Exploring Weakly-Hard Paradigm for Networked Systems

C. Huang, K. Wardega*, W. Li & Q. Zhu

Northwestern BOSTON UNIVERSITY

DESTION'19 - Montreal, QC.

Background

Timing and Failure

The design of systems that work

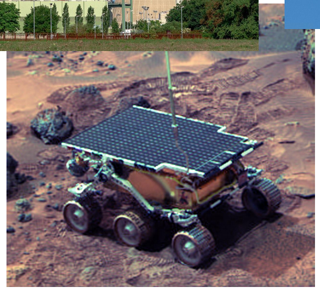
Hard Real-Time Model

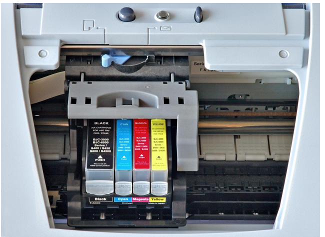
WCET Analysis	Deadlines	Scheduling
Characterize the system. How bad can it get, exactly?	Give each task a deadline.	Plan everything. Power, resources, period.
Can I even do this? *Is it just hard? Or impossible?*		

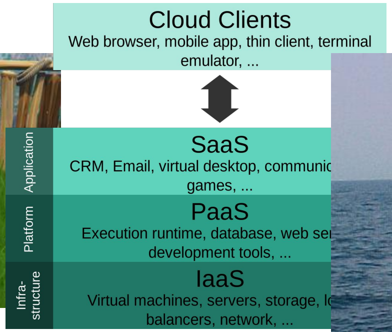
Firm/Soft Real-Time Model

pWCET* Analysis	Distributions*	Scheduling
Characterize the system. How bad can it get, probably*?	Assign each task a distribution*.	Plan everything. Power, resources, period. *Is this useful for my application?*

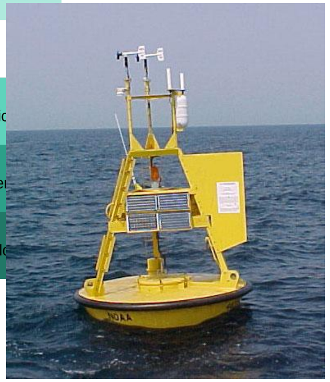
Timing Constraints vs Guarantees







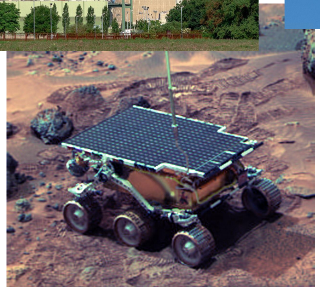
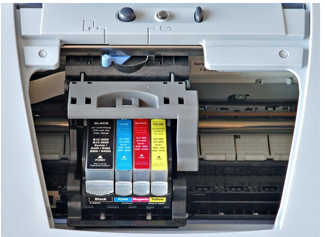
NETFLIX

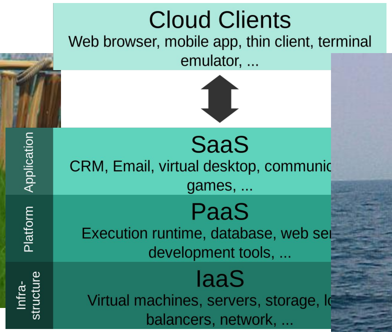


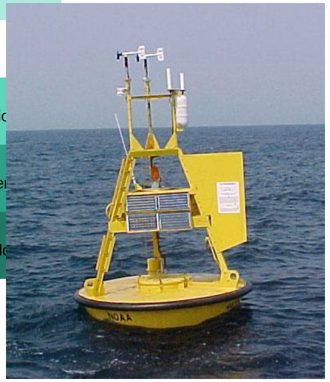
Hard **Firm** **Soft**

Timing Constraints vs Guarantees





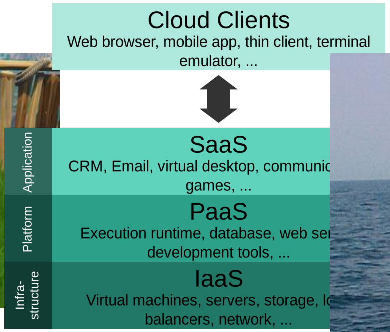
NETFLIX

Hard **Firm** **Soft**

Trace guarantee
(deterministic)

Set of traces guarantee
(probabilistic)

Timing Constraints vs Guarantees



NETFLIX

Hard | **Firm** | **Soft**

?

Trace guarantee (deterministic)

Trace guarantee (deterministic)

Set of traces guarantee (probabilistic)

Can I get trace guarantees if computing WCETs is hard?

How about introducing *bounded non-determinism?*

		Meet	Consecutive	Meaning
1	$\langle m,K \rangle$	✓	✗	Meet at least *m* deadlines of every *K*
2	$\langle\langle m,K \rangle\rangle$	✓	✓	Meet at least *m* consecutive deadlines in every *K*
3	(m,K)	✗	✗	Miss no more than *m* deadlines of every *K*
4	$((m,K))$	✗	✓	Miss no more than *m* consecutive deadlines in every *K*

		Meet	Consecutive	Meaning
1	$\langle m,K \rangle$	✓	✗	Meet at least *m* deadlines of every *K*
2	$\langle\langle m,K \rangle\rangle$	✓	✓	Meet at least *m* consecutive deadlines in every *K*
3	(m,K)	✗	✗	Miss no more than *m* deadlines of every *K*
4	$((m,K))$	✗	✓	Miss no more than *m* consecutive deadlines in every *K*

Weakly-Hard Real-Time Model

“Analysis”	“Deadlines”	Scheduling
Characterize the system. pWCET or (*m*,*K*)-type constraints.	Give each task a deadline. Deadline misses are bounded.	Plan everything. Power, resources, period.
Easier than analysis during design of a Hard Real-Time System.		*Traded probabilities for bounded non-determinism.* ***Potentially higher performance! Still get trace guarantees!***

Applications of the Weakly-Hard Model

2001

Weakly-Hard Constraints

Methodology introduced.
(Bernat, IEEE Trans. Comp.)

