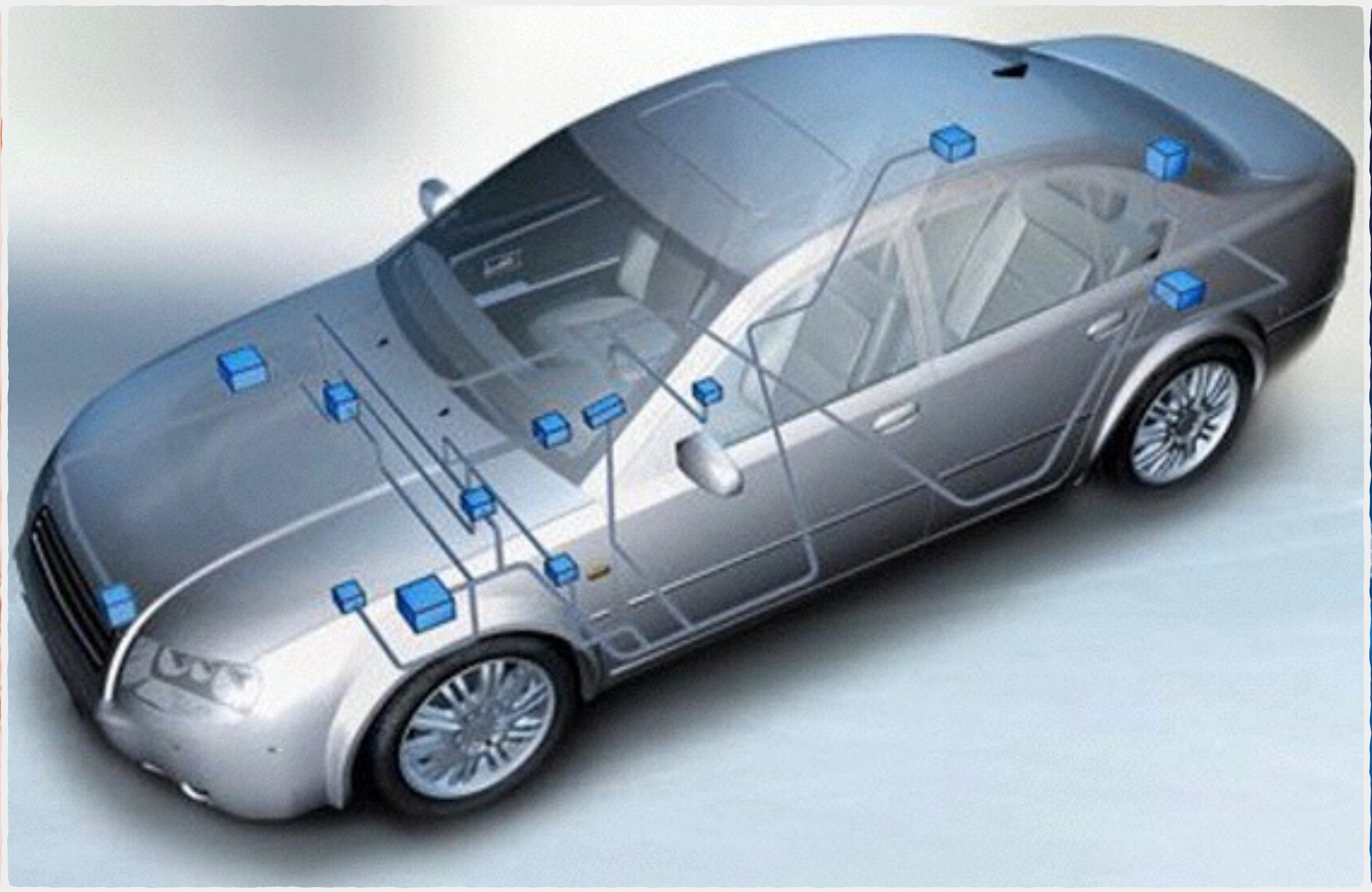
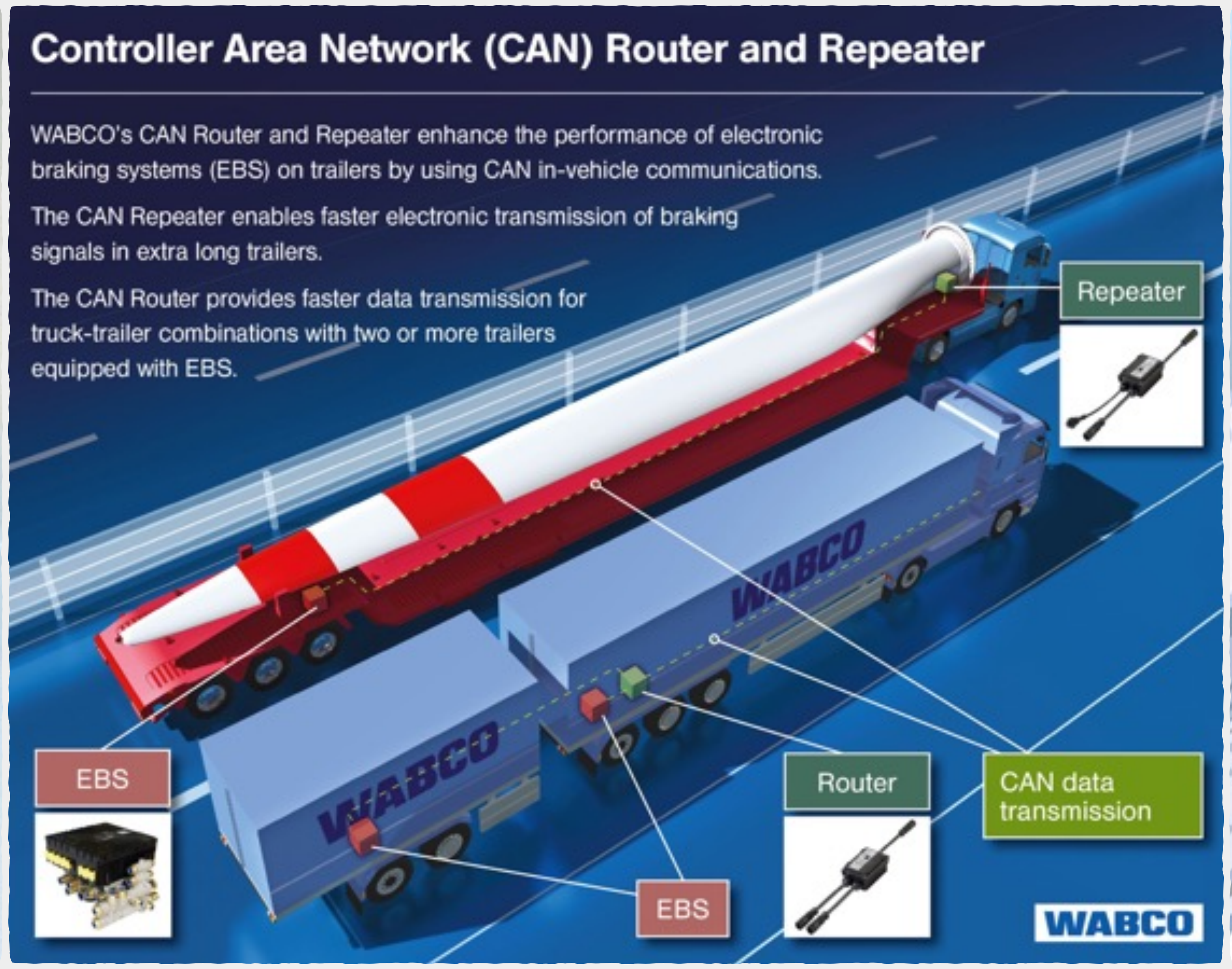




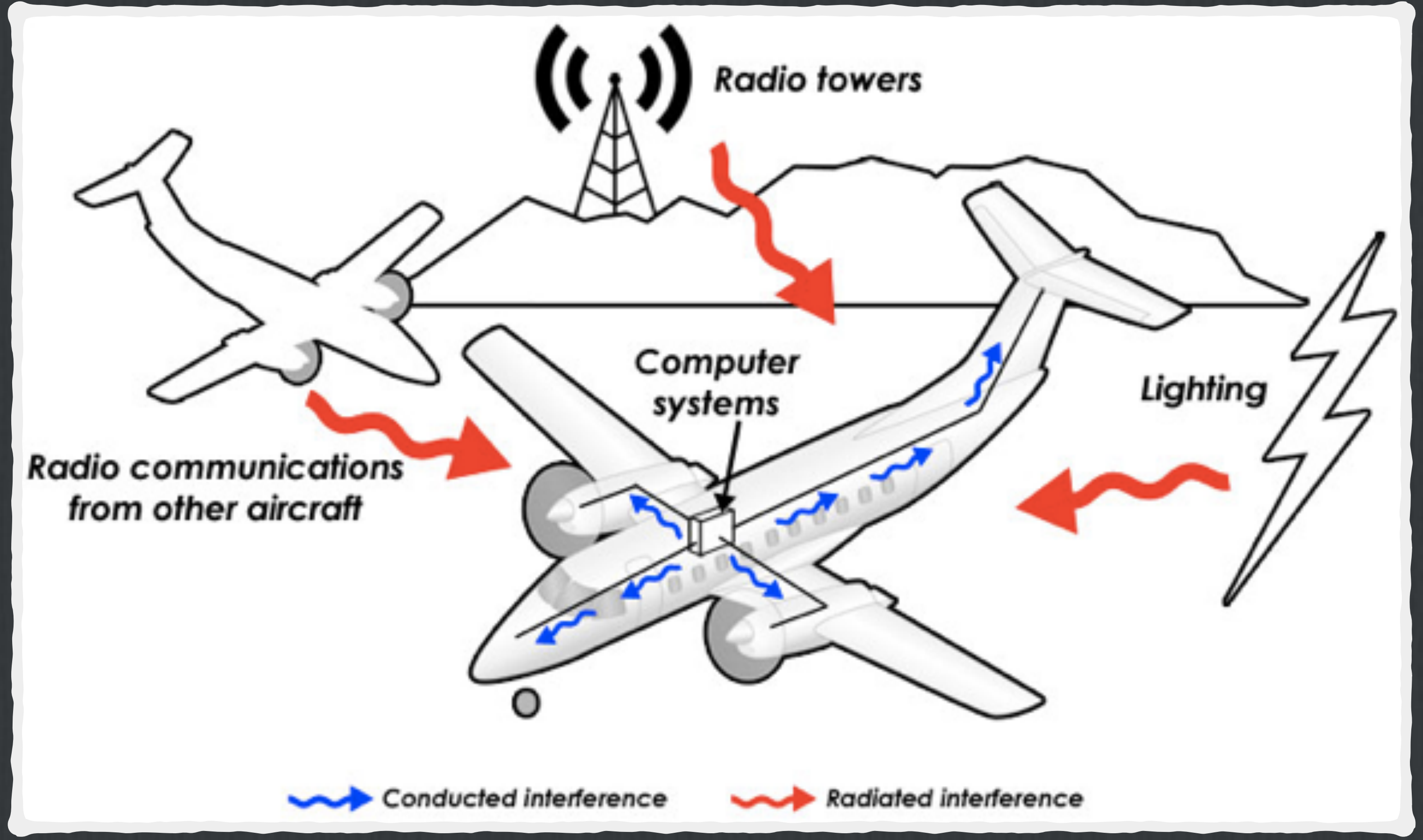
Quantifying the Resiliency of Fail-Operational Real-Time Networked Control Systems

Arpan Gujarati, Mitra Nasri, Björn B. Brandenburg

MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS

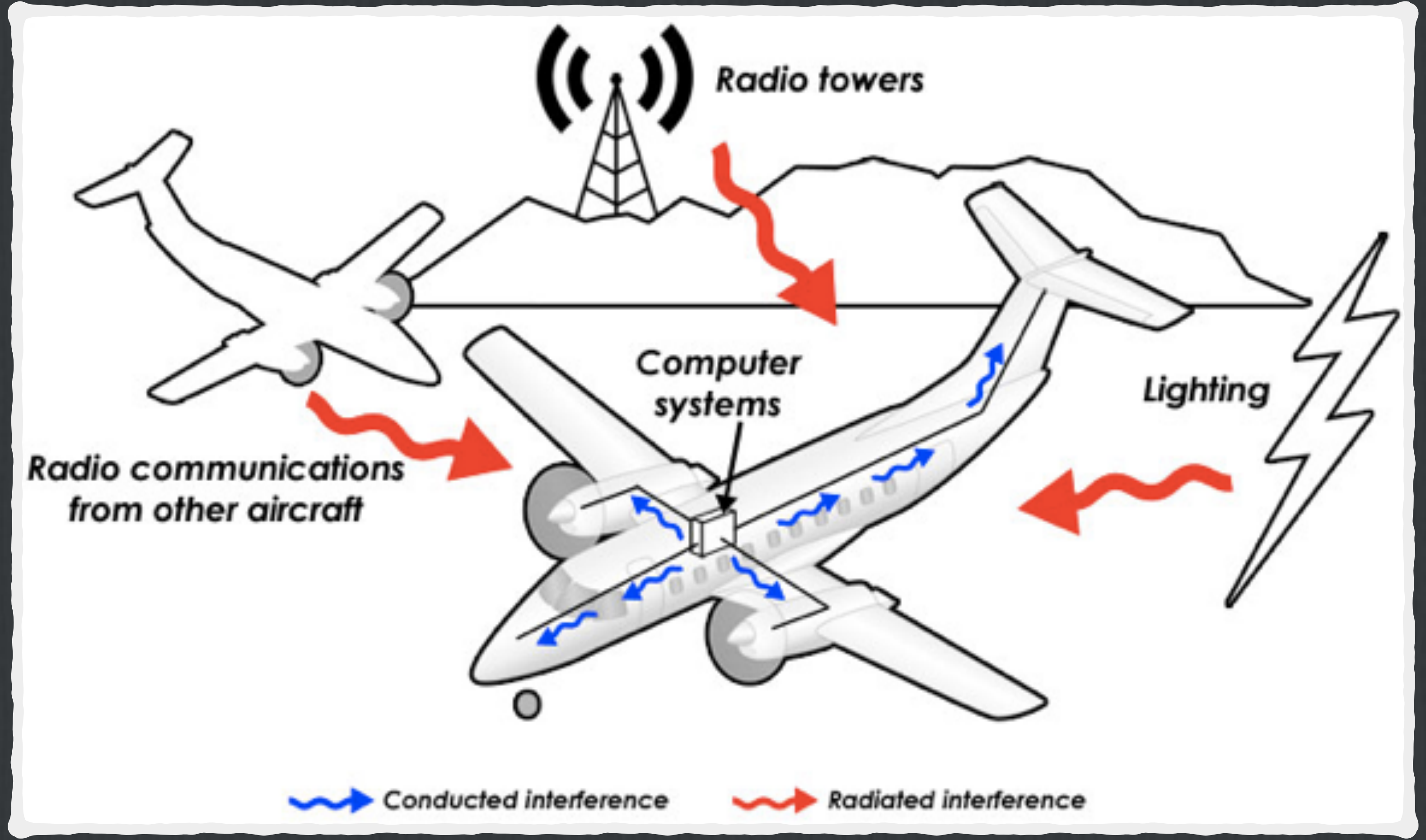
Embedded systems are susceptible to environmentally-induced transient faults

- Harsh environments
  - ➡ Robots operating under hard radiation
  - ➡ Industrial systems near high-power machinery
  - ➡ Electric motors inside automobile systems
- Bit-flips in registers, buffers, network



Embedded systems are susceptible to environmentally-induced transient faults

- Harsh environments
  - ➡ Robots operating under hard radiation
  - ➡ Industrial systems near high-power machinery
  - ➡ Electric motors inside automobile systems
- Bit-flips in registers, buffers, network



Example*

*Mancuso R. Next-generation safety-critical systems on multi-core platforms. PhD thesis, UIUC, 2017.

- ➡ One bit-flip in a 1 MB SRAM every 10^{12} hours of operation
- ➡ 0.5 billion cars with an average daily operation time of 5%
- ➡ About 5,000 cars are affected by a bit-flip every day

Failures and errors due to transient faults in distributed real-time systems

Failures and errors due to transient faults in distributed real-time systems

- Transmission errors
  - ➡ Faults on the network
- Omission Errors
  - ➡ Fault-induced kernel panics
- Incorrect computation Errors
  - ➡ Faults in the memory buffers

Failures and errors due to transient faults in distributed real-time systems

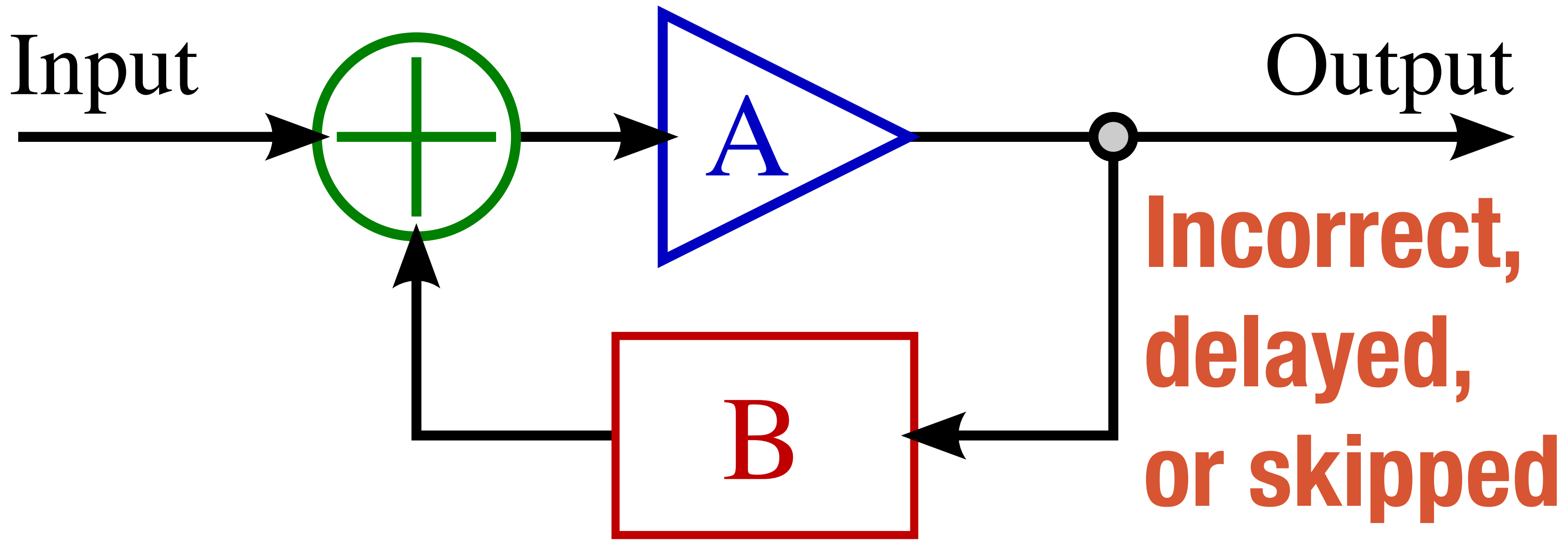
- ☐ Transmission errors
  - ➡ Faults on the network
- ☐ Omission Errors
  - ➡ Fault-induced kernel panics
- ☐ Incorrect computation Errors
  - ➡ Faults in the memory buffers

Failures in:

➡ value domain (incorrect outputs)

➡ time domain (deadline violations)

E.g., safety-critical control system



Mitigating the effects of transient faults in distributed real-time systems

Mitigating the effects of transient faults in distributed real-time systems

- Transmission errors
  - ➡ Faults on the network
- Omission Errors
  - ➡ Fault-induced kernel panics
- Incorrect computation Errors
  - ➡ Faults in the memory buffers

Retransmissions at the network layer

Dual modular redundancy (DMR)

Triple modular redundancy (TMR)

Mitigating the effects of transient faults in distributed real-time systems

How can we objectively compare the reliability offered by different mitigation techniques?

- Omission Errors
  - ➡ Fault-induced kernel panics
- Incorrect computation Errors
  - ➡ Faults in the memory buffers

Retransmissions at the network layer

Dual modular redundancy (DMR)

Triple modular redundancy (TMR)

Mitigating the effects of transient faults in distributed real-time systems

How does the real-time requirement affect system reliability?
When does it really become a bottleneck?

- Omission Errors
  - ➡ Fault-induced kernel panics
- Incorrect computation Errors
  - ➡ Faults in the memory buffers

Dual modular redundancy (DMR)

Triple modular redundancy (TMR)

Mitigating the effects of transient faults in distributed real-time systems

How does the real-time requirement affect system reliability?
When does it really become a bottleneck?

