Design:

Our distributed group membership service implements two failure detection protocols with integrated suspicion mechanisms: **Gossip** and **SWIM (PingAck)**. The system maintains full membership lists at each node. The architecture consists of five core components: a centralized Membership Manager for thread-safe state management, separate Protocol Handlers for Gossip and SWIM protocols, a UDP-based Message System, a modular Suspicion Engine integrated into both protocols, and an RPC-based CLI Interface.

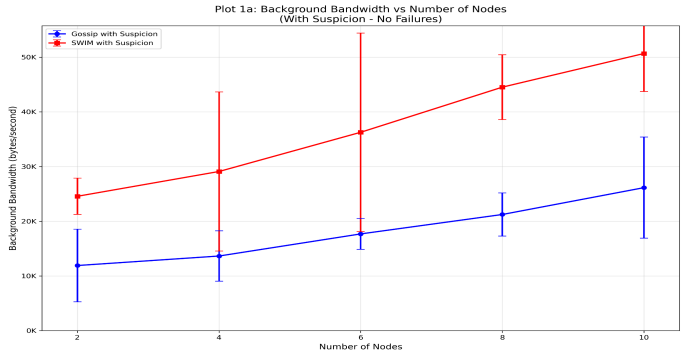
The **Gossip Protocol with Suspicion** operates by having each node periodically select a random target and send its information via UDP. The suspicion mechanism is integrated such that when a node's timestamp exceeds the `Tsuspect`, it transitions to the `Suspected` state. After a `Tfail` timeout, suspected nodes transition to the `Failed` state and are eventually cleaned up after `Tcleanup`. The protocol employs a round-robin approach with a random permutation for fair target selection and relies on timestamp-based detection. Each gossip message contains the complete membership state to ensure rapid propagation and eventual consistency.

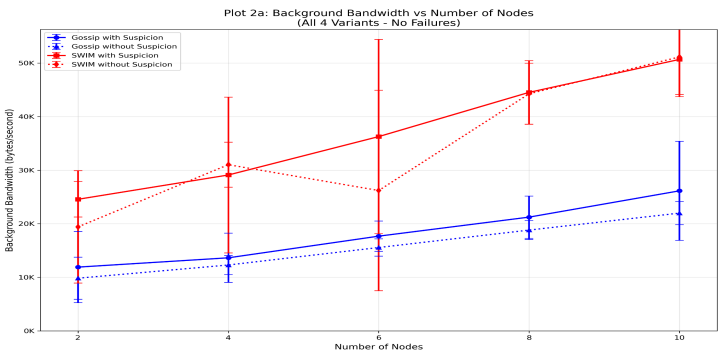
The **SWIM Protocol with Suspicion** implements a two-phase detection mechanism where each node periodically pings a random target and waits for acknowledgment. If no direct acknowledgement is received within `TpingFail`, the node is moved to an indirect ping queue, where 3 other members are asked to ping the suspected node. If indirect pings time out after `TpingReqFail`, the node transitions to the `Suspected` state. The protocol maintains separate maps for tracking direct and indirect ping acknowledgments, ensuring reliable failure detection while reducing false positives through the two-phase approach. Suspected nodes are propagated through ping/pong messages and eventually transition to the `Failed` state after a `Tfail` timeout.

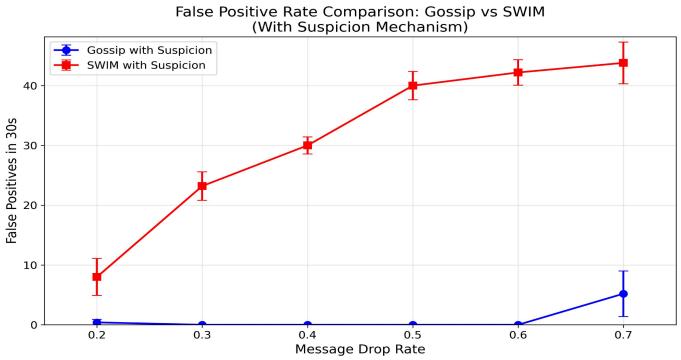
Protocol Switching is implemented through a dynamic switching mechanism that allows seamless transitions between Gossip and SWIM protocols without stopping the system. When switching, the system broadcasts `UseGossip` or `UseSwim` messages to all members, ensuring coordinated protocol changes. The switching mechanism preserves the existing membership list and maintains continuity of failure detection during the transition.

