2022
CPS-IoT Week

SmarTiSCH: An Interference-Aware Engine for IEEE 802.15.4e-based Networks

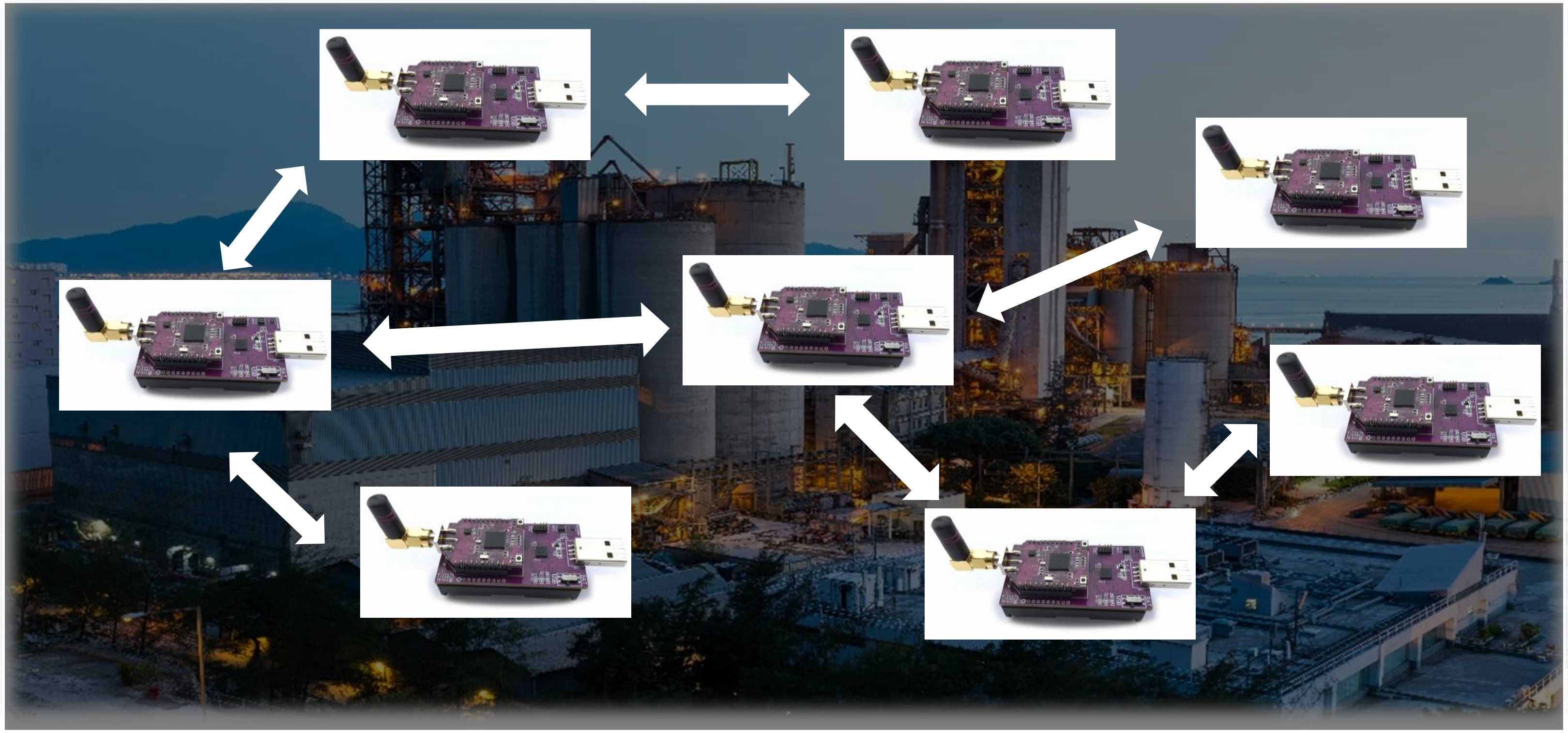
Zihao Yu†, Xin Na†, Carlo Alberto Boano‡, Yuan He†, Xiuzhen Guo†,
Pengyu Li†, and Meng Jin §

†School of Software & BNRist, Tsinghua University, China

‡Institute of Technical Informatics, Graz University of Technology, Austria

§ Shanghai Jiao Tong University, Shanghai, China

Industrial IoT

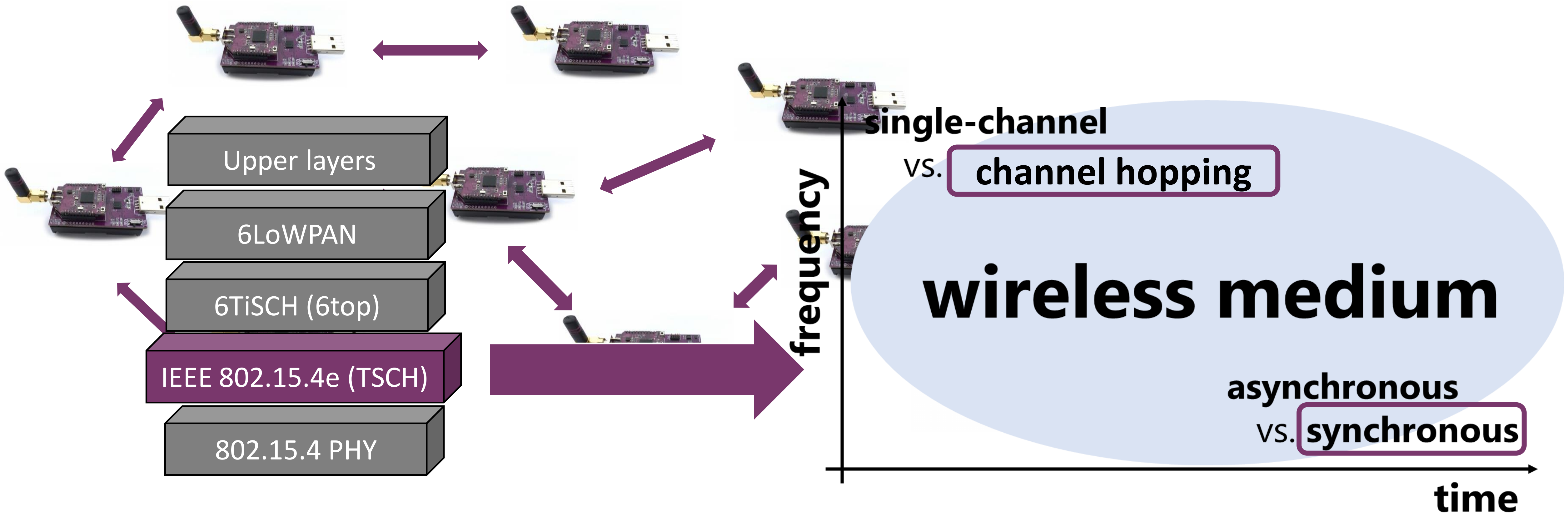


Industrial IoT: Mission-critical applications

Stringent requirement
on reliability

IEEE 802.15.4e

Mission-critical IoT applications ⇨ IEEE 802.15.4e standard: TSCH (time-slotted channel hopping)



Upper layers

6LoWPAN

6TiSCH (6top)

IEEE 802.15.4e (TSCH)

