

eAFH: Informed Exploration for Adaptive Frequency Hopping in Bluetooth Low Energy

Valentin Poirot ^{1,2}, Olaf Landsiedel ^{1,2}

¹ *Kiel University, Germany*

² *Chalmers University of Technology, Sweden*

Motivation: Mobility with BLE

- We bring Bluetooth and Bluetooth Low Energy with us everywhere
 - Headphones, smart watches
- Bluetooth subject to interference
 - Stronger Wifi signals, concurrent Bluetooth transmissions
 - Deals with Adaptive Frequency Hopping and channel exclusion
- Interference changes as we move around!
 - Different Wifi channels affected

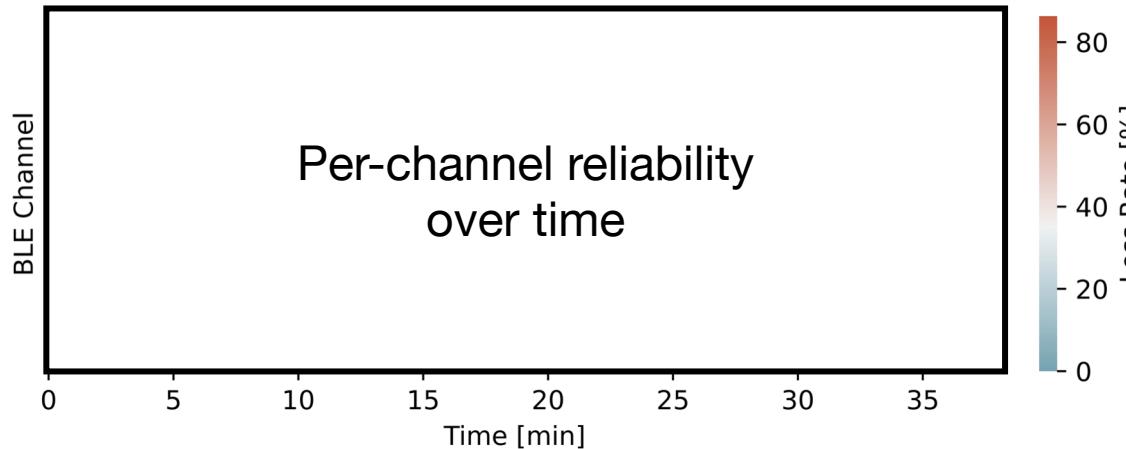


© Pixabay

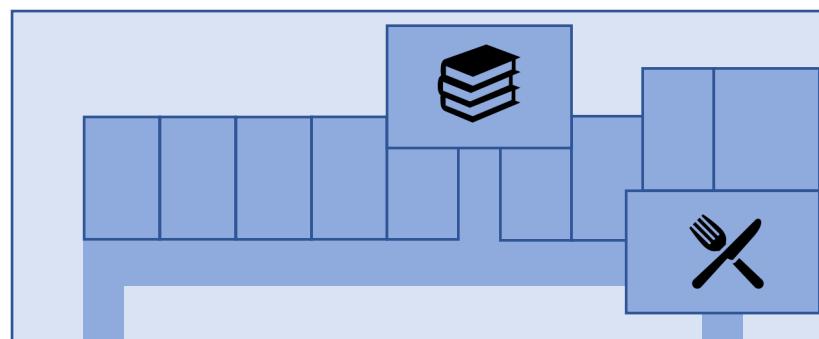


© Garmin

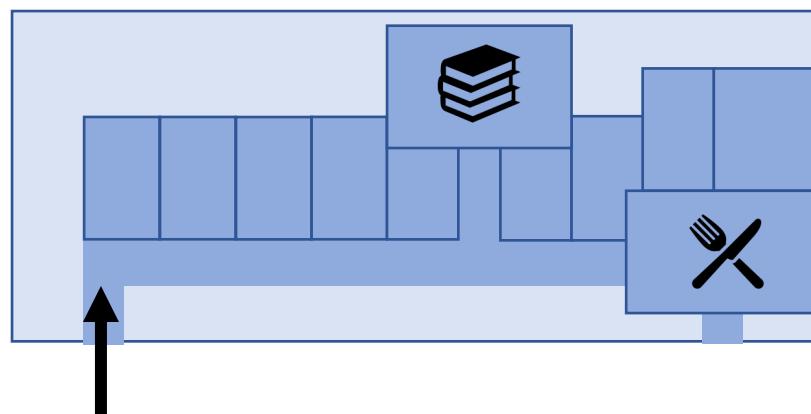
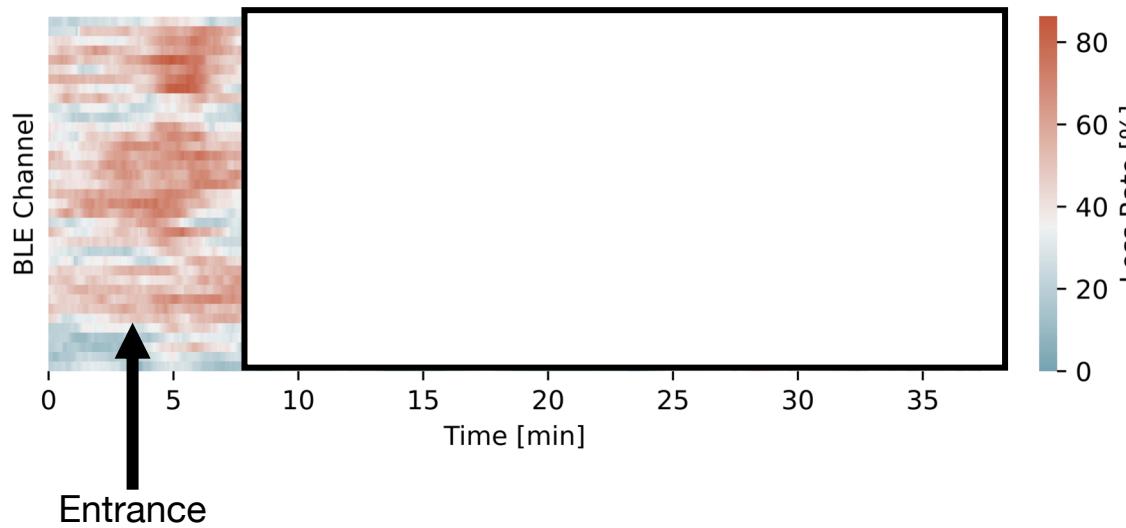
Motivation: Walking in a shopping mall



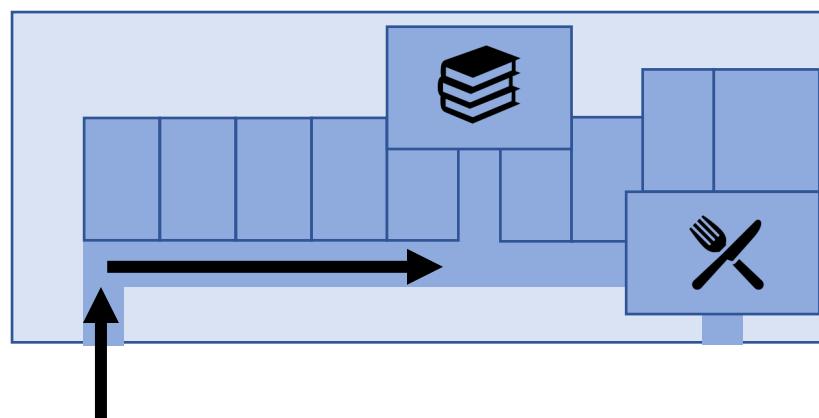
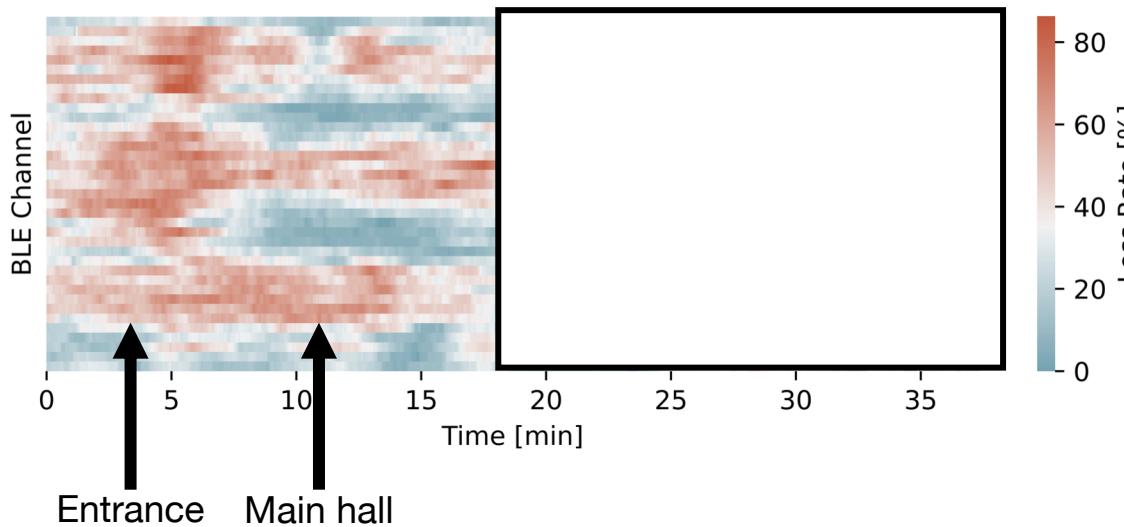
Nearby shopping mall
Mid December 2021
Saturday, ~ 14:00



