Multi-Connection Scheduling based on Connection Subrating for Fair Resource Allocation in Bluetooth Low Energy Networks

Networked Systems Laboratory

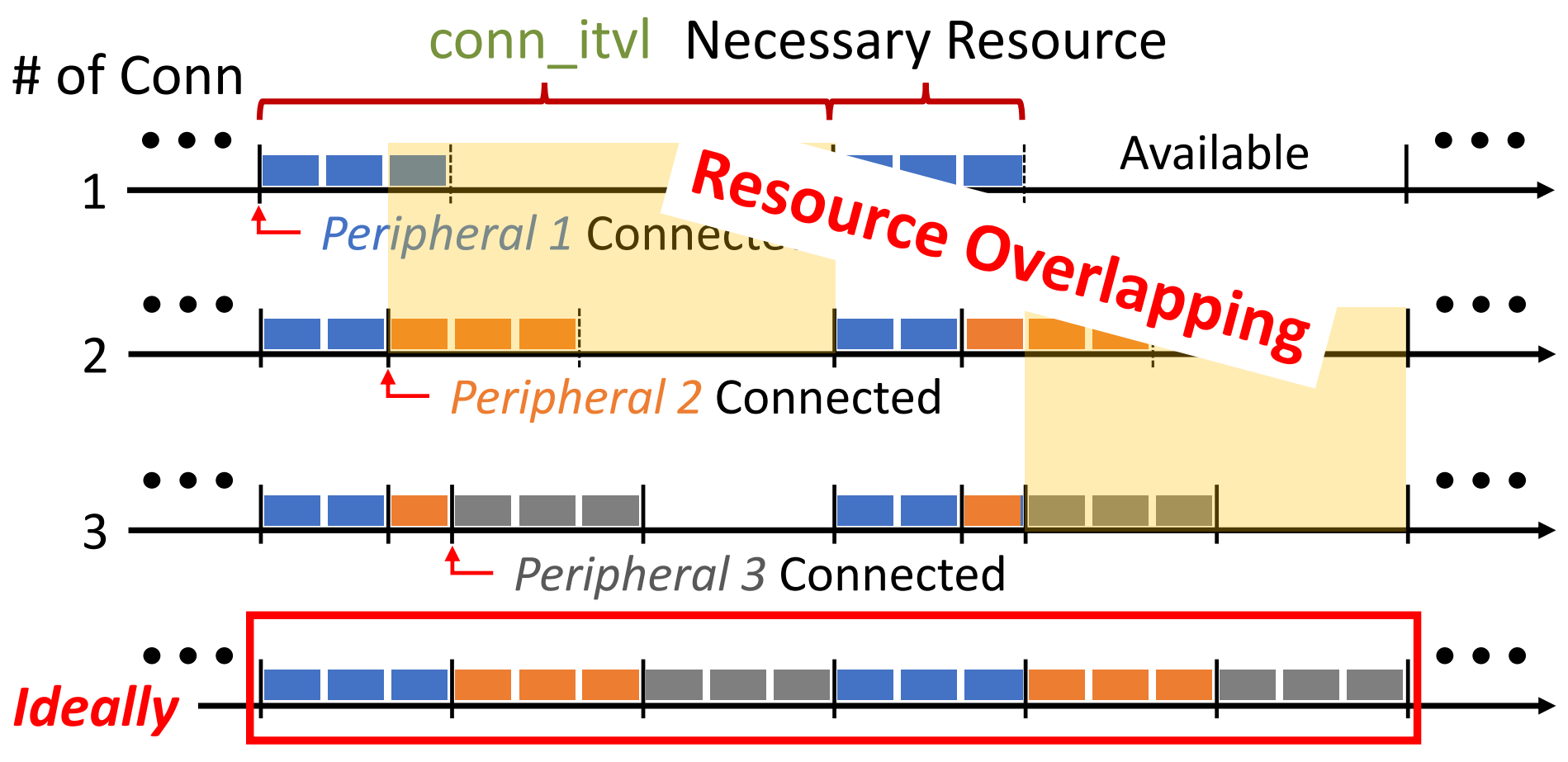
Moonbeom Kim, Jeongyeup Paek
Chung-Ang University
Department of Computer Science and Engineering
mbkim@cau.ac.kr, jpaek@cau.ac.kr

