



电子科技大学
University of Electronic Science and Technology of China



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

Processor-Sharing Internet of Things Architecture for Large-scale Deployment

Qianhe Meng¹, Han Wang¹, Chong Zhang^{1,2}, Yihang Song¹, Songfan Li³,
Li Lu¹, and Hongzi Zhu⁴

¹University of Electronic Science and Technology of China ²Southwest Petroleum University

³The Hong Kong University of Science and Technology ⁴Shanghai Jiao Tong University

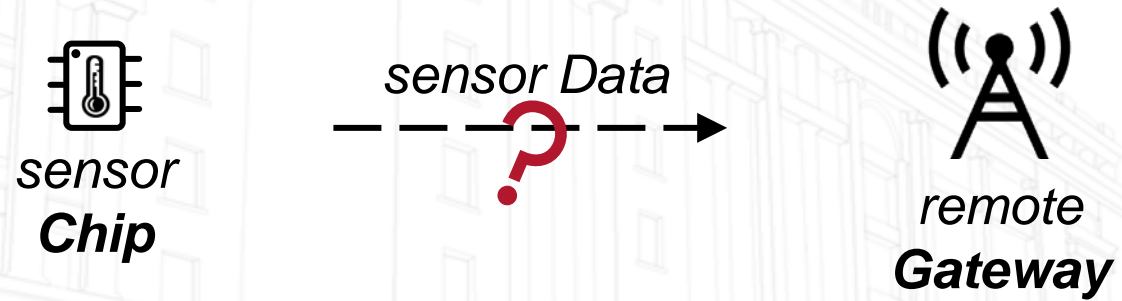
Sensor chips are keen to be **deployed** at a **large-scale**.



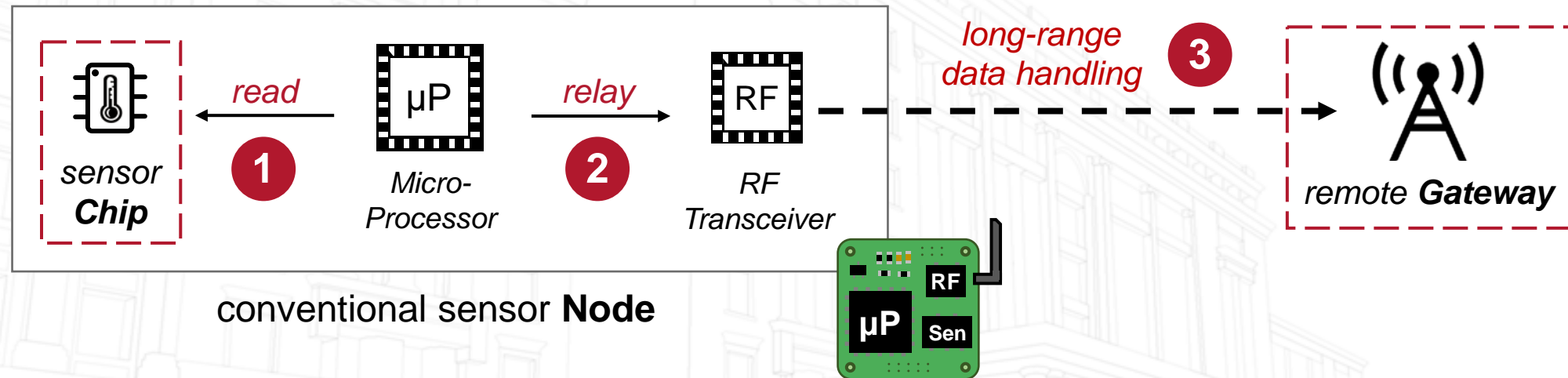
scenario #1: wildfire detection



scenario #2: structural health monitoring



Embedded Solution: Embedding each sensor chip into a full-fledged embedded device.



Nodes are **EXPENSIVE**
for large-scale deployment

Unaffordable **Manufacturing** Cost

*full-fledged with μ P and RF transceiver
→ **more than \$ 10 per node!***

Unaffordable **Maintenance** Cost

*power-hungry RF components
→ **frequent battery changes!***

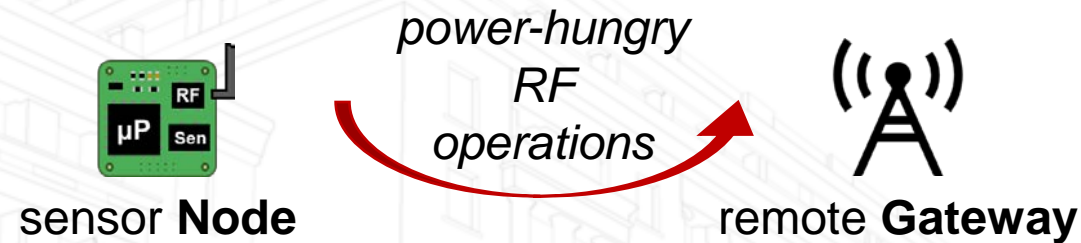
A faint, light-gray line drawing of a large, multi-story building with many windows and classical architectural features serves as the background for the slide.

Simplify nodes
for the ease of large-scale deployment.

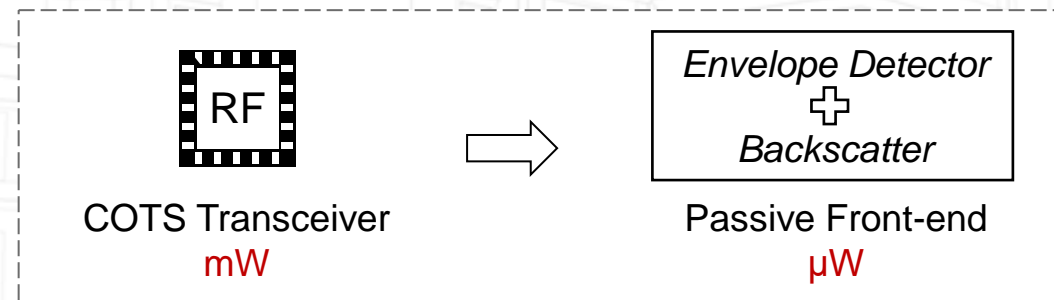
Shifting node's functionality **to** the remote gateway.



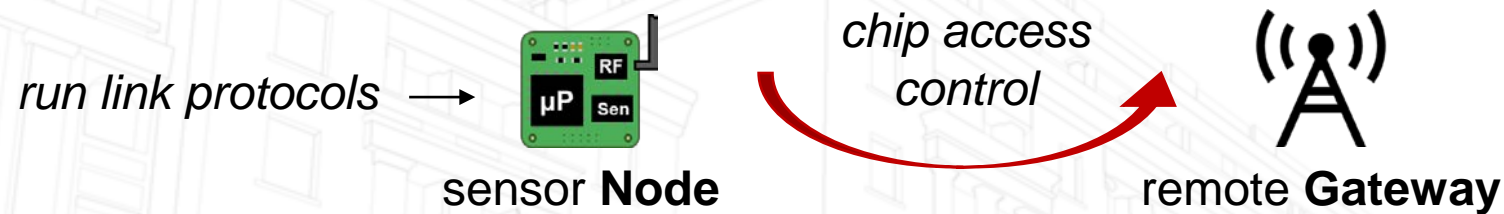
Direction #1: communication offload (e.g., backscatter)



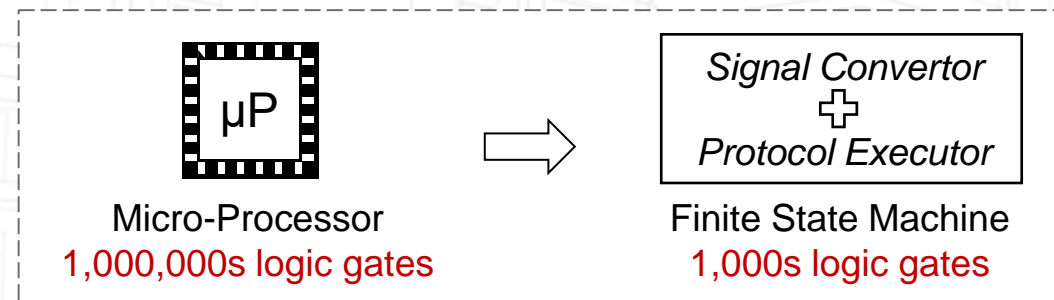
Drawback: Two-way communication range **limited to ~30 meters**.



Direction #2: computation offload (e.g., processor-free)



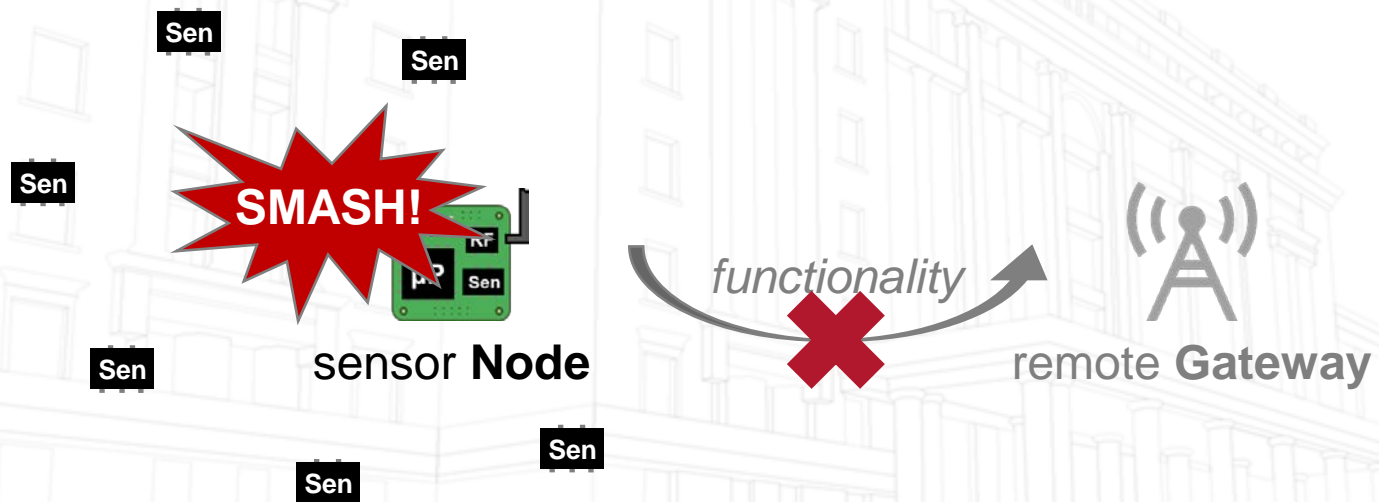
Drawback: Comm. overheads increased, thus network **scalability sacrificed**.



