

ECEN 5823-001

Internet of Things Embedded Firmware

Lecture #10
27 September 2018

Agenda

- Class Announcements
- I2C Load Power Management Rubric
- BLE Client assignment
- Bluetooth Low Energy / Smart

Class Announcements

- Quiz #5 is due at 11:59 on Sunday, September 30th, 2018
- Homework #3: HTP BLE Assignment is due on Sunday, September 30th, at 11:59pm
- Mid-term will be held in class on Thursday, October 18th
- Software Best Practices do not include “for loops” for delays. Why?
 - Are optimized out by the compiler optimizer when used
 - Non-transportable code – dependent on frequency, compiler, optimizer, processor architecture
- Starting with the BLE assignment, for/while delay loops will have a minus 1 point deduction

I2C Load Power Management Rubric

1. Total points for this exercise is 10 points
 - a. 2.0 pts for the questions
 - b. 8.0 pts of the code
2. Question scoring. Max score is 2.0 pts.
 - a. Question 1: Will not score due to multiple ways to implement
 - b. Question 2: $< 3\mu\text{A}$ (1.0 pts)
 - c. Question 3: Not scoring due to different ways of measuring
 - d. Question 4: 87-90ms (1.0 pts)
3. Functional code delivered per exercise. Max score is 5.0 pts.
 - a. Measured Period of measurements = 2.0s (1.0 pts)
 - b. Length of POR of Si7021 = 80mS (1.0 pts)
 - c. Total time once LPM On to going Off ~ 87-90mS (1.0 pts)
 - d. Length of conversion time = 7-10mS (1.0 pts)
 - e. Function temperature output being read via I2C (1.0 pts)
 - f. Current will Load Power Management Off $< 3\mu\text{A}$ (1.0 pts)
 - g. Verify LPM being turned ON and OFF (1.0 pts)
 - h. Program is written to use a scheduler to service interrupts (1.0 pts)
4. Best Practices
 - a. Lack of Silicon Labs IP statement for sleep routines (-1.0 pts)
 - b. Not following course documentation practices (-1.0 pts)



BLE Client Assignment

ECEN 5823

BLE Client Assignment

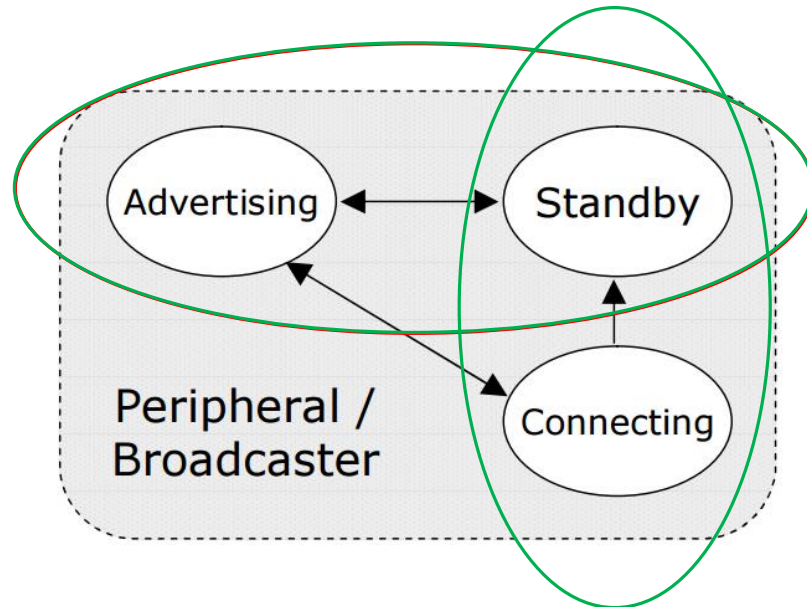
Fall 2018

Objective: To create BLE client that connects and gets the temperature readings from the already developed BLE Server application.

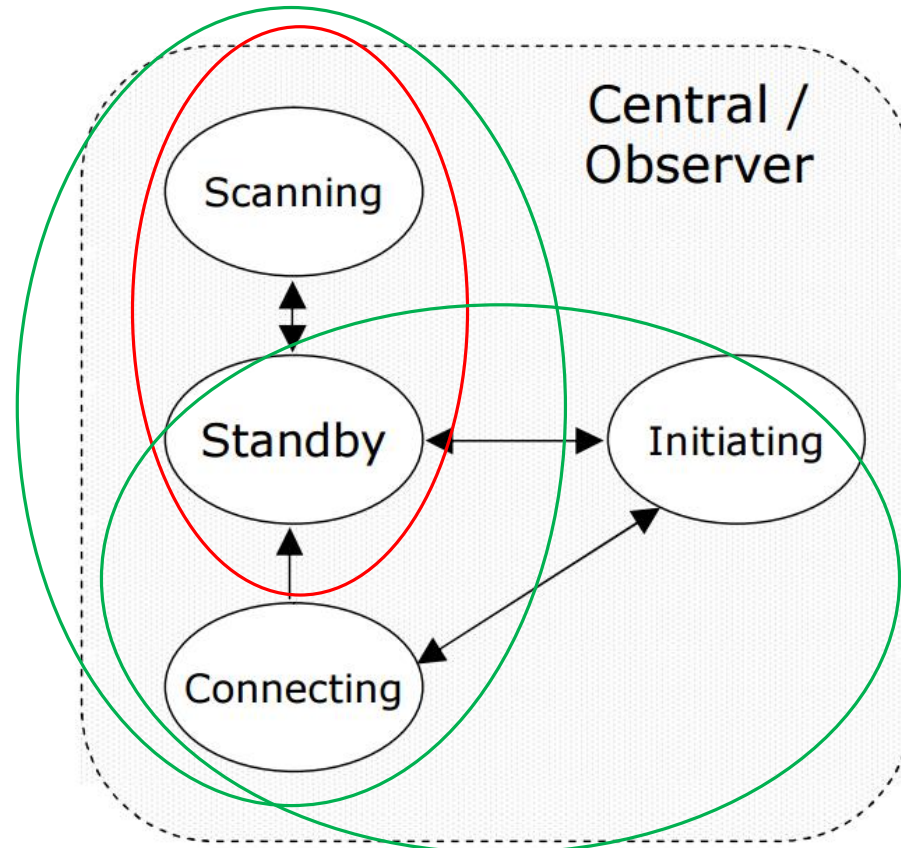
Note: This assignment will begin with the completed BLE Server assignment.

The BLE client project can be made by copying the BLE server project you created by removing unnecessary events in the while(1) loop or can use the soc-empty template project you get from the Simplicity Studio.

