pTunes: Runtime Parameter Adaptation for Low-power MAC Protocols

Marco Zimmerling, Federico Ferrari, Luca Mottola[*], Thiemo Voigt[*], Lothar Thiele

Computer Engineering and Networks Lab, ETH Zurich

[*]Swedish Institute of Computer Science (SICS)

Configuring a MAC Protocol is Not Easy

Application Requirements

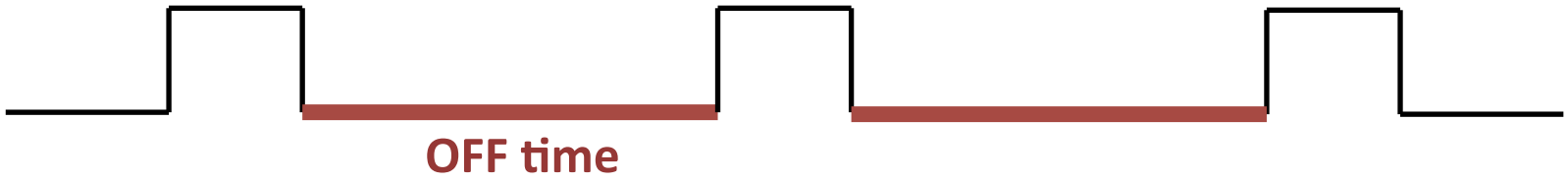
End-to-end Reliability

Network Lifetime

End-to-end Latency

?

Low-power MAC protocol



A Real-world Example

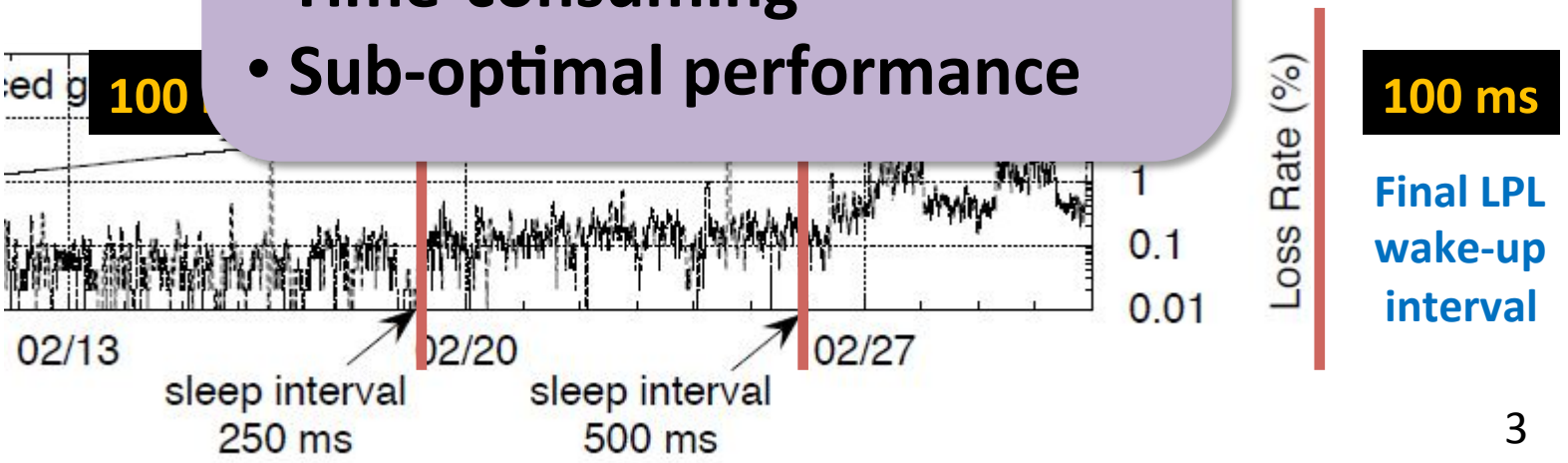
Adaptive Control Application

Ceriotti et al., *IPSN'11*

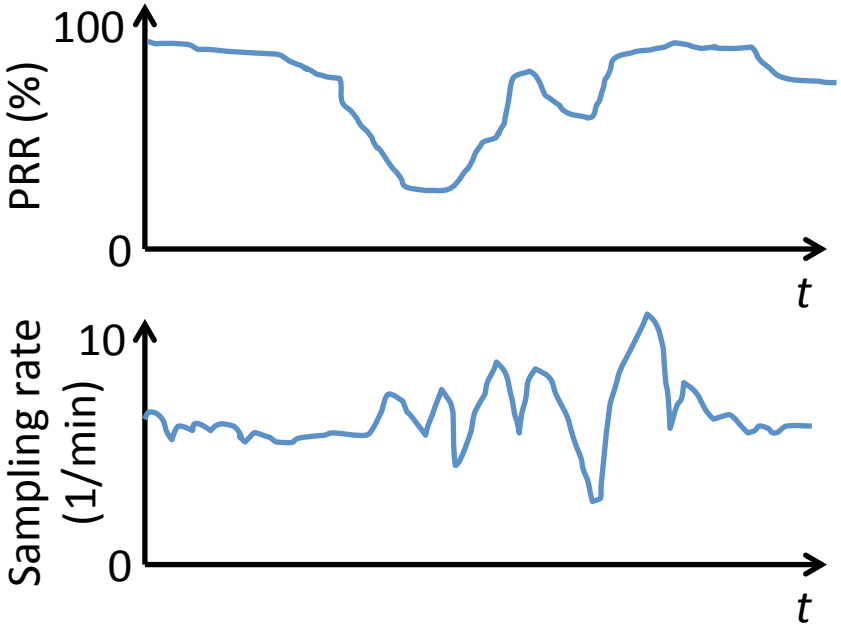
Current practice:
- Experience
- Field trials
- Over-provision

- **Requires expert knowledge**
- **Deployment-specific**
- **Time-consuming**
- **Sub-optimal performance**