- Omission Errors
  - ➡ Fault-induced kernel panics

Dual modular redundancy (DMR)

What if the system is weakly-hard real-time,
i.e., it can tolerate a few failures?

dular redundancy (TMR)

Problem: Reliability analysis of networked control systems

Problem: Reliability analysis of networked control systems

Given

1. Networked control system (messages, period)
2. Robustness specification (weakly-hard constraints)
3. Active replication scheme (DMR, TMR, others)
4. Peak transient fault rates (for the network and the hosts)

Problem: Reliability analysis of networked control systems

Given

1. Networked control system (messages, period)
2. Robustness specification (weakly-hard constraints)
3. Active replication scheme (DMR, TMR, others)
4. Peak transient fault rates (for the network and the hosts)

Objective

A safe upper bound on the failure rate of the networked control system

Problem: Reliability analysis of networked control systems

Given

1. Networked control system (messages, period)
2. Robustness specification (weakly-hard constraints)
3. Active replication scheme (DMR, TMR, others)
4. Peak transient fault rates (for the network and the hosts)

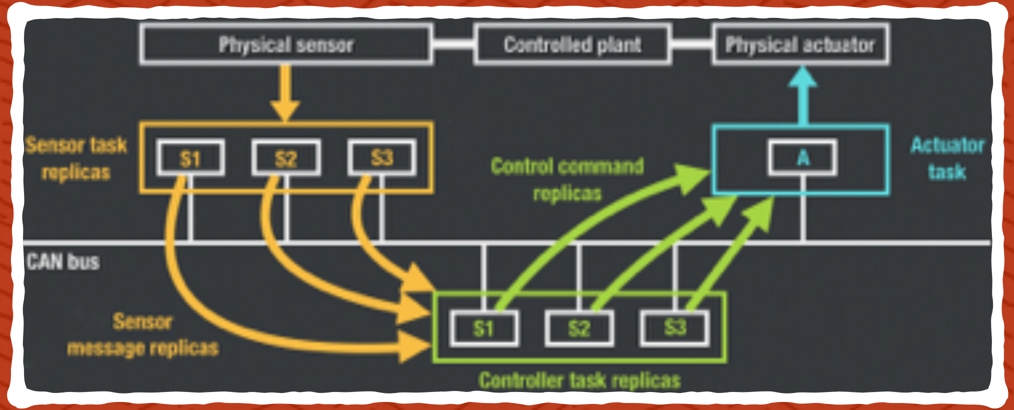
Objective

A safe upper bound on the failure rate of the networked control system

Failures-In-Time (FIT) = Expected # failures in one billion operating hours

Outline

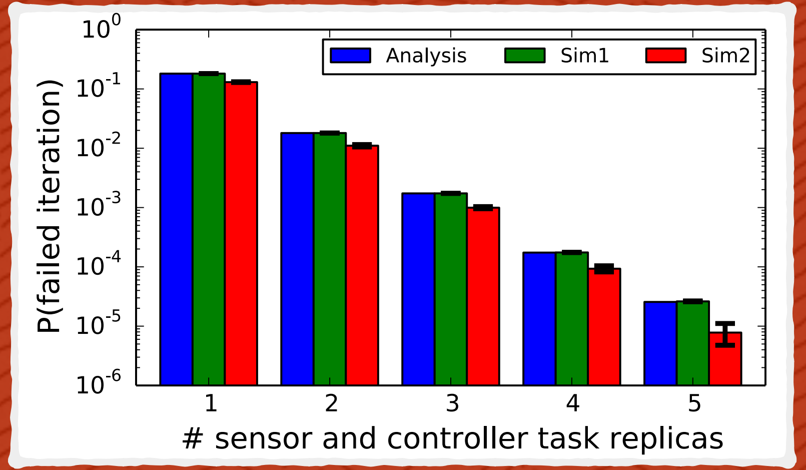
Analysis of a Controller Area Network (CAN) based networked control system



System Model

$$\int_0^\infty t \cdot f(t)\, dt$$

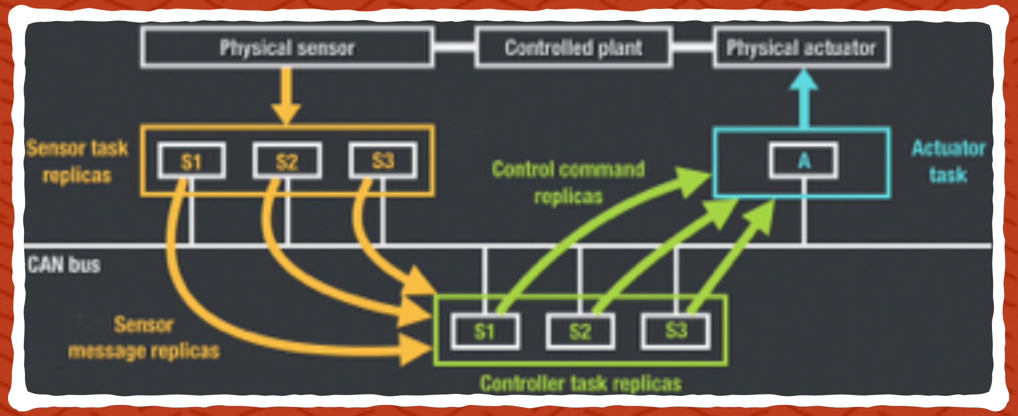
Analysis



Evaluation

Outline

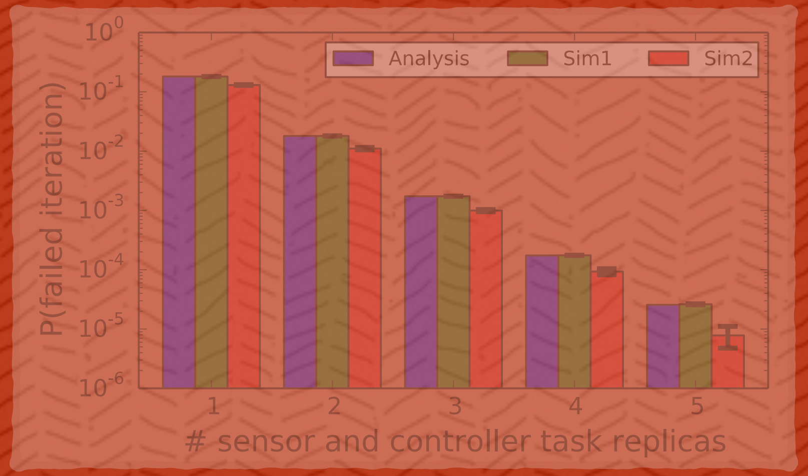
Analysis of a Controller Area Network (CAN) based networked control system



System Model

$$\int_0^\infty t \cdot f(t)\, dt$$

Analysis

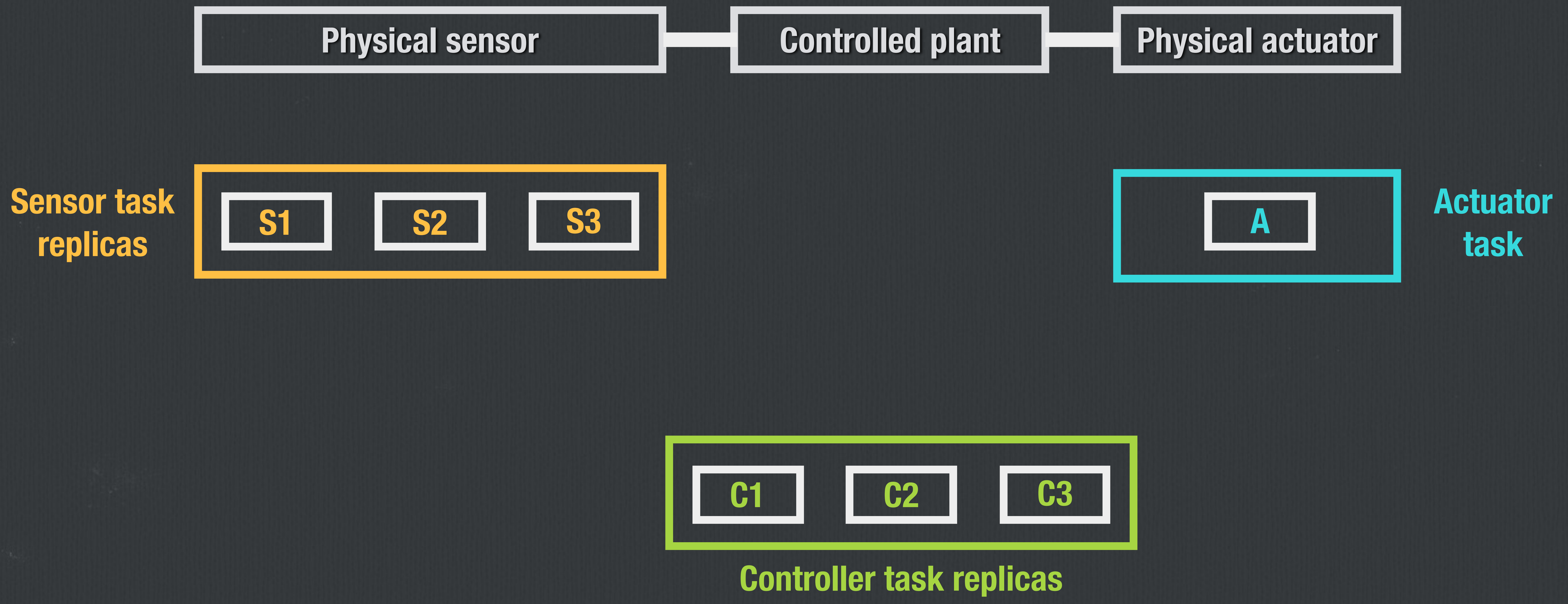


Evaluation

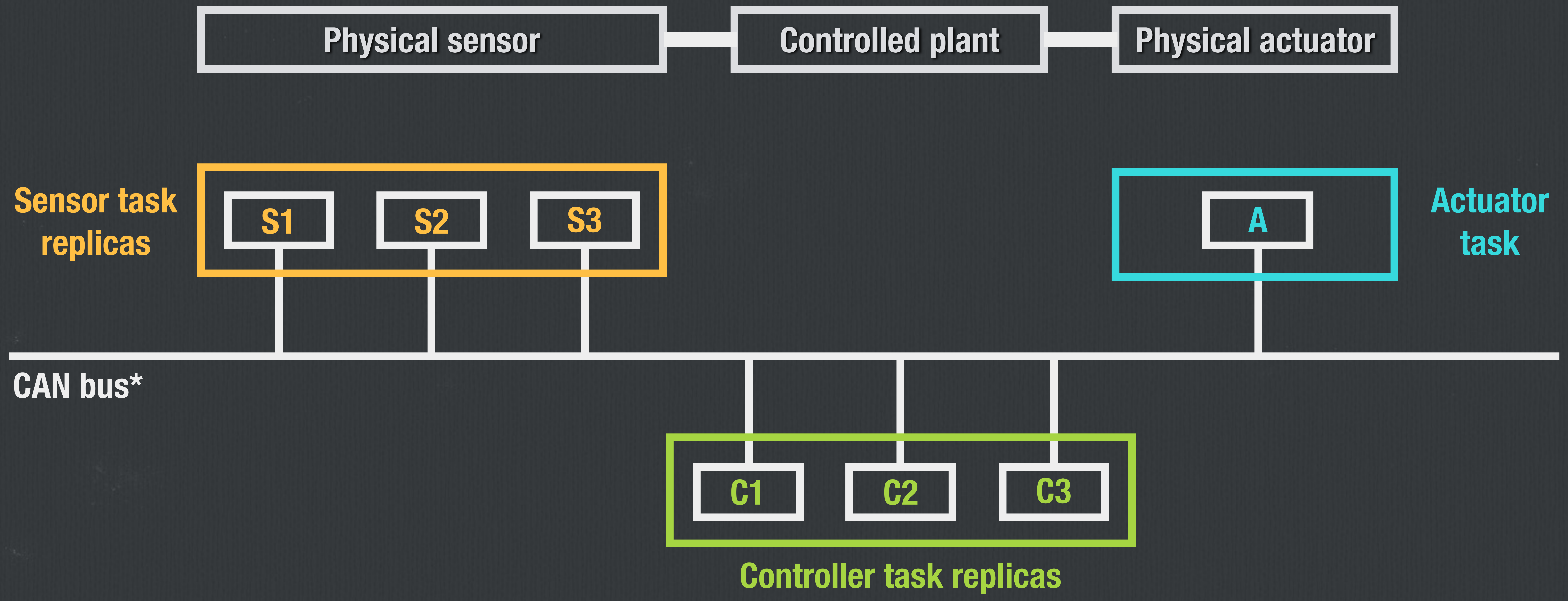
Fault tolerant single-input single-output (FT-SISO) networked control loop

Fault tolerant single-input single-output (FT-SISO) networked control loop

Fault tolerant single-input single-output (FT-SISO) networked control loop

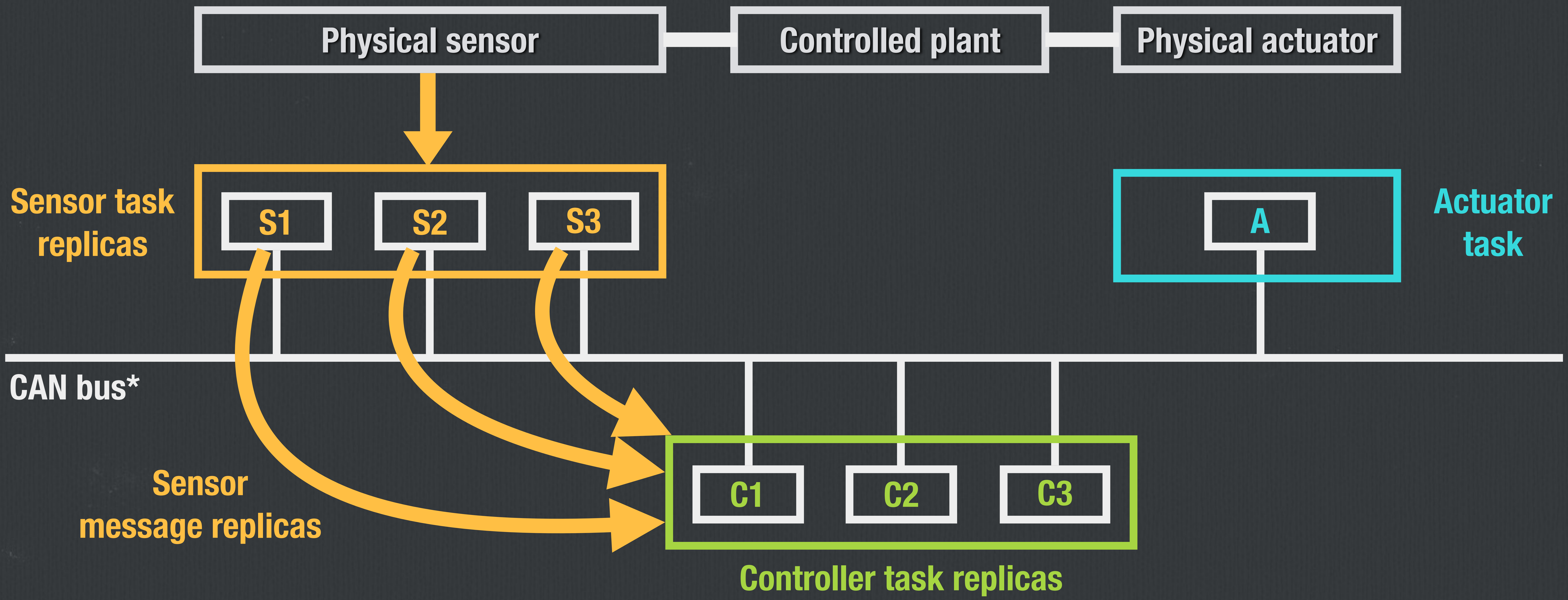


Fault tolerant single-input single-output (FT-SISO) networked control loop



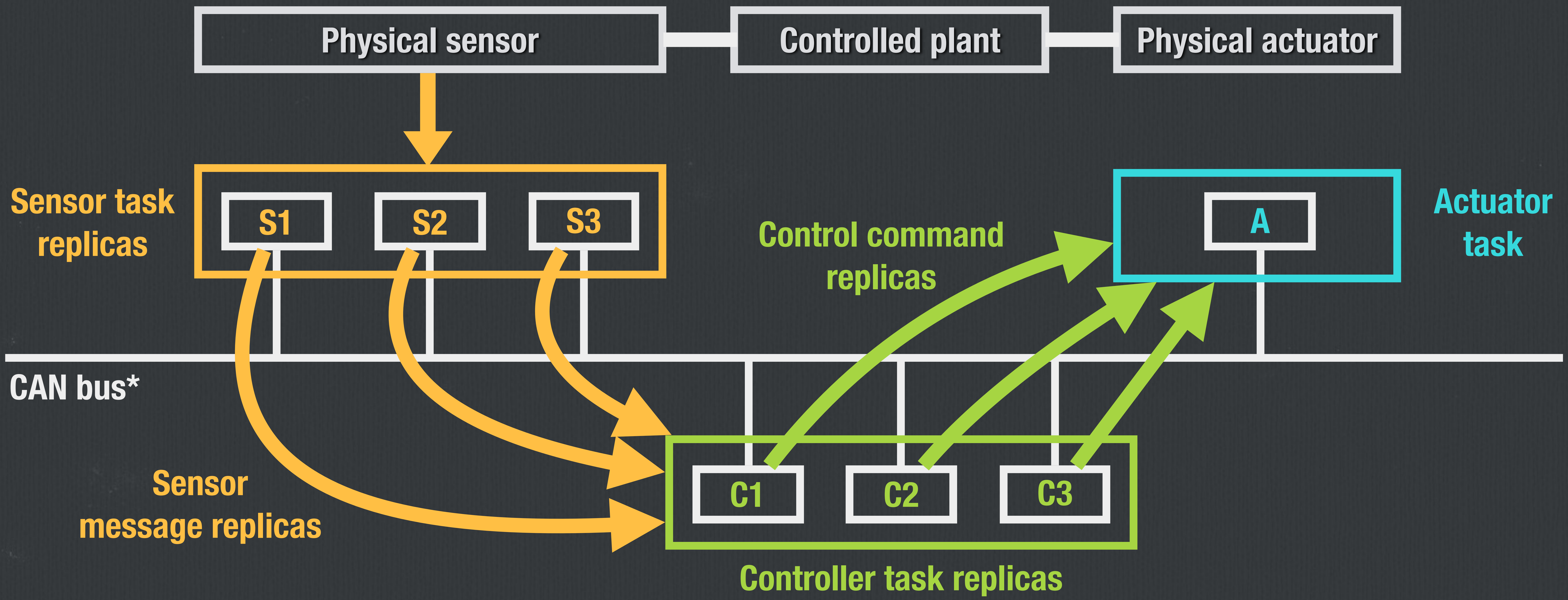
* Controller Area Network

Fault tolerant single-input single-output (FT-SISO) networked control loop



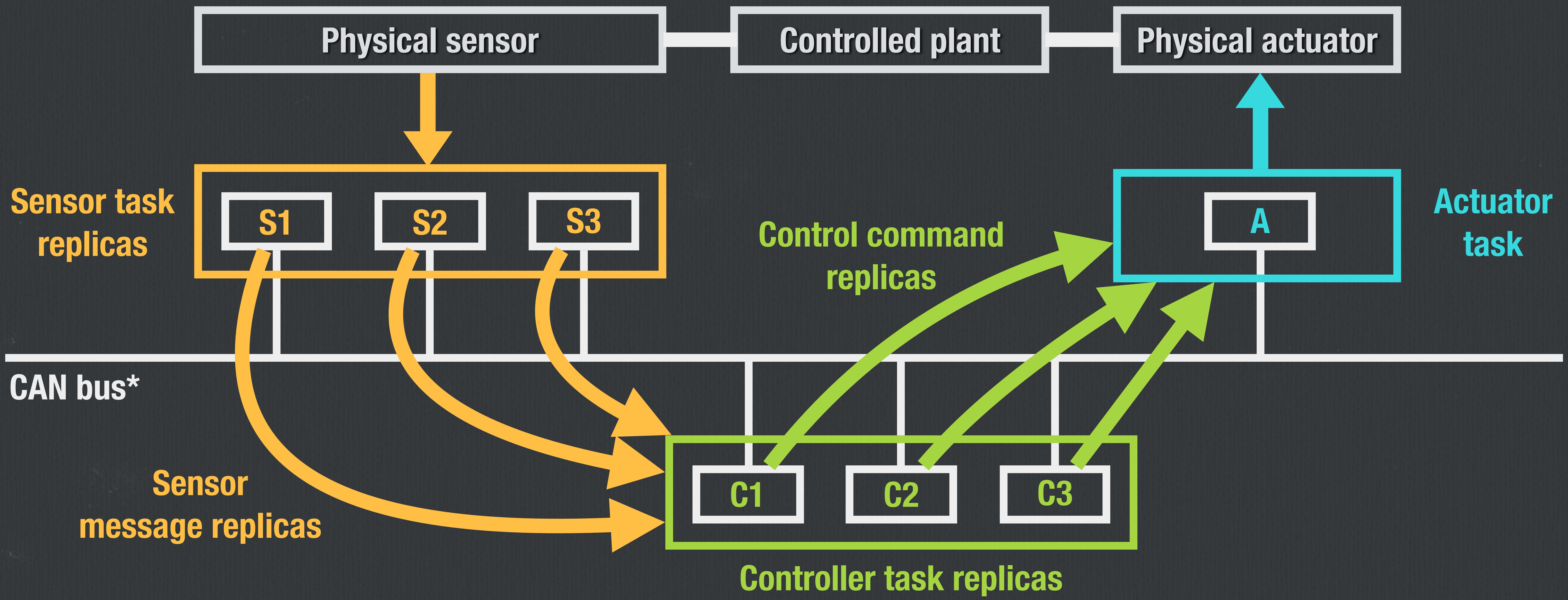
* Controller Area Network

Fault tolerant single-input single-output (FT-SISO) networked control loop



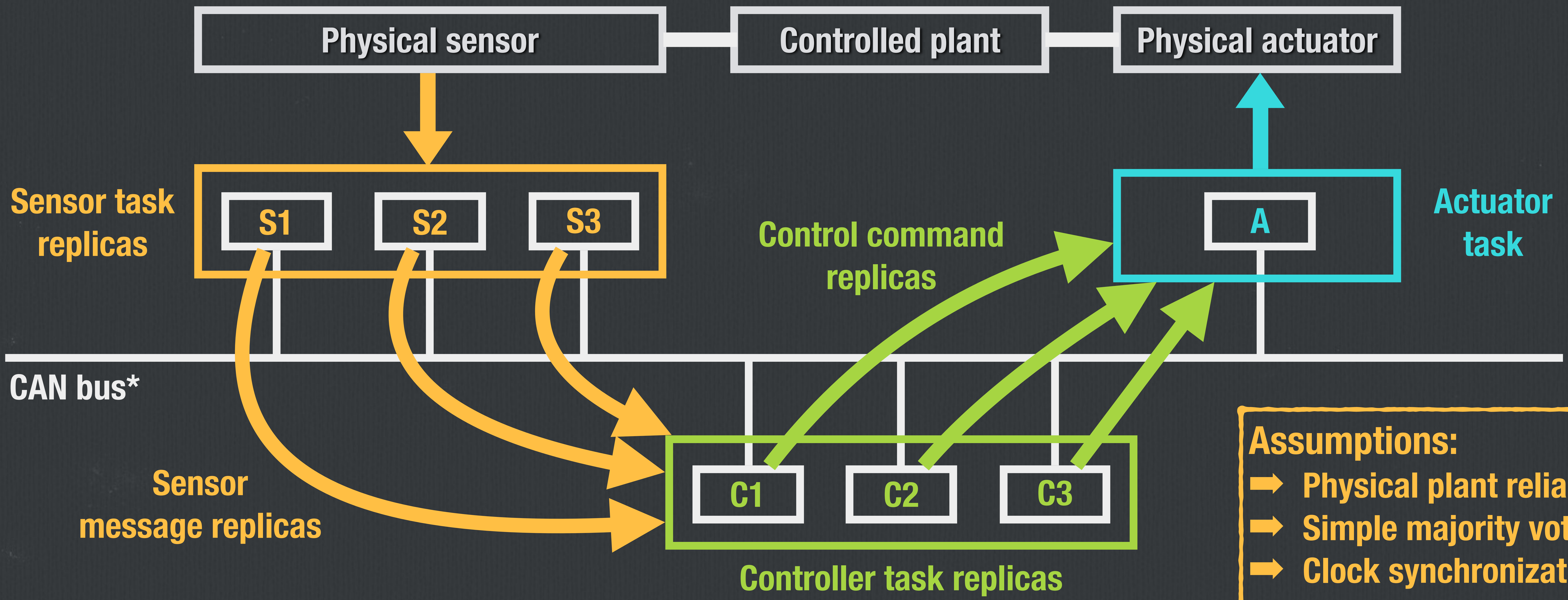
* Controller Area Network

Fault tolerant single-input single-output (FT-SISO) networked control loop



* Controller Area Network

Fault tolerant single-input single-output (FT-SISO) networked control loop



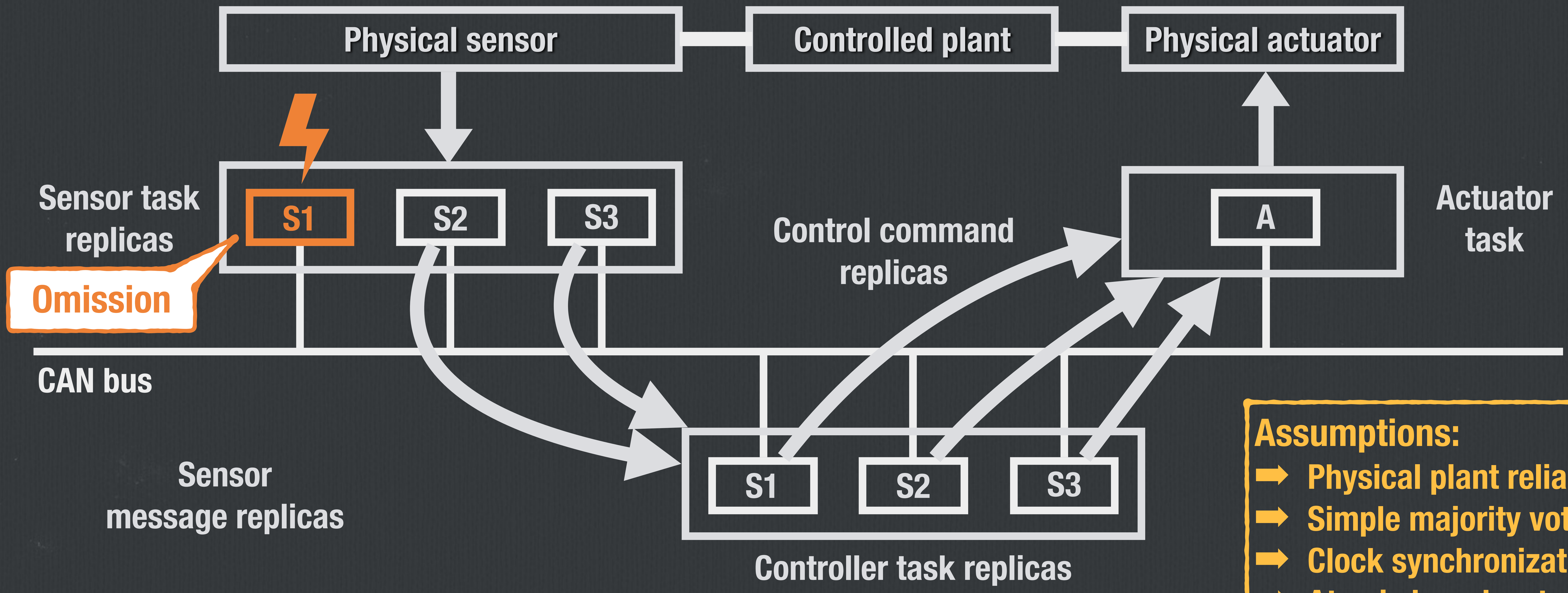
* Controller Area Network

Assumptions:

- ➡ Physical plant reliable
- ➡ Simple majority voting
- ➡ Clock synchronization
- ➡ Atomic broadcast

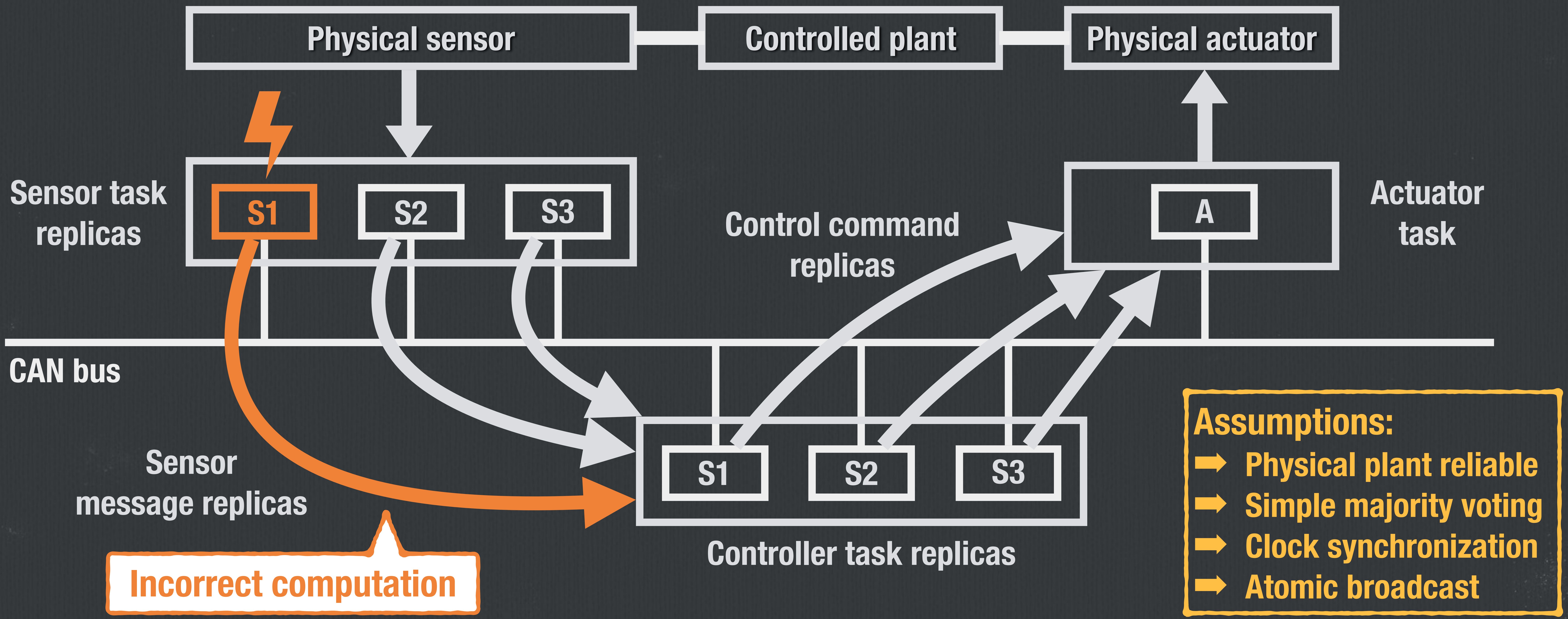
Failures and errors in a FT-SISO networked control loop

Failures and errors in a FT-SISO networked control loop

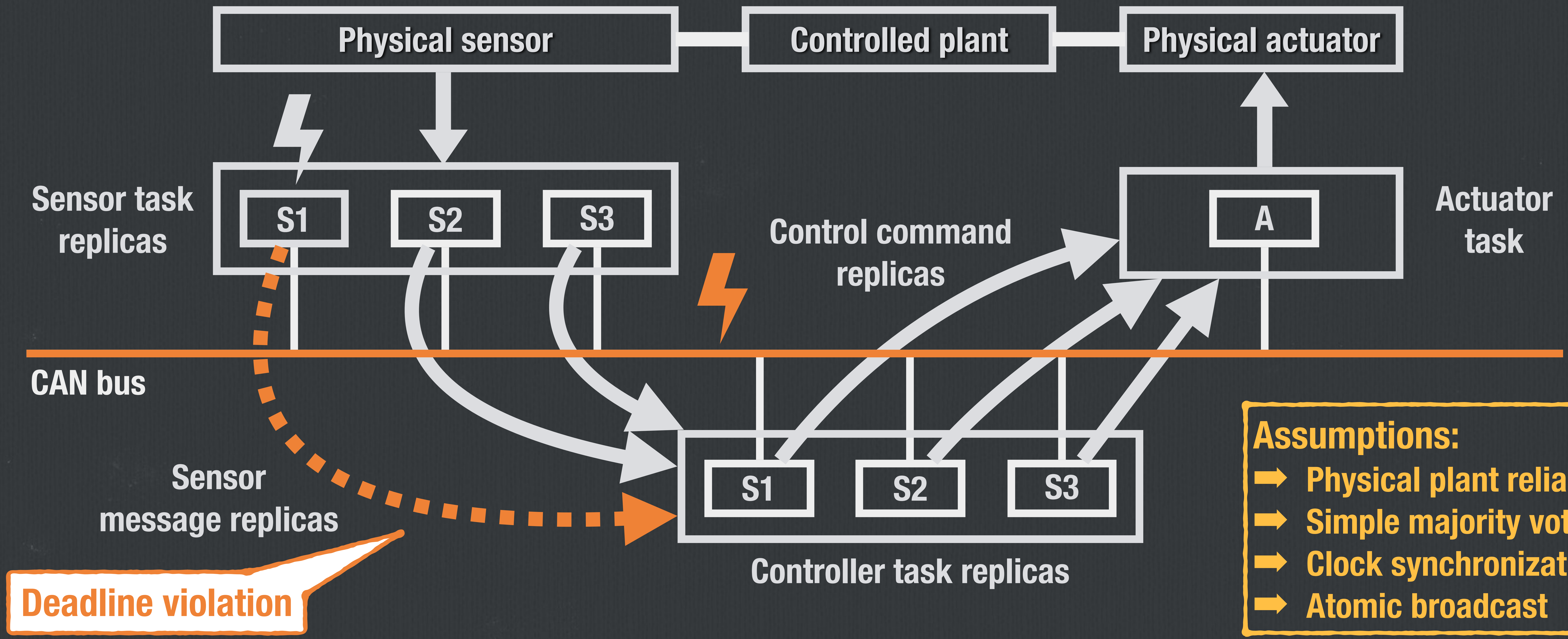


Assumptions:
- ➡ Physical plant reliable
- ➡ Simple majority voting
- ➡ Clock synchronization
- ➡ Atomic broadcast

Failures and errors in a FT-SISO networked control loop

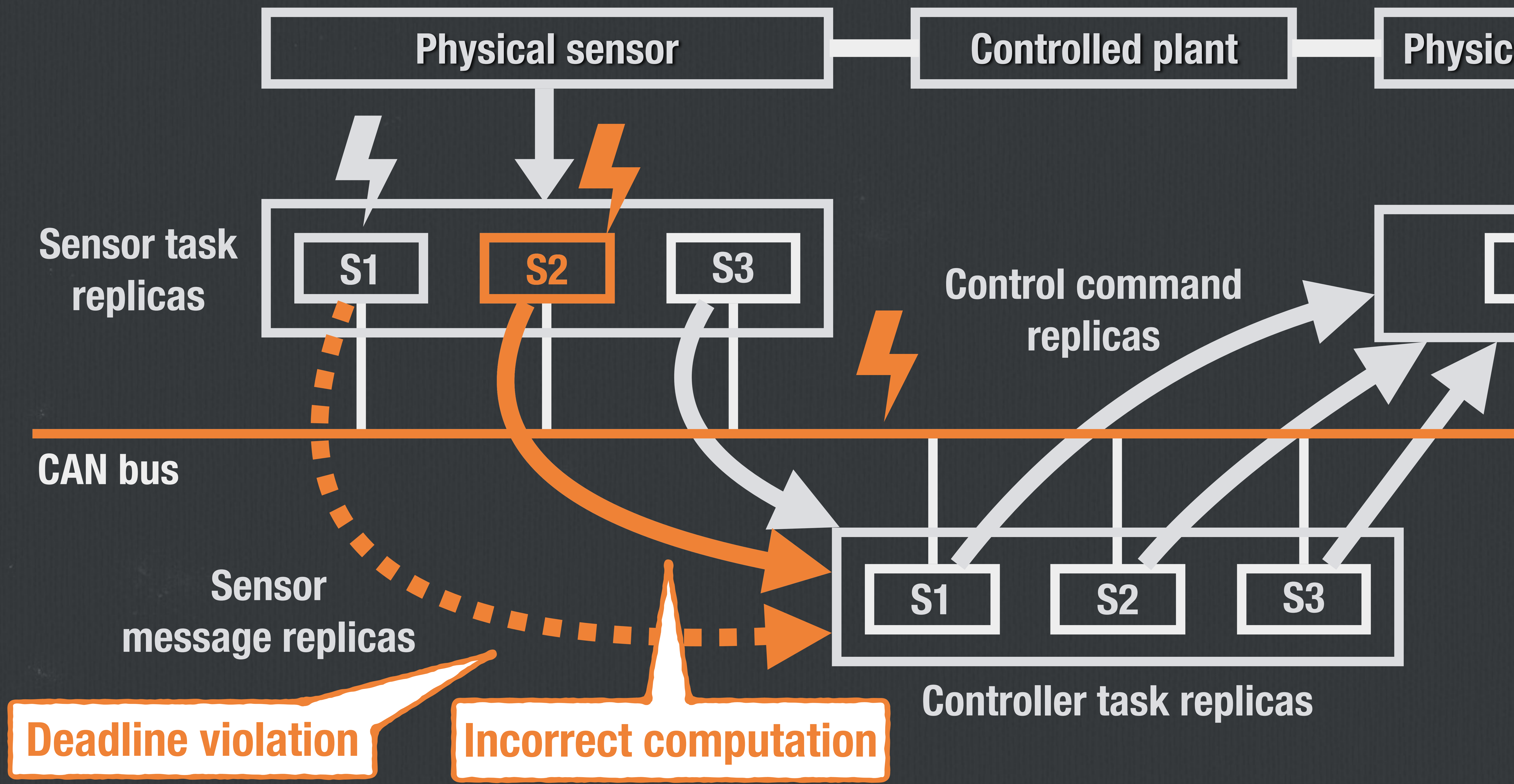


Failures and errors in a FT-SISO networked control loop



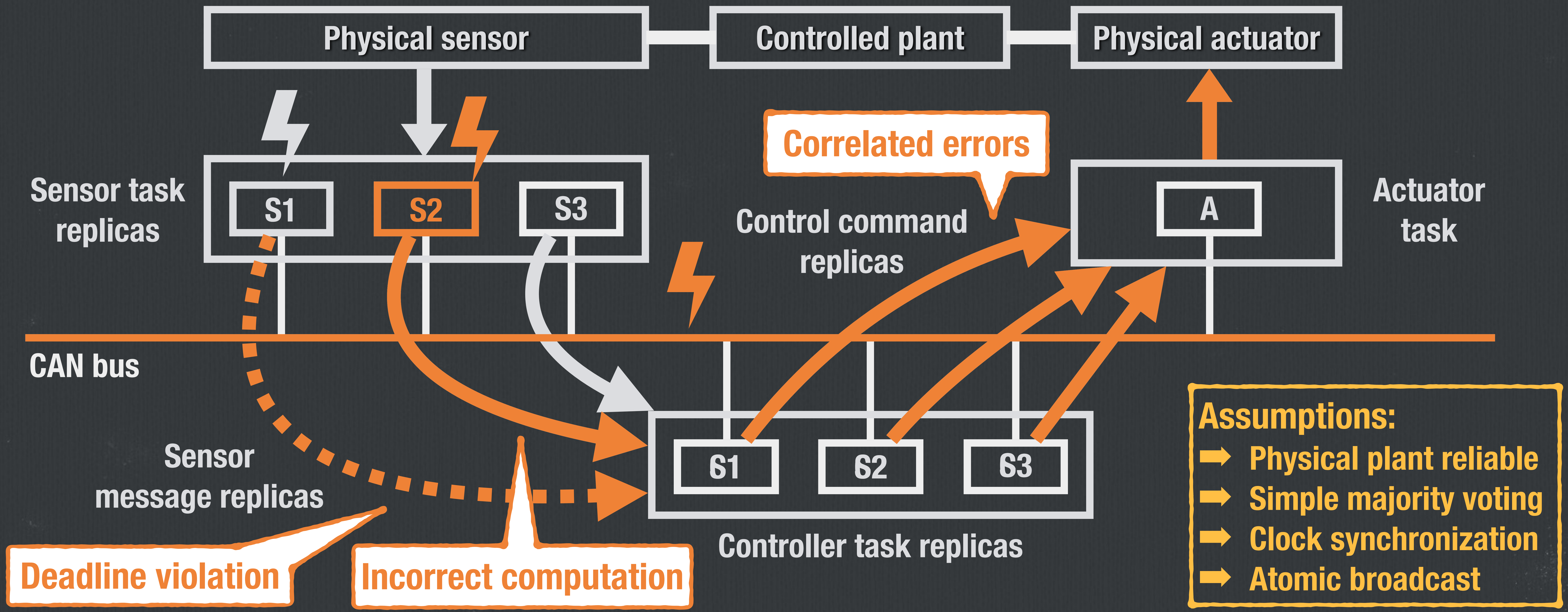
Assumptions:
- ➡ Physical plant reliable
- ➡ Simple majority voting
- ➡ Clock synchronization
- ➡ Atomic broadcast

Failures and errors in a FT-SISO networked control loop

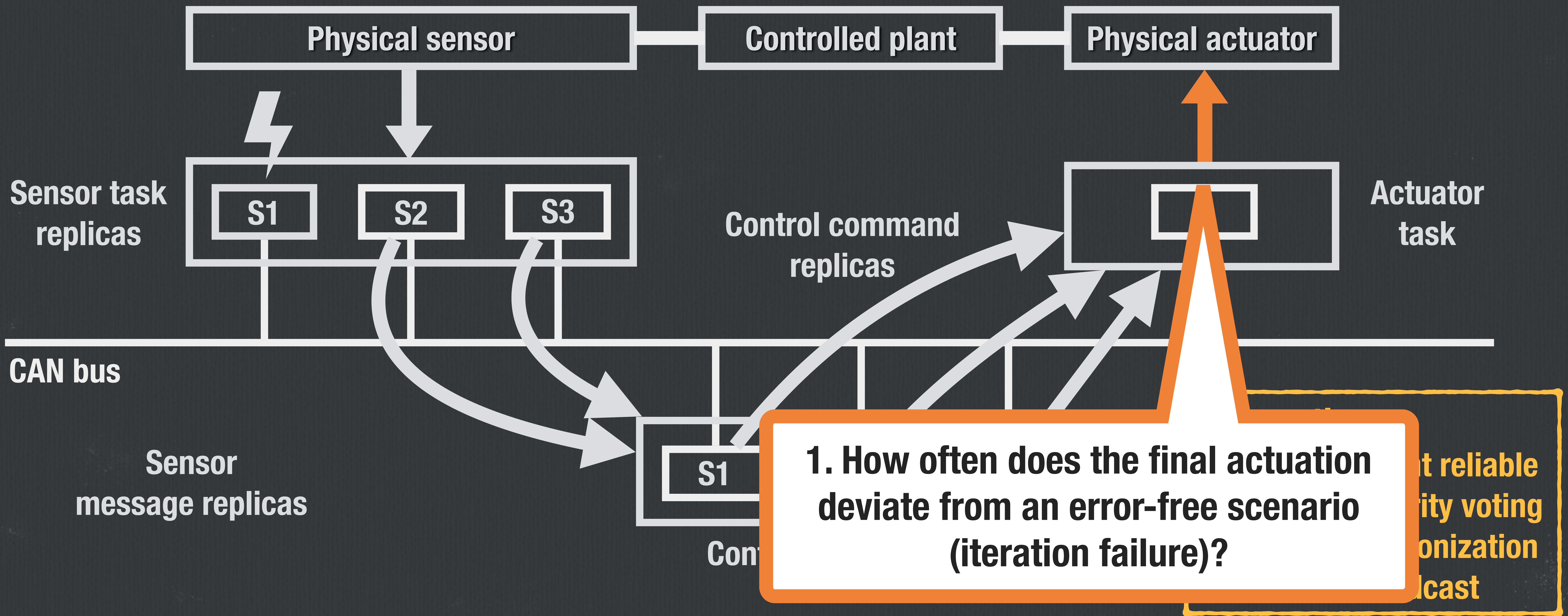


Assumptions:
- Physical plant reliable
- Simple majority voting
- Clock synchronization
- Atomic broadcast

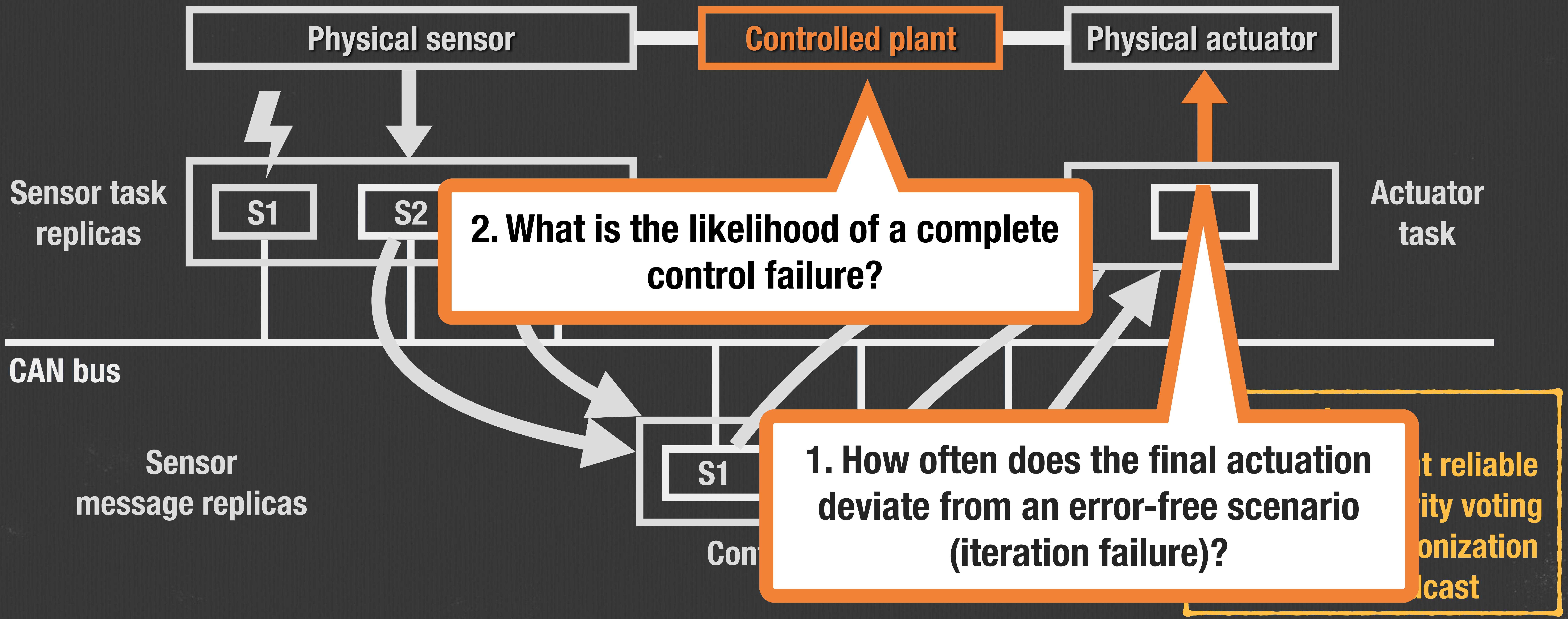
Failures and errors in a FT-SISO networked control loop



Failures and errors in a FT-SISO networked control loop



Failures and errors in a FT-SISO networked control loop



1. Modeling control loop iteration failures

Control loop iterations I_1 I_2 I_3 ... I_{n-1} I_n I_{n+1} ...