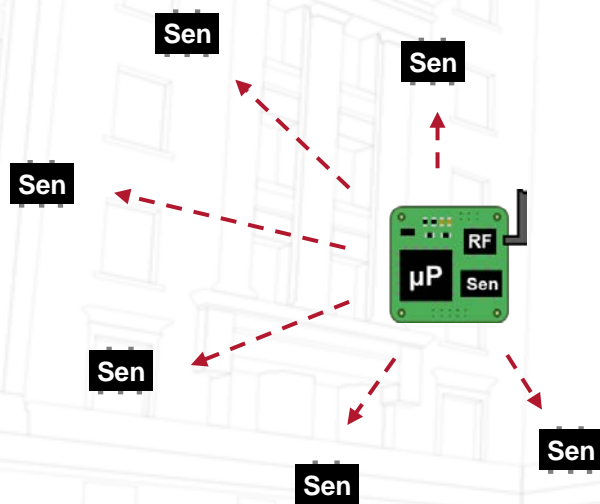
Insight: Current effort **trades** the network performance **for** the simplicity of sensor nodes. → More gateways to be deployed in trade-off!

Deployment costs are **INCREASED** at the **SYSTEM-LEVEL!**

Smash a full-fledged sensor node into the AIR
for **broader sensor-chip coverage**.



Could a processor access its
neighboring sensor chips
with **negligible** overheads?



deployment costs

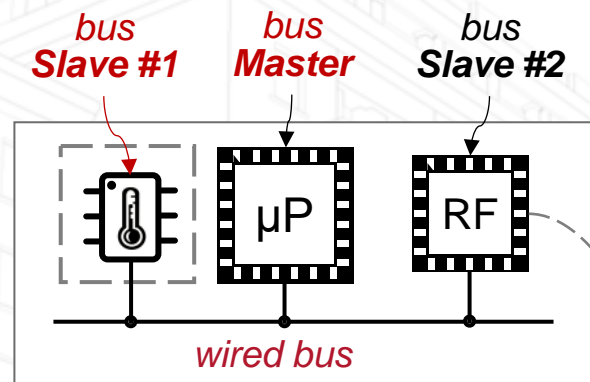


network scalability

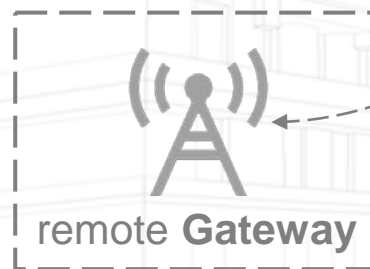


remote **Gateway**

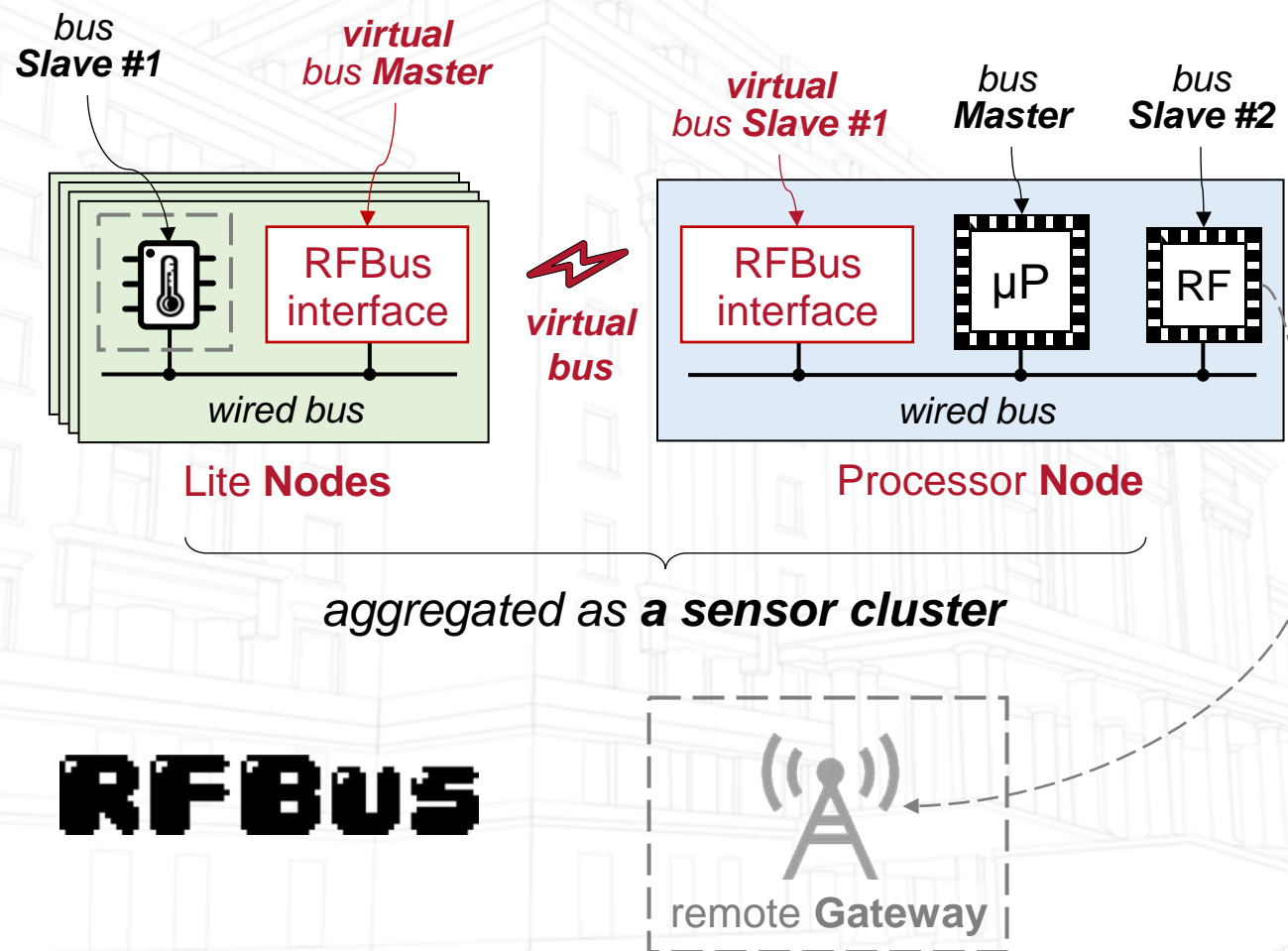
Basic Knowledge: Chips are inter-connected through **computer bus**.



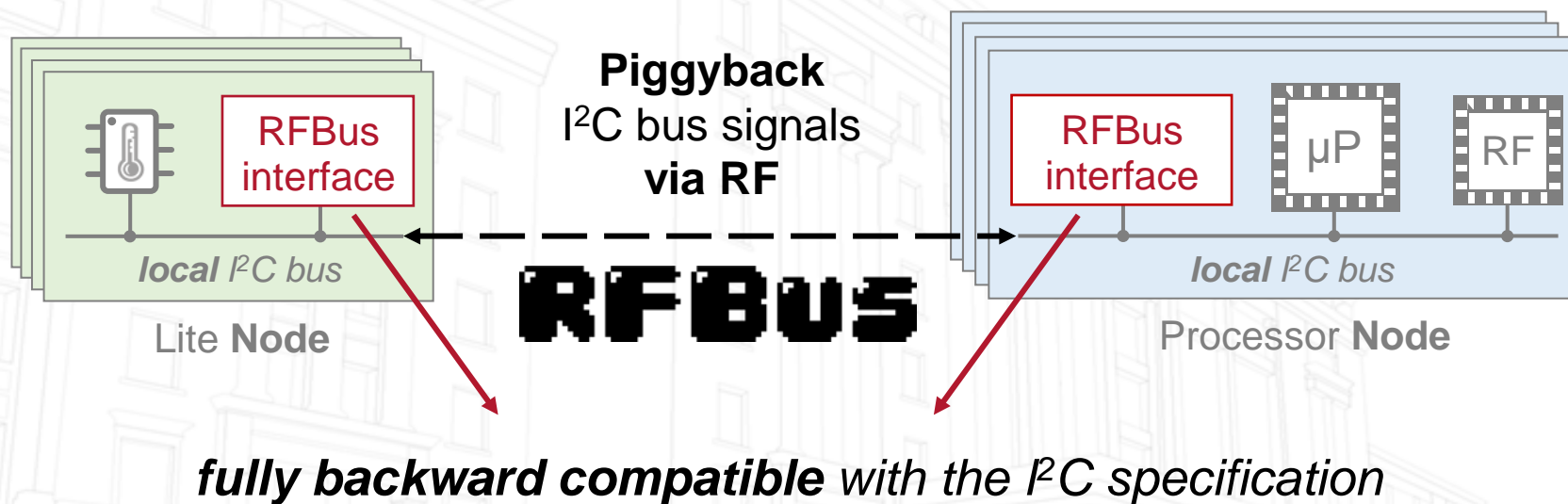
conventional sensor **Node**



Basic Idea: Constructing a *virtual* bus among chips.



Sweet Spot: Inherit link-layer services from I²C protocol transparently, including chip address, anti-collision and reliable delivery.



