- PPP (Point-to-Point Protocol) is a general method for delivering packets across links
 - Framing uses a flag (0x7E) and byte stuffing
 - Default is unnumbered mode: connectionless unacknowledged service
 - Errors are detected with a checksum



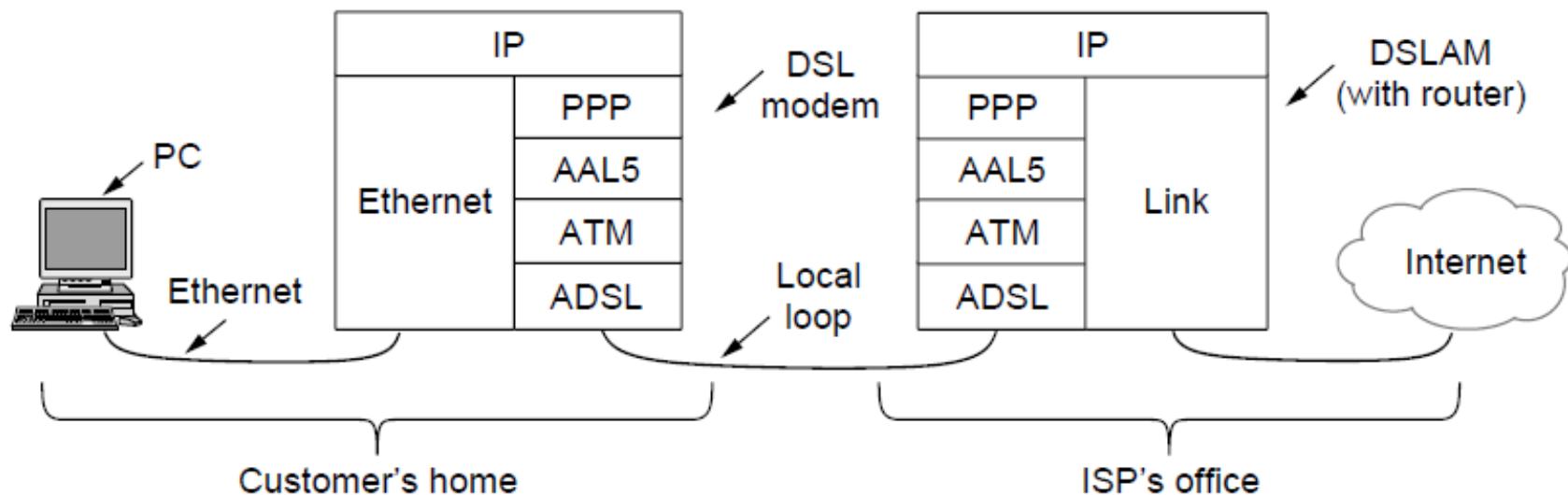
Packet over SONET

- Packet over SONET: carry IP packets over SONET optical fibre links
- Uses PPP (Point-to-Point Protocol) for framing



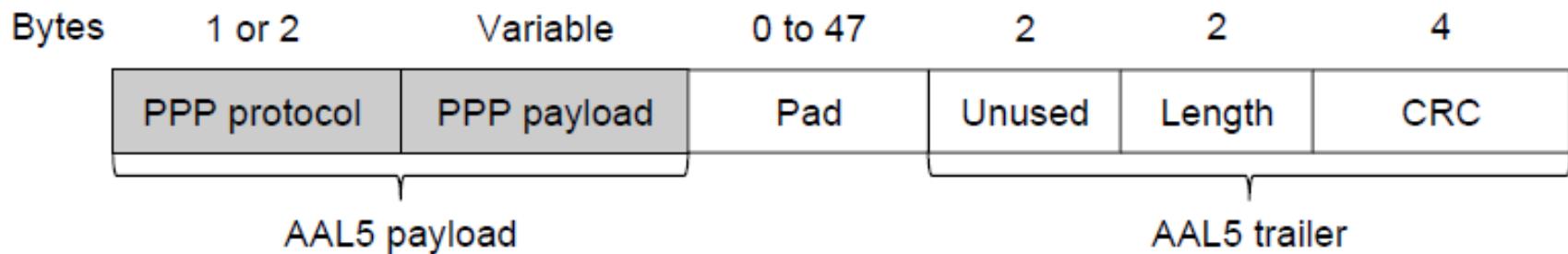
ADSL

- Widely used for broadband Internet over local loops
 - ADSL runs from modem (customer) to DSLAM (ISP)
 - IP packets are sent over PPP and AAL5/ATM (over)



ADSL

- PPP data is sent in ATM cells over ADSL
 - ATM is a link layer protocol that uses short, fixed-size cells (53 bytes); each cell has a virtual circuit identifier
 - 1) PPP frame is converted to an AAL5 frame (PPPoA)
 - 2) AAL5 frame is converted to ATM cells



Structure of AAL5 frame

which will be divided into 48-byte pieces, each of which goes into one ATM cell with 5-byte header

MAC Sub-Layer

COMP90007 Internet Technologies

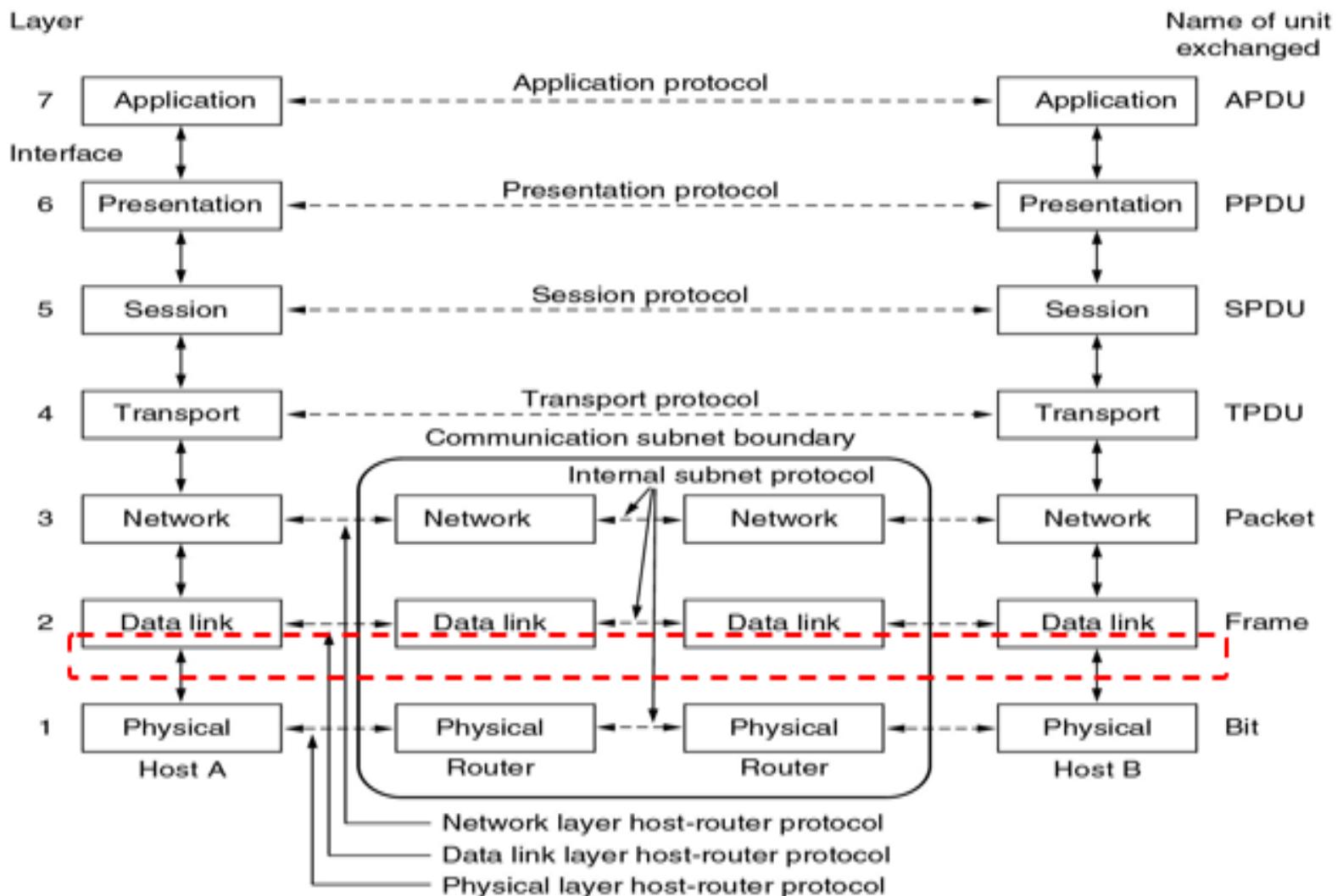
Lecturer: Ling Luo

Semester 2, 2020

Introduction

- On **point to point networks**, there are only singular sender and receiver pairs, eliminating transmission contention
- On **broadcast networks**, determining right to transmit is a complex problem
- **Medium Access Control (MAC)** sub-layer is used to assist in resolving transmission conflicts

MAC Sub-layer



Types of Channel Allocation Mechanisms

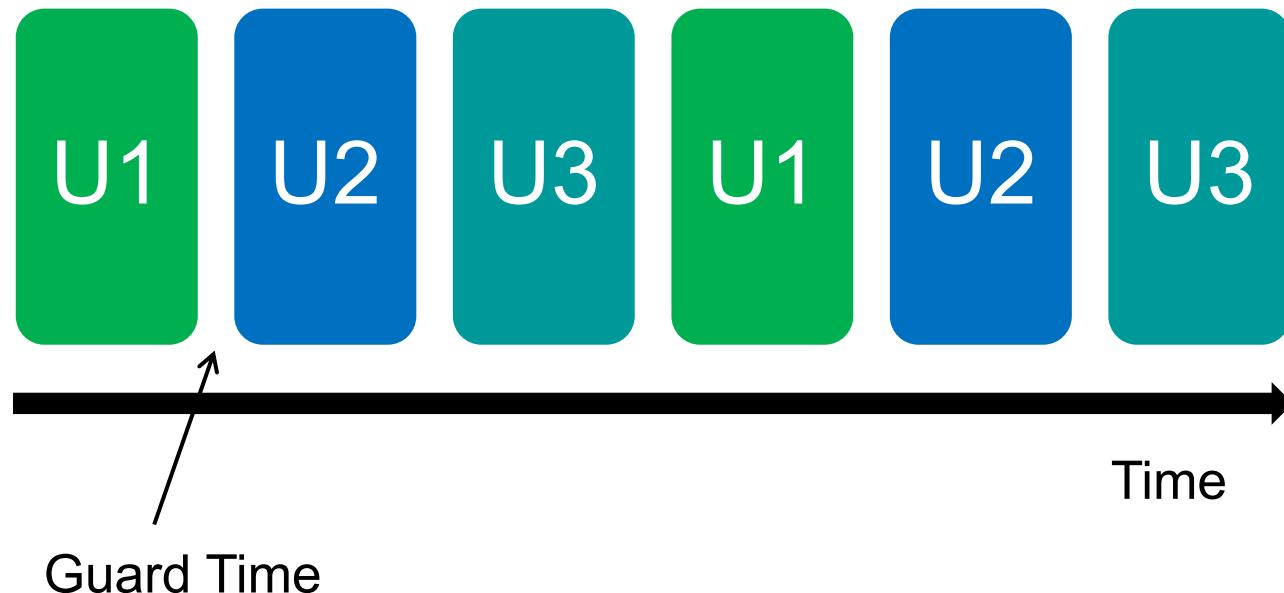
- Various methods exist for allocating a single broadcast channel amongst competing users
 - **Static Channel Allocation**
 - **Dynamic Channel Allocation**

Static Channel Allocation (1)

- Arbitrary division of a channel into segments and each user is allocated a dedicated segment for transmission
- Time Division Multiplexing (TDM) and Frequency Division Multiplexing (FDM)

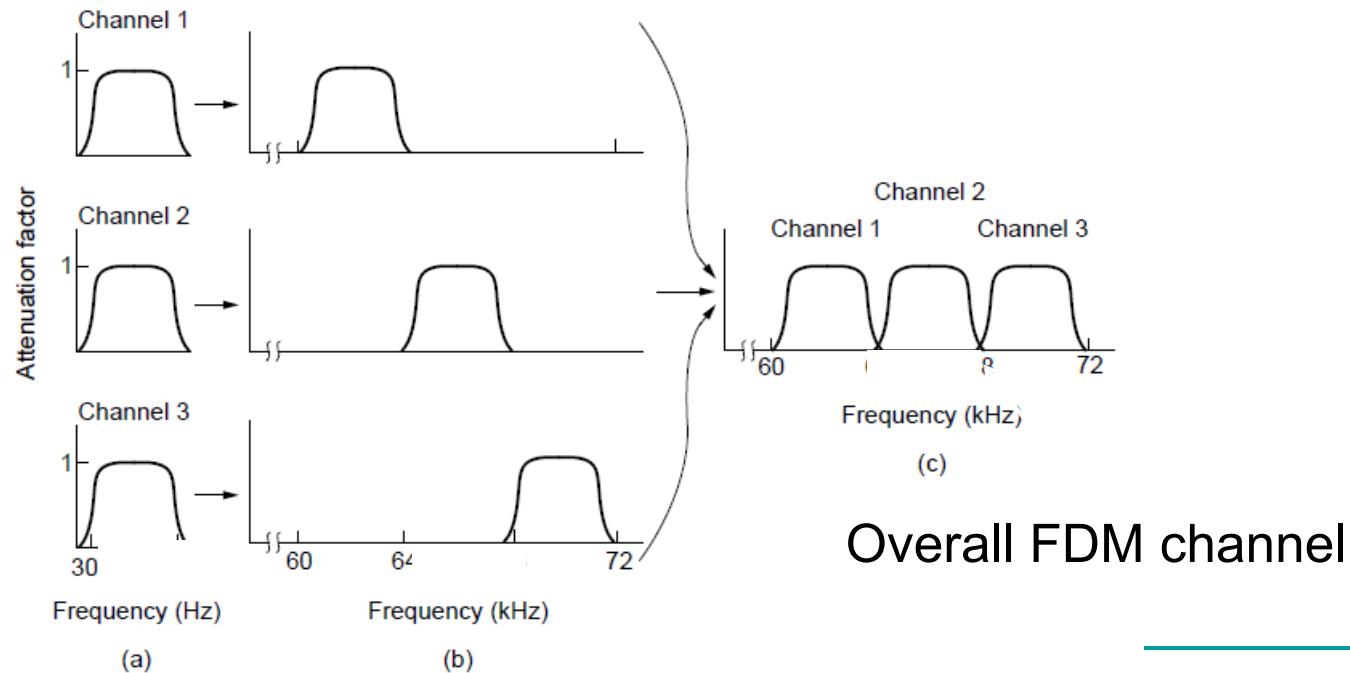
Time Division Multiplexing

- TDM: users take turns on a fixed schedule
- e.g. 2G uses TDM



Frequency Division Multiplexing

- FDM shares the channel by placing users on different frequencies.
- e.g. TV and Radio use FDM



Static Channel Allocation (2)

- Usually good for fixed number of users
- Significant inefficiencies arise when:
 - Number of senders > allocated segments
 - Number of senders is not static
 - Network Traffic is bursty: static methods TDM and FDM try to give consistent access to the network

Dynamic Channel Allocation (1)

- Channel segmentation is dynamic, segment allocation is dynamic
- Assumptions for dynamic channel allocation:
 - 1) Independent transmission stations
 - 2) Single channel for all communication
 - 3) Simultaneous transmission results in damaged frames (collision)

Dynamic Channel Allocation (2)

4) Time

- Continuous: Transmission can begin at any time
- Slotted: Transmission can begin only within discrete intervals

5) Carrier Sense

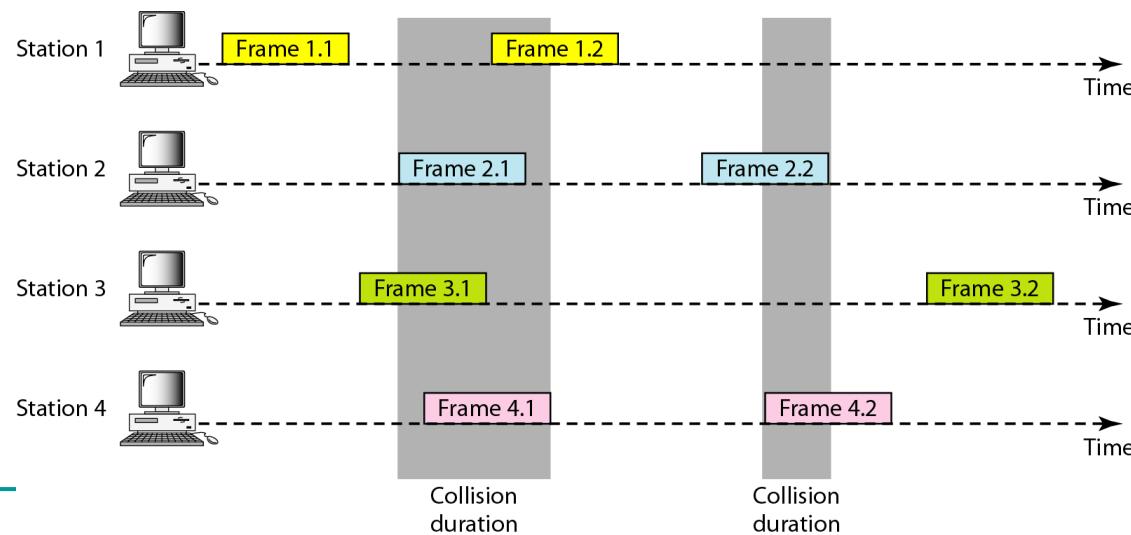
- Carrier Sense: Detection of channel use prior to transmission
- No Carrier Sense: No detection of channel use prior to transmission

Multiple Access Protocols

- Contention
 - ALOHA
 - Carrier Sense Multiple Access
- Collision Free
- Limited Contention
- MACA/MACAW (for Wireless LANs)

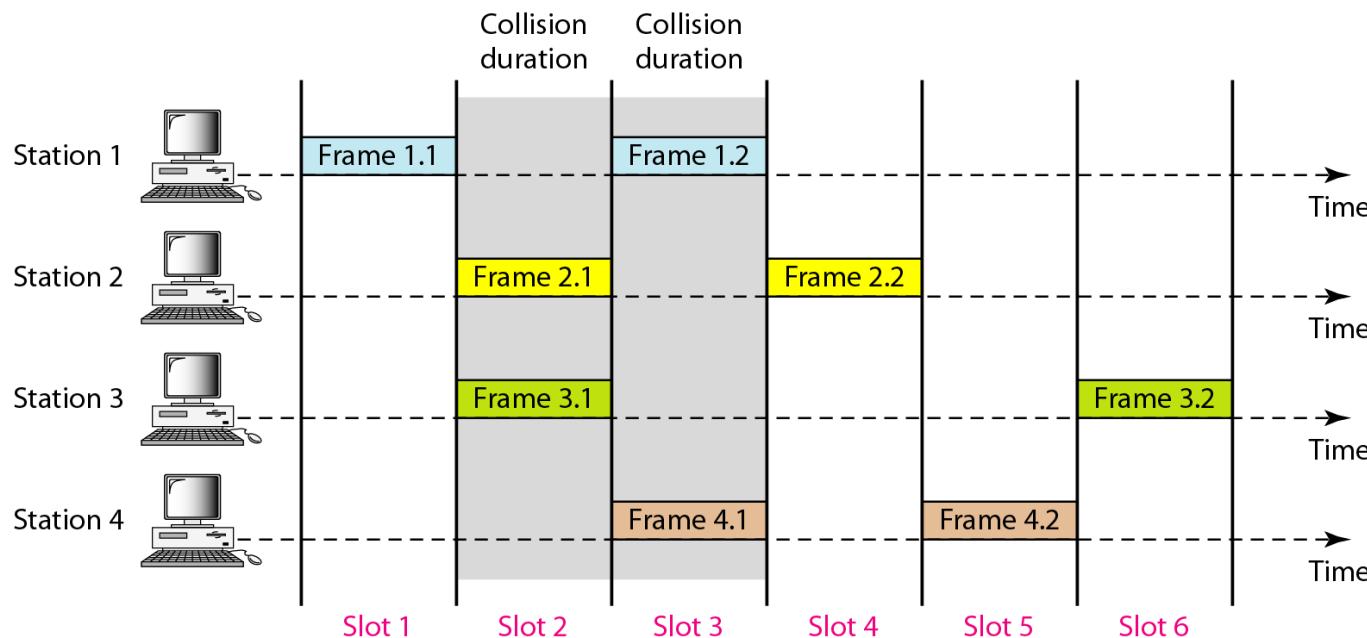
ALOHA

- Users transmit frames whenever they have data; retry after a random time if there are collisions (or no Ack is arrived)
- Requires no central control mechanism
- Efficient under low load but inefficient under high traffic loads



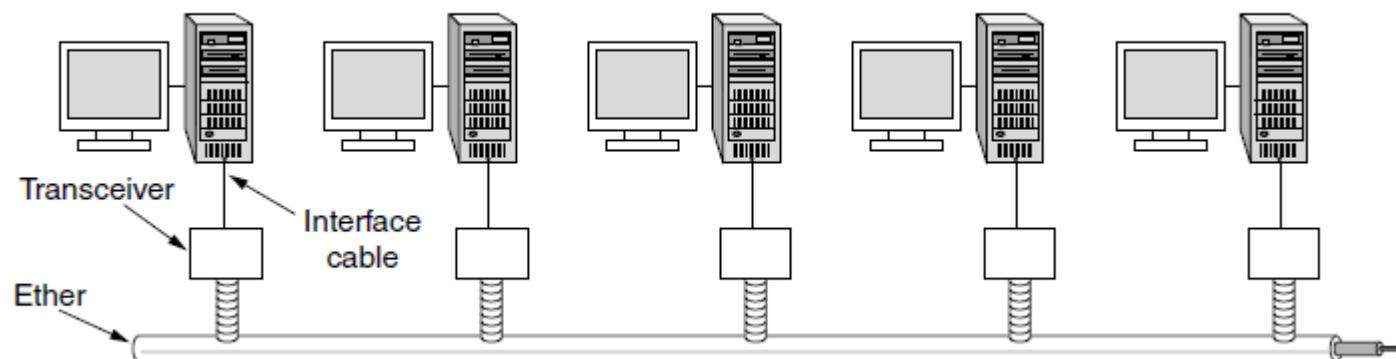
Slotted ALOHA

- Allows the users to start sending only at the beginning of defined slots.
- Increase efficiency of pure ALOHA by reducing possibility of collisions



Carrier Sense Multiple Access (CSMA)

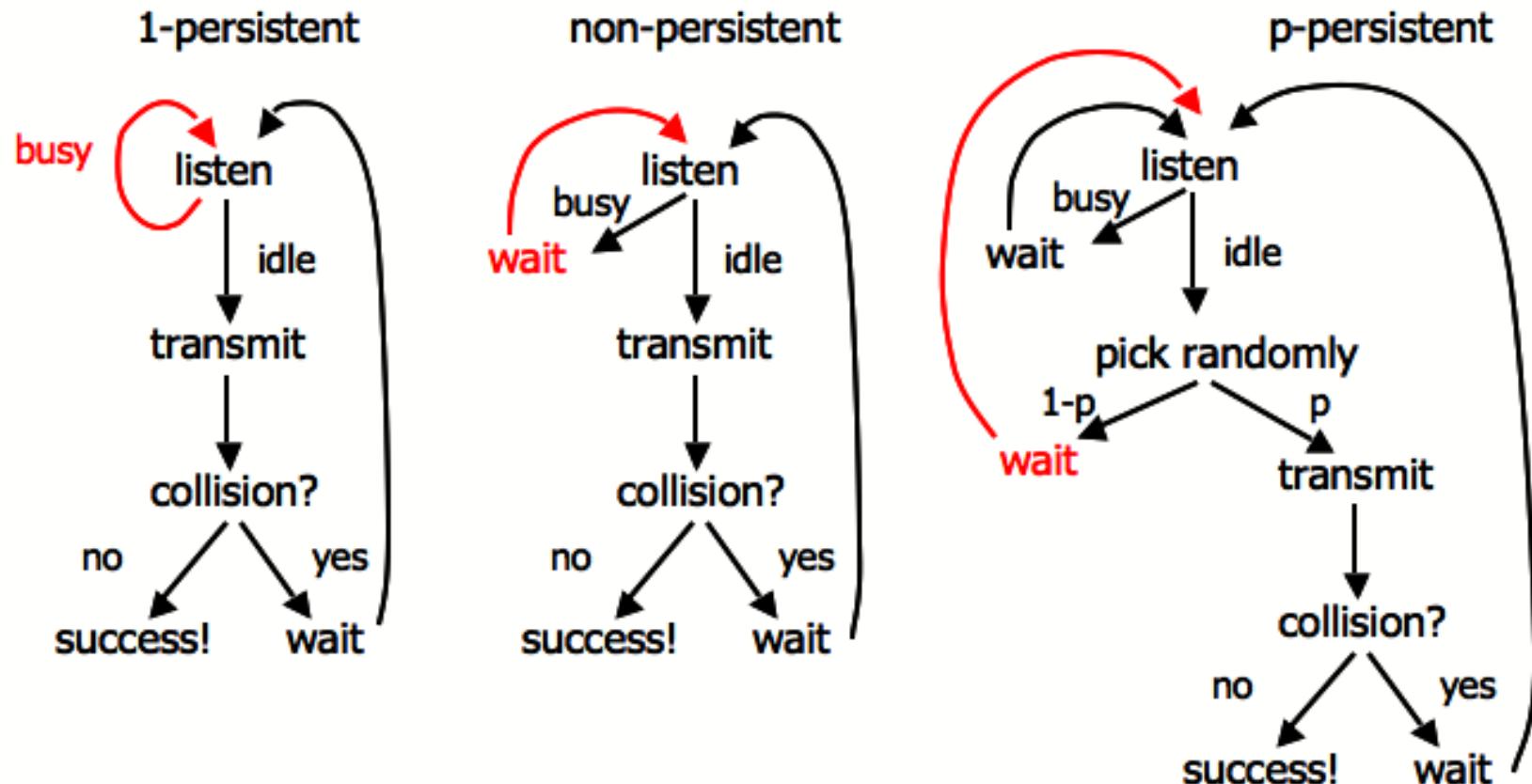
- **Require transmission state detection** to determine transmission rights dynamically, there are specific protocols which are used
 - Persistent and Non-Persistent CSMA
 - CSMA with Collision Detection



Persistent and Non-Persistent CSMA (1)

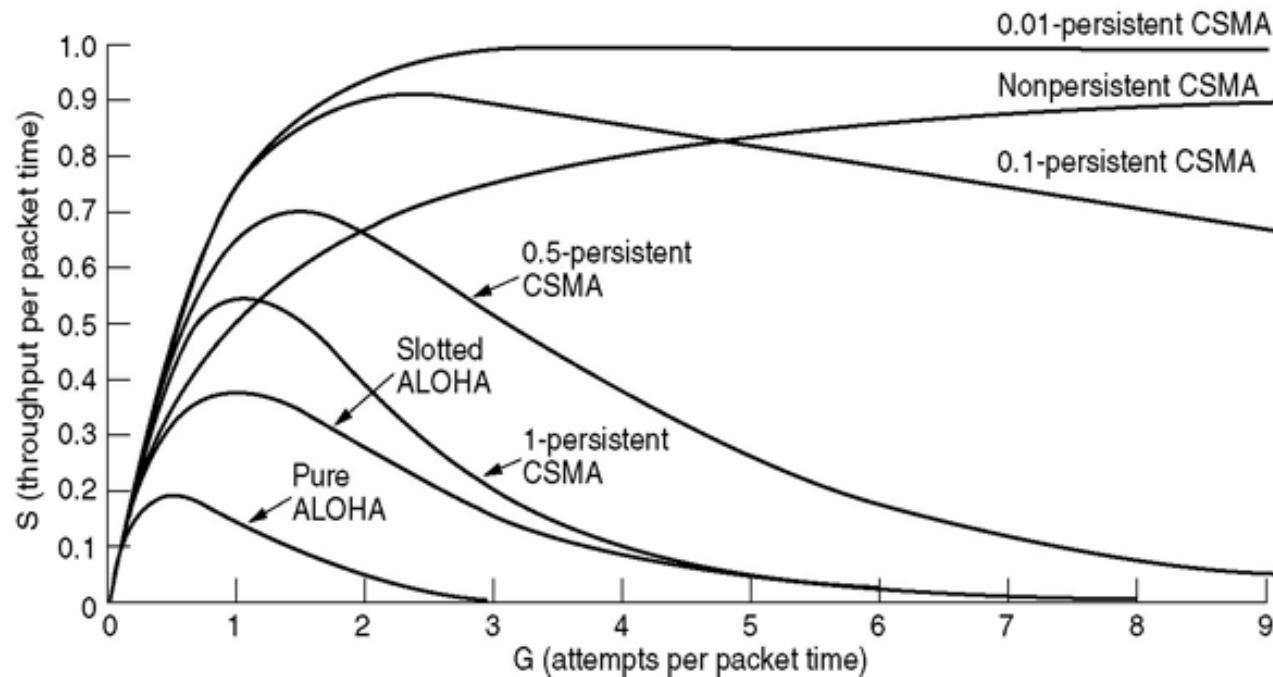
- When a sender has data to transmit, first check channel to detect other active transmission
- **1-persistent CSMA**
 - Continuously check, and wait until channel idle; transmit one frame and check collisions; if collision, wait for a random time and repeat
- **Non-persistent CSMA**
 - If channel busy, wait random period and check again; if idle, start transmitting
- **p-persistent CSMA**
 - If channel idle, transmit with probability p, or wait with probability (1-p) and check again

Persistent and Non-Persistent CSMA (2)



CSMA Variants

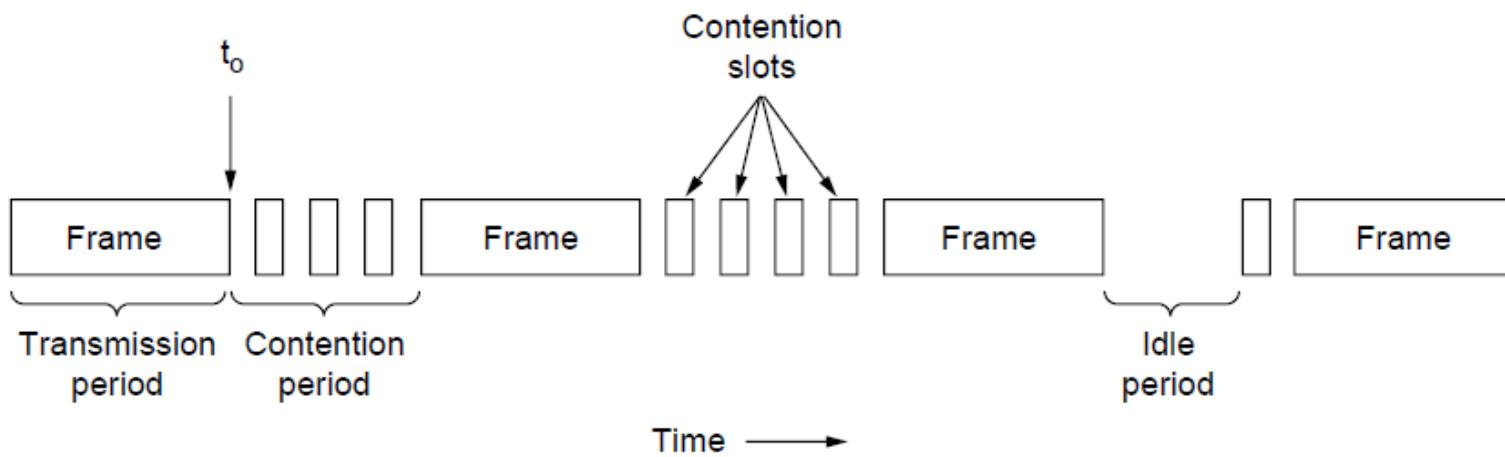
- Comparison of the efficiencies (channel utilisations) for various protocols



CSMA outperforms ALOHA, and being less persistent is better under high load

CSMA with Collision Detection

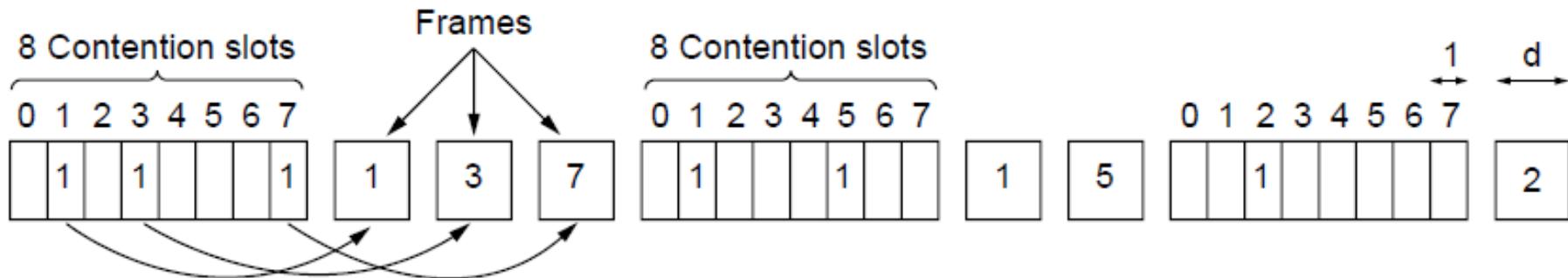
- Process: After collision detected, abort transmission, wait random period, try again
- Channel must be continually monitored
- Reduce contention times to improve performance



Collision Free Protocols (1)

■ Bit Map Protocol

- Reservation-based protocol
- 1 bit per station overhead
- Division of transmission right, and transmission event - no collisions as this is a reservation-based protocol



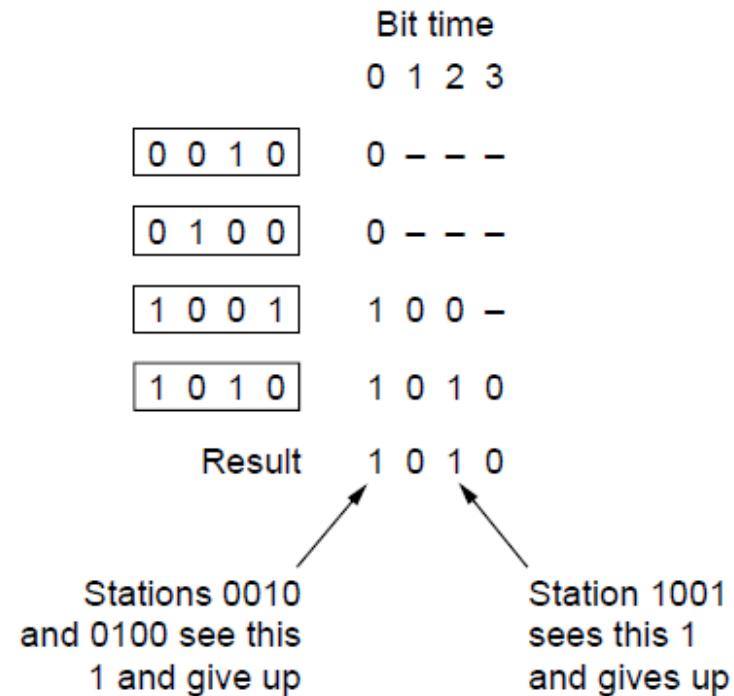
Collision Free Protocols (2)

- Binary Countdown Protocol
 - Uses binary station addressing
 - Higher numbered stations have a higher priority
 - No collisions as higher-order bit positions are used to arbitrate between stations wanting to transmit

Collision Free Protocols (3)

■ Binary Countdown Protocol

- Stations send their address in contention slot ($\log_2 N$ slots instead of N)
- Channel medium ORs bits; stations give up when they send a “0” but see a “1”
- Station that sees its full address is next to send



Contention vs. Collision Free

- **2 strategies: contention and collision free**
 - Under **low loads** (collisions are rare), the collision free is less attractive due to the overhead.
 - Under **higher loads**, contention method is less attractive due to higher number of collisions.
- Both become inefficient at different points

COMP90007 Internet Technologies

Week 4 Workshop

Semester 2, 2020

Question 1 (Sampling)

- Consider a telephone signal that is bandwidth limited to 4 kHz.
 - (a) At what rate should you sample the signal so that you can completely reconstruct the signal?
 - (b) If each sample of the signal is to be encoded at 256 levels, how many bits/symbol are required for each sample?
 - (c) What is the minimum bit rate required to transmit this signal?
- **Note:** This is a direct application of the Sampling Theorem and forms the basics of the application of the theorem, i.e. without considering data rates.

Question 2 (Sampling)

- Is the Sampling theorem true for optical fibre or only for copper wire?

Question 3 (Max Data Rate)

- Given a noiseless 4 kHz channel, what is the maximum data rate of the communications channel?

Question 4 (Max Data Rate)

- The bandwidth of a television video stream is 6 MHz. How many bits/sec are sent if four-level digital signals are used? Assume a noiseless channel

Question 4 (Max Data Rate)

- The bandwidth of a television video stream is 6 MHz. How many bits/sec are sent if four-level digital signals are used? Now assume a S/N of 20db (i.e. 100).

Question 5

The following character encoding is used in a data link protocol:

A: 01000111

B: 11100011

FLAG: 01111110

ESC: 11100000

Show the bit sequence transmitted (in binary) for the four-character frame payload *A B ESC FLAG*, when each of the following framing methods are used:

- (a) Character count
- (b) Flag bytes with byte stuffing
- (c) Starting and ending flag bytes, with bit stuffing

Question 6

The following data fragment occurs in the middle of a data stream for which the byte-stuffing algorithm as described in the lecture is used:

A B ESC C ESC FLAG FLAG D.

What is the output after stuffing?

COMP90007 Internet Technologies

Week 4 Workshop

Semester 2, 2020

Suggested solutions

Question 1 (Sampling)

- Consider a telephone signal that is bandwidth limited to 4 kHz.
 - (a) At what rate should you sample the signal so that you can completely reconstruct the signal?
 $\text{min. sampling rate} = 2 \times 4000 = 8 \text{ kHz} = 8000 \text{ samples/s}$
 - (b) If each sample of the signal is to be encoded at 256 levels, how many bits/symbol are required for each sample?
 $256 \text{ possible values per sample requires } \log_2(256) = 8 \text{ bits/sample}$
 - (c) What is the minimum bit rate required to transmit this signal?
 $8 \text{ bits/sample} \times 8000 \text{ samples/s} = 64 \text{ kbps}$
- **Note:** This is a direct application of the Sampling Theorem and forms the basics of the application of the theorem, i.e. without considering data rates.

Question 2 (Sampling)

- Is the Sampling theorem true for optical fibre or only for copper wire?
 - The Sampling theorem is a property of mathematics and has nothing to do with technology.
 - The Sampling theorem states that if you have a function which does not contain any frequency components (sines or cosines) above f , then by sampling at a frequency of $2f$, you capture all the information there is. The Sampling theorem is independent of the transmission medium.

Question 3 (Max Data Rate)

- Given a noiseless 4 kHz channel, what is the maximum data rate of the communications channel?
 - A noiseless channel can carry an arbitrarily large amount of information, e.g. there can be an infinite number of signalling levels, this is because there is no noise. This is a neat observation and the level information is not restricted by the question in any way. Shannon specifies a limit on the information rate based on given noise.

Question 4 (Max Data Rate)

- The bandwidth of a television video stream is 6 MHz. How many bits/sec are sent if four-level digital signals are used? Assume a noiseless channel
 - The maximum baud rate is 12 symbols/sec
 - Four levels of signalling provide: $\log_2 4 = 2$ bits/symbol
 - Hence, the total data rate is: $12 \text{ million symbols/s} \times 2 \text{ bits/symbol} = 24 \text{ Mbps}$

Question 4 (Max Data Rate)

- The bandwidth of a television video stream is 6 MHz. How many bits/sec are sent if four-level digital signals are used? Now assume a S/N of 20db (i.e. 100).

- Using Shannon's theorem, we have:
$$B \times \log(1+S/N) = 6\text{MHz} \times \log_2(1+100) = 6\text{MHz} \times 6.65 = 39.9\text{Mbps}$$

Note: Using Nyquist's theorem, we have: $2B \times \log_2 V$

$$= 2 * 6\text{MHz} \times \log_2 4 = 12\text{MHz} \times 2 = 24\text{Mbps}$$

The bottleneck is therefore the Nyquist limit, giving a maximum channel capacity of 24Mbps.

Question 5

The following character encoding is used in a data link protocol:

A: 01000111

B: 11100011

FLAG: 01111110

ESC: 11100000

Show the bit sequence transmitted (in binary) for the four-character frame payload *A B ESC FLAG*, when each of the following framing methods are used:

- (a) Character count
- (b) Flag bytes with byte stuffing
- (c) Starting and ending flag bytes, with bit stuffing

Answer:

1. 00000101 01000111 11100011 11100000 01111110
5 A B 'ESC' 'FLAG'

2. 01111110 01000111 11100011 11100000 11100000 11100000 01111110 01111110
FLAG A B ESC 'ESC' ESC 'FLAG' FLAG

3. 01111110 01000111 110100011 111000000 011111010 01111110
FLAG A B 'ESC' 'FLAG' FLAG

Question 6

The following data fragment occurs in the middle of a data stream for which the byte-stuffing algorithm as described in the lecture is used:

A B ESC C ESC FLAG FLAG D.

What is the output after stuffing?

Answer:

After stuffing we get:

A B ESC ESC C ESC ESC FLAG ESC FLAG D.

MAC Sub-Layer

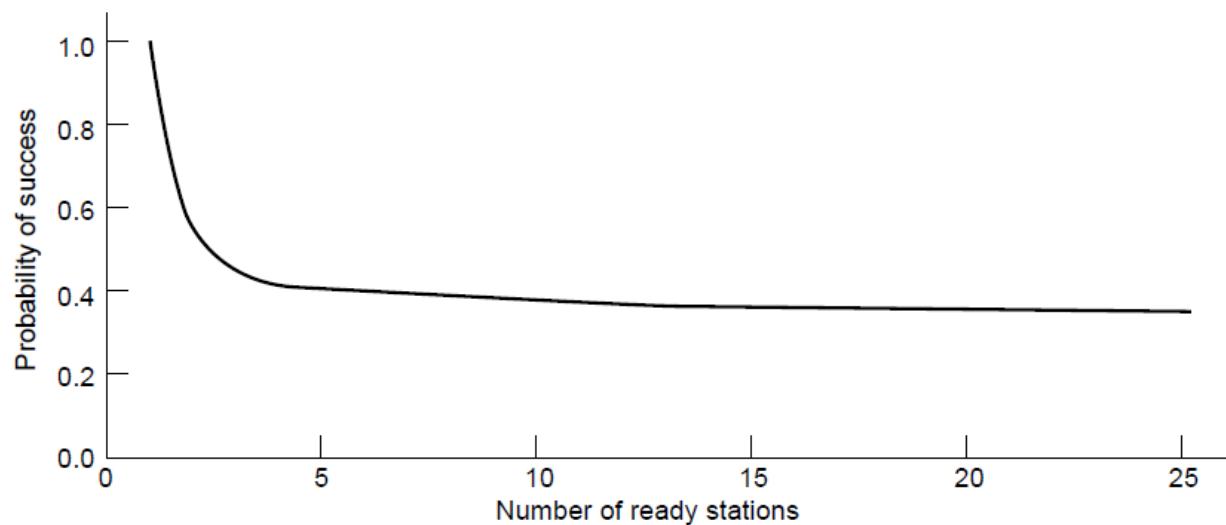
COMP90007 Internet Technologies

Lecturer: Ling Luo

Semester 2, 2020

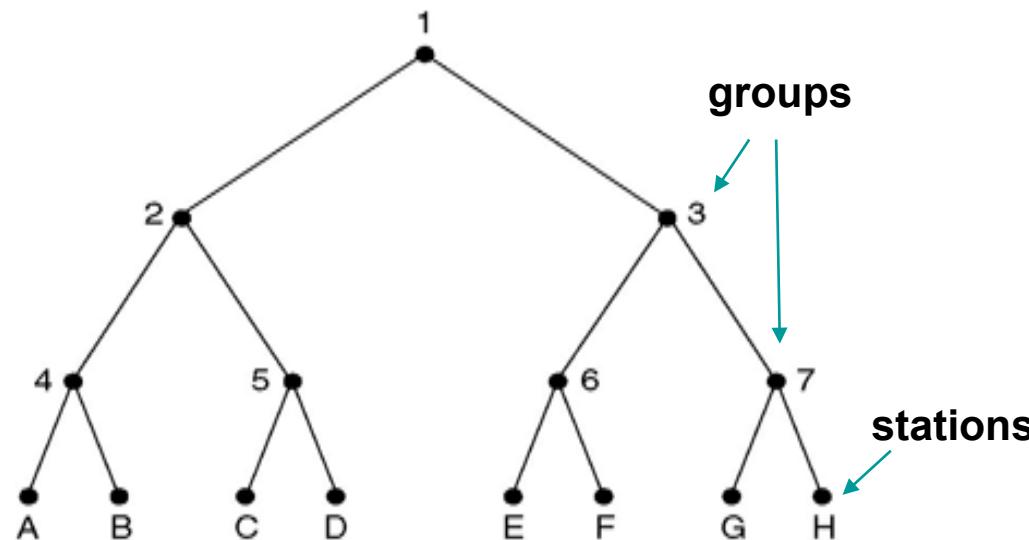
Limited Contention Protocols

- Idea: divide stations into groups, within which only a very small number are likely to transmit data.
- Increase the probability of stations acquiring transmission rights by dividing stations and using a binary algorithm to determine right allocation
- Avoid wastage due to idle periods and collisions



Adaptive Tree Walk Protocol

- All stations compete for right to transmit, if a collision occurs, binary division is used to resolve contention
- Stations are divided into groups to poll
 - Depth first search under nodes with poll collisions
 - Start search at lower levels if >1 station want to transmit



Example 1: D G

Slot 1 → D, G – collision

Slot 2 → D

Slot 3 → G

Example 2: B D G

Slot 1 → B, D, G – collision

Slot 2 → B, D - collision

Slot 3 → B

Slot 4 → D

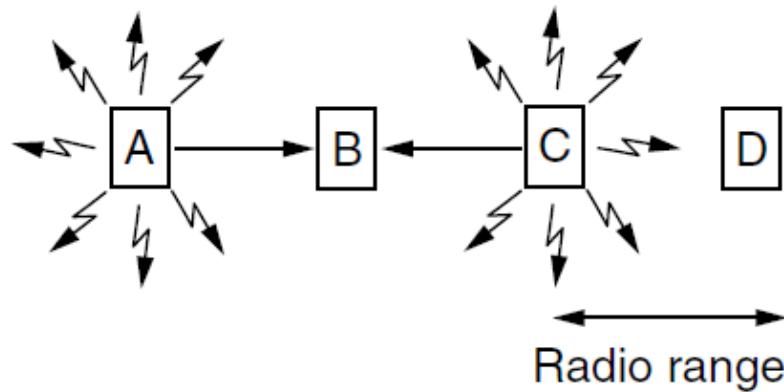
Slot 5 → G

Wireless LAN Protocols

- Wireless Complications: stations have coverage regions, which leads to **hidden** and **exposed terminal** problems.
- When a station is in the range of two transmitters or relays, interference affects signal reception.
- Require **detection of transmissions around receiver, not just carrier sensing.**
- Transmission Protocols for Wireless LANs (802.11)
 - Multiple Access with Collision Avoidance for Wireless (MACAW)

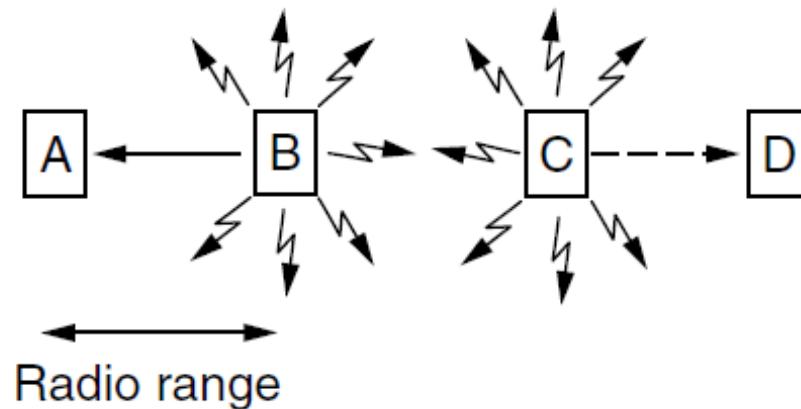
Hidden and Exposed Terminals (1)

- **Hidden terminals** are senders that cannot sense each other but nonetheless collide at intended receiver
 - A and C are hidden terminals when sending to B
 - Want to prevent; loss of efficiency



Hidden and Exposed Terminals (2)

- **Exposed terminals** are senders who can sense each other but still transmit safely (to different receivers)
 - $B \rightarrow A$ and $C \rightarrow D$ are exposed terminals
 - Desirably concurrency; improves performance



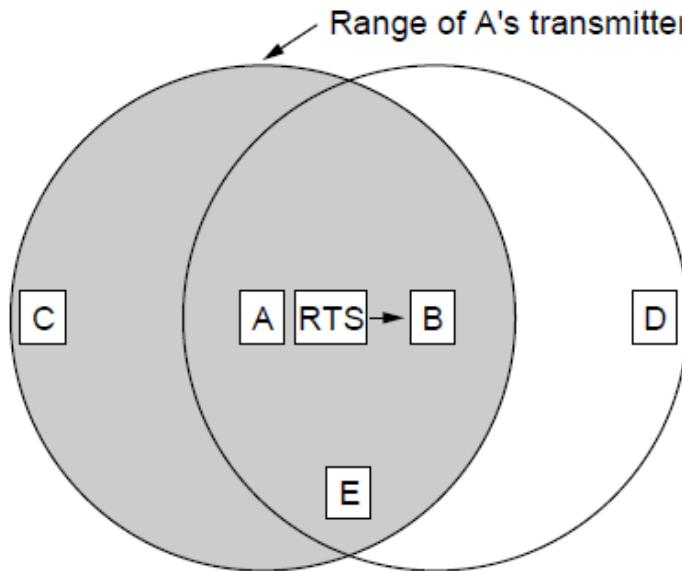
MACA (1)

- MACA: Multiple Access with Collision Avoidance
- Sender asks receiver to transmit short control frame
- Stations near receiver hear control frame
- Sender can then transmit data to receiver

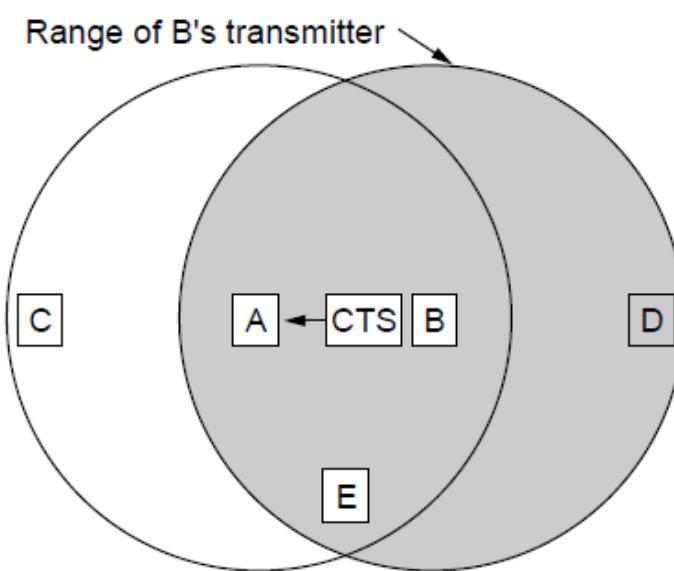
MACA (2)

MACA protocol grants access for A to send to B:

- ❑ A sends RTS to B [left]; B replies with CTS [right]
- ❑ A can send with exposed but no hidden terminals



A sends RTS to B; C and E hear and defer for CTS



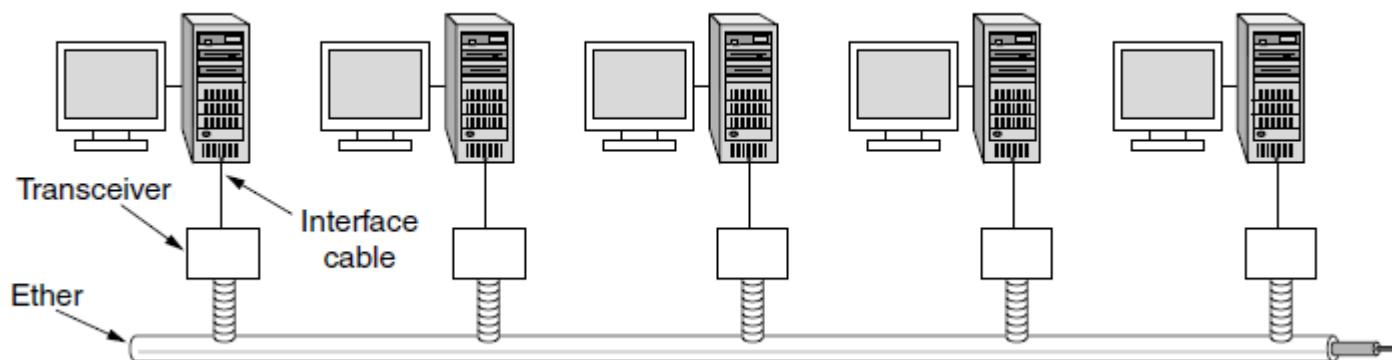
B replies with CTS; D and E hear and defer for data

Ethernet

- MAC Sub-Layer Case Study
 - Classic Ethernet
 - Switched Ethernet

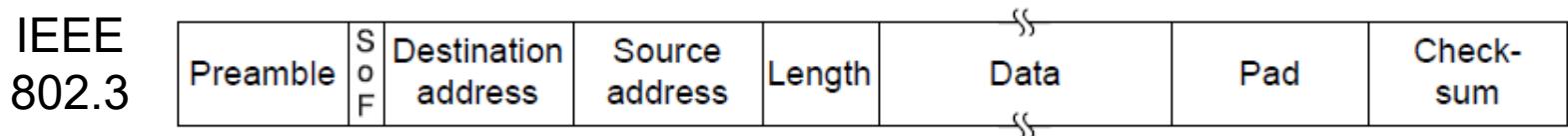
Classic Ethernet

- Each type of Ethernet has a maximum cable length per segment.
- Multiple cable lengths can be connected by repeaters - a physical device which receives, amplifies and retransmits signals in both directions.



