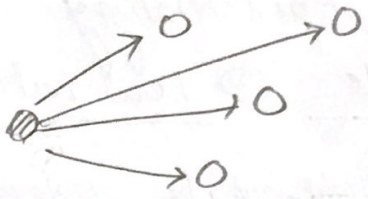
GOSSIP AND EPIDEMIC PROTOCOLS :-

① Multicast Problem

Problems → Fault Tolerance
→ Scalability

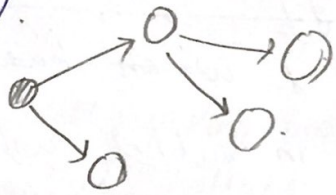


APPROACH 1 : Centralized :

① Sender sends all receivers UDP/TCP packets

Problem : ① Sender fails halfway. Only some receivers get packets.
② Sender overhead high.
③ Latency (As high as O(N)). → For last node

APPROACH 2 : Tree based



① Nodes develop a spanning tree

Ex. IP Multicast, SRM, RMTP, TRAM

⇒ Latency might be reduced to O(logN) in a balanced tree.
⇒ Less overhead.

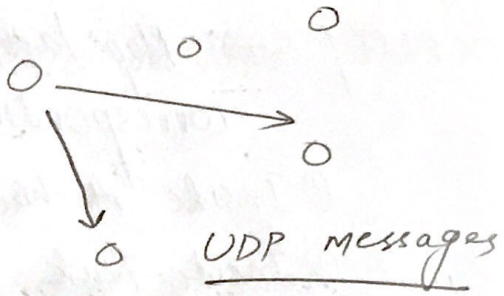
PROBLEM : ① Setup and Tree maintenance
② Failure of non-leaf requires maintenance & repair.

Generally use ACK/NAK (Can have O(N) ACK/NAK Overhead)

PROTOCOLS : ① SRM : → NAK
→ Exponential backoff to avoid overflow

② RMTP : → ACK
→ ACK sent to designated receivers.

APPROACH 3 :- GOSSIP PROTOCOL



Periodically transmit to 'b' random targets

↓

span / fanout

UDP messages

* After being infected by a gossip, it will start gossiping messages.

(Slightly higher messages but lower overhead)

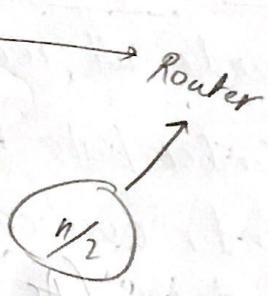
① Push Gossip ② Pull Gossip (request for messages) ③ Hybrid (Push-Pull)

ANALYSIS → Epidemiology

① Reliable ② Low Latency ③ Light weight

PROBLEM: Might not be Topology aware.

Traffic through router becomes high if nodes are picked at random



SOLUTION: Topology-Aware GOSSIP

In subnet 1 which has n_i nodes pick target in subnet with $(p = 1 - 1/n_i)$

Router load = O(1) from each subnet