BBC: Enabling BLE to Support Bluetooth Classic

Hsun-Wei Cho Kang G. Shin

Real-Time Computing Laboratory

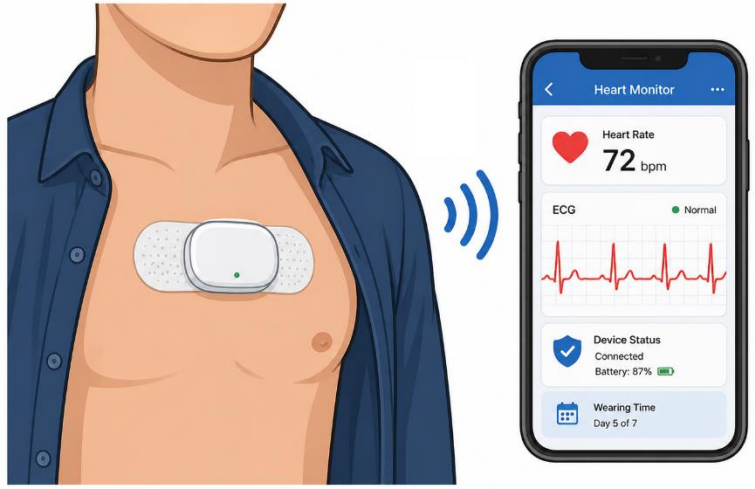
University of Michigan

Outline

- Overview
- System Design
  - Architecture
  - Connection Establishment
  - Two-way Communication
  - Reliable Packet Delivery
  - Authentication, Encryption & Transport Support
- Implementation and Evaluation
- Conclusion

Bluetooth devices are ubiquitous

- Headphones, speakers, fitness trackers, medical devices
- ~5B devices are shipped every year [1]



[1] https://www.bluetooth.com/2025-market-update/trends/

Bluetooth has two distinct protocols

- **Bluetooth Classic:** Original standard; dominant in audio streaming
- **BLE:** Low-power standard; in active development

Bluetooth Classic

Bluetooth Low Energy (BLE)

More and more Bluetooth devices rely on BLE

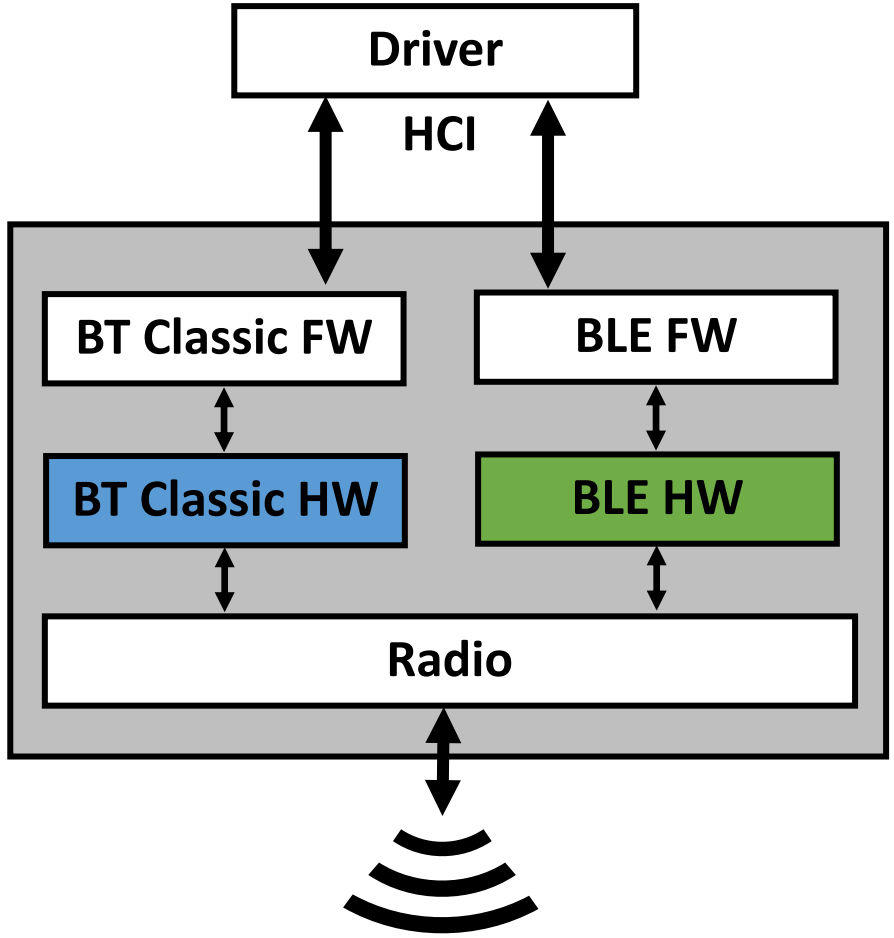
- Single-mode BLE devices expected to grow 22% annually [1]
  - 10x the growth of any other types
- New standard releases heavily focus on BLE
- No significant development for Bluetooth Classic
- **However, most audio streaming devices (80 to 85% in 2024 [2]) still rely on Bluetooth Classic**

