# Bluetooth Multilink Unleash Your Imagination

## – GATT proprietary connected Mesh with all nodes powered by button cell

## 蓝牙多连接，释放无限想象

- 构建基于GATT的全网低功耗节点的私有Mesh

Kevin Ai 艾敏华

FAE Team Leader – Mainland China

kevin.ai@nordicsemi.no

Bluetooth Asia 2019

# Agenda

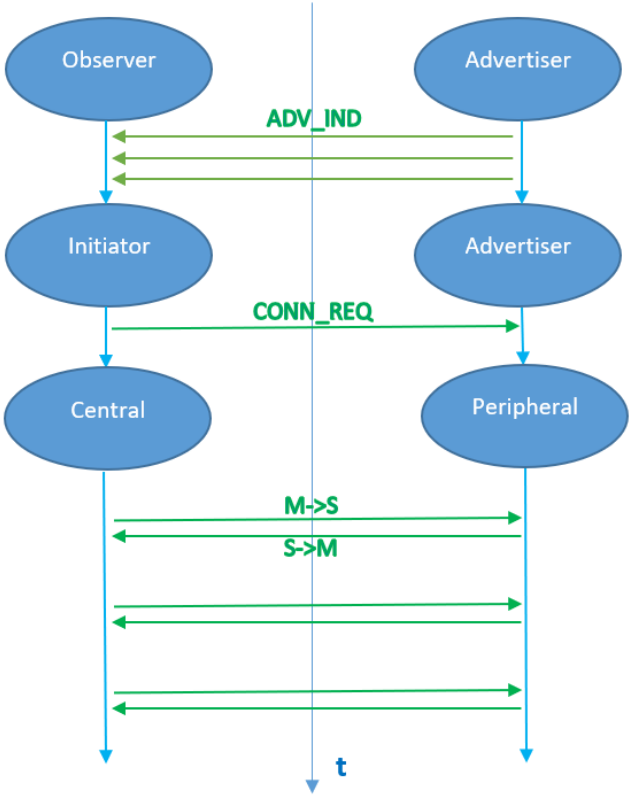
1 **Bluetooth multi-link concepts and implementations**

2 **True low power relay network and mesh design**

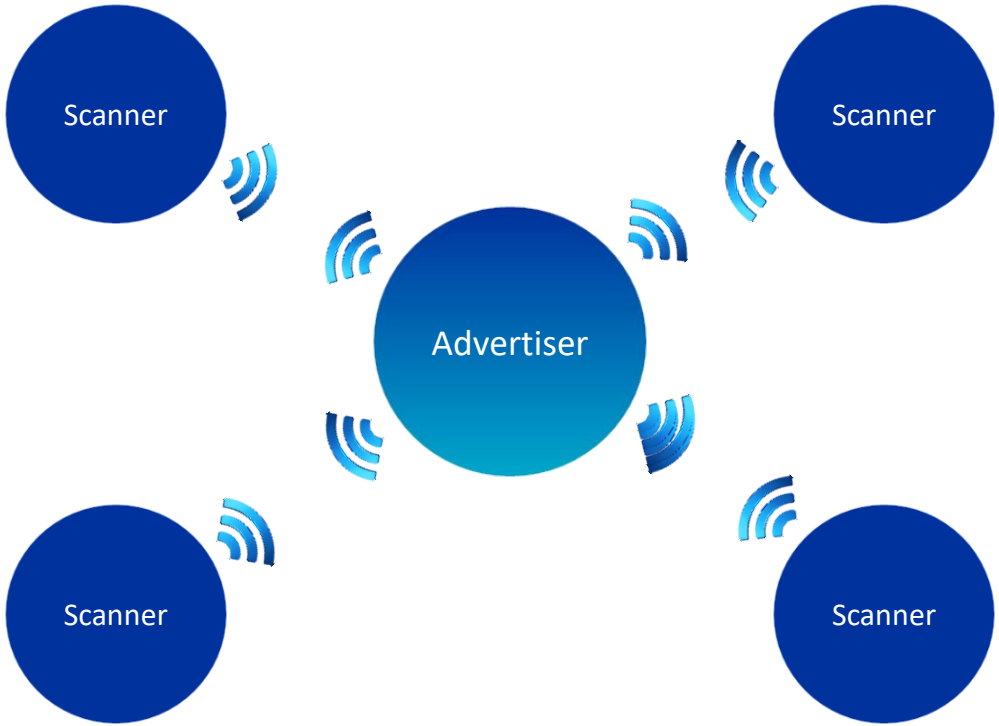
3 **Bluetooth multi-link use case study**

# LE roles

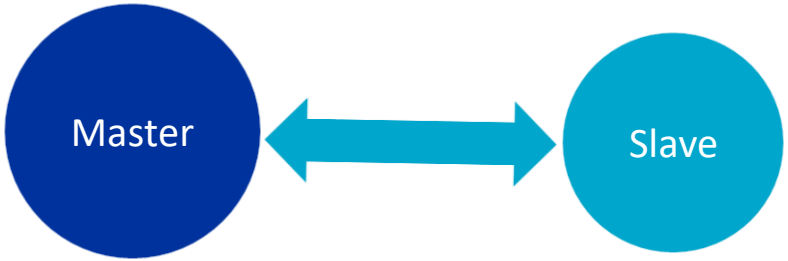
- Advertiser/Broadcaster
- Observer/scanner
- Initiator
- Master/central
- Slave/peripheral