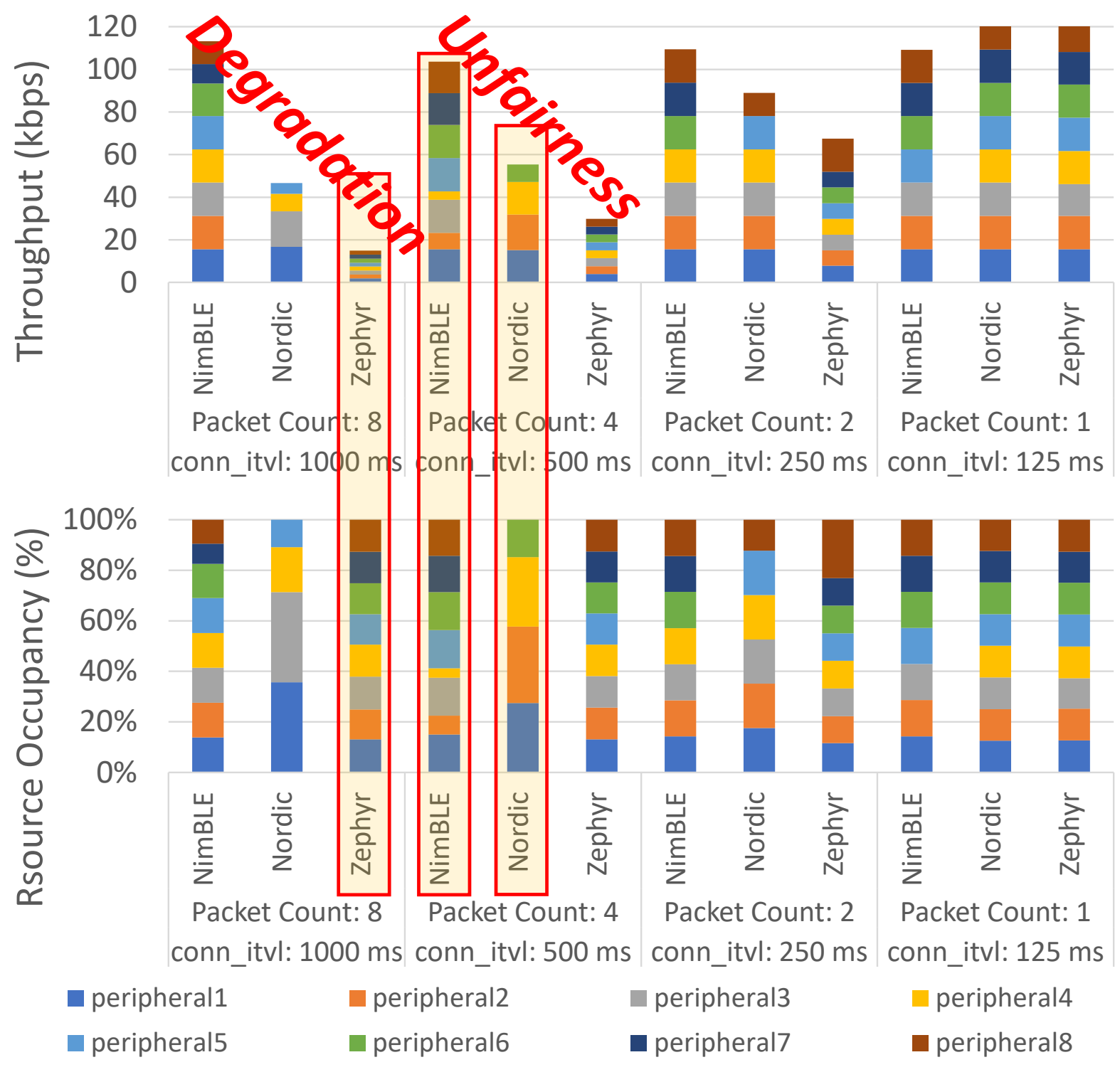
Introduction - Bluetooth Low Energy (BLE)

Resource Allocation Unfairness

- The *Bluetooth SIG* **does not specify algorithm for scheduling and managing resources** of multiple connections.
- Most commercial *Central BLE devices* **independently handle the connection schedule and resources** for each connected *Peripheral*.
- When a new connection is established, the *Central* assigns resources for the new connection **without considering the QoS of previous ones as well as the current status**.