BLE: Advertising (Peripheral and Central States)



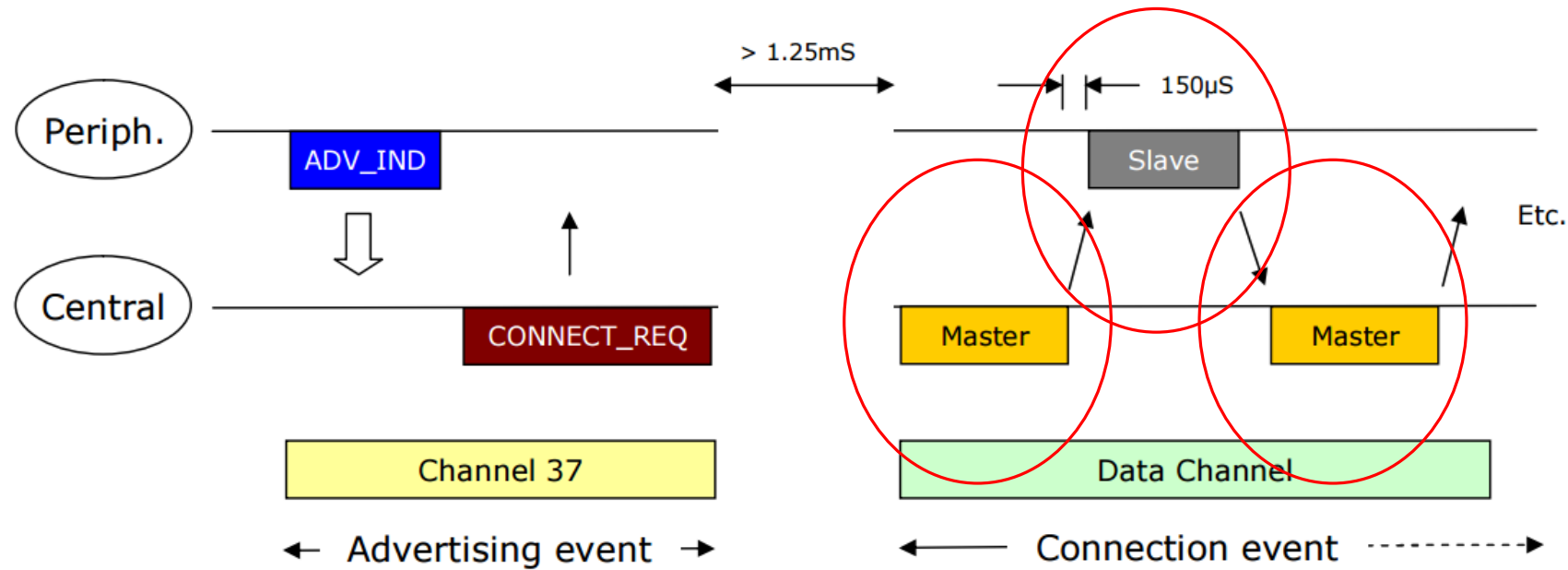
A Broadcaster cannot enter the Connecting state.



An Observer cannot enter the Initiating State.

BLE: Data transactions

In looking at the data connection event, how is it different than a Bluetooth Classic communications?

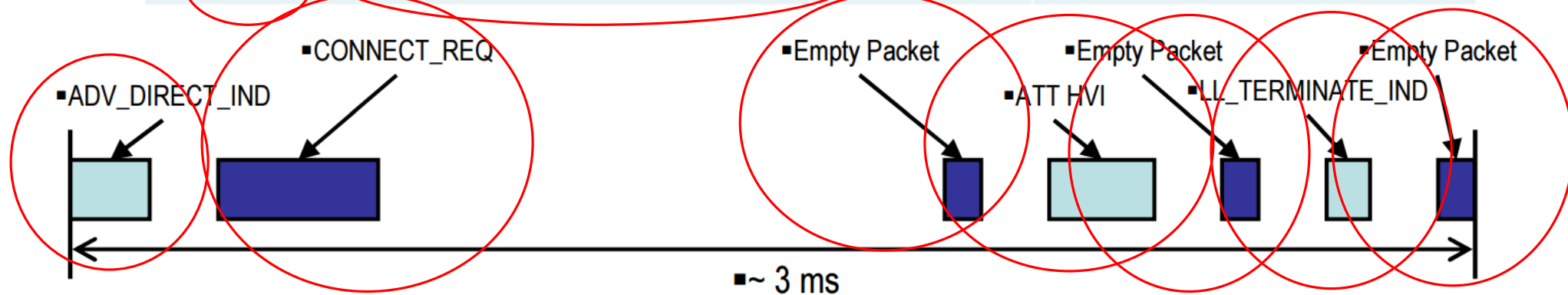


- Once a connection is made:
 - Master informs slave of hopping sequence and when to wake
 - All subsequent transactions are performed in the 37 data channels
 - Transactions can be encrypted
 - Both devices can go into deep sleep between transactions