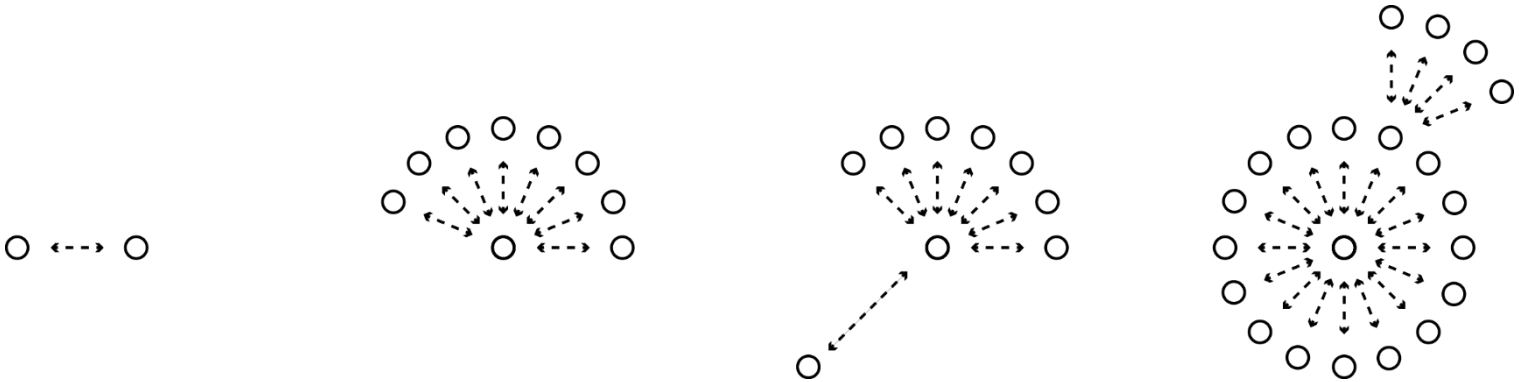
# LE topology – Broadcast



# LE topology – Connection/Link



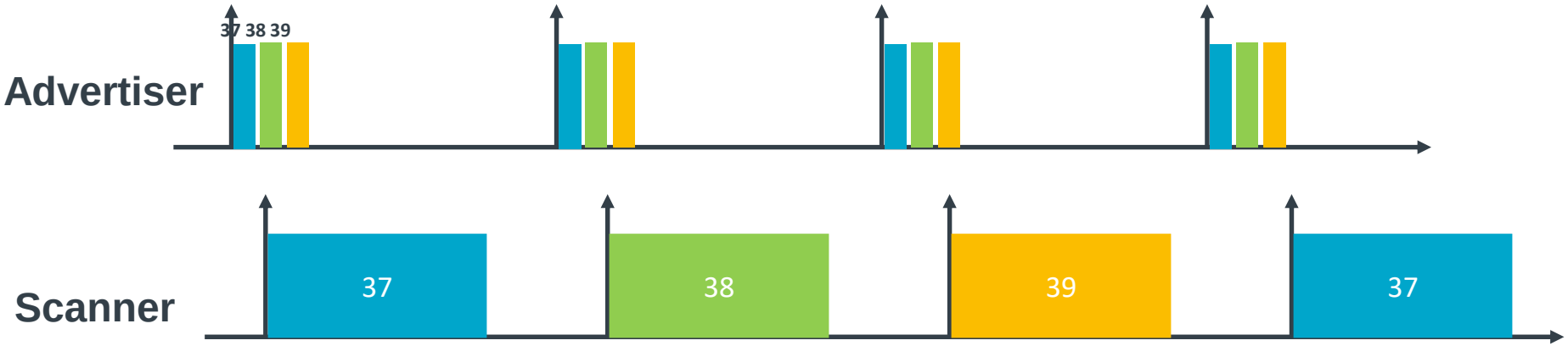
# Bluetooth connection evolution



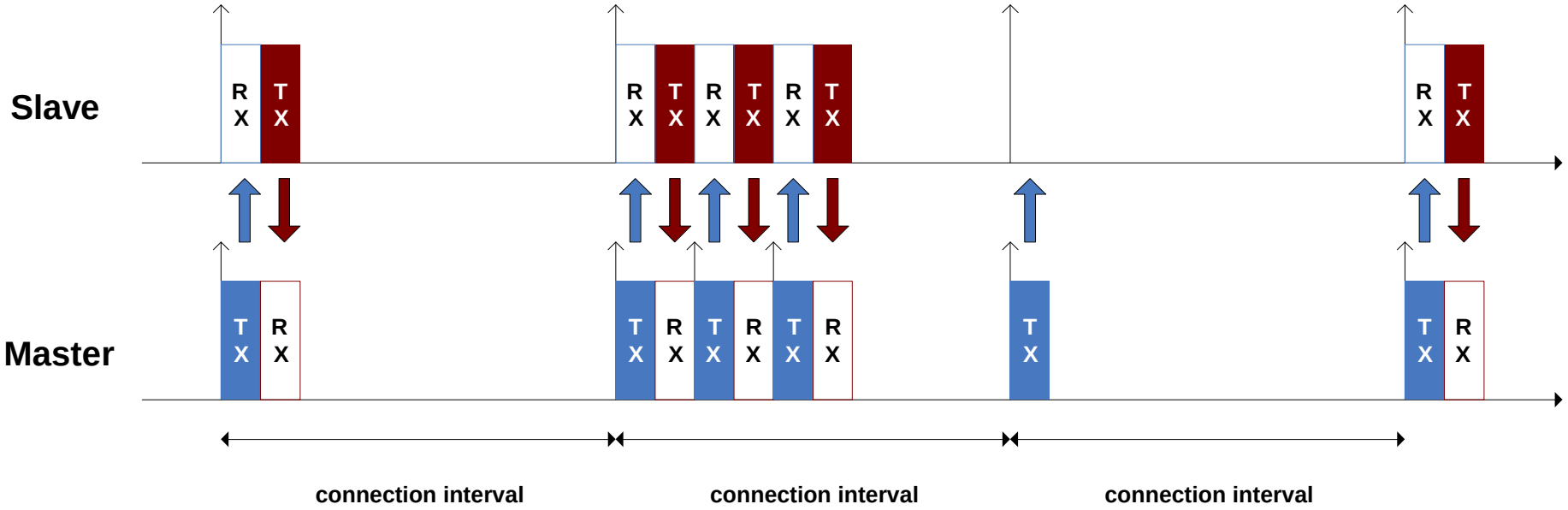
| Single link peripheral | Peripheral / multi-link master<br><br>Up to 8 links | Concurrent peripheral / multi-link central<br><br>Up to 8 links | Concurrent multi-link peripheral and central<br><br>Up to 20 links |
|---|---|---|---|