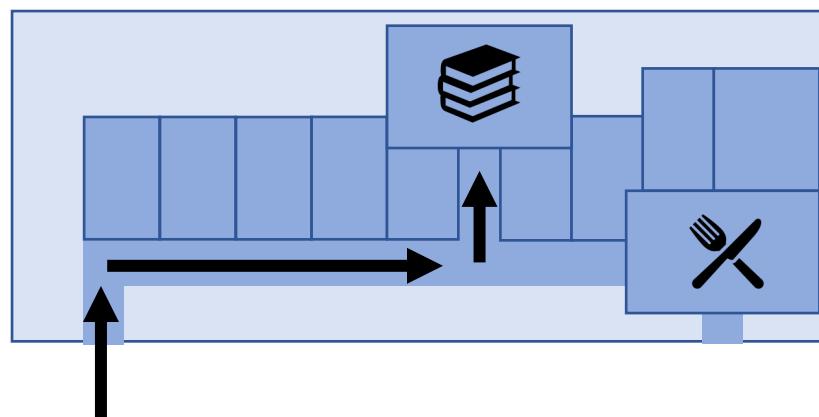
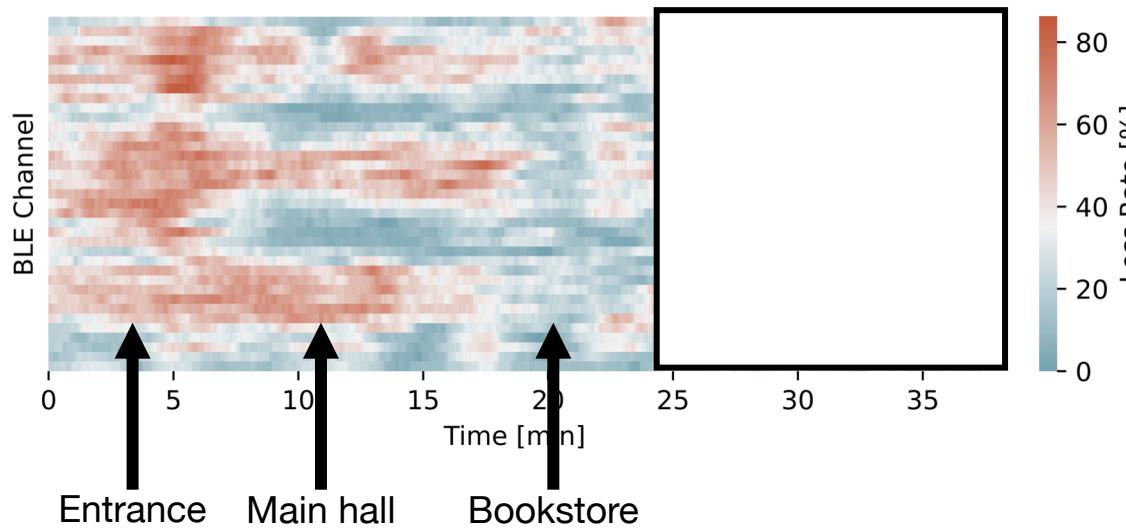
Motivation: Walking in a shopping mall



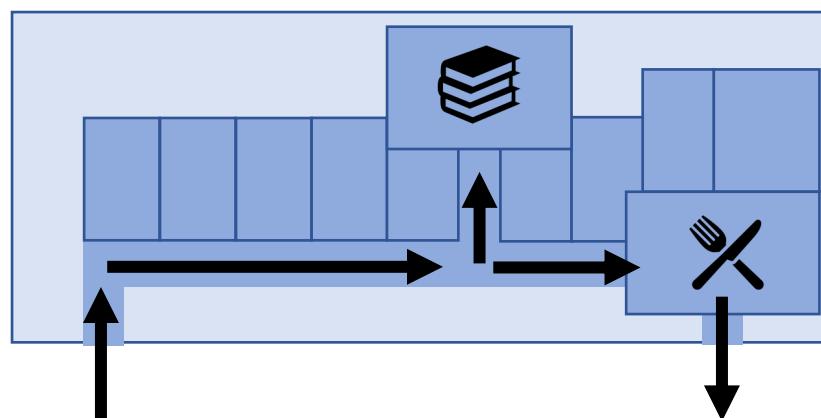
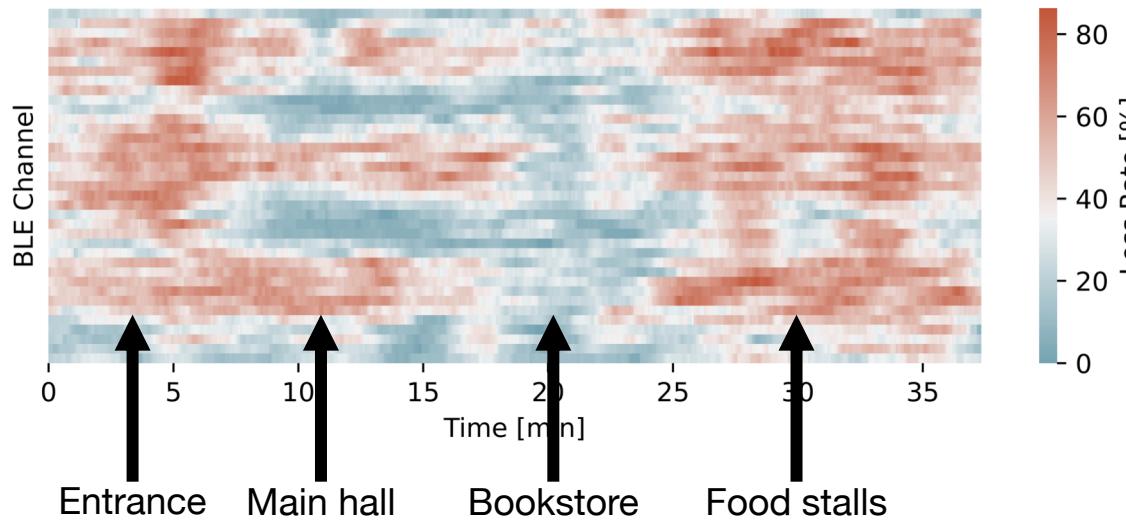
Motivation: Walking in a shopping mall



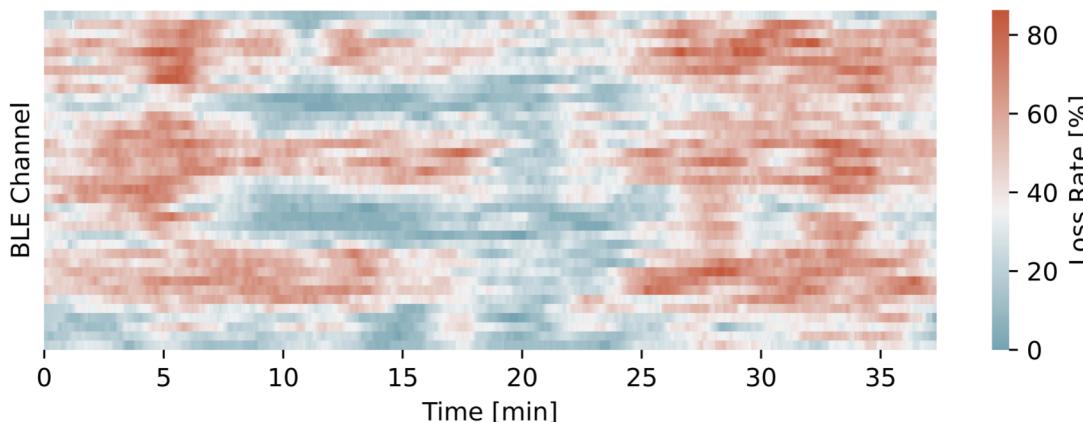
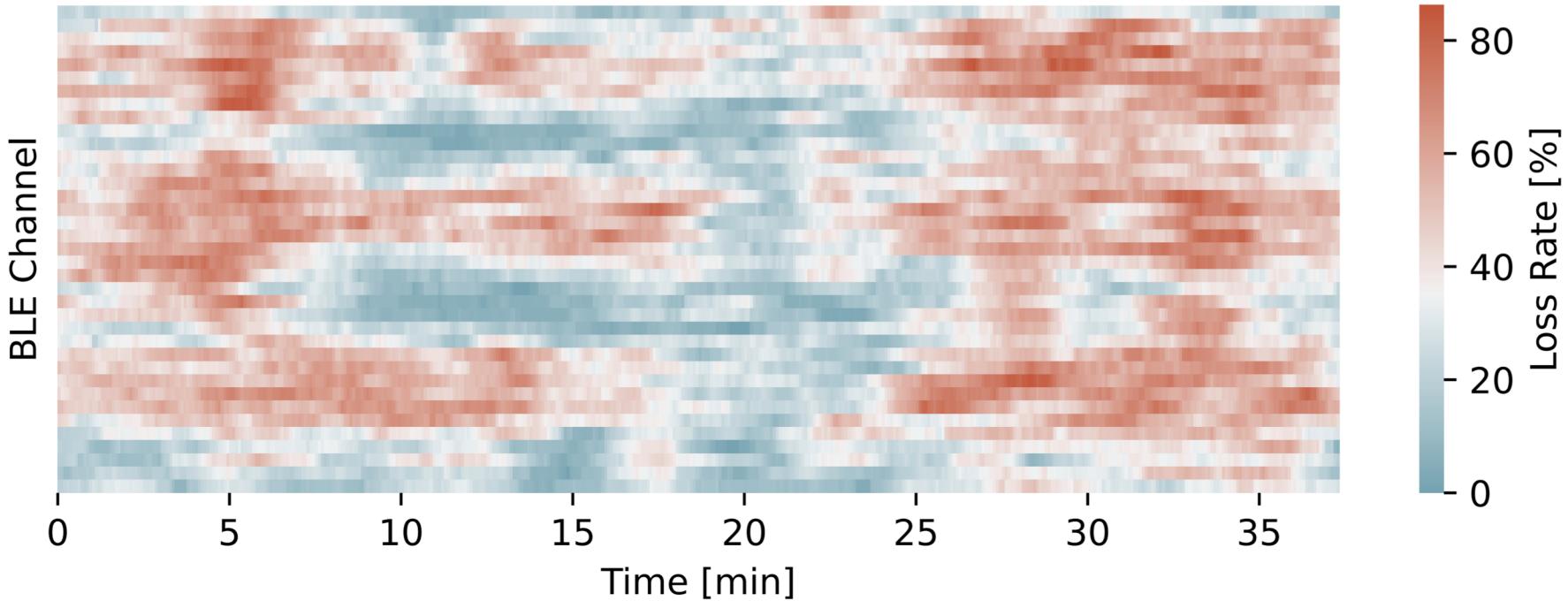
Motivation: Walking in a shopping mall