2002

Controller Area Networks

Weakly-Hard vs
overly-pessimistic Hard
real-time. (Broster, ECRTS)

2005

Energy-Constrained RTS

Maximize performance within
energy limits under Weakly-Hard
constraints. (AlEnawy, RTSS)

2008

Multimedia

Scheduling for a multimedia
service under overload
conditions. (Wu, UMEDIA)

2015

RT Linear Control

Reachability analysis via
bounded time safety
verification. (Duggirala, RTSS)

2019

RT Nonlinear Control

Sufficient conditions for safety
analysis of weakly-hard control.
(Huang, HSCC)

Applications of the Weakly-Hard Model

Embedded Systems

2001

Weakly-Hard Constraints

Methodology introduced.
(Bernat, IEEE Trans. Comp.)

2002

Controller Area Networks

Weakly-Hard vs
overly-pessimistic Hard
real-time. (Broster, ECRTS)

2005

Energy-Constrained RTS

Maximize performance within
energy limits under Weakly-Hard
constraints. (AlEnawy, RTSS)

2008

Multimedia

Scheduling for a multimedia
service under overload
conditions. (Wu, UMEDIA)

2015

RT Linear Control

Reachability analysis via
bounded time safety
verification. (Duggirala, RTSS)

2019

RT Nonlinear Control

Sufficient conditions for safety
analysis of weakly-hard control.
(Huang, HSCC)

Applications of the Weakly-Hard Model

Not an Embedded System

20??

Networked systems

2001

Weakly-Hard Constraints

Methodology introduced. (Bernat, IEEE Trans. Comp.)

2002

Controller Area Networks

Weakly-Hard vs overly-pessimistic Hard real-time. (Broster, ECRTS)

2005

Energy-Constrained RTS

Maximize performance within energy limits under Weakly-Hard constraints. (AlEnawy, RTSS)

2008

Multimedia

Scheduling for a multimedia service under overload conditions. (Wu, UMEDIA)

2015

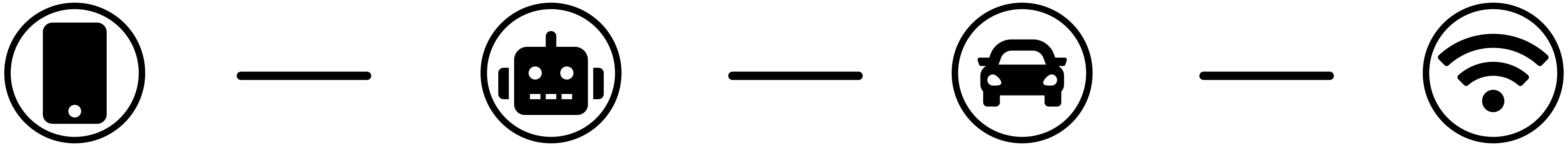
RT Linear Control

Reachability analysis via bounded time safety verification. (Duggirala, RTSS)

2019

RT Nonlinear Control

Sufficient conditions for safety analysis of weakly-hard control. (Huang, HSCC)



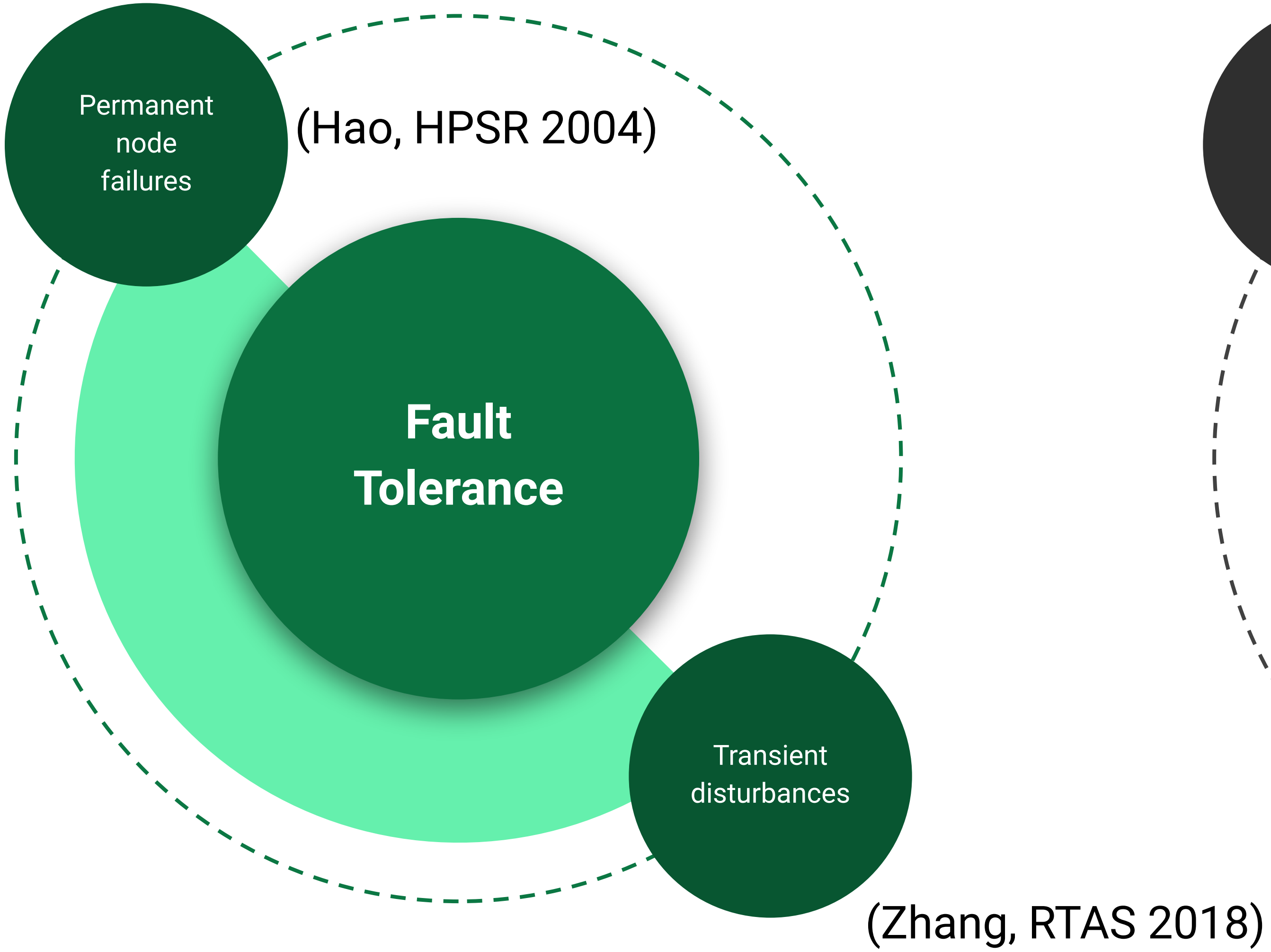
Mobile Cloud **Cloud Robotics** **Connected Vehicles** **IoT**