2011 → Today

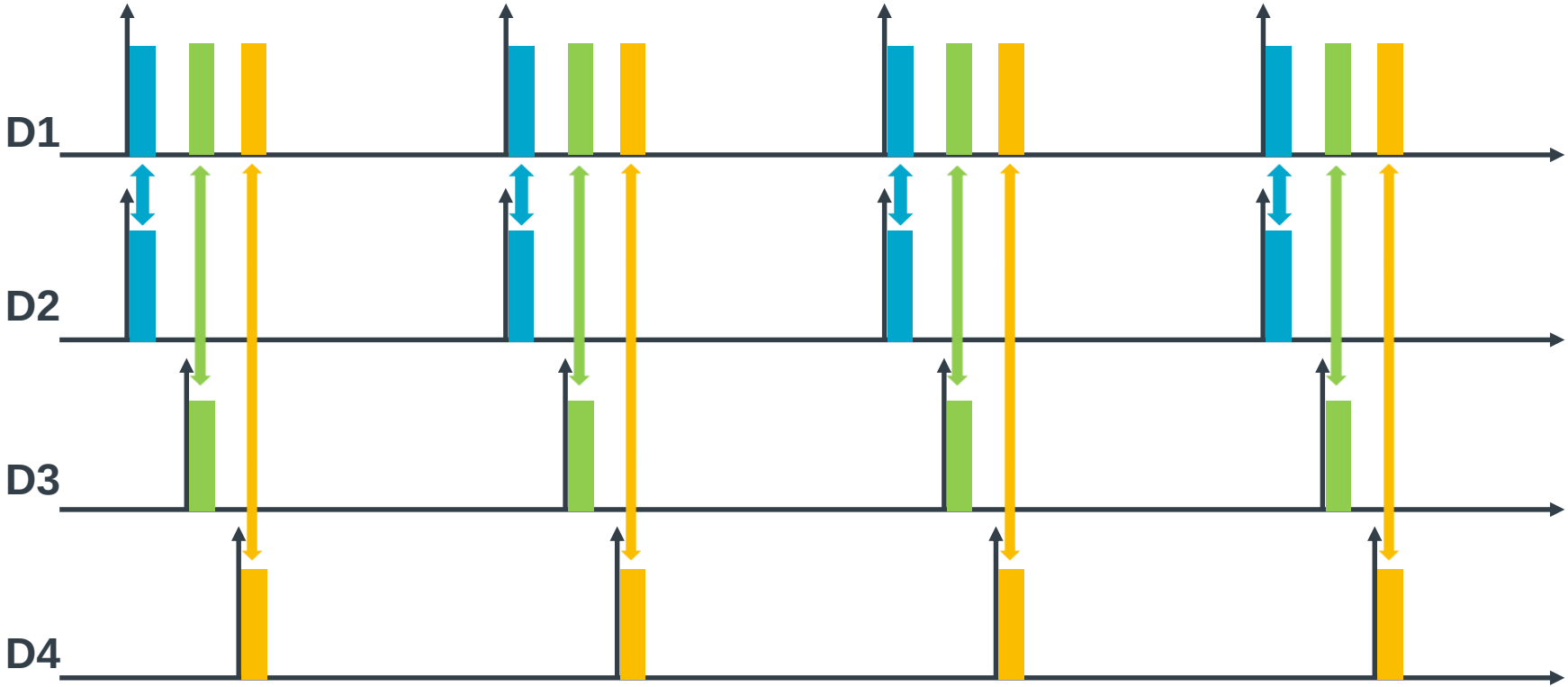
# Advertising/Scanning timing



# 1-Link connection timing

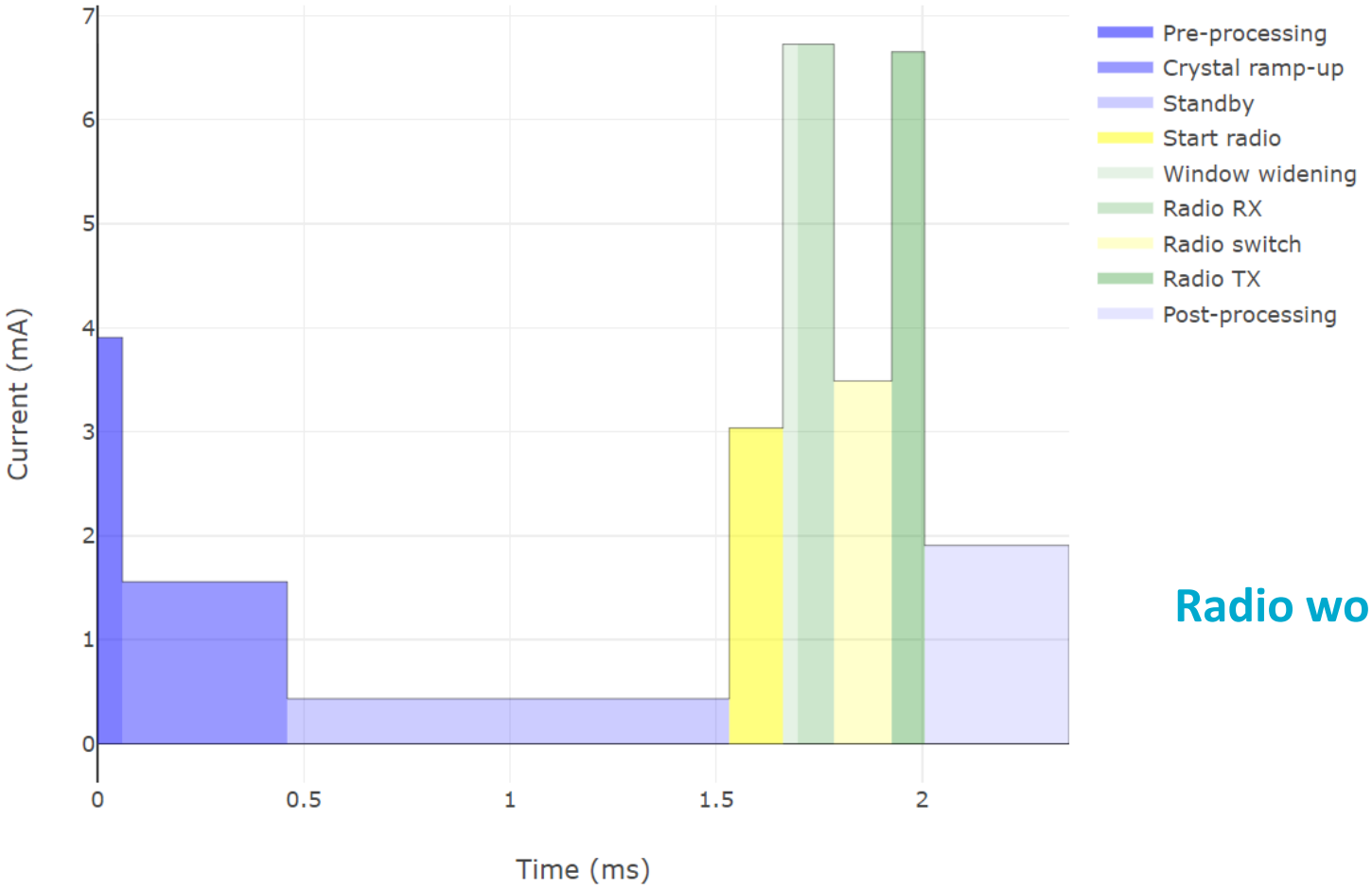


# Multi-Link connection timing



# Challenges for multilink implementation

- Fast radio startup and switch
- Good scheduling
- Accurate time slot
- Robustness
- Flexibility
- Nice and clean SDK
- …




# Radio working timing

# Radio performance


Fast radio startup

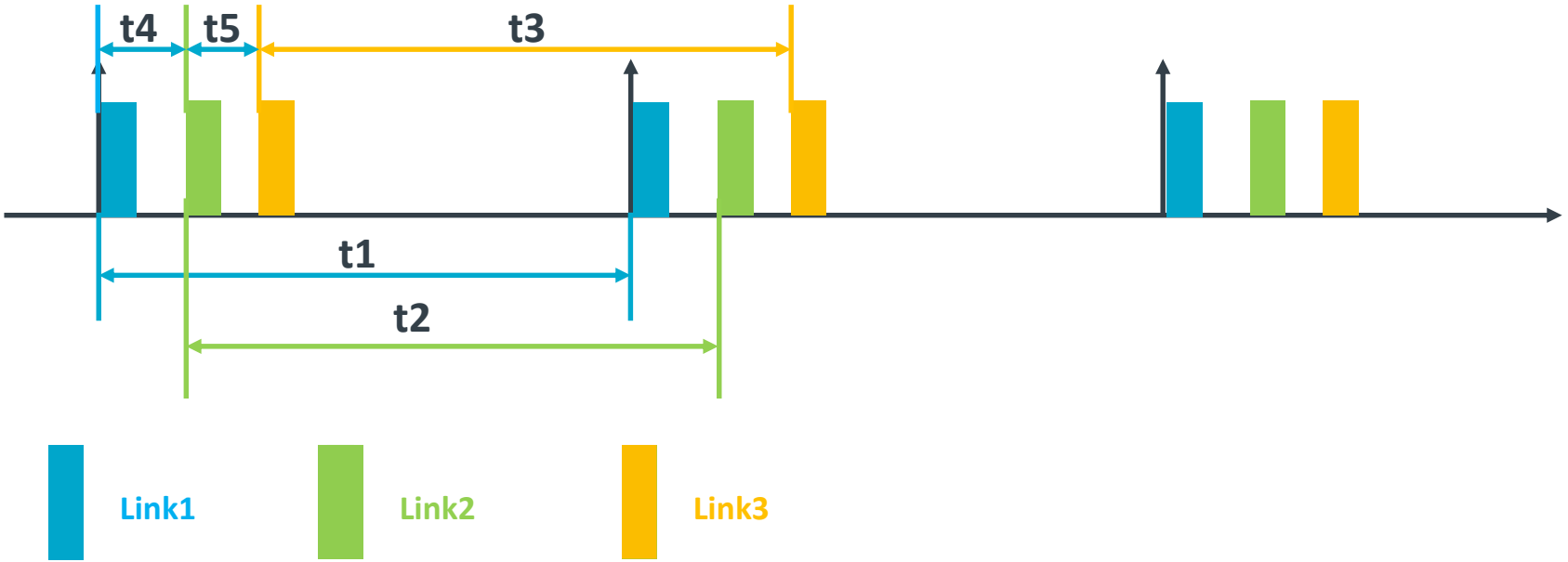

Fast radio switch


Fast execution


Radio Active


Idle w/ RTC

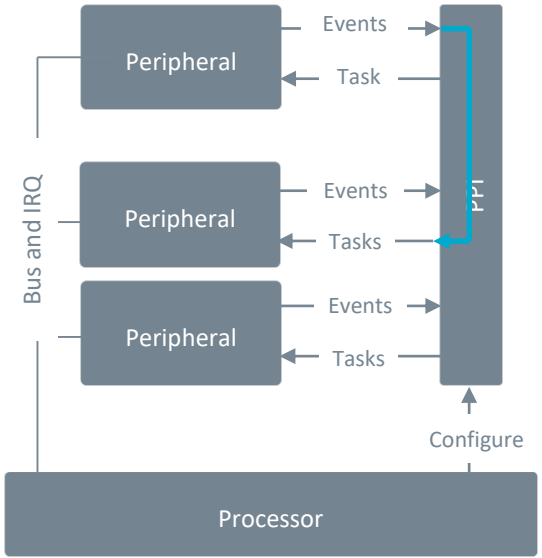
# Demanding Time requirement




**t1/t2/t3: connection interval**

**t4/t5: inter-link time**

# PPI-Programmable Peripheral Interconnect




Autonomous Peripheral Interaction

Complements Regular Interrupts

Independent Operation

Configurable Task – Events Mapping

Offload CPU

Power Saving

Real-time Operation

# PPI channels in Bluetooth protocol stack (Softdevice)

| Channel | EEP | TEP |
|---|---|---|
| 20 | TIMER0->EVENTS_COMPARE[0] | RADIO->TASKS_TXEN |
| 21 | TIMER0->EVENTS_COMPARE[0] | RADIO->TASKS_RXEN |
| 22 | TIMER0->EVENTS_COMPARE[1] | RADIO->TASKS_DISABLE |
| 23 | RADIO->EVENTS_BCMATCH | AAR->TASKS_START |
| 24 | RADIO->EVENTS_READY | CCM->TASKS_KSGEN |
| 25 | RADIO->EVENTS_ADDRESS | CCM->TASKS_CRYPT |
| 26 | RADIO->EVENTS_ADDRESS | TIMER0->TASKS_CAPTURE[1] |
| 27 | RADIO->EVENTS_END | TIMER0->TASKS_CAPTURE[2] |
| 28 | RTC0->EVENTS_COMPARE[0] | RADIO->TASKS_TXEN |
| 29 | RTC0->EVENTS_COMPARE[0] | RADIO->TASKS_RXEN |
| 30 | RTC0->EVENTS_COMPARE[0] | TIMER0->TASKS_CLEAR |
| 31 | RTC0->EVENTS_COMPARE[0] | TIMER0->TASKS_START |

# Scheduling timing-activities

- Only 1 radio peripheral
- Different timing-activities/events
  - Central role timing-activities
  - Peripheral role timing-activities
  - Advertiser timing-activities
  - Scanner/Initiator timing-activities
  - Flash Access timing-activities
  - Radio time-slot timing-activities
  - QoS channel survey timing-activities

# Scheduling principles

- Advertiser and broadcaster timing-events are scheduled as early as possible
- Peripheral link timing-events follow the timings dictated by the connected peer
- Central link timing-events are added relative to already running central link timing-events, close to each other (without any free time between them)
- If timing-events collide, their scheduling is determined by a priority system