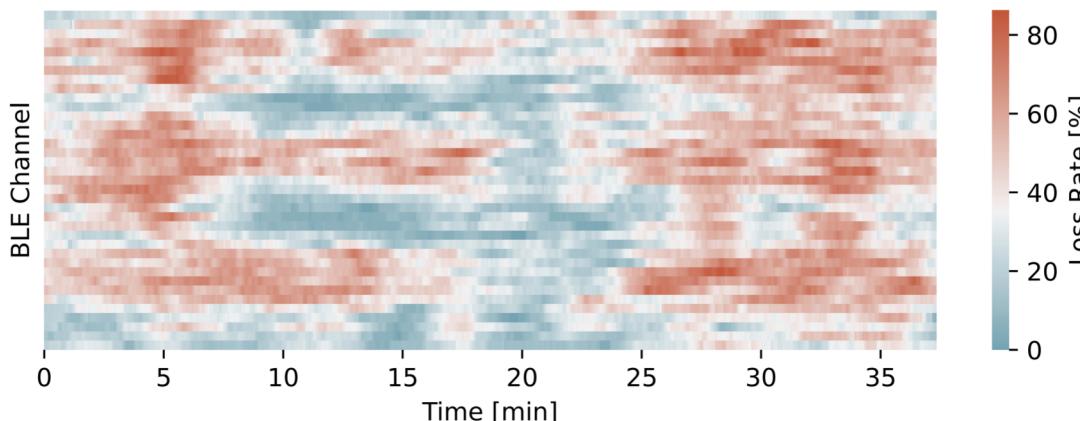
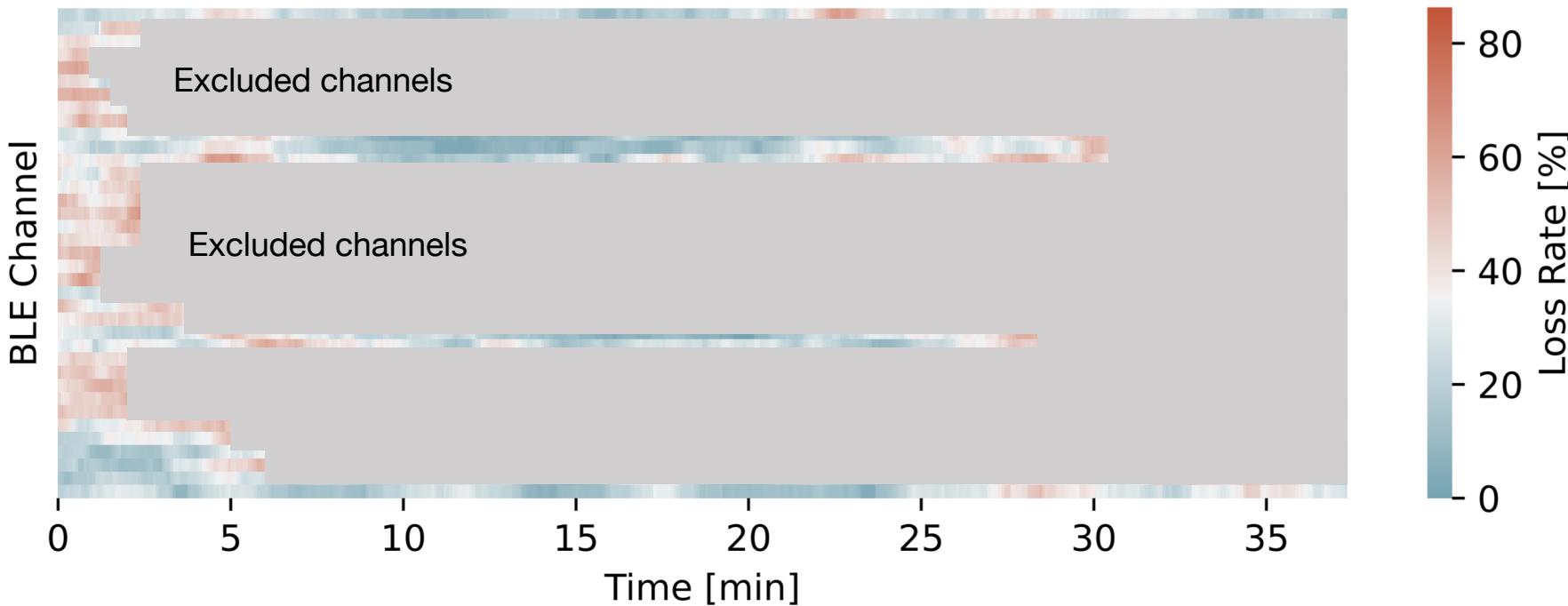
Motivation: Walking in a shopping mall



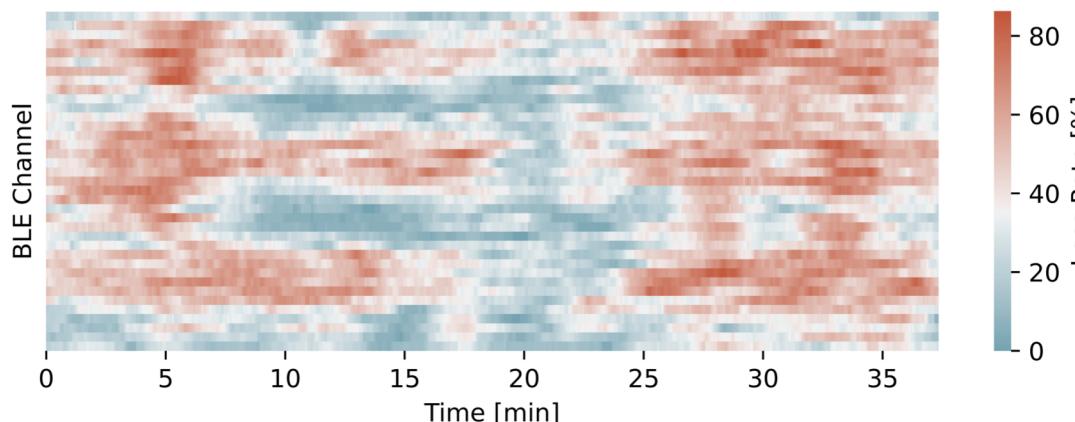
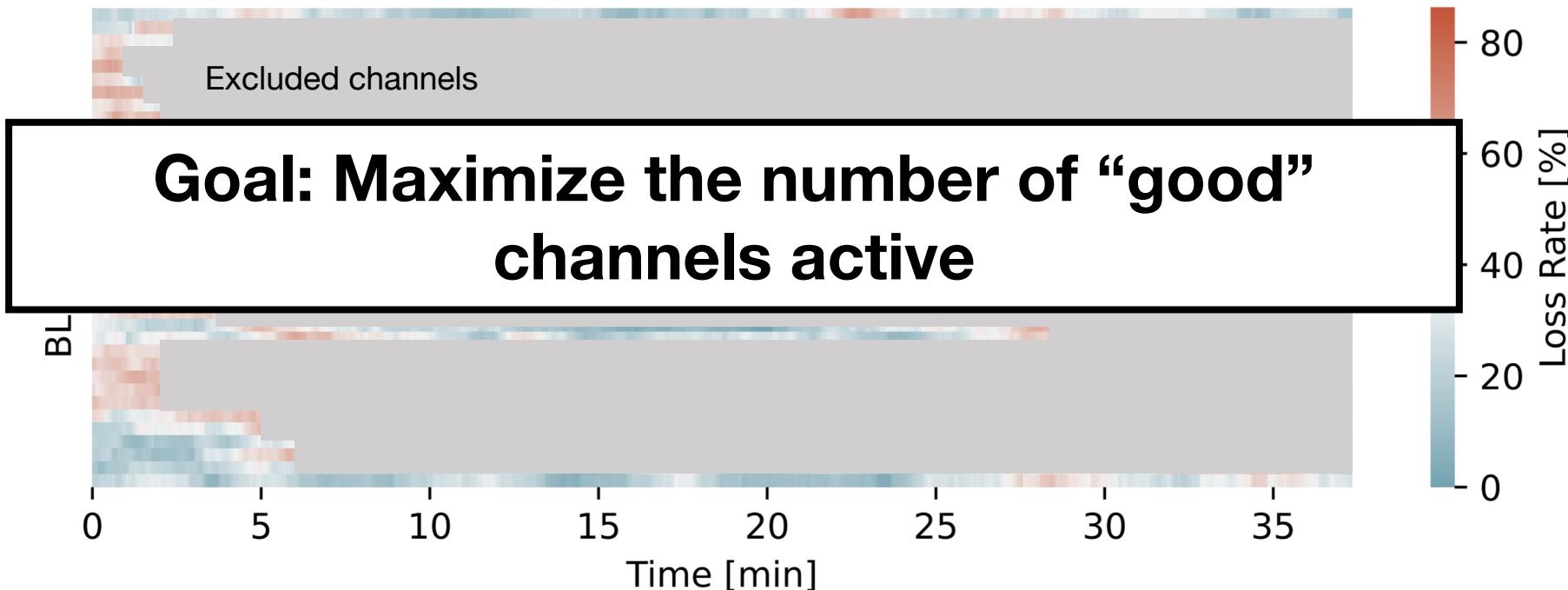
Problem: How to manage channels?