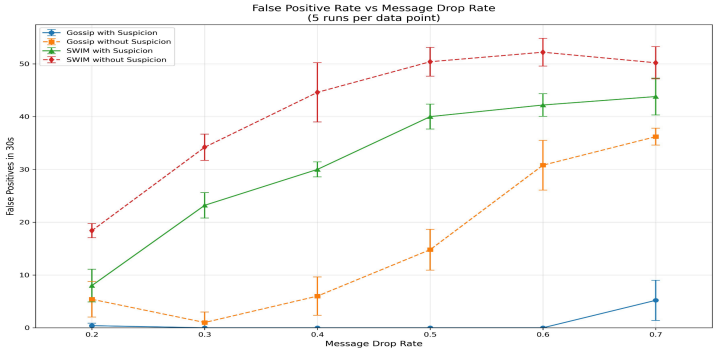
The design meets the **6-second completeness** requirement through fast detection and efficient propagation. Detection takes at most 3s: The gossip protocol uses a 1s timeout while the ping protocol uses a 500ms timeout to meet the 3s bound. Propagation takes ~1 second: each node contacts one random target every 100ms using round-robin with random permutation. With 10 nodes, each node contacts all 9 others within 900ms, while overlapping propagation from multiple nodes ensures complete coverage. The 6-second total provides a conservative safety margin for network delays and system convergence.

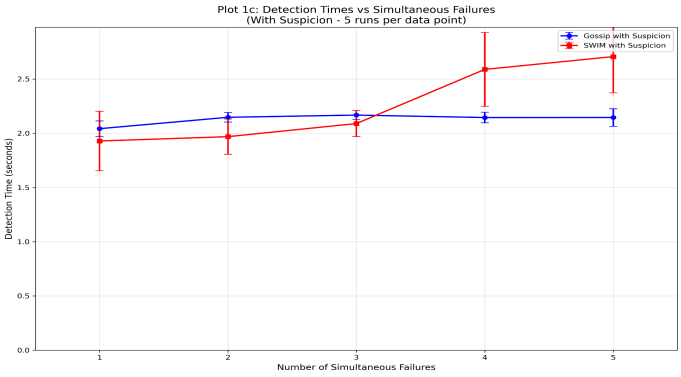
Analysis:

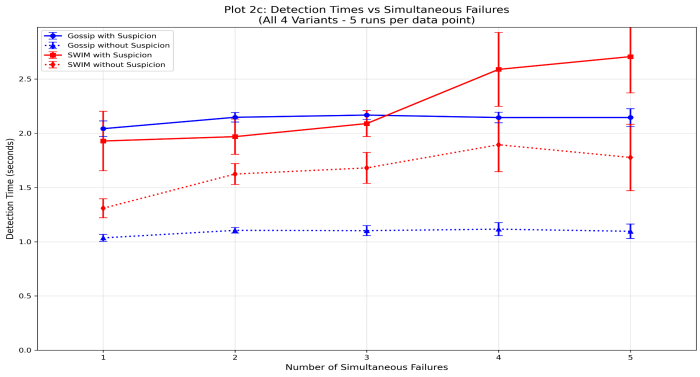












D: Gossip benefits more from suspicion. It maintains a near-zero false positive rate until the drop rate exceeds 60%, and the gap between the lines of Gossip with/without suspicion becomes larger as drop rate increases, implying that the impact of suspicion becomes more significant as drop rate increases. Gossip's simple timeout-based design is vulnerable to temporary message loss, and suspicion directly counteracts this weakness by adding a "grace period" for delayed heartbeats to refute an incorrect failure detection. While still beneficial to SWIM, its inherent resilience from indirect pings means the additional improvement gained from the suspicion is less significant.

E: Gossip w/o suspicion outperforms SWIM w/o suspicion. Besides fewer false positives, it has lower bandwidth by piggybacking heartbeats instead of sending active probes. It also maintains a consistent detection time of ~2.1s while SWIM's detection time increases with the number of simultaneous failures, exceeding that of Gossip after 3 failures. A possible reason can be that we are experimenting with high drop rates (20% to 70%), and SWIM's multi-message indirect pinging mechanism creates more opportunities for false positives when individual messages are dropped. The relatively small k (k=3 indirect pings) is insufficient to overcome this cumulative probability of failure in such lossy conditions.