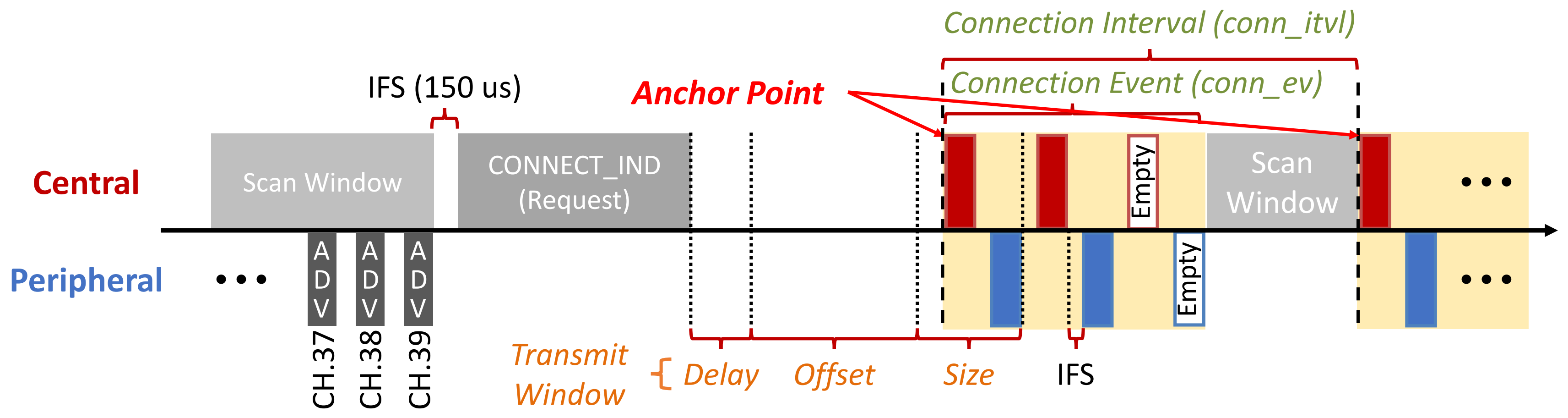
➔ **Cause the Resource Overlapping & Performance Degradation Problems!**



