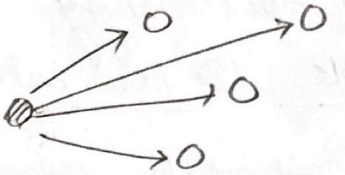
GOSSIP AND EPIDEMIC PROTOCOLS :-

(A) Multicast Problem

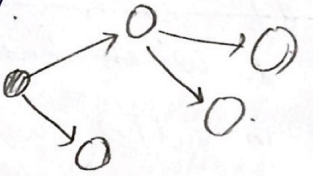
Problems → Fault Tolerance
→ Scalability



APPROACH 1 : Centralized :

(A) 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

(A) Nodes develop a spanning tree

Ex. IP Multicast, SRM, RMTP, TRAM

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

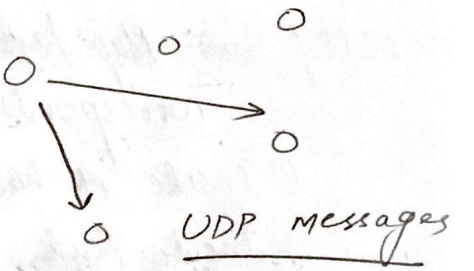
PROBLEM: (1) Setup and Tree maintenance
(2) Failure of non-leaf requires maintenance & repair.

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

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

(Slightly higher messages but lower overhead)

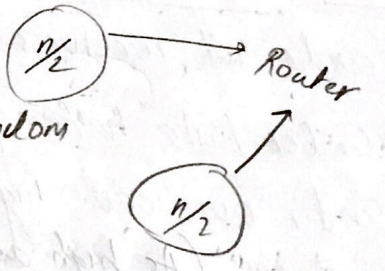
① Push Gossip ② Pull Gossip (request for messages) ③ Hybrid (Push-Pull)

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