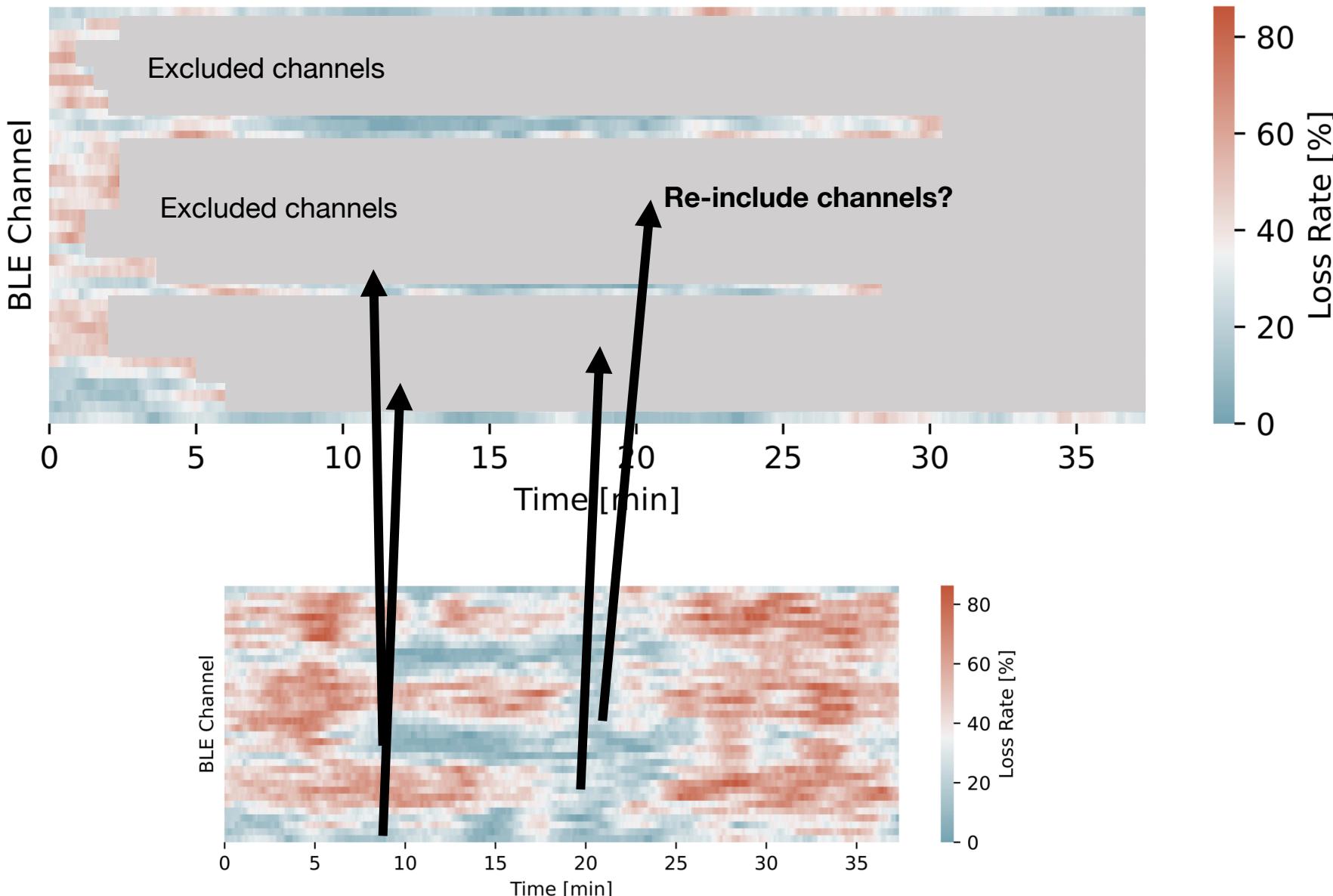
Problem: How to manage channels?



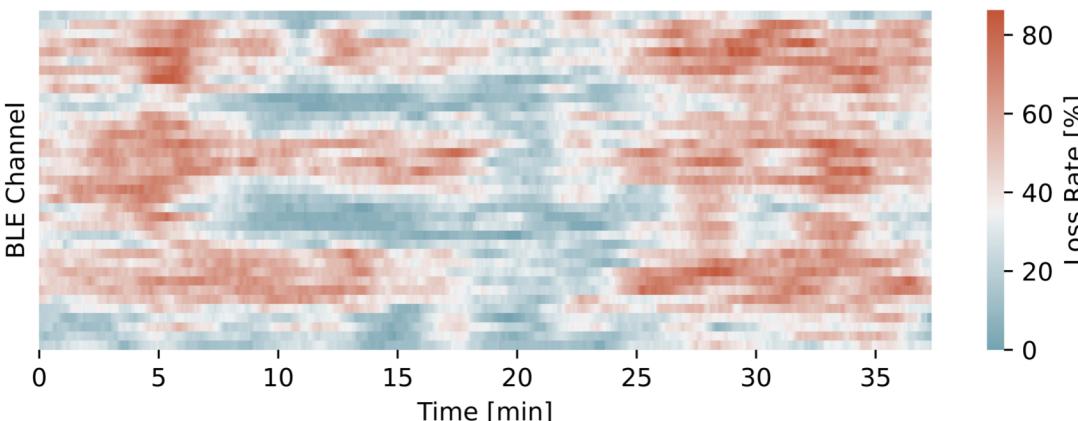
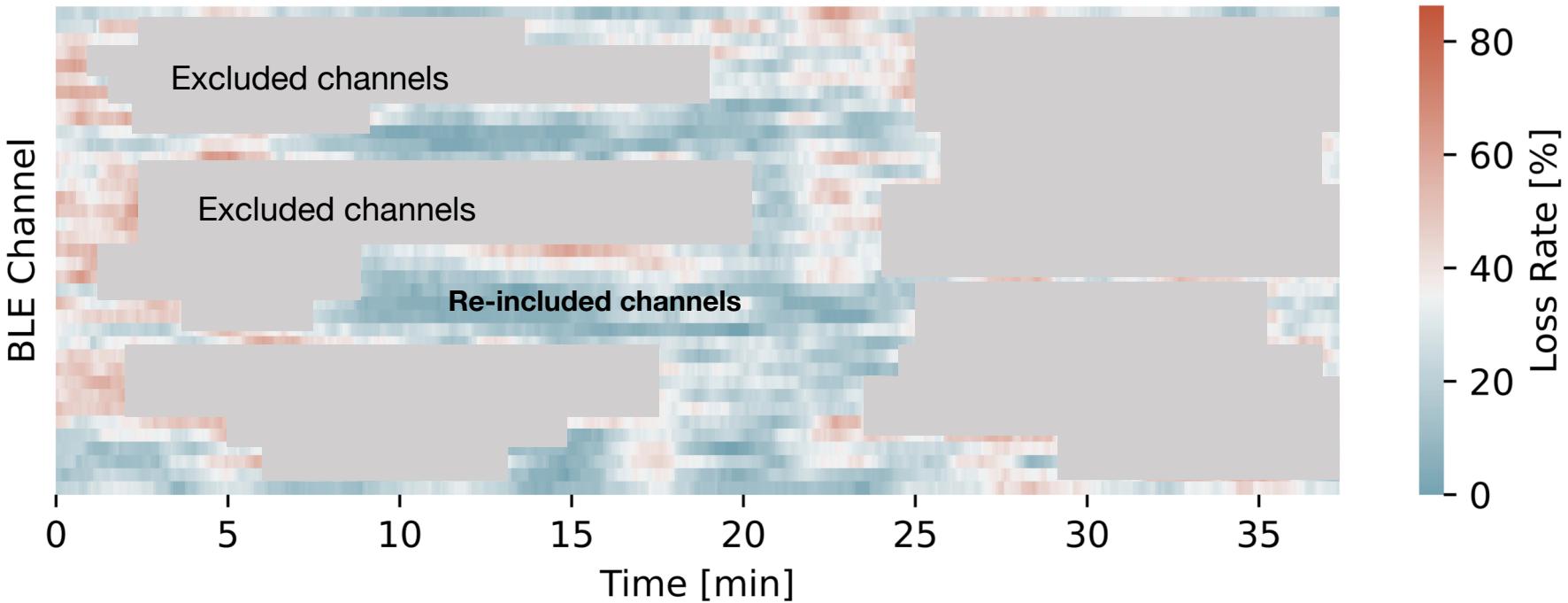
Problem: How to manage channels?



