Understanding Reliability in Bluetooth® Technology

Author: Martin Woolley

Version: 1.0.2

Revision Date: 2 December 2020

Bluetooth®

Revision History

Version	Date	Author	Changes
1.0	9 October 2020	Martin Woolley	Initial Version
1.0.1	4 November 2020	Martin Woolley	Fixed several incorrect section reference numbers
1.0.2	2 December 2020	Martin Woolley	Corrected the SN and NESN values in Figure 13

Table of Contents

Table of Contents

Table of Contents

1.0 Reliable Communication and Bluetooth® Technology

1.1 The Joy of Radio

Wireless communication systems usually employ radio as the underlying, physical basis for getting data from one device to another. Bluetooth itself is a radio communications technology. But there's a problem.

Radio is Undeniably, Unambiguously, Uncontestably, Unreliable. And that's a fact.

So logically, if Bluetooth technology uses radio and radio is unreliable, how can Bluetooth technology ever be described as reliable?

Let's start by defining the problem and then move on to answer that key question. We'll focus primarily on Bluetooth Low Energy (LE) in this paper.

Radio is Undeniably, Unambiguously, Uncontestably, Unreliable. And that's a fact.

2.0 What Do We Mean by Reliability?

2.1 A Simple Definition

A plain English definition of reliability which is suited to our purposes follows:

Communication can be regarded as reliable if the data sent is the data received and any intended action relating to that data happens as expected.

We intuitively know what we mean when we talk about reliability. It's not rocket science.

Communication can be regarded as reliable if the data sent is the data received and any intended action relating to that data happens as expected.

2.2 Extending our Understanding of Reliability

It may not be rocket science, but sometimes requirements for reliability in a product or solution are a little more nuanced. The basic requirements that data should arrive in the correct state and that actions should occur as expected may be supplemented or refined with other requirements such as:

- **A tolerance for failure:** if the expected result is observed in 99.9999% of cases, we may still regard the operation of the system as reliable. Sometimes this type of reliability requirement is expressed with respect to a time period such as a tolerance for no more than 1 failed operation every 24 hours and we may even talk about *mean time between failures*.
- **Latency:** the system may be regarded as operating reliably only if the lights always switch on no more than 500ms after the light switch was activated.
- **Resilience:** operations are still carried out correctly even when certain system, product or component failures have occurred. If one of the specified failures occurs and interrupts the intended service or functionality, then the system is said to be unreliable. If the system can experience such failures and continue to deliver its intended service or function, it is said to be reliable.

Reliability needs to be *fit for purpose* and is not an absolute. Like security, this is a concept best understood in the context of a set of requirements.

And talking of security, it should be noted that changes to data that take place somewhere between the transmitter and receiver could be caused by naturally occurring phenomena or be deliberately brought about, with malicious intent.

In some scenarios, there will be a great deal of tolerance for some transmitted data not being received. If the heart rate monitoring application I use on my smartphone while cycling does not receive a certain proportion of the values that are transmitted by the

heart rate monitor strapped around my chest, that is probably still OK. The overall functionality of the system would be unaffected and I would almost certainly be unaware that this was happening. The reliability of the system in this context is fit for purpose. On the other hand, there may be situations which demand a much higher level of reliability, with very little tolerance for failures. The same heart rate monitor in a medical rather than sports and fitness context is likely to need to hit far more stringent reliability targets.

3.0 Reliability Issues in Wireless Communication Systems

There are many types of issue and circumstances that can cause reliability problems in wireless communications systems.

3.1 Collisions

Imagine a system that uses patterns of either red or blue light to convey information. Receivers in this system only understand the meaning of red or blue light and the behaviour of this simple and fictitious signalling system is undefined if a different colour of light is encountered.

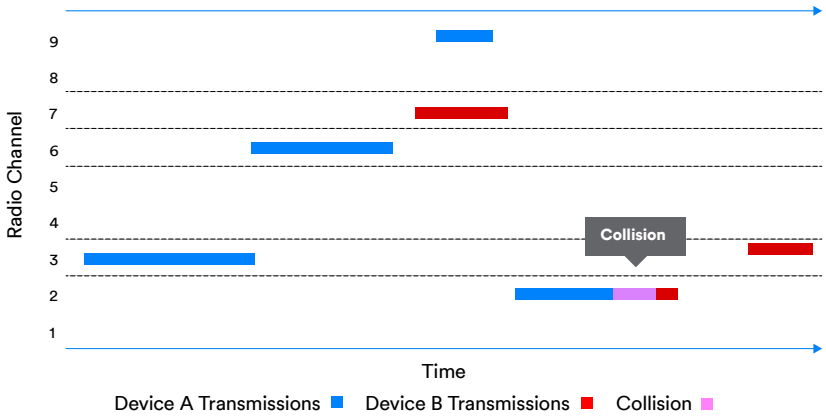

Figure 1 - A collision taking place on channel #2

Now imagine we have a room, equipped with a blue light and a red light, each controlled independently by separate switches on the wall. There's a light sensor in the middle of the room, which registers the colour of light and responds as it has been programmed to, depending on whether red or blue light is sensed.

The two switches are turned on and off at random intervals and for a random duration by a couple of poorly trained but otherwise excellent technicians. Every now and again, the red light and blue light are both on at the same time. For the period where the two lights overlap, the sensor measures the light colour to be purple. This colour is not recognised and so its receipt is treated as an error. To use the language of telecommunications, a collision has occurred and the information transmitted by both lights has been corrupted and lost.

Radio transmissions can collide in the same way if two in-range transmissions take place on the same channel and over time periods which overlap. As was the case in the coloured light scenario, information within the two radio transmissions is corrupted and lost.

When a device transmits some data, the digital bits that the data is comprised of

are converted into analogue symbols and they are transmitted, one at a time on the selected radio channel. The symbols are transmitted at a symbol rate, which is a measure of how quickly you can change from one symbol to another, and therefore a given number of bits will take a certain elapsed time to be transmitted. The longer the transmission takes place, depending on the symbol rate and the number of bits to be transmitted, the greater is the probability that a collision will occur.

If two devices repeatedly transmit at the same time on the same channel, using the same time interval and duration parameters, persistent collisions will occur.

3.2 Multi-path Propagation and Time-Dispersion

Radio signals like light, can reflect off surfaces or be refracted as they pass through objects. In communication systems, this can result in a signal arriving at a receiver multiple times, from different directions. This is known as multipath propagation.

A consequence of multipath propagation is that copies of a signal that have taken a different path may arrive at the receiver at slightly different times, a phenomenon known as *time dispersion*. Time dispersion can cause a type of interference which is known as *Inter-Symbol Interference (ISI)*.

3.3 Transmitter / Receiver Synchronisation

The type of radio used in Bluetooth® devices is called a half-duplex radio. This means that two devices can communicate with each other in each direction, but not simultaneously. First one radio transmits while the other radio receives, and then the other radio transmits while the first radio receives. A radio of this type can be in one of three states at any one point in time; either transmitting on a given radio channel, listening to a particular channel or idle.