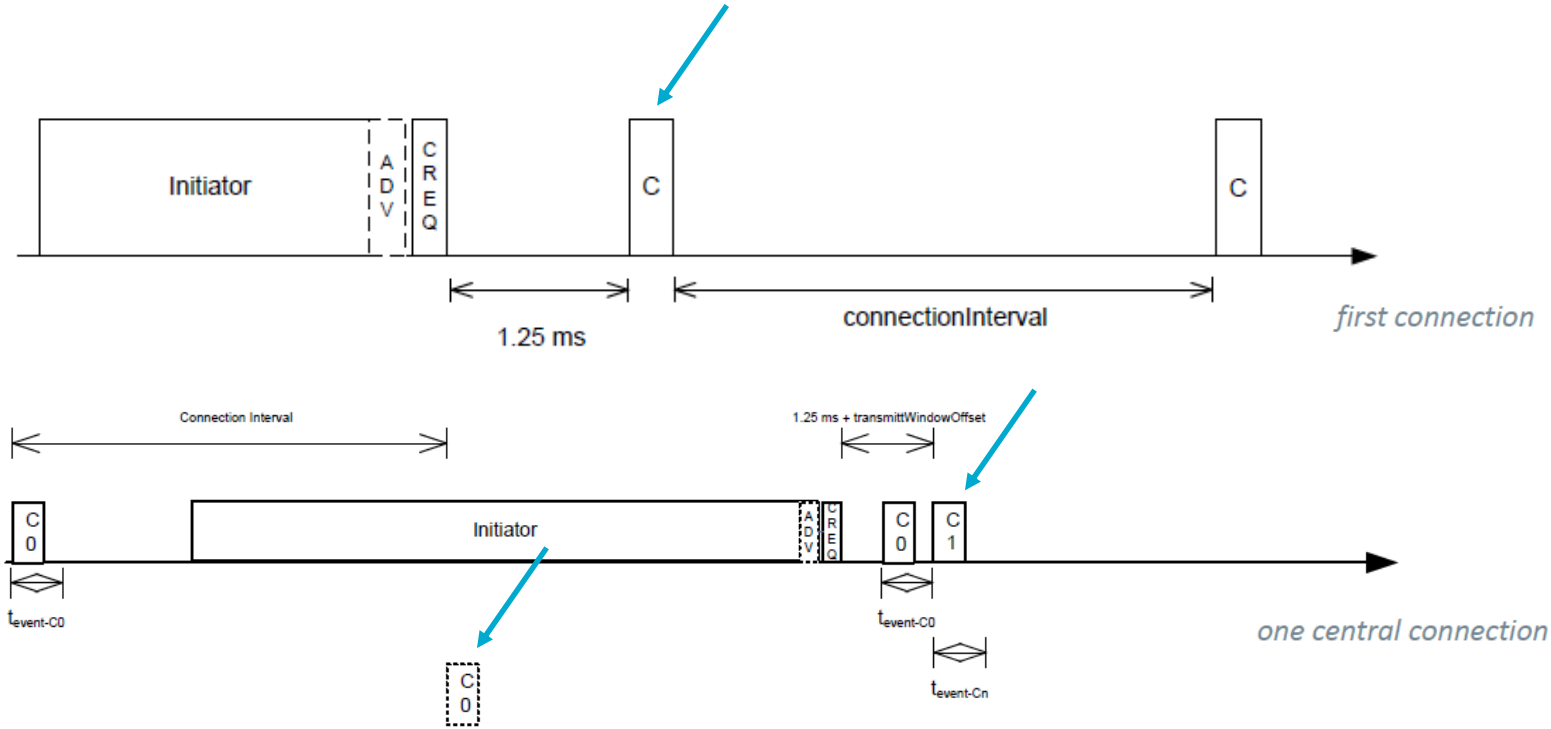
# Priority

| Priority (Decreasing order) | Role state |
|---|---|
| First priority | • Central connections that are about to time out<br>• Peripheral connection setup (waiting for ack from peer)<br>• Peripheral connections that are about to time out |
| Second priority | • Central connection setup (waiting for ack from peer)<br>• Initiator<br>• Connectable advertiser/Broadcaster/Scanner which has been blocked consecutively for a few times |
| Third priority | • All *Bluetooth* low energy roles in states other than above run with this priority<br>• Flash access after it has been blocked consecutively for a few times<br>• Radio Timeslot with high priority |
| Fourth priority | • Flash access<br>• Radio Timeslot with normal priority |
| Last priority | • QoS channel survey |

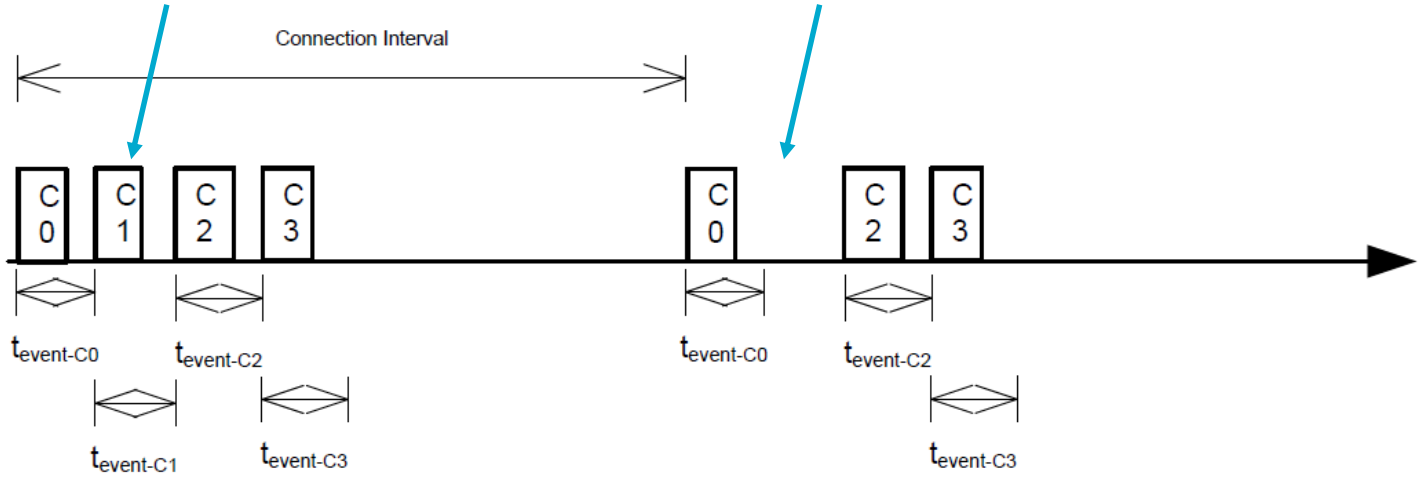
# Scheduling priorities

- Higher priority timing-activity get the timing-event if overlapped by lower priority timing-activity
- First come first served for the same priority timing-activities
- Once started the timing-event cannot be preempted by any other timing-activities
- The different timing-activities have different priorities at different times, dependent upon their states