TinyOS LPL



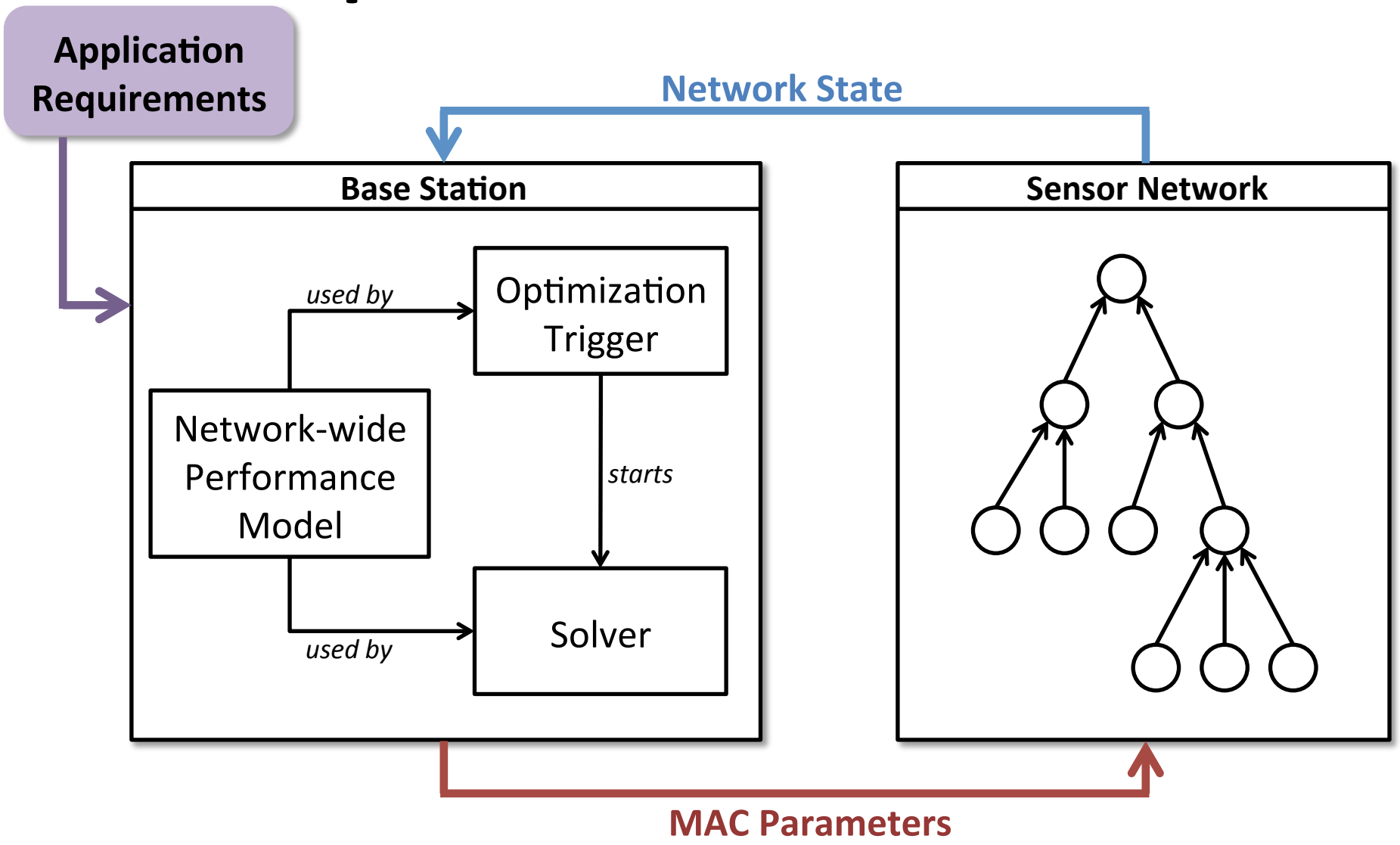
Adapting a MAC Protocol is Even Harder



Need to adapt *at runtime* to changes in

- Topology (*e.g.*, node failures, disconnects)
- Wireless link quality (*e.g.*, interference)
- Traffic load (*e.g.*, varying sampling rate)