Single-mode BLE devices are not compatible with Bluetooth Classic protocols

[1] https://www.bluetooth.com/2025-market-update/trends/
[2] https://www.bluetooth.com/2024-market-update/#audio-streaming

Existing solutions do not close this gap

- LE Audio
  - **Optional** feature
  - Requires BT 5.2 (2020)
  - New audio codec with high compression ratios
- Dual-mode chips
  - More complicated than single-mode BLE chips
  - Not applicable to fast-growing single-mode BLE devices



Typical implementations [1] of “dual-mode” chips

[1] https://t-2-m.com/semiconductor-ip-core/bluetooth-dual-mode-5-3-hw-linklayer-baseband-controller-ip

Problem: *More devices use BLE but most headphones use Bluetooth Classic*

Question: *Can we use BLE(-only) chips for Bluetooth Classic communication?*

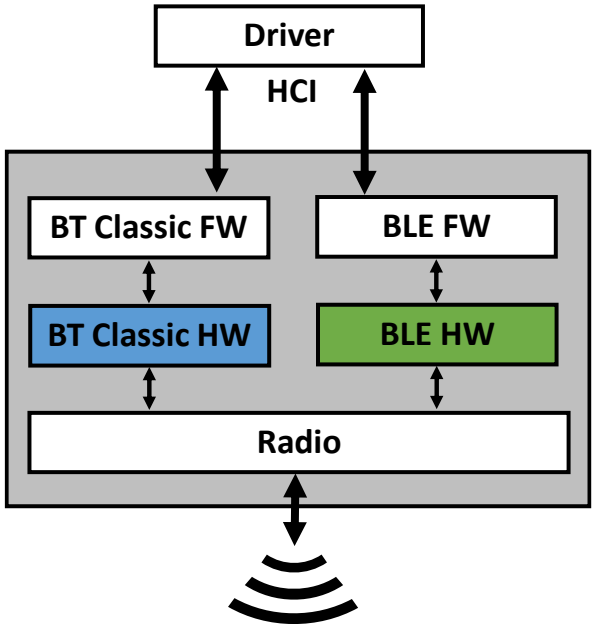
Solution: *BBC*

Future devices can use simple BLE chips
Current BLE devices can now stream audio

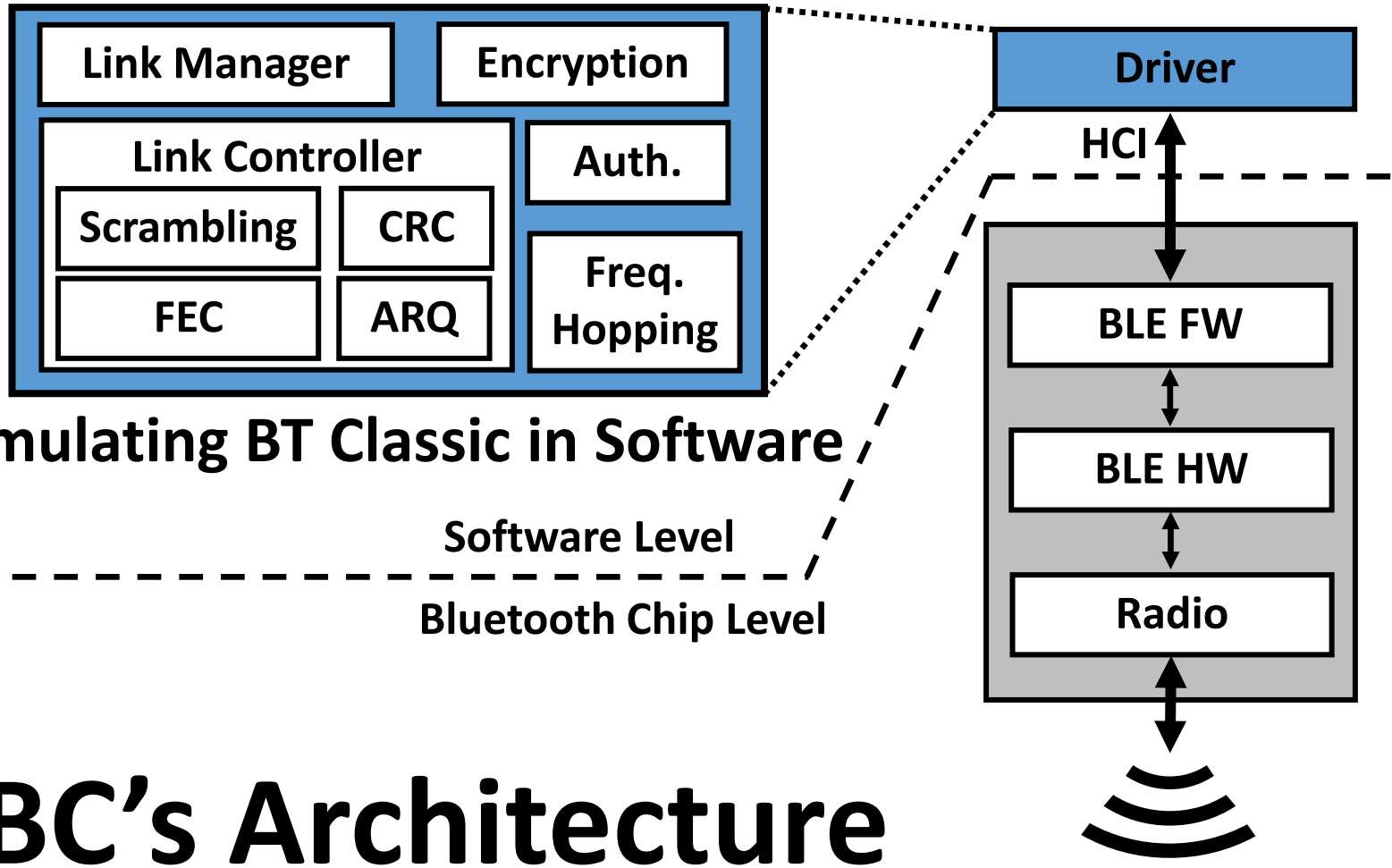
Outline

- Overview
- System Design
  - Architecture
  - Connection Establishment
  - Two-way Communication
  - Reliable Packet Delivery
  - Authentication, Encryption & Transport Support
- Implementation and Evaluation
- Conclusion