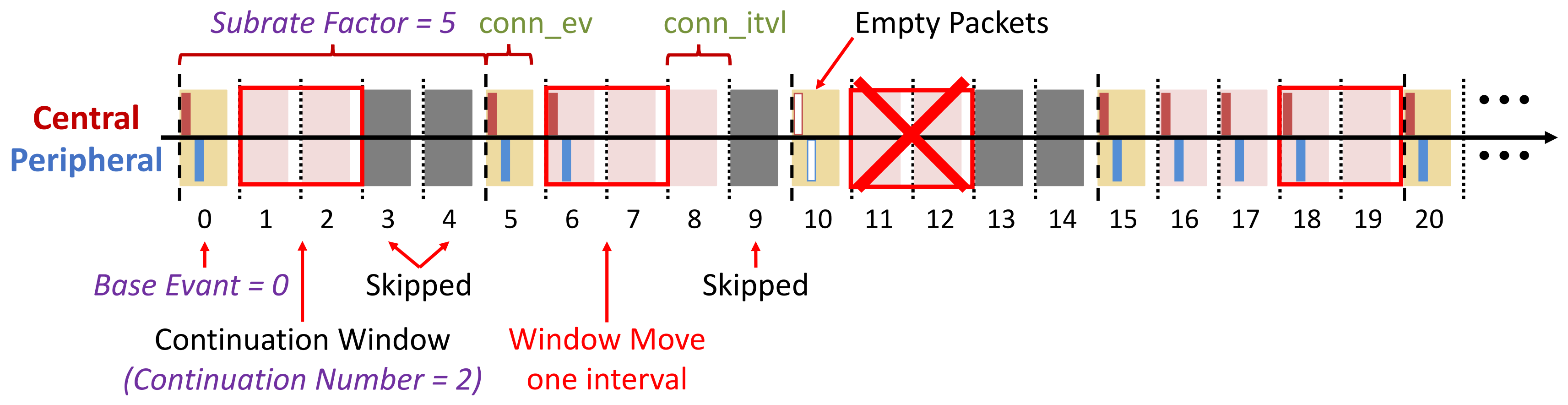


BLE Connection Establishment

- The ***Central*** *(i.e., Master)* manages most of the procedures and parameters for connection with the ***Peripheral*** *(i.e., Slave)*.



- ***Connection Subrating*** for low-duty cycle connection with power-saving, and fast connection update.



Design – Subrating-based Connection Scheduling (*SCS*)