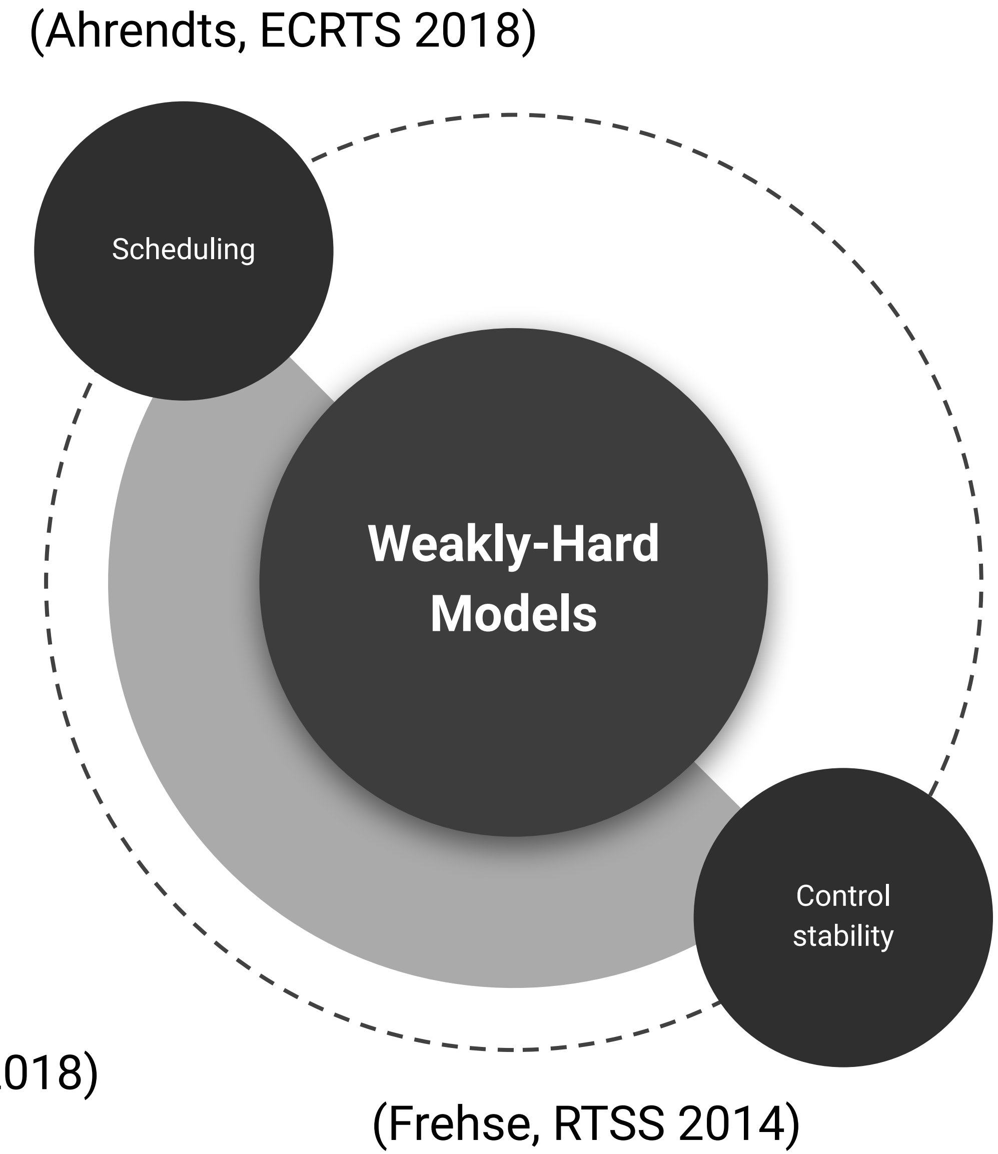
QoS Control Consensus Reliability

Flooding Reachability Stability

WCET-based analysis of networked systems is likely impossible.