Architecture

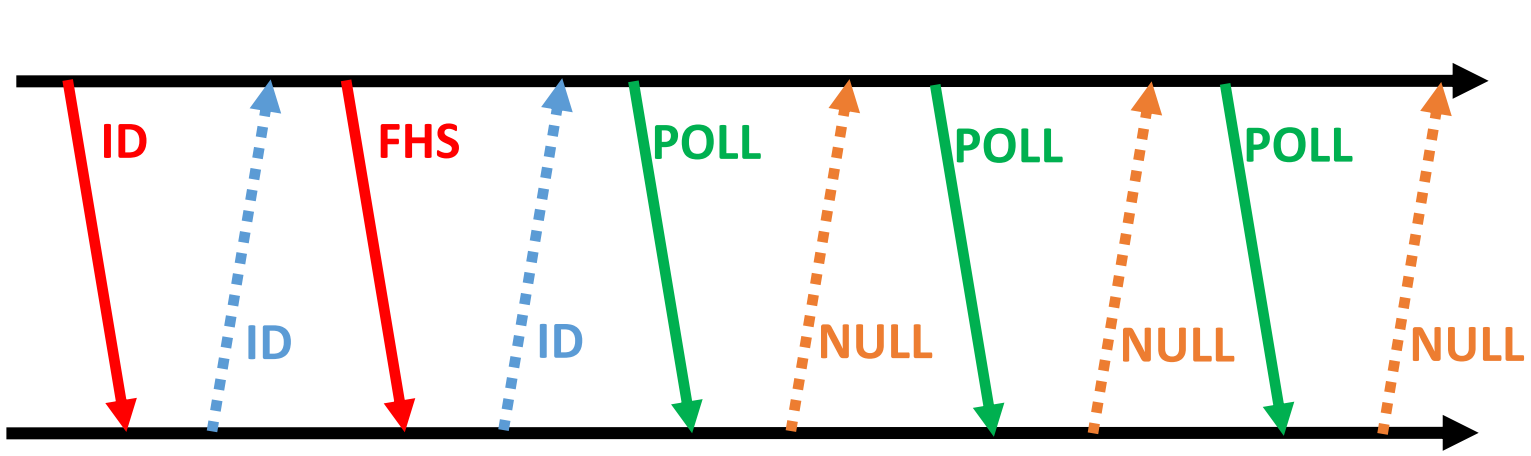


Traditional Architecture



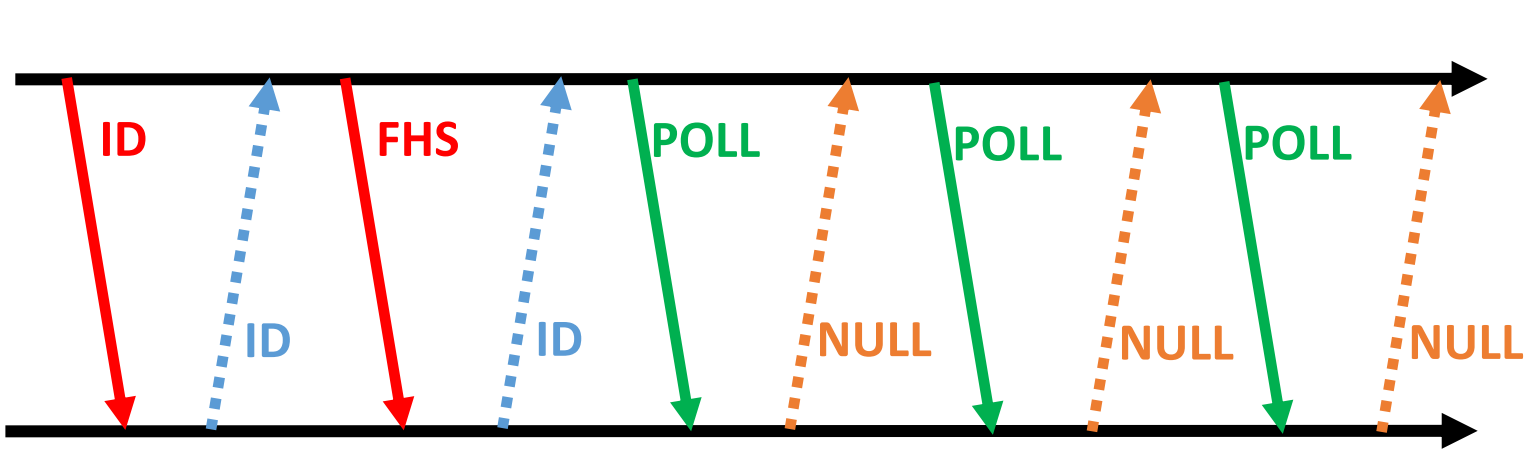
BBC's Architecture

Connection Establishment



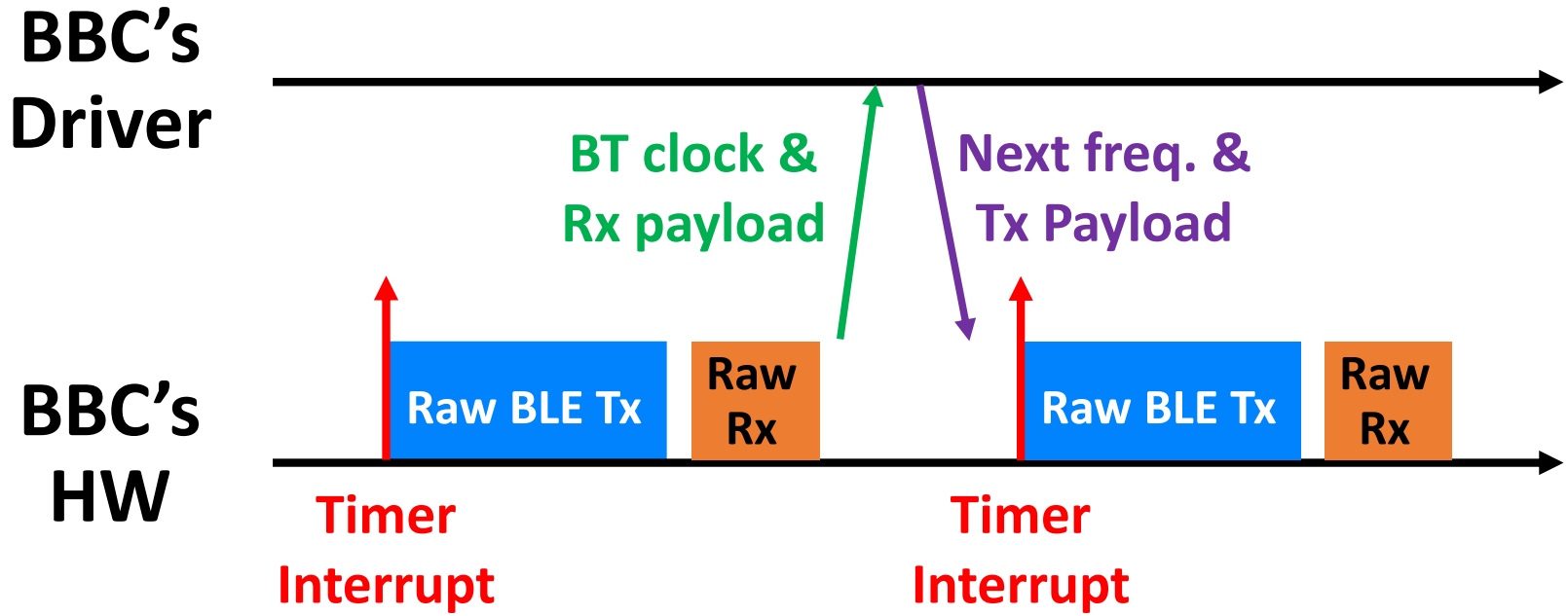
- Typical **paging** is a highly random process handled by HW:
  - Pseudo-random frequency hops
  - Pseudo-random bit processing
  - Multiple pseudo-random hopping sequences
- BLE chip lacks HW to handle these in real time