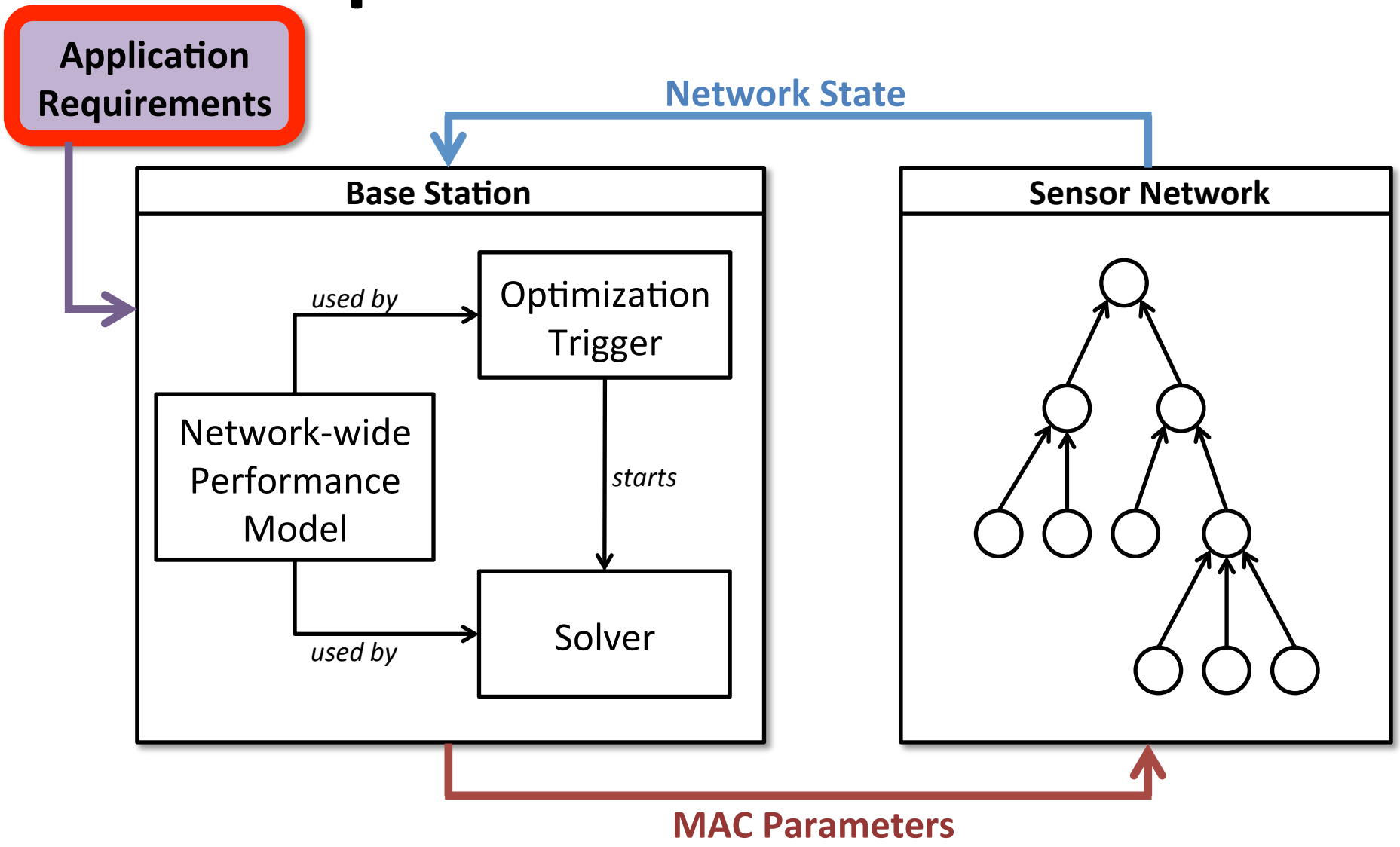
pTunes in a Nutshell



Contributions

1. pTunes framework for runtime adaptability of existing low-power MAC protocols
2. Flexible modeling approach
3. Efficient runtime support to “close the loop”

pTunes in a Nutshell



Application Requirements

- **pTunes** targets data collection scenarios
  - Tree routing
  - Low-power MAC
- Example requirements specification

