EE 382N: Distributed Systems **Fall 2017**

Lecture 1: Aug 18

Lecturer: Vijay Garg *Scribe: Pankaja A*

1.1 Introduction

This is the first lecture for this course. The professor started with a walk-through of course contents and grading policy.Please refer to canvas for this information.

The main topics discussed in this lecture are:

- Goals of the course
- What are distributed systems
- Puzzles

This set of lecture notes will briefly re-examine the topics covered in this lecture, in the order in which they appeared during class.

1.2 Goals of the course

The goals of this course is to have basic understanding of the topics listed in syllabus(refer canvas). At the end of this course, write programs using sockets.

1.3 What are distributed systems?

1.3.1 Concepts: Time

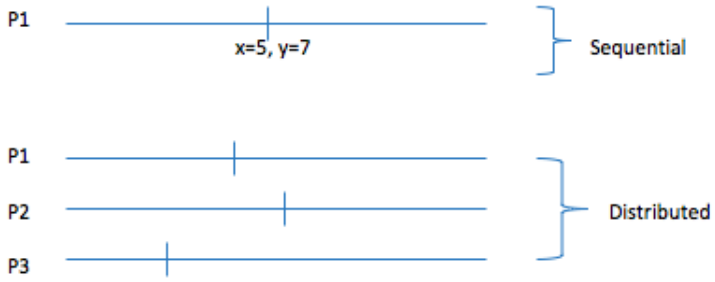
Consider two events e and f in sequential world. Lets say event e happened at $08:50$ and event f happened at $08:55$. Now, just based on the time at which these two events happened, we can certainly say that "f must have happened after e". This only holds good for sequential world though.

Now lets consider distributed world. In distributed systems, it is impossible to synchronize the clocks. The only way to synchronize is to send messages and receive messages.

Say we have a process $P1$ running. In sequential world it is possible to get one particular instance of this execution and obtain its value. This is possible as there is a shared memory and also the clocks are synchronized.

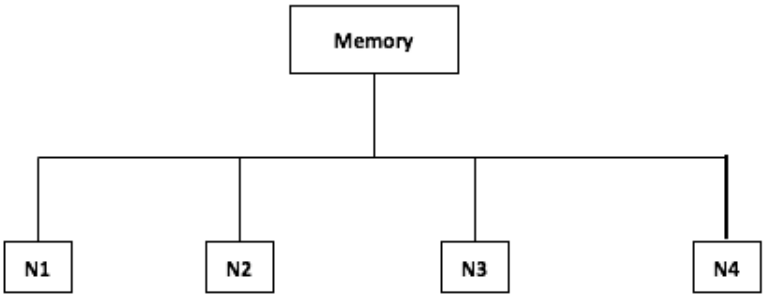
But, in distributed systems - there is no shared clock and there are no synchronized clocks.Hence it is not possible to define one particular instance in the execution of all these processes. Also, remember that these

process are running at different locations, one process may be running in one room and other process may be running in some remote satellite.



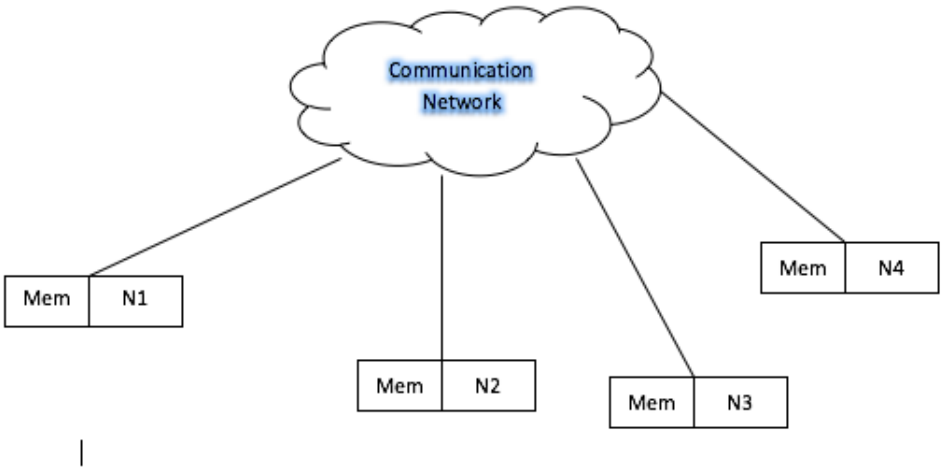
1.3.2 Centralized v/s Distributed

Consider the centralized system as pictured below.



N1,N2,N3,N4 are the nodes and the memory is shared by all the nodes. Memory is not scale-able here and thus it is bottleneck.

In Distributed system(as pictured below], all the nodes are connected via a communication network.Every node has its own memory, and there is no shared memory.



Lets say, if a node $N1$ sends message to another node $N2$. If $N1$ does not hear back from $N2$ - there can be two possibilities. Either node $N2$ is dead or the communication link itself is slow. Since its not possible to differentiate between these two cases, agreeing on a particular state would be a problem.

Thus a Distributed system is the one which has

- No shared memory
- No shared clock
- No perfect failure detection

1.4 Puzzles

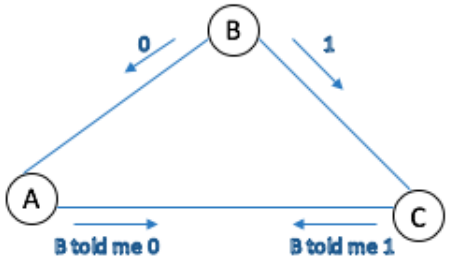
We covered two puzzles in this lecture.

1.4.1 Byzantine general agreement problem

This problem was introduced by 3 people - Lamport, Shostak and Pease in 1982.