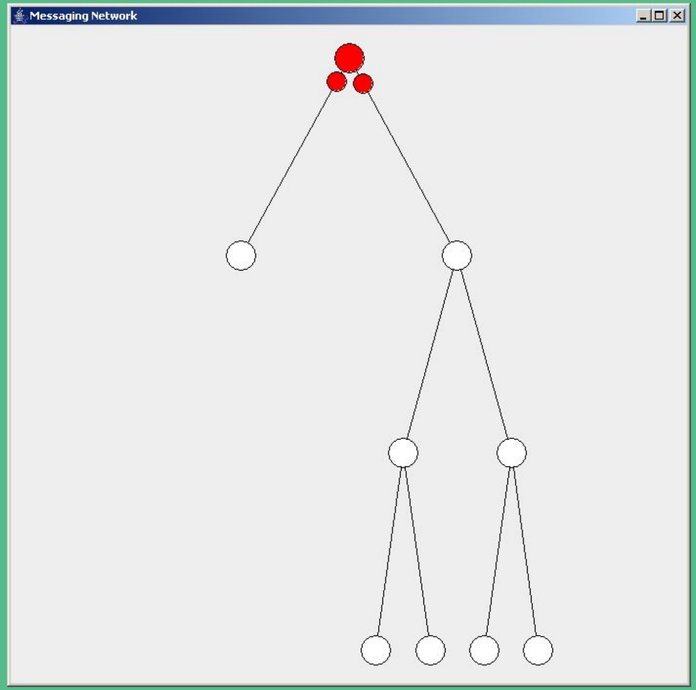
Related Work





Network Flooding

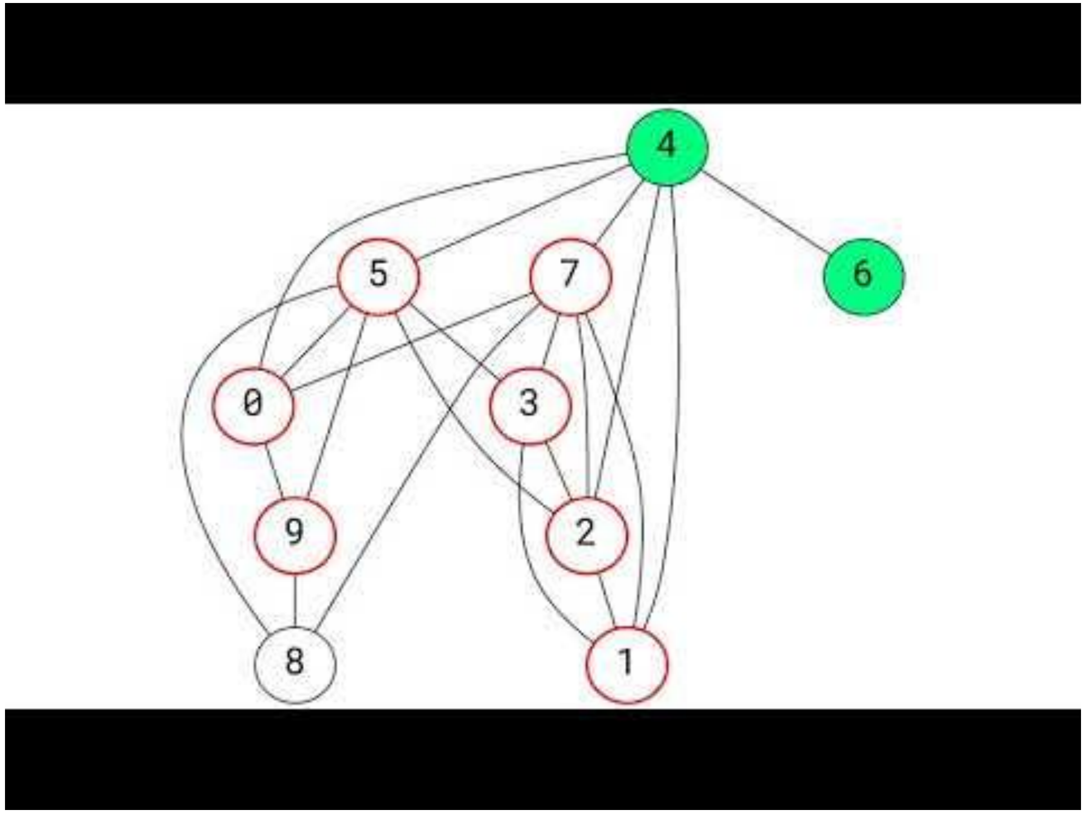
Capturing Node or Link Failures

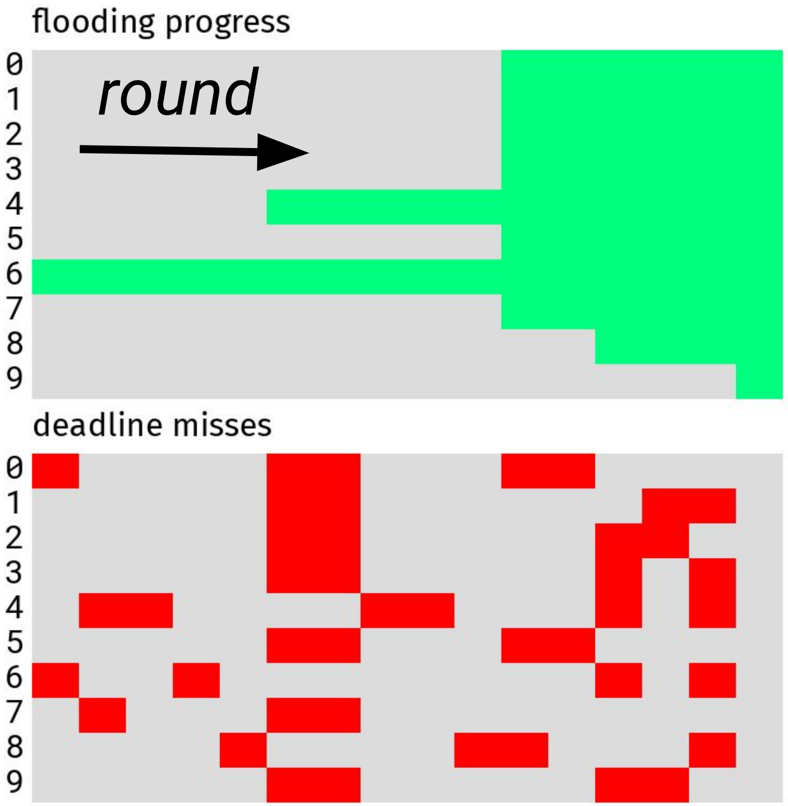


Florian Lindner

Glossy Low-Power Wireless Bus (Ferrari, IPSN’11)