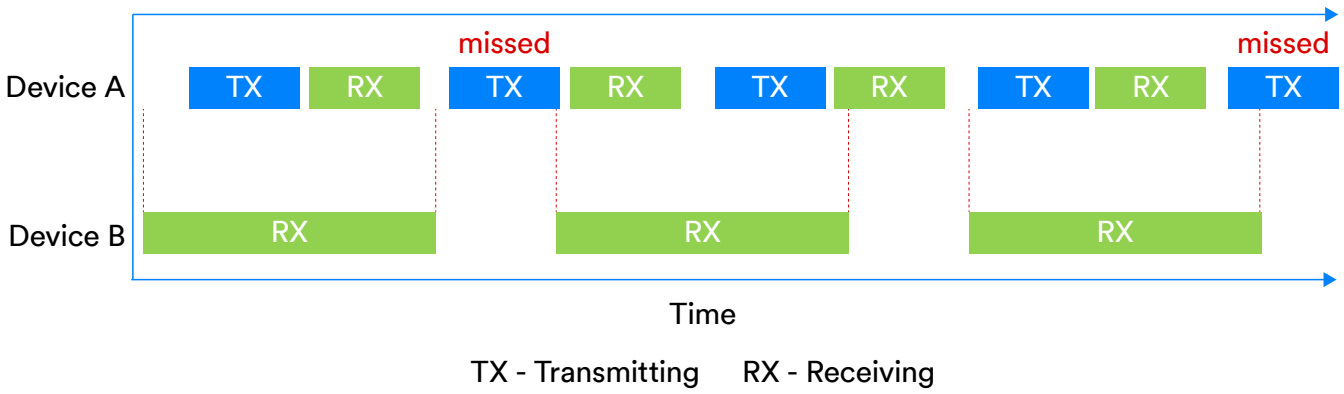

Figure 2 - Missed packets in unsynchronised communication

If a receiver device is not listening when another in-range device transmits some data or it is not listening on the channel that the transmitter is using then the transmitted data will not be received.

The percentage of time that a device spends listening for transmissions is called the *RX duty cycle*. The higher the RX duty cycle, the more likely it is that the receiver

will be listening when the transmitter sends data. This will be especially important when two devices have not synchronised their transmit/receive timings.

Figure 2 depicts a situation involving two devices. Device A is transmitting data periodically. Device B is *scanning* (listening) for transmissions periodically but the scheduling of the periods when it will be scanning is in no way synchronised with the timing of transmissions from Device A. Consequently, some of the time Device B is scanning for the whole time that Device A is transmitting a packet and so is able to receive it in full. At other times it is not and so the packet transmitted by Device A is lost.

Note that Device B is scanning for approximately three quarters of the time. Its RX duty cycle is therefore approximately 75%.

3.4 Signal Strength and Receiver Sensitivity

Signal strength, measured by a receiving device is called the Received Signal Strength Indicator (RSSI). Signal strength can have an impact on reliability in a number of ways.

A strong transmitted signal can saturate a radio receiver and errors can result when attempting to decode the received signal.

The weaker a signal is, the closer its level gets to the level of any background *noise*. Noise in this context is defined as unwanted radio signals caused by naturally occurring and human-made electromagnetic radiation. The relationship between the signal strength and background noise levels is called the *signal to noise ratio*. When the signal to noise ratio is reducing, it eventually becomes difficult to decode the information contained within the transmitted signal without error. The rate at which attempts to decode received analogue symbols to produce the corresponding digital bits fails is called the *Bit Error Rate (BER)*. When the BER is sufficiently high, communication fails completely.

The further from the transmitter a receiver is located, the lower the RSSI will be and so the potential for errors will be higher, due to a lower signal to noise ratio. The reduction of the signal strength as the distance from the transmitter increases is known as *path loss*.

The Bluetooth Core Specification states that a receiver must exhibit a BER of no more than 0.1% at a signal strength of -70 dBm. This is known as the *receiver sensitivity*.

Two devices, A and B may have different transmission power levels. This can lead to situations where Device A is comfortably within range of Device B and can therefore transmit data to it reliably, but B is approaching its sensitivity limit and therefore the signal to noise ratio experienced by A is low and causing errors to be experienced. We sometimes refer to this situation as involving *asymmetric radio links*.

3.5 Modulation Schemes

A modulation scheme is the means by which information is encoded in a radio signal for transmission. Modulation schemes exploit one or more of the fundamental properties of

radio for this purpose. Some schemes use amplitude, some use frequency and some use phase, for example.

Modulation schemes do not cause reliability problems as such, but some perform better than others, increasing the probability that the receiver will be able to correctly decode a signal and extract the information that it contains.

3.6 Coexistence and Collocation

Different radio technologies may use the same part of the radio spectrum. Bluetooth® technology and Wi-Fi[1] both use the Industrial, Scientific and Medical (ISM) 2.4GHz band, for example.

When two or more radio technologies share a part of the radio spectrum, we have what is known as a *coexistence* issue. One technology may interfere with the other if suitable mitigation steps are not taken.

When two or more radio technologies are supported by the same device, they are said to be collocated. *Collocated* radios may interfere with each other without measures being taken to minimise or eliminate this issue.

3.7 Buffer Overflow

A buffer is a temporary store in a computer or microcontroller's memory. Available memory is always limited and each protocol layer may require one or more buffers. Each buffer has a maximum size.

When one device sends data to another, it sends the data in a series of discreet packets. Those packets will be received by other devices and are typically placed temporarily in a buffer. Packets are then passed up through the various layers of the protocol stack, possibly resting temporarily in other buffers along the way.

If the rate of arrival of packets into a buffer exceeds the rate at which packets are removed from it then eventually the buffer will overflow, with some packets being discarded. When this happens, the data contained within these packets is lost and communication may be regarded as having failed.

3.8 Single Points of Failure

Communication systems are just that. *Systems*.

Systems by definition, consist of multiple inter-related components and in some cases, a component may be key to the overall, reliable operation of the system. Failure of that one key component can therefore cause the whole system to fail. A component with this property is known as a *single point of failure*.

Technical architects and network designers will seek to avoid the existence of single points of failure in their designs.

[1] *When using the 802.11b, 802.11g or 802.11n MAC/PHY*

3.9 Software Defects

Perfectly designed systems will exhibit problems if not implemented correctly and software can sometimes contain defects or *bugs*. It is one of the primary purposes of testing to find defects such that they can be corrected before an application or product is released. But sometimes, even with the most rigorous of testing regimes, defects can go undetected and this may give rise to reliability problems in applications or products. Users may then incorrectly conclude that the communications technology used is itself at fault.

3.10 The 100% Reliability Mirage

100% reliability is unlikely to be achievable in any real-world system, whether it uses wired or wireless communication. A system can fail in a great many ways, some relating to communication of data and some not. Anybody involved in the subject of *disaster recovery* will appreciate this.

Where requirements for reliability are high and the potential consequences of failures are critical and unacceptable, it is common to build fail-safe mechanisms into systems so that even if the probability of failure is very, very low, the system will fail in a safe way on those very rare and unusual occasions where it does.

[1] *When using the 802.11b, 802.11g or 802.11n MAC/PHY*

4.0 Creating Reliability from Unreliable Foundations

Bluetooth® technology uses radio and radio is unreliable. But Bluetooth communication works very well, so how is this apparent contradiction to be explained?

The answer lies in numerous aspects of the design of the Bluetooth communication system, including its use of radio and its *protocols*.

Bluetooth technology is a modular system and more than one configuration of stack is possible.

Smartphones and connectable peripheral devices will include a Bluetooth Low Energy (LE) controller with a host component that supports the Generic Access Profile (GAP) and Generic Attribute Profile (GATT) and protocols such as the Attribute Protocol (ATT) and Security Manager Protocol (SMP). Figure 3a shows this stack configuration.

A Bluetooth mesh device will also contain a Bluetooth LE controller but the host part will contain the layers of the Bluetooth mesh networking stack. Figure 3b depicts a Bluetooth mesh stack.

Whatever the stack configuration, each layer has clearly defined responsibilities and a means of passing data to the adjacent layers above and below. Features of Bluetooth technology which mitigate or reduce the probability of certain types of potential reliability problems exist in various parts of the stack. Some such mechanisms apply to all possible uses of Bluetooth technology, while some of them apply to only particular scenarios.