1. Modeling control loop iteration failures

Control loop iterations I_1 I_2 I_3 … I_{n-1} I_n I_{n+1} …

1 Final actuation is successful

Error-free

1

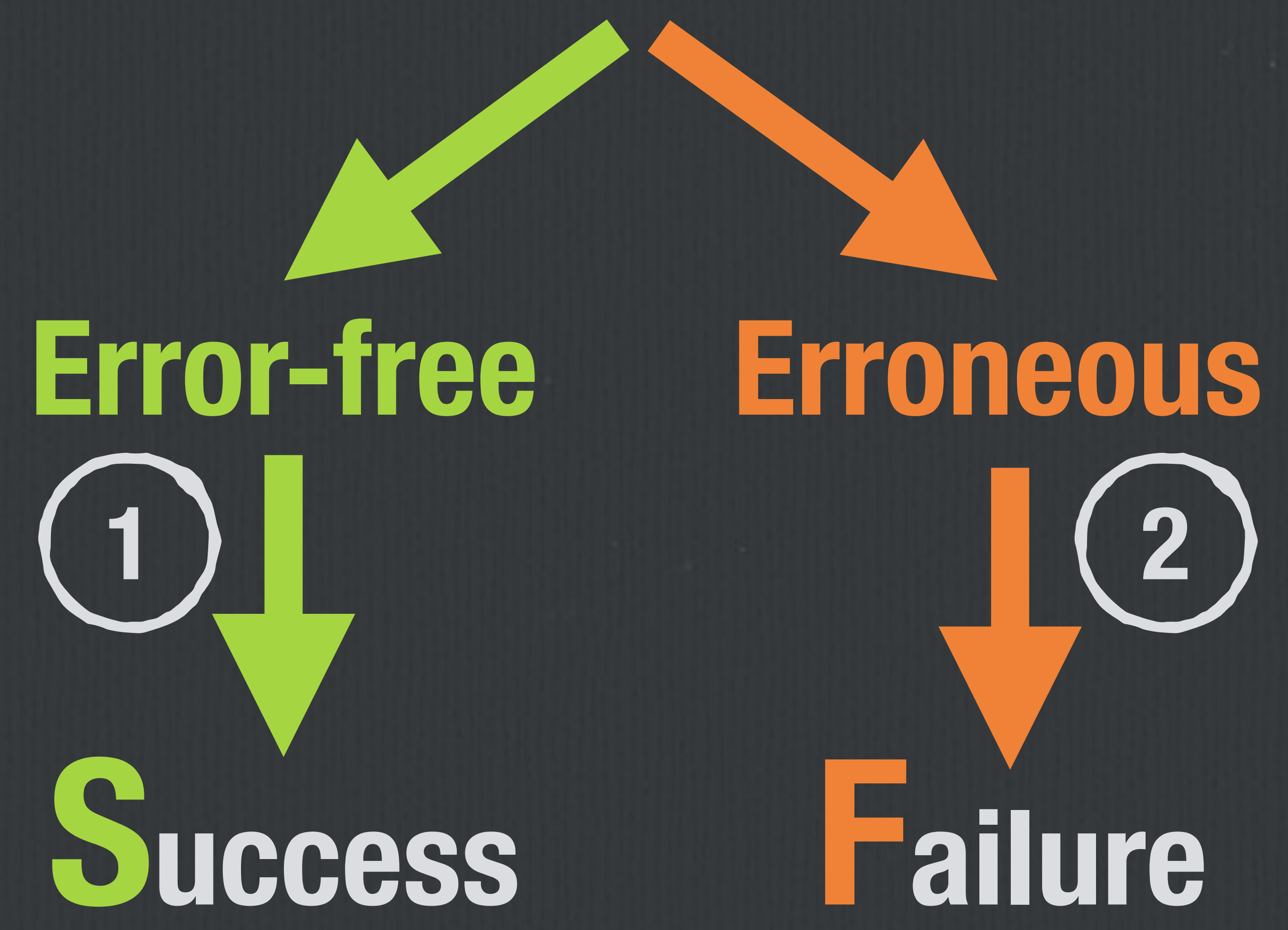
Success

1. Modeling control loop iteration failures

Control loop iterations $I_1 \; I_2 \; I_3 \; \cdots \; I_{n-1} \; I_n \; I_{n+1} \; \cdots$

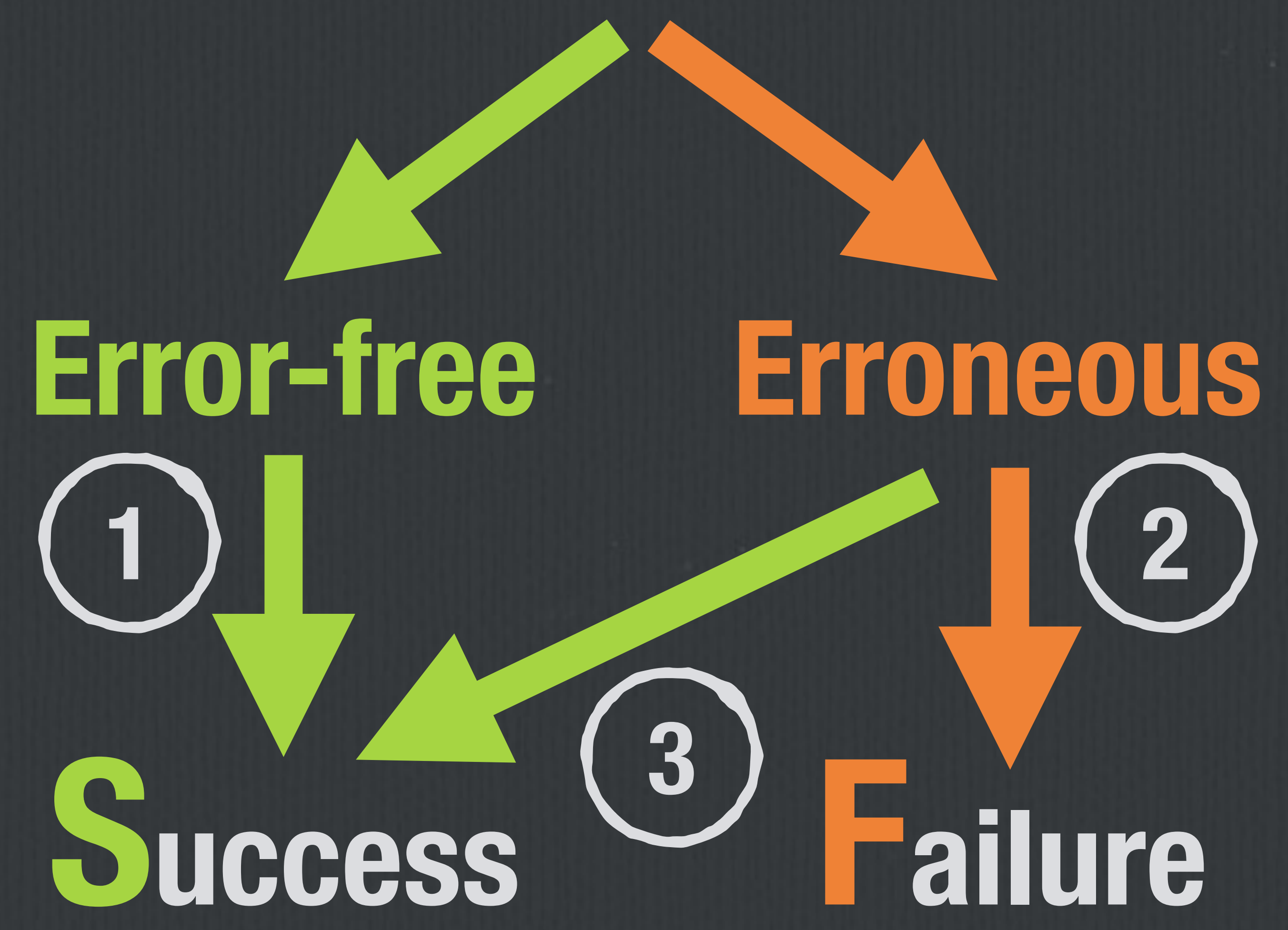
1 Final actuation is successful

2 Final actuation failed (different from 1)



1. Modeling control loop iteration failures

Control loop iterations I_1 I_2 I_3 ... I_{n-1} I_n I_{n+1} ...



1. Final actuation is successful
2. Final actuation failed (different from 1)
3. Final actuation is successful (same as 1) despite the errors

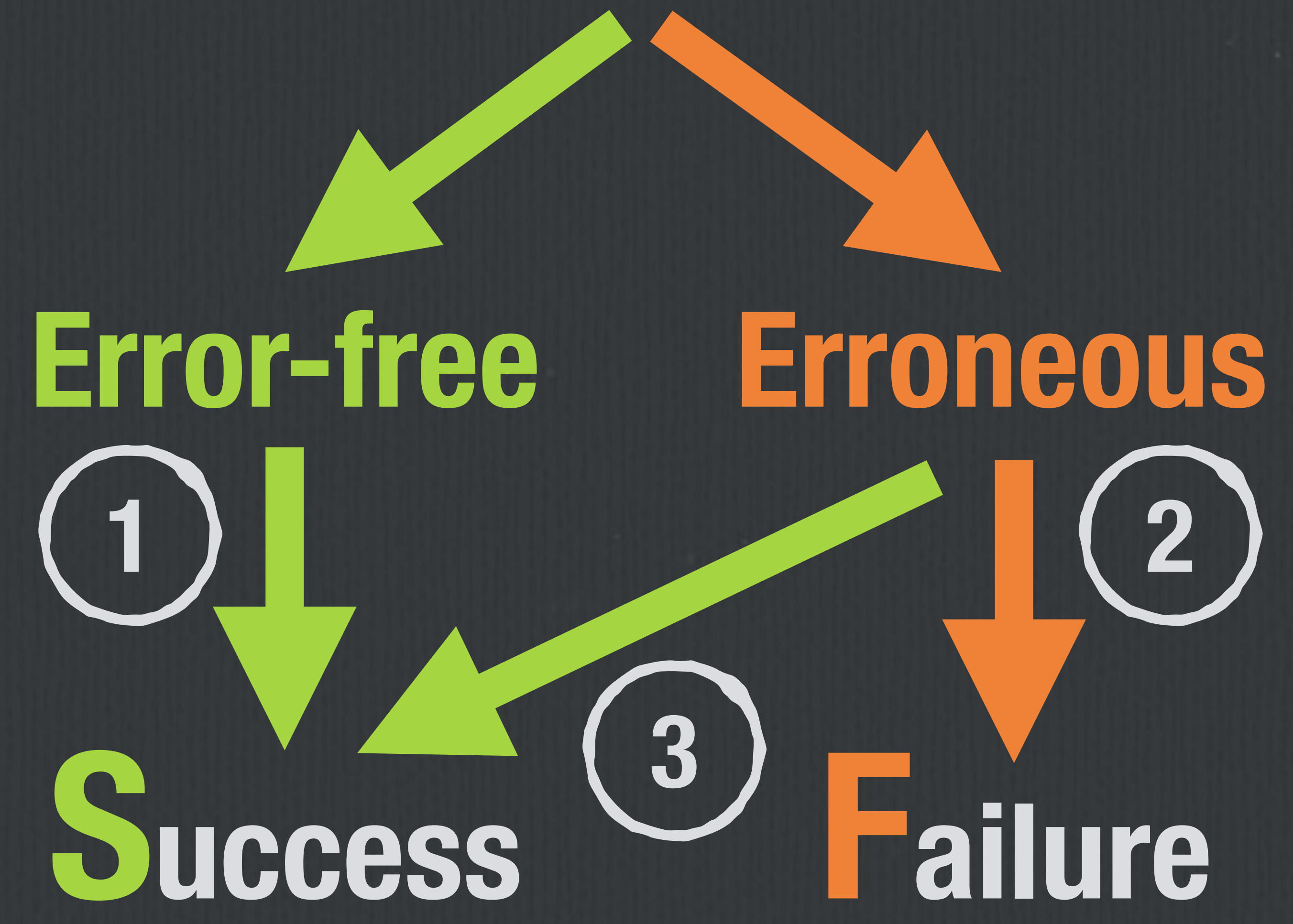
1. Modeling control loop iteration failures

Control loop iterations $I_1 \; I_2 \; I_3 \; \cdots \; I_{n-1} \; I_n \; I_{n+1} \; \cdots$

1 Final actuation is successful

2 Final actuation failed (different from 1)

3 Final actuation is successful (same as 1) despite the errors



Explicitly account for fault tolerance

2. Modeling control failure based on the (m, k)-firm constraint

2. Modeling control failure based on the (m, k)-firm constraint

Control loop iterations

Success Failure

time

S S S F S S S S F S F S S S

2. Modeling control failure based on the (m, k)-firm constraint

Control loop iterations

Success **F**ailure

time

S S S F S S S S F S F S S S

Hard constraint

Control failure upon first iteration failure

2. Modeling control failure based on the (m, k)-firm constraint

Control loop iterations

Success **F**ailure

time →

S S S F S S S S F S F S S S

Hard constraint

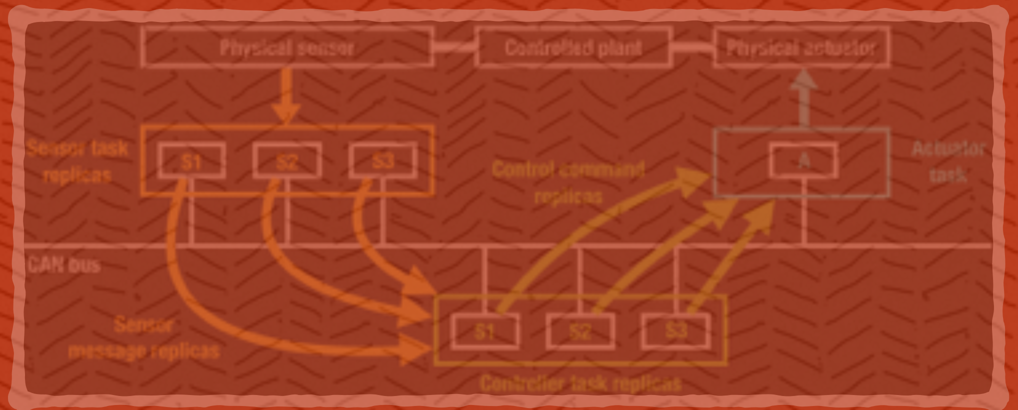
Control failure upon first iteration failure

(2, 3) constraint

Control failure when less than 2 iterations successful in 3 consecutive iterations

Outline

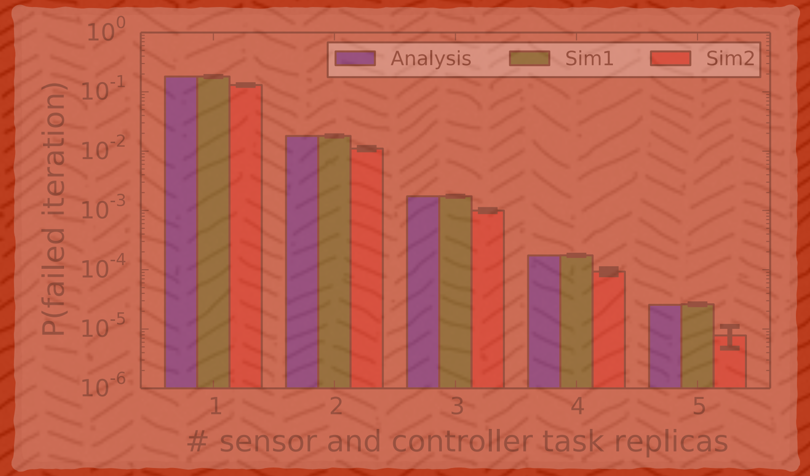
Analysis of a Controller Area Network (CAN) based networked control system