This problem is built around an imaginary General who makes a decision to attack or retreat, and must communicate the decision to his lieutenants/other generals so that the resulting action is coordinated - either all attack or all retreat.

Some of these generals are disloyal.

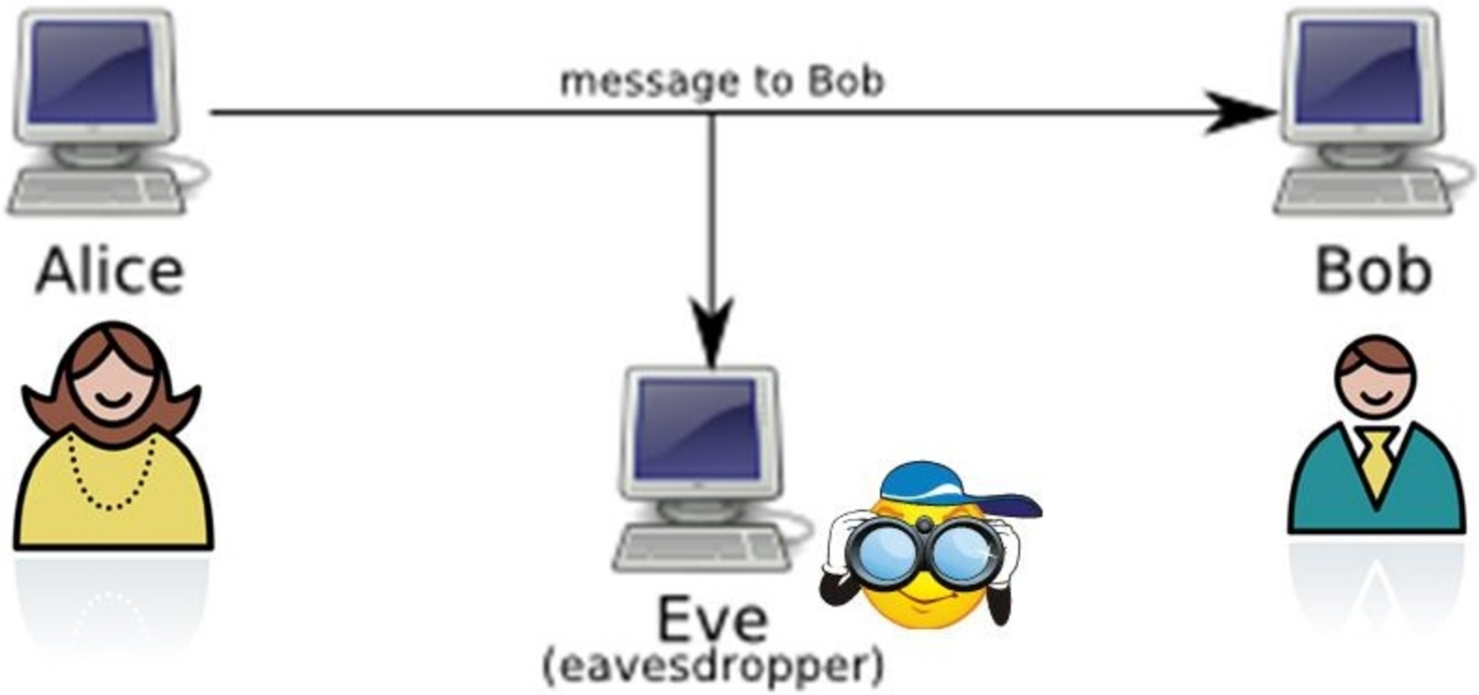


Consider a system of 3 generals, and lets say one of them (B) is disloyal. B sends message 0(retreat) to A and sends message 1(attack) to C. Now A and C try to validate their messages, nothing can be inferred as to who is disloyal and sending conflicting messages. Its equally possible that A or C is disloyal and sending wrong message even when B send them consistent message.

Thus when the number of nodes is 3, and if there is 1 disloyal general, there is no solution.

To generalize,if $1/3^{rd}$ of the people are disloyal then there is no solution.

1.4.2 Alice,Eve and Bob



This puzzle has 3 people - Alice, Eve and Bob. Alice is calling Bob for the first time to arrange for a movie/dinner date. Now, there is Eve who is listening to the conversion(on a line) between Alice and Eve. There is nothing shared between Alice and Bob.The goal is Alice needs to communicate the intended place/time for a date to Bob without Eve getting any information regarding the same.

Solution 1:

- Alice prepares her message, put a lock on the message and sends to Bob.
- Bob receives the message, puts his lock on the same message and sends back to Alice.
- Alice receives the message from Bob, Alice removes her lock and sends the message back to Bob, He removes his lock and he can read the message.

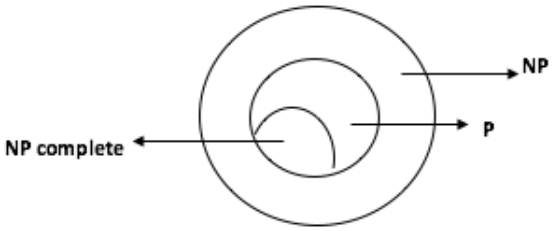
The assumption for this solution to work is that the encryption is commutative.

Solution 2:

- Bob sends his public key to Alice
- Alice encrypts it and send it back to Bob. Bob can then decrypt it and read the message

1.5 P v/s NP

There was a question asked by one student regarding P v/s NP problems, so professor briefly covered it in response to his question.



Most people believe $P \neq NP$

NP problems is a class of problems using which one can verify efficiency.

P problems is a class of problems using which one can compute solutions efficiently.

Both P and NP problems will be covered in detail in subsequent lectures.