Worst-case Flooding Latency





$(m, K) = (2, 5)$ The flood is initialized by node #6

Flooding Specification

INIT, FLOOD

+

Synchronous Updates

EVOLVE, PERSIST

+

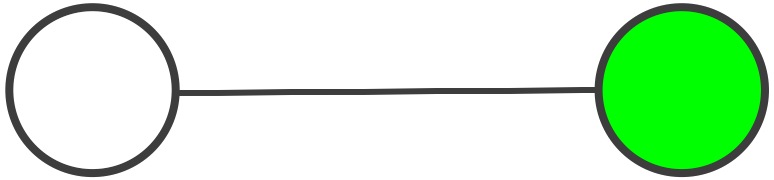
Weakly-Hard Constraints

(m,K) on every node

Throw it to the SMT solver

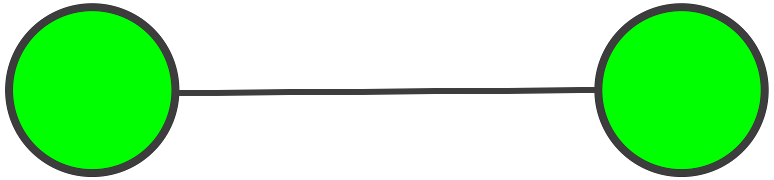
and iterate over the finite horizon

Example: the EVOLVE Constraint



Node *i* **Node *j***

Time *t*, *i* and *j* on, *j* has the packet

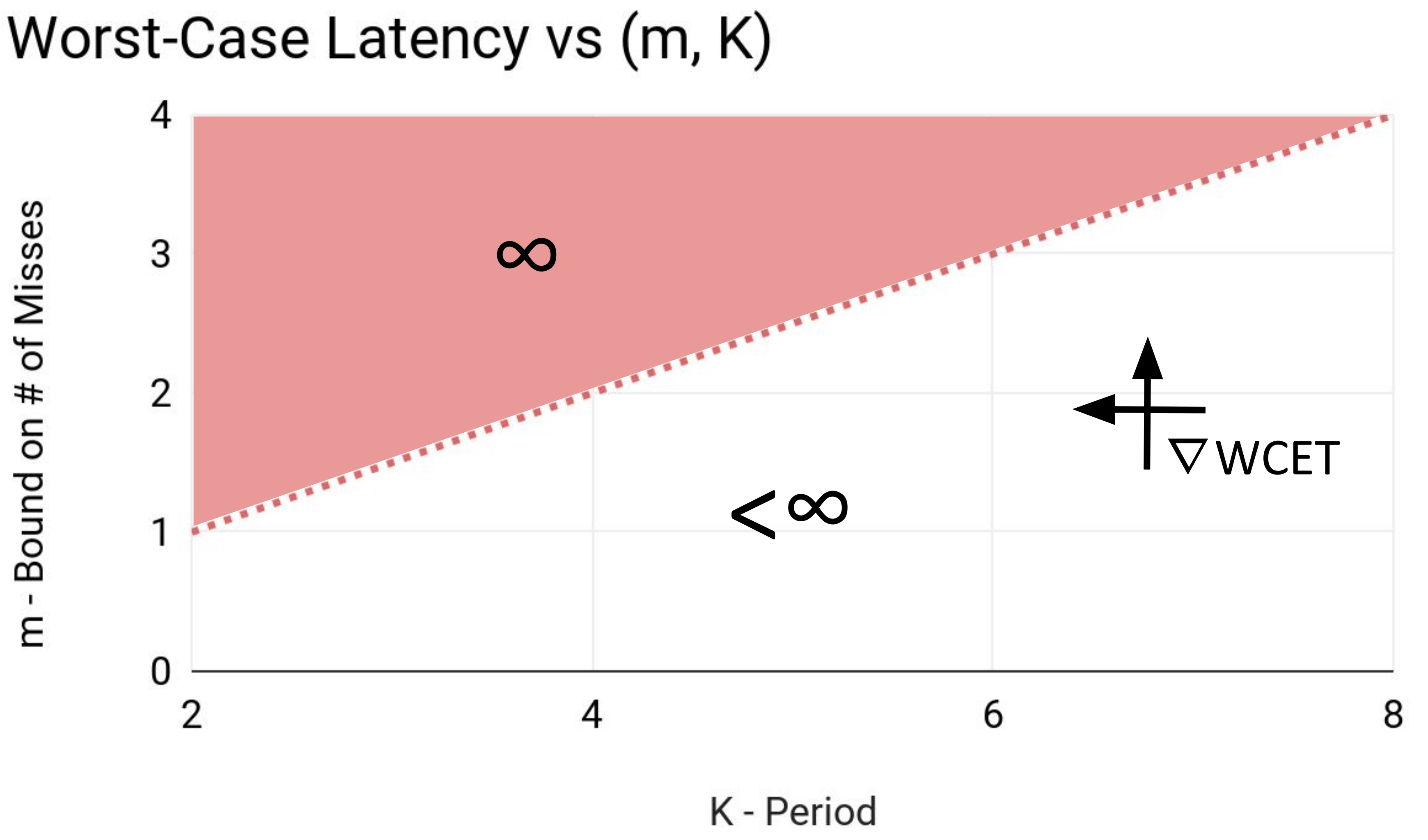


Node *i* **Node *j***

Time *t+1*, *i* and *j* on, *i* and *j* have the packet

$$\bigwedge_{i,j} \bigwedge_{t<T-1} p_{i,t} \wedge \neg m_{i,t} \wedge \neg m_{j,t} \wedge a_{i,j} \rightarrow p_{j,t+1}$$

The EVOLVE constraint