Connection Establishment



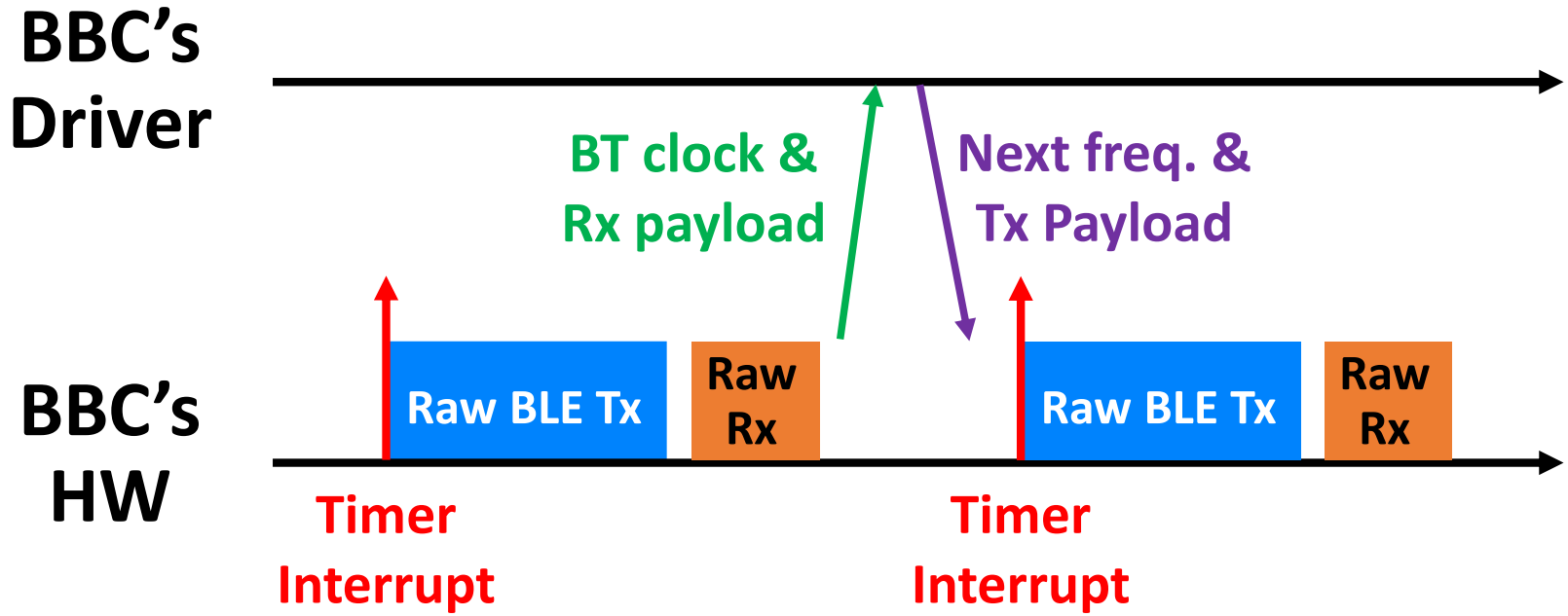
- Observations:
  - BBC has complete control over ID, FHS, Central's address, and POLL
  - Paging can be made deterministic
- Solution: BBC pre-generates valid frequency hops and packet bits
- Enable BLE chip to page headphones without dedicated HW