Ethernet Frame Format

- MAC protocol is 1-persistent CSMA/CD
 - Random delay (backoff) after collision is computed with BEB (Binary Exponential Backoff, i.e., random number 0 to $2^i - 1$)
- Frame format is still used with modern Ethernet



Preamble (7B) – synchronisation between sender and receiver

Start of Frame (1B) – FLAG bytes

Dest. & Source addresses (6B + 6B) – to identify sender and receiver

Type or Length (2B) – specifies which process to give the frame to

(0x0800 means data contains IPv4)

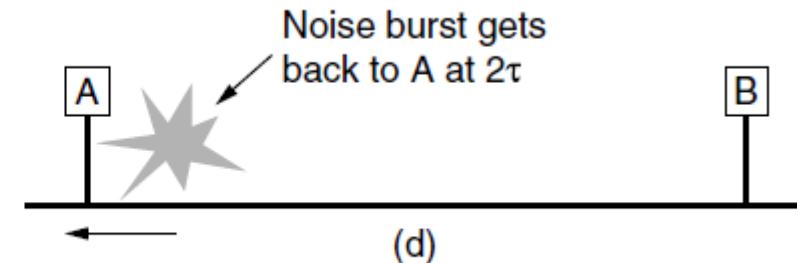
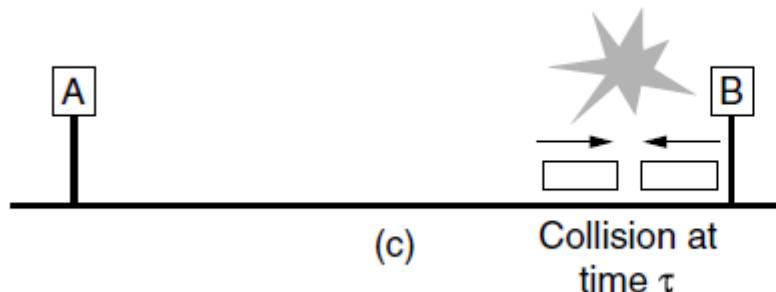
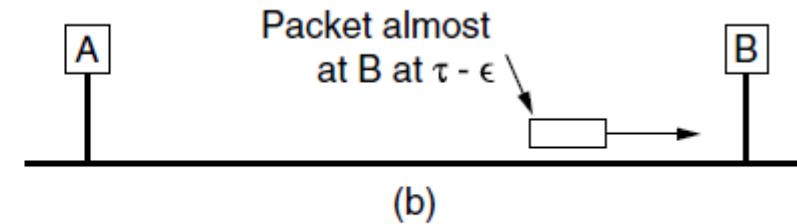
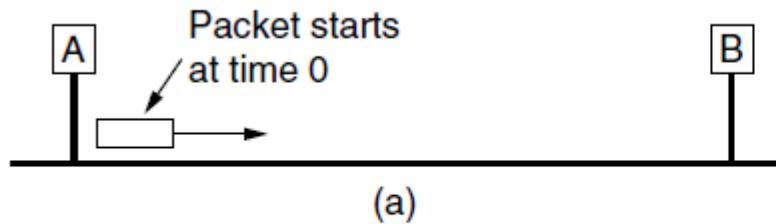
Data (0~1500B)

Pad(0~46B) – minimum size of the message of the Ethernet is 64 Bytes

CRC (4B) – 32 bits checksum

Classic Ethernet Minimum Packet Size

- Collisions can occur and take as long as 2τ to detect
 - τ is the time it takes to propagate over the Ethernet
 - Leads to minimum packet size for reliable detection



MAC Addressing

- Source and Destination Addressing can be done at a local or global levels
- The **MAC Address** provides the unique identifier for a physical interface
- MAC Address is a 48-bit number encoded in the frame, written in hexadecimal notation
 - e.g. 00:02:2D:66:7C:2C

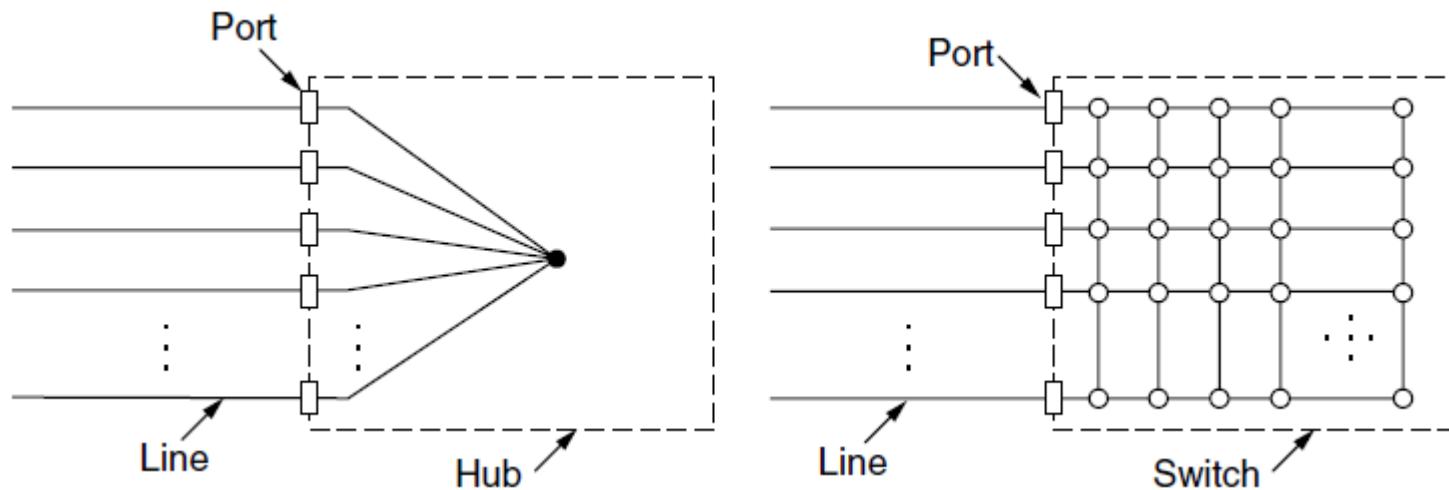
Ethernet Performance

$$\text{Channel Efficiency} = \frac{1}{1 + (2BLe)/(cF)}$$

- F: frame length
 - B: bandwidth
 - L: cable length
 - c: speed of signal propagation; e: constant ≈ 2.71828
 - Optimal case: e contention slots per frame
-
- When cF is large, the channel efficiency will be high.
 - Increasing network bandwidth or distance (BL) reduces the efficiency for a given frame size.

Switched Ethernet

- Hubs wire all lines into a single CSMA/CD domain
- Switches isolate each port to a separate domain
 - Much greater throughput for multiple ports
 - No need for CSMA/CD with full-duplex lines



Summary of Multiple Access Protocols

- Contention
 - ALOHA, Slotted ALOHA
 - Carrier Sense Multiple Access: 1-persistent, non-persistent, p-persistent
- Collision Free: bit map, binary countdown
- Limited Contention: adaptive tree walk
- MACA/MACAW (for Wireless LANs): RTS and CTS

Network Layer

COMP90007 Internet Technologies

Lecturer: Ling Luo

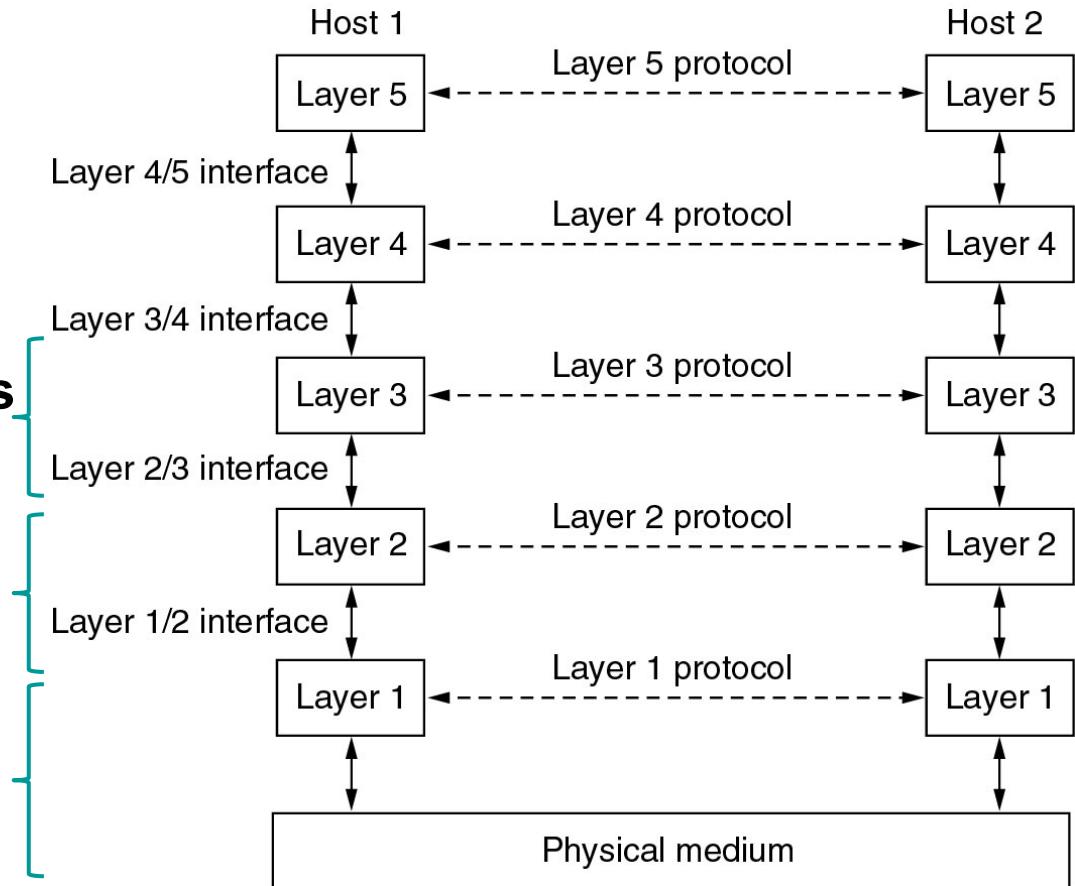
Semester 2, 2020

Network Layer

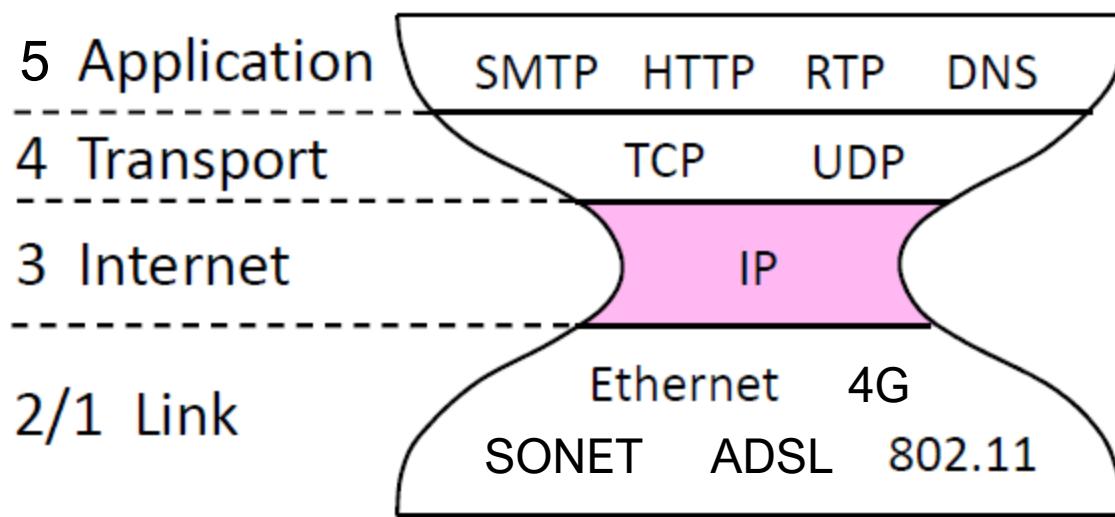
**Connecting different networks
(internetworking)**

Framing, error and flow control, MAC

Different cables, wireless, signal, digital to analogue

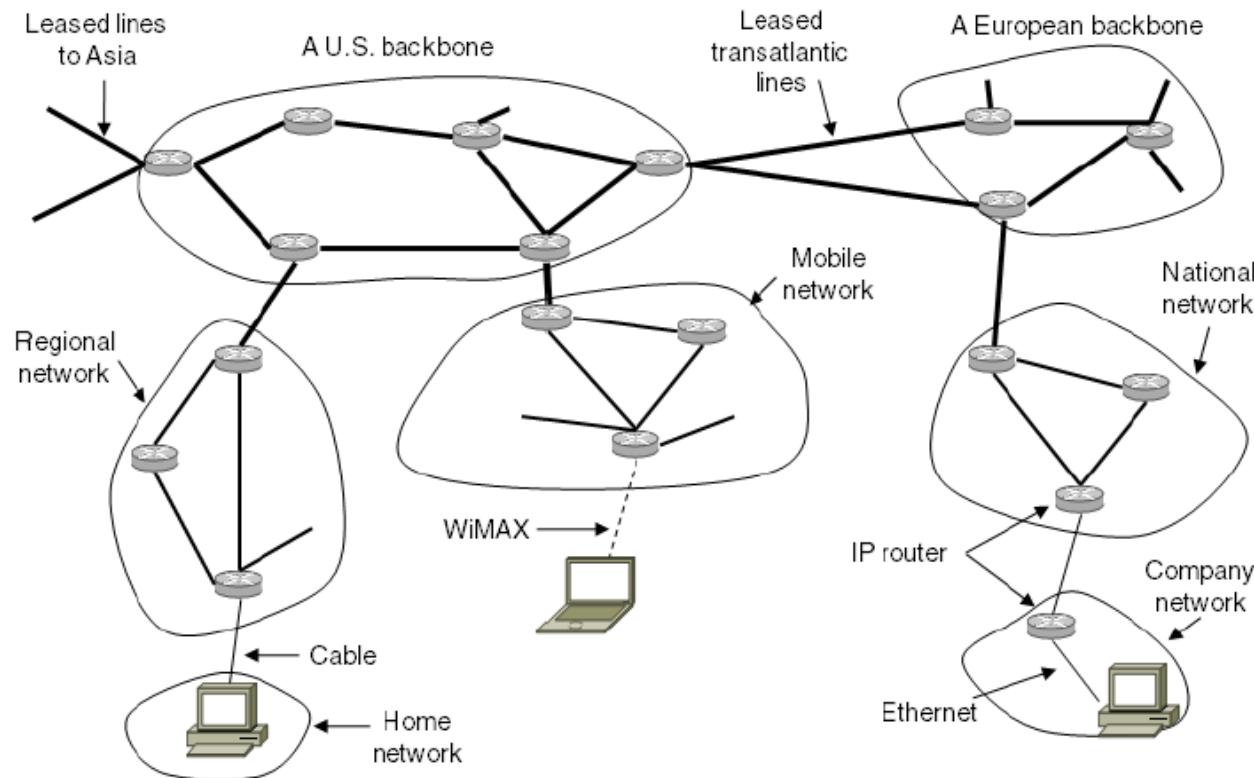


Internet

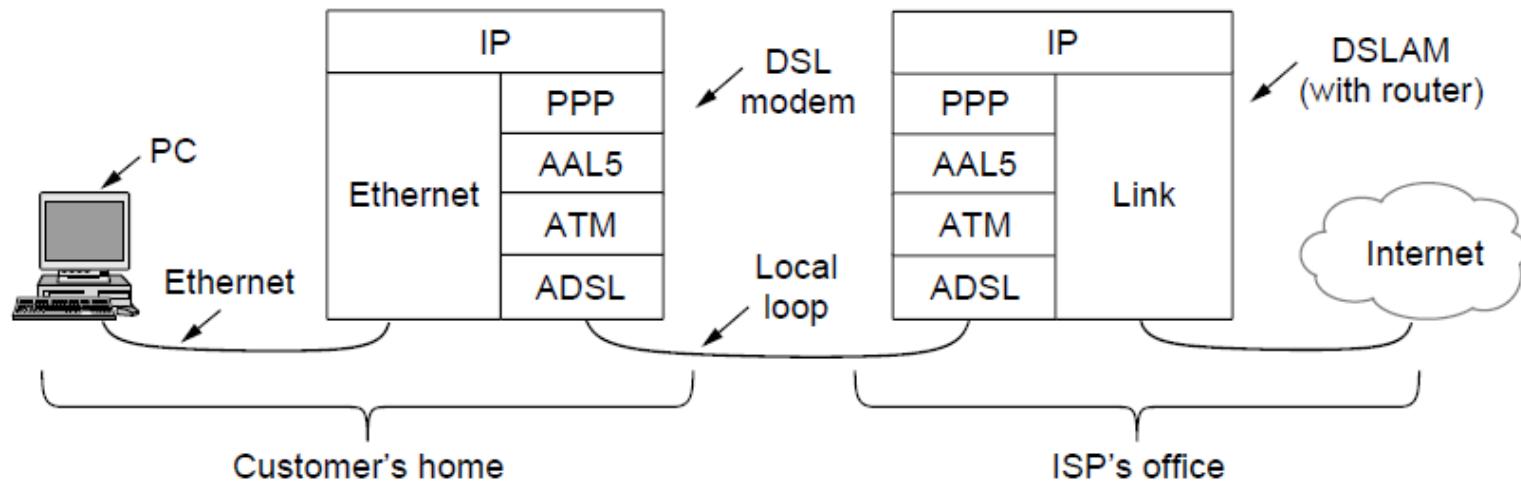


Network Layer in the Internet (1)

- Internet is a collection of many networks that is interconnected by the IP protocol



Network Layer in the Internet (2)

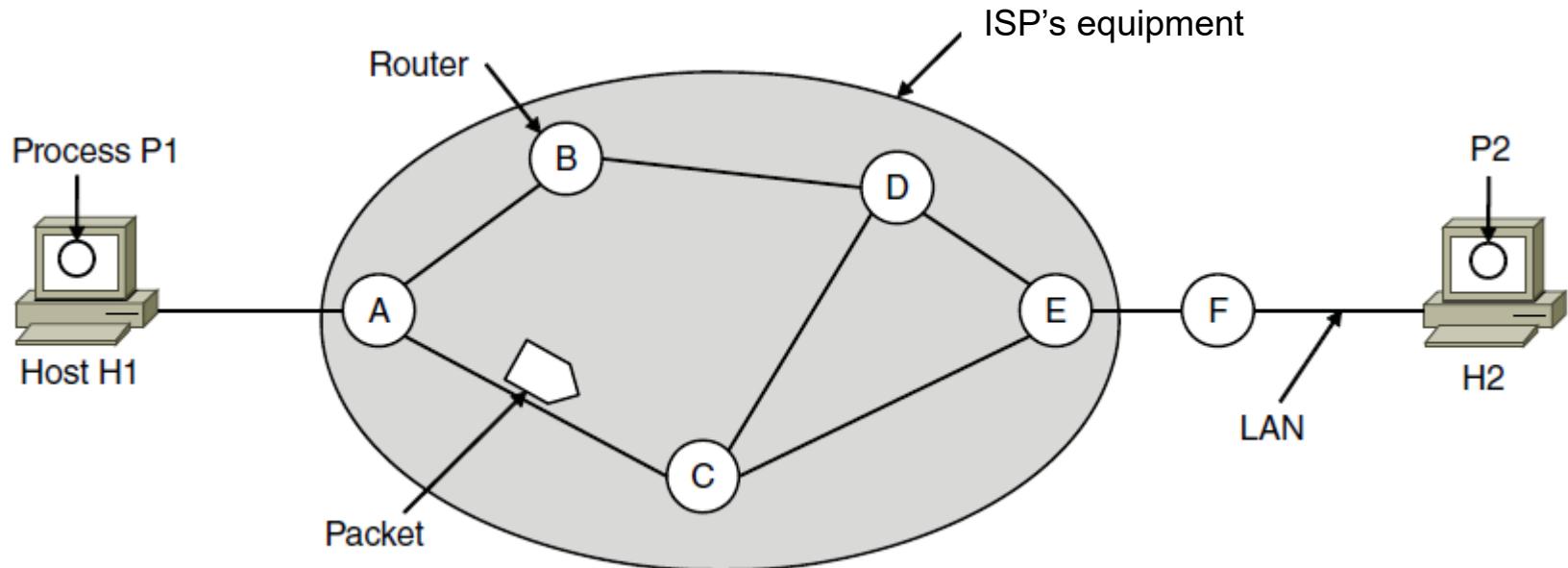


Internet Protocol (IP)

- The network layer protocol IP holds the whole Internet together.
- Provides a **best-effort** service to route datagrams from source host to destination host
- These hosts may be
 - On the same network
 - On different networks

Store-and-Forward Packet Switching

- Hosts generate packets and inject into the network
- Routers treat packets as messages, receiving/storing them and then forwarding them based on how the message is addressed
- **Router routes packets through the network**



Services Provided to the Transport Layer

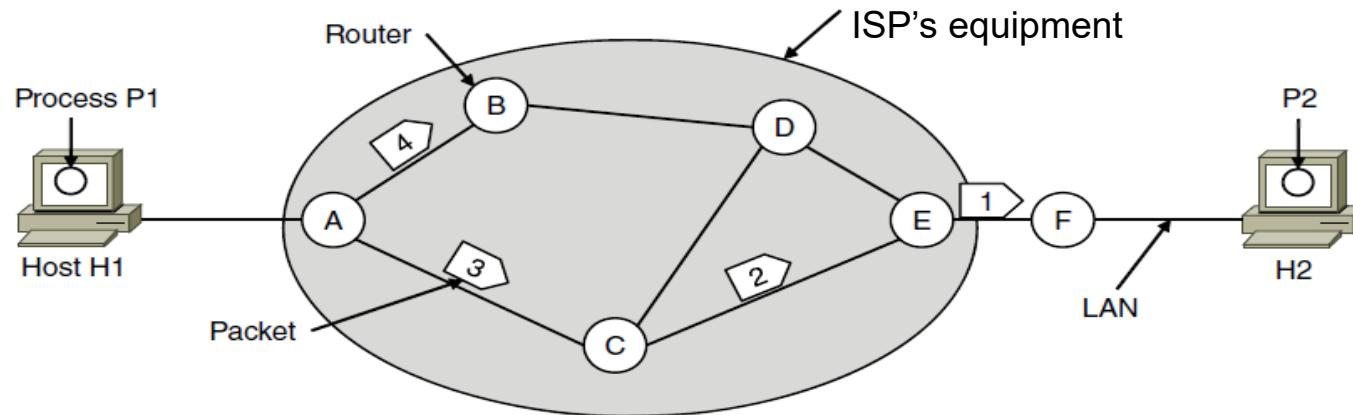
- Design goals:
 - Services should be **independent of router technologies**
 - Transport layer should be shielded from number, type and topology of routers
 - Network addressing should use a uniform numbering plan (network identifier)

Types of Services

- **Connectionless:** Packets (datagrams) are injected into subnet **individually** and routed **independently** to destination
 - Internet: move packets in a potentially unreliable subnet; QoS is not easily implemented
 - Flow and error control done by other layers
- **Connection-oriented:** Packets travelling between destinations, following the **same route**
 - Telecommunication: guarantee reliability; QoS is important

Routing within a Datagram Subnet

- **Connectionless - post office model:** packets are routed individually based on destination addresses in them
- Packets can take different paths
- E.g., P1 sends a long message to P2



A's table (initially)

A	☒
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	☒
B	B
C	C
D	B
E	B
F	B

C's Table

A	A
B	A
C	☒
D	E
E	E
F	E

E's Table

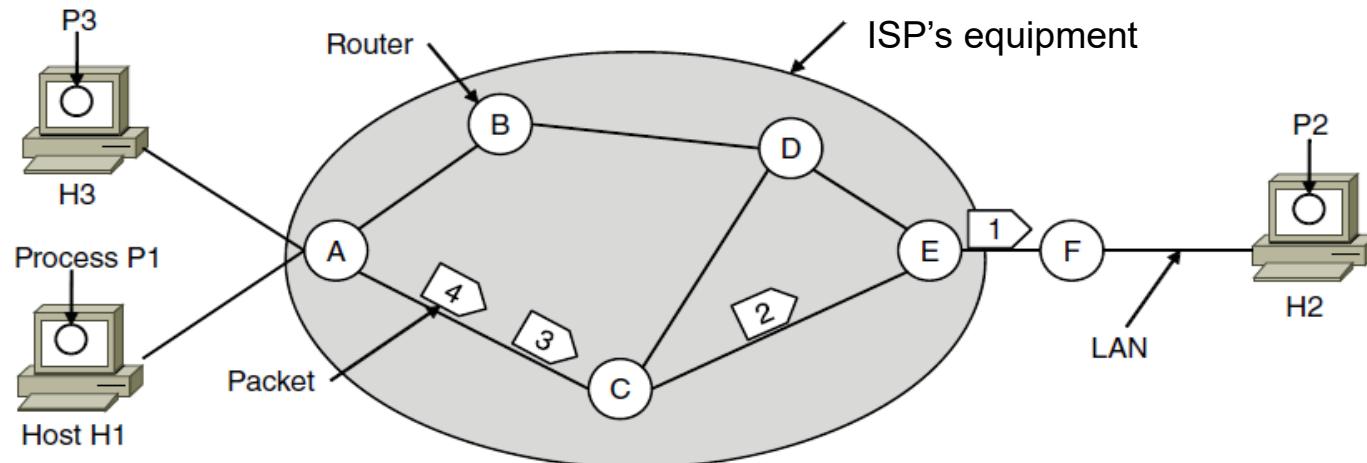
A	C
B	D
C	C
D	D
E	☒
F	F

Routing table (can be fixed or change over time)

Routing algorithm – manages the routing table

Routing within a Virtual-Circuit Subnet

- **Connection-oriented - telephone network model:** Packets are routed through virtual circuits (created earlier) based on tag number (not full address but unique at a given link) in them
 - Packets take the same path to avoid having to choose a new route for every packet
 - e.g., MultiProtocol Label Switching Network



connection identifier	A's table	C's Table	E's Table												
	<table border="1"><tr><td>H1</td><td>1</td></tr><tr><td>C</td><td>1</td></tr></table> <p>in out</p>	H1	1	C	1	<table border="1"><tr><td>A</td><td>1</td></tr><tr><td>E</td><td>1</td></tr></table>	A	1	E	1	<table border="1"><tr><td>C</td><td>1</td></tr><tr><td>F</td><td>1</td></tr></table>	C	1	F	1
H1	1														
C	1														
A	1														
E	1														
C	1														
F	1														

Datagram vs. Virtual-Circuit Subnets

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Compromises in VC and Datagram Subnets (1)

- Setup time vs. address parsing time
 - VC: requires setup time and resources, but packet transmission is very fast after that
 - Datagram: more complicated lookup procedure
- Memory of router
 - VC: requires entry per virtual circuit
 - Datagram: requires large tables of every possible destination route
- Bandwidth
 - VC: saves potential overhead in full addressing of each packet and computation of path. Still needs them during setup
 - Datagram: full destination address in every packet

Compromises in VC and Datagram Subnets (2)

- QoS and congestion avoidance
 - VC: easier to provide QoS, able to reserve CPU, bandwidth and buffer in advance
- Longevity
 - VC: can be setup for repeating and long-running uses e.g. Permanent VC's
- Vulnerability
 - VC: particularly vulnerable to hardware/software crashes, all VC's aborted and no traffic until they are rebuilt
 - Datagram: can use an alternative route

Different Networks

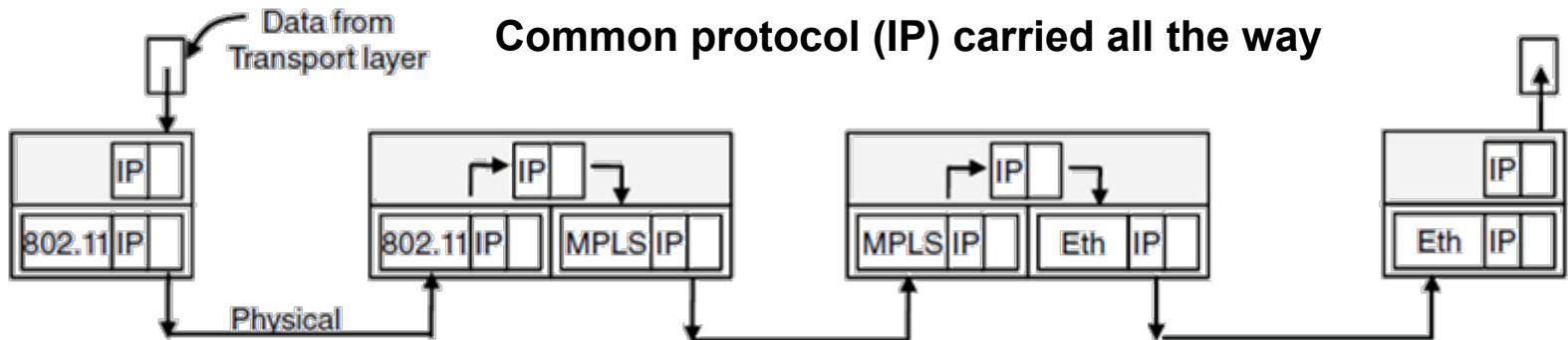
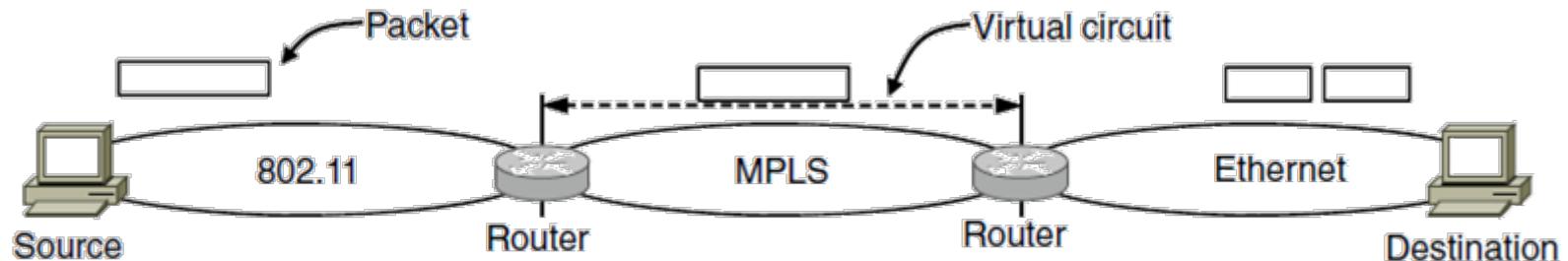
- Service offered: connectionless vs. connection-oriented
- Packet size: different max
- Addressing: different sizes, flat or hierarchical
- Quality of service: present or absent
- Reliability: different levels of loss
- Security: privacy rules, encryption
- Parameters: different timeouts

Internetworking

- Internetworking joins multiple, different networks into a single larger network
- Issues when connecting networks:
 - Different network types and protocols
 - Different motivations for network choices
 - Different technologies at both hardware and software levels

How Different Networks are Connected

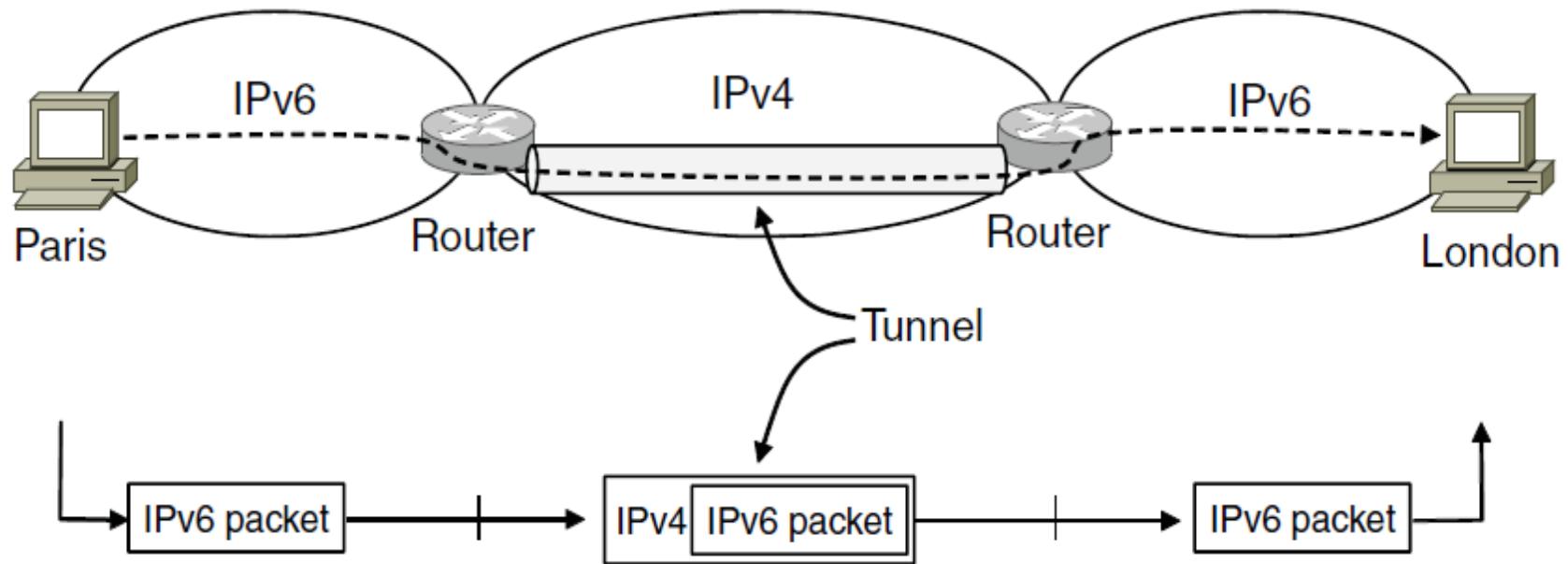
- Internetworking based on a common network layer – IP



Tunneling

- Tunneling is a special case used when the source and destination are on the same network, but there is a different network in between.
 - Source packets are encapsulated in packets, travelling through connecting network

Tunneling IPv6 Packets through IPv4



Network Layer

COMP90007 Internet Technologies

Lecturer: Ling Luo

Semester 2, 2020

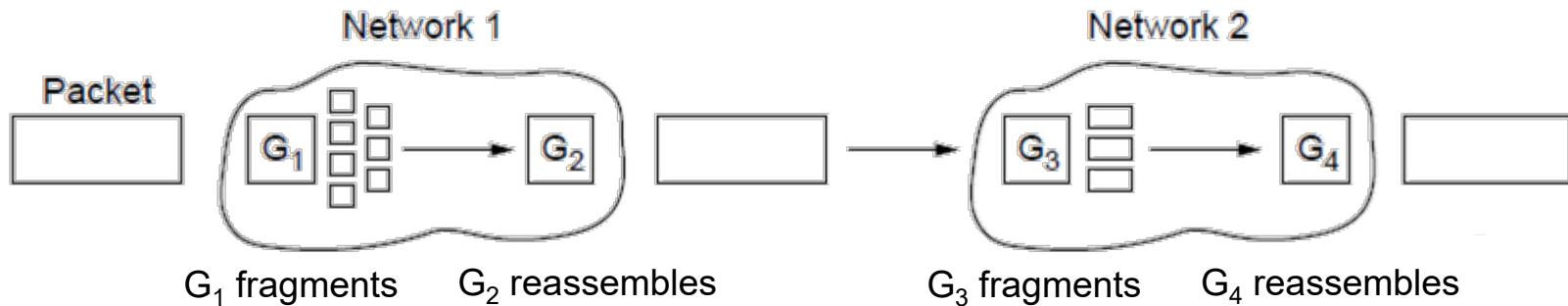
Fragmentation

- All networks have a maximum size for packets (Maximum Transmission Unit, MTU)
 - Hardware and operating system
 - Protocols and standards compliance
 - Efficiency of transmission
- Fragmentation divides packets into fragments
 - Large packets need to be routed through a network whose maximum packet size is too small.

Types of Fragmentation (1)

■ Solution: Fragmentation and Reassembly.

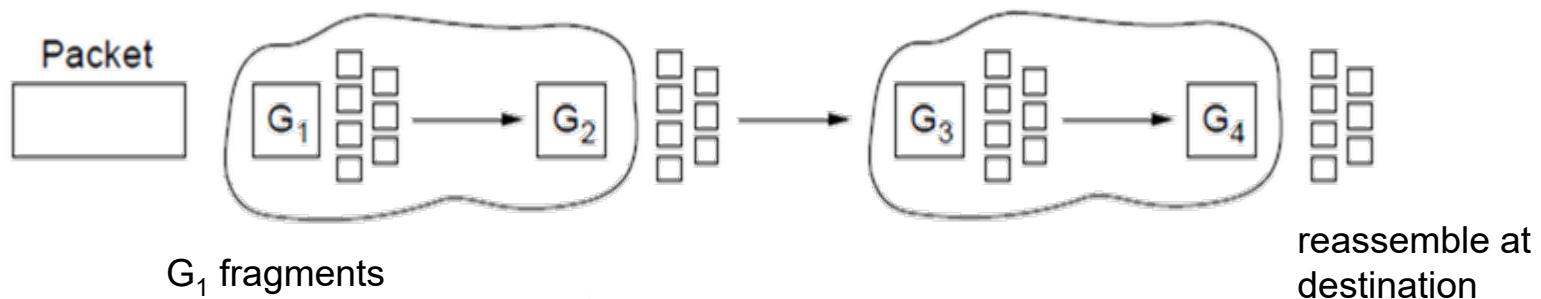
Transparent: packets fragmented & reassembled in each network. Route constrained, more work.



Types of Fragmentation (2)

■ Solution: Fragmentation and Reassembly.

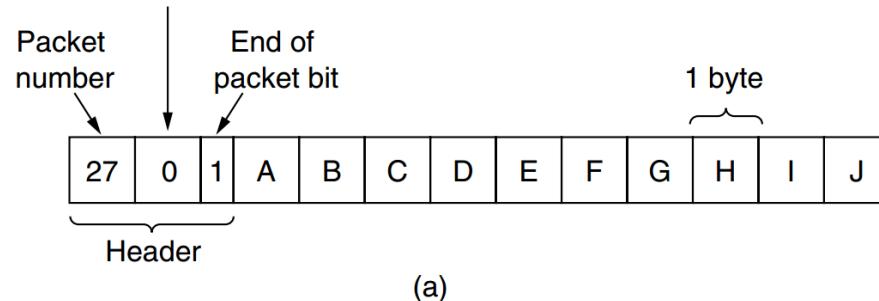
Non-transparent: fragments are reassembled at destination.
Router has less work, but each packet requires packet number,
byte offset, end of packet flag. IP works this way.



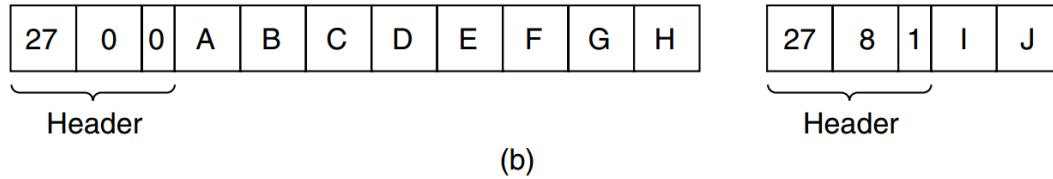
IP-Style Fragmentation

Original packet:
(10 data bytes)

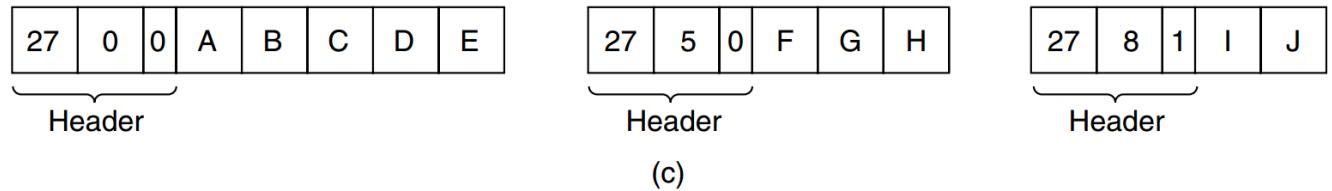
Number of the first elementary fragment in this packet



Fragmented:
(to 8 data bytes)

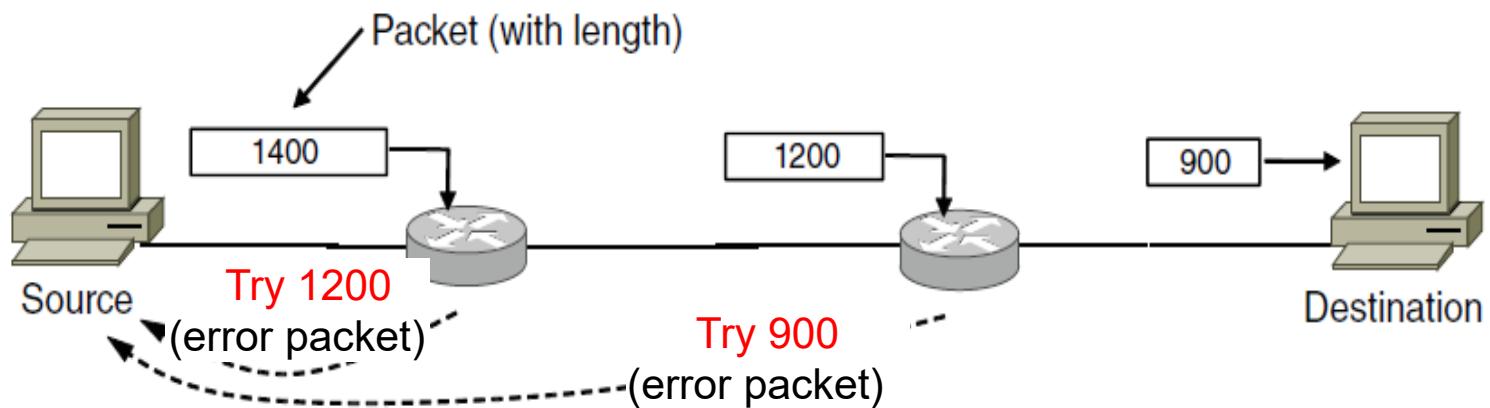


Re-fragmented:
(to 5 bytes)



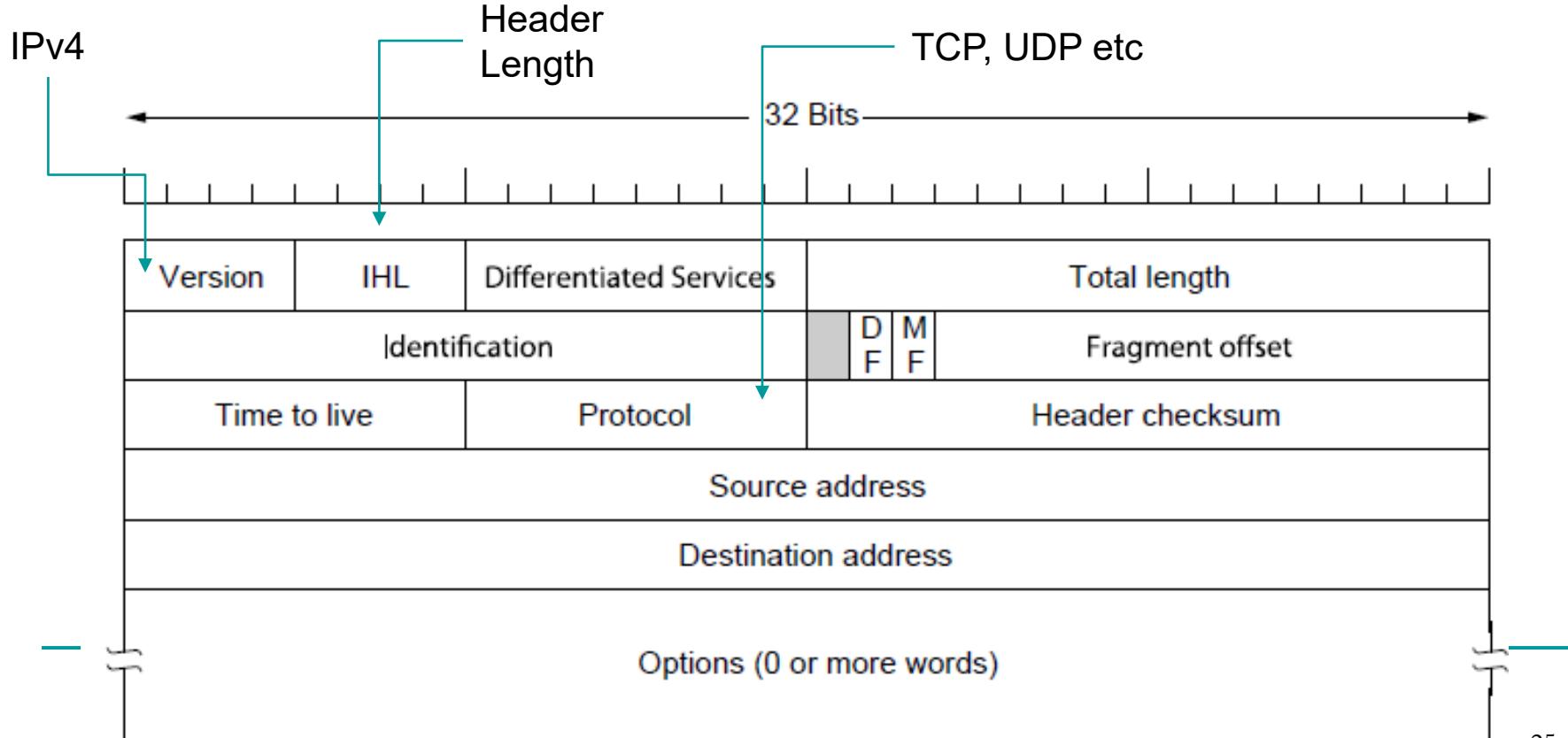
Path MTU Discovery

- Alternative to Fragmentation
- Advantage: the source knows what length packet to send
- If the routes and path MTU change, new error packets will be triggered and the source will adapt to the new path



IPv4 Datagram Structure (1)

- **IPv4** (Internet Protocol) **datagram** consists of a header and payload
- **IPv4 header** is carried on all packets and has fields for the key parts of the protocol
- **Header format:** 20-byte fixed part + variable-length optional part



IPv4 Datagram Structure (2)

- IHL: Internet Header Length, in 32-bit units, min is 5 and max is 15
- Differentiated services: different classes of service
- Total Length: header and payload, max length 65535 bytes
- Identification:
 - Allows host to determine which datagram the new fragment belongs to.
 - All fragments of same datagram have same ID
- DF: Don't Fragment
 - Now it is used as part of the process to discover the path MTU, which is the largest packet that can travel along a path without being fragmented
- MF: More Fragment, is this the last one?
- Fragment offset: where in the datagram the current fragment belongs

IPv4 Datagram Structure (3)

- TTL: Time to live, limits packet lifetimes in hops or seconds
- Protocol: TCP, UDP ...
- Header Checksum: verifies the header only
- Source Address: IP address of the sender
- Destination Address: IP address of the receiver
- Options: e.g. security, strict vs. loose source routing, record route, timestamp

IP Addresses (1)