802.15.4 PHY

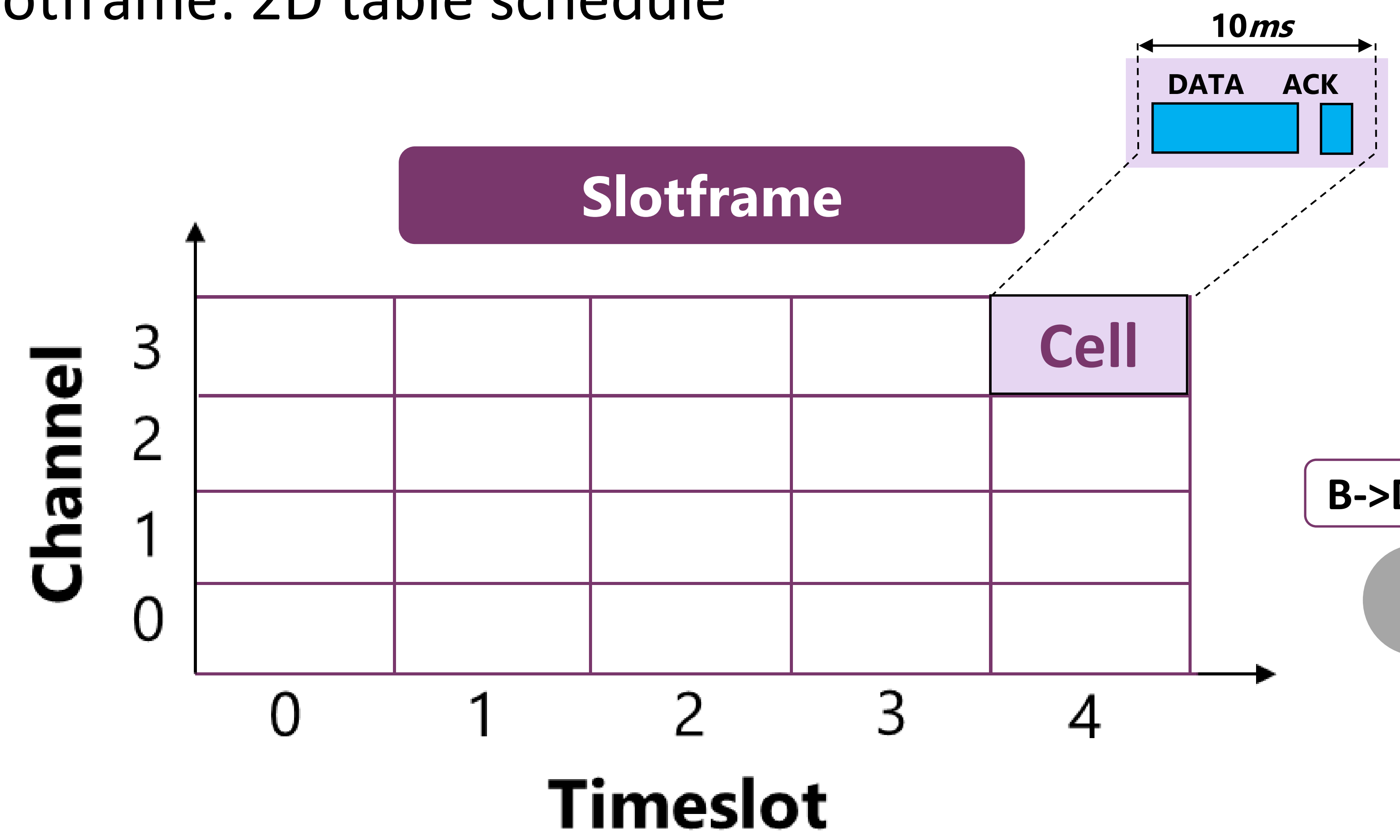
single-channel vs. channel hopping

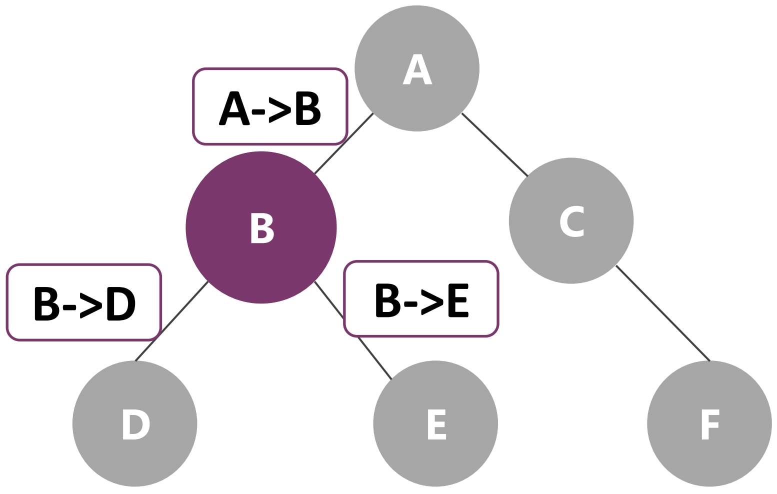
wireless medium

asynchronous vs. synchronous

TSCH Protocol

Slotframe: 2D table schedule



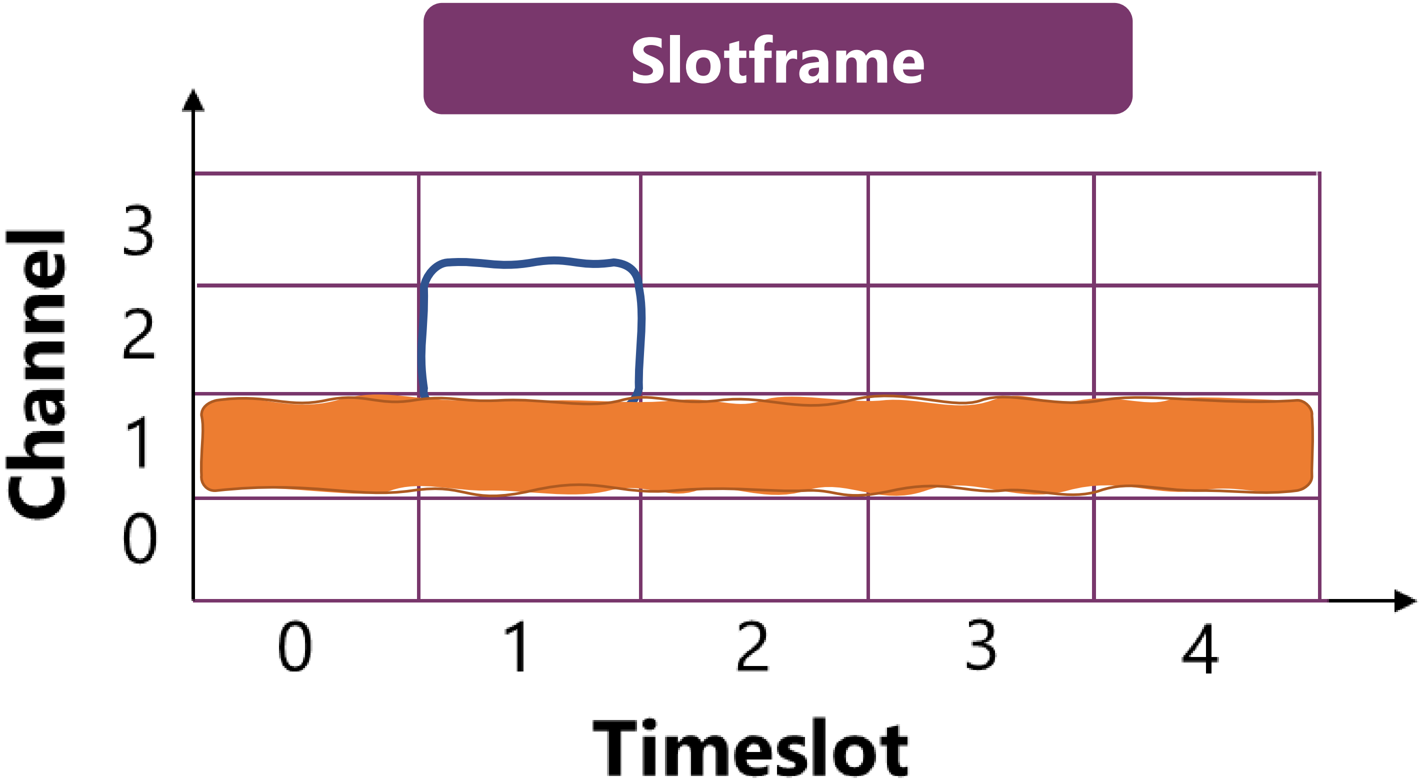


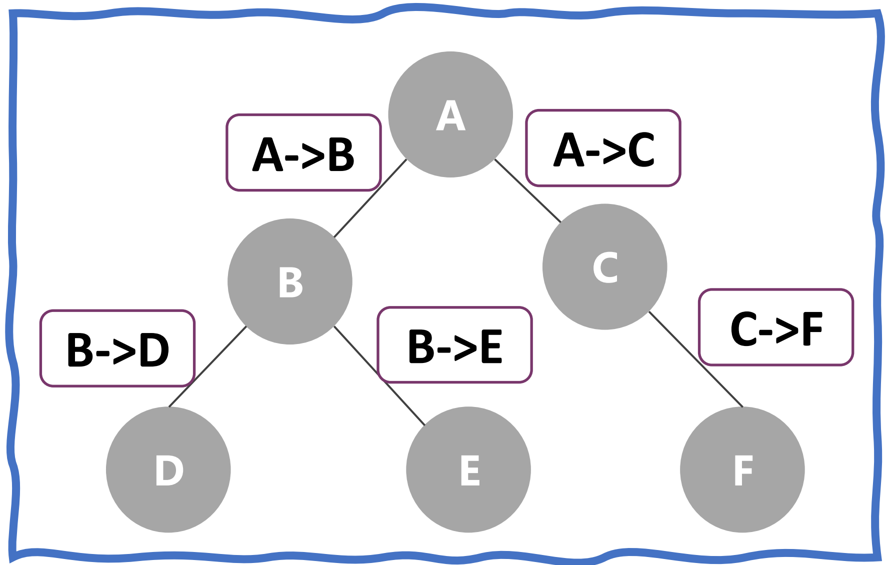
A simplified network

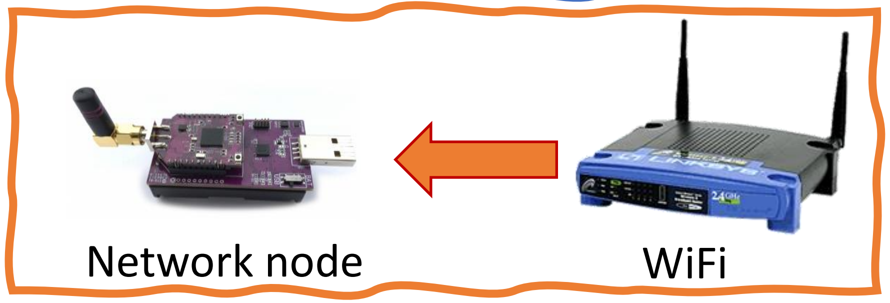
Scheduler: how to allocate the cells with minimum energy expenditure?