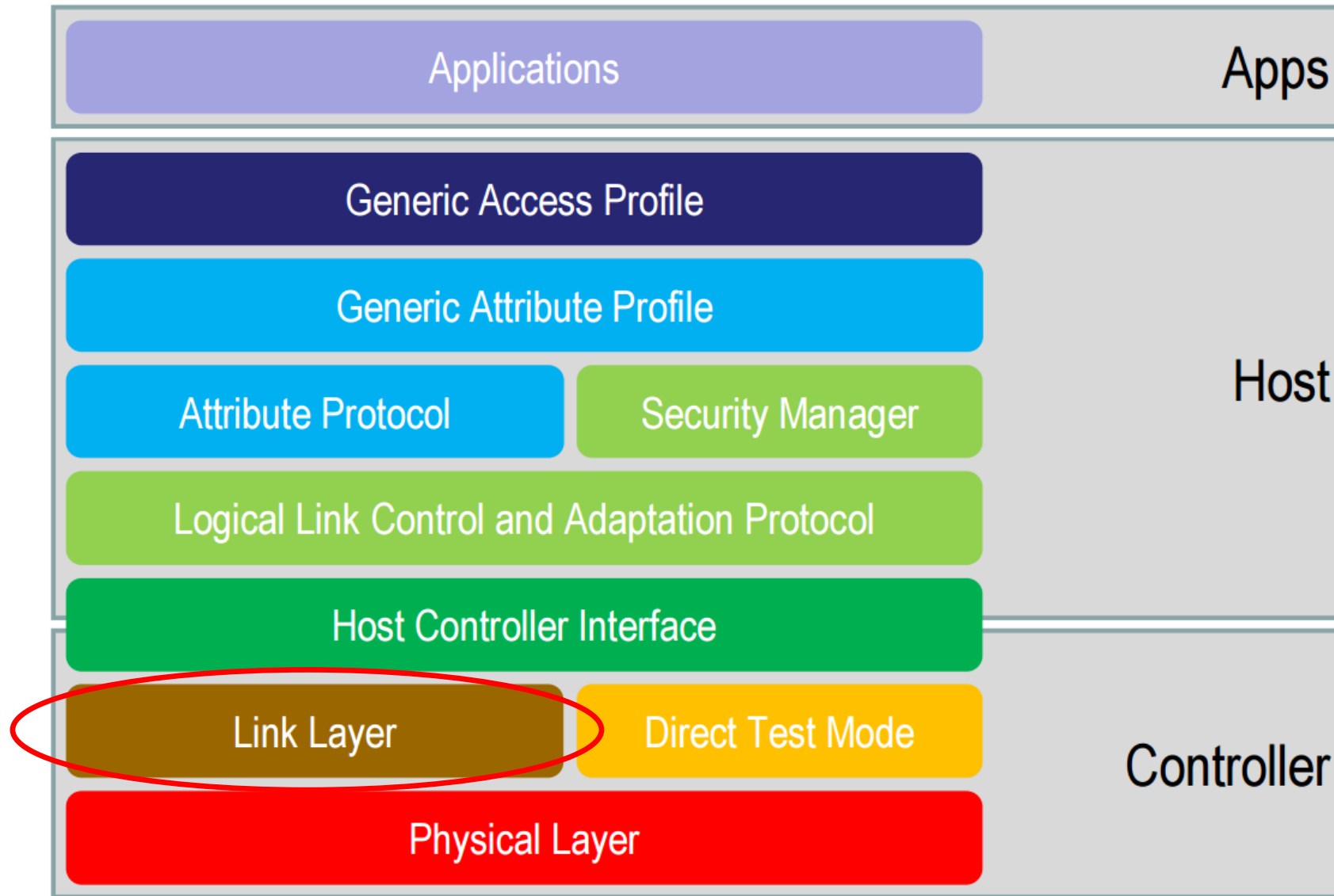


BLE – Minimum time for data transaction

Time (us)	Master Tx	Radio Active (us)	Slave Tx
0		176	ADV_DIRECT_IND
326	CONNECT_REQ	352	
1928	Empty Packet	80	
2158		144	Attribute Protocol Handle Value Indication
2452	Empty Packet (Acknowledgement)	80	
2682		96	LL_TERMINATE_IND
2928	Empty Packet (Acknowledgement)	80	



BLE: Stack



BLE: Optimizing for Low Power/Energy

- What are the primary ways that BLE reduces energy?
 - Keeping the packets short
 - Using a high physical bit rate
 - Providing low overhead
 - Optimized acknowledgement scheme
 - Single-channel connection events
 - Using offline encryption (encrypting when the radio is off)
- What are two types of power consumption that are critical for lower power consumption?
 - Low peak-power consumption to optimize the use of button-cell batteries ($P=I^2R$)
 - Low power-per-application bit to enable a device to be used a long time sending a defined quantity of application data

BLE: Short Packets

- Low power designer dilemma:
 - To make a radio more stable, more circuitry is required that increases cost and power consumption. How does BLE solve this dilemma?
 - The BLE radio solves this dilemma by making the packet length significantly small that heating effects are minimized
 - The heating effect does not require a very long packet to cause this heating problem
 - The 3 millisecond packets in Bluetooth Classic are long enough to cause this heating issue
 - The BLE specifications take into account the physical properties of the semiconductor that are used to create the radios
 - The goal is to keep packets no more than a few hundredths of milliseconds in length to prevent any semiconductor heating problems
 - Resulting in no requirement for calibration or stabilization circuitry for the radio

BLE: Short Packets

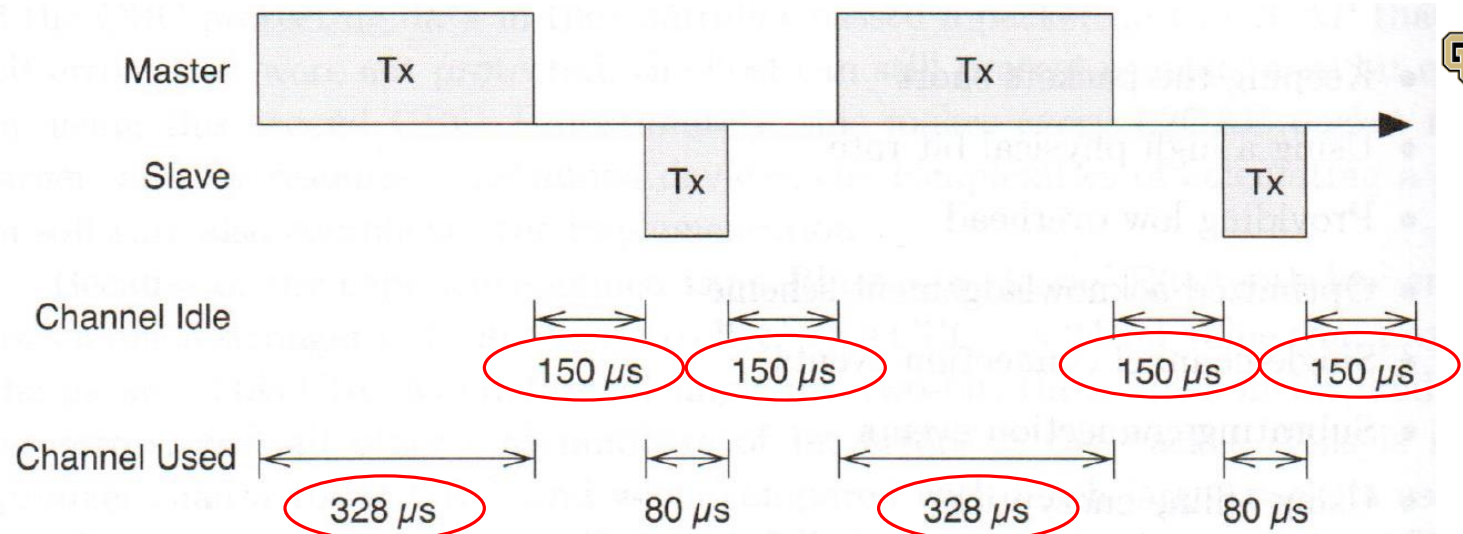
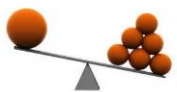


Figure 7-31 Short packets

- The packets are short enough that any drift of the frequency due to heating will **not** be **outside** the BLE radio specifications
 - The longest packet in an Advertising Event is 378uS
 - The longest packet in a Connection Event is 328uS
- To further reduce the issue of silicon heating, the specification requires a 150uS gap between “very long” packets to enable the silicon to cool down between packets
- Removing the requirement of calibrating of the frequencies between transmitting and receiving or receiving and transmitting packets.



