CHALMERS
UNIVERSITY OF TECHNOLOGY

C A U
Kiel University
Christian-Albrechts-Universität zu Kiel

Paxos Made Wireless: Consensus in the Air

Valentin Poirot[†], Beshr Al Nahas[†], Olaf Landsiedel[‡†]

[†]*Chalmers University of Technology, Sweden*

[‡]*Kiel University, Germany*

International Conference on Embedded Wireless Systems and Networks 2019, Beijing, China

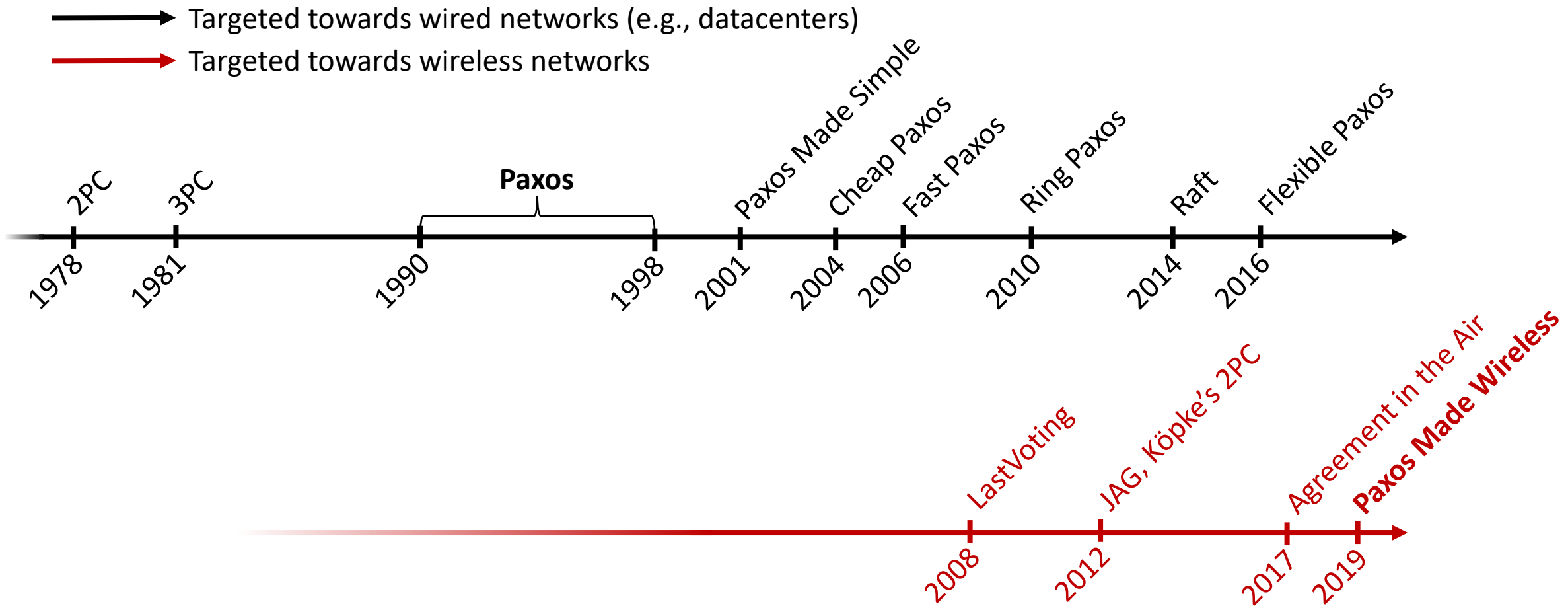
Consensus: a primordial primitive

- The ability for a group to agree on something

- In typical (wired) networks (e.g., datacenters):
  - Distributed databases
  - Distributed locks
  - Common configuration (state machine replication)

"It looks like we have a consensus."

A brief history on agreement and consensus



Consensus: a primordial primitive

- The ability for a group to agree on something

"It looks like we have a consensus."

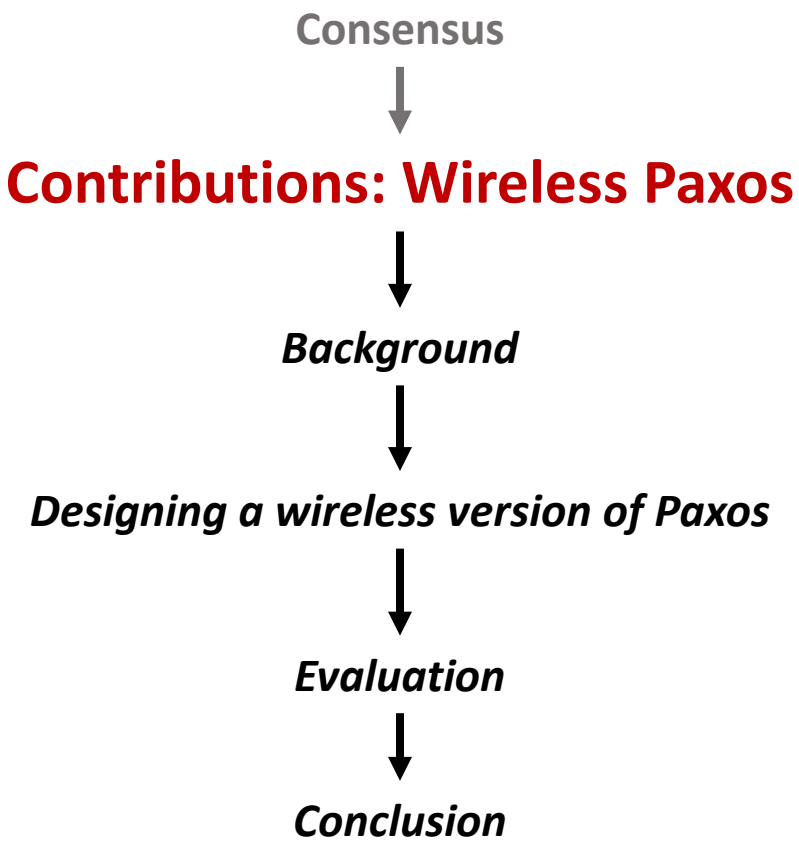
- In typical (wired) networks (e.g., datacenters):
  - Distributed databases
  - Distributed locks
  - Common configuration (state machine replication)

- In **wireless sensor networks**?
  - Network configuration: global (schedule), local (e.g., 6TiSCH)
  - Common destination point in Unmanned Aerial Vehicles (UAVs)
  - Unique decision in wireless closed-loop control (with distributed controllers)

Fault-tolerant Consensus

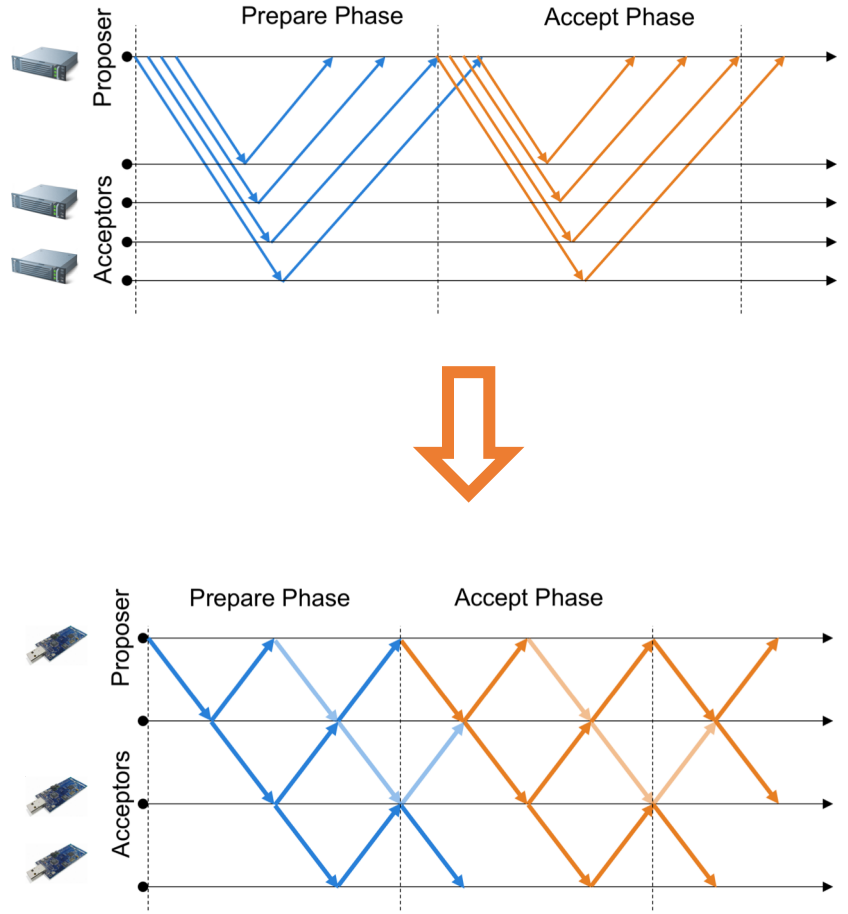
- Many possible failures
  - **Highly dynamic links** → message losses, network segmentation
  - **Nodes running on batteries, intermittent power** → node crashes, recovery
- 2PC, 3PC, and dissemination ≠ fault-tolerant consensus
  - **2PC blocks** upon node failures
  - **3PC is not blocking**, but can **lead to inconsistencies** in corner cases
  - Glossy provides dissemination with **high reliability**
    - But **no feedback**
- Paxos provides fault-tolerant consensus
  - As long as a **majority is up and running**
  - **Proven** to be correct