Problem: interference

- **Internal interference**: internal collisions between links
- **External interference**: impact from external devices like WiFi devices

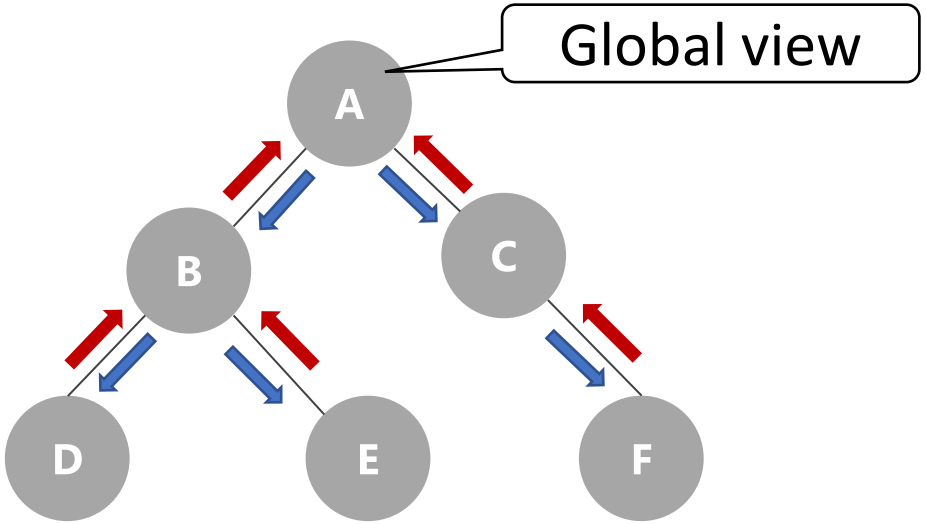




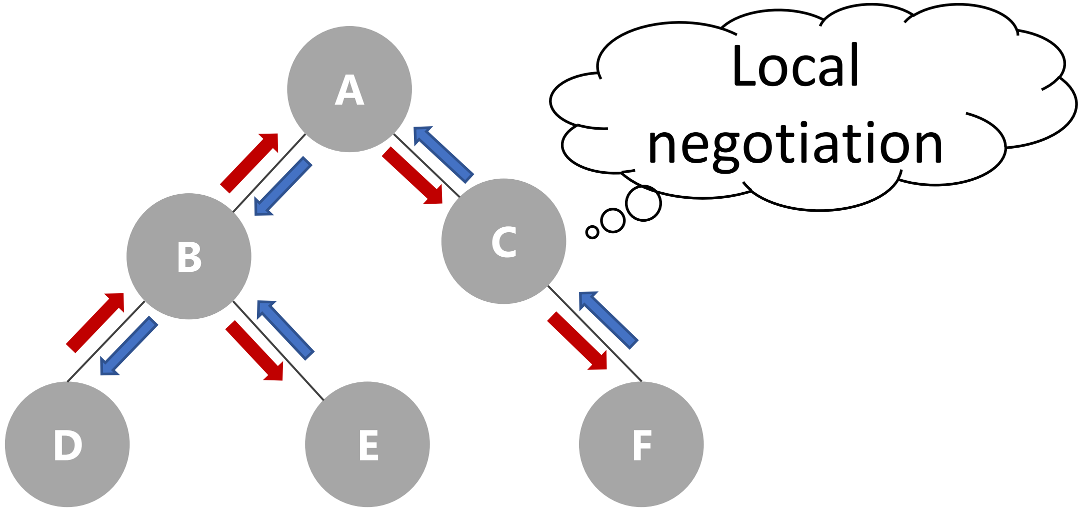


Previous works: centralized/distributed scheduler

Centralized scheduler



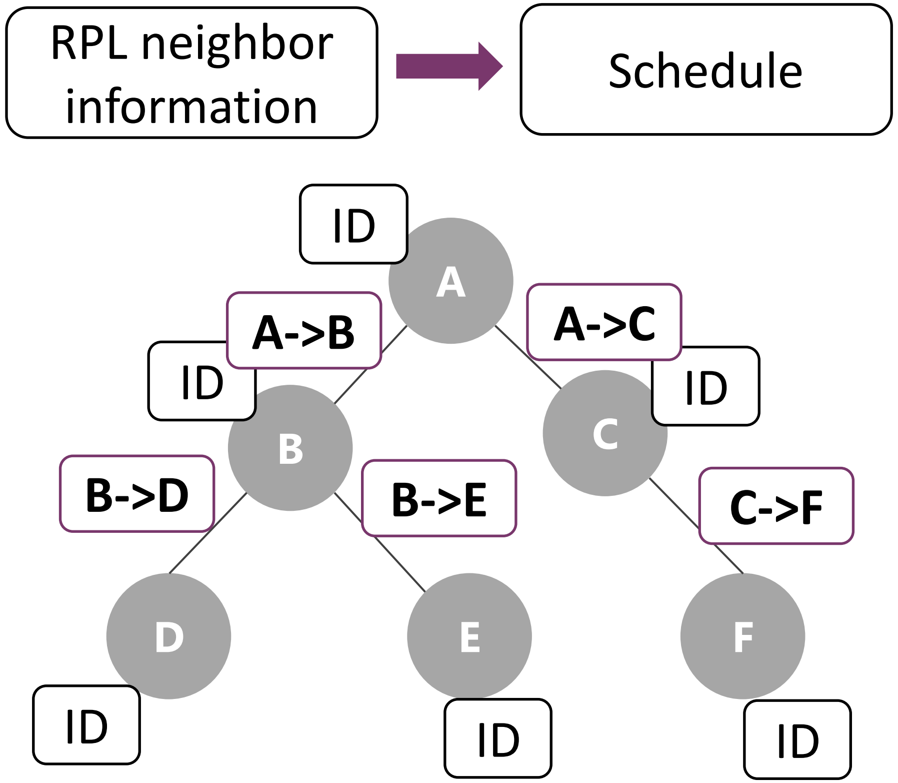
Distributed scheduler

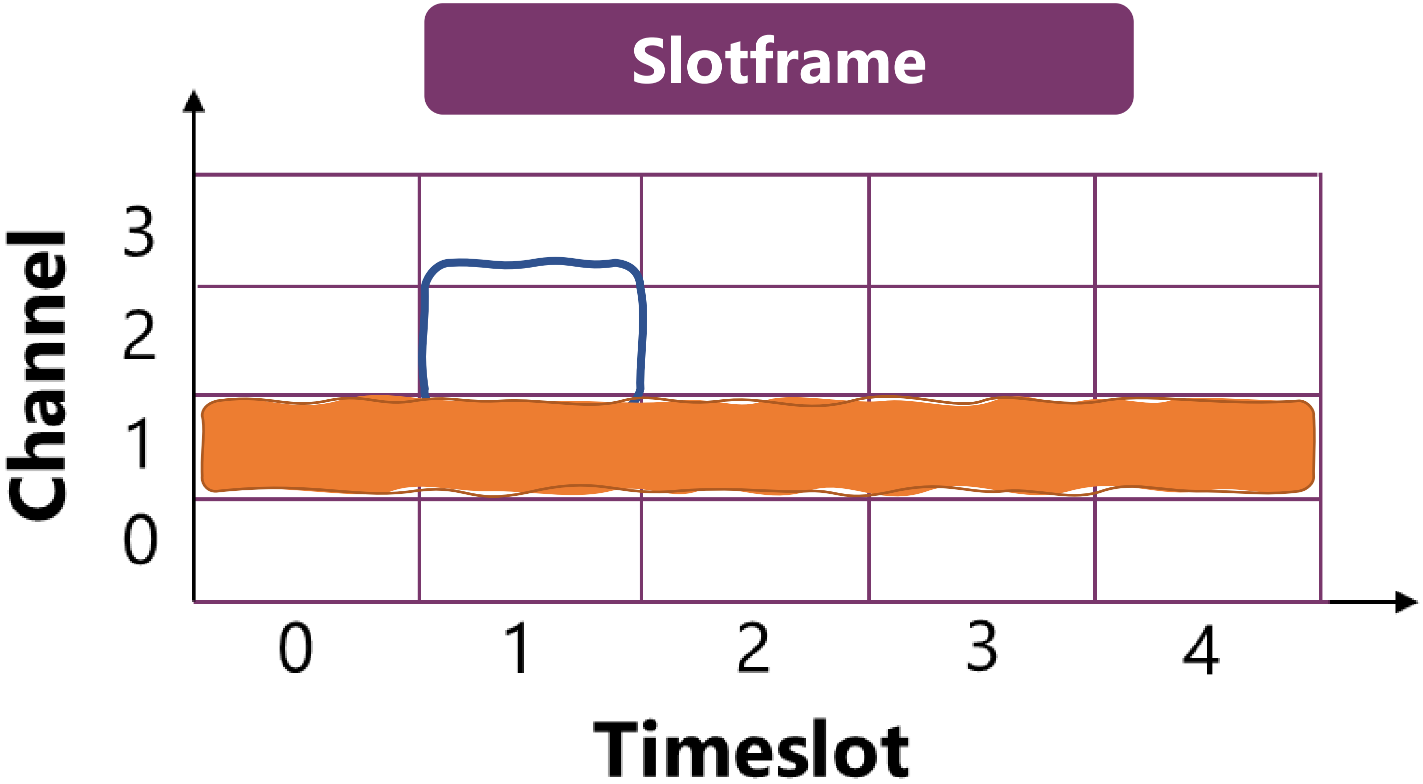


Both the schedulers rely on **extra control traffic** to detect and handle interference!