- IP address (IPv4) is 32-bit long, written in dotted decimal notation

128.18.3.11

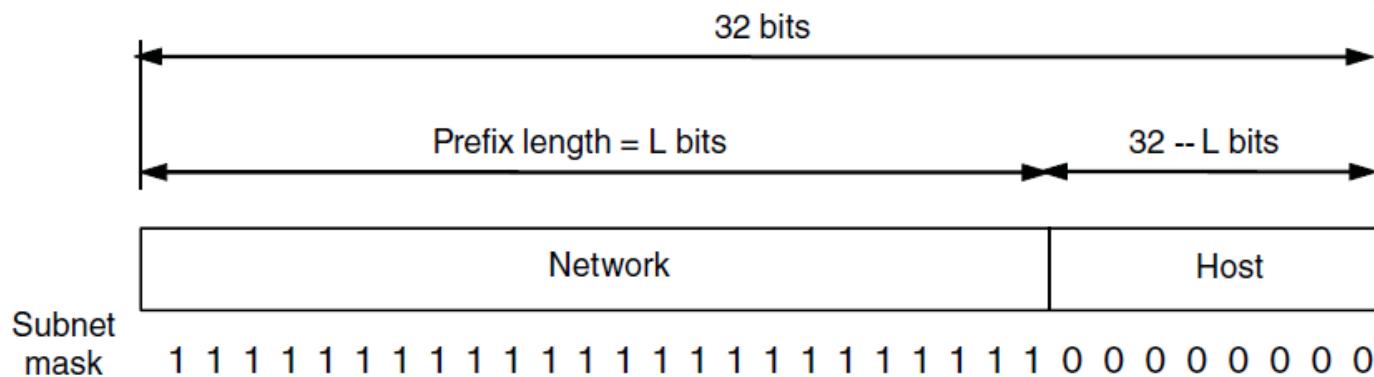
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	0	0	1	0	0	1	0

range: 0-255

- Addresses are hierarchical and can be allocated in blocks
e.g. 256 addresses in the block 128.18.3.0 – 128.18.3.255
- Overall, IP allocation is managed by Internet Corporation for Assigned Names and Numbers (ICANN)

IP Addresses (2)

- network portion + host portion
- **Prefix:** determined by the network portion, all hosts on a single network has the same network portion.
prefix is written as: lowest address/bit-length
 $128.18.3.0/24, 18.2.0.0/16$
- **Subnet mask:** all 1s in the network portion
- **Extract prefix:** ANDed the IP address with the subnet mask



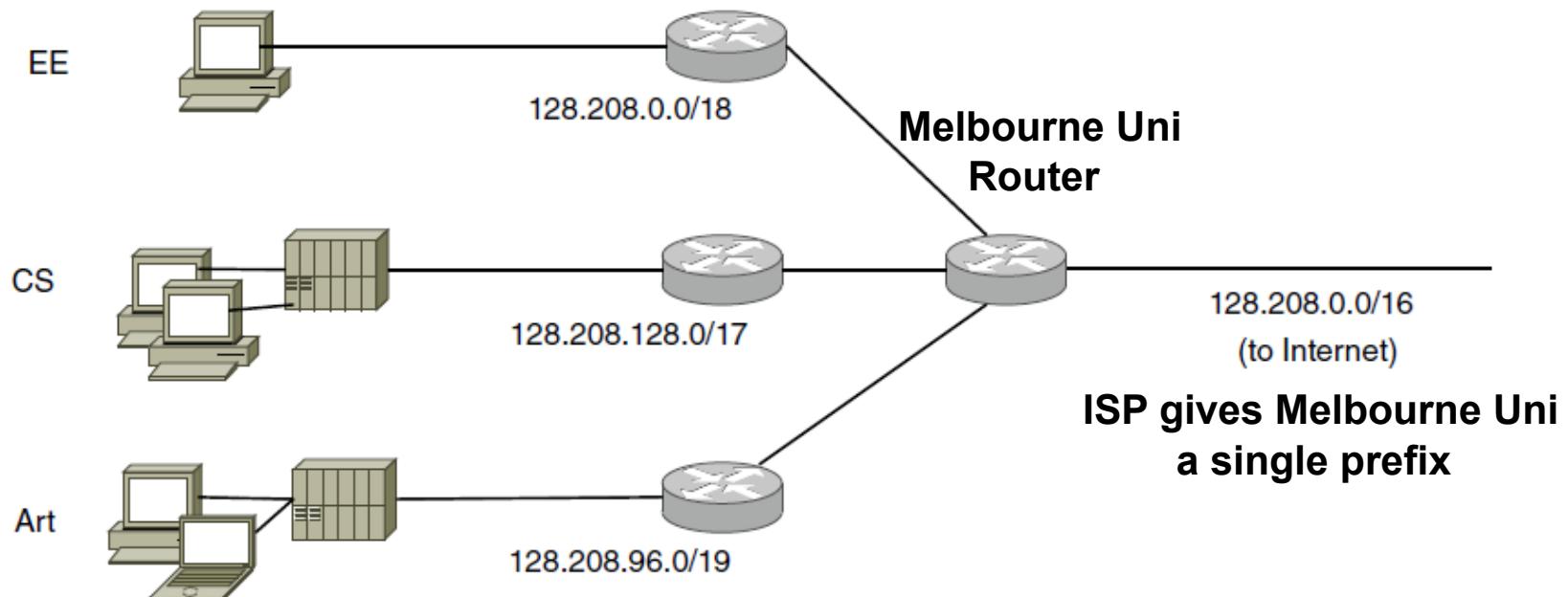
IP Addressing and Routing Tables

- Routing tables are typically built on a triplet:
 - Prefix Address
 - Subnet Mask
 - Outgoing Line (physical or virtual)
- Example: a row of a routing table

Prefix	Subnet Mask	Interface
128.18.3.0/24	255.255.255.0	Eth 0

Subnets (1)

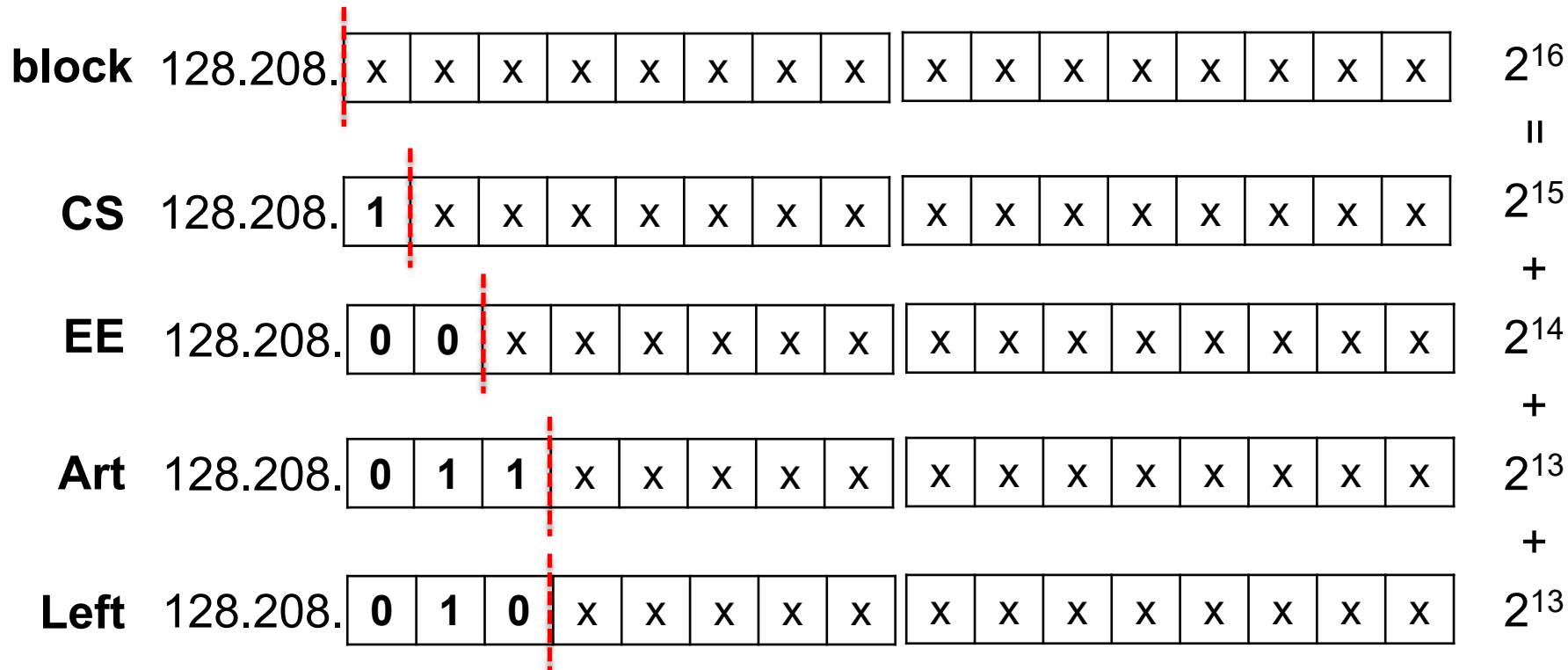
- Subnetting allows networks to be split into several parts for internal uses whilst acting like a single network for external use
- Looks like a single prefix outside the network



Network is divided into subnets internally

Subnets (2)

128.208.0.0/16 → number of addresses 2^{16}



Network Layer

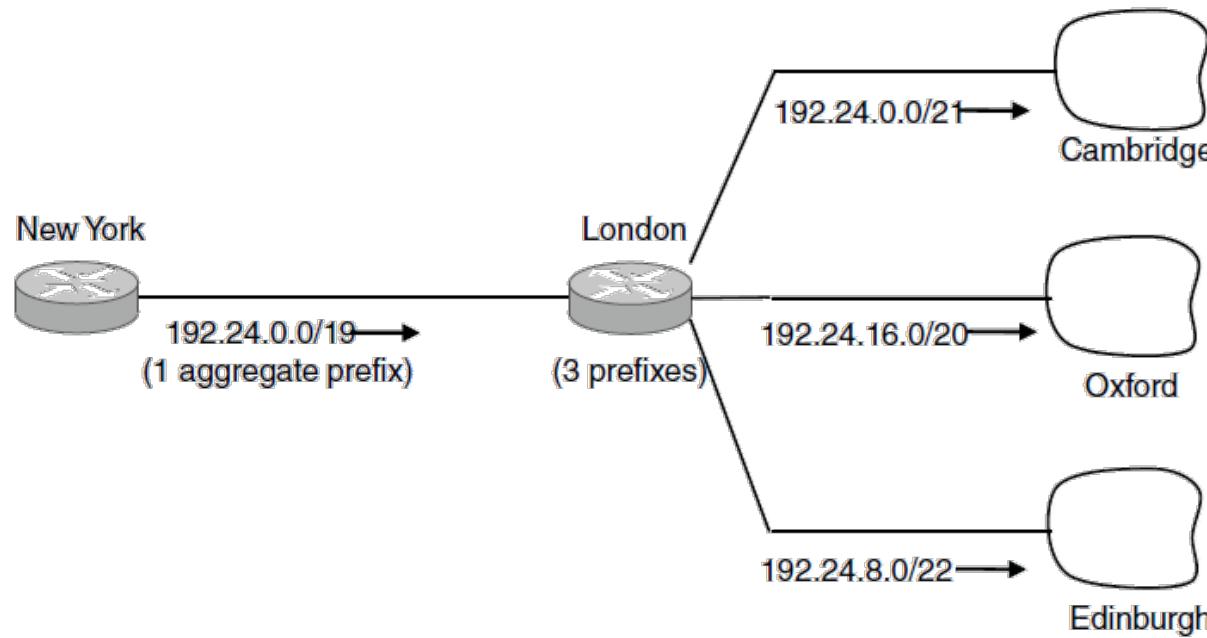
COMP90007 Internet Technologies

Lecturer: Ling Luo

Semester 2, 2020

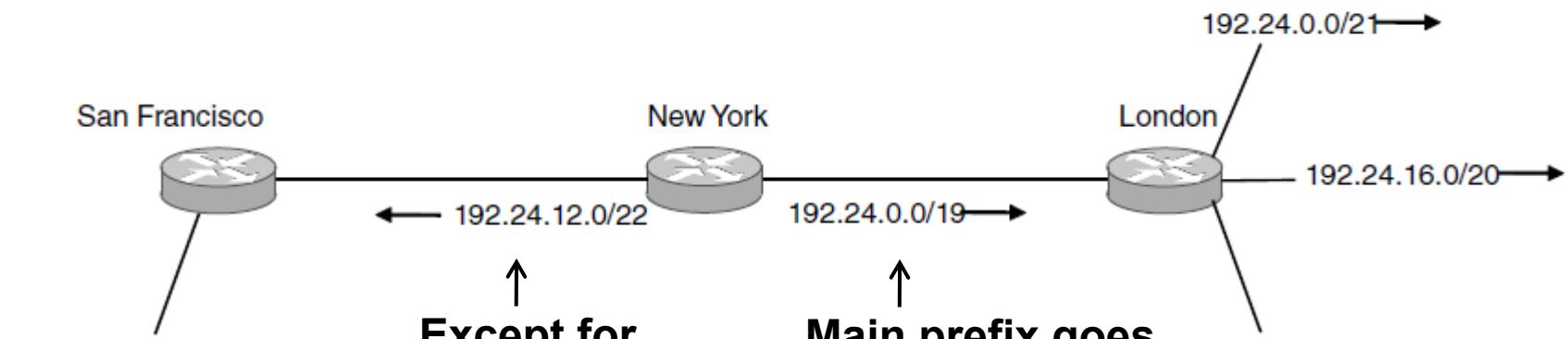
Classless Inter-Domain Routing (1)

- Routing table explosion? Backbone router connecting networks around the world → 300k networks
- Aggregation: process of joining multiple IP prefixes into a single larger prefix to reduce size of routing table



Classless Inter-Domain Routing (2)

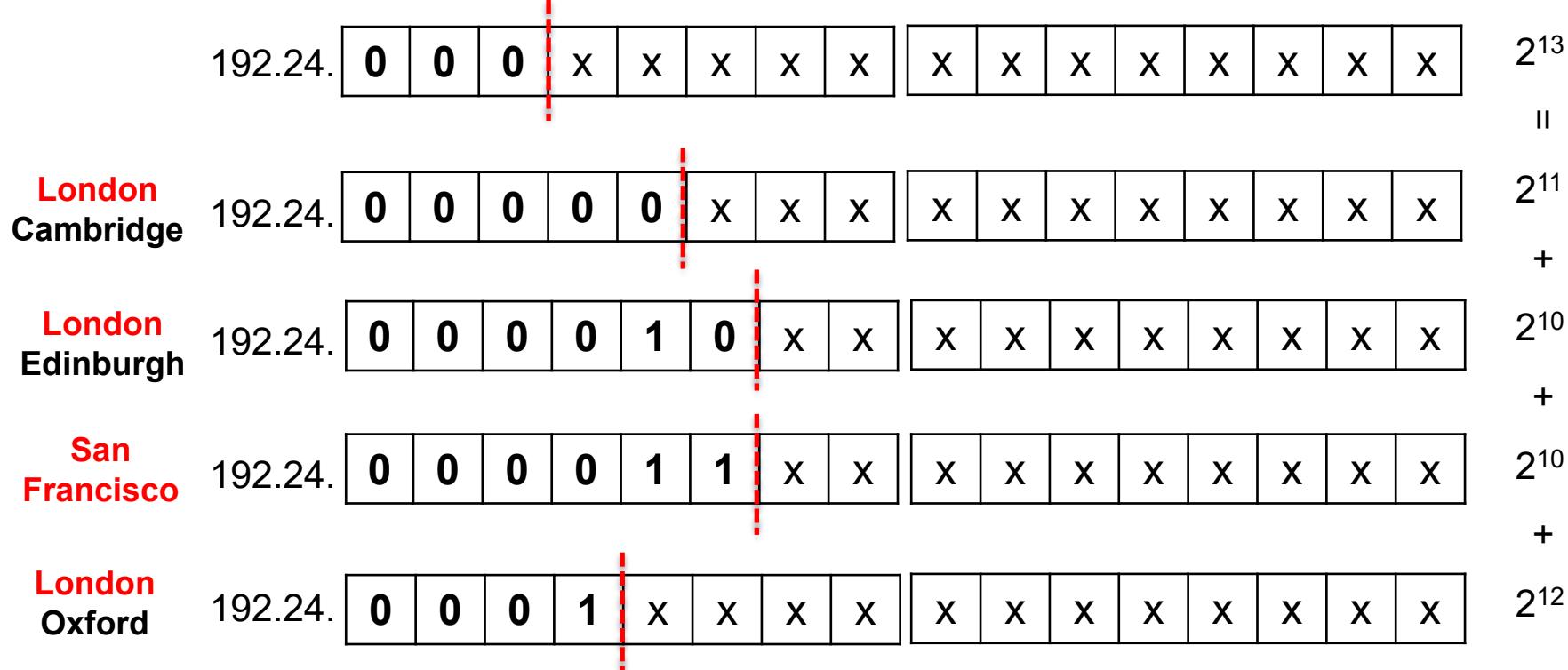
- Packets are forwarded to the entry with the longest matching prefix (i.e. smallest address block)
- Complicates forwarding process but adds flexibility
 - 1) Check address whether matches the longest prefix → /22
 - 2) If not, then see if it matches /19



Prefix Address	Subnet Mask	Interface
192.24.12.0/22	255.255.252.0	Eth 0 (to SF)
192.24.0.0/19	255.255.224.0	Eth 1 (to London)

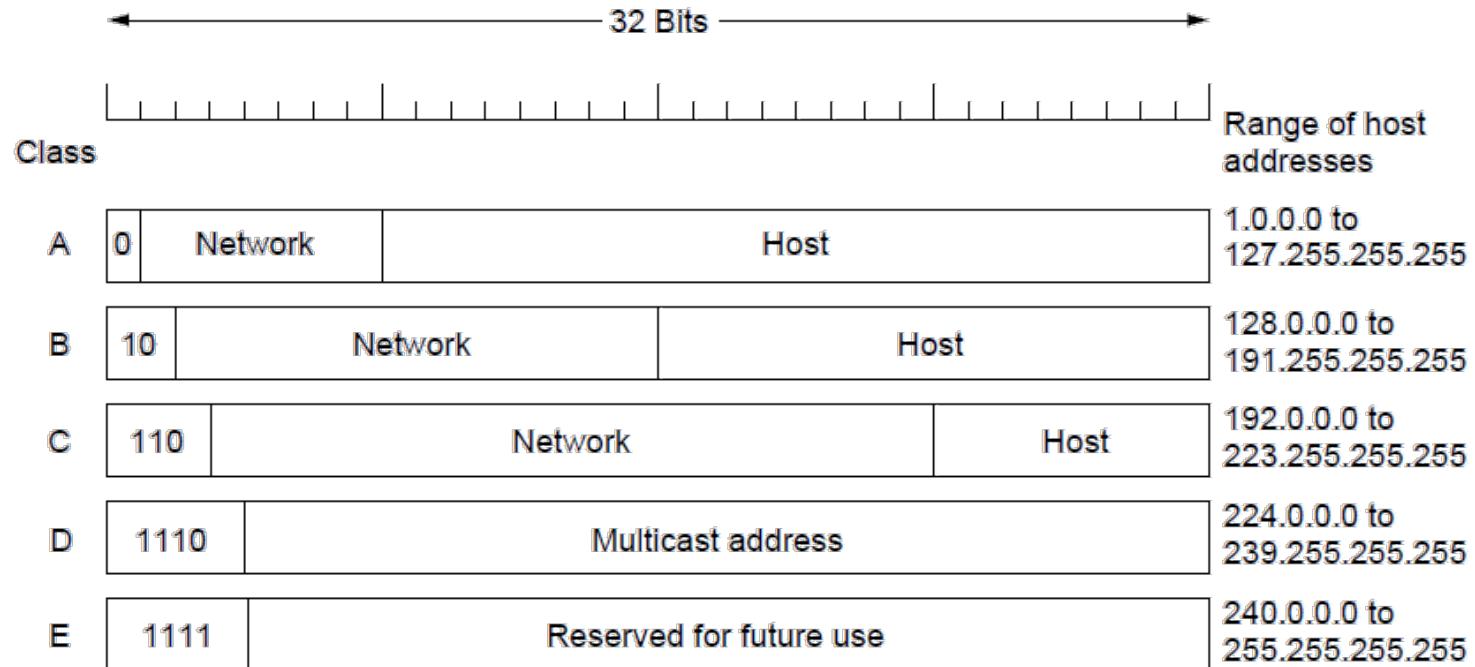
Classless Inter-Domain Routing (3)

192.24.0.0/19 → number of addresses 2^{13} (8192)



Classful Addressing

- Old design: addresses came in blocks of fixed size (Class A, B, C, D, E)
 - Carries size as part of address, but lacks flexibility

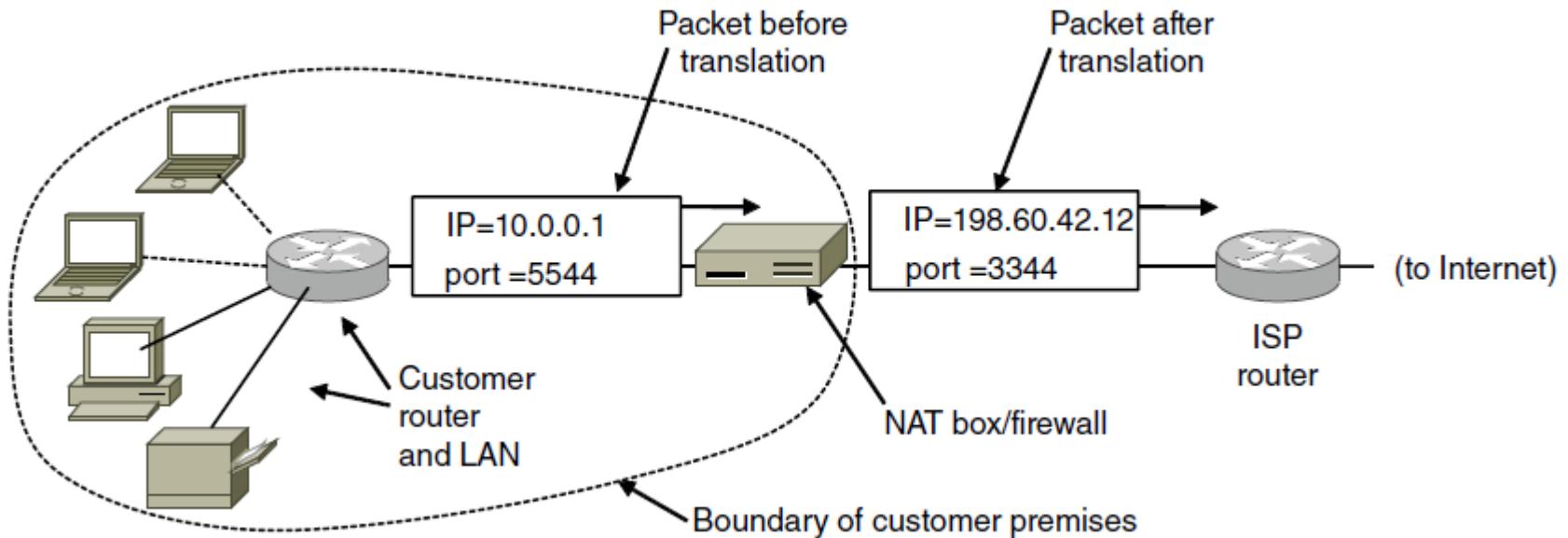


Private IP Ranges

- Range of IP addresses that CANNOT appear in the Internet
- Reserved only for private networks
 - 10.0.0.0/8 ($2^{24} = 16,777,216$ hosts)
 - 172.16.0.0/12 ($2^{20} = 1,048,576$ hosts)
 - 192.168.0.0 /16 ($2^{16} = 65,536$ hosts)

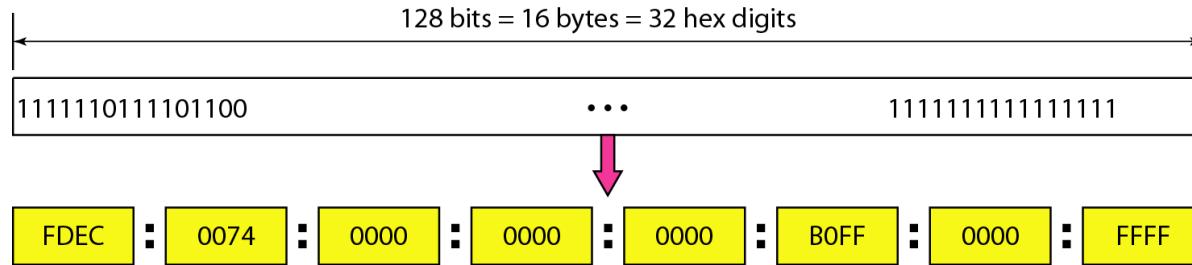
Network Address Translation (NAT)

- NAT box maps one external IP address to many internal IP addresses
 - Uses TCP/UDP port to distinguish connections
 - Violates layering; popular tool in conserving global address space



IPv6 (1)

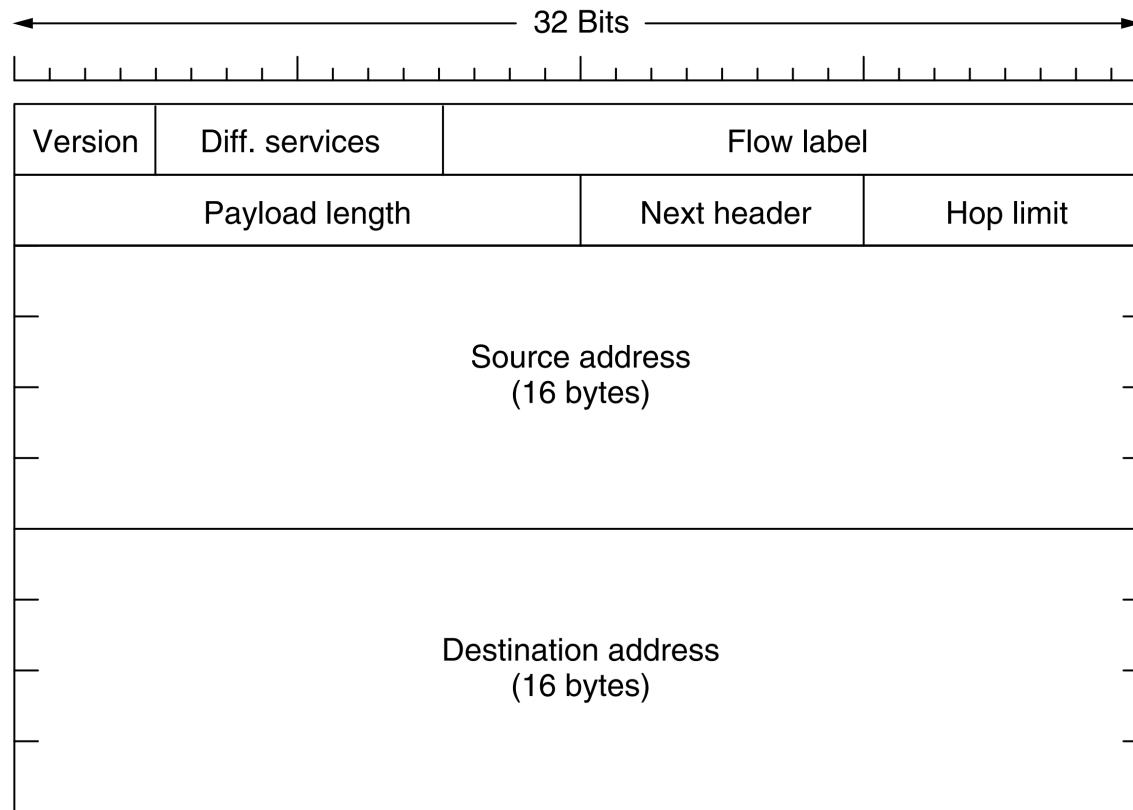
- Larger address space: 128-bit address
use hexadecimal colon notation



- The format of header is simplified: required fields + options
- Support for more security: encryption and authentication
- Transition: dual stack, tunneling

IPv6 (2)

- Required fields in IPv6 header (40 bytes)



Internet Control Protocols

- IP works with the help of several control protocols:
 - ICMP (Internet Control Message Protocol) is a companion to IP that returns error info
 - Required, and used in many ways, e.g., traceroute, ping
 - ARP (Address Resolution Protocol) finds MAC address of a local IP address
 - Host queries an address and the owner replies
 - DHCP (Dynamic Host Control Protocol) assigns a local IP address to a host
 - Gets host started by automatically configuring it
 - Host sends request to server, which grants a lease

ICMP

- Used for testing and monitoring ambient conditions between hosts and routers

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and Echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router

Network Layer

COMP90007 Internet Technologies

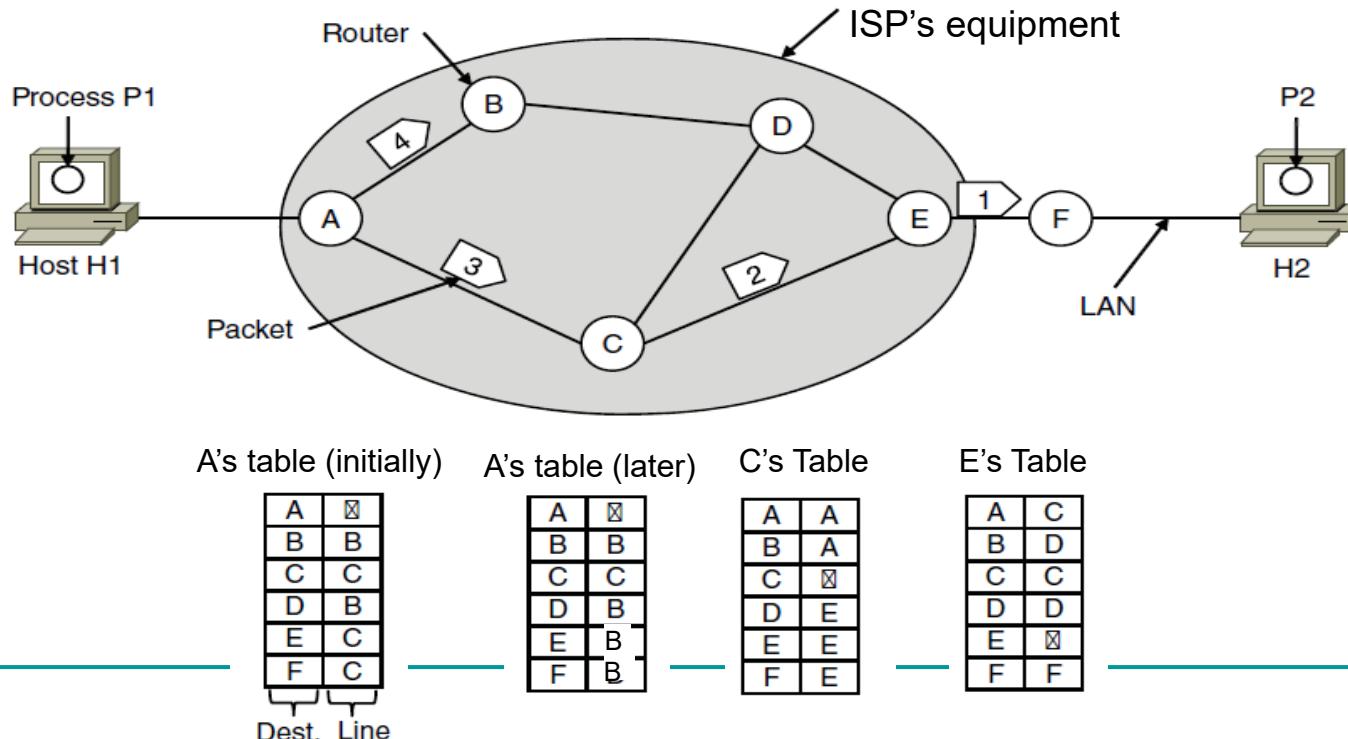
Lecturer: Ling Luo

Semester 2, 2020

Routing

Consider the network as a graph of nodes and links:

- Routing is the process of discovering network paths
- Decide what to optimize: hops, delay, etc.
- Update routes for changes in topology (e.g., failures)



Routing Algorithms (1)

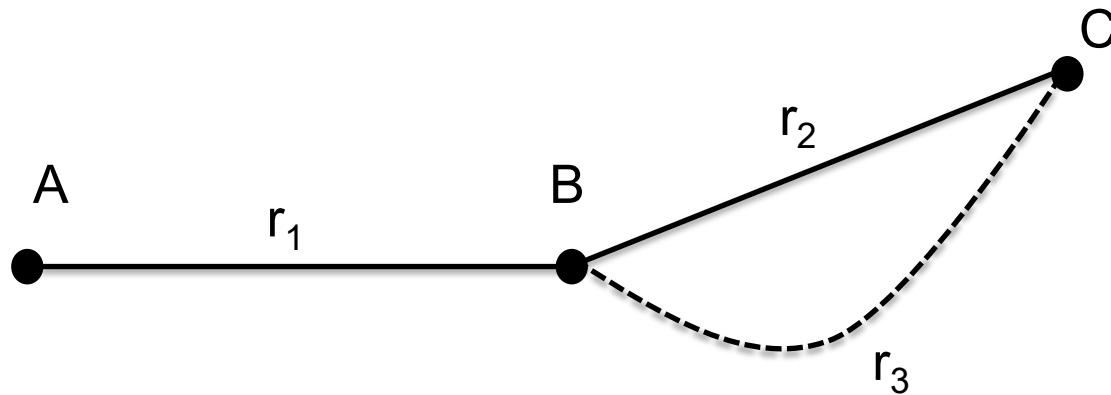
- The routing algorithm is responsible for deciding on which output line an incoming packet should be transmitted
- **Non-Adaptive Algorithms**
 - Static decision-making process (static routing)
- **Adaptive Algorithms**
 - Dynamic decision-making process (dynamic routing)
 - Changes in network topology, traffic, etc.

Routing Algorithms (2)

- Non-adaptive
 - Shortest path routing
 - Flooding
- Adaptive
 - Distance vector routing
 - Link state routing
- Hierarchical routing
- Broadcasting routing
- Multicasting routing

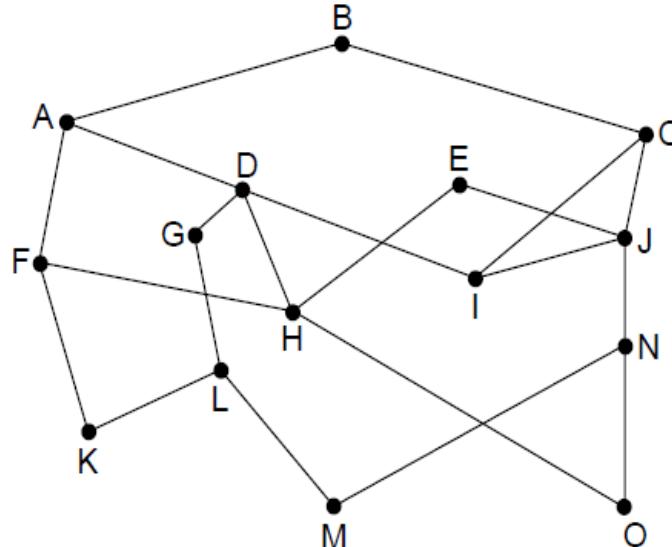
Optimality Principle

- If router B is on the optimal path from router A to router C, then the optimal path from B to C also falls along the same route.

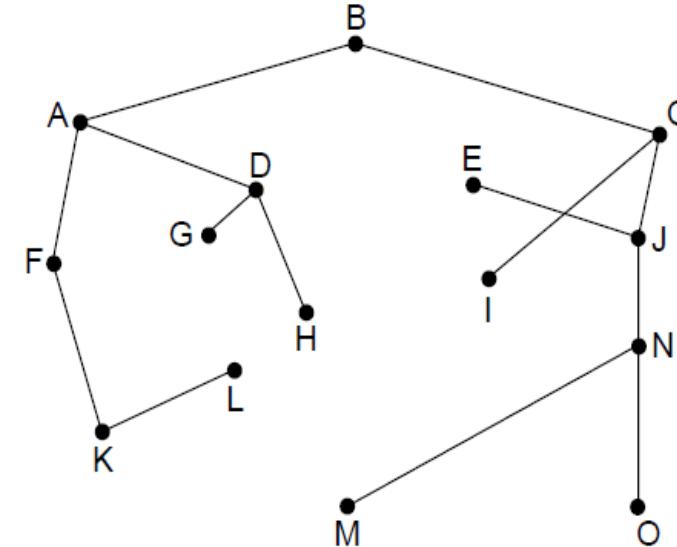


Sink Tree

- **Sink Tree:** the set of optimal routes from all sources to a given destination forms a tree rooted at the destination
- Goal of a routing algorithm: discover and utilise the sink trees for all routers



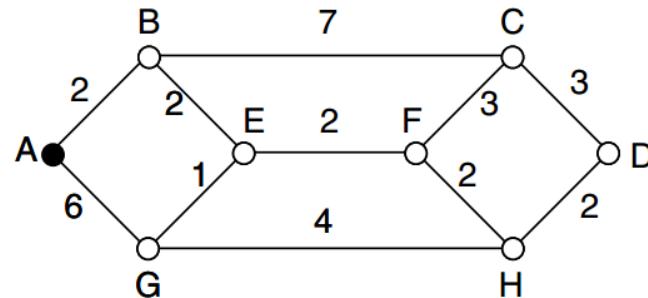
Network



Sink tree of best paths to router B

Shortest Path Routing

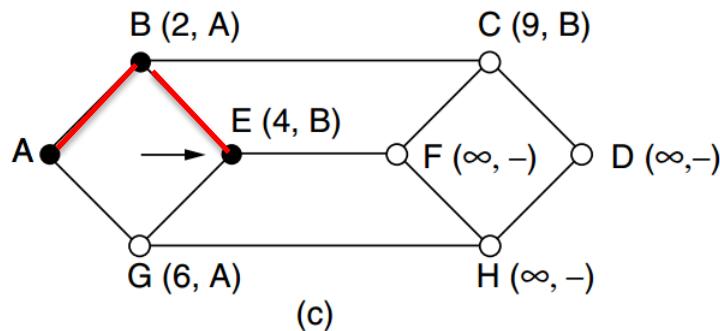
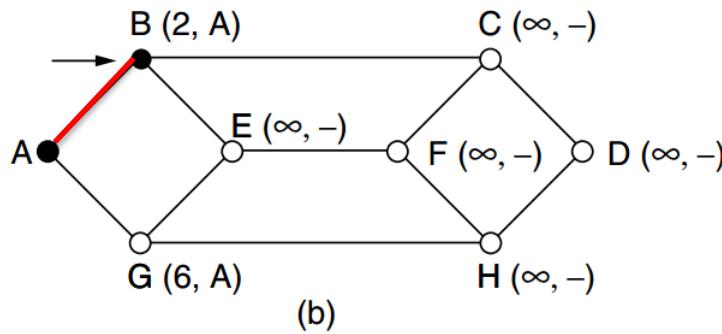
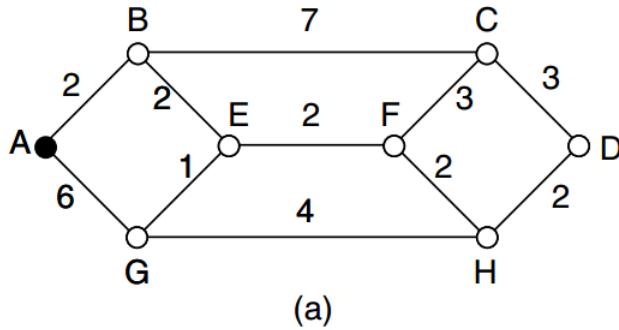
- A non-adaptive algorithm
- Shortest path can be determined by building a graph with each node representing a router, and each arc representing a communication link
- Metrics: number of hops, distance, delay etc.
- To choose a path between 2 routers, the algorithm finds the shortest path between them on the graph



Shortest Path: Dijkstra's Algorithm (1)

- Computes a sink tree on the graph:
 - Each link is assigned a non-negative weight/distance
 - Shortest path is the one with lowest total weight
 - Using weights of 1 gives paths with fewest hops
- Algorithm:
 - 1) Create a set P , **tracking the nodes added in the tree**. Initialize it as empty.
 - 2) For each node, assign a **distance value d from the node to sink**. Initialize the distance for all nodes as infinity.
 - 3) Start from the sink node, assign distance as 0.
 - 4) **Repeat** when P doesn't include all nodes:
 - i. For all the nodes not in P , compare distance d
 - ii. Pick a node v with min distance and add it to P
 - iii. Update d for all the adjacent nodes of v (newly added node)

Shortest Path: Dijkstra's Algorithm (2)



Distance to A

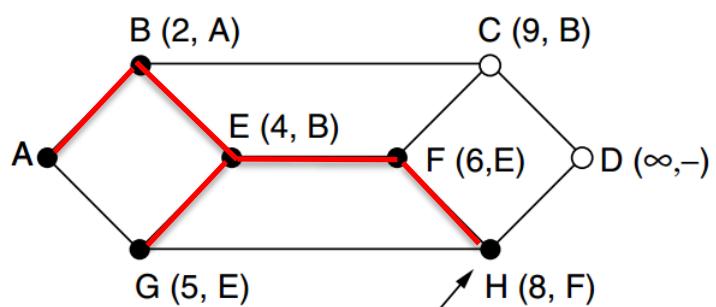
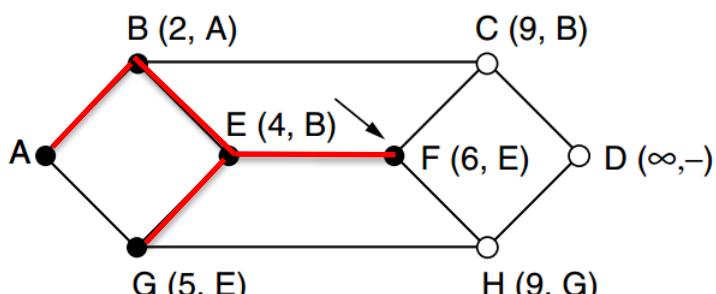
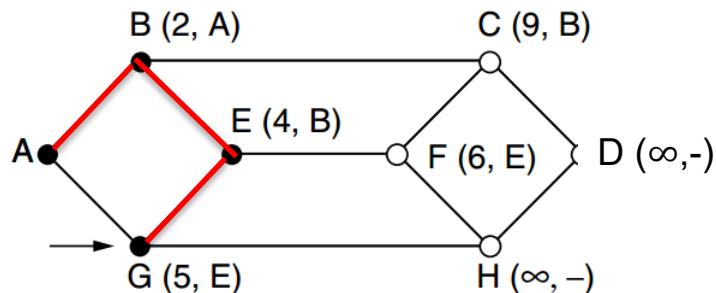
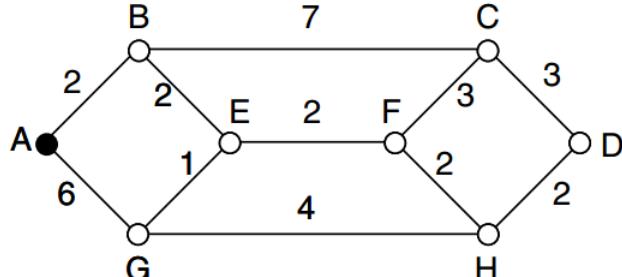
n	A	B	C	D	E	F	G	H
1	0	∞	∞	∞	∞	∞	∞	∞
2	--	2	∞	∞	∞	∞	6	∞
3	--	--	9	∞	4	∞	6	∞

Set P

{A}

{A, B}

{A, B, E}



Distance to A

Set P

n	A	B	C	D	E	F	G	H	
1	0	∞	∞	∞	∞	∞	∞	∞	{A}
2	--	2	∞	∞	∞	∞	6	∞	{A, B}
3	--	--	9	∞	4	∞	6	∞	{A, B, E}
4	--	--	9	∞	--	6	5	∞	{A, B, E, G}
5	--	--	9	∞	--	6	--	9	{A, B, E, G, F}
6	--	--	9	∞	--	--	--	8	{A, B, E, G, F, H}
7	--	--	9	10	--	--	--	--	{A, B, E, G, F, H, C}
8	--	--	--	10	--	--	--	--	{A, B, E, G, F, H, C, D}

...

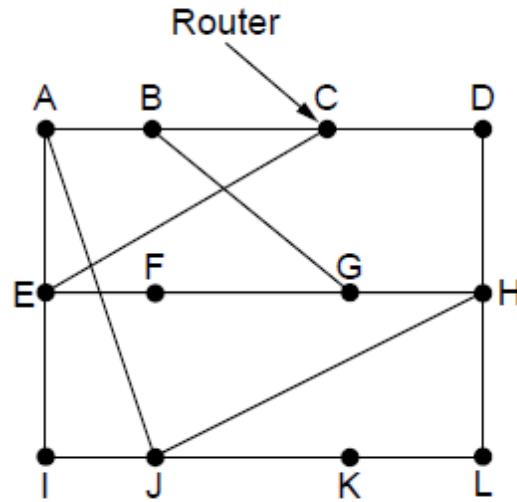
Flooding

- A non-adaptive algorithm
- Every incoming packet is sent out on **every outgoing line except the one on which it arrived**
- Inefficient: generates a large number of duplicate packets
- Selective flooding is an improved variation
 - Routers send packets only on links which are approximately in the right direction

Distance Vector Routing (1)

- A dynamic algorithm
 - Each router maintains a table which includes the best known distance to each destination and which line to use to get there
 - Tables are updated by exchanging information with neighboring routers
 - Global information shared locally
- Algorithm:
 - 1) Each node knows distance of links to its neighbors
 - 2) Each node **advertises** vector of lowest known distances to **all neighbors**
 - 3) Each node uses received vectors to **update** its own
 - 4) Repeat periodically

Distance Vector Routing (2)



Network

To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	-
K	24	22	22	0	6	K
L	29	33	9	9	15	K

JA delay is 8 JI delay is 10 JH delay is 12 JK delay is 6

New vector for J

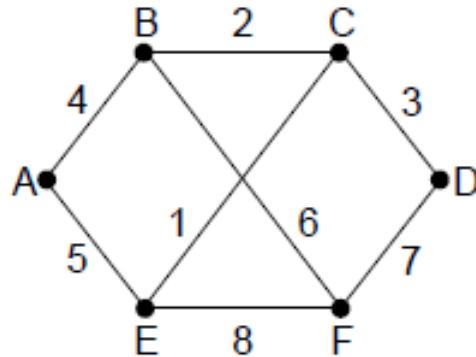
Vectors received at J from neighbors A, I, H and K

Link State Routing

- A dynamic algorithm
 - An alternative to distance vector: **too long to converge** after the network topology changed
 - Widely used in the Internet, e.g. Open Shortest Path First (OSPF)
 - More computation but simpler dynamics
 - Local information shared globally, using flooding
- Algorithm: each router has to
 - 1) Discover neighbors and learn network addresses
 - 2) Measure delay or cost to each neighbor
 - 3) **Build link state packet**
 - 4) Send this packet to **all other routers**
 - 5) **Compute the shortest path** to every other router, e.g. using Dijkstra's algorithm

Building Link State Packets

- Link State Packet (LSP) for a node lists neighbors and weights of links to reach them



Network

	Link	State	packets	
A	B Seq. Age B 4 E 5	C Seq. Age B 2 D 3 F 6 E 1	D Seq. Age A 5 C 1 F 8	F Seq. Age B 6 D 7 E 8
B				
C				
D				
E				
F				

LSP for all nodes

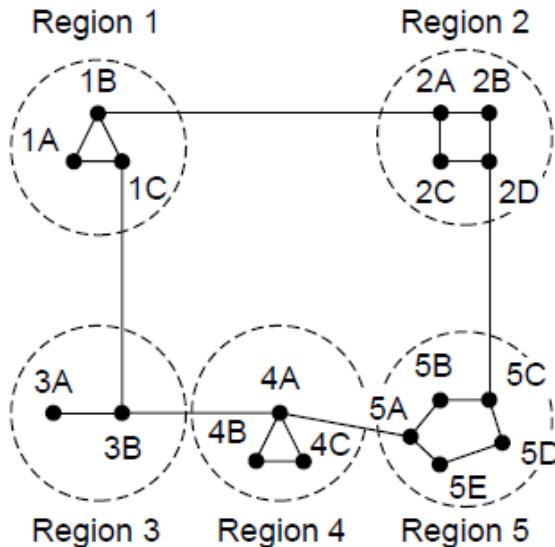
- When to build LSP?
 - Periodically at regular intervals
 - Build them when some significant event occurs, e.g. a line or neighbor going down or coming back up again, or changing its properties considerably

Hierarchical Routing (1)

- As networks grow in size, routing tables expand and this impacts CPU and memory requirements
- Dividing all routers into regions increases efficiencies
 - Each router knows everything about other routers in its region but **nothing about routers in other regions**
 - Routers which connect to two regions act as exchange points for routing decisions

Hierarchical Routing (2)

- Hierarchical routing reduces the work of computation but may result in slightly longer paths than flat routing



Full table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

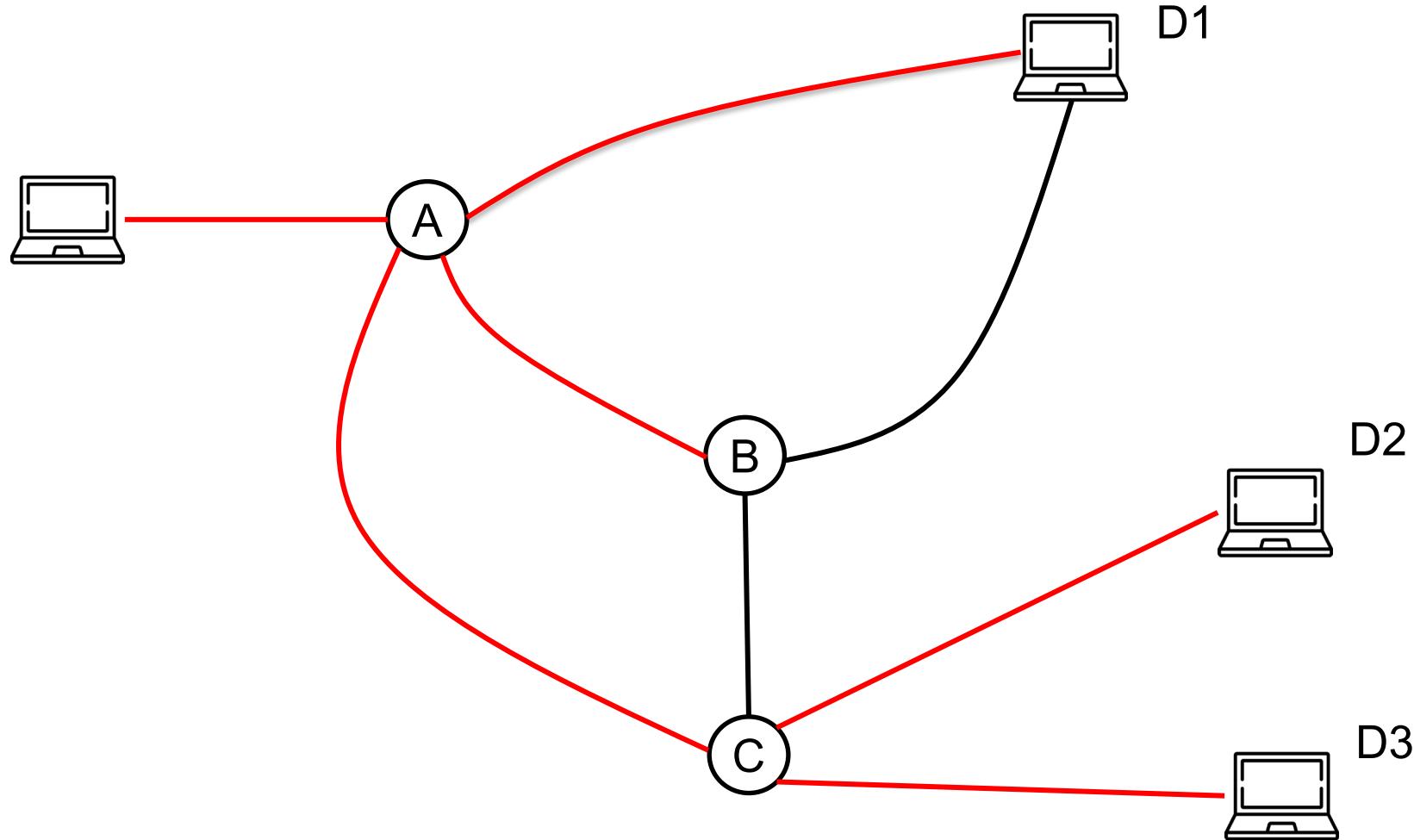
Hierarchical table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Broadcast Routing (1)

- Broadcast routing allows hosts to send messages to all other hosts.
 - Single distinct packet to each destination: inefficient, and source needs all destination addresses
 - Multi-destination routing: a router copies the packet for each outgoing line. Use bandwidth more efficiently, but source needs to know all the destination addresses
 - Flooding
 - Reverse path forwarding

Broadcast Routing (2)



Broadcast Routing (3)

- Reverse path forwarding

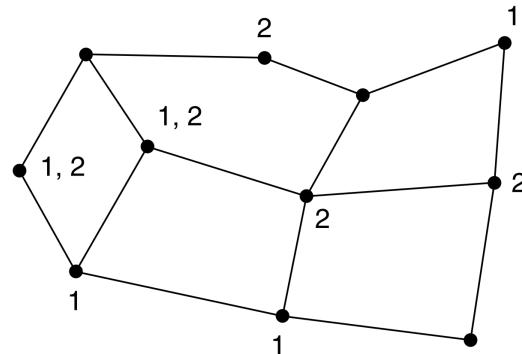
The router checks if the broadcast packet arrived on the line normally used for sending packets to the source of the broadcast:

- Yes**: there is a **high probability** that the route used to transmit this packet is **the best**, and this packet is the **first copy**. The router then copies the packet and forwards them onto all other lines.
 - No**: the packet is **discarded as a likely duplicate**.

