



# Data Communications

DCF255

Lecture 1 | Introduction to Course and Data Communication Networks

# Agenda

- Overview of Course and Expectations
- Understanding basic terminology – Data Communications v Networking
- Why do programmers need to understand data communications?
- Brief history of Data Communication Networks

# Course Overview

Brief description of DCF<sub>255</sub> Expectations

# Course Description

- [DCF255](#)

# Course Web Site

- [MySeneca](#)

# Course Evaluation

|  |  |       |             |
|--|--|-------|-------------|
| ▪ Lab Assignments: (8) 1.875 each  | 15%  |       |             |
| ▪ Group Assignment   | 15%  | Total | 30%         |
| Working in groups of 3 students each group will provide a 4-6 page report on an Internet protocol. The report will be submitted using BB portfolio tool as well as a hard copy of the report |  |       |             |
| ▪ Quizzes: (best 8 out of 10)  | 35%  |       |             |
| ▪ Test 1 – weeks 1-5 (Data Comm)   | 15%  |       |             |
| ▪ Final Exam – weeks 7-11 (Networking)   | 20%  | Total | 70%         |
| ▪ <b>Grand Total</b>   |  |       | <b>100%</b> |
| ▪ <b>Note:</b>   | <b>Students must pass the Final exam and all tests with a weighted average of 50% to pass the course</b> |       |             |
|  |  |       |             |

# Success Factors Summary

- Attend class
  - Do not multi-task
  - Make notes from Word document file
  - Come prepared to ask questions in during online session
  - Complete course work on time
  - Review notes and study hard
  - Write all the quizzes and study hard to do your best

# Terminology

Data Communications v Networking

# Terminology

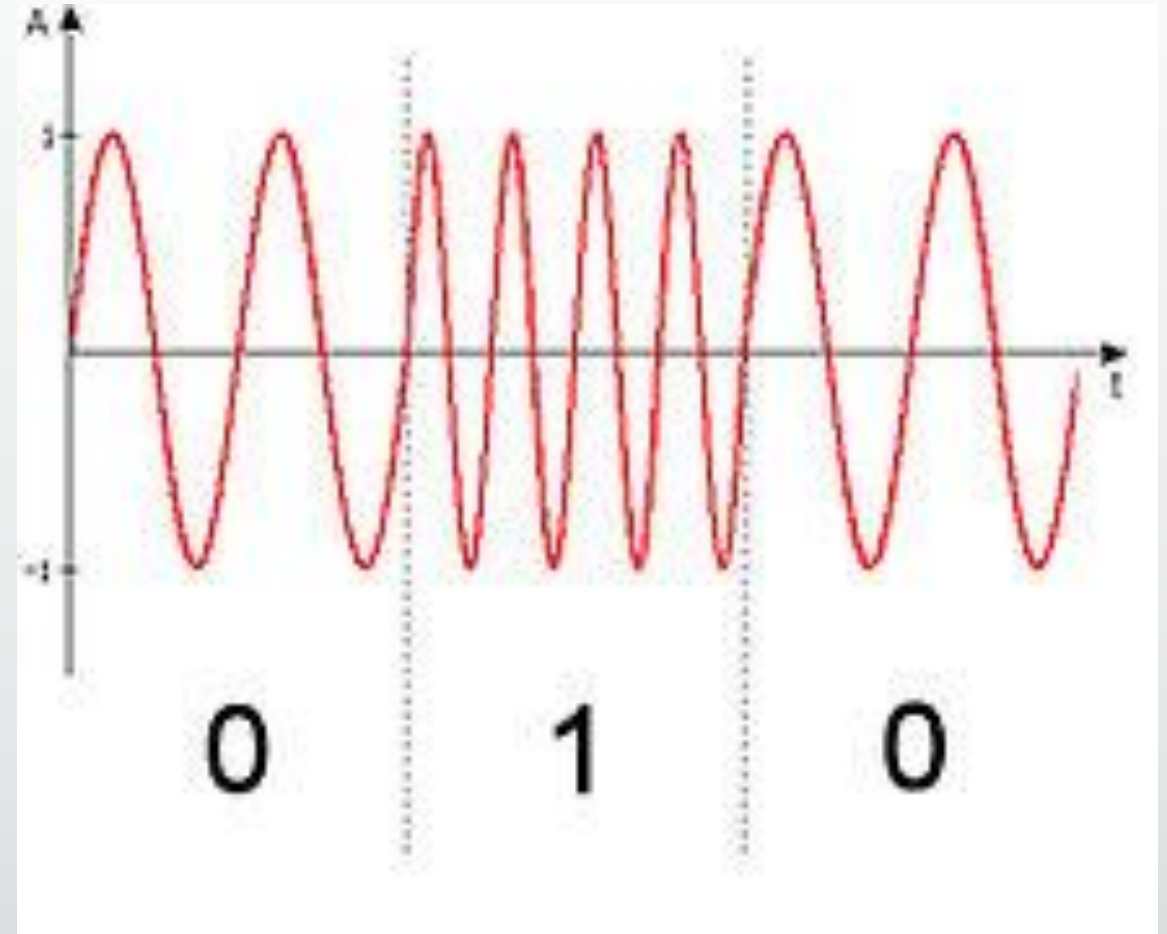
- A computer is a programmable machine that can perform various computations, store data, and create documents
- It follows a set of prerecorded instructions called a program.
- The data generated or stored by the computer is a collection of zeros and ones.





# Terminology

- Data communications :
- “is the sending of signals that represent zeros and ones over a point-to-point circuit between two computers.”