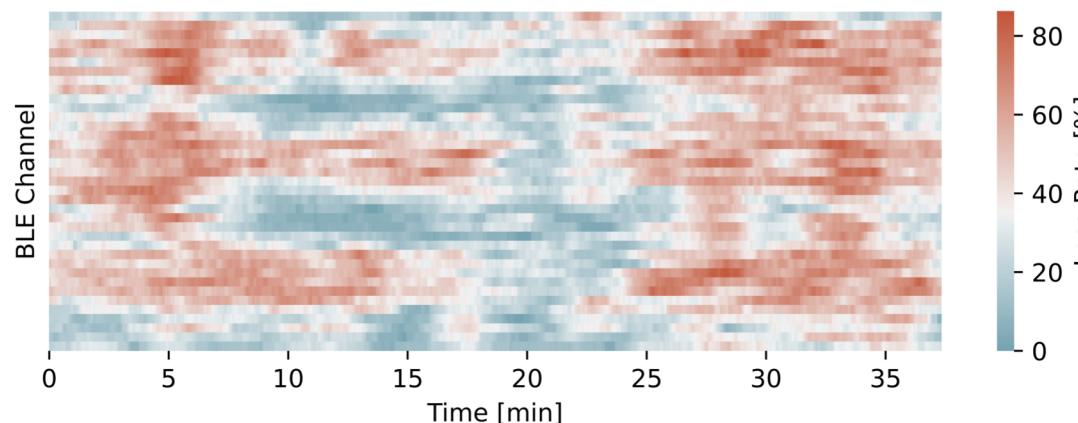
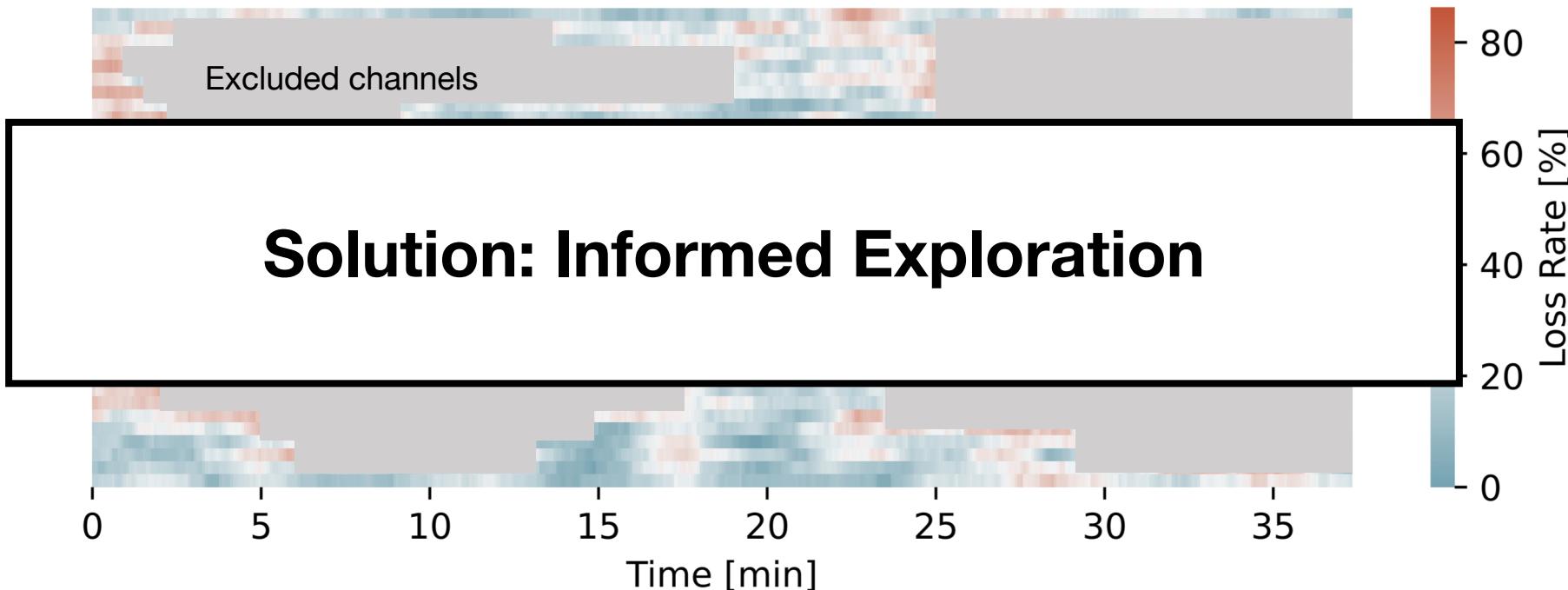
Problem: How to manage channels?



Problem: How to manage channels?

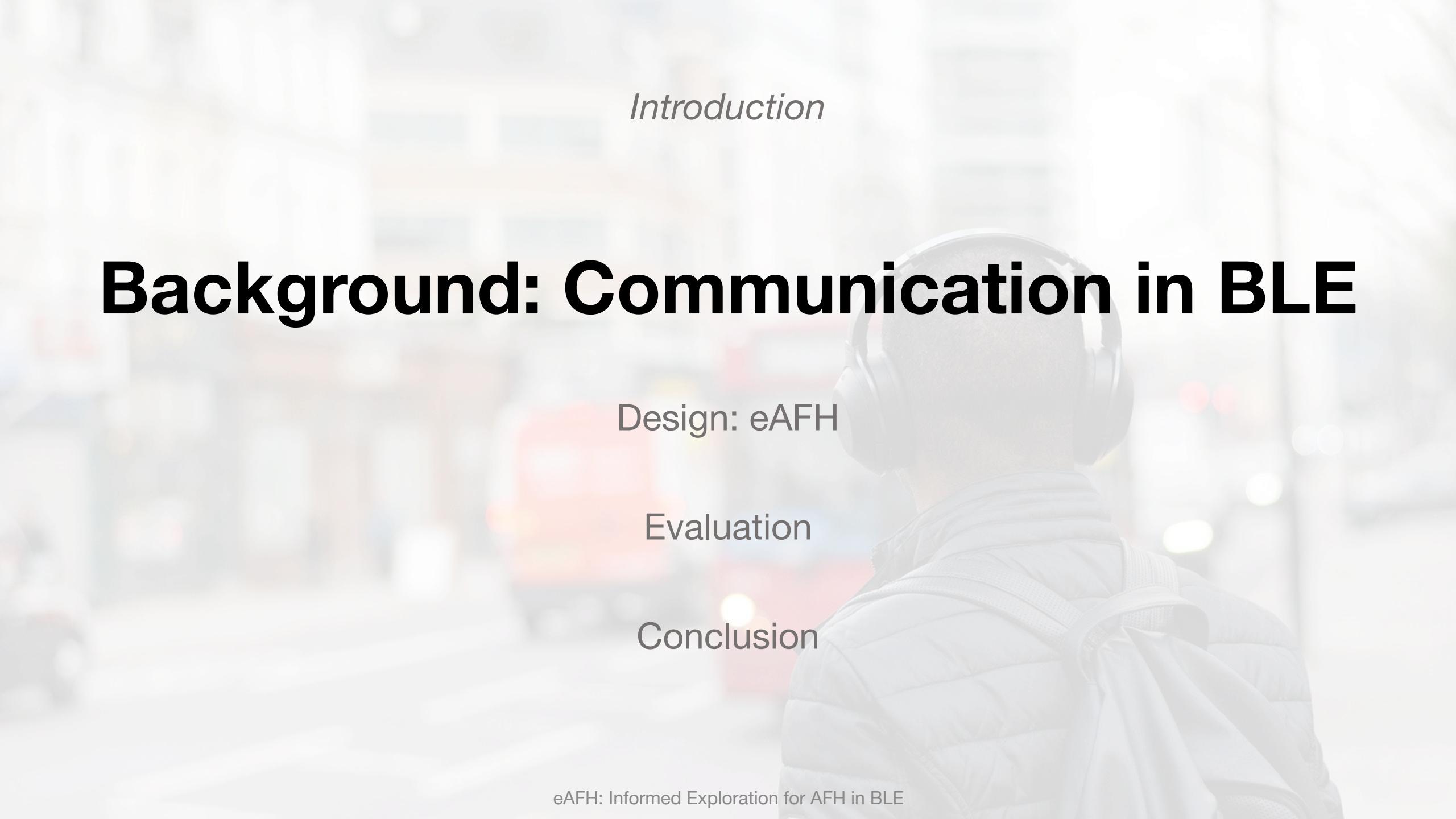


Problem: How to manage channels?



Contributions: eAFH

- We present **eAFH**: channel management for BLE
 - Excludes channels suffering losses
 - **Re-includes channels via informed exploration**
- **eAFH only needs to be implemented** by the central
 - Works with all commercial peripherals available today!
- We make eAFH open-source
 - github.com/ds-kiel/eAFH
 - for the Zephyr RTOS
- eAFH achieves 98 - 99.5% link-layer reliability against dynamic WiFi interference
 - providing a 40% increase in channel diversity
 - 1% control overhead cost

A grayscale photograph of a person from the back, wearing a backpack and headphones. The background is blurred, suggesting a city street at night with bokeh lights.

Introduction

Background: Communication in BLE

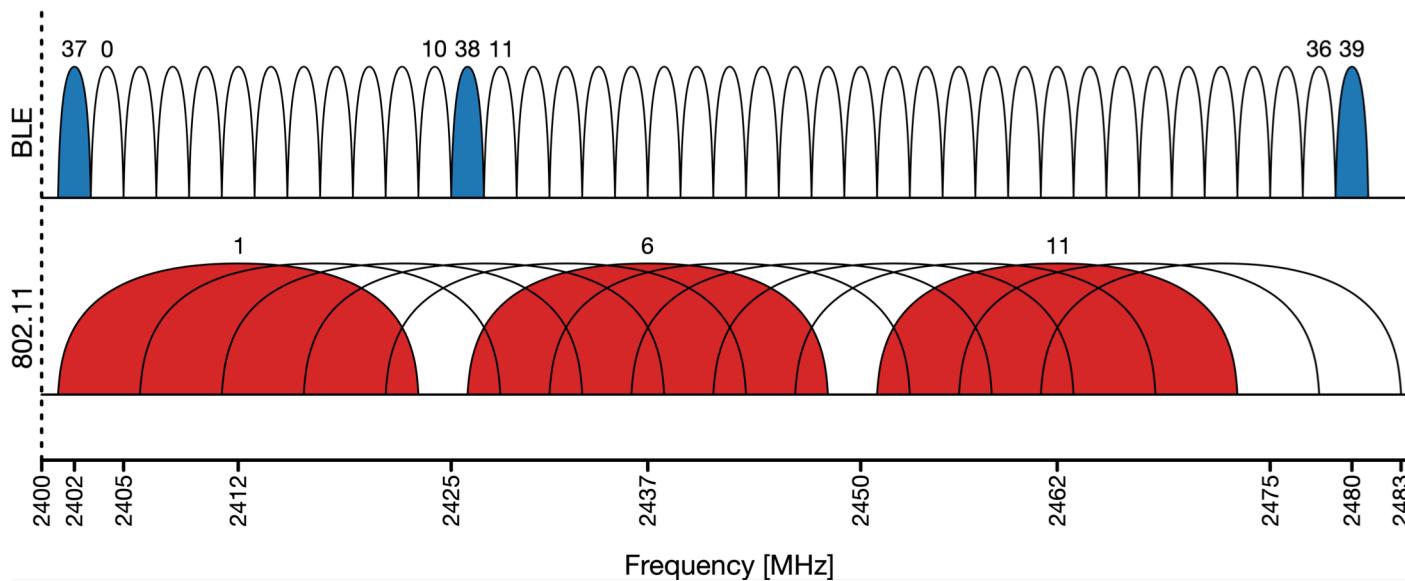
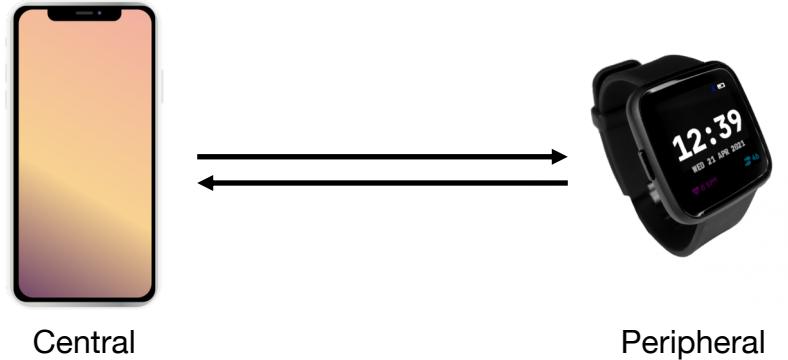
Design: eAFH