System Model

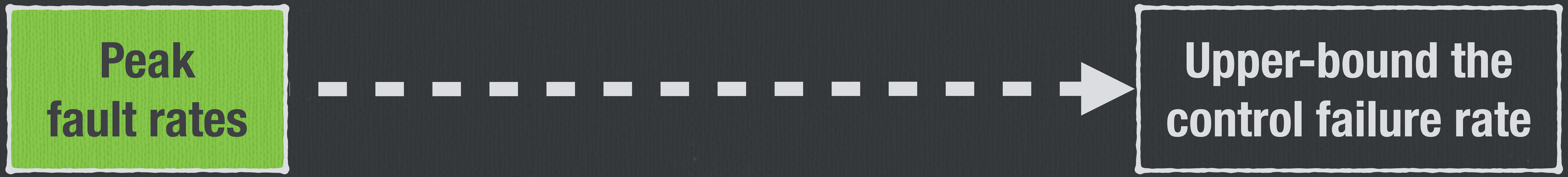
$$\int_0^\infty t \cdot f(t)\, dt$$

Analysis

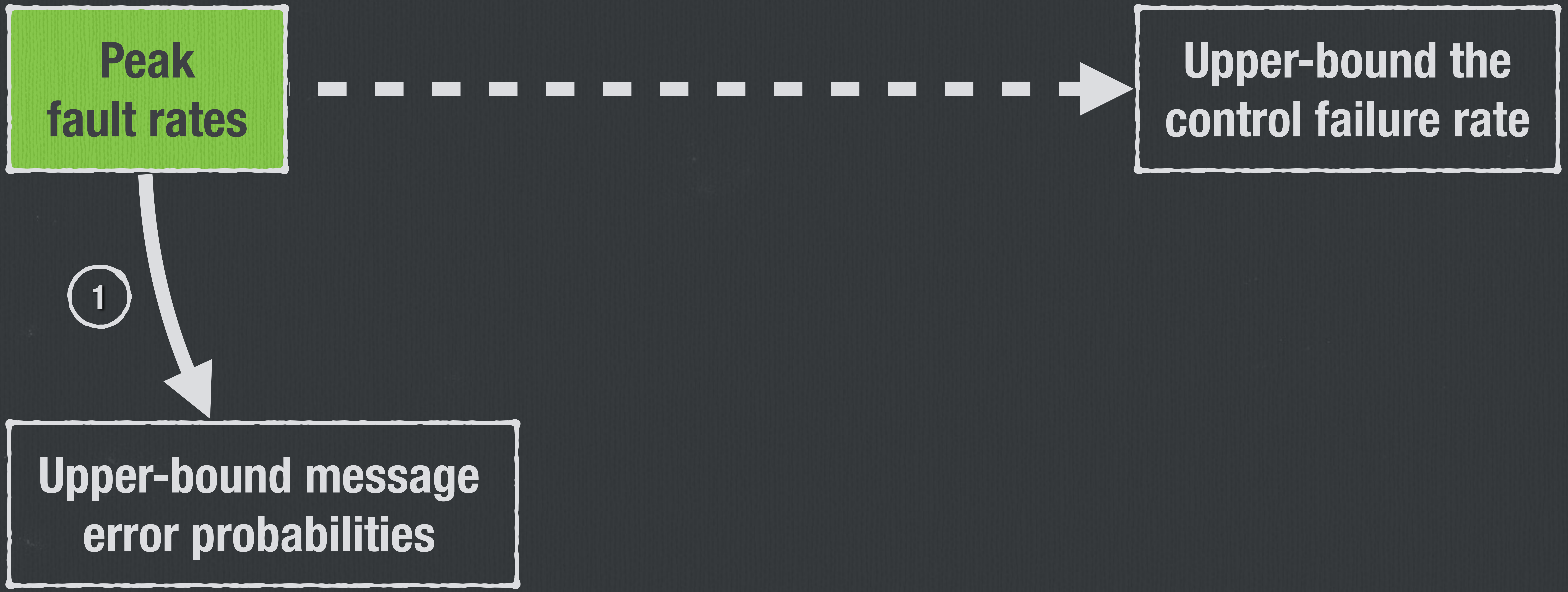


Evaluation

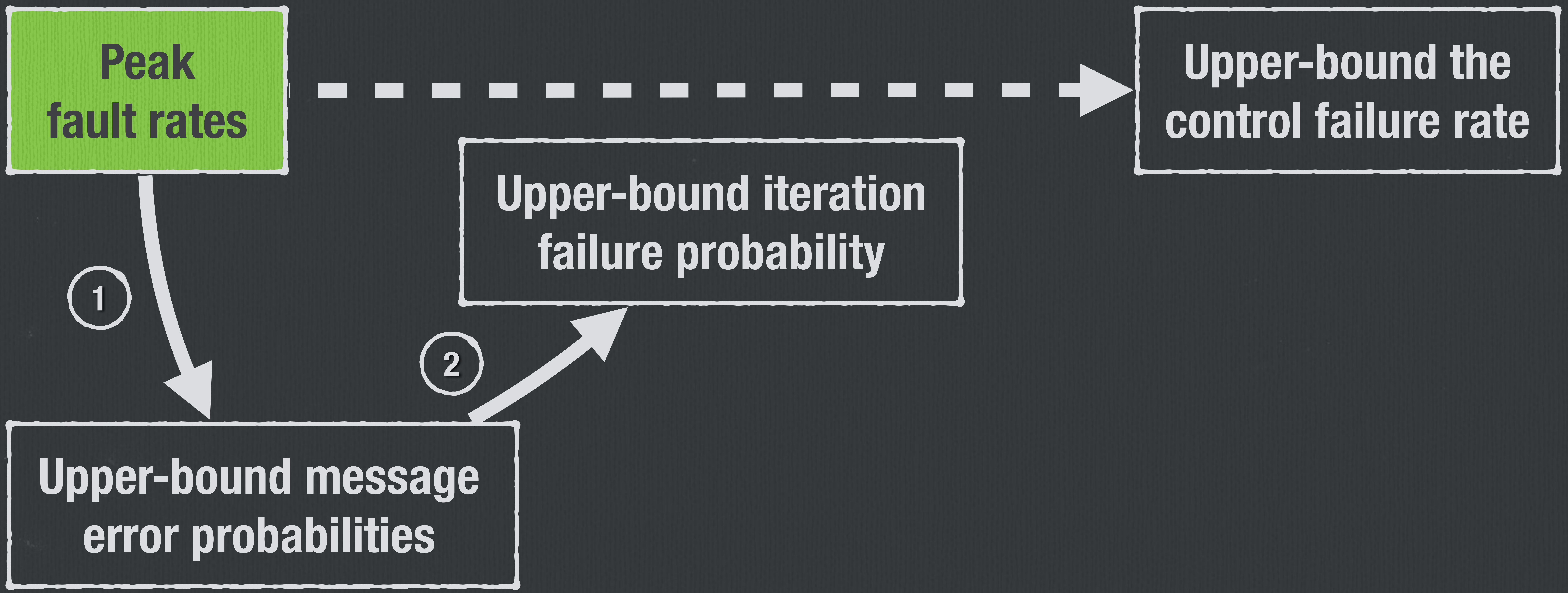
Analysis steps



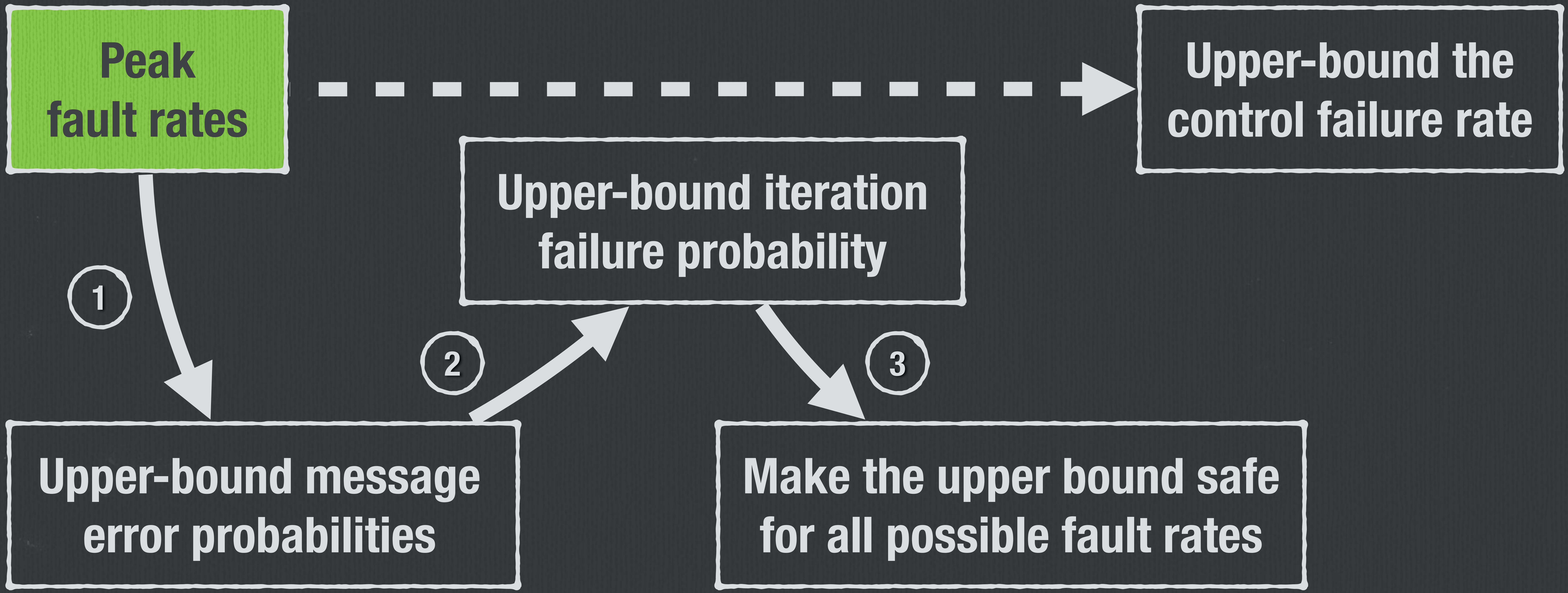
Analysis steps



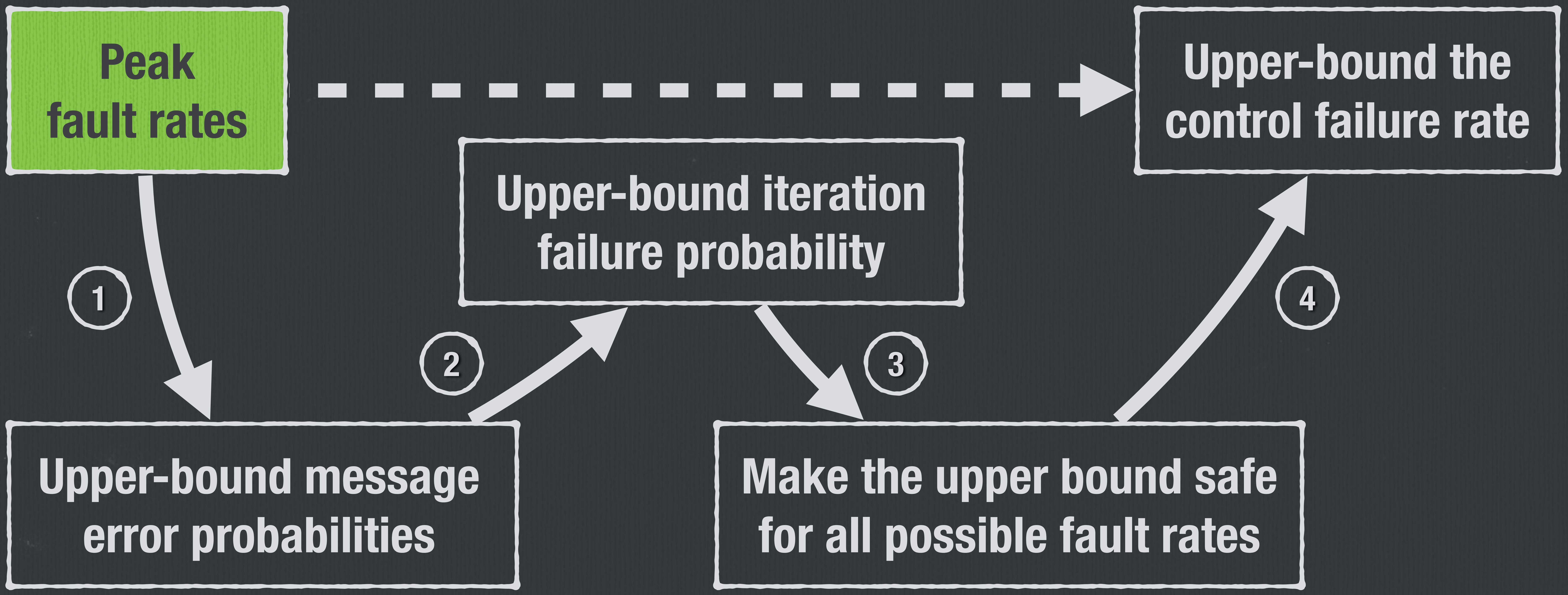
Analysis steps



Analysis steps

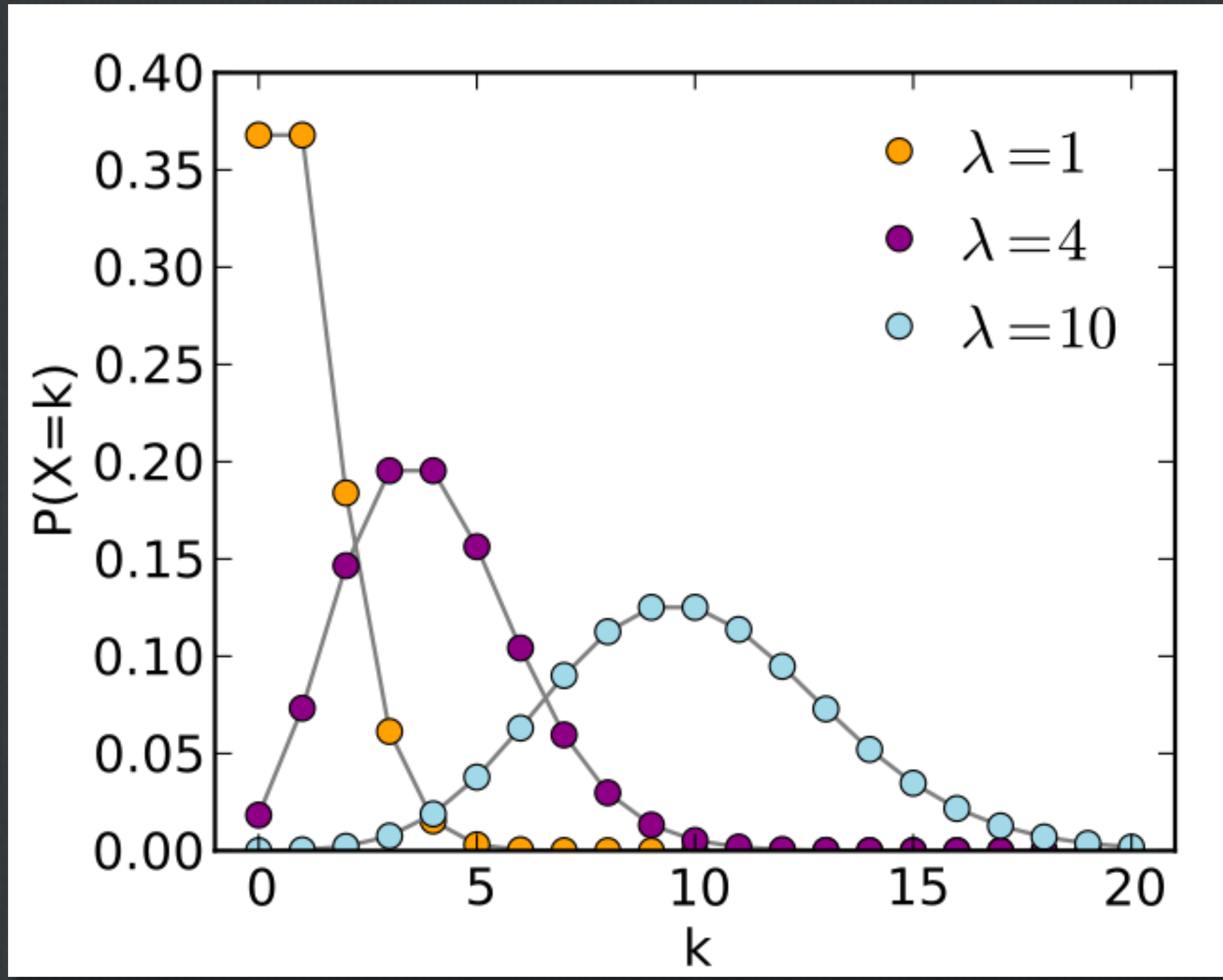


Analysis steps



Upper-bounding the message error probabilities

Using poisson model
for fault arrivals



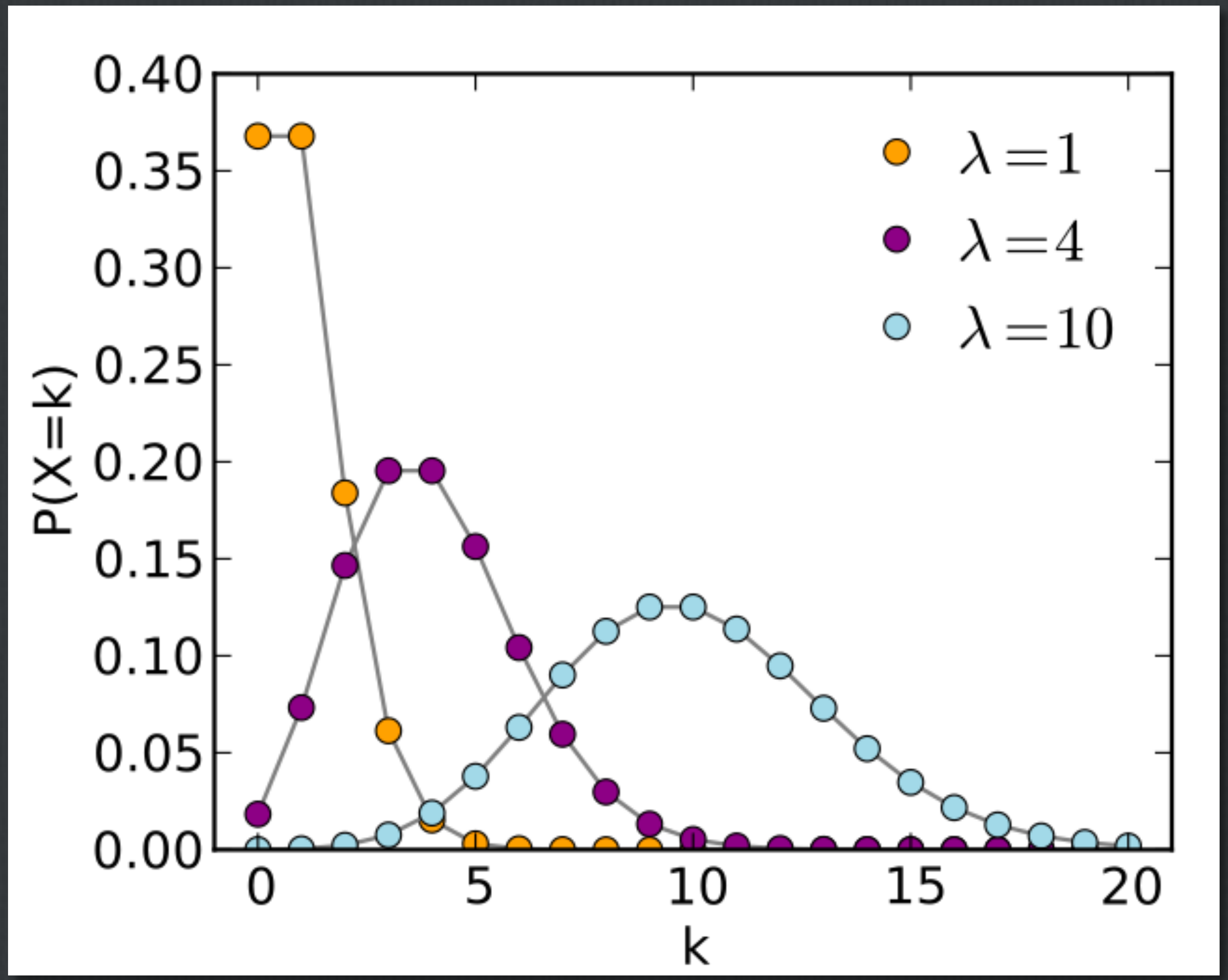
Upper-bounding the message error probabilities

Peak fault rates

Based on the message parameters

$P_1 \geq$ **P** (msg. is omitted at time t)
$P_2 \geq$ **P** (msg. is incorrectly computed)
$P_3 \geq$ **P** (msg. is misses its deadline)

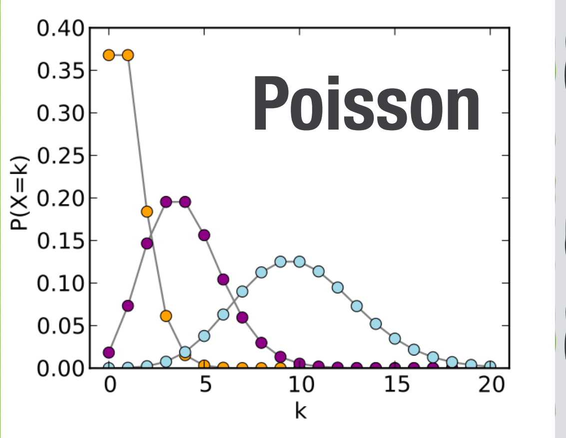
Using poisson model for fault arrivals



Analysis steps

Peak fault rates

① **Message error probability bounds** $P_1, P_2, P_3, \ldots$



② **Upper-bound iteration failure probability**

③ **Make the upper bound safe for all possible fault rates**

④ **Upper-bound the control failure rate**

Upper-bounding the iteration failure probabilities

Accounting for

- ➡ all possible error scenarios
- ➡ error propagation and correlation
- ➡ voting protocol

Upper bounds on message error probabilities

$P_1 \geq$ P (msg. is omitted at time t)

$P_2 \geq$ P (msg. is incorrectly computed)

$P_3 \geq$ P (msg. is misses its deadline)

Upper-bounding the iteration failure probabilities

Accounting for
- ➡ all possible error scenarios
- ➡ error propagation and correlation
- ➡ voting protocol

$$V_n \, (P_1, P_2, P_3, \ldots) \geq \mathrm{P} \, (I_n = \mathbf{F})$$

Upper bounds on message error probabilities

$P_1 \geq$ P (msg. is omitted at time t)

$P_2 \geq$ P (msg. is incorrectly computed)

$P_3 \geq$ P (msg. is misses its deadline)

Analysis steps

Peak fault rates

Upper-bound the control failure rate

$V_n\,(P_1, P_2, P_3, \ldots) \geq \mathrm{P}\,(I_n = \mathrm{F})$

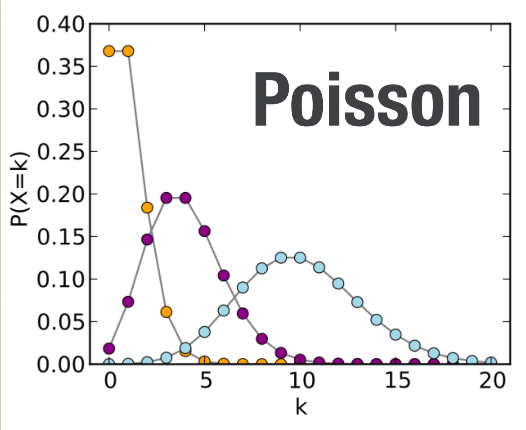
1

2

3

4

Message error probability bounds $P_1, P_2, P_3, \ldots$

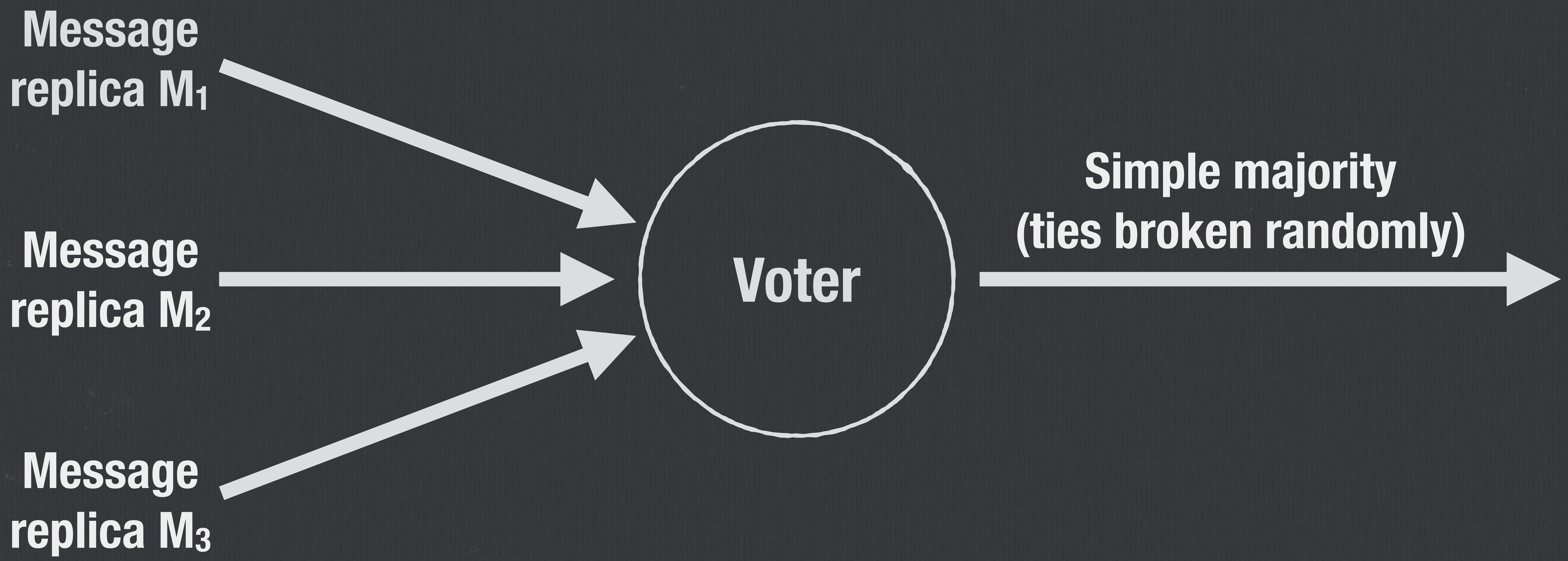


Make the upper bound safe for all possible fault rates

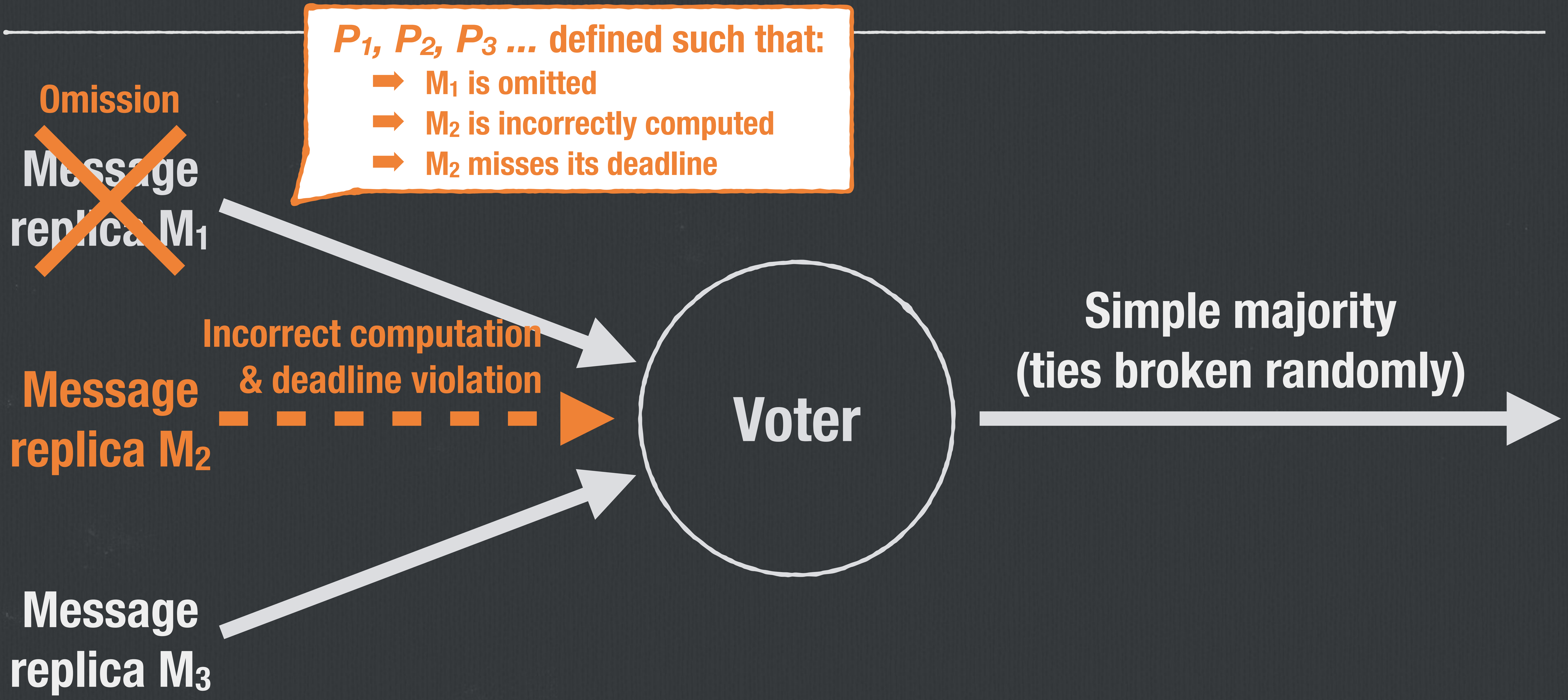
Is the upper bound $V_n(P_1, P_2, P_3, \ldots)$ safe for all possible fault rates?

Let's look at a simple example!