Two-way Communication: Packet-level



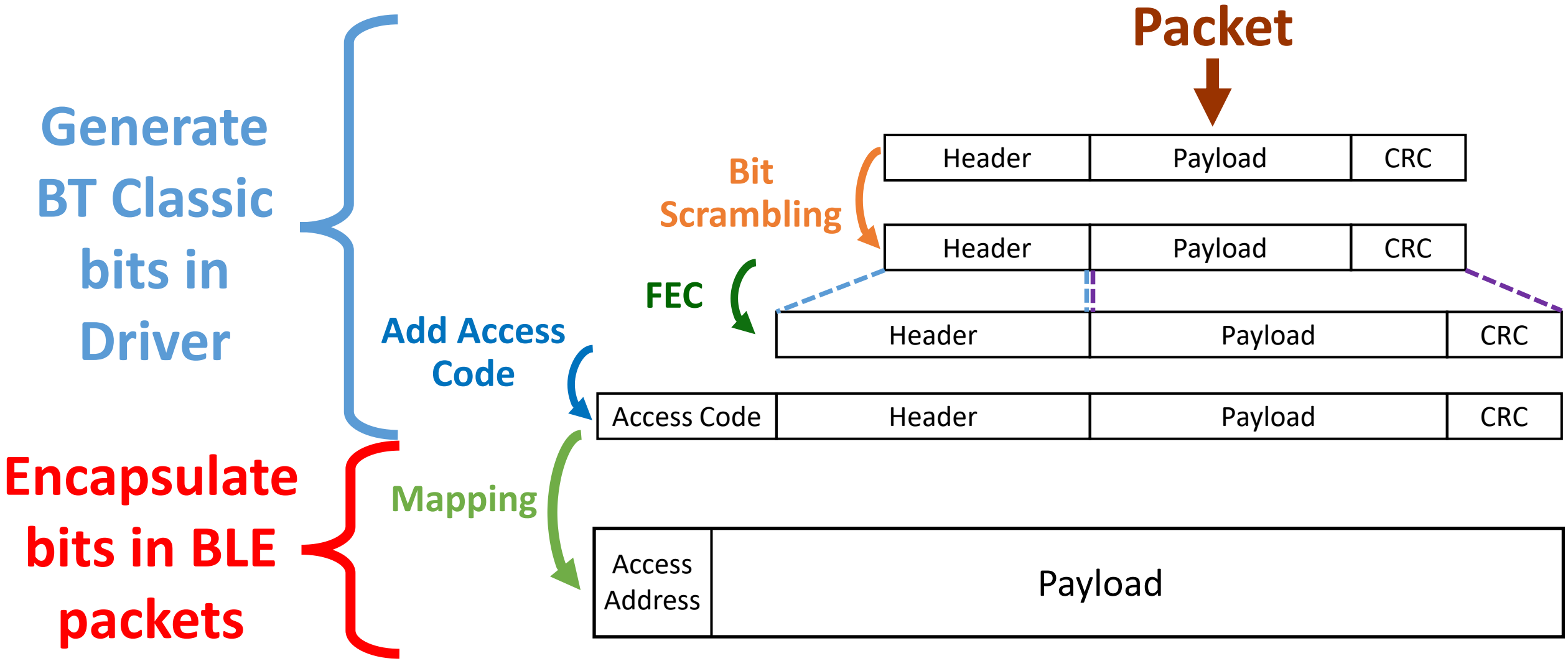
- Bluetooth Classic packets must align with each time slot
- Time slot's clock value is used in:
  - Bit processing and generation
  - Frequency hopping computation

Two-way Communication: Packet-level

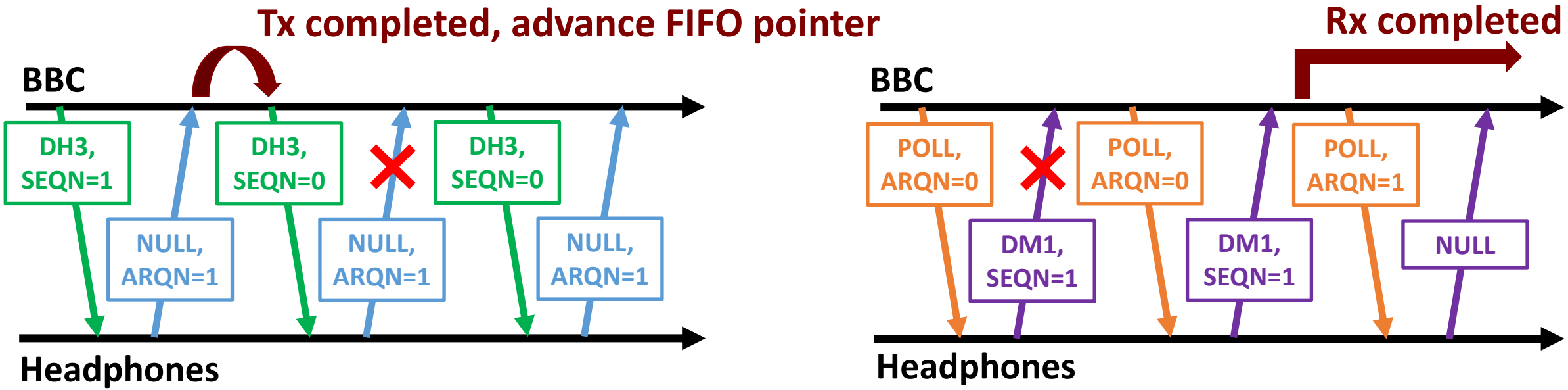


- BBC uses a "ping-pong" design
  - Use a HW timer to ensure precise time alignment
  - HW sends timer value and Rx payload to Driver
  - Driver processes Rx payload; sends Tx payload and frequency hops to HW
- Enable SW processing while maintaining HW's time precision