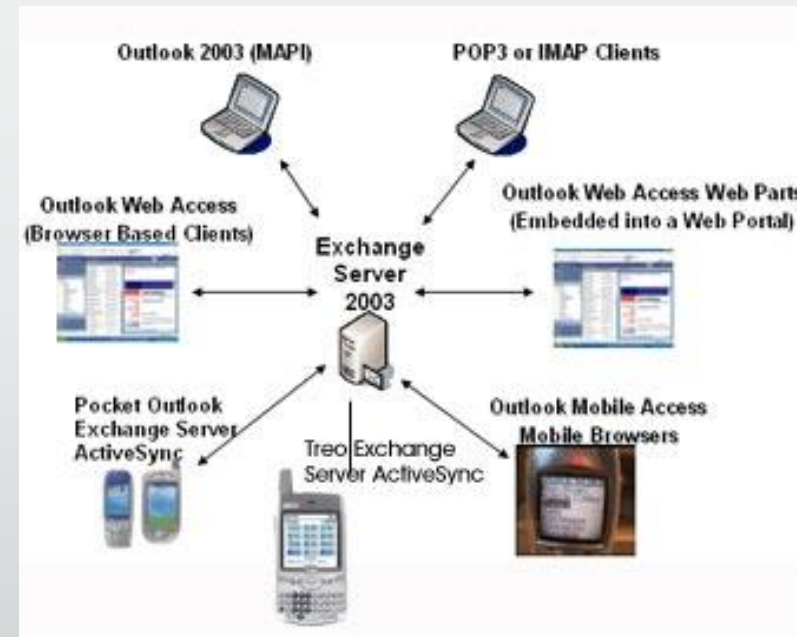
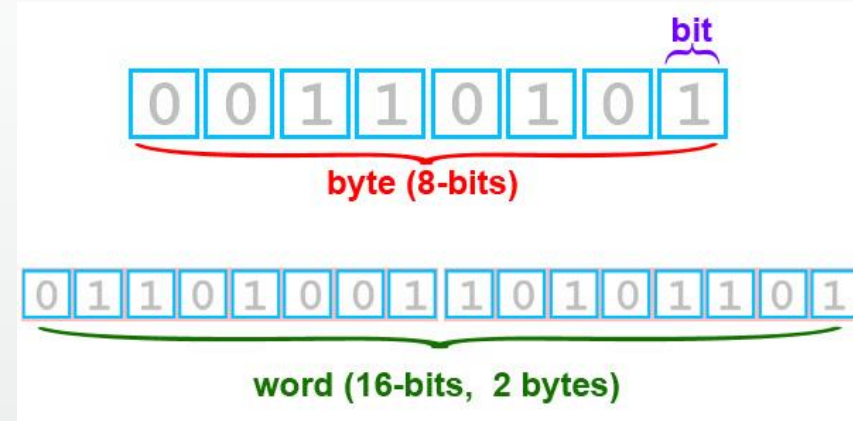
# Terminology

- Networking :
- “begins when point-to-point circuits are joined together into a collection of computers for the exchange information and the sharing of resources.”



# Summary

- Data Communications is concerned with bits and bytes of digital data getting from point A to point B across a direct link.
- Networking is concerned with the interoperability and exchange of information among applications

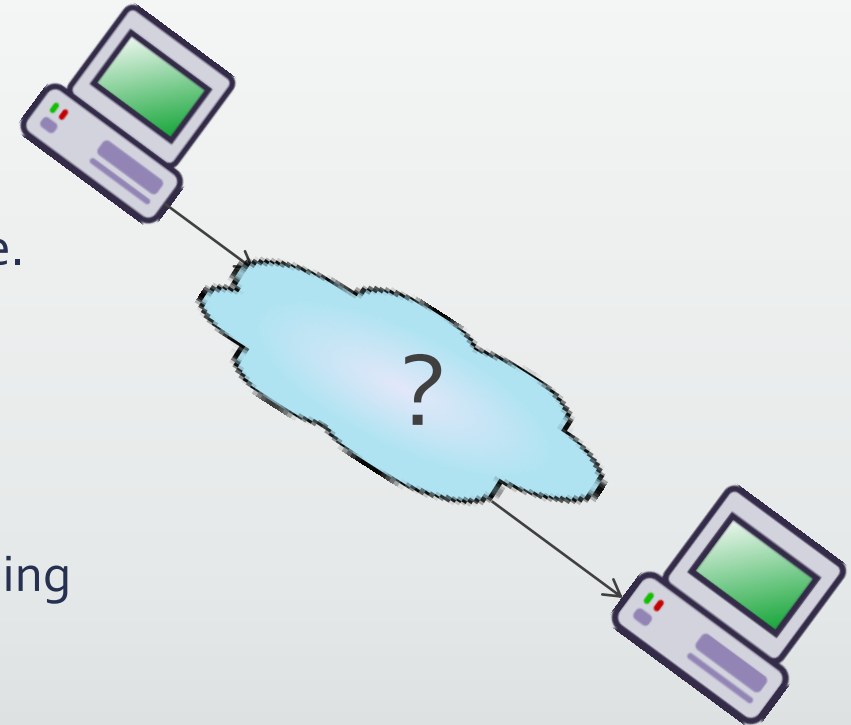


# Data Communications v Networking

I'm a programmer, why do I need to know about data communications?

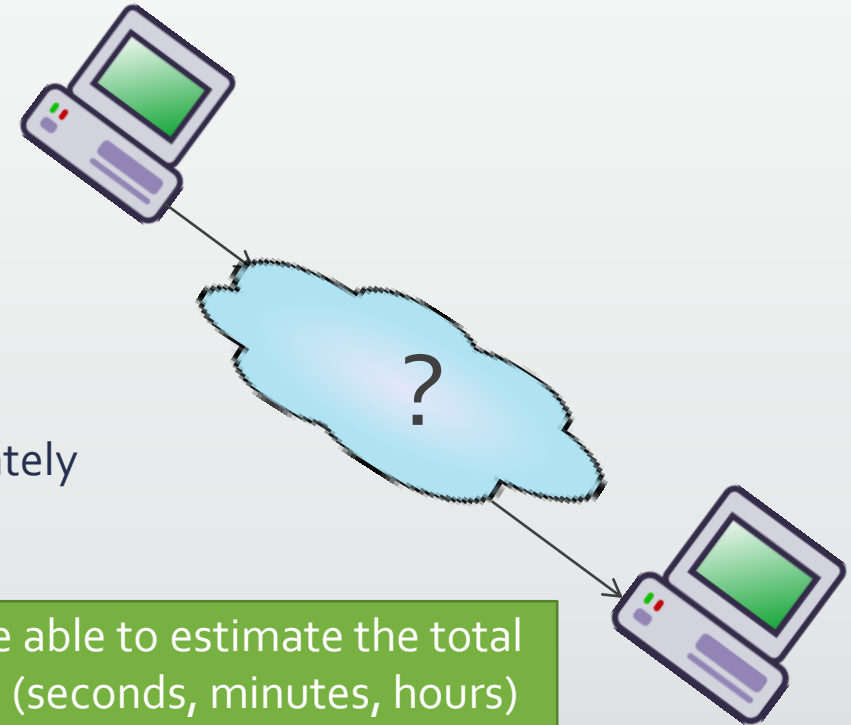
# 1. Programmers need to know what happens in the network cloud to write useful applications.