Recent works: autonomous scheduler

Autonomous scheduler



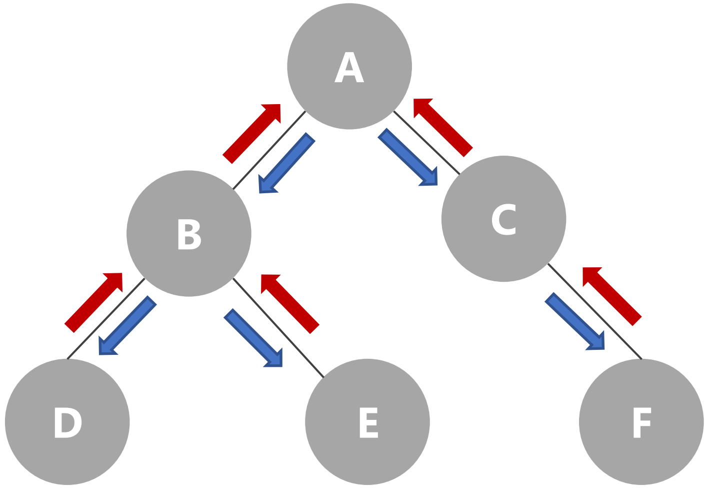


Vulnerable to interference due to unawareness!

Dilemma: overhead or interference?

Awareness of interference

Centralized/distributed scheduler

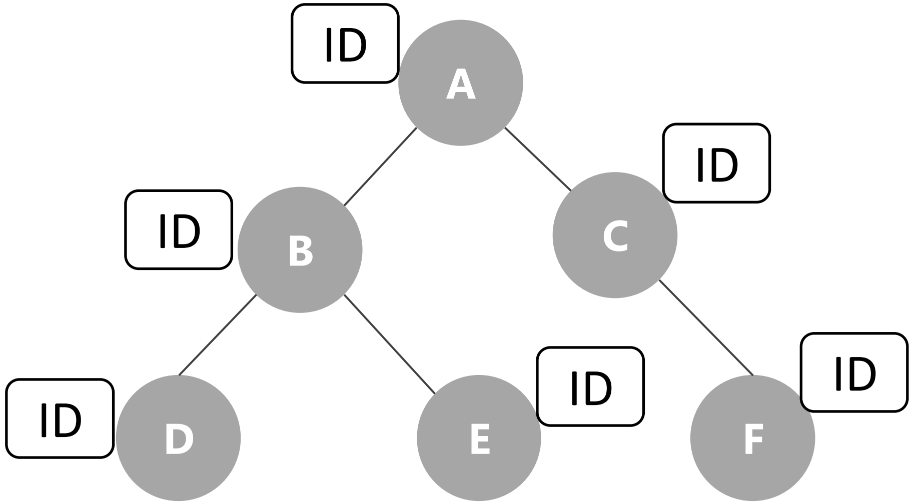


With extra traffic

Both?