Two-way Communication: Bit-level



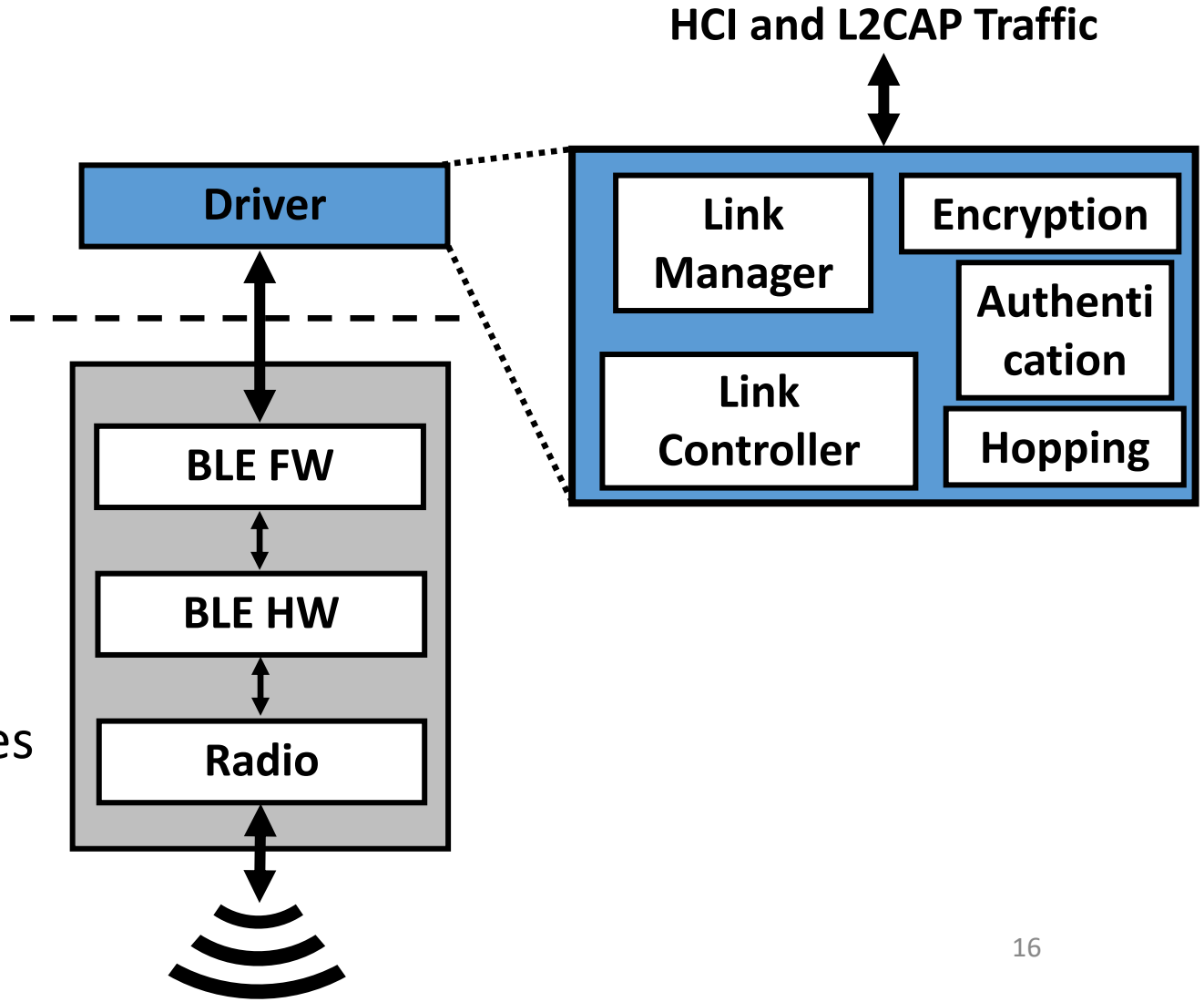
Reliable Packet Delivery



- Outgoing packets are queued in a FIFO
- SEQN and ARQN bits are used for reliable communication
- BBC implements SEQN/ARQN logic in SW

Authentication, Encryption & Transport Support

- Emulate the **Link Manager**
  - Link connection handshakes
  - Link capability negotiations
- Handle **authentication**
  - Two rounds of SAFER+ ciphers
- Handle **encryption**
  - Implement hash functions
  - Implement the E0 algorithm
  - Pre-calculate encryption sequences
- Support HCI requests and L2CAP layer