Evaluation

Conclusion

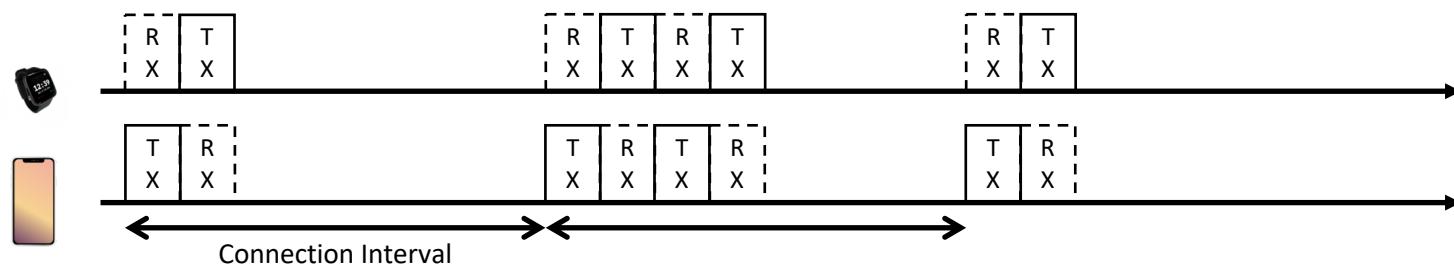
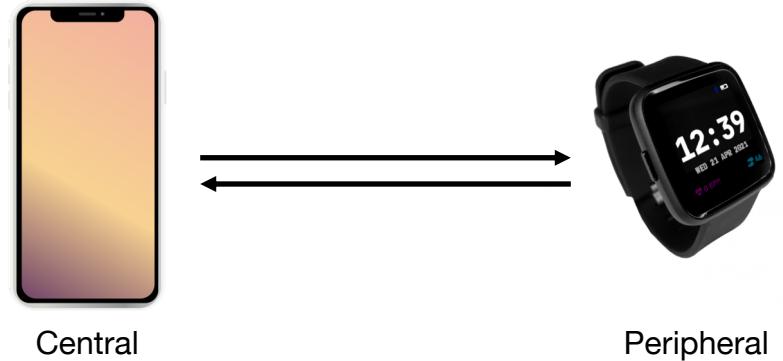
Background: Communication in BLE

- Two modes: Connectionless or Connection-oriented
- 40 channels: 3 for advertisements, 37 for connections



Background: Communication in BLE

- Two modes: Connectionless or Connection-oriented
- 40 channels: 3 for advertisements, 37 for connections
- Adaptive Frequency Hopping
 - Hops to a new frequency after each connection event
 - Central can discard channels from the channel map (hopping sequence)
 - Retransmit *until* a packet is received



A grayscale photograph of a person from behind, wearing large over-ear headphones and a backpack. They are standing on a city street at night, with blurred lights from traffic and buildings in the background.

Introduction

Background: Communication in BLE

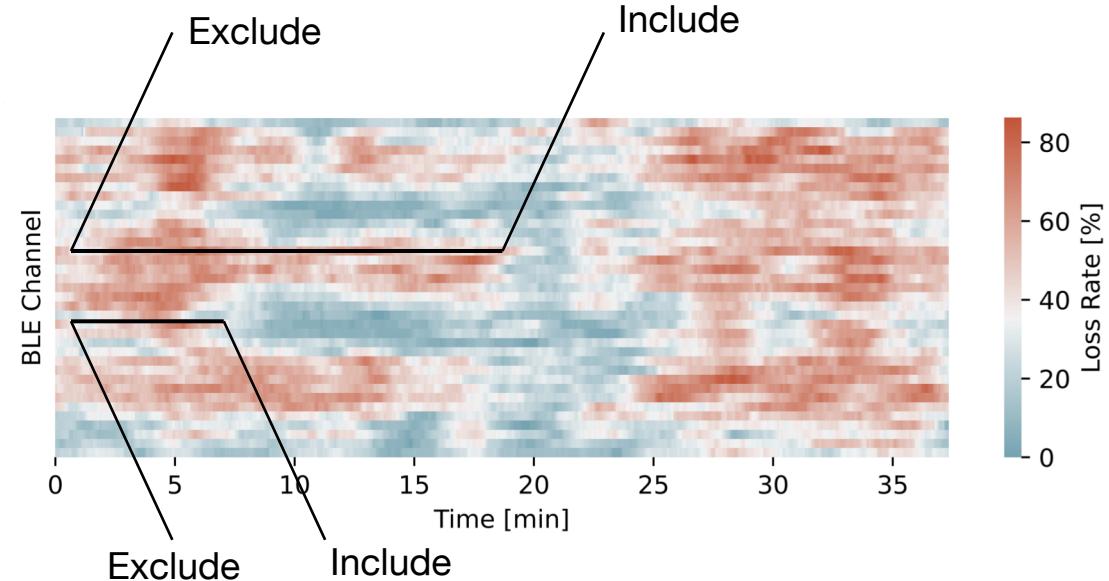
Design: eAFH

Evaluation

Conclusion

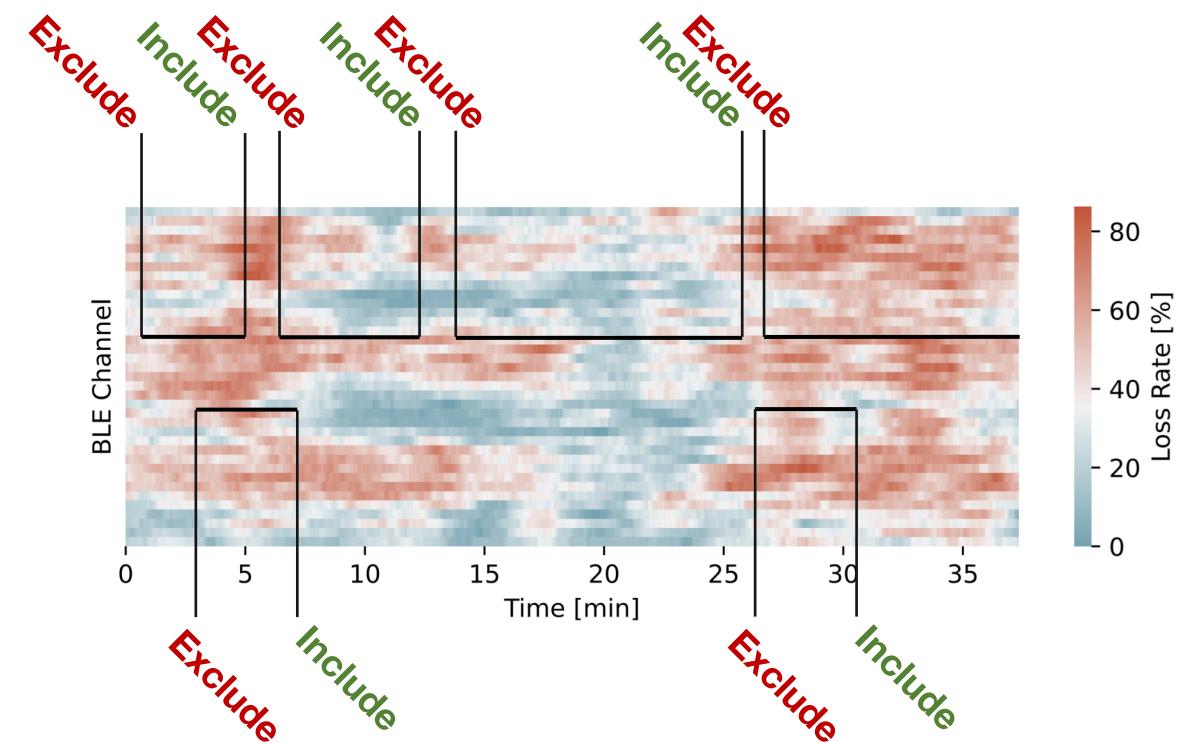
Channel Management and Exploration

- How to exclude channels?
 - Packet Delivery Ratio < threshold
- When to re-include channels?
 - Ideally, PDR > threshold
 - Cannot compute the PDR without transmissions!
 - Static timeout? Some channels are affected constantly, others not!
- Idea: find a **heuristic** estimating our **uncertainty** about the channel performance
 - Re-include the channel only if we are uncertain about its real performance
 - Uncertainty evolves with **time** and **past performance**