- Early days of networking not necessary for programmers to understand networking
- Today programming and networking “converging”
- Program code and data no longer on same machine. Now stored on different machines.
- To write effective programs, programmers need to understand how the network is structured
- Networking personnel must understand programming in order to set up the ACLs properly



# Example: How many bits and how long to transport one 8 1/2 X 11 page of information?

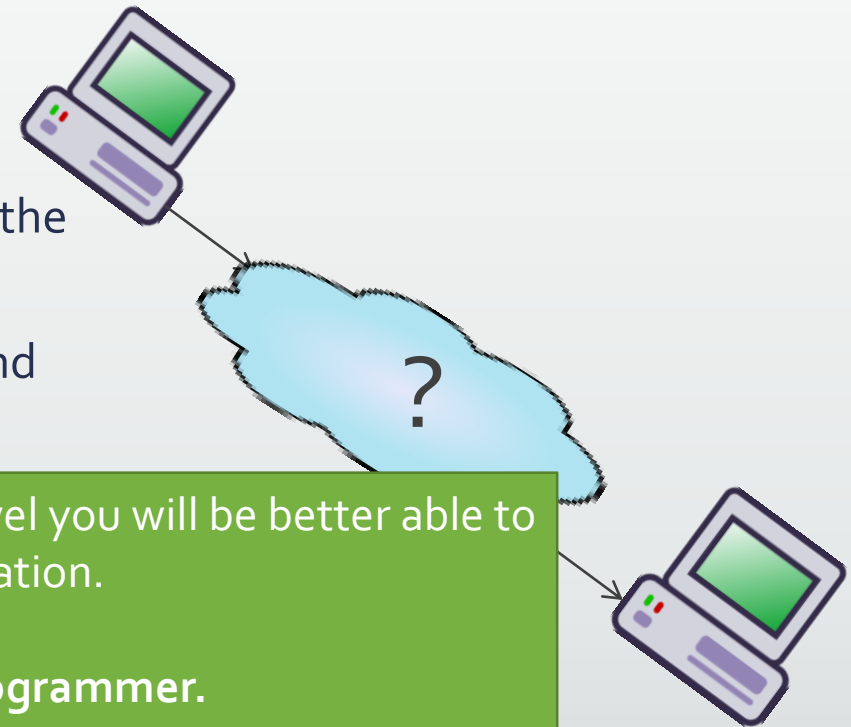
- 10 characters X 6.5 inches X 27 lines = 1,755 characters
- Using UniCode encoding:  
1755 characters X 16 bits = 28,080 bits per page
- How long will it take to transmit?
  - Dialup @ 56000 bps =  $28,080 / 56000 = 0.5$  seconds approximately
  - 4G @ 1.5Mbps =  $28,080 / 1500000 = 0.019$  approximately



To properly design a mobile application the programmer must be able to estimate the total amount of data to be transmitted within a particular time period (seconds, minutes, hours) to know how much throughput is needed by the network

# Network speed is affected by:

- Weather – rain or snow can slows the network
- Bandwidth is share so the more users the slower the network
- Distance from Cell tower – the further way the slower the network
- Amount of Network Congestion – latency at towers and routers slows the network