**A chip-oriented multi-to-multi RF network
powered by I²C.**

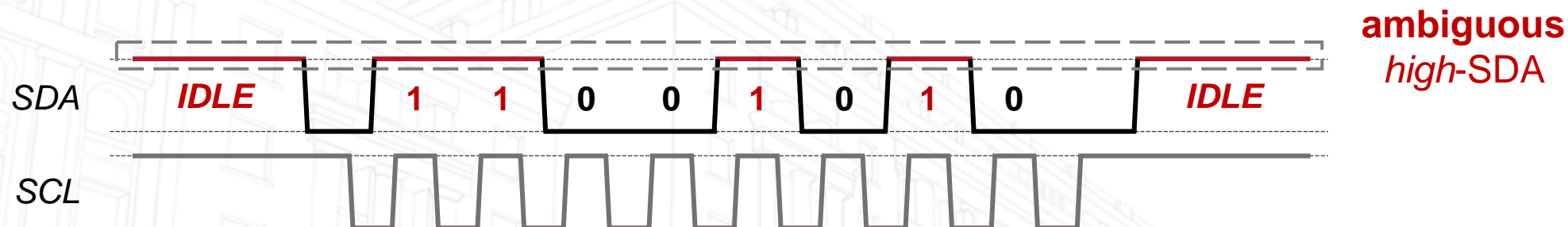
Design #1: RF Open-Drain (***PHY layer***)

Design #2: RFBus Front-End (***PHY layer***)

Design #3: Half-duplex RF signaling (***link layer***)

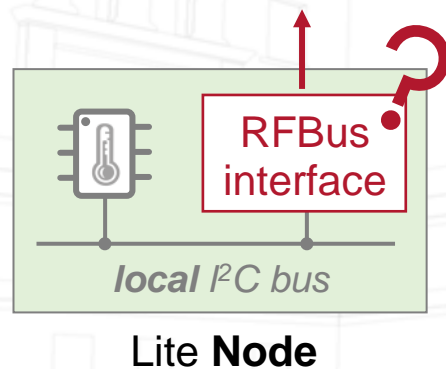
Design #4: Multi-to-Multi Chip Networking (***link layer***)

Problem: I²C express **trinary** bus information through **binary** voltage-level.



Lite node asks:

*‘Does the input high-SDA represents **logic ‘1’** or **IDLE**?’*



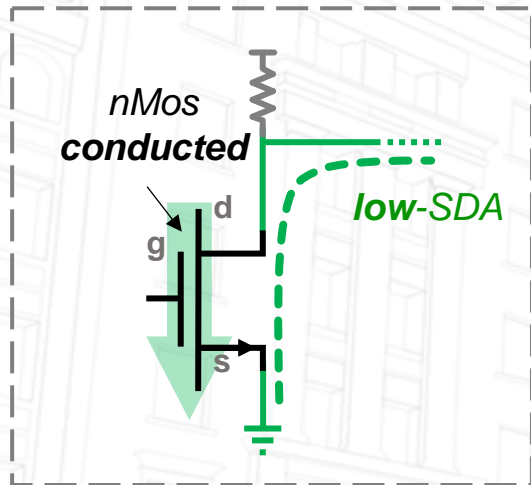
if logic ‘1’

piggyback it via RF;

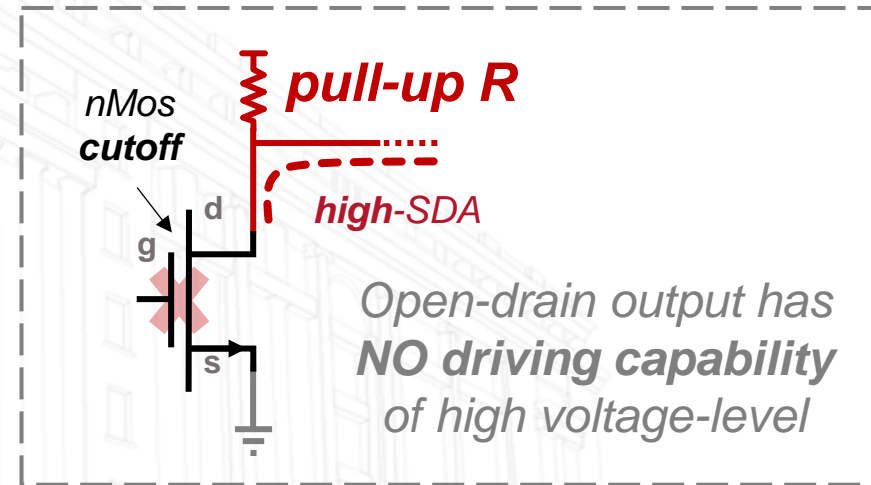
else (IDLE)

keep silent (release I²C bus);

Insight: Trinary I²C bus information is expressed by binary nMos control.



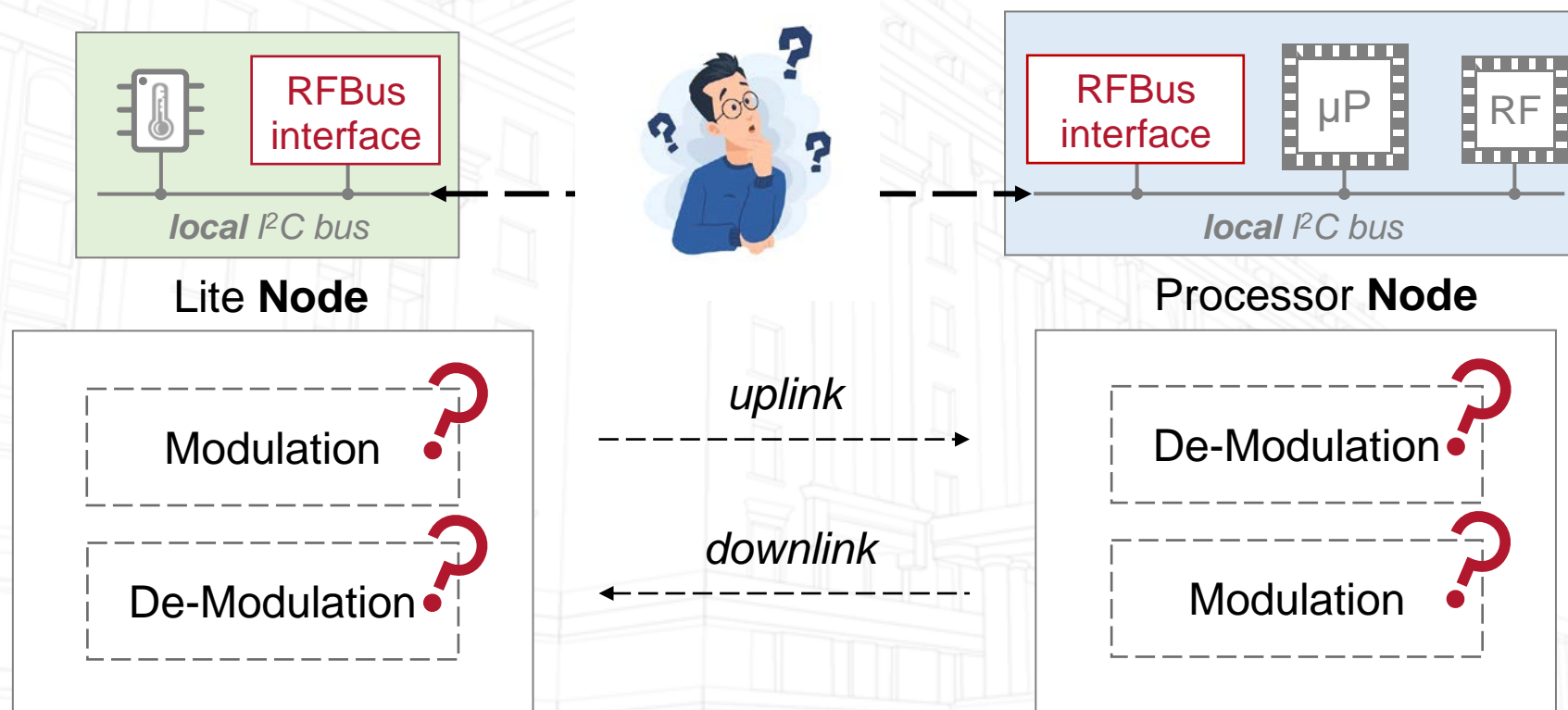
SDA expresses **logic '0'**



SDA expresses **logic '1' or IDLE**

Solution: Piggyback **low-SDA** via **RF** only.