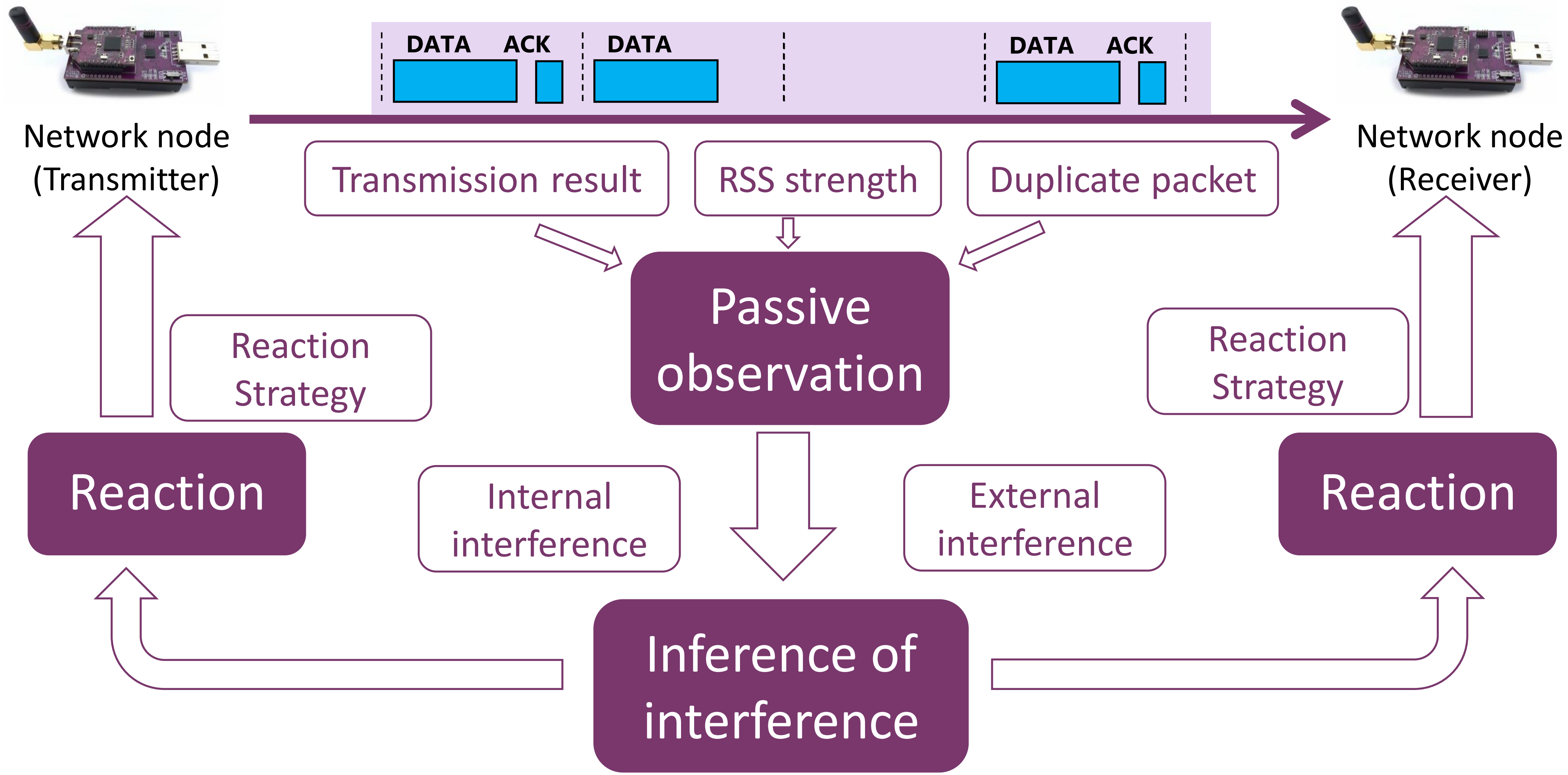
No control traffic

Autonomous scheduler



Vulnerable to interference

SmarTiSCH: an interference-aware engine



Challenges

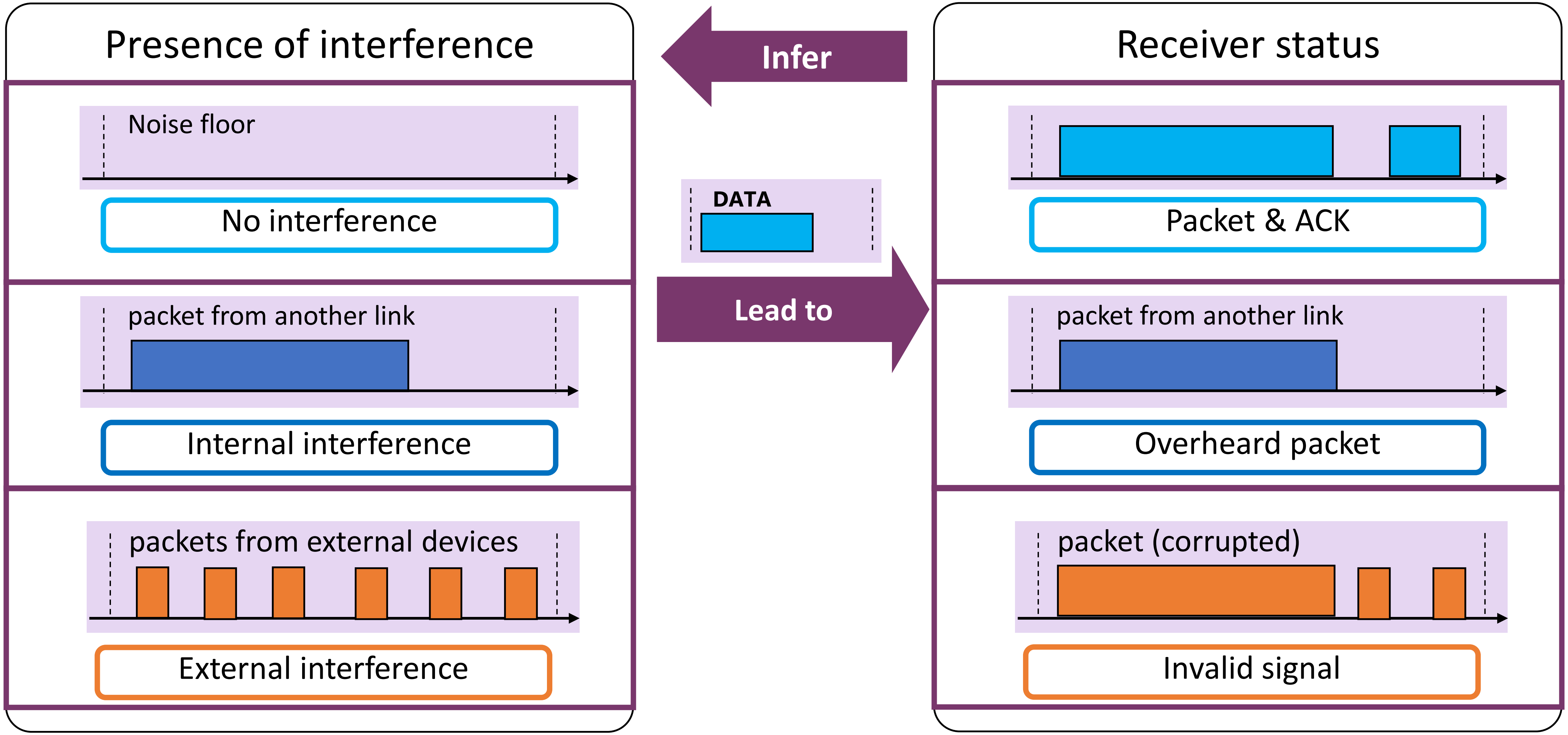
Challenge 1 – To **obtain the awareness** of interference based on data transmission

- How to infer the type of interference from **limited information**?
- How to solve the **asymmetry** between the observation of the transmitter and the receiver?
- How to handle the **ambiguity** of the inference process?

Challenge 2 –To **share and utilize the awareness** under interference without extra control traffic

- How to reliably share the information **under interference**?
- How to build consensus between the transmitter and the receiver **without control packets**?
- How to **react** to interference by updating the schedule?

SmarTiSCH: passive observation



SmarTiSCH: inference of interference

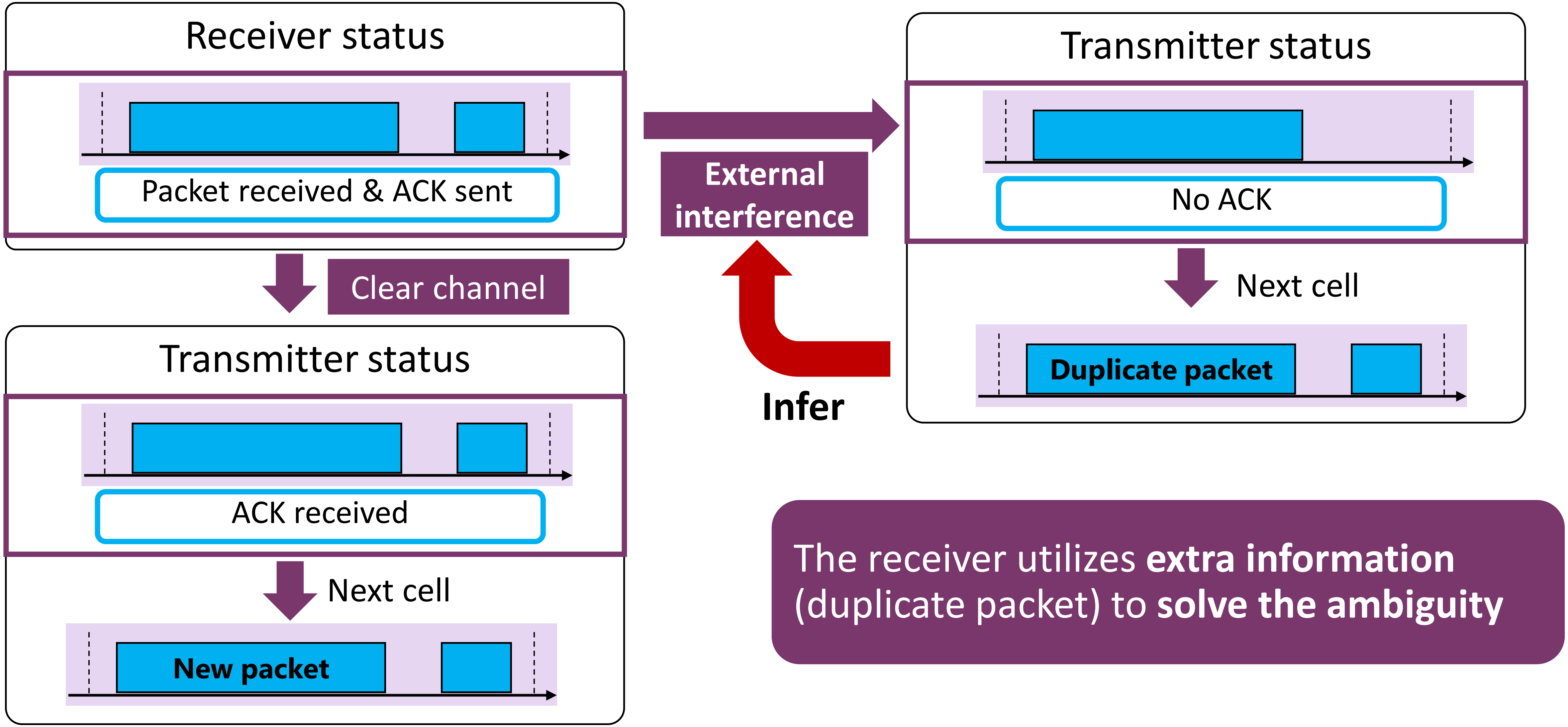
Asymmetry between the observations of the transmitter and the receiver

		Transmitter status			
		No traffic	CCA busy	ACK reveived	No ACK
Receiver status	Clear channel	No internal No external			
	Overheard packets	Internal No external	Internal External		
	ACK sent			No internal No external	No internal External
	Invalid signals	(internal) External	(internal) External		Internal OR External

Ambiguity?

A **receiver-dominant** design principle to infer the presence of interference

SmarTiSCH: inference of interference



Receiver status

Packet received & ACK sent

Clear channel

Transmitter status

ACK received

Next cell

New packet

External interference

Transmitter status

No ACK

Next cell

Duplicate packet

Infer

The receiver utilizes **extra information** (duplicate packet) to **solve the ambiguity**

Challenges

Challenge 1 – To **obtain the awareness** of interference based on data transmission

- How to infer the type of interference from **limited information**?
- How to solve the **asymmetry** between the observation of the transmitter and the receiver?
- How to handle the **ambiguity** of the inference process?

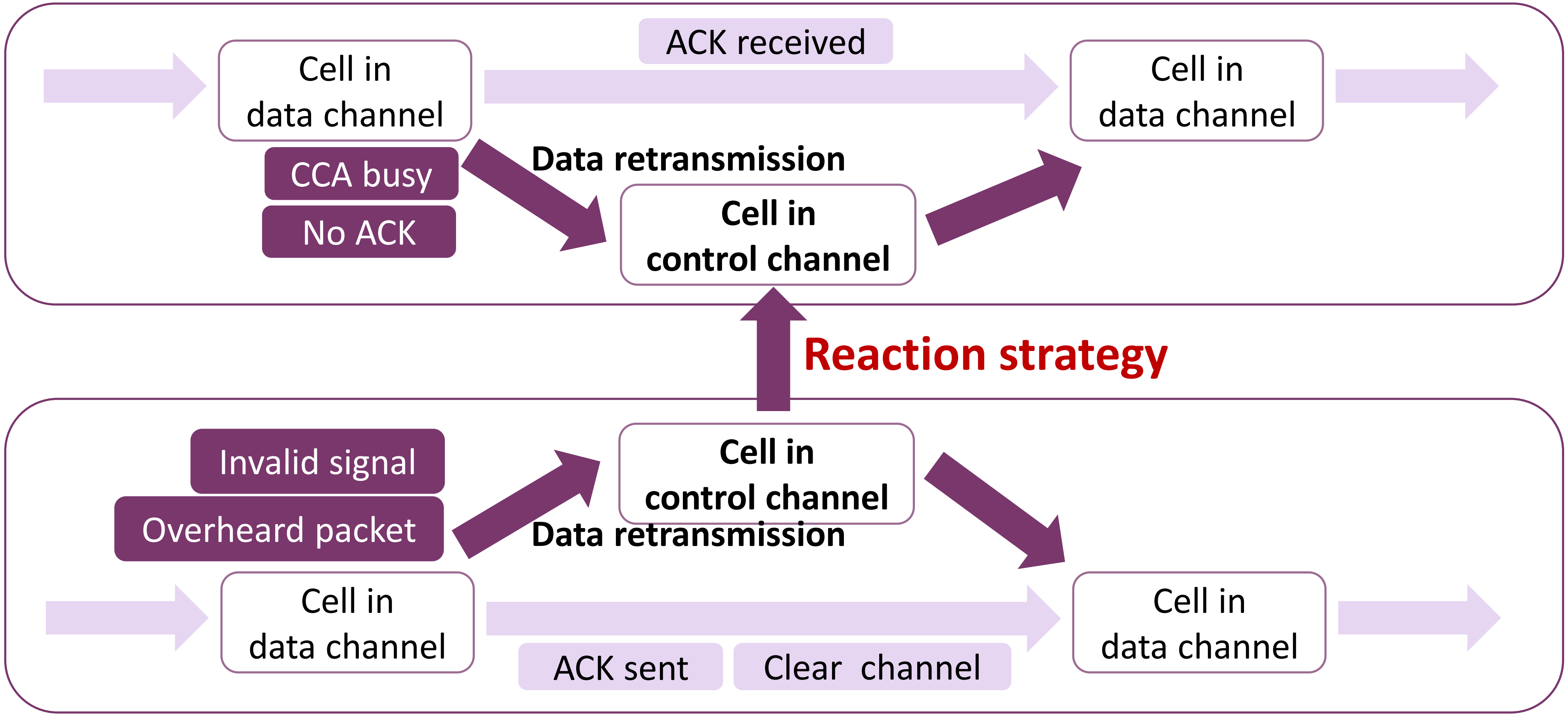
Challenge 2 –To **share and utilize the awareness** under interference without extra control traffic