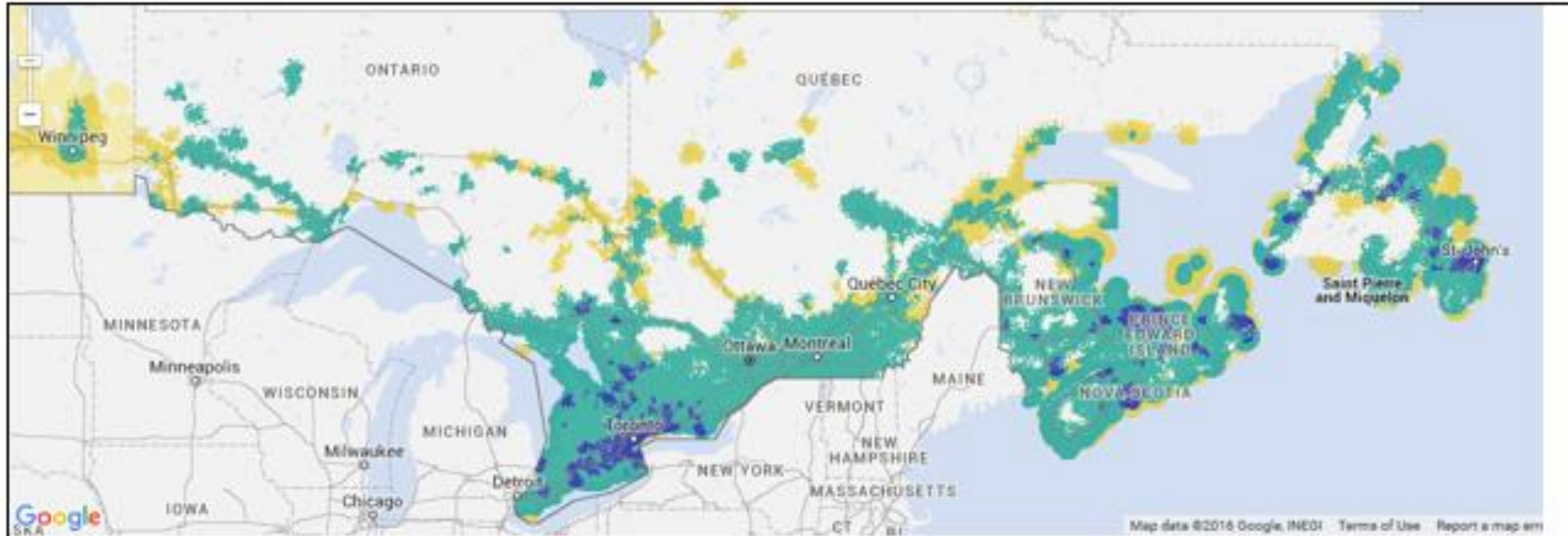


By understanding the layers of abstraction below the network level you will be better able to write and to troubleshoot a network application.

The goal is for you to be a more productive programmer.



# Canadian Network Speeds Summary



## Legend





|   |                  |
|---|------------------|
|  | Dialup 56 kbps   |
|  | 4G – 1.5 Mbps    |
|  | LTE – 2.5 Mbps   |
|  | LTE-A – 5.5 Mbps |

Table 1: Partial Map of Mobile Telephone Networks in Canada



## 2. Programmers must also know how hackers can exploit applications

- Writing secure applications means you understand how hackers can exploit applications for malicious purposes
- Programmers need to understand how hackers will try and use the application in ways it was not intended.
- Buffer Overflow, SQL infection attacks are examples where the programmer was overly trusting of user input.



# Programmers must also know how hackers can exploit applications

- Everyone knows about IP address such as 192.168.0.1, but as programmers, we need to understand the process of using sockets in applications and how application data can be changed by malicious individuals.
- A good mobile application will have a “choke-point” where all input data will be checked to ensure the data is in the format and length expected by the application.
- And if not, how to sanitize the input using stored procedures or regular expressions
- A common exploit is MAC spoofing or ARP Cache poisoning

```
import socket
import sys

HOST = '' # Symbolic name, meaning all available interfaces
PORT = 8888 # Arbitrary non-privileged port

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print 'Socket created'

#Bind socket to local host and port
try:
    s.bind((HOST, PORT))
except socket.error as msg:
    print 'Bind failed. Error Code : ' + str(msg[0]) + ' Message ' + msg[1]
    sys.exit()

print 'Socket bind complete'

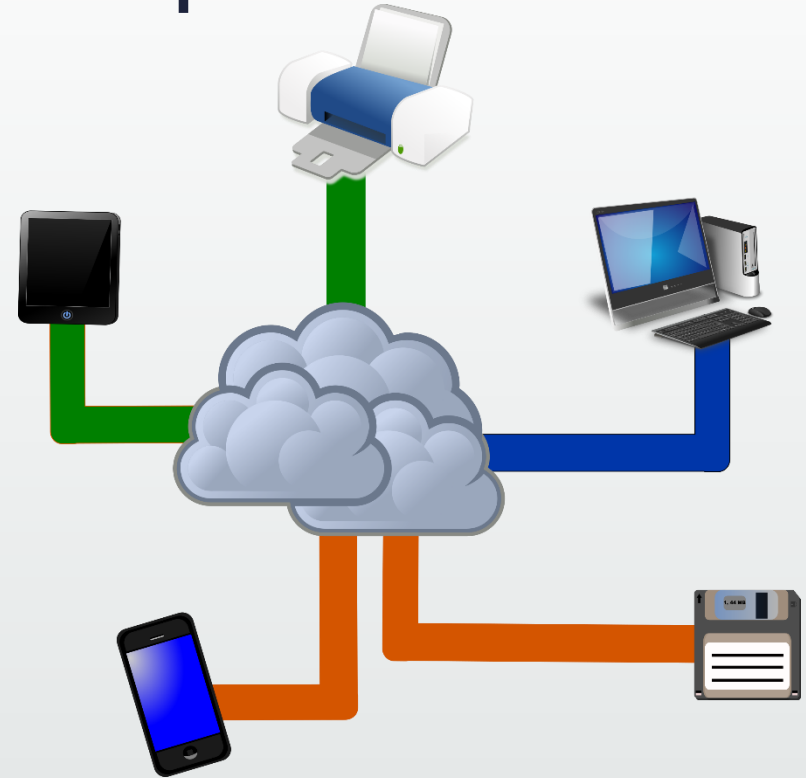
#Start listening on socket
s.listen(10)
print 'Socket now listening'

#now keep talking with the client
while 1:
    #wait to accept a connection - blocking call
    conn, addr = s.accept()
    print 'Connected with ' + addr[0] + ':' + str(addr[1])

s.close()
```

### 3. John Gage has described the IT environment of today as “The network is the computer.”

- The Cloud is now a development platform
- The cloud can run programs and store data just like the computer
- And through virtualization the cloud is a network of thousands of computers with unparalleled processing power.