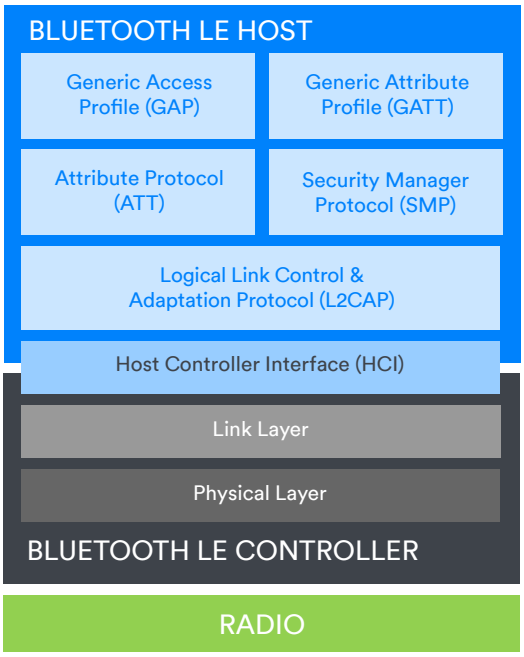

Figure 3a - A stack configuration supporting Bluetooth LE with GAP/GATT/ATT

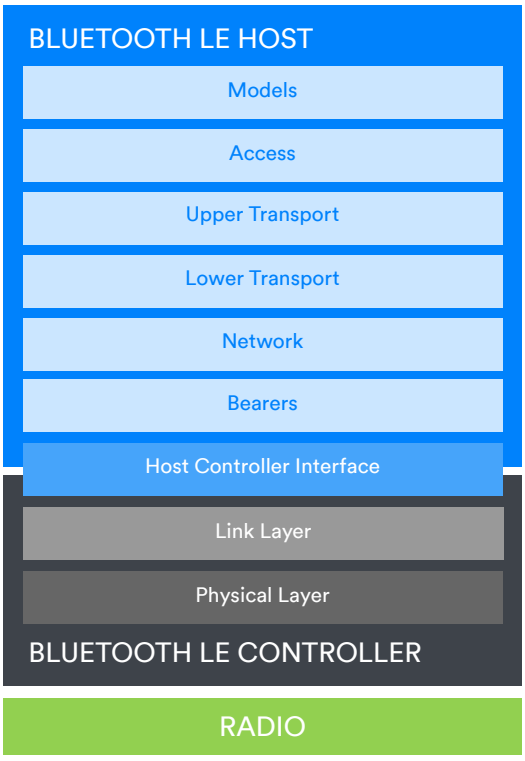

Figure 3b - A stack configuration supporting Bluetooth mesh

4.1 Generally Applicable Features and Mitigation Techniques

We'll start with a guided tour of the reliability enhancing features of Bluetooth technology that are generally applicable in all scenarios. Figure 4 shows an example Bluetooth *air interface packet* and will be referred to.

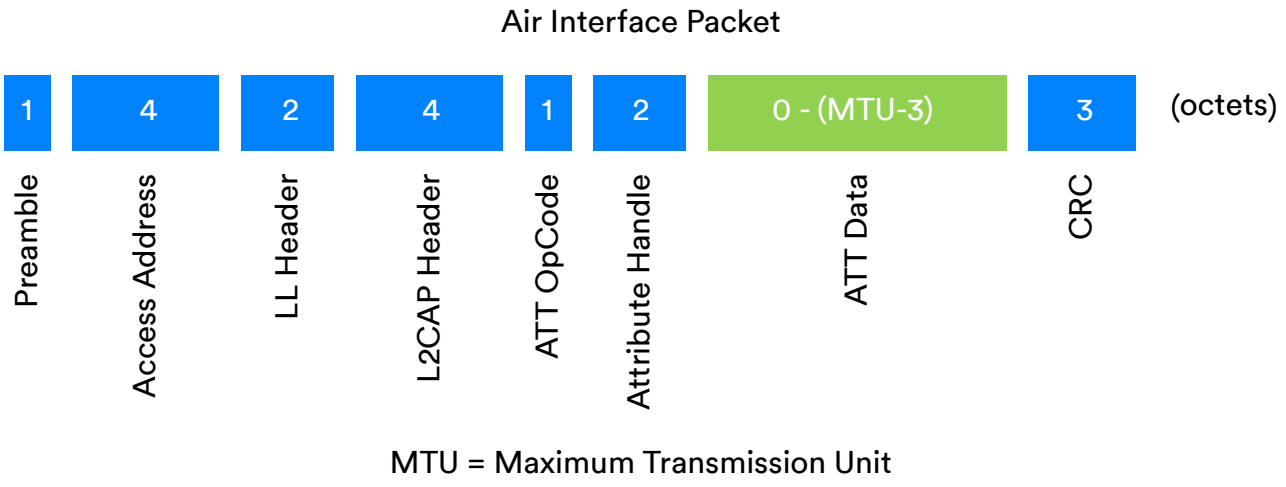

Figure 4 - An example Bluetooth packet containing an ATT PDU

4.1.1 The Bluetooth® Modulation Scheme

Reliability in Bluetooth technology starts with the most fundamental of issues, concerning exactly how radio is used as a carrier for digital data. In the Bluetooth stack, these issues are dealt with in the physical (PHY) layer.

One of the primary problems that the physical layer must deal with, is to be able to recognize Bluetooth radio transmissions and extract the data encoded in a signal correctly. This is an absolutely fundamental step on the road to reliability.

Radio is an analogue, physical phenomena. Physicists typically model radio signals in terms of waves. Radio waves possess electromagnetic energy and have a collection of fundamental properties including an amplitude, wavelength, and frequency. These concepts are illustrated in Figures 5 and 6.

As defined previously, strategies which use the fundamental properties of waves in some way to encode information are called *modulation schemes*. There are many modulation schemes. Some use the changing amplitude of a signal; some encode information using radio phase and some use frequency changes.

When reliability is an important design goal for a radio communication system, some modulation schemes are better than others. Amplitude-based modulation schemes are somewhat susceptible to interference due to noise, whereas frequency-based schemes are less vulnerable in this respect.

Bluetooth technology uses a special *binary frequency shift keying* modulation scheme called G*aussian Frequency Shift Keying (GFSK).* It's a *binary* modulation scheme because each symbol represents only one bit, with a value of either zero or one.

Binary frequency shift keying encodes digital data by selecting a central frequency known as the *carrier* and then shifting it up by a given *frequency deviation* to represent a 1 or down by the same frequency deviation to represent a 0. The minimum size of the