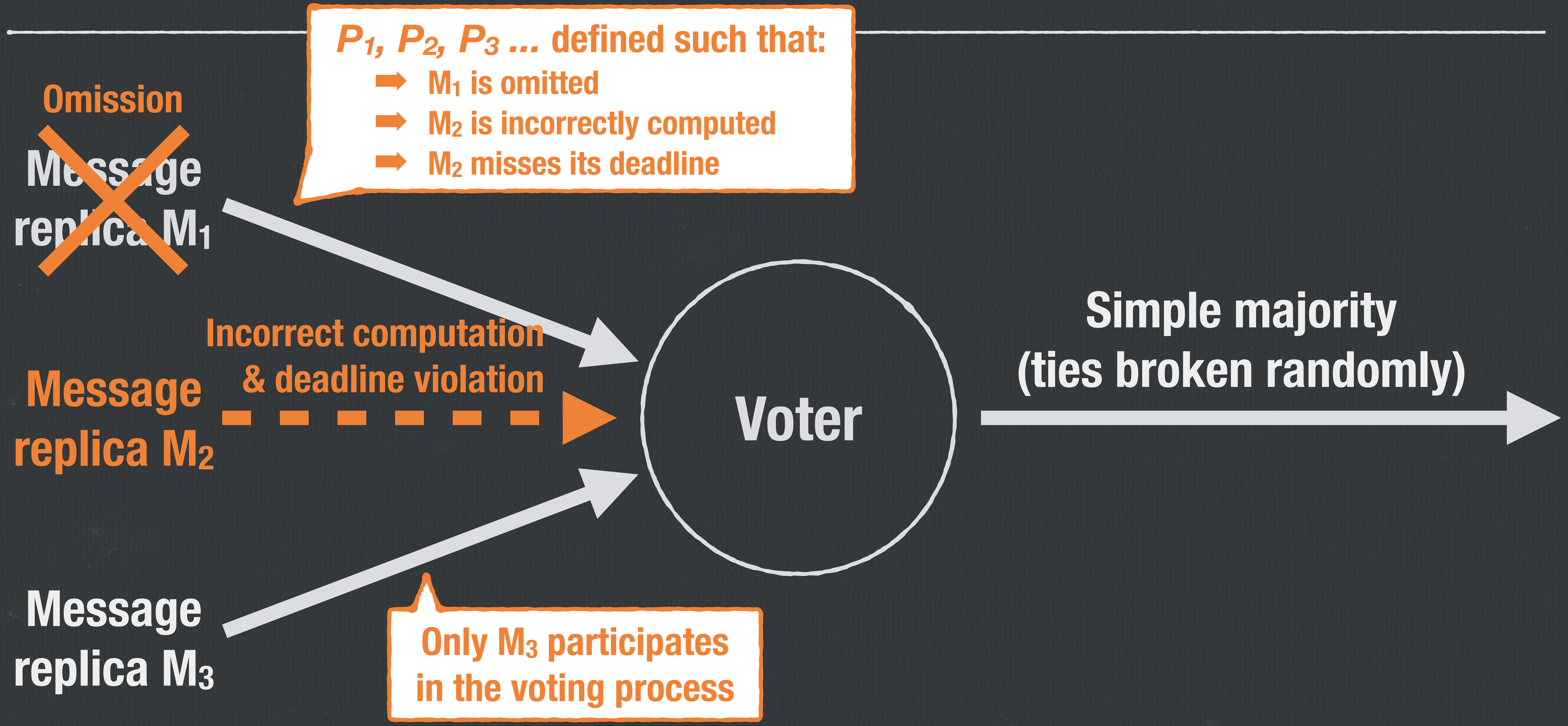
Is the upper bound $V_n\,(P_1, P_2, P_3, \ldots)$ **safe** for all possible fault rates?



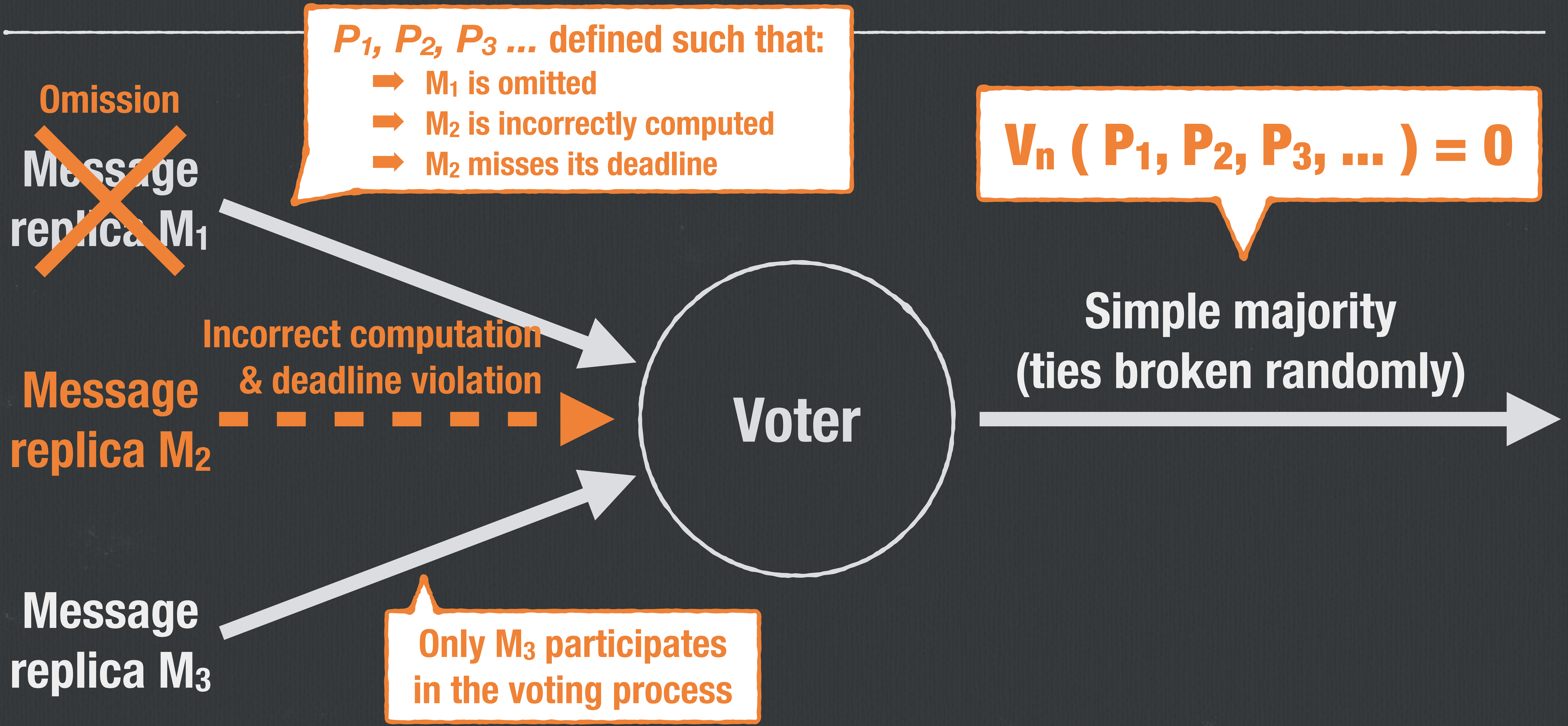
Is the upper bound $V_n(P_1, P_2, P_3, \ldots)$ safe for all possible fault rates?



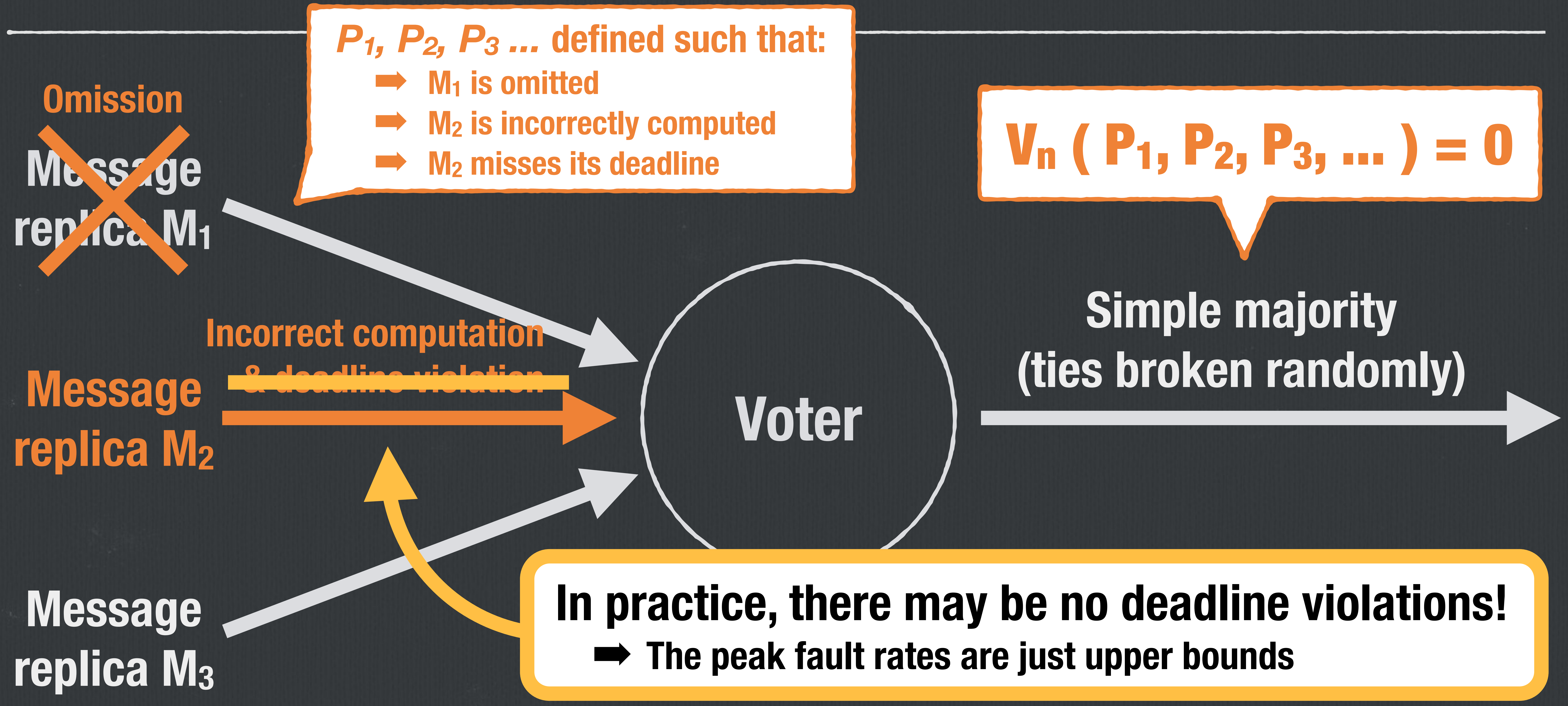
Is the upper bound V_n (P_1, P_2, P_3, …) safe for all possible fault rates?



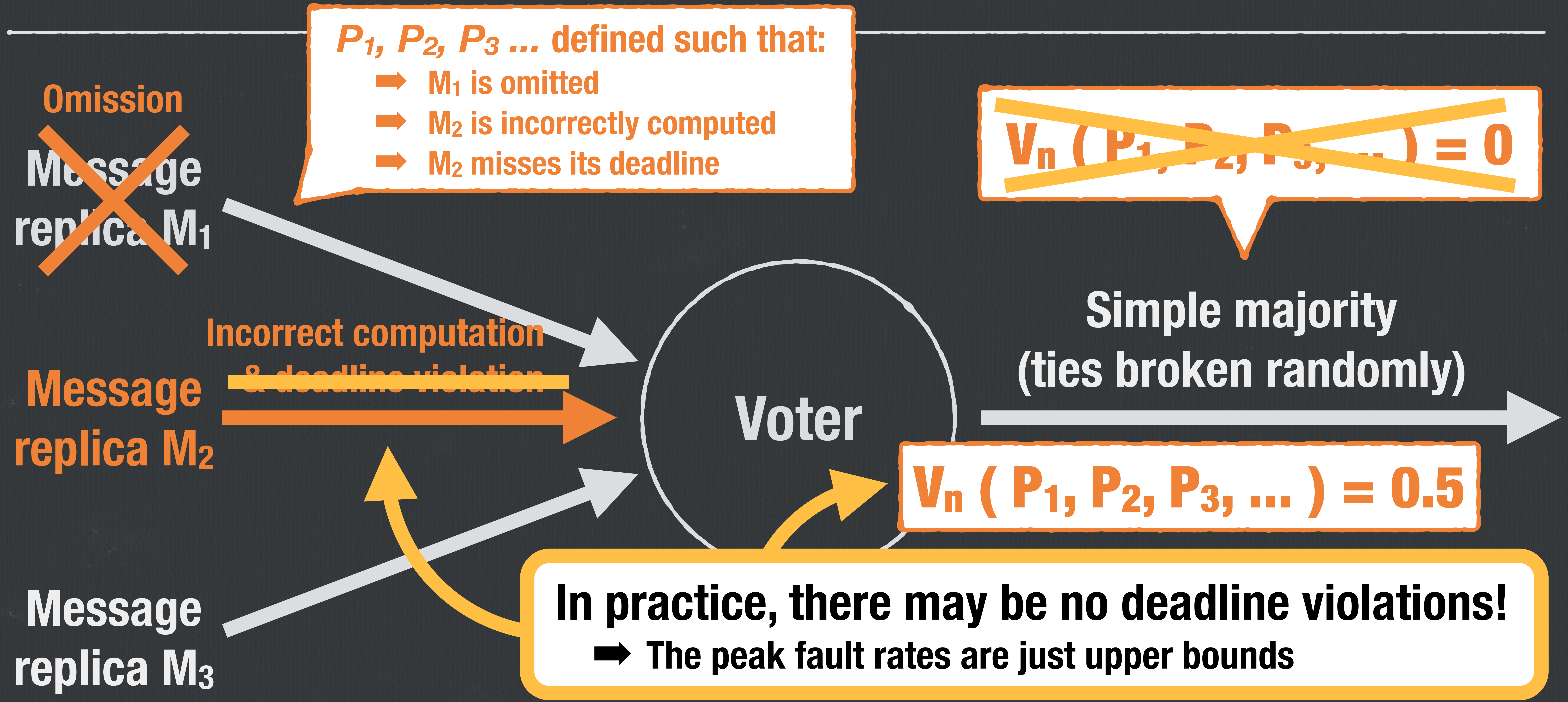
Is the upper bound $V_n (P_1, P_2, P_3, \ldots)$ safe for all possible fault rates?



Is the upper bound $V_n (P_1, P_2, P_3, \ldots)$ safe for all possible fault rates?



Is the upper bound $V_n(P_1, P_2, P_3, \ldots)$ **safe** for all possible fault rates?



Is the upper bound $V_n (P_1, P_2, P_3, \ldots)$ safe for all possible fault rates?

$$\boxed{V_n (P_1, P_2, P_3, \ldots) \geq \mathrm{P} (I_n = \mathbf{F})}$$

Safe if V_n is monotonic in $P_1, P_2, P_3, \ldots$

Is the upper bound $V_n\,(P_1, P_2, P_3, \ldots)$ **safe** for all possible fault rates?

$$\boxed{V_n\,(P_1, P_2, P_3, \ldots) \geq \mathrm{P}\,(I_n = \mathbf{F})}$$

Safe if V_n is monotonic in $P_1, P_2, P_3, \ldots$

+

A fudge factor Δ is added to ensure monotonicity*

*Arpan Gujarati, Mitra Nasri, and Björn B Brandenburg. Quantifying the resiliency of fail-operational real-time networked control systems. Technical Report MPI-SWS2018-005, Max Planck Institute for Software Systems, Germany, 2018. URL: http://www.mpi-sws.org/tr/2018-005.pdf.

Is the upper bound $V_n (P_1, P_2, P_3, \ldots)$ safe for all possible fault rates?

$$\boxed{V_n (P_1, P_2, P_3, \ldots) \geq \mathrm{P} (I_n = \mathbf{F})}$$

Safe if V_n is monotonic in $P_1, P_2, P_3, \ldots$

+

A fudge factor Δ is added to ensure monotonicity*

=

$$\boxed{U_n (P_1, P_2, P_3, \ldots) \geq \mathrm{P} (I_n = \mathbf{F})}$$

*Arpan Gujarati, Mitra Nasri, and Björn B Brandenburg. Quantifying the resiliency of fail-operational real-time networked control systems. Technical Report MPI-SWS2018-005, Max Planck Institute for Software Systems, Germany, 2018. URL: http://www.mpi-sws.org/tr/2018-005.pdf.

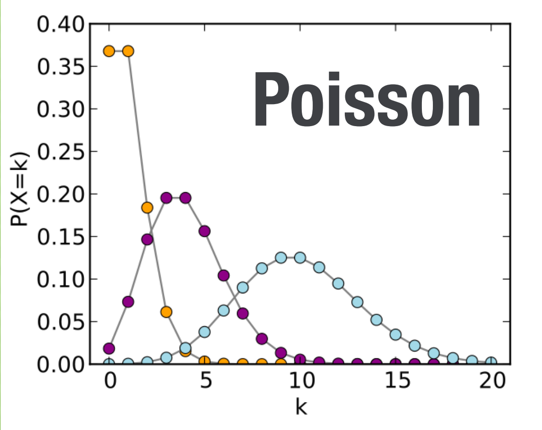
Analysis steps

Peak fault rates

Upper-bound the control failure rate

1

Message error probability bounds $P_1, P_2, P_3, \ldots$



2

$V_n (P_1, P_2, P_3, \ldots) \geq P (I_n = \mathbf{F})$

3

$U_n (P_1, P_2, P_3, \ldots) \geq P (I_n = \mathbf{F})$

Safe

4

Upper-bounding the control failure rate
(Failures-In-Time or FIT)

$U_n\ (\ P_1,\ P_2,\ P_3,\ \ldots\ )$
$\geq \mathrm{P}\ (\ I_n = \mathbf{F}\ )$

Upper-bounding the control failure rate (Failures-In-Time or FIT)

$$U_n(P_1, P_2, P_3, \ldots) \geq \mathbf{P}(I_n = \mathbf{F})$$

FIT (expected # failures in 1 billion hours) $= 10^9 \,/\, MTTF$ (in hours) (Mean Time To first control Failure)

$$= \frac{10^9}{\int_0^\infty t \cdot f(t)\, dt}$$

(probability density function)

Upper-bounding the control failure rate
(Failures-In-Time or FIT)

$U_n(P_1, P_2, P_3, \ldots) \geq \mathbf{P}(I_n = \mathbf{F})$

$$\textit{FIT} = 10^9 \,/\, \textit{MTTF} \text{ (in hours)}$$

(expected # failures in 1 billion hours)

(**M**ean **T**ime **T**o first control **F**ailure)

$$= \frac{10^9}{\int_0^\infty t \cdot f(t)\, dt}$$

(probability density function)

(probability density function) $f(t)$ = P (first control failure at time t)

= P (first violation of (2, 3)-firm constraint at time t)

= P (first instance of **FSF** | **FFS** | **SFF** | **FF** at time t)

Upper-bounding the control failure rate
(Failures-In-Time or FIT)

$U_n (P_1, P_2, P_3, \ldots) \geq P (I_n = F)$

Using prior work*

$FIT = 10^9 \,/\, MTTF$ (in hours)

(expected # failures in 1 billion hours)

(Mean Time To first control Failure)

$$= \frac{10^9}{\int_0^\infty t \cdot f(t)\, dt}$$

(probability density function)

(probability density function)

$f(t)$ = P (first control failure at time t)
= P (first violation of (2, 3)-firm constraint at time t)
= P (first instance of FSF | FFS | SFF | FF at time t)

*M. Sfakianakis, S. Kounias, and A. Hillaris. "Reliability of a consecutive k-out-of-r-from-n: F system." IEEE Transactions on Reliability 41, no. 3 (1992): 442-447.

Upper-bounding the control failure rate (Failures-In-Time or FIT)

U_n (P_1, P_2, P_3, …) $\geq$ P (I_n = F)

Using prior work*

Scalable and numerical, but sound, analysis

$$FIT = 10^9 \,/\, MTTF \text{ (in hours)}$$

(expected # failures in 1 billion hours)

(Mean Time To first control Failure)

$$= \frac{10^9}{\int_0^{\infty} t \cdot f(t)\, dt}$$

(probability density function)

(probability density function)

$f(t)$ = P (first control failure at time t)

= P (first violation of (2, 3)-firm constraint at time t)

= P (first instance of FSF | FFS | SFF | FF at time t)

*M. Sfakianakis, S. Kounias, and A. Hillaris. "Reliability of a consecutive k-out-of-r-from-n: F system." IEEE Transactions on Reliability 41, no. 3 (1992): 442-447.