- How to reliably share the information **under interference**?
- How to build consensus between the transmitter and the receiver **without control packets**?
- How to **react** to interference by updating the schedule?

SmarTiSCH: reaction to interference

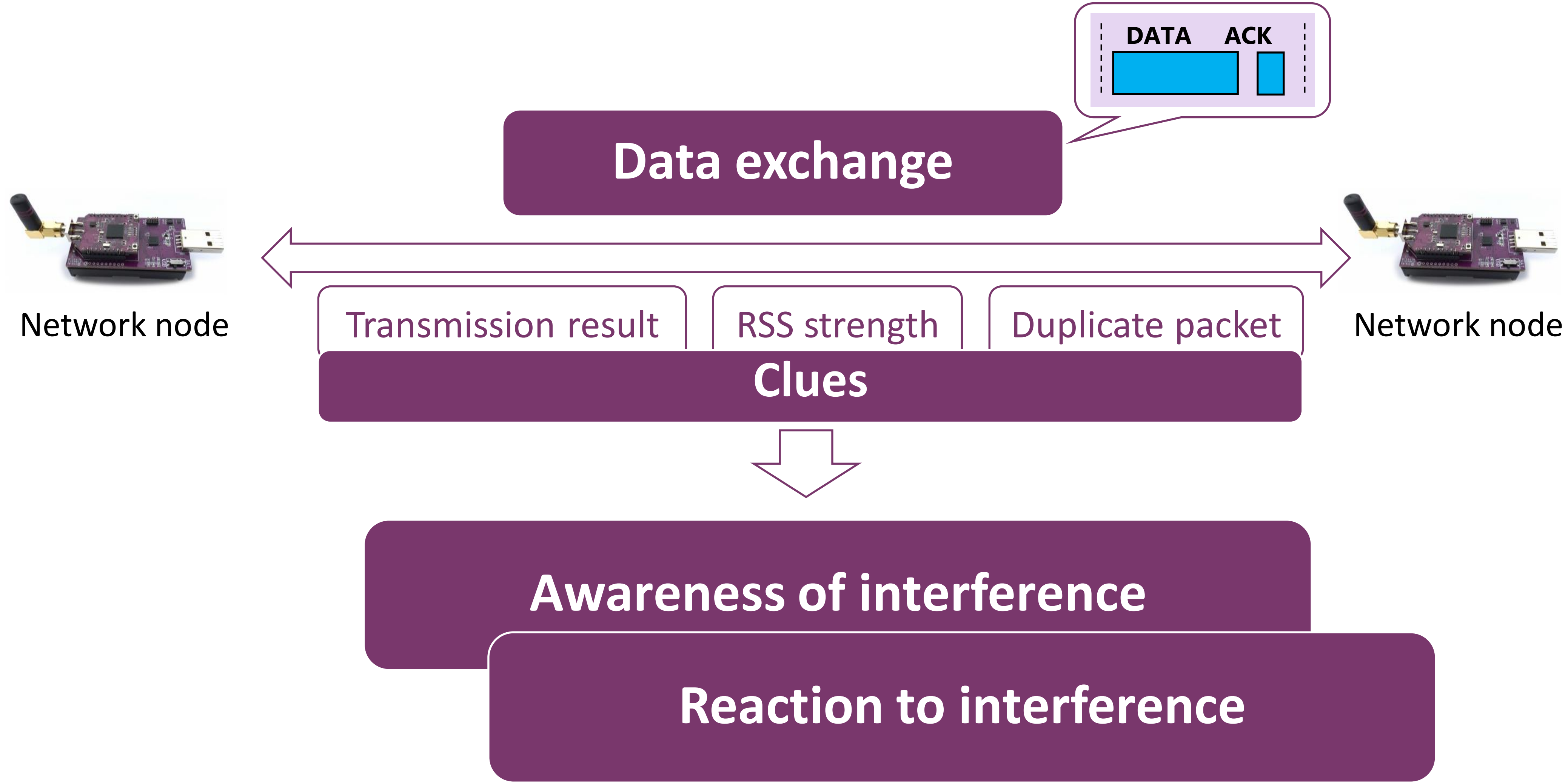
Transmitter

Receiver



The transmitter and the receiver enter the control channel for reliable information exchange

Observation: data exchange provide clues!



Data exchange

Network node

Network node

Transmission result

RSS strength

Duplicate packet

Clues

Awareness of interference

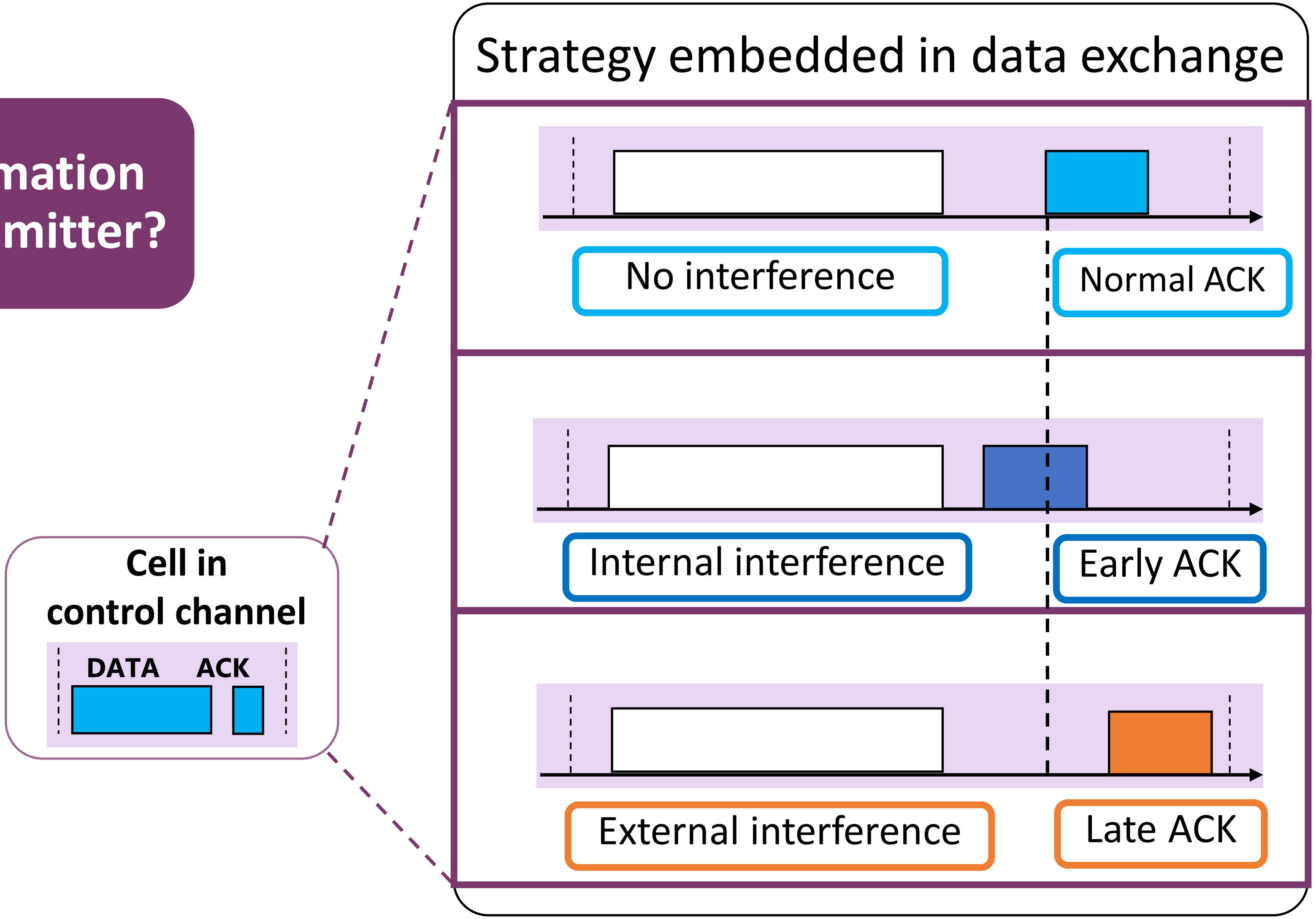
Reaction to interference

SmarTiSCH: reaction to interference

How to transfer the information from receiver to the transmitter?