BLE: Short Packets

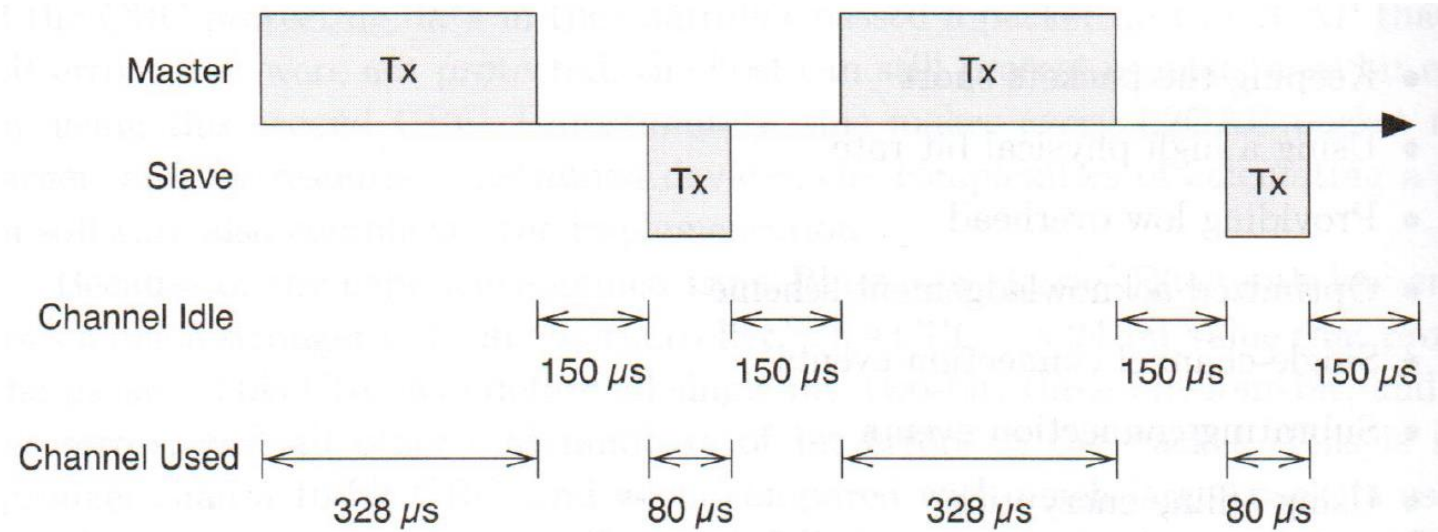


Figure 7-31 Short packets

- The addition of a 150uS cool down periods reduces the maximum duty cycle of transmitting data in one direction on an encrypted link
 - $\frac{\text{maximum size packet} + \text{acknowledge packet}}{\text{total time to send and acknowledge data}}$
 - $\frac{(328+80)}{(328+150+80+150)} = \frac{408}{708} = \sim 58\%$
- 58% is a very low duty cycle for wireless technology
 - For example, Bluetooth classic is 72% (Increase packet length and time has more effect on the numerator than the denominator)

Reducing bandwidth

BLE: High Bit Rate

- CMOS technology is optimized for gates that do not change state since in digital systems, most gates do not change state
 - Running a 2.4GHz oscillator used for the radio modulation is contrary to what CMOS is optimized, thus consuming a significant amount of energy
 - Since basic technology of the CMOS semiconductor defines a base amount of current/energy for the radio, the efficiency of the modulated signal becomes significant.
 - The quicker that a given amount of data can be transmit, the more efficient the radio
 - The BLE radio is designed for 1Msps transmit rates
 - If BLE could only transmit 250Ksps, the radio power would be 4x
 - Note, modulations schemes that are more complex and transmit data at higher data rates can result in more power/energy due to the complexity of the radio

BLE: Low Overhead

- As discussed, the shorter the packet, the more energy efficient the radio
 - Reducing the time that the radio is enabled and the time generating the 2.4GHz oscillator
 - BLE overhead includes:
 - Preamble
 - Access address
 - Header
 - Length
 - CRC
 - And optional MIC value

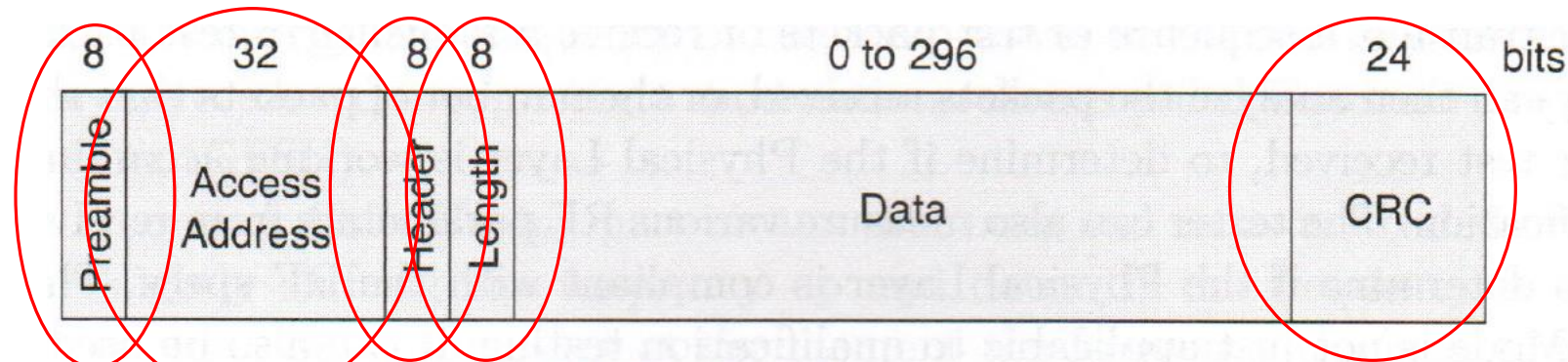


Figure 3–2 The Link Layer packet structure

BLE: Overhead

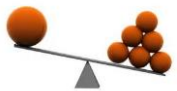
- What is one way of measuring efficiency of a radio protocol?
 - For an unencrypted packet, size of the application data compared with the total packet size required to transmit the application data
 - For an encrypted packet, the efficiency is lower, primarily because of the additional 4 octets of MIC included in each packet
- Compared to other low energy radio technologies, the BLE efficiency is very good
 - For example, ZigBee has a packet overhead of 15 to 31 octets compared to BLE's 10 for unencrypted
 - With ZigBee transmitting 4x slower physical data rates than BLE, a short 4 octet application data in ZigBee could require up to **10 times** more energy to transmit than BLE