Overview


Consensus
Contributions: Wireless Paxos
Background
Designing a wireless version of Paxos
Evaluation
Conclusion

Contributions: Paxos Made Wireless

- We bring **fault-tolerant consensus** to **low-power wireless networks**
- Paxos as a **many-to-many scheme**
  → By **distributing** part of the proposer's logic to acceptors
  → Building on top of **concurrent transmissions** and **in-network processing**
- Available as an **open source library**[1]
  → No need to lose time to understand and implement Paxos again



[1] github.com/iot-chalmers/wireless-paxos

Overview

Consensus

↓

Contributions: Paxos Made Wireless

↓

Background

↓

Designing a wireless version of Paxos

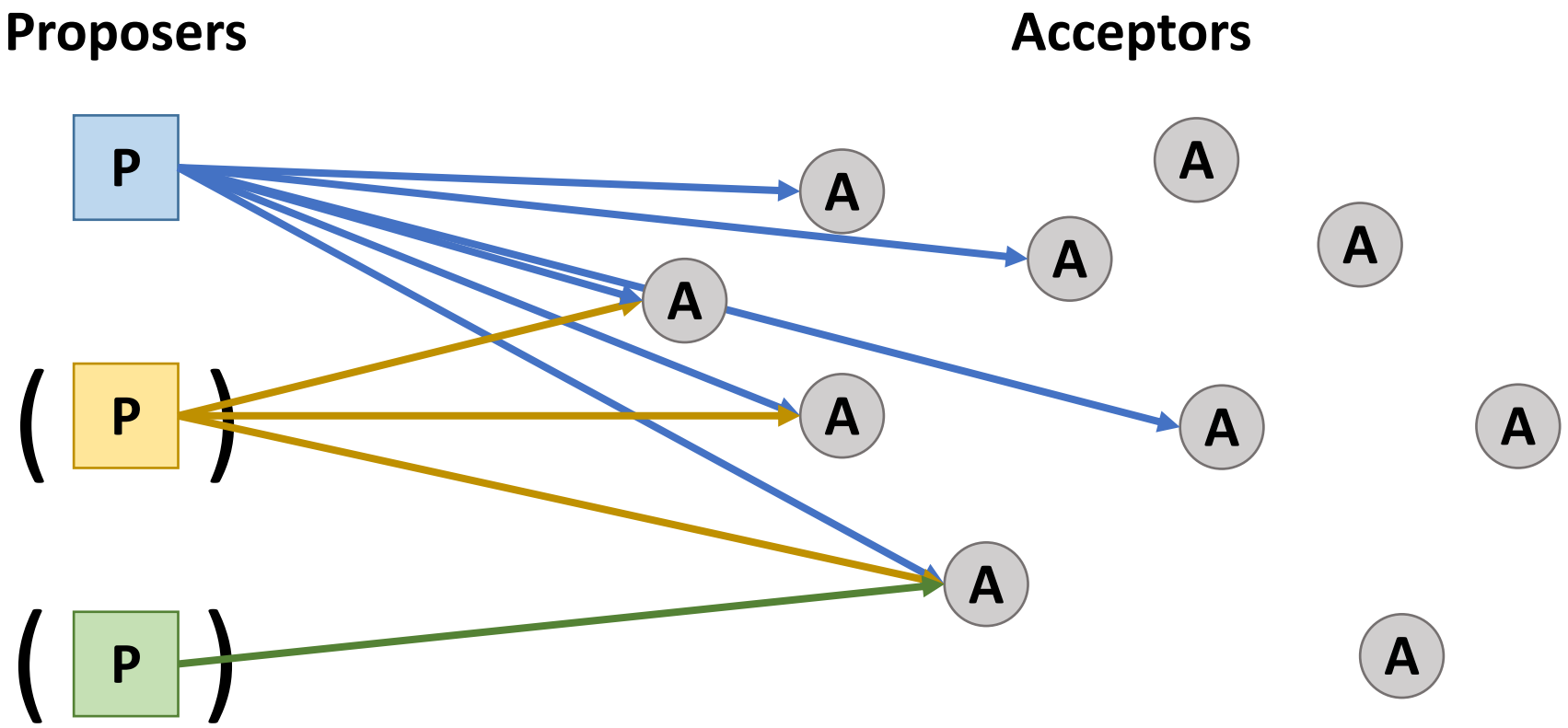
↓

Evaluation

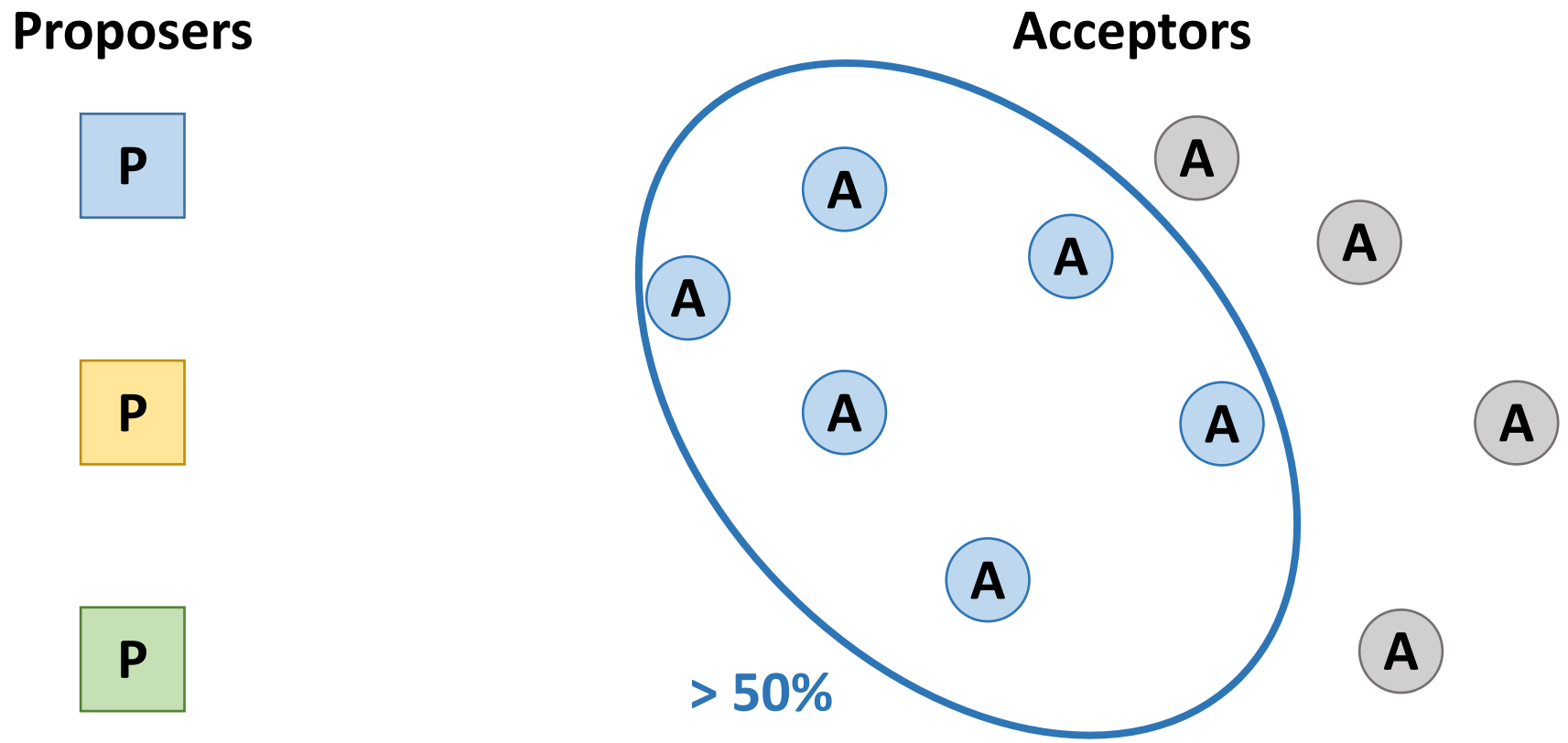
