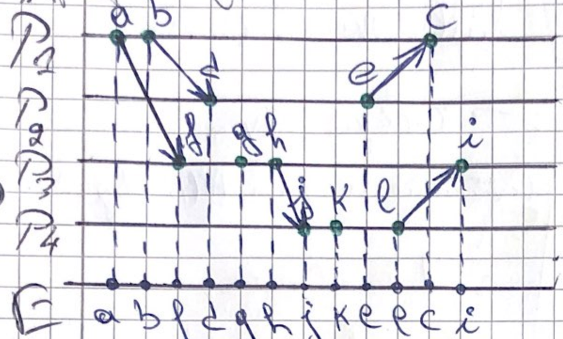
Homework #02

Problem 2.1:



* Causality Chains:

P_1: $a < b < c$

P_2: $d < e$

P_3: $f < g < h < i$

P_4: $j < k < l$

$\forall e \in S_e$
$\exists e' \in r_e$ s.t $e < e'$

$a < f$
$b < d$
$e < c$
$h < j$
$l < i$

$b \| f$
$c \| j$
$e \| h$

* Lamport's Logical Clock:

$(\forall e \in V_i)\ \theta_{p} := \theta_p + 1;$

$(\forall e \in V_s)\ \theta_{p} := \theta_p + 1;$

$(\forall e \in V_r)\ \theta_{p} := \max(\theta, \theta_p) + 1;$

Med Lamine Cherfi
CA-CS-802

Considering a is the first event

$\theta_{p_1} := (1, 2, 5)$ $\theta_a = 1$

following the causal order relation $<$

$\theta_{p_2} := (3, 4)$

$\theta_{p_3} := (2, 3, 4, 8)$

$\theta_{p_4} := (5, 6, 7)$

The clock values for each of the events concerne with the previously Causalities chains presented

$\theta_a = 1; \theta_b = 2; \theta_f = 2; \theta_d = 3; \theta_g = 3; \theta_h = 4$

$\theta_j = 5; \theta_k = 6; \theta_e = 4; \theta_l = 7; \theta_c = 5; \theta_i = 8$

Problem 2.2:

$\Longrightarrow$ Ring: 1. given N nodes, we have N edges as every Node is connected to its neighbours forming a cycle. circular data path.

2. Advantages: Simplified topology. • Data flows in one direction (if unidirectional) which reduces chances of packet colision. • Additional stations can be added later without affecting Network performance. • Equal access to resources • Cheap to install and expand.

Disadvantages: If Unidirect link has to pass through all nodes

• If one Node fails, whole network fails

• Addition & removal of Nodes difficult may cause issue in Network Activity

- Difficult to troubleshoot ring

⇒ **tree**: given a tree network w/ N nodes, we have $N-1$ edges.

Advantages: Low communication Cost. • High Scalability as Nodes Can be added to leafs simply. (Leaf Nodes Control more Nodes)
- If one Node fails, others not affected
- Highly Secure & Reliable
- Easy fault detection & Maintenance

Disadvantages:
- if the first Level station is erroneous, the next Level station will also go under problems.
- Difficult to configure compared to other networks topologies.
- Dense Network traffic as Data needs to travel from central Cable
- Requires large N° of cables compared (Star, ring)

⇒ **Star**: given a N nodes star network, we have $N-1$ edges.

Advantages: • Reliable (1 cable or station fails others would still work.
- High Performance. (No Data collisions can occur)
- Less expensive (each station only needs 1 I/O port and 1 link connecting to Hub)
- Easy fault detection (links easily identified)
- No disruptions to Network when Adding or Removing stations

Disadvantages: if Hub goes everything else goes down.
- Performance is predicated on the one concentrator (Hub)
- Central system of star requires more resources & regular maintenance
- if connecting network device (switch) fails, Nodes attached are disabled.

⇒ **Clique**: given a N nodes clique topology, we have $N(N-1)/2$ edges.

Advantages:
- Low latency of data transmission between any two Nodes (Each Node has a direct connection to All others)
- High Redundancy, if one link fails, many others still available making it Highly Resilient to failures (Multiple D.C. provide)
- Simplified Routing. No need for complex Algos best path for Data
- D.C. Maximizes Bandwidth (transmission utilization, reducing congestion & Latency (No intermediary nodes) passes

Disadvantages:
- Scalability issues, N° of connections grows quadratically w/ N° of nodes $N(N-1)/2$.
- High Cost, need for Large N° of direct connections
- Complex Management & Maintenance
- Requires a Substantial amount of resources to maintain All D.C

D.C = direct connection.

- Physical Limitation, Physical Layout of cabling or wireless connections to support a clique topology can be challenging especially if Network grows.

⟹ HyperCube: given an n-dimensional hypercube, we have $N = 2^n$ nodes, each node has n connections, each shared by two nodes, avoiding double counting $E = \frac{N \cdot n}{2} = \frac{2^n \cdot n}{2} = 2^{n-1} \cdot n$

- Advantages: • n diameter for a Hypercube of n dimension (low diameter)
  - Fault Tolerance, HC provides multiple paths between nodes. If one path fails, data is rerouted.
  - Scalability, HC scales efficiently as network grows n-dim HC can connect 2^n N, providing a clear & structured way to expand network
  - Balanced Load distribution, regular and symmetric structure of H.C ensures it, preventing bottlenecks and evenly spreading traffic.

Disadvantages:
- Complexity in Design & Implementation as dimension increases. (Managing connec / propa functions)
- High Cost for large dim, Although H.C scales well Each additional dim doubles the N of required connections leading to a high infrastructure cost.
- Routing Complexity, though there're multiple path, selecting optimal one, requires sophisticated routing Algos to minimize latency & avoid congestion

Problem 2.3:

- In Order to ensure eventual delivery requirement, Balanced Sliding-Window Protocol requires both fairness assumptions satisfied F_1 & F_2

Counterexample:

Consider the Balanced Sliding-Window Protocol, where sender has a window of packets to send $P_1 \dots P_i \dots P_m$ ($\forall i \in N^*$)

Let's Assume packet P_i ($\forall i \in N$) is within the sending window and is applicable for sending for an infinitely long period. However, due to violation of F_1, P_i is not sent infinitely often, Sender only sends it for a finite N of times or not at all after some initial attempts. As F_1 does not hold there's not guarantee P_i will continue to be sent even though it's Applicable.

- Fairness assumption (F_2) ensures that if packets P_i were sent infinitely often, it would be received infinitely often. However, this assumption is irrelevant if P_i were not sent infinitely often in the first place.