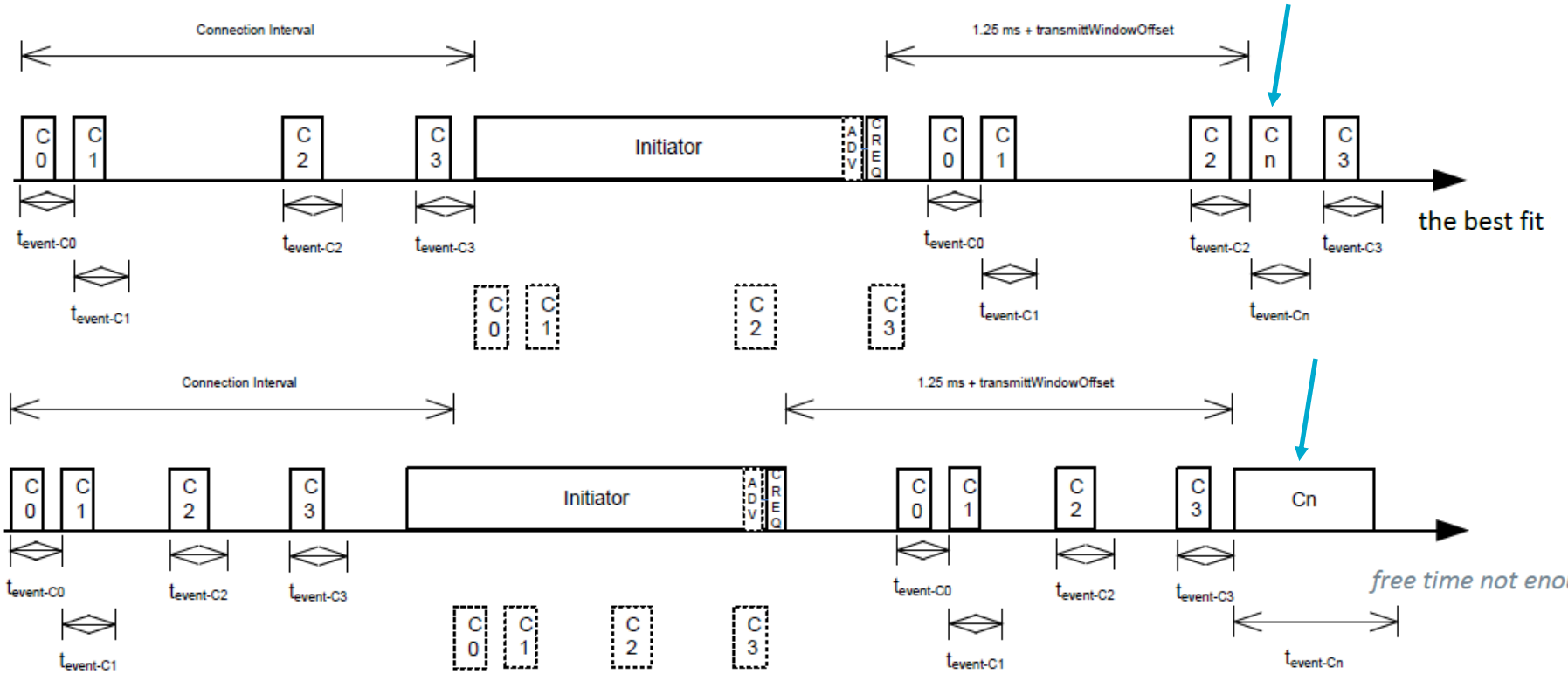
# Initiator timing 1



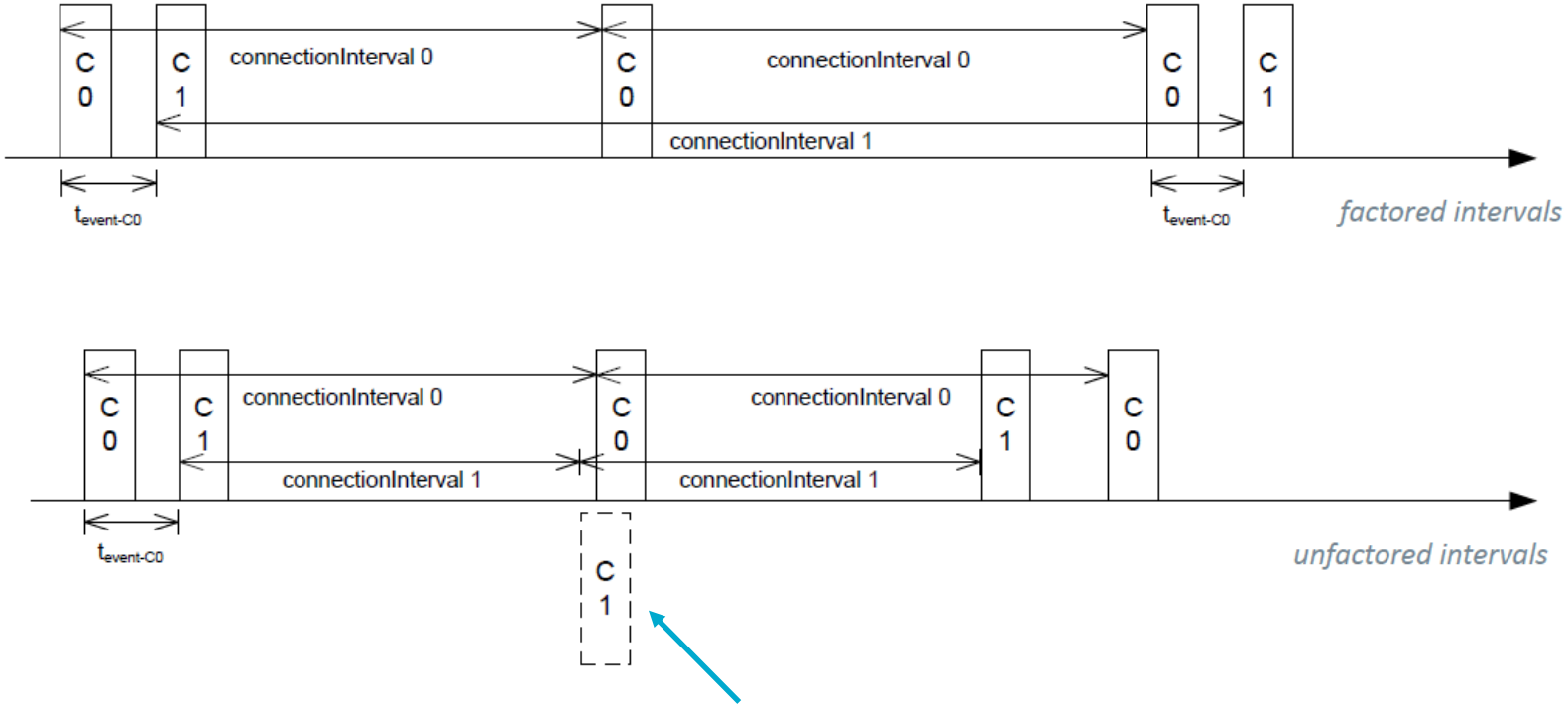
# Disconnection



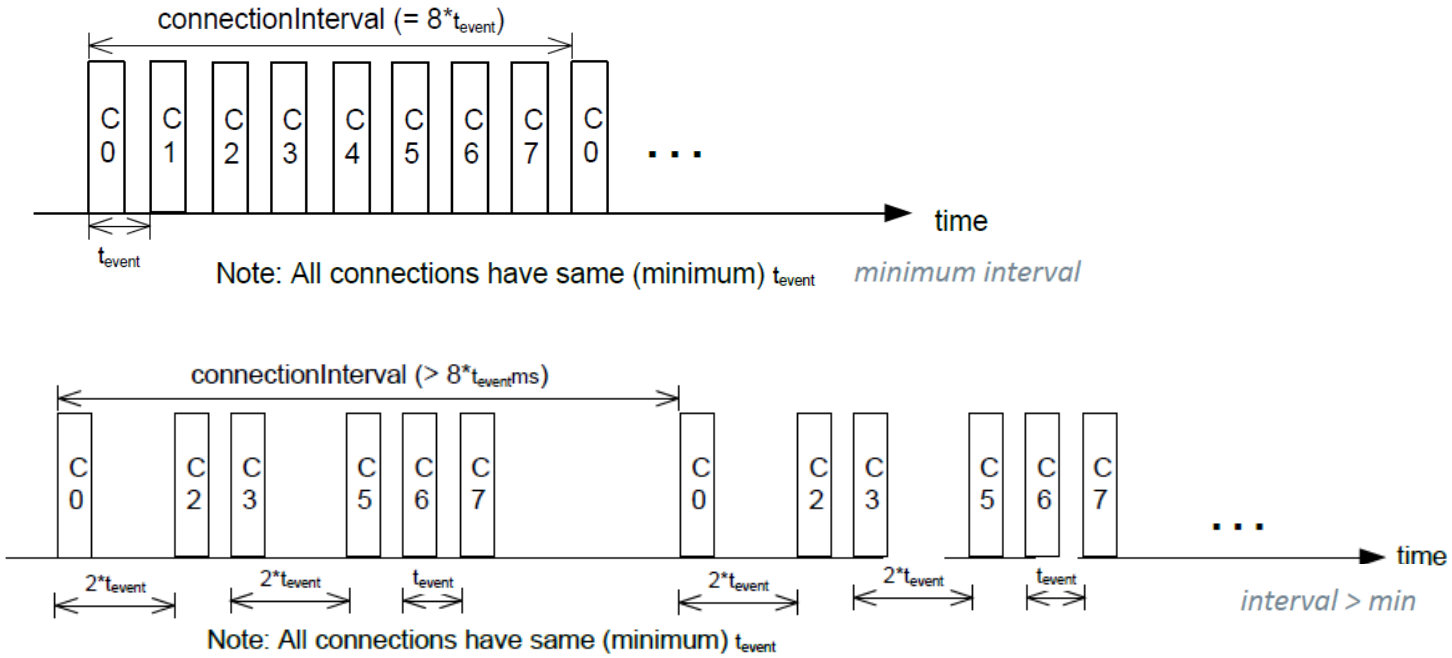
# Initiator timing 2



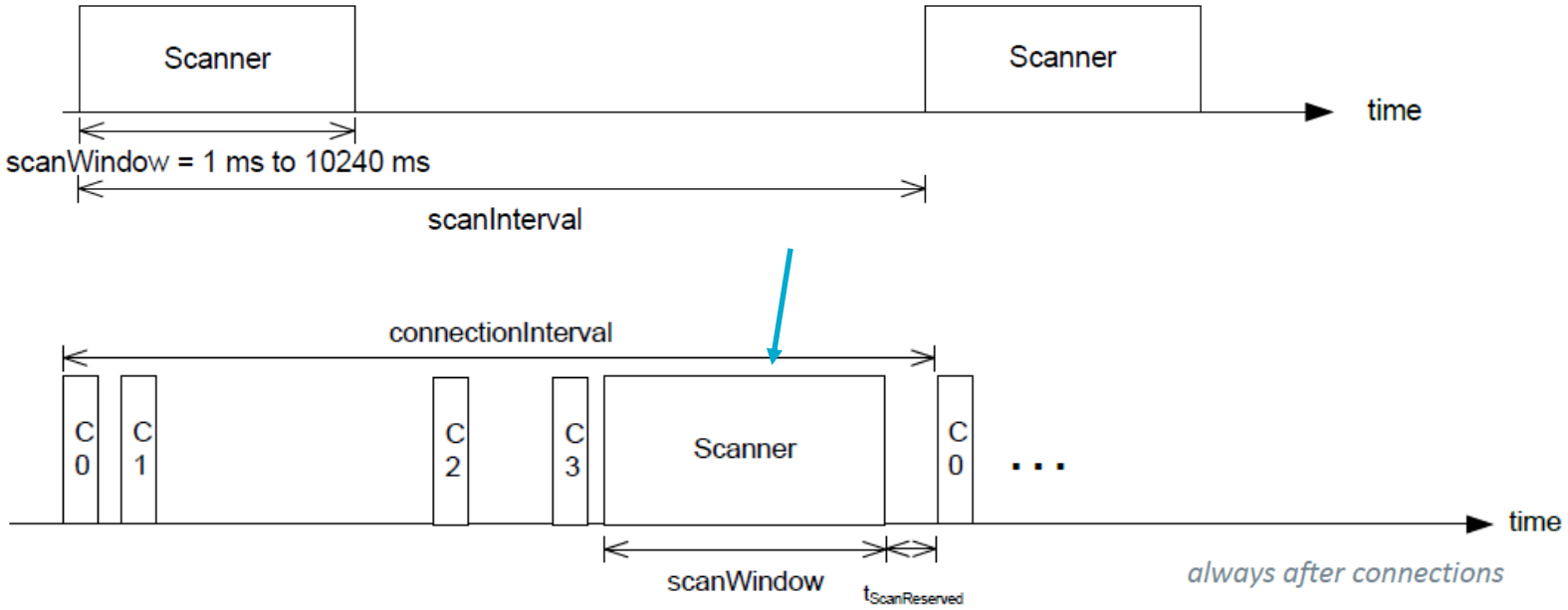
# Central connection timing 1



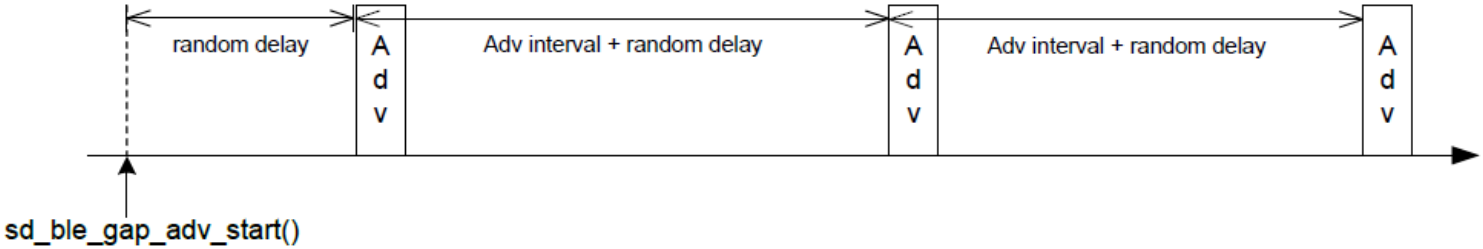
# Central connection timing 2

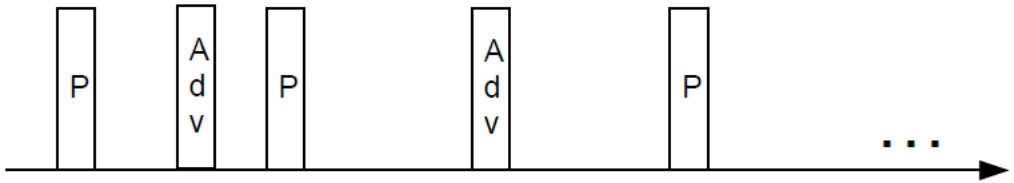


# Scanner timing



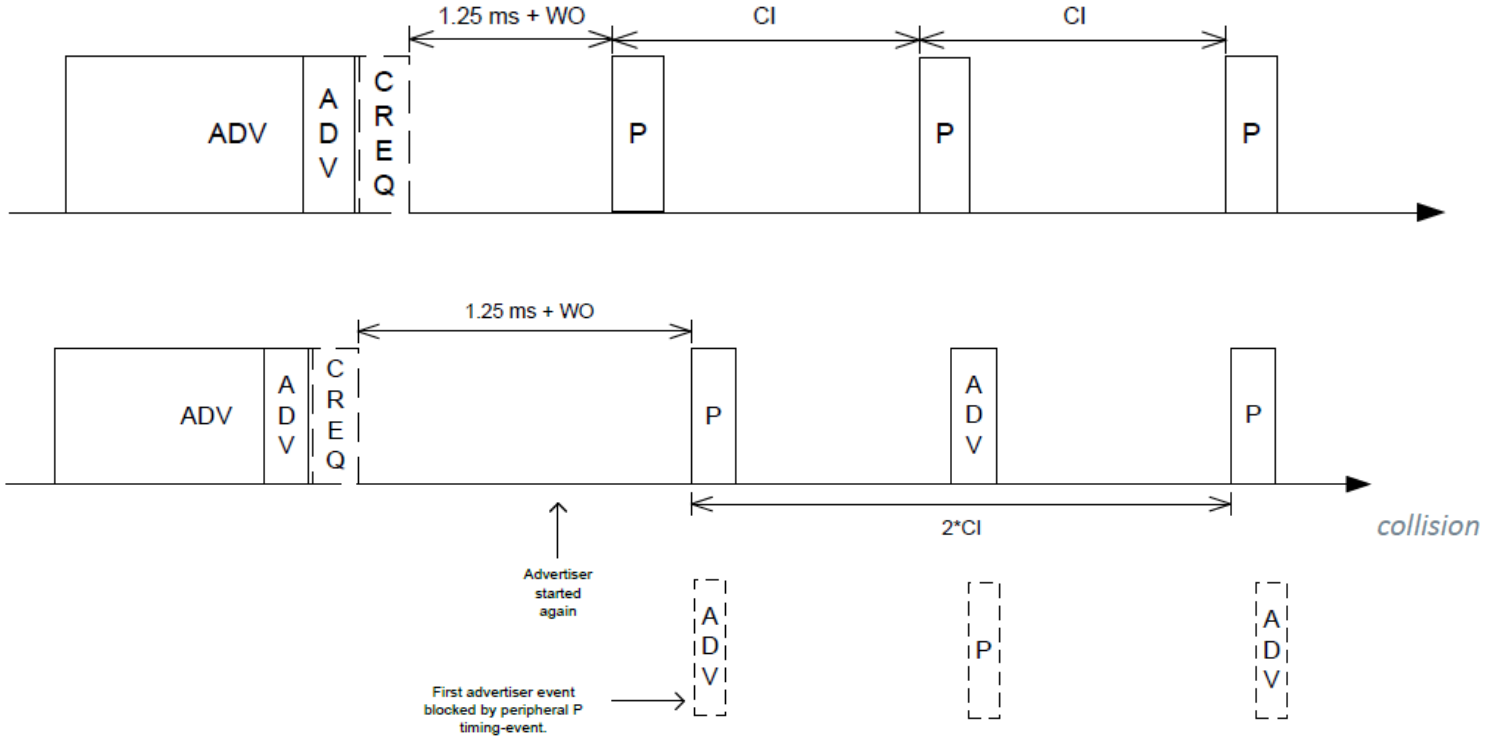
# Advertiser timing



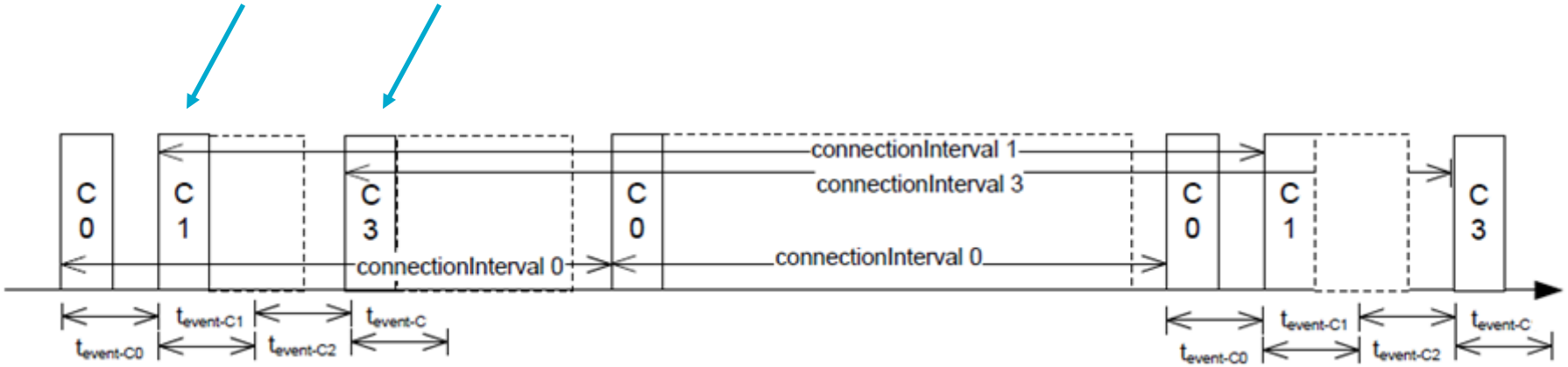




*Advertiser collision*

# Peripheral connection timing



# Connection event length extension



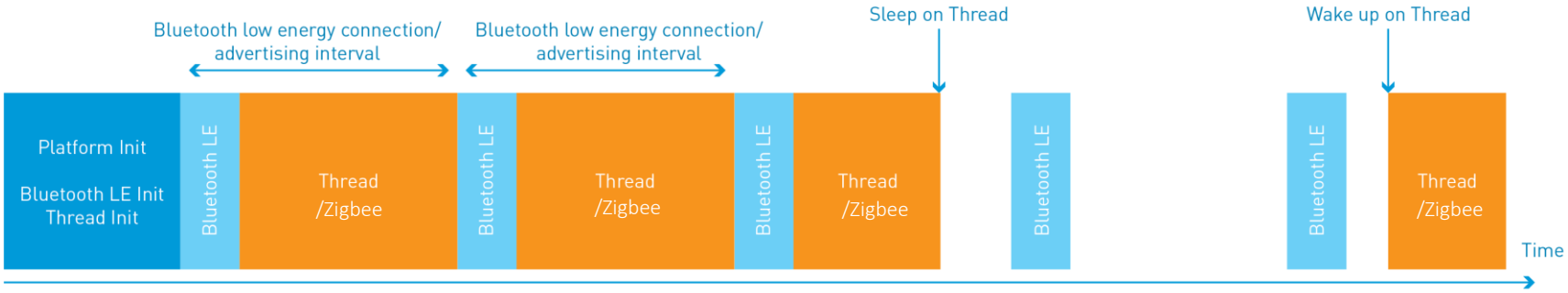
# Suggested intervals and windows

- Interval of all roles have a common factor which is ≥ Σtevent-Cx + (tScanReserved + ScanWindow) + tSlaveEventNominal + tAdvEventMax
- When long Link Layer Data Channel PDUs are in use, it is recommended to increase the event length of a connection
