Machine Programming 2 – Distributed Group Membership

An Phan(anphan2), Chang Liu(changl25)

Design

The system consists of multiple machines, each hosting a server running continuously. With network latencies considered, the objective is to provide time-bounded completeness in identifying machine failures, joins, or leaves within 5 seconds and 6 seconds for any membership list.

Components

Machine Servers: Each machine runs a server process responsible for sending and receiving heartbeat messages. These servers operate continuously.

Introducer: At the start, all machines are connected to an introducer and get introduced to the network

Membership Lists: Each machine maintains a local membership list containing information about all other machines in the group, their statuses, heartbeats, and timestamps.

Gossip-based Heartbeating: All machines periodically increment their heartbeat and send their membership list to *n_send* random members in their membership list. After they receive membership list from other, they update their membership list based on the heartbeat.

Gossip+S Mechanism: In the Gossip+S variant, machines maintain a suspicion status and incarnation number. This will be use to update the suspicion state machine.

Meeting 5 Second Completeness

Join bound:

The join bound is the same for both protocols, the introducer will spread the message about the new node when they join, it will take T_period*logN to be spread which is around 1 second.

Fail Bound:

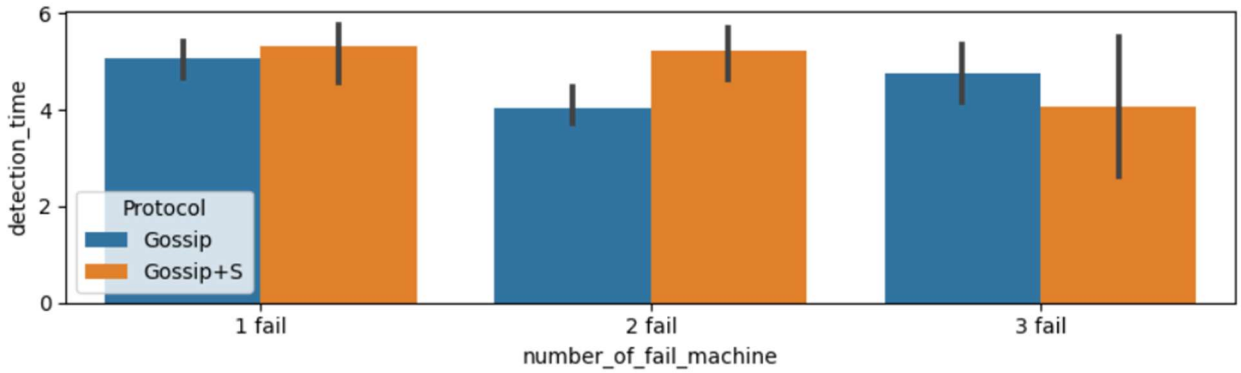
Gossip (T_fail = 4, T_cleanup = 4, T_period = 0.25, n_send = 3): It will take the last message from the fail node around 4 period which is 1 second to get to the detect node, then it will start time out with 4 second make the time bound be 5 second.

Gossip+S (T_fail = 2, T_cleanup = 4, T_suspect = 2, T_period = 0.25, n_send = 3): The same argument as above so it will take roughly 2 + 2 + 1 = 5 second for the fail node to be deleted.

Experimental Results:

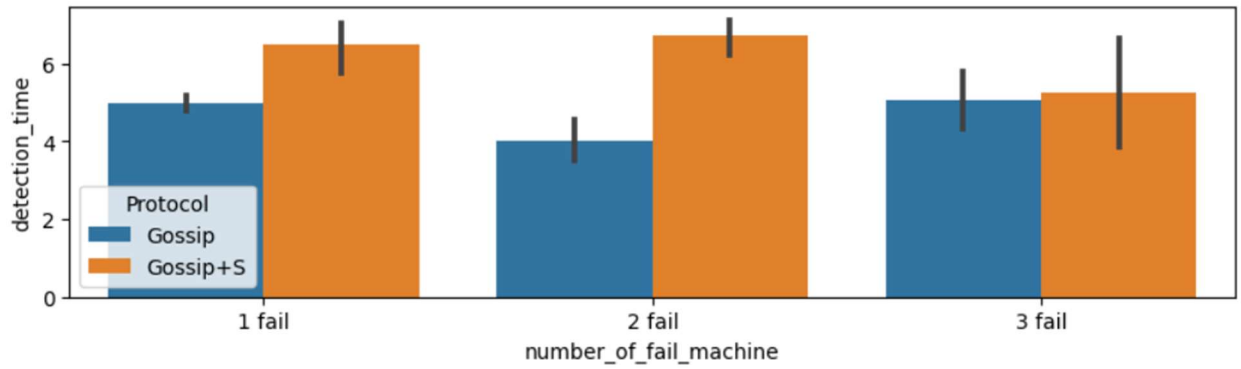
1. We fix the time bound with the parameters stated previously for the Gossip and Gossip+S protocol and run both on 10 machines for 10 minutes for 5 times.

	Gossip	Gossip+S
Bandwidth (bytes/s)	5840±674	8133±1023
False positive rate	2.47±0.34%	1.27±0.23%



2. We try to fix the bandwidth around the 5840 bytes/sec shown above by making the gossip+S period longer T_period = 0.35.

	Gossip	Gossip+S
Bandwidth (bytes/s)	5840±674	6172±824
False positive rate	2.47±0.34%	2.07±0.49%



Discussion

In cases where we maintain a fixed detection time, the Gossip+S protocol exhibits a 40% increase in bandwidth usage compared to the Gossip protocol while achieving significantly lower false positive rates. Moreover, the detection time experiences a slight increase, particularly when multiple machines fail simultaneously. However, when we keep bandwidth fixed, the Gossip+S protocol continues to outperform the Gossip protocol in terms of false positive rates. Nevertheless, it results in considerably higher detection times. Interestingly, the detection time for the Gossip protocol remains relatively stable and shows minimal variation with the number of failing machines.