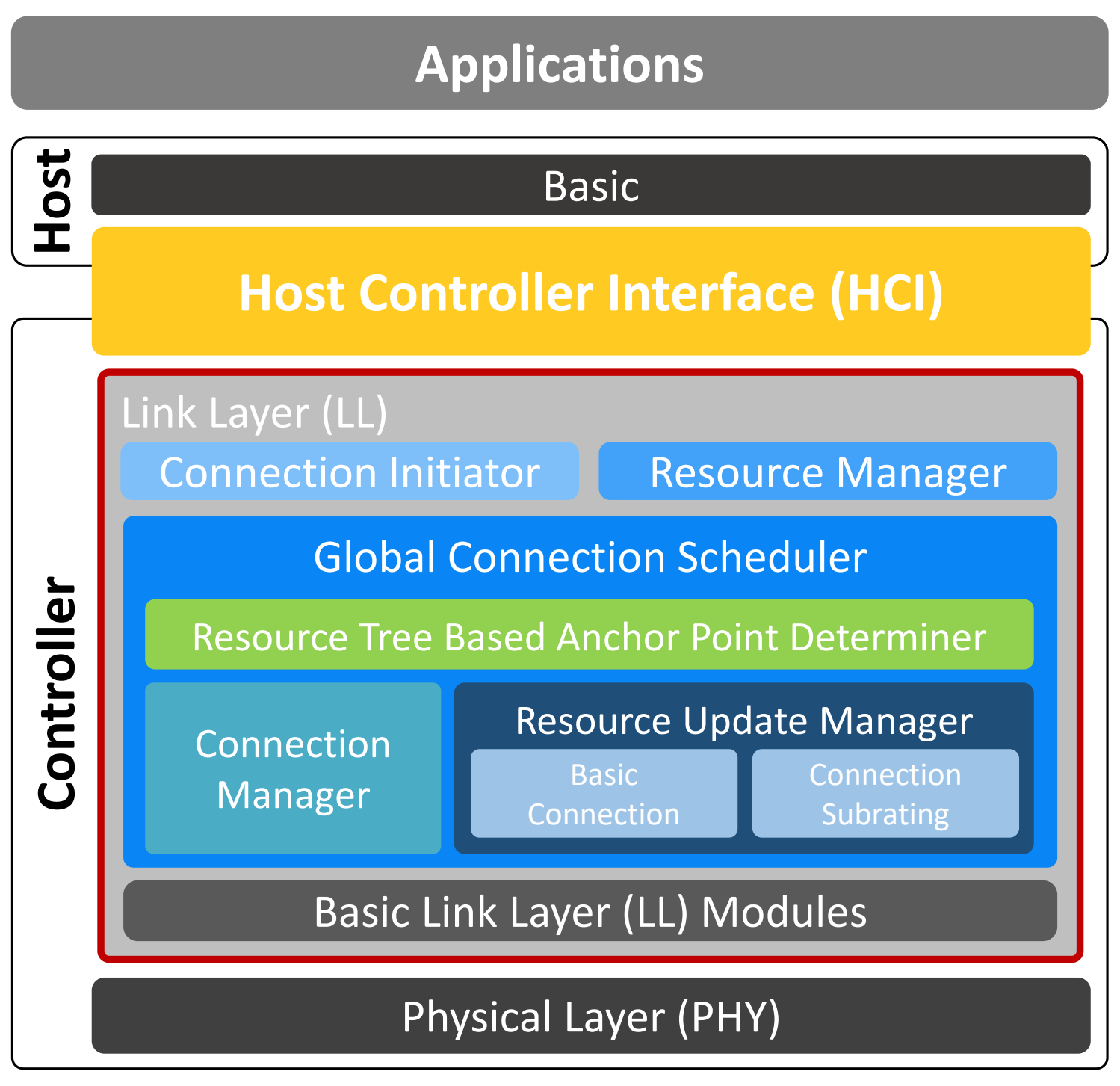
Connection Scheduling

- ***Connection Initiator***
  - Fixate the conn_itvl to 7.5 ms.
  - Mimic the original conn_itvl$_o$ using *subrate factor (sf)*.

$$sf = ceil\,(conn\_itvlO/7.5\,ms),$$

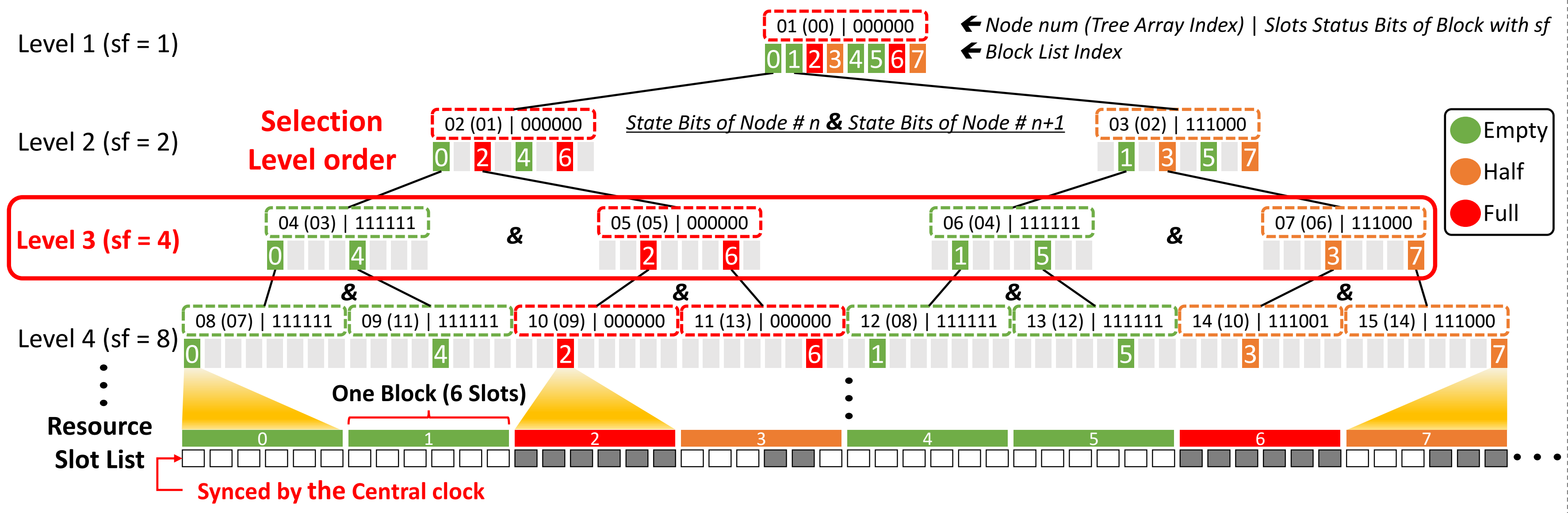
$$f(sf) = \begin{cases} sf & ,sf = 2^n \\ sf = sf \pm \min(sf - 2^{n-1},\ 2^n - sf) & ,sf \neq 2^n \end{cases}$$

  - Range of mimicked *conn_itvl*: n = 0~9 ➔ 7.5, 15, 30, … , 3840 ms



BLE Central Stack of *SCS*

- ***Resource Tree based Anchor Point Determiner***
  - Propose a *"Bitwise Operation-based Resource State Tree"* for fast resource search, update, and reduced of memory usage.
  - One slot represents 1.25 ms.