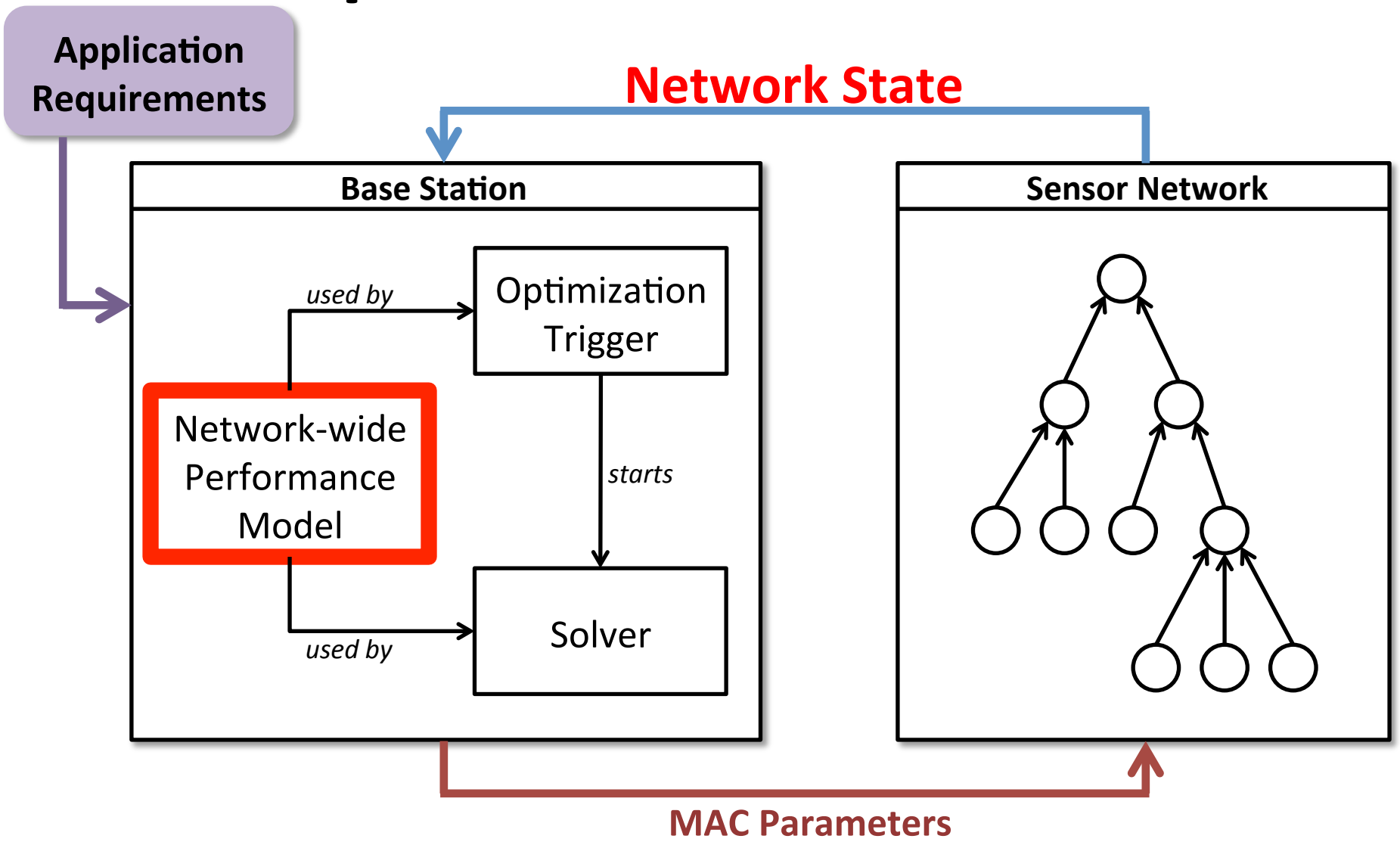
Maximize: ***Network lifetime***

Subject to: ***End-to-end reliability*** greater than **95 %**

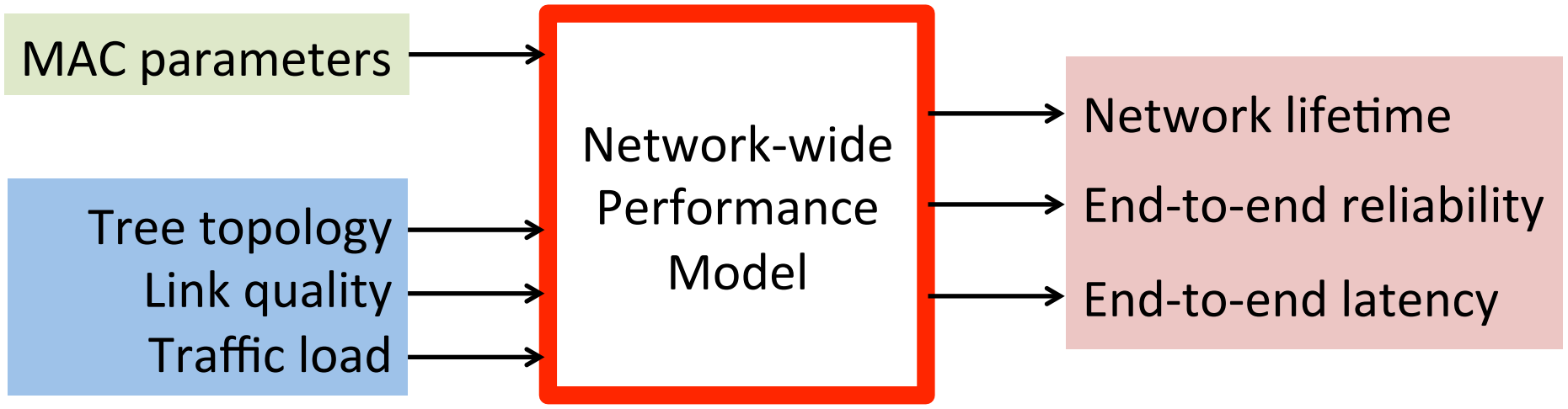
End-to-end latency below **1 second**

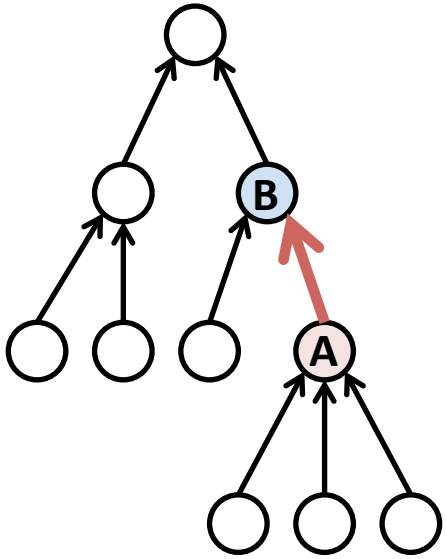
pTunes determines *at runtime* MAC parameters whose performance meets the requirements

pTunes in a Nutshell



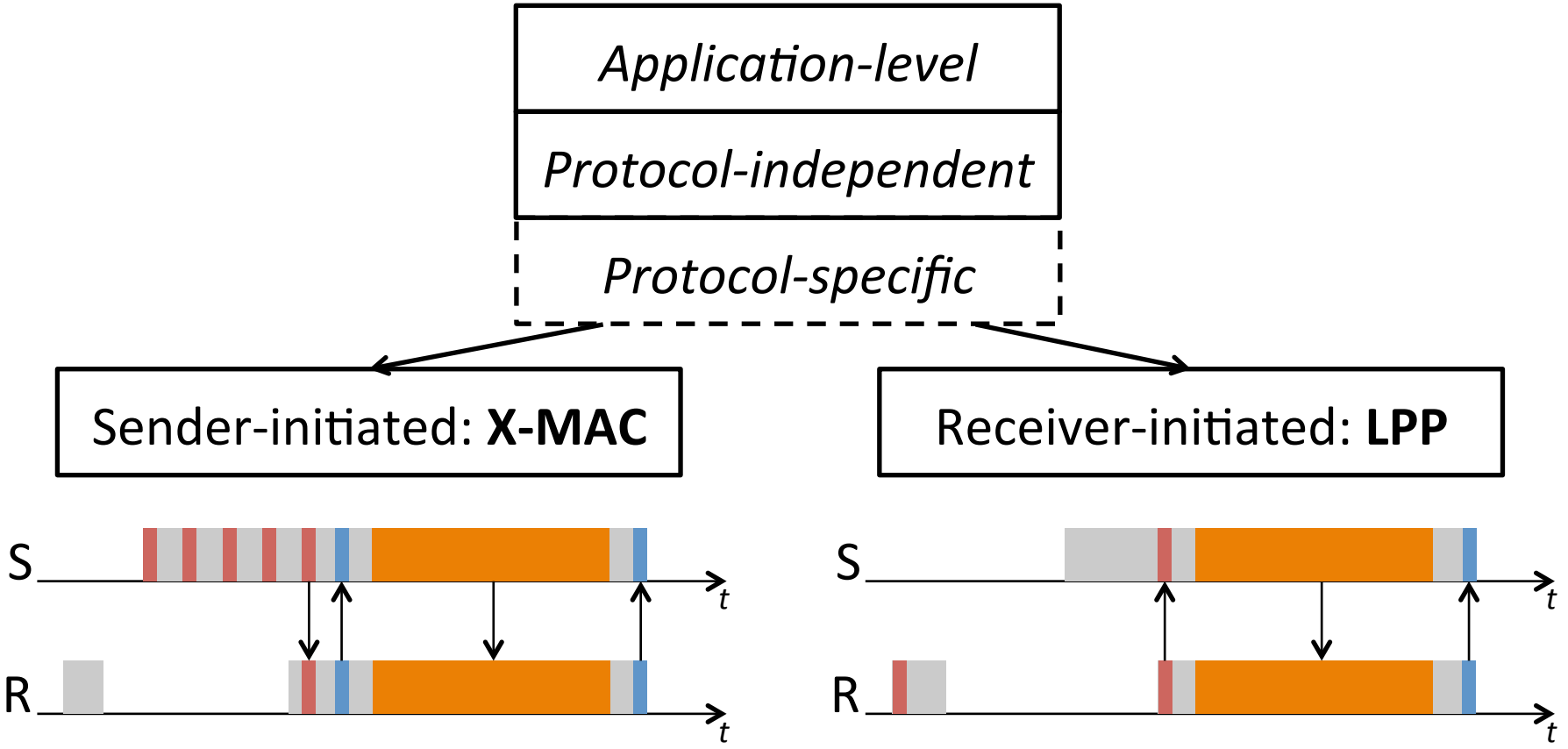
Network-wide Performance Model





Using the model, pTunes can *predict* how changes in the MAC parameters affect performance