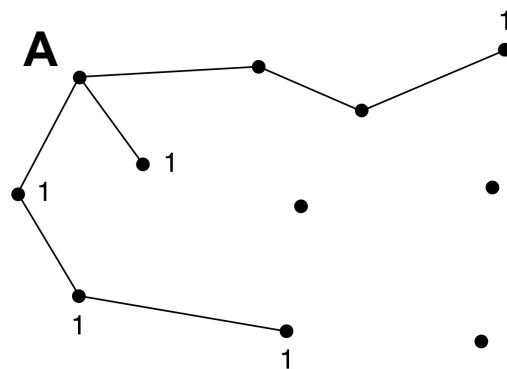
Multicast Routing (1)

- Multicast routing allows hosts to send a message to a well-defined group within the whole network
- Each router computes a spanning tree covering all other routers
 - Spanning tree: subset of the graph that includes all nodes, but no loops.
 - Prunes the spanning tree to eliminate all lines which do not lead to members of the group

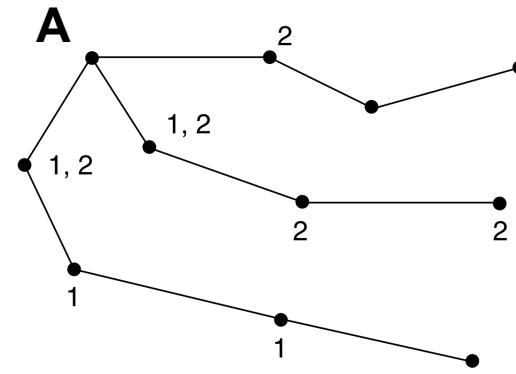
Multicast Routing (2)



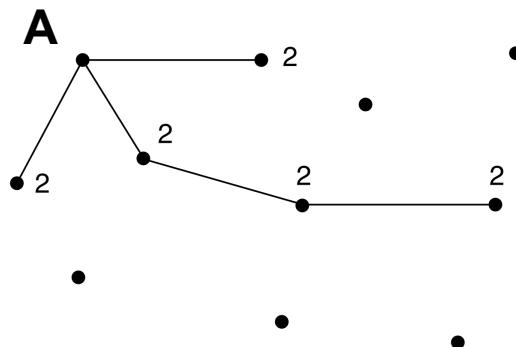
network



multicast tree for Group 1



spanning tree for router A



multicast tree for Group 2

COMP90007 Internet Technologies

Week 6 Workshop

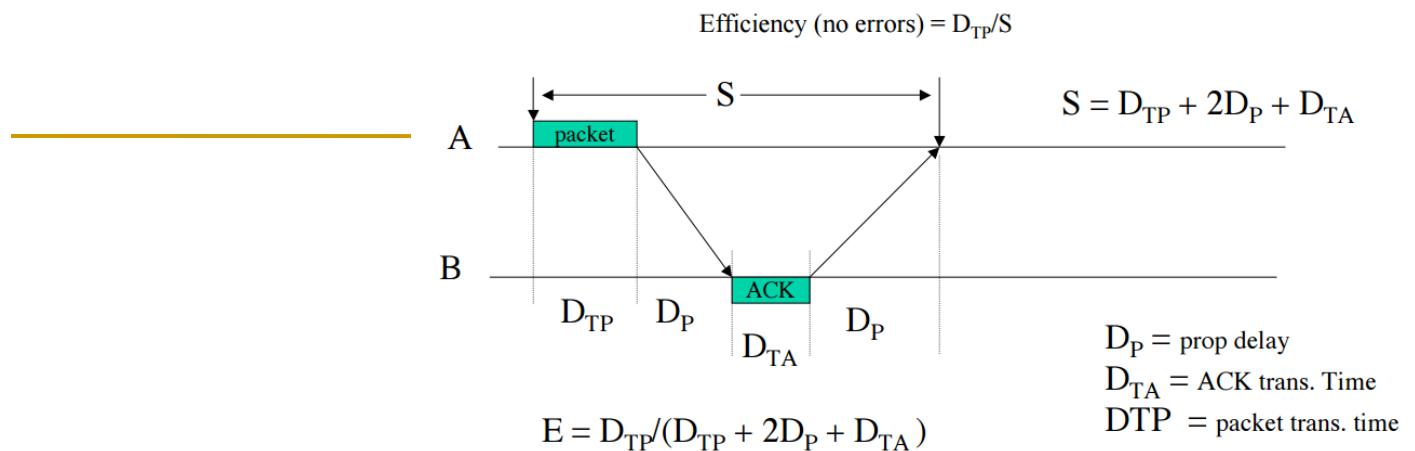
Semester 2, 2020

Question 1

Using the polynomial code method, compute the CRC for the frame: 1101011111 having a generator polynomial $G(x)$ as $x^4 + x + 1$.

Question 2

A channel has a bit rate of 4 kbps and a propagation delay of 20 ms. For what range of frame sizes does stop-and-wait give an efficiency of at least 50 percent?

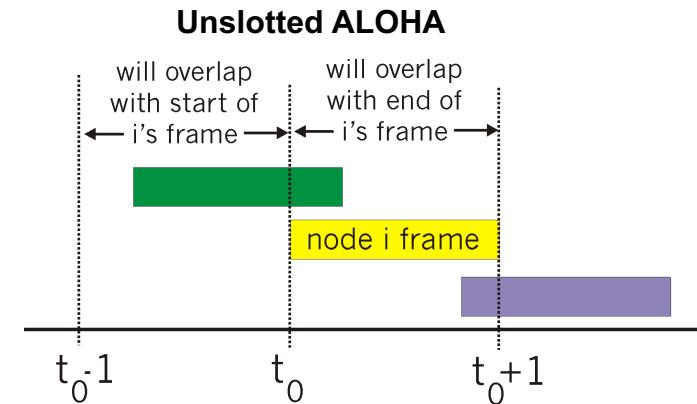
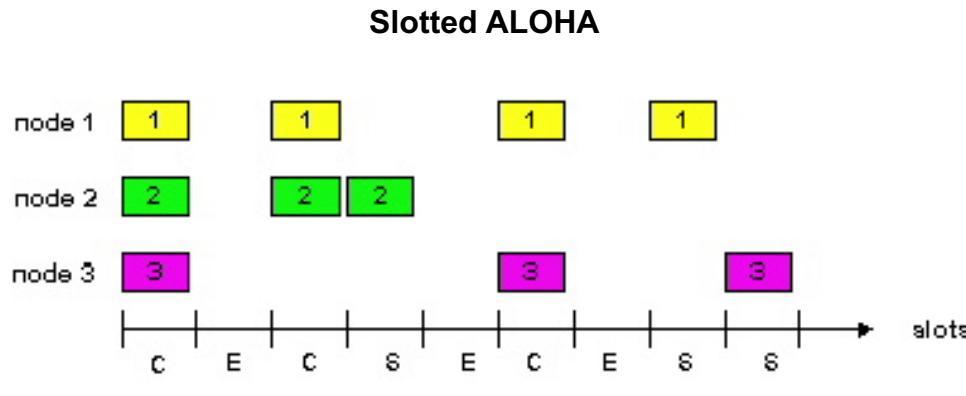


Question 3

Why would anyone like to use the Go-Back-N protocol if we already introduced a superior protocol that can repeat only the missing frames, i.e., the Selective Repeat protocol?

Question 4

Consider the delay of pure ALOHA versus slotted ALOHA at low load. Which one is less? Explain your answer.



Question 5

For medium access control one can use dynamic allocation of channels in comparison to static allocation. Dynamic allocation is far more adaptive. Thus, why would anyone use static allocation mechanisms?

COMP90007 Internet Technologies

Week 6 Workshop

Semester 2, 2020

Suggested solutions

Question 1

Using the polynomial code method, compute the CRC for the frame: 1101011111 having a generator polynomial G(x) as $x^4 + x + 1$

Frame:	1 1 0 1 0 1 1 1 1 1	
Generator:	1 0 0 1 1	
1 0 0 1 1	/	1 1 0 0 0 0 1 1 1 0 ← Quotient (thrown away)
		1 0 0 0 0 0 ← Frame with four zeros appended
		1 0 0 1 1 ↓
		1 0 0 1 1 ↓
		0 0 0 0 1 ↓
		0 0 0 0 0 ↓
		0 0 0 1 1 ↓
		0 0 0 0 0 ↓
		0 0 1 1 1 ↓
		0 0 0 0 0 ↓
		0 1 1 1 1 ↓
		0 0 0 0 0 ↓
		1 1 1 1 0 ↓
		1 0 0 1 1 ↓
		1 1 0 1 0 ↓
		1 0 0 1 1 ↓
		1 0 0 1 0 ↓
		0 0 0 1 0 ↓
		0 0 0 0 0 ↓
		1 0 ← Remainder
Transmitted frame:	1 1 0 1 0 1 1 1 1 1 0 0 1 0	← Frame with four zeros appended minus remainder

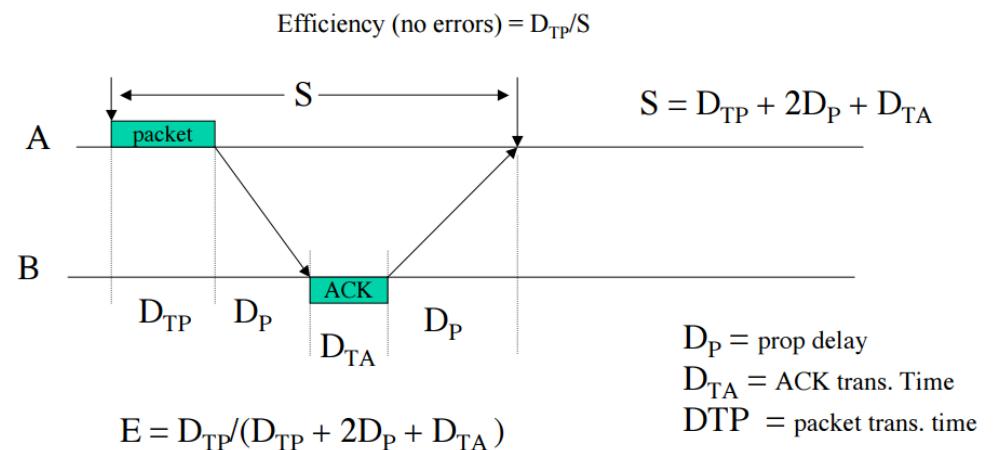
Question 2

A channel has a bit rate of 4 kbps and a propagation delay of 20 ms. For what range of frame sizes does stop-and-wait give an efficiency of at least 50 percent?

Answer:

Efficiency will be 50% when the time to transmit the frame equals the round trip propagation delay.

At a transmission rate of 4 kbps, 40 ms will transfer 160 bits. For frame sizes greater than 160 bits, stop-and-wait is reasonably efficient.



Question 3

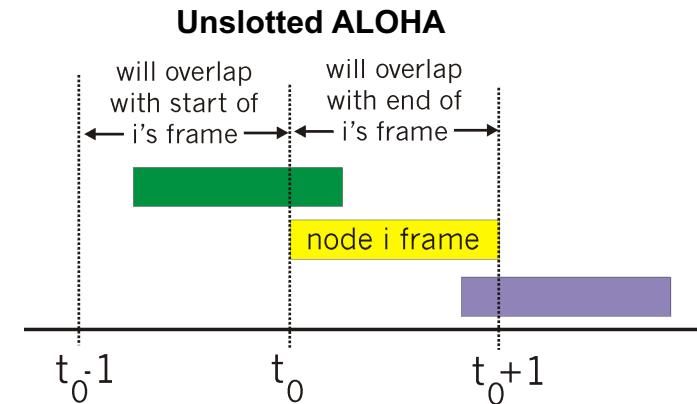
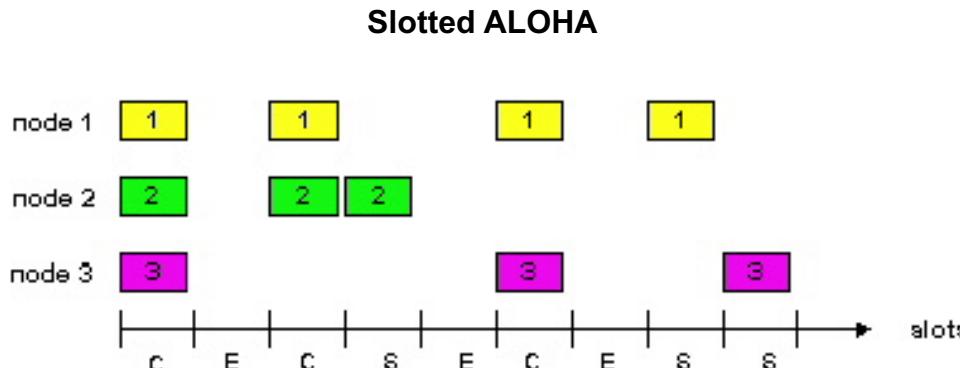
Why would anyone like to use the Go-Back-N protocol if we already introduced a superior protocol that can repeat only the missing frames, i.e., the Selective Repeat protocol?

Answer:

This is a standard case of speed vs memory in computing. Yes, Selective Repeat would be fast in recovering frames as the receiver does not throw away frames that come out of sequence but this comes with the cost that the receiver now has to have a larger than single frame size as its buffer, i.e. more memory needed.

Question 4

Consider the delay of pure ALOHA versus slotted ALOHA at low load. Which one is less? Explain your answer.



Answer:

With slotted ALOHA, it has to wait for the next slot. This introduces half a slot time of delay. With pure ALOHA, transmission can start instantly. At low load with minimal collisions, pure ALOHA will have less delay.

However, at higher loads, there is more probability for collisions in pure ALOHA compared to slotted ALOHA. This is because frames can collide in midway. By enforcing synchronisation, slotted ALOHA is able to achieve much greater efficiency.

Question 5

For medium access control one can use dynamic allocation of channels in comparison to static allocation. Dynamic allocation is far more adaptive. Thus, why would anyone use static allocation mechanisms?

Ans. Static allocation is still useful when the number of senders are known and fairly stable. In such a case, one does not need to deal with collision resolution etc through complex algorithms. Especially if all senders are in need of the channel regularly, why would we bother trying to allocate channels dynamically. A good example is FM radio where all channels are regularly used and fairly stable in terms of number of them and a fair static allocation would suffice.

COMP90007 Internet Technologies

Week 7 Workshop

Semester 2, 2020

Suggested solutions

Question 1

What are the benefits and disadvantages of Transparent fragmentation in Network Layer?

Ans. Good design paradigm and encapsulation of fragmentation within each network. Transparent fragmentation is straightforward to implement and use but has problems. For one thing, the exit router must know when it has received all the pieces, so either a count field or an “end of packet” bit must be provided. Also, because all packets must exit via the same router so that they can be reassembled, the routes are constrained. By not allowing some fragments to follow one route to the ultimate destination and other fragments a disjoint route, some performance may be lost. More significant is the amount of work that the router may have to do. It may need to buffer the fragments as they arrive, and decide when to throw them away if not all of the fragments arrive. Some of this work may be wasteful, too, as the packet may pass through a series of small packet networks and need to be repeatedly fragmented and reassembled.

Question 2

Convert the IP address 11000001, 01010010, 11010010, 00001111 to dotted decimal notation.

Ans. 193.82.210.15

Question 3

Convert the IP address 240.68.10.10 to binary format

Use the following key:

10000000	2^7	128
01000000	2^6	64
00100000	2^5	32
00010000	2^4	16
00001000	2^3	8
00000100	2^2	4
00000010	2^1	2
00000001	2^0	1

Ans. 1111 0000 . 0100 0100 . 0000 1010 . 0000 1010

Question 4

A network on the Internet has a subnet mask of 255.255.240.0. What is the maximum number of hosts that it can handle?

Answer:

255.255.240.0 in binary is 11111111 11111111 11111111 11110000 00000000

The mask is 20 bits long, so the network part is 20 bits. The remaining 12 bits are for the host, so 4096 host addresses exist.

Question 5

IPv6 uses 16 bytes addresses. If a block of 1 million addresses is allocated every picosecond, how long will the addresses last?

Answer:

With 16 bytes there are 2^{128} or addresses. If we allocate them at a rate of $10^6 / 10^{-12} = 10^{18}$ addresses per second. Therefore it will take 3.4×10^{20} seconds to run out of IP addresses, which is about 10^{13} years.

This number is 1000 times the age of the universe. Of course, the address space is not flat, so they are not allocated linearly, but this calculation shows that even with an allocation scheme that has an efficiency of 1/1000 (0.1 percent), one will never run out.

Question 6

A router adds an entry in its table that can be represented with mask as 135.46.56.0/21. What is the maximum number of hosts that this network can represent?

Ans. 21 bits means network has 21 bits reserved, and remaining 11 bits are for hosts.

Hence maximum number of hosts is $2^{11} = 2048$

Question 7

If there are n independent paths between two nodes in a network, and the probability that an individual path is working is p , what is the probability of these two nodes being connected? Assume path failures are independent.

Hint: first try to calculate what is the probability that all paths have failed

Answer:

$$\begin{aligned} \text{Pr(nodes connected)} &= 1 - \text{Pr(no connection)} \\ &= 1 - \text{Pr(all paths failed)} \\ &= 1 - \text{Pr(individual path failure)}^n && (\text{assuming independent events}) \\ &= 1 - [1 - \text{Pr(individual path working)}]^n \\ &= 1 - (1 - p)^n \end{aligned}$$

COMP90007 Internet Technologies

Week 8 Workshop

Semester 2, 2020

Suggested solutions

Question 1

A router has just received the following IP addresses:
57.6.96.0/21, 57.6.104.0/21, 57.6.112.0/21 and
57.6.120.0/21. If all of them use the same outgoing line,
can they be aggregated? If so, to what? If not, why not?

Answer:

They can be aggregated to 57.6.96.0/19

Question 2

A router has the following entries in its routing table:

<u>Prefix</u>	<u>Next hop</u>
151.46.184.0/22	Interface 0
151.46.188.0/22	Interface 1
151.53.40.0/23	Router 1
default	Router 2

For each of the following IP addresses, what does the router do if a packet with that address arrives?

(a) 151.46.191.10

⇒ Interface 1

(b) 151.46.187.2

⇒ Interface 0

Question 3

Why do we need routing algorithms in the Network layer?
What are the key categories of routing algorithms?

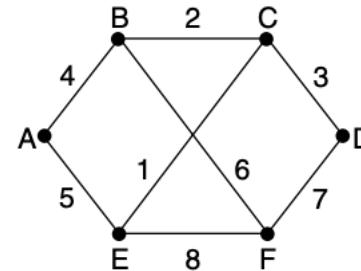
Answer: Routing algos are needed to help decide on which output line an incoming packet should be transmitted.

Key Categories:

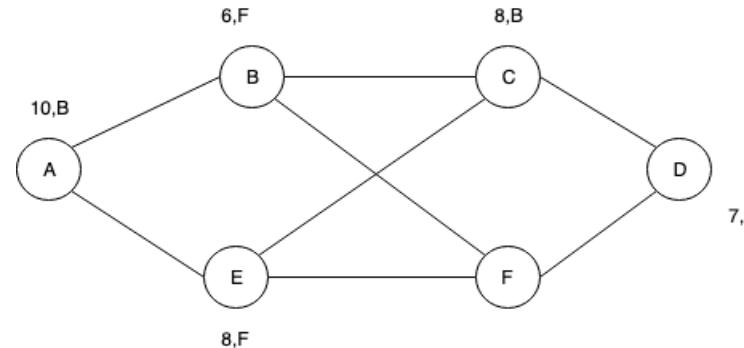
- Non-Adaptive Algorithms
- Adaptive Algorithms

Question 4

Compute the sink tree for Node F in the graph below:



Ans. Refer to Dijkstra's algorithm on the Slides 49-51 of Network Layer



Question 5

Distance vector routing is used for the diagram shown below, and the following vectors have just come in to router C: from B: (5, 0, 8, 12, 6, 2); from D: (16, 12, 6, 0, 9, 10); and from E: (7, 6, 3, 9, 0, 4). The cost of the links from C to B, D, and E, are 6, 3, and 5, respectively. What is C's new routing table? Give both the outgoing line to use and the expected delay.

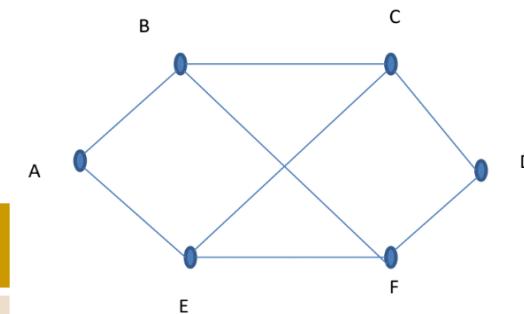
Answer: Using the delays 6, 3, and 5 for B, D, and E, the vectors will be written as:

All Routers	Via B	Via D	Via E
A	11	19	12
B	6	15	11
C	14	9	8
D	18	3	14
E	12	12	5
F	8	13	9

28/9/20

All Routers	Outgoing Line	Expected Delay
A	B	11
B	B	6
C	-	0
D	D	3
E	E	5
F	B	8

© University of Melbourne 2020



Week 9: Application Layer

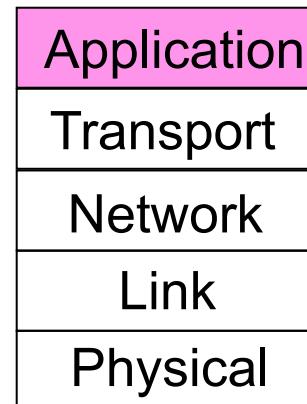
Internet Technologies COMP90007

Lecturer: Muhammad Usman

Semester 2, 2020

The Last Layer in Hybrid Stack

- **Application Layer**
- We will look at key implementations to study this: **Domain Name System** first

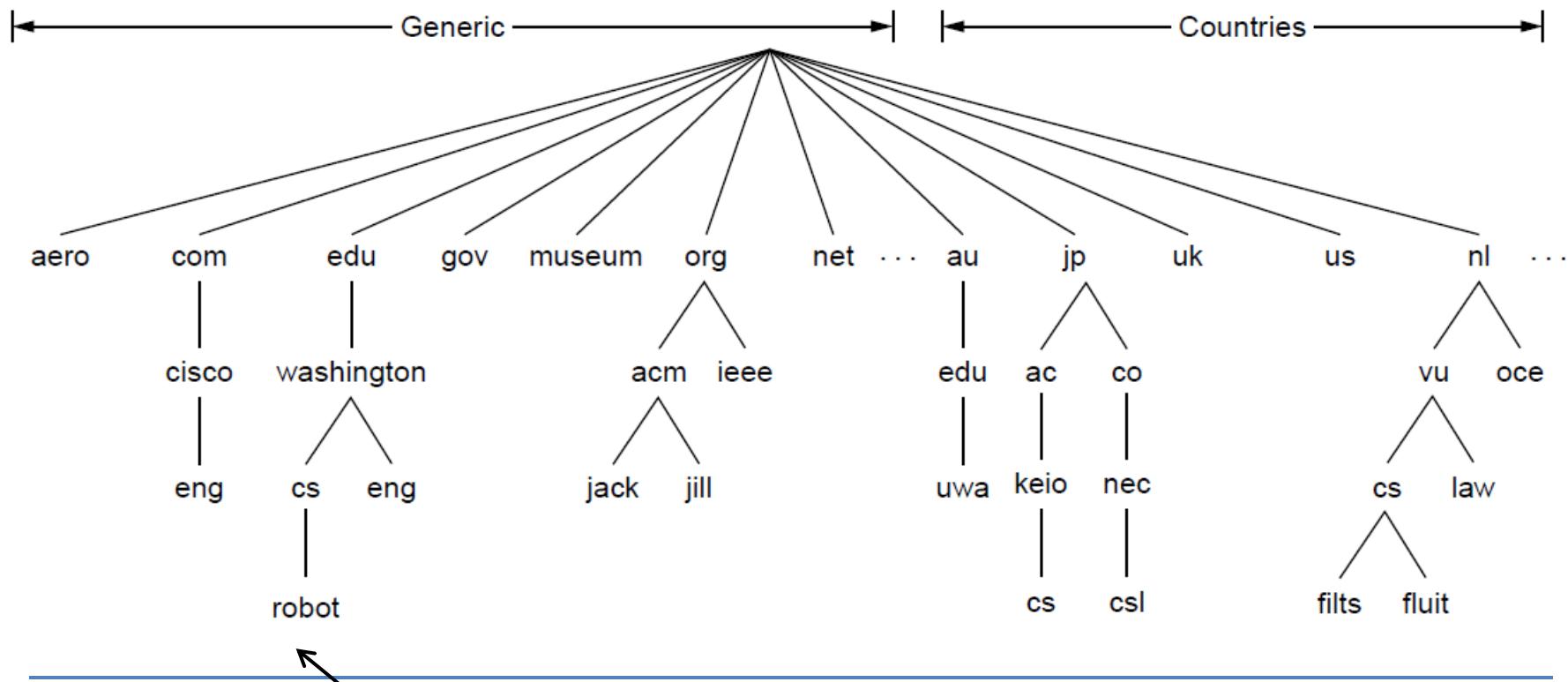


DNS (Domain Name System)

- Problem?
 - IP address (32 bit), e.g., 121.7.106.83 – used for addressing datagrams
 - **www.yahoo.com – used by humans**
- **Question:** how do you map between IP address and name, and vice versa?
- **Domain Name System:**
 - *distributed database* implemented in a hierarchy of many *name servers*
 - *application-layer protocol* that allows a host to query the database in order to *resolve* names (address/name translation)
 - used by other application-layer protocols (http, ftp, smtp)

Conceptual Divisions of DNS Namespace

- A hierarchical naming convention; the top of the hierarchy is managed by ICANN (*The Internet Corporation for Assigned Names and Numbers*).



The computer `robot.cs.washington.edu`

Name Space

- Internet historically divided into over 250 top-level domains (TLD).

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes
biz	Businesses	2001	No
coop	Cooperatives	2001	Yes
info	Informational	2002	No
museum	Museums	2002	Yes
name	People	2002	No
pro	Professionals	2002	Yes
cat	Catalan	2005	Yes
jobs	Employment	2005	Yes
mobi	Mobile devices	2005	Yes
tel	Contact details	2005	Yes
travel	Travel industry	2005	Yes
xxx	Sex industry	2010	No

Why not centralize DNS?

- Single point of failure
- Traffic volume
- Distant centralized database
- Maintenance
- Does not scale well

DNS Services

- hostname to IP **address translation**
- host **aliasing** – alias names for canonical names
 - e.g., canonical `relay1.westcoast.enterprise.com` aliased to `www.enterprise.com`
- mail server aliasing
 - e.g., `Bob@relay1.westcoast.hotmail.com` aliased to `Bob@hotmail.com`
- **load distribution**
 - *busy sites are replicated over multiple servers*
 - *a set of IP addresses is associated with one canonical name*
 - *DNS server rotates the order of the addresses to distribute the load*

Domain Name Characteristics

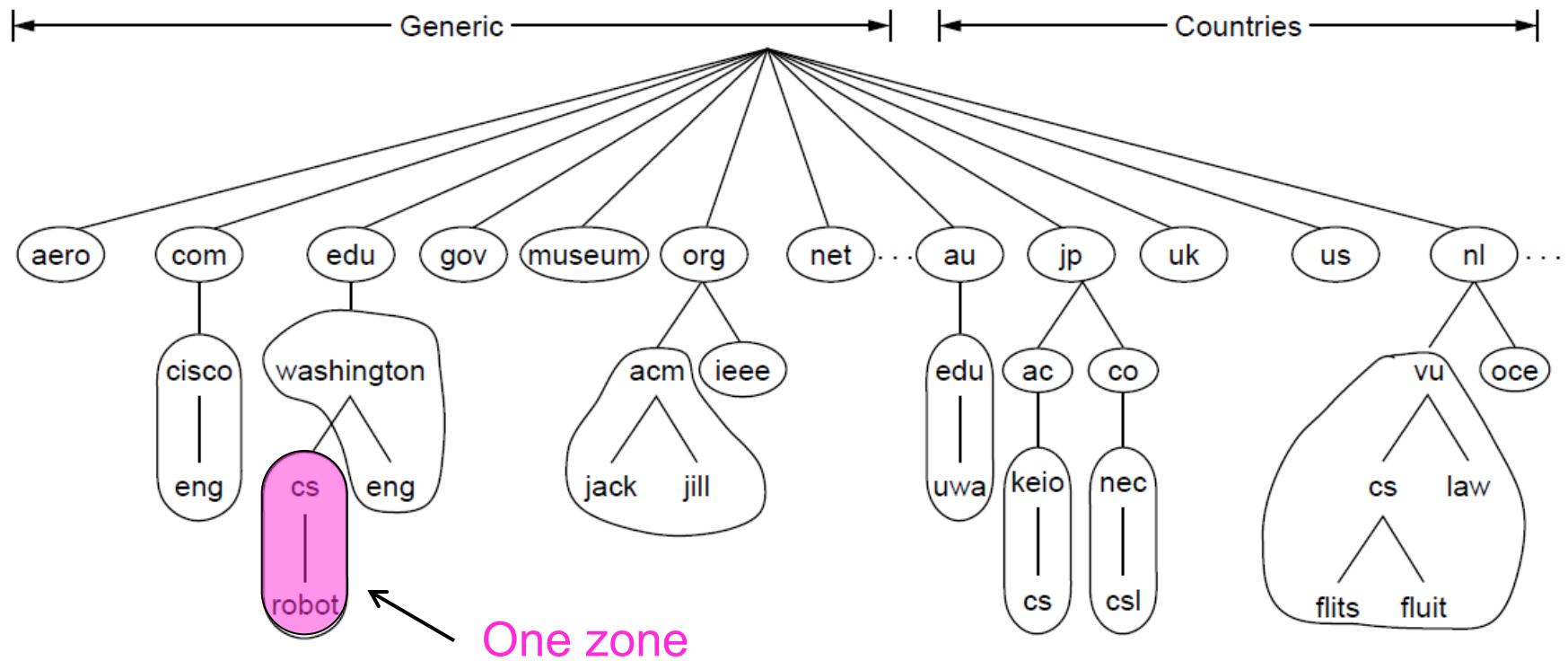
- Domain names:
 - Are case insensitive
 - Can have up to 63 characters per constituent
 - Can have up to 255 chars per path
 - Can be internationalised (since 1999)
- Naming conventions usually follow either organisational or physical boundaries eg.
 - au.ibm.com / uk.ibm.com (for email)
 - ibm.com.au / ibm.co.uk (for web)
- Absolute domain names ends in a “.”
- Relative domain names partially specify the location and can be used only within the context of an absolute domain name

Zone Name Servers

- DNS namespace divided into **non-overlapping zones**
- Each zone contains a part of the DNS tree and also name servers authoritative for that zone -
 - usually 2 name servers for a zone (called the primary and secondary name servers),
 - sometimes secondary is actually outside the zone (for reliability)
- Name servers are arranged in a hierarchical manner extending from a set of root servers

Name Servers

- The DNS name space is divided into nonoverlapping zones; each circled contains some part of the tree.



Root Name Servers

- The root servers form the authoritative cluster for enquiry in the event of locally-unresolvable name queries
- There are 13 root servers globally
 - In some cases, a root server is a cluster of servers

Resource Records

- The **Resource Records** (RR) are the key objects in the Domain Name System
- A RR consists of a domain name, TTL, class, type, value
 - Domain Name: which domain this record applies to
 - TTL: indicates stability or temporal extent of the record
 - Class: IN for internet (others exist, but deprecated)
 - Type: a closed vocabulary of the following:
 - A : The Internet address of the host
 - CNAME : The canonical name for an alias
 - MX : The mail exchanger
 - NS : The name server
 - PTR : The host name if the query is in the form of an Internet address; otherwise the pointer to other information
 - SOA : The domain's start-of-authority information
 - Value: data (semantics depend on record type)

Asking for Domain Name: Example

User requests the URL

www.someschool.edu/index.html

1. User machine runs the client side of the DNS software
2. Browser extracts the hostname from the URL, and passes it to the client-side of the DNS application
3. DNS client sends a query containing the hostname to a DNS server
4. DNS client eventually receives a reply containing the IP address for the hostname
5. Browser initiates a TCP connection to the process located at port 80 at the IP address

A Typical DNS Query: dig

```
dig www.unimelb.edu.au
; <>> DiG 9.3.0s20021217 <>> www.unimelb.edu.au
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19905
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2
;; QUESTION SECTION:
;www.unimelb.edu.au. IN A
;; ANSWER SECTION:
www.unimelb.edu.au. 15589 IN A 128.250.148.40
;; AUTHORITY SECTION:
unimelb.edu.au. 1217 IN NS ns2.unimelb.edu.au.
unimelb.edu.au. 1217 IN NS ns1.unimelb.edu.au.
;; ADDITIONAL SECTION:
ns1.unimelb.edu.au. 491 IN A 128.250.20.2
ns2.unimelb.edu.au. 494 IN A 128.250.144.180
;; Query time: 393 msec
;; SERVER: 128.250.66.5#53(128.250.66.5)
;; WHEN: Fri Apr 18 05:46:56 2014
;; MSG SIZE rcvd: 120
```

Domain Resource Records

```
; Authoritative data for cs.vu.nl
cs.vu.nl.      86400  IN  SOA   star boss (9527,7200,7200,241920,86400)
cs.vu.nl.      86400  IN  MX    1 zephyr
cs.vu.nl.      86400  IN  MX    2 top
cs.vu.nl.      86400  IN  NS    star ← Name
star           86400  IN  A     130.37.56.205
zephyr         86400  IN  A     130.37.20.10
top            86400  IN  A     130.37.20.11 ← IP addresses of
www            86400  IN  CNAME star.cs.vu.nl
ftp             86400  IN  CNAME zephyr.cs.vu.nl

flits          86400  IN  A     130.37.16.112
flits          86400  IN  A     192.31.231.165
flits          86400  IN  MX   1 flits
flits          86400  IN  MX   2 zephyr
flits          86400  IN  MX   3 top

rowboat        IN  A     130.37.56.201
                IN  MX   1 rowboat
                IN  MX   2 zephyr ← Mail
little-sister   IN  A     130.37.62.23
gateways
laserjet       IN  A     192.31.231.216
```

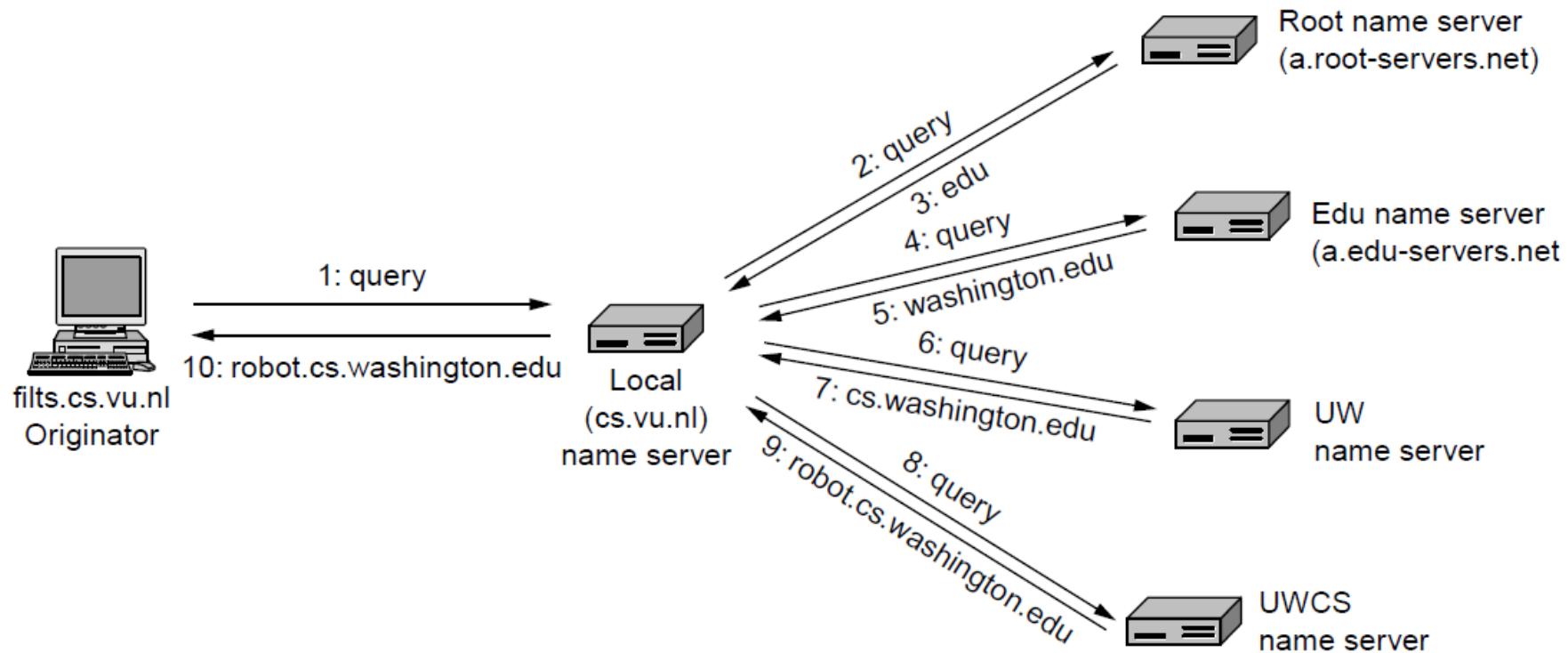
- A portion of a possible DNS database for cs.vu.nl.

DNS In Action

- Finding the IP address for a given hostname is called **name resolution** and is done with the DNS protocol.
- Resolution:
 - Computer requests local name server to resolve
 - Local name server asks the root name server
 - Root returns the name server for a lower zone
 - Continue down zones until name server can answer
- DNS protocol:
 - Runs on UDP port 53, retransmits when lost messages
 - Caches name server answers for better performance

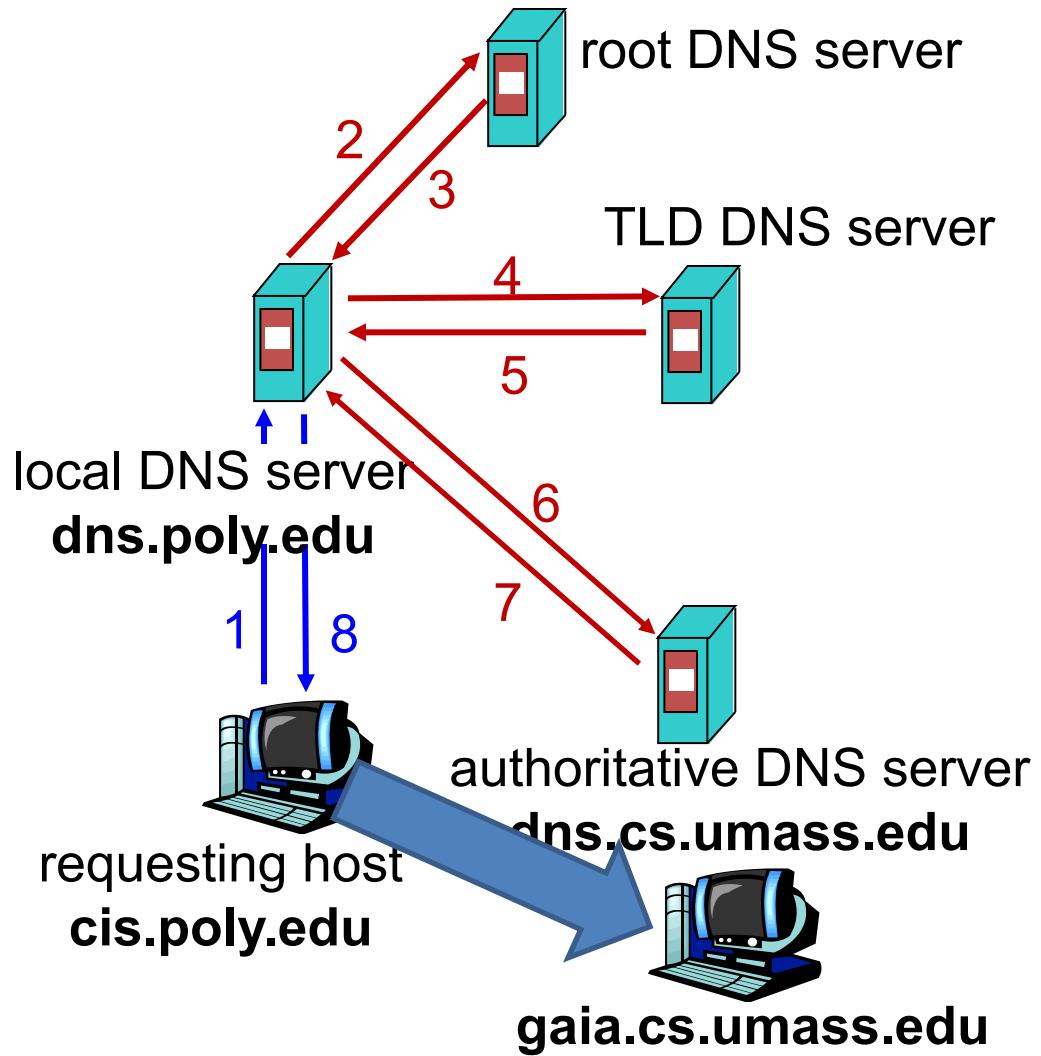
Example

- Example of a computer looking up the IP for a name



DNS Name Resolution Contd

- host at cis.poly.edu wants IP address for gaia.cs.umass.edu
- iterated query:
 - contacted server replies with name of server to contact
 - “I don’t know this name, but ask this server”
- recursive query:
 - server obtains mapping on client’s behalf



DNS: Caching & Updating Records

- Once (any) name server learns a mapping, it *caches* the mapping
 - IP addresses of TLD servers typically cached in local name servers
 - root name servers not often visited
 - Cache entries timeout (disappear) after some time

Week 9: Application Layer

Internet Technologies COMP90007

Lecturer: Muhammad Usman

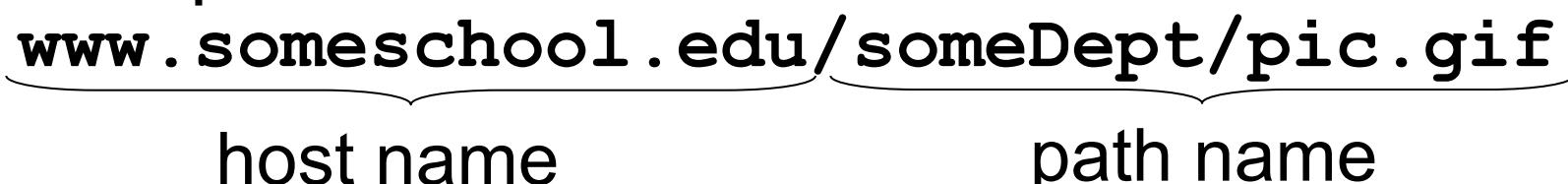
Semester 2, 2020

World Wide Web (WWW)

■ World Wide Web key components are?