Resource Allocation and Update

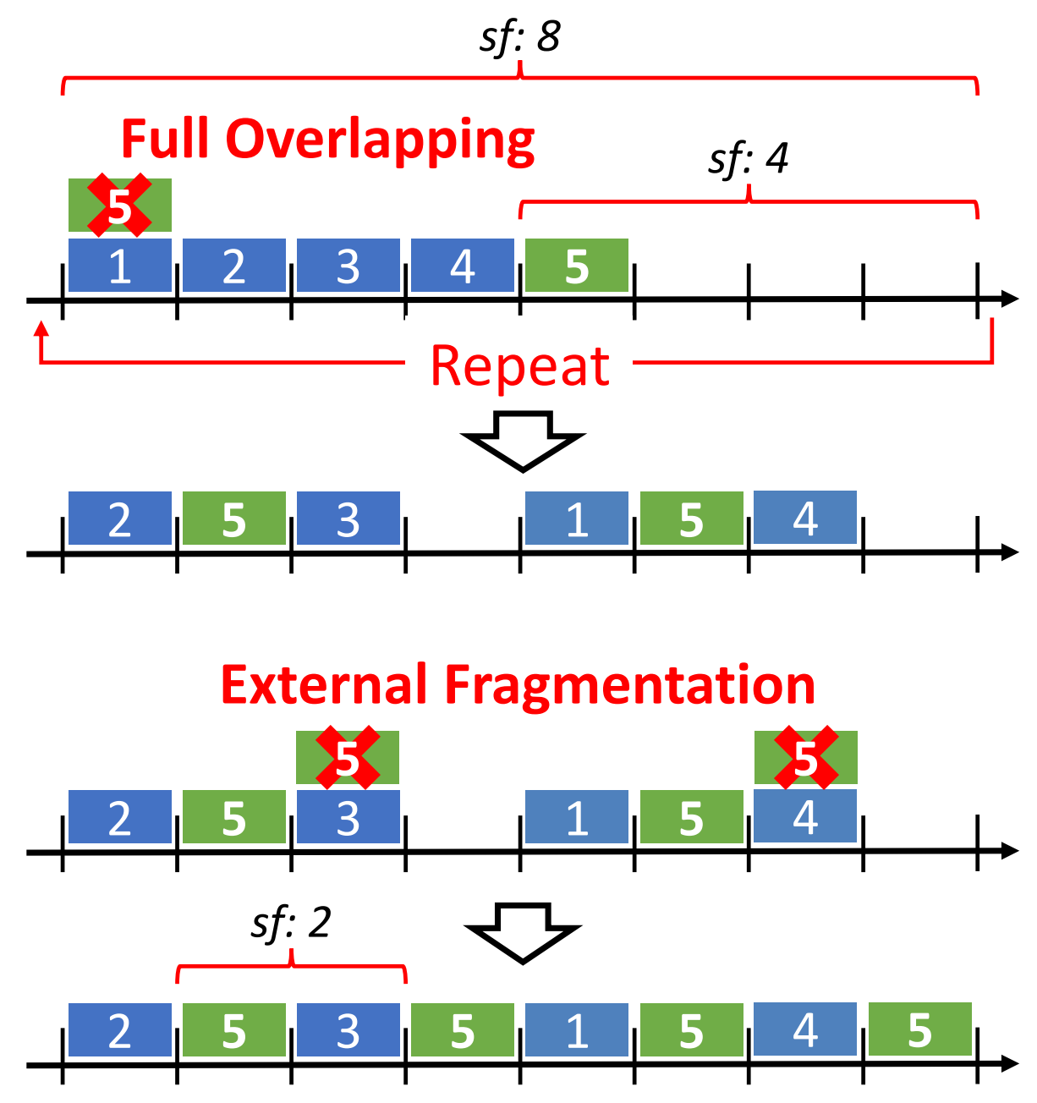
- ***Resource Manager***
  - Propose a *"ZigZag Allocation with Resource Split"* to achieve multiple connections with full overlapping prevention and external fragmentation mitigation.
    - Ex) *ZigZag*