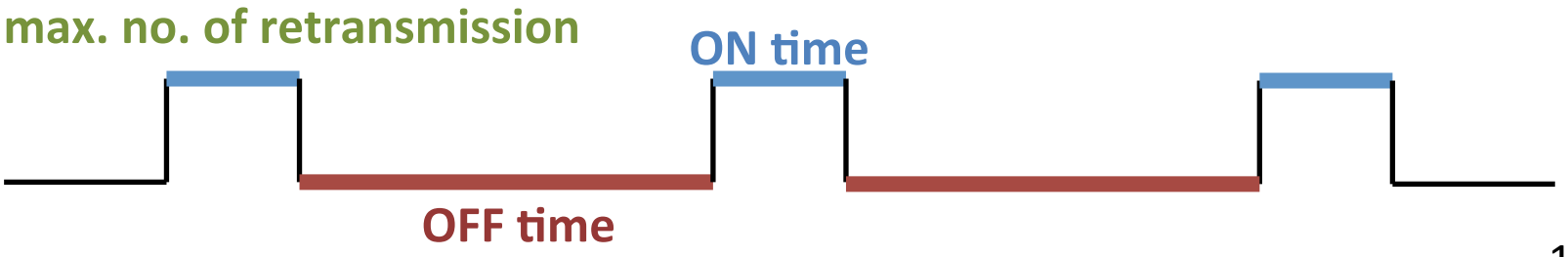
Layered Modeling Approach



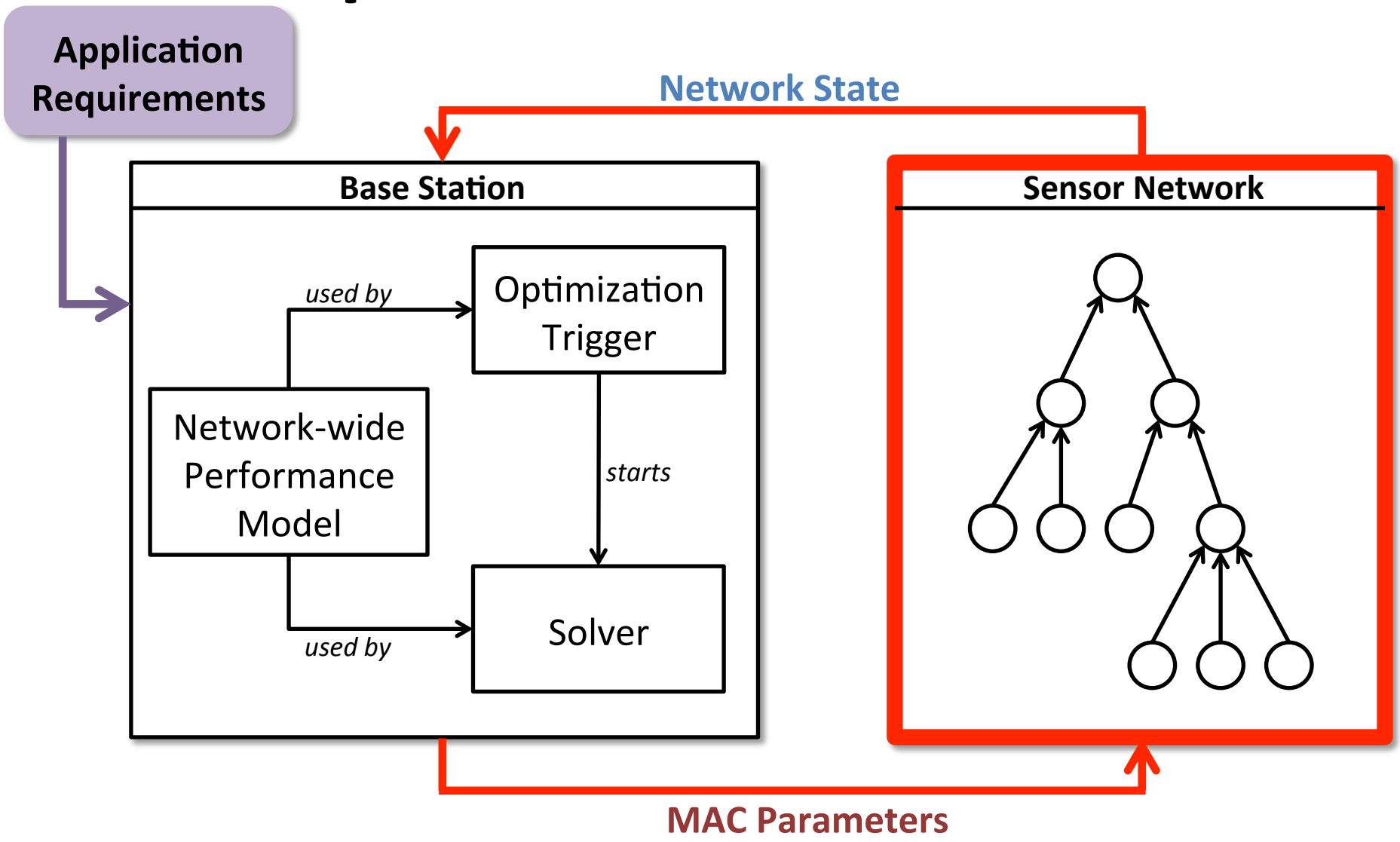
Only **the lowest layer must be changed to adapt the model in pTunes to a given MAC protocol**