Estimating uncertainty

- Re-include after timeout
 - Explore the channel to compute a new PDR
 - Only use the *recent PDR* for exclusion
- Idea 1: Increase timeout duration based on past performance *across exclusions*
 - Keep memory of past losses to ensure longer timeouts
 - Timeout increases exponentially



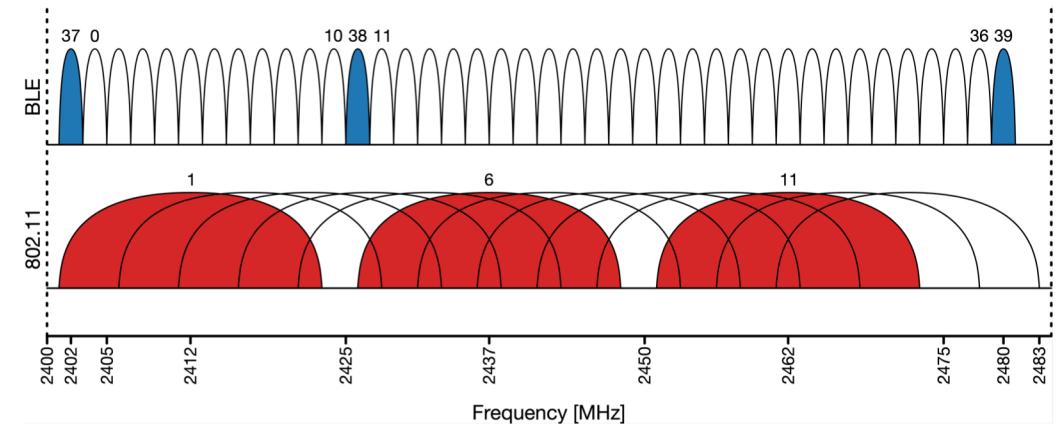
Estimating uncertainty

Idea 2: Losses are correlated across channels

- Especially with Wifi traffic
- Take into account the nearby PDR to increase the timeout duration

$$\text{Uncertainty}(c) = \frac{t}{\text{past losses}} - \text{losses}_{\text{nearby-channels}}$$

Idea 1 *Idea 2*

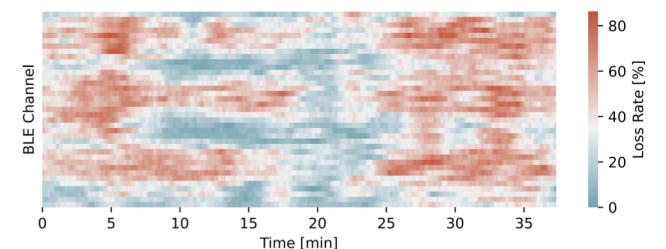


If $\text{Uncertainty} > \text{threshold}$

- Re-include channel c

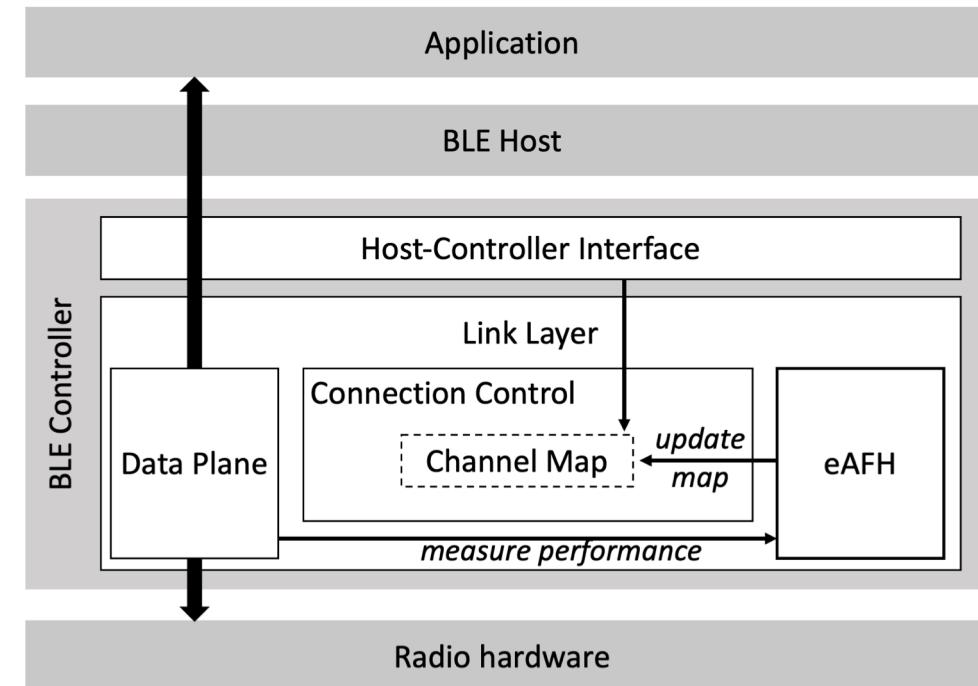
Else

- t++



eAFH

- eAFH: channel management for BLE
 - Excludes channels via PDR
 - **Re-includes channels via uncertainty**
- Implemented within the BLE stack
 - Application-agnostic
- Standard compliant
 - Only executed by the central
 - Works with any peripheral
- Targets dynamic scenarios with mobility



Introduction

Background: Communication in BLE

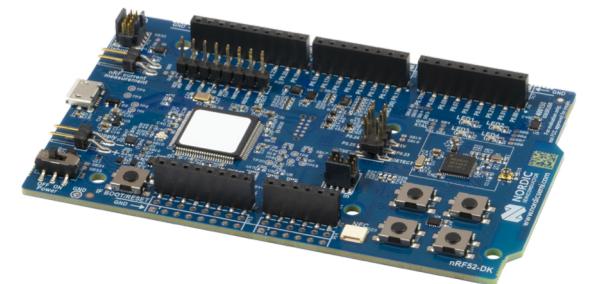
Design: eAFH

Evaluation

Conclusion

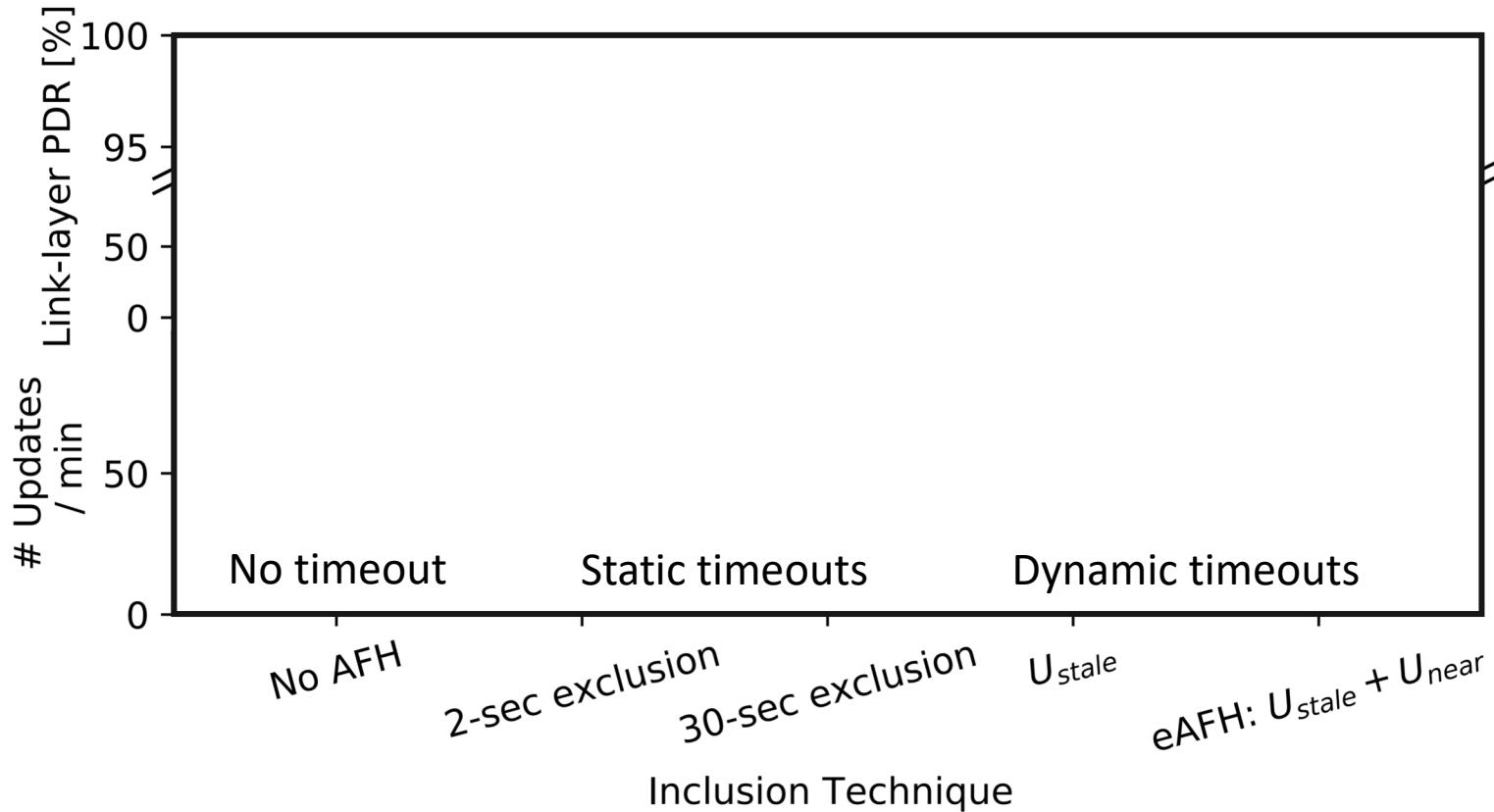
Evaluation Setup

- Implemented for the Zephyr OS
 - Part of the Link layer of the BLE controller
- Platform: nRF52
 - Microcontroller with 64 MHz MCU , 256 kB RAM, BLE 5
 - Packet exchange every 20 ms
- Effects of Wifi interference
 - Constant traffic
 - Traffic evolving over time