Table 7-6 The Overhead for Application Data

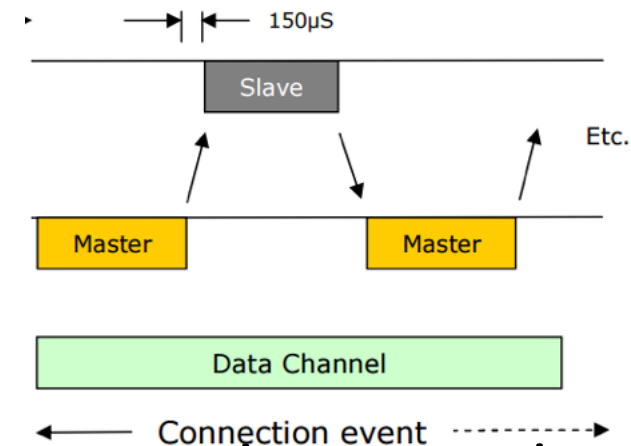
Packet Type	Application Data Size (octets)	Overhead (octets)	Efficiency (%)
Unencrypted	4	10	29%
Unencrypted	8	10	44%
Unencrypted	16	10	62%
Unencrypted	27	10	73%
Encrypted	4	14	22%
Encrypted	8	14	36%
Encrypted	16	14	53%
Encrypted	27	14	66%

BLE: Acknowledgement Scheme

- The Link Layer acknowledgement scheme does not require an acknowledgement of a packet to be performed or even delivered immediately
 - This non-requirement of an acknowledgement is a departure from Bluetooth Classic
 - In Bluetooth Classic, the receiver must acknowledge the packet at the next opportunity it has to transmit
 - If the acknowledgement is not received immediately, the receiver must signal a negative acknowledgement in the next transmit packet
 - In BLE, every packet sent can acknowledge the last packet transmitted, even if this was transmitted some time ago
 - The BLE scheme enables the transmit acknowledgement to occur when convenient such as when it is ready to transmit for some other reason or allow it to finish transmitting a large amount of data quickly



BLE: Single-Channel Connection Events



- All communication between a master and a slave occur in connection events
 - A connection event is a packet transmitted by the master, followed by the slave, followed by a series of alternating packets send by the slave and master
 - During a connection event, the master and slave stay on the same frequency
 - The assumption is that if the master packet was successfully transmitted, the interference is low enough to enable a good channel for the slave response

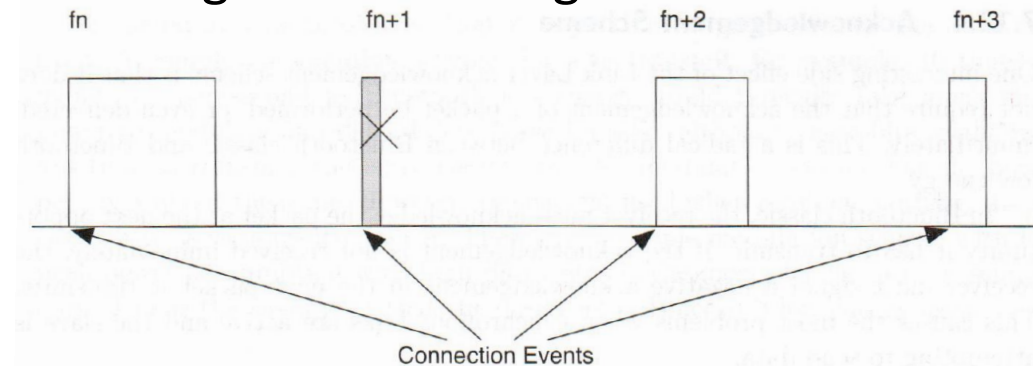


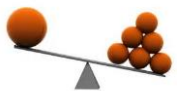
Figure 7-32 Single-channel connection events