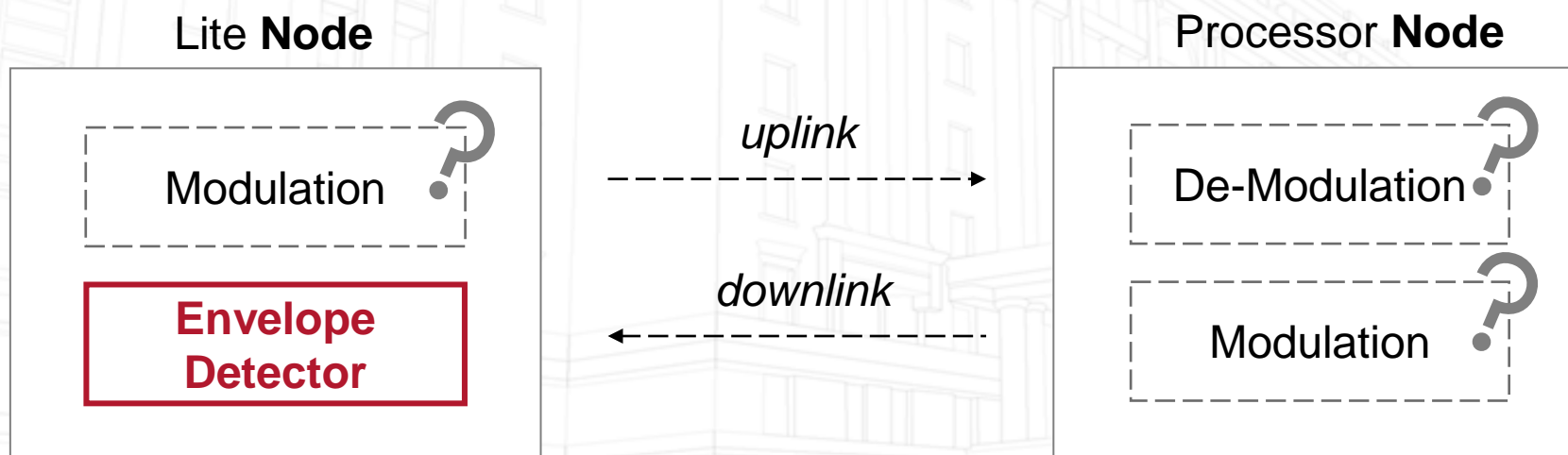
Principle: Inter-node communication with negligible overheads.



low-cost, low-power & fit-for-purpose performance

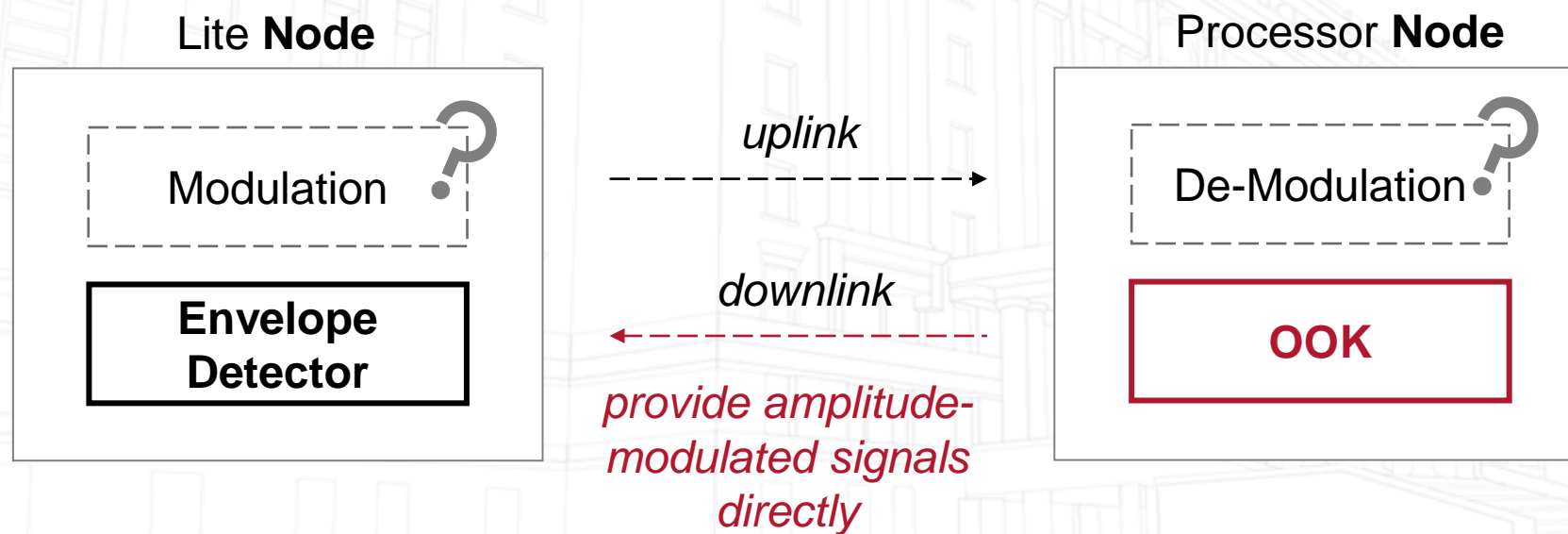
Design Concern: Lite node is RFBUS slave, thus must listen to the RFBUS continuously.

Ultra-low power RX at lite node



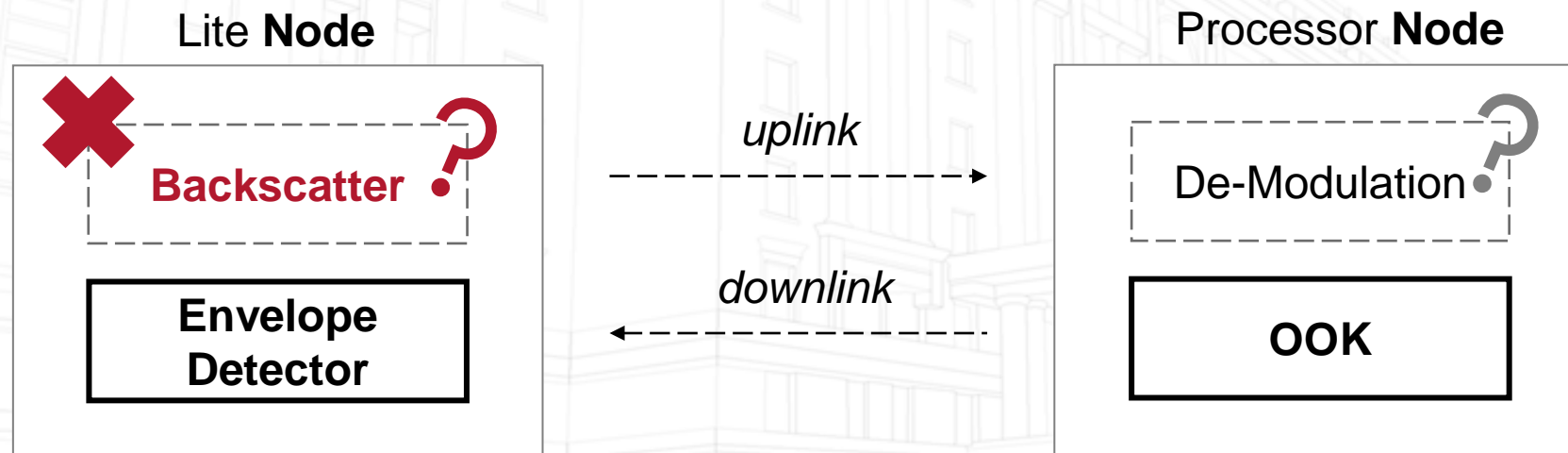
Design Concern: Envelope detector demodulates according to RF signal energy.

OOK TX at processor node

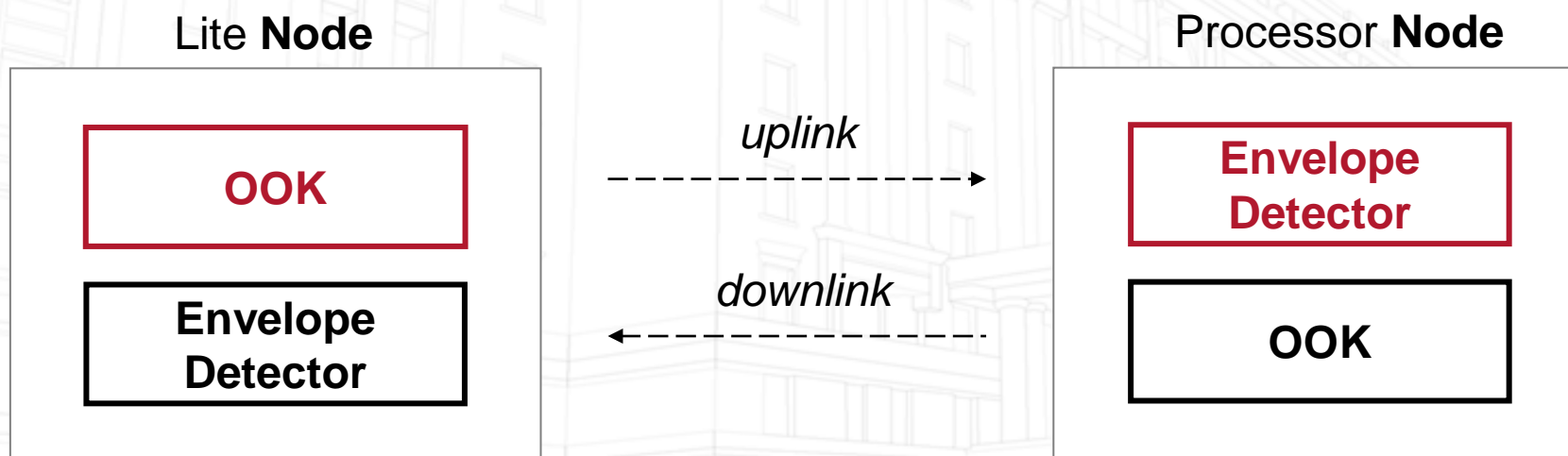


Insight: Backscatter is not suitable for RFBUS.

- Backscatter calls for sensitive RX → *deployment costs of processor node increased.*
- Backscatter causes unequal two-way communication range → *inefficient energy utilization.*



A **symmetric, bidirectional** RF chain
between the processor and lite nodes.