↓

Conclusion

Paxos – the basics

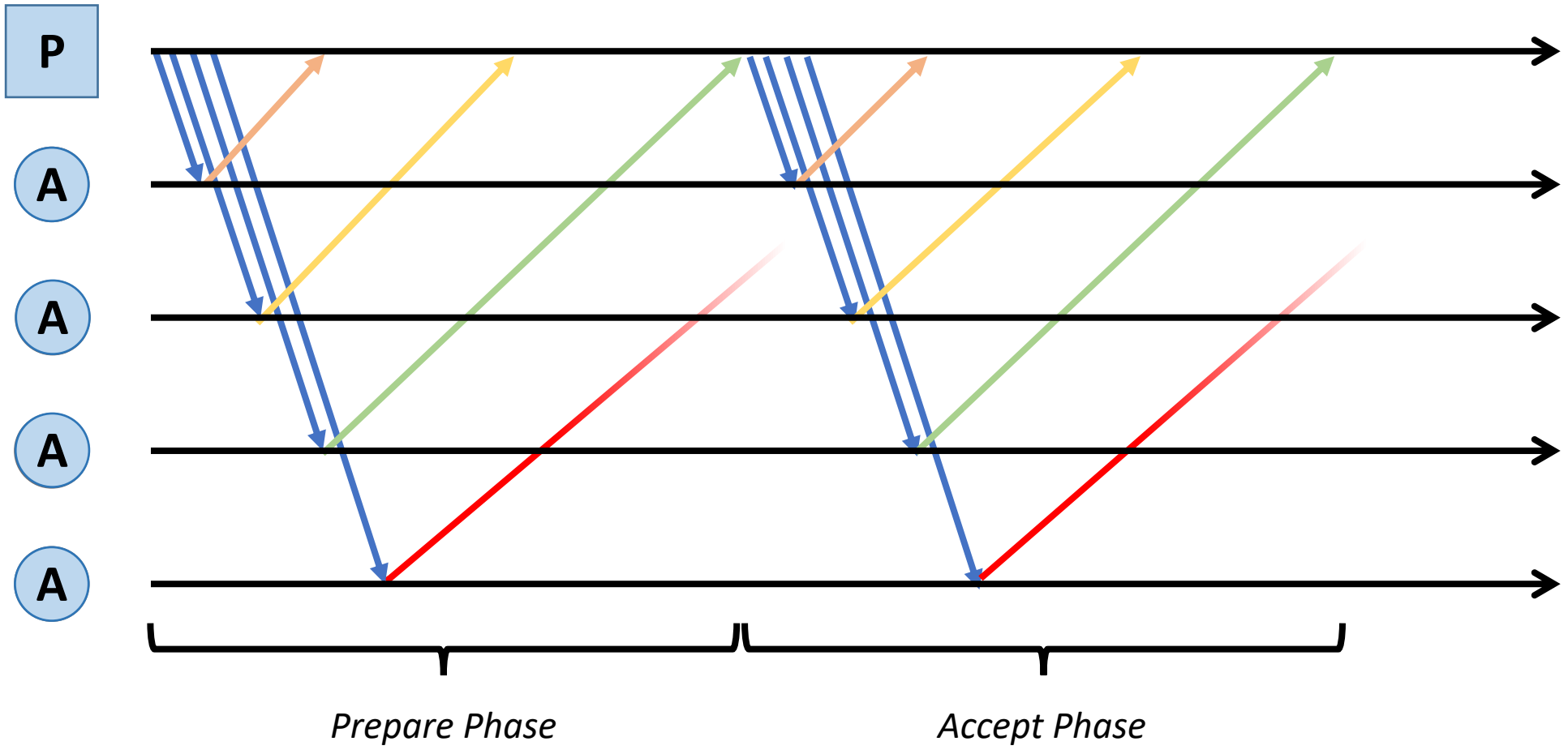


Paxos – the basics

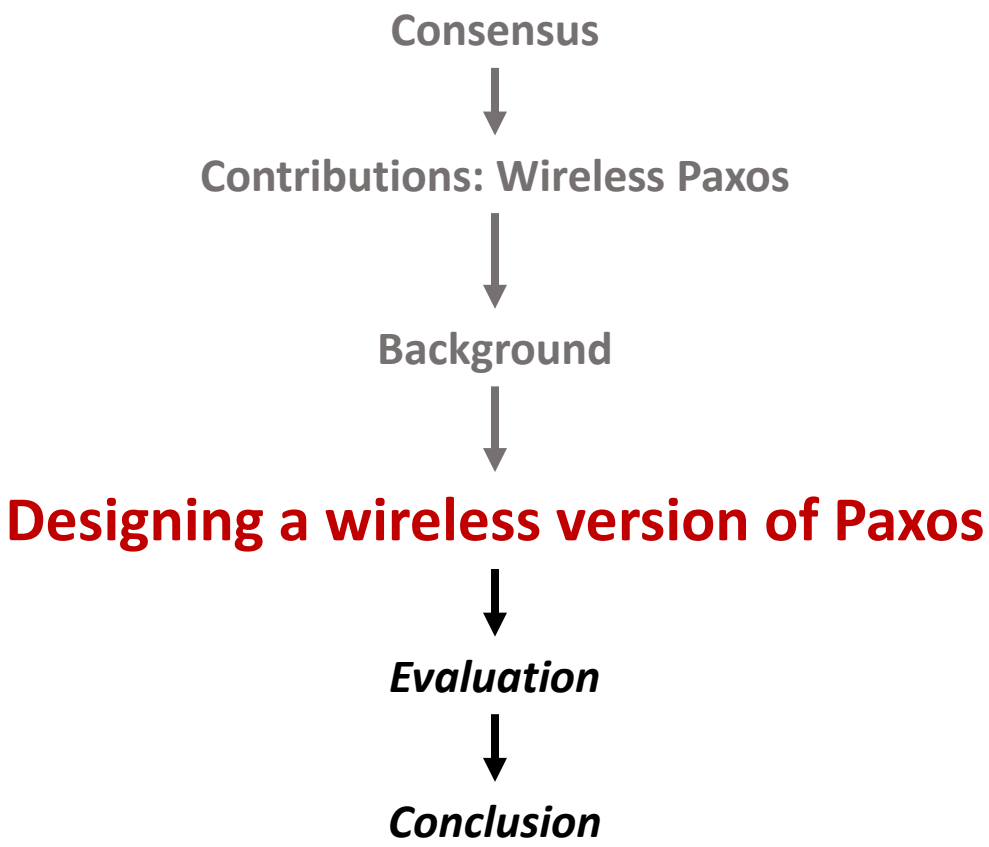


Paxos – the basics

> 50% responses received by Proposer, value is *chosen*

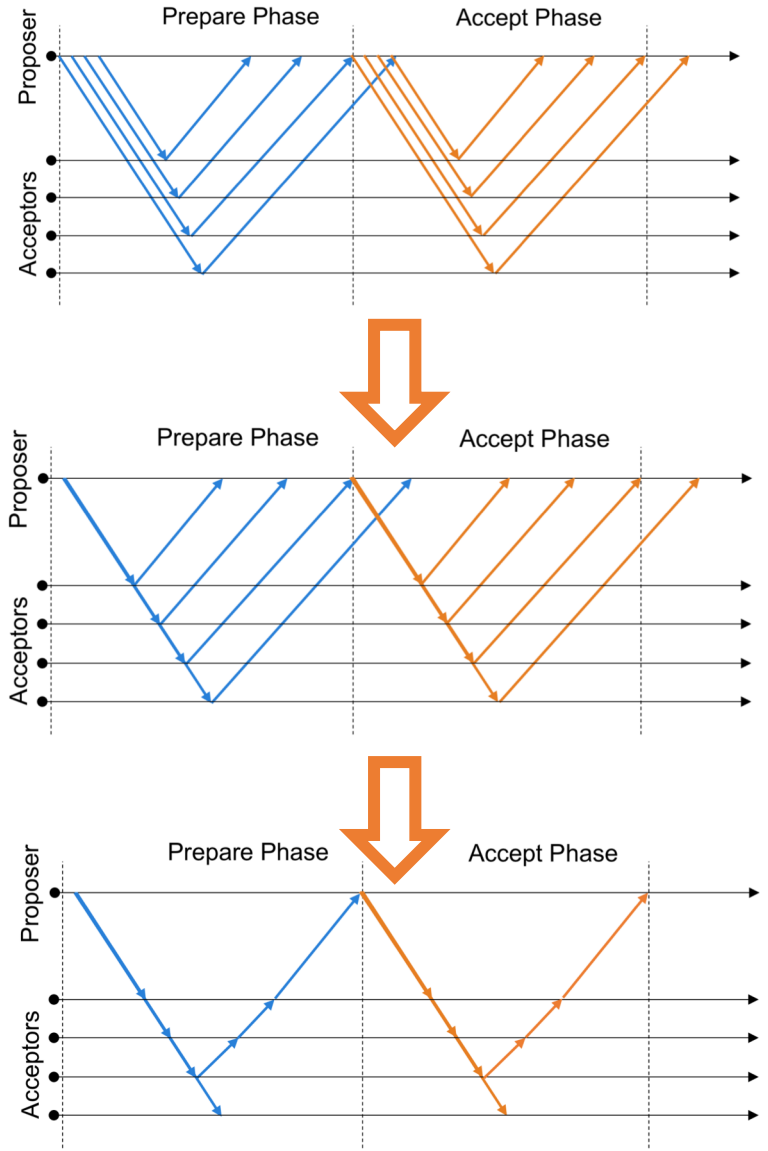


Overview


Consensus
Contributions: Wireless Paxos
Background
Designing a wireless version of Paxos
Evaluation
Conclusion

Rationale – Paxos is expensive

- Required: **2N+2 end-to-end messages**
  - Example: Euratech has **188** nodes → **378 end-to-end messages**
- Using **multicast** for requests?
  → **2 broadcasts and 189 end-to-end messages** in Euratech
- Using **special topology** for responses (e.g., Ring Paxos)
  → **2 broadcasts, 189 unicast messages** (+ cost of routing) in Euratech



Rationale – Reducing costs?