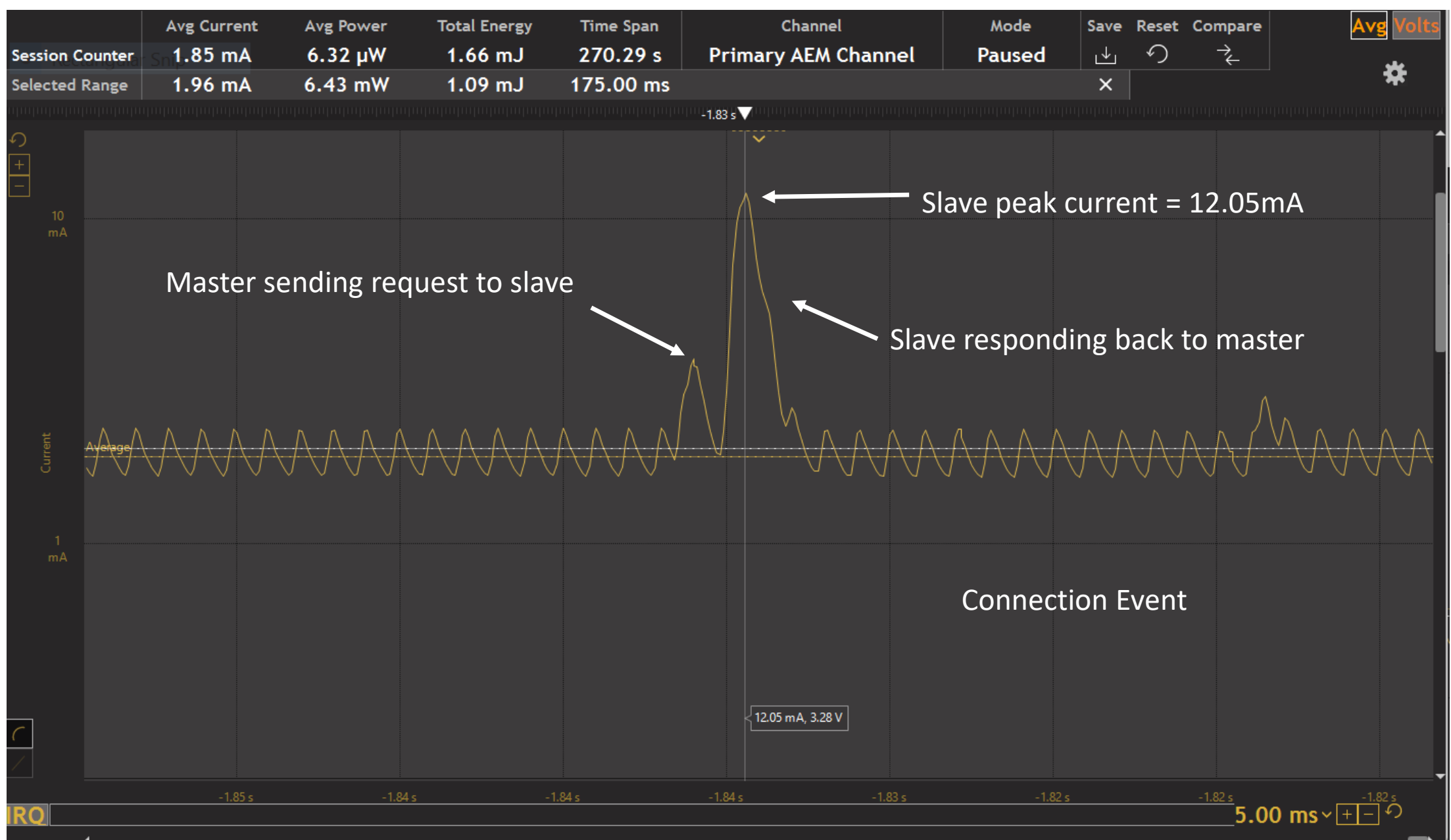
BLE: Single Channel Connection Events

- If the assumption to use Single Channel Connection Events is that the channel is clear for the slave if open for the master, why not use this frequency channel all the time?
 - The master cannot signal this change to all the slaves
 - Consumes a lot of power to resynchronize the single frequency once the channel begins to fail
 - With interference having a “bursty” nature, such as a WiFi packet occurring, staying on one channel continually begins to fail
 - The one-channel model also reduces the number of co-located networks because each will naturally drift to a clean frequency, and once filled, no more co-located networks would be able to be established without interfering on an existing network
- Frequency-hopping algorithm distributes network traffic in both time and frequency allowing for many more simultaneous networks to be active

BLE: Subranging Connection Events

- Low latency
 - Low latency means frequent contact between the slave and the master
 - Frequent contact means high use of the radio
 - High use of radio results in higher energy use and lower battery life
- Low power/energy
 - Low power means the slave is listening to the master infrequently
 - If the master is continually polling the slave and the slave has to listen to each of the polling connection events, it would not be low power
- How to balance between low latency and low power/energy?
 - Allow the slave to ignore most connection events from the master





BLE: Subrating Connection Events

- The feature of allowing the slave to ignore X number of connection events is called **slave latency**.
- The more connection events that a slave can miss, the lower power the slave can be.
- The limit to slave latency is that it cannot be longer than the supervision timeout of the connection.

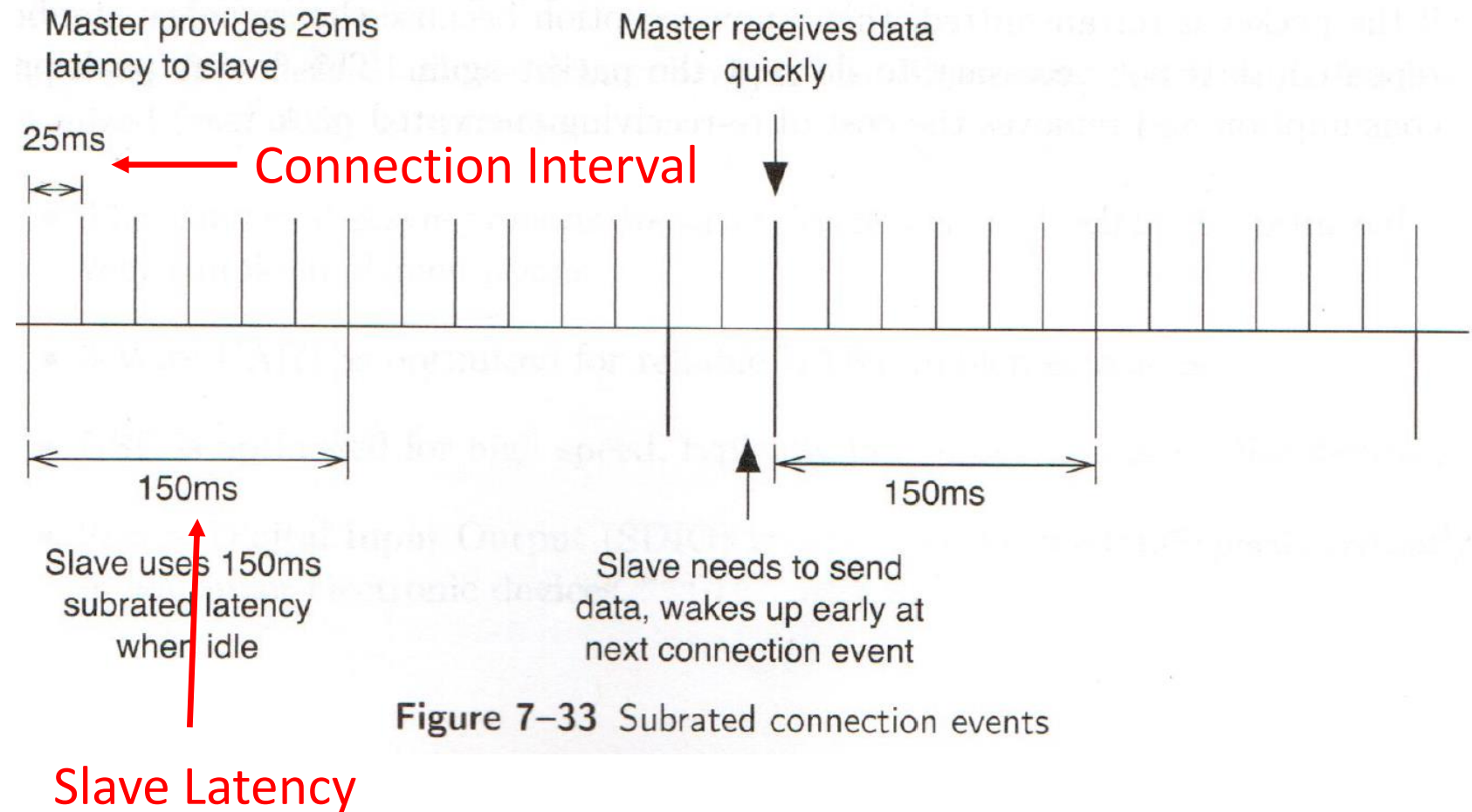


Figure 7-33 Subrated connection events

BLE: Subranging Connection Events

- It is not recommended to have a slave latency that gives fewer than 6 opportunities for the slave to resynchronize before the supervision timeout
- For example, if the supervision timeout is 600ms and the connection interval is 25ms, the slave latency should not be longer than 450ms
- Resulting in the slave to transmit data in an average of 25ms, connection interval, but only needing to connect 1 out of 24 connection events