      Peripheral 1, 2, 3, and 4: subrate factor (8), necessary resources (1 Block)

      Peripheral 5: subrate factor (4), necessary resources (1)
    - Ex) *Resource split*

      Peripheral 5: subrate factor (4), necessary resources (2)

      ➔ subrate factor (2), necessary resources (1)



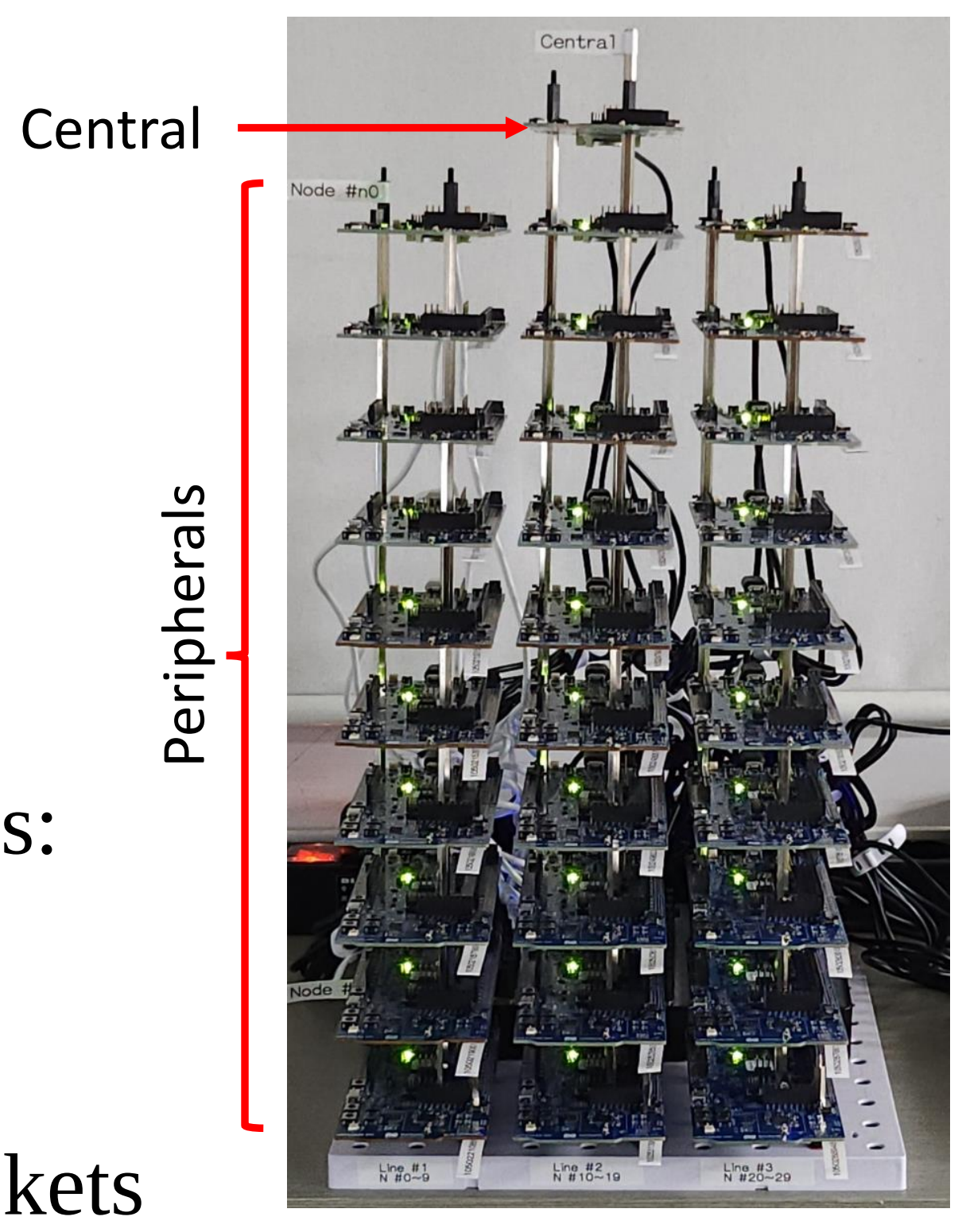
  - Adopt an *"Exponentially Weighted Moving Average (EWMA)"* to periodically measure and manage the resources.

```
resource_diff = allocated_resource - conn_ev_EWMA

if ( | resource_diff | <  threshold + guard_slots )
    Reduce resources by resource_diff
    Update Resource Slot List and Bitwise Operation-based Resource States Tree

else if (  resource_diff  >=  thresholds )
    Increase resources by resource_diff + gard_slots
    Update Resource Slot List and Bitwise Operation-based Resource States Tree
```