- Client and Server software – **Firefox** is the client software for web access where **Apache** is on the server side
- Web mark-up languages - **HTML** – how webpages are coded
- Web scripting languages – More dynamicity to webpages - **Javascript**
- **HTTP** – about how to transfer

Web Access

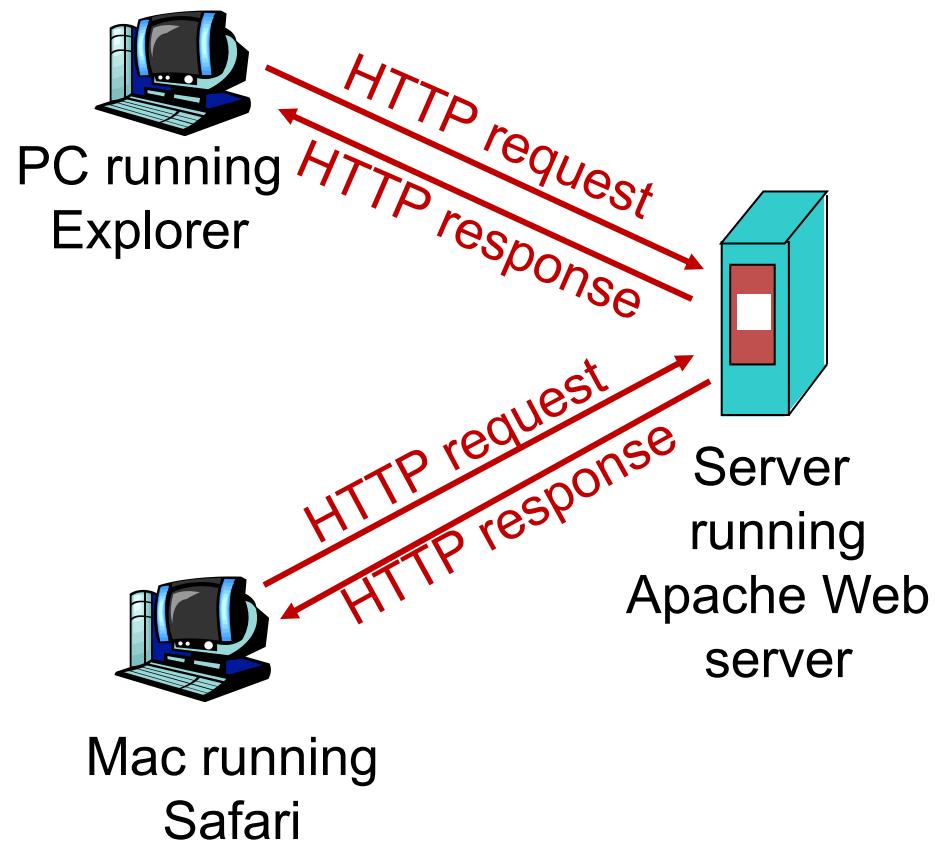
- A web page consists of objects
- An object can be HTML file but also JPEG image, Java applet, audio file, ...
- A web page consists of a base HTML file which includes several referenced objects
- Each object is addressable by a URL (uniform resource locator)
- Example URL:
The URL is shown as "www.someschool.edu/someDept/pic.gif". Two curly braces are placed under the URL. The first brace covers "www.someschool.edu", and the second brace covers "/someDept/pic.gif".

host name path name

HTTP: hypertext transfer protocol

HyperText “text ... cross-referencing between sections of text and associated graphic material”

- HTTP is at the application layer
- client/server model
 - **client:** browser that requests, receives and displays Web objects
 - **server:** Web server sends objects in response to requests



HTTP Connections

- Non-persistent HTTP
 - at most one object sent over a TCP connection
- Persistent HTTP
 - multiple objects can be sent over a single TCP connection between client and server

Non-persistent HTTP (I)

suppose user enters URL:

www . someSchool . edu /someDepartment /home . index

1a. HTTP client initiates TCP connection to HTTP server (process) at www . someSchool . edu on port 80

2. HTTP client sends a HTTP **request message** (containing URL) into TCP connection socket. Message indicates that client wants object **someDepartment /home . index**

contains text and references to 10 images

1b. HTTP server at host www . someSchool . edu waiting for TCP connection at port 80. Accepts connection, notifying client

3. HTTP server receives request message, forms **response message** containing requested object, and sends message into its socket

time

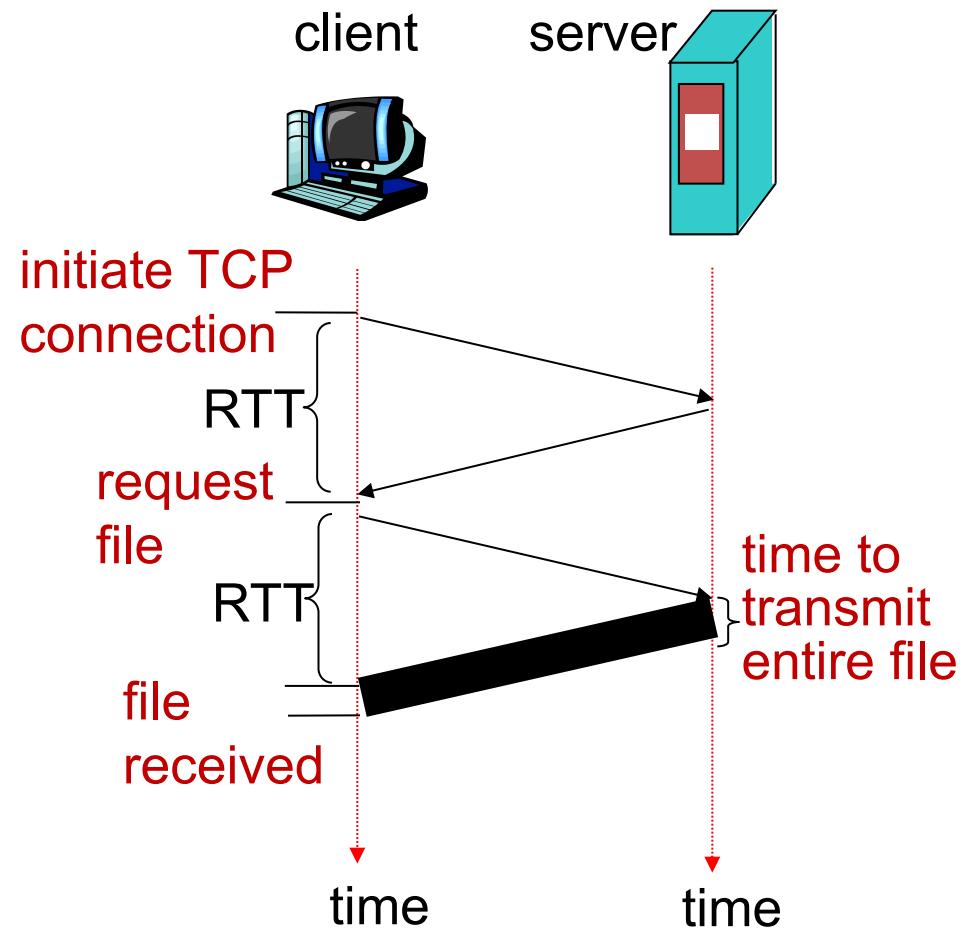
Non-persistent HTTP (II)

time

- 
4. HTTP client receives response message containing HTML file
 5. HTTP server closes TCP connection.
 6. Parses HTML file, and finds 10 referenced jpeg objects
 7. Steps 1-6 repeated for each of the 10 jpeg objects

Non-Persistent HTTP: Response Time

- Round Trip Time (RTT) – time for a small packet to travel from client to server and back
- Response time
 - one RTT to initiate TCP connection
 - one RTT for HTTP request and first few bytes of HTTP response to return
 - file transmission time
- Total response time =
$$2 \text{ RTT} + \text{file transmission time}$$



Non-Persistent HTTP – Issues

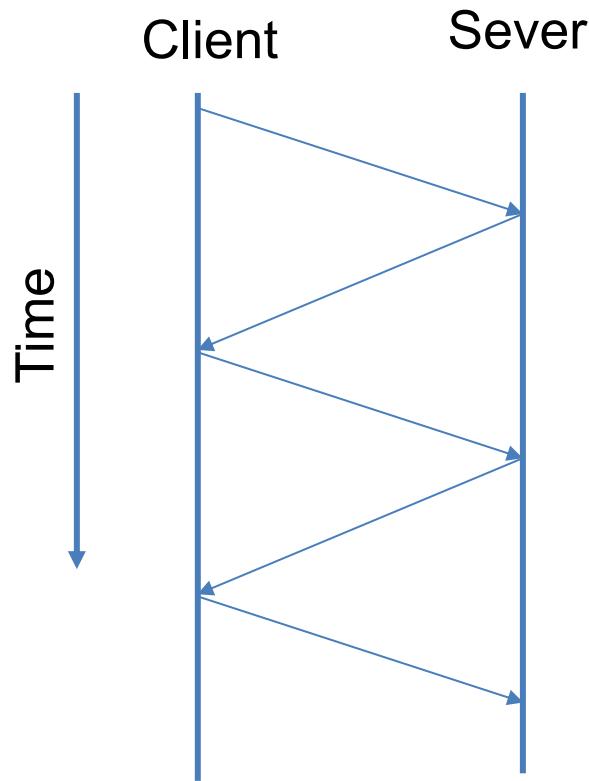
- Requires new connection per requested object
- OS overhead for each TCP connection
- Delivery delay of 2 RTTs per requested object

Persistent HTTP

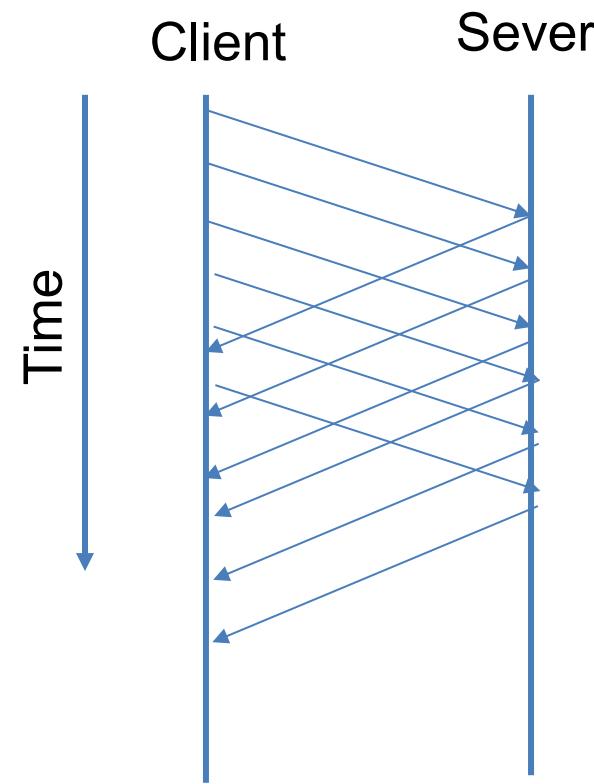
- Server leaves connection open after sending response
- Subsequent HTTP messages between same client/server sent over open connection
- Pipelining – client sends request as soon as it encounters a referenced object
 - → as little as one RTT for all the referenced objects
- Server closes a connection if it hasn't been used for some time

Sequential vs Pipeline

Sequential



Pipeline



HTTP Request Message: Example

request line

(GET,
POST,
HEAD
commands)