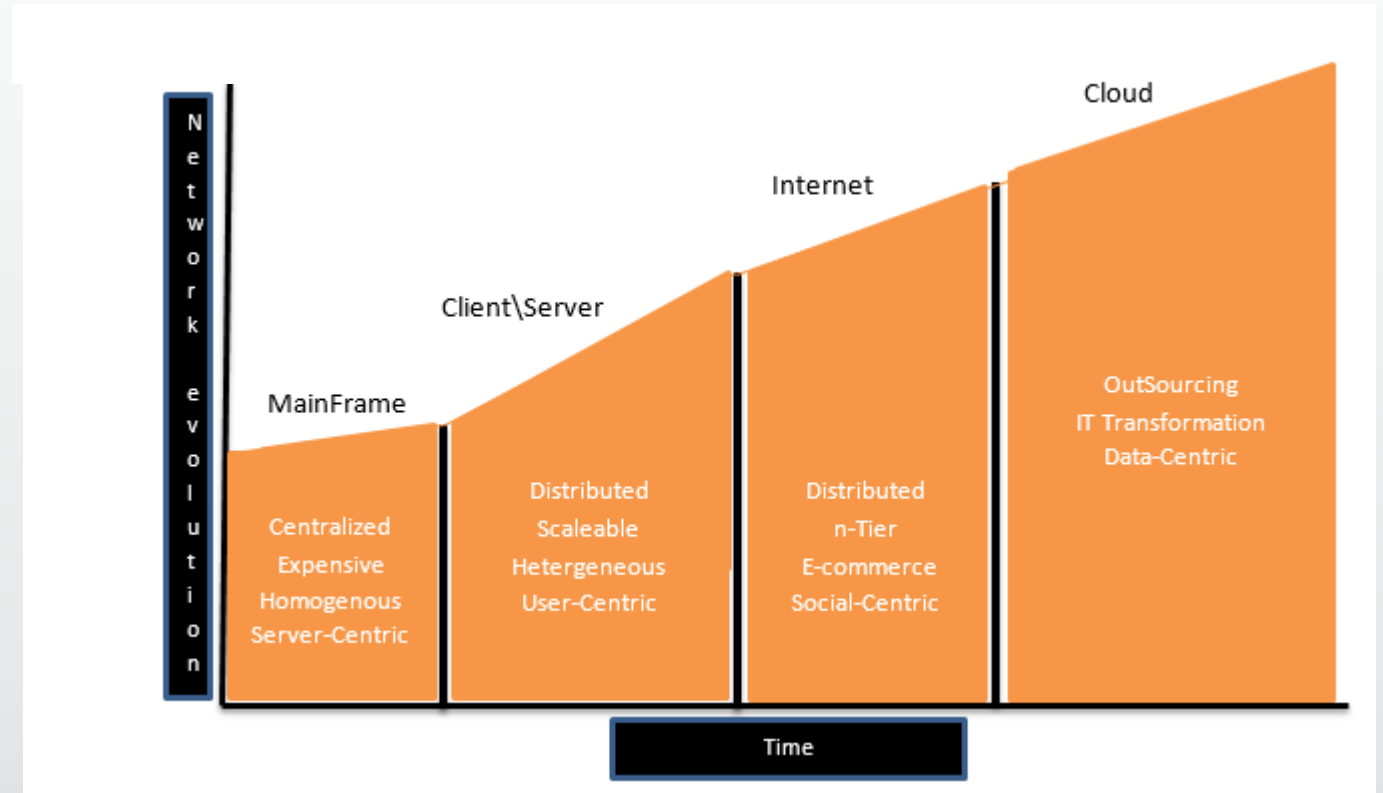


# Communication Networks

A Brief History

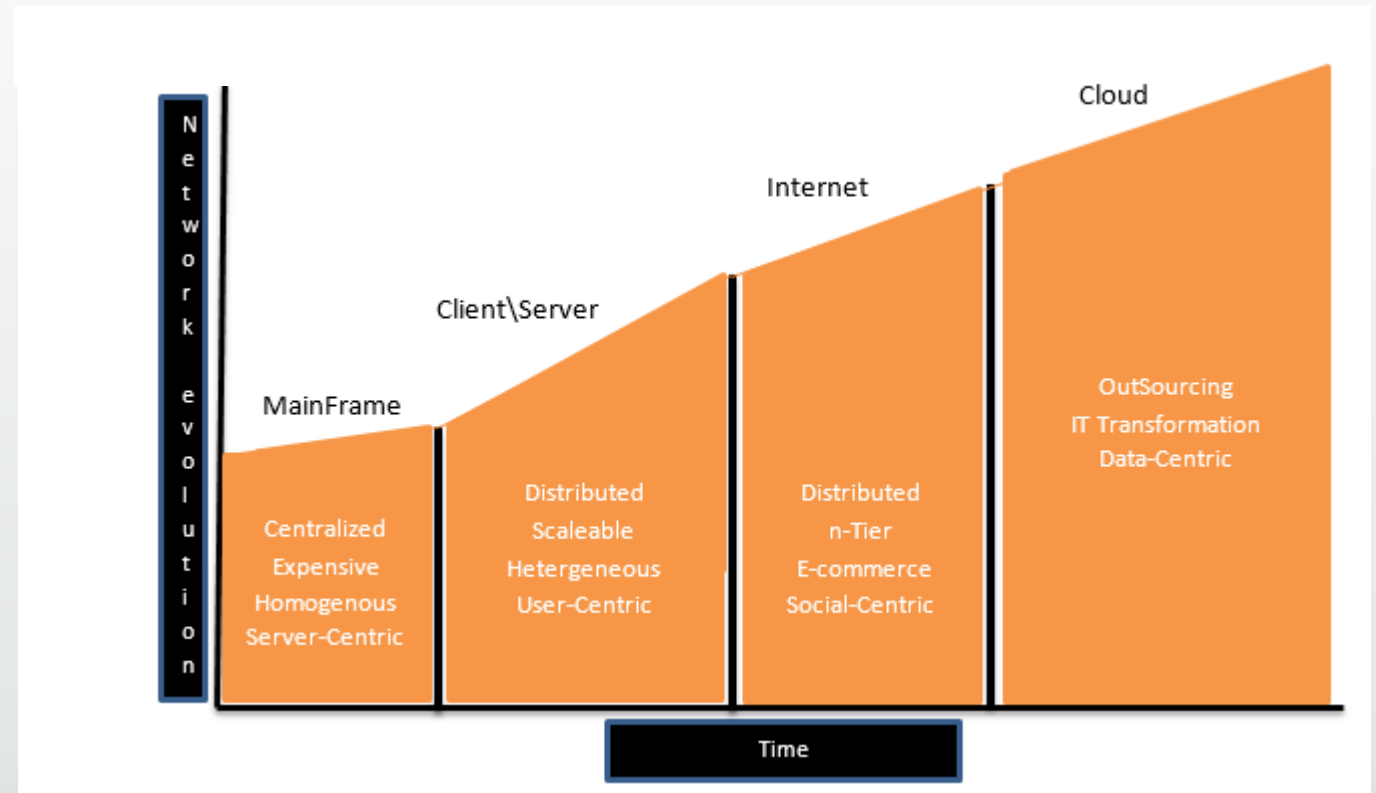
# Host to Mainframe: 1950-1979

- Mainframe is a large powerful unit
- Centralized processing, meaning that the data and executable code ran on the same machine
- Mainframes can only “talk” to dumb terminals, not PCs
- Very expensive, and usually hardware and software sold from same vendor