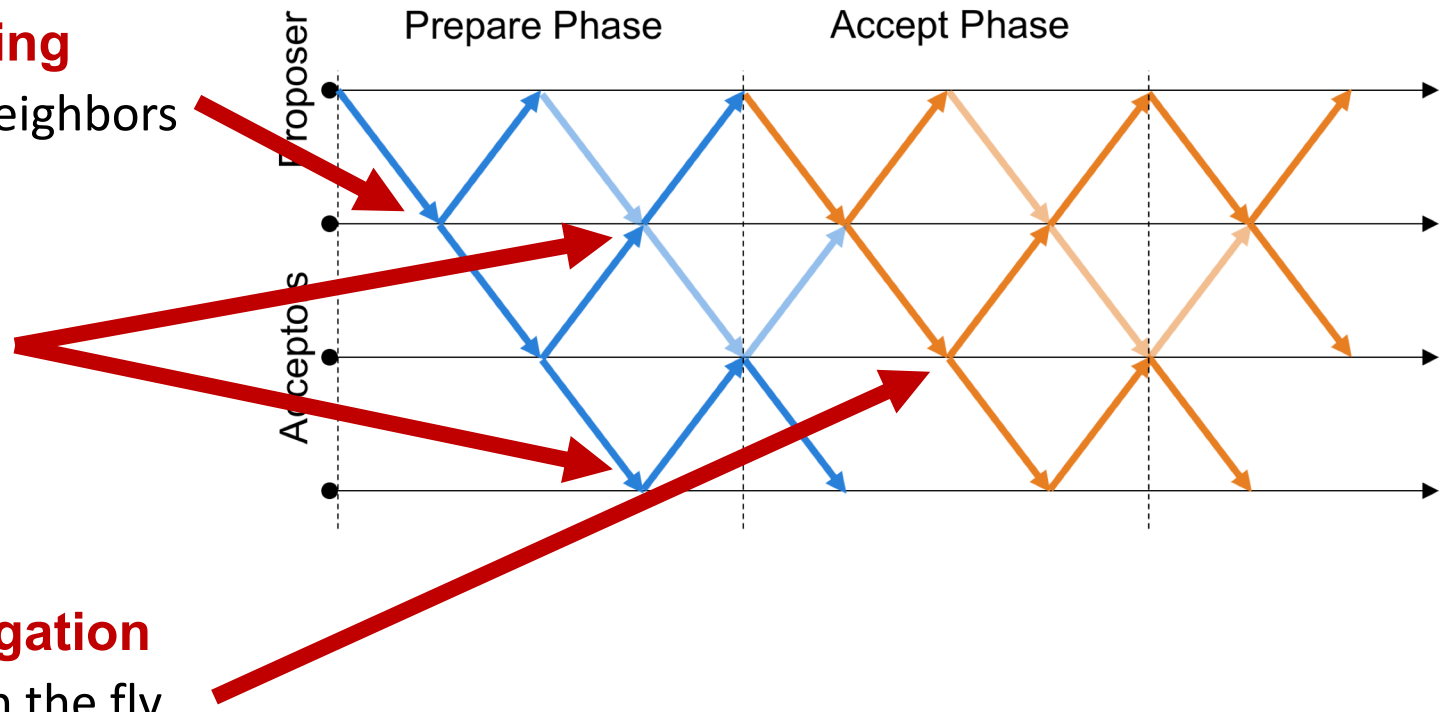
- Paxos has two phases
  - Each phase is a **request (dissemination)** and **responses (collection)**

- All responses are not required by the proposer!
  - The proposer must **detect a majority**
  - The proposer selects the response with the **highest proposal number**

→ **Distribute part of the proposer logic to all acceptors**
  - Acceptors **aggregate** the maximum proposal number and participation **along the way**
  - **No need** for unicast messages, routing, or special topology anymore
  - Fits to the approach used by **Agreement in the Air** (and **Chaos**)

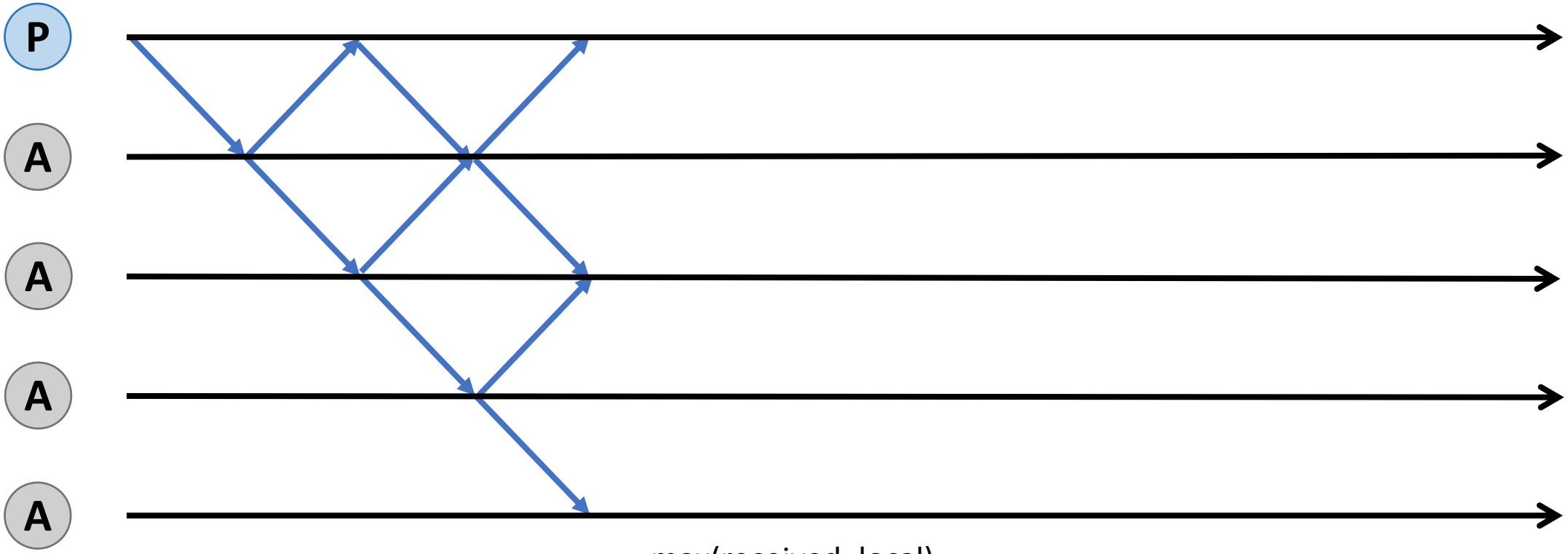
Wireless Paxos

- **Broadcast-oriented and flooding**
  - Data is broadcasted to local neighbors
- **Concurrent transmissions**
  - Nodes transmit concurrently
- **Local computation and aggregation**
  - The response is aggregated on the fly



Wireless Paxos

Proposer sees majority.

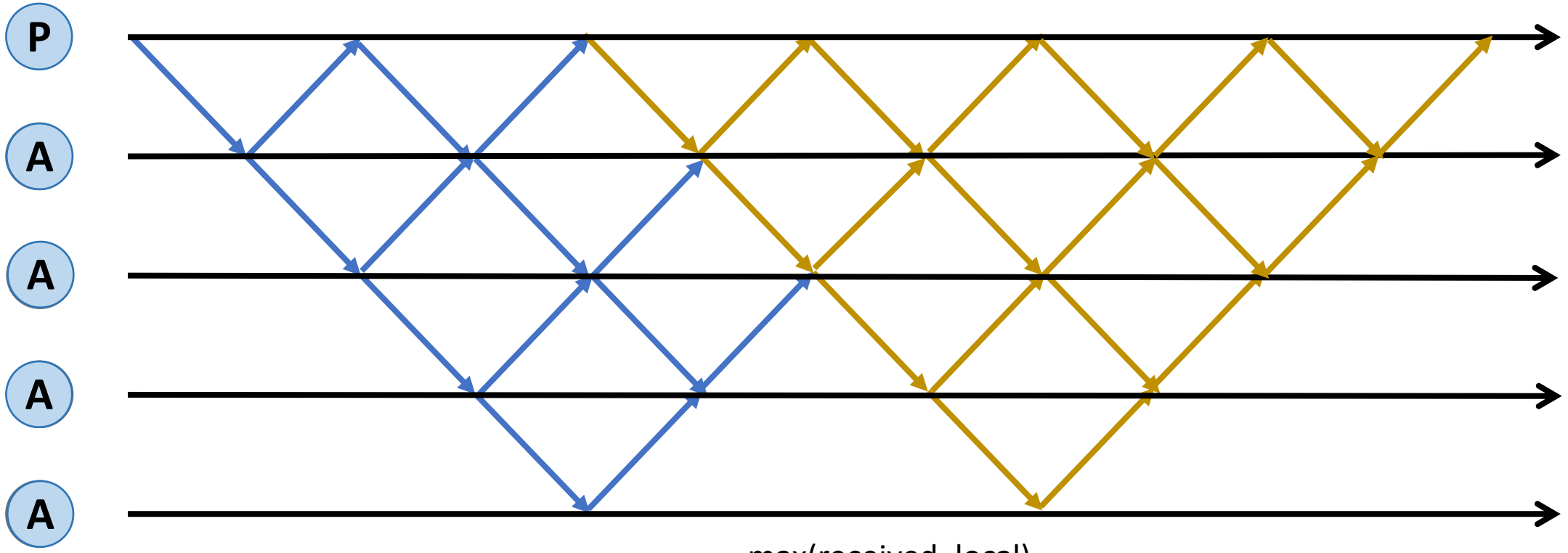


max(received, local)