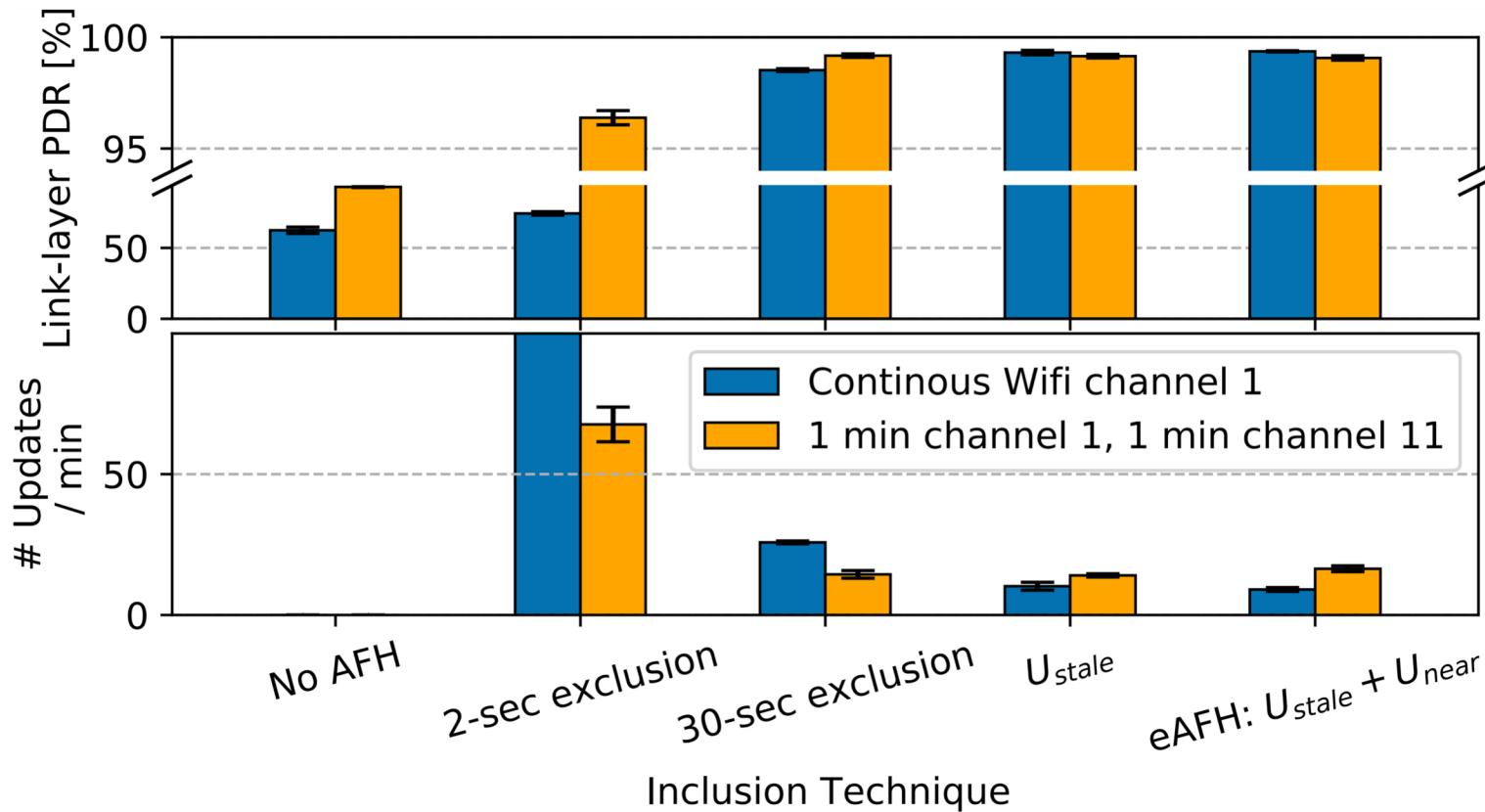
Evaluation: exploration overhead

- Performance (PDR) and overhead (updates per minute) against:
 - Constant jamming & one-off jamming



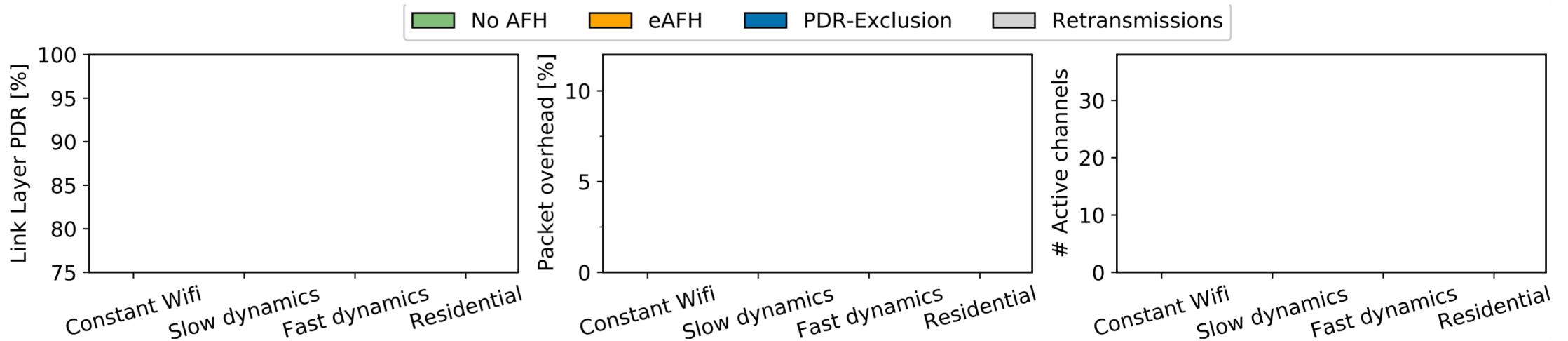
Evaluation: exploration overhead

- Performance (PDR) and overhead (updates per minute) against:
 - Constant jamming & one-off jamming



Evaluation: State-of-the-Art Comparison

- PDR-Exclusion: Excludes based on PDR & re-includes everything once too few channels are active



- Willing to risk some retransmissions to ensure channel diversity

Introduction

Background: Communication in BLE

Design: eAFH

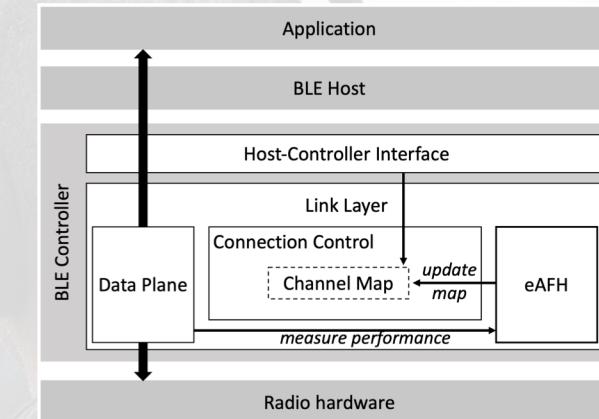
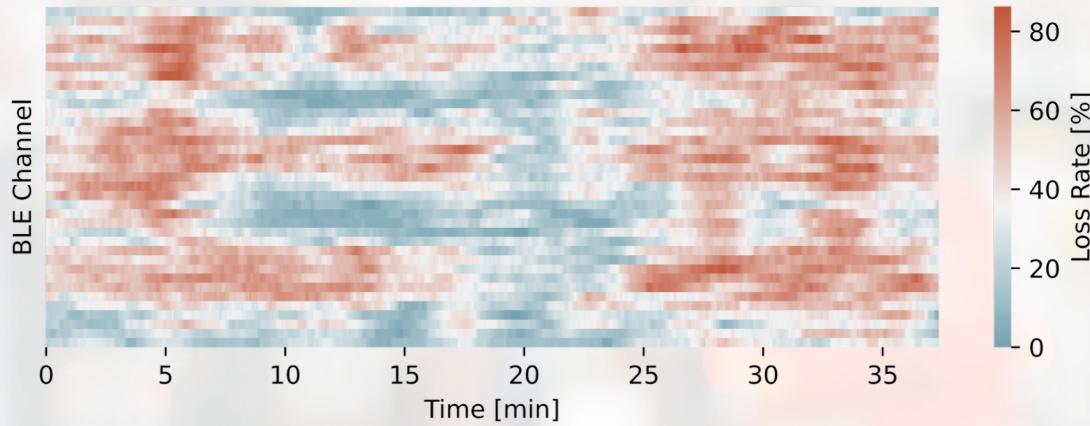
Evaluation

Conclusion

Conclusion

- We present **eAFH**: channel management for BLE
 - Excludes channels suffering losses
 - **Re-includes channels via informed exploration**
- **eAFH only needs to be implemented** by the central
 - Works with all commercial peripherals available today!
- We make eAFH open-source
 - github.com/ds-kiel/eAFH
 - for the Zephyr RTOS
- eAFH achieves 98 - 99.5% link-layer reliability against dynamic WiFi interference
 - providing a 40% increase in channel diversity
 - 1% control overhead cost

eAFH: Informed Exploration for Adaptive Frequency Hopping in Bluetooth Low Energy



vpo@informatik.uni-kiel.de

github.com/ds-kiel/eAFH