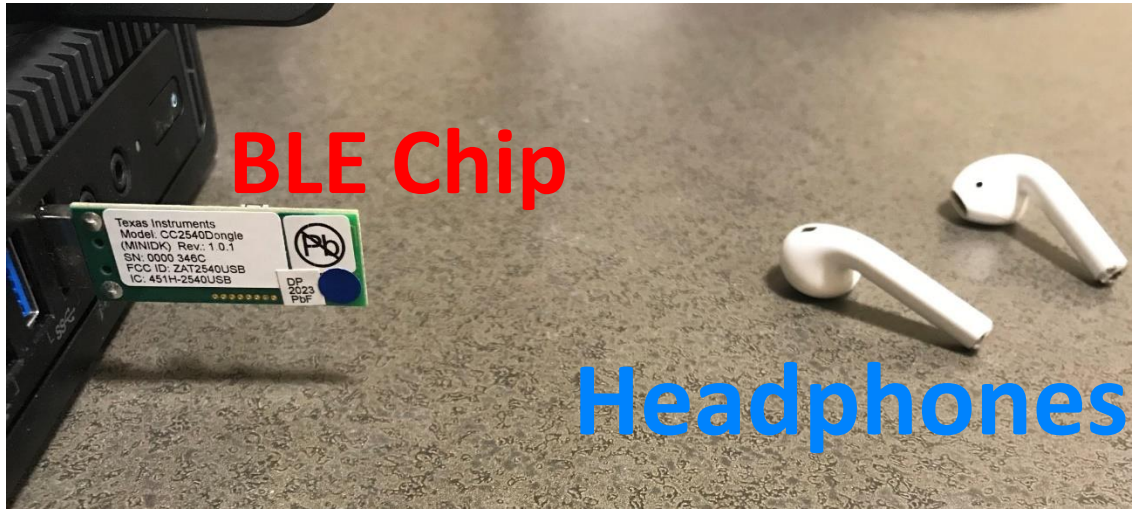


Outline

- Overview
- System Design
  - Architecture
  - Connection Establishment
  - Two-way Communication
  - Reliable Packet Delivery
  - Authentication, Encryption & Transport Support
- Implementation and Evaluation
- Conclusion

Implementation and Evaluation

- Hardware: Texas Instruments CC2540
  - BLE single mode
  - Bluetooth 4.0 and **no LE audio support**
- Wrote custom Kernel Driver on Ubuntu 20.04
- Use Teledyne LeCroy FTS4BT to measure PER and throughput
- Use unmodified headphones to measure audio streaming performance



Headphones	Bluetooth Chip Used
Sennheiser CX150	Qualcomm QCC3024
Sony SBH20	CSR CSR8640
Apple Airpods 2	Apple H1

Evaluation: Packet Error Rate and Throughputs

Packet Type	POLL			DM1			DH3		
Distance	1m	5m	10m	1m	5m	10m	1m	5m	10m
Correctly Received Packets	4094	4087	4077	4094	4075	4074	3965	3954	3897
Packet Error Rate (%)	0.05	0.22	0.46	0.05	0.51	0.54	3.20	3.47	4.86
Throughputs (kbps)	No payload			54.37	54.12	54.11	566.87	565.30	557.15

- Use FTS4BT to receive 4096 packets
- DM1: 17 bytes
- DH3: 183 bytes
- Throughput is sufficient to support A2DP at high quality

Evaluation: System Performance

Sennheiser CX150BT

	1m		5m		10m	
	Ack'd/Sent	PER (%)	Ack'd/Sent	PER (%)	Ack'd/Sent	PER (%)
POLL	8714/8825	1.26	8472/8617	1.68	8276/8549	3.19
DM1	18/18	0.00	18/18	0.00	19/19	0.00
DH3	14709/15157	2.96	14723/15365	4.18	14588/15432	5.47
Total	23441/24000	2.33	23213/24000	3.28	22883/24000	4.65
Total Payload	2490987 Bytes		2493419 Bytes		2470472 Bytes	
Throughput (Average/Peak)	332.13 kbps/338.90 kbps		332.46kbps/338.54 kbps		329.40 kbps/338.19 kbps	

Same audio quality as Bluetooth Classic chips!

Sony SBH20

	1m		5m		10m	
	Ack'd/Sent	PER (%)	Ack'd/Sent	PER (%)	Ack'd/Sent	PER (%)
POLL	8797/8873	0.86	8564/8654	1.04	8563/8701	1.59
DM1	17/17	0.00	17/18	5.56	17/17	0.00
DH3	14701/15110	2.71	14701/15328	4.09	14701/15282	3.80
Total	23515/24000	2.02	23282/24000	2.99	23281/24000	3.00
Total Payload	2491546 Bytes		2491546 Bytes		2491546 Bytes	
Throughput (Average/Peak)	332.21 kbps/336.13 kbps		332.21 kbps/338.54 kbps		332.21 kbps/336.22 kbps	

Evaluation: Computation, Latency, Codec

- Low computation requirement
  - Integer operations
  - ~1.6% of packet Tx interval
- Comparable end-to-end latency
  - Dominated by audio codec
  - Within <10% of COTS BT dongle
- Support high-quality audio codec
  - AAC, aptX
  - BBC acts as a general transport layer

Computation Time (%)

	POLL	DM1	DH3	Encryption
Mean (μs)	1.19	2.32	10.03	29.67
Std. Deviation (μs)	0.65	1.03	2.70	7.62

End-to-end Latency

	SBH20	CX150	Airpods
BBC (ms)	161.3	229.9	181.8
CSR dongle (ms)	149.3	216.0	165.3

BBC with high-quality audio codec

	CX150	Airpods	Momentum 2
Codec	AAC	AAC	aptX
PER (%)	6.9	25.2	6.6
Throughput (kbps)	307.40	304.51	352.01

Conclusion

- Enable **BLE** chips to communicate with **Bluetooth Classic devices**
- **Emulate Bluetooth Classic operations using SW**
- Establish connections and maintain two-way communication **without using Bluetooth-Classic-specific HW blocks**
- **Add audio connectivity** to existing (and fast-growing) BLE devices
- Enable future devices to use **simple BLE chips for existing headphones**

BLE

Bluetooth Classic