Prep.	Proposal:*n*	…	…	X	X	X		

Wireless Paxos

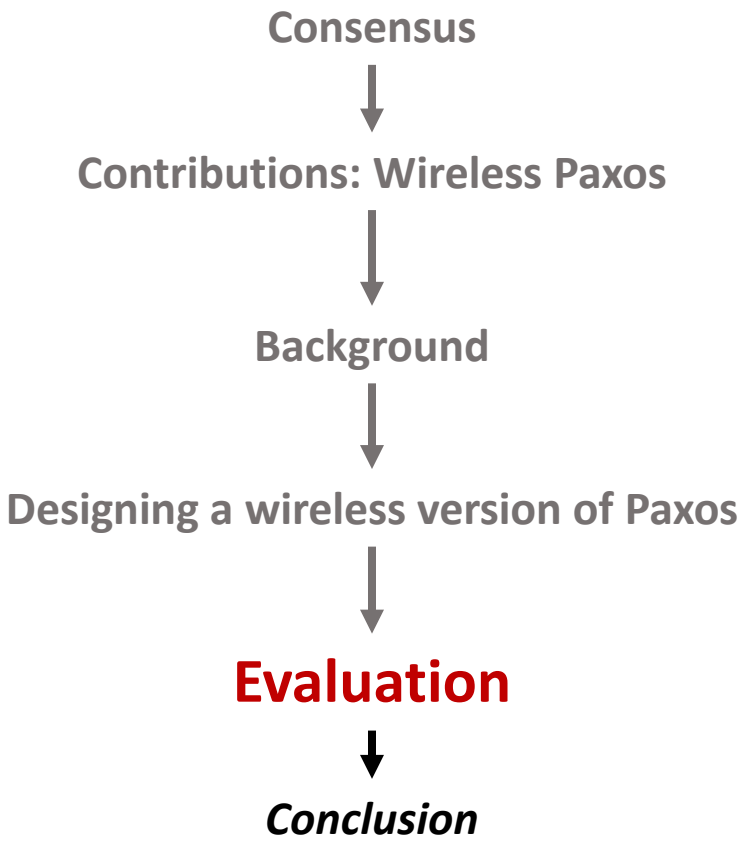
Result has converged, all nodes are aware that all other nodes agreed.



max(received, local)

Acc.	Proposal:*n*	value	…	X	X	X	X	X

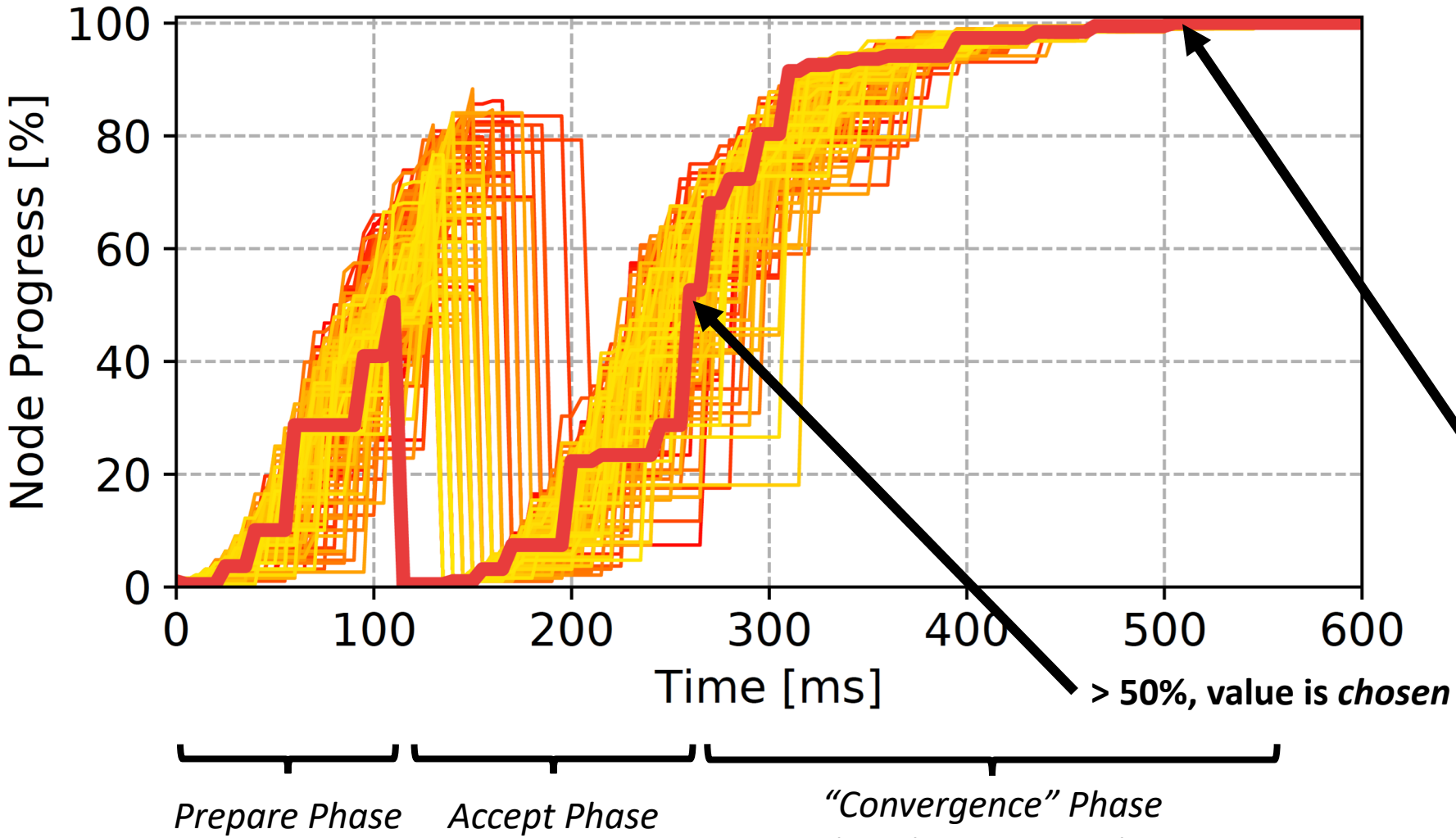
Overview


Consensus
Contributions: Wireless Paxos
Background
Designing a wireless version of Paxos
Evaluation
Conclusion

What has been evaluated?

- **Wireless Paxos** *and Wireless Multi-Paxos*
- **Very dense (Euratech) and low density (Flocklab) deployments**
- *Effect of multiple proposers*
- **Comparison of primitive costs (Dissemination, Agreement, Consensus)**
- *Consistency under injected failures*

Wireless Paxos in action – a typical round in Euratech (188 nodes)