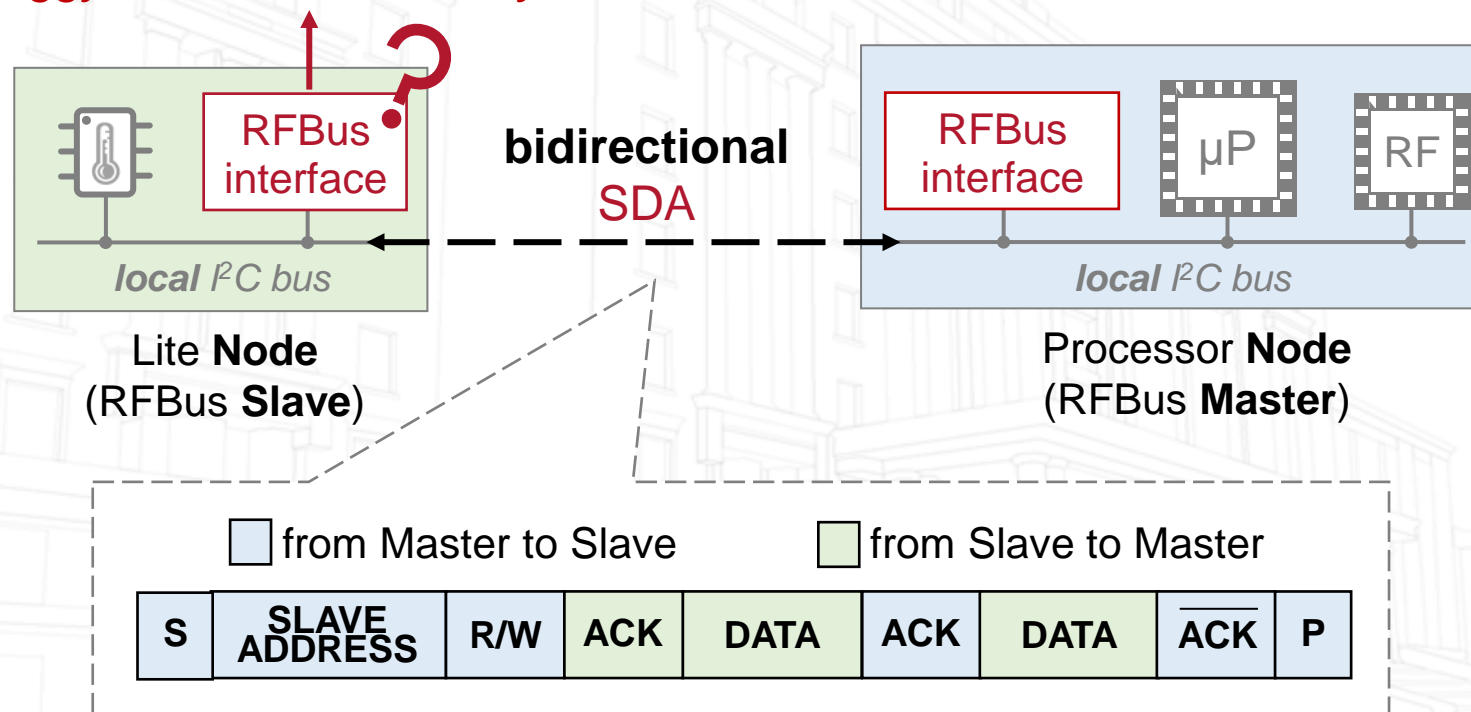


identical two-way communication range

Problem: Rhythm of bidirectional I²C communication is **agnostic** to lite node.

Lite node asks:

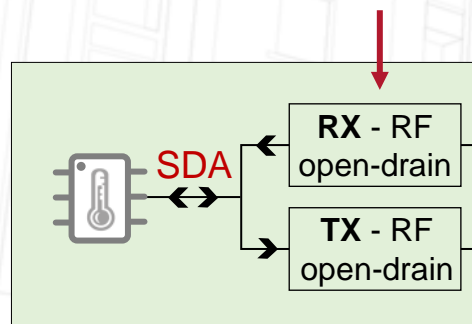
'Should I piggyback current SDA symbol?'



Solution: Frequency division multiplexing without protocol parsing at lite node.

RFBUS replies:

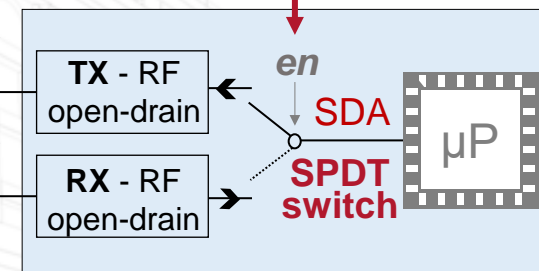
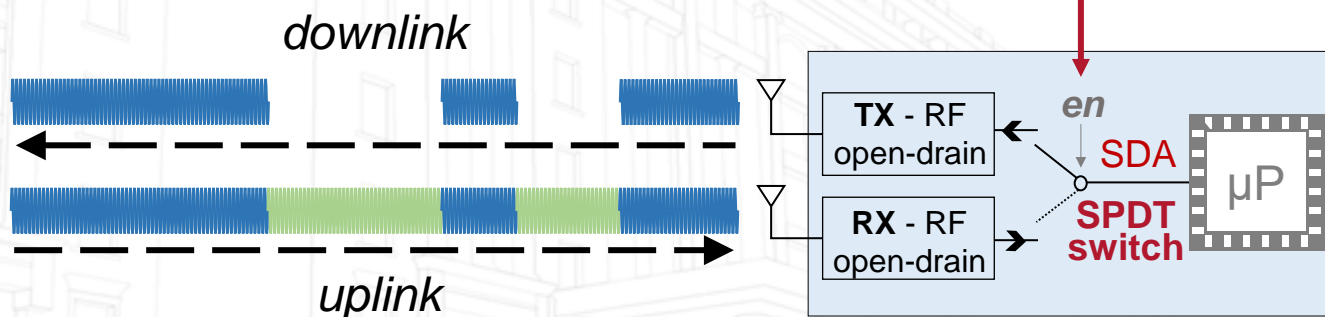
'Don't worry, just piggyback all SDA symbols.'



Lite Node
(RFBUS Slave)

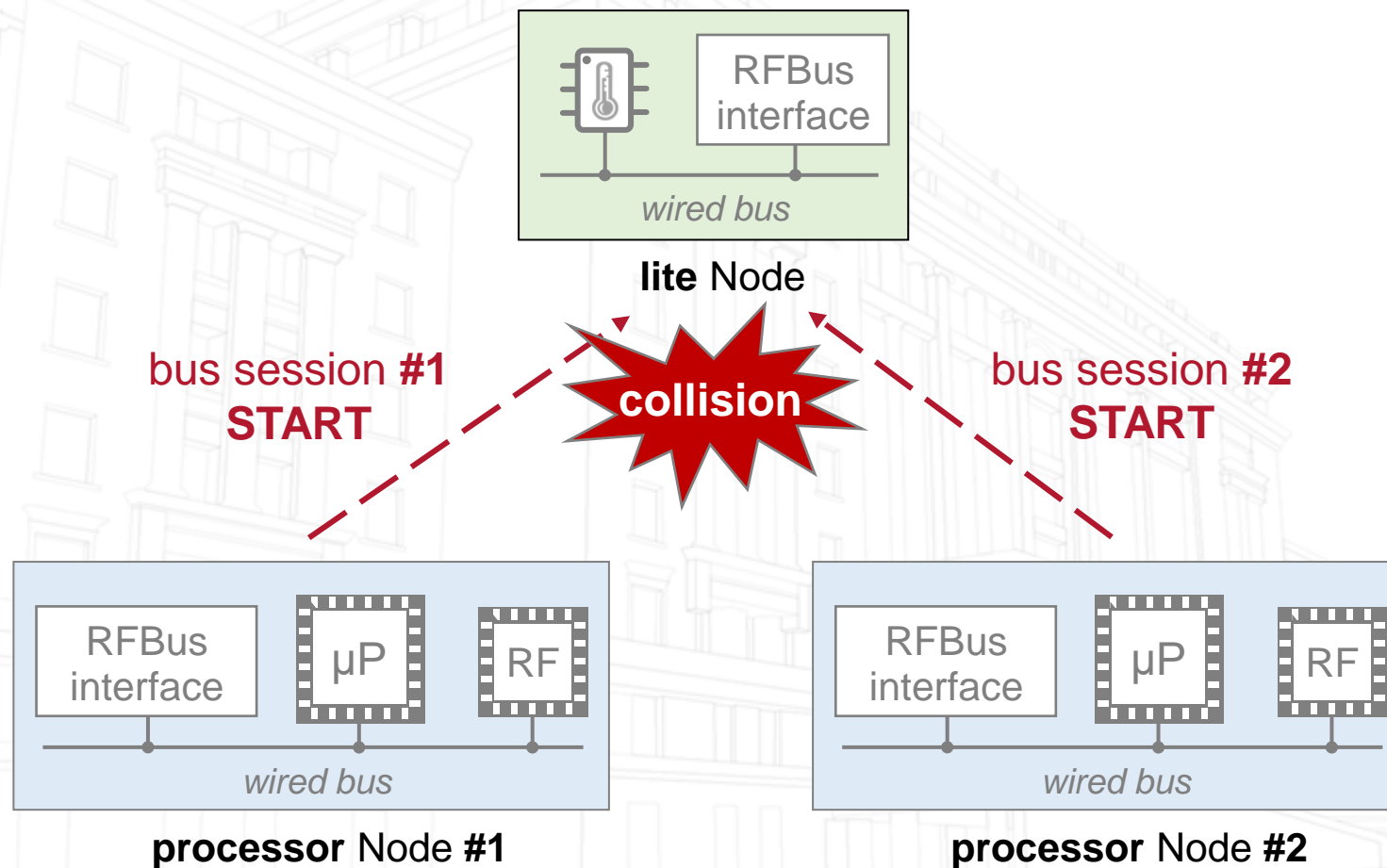
RFBUS commands:

'Hey, only load SDA symbols generated by sensor chip to your local SDA!'