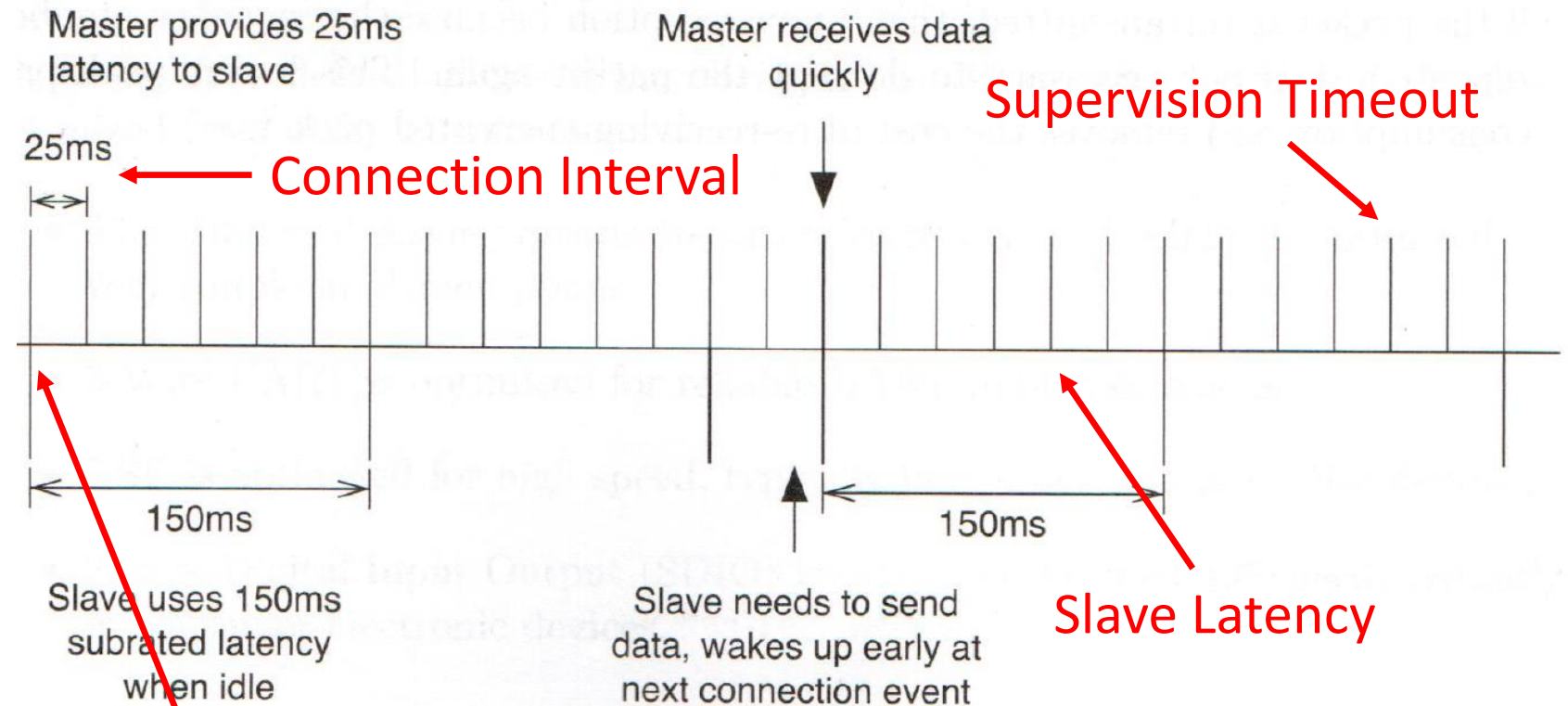
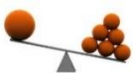


Figure 7-33 Subranged connection events

End of last master-slave connection event



Optimizing to the slave or less resource rich device

2.8.1.4 cmd_le_connection_set_parameters

This command can be used to request a change in the connection parameters of a Bluetooth connection.

Table 2.137. Command

Byte	Type	Name	Description
0	0x20	hlen	Message type: Command
1	0x09	lolen	Minimum payload length
2	0x08	class	Message class: Connection management
3	0x00	method	Message ID
4	uint8	connection	Connection Handle
5-6	uint16	min_interval	Minimum value for the connection event interval. This must be set be less than or equal to max_interval. <ul style="list-style-type: none"> • Time = Value x 1.25 ms • Range: 0x0006 to 0x0c80 • Time Range: 7.5 ms to 4 s
7-8	uint16	max_interval	Maximum value for the connection event interval. This must be set greater than or equal to min_interval. <ul style="list-style-type: none"> • Time = Value x 1.25 ms • Range: 0x0006 to 0x0c80 • Time Range: 7.5 ms to 4 s
9-10	uint16	latency	Slave latency. This parameter defines how many connection intervals the slave can skip if it has no data to send <ul style="list-style-type: none"> • Range: 0x0000 to 0x01f4 Use 0x0000 for default value
11-12	uint16	timeout	Supervision timeout. The supervision timeout defines for how long the connection is maintained despite the devices being unable to communicate at the currently configured connection intervals. <ul style="list-style-type: none"> • Range: 0x000a to 0x0c80 • Time = Value x 10 ms • Time Range: 100 ms to 32 s • The value in milliseconds must be larger than $(1 + \text{latency}) * \text{max_interval} * 2$, where max_interval is given in milliseconds It is recommended that the supervision timeout is set at a value which allows communication attempts over at least a few connection intervals.

Blue Gecko BLE API

Foscam Incident

- August 2013

Gilbert says he first heard a voice from down the hall. As he and his wife got closer, what it was saying got worse.

"He said, 'Wake up Allyson, you little (expletive),'"
Gilbert said.

And soon he knew it was coming from the camera.

"I see the camera move on us," Gilbert said.

Gilbert immediately pulled the plug and started doing research. He believes someone hacked his router as well as the camera. The person could see Allyson's name on the bedroom wall to call her by it.




Foscam Incident

- Researchers had discovered that an attacker can use the following IP address of the baby monitor to download the entire memory contents of the baby monitor
 - `http://[IP Address]/proc/kcore`
- Once access to the memory content, the hacker can use a hex editor to obtain:
 - Username
 - Password
- Question is, how did the hacker locate a specific baby monitor that is exposed to the internet?