No extra control packet

Information embedded in the time of ACK



Cell in control channel

Strategy embedded in data exchange

No interference — Normal ACK

Internal interference — Early ACK

External interference — Late ACK

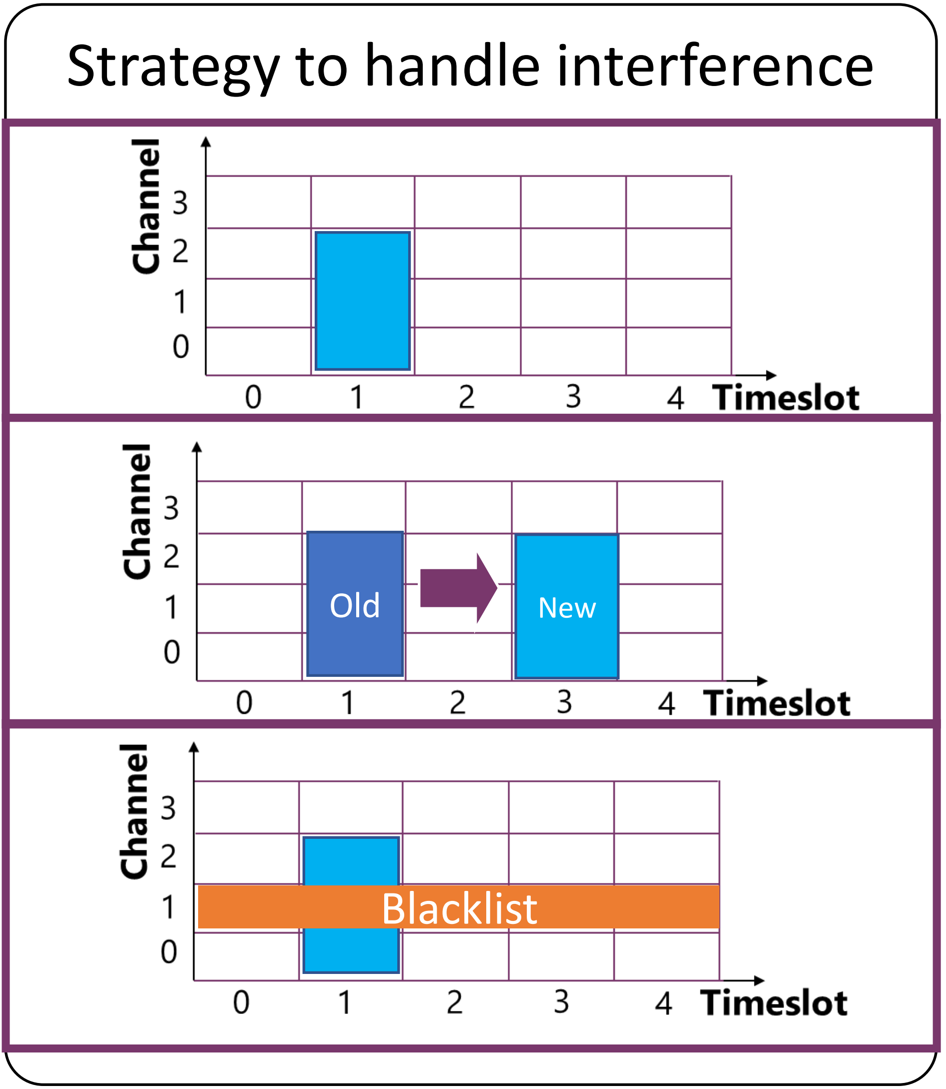
SmarTiSCH: reaction to interference

How to react to interference in three different conditions?

Normal ACK → No interference →

Early ACK → Internal interference →

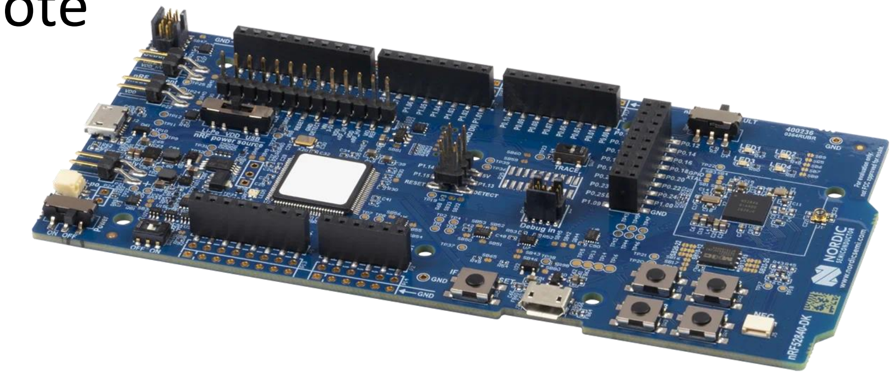
Late ACK → External interference →



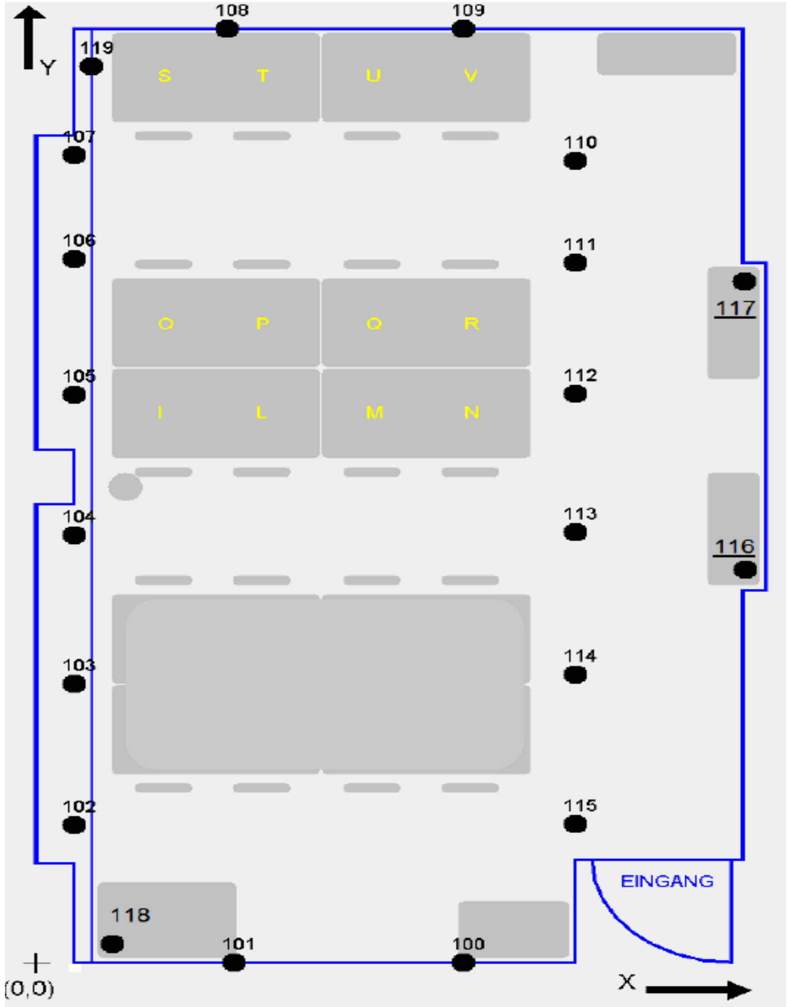
Strategy to handle interference

SmarTiSCH: Implementation

- SmarTiSCH: link-based Orchestra with our engine
- Hardware: nRF52840 nodes
- Software: ContikiNG, and use RPL with storing mode on top
- Testbed: 20 nodes in a 50m^2 area
- External interference: JamLab-NG
- Application: a data collection application at the root note