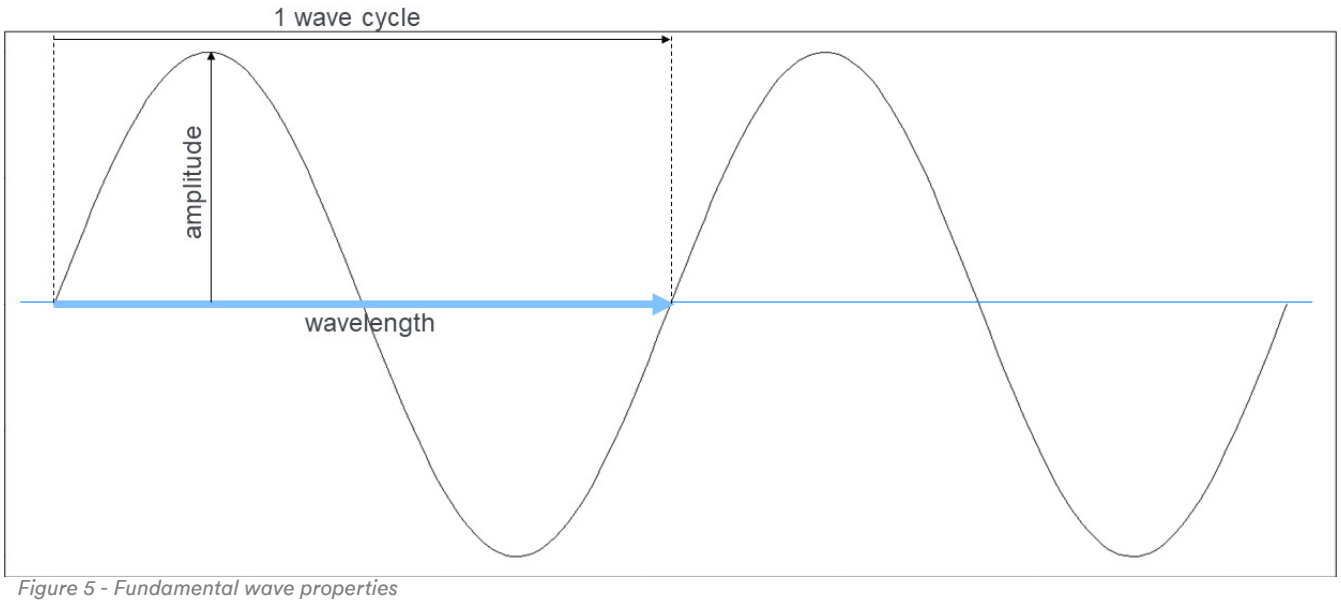

Figure 5 - Fundamental wave properties

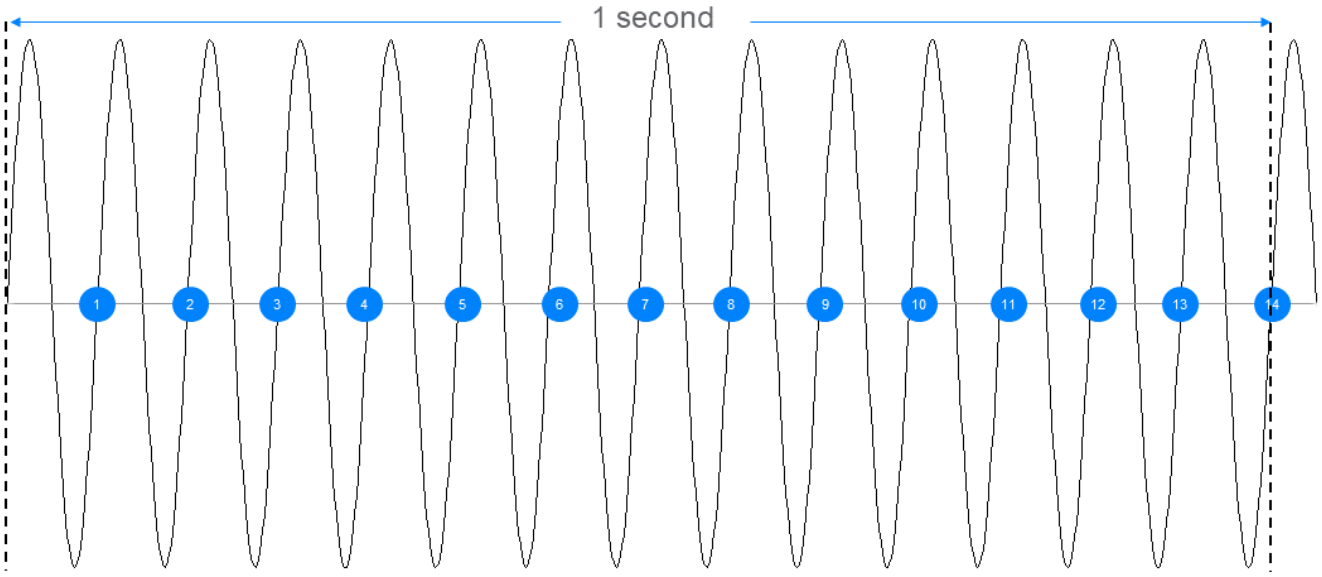

Figure 6 - Frequency

frequency deviation that is permitted is specified in the Bluetooth® Core Specification and depends on the selected symbol rate, which is either 1 or 2 mega-symbols per second (Msym/s) in Bluetooth LE. For the 1 Msym/s symbol rate, a minimum frequency deviation of 185 kHz is specified, whereas for the faster symbol rate, the minimum frequency deviation is 370 kHz. These values were chosen carefully, to help make the recognition of encoded 1s and 0s in a signal reliable.

Frequency shift keying (FSK) modulation schemes, by definition involve a frequency change each time the symbol value changes. An abrupt, near-instantaneous change of frequency will generate noise and noise causes interference. Furthermore, in real circuits, there is the possibility of *spectral leakage* where the signal spills over unintentionally, into

other frequencies, making the task of decoding it at the receiver all the more difficult.

Bluetooth® technology reduces interference through its use of a superior FSK modulation scheme, GFSK. The Gaussian aspect of GFSK modifies the standard FSK approach by including a filter which causes frequency transitions to be smoothed and therefore less noisy and the spectral width to be narrower, reducing the potential for interference with other frequencies.

4.1.2 Preamble

The first field in all Bluetooth LE packets is called the *preamble*. It is 8 bits long and contains an alternating pattern of binary ones and zeroes. Its purpose is to provide the receiver with material it can use to find the frequencies being used to encode digital ones and zeroes in the remainder of the packet. It is also used by the radio's automatic gain control which optimises the signal strength.

Accurately establishing the frequencies used in a signal and setting the radio's parameters to an optimal state is the first step in ensuring reliable receipt of a packet.

4.1.3 Access Address

When the Bluetooth controller is listening for data on a channel, it will receive **all** radio signals within the frequency range defined by that channel. Received signals may be:

- Bluetooth packets sent to this device.
- Bluetooth packets which are not intended for this device.
- Packets relating to other wireless communications technologies which are operating in the same ISM band and using frequencies in the Bluetooth radio channel currently being scanned.
- Background noise.

The Bluetooth controller must be able to distinguish between these signals and accurately pick out those that encode Bluetooth packets sent to this device. Anything else must be ignored.

All Bluetooth packets contain a 32-bit *access address* which allows signals that are almost certainly Bluetooth to be quickly picked out at the earliest opportunity, and other signals to be immediately discarded.

There are two types of access address.

The *advertising access address* is a fixed value of 0×8E89BED6 which most advertising packets use. This value was chosen because it has good *correlating properties*. Correlation is the mathematical procedure used to recognise specific patterns in a signal.

Packets exchanged during communication between two connected devices contain an *access address* with a value assigned by the link layer which uniquely identifies all packets relating to that connection. These generated access address values are largely random but

subject to additional rules which are designed to increase the reliability of recognising access addresses correctly.

Packets relating to distinct *periodic advertising chains* and to distinct B*roadcast Isochronous Streams* (BIS) each have a unique access address.

The access address allows signals which are *relevant* to the receiving device to be selected. It is a responsibility of the Bluetooth® stack's *Link Layer* to check access addresses.

The probability of mistaking random background electro-magnetic noise for a Bluetooth signal is extremely small, thanks to the 32-bit length of the access address. In the unlikely event that the pattern of random background noise matches an access address which is relevant to the receiver, further bit stream processing will quickly determine that it is not a valid Bluetooth packet.

Quickly selecting only relevant signals and discarding others, is another key step in Bluetooth receiver operation which contributes to reliable communication.

4.1.4 The Cyclic Redundancy Check (CRC)

All Bluetooth packets contain a *Cyclic Redundancy Check (CRC)* field which appears at or near to the end of the packet. CRCs are a commonly used mechanism for detecting cases where transmitted data has been unintentionally changed due to issues like collisions.