Progress: number of participating nodes

Proposer

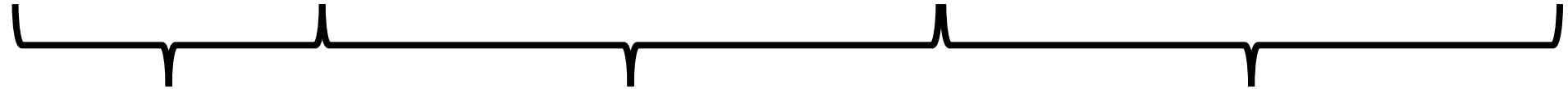
Acceptors

> 50%, value is *chosen*

All nodes are aware that the entire network chose the value

Comparing the cost of primitives

Protocol	Glossy	2PC	3PC	WPaxos	WMulti-Paxos
Slot length [ms]	4	4	4	5	6



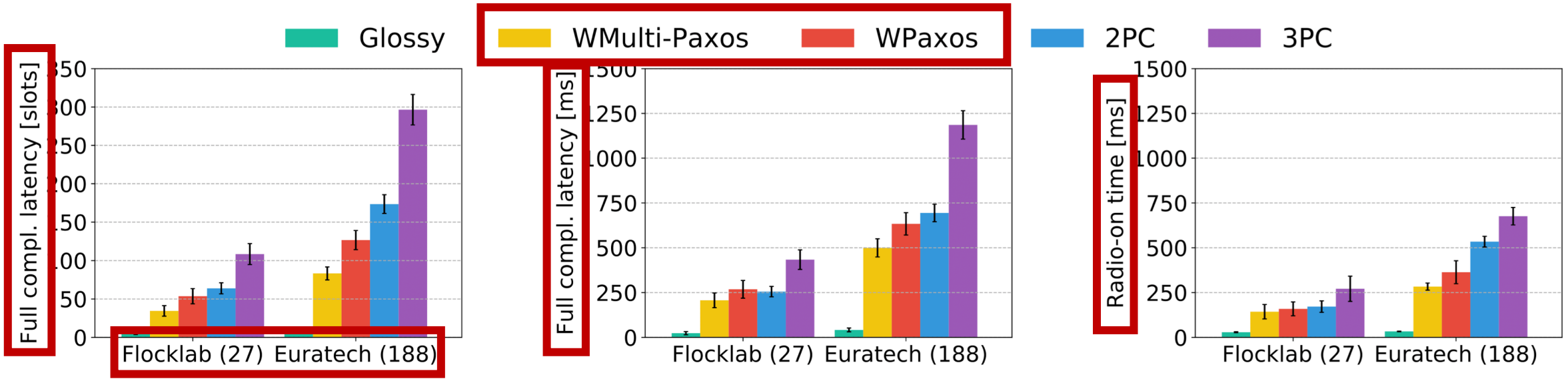
Dissemination

Agreement

Consensus

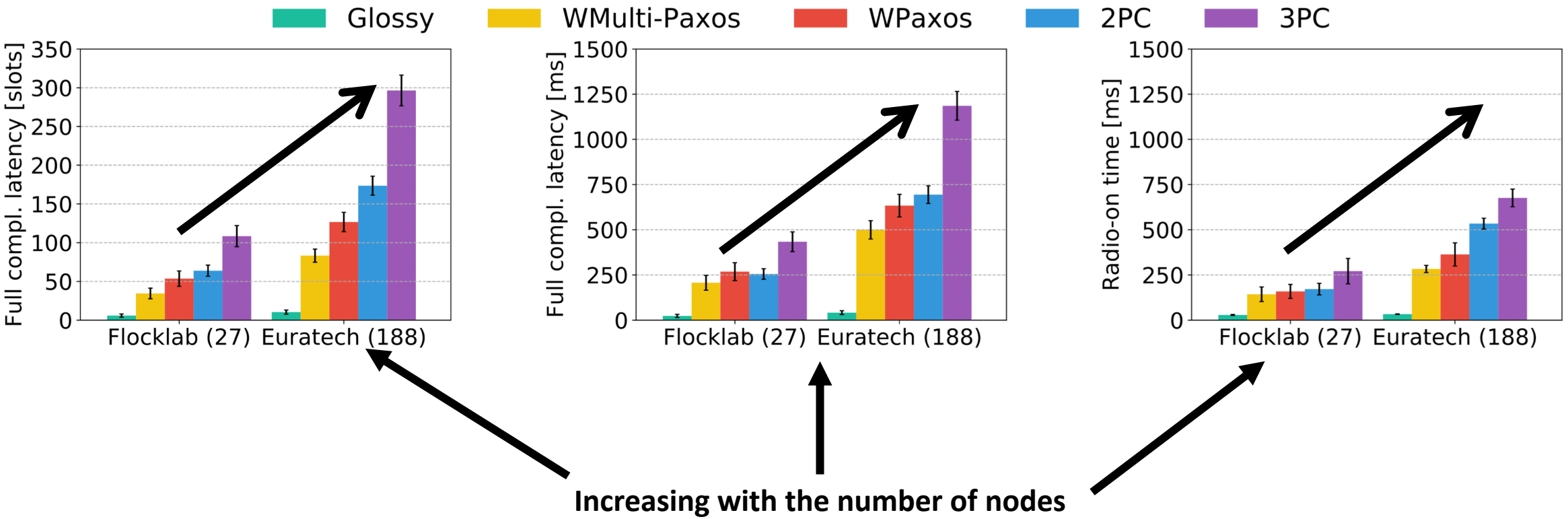
Comparing the cost of primitives

Protocol	Glossy	2PC	3PC	WPaxos	WMulti-Paxos
Slot length [ms]	4	4	4	5	6



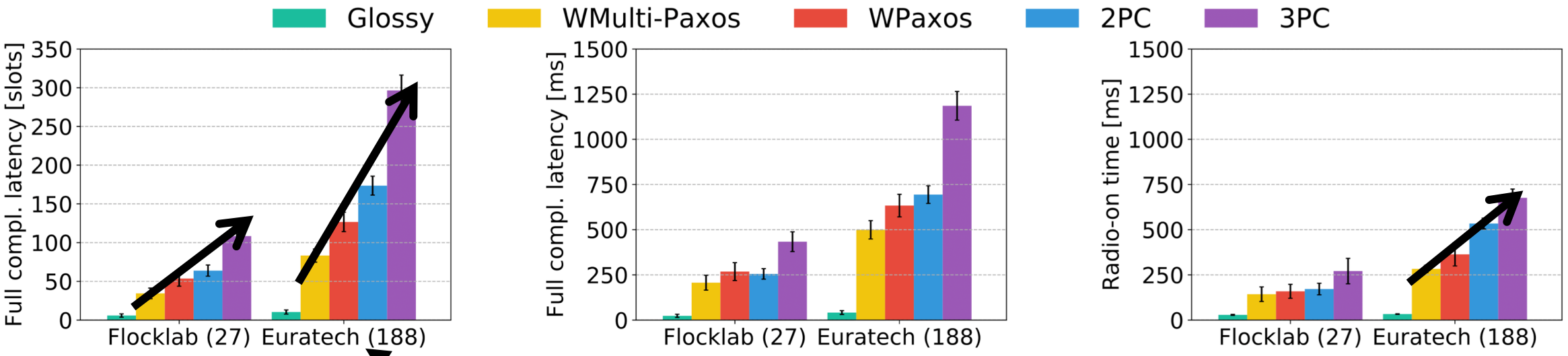
Comparing the cost of primitives

Protocol	Glossy	2PC	3PC	WPaxos	WMulti-Paxos
Slot length [ms]	4	4	4	5	6



Comparing the cost of primitives

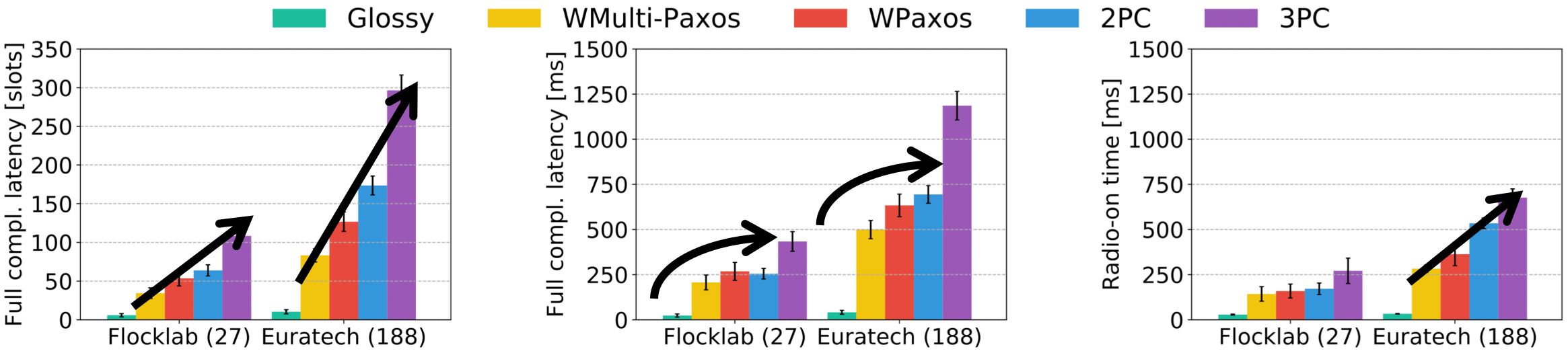
Protocol	Glossy	2PC	3PC	WPaxos	WMulti-Paxos
Slot length [ms]	4	4	4	5	6



Proportional to the number of phases

Comparing the cost of primitives

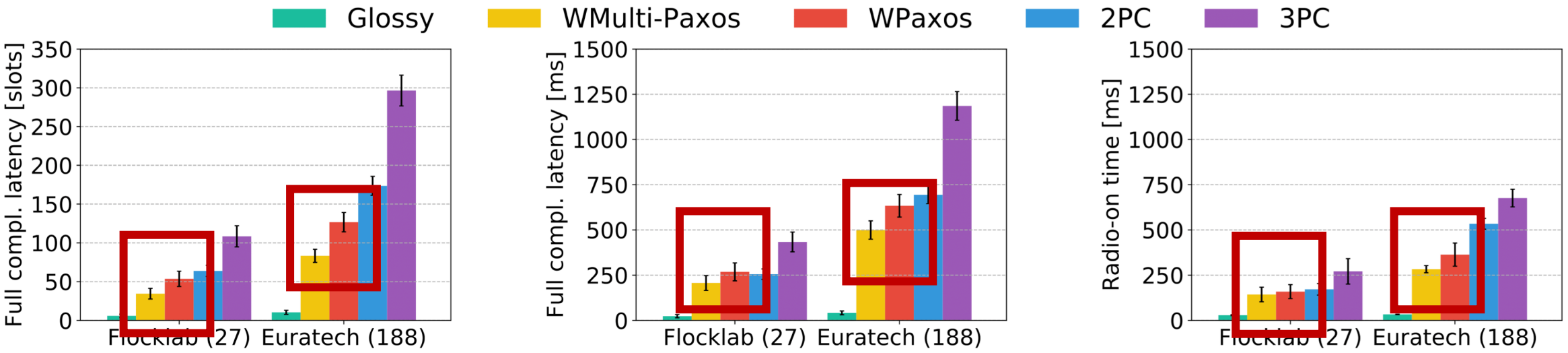
Protocol	Glossy	2PC	3PC	WPaxos	WMulti-Paxos
Slot length [ms]	4	4	4	5	6