- A recommended configuration for having fewer colliding Peripherals is to set a short event length and enable the Connection Event Length Extension in the SoftDevice
- Set the supervision time-out for connections long enough to avoid loss of connection when packets are dropped

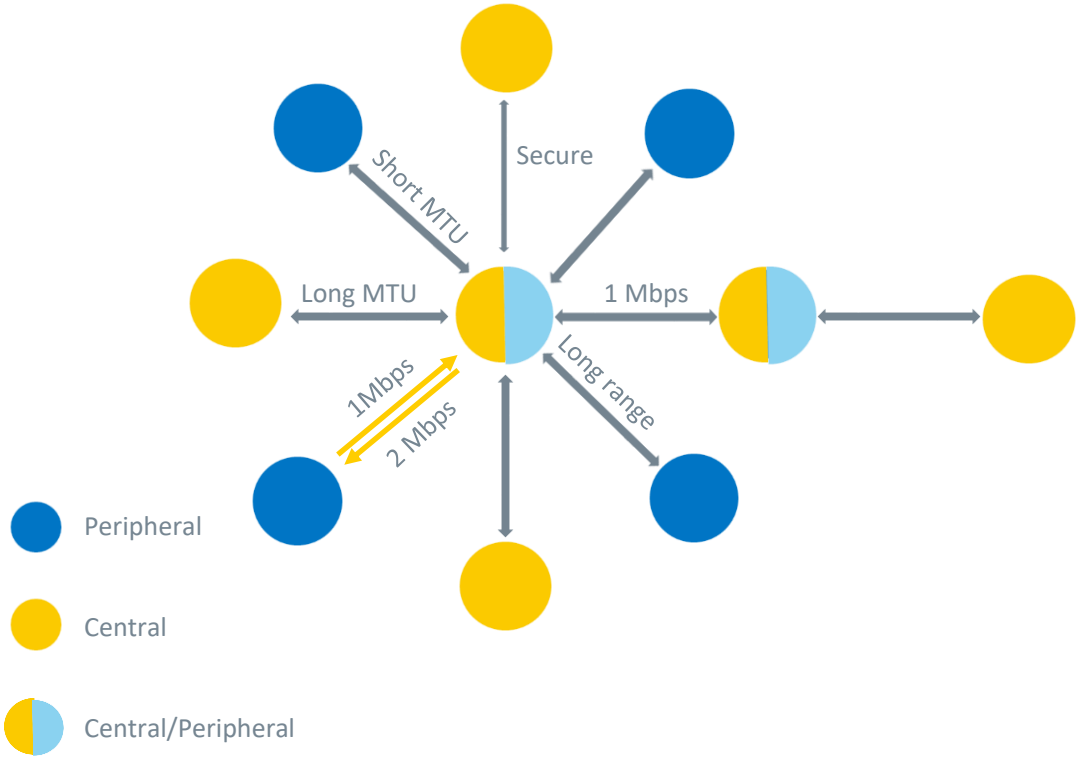
# Time-slot API

- Multilink connotation can extend to Zigbee/Thread/2.4G
- Time slot API is used to multiplex radio resource
- Time slot is a part of softdevice scheduling mentioned previously
- Concurrent LE and Zigbee/Thread/2.4G

# Concurrent LE and other protocols



# Flexible Bluetooth Multilink




- Concurrent master and slave
- Concurrent client and server
- Configurable MTU for each link
- Configurable PHY for each link
- Secure the link or not
- Up to 20 active links

# Easy to use

- Complicated multilink doesn’t mean complicated SDK
- To start a new advertising or a new peripheral link, call **sd_ble_gap_adv_start** no matter how many running links are
- To start a new scanning or a new central link, call **sd_ble_gap_scan_start** no matter how many links are running