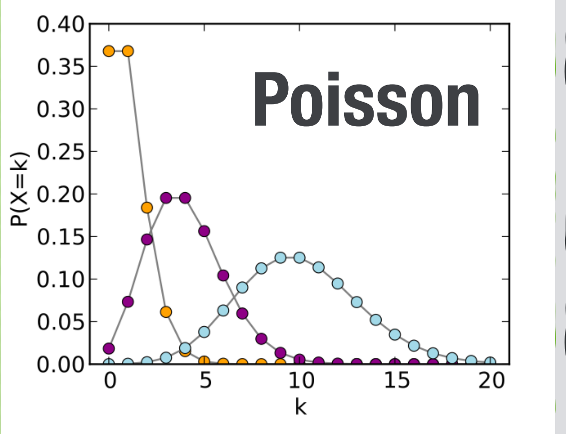
Analysis steps

Peak fault rates

FIT_{UB} for a single control loop

1

Message error probability bounds $P_1, P_2, P_3, \ldots$



2

$V_n\,(P_1, P_2, P_3, \ldots) \geq P\,(I_n = F)$

3

$U_n\,(P_1, P_2, P_3, \ldots) \geq P\,(I_n = F)$

Safe

4

Analysis steps

Peak fault rates

FIT_{UB} for a single control loop

FIT_{UB} for L_1

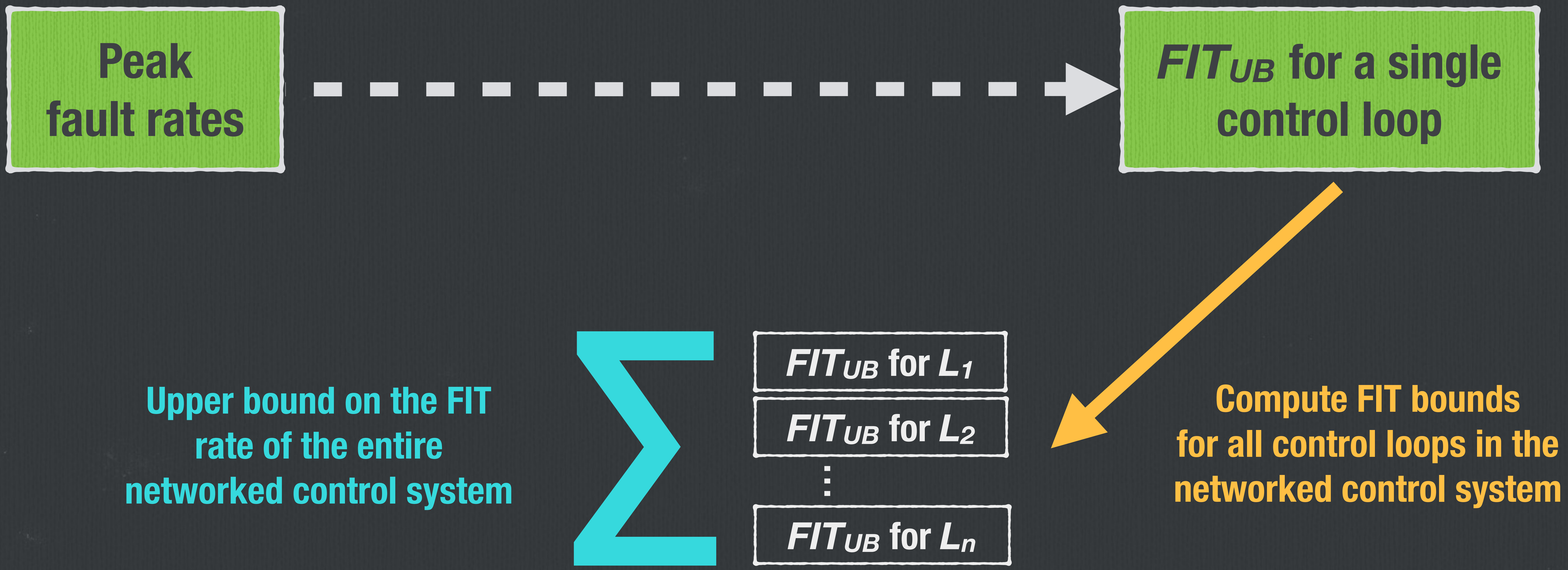
FIT_{UB} for L_2

⋮

FIT_{UB} for L_n

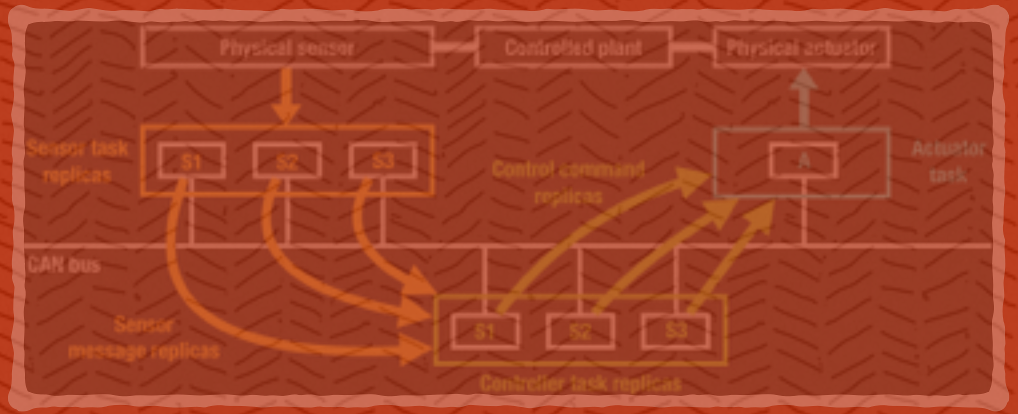
Compute FIT bounds for all control loops in the networked control system

Analysis steps



Outline

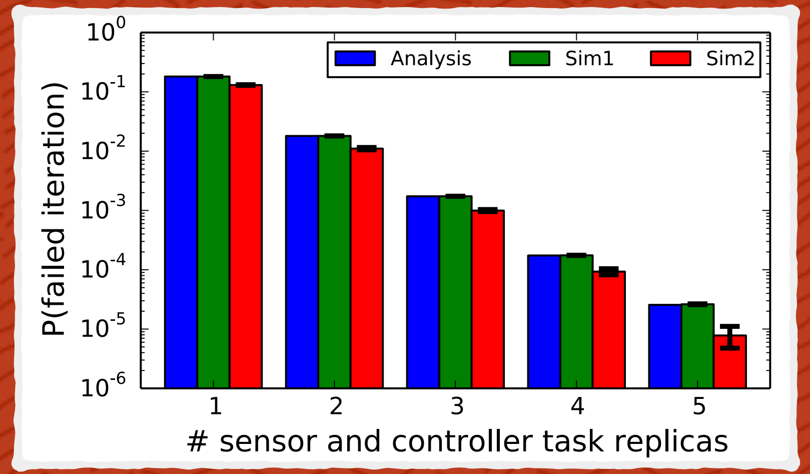
Analysis of a Controller Area Network (CAN) based networked control system



System Model

$$\int_0^\infty t \cdot f(t)\, dt$$

Analysis



Evaluation

Evaluation overview

Evaluation overview

- How accurate is the analysis?
  - Comparison with simulation results

Evaluation overview

- ☐ How accurate is the analysis?
  - ➡ Comparison with simulation results
- ☐ Case study: FIT vs. (m, k) constraints vs. replication schemes

CAN-based active suspension workload*

- Four control loops L_1, L_2, L_3, L_4
  - ➡ to control the four wheels with magnetic suspension

Messages	Length	Period (ms)	Deadline (ms)	Priority
Clock sync.	1	50	50	High
Current mon.	1	4	4	
Temperature	1	10	10	
L_1 messages	3	1,75	1,75	
L_2 messages	3	1,75	1,75	
L_3 messages	3	1,75	1,75	
L_4 messages	3	1,75	1,75	
Logging	8	100	100	Low

**Adolfo Anta and Paulo Tabuada. On the benefits of relaxing the periodicity assumption for networked control systems over CAN. In Proceedings of the 30th Real-Time Systems Symposium, pages 3–12. IEEE, 2009.*

CAN-based active suspension workload*

- Four control loops L_1, L_2, L_3, L_4
  - ➡ to control the four wheels with magnetic suspension

This talk: Control loop L_1's tasks were replicated

Messages	Length	Period (ms)	Deadline (ms)	Priority
Clock sync.	1	50	50	High
Current mon.	1	4	4	
Temperature	1	10	10	
L_1 messages	3	1,75	1,75	
L_2 messages	3	1,75	1,75	
L_3 messages	3	1,75	1,75	
L_4 messages	3	1,75	1,75	
Logging	8	100	100	Low

Adolfo Anta and Paulo Tabuada. On the benefits of relaxing the periodicity assumption for networked control systems over CAN. In Proceedings of the 30th Real-Time Systems Symposium, pages 3–12. IEEE, 2009.

CAN-based active suspension workload*

- Four control loops L_1, L_2, L_3, L_4
  - ➡ to control the four wheels with magnetic suspension

This talk: Control loop L_1's tasks were replicated

In the paper: Experiments with all replica schemes

Messages	Length	Period (ms)	Deadline (ms)	Priority
Clock sync.	1	50	50	High
Current mon.	1	4	4	
Temperature	1	10	10	
L_1 messages	3	1,75	1,75	
L_2 messages	3	1,75	1,75	
L_3 messages	3	1,75	1,75	
L_4 messages	3	1,75	1,75	
Logging	8	100	100	Low

*Adolfo Anta and Paulo Tabuada. On the benefits of relaxing the periodicity assumption for networked control systems over CAN. In Proceedings of the 30th Real-Time Systems Symposium, pages 3–12. IEEE, 2009.

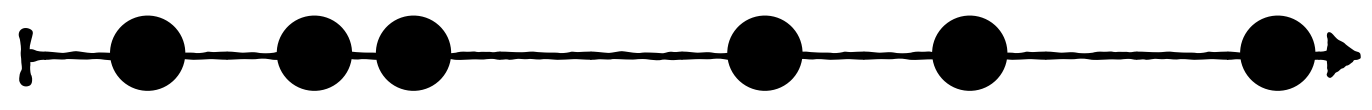
How accurate is the analysis?

Iteration failure probability bound

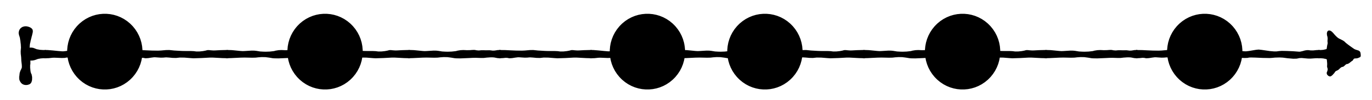
$$\boxed{U_n (P_1, P_2, P_3, \ldots) \geq \mathrm{P} (I_n = \mathrm{F})}$$

Discrete event simulation of a CAN-based system

Poisson process for CAN bus faults



Poisson process for faults on Host 1



... and so on

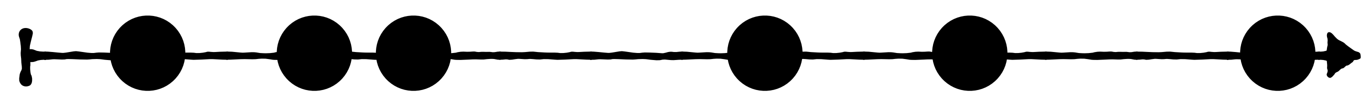
How **accurate** is the analysis?

Iteration failure probability bound

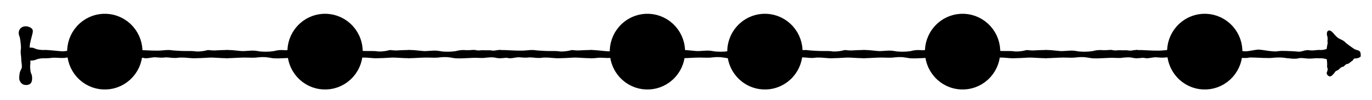
$$U_n (P_1, P_2, P_3, \ldots) \geq \mathrm{P} (I_n = \mathbf{F})$$