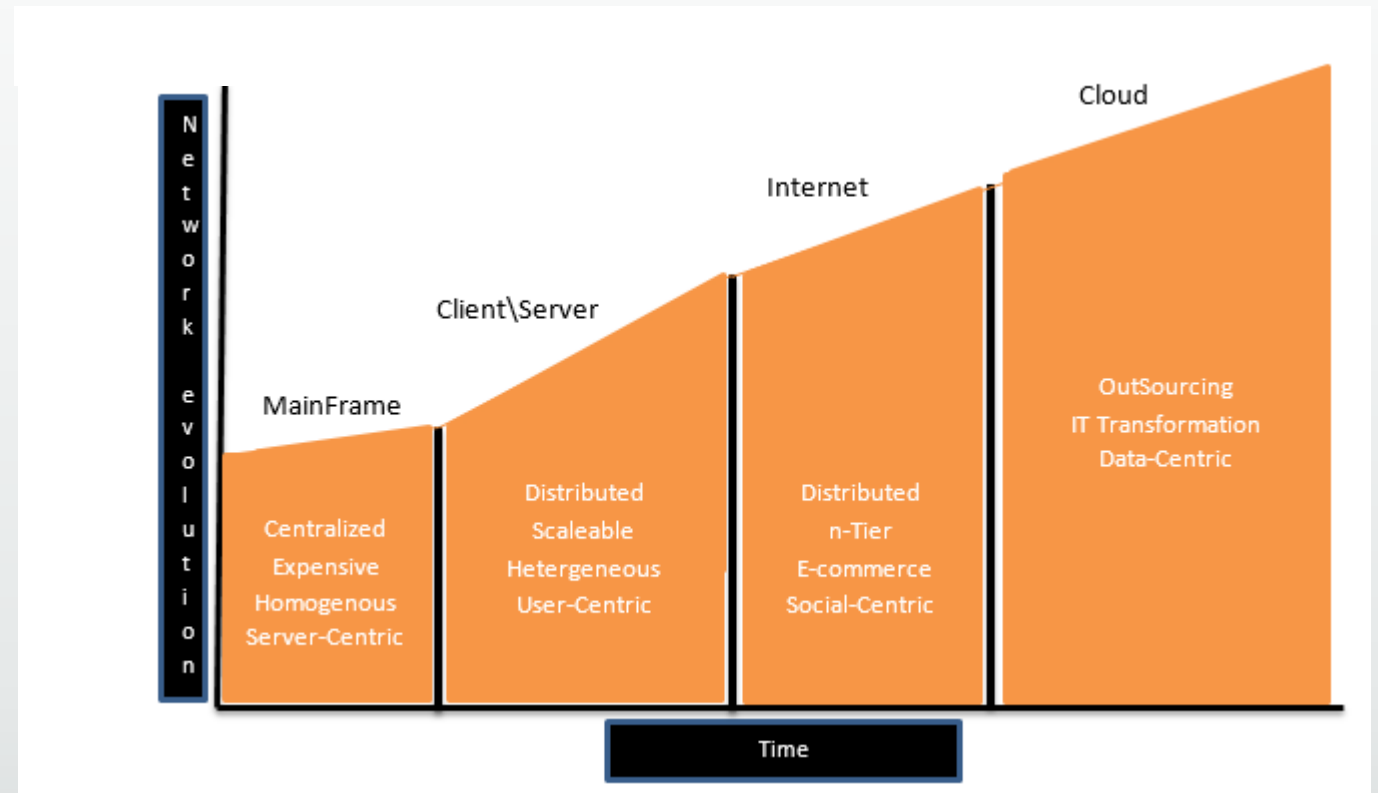
# Client-Server: 1980-1990

- Client-Server architecture developed from the marketing of the PC
- Processing was distributed, meaning that the code and data did not have to be on the same computer.
- Led to a heterogeneous system where hardware and software not provided from same vendor
- Client-Server more affordable and led to 2-Tier programming. Client interface front end with a server processing back end