When a new packet is formulated by the link layer, a CRC value is calculated by applying the CRC algorithm to the other bits in the packet. The resultant 24-bit value is then added to the packet.

On receiving a packet, the link layer in the receiving device recalculates the CRC and compares the result with the CRC value included in the received packet. If the two values are not the same, it is concluded that one or more bits in the transmitted packet have been changed and the packet is discarded.

It should be noted that the CRC is not a security mechanism since a packet could be deliberately altered and the CRC easily recalculated.

4.1.5 The Message Integrity Code (MIC)

Bluetooth LE packets may be encrypted. All encrypted packets include a field called the *Message Integrity Check (MIC)*. The MIC is in fact a *message authentication* code but since the acronym MAC has other uses in the field of communications, in the Bluetooth specification, *MIC* is used.

The MIC is not a reliability feature per se. It is a security feature whose purpose is to enable the detection of attempts to deliberately tamper with the contents of a packet. But since part of our informal definition of reliability is that the data transmitted should be the data received and we acknowledge that changes may be unintentional or deliberate, we include it here for completeness.

After all, can insecure communication ever really be thought of as being reliable?

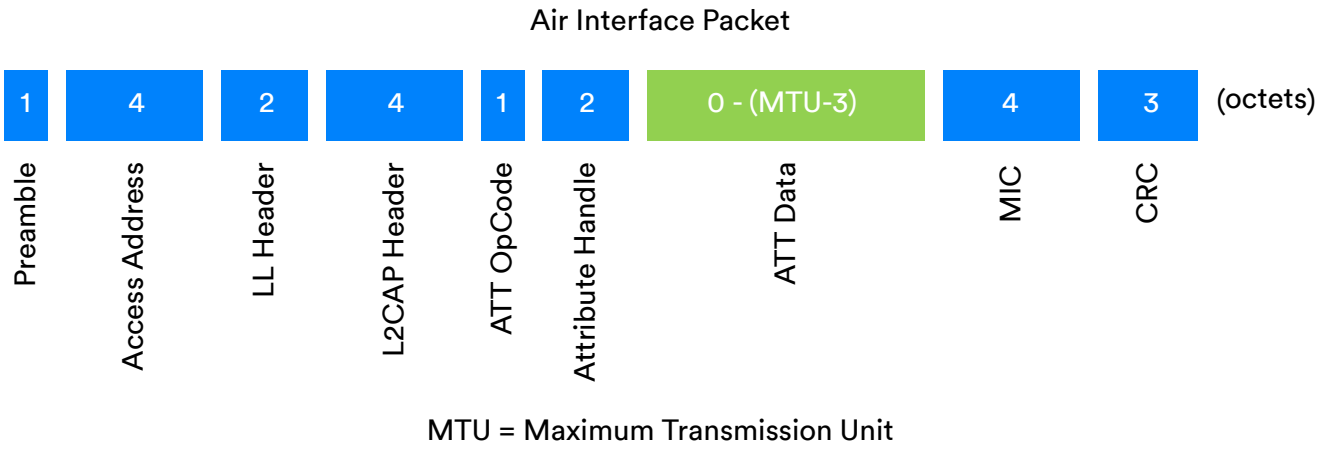

Figure 7 - An encrypted Bluetooth LE packet with the MIC field

4.1.6 Spread Spectrum

Bluetooth® technology uses the 2.4GHz ISM radio band. 2.4 GHz ISM does not define a single frequency, but rather it defines a range of frequencies, in this case starting at 2400 MHz and ending at 2483.5 MHz. When used with Bluetooth LE, this frequency range is divided into 40 channels, each 2 MHz wide. Bluetooth BR/EDR divides it into 80 channels of 1 MHz width.

Each channel is numbered, starting at channel zero. Channel zero has a centre frequency of 2402 MHz, leaving a gap of 1 MHz between the lowest frequency delimiting channel zero and the start of the ISM 2.4 GHz band. Channel 39 has a centre frequency of 2480 MHz, which leaves a gap of 2.5 MHz to the end of the ISM 2.4 GHz band.

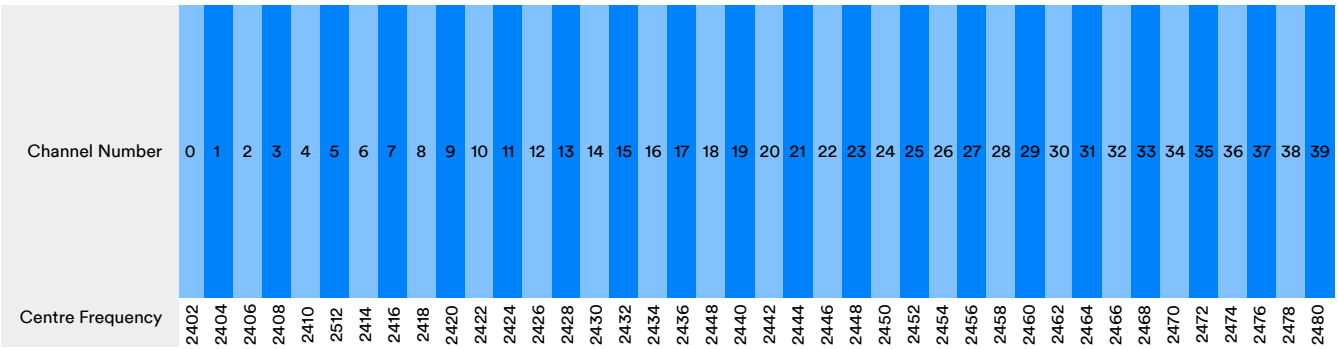

Figure 8 - Bluetooth LE channels within the ISM 2.4 GHz band

Figure 8 depicts the division of the ISM band into radio channels for use by Bluetooth LE. Note that *channel number* always ascends in a contiguous sequence from 0 to 39 whereas a *channel index*, which we will cover in section 4.2.2, is assigned to the set of ISM channels in a slightly different way.

Communication of data over Bluetooth® technology makes use of more than one radio channel. Using multiple radio channels makes Bluetooth communication highly reliable in busy radio environments, where collisions and interference are likely to occur.

The use of multiple frequencies in this way, is called a *spread spectrum* technique and Bluetooth can be said to be a spread spectrum radio communications technology. The details of how spread spectrum techniques are employed vary in a number of different situations and the topic will be re-examined in sections 5.2, 5.3 and 5.4.

4.1.7 Addressing Coexistence and Collocation Issues

The use of the same radio band by a number of different radio technologies at the same time, poses potential challenges. It is possible for one technology to interfere with the transmissions of another technology, notably through the occurrence of *collisions* (see 3.1). Collectively, such issues are known as *coexistence* problems. Bluetooth, Wi-Fi, cordless DECT phones and even microwave ovens, all operate in the 2.4 GHz ISM band and so the potential for coexistence problems between these technologies and device types exists.

Coexistence issues are primarily addressed in Bluetooth through the use of spread spectrum techniques. Even greater reliability is achieved when two devices are *connected*, through the particular way in which spread spectrum techniques are used in Bluetooth in that scenario and this will be explored in section 5.2.

Collocation is the term used to describe the existence of more than one radio *within the same device*, each supporting a different communications technology or set of technologies. There is scope for interference between the different radios in a device. A Long-Term Evolution (LTE) radio, as used in 4G mobile phone systems can operate in frequency bands that are adjacent to the 2.4 GHZ ISM band, which gives rise to potential problems such as preventing one radio from receiving whilst the other is transmitting. Most collocation issues fall outside of the scope of the Bluetooth Core Specification itself, but advice to implementers is provided. Mitigating measures include the use of filters which reduce interference between radios and radio time-slot scheduling considerations which implementers are advised to accommodate.