← → ↺ <https://www.shodan.io> ☆ 🗨️ 🌐 ☰

Apps ★ Bookmarks IRU Google Drive Identity Theft Protection CU D2L


Shodan Developers Book View All...

 SHODAN 🔍 Explore Enterprise Access Contact Us New to Shodan? Login or Register

The search engine for the Internet of Things

Shodan is the world's first search engine for Internet-connected devices.

Create a Free Account Getting Started



Explore the Internet of Things

Use Shodan to discover which of your devices are connected to the Internet, where they are located and who is using them.



See the Big Picture

Websites are just one part of the Internet. There are power plants, Smart TVs, refrigerators and much more that can be found with Shodan!



Monitor Network Security

Keep track of all the computers on your network that are directly accessible from the Internet. Shodan lets you understand your digital footprint.



Get a Competitive Advantage

Who is using your product? Where are they located? Use Shodan to perform empirical market intelligence.



Foscam Incident

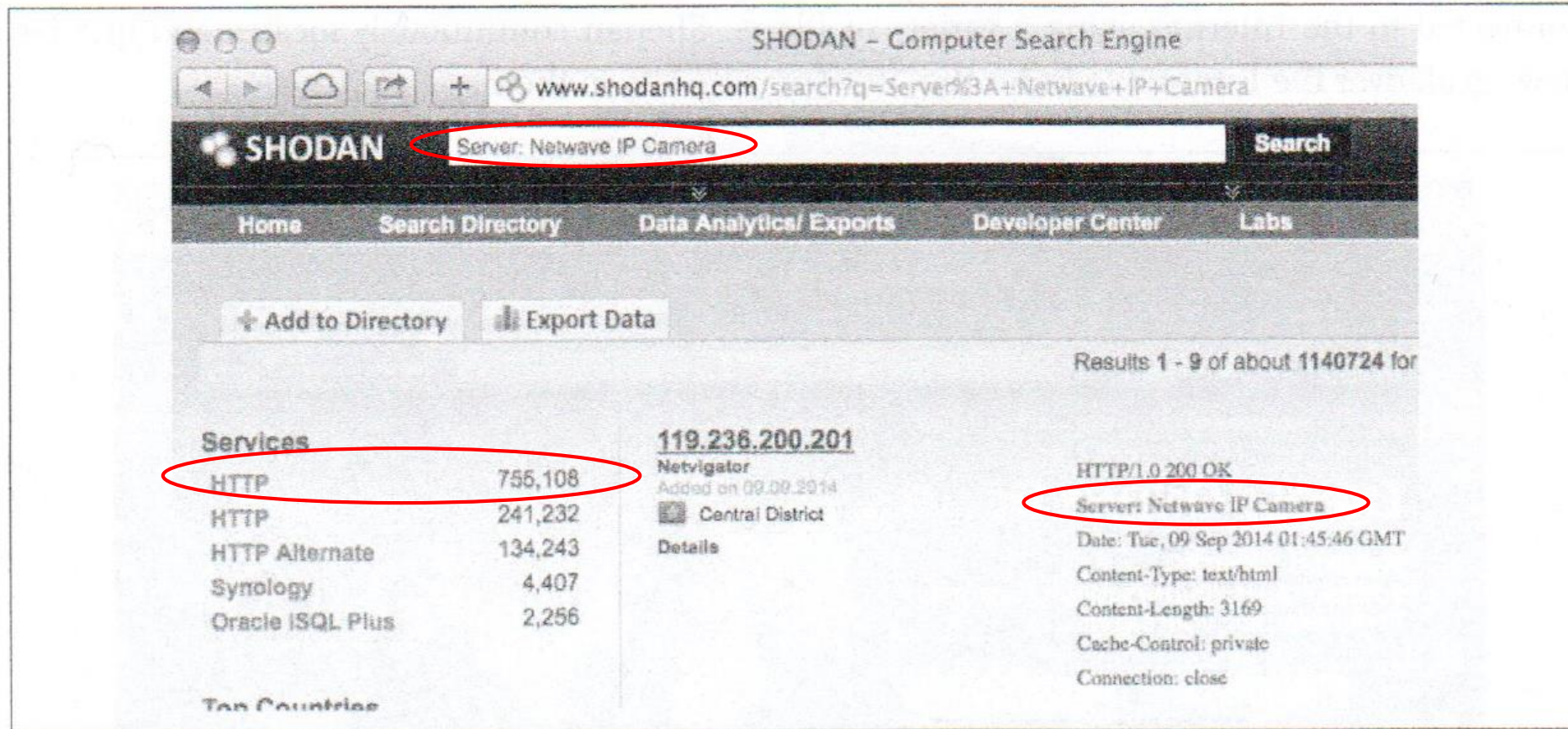


FIGURE 3-3. Shodan query to locate Foscam devices on the Internet

Foscam – Exploiting Default Credentials

- Foscam devices were originally known to have default username of “admin” and a blank password
- Most users are likely to use the default username and password unless the setup requires the user to select a stronger username or password
- In August 2013, Foscam released an upgrade to the firmware and prompted the users to change the user name and blank passwords, but the upgrade was not automatic
- Users had to locate the software update manually and then apply it using a web interface
- Researchers have concluded that 0 Foscam cameras in the “wild” run the latest firmware

Foscam saga continues

- In April 2014, another incident similar to the Gilbert's in August 2013 occurred

CINCINNATI, OH (FOX19) -Heather Schreck was asleep around midnight in her Hebron home when a voice startled her.

"All of a sudden, I heard what sounded like a man's voice but I was asleep so I wasn't sure," Heather said.

Disoriented and confused, Heather picked up her cell phone to check the camera in her 10-month-old daughter Emma's room. The camera was moving, but she wasn't moving it.




"About the time I saw it moving, I also heard a voice again start screaming at my daughter. He was screaming, 'Wake up baby. Wake up baby.' Then just screaming at her trying to wake her up."

That's when Heather's husband, Adam, ran into Emma's room. Adam said the camera then turned from his petrified daughter to point directly at him.

Foscam saga continues

- The April 2014 Foscam incident exemplifies how security vulnerabilities in IoT devices can persist if device manufacturers do not implement a seamless method to push security patches to existing devices
- Research proves a manual procedure required to update a device pretty much guarantees most people are unlikely to do so
 - Few people are likely to make the effort to find and apply security patches

How to implement DFU via Bluetooth OTA?

- Why firmware updates?
 - To improve SECURITY!
 - Provide feature upgrades
 - Fix product bugs / issues
 - Update to the latest Bluetooth Stack
- What is updated during an DFU OTA?
 -  • Bootloader
 -  • Bluetooth Stack
 -  • Application