Discrete event simulation of a CAN-based system

Poisson process for CAN bus faults



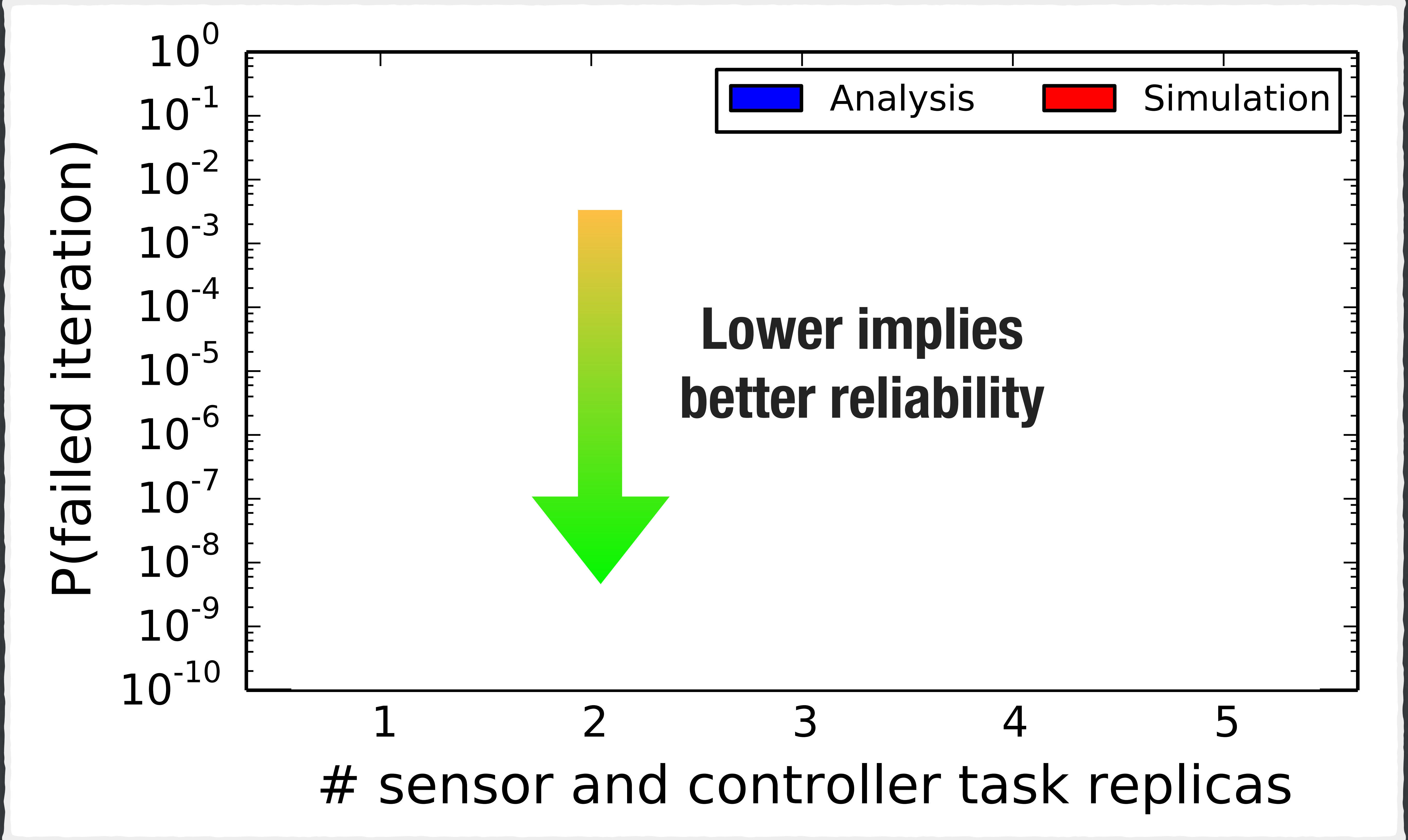
Poisson process for faults on Host 1



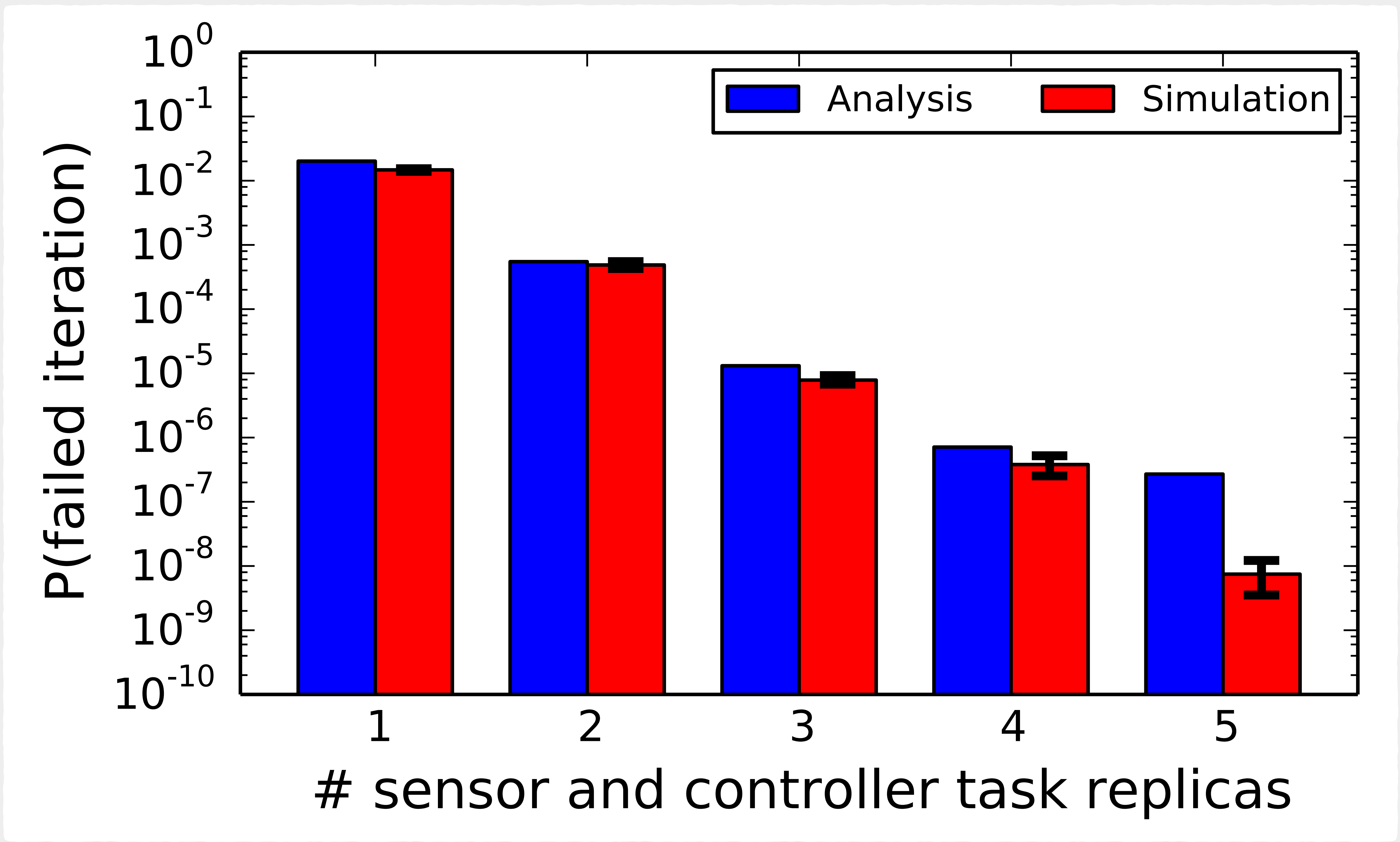
... and so on

Simulation is not safe

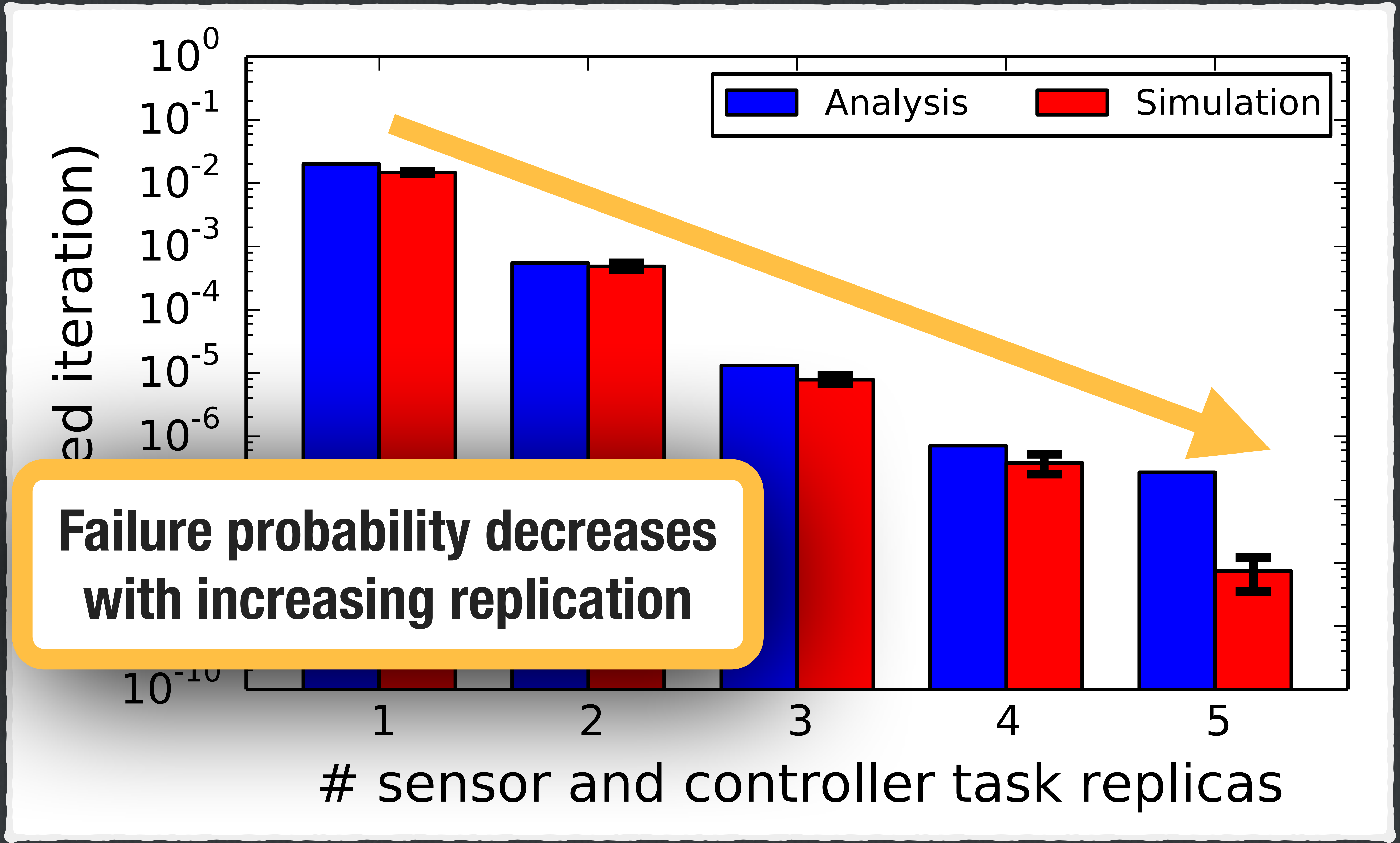
Analysis versus simulation



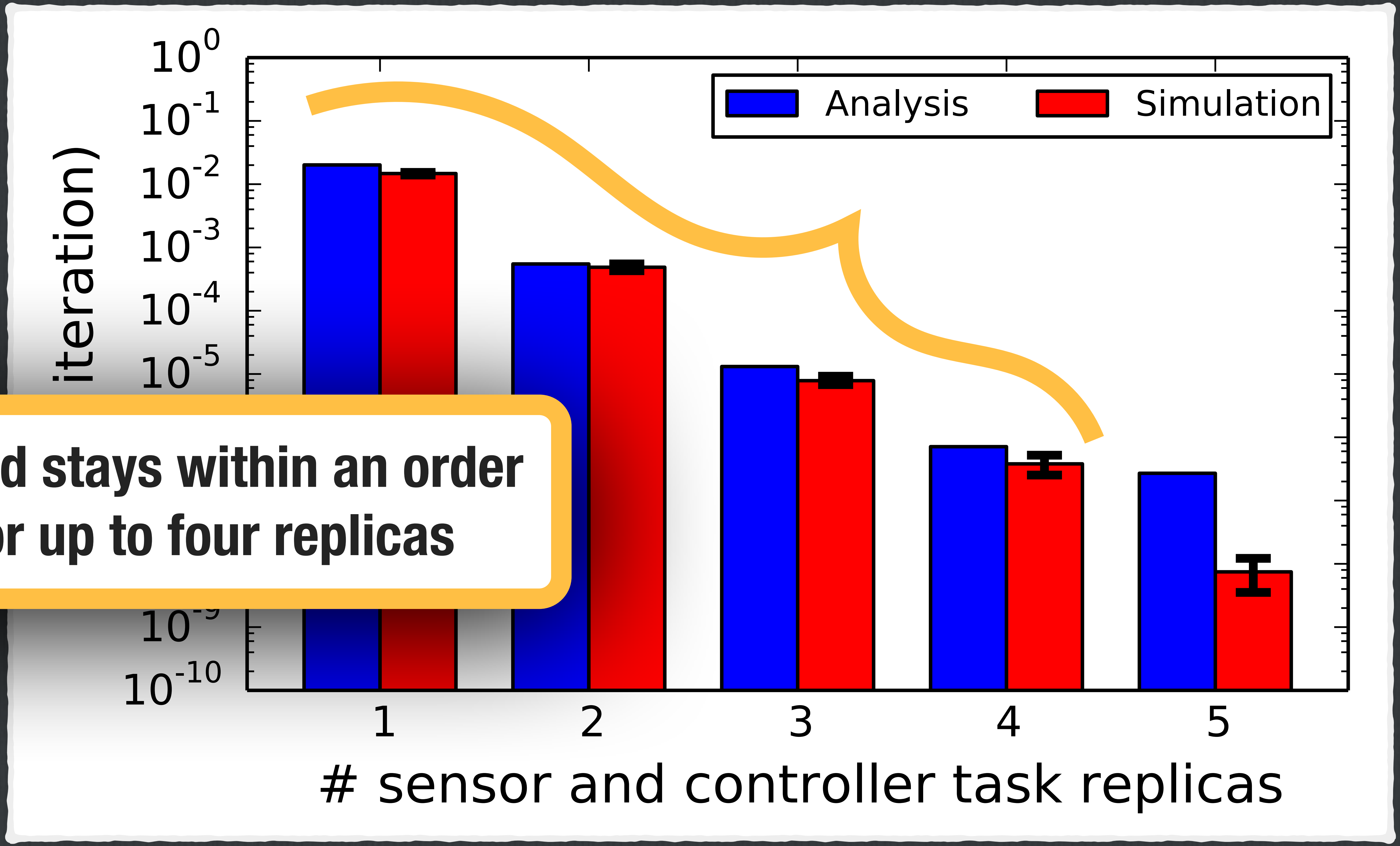
Analysis versus simulation



Analysis versus simulation

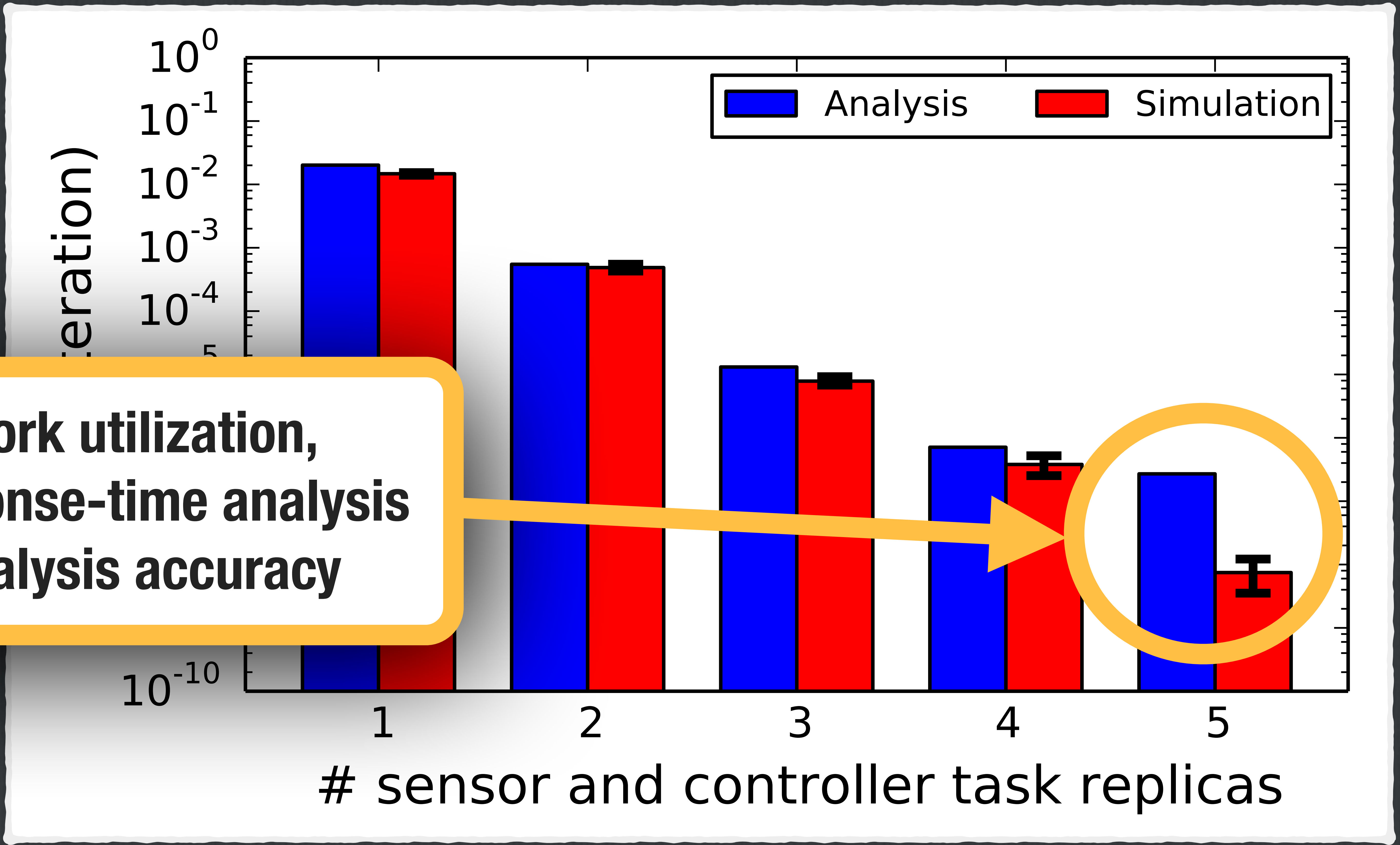


Analysis versus simulation



Pessimism incurred stays within an order of magnitude for up to four replicas

Analysis versus simulation



At high network utilization, worst-case response-time analysis affects the analysis accuracy

Case study

Case study

- FIT analysis for different (m, k)-firm constraints
  - ➡ (9, 100) ~ 9%
  - ➡ (19, 20) ~ 95%
  - ➡ (99, 100) ~ 99%
  - ➡ (9999, 10000) ~ 99.99%

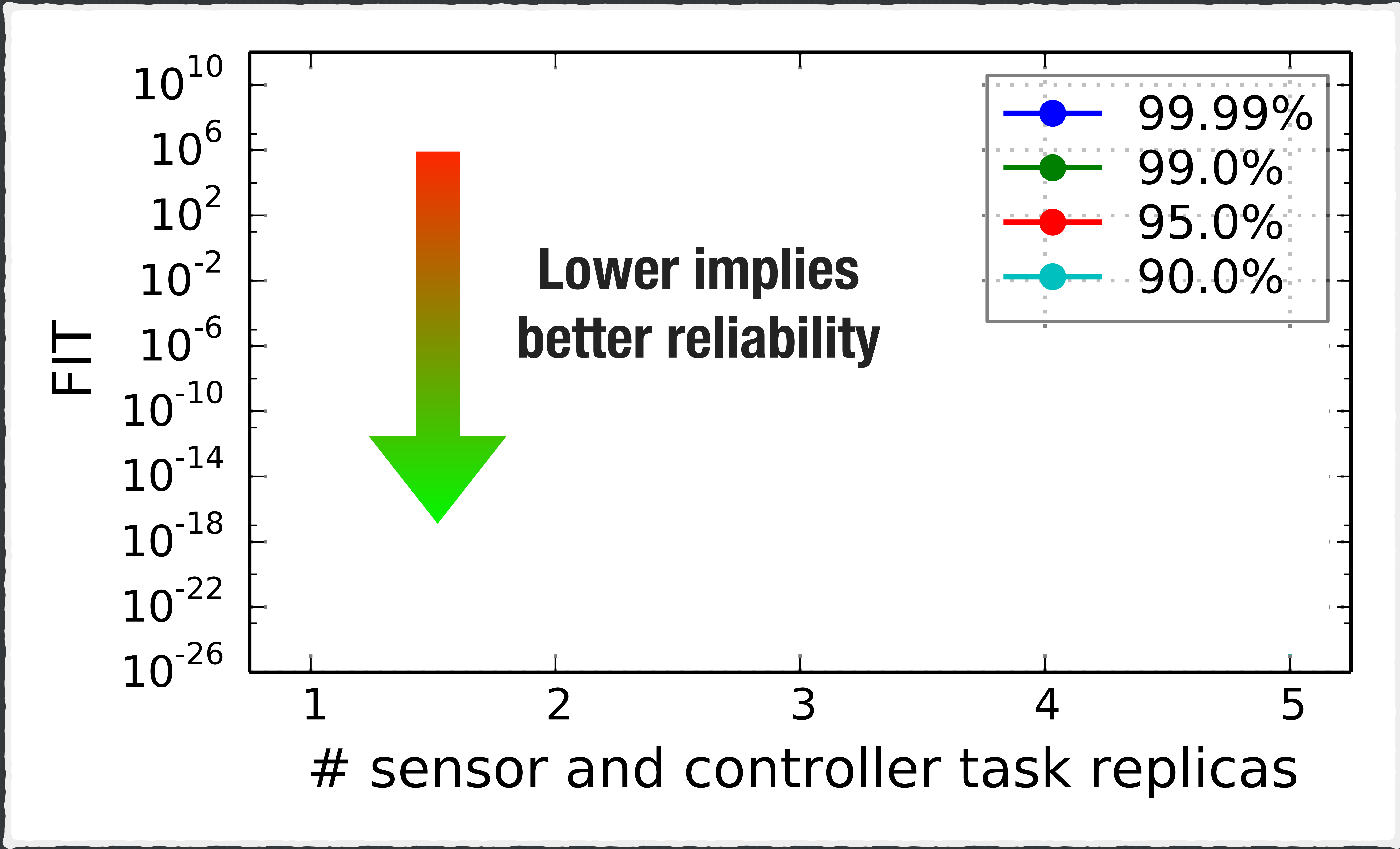
Case study

- FIT analysis for different (m, k)-firm constraints
  - (9, 100) ~ 9%
  - (19, 20) ~ 95%
  - (99, 100) ~ 99%
  - (9999, 10000) ~ 99.99%
- Replication factor of loop L_1's tasks varied from 1 to 5

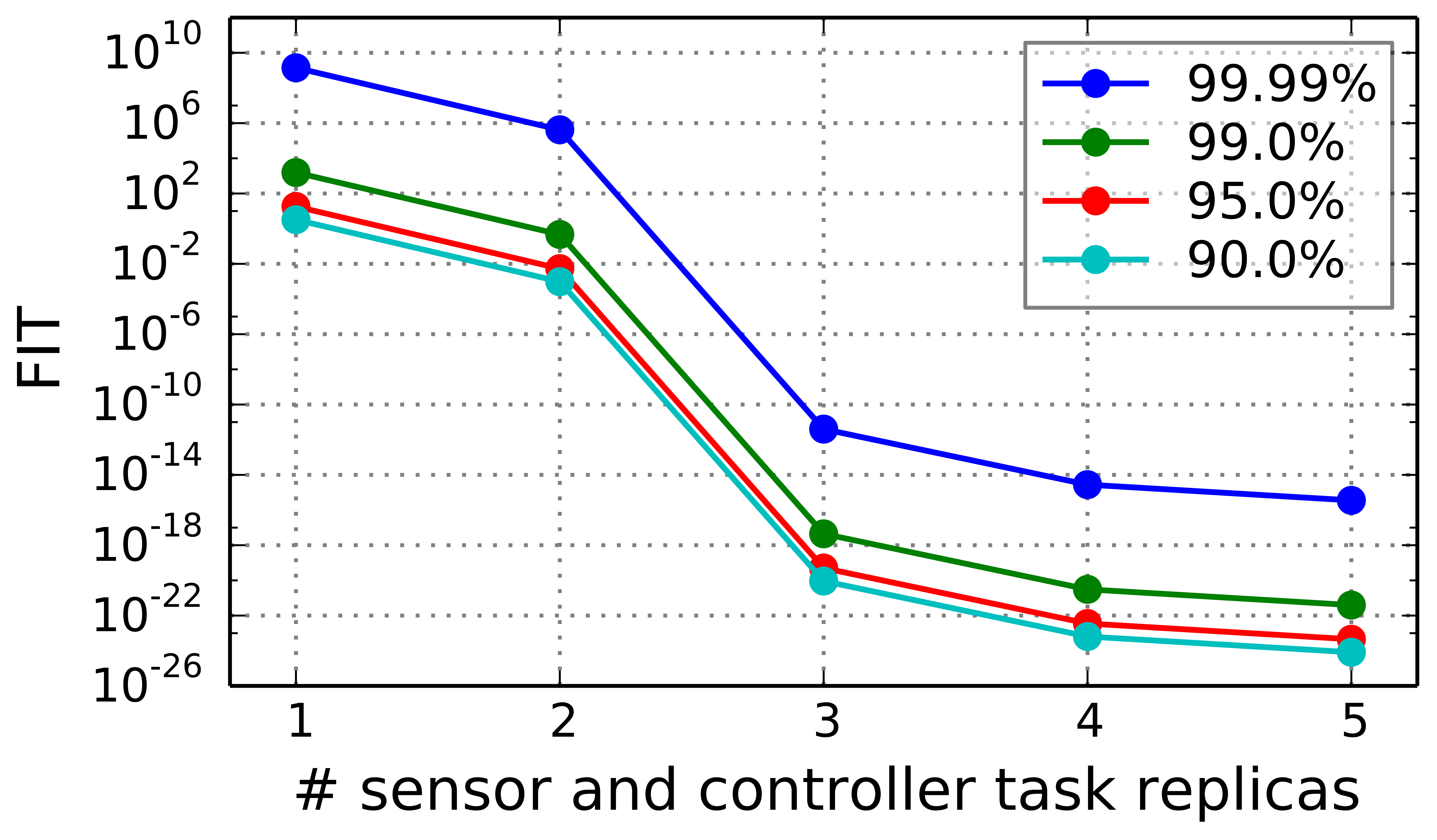
Case study

- FIT analysis for different (m, k)-firm constraints
  - ➡ (9, 100) ~ 9%
  - ➡ (19, 20) ~ 95%
  - ➡ (99, 100) ~ 99%
  - ➡ (9999, 10000) ~ 99.99%
- Replication factor of loop L_1's tasks varied from 1 to 5
- What should be the replication factor to achieve FIT under 10^{-6}?

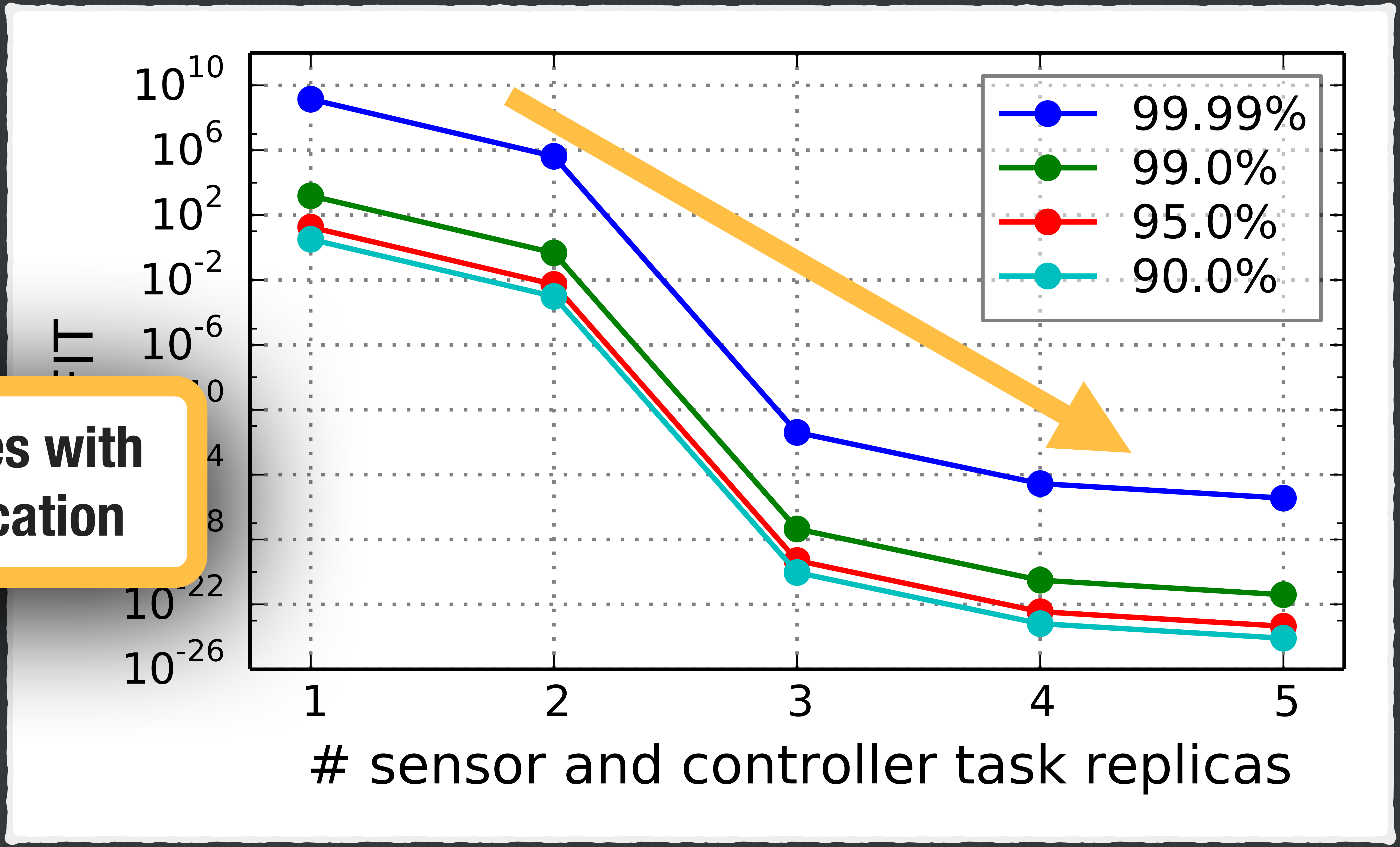
FIT rate vs. replication factor vs. (m, k) parameters



FIT rate vs. replication factor vs. (m, k) parameters

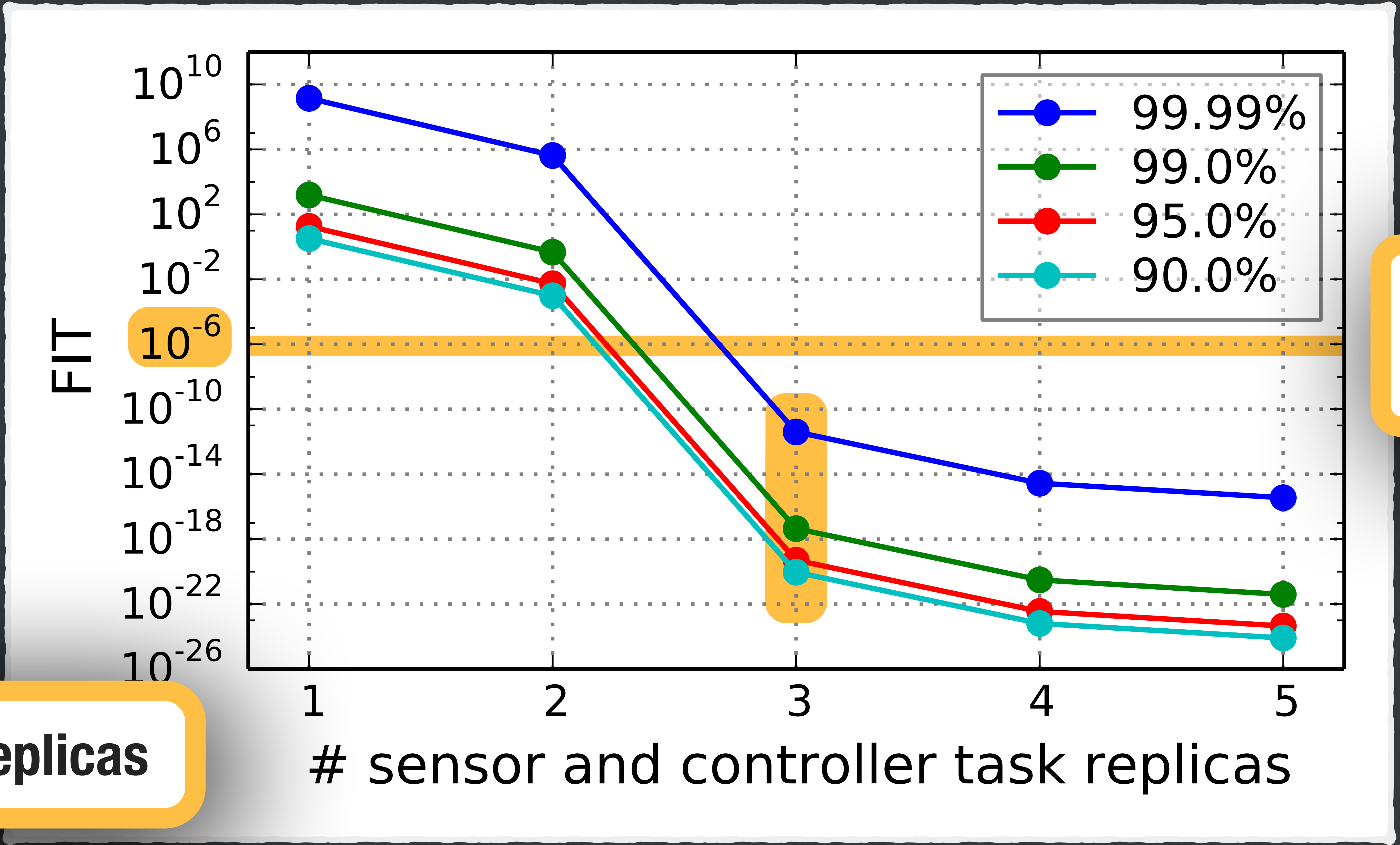


FIT rate vs. replication factor vs. (m, k) parameters



FIT rate decreases with increasing replication

FIT rate vs. replication factor vs. (m, k) parameters



If the desired FIT rate is under 10^{-6}

Prefer three replicas

Summary

Summary

- ☐ A safe Failures-In-Time (FIT) analysis for networked control systems
  - ➡ CAN-based networked control system model

Summary

- A safe Failures-In-Time (FIT) analysis for networked control systems
  - ➡ CAN-based networked control system model
- Focus on failures and errors due to transient faults
  - ➡ omission errors
  - ➡ incorrect computation errors
  - ➡ transmission errors

Summary

- A safe Failures-In-Time (FIT) analysis for networked control systems
  - ➡ CAN-based networked control system model

- Focus on failures and errors due to transient faults
  - ➡ omission errors
  - ➡ incorrect computation errors
  - ➡ transmission errors

- ... and on robust systems that can tolerate a few iteration failures
  - ➡ (m,k)-firm model for control failure

Summary

- ☐ A safe Failures-In-Time (FIT) analysis for networked control systems
  - ➡ CAN-based networked control system model

- ☐ Focus on failures and errors due to transient faults
  - ➡ omission errors
  - ➡ incorrect computation errors
  - ➡ transmission errors

Future work: Byzantine errors + BFT protocols

- ☐ ... and on robust systems that can tolerate a few iteration failures
  - ➡ (m,k)-firm model for control failure

Summary

- ☐ A safe Failures-In-Time (FIT) analysis for networked control systems
  - ➡ CAN-based networked control system model
- ☐ Focus on failures and errors due to transient faults
  - ➡ omission errors
  - ➡ incorrect computation errors
  - ➡ transmission errors

Future work: Byzantine errors + BFT protocols

- ☐ ... and on robust systems that can tolerate a few iteration failures
  - ➡ (m,k)-firm model for control failure

Accounting for other robustness criteria