Radio time-slot scheduling is a complex issue, concerned with determining when the radio is and is not available for use. Some aspects of scheduling fall within the scope of the Bluetooth Core Specification. Issues relating to collocation with other radios and other considerations and constraints, such as those which an operating system might impose, do not. A feature known as *Slot Availability Masks (SAMs)* is defined however, and this allows two Bluetooth devices to provide information to each other about what time-slots are available for use and by taking this information into account, the scheduling used by each device may be optimised to avoid using time slots where collocation-related interference is likely.

4.1.8 The LE Coded PHY

Bluetooth LE offers three different ways of using the radio. The three alternatives

are part of the physical layer and each is referred to with the abbreviation, *PHY*. The three defined PHYs are:

- LE 1M - 1 Msym/s symbol rate
- LE 2M - 2 Msym/s symbol rate
- LE Coded - 1 Msym/s symbol rate with Forward Error Correction (FEC)

The LE Coded PHY increases the receiver sensitivity so that a BER of 0.1% is not encountered until the receiver is at a greater range from the transmitter than would be the case with the LE 1M PHY.

LE Coded is used with a parameter called S set to either 2 or 8. When S=2, LE Coded approximately doubles the range over which communication is reliable. When S=8, range is approximately quadrupled.

Reliable communication at longer range is accomplished by the LE Coded PHY without increasing the transmission power through the inclusion of extra data in each packet which allows errors to be both detected and *corrected* using a mathematical technique called Forward Error Correction. The increased range is accompanied by a resultant reduction in data rate however, with S=2 yielding 500 Kb/s and S=8 delivering 125 Kb/s.

The primary purpose of the LE Coded PHY is to increase range, but it does so by reducing the bit error rate at lower signal strengths so that communication at longer ranges is sufficiently reliable.

4.2 Reliability in Bluetooth® Connection-Oriented Communication

In this section, we'll explore how reliability is achieved when two Bluetooth devices are connected to each other.

4.2.1 Connections

A device may be connected to several other devices simultaneously and use of the radio is divided amongst the connections using a time-sharing strategy. When two Bluetooth LE devices connect, they agree a number of parameters which then govern how they subsequently communicate. Key amongst these parameters is the *connection interval* which controls how often a connection may use the radio.

Every time the connection interval starts for a connection, we say that there has been a *connection event* and the first device (the *Central*) will transmit a packet. The second device (the *Peripheral*) in the connection, working to the same connection interval, will be ready to receive that packet. After a fixed delay of 150 microseconds, the Central then switches to listening and the Peripheral may then transmit. This exactly timed exchange of packets may be repeated a number of times during the connection event, subject to implementation details that fall outside of the Bluetooth Core Specification.

In this way, at precisely timed intervals, each connection is serviced and each pair of devices transmit and receive packets at exactly the right time, perfectly

synchronised with each other. When a packet is transmitted, the target device is listening as required and so is ready to receive the packet.

Figure 9 provides a simplified depiction of how the radio is shared across four connections. Note that

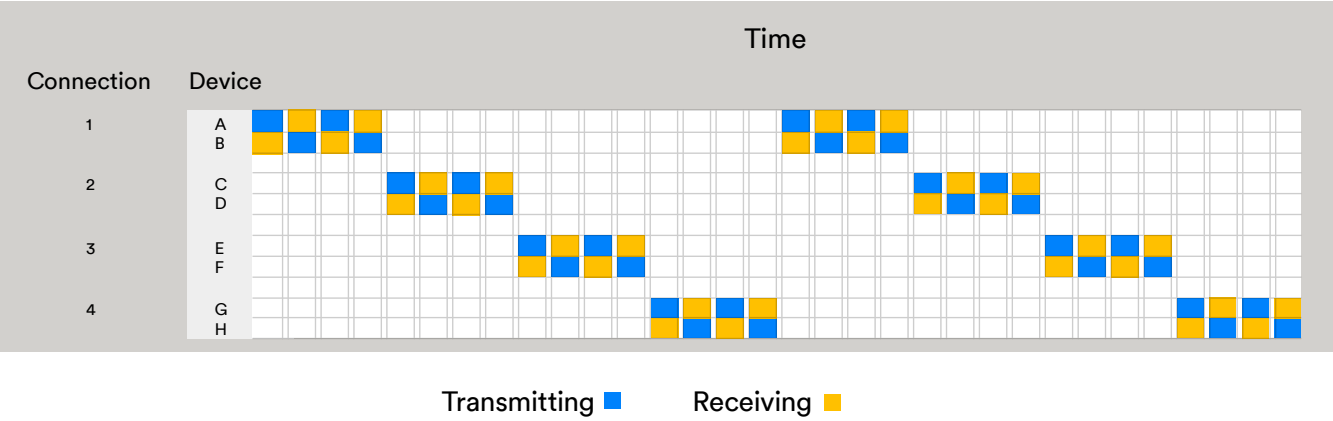

Figure 9 - A simplified illustration of connection use and radio sharing

in this case, all connections have the same connection interval and exchange the same number of packets in each connection event, which would not necessarily be the case.

4.2.2 Adaptive Frequency Hopping

One of the major challenges in radio communications concerns *collisions*, which are particularly problematic in busy radio environments. Section 3.1 explored collisions and explained that a collision occurs when two or more devices transmit data on the same radio channel in overlapping time periods, and that different radio technologies such as Bluetooth® and Wi-Fi can interfere with each other if their use of the radio spectrum overlaps.

Bluetooth technology mitigates the risk of collisions through its use of *spread spectrum* techniques. When two devices are connected, this involves a specific technique known as *adaptive frequency hopping*.

At each *connection event*, as described in 5.2.1, a pair of connected devices have the opportunity to use their radios to exchange packets at precisely timed intervals. But in addition to this, at the start of each connection event, *frequency hopping* occurs, with a radio channel being deterministically selected from the set of available channels using a *channel selection algorithm*. Each device in the connection will then switch to the selected channel and over time and a series of connection events, communication will take place using a frequently changing series of different channels, distributed across the 2.4 GHz band, thereby significantly reducing the probability of collisions occurring.

Of the 40 channels defined for use by Bluetooth LE, 37 of these channels (known as the *general purpose channels*) are available for use during connected communication.

Frequency hopping makes a great contribution to reliability in communication between connected devices but Bluetooth goes one step further.

In a given environment, some Bluetooth® radio channels might not be functioning well, perhaps because interference is impacting them, whereas other channels are working reliably. Over time, the list of reliable channels and unreliable channels may change, as other wireless communication devices in the environment come and go.

The primary device in a connection maintains a *channel map* which classifies each channel that is working well as *used* or otherwise as *unused*. The channel map is shared with the second device using a link layer procedure so that they each have the same information about which channels will be used and which will be avoided.

Devices use implementation-specific techniques to monitor how well each channel is functioning. If it is determined that one or more previously working channels are no longer working well enough, the channel map is updated. Conversely, if a previously bad channel is found to be working well now, its status will also be updated in the channel map. Channel map updates are then shared with the second device. In this way, Bluetooth ensures that it uses only known good channels, avoids problematic channels and keeps the channel map up to date so that it is always the optimal subset of channels that are being used. This is the *adaptive* aspect of the Bluetooth *adaptive frequency hopping system*.