Layering in Action: X-MAC End-to-end Reliability

Application-level (source-sink paths)	$R = \frac{1}{\lvert\mathcal{M}\rvert} \sum_{n \in \mathcal{M}} R_{\mathcal{P}_n} = \frac{1}{\lvert\mathcal{M}\rvert} \sum_{n \in \mathcal{M}} \left(\prod_{l \in \mathcal{P}_n} R_l \right)$
Protocol-independent (child-parent link)	$R_l = 1 - (1 - p_{s,l})^{N+1}$
X-MAC-specific (single transmission)	$p_{s,l} = \left(1 - (1 - p_l)^{(T_{on} - T_{str})/T_{it}}\right) \cdot p_l^2$



pTunes in a Nutshell

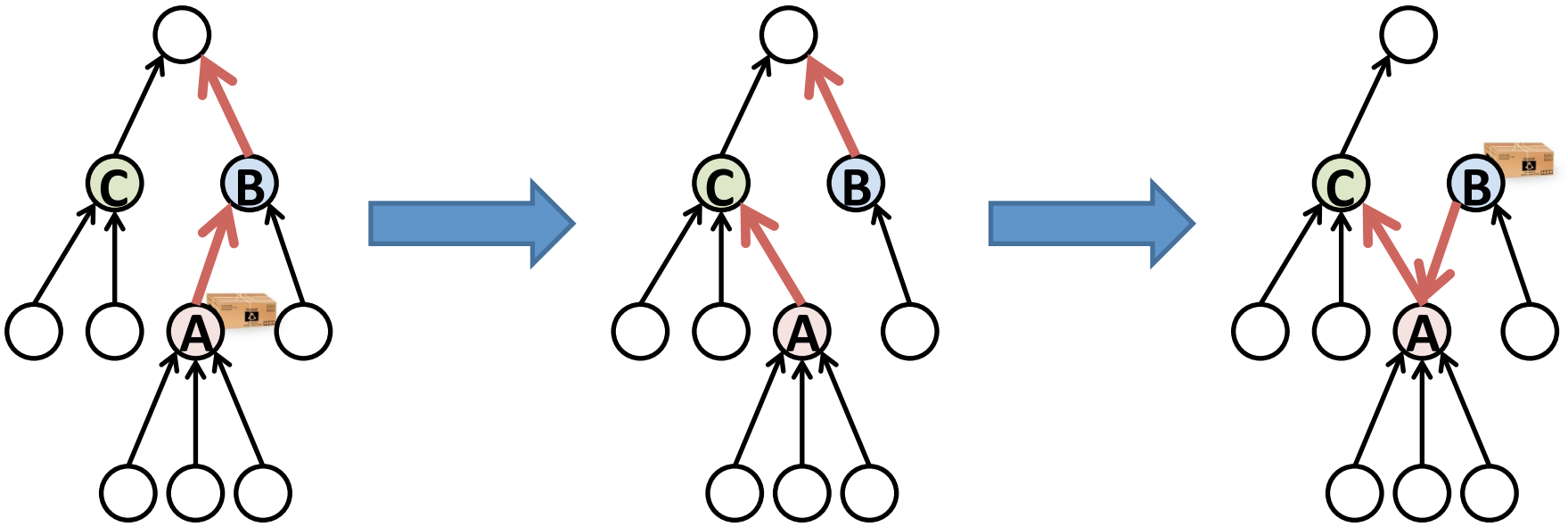


Communication Support to Close the Loop

- Piggybacking introduces a dependence on the ***rate*** and the ***reliability*** of application traffic
- Running a dissemination protocol concurrently may degrade the reliability of application traffic

Need to *decouple* network state collection and MAC parameter dissemination from application data traffic