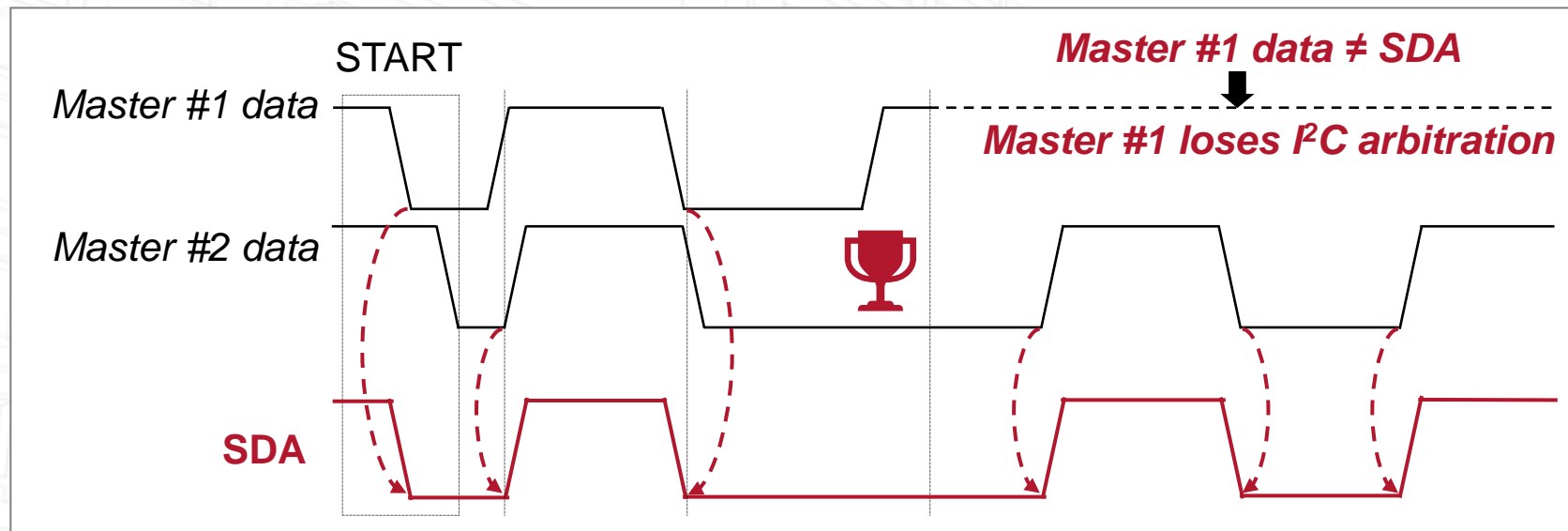


Processor Node
(RFBUS Master)

Problem: Bus collisions among multiple processor nodes.

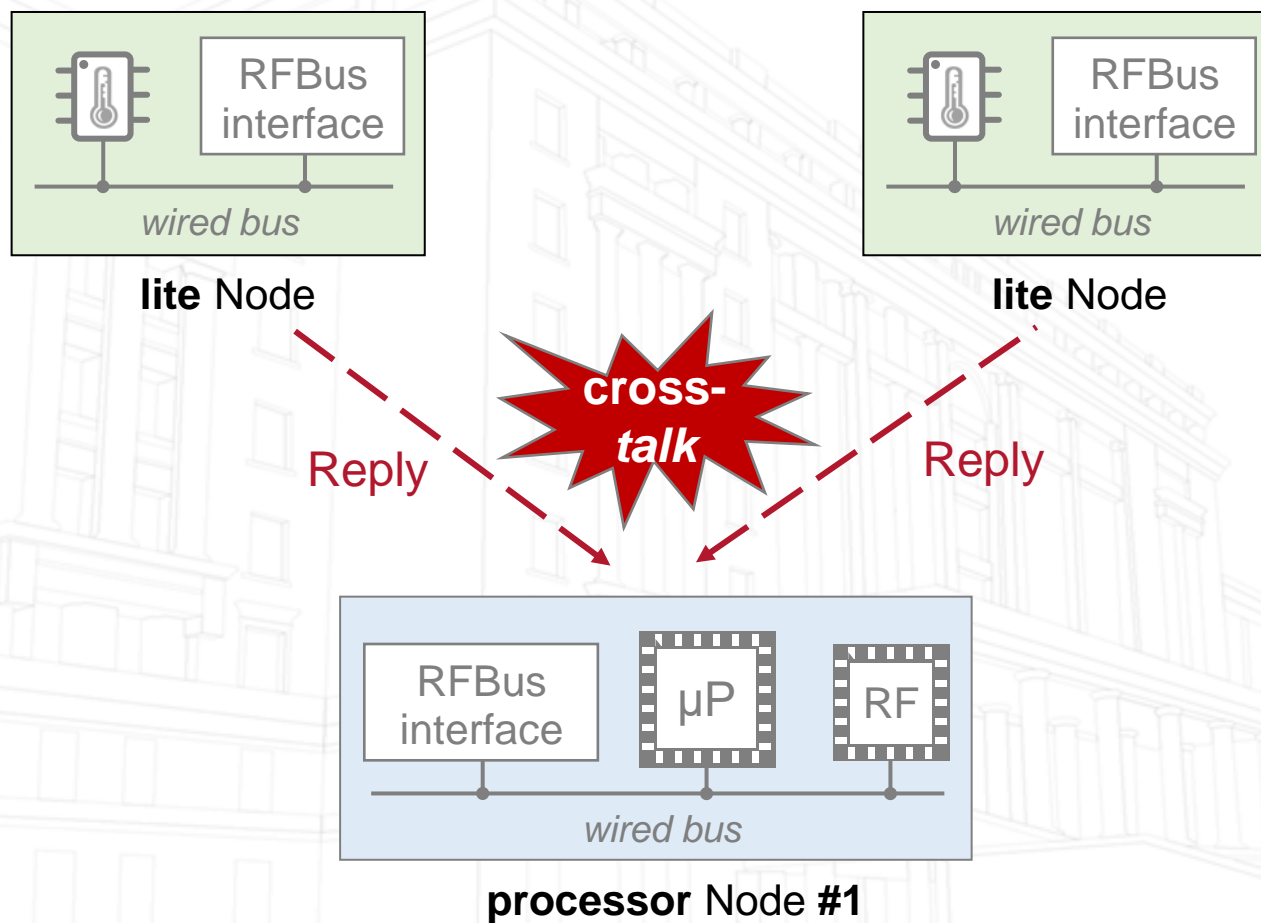


Naturally resolved by I²C arbitration.



The principle of I²C arbitration shown in a two-master case.

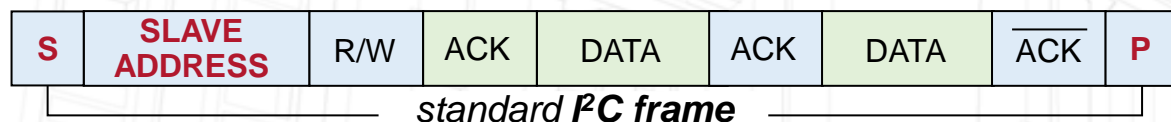
Problem: Cross-talk among multiple lite nodes.



Intuitive Solution: Control lite node's RF by parsing I²C protocol.

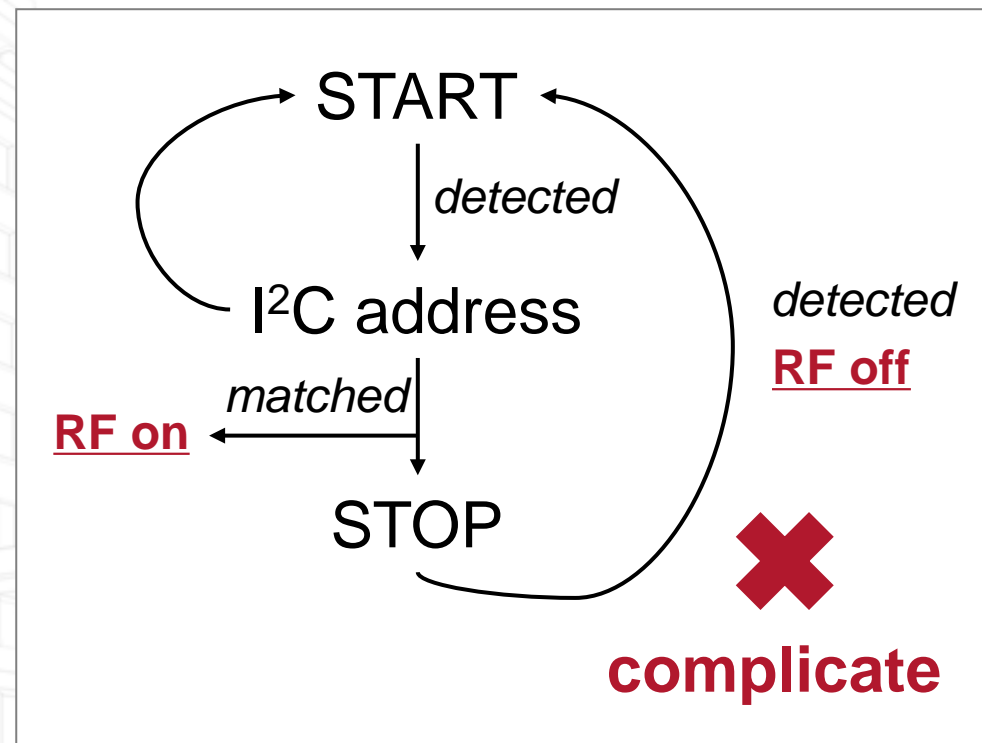
□ from processor node
to lite node

□ from lite node
to processor node

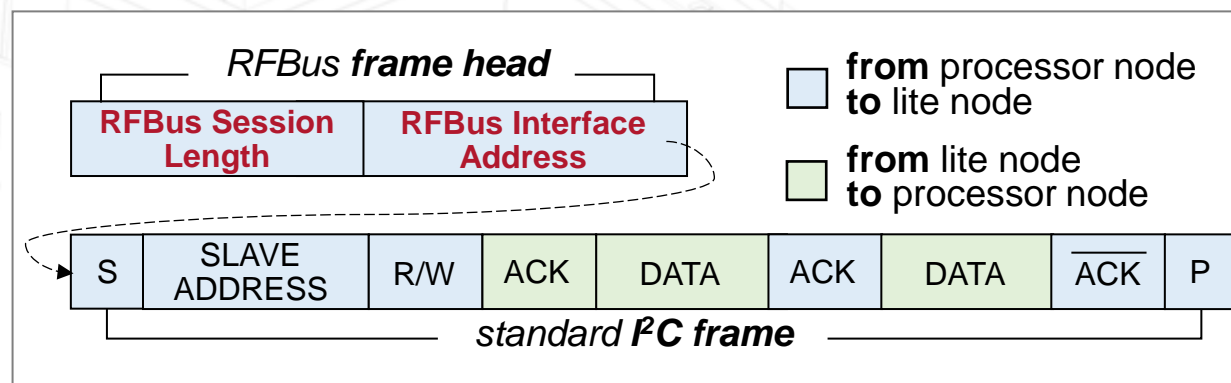


7-bit I²C address =
4-bit Manufacturer ID +
3-bit hardware-defined address

✗ Only $2^3 = 8$ usable addresses



Our Solution: Configure lite node's RFBus interface before each I²C session.