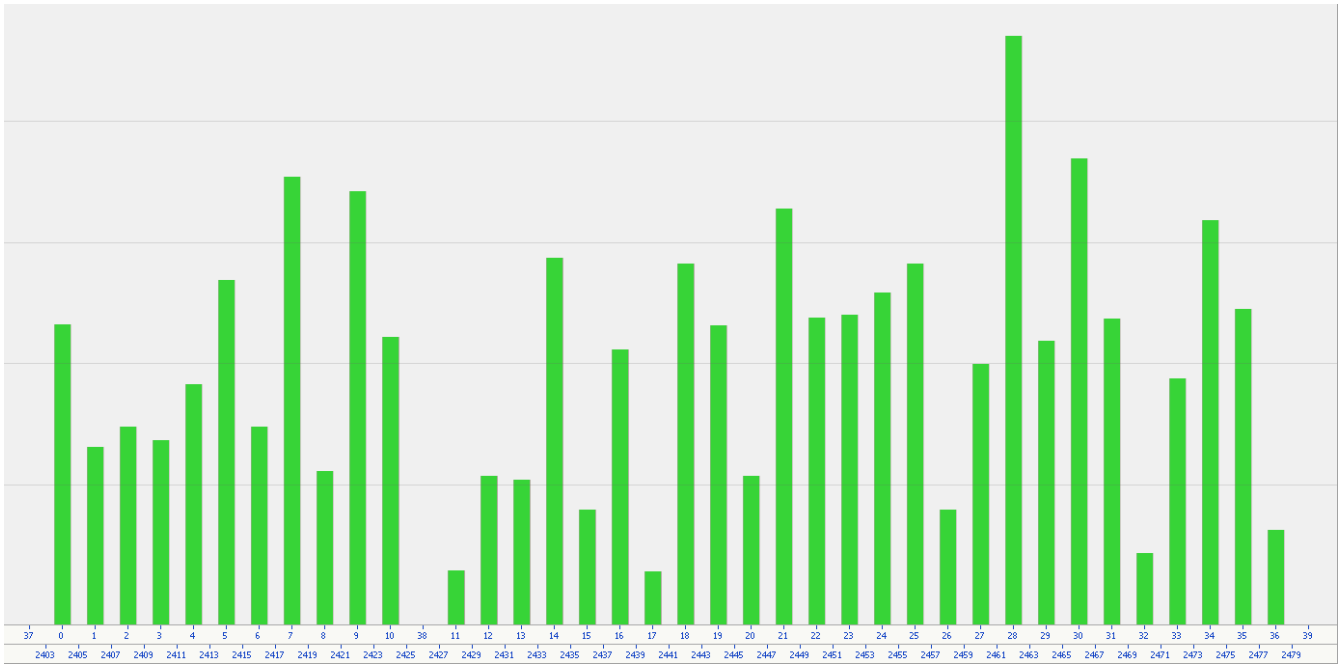

Figure 10 - Adaptive Frequency Hopping distributing communication across channels

Figure 10 shows the way the Bluetooth channels were used by two connected devices during testing and illustrates the highly effective way in which radio use is spread across the ISM 2.4 GHz spectrum. At the bottom of the chart you can see the *channel index* and frequencies in MHz. The channel index is an indirect way of referencing a radio channel and will be discussed further in section 5.3.

4.2.3 Ordering and Acknowledgements at the Link Layer

The Bluetooth® LE link layer uses a type of signalling between connected devices which ensures that data is processed in the right order, that the receipt of packets can be acknowledged, and for this to be used to decide whether to move on to the next packet or instead, to retransmit the previous one.

All link layer data packets contain three important fields which contribute to communication being reliable. These fields are called the Sequence Number (SN), Next Expected Sequence Number (NESN), and the More Data field. All three of these fields are single bit fields and their use provides a system of acknowledgements and a method for checking for the correct ordering of received packets.

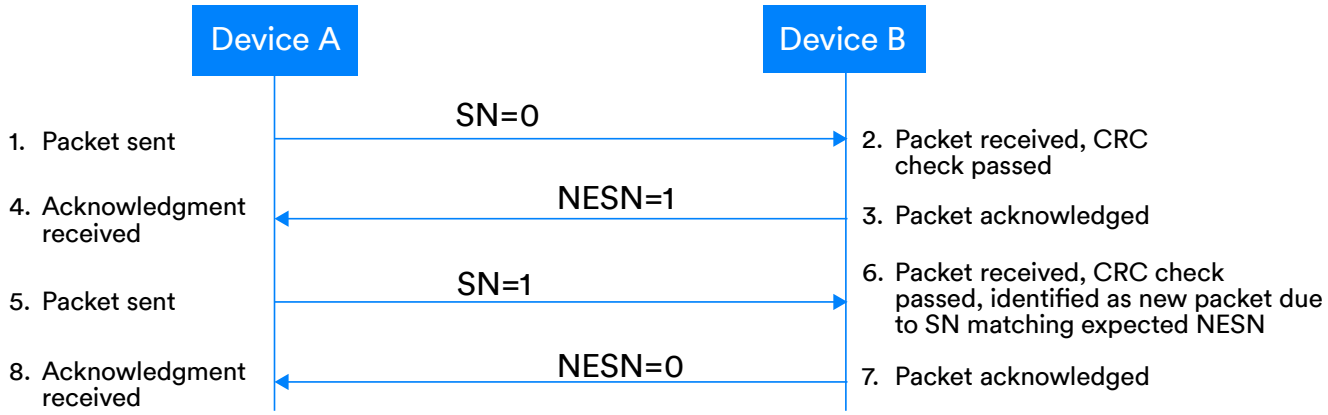

Figure 11 - A successful exchange of packets at the link layer

Communication starts with the primary device (Device A) sending a link layer data packet with SN and NESN both set to zero. From this point on, at each packet exchange that takes place, if all is well, the value of the SN field as set by Device A, will alternate between zero and one. The secondary device (Device B) always knows therefore, what the SN value of the next packet to be received should be and checks for this.

If Device B receives a packet from Device A with the expected SN value, it responds with a link layer data packet that has NESN set to the logical value *NOT(SN)*. So for example, if the received SN value was 1 then NESN in the response will be 0.

When Device A receives a response from Device B with NESN set to the value that Device A intends to use for SN in its next packet, Device A takes this to be an acknowledgement from Device B, confirming that it received the last transmitted packet correctly.

If Device B receives a packet with the wrong SN value, it assumes that the packet is the retransmission of the previous packet received, acknowledges it but does not pass it up the stack for further processing.

If Device A receives an unexpected NESN value in a reply from Device B or does not receive a reply at all, it resends the packet with the same SN value used originally.

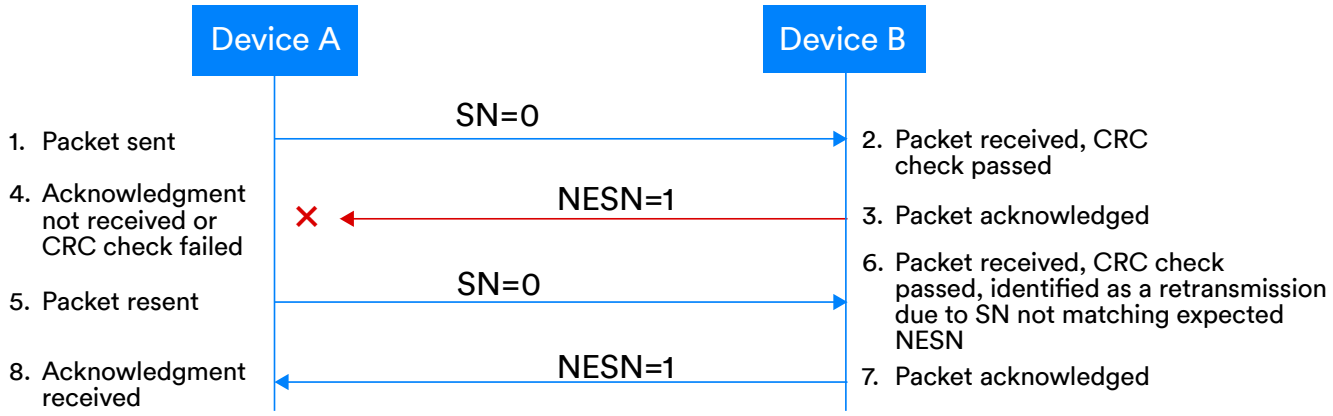

Figure 12 - Link layer retransmissions

Different controller implementations are free to implement varying algorithms regarding how many times to resend before concluding communication to have failed.