header
lines

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
```

indicates
end of
header
lines

\r\n

Persistent HTTP

HTTP Response Message: Example

200 OK – request succeeded, requested object later in this msg

....

404 Not Found – requested document not found on this server

status line:

HTTP/1.1 200 OK\r\nDate: Sun, 26 Sep 2010 20:09:20 GMT\r\nServer: Apache/2.0.52 (CentOS) \r\nLast-Modified: Tue, 30 Oct 2007 17:00:02 GMT\r\nContent-Length: 2652\r\nKeep-Alive: timeout=10, max=100\r\nConnection: Keep-Alive\r\nContent-Type: text/html; charset=ISO-8859-1\r\n\r\ndata data data data data ...

header
lines

data, e.g.,
requested
HTML file

HTTP Request Methods

Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page <small>(write a new page / resource)</small>
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options

HTTP Error Codes

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

Cookies

- **The http servers are stateless**
- Cookies to place small amount (<4Kb) of info on users computer and re-use deterministically (RFC 2109)
- Questionable mechanism for tracking users (invisibly perhaps)

User-server Interaction:Cookie Example 1

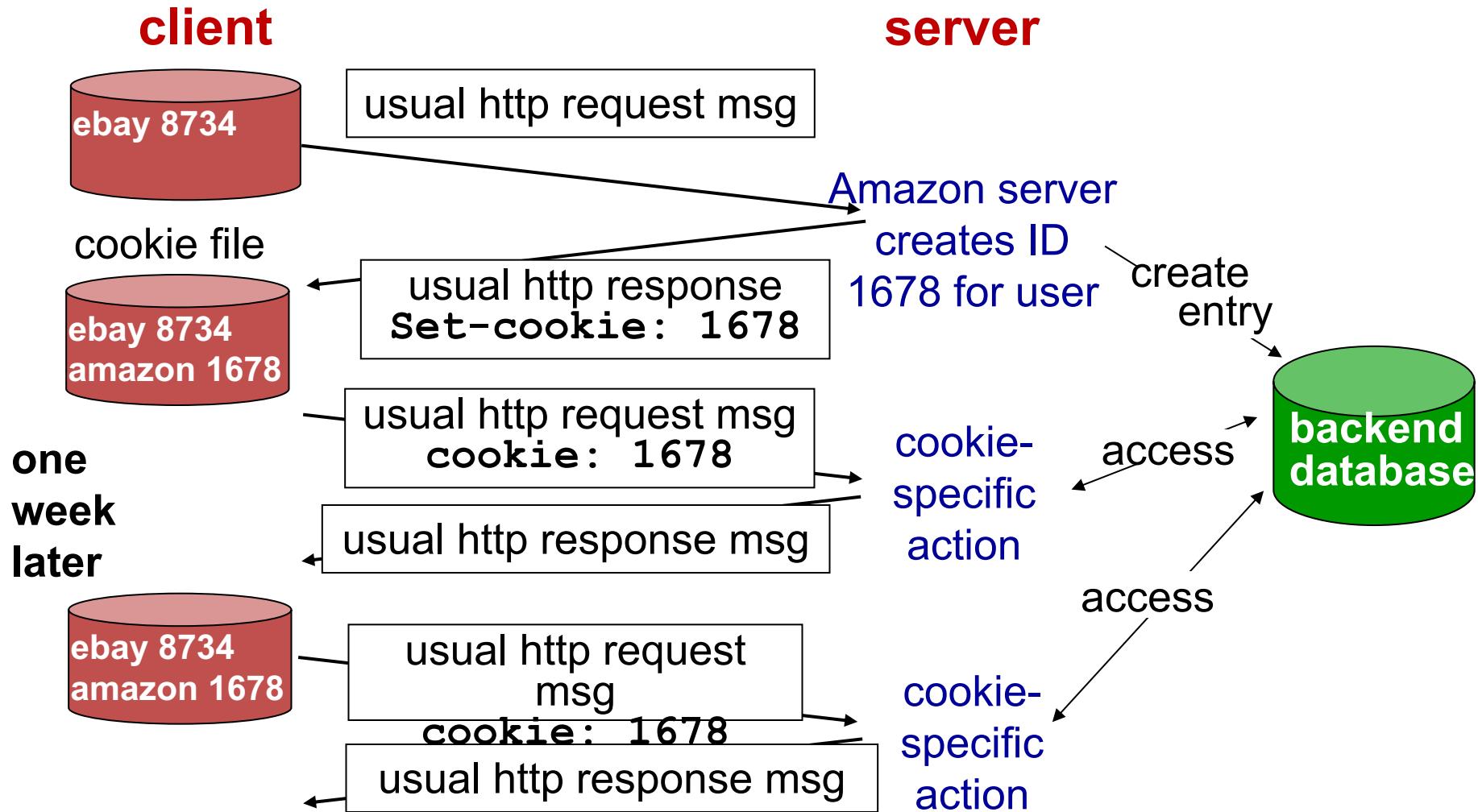
Susan always accesses the Internet from her (*cookie-enabled*) home PC. She visits a specific (*cookie-enabled*) e-commerce site for the first time

- When the initial HTTP requests arrives at the site, the site creates:
 - unique ID
 - entry in backend database for ID
- The e-commerce site then responds to Susan's browser, including in the HTTP response
 - Set-cookie: 1234 — **ID**

User-server Interaction: Cookie Example (Contd)

- Susan's browser appends a line to a cookie file that it manages
 - www.e-commerce-site.com 1234
- Next time Susan request a page from that site, a cookie header line will be added to her request
 - Cookie: 1234
- The server will then perform a cookie-specific action

Keeping state with Cookies: Example 2



Beyond User Tracking: Advantages of Cookies

- Authorization
- Shopping carts
- Recommendations
- User session state

Cookies vs Sessions

- **Both introduce “memory” or state into HTTP and are about multiple TCP connections**

Sessions

- Sessions information regarding visitor's interaction stored at the server side: up to some hours
- When user closes the website, the session ends
- Sessions information size can be large

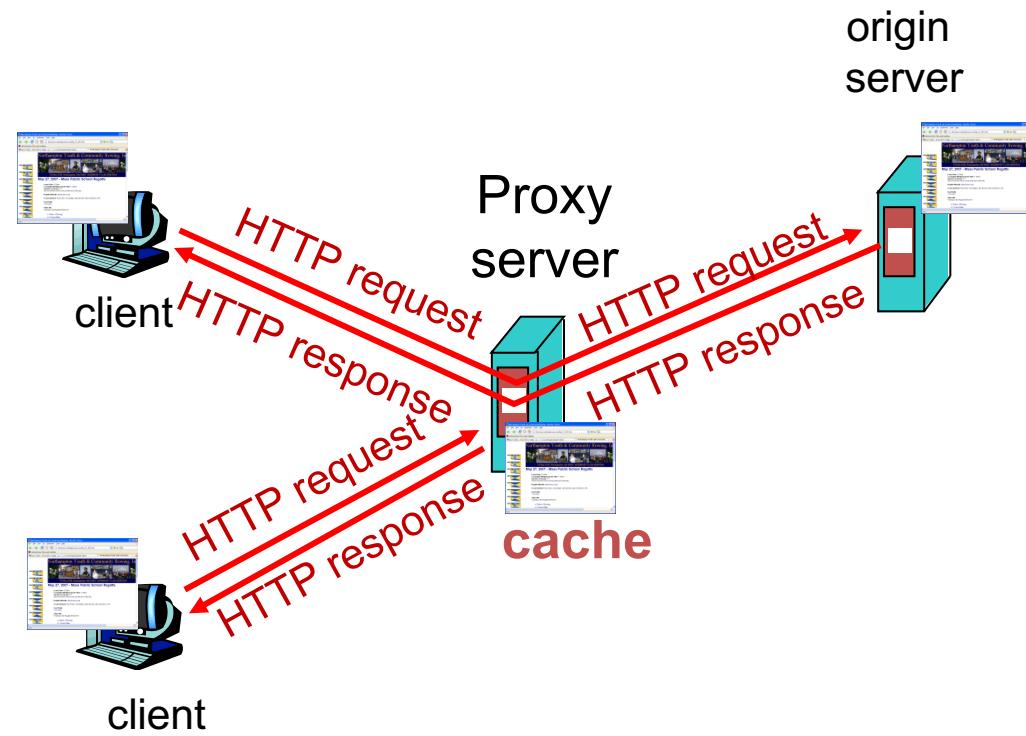
Cookies

- Cookies are transferred between server and client
- Cookie information stored at both client and server
- Maintain client information until deleted
- Cookies information size limited

Web Caches (Proxy Server)

Goal: satisfy client request without involving origin server

- ❖ User sets browser to access Web via cache
→ browser sends all HTTP requests to cache
 - **if object in cache,** cache returns object
 - **else** cache requests object from origin server, then returns object to client



More about Web Caching

- Cache acts as both client and server
- Typically cache is installed by ISP (university, company, residential ISP)
- Causes problems for frequently changing data though

Why Web caching?

- Reduce response time for client request
- Reduce traffic on an institution's access link

COMP90007 Internet Technologies

Week 9 Workshop

Semester 2, 2020

Suggested solutions

Question 1

In determining maximum packet lifetime, we have to be careful and pick a large enough period to ensure that not only the packet but also its acknowledgements have vanished. Discuss why this is needed.

Answer:

Look at the second duplicate packet in Fig. 6-11(c). When the packet arrives, it would be a disaster if acknowledgements to y were still floating around.

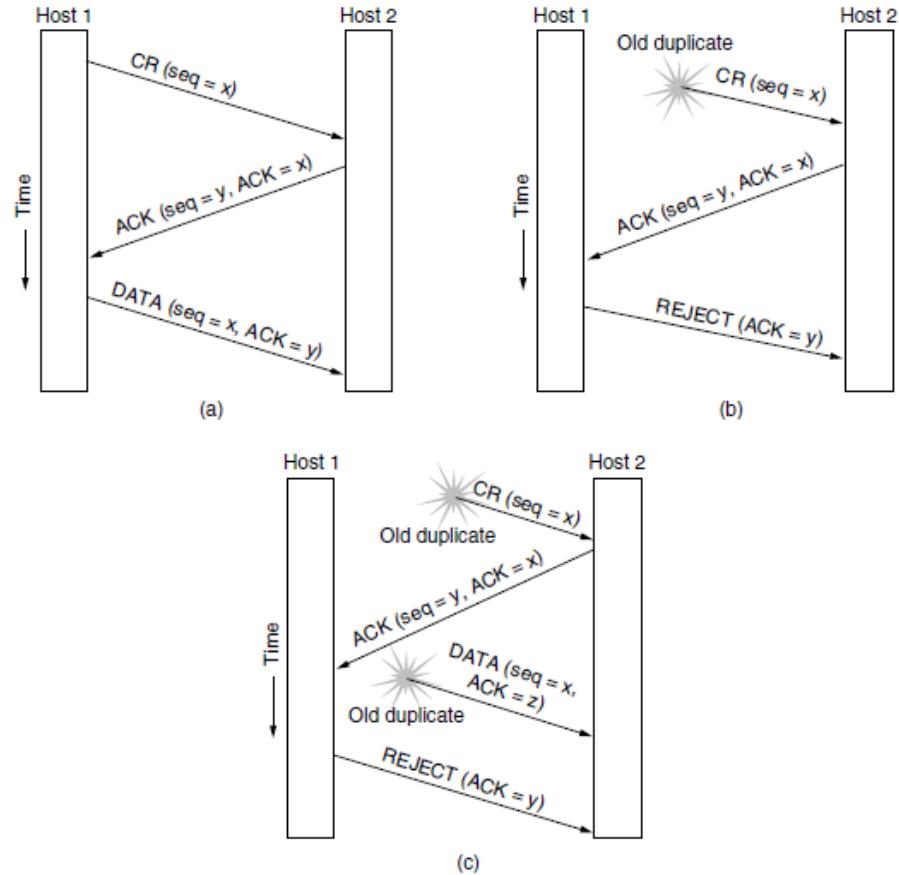


Figure 6-11. Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST. (a) Normal operation. (b) Old duplicate CONNECTION REQUEST appearing out of nowhere. (c) Duplicate CONNECTION REQUEST and duplicate ACK.

Question 2

Imagine that a two-way handshake, rather than a three-way handshake were used to set up connections. In other words, the third message was not required. Are deadlocks now possible? Give an example or show that none exist.

Answer:

Deadlocks are possible!

For example, a packet arrives at A out of the blue, and A acknowledges it. The acknowledgement gets lost, but A is now open while B knows nothing at all about what has happened. Now the same thing happens to B, and both are open, but expecting different sequence numbers. Timeouts have to be introduced to avoid the deadlocks at least.

Question 3

Does the 3 way handshake based connection release protocol create a flawless disconnection?

Answer: No. The three-way handshake-based solution is an approximation. E.g.: Imagine the timeout for case (b) on page 521, if the timeout triggers while there is data lingering in the network then the data will be lost as connection will be terminated early.

Question 4

What is the 2 army problem? Where does it occur in networking? Provide an example.

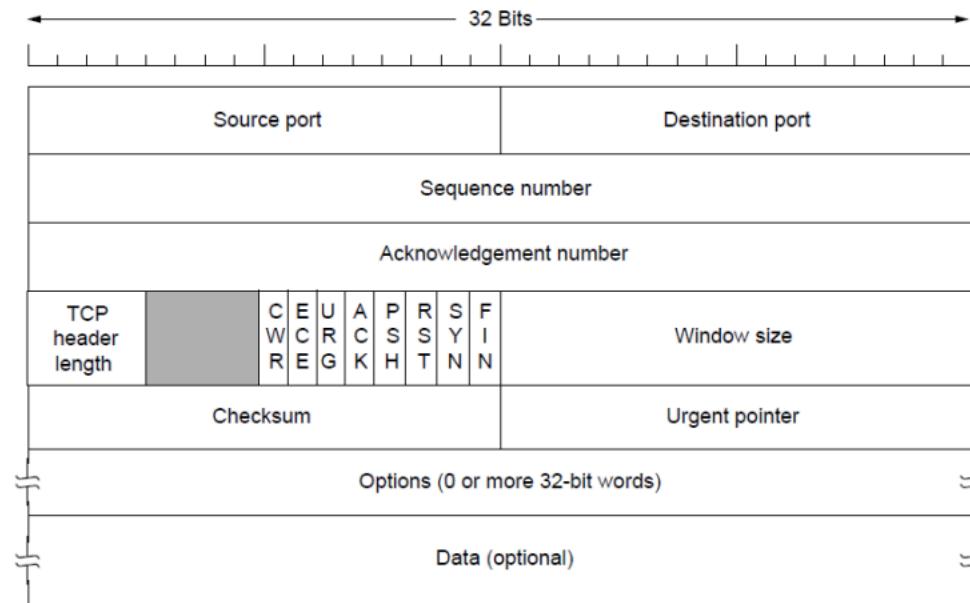
Ans. Refer to Page 519 for the explanation of two armies, one of which is split up with an enemy in the middle, and how they communicate with each other to try and coordinate for launching an attack.

Example – Connection Release.

Question 5

What information is sent with the TCP Segment header, explain each field briefly?

Ans.

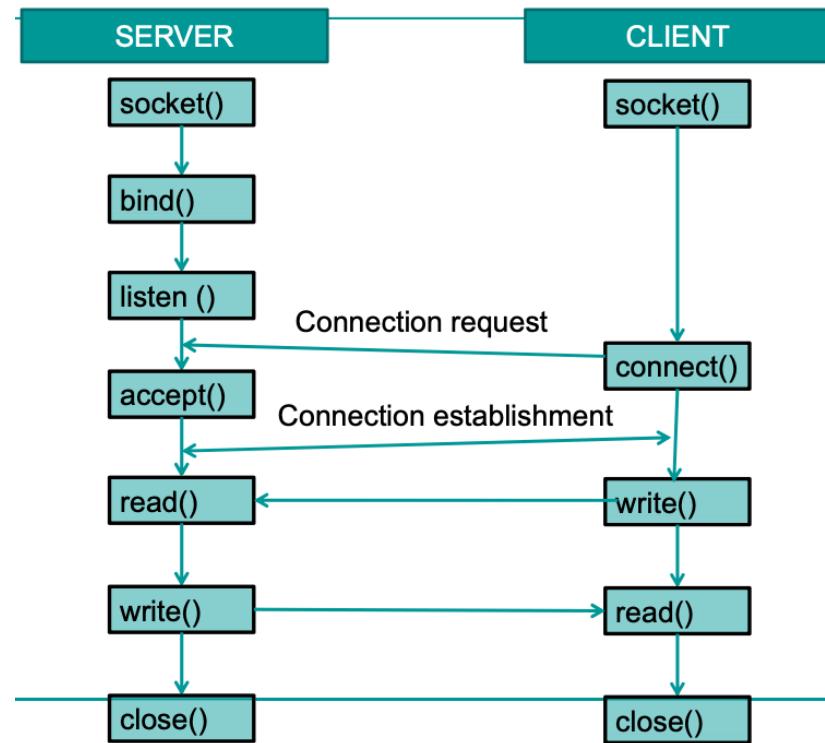


See explanation for each field in slides.

Question 6

Describe with a simple flowchart how a single socket-based client-server communication works?

Answer:



Week 10: Application Layer

Internet Technologies COMP90007

Lecturer: Muhammad Usman

Semester 2, 2020

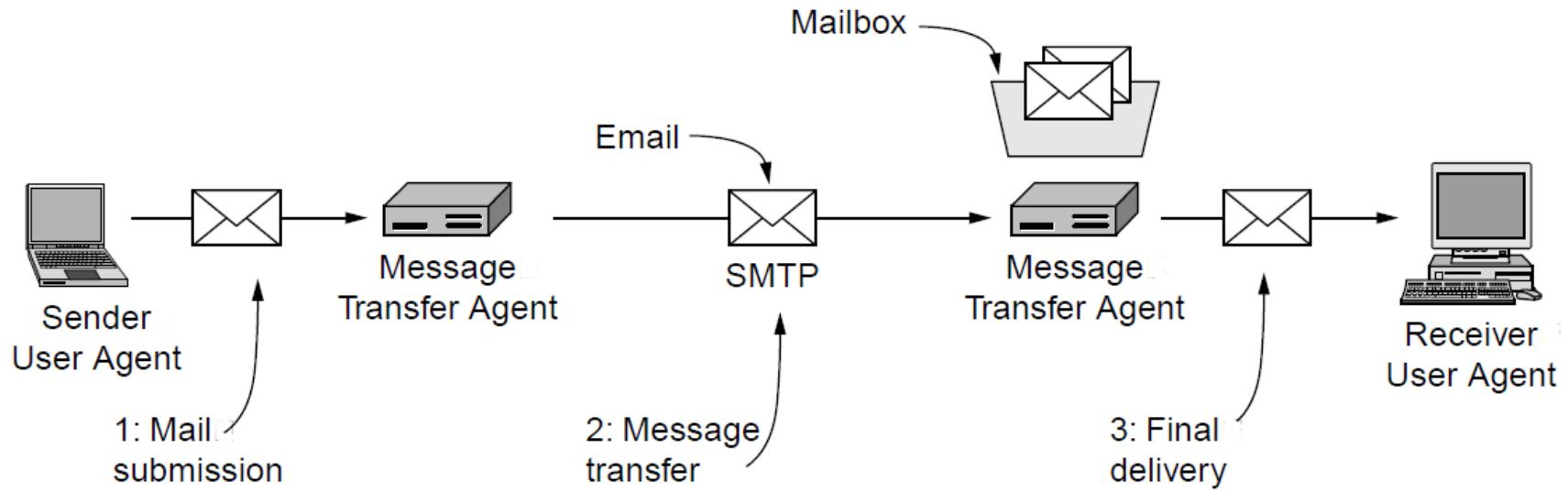
Email

- Email involves
 - User Agent: Thunderbird, Outlook
 - Message Transfer Agent: Exchange
 - Message Transfer Protocols

Email Services

- Email has a long heritage (since 1960's)
- Standards for Internet-enabled email are based on 2 RFC's
 - RFC 821 (transmission)
 - RFC 822 (message format)
 - RFC 2821 and RFC 2822 (revised versions of earlier RFCs)

Architecture and Services



User agents

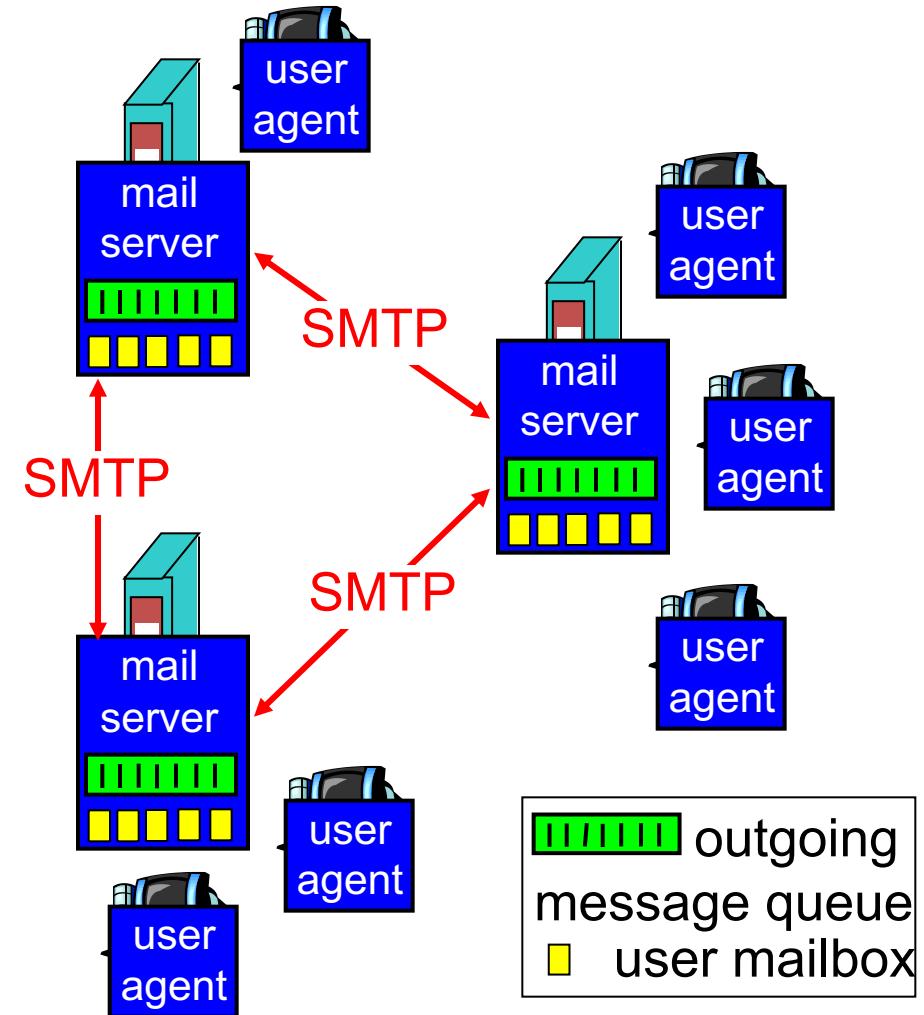
Allow user to read and send email

Message transfer agents

Transport messages from source - destination

Electronic Mail - Overview

- **SMTP (Simple Mail Transfer Protocol)** is used to ***send*** messages from the sender's
 - **mail server** to the receiver's **mail server**
 - **user agent** to the sender's **mail server**



User Agent

- Basic functions: compose, report, display, dispose
- **Envelope** and **contents**: encapsulation of transport related information
 - Envelope - destination address, priority, and security level, all of which are distinct from the message itself
 - Mail servers use the envelope for routing
- **Header** and **body**: header - user agent control info; body for human recipient
 - contains control information for the user agents
- User must provide message, destination, optional other parameters
- Addressing scheme **user@dns-address**

RFC 822: Message

- RFC 822 doesn't distinguish header and envelope fields
- RFC 822 allows users to invent new headers for private use but they must start with X-

Multipurpose Internet Mail Extensions (MIME) #1

- In the early days of email, messages were in English and used only simple text: **RFC822 was enough for these simple constraints**
- In time the inadequacy of RFC822 became apparent
 - Languages with **accents** (French, Spanish)
 - **Non-Latin alphabets** (eg Cyrillic)
 - **Non-alphabetic language** (eg Chinese, Japanese)
 - Messages with content other than text (**audio, images**)
- As a result, MIME (RFC 1341) was written (later updated in RFCs 2045-2049)

Multipurpose Internet Mail Extentsions (MIME) #2

- **MIME retains RFC822 format but adds** structural elements to the message body and defines encoding rules for non-ASCII messages
- MIME has 5 additional message headers:
 - MIME-Version: identifies the MIME version
 - Content-Description: human readable describing contents
 - Content-Id: unique identifier
 - Content-Transfer-Encoding: how body is wrapped for transmission
 - Content-Type: type and format of content (e.g., text/plain, html, video, etc..)

MIME Types and Subtypes

Type	Example subtypes	Description
text	plain, html, xml, css	Text in various formats
image	gif, jpeg, tiff	Pictures
audio	basic, mpeg, mp4	Sounds
video	mpeg, mp4, quicktime	Movies
model	vrml	3D model
application	octet-stream, pdf, javascript, zip	Data produced by applications
message	http, rfc822	Encapsulated message
multipart	mixed, alternative, parallel, digest	Combination of multiple types

Message Format

- Typical multipart message containing HTML and audio alternatives is given here

From: alice@cs.washington.edu
To: bob@ee.uwa.edu.au
MIME-Version: 1.0
Message-Id: <0704760941.AA00747@cs.washington.edu>
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
Subject: Earth orbits sun integral number of times

This is the preamble. The user agent ignores it. Have a nice day.

--qwertyuiopasdfghjklzxcvbnm
Content-Type: text/html

<p>Happy birthday to you

Happy birthday to you

Happy birthday dear Bob

Happy birthday to you</p>

--qwertyuiopasdfghjklzxcvbnm
Content-Type: message/external-body;
access-type="anon-ftp";
site="bicycle.cs.washington.edu";
directory="pub";
name="birthday.snd"

content-type: audio/basic
content-transfer-encoding: base64
--qwertyuiopasdfghjklzxcvbnm--

Message Transfer

- Transfer
 - SMTP (Simple Message Transfer Protocol)
- Delivery
 - POP3 (Post Office Protocol 3)
 - Download to a single device
 - IMAP (Internet Message Access Protocol)
 - Designed with multiple devices in mind

SMTP

- Simple Message Transfer Protocol
- Simple ASCII protocol, operating on TCP port 25
- RFC 821: Simple Mail Transfer Protocol
- RFC 2821: Extended Simple Mail Transfer Protocol

SMTP Steps

■ Basic steps SMTP:

- User agent submits to MTA (mail transfer agent) on port 587
- One MTA to the next MTA on port 25
- Other protocols used for final delivery (IMAP, POP3)

IMAP

- Used for final delivery
- Internet Message Access Protocol (IMAP)
- RFC 3501 defines version 4
- User agent runs an IMAP client for this
- Protocol has command like:
 - Login, List, Copy, Create, Delete, etc
- Main difference with POP3 is mail remains on server
- Complex protocol but makes mail machine independent

Webmail

- Gmail and alike
- A service run by a company server
- An interface to managing email over the Web
- Mainly it is an user interface

Spam

- Unwanted email
- Main countermeasures are
 - Filters based on email content
 - Blacklisting known spam addresses
 - Parking email from unknown sources
 - Collecting spam and creating a knowledge-base
 - Detecting mass emails
 - ...

Week 10: Application Layer

Internet Technologies COMP90007

Lecturer: Muhammad Usman

Semester 2, 2020

A Key Application Layer Worry: Dealing with Multimedia Data

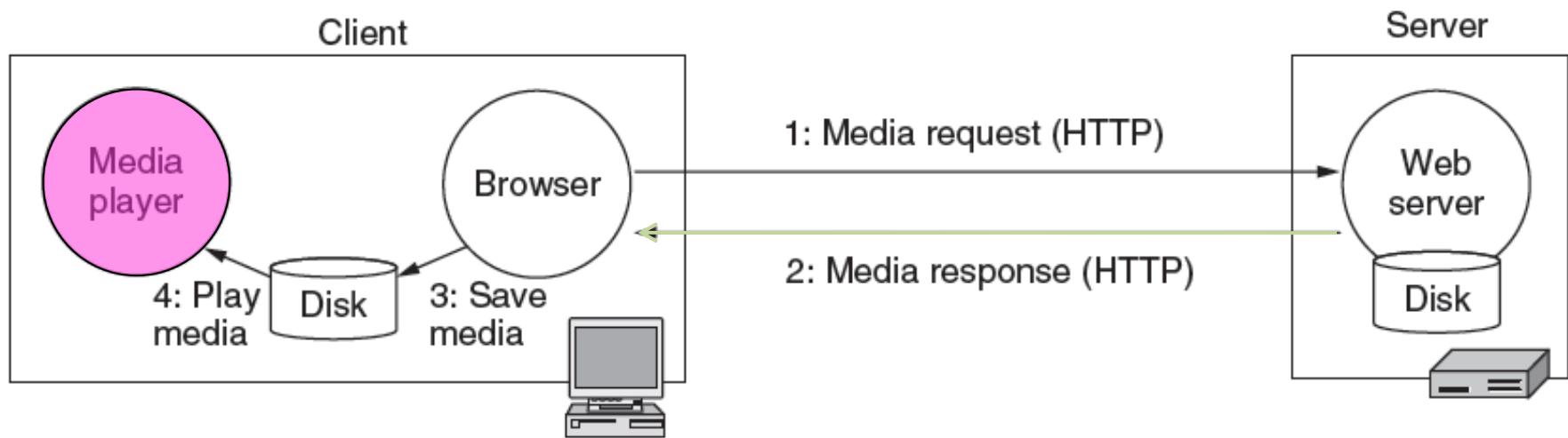
- What is different with Multimedia data?
 - Higher bandwidth requirements
 - Higher QoS requirement, i.e., **delay sensitivity**

Delivery Through Specialized Companies

- Separate providers

- Not all communication is one-to-one, quite a bit is multicast/broadcast which is different to most traffic
- Specialized infrastructure also needs special attention
 - We use separate multimedia servers from web servers: Streaming multimedia service providers are often separated and highly specialised, compared to traditional web hosts

A Basic Model for Multimedia on the Web



Problems with the Basic Model

- The entire media file must be transmitted over the network before playback starts
 - Imagine waiting for all the movie to come to your side for everything you watch: That is not tolerable

Problems with the Basic Model

- Basic model assumes mainly point-to-point data distribution rather than a point-to-multipoint (broadcast) distribution model
 - Recall special methods for efficient multicast; none can be used if we do not realize this special need/mode

Problems with the Basic Model

- Basic model relies on simple browser/plugin/helper integration and traditional service types:
 - This limits the capabilities of the software for delivery

Streaming Media Protocols

- ❑ HTTP
- ❑ RTP - Real-time Transport Protocol (works over UDP allows for time-stamping etc)
- ❑ ...
- ❑ MPEG-4 (allows for compression)
- ❑ ...
- ❑ Microsoft's Windows Media (closed protocol)
- ❑ ...
- ❑ **...many of these protocols may be used at one time to achieve a successful media stream...**

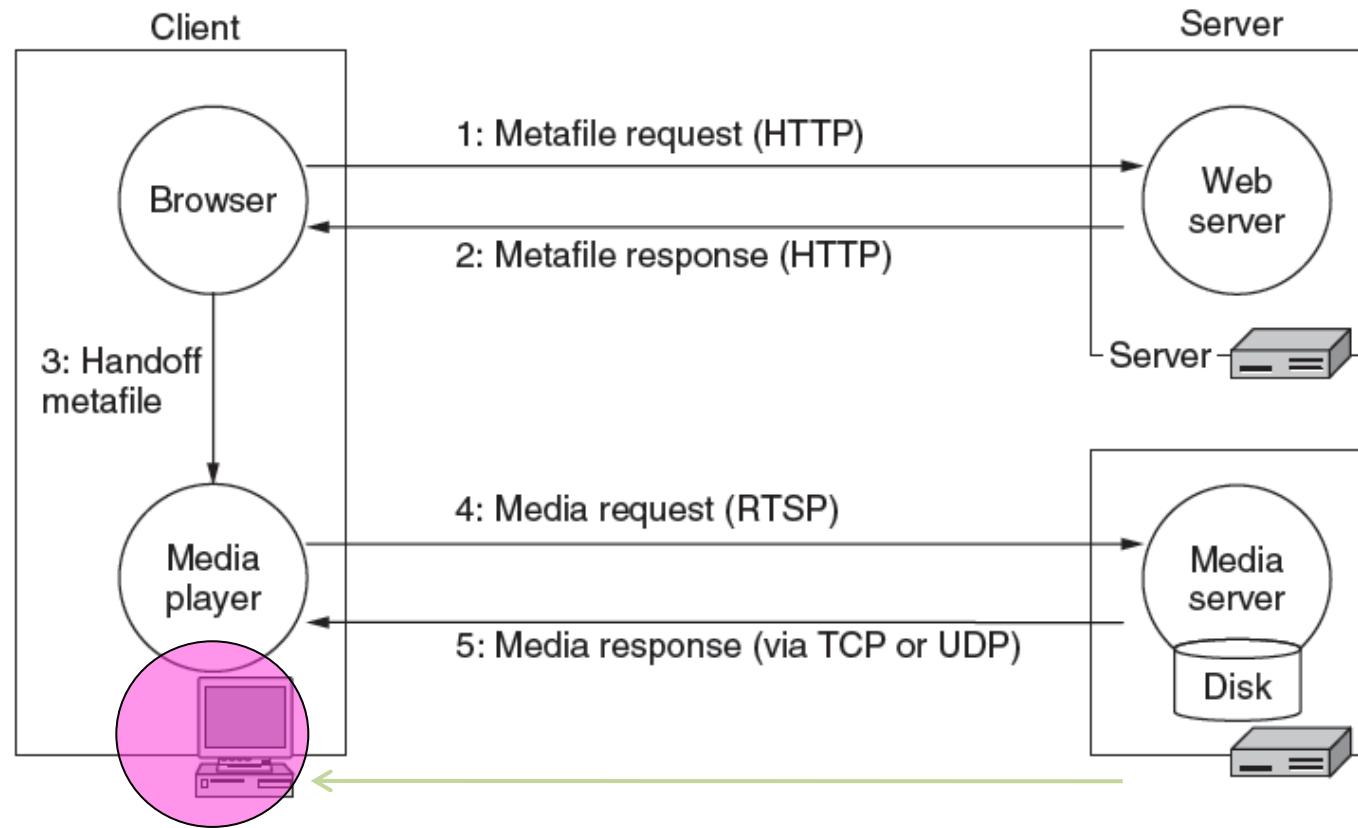
Specialized Multimedia Software

- 4 main tasks of the multimedia playback software
 - First is to deal with the user interface side of the story
 - Functions such as volume control, playback, next, etc..
 - This is commonly what most people see/know
 - But there are 3 others that are less about the UI/controls

Specialized Multimedia Software Contd

- Others are:
 - Handle transmission errors in conjunction with transport protocols
 - Using RTP/UDP errors will likely occur, playback software must manage/mask them gracefully
 - Eliminate jitter
 - Small buffer, quick playback but susceptible to jitter/delay
 - Large buffer, delay at start of playback while buffer fills, but less susceptible to delay/jitter
 - Sometimes compress and almost always decompress the multimedia files to reduce size

Specialized Model



Handling Errors: A Common Method

Forward Error Correction (FEC) is simply the error-correcting encoding of data

For every X data packets Y **new packets are added similar to methods we have seen**

These contains **redundant bits that are used to deal with errors**

Methods use **parity or exclusive-OR** sums of the bits in each of the data packets but are more complex than methods we saw so far

Examples are Reed-Solomon, Tornado codes, etc.

Handling Errors: Other Directions

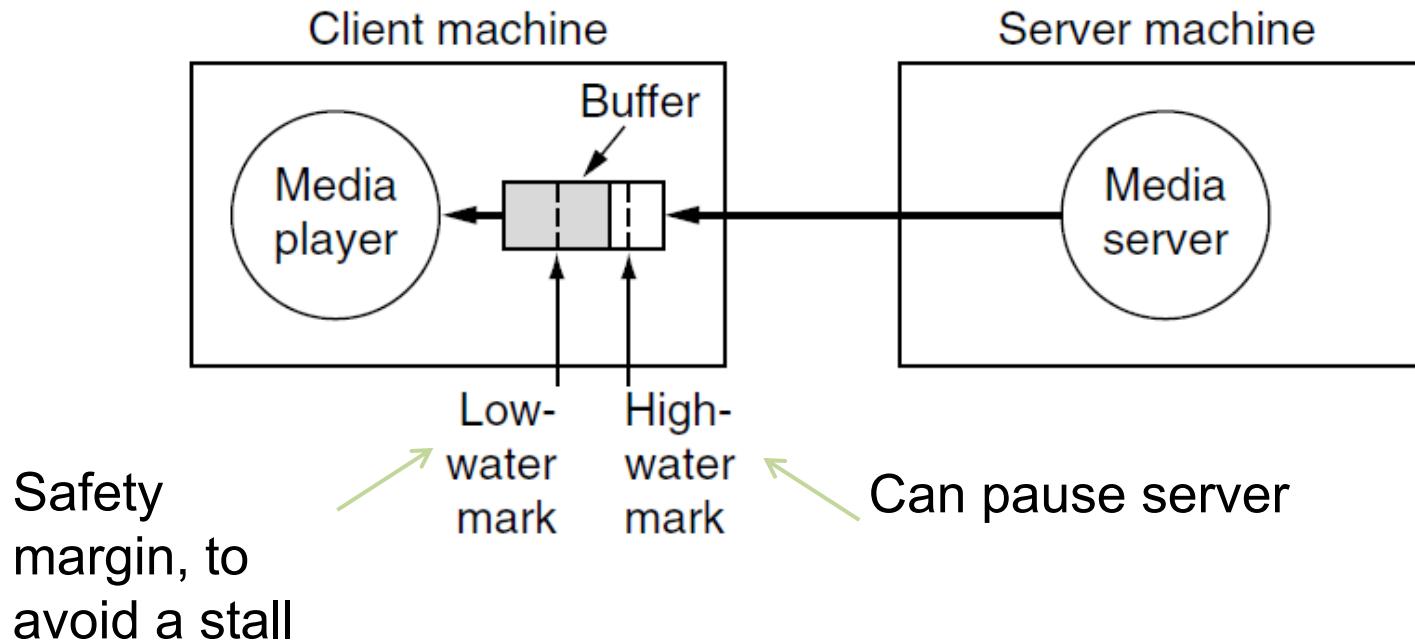
Error Resilience: Remarking for re-sync so that a packet loss does not create a total loss, mainly on sender side

Error Concealment: Done by the receiver e.g., interpolation between frames to reduce displeasing experiences

Retransmission: Less meaningful for streaming data but for watching a movie this can be deployed for lost packets of the movie

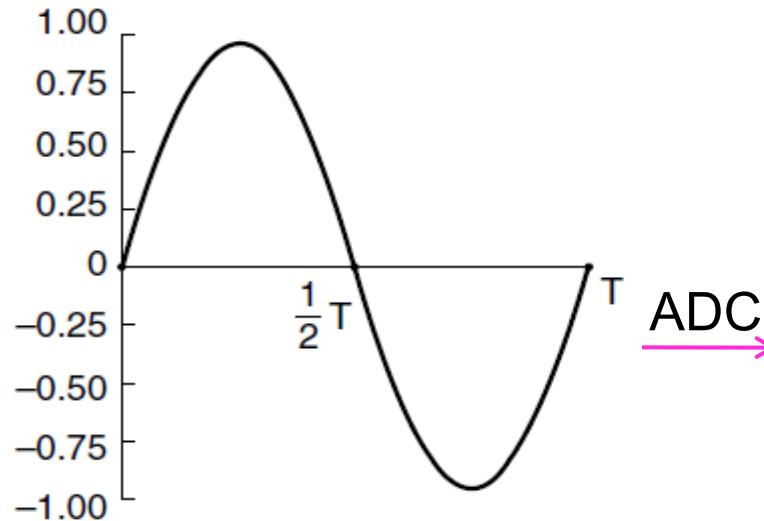
Jitter Management is Crucial for Multimedia

Jitters happens because of variable bandwidth and loss/retransmissions. So we use buffers...

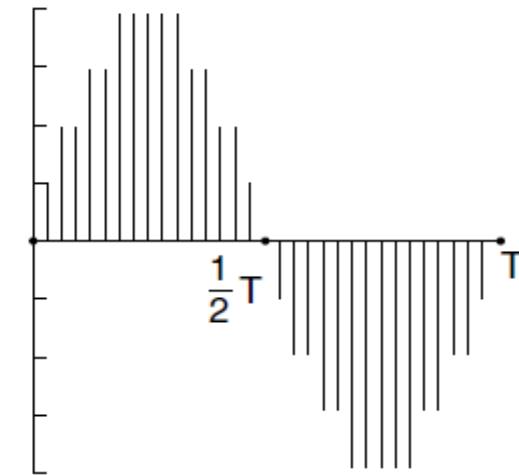


Dealing with Large File: Compression

ADC (Analog-to-Digital Converter) produces digital data, say from a microphone



Continuous audio
(sine wave)



Digital audio
(sampling theory in play)

Example: Audio Compression

- We can use Nyquist and Shannon theorems again: to convert analog data to digital first
- Then apply techniques to eliminate some data...

For example: perceptual coding is that some data can mask other data, e.g., in audio, which can be used to eliminate the data

- Frequency masking: Some sounds mask/hide others so there is no point encoding them
- Temporal masking: Human ears can miss soft sounds immediately after loud sounds, takes time for the ear to adjust, no need to store them either

An Example Format: MP3

- MP3 is MPEG Audio Layer 3
- MP3's compression is **based on perceptual coding**
- MP3 audio compression results in significant **file size savings without a perceived loss of audio quality**
- Typical MP3 audio compression rates for CD quality audio reduce the need for bandwidth **from 1.4Mbps for stereo down to 96-128Kbps**

For Digital Video

- Video is digitized as pixels
 - TV quality: 640x480 pixels, 24-bit color, 30 times/sec
~ 200Mbs uncompressed
- Video is sent compressed due to its large bandwidth requirements
 - Lossy compression exploits human perception
 - E.g., JPEG for still images, MPEG for video
 - Large compression ratios achieved (often 50X for video)

Compression with JPEG

- JPEG lossy compression
- JPEG often provides compression ratios of 20:1
- JPEG compression is **symmetric, decoding takes as long as encoding**
- This is not the case in all types of compression

MPEG

- MPEG - Motion Picture Experts Group
- MPEG can compress both audio and video together
- The evolution of MPEG:
 - MPEG-1: VCR quality at 1.2 Mbps (40:1)
 - MPEG-2: Broadcast quality at 4-6Mbps (200:1)
 - MPEG-4: DVD quality at 10Mbps (1200:1)

Week 10: Network Security

Internet Technologies COMP90007

Lecturer: Muhammad Usman

Semester 2, 2020

What is Network Security?

- Network security is a combo of 4 related areas:
 - **Secrecy** (Keeping information hidden from a general audience)
 - **Authentication** (Ensuring the user you are giving content to has valid credentials)
 - **Non-repudiation** (Prove a content was created by a named user)
 - **Integrity control** (Ensure that a content has not been tampered with)
- All of these are **equally valid** and has been around for all systems for some time, but have different and sometimes more challenging implications in a networked environment
- Aspects of security can be found at all layers of a protocol stack, **there is no way to secure a network by building security into one layer only**
- Most security implementations are **based on common cryptographic principles** and appear on almost all layers

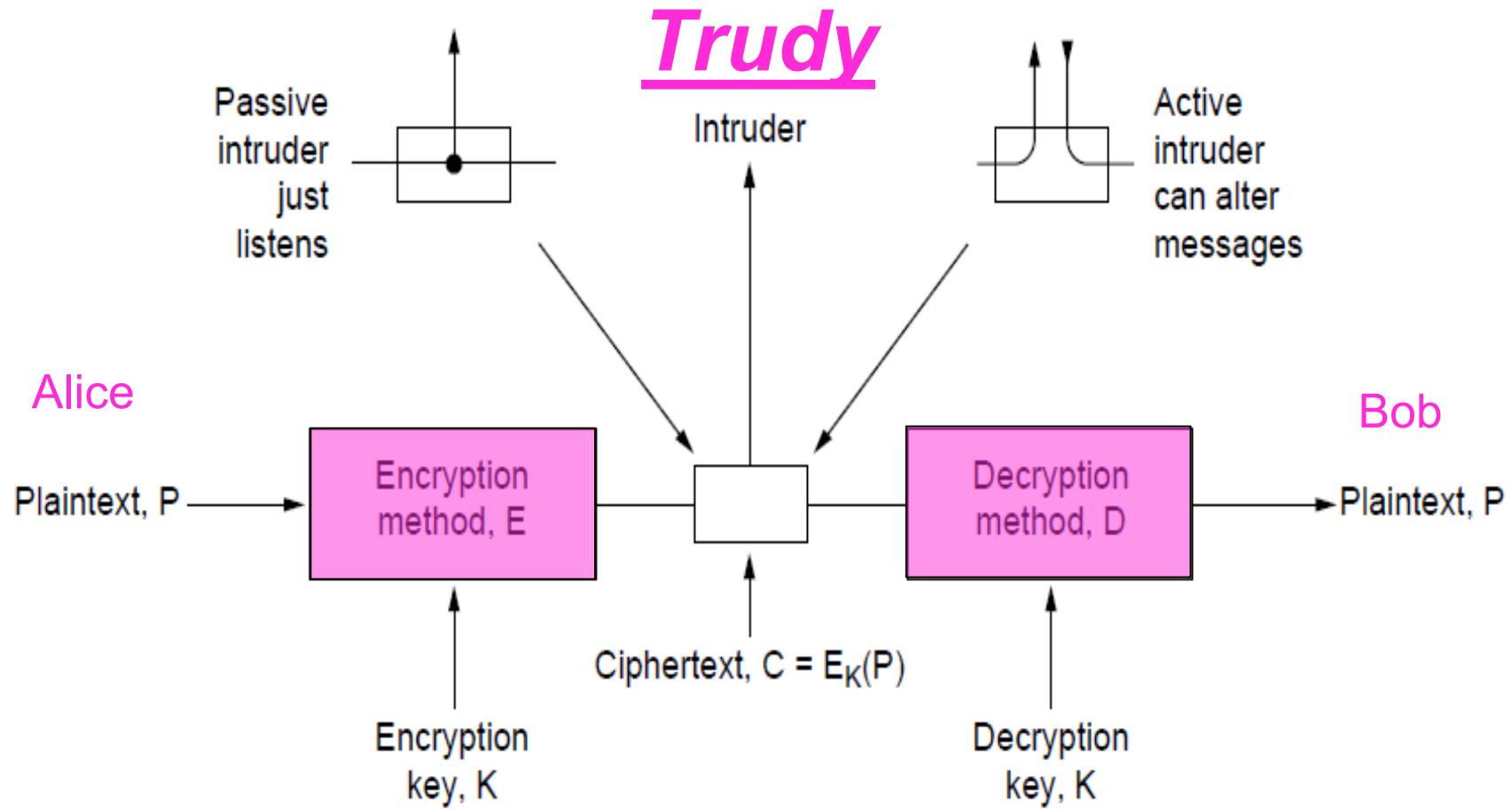
Some Well-Known Characters..

Adversary	Goal
Student	To have fun snooping on people's email
Cracker	To test out someone's security system; steal data
Sales rep	To claim to represent all of Europe, not just Andorra
Businessman	To discover a competitor's strategic marketing plan
Ex-employee	To get revenge for being fired
Accountant	To embezzle money from a company
Stockbroker	To deny a promise made to a customer by email
Con man	To steal credit card numbers for sale
Spy	To learn an enemy's military or industrial secrets
Terrorist	To steal germ warfare secrets

Cryptography

- A key area/set of algorithms for creating secrets, authenticating users, making sure messages are not tampered with, and edits are not denied by the original author....

Encryption Model



Key Cryptography Concepts

- Three foundations:
 - Plaintext
 - Keys
 - Ciphertext
- **Plaintext** messages to be encrypted can be transformed (encrypted/decrypted) by a function that is parameterized by a **key**, the output of the transformation process is **ciphertext**
- **Kerckhoff's principle:** Cryptographic Algorithms and related functions (E , D) are public; only the keys (K) are secret

A Simple Example

- Key/Method for Encryption: this function is a very simple one, transform every character to the next one in the alphabet for a given plaintext (and “z” becomes “a” to circle around the end)
- Input plaintext
 - “where”
- Output ciphertext
 - “xifsf”
- Decryption is the simple method to go back in the alphabet to reverse the effect of encryption

The Notation

- C = ciphertext, P = plaintext, E = encryption, D = decryption, K = key
- $C = E_K(P)$
- $P = D_K(C)$
- $D_K(E_K(P)) = P$
- In fact what we want in simple crypto-based network security is efficient methods where
$$D_{K1}(E_{K2}(P)) = P \text{ if and only if } K1=K2.$$

Keys Plays an Important Role

- A key is a string that allows the selection of one of many potential encryptions
- The key can be changed as often as required
- Algorithms are more likely to be at the hands of attackers anyhow, not changed as frequently
- Cipher is a term commonly used as the term for algorithm here
- The size of the overall key space is determined by the number of bits in the key string
- The longer the key, the more effort is required to break a given encryption

A common function used in ciphers: Recall XOR

- An XOR is an “exclusive or” function used regularly.
- A XOR B means A or B, but not both
- XOR is commonly used in cryptography

A	B	A XOR B
F	F	F
F	T	T
T	F	T
T	T	F

Truth values	Binary Equivalents
T	1
F	0

Some Main Types of Ciphers

- Substitution cipher
 - Each letter or group of letters is replaced systematically by other letters or groups of letters (our example)
- Transposition cipher
 - All letters are re-ordered without disguising them
- One-time pad
 - Uses a random bit string as the key: convert the plaintext into a bit string, then XOR the two strings bit by bit
 - Harder to break
 - How to share the random key and its size are factors

Example revisited: Substitution cipher example

- Substitution ciphers replace each group of letters in the message with another group of letters based on a key with an intention to disguise the message.

plaintext:	a b c d e f g h i j k l m n o p q r s t u v w x y z
ciphertext:	Q W E R T Y U I O P A S D F G H J K L Z X C V B N M

- If “were” was the ciphertext received then it becomes

“bcdc”

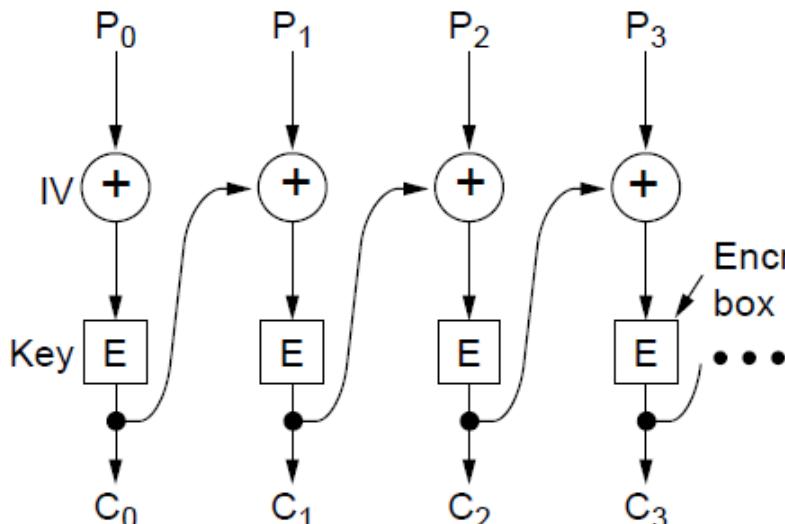
when decrypted

Modern Key-based Algorithms

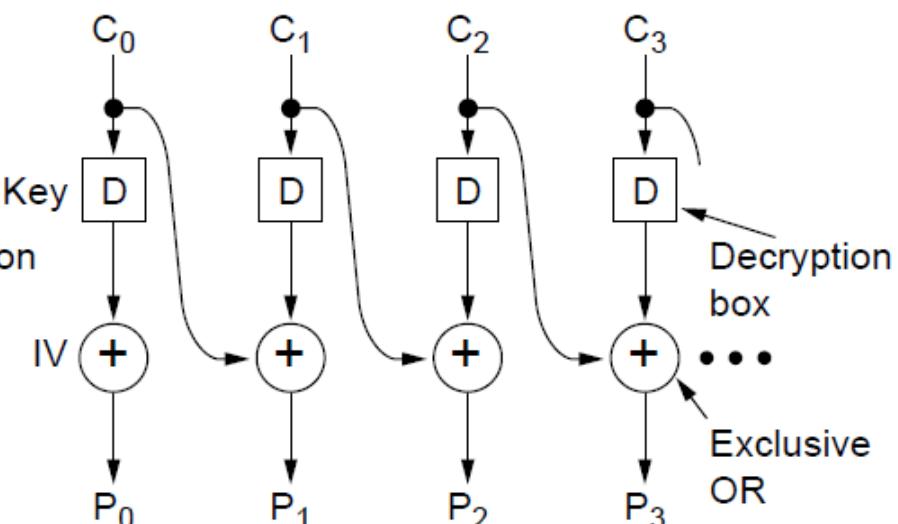