RFBus frame structure

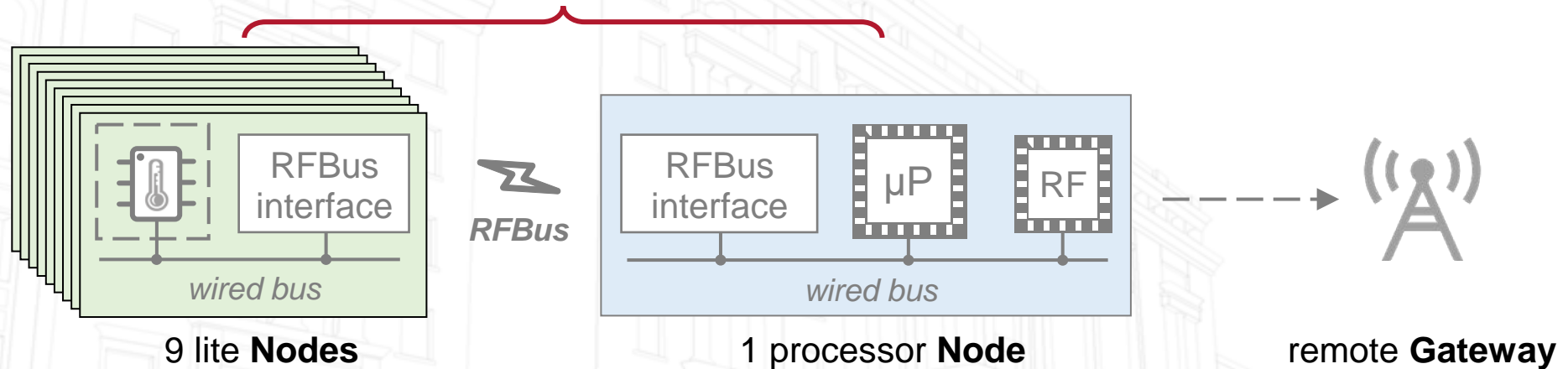
usable I²C address

$2^3 \times 2^8 = 2048$ usable addresses in RFBus

RFBus address

part1 RFBus Network

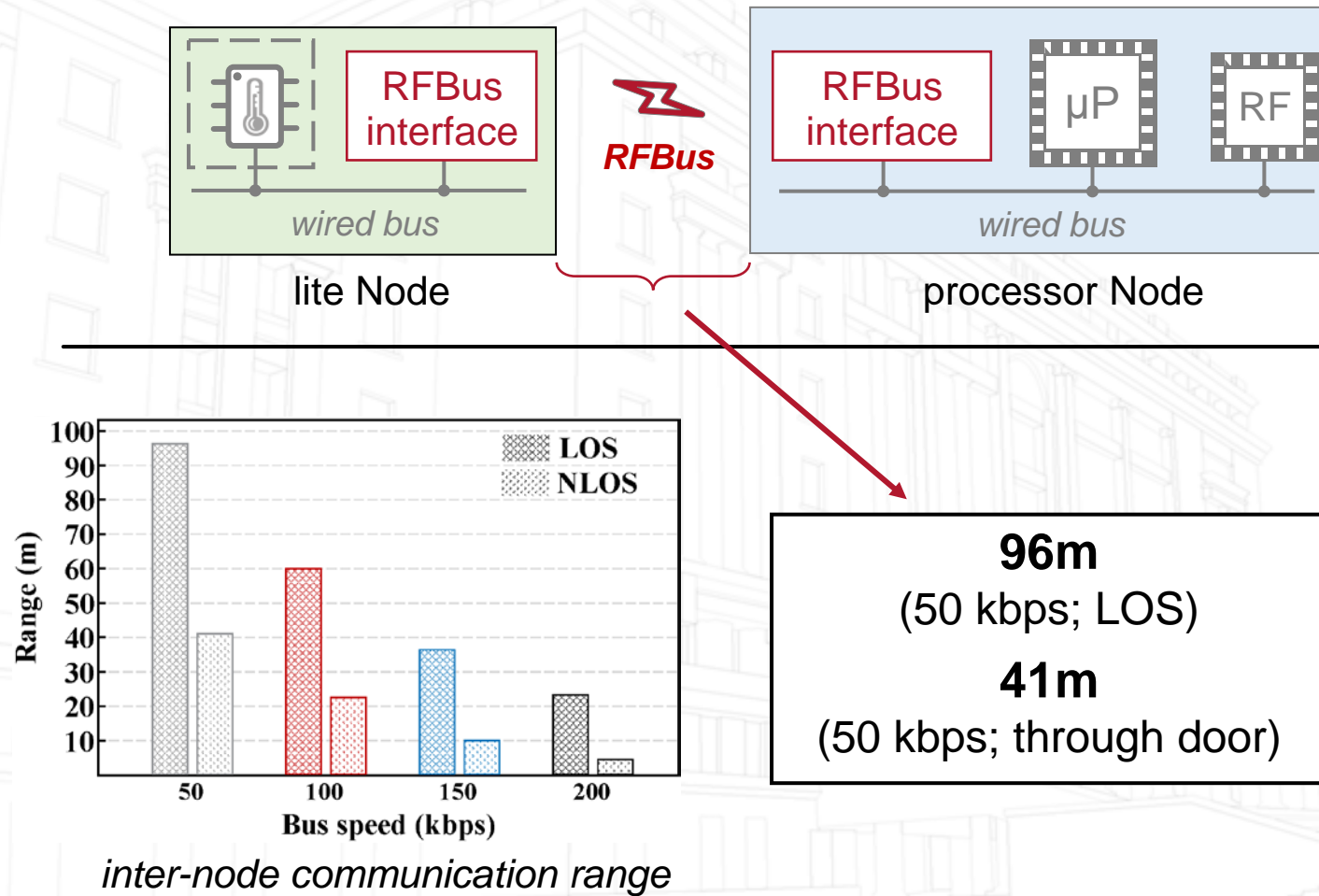
- > Inter-Node Communication Range
- > Network Throughput
- > Task Throughput



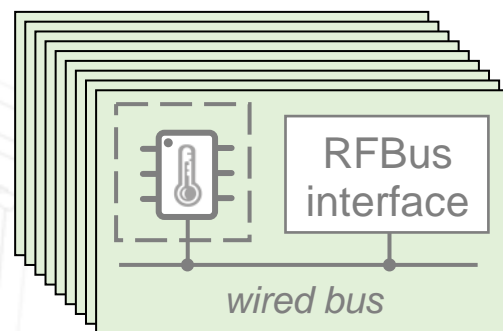
part2 Processor-Sharing Architecture

- > Power Consumption
- > Response Time
- > Manufacturing Cost

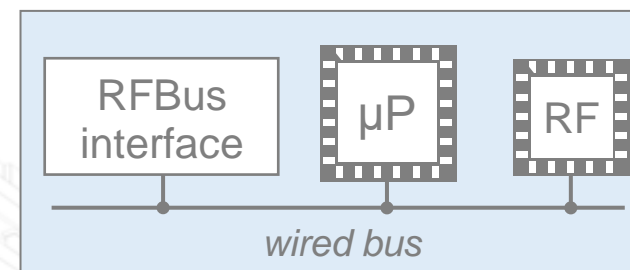
Setup: passive RX ; 17dBm+2dBi TX



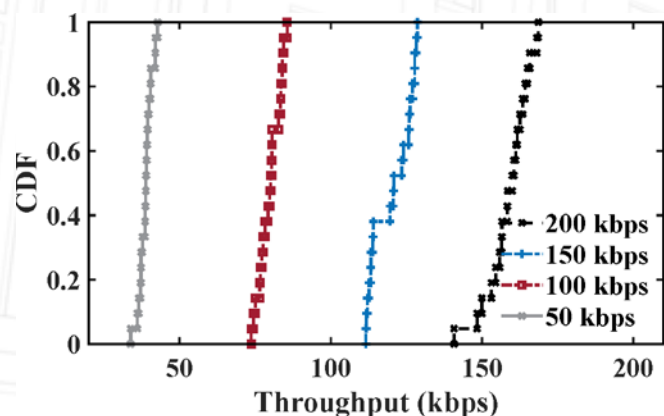
Evaluation > RFBus Network > Network and Task Throughput



