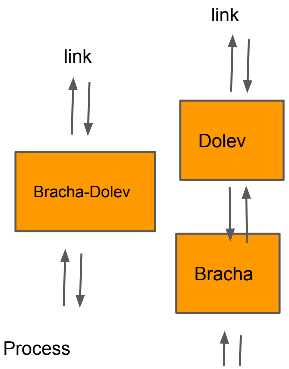
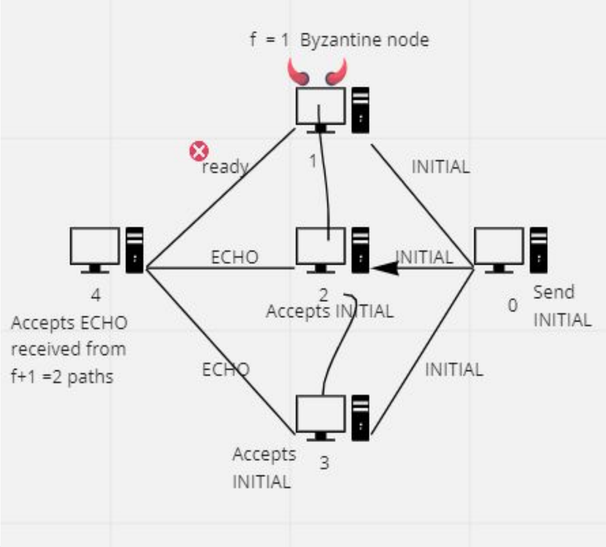
(1)What modifications from Brcha-Dolev broadcasting protocol can be applied to Brcha-CPA broadcasting protocol

Background(How Bracha-Dolev work)

-Bracha-Dolev is Byzantine reliable broadcast protocol used to reach consensus in a network
 -it results from combining Dolev's and Bracha's broadcast protocol



Bracha:
When receiving initial (from broadcaster)send echo
When receiving (n+f)/2 + 1 echo or f+1 ready send ready
after receiving 2f+1 ready accept the message
Dolev: Works on Bracha's messages
-Always relays messages other processes sent to neighbours not included in the path
-Forward to Bracha if a message is received from f+1 disjoint paths



Background(Replacing Dolev)

Disadvantages of Bracha-Dolev:
-Dolev checks the message is received from (f+1)-disjoint paths which is very computationally expensive
-Bracha-Dolev has a very high message complexity(number of messages sent)

Replace Dolve By CPA in Brach-Dolev
Advantages:
-Better message complexity
-Less computational expensive
Disadvantages:
-We can't apply CPA to all graphs

The main question

A lot of optimizations have been applied to Bracha-Dolev by Silvia Bonomi et al [1], Can we apply any of these optimizations to Brach-CPA? What is the decrease percentage of message complexity after applying these optimizations? On Which types of graphs does CPA have the highest probability of succeeding ?

Research Method

-Choosing optimizations to apply to Bracha-CPA from the paper[1].
-Compare Bracha-CPA with the optimizations with plain Bracha-CPA and Brach-Dolev using omnet++
-check on which type of graph we can apply CPA by implementing F, L, R partitioning algorithm[2] in Python

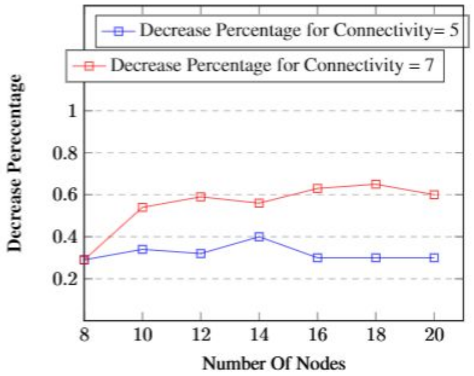
References

[1]Silvia Bonomi, Giovanni Farina, and Sebastien Tixeuil. Multi-hop byzantine reliable broadcast with honest dealer made practical.Journal of the Brazilian Computer Society, 25(1):1–23, 2019.
[2]Lewis Tseng, Nitin Vaidya, and Vartika Bhandari. Broadcast using certified propagation algorithm in presence of byzantine faults.Information Processing Letters, 115(4):512–514, 2015.

Results

We applied the following optimizations to Bracha-CPA:
-Single-hop Send message(processes don't relay send)
-Echo to Echo transitions(Dolev wants to relay an echo and Bracha want to send an echo, we send echo_echo message)
-Echo to ready transitions(Dolev wants to relay an echo and Bracha want to send a ready, we send echo_ready message)
-Ignore Echos if Dolev-deliver a ready from the same process
-Ignores all Echo if the process accepts the message

Comparison between Bracha-CPA and Brcha-Dolev



Conclusion

-Bracha-CPA with the optimizations has up to 60% less message complexity compared to Bracha-Dolev
-Bracha-CPA with the optimizations has up to 20% less message complexity compared to plain Bracha-CPA(didn't include graph due to lack of space)
-When we a maximum number of Byzantine nodes, here is a list of types of graphs sorted based on the success probability of using CPA(starting from the worst):
Generalized-wheel, k-regular, Multi-partite-wheel, k-diamond and k-pasted

Qusay Fantazia Q.Fantazia@student.tudelft.nl