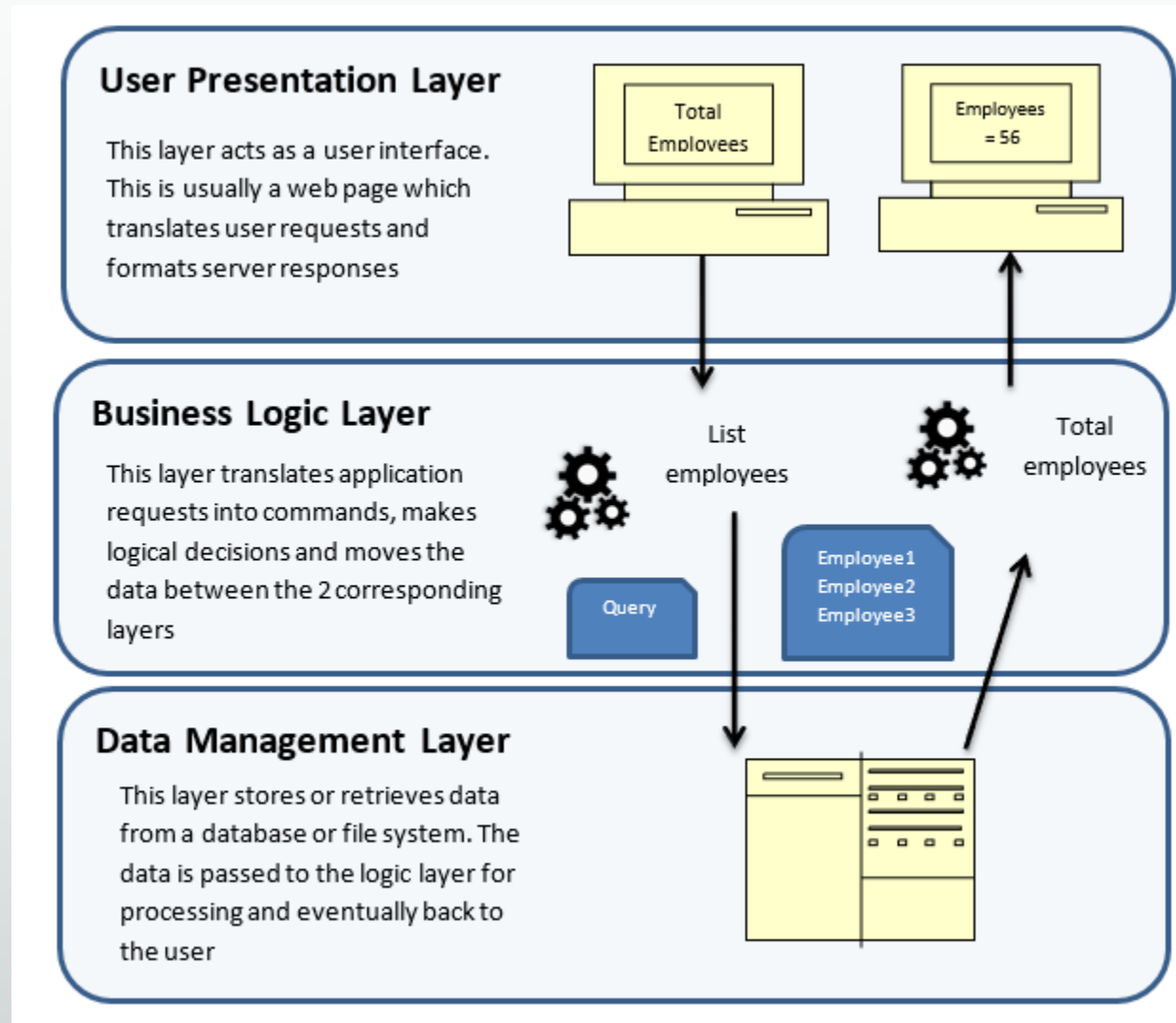


# Internet: 1991-2009

- Led to n-Tier Programming where the client interface, business logic, and data are stored on different machines
- Led to a more heterogeneous system because vendor's started producing "middleware" to make software and hardware from different vendors work together
- Greatly lowered the cost of LAN networks