Algorithm's complexity impacts the latency

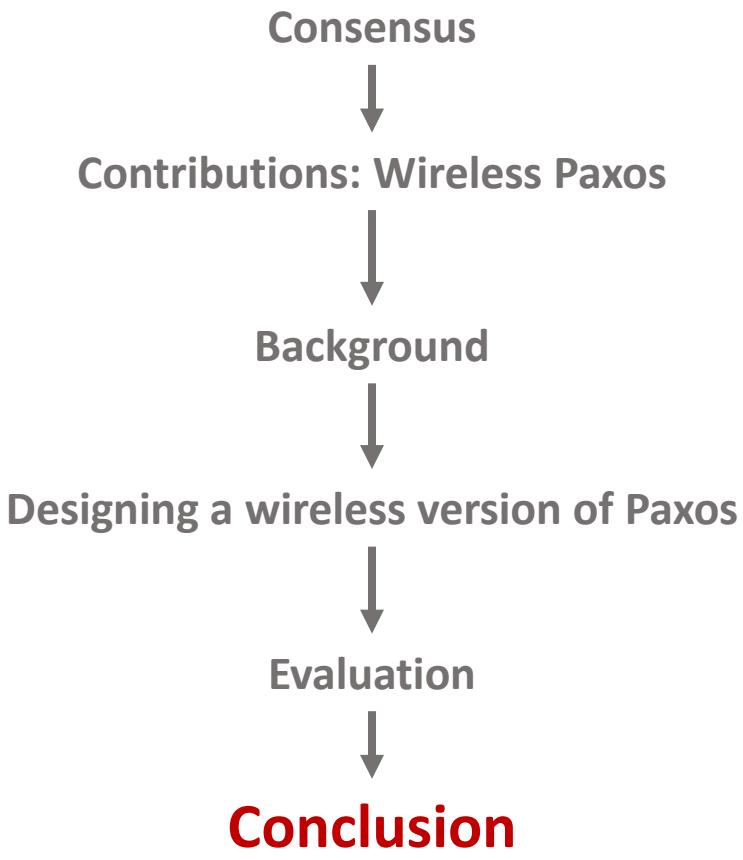
Comparing the cost of primitives

Protocol	Glossy	2PC	3PC	WPaxos	WMulti-Paxos
Slot length [ms]	4	4	4	5	6



Wireless Paxos and Wireless Multi-Paxos remain low-latency

Overview


Consensus
Contributions: Wireless Paxos
Background
Designing a wireless version of Paxos
Evaluation
Conclusion

Conclusion

Wireless Paxos

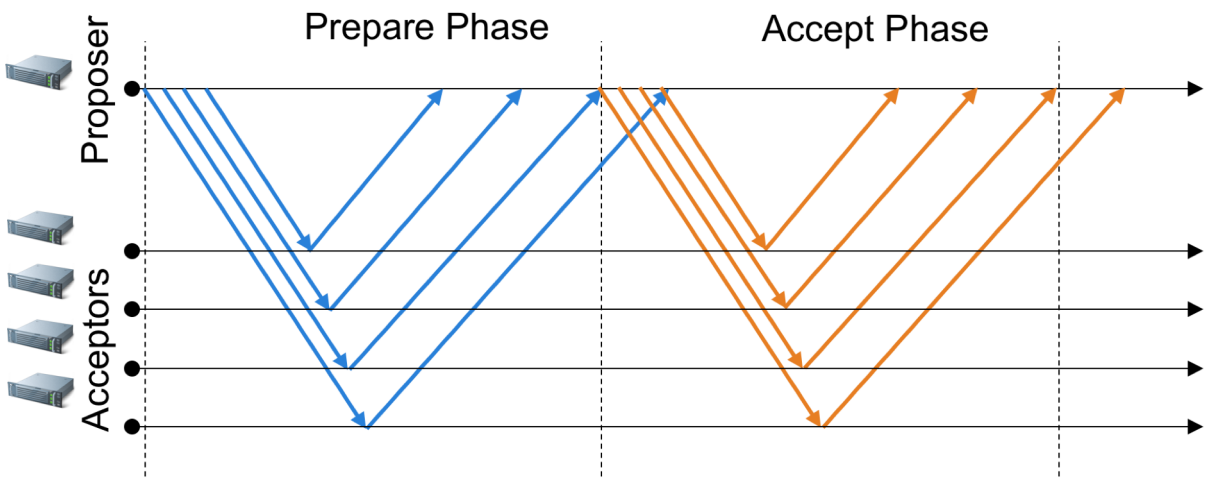
- Fault-tolerant consensus for low-power wireless networks
- Paxos as a many-to-many scheme
- Based on flooding, concurrent transmissions and in-network processing

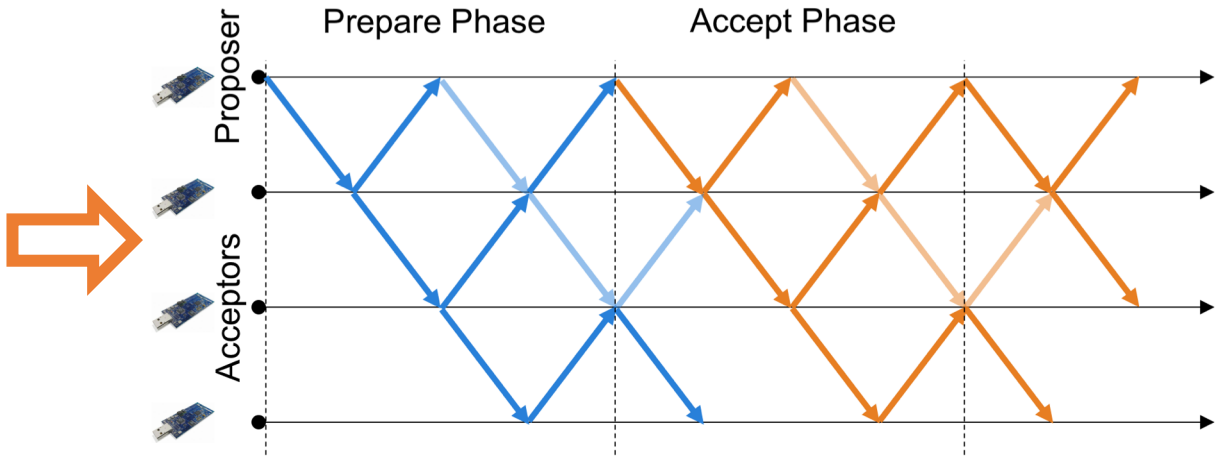
Results

- Consensus in **289 ms for 188 nodes** in Euratech
- **Consensus stays consistent** under injected failures

Paxos Made Wireless: Consensus in the Air

THANK YOU!





CHALMERS
UNIVERSITY OF TECHNOLOGY

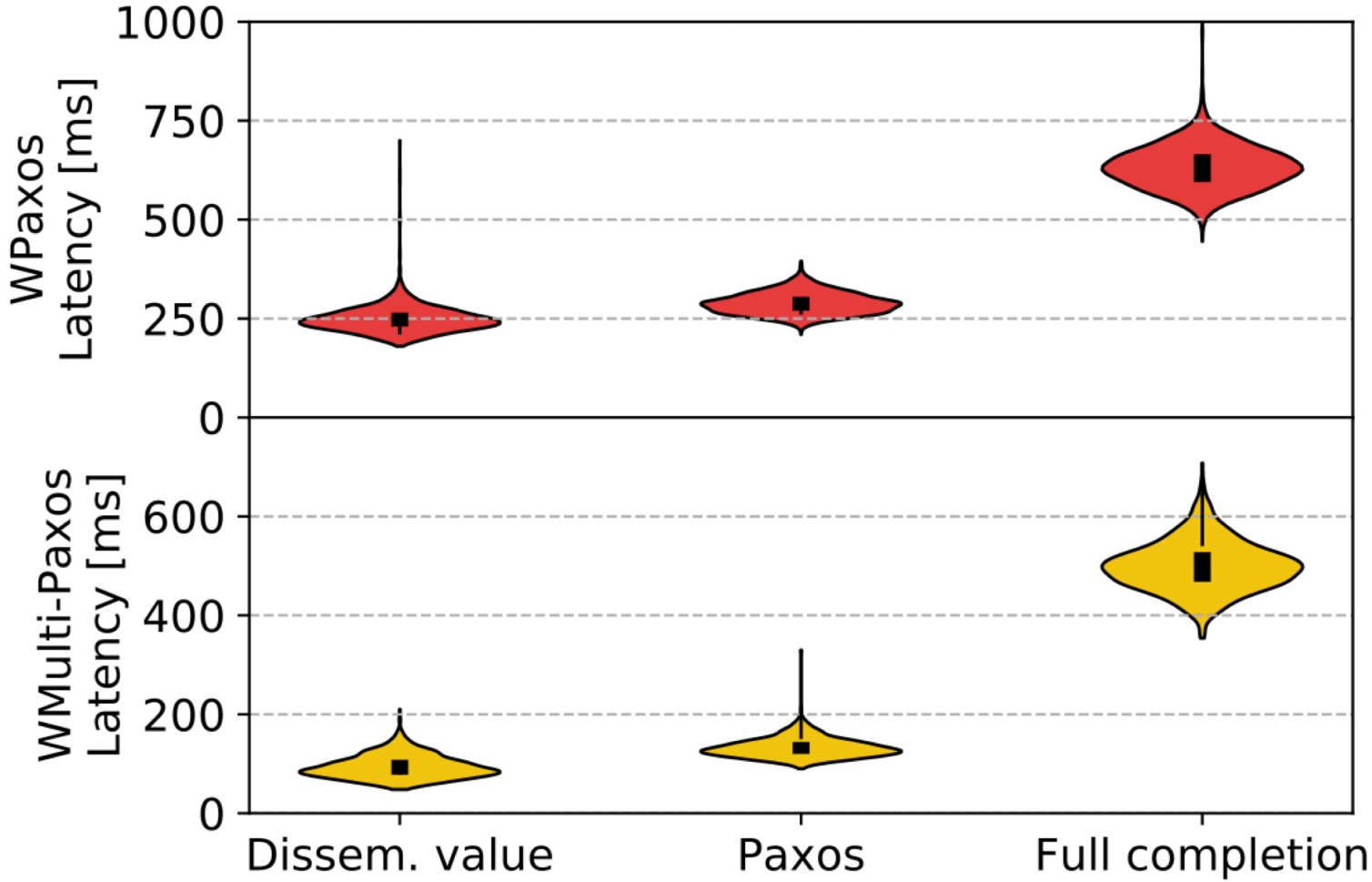
C A U
Kiel University
Christian-Albrechts-Universität zu Kiel