Evaluation – Comparison with Popular BLE Stacks

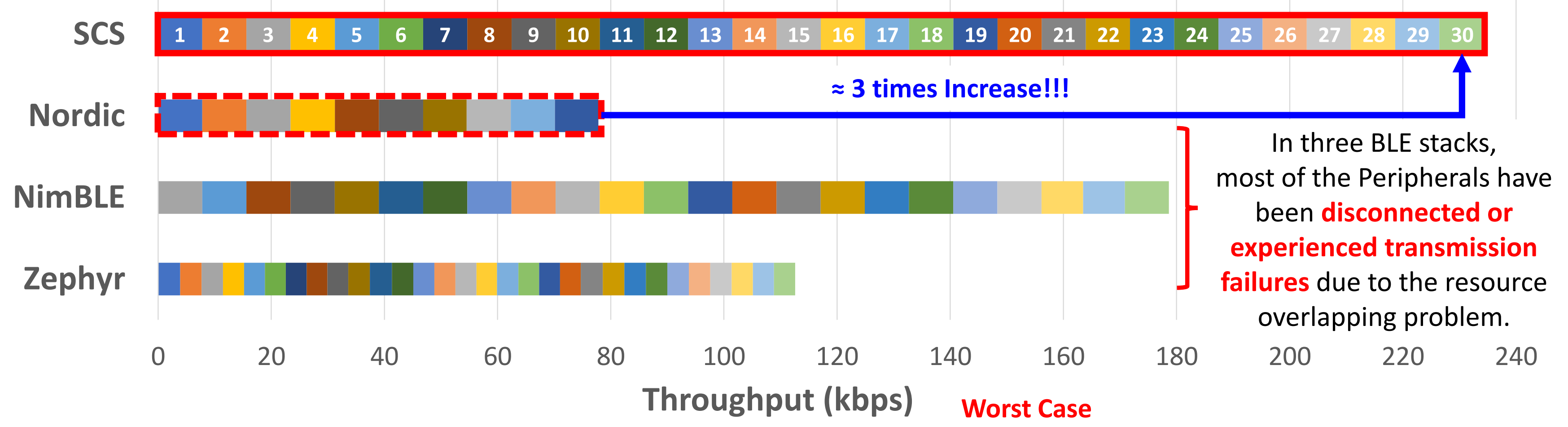
Experiment Setup

- nRF52840 DK BLE Board
  - Central (1) and Peripherals (30)
- Comparison Group
  - NimBLE, NORDIC, and Zephyr Project
- Scenario
  - All Peripherals have the following conditions:
    - Payload size: 244 bytes (max.)
    - conn_itvl: 500 ms
  - Number of transmission per *conn_itvl*: 2 packets
- Each Peripheral is established connection with a 10 second interval.



Preliminary Results

- **Throughput** and **Occupancy** of Each Peripheral
  - *SCS* achieves not only approximately 100% of the expected maximum reachable throughput (234.24 kbps) but also resource allocation fairness.



In three BLE stacks, most of the Peripherals have been **disconnected or experienced transmission failures** due to the resource overlapping problem.

Worst Case

Theoretical	*SCS*	Nimble	Nordic	Zephyr
Avg. each Peripherl	**≈ 7.808 kbps**	≈ 5.955 kbps	≈ 2.594 kbps	≈ 3.753 kbps
(%)	**≈ 100 %**	≈ 76 %	≈ 33 %	≈ 48 %