- **Two main categories**
- **Symmetric key algorithms** use the same key for both encryption and decryption
- Symmetric key algorithms can use permutation, substitution and a combination of both to encrypt and decrypt
- **2 Symmetric Key Algorithms**
 - Data Encryption Standard (DES)
 - Uses 64 bit blocks and 56 bit keys
 - 2^{56} key space
 - Triple DES has a 3×2^{56} key space
 - Advanced Encryption Standard (AES) in use since 2000s
 - Uses 128 bit blocks and 128 bit keys
 - 2^{128} key space
 - Still substitution and permutation based with multiple rounds

Cipher Block Chaining Mode

- Same text leads to same ciphertext unless something else is done, thus:
- In block chaining mode, each plaintext block is XOR'ed with the previous ciphertext block before being encrypted
- (a) encryption, (b) decryption



CBC mode encryption



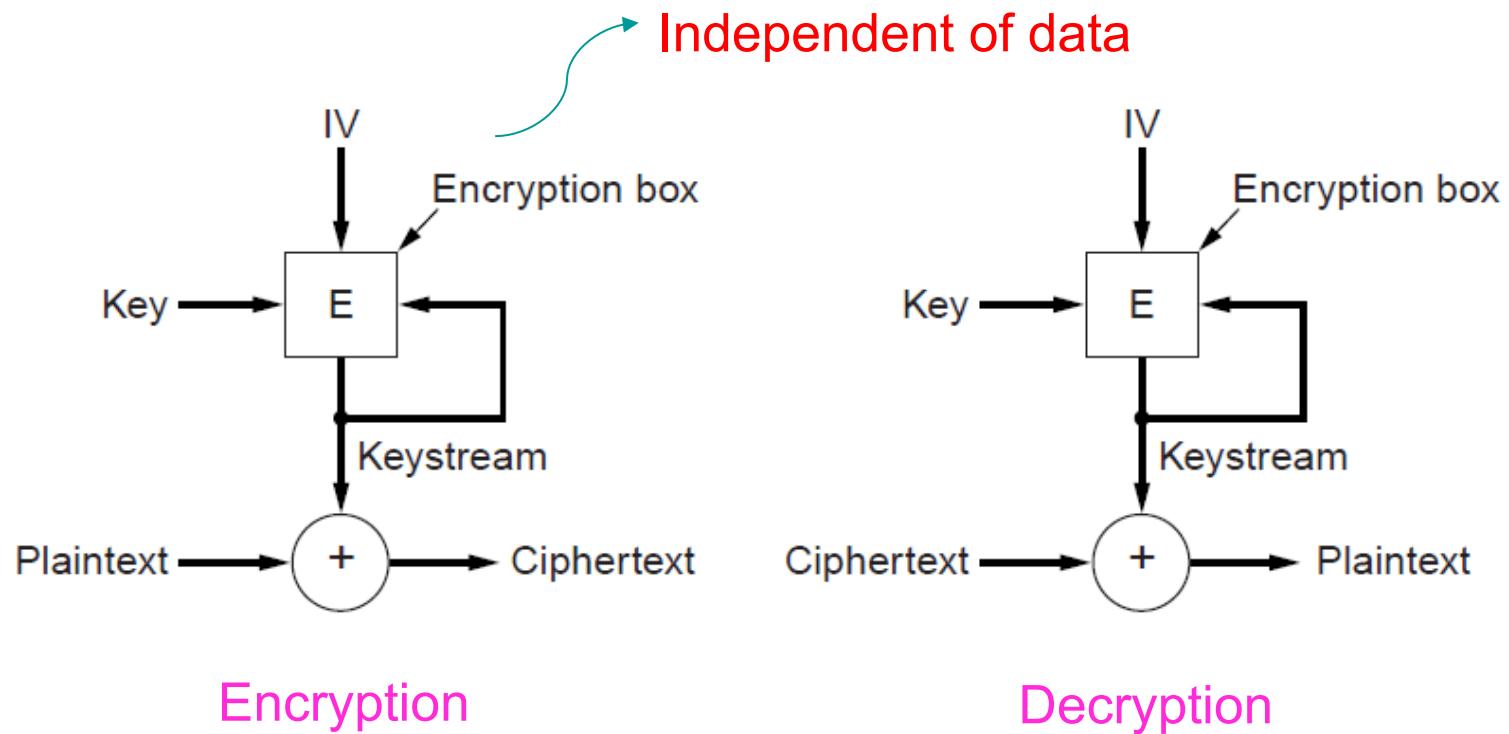
CBC mode decryption

Cipher Feedback Mode

- In cipher feedback mode, byte-by-byte encryption is used rather than block-by-block encryption
- Good for things like encrypting someones key strokes on a keyboard
where a lot of data is not immediately available

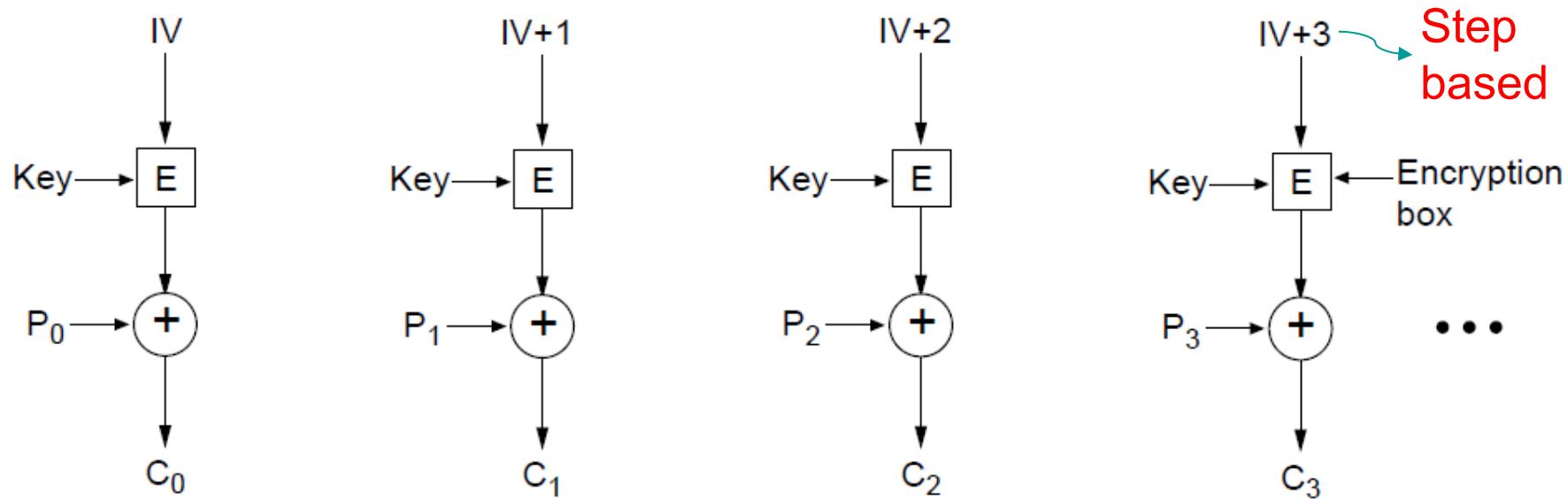
Stream Cipher Mode

- What if data transmission errors occur? In stream cipher mode, recursive sequential block encryption is used as a one-time pad, and XOR'ed with plaintext to generate ciphertext



Counter Mode

- But how about random access to data after encryption?
- In counter mode, plaintext is not directly encrypted, but an initialisation parameter plus an arbitrary constant is encrypted, and the resulting ciphertext is XOR'ed with plaintext



Many Symmetric Key Algorithms Exist

Cipher	Author	Key length	Comments
Blowfish	Bruce Schneier	1–448 bits	Old and slow
DES	IBM	56 bits	Too weak to use now
IDEA	Massey and Xuejia	128 bits	Good, but patented
RC4	Ronald Rivest	1–2048 bits	Caution: some keys are weak
RC5	Ronald Rivest	128–256 bits	Good, but patented
Rijndael	Daemen and Rijmen	128–256 bits	Best choice
Serpent	Anderson, Biham, Knudsen	128–256 bits	Very strong
Triple DES	IBM	168 bits	Second best choice
Twofish	Bruce Schneier	128–256 bits	Very strong; widely used

COMP90007 Internet Technologies

Week 10 Workshop

Semester 2, 2020

Suggested solutions

Question 1

Why does UDP exist? Would it not have been enough to just let the user processes send raw IP packets?

Answer:

No. IP packets contain IP addresses, which specify a destination machine. Once such a packet arrived, how would the network handler know which process to give it to? UDP packets contain a destination port. This information is essential so they can be delivered to the correct process.

Question 2

Both UDP and TCP use port numbers to identify the destination entity when delivering a message. Discuss possible reasons for why these protocols invented a new abstract ID (port numbers), instead of using process IDs (which already existed when these protocols were designed?)

Answer:

Here are three reasons.

- First, process IDs are OS-specific. Using process IDs would have made these protocols OS-dependent.
- Second, a single process may establish multiple channels of communications. A single process ID (per process) as the destination identifier cannot be used to distinguish between these channels.
- Third, having processes listen on well known ports is easy, but well-known process IDs are impossible.

Question 3

What is the key difference between TCP Tahoe and TCP Reno?

Answer:

- The key difference, and benefit of Reno, is that Reno avoids Slow start when it can and can do Fast recovery at certain cases.

Question 4

Recall the Leaky Bucket algorithm we saw in class. If a sender has a burst data rate of 20KB/s for 20 seconds as data to send and we have a bucket size of 100KB with a output rate of 10KB/s: Does the Leaky Bucket algorithm achieve its aim of regulating output properly? If so explain how and show your calculations. If not, find the appropriate bucket size and discuss why we need more/less of a bucket size.

Answer: The bucket is too small, we should have had a bucket size of 200KB. The input data is $20 \times 20 = 400\text{KB}$, in the same time the bucket can empty only $20 \times 10 = 200\text{KB}$ and can hold another 100KB. So another 100KB bucket size is needed to deal with 400KB in total.

Question 5

TCP relies on timers for resending in case some ACKs are missing. If we set such timers to a fixed value of say 100ms, discuss what would be the advantages and disadvantages of such a static protocol design.

Answer: A fixed 100ms timer is easy to implement and does not require monitoring network status and load. If conditions are stable and 100ms is set accordingly then all should work fine with little overhead. However, in most cases, network conditions change. Thus 100ms could lead to prematurely timers going off, which means many redundant resends, especially when network is busy... When network conditions indicate fast traffic, then the opposite would be true, i.e., static timers will go off too late for missing segments and applications would wait extra for no good reason. In case of dynamic conditions in the network, timers should be set based on measurements rather than to a static value.

Week 11: Network Security

Internet Technologies COMP90007

Lecturer: Muhammad Usman

Semester 2, 2020

Public Key Algorithms

- Fundamentally different to symmetric key ones
- Diffe & Hellman proposed the new model
 - **Asymmetric keys**
 - **Two keys are used**
 - Not easily derivable from each other
 - Hence addressing a fundamental issue of key sharing

Asymmetric Keys

- Diffe-Hellman key system
 - **Key 1: public key**, usable by anyone **to encrypt** messages to the owner of the key, this key known to all
 - **Key 2: private key**, required **to decrypt** the message and known only by the owner of this key

The Process

- $C = \text{ciphertext}$, $P = \text{plaintext}$, $E = \text{encryption}$, $D = \text{decryption}$
K1, K2 = keys
- $C = E_{K1}(P)$
 - Sender knows the public key $K1$ and the P
- $P = D_{K2}(C)$
 - Only receiver knows private $K2$ which can undo $K1$'s effect
- $D_{K2}(E_{K1}(P)) = P$

RSA: An Asymmetric Key Algorithm

- **RSA - Rivest, Shamir, Adleman**
- Famous and robust algorithm
- Key generation:
 - Choose two large primes, p and q
 - Compute $n = p \times q$ and $z = (p - 1) \times (q - 1)$.
 - Choose d to be relatively prime to z, i.e., no common factors
 - Find e such that
 - **$(d \times e) \bmod z = 1$**
 - Public key is (e, n), and private key is (d, n)
- Encryption:
 - $\text{Cipher} = \text{Plain}^e \pmod{n}$
- Decryption:
 - $\text{Plain} = \text{Cipher}^d \pmod{n}$

RSA Example

- Let $p=3, q=11$: then z is $(3 - 1) \times (11 - 1) = 20$
- What is a potential d ?
- If $d = 7$ then 7 and 20 has no common factors
- What is an e ?
- If $e = 3$, then $(d \times e)$ is 1 in mod z
- What are the two key tuples then?
- Enc: 3, 33 Dec: 7, 33 (as $n=3 \times 11=33$ and $d=7$ and $e=3$)

Plaintext (P)		Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	$P^3 \pmod{33}$	C^7	$C^7 \pmod{33}$
S	19	6859	28	13492928512	19
U	21	9261	21	1801088541	21
Z	26	17576	20	1280000000	26
A	01	1	1	1	01
N	14	2744	5	78125	14
N	14	2744	5	78125	14
E	05	125	26	8031810176	05

Encryption: $C = P^3 \pmod{33}$

Decryption: $P = C^7 \pmod{33}$

S is the 19th character in the alphabet...

RSA Security

- RSA's security is based on the difficulty involved in factoring large numbers in math theory - approx 10^{25} years to factor a 500 digit number and RSA uses 1024 bits!
- Disadvantage: RSA is too slow for encrypting/decrypting large volumes of data

RSA Security Contd

- ...but is widely used for many other things such as **secure key distribution**
- Then RSA can be used in tandem with symmetric key algorithms...

Another Use of Cryptography: Digital Signatures

- Cryptographic approaches can also be used to ensure **authenticity** and allow for **non-repudiation**
- Requirements
 - Receiver can **verify the claimed identity of the sender**
 - **Sender cannot deny she created** contents of the message
 - **Receiver cannot have derived the message themselves**

Digital Signatures

■ Approaches

- Using symmetric keys via an intermediary
 - You need a BIG BROTHER to do all the messaging, not good!
- Using public keys as individuals

Using Public Keys

- Sender Alice uses *private key on P*
- *Receiver Bob uses her public key to undo and get P*
- RSA can do this as well, as *E(D(P)) = P in RSA*

- Alice cannot deny signing as she only knows her private key

Signatures with Message Digests

- Basic concept of a message digest is to use a one-way hash function for an arbitrary length of plaintext, so that it becomes a "unique" small fixed-length bit string
- Thus no need to deal with huge message text and encryption just for authentication purposes
- A message digest (MD) has four important properties:
 - 1 Given P, it is easy to compute $MD(P)$
 - 2 Given $MD(P)$ it is effectively impossible to find P
 - 3 Given P, no one can find P' such that $MD(P') = MD(P)$
 - 4 A change in even a single bit of input produces a very different output

Famous Message Digest Algorithms

- MD5
- SHA-1
- Outputs
 - Given "this is a test" (text could have been longer)
 - MD5:
e19c1283c925b3206685522acfe3e6
 - SHA-1:
6476df3aac780622368173fe6e768a2edc3932c8

Public Key Management

- There is specific PK infrastructure to avoid compromising the security of PK's during the initial distribution process.
- Certification Authority (CA)
 - A trusted intermediary who uses non-electronic identification to identify users prior to certifying keys and certificates
- X.509
 - An international standard for certificate expression
- PKI (Public Key Infrastructure) is a
 - Hierarchically structured certificate authorities allow for the establishment of a chain of trust or certification path
 - *Verisign* was such a company

Certificate Issuing

- A Certificate authority (CA) says:

I hereby certify that the public key

19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A

belongs to

Robert John Smith

12345 University Avenue

Berkeley, CA 94702

Birthday: July 4, 1958

Email: bob@superduper.net.com

SHA-1 hash of the above certificate signed with the CA's private key

Week 11: Network Security

Internet Technologies COMP90007

Lecturer: Muhammad Usman

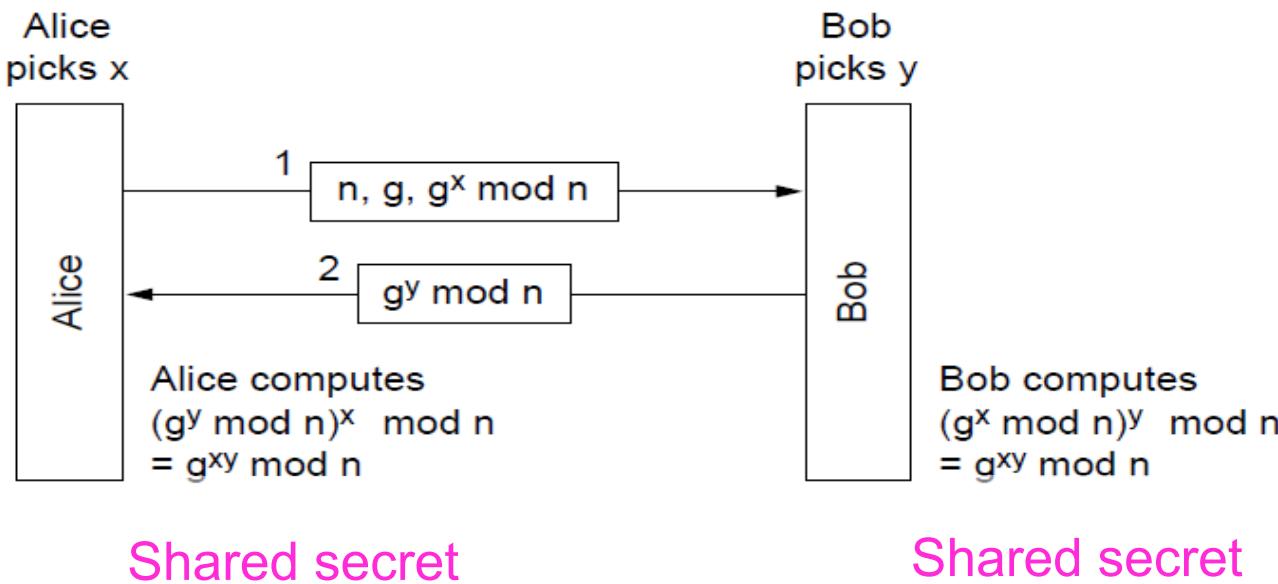
Semester 2, 2020

Authentication

- **Authentication is a primary tenet** of network security
- However, **authentication process itself needs to be secure** also
- A fundamental principle: **minimise the use of permanent keys in establishment of secure connections** (the less packets are exchanged using such keys, the less exposure to potential attackers)
- Four methods in common use:
 - Shared keys
 - Key distribution
 - Kerberos
 - Public keys

Authentication Based on a Shared Secret Key

- How to create a key with Diffie-Hellman key exchange:



Is there a way to break this? Still open to man-in-the-middle attack!

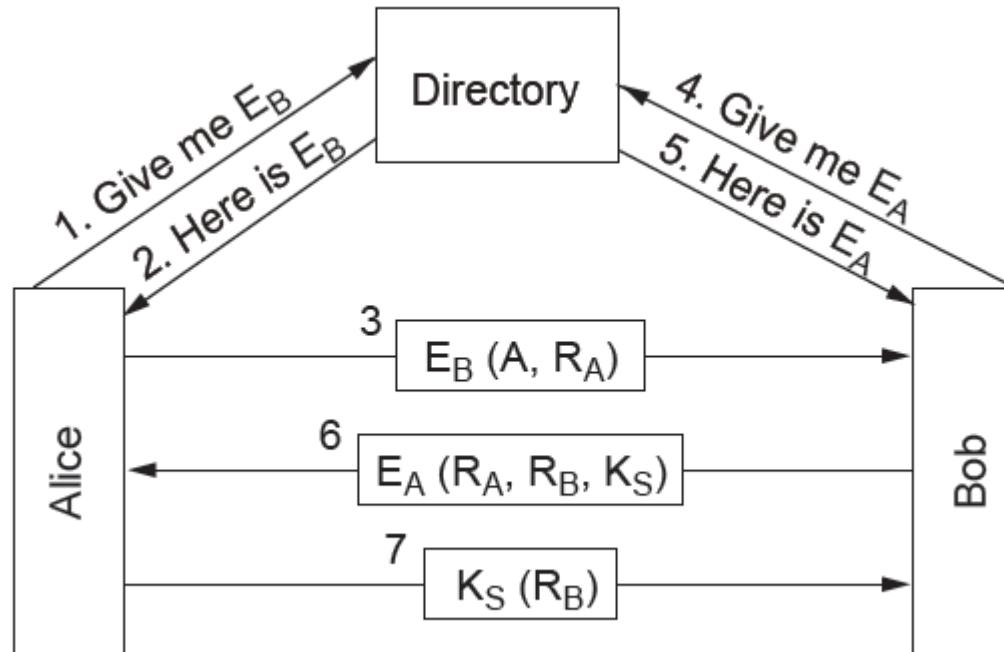
Authentication Using a Key Distribution Center

- In this method, **a trusted intermediary is used** to facilitate
- Users each share a key with a central key distribution centre, and authenticate to the KDC directly
- The KDC acts as a relay between the two parties
- There are issues here as well:
 - Open to **replay-attack**
- Solutions exist to patch the KDC mechanism
 - E.g. timestamps

Authentication Using Kerberos

- Similar to KDC a popular protocol emerged and in frequent use today: Kerberos
- In this method, a multi-component system is required
 - Authentication Server
 - Ticket Granting Server (TGS)
 - Recipient
- Authentication is managed centrally, and then party to party communication is facilitated by single use tickets
- Still disadvantages remains: Does not scale to large numbers; different businesses need to trust each other's TGSs...

Authentication Using Public Key Cryptography



IPSec

- Where to put security?
 - Some say application layer: but users may not want such things
 - Some say lower layers: but not as strong as having it at app layer
 - Outcome is **security can/should be in multiple layers**
- One can put security at application level but also...
- **IPSec (RFC 2401,..) puts it at the network level** as well
- In the IPSec model, **encryption is compulsory, but a null encryption algorithm can be used** between points
- The main IPSec framework features are **secrecy, data integrity, and replay** attack protection
- The IPSec framework allows multiple algorithms and multiple levels of granularity, connection-oriented (**connections are named as SA's security associations**)

IPSec Implementation

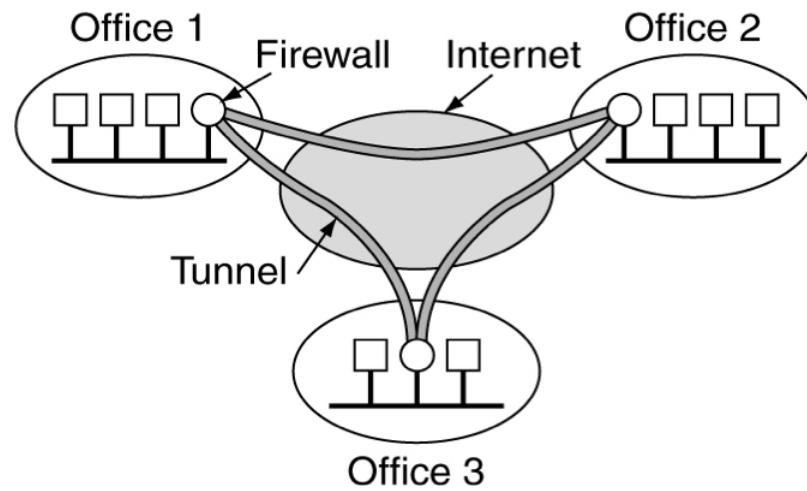
- IPSec has two main implementation components
 - Things being added to packets in transit
 - ISAKMP key management: Internet Security Association and Key Management Protocol for establishing keys
- IPSec has 2 modes
 - Transport mode - uses header insertion after IP Header
 - Tunnel mode - uses packet encapsulation

Virtual Private Networks

- Unlike a physical network based on leased lines between locations for which secure transit is required
- A Virtual Private Network (VPN) is a virtual layer on top of an IP network which provides a secure end-to-end connection over public infrastructure
- A common VPN implementation model:
 - Use a firewall at each end of a connection
 - Setup a SA to create an IPSec tunnel between the two end points
- Communication on this infrastructure is transparent to end users

VPN

A virtual private network



Firewalls

- While IPSec ensures security in transit, a **firewall ensures security at the network perimeter**
- Firewalls are positioned at the network boundary, and **provide a controlled series of routes between the internal and external networks**
- Three characteristics of firewalls
 - All **inbound and outbound** traffic must transit the firewall
 - Only **authorised traffic** must pass through the firewall
 - Firewalls should be **immune to penetration** themselves

Firewall Scope

- Check packets for “bad” packets
 - Administrators can **write rules for this**, e.g., distinguish regular HTTP from P2P related HTTP
- **Not everything is inside the wall**
- Web servers and email servers etc **need to be exposed to allow more open communication**
 - Best firewall is NOT disconnecting everything from the Internet
- Through **further rules packets go in-between this gray area and the LAN**
- Firewalls don't provide protection against inhouse threats
- Applications can still distribute viruses (via bad attachments for example)

Wireless Security Context

- Wired networks are relatively easy to secure because they require physical access to intercept traffic
- Wireless networks are more difficult to secure because of **omnidirectional signal propagation**
- Additionally by default **most wireless network equipment operates in an insecure and promiscuous manner**
- 802.11 has a native secure protocol, **Wired Equivalency Protocol** (WEP), which is a 40-bit encryption based on RC4 algorithm

Wireless Security Issues

- Two inherent insecurities
 - 40 bit encryption is breakable with low-moderate computational resources
 - RC4 re-uses keys, so capturing a small volume of encrypted traffic will guarantee key identification
- Given these constraints, how can wireless networks be secured?

Securing Wireless

- Additional encryption (128bit WEP)
 - Increased security through longer key lengths
- MAC Address Filtering
 - Only allow specified MAC interfaces to establish connections
- ...
- WPA2 (WiFi Protected Access 2)
- ...
- Multilayered security
 - Use a VPN over wireless

COMP90007 Internet Technologies

Week 11 Workshop

Semester 2, 2020

Suggested solutions

Question 1

What are the disadvantages of having only one central DNS server that serves all machines connected to the internet?

Ans. Some of the disadvantages of a single DNS are:

- Single point of failure
- Traffic congestion at server
- Distant centralised server for remote queries
- Maintenance issues, not only for keeping large amount of data upto date but also the prospect of simple service maintenance could cause big disruptions.
- May not be able to service all queries fast enough, also scaling on the computation front may be an issue.

Question 2

What does iterative mode of execution when querying a DNS mean? Where is it used? What is the recursive mode? Please explain with an example.

Ans. Iterative queries are requests that are propagated from one name server to another, gathering partial results in the form of which name server might know the location of the authoritative record till we reach that location. At which point, we return the final answer (Resource record mapping) to our end user who requested the domain initially, this is referred to as recursive, where we return the final answer only to the end user and not partial answers. Recursive queries are when a local PC/device delegates to local server the DNS query to recursively follow-up the query with other servers in the DNS system.

Question 3

What is non-persistent HTTP connections? Explain with an example request.

Ans. Sessions where only one object/response is returned and a new connection is established for every response to be provided.

Question 4

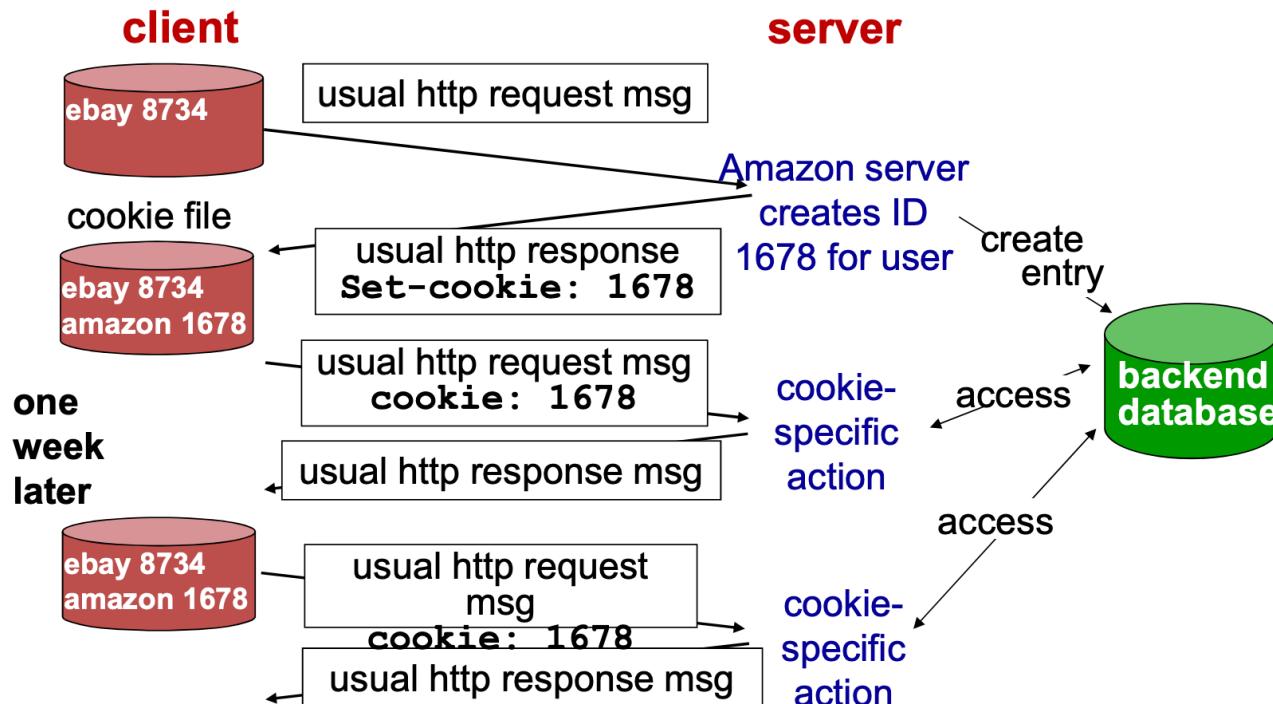
What are the benefits of a persistent HTTP connection?

Ans. A connection established remains open for some time after the transmission of an object, allowing multiple objects to be sent using the same connection as opposed to tearing down an existing connection and recreating it for every response to be sent.

Question 5

Give an example execution of an HTTP request with a cookie being used?

Ans.



Question 6

Web Caching can cause problems such as stale data being served to clients. What are the benefits of web caching?

Ans.

- Reduced response time to distant servers.
- Reduced traffic to congested servers.

Question 7

What is perceptual coding in terms of compressing media to deliver data over the internet? Give two examples?

Ans. Perceptual coding is that some media content such as audio can be coded into digital form without loss of any perceived quality. For example some sounds can mask other sounds for human hearing and at that point those sounds that are identified can be used to reduce the data size to be transmitted. For example -

Frequency masking: Some sounds at certain frequencies can mask/hide others so there is no point encoding the ones humans cannot hear.

Temporal masking: Human ears can miss soft sounds immediately after loud sounds, takes time for the ear to adjust, so no need to put these in the compressed data as well.

Question 8

What is SMTP protocol and where is it used?

Ans. SMTP is an application layer protocol for mail transfer. It is used from the user agent to the MTA and between MTAs.

Question 9

What are the two missing layers of the OSI protocol that we did not see in the Internet so far? Give one service for each.

Ans. Presentation and Session Layer.

Services can be: formatting, encryption, compression for presentation layer and authentication, authorization, session management for the Session layer.

Networking Applications of Artificial Intelligence and Data Mining

Chris Leckie

Artificial Intelligence

Definition: developing computer systems that can perform tasks that traditionally can only be done by a human

For example:

- Playing a good game of chess
- Self-driving car
- Translating spoken English into spoken Spanish in real-time
- Detecting that a user's account has been hacked

What types of intelligent behaviour are needed in these applications?

**Today, we'll focus on one major area
of Artificial Intelligence:**

Data Mining and Machine Learning

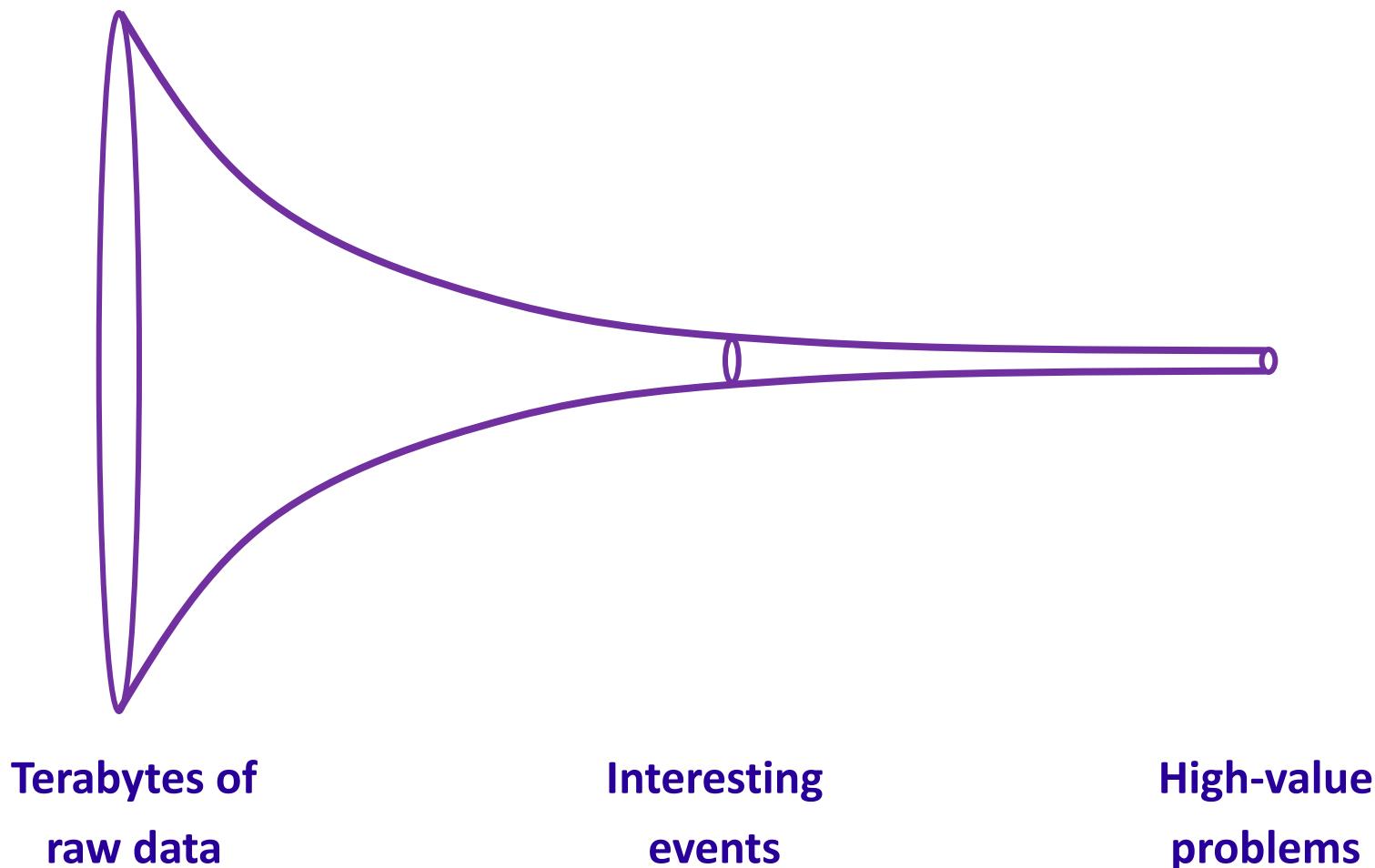
Overview

Data mining / Machine Learning aim to find useful patterns in large data sets

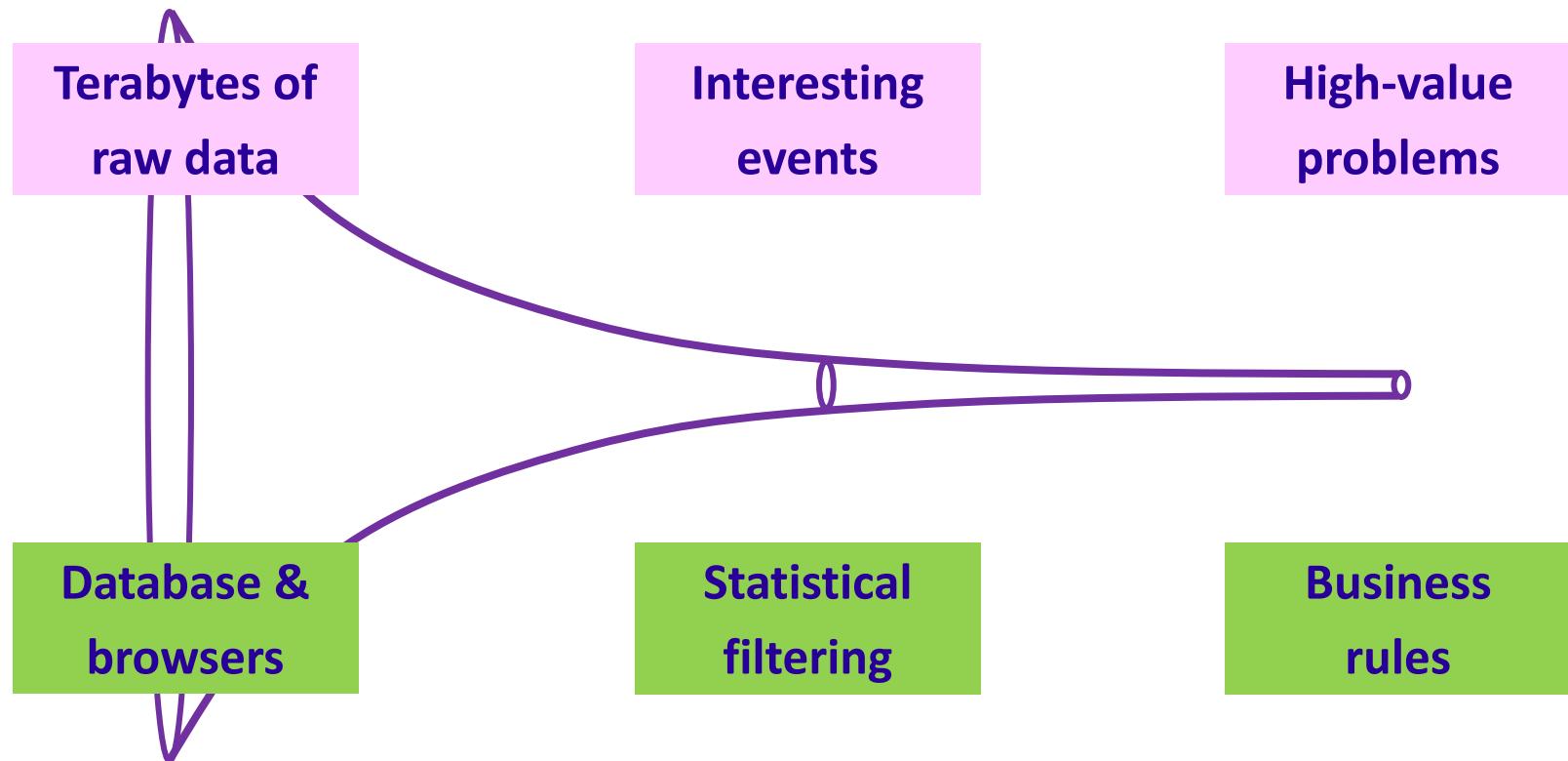
For example:

- **Market segmentation studies**
 - Find categories of customers with similar buying behaviour
- **Predictive modelling**
 - Find customers who are likely to commit fraud based on their transaction history

The Common Theme – Big Data



Automating the Data Analysis Pipeline



Part of the field of **data mining / machine learning**

Types of Learning Problems in Data Mining / Machine Learning

Supervised Learning:

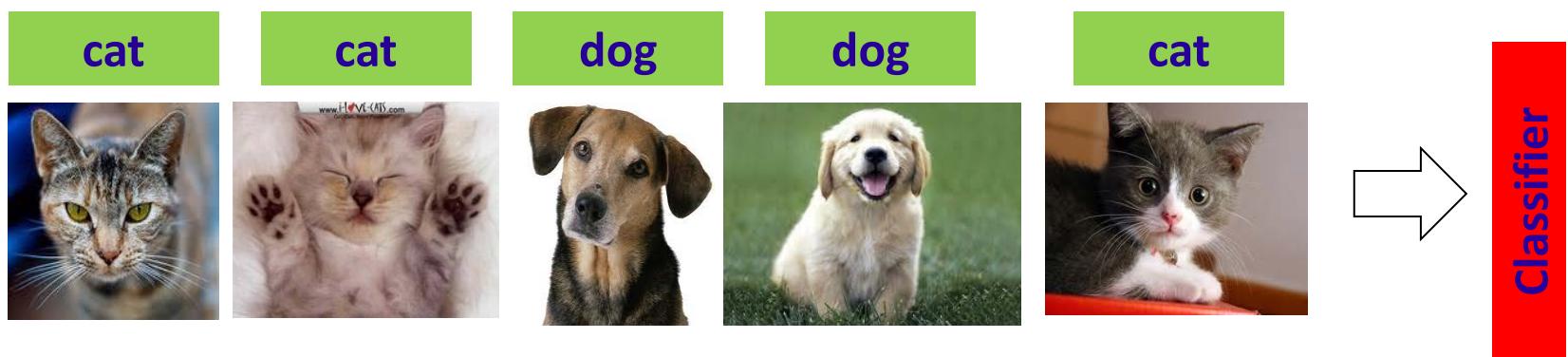
Learn a classifier from a set of labelled examples so that you can classify new unlabelled examples in the future

Unsupervised Learning:

Cluster a set of unlabelled examples to learn the natural categories or types of objects

Learning a Classifier (Supervised Learning)

Training a classifier



Classifying new examples



Clustering to Learn Categories (Unsupervised Learning)

What are the natural categories in a database?



Consider a database of animals.