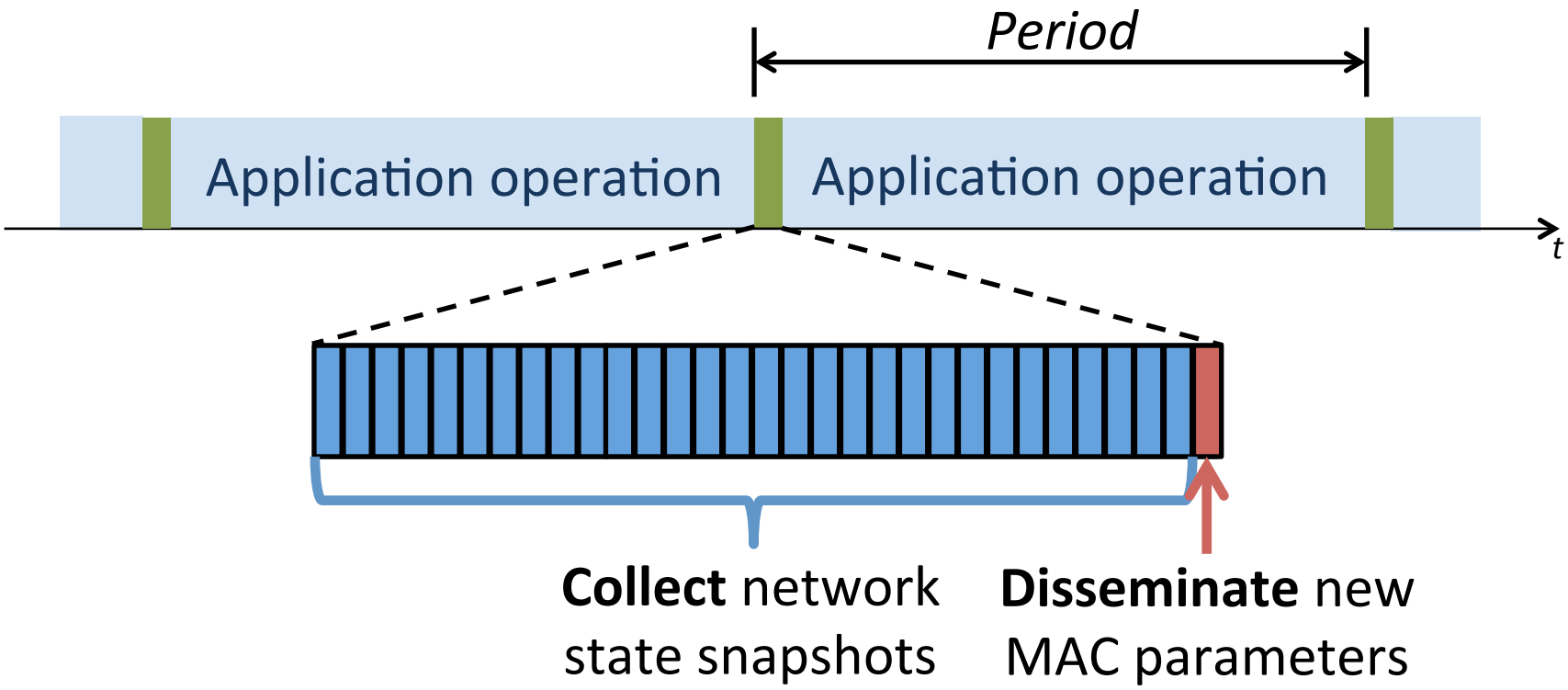
The Need for Consistency



Ⓐ reports Ⓑ and Ⓑ reports Ⓐ as parent, resulting in a loop that never existed for real

Need to collect *consistent* snapshots of network state from all nodes

Closing the Loop: Our Solution



- Phases consist of ***non-overlapping slots***, one for each node
- Each slot corresponds to a distinct ***Glossy network flood***

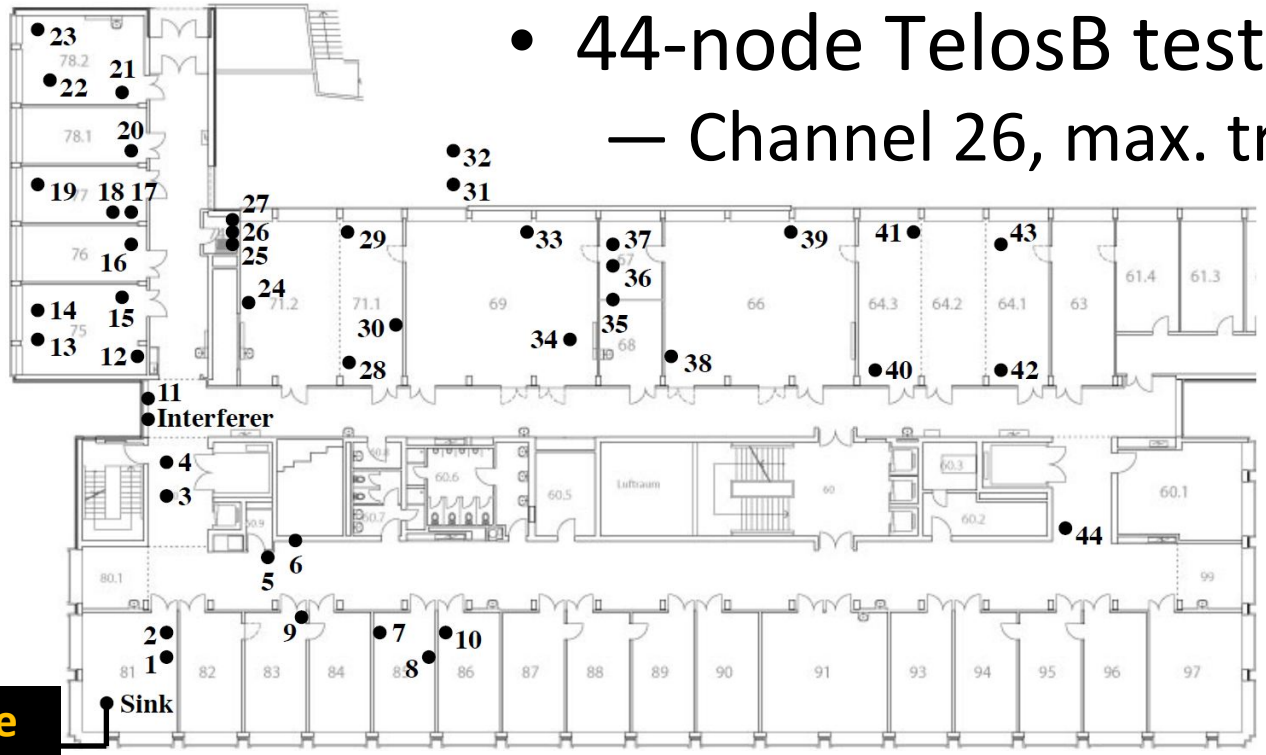
Our Solution Achieves

- ***Temporal decoupling from application***
- ***Timeliness***
  - Fast collection and dissemination
- ***Consistency***
  - Snapshots taken at all nodes at the same time
  - Simultaneous transition to new MAC parameters
- ***Energy efficiency*** *(44-node TelosB testbed)*

Period	Excess Radio Duty Cycle
1 minute	0.35 %
5 minutes	0.07 %

Testbed Evaluation: Setup

- Implementation
  - Sensor nodes: Contiki, Rime stack, X-MAC, LPP
  - Base station: Java, ECLiPSe



- 44-node TelosB testbed
  - Channel 26, max. transmit power

Testbed Evaluation: Methodology

- Metrics
  - ***Projected network lifetime:*** measured in software and computed based on 2000 mAh @ 3V batteries
  - ***End-to-end reliability:*** packet sequence numbers
  - ***End-to-end latency:*** packet time stamps
- Requirements specification

  Maximize: ***Network lifetime***

  Subject to: ***End-to-end reliability*** greater than **95 %**

  End-to-end latency below **1 second**
- Compare to static MAC parameters determined using **pTunes** and extensive experiments

Testbed Evaluation: Overview

1. Our X-MAC and LPP models are very accurate
2. pTunes provides higher bandwidth against increasing traffic and prevents queue overflows
3. pTunes achieves up to three-fold lifetime gains
4. Adaptation to traffic fluctuations
5. Adaptation to changes in link quality
6. Interaction with routing

Adaptation to Traffic Fluctuations

End-to-end reliability [%]
greater than 95 %

End-to-end latency [sec]
below 1 second

Network lifetime [days]