Paxos Made Wireless: Consensus in the Air

Valentin Poirot[†], Beshr Al Nahas[†], Olaf Landsiedel[‡†]
[†]*Chalmers University of Technology*
[‡]*Kiel University*

International Conference on Embedded Wireless Systems and Networks 2019, Beijing, China

Latency of consensus



- **Dissem. value:**

All nodes received decision, no convergence yet

- **Paxos:**

Proposer received a maj. of replies

- **Full Completion:**

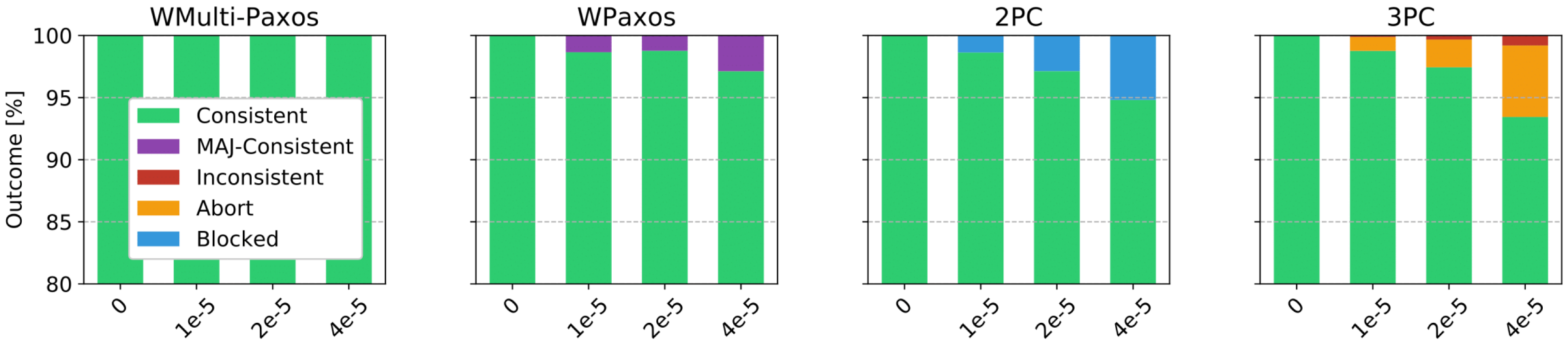
All nodes heard of all other nodes

Agreement over Glossy vs Wireless Paxos?

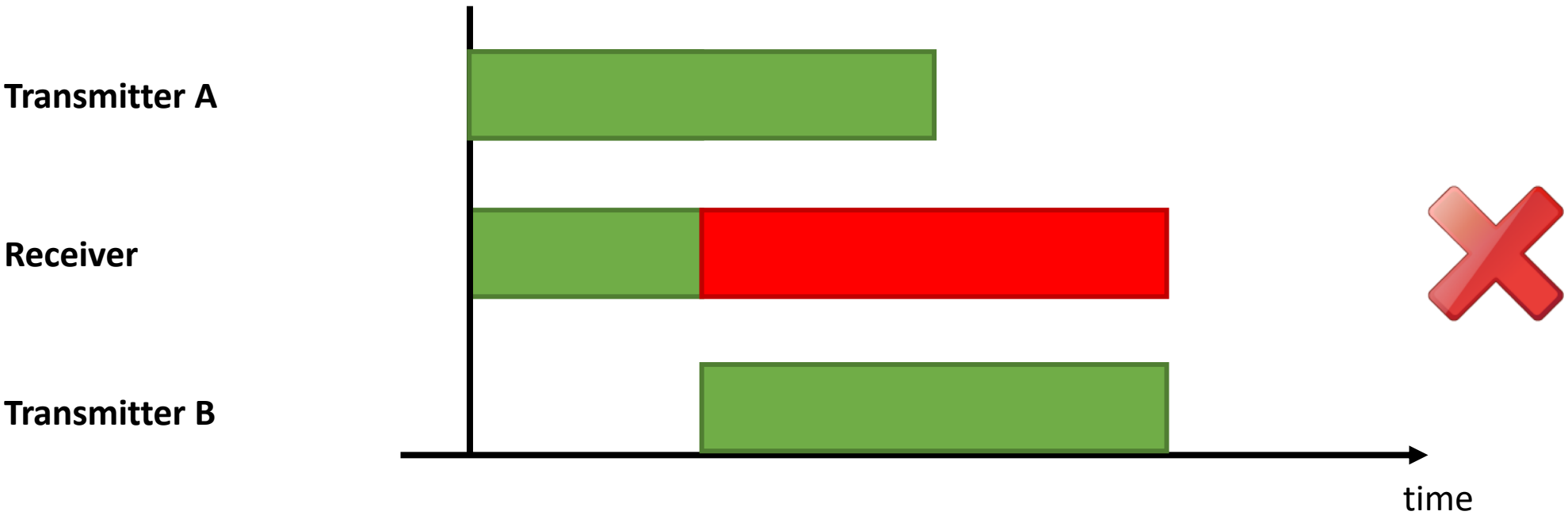
Table 1. Estimating the Cost of Feedback in Euratech with 188 nodes. *By repeating the flood multiple times, reliability is improved but no feedback is available (Repeated Glossy). Each node can report its status with a new flood (Glossy with Feedback) at a very high cost. Wireless Multi-Paxos provides consensus at a lower cost.*

Protocol	Repeated Glossy	Glossy with Feedback	Wireless Multi-Paxos
Latency [ms]	100	3760	500

Consensus consistency



Concurrent transmissions

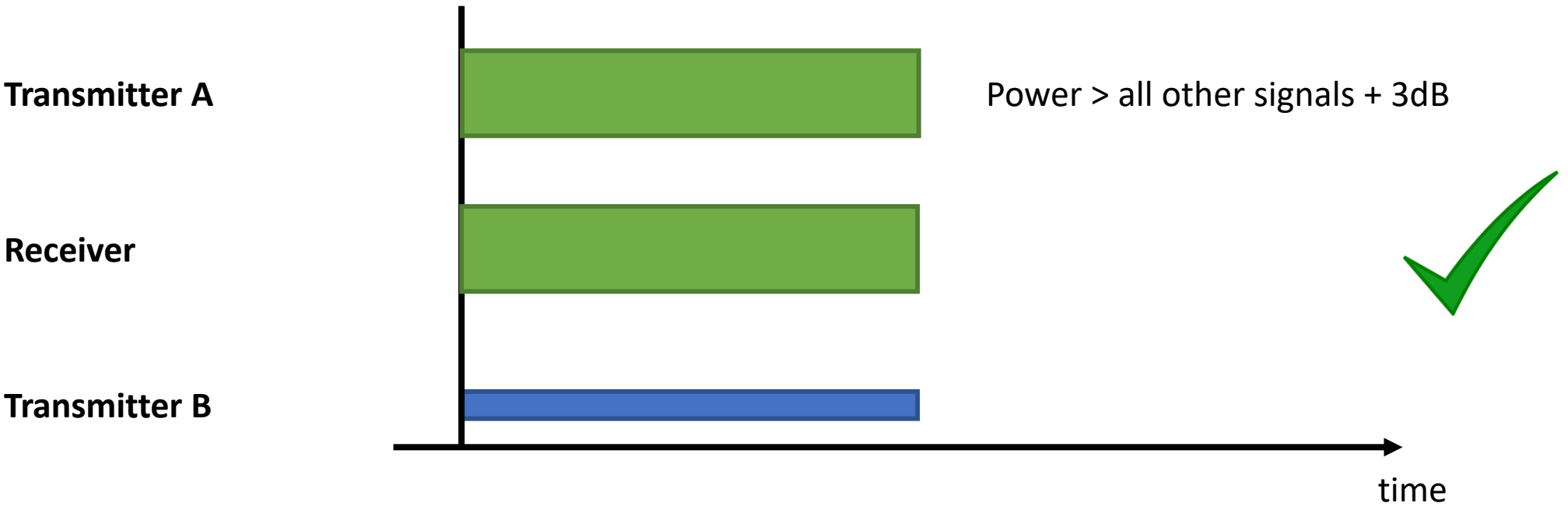


Conditions: *different* starting time

Concurrent transmissions

Conditions: *same* starting time, *same* data

Concurrent transmissions



Conditions: *same* starting time, *different* data, *higher* received power