Worst-case latency increases as either *m* increases or *K* decreases

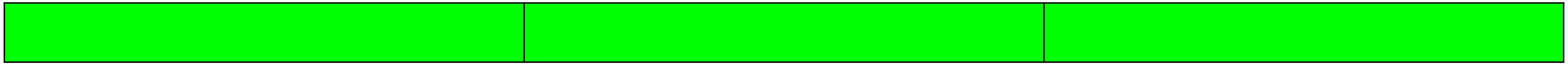
Table 1: Average increase in maximum latency under various (m, K) constraints relative to D_G.

		K						
		3	4	5	6	7	8	9
m	1	2.59x	2.59x	2.00x	2.00x	2.00x	2.00x	2.00x
	2	-	-	4.17x	4.17x	4.17x	3.00x	3.00x
	3	-	-	-	-	5.76x	5.76x	5.76x
	4	-	-	-	-	-	-	7.35x

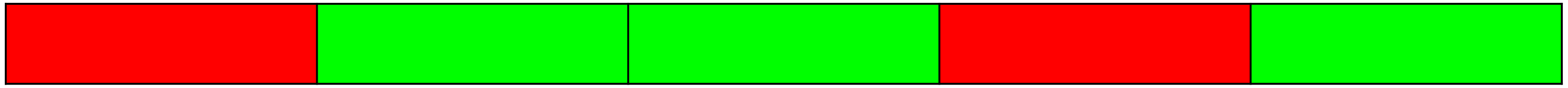
The fraction of *m* over *K* is not what drives high worst-case latency (consecutive misses do).

Richer Design Choices with Weakly-Hard

Under a hard timing model



Under a weakly-hard timing model

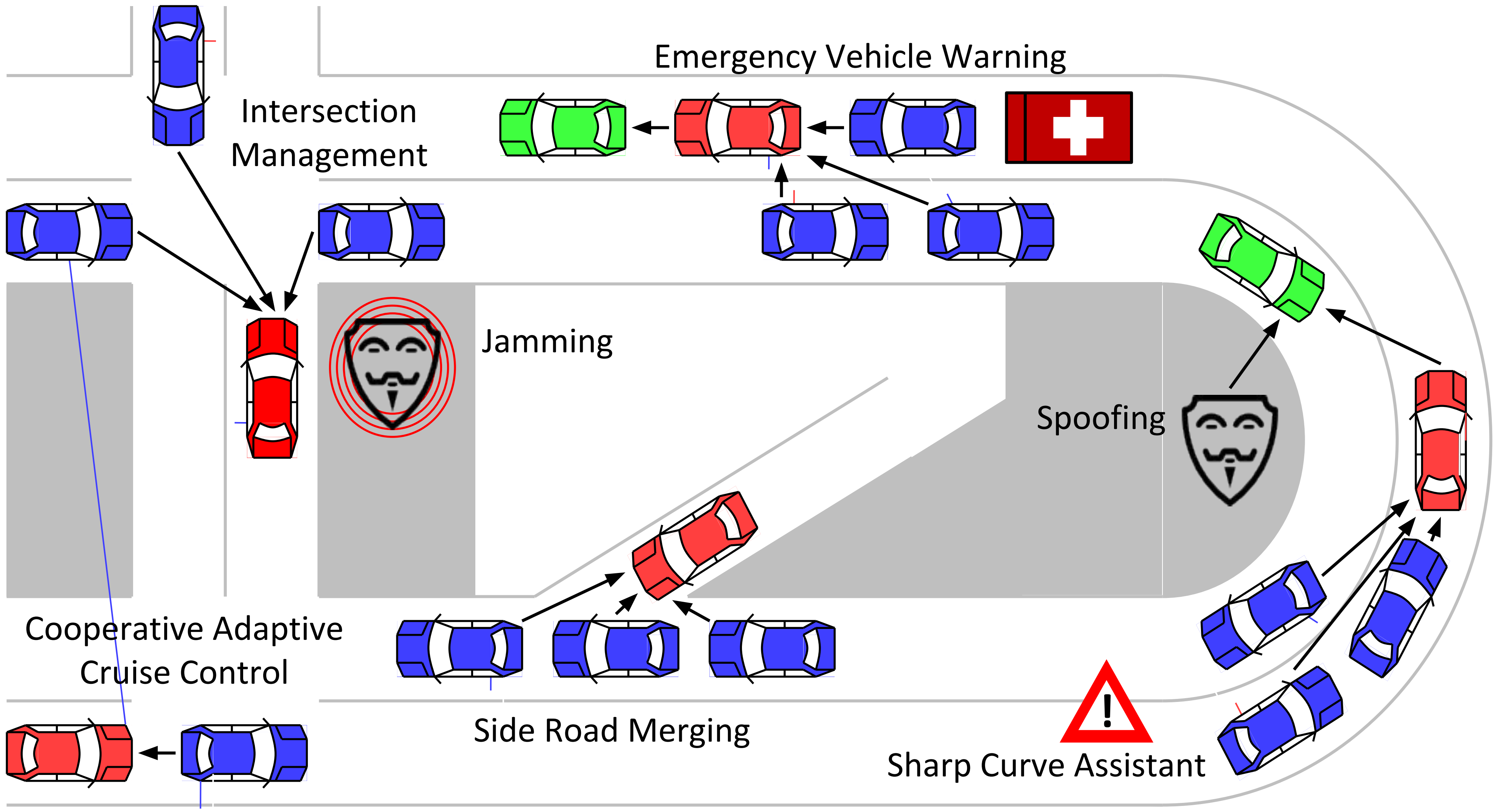


Allowing bounded misses can enable shorter periods, potentially enabling faster runtimes and better performance