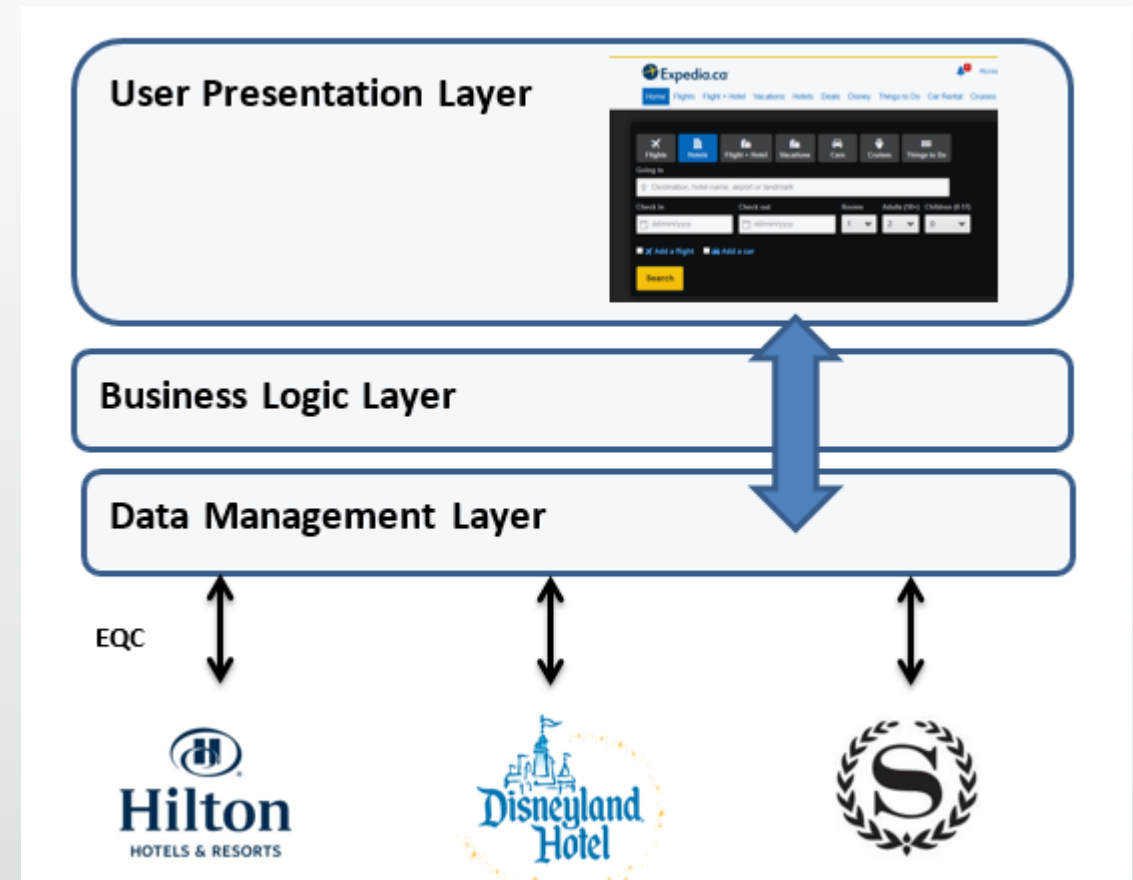
# n-Tier Programming





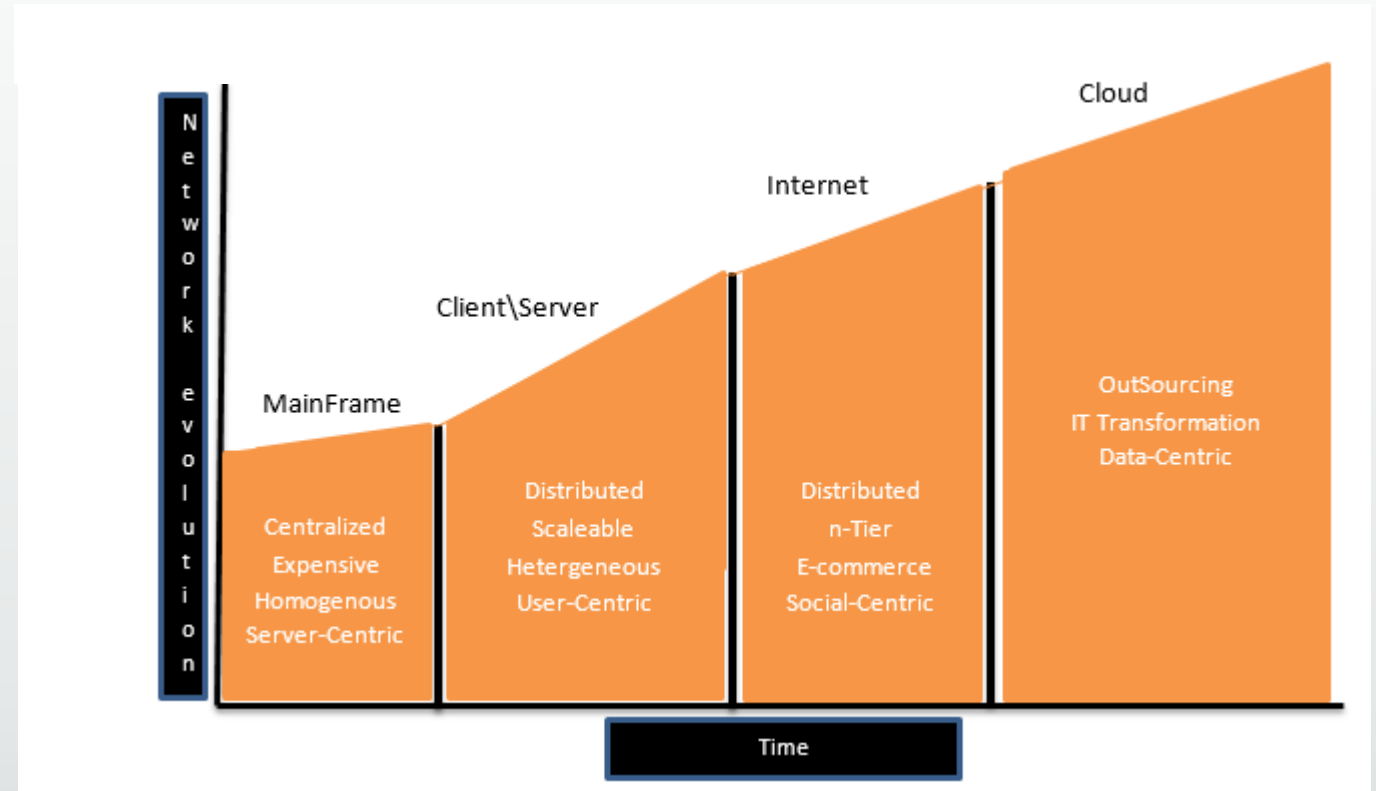
# Expedia Example:

- Originally developed by Microsoft as a research project in n-Tier Programming Expedia is now a premier travel provider
- Hotels have time sensitive inventory for example a block of rooms which sell for \$170 a night, but have not sold and its early afternoon.
- Hotels use Expedia's Quick Connect software to list that block of rooms available for sale directly to Expedia's Data Management Layer. Expedia buys these rooms, e.g. \$95
- When you go on the Internet to search for deals, Expedia's User Presentation Layer displays the rooms which Expedia resells to you e.g. \$120-\$130



# Cloud-Computing: 2010- present

- Cloud computing is an extensive of Client-Server architecture
- Cloud is “data-centric” allowing users to connect to the network with a smart phone or computer and access information from anywhere
- Cloud computing increases the use of “out-sourcing” for software (SaaS), infrastructure (IaaS), or platform (PaaS) services
- Cloud computing will revolutionize how IT handles network services, fewer personnel will be needed
- Cloud has thousands of virtual clients which can process data faster than in-house systems. This will change the nature of business systems, such as buying and, selling



# Cloud and Programming

- The Cloud has now become a development platform, Microsoft Azure, Google's Cloud
- .The 15<sup>th</sup> version of MSOffice is fully compatible with the cloud to allow users to use the cloud as their local PC.
- Developers are rewriting existing programs to run on clouds
- Popular programming languages for cloud are: SQL, Python, R Math, Erlang, XML Data, Clojure Math Language, GO, GFM domain, RUBY, PHP, JAVA etc.
- Joseph Hellerstein has developed a new language called [BLOOM](#) which will help programmers write complex data-centric cloud applications.



# Summary

1. Programmers should understand data communications in order to be better programmers, designing useful mobile applications that are consistent with network speeds and throughput.
2. Programmers should understand data communications in order to know how hackers can take advantage of programming mistakes. This will help programmers to write more secure applications
3. Lastly, programmers should understand data communications because the world of networking and programming is forever changed with the advent of client-server architecture and cloud computing. The latter is changing the nature of programming and business.