- That’s it. Let softdevice handle the rest for you!

# Agenda

1 **Bluetooth multi-link concepts and implementations**

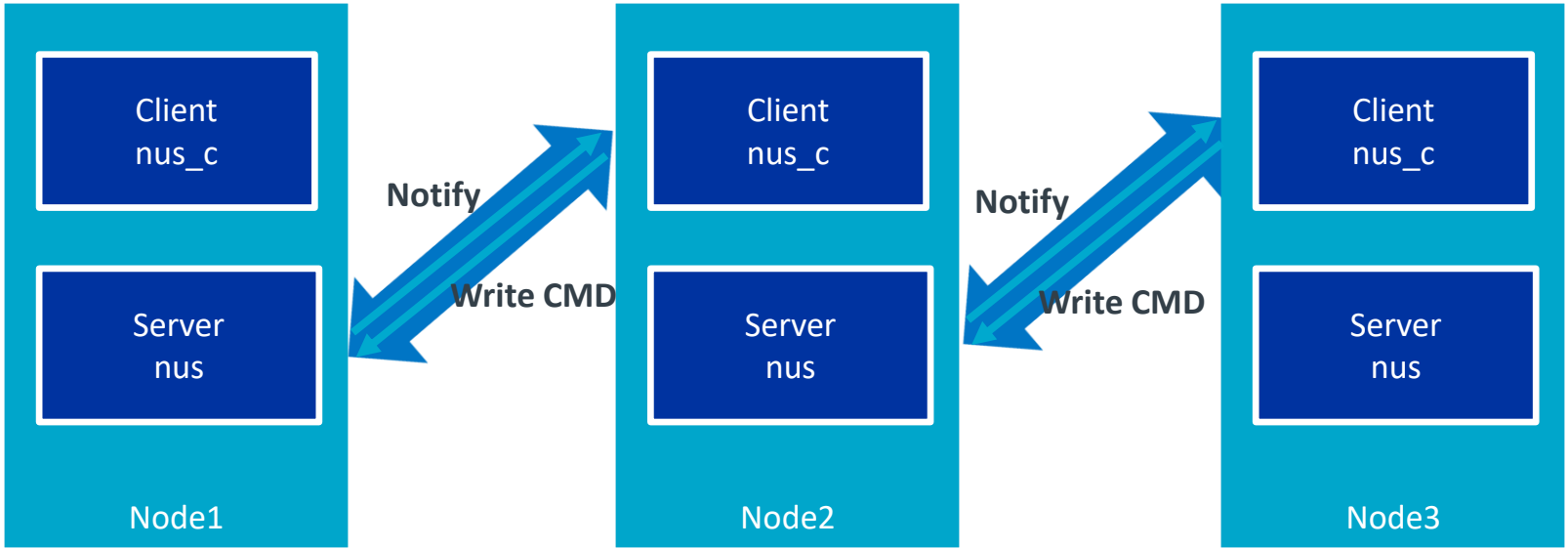
2 **True low power relay network and mesh design**

3 **Bluetooth multi-link use case study**

# Challenges when building a relay network by Bluetooth multilink

- All the nodes have the same firmware
- Master/Slave asymmetry
- Client/Server asymmetry
- One link for one pair
- Determined roles for the specific link

# ATT transparent communication (NUS)



# Interactive API abstraction

- Server -> Client, sd_ble_gatts_hvx(conn_handle, &hvx_params);
- Client -> Server, sd_ble_gattc_write(conn_handle, &write_params);
- conn_handle, connection handle, to identify a link
- nus_send(conn_handle,p_data,length)
- Be aware that server can send/receive data to/from client, client can send/receive data to/from server as well. The difference between them is who is the owner to build ATT database