V2V Networks

à la VANET

Beyond single-vehicle autonomous driving


Emergency Vehicle Warning
Intersection Management
Jamming
Spoofing
Cooperative Adaptive Cruise Control
Side Road Merging
Sharp Curve Assistant

Communication Disturbance in V2V

Packet Delay & Loss

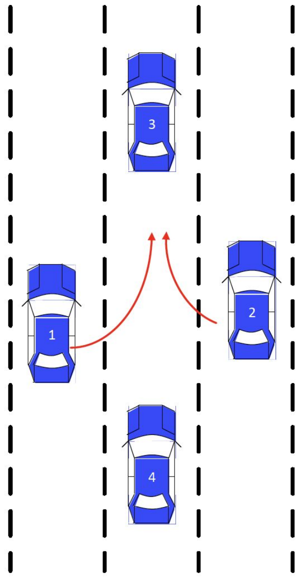
- Communication delay
- Packet collision/loss
- Jamming/flooding attacks

Prior Work

- Doesn’t consider packet loss
- Deadlock & unsafe situations
- Liveness issues

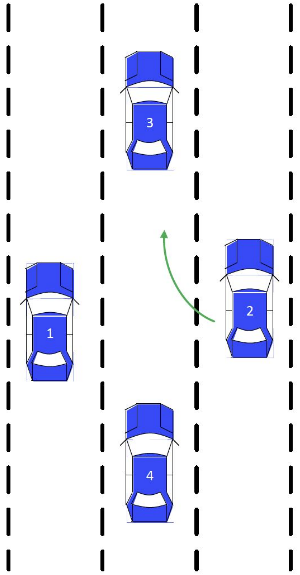
How should one measure the impact of disturbances? How should one derive the communication requirements?

Cooperative Lane Changing Under Disturbance



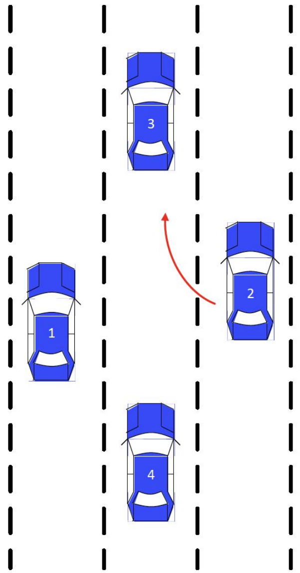
Vehicle	Agreement
1	1 merge
2	2 merge
3	2 merge
4	1 merge

a. unsafe scenario under partial agreement



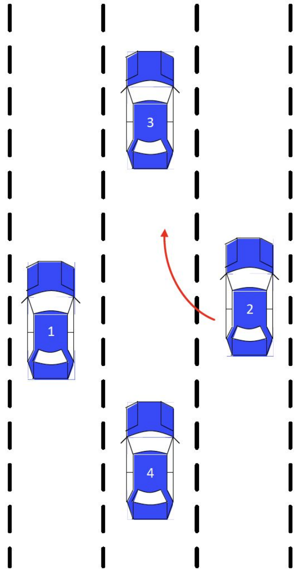
Vehicle	Agreement
1	2 merge
2	2 merge
3	2 merge
4	2 merge

b. safe scenario under global agreement (consensus)