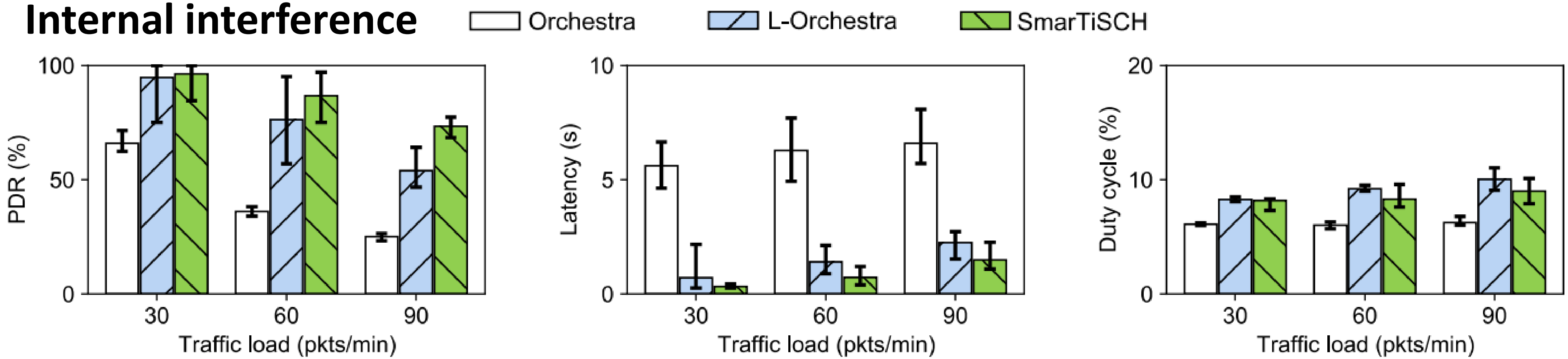
nRF52840 node



Testbed

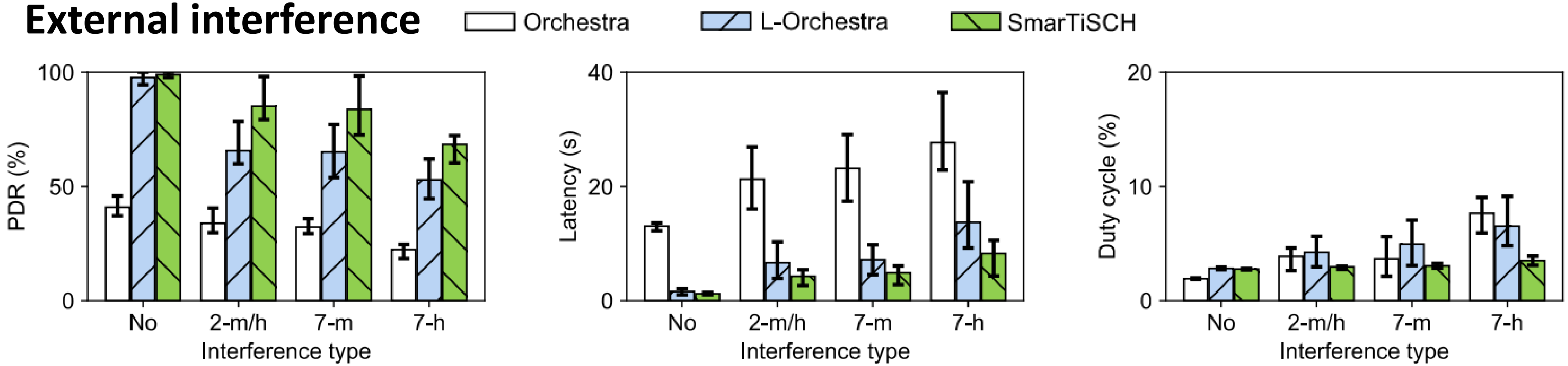
SmarTiSCH: Evaluation

Internal interference



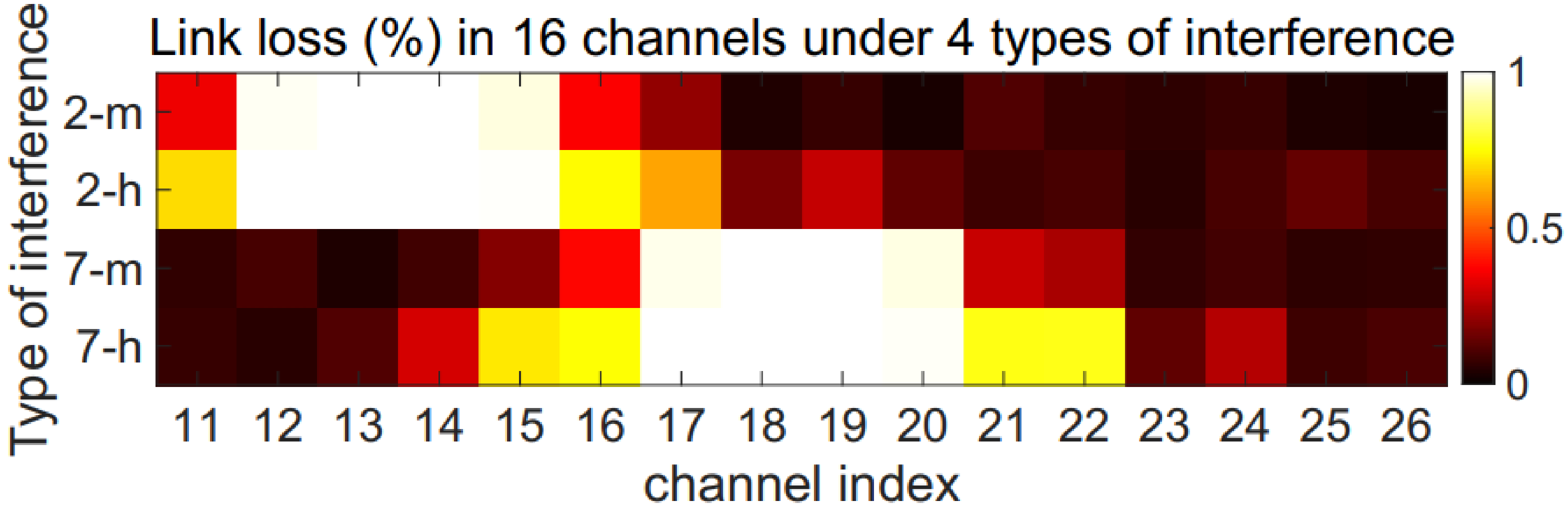
(a) PDR (b) Latency (c) Duty cycle

External interference

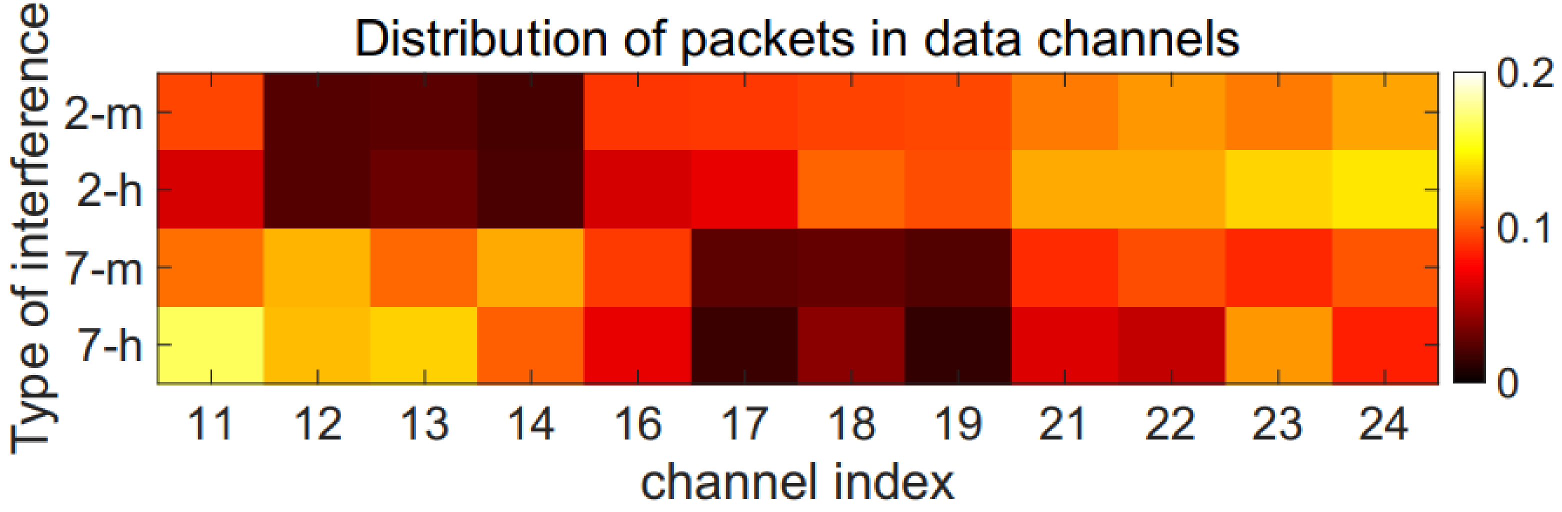


(a) PDR (b) Latency (c) Duty cycle

SmarTiSCH: Evaluation



Link loss in different channels



SmarTiSCH

Packet distribution in different channels

Conclusion

- SmarTiSCH increases the awareness and robustness to interference without extra cost of scheduling.
- Design: passive observation, inference of interference, reaction
- Evaluation: SmarTiSCH expands the channel resources, increases the network capacity, and improves the network performance without extra control traffic.

Systems & Ubiquitous Networking (SUN) Group