# Relay topology

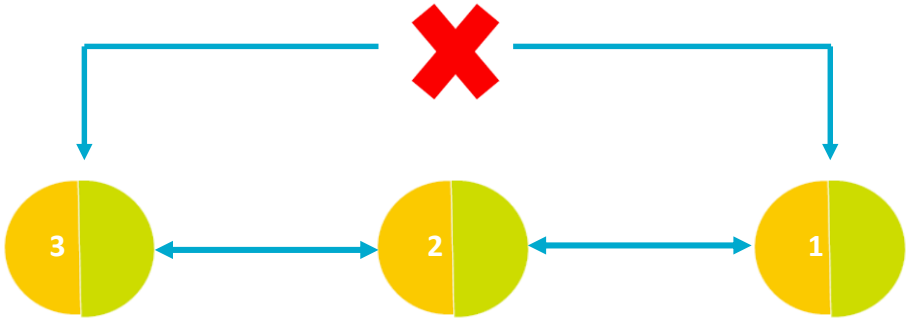
# Challenges upgrading relay to mesh

- Select a provisioner – first master device
- Assign an unique address for each node – provisioner coordinate
- Distinguish provisioned device and un-provisioned device – advertising name
- Routing or reconnection – name and device address filter
- Recover – name filter and authentication data persistent

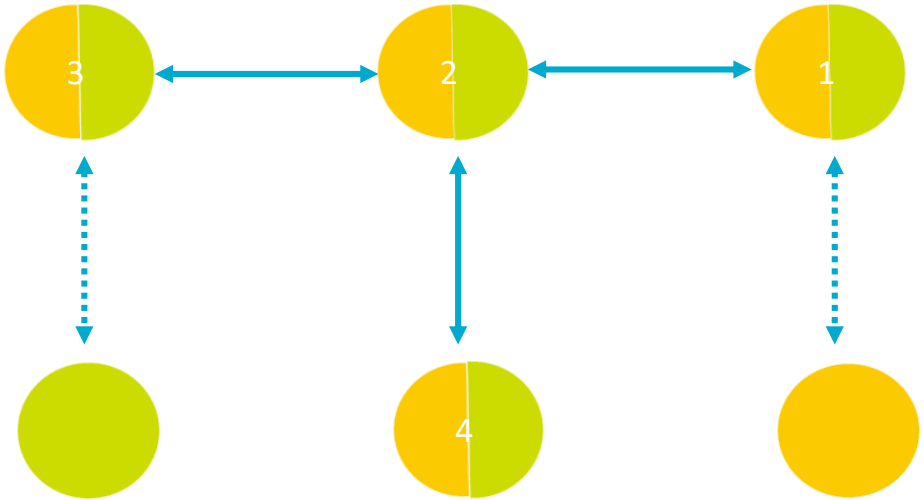
# Building a GATT mesh example 1

- 2 central link 1 peripheral link
- Use NUS and NUS_C for GATT interaction
- Use HRS_c for secure link
- Input command by UART
- Print logs by RTT

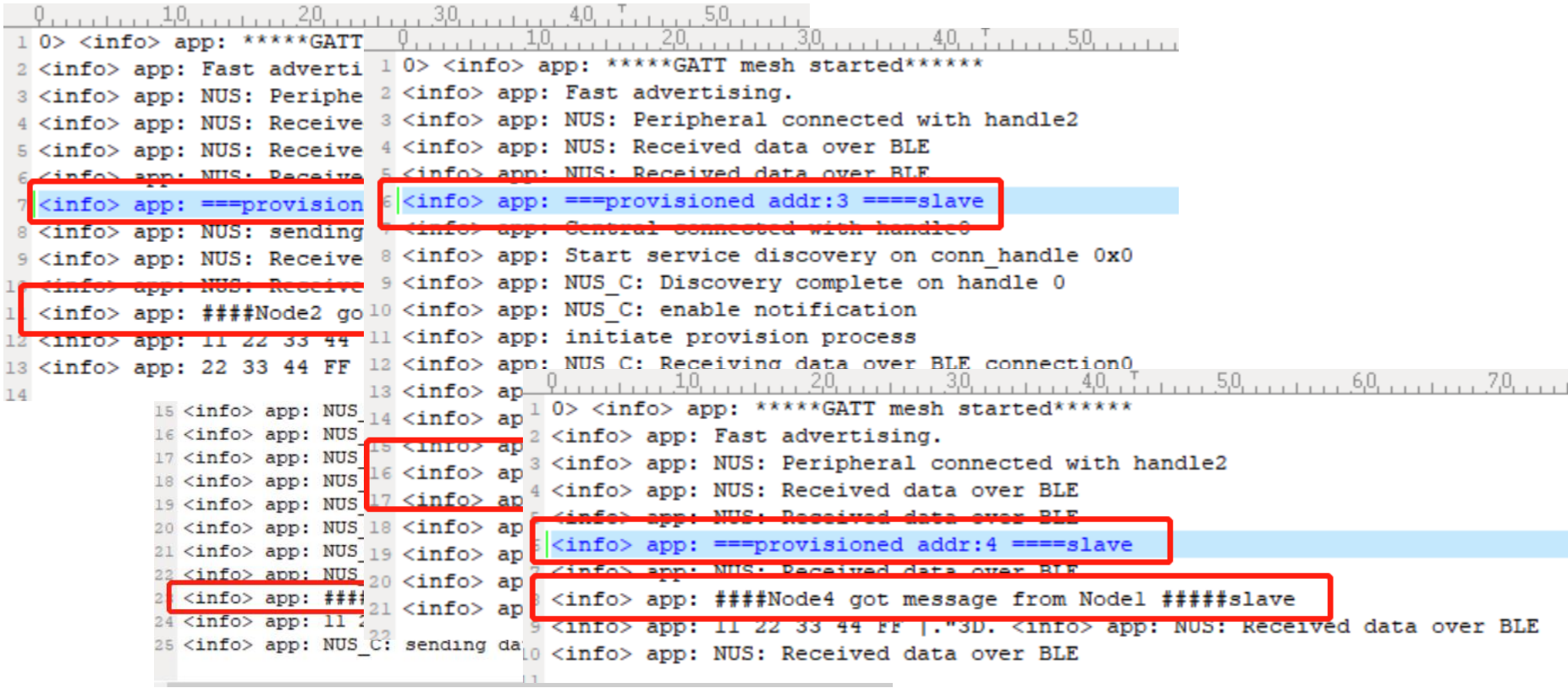
# Building a GATT mesh example 2



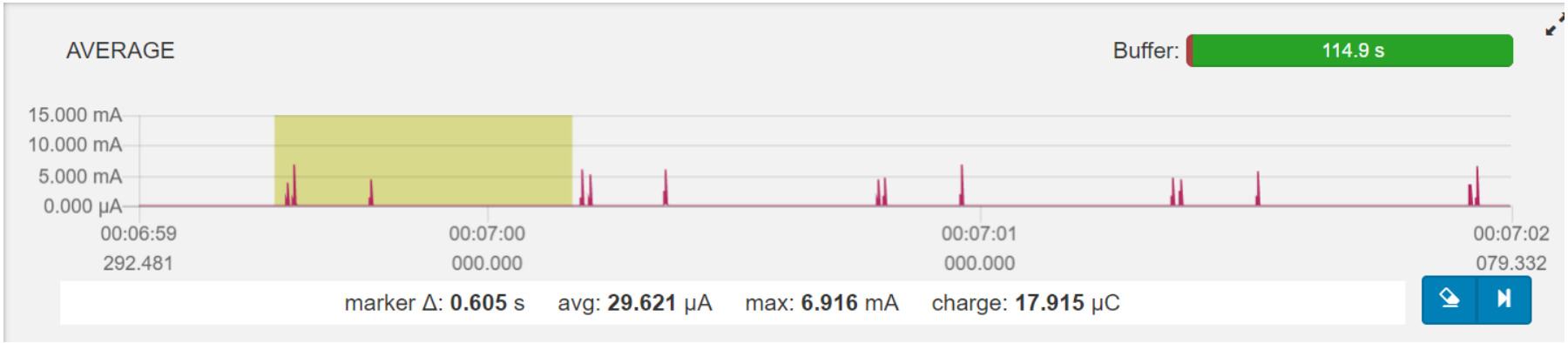
# The Mesh example topology



# Demo logs



# Node2 power consumption



# GATT mesh advantages

- All low power nodes
- Specified ‘physical’ routing
- Fast setup
- Best of class security. Each link is secured separately
- Easy to use and deploy
- Compatible with other LE apps

# Agenda

1 **Bluetooth multi-link concepts and implementations**

2 **True low power relay network and mesh design**

3 **Bluetooth multi-link use case study**

# ESL AP

- Advertising/Peripheral.
  Interact with ESL
- Scanning/Central.
  Discover other devices

# Smart door lock

- Peripheral for configuration and unlocking
- Central for unlocking
- Concurrent Zigbee to interact with smart home system

# Smart power meter

- Master role connecting to breakers
- Slave sole connecting to phone or aggregator

# Smart Bus Display

- 4 screens or more are interconnected
- 4 master link 1 slave link
- or GATT Mesh

# Night light relay

- Bluetooth multilink relay network
- Phone configuration
- Low power

# 蓝牙多连接，释放无限想象

- 构建基于GATT的全网低功耗节点的私有Mesh


- Kevin Ai
- kevin.ai@nordicsemi.no

- 5/23/2019