Vehicle	Agreement
1	2 merge
2	2 merge
3	1 merge
4	2 merge

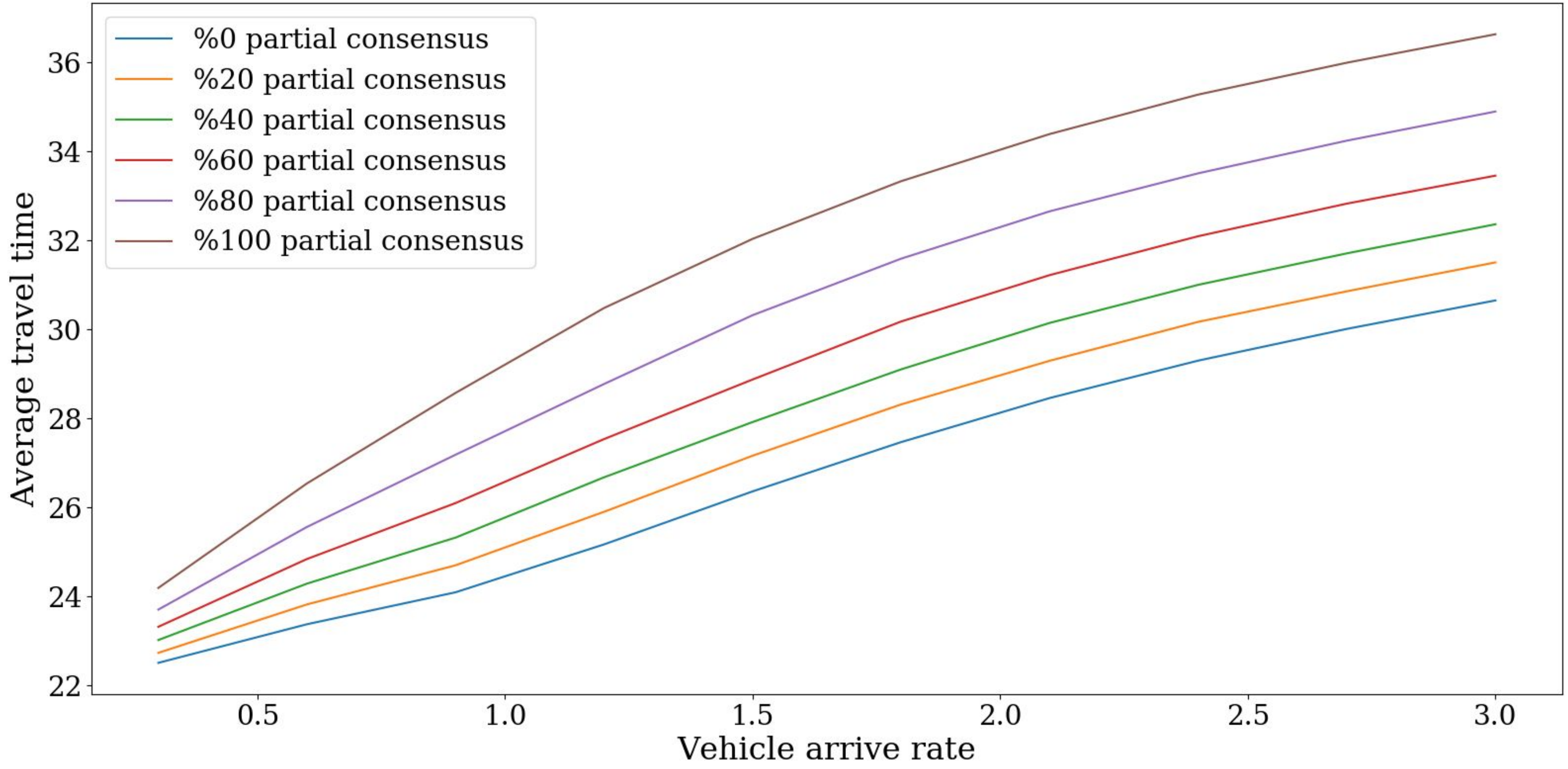
c. less safe than (b) but safer than (a)



Vehicle	Agreement
1	2 merge
2	2 merge
3	1 merge
4	1 merge

d. less safe than (c) but safer than (a)

Impact of Disturbances on Lane Changing



Performance degrades as disturbances cause increasing rates of partial consensus.

A Weakly-Hard Model of V2V disturbances allows us to directly reason about safety and performance

Research Directions

Networked Systems through the lens of Weakly-Hard Models

Stabilization

Given any, even faulty, initial state, the system should reach a correct state.

Reachability

Packets of a given class should only reach the designated host.

Consensus

Leader selection in the presence of faulty nodes or links.

Reliability

Correctness should be tolerant to occasional link failures.

QoS

Bounded latency in packet routing.

Thanks!

Credits & References

Nuclear power plant photo: wikipedia user Avda, CC BY-SA 3.0

Boeing MAX 737 photo: wikipedia user Acefitt, CC BY-SA 4.0

Inkjet printer photo: André Karwath, CC BY-SA 2.5

Car manufacturing photo: wikipedia user Siyuwj, CC BY-SA 3.0

Icons: Font Awesome CC BY 4.0 License

Wu, T., & Jin, S. (2008). Weakly hard real-time scheduling algorithm for multimedia embedded system on multiprocessor platform. *Proceedings - 2008 the 1st IEEE International Conference on Ubi-Media Computing and Workshops, U-Media2008*, 320–325. https://doi.org/10.1109/UMEDIA.2008.4570910

Alenawy, T. A., & Aydin, H. (2005). Energy-Constrained Scheduling for Weakly-Hard Real-Time Systems.

Duggirala, P. S., & Viswanathan, M. (2016). Analyzing Real Time Linear Control Systems Using Software Verification. *Proceedings - Real-Time Systems Symposium*, *2016-Janua*, 216–226. https://doi.org/10.1109/RTSS.2015.28

Broster, I., Bernat, G., & Burns, A. (2002). Weakly hard real-time constraints on controller area network. *Proceedings - Euromicro Conference on Real-Time Systems*, 134–141. https://doi.org/10.1109/EMRTS.2002.1019193

Bernat, G., Burns, A., & Member, S. (2001). Weakly Hard Real-Time Systems, *50*(4), 308–321.

Ferrari, F., Zimmerling, M., Thiele, L., & Saukh, O. (2011). Efficient Network Flooding and Time Synchronization with Glossy. *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 73–84.

Frehse, G., Hamann, A., Quinton, S., & Woehrle, M. (2015). Formal analysis of timing effects on closed-loop properties of control software. *Proceedings - Real-Time Systems Symposium*, *2015-Janua*(January), 53–62. https://doi.org/10.1109/RTSS.2014.28

Ahrendts, L., Quinton, S., Boroske, T., & Ernst, R. (n.d.). Verifying Weakly-Hard Real-Time Properties of Traffic Streams in Switched Networks.

Zhang, T., Gong, T., Yun, Z., Han, S., Deng, Q., & Hu, X. S. (2018). FD-PaS: A fully distributed packet scheduling framework for handling disturbances in real-time wireless networks. *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS*, 1–12. https://doi.org/10.1109/RTAS.2018.00007

Bin Hao, Jian Tang, & Guoliang Xue. (2004). Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation, 246–250. https://doi.org/10.1109/hpsr.2004.1303479