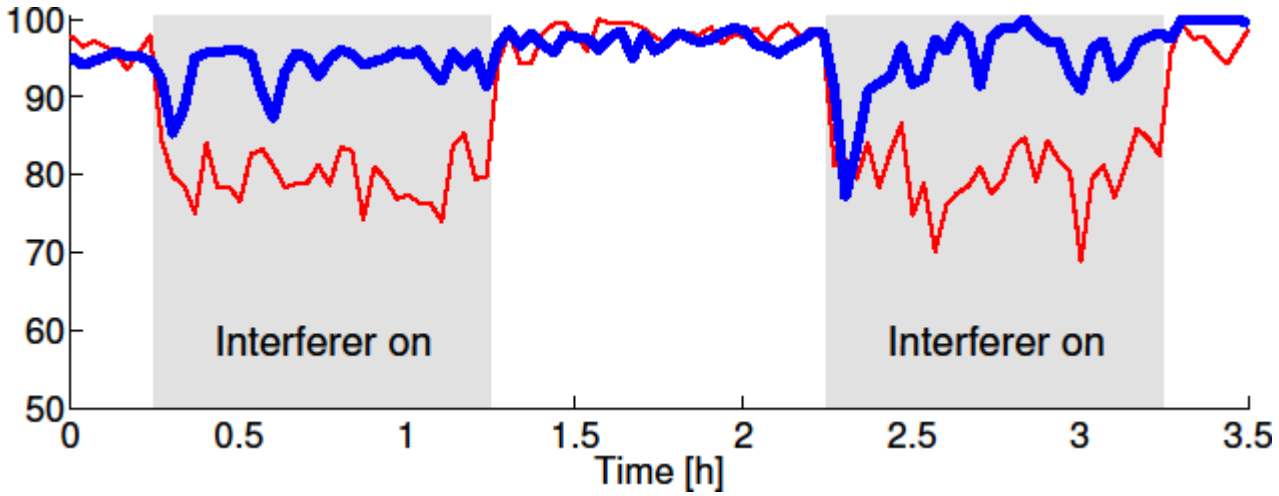
pTunes satisfies end-to-end requirements at high traffic while extending network lifetime at low traffic

Adaptation to Changes in Link Quality

Interferer jams using modulated carrier: 1 ms ON / 10 ms OFF

Adaptation to Changes in Link Quality

End-to-end reliability [%]
greater than 95 %



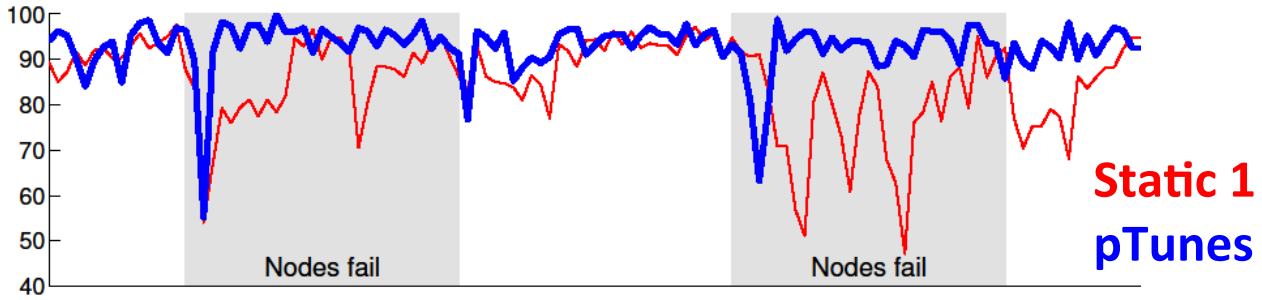
pTunes reduces packet loss by 80 % during periods of controlled wireless interference

Interaction with Routing

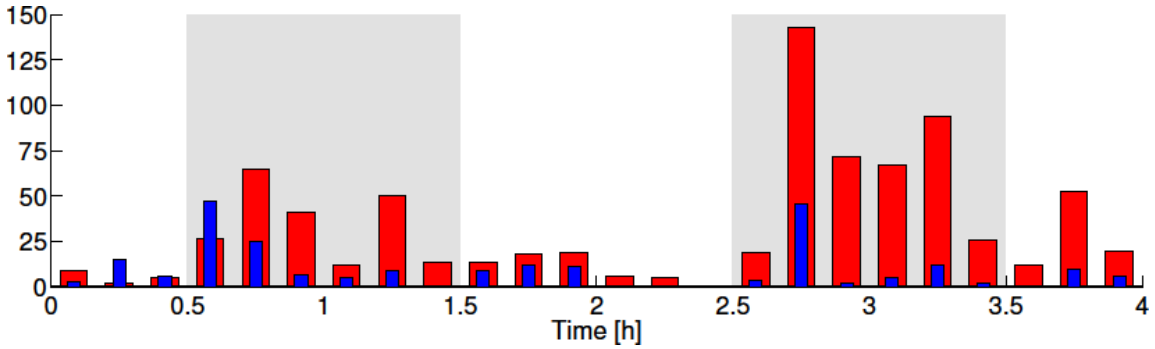
Turn off 8 nodes × within the sink's neighborhood

Interaction with Routing

End-to-end reliability [%]
greater than 95 %



Total number of parent switches



pTunes helps the routing protocol quickly recover from node failures, thus reducing packet loss by 70 %

Conclusions

- **pTunes** framework for runtime adaptability of existing low-power MAC protocols
- Flexible modeling approach
- Efficient system support to “close the loop”
- Testbed experiments demonstrate that
  - **pTunes** aids in meeting the requirements of real-world applications as the network state changes
  - **pTunes** eliminates the need for time-consuming, and yet error-prone, manual MAC configuration

FNSNF
Fonds National Suisse
Schweizerischer Nationalfonds
Fondo Nazionale Svizzero
Swiss National Science Foundation
NCCR MICS
National Competence Center In Research
Mobile Information and Communication Systems
nano-tera.ch
NET
Cooperating Objects
Network of Excellence
Swedish Foundation for Strategic Research