9 lite Nodes



1 processor Node



Limited by the **conducting frequency** of RF Schottky diode

168 kbps (max)
network throughput

Strategy: querying

50 kbps	100 kbps	150 kbps	200 kbps
169 Hz	339 Hz	452 Hz	678 Hz

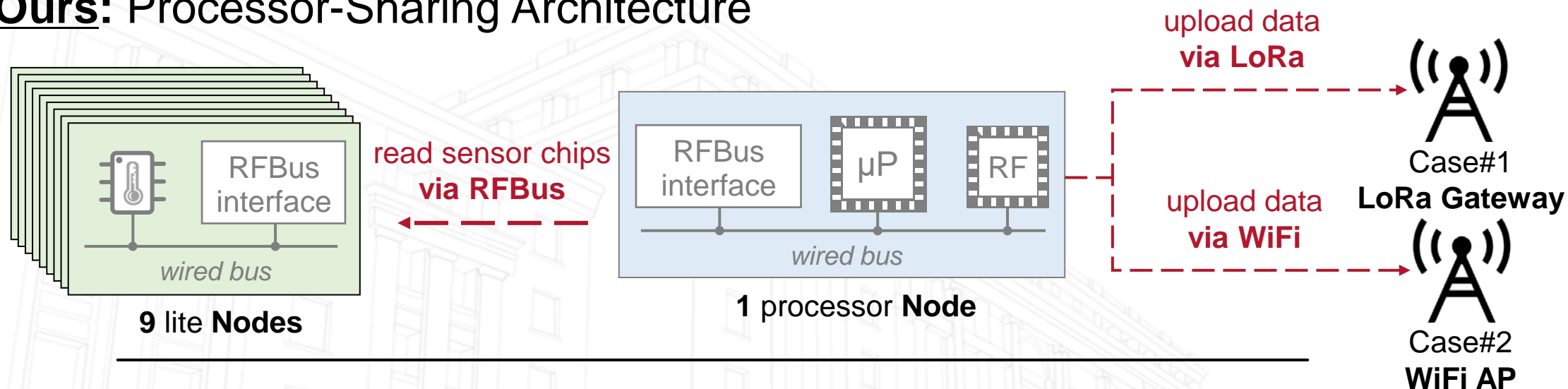
Strategy: polling

50 kbps	100 kbps	150 kbps	200 kbps
218 Hz	403 Hz	550 Hz	826 Hz

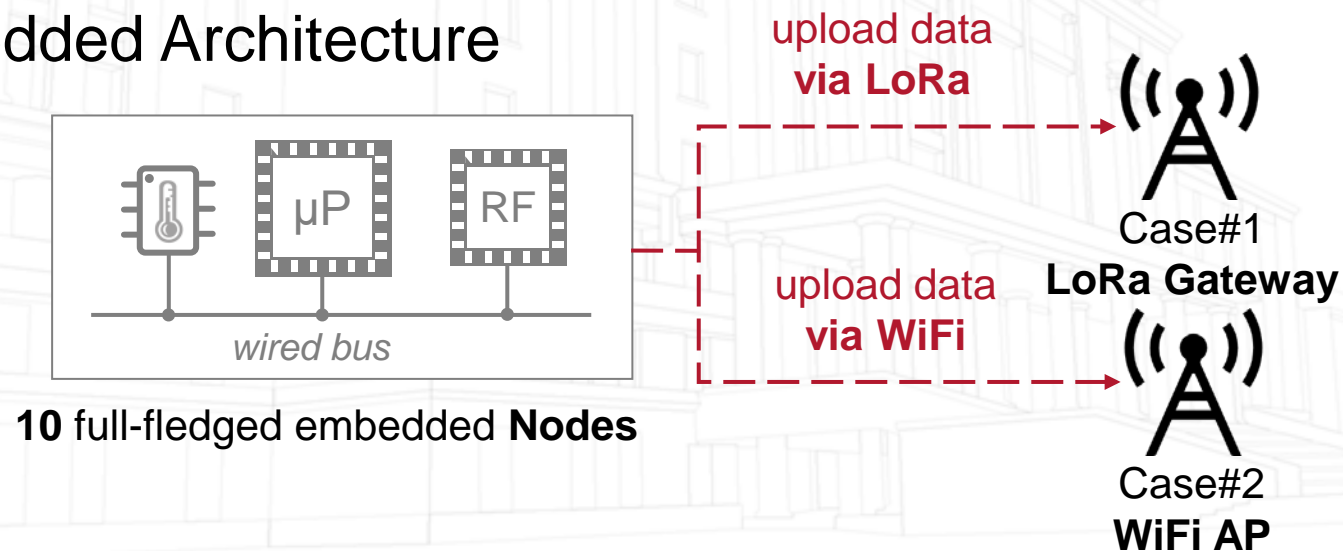
A processor node reads these **30 sensor chips** up to **826 times** in pipeline within **1 second**.

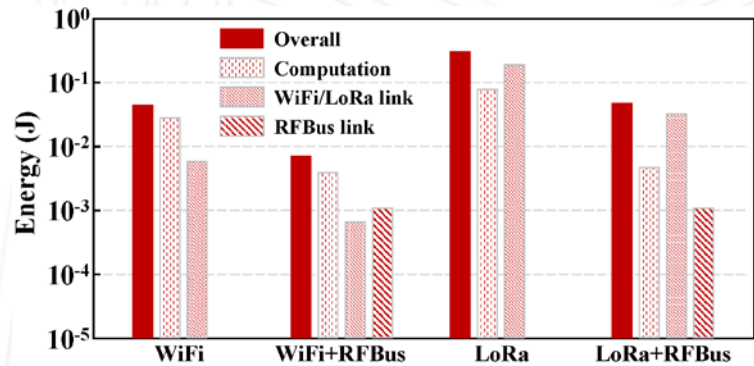
24.8 kHz (max)
task throughput

Ours: Processor-Sharing Architecture



Benchmark: Embedded Architecture



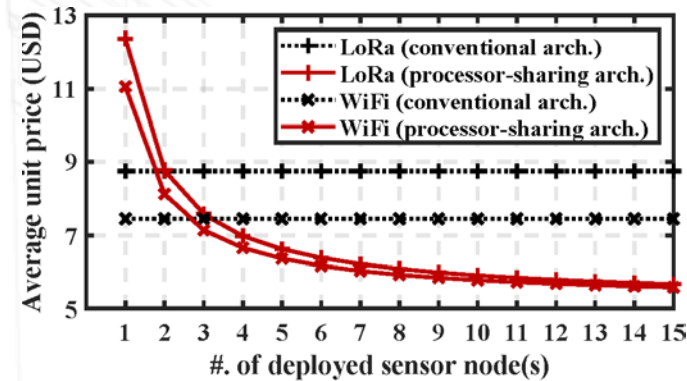


Node's power consumption compared with:

- embedded **WiFi** node **6.09** × ↓
- embedded **LoRa** node **6.69** × ↓

WiFi		LoRa	
embedded architecture	P.S. architecture	embedded architecture	P.S. architecture
364 ms	241 ms	1029 ms	300 ms

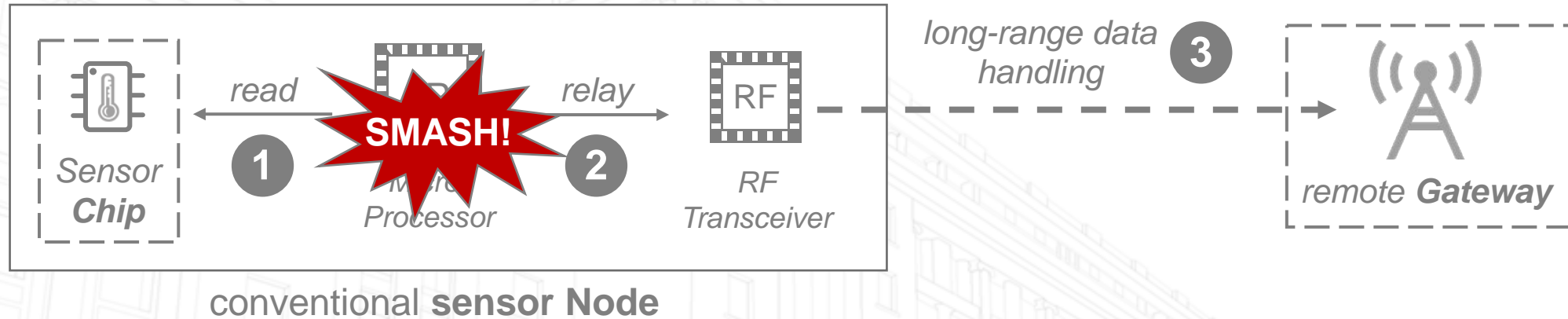
Response time comparison.



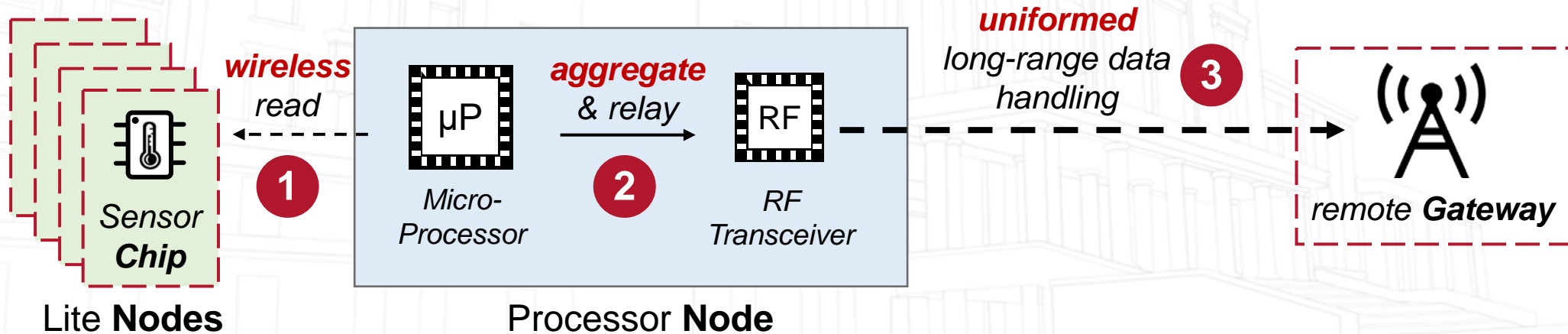
Node's manufacturing cost compared with:

- embedded **WiFi** node **23.5%** ↓
- embedded **LoRa** node **33.5%** ↓

Embedded Solution:



Processor-Sharing Solution:



Processor-Sharing Internet of Things Architecture for Large-scale Deployment

Qianhe Meng

qianhe@std.uestc.edu.cn

more at <https://mqhyes.github.io/>