How many different types of animals are there here?



Examples of Applications of Data Mining

Supervised Learning:

- Fraud detection from credit card transactions
- Face recognition in Facebook
- Diagnosing cancer from genetic test on blood samples

Unsupervised Learning:

- Modelling different types of network traffic (web, video, music, etc)
- Building an index of the types of documents on a web site
- Identifying different categories of customers on a retail website

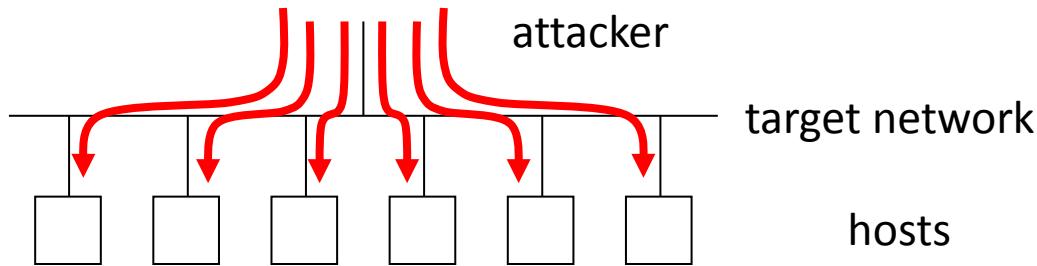
Learning Unusual Patterns (Anomaly Detection)

- Learn a model of “normal” database records
- Use this model to test new records for anomalies
- Any anomalies can be either interesting or errors

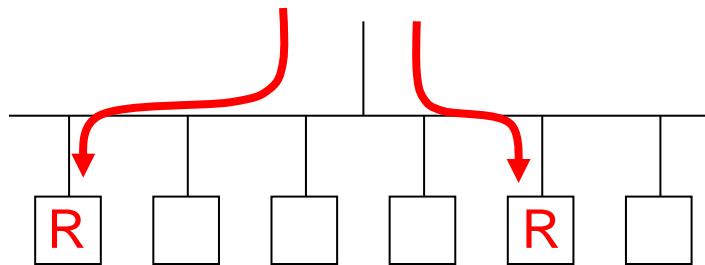
Example of Machine Learning in Cyber Security

Examples of Network Intrusion

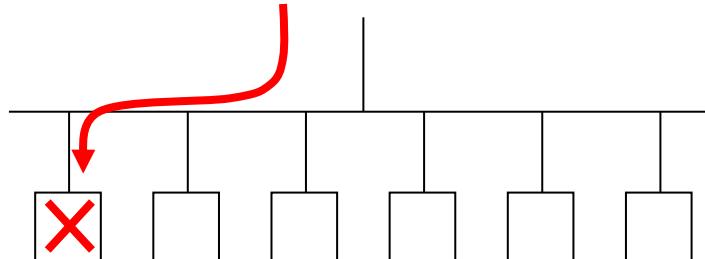
Probe for hosts with known weaknesses



Gain root access to hosts

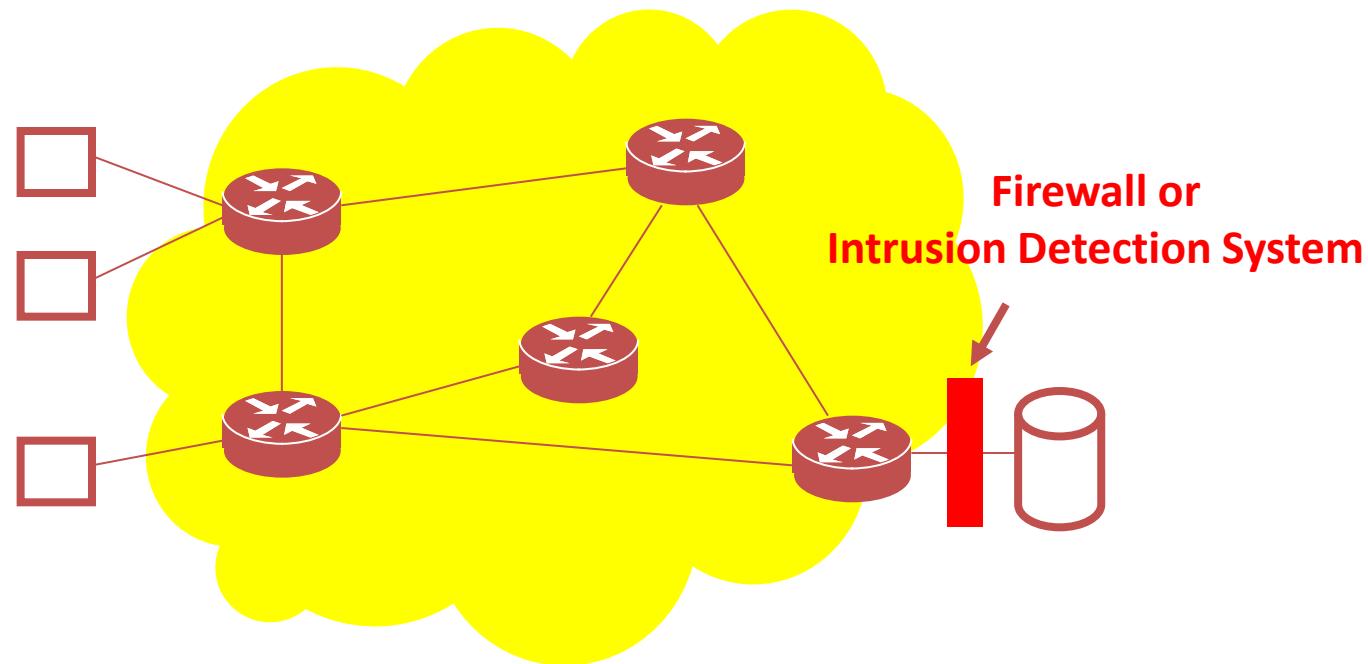


Denial-of-service attack using malformed packets



Existing Approaches to Defend Against Network Attacks

Write rules to detect *known* types of attack



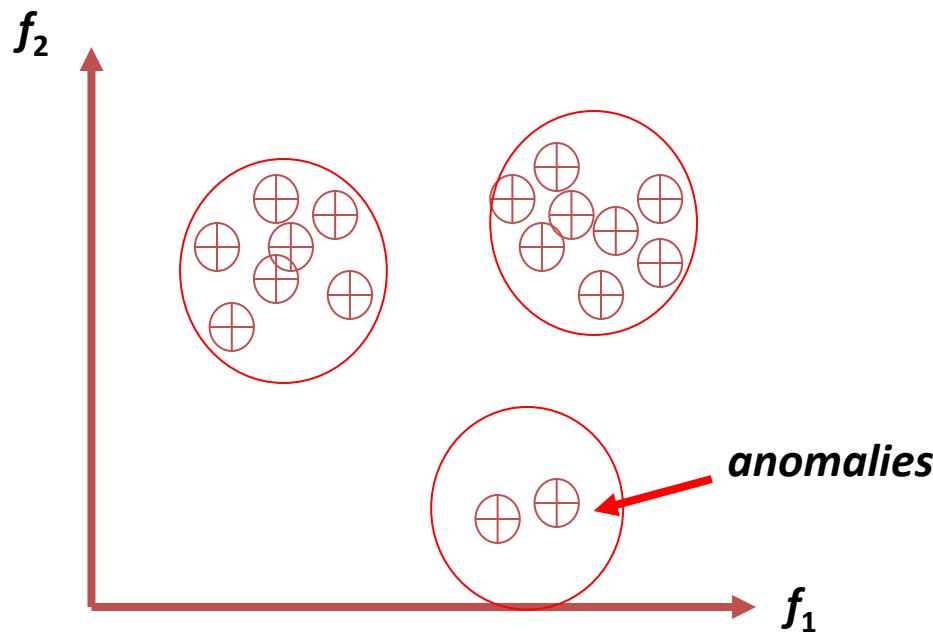
Drawback: unable to detect new attacks

Alternative Approach: Anomaly Detection

- Learn a model of “normal” traffic
- Use this model to test new traffic for anomalies
- Any anomalies are treated as an attack

Cluster-based Anomaly Detection

- Map network connections into a feature space $\{f_1 \dots f_k\}$
- Cluster similar connections
- Use large clusters to represent normal traffic



Challenge: changing traffic patterns cause false alarms

Summary

How would you define Artificial Intelligence (AI)?

What are some example applications of AI?

**What is the difference between supervised
and unsupervised learning in data mining / machine learning?**

What are some example applications of data mining?

What is anomaly detection?

How can anomaly detection be used in network security?

Week 12: Revision Lecture

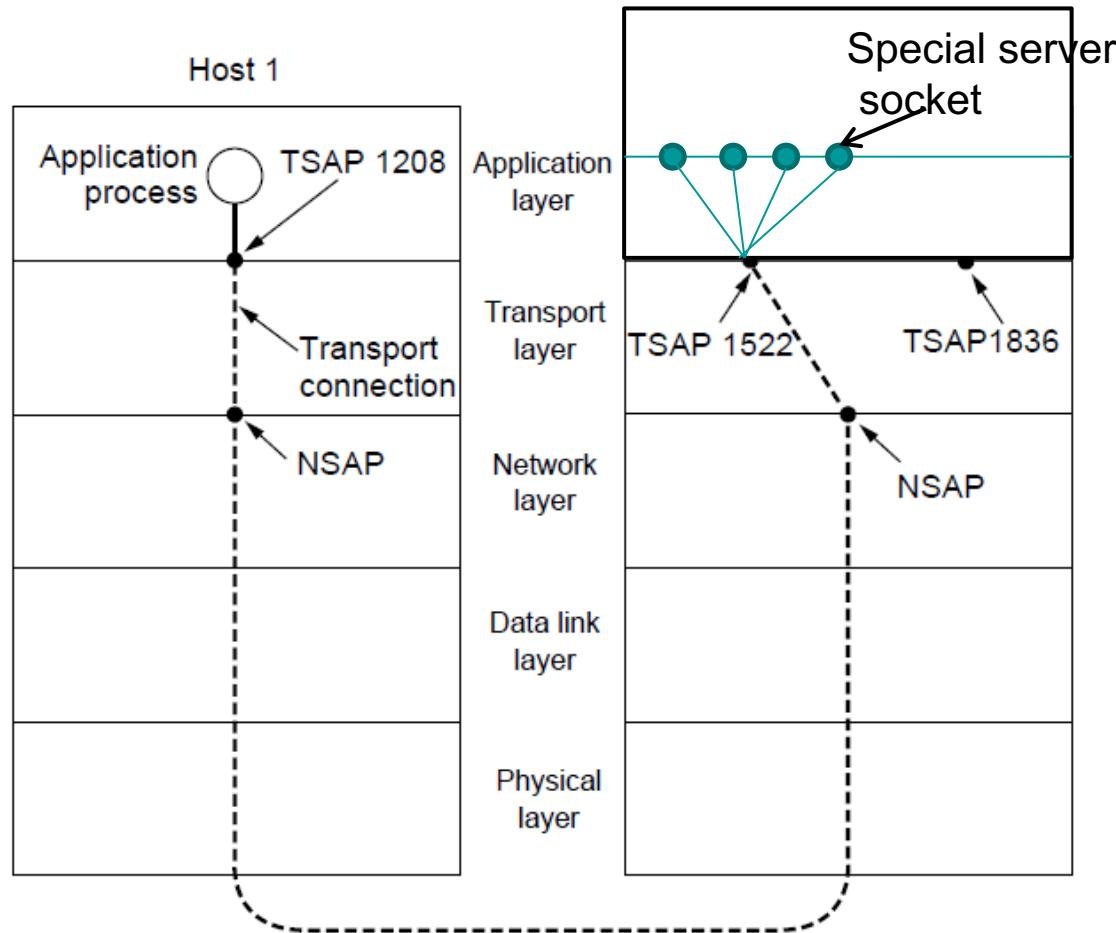
Internet Technologies COMP90007

Lecturer: Muhammad Usman

Semester 2, 2020

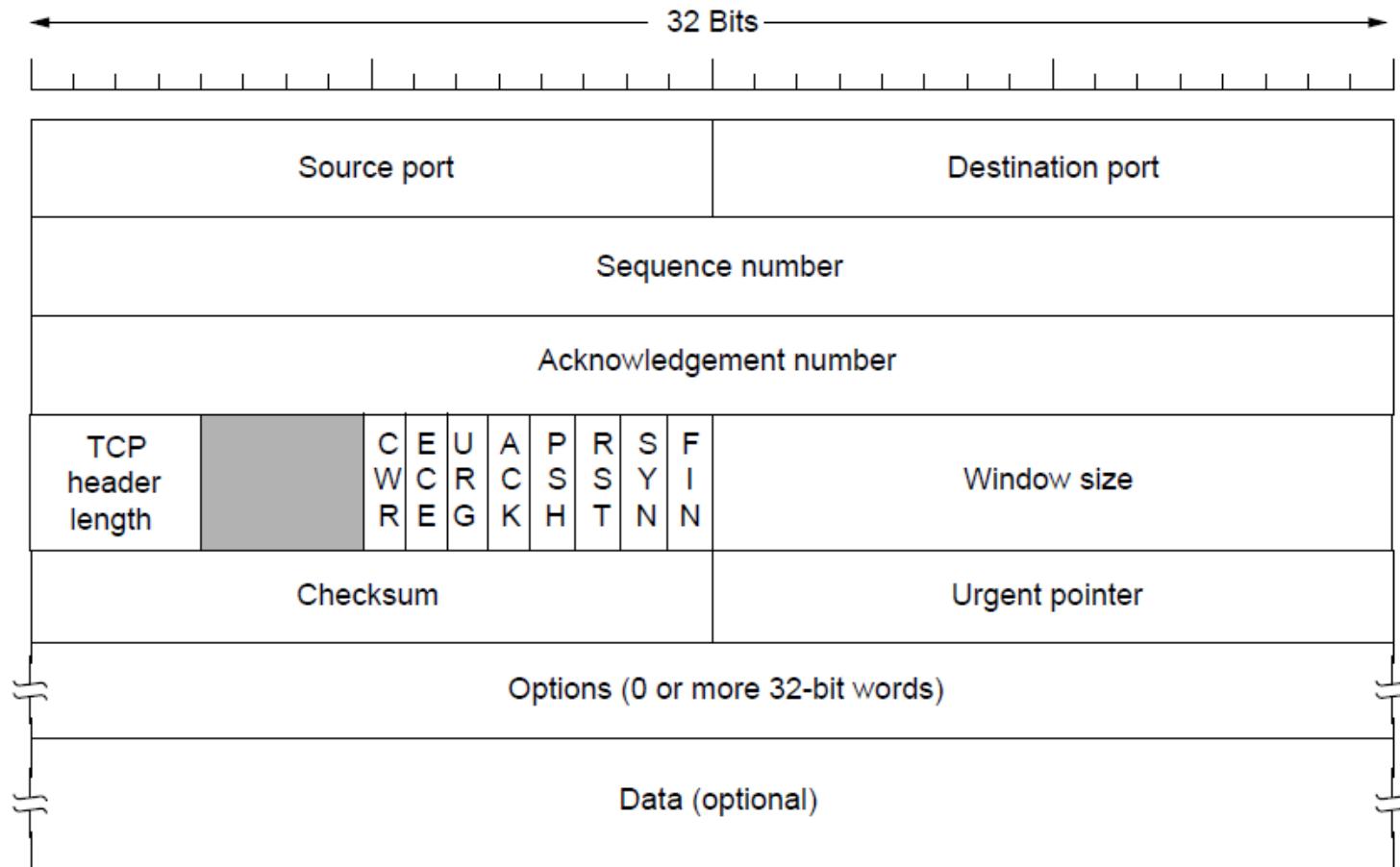
Addressing at Transport Layer

- Socket library provides a multiplexing tool on top of TSAPs to allow servers to service multiple clients
- It simulates the server using a different port to connect back to the client



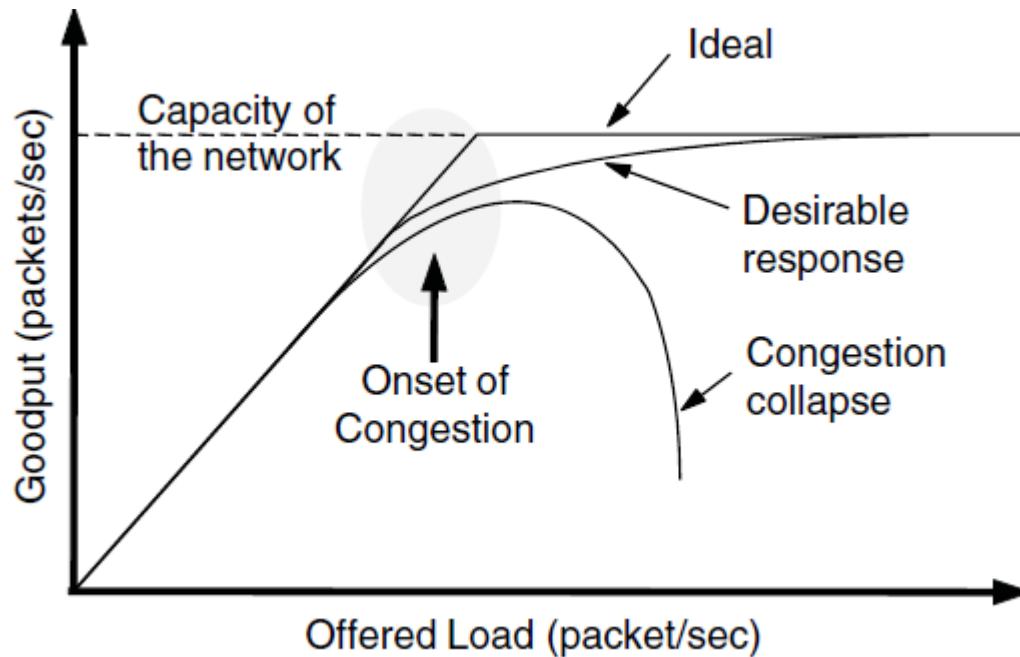
TCP Segment Header

- TCP header includes addressing (ports), sliding window (seq. / ack. number), flow control (window), error control (checksum) and more



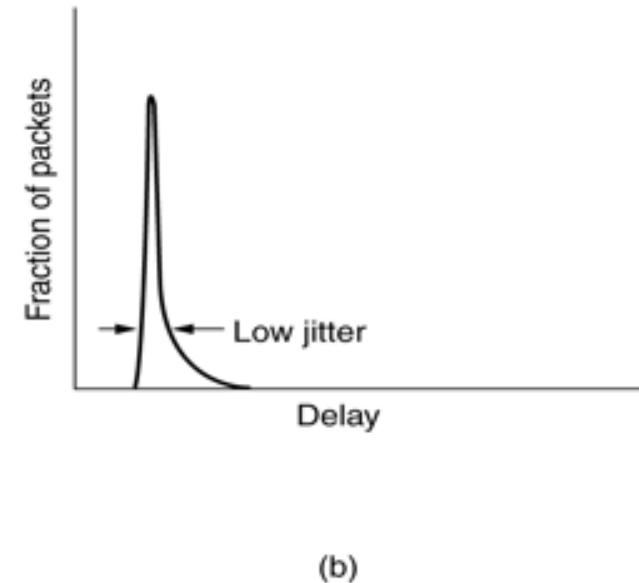
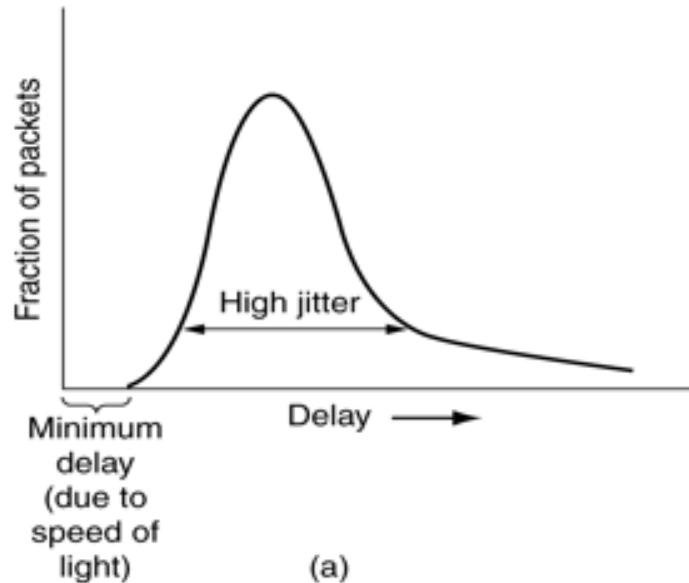
What happens in congestion?

- Congestion results when too much traffic is offered; performance degrades due to loss/retransmissions
 - Goodput (=useful packets) trails offered load



What is Jitter?

- Jitter is the variation in packet arrival times
 - a) high jitter
 - b) low jitter

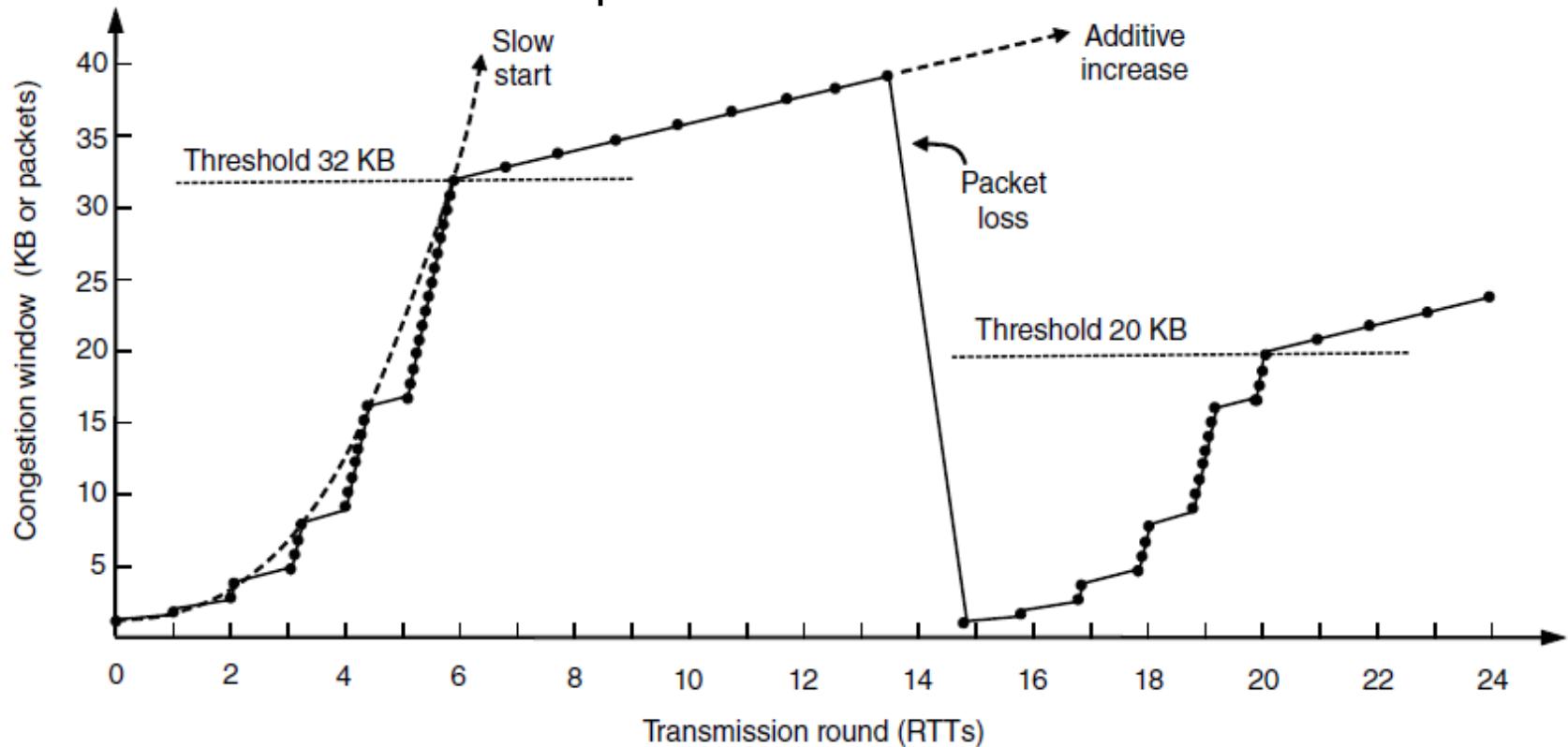


Techniques for Achieving Good QoS

- **Over-provisioning**
 - more than adequate buffer, router CPU, and bandwidth (expensive and not scalable ...)
- **Buffering**
 - buffer received flows before delivery - increases delay, but smoothes out jitter, no effect in reliability or bandwidth
- **Traffic Shaping**
 - regulate the average rate of transmission and burstiness of transmission
 - **leaky bucket**
 - **token bucket**
- **...**

Internet Congestion Control

Slow start followed by additive increase (TCP Tahoe)
Threshold is half of previous

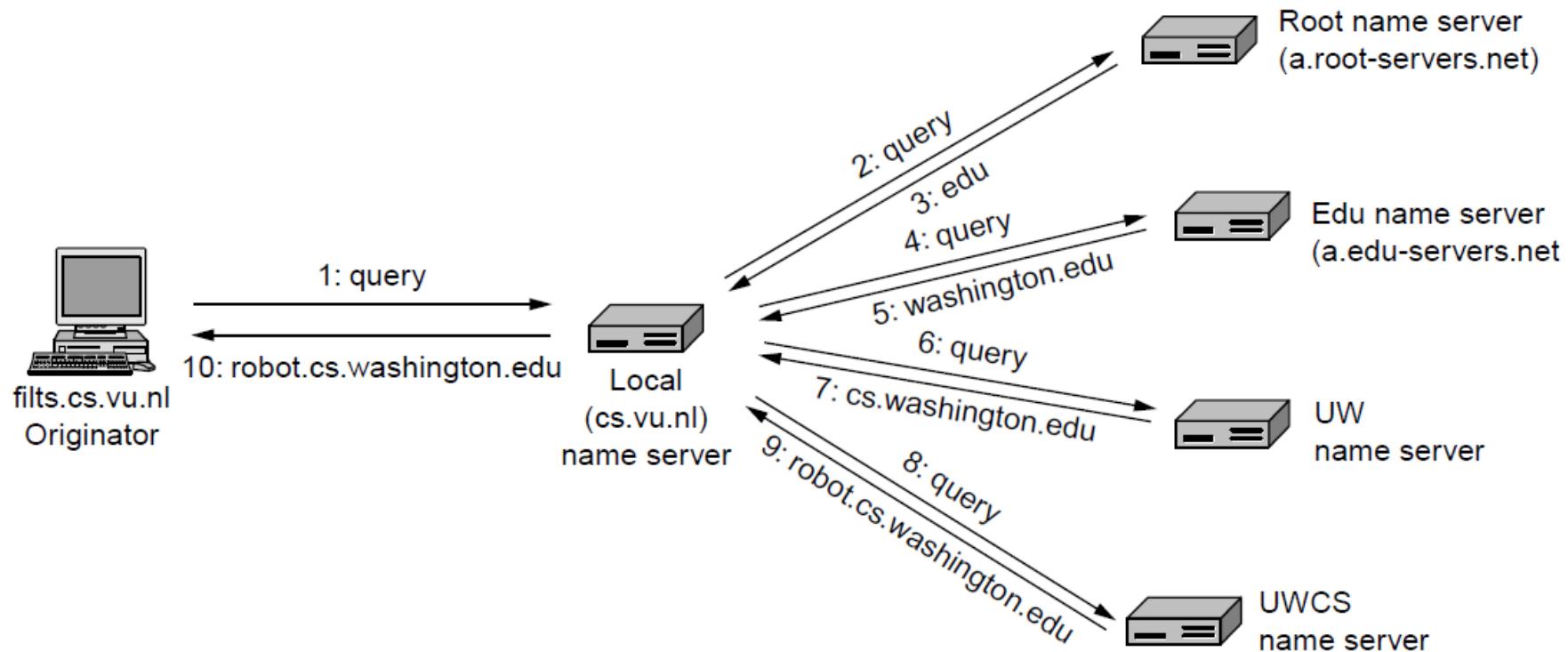


Application Layer: DNS First

- Problem?
 - IP address (32 bit), e.g., 121.7.106.83 – used for addressing datagrams
 - www.yahoo.com – used by humans
- Question: how do you map between IP address and name, and vice versa?
- Domain Name System:
 - *distributed database* implemented in a hierarchy of many *name servers*
 - *application-layer protocol* that allows a host to query the database in order to *resolve* names (address/name translation)
 - used by other application-layer protocols (http, ftp, smtp)

Example

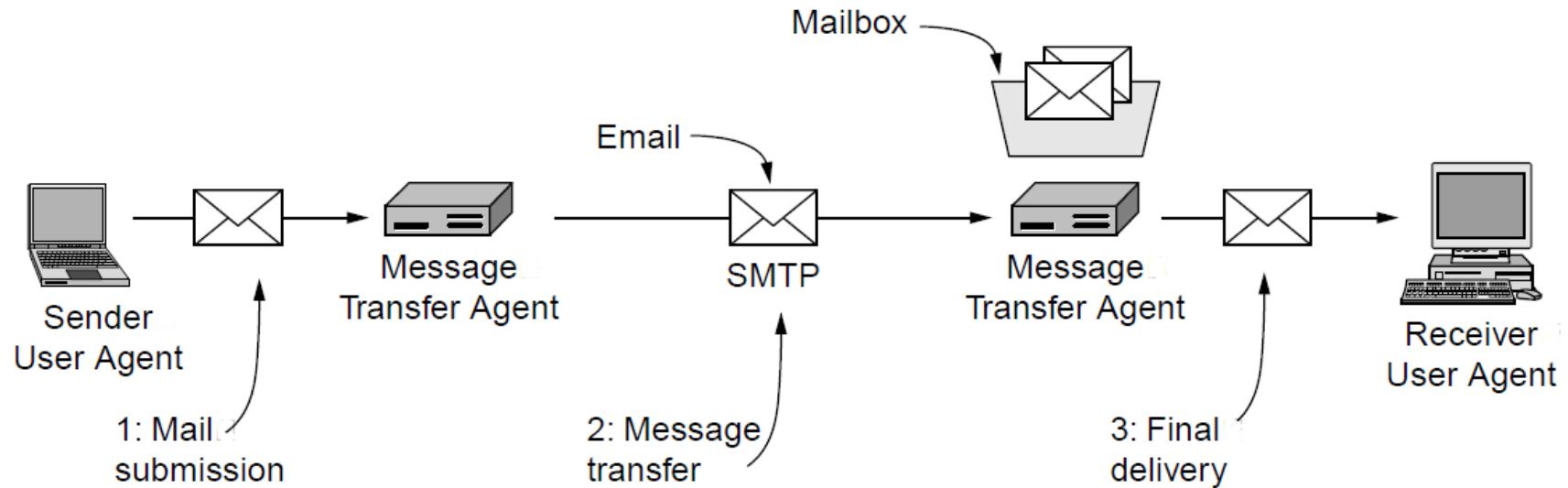
- Example of a computer looking up the IP for a name



The World Wide Web (WWW)

- World Wide Web key components are?
 - Client and Server software – Firefox is the client software for web access where Apache is on the server side
 - Web mark-up languages - HTML – how webpages are coded
 - Web scripting languages – More dynamicity to webpages - Javascript
 - HTTP – about how to transfer

Email



User agents

Allow user to read and send email

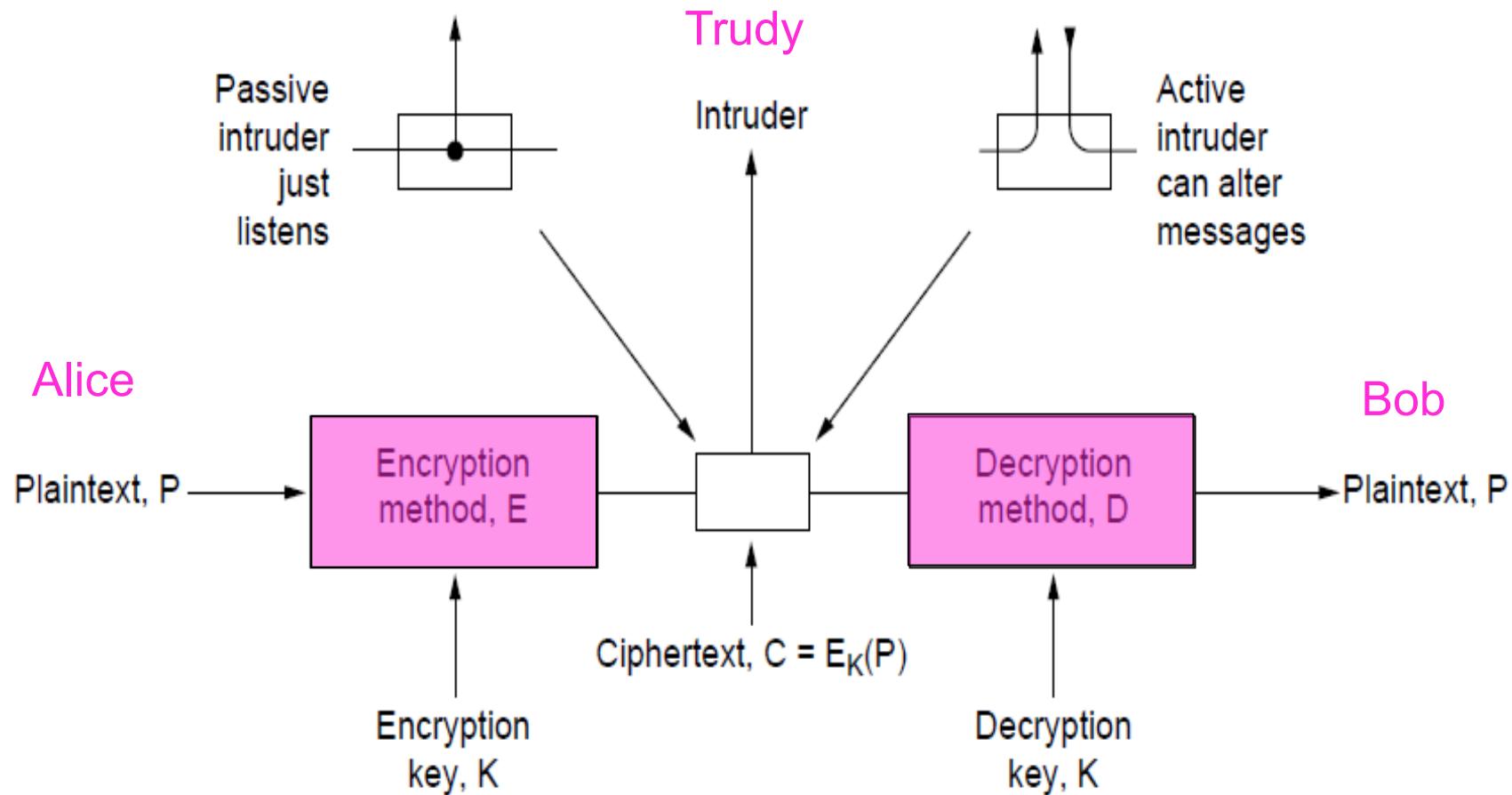
Message transfer agents

Transport messages from source - destination

Special Topic: Network Security

- Network security is a combo of 4 related areas:
 - **Secrecy** (Keeping information hidden from a general audience)
 - **Authentication** (Ensuring the user you are giving content to has valid credentials)
 - **Non-repudiation** (Prove a content was created by a named user)
 - **Integrity control** (Ensure that a content has not been tampered with)

Basics of Crypto: The Model



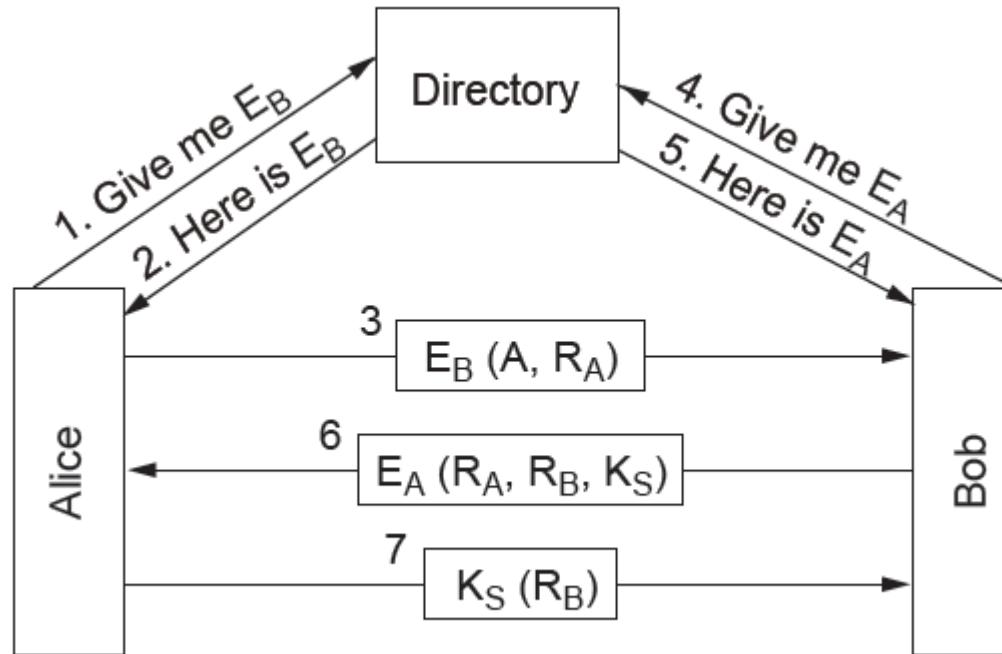
Modern Key-based Algorithms

- **Two main categories**
- Symmetric key algorithms use the same key for both encryption and decryption
- Symmetric key algorithms can use permutation, substitution and a combination of both to encrypt and decrypt
- **Symmetric Key Algorithms**
Numerous algorithms exist
We saw key solutions to certain types of attacks/problems

Asymmetric Key Algorithms

- **RSA - Rivest, Shamir, Adleman**
- Famous and robust algorithm
- Key generation:
 - Choose two large primes, p and q
 - Compute $n = p \times q$ and $z = (p - 1) \times (q - 1)$.
 - Choose d to be relatively prime to z, i.e., no common factors
 - Find e such that
 - **(d x e) mod z = 1**
 - Public key is (e, n), and private key is (d, n)
- Encryption:
 - $\text{Cipher} = \text{Plain}^e \pmod{n}$
- Decryption:
 - $\text{Plain} = \text{Cipher}^d \pmod{n}$

An Application: Authentication Using Public Key Cryptography



Review (1)

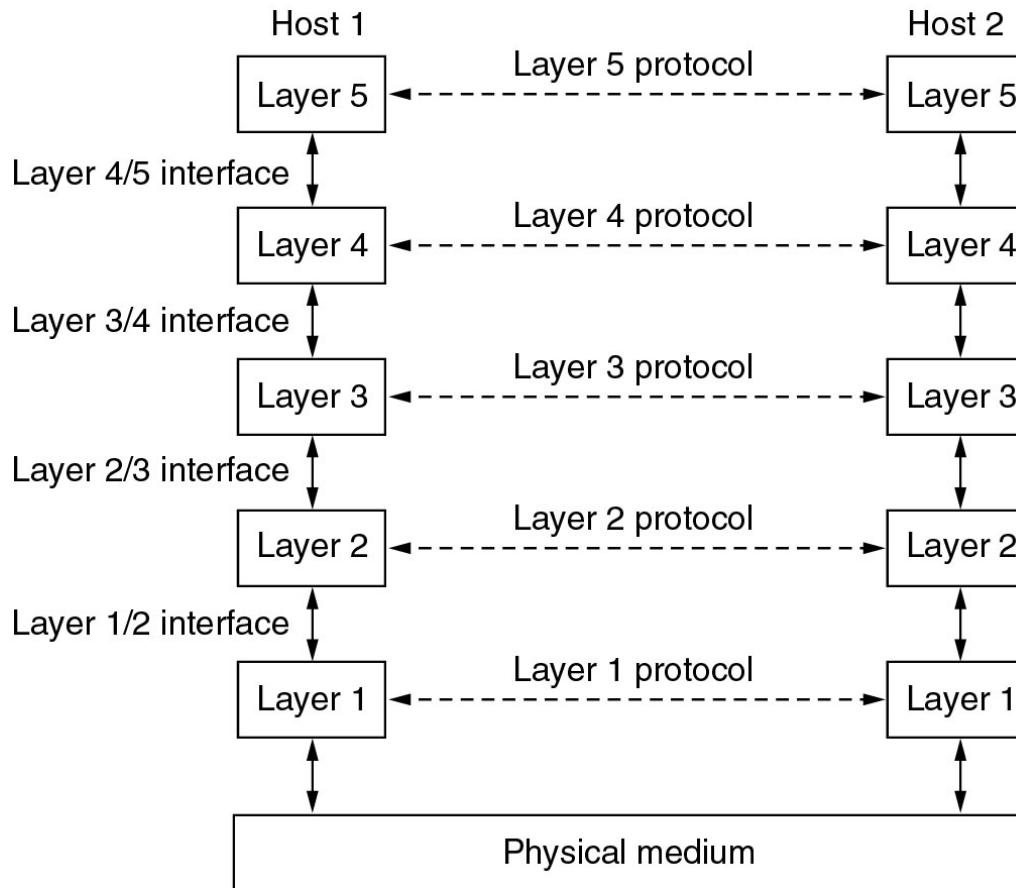
COMP90007 Internet Technologies

Lecturer: Ling Luo

Semester 2, 2020

Network Protocol Hierarchies

- Layers, protocols and interfaces



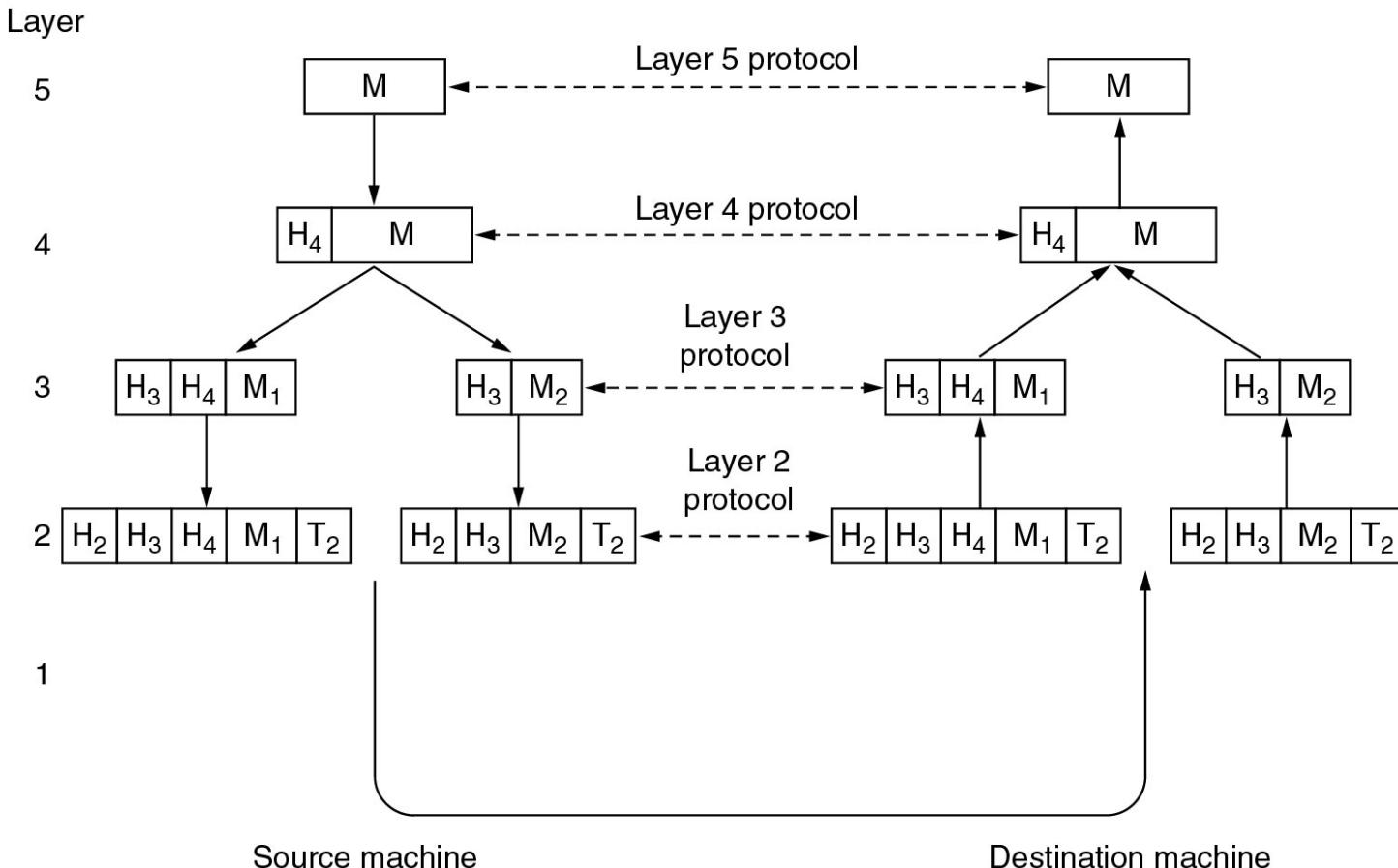
Consider the network as a stack of **layers**

Each layer offers **services** to layers above it through **interface**

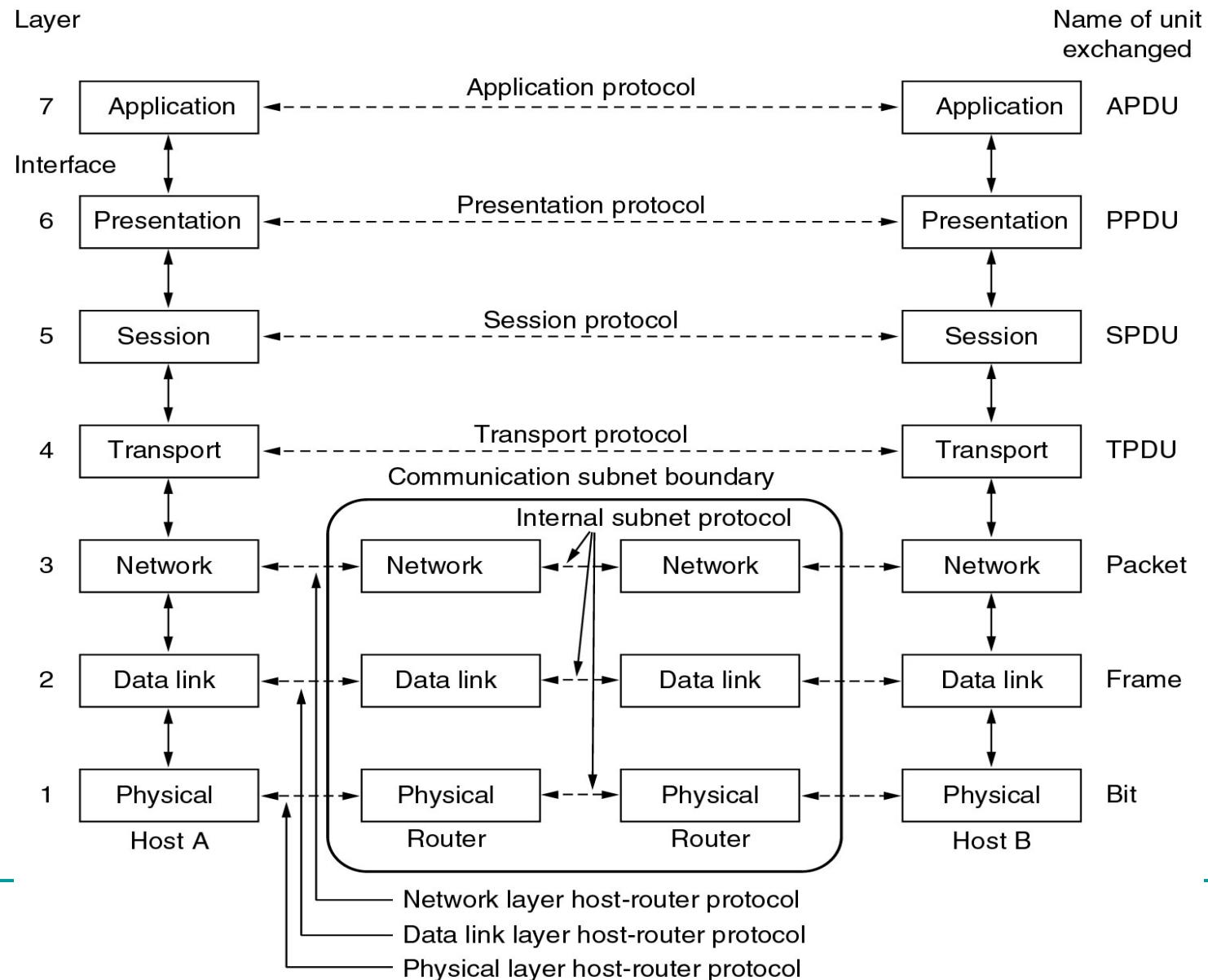
Protocol is an agreement between the communicating parties on how communication is to proceed

Network Protocol Hierarchies (2)

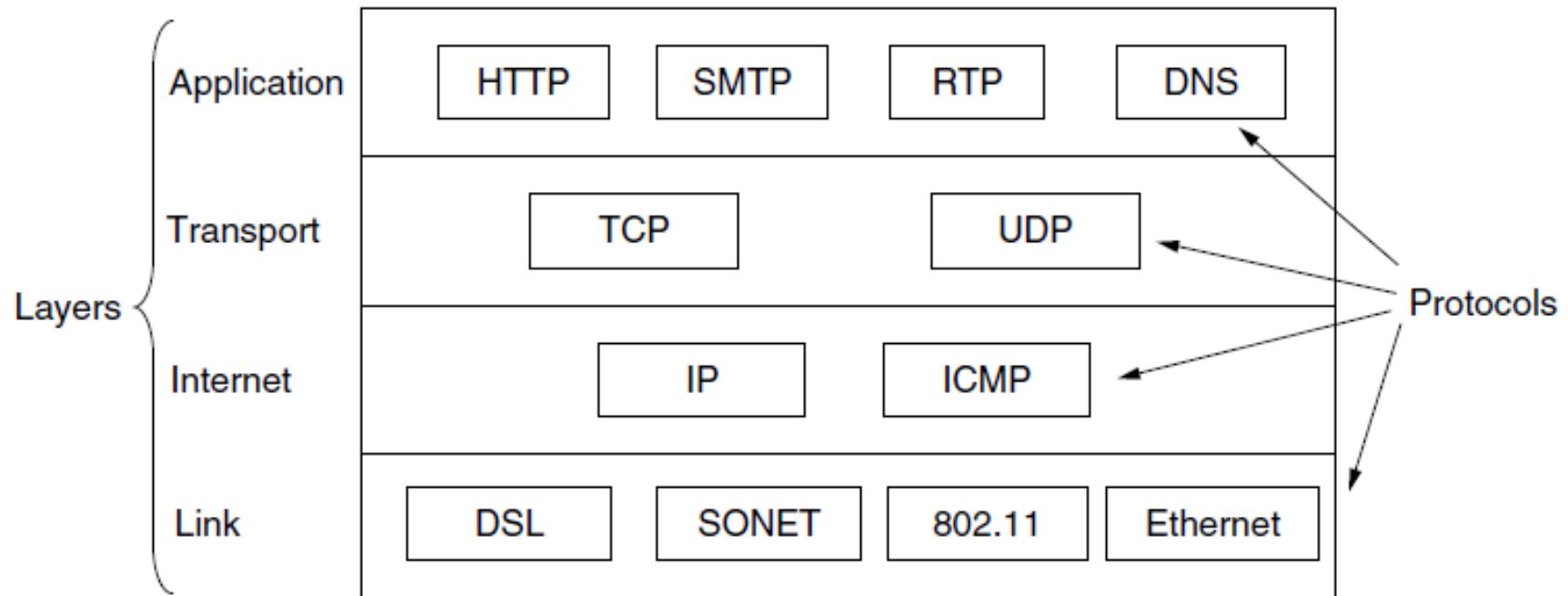
- Information flow supporting virtual communication in layer 5



OSI Reference Model



TCP/IP: Protocols



Hybrid Model

- The hybrid reference model used in this semester

5	Application layer
4	Transport layer
3	Network layer
2	Data link layer
1	Physical layer

Physical Layer

- The physical layer is concerned with the electrical, timing and mechanical interfaces of the network
 - **Mechanical:** material, cable length ...
 - **Electrical:** voltage levels, signal strength ...
 - **Timing:** data rate, latency

Compare Transmission Media (1)

Wired: twisted pairs, coaxial cable, fibre optics ...

Wireless: radio, microwave, infrared, satellite ...

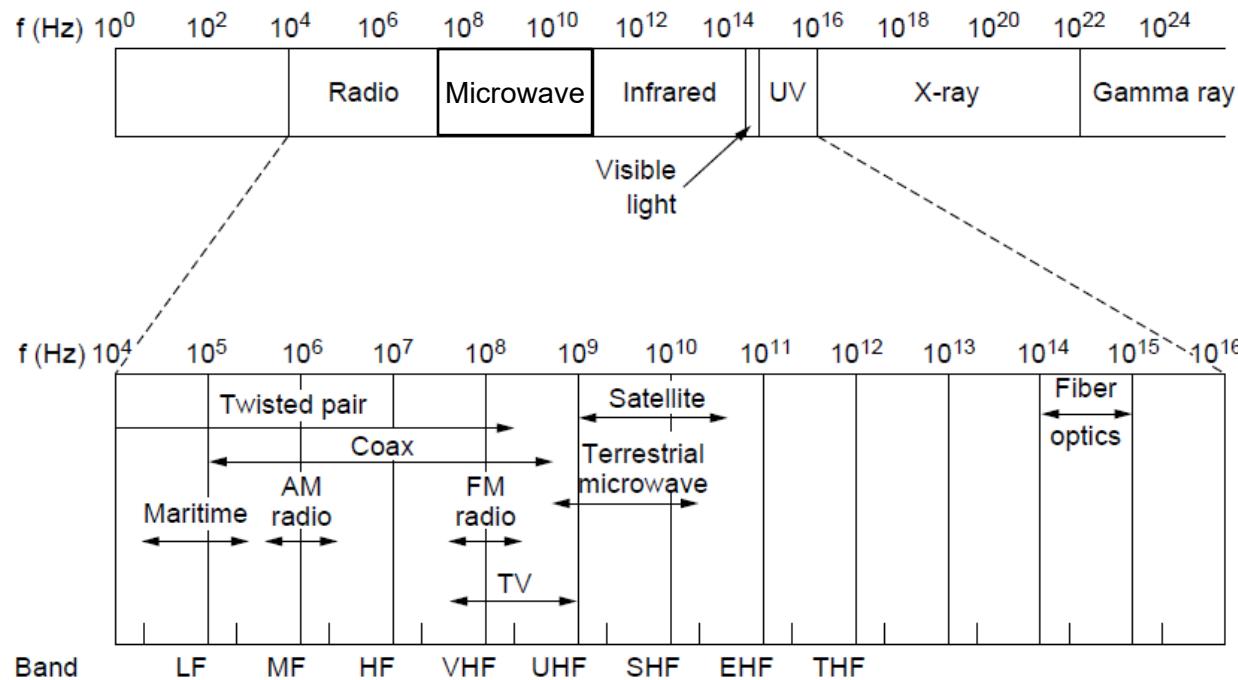
Compare the properties of wires and fibre:

Property	Wires	Fibre
Distance	Short (100s of m)	Long (tens of km)
Bandwidth	Moderate	Very High
Security	Easy to tap	Hard to tap
Cost	Inexpensive	More Expensive
Convenience	Easy to use	Harder to use

Compare Transmission Media (2)

Wireless Electromagnetic Spectrum

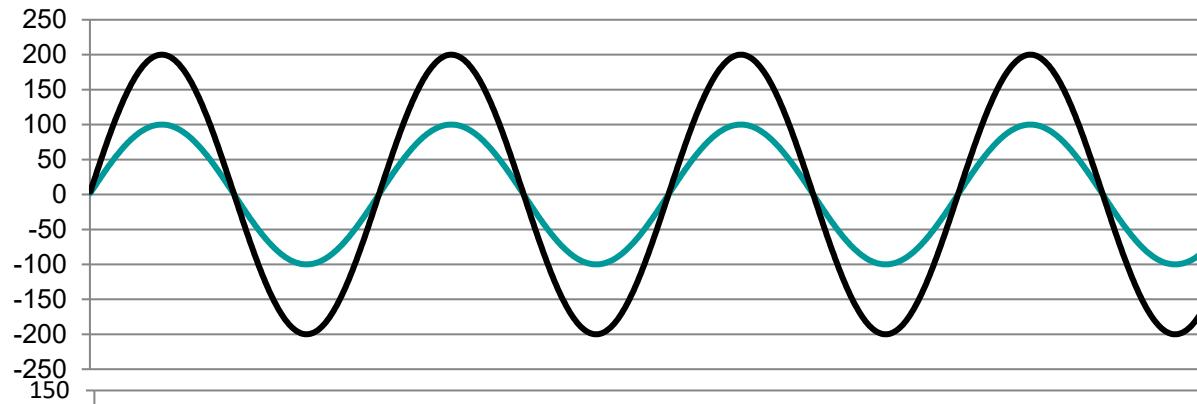
- Radio: wide-area broadcast
- Microwave: LANs and 3G/4G
- Infrared/Light: line-of-sight



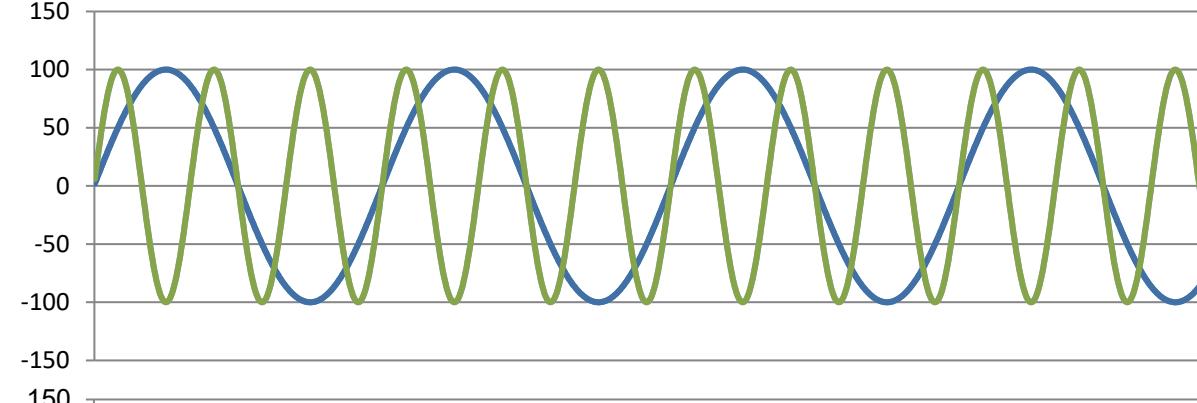
Data Communication using Signals (1)

- Information is transmitted by varying a physical property e.g. voltage, current
- For a sinewave :
function: $c * \sin(a * t + b)$
c: amplitude, $a/(2\pi)$:frequency and b:phase
can change the behaviour of the function.

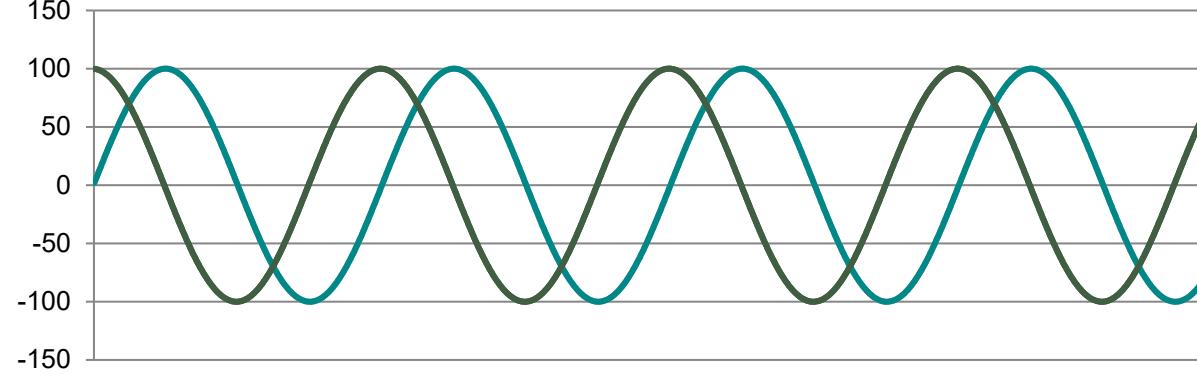
Data Communication using Signals (2)



Change in
Amplitude



Change in
Frequency



Change in
Phase

Maximum Data Rate of a Channel

- Nyquist's theorem relates the data rate to the bandwidth (B) and number of signal levels (V) (**channel without noise**):

$$\text{Max. data rate} = 2B \log_2 V \text{ bits/sec}$$

- Shannon's theorem relates the data rate to the bandwidth (B) and signal strength (S) relative to the **noise (N)**:

$$\text{Max. data rate} = B \log_2(1 + S/N) \text{ bits/sec}$$

Message Latency

- **Transmission delay**
 - $T\text{-delay} = \text{Message in bits} / \text{Rate of transmission}$
 $= M/R$ seconds
- **Propagation delay**
 - $P\text{-delay} = \text{length of the channel} / \text{speed of signals}$
 $= \text{Length} / \text{Speed of signal}$ (2/3 of speed of light for wire)
- **Latency = L = T-delay + P-delay**

Data Link Layer

- Functions of the data link layer:
 1. Provide a well-defined service interface to network layer
 2. Handling transmission errors
 3. Data flow regulation
- Primary process:
 - Take **packets** from network layer, and encapsulate them into **frames** (containing a header, a payload, a trailer)

Framing

- Methods:
 - Character (Byte) count
 - Flag bytes with byte stuffing
 - Start and end flags with bit stuffing
- Most data link protocols use a combination of character count and one other method

Error Control

- Adding check bits to ensure that a garbled message by the physical layer is not considered as the original message by the receiver
- Error Control deals with
 - **Detecting** the error
 - Parity
 - Checksum
 - Cyclical Redundancy Check (CRC)
 - **Correcting** the error
 - Hamming Code

Error Bounds – Hamming distance

Flow Control

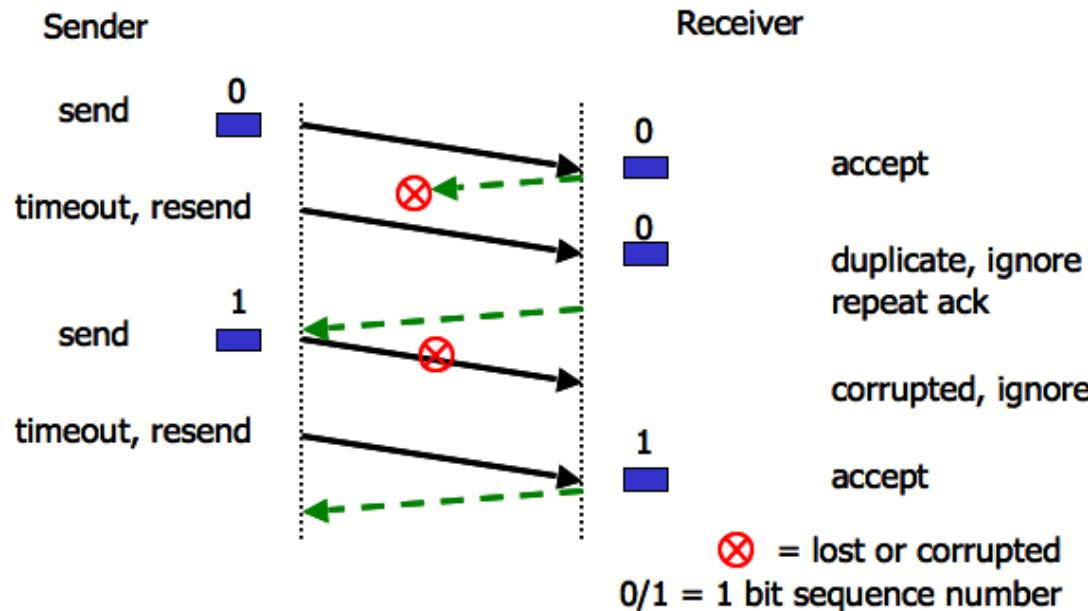
- The **fast senders** vs **slow receivers** problem requires a solution
- Principles to control when sender can send next frame

Feedback based flow control: ack

- Stop and wait
- Sliding window: go-back-N, selective repeat

Stop and Wait Protocol

- ARQ (Automatic Repeat reQuest)
 - Ack and Timeout



Link Utilisation in Stop and Wait Protocols

Link Utilisation (U) measures efficiency in communication.

T_f = Transmission delay, time needed to transmit a frame of length L;

T_p = Propagation delay;

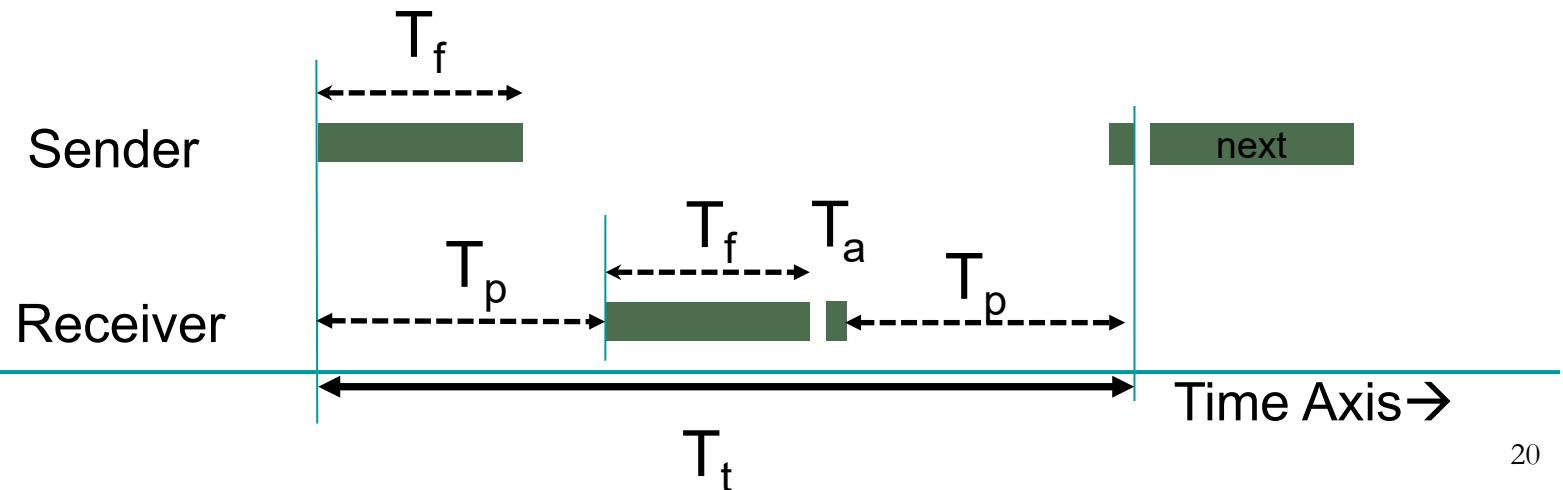
T_a = Time for transmitting an Ack, and we can assume $T_a = 0$.

$$T_t = T_f + 2T_p$$

$$U = (\text{Time of transmitting a frame}) / (\text{Total time for the transfer}) = T_f / T_t$$

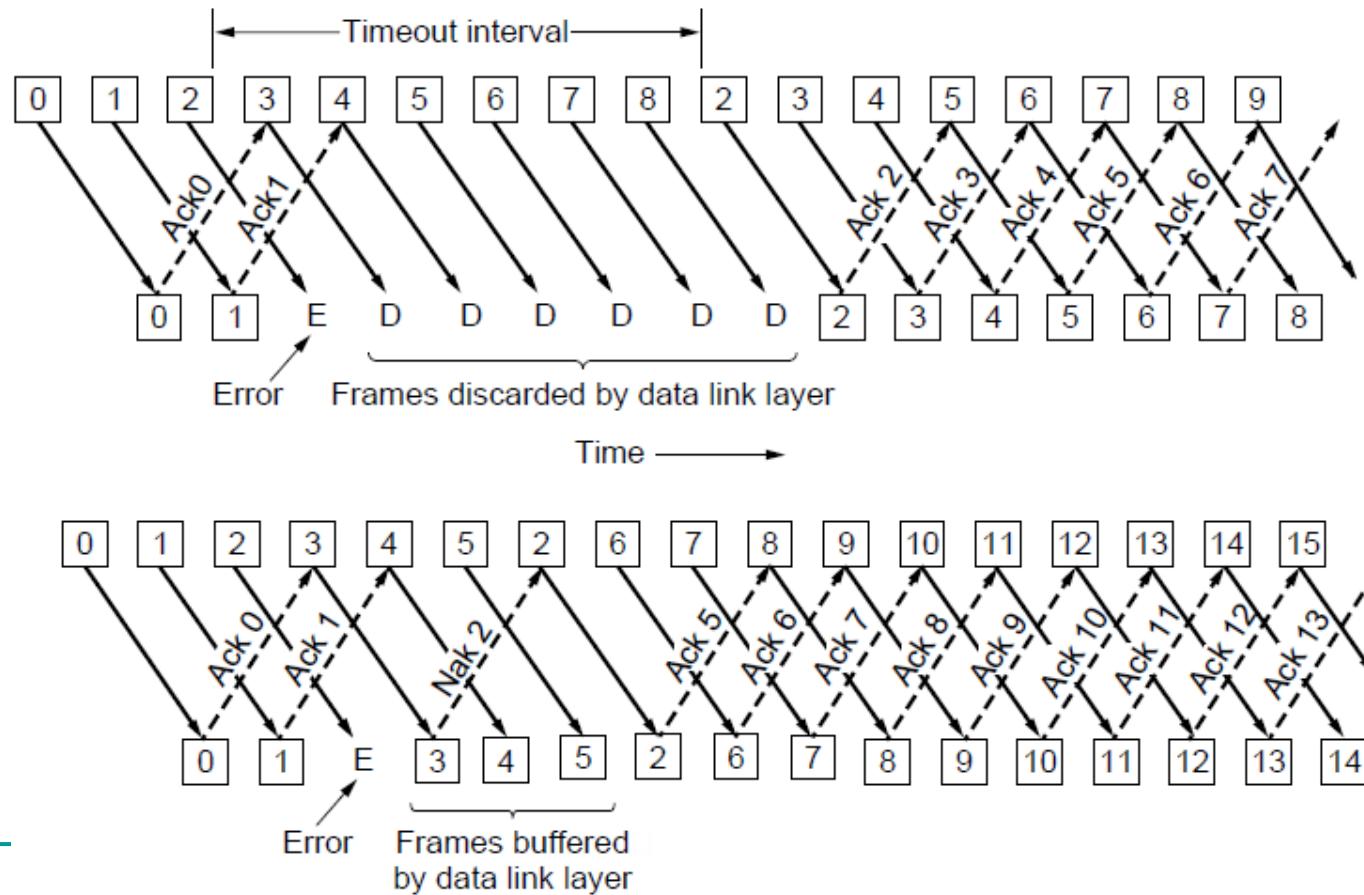
Given bit rate B and $T_f = L/B$, we have

$$U = T_f / (T_f + 2T_p) = (L/B) / (L/B + 2T_p) = L / (L + 2T_p B).$$



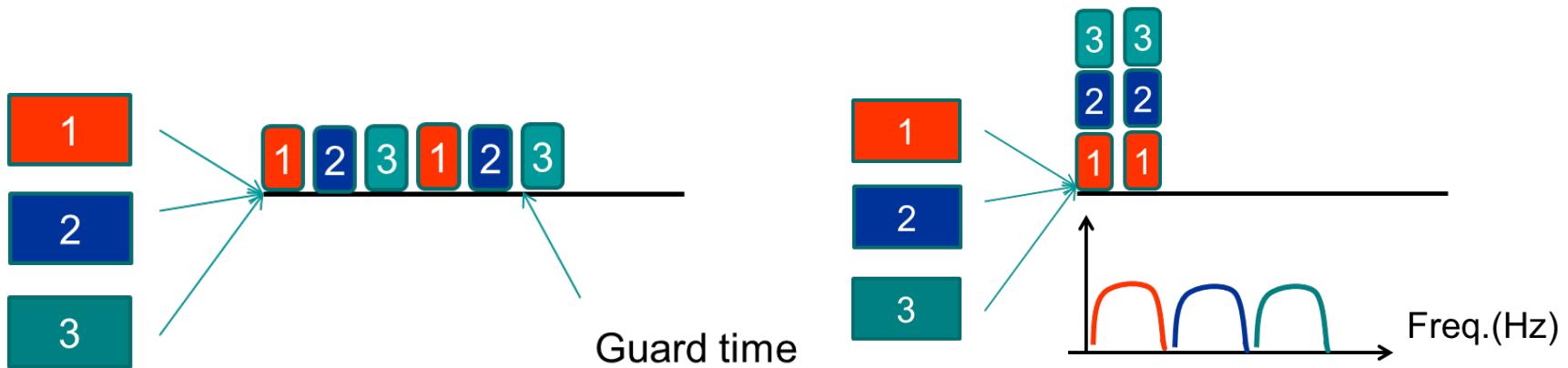
Go-Back-N vs. Selective Repeat

- Trade-off between efficient use of bandwidth and data link layer buffer space



Medium Access Control

- MAC sub-layer is used to assist in resolving transmission conflicts
- Static channel allocation
 - Time division multiplexing
 - Frequency division multiplexing

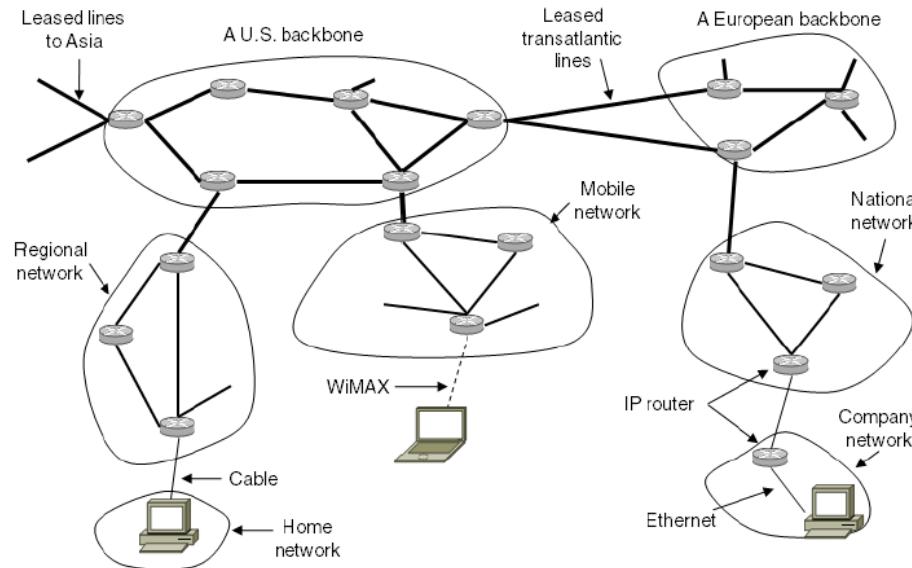


Medium Access Control

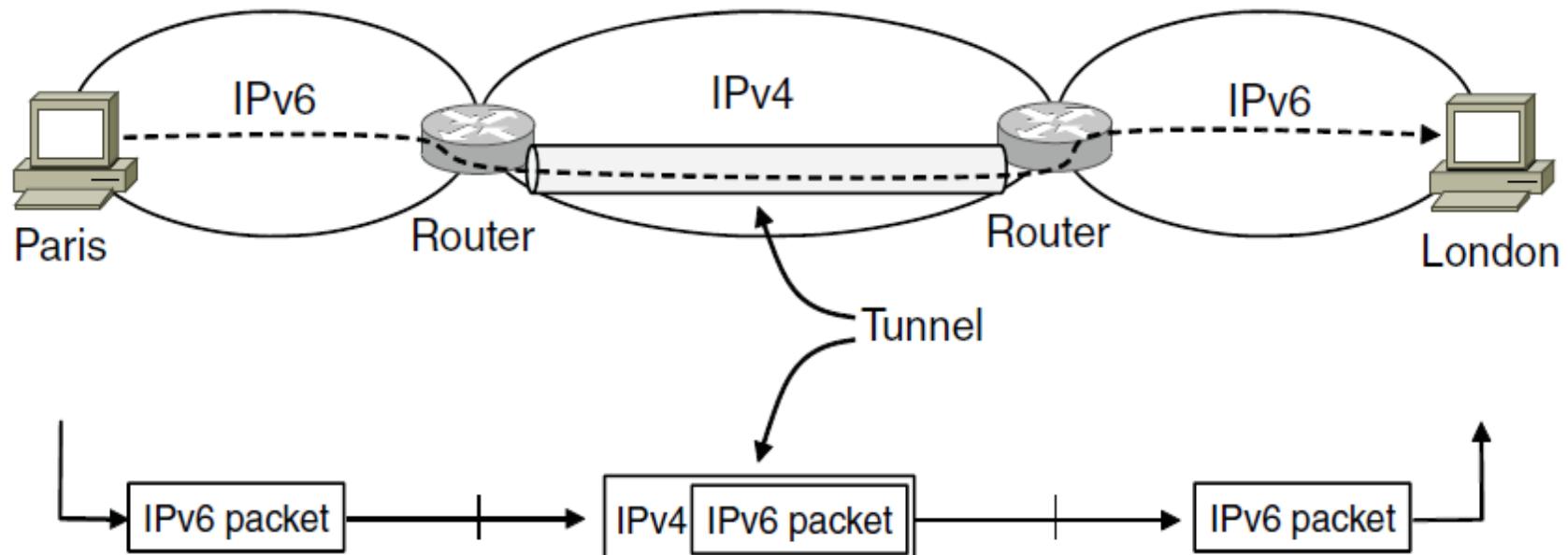
- Contention
 - ALOHA, Slotted ALOHA
 - Carrier Sense Multiple Access: 1-persistent, non-persistent, p-persistent, with collision detection
- Collision Free: bit map, binary countdown
- Limited Contention: adaptive tree walk
- Wireless LANs: MACA/MACAW

Network Layer

- Internet is a collection of many networks that is interconnected by the IP protocol
- Provides a **best-effort** service to **route datagrams** from source host to destination host



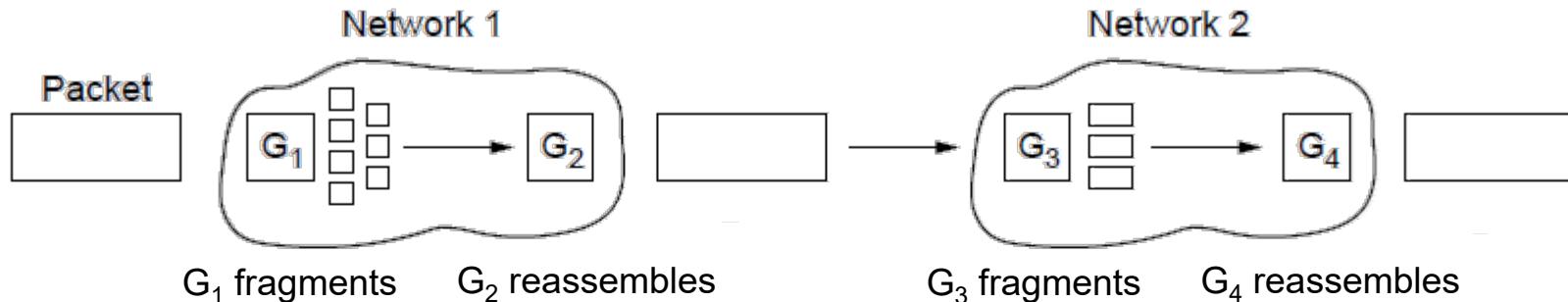
Connecting Different Networks (1)



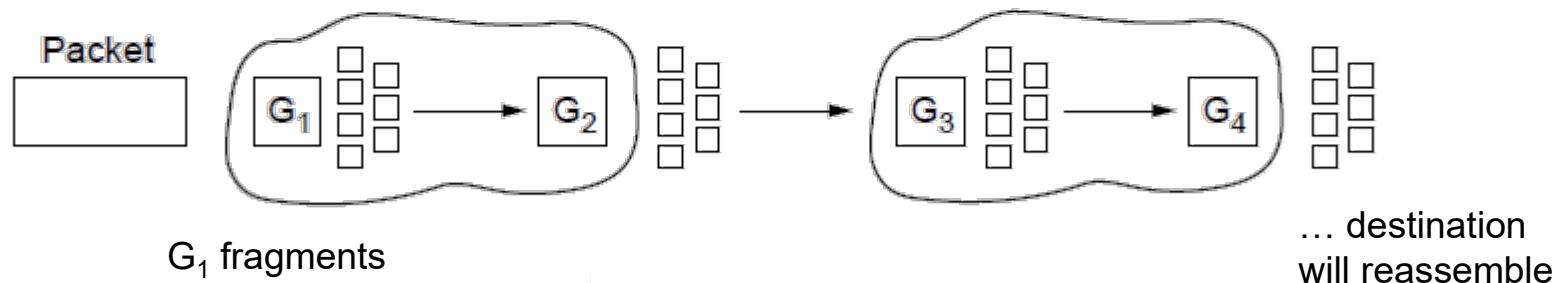
Tunneling IPv6 Packets through IPv4

Connecting Different Networks (2)

- Large packets need to be routed through a network whose maximum packet size is too small.
- **Solution: Fragmentation and Reassembly.**



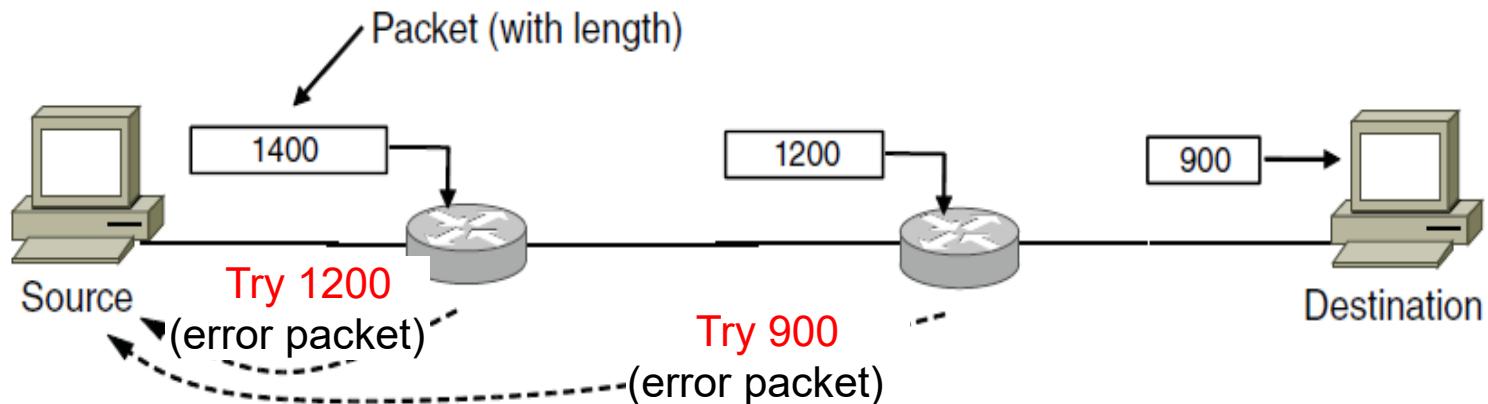
a) **Transparent** – packets fragmented / reassembled in each network.



- **b) Non-transparent** – fragments are reassembled at destination.
Each packet requires packet number, byte offset, end of packet flag.

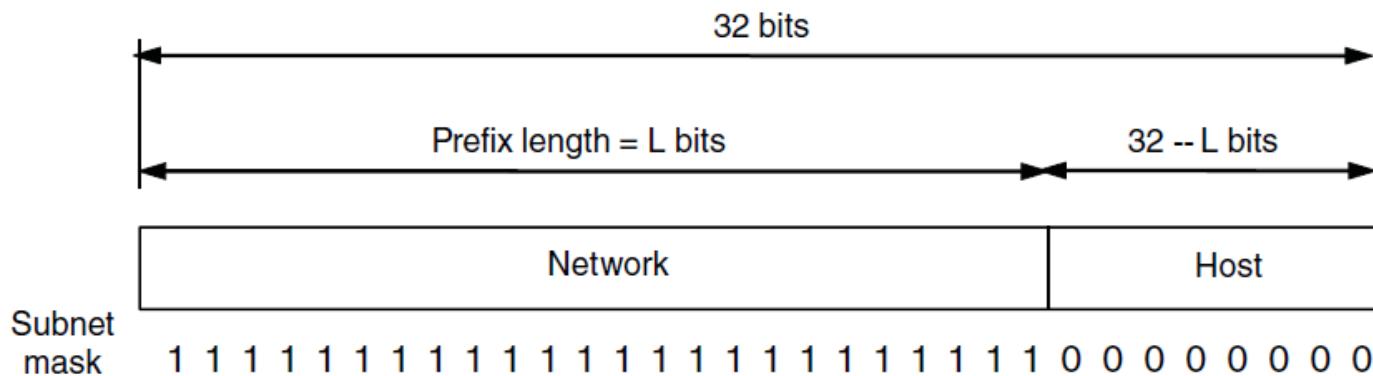
Connecting Different Networks (3)

- Path MTU Discovery: alternative to Fragmentation
- Advantage: the source knows what length packet to send
- If the routes and path MTU change, new error packets will be triggered and the source will adapt to the new path



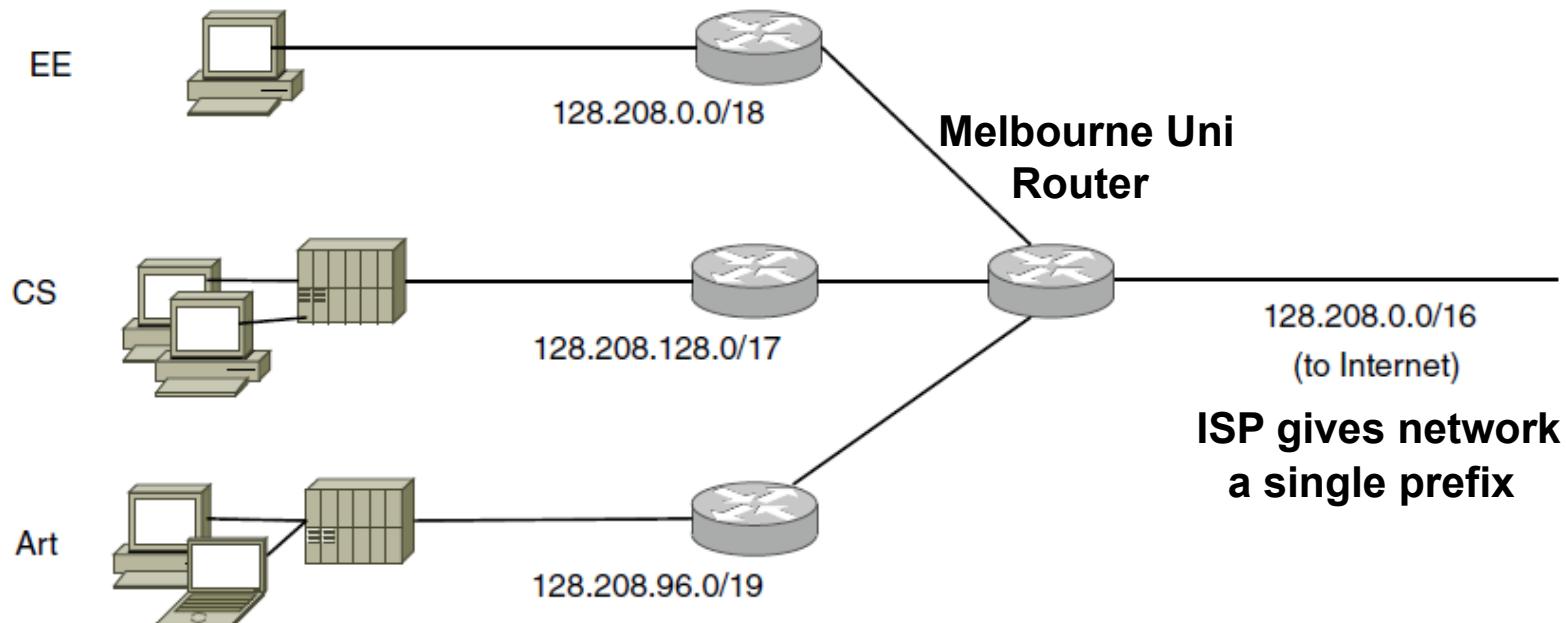
IP Addresses

- network portion + host portion
- **Prefix:** determined by the network portion, all hosts on a single network has the same network portion.
prefix is written as: lowest address/bit-length
 $18.2.31.0/24, 18.2.0.0/16$
- **Subnet mask:** all 1s in the network portion
- **Extract prefix:** ANDed the IP address with the subnet mask



Subnets

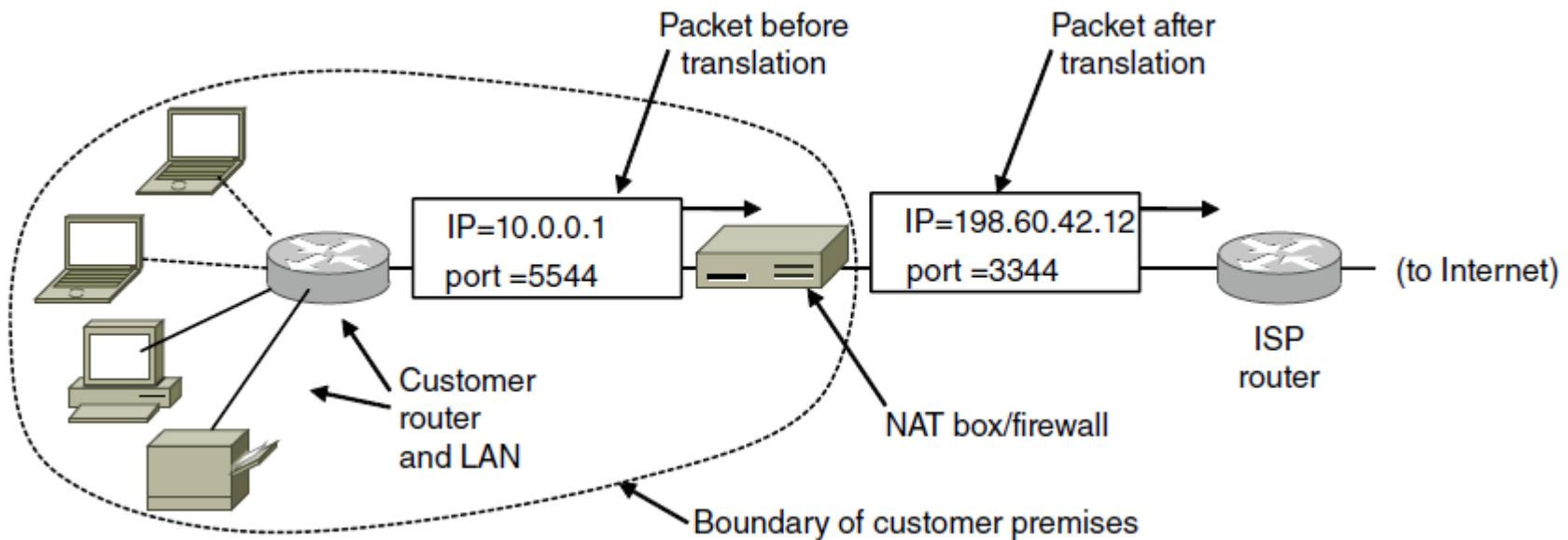
- Subnetting allows networks to be split into several parts for internal uses whilst acting like a single network for external use
- Looks like a single prefix outside the network



Network is divided into subnets internally

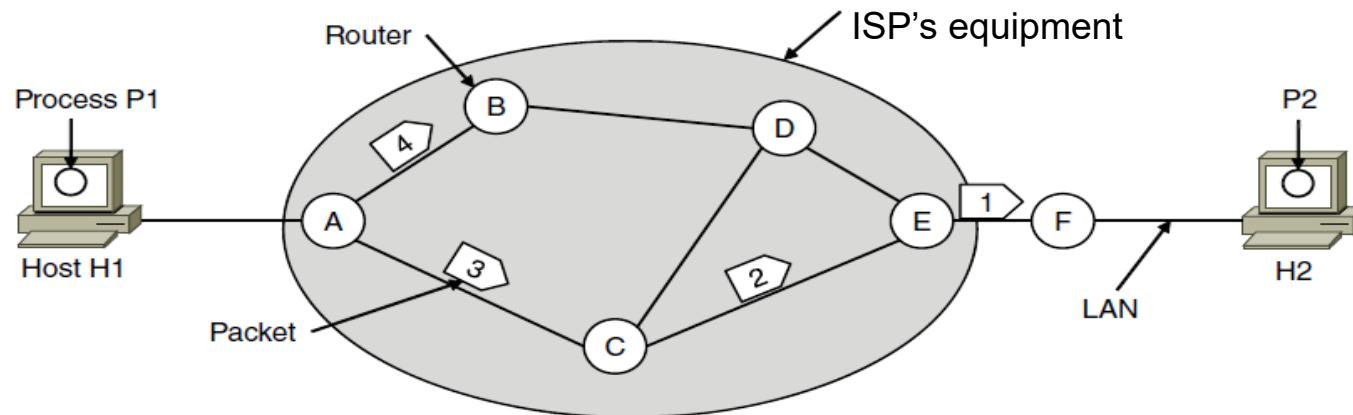
Network Address Translation (NAT)

- NAT box maps one external IP address to many internal IP addresses
 - Uses TCP/UDP port to distinguish connections
 - Violates layering; popular tool in conserving global address space



Routing within a Datagram Subnet

- **Connectionless - post office model:** packets are routed individually based on destination addresses in them
- Packets can take different paths
- E.g., P1 sends a long message to P2



A's table (initially)

A	X
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	X
B	B
C	C
D	B
E	B
F	B

C's Table

A	A
B	A
C	X
D	E
E	E
F	E

E's Table

A	C
B	D
C	C
D	D
E	X
F	F

Routing table (can be fixed or change over time)

Routing algorithm – manages the routing table

Virtual-Circuit vs. Datagram Subnets

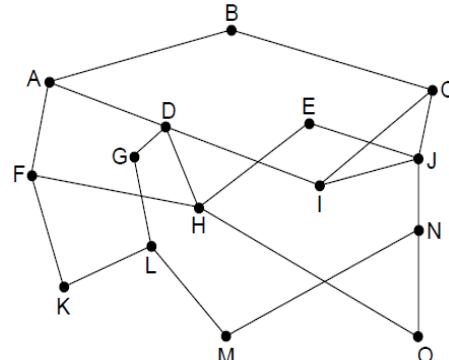
Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Routing Algorithms

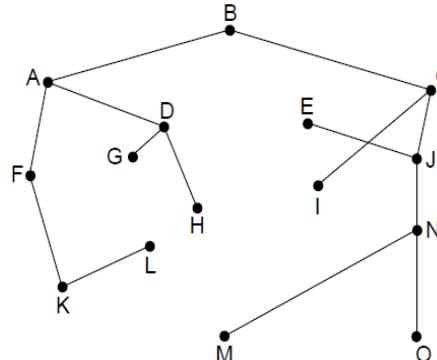
- Non-adaptive
 - Shortest path routing
 - Flooding
- Adaptive
 - Distance vector routing
 - Link state routing
- Hierarchical routing
- Broadcasting routing
- Multicasting routing

Shortest Path Routing

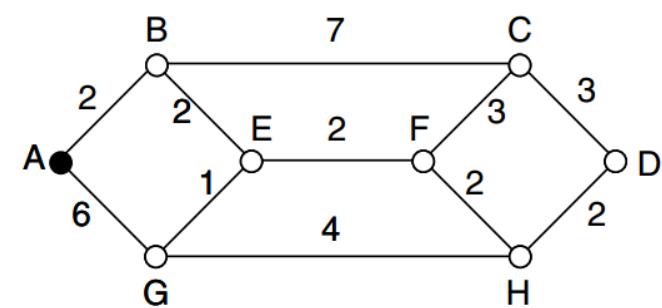
- To choose a path between 2 routers, the algorithm finds the shortest path between them on the graph
- Metrics: number of hops, distance, delay etc.



Network

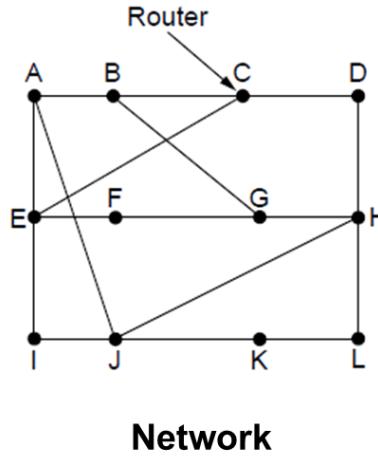


Sink tree of best paths to router B



Distance Vector Routing

- Each router maintains a table which includes the best known distance to each destination and which line to use to get there.
- Global information shared locally.



New estimated delay from J ↓ Line

To	A	I	H	K	Line
A	0	24	20	21	8 A
B	12	36	31	28	20 A
C	25	18	19	36	28 I
D	40	27	8	24	20 H
E	14	7	30	22	17 I
F	23	20	19	40	30 I
G	18	31	6	31	18 H
H	17	20	0	19	12 H
I	21	0	14	22	10 I
J	9	11	7	10	0 -
K	24	22	22	0	6 K
L	29	33	9	9	15 K

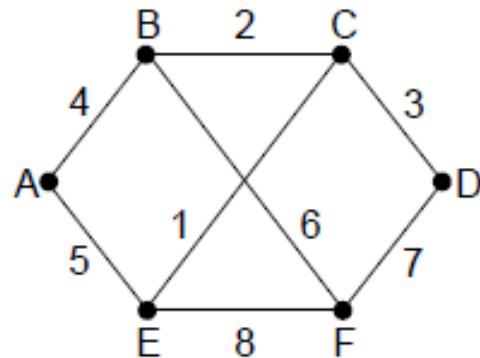
JA delay is 8 JI delay is 10 JH delay is 12 JK delay is 6

New vector for J

Vectors received at J from neighbors A, I, H and K

Link State Routing

- ❑ An alternative to distance vector: **too long to converge** after the network topology changed
- ❑ Widely used in the Internet
- ❑ More computation but simpler dynamics
- ❑ Local information shared globally using flooding



Network

Link	State	Packets
A	B	E
Seq.	Seq.	Seq.
Age	Age	Age
B 4	B 2	C 5
E 5	D 3	B 6
	C 2	D 7
	F 6	E 8
	E 1	

LSP for each node

Broadcast Routing

- Broadcast routing allows hosts to send messages to all other hosts.
 - Single distinct packet to each destination
 - Multi-destination routing
 - Flooding
 - Reverse path forwarding

COMP90007 Internet Technologies

Week 12 Workshop

Semester 2, 2020

Question 1

Given the RSA algorithm we studied last week, if $p = 3$, $q = 11$ and if $d = 3$ and $e = 7$ instead of the version we saw in class, using the same character mapping we saw in class though, where A is 01 and B is 02, and C is 03 and so on, how would RSA work? Would it work at all? Show in detail what numbers would be computed and transmitted at both ends of a transmission if we want to send across a "D". Show where it fails if it does not work properly?

Question 2

Using the RSA algorithm we saw in class, please design a simple algorithm to sign documents where the sender cannot refute the fact that a document was signed by herself later on. Which property of RSA your algorithm relies on?

Question 3

Given the Diffie-Hellman key challenge in the lectures please develop the full flow chart for the man-in-the-middle (MITM) attack, with step numbers and messages sent, show details about how this attack would work.

Question 4

Leveraging the authentication protocol using Public-Key cryptography, we send across two additional numbers, RA and RB. Why are these needed? Why not Alice sends only her name to Bob but needs a RA as well?

Question 5

Please list, summarize the four key areas/aspects of network security.

COMP90007 Internet Technologies

Week 12 Workshop

Semester 2, 2020

Suggested solutions

Question 1

Given the RSA algorithm we studied last week, if $p = 3$, $q = 11$ and if $d = 3$ and $e = 7$ instead of the version we saw in class, using the same character mapping we saw in class though, where A is 01 and B is 02, and C is 03 and so on, how would RSA work? Would it work at all? Show in detail what numbers would be computed and transmitted at both ends of a transmission if we want to send across a "D". Show where it fails if it does not work properly?

Ans: It Works! $p = 3$, $q = 11$ means z is $(3 - 1) \times (11 - 1) = 20$

d is chosen to be 3 which has no common factors with z which is good. e is 7 which means $(d \times e)$ is $3 \times 7 = 21$. Thus $21 \bmod 20$ is 1 which is another good choice!

n is $p \times q = 33$

For encryption the pair to use 7,33 which is the public key ,and for decryption 3, 33 is used which is the private key! To send "D", first we see that it has numerical value is 4 as per this question's suggestion. And $4^7 = 16384$

$16384 \bmod 33 = 16$ is found next (Ok to use a calculator here but not necessary if you see 4^7 is 2^{14})

16 is sent in transmission and then we take $16^3 = 4096$ upon receipt, and then $4096 \bmod 33$ to get 4 which concludes decryption, 4 is "D" in our coding. Eureka!

Question 2

Using the RSA algorithm we saw in class, please design a simple algorithm to sign documents where the sender cannot refute the fact that a document was signed by herself later on. Which property of RSA your algorithm relies on?

Ans. The algorithm is as follows:

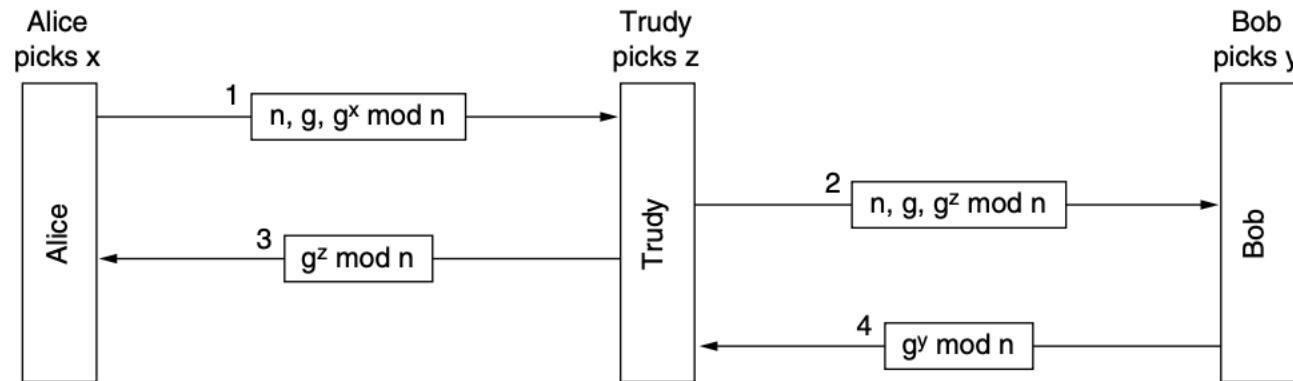
- Someone sends a document for signature to person A.
- Our person A signs it by using her private key PrK to, pretty much uses private key to lock the document.
- Then this message is sent to whoever needs it. We can add the plain text message to this communication as well or the original document can be accessible from a webpage etc.
- Receiver uses the public key of the sender A, say PuK, to open the message and if the text matches the original plaintext of the document then sender A should be the one who signed this document as there is no other person who can lock the document with her private key as only she knows that key; which only our public key is capable of countering the effect of...

We rely on the property that $E(D(P)) = P$ in RSA as well as $D(E(P)) = P$ using these key pairs.

Question 3

Given the Diffie-Hellman key challenge in the lectures please develop the full flow chart for the man-in-the-middle (MITM) attack, with step numbers and messages sent, show details about how this attack would work.

Ans.



Refer to section 8.7.2 of Tanenbaum.

Question 4

Leveraging the authentication protocol using Public-Key cryptography, we send across two additional numbers, RA and RB. Why are these needed? Why not Alice sends only her name to Bob but needs a RA as well?

Ans. Without RA Bob can still send back an acknowledgement but Alice cannot be sure that whether the responding person is Bob or not. The RA is needed to prove that Bob opened the initial message with his private key, saw RA, and in the response message sends it to Alice to prove this. Same is true for the role of RB.

Question 5

Please list, summarize the four key areas/aspects of network security.

Ans. The four key areas/aspects are:

Secrecy: keeping information hidden from a general audience, i.e., except the intended party

Authentication: Ensuring the user you are giving the content to has the valid id/credentials

Non-repudiation: Proving that the content belongs to/send by a named sender

Integrity control: Ensuring the content is not tampered with, e.g., during transport