



# **Byzantine Failures in Distributed Systems**

## **Trust and Resilience in Distributed Systems**

Burak Metehan Tunçel  
[tuncel.burak@metu.edu.tr](mailto:tuncel.burak@metu.edu.tr)

Wireless Systems, Networks and Cybersecurity Laboratory  
Department of Computer Engineering  
Middle East Technical University  
Ankara Turkey

March 28, 2024

# Outline of the Presentation

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works
  - Model, Definitions
  - Background, Previous Works
  - Background, Previous Works
- 5 Contribution
  - Main Point 1
- 6 Experimental results/Proofs
  - Main Result 1
- 7 Conclusions

# Agenda

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works
- 5 Contribution
- 6 Experimental results/Proofs
- 7 Conclusions

# Clock Synchronization

## Clock Synchronization in Distributed Systems

Ensuring accurate and consistent timekeeping across geographically distributed systems is fundamental for coordinated operation.

Traditional clock synchronization algorithms assume processes behave correctly and exchange reliable information; however, real-world systems are not immune to failures. In some scenarios, processes may exhibit malicious or arbitrary behavior.

Therefore, developing robust clock synchronization algorithms that can tolerate these failures is crucial for ensuring the reliability and integrity of distributed systems.

# Agenda

- 1 The Problem
- 2 The Contribution**
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works
- 5 Contribution
- 6 Experimental results/Proofs
- 7 Conclusions

# What is the solution/contribution

- Robust clock synchronization algorithms tolerating Byzantine failures, which exhibit malicious or arbitrary behavior, is crucial.
- **The Mahaney-Schneider synchronizer** stands out as a practical and efficient solution for achieving this goal.
- Leveraging message exchange and statistical analysis to filter out misleading information.
- Converging towards a reliable estimate of the system time, even in the presence of Byzantine processes.

# Agenda

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance**
- 4 Background/Model/Definitions/Previous Works
- 5 Contribution
- 6 Experimental results/Proofs
- 7 Conclusions

# Motivation/Importance

- **Trust in Decentralized Systems:** BFT is essential for establishing trust in decentralized systems like blockchain, ensuring operations continue even when some participants are unreliable.
- **Real-World Applications:** It's crucial for safety in critical applications such as aviation and autonomous vehicles, where failure can have dire consequences.
- **Philosophical Reflection:** BFT prompts reflection on trust and consensus in systems where misinformation or malice is possible, highlighting its broader implications.
- **Evolving Distributed Systems:** As these systems grow in scale and importance, the role of BFT in ensuring resilience against problematic conditions becomes increasingly vital.



# Agenda

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works**
- 5 Contribution
- 6 Experimental results/Proofs
- 7 Conclusions

# Definitions

- Byzantine failures are the scenarios where processes exhibit arbitrary behavior including sending incorrect information, withholding data, or even actively trying to disrupt the synchronization process.
- Byzantine Fault Tolerance (BFT) is a property of distributed systems that allows them to continue operating correctly even when some of the nodes fail in arbitrary or malicious ways.

# Background

- Clock synchronization in distributed systems ensures consistent timekeeping across geographically.
- Traditional algorithms assume well-behaved processes such as Lamport's logical clocks and Cristian's algorithm.
- Traditional algorithms fail with Byzantine failures.

# Background

- BFT protocols can tolerate Byzantine failures but are computationally complex and expensive.
- The Mahaney-Schneider synchronizer is a BFT clock synchronization algorithm that is efficient and robust.
- Work even in the presence of Byzantine failures.

# Agenda

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works
- 5 Contribution**
- 6 Experimental results/Proofs
- 7 Conclusions

# Main Point 1

TODO: Complete after code implementation and experiments.

# Agenda

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works
- 5 Contribution
- 6 Experimental results/Proofs**
- 7 Conclusions

# Main Result 1

TODO: Complete after code implementation and experiments.



# Agenda

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works
- 5 Contribution
- 6 Experimental results/Proofs
- 7 Conclusions**

# Conclusions

TODO: Complete after code implementation and experiments.

# References

# Questions

THANK YOU

Byzantine Failures in Distributed Systems  
Trust and Resilience in Distributed Systems

presented by Burak Metehan Tuncel  
[tuncel.burak@metu.edu.tr](mailto:tuncel.burak@metu.edu.tr)



March 28, 2024