As discussed in section 4.1, each packet contains a CRC field and encrypted packets also contain an MIC field. On receiving a packet, the link layer checks the CRC and if present, the MIC. If either check fails, the packet is not acknowledged and this generally results in the originator of the packet resending it.

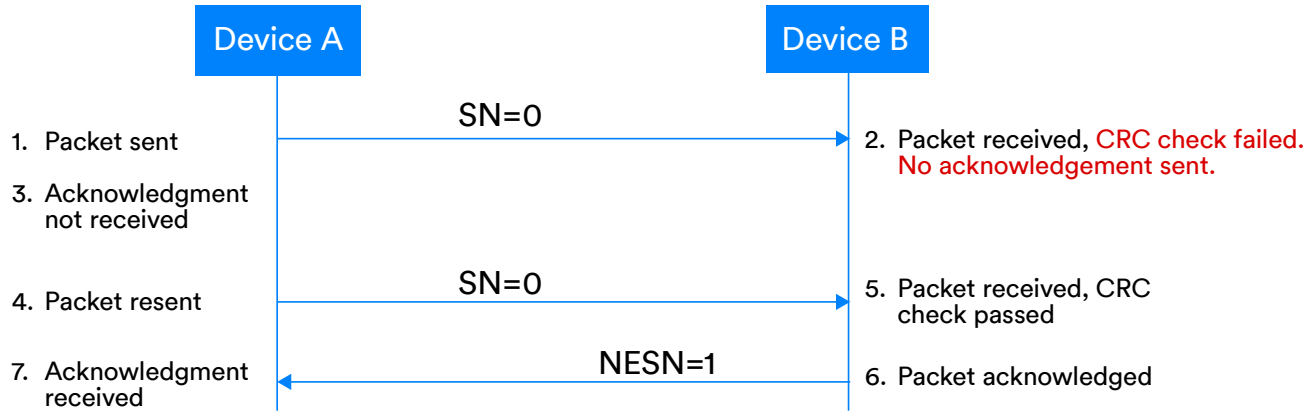

Figure 13 - Link layer handling CRC failure

4.2.4 Flow Control

Many of the reliability issues that the Bluetooth stack is designed to address stem from the fact that radio is used as the carrier of data. But there are other sources of reliability problems that are an issue for wired communication, too.

A device sending data to another device will do so by sending the data in a series of *packets*. If the receiving device is unable to process them quickly enough, it may be forced to start discarding packets and the data they contain as buffers hit their maximum capacity. This was described in section 3.7.

Flow control is the name given to various strategies for ensuring that data is communicated at a rate that can be accommodated by the receiving device or component.

In Bluetooth LE, a simple form of flow control is available at the link layer. By not updating the NESN when sending a reply (ref 4.2.3), the second device can cause the first to resend the original packet at a later time. Since the original packet has already been received and processed, the retransmitted copy will be discarded, slowing the rate of arrival of new packets which need complete processing.

The Logical Link Control and Adaptation Protocol (L2CAP), supports multiple, more sophisticated flow control modes such as the Enhanced Credit Based Flow Control Mode, as used with the Enhanced Attribute Protocol (EATT), which is discussed in section 4.2.5. Credit based flow control is a particular approach to this issue and, in general, it works as follows:

- The transmitting device knows the capacity of the receiving device in terms of the number of PDUs it can handle without losing data (e.g. through its buffer overflowing).
- The transmitter sets a counter to this receiver capacity limit. Every time a PDU is sent by the transmitter, the counter is decremented. When the counter value reaches zero, the transmitter knows the receiver is at full capacity and so stops sending further PDUs temporarily while the receiver processes its backlog.
- After the receiver reads and processes one or more PDUs from its buffer, it sends back a corresponding number of credits to the transmitter which uses this value to increment its counter. With the counter at a non-zero value, the transmitter may continue to send further PDUs.

4.2.5 The Attribute Protocol and the Enhanced Attribute Protocol

4.2.5.1 ATT Transactions

Bluetooth devices may contain a collection of special data entities known as services, characteristics, and descriptors, each of which is a type of *attribute*. Attributes of all types are organized within something called an attribute table. The Attribute Protocol (ATT) is used by an ATT client to discover details of the attribute table in a remote, connected device which is known as the ATT server. Client and server can each use the attribute protocol to interact with the other in a variety of ways.

ATT defines the concept of a transaction. Request PDUs from a client require a response PDU to be returned by the server. Indications sent by a server must be replied to by the client with a confirmation PDU. Each request/response pair or indication/confirmation pair forms a transaction and transactions are a reliability mechanism which indicate whether or not a request/indication was successfully received and processed at the ATT layer of the stack.

Most ATT PDU types are transaction-oriented, but ATT also includes a few PDU types which are not associated with transactions, namely *commands* and *notifications*. Link layer acknowledgements provide assurance that an ATT PDU sent by one device will reach the remote device or if not, the failure will be detected by the sending device.

But for those ATT PDUs which are not transactional, it is possible for a PDU to be received and this confirmed at the link layer, but for the PDU to be then discarded higher up the Bluetooth® stack, perhaps because of buffer overflow. As such, these types of ATT PDU are regarded as *unreliable*.

The Enhanced Attribute Protocol (EATT) is an improved version of ATT which amongst other things, uses the Enhanced Credit Based Flow Control Mode in L2CAP. The use of flow control for EATT means that even the non-transactional parts of EATT can be regarded as *reliable*.

4.2.5.2 Queued Writes

Sometimes data must be written to a device characteristic in multiple steps. This may be because the amount of data to be written exceeds the Maximum Transmit Unit (MTU) size supported. It is common in cases like this for the result of the series of write operations to be valid only if every one of them succeeded. If any of the writes fail, the device must reset the characteristic data value to its state prior to the first of the series of write operations. This *all or nothing* approach to transactions and data change is known as *atomicity*.

To allow multi-step writes to be executed in such a way that the integrity of the data is assured and the overall operation be atomic, ATT provides a set of PDUs which allow *queued writes* to be performed. An ATT client sends the ATT server a series of ATT_PREPARE_WRITE_REQ PDUs, each containing a part of the overall value to be transferred and to which the server responds with an ATT_PREPARE_WRITE_RSP PDU. The response PDU contains a copy of the data written which allows the sender to verify the value.

When all required writes have been performed and each has resulted in a response indicating successful processing of the request, the ATT client completes the operation with an ATT_EXECUTE_WRITE_REQ PDU, which generates a ATT_EXECUTE_WRITE_RSP PDU sent from the server to the client. It is when the ATT_EXECUTE_WRITE_REQ is received by the server that the new value, received via the previous series of ATT_PREPARE_WRITE_REQ is committed.

4.2.6 LE Power Control

Low power consumption is a common design goal for Bluetooth LE products. One of the decisions which product designers must make and which affects power consumption, concerns the transmission power level that the device will use. Informing this decision will be expectations regarding use cases and the range at which the product will typically need to operate, in communicating with another device such as a smartphone.

This approach will inevitably involve compromises, and there will be situations where users are using the product near to the limit of the range supported by the selected transmission power level. Due to path loss and a reduced signal to noise ratio (see section 3.4) at this point, errors are more likely to be experienced, resulting initially in slower communication due to CRC failures and resultant retransmissions and ultimately in connection loss.

Bluetooth LE has a dynamic power control feature which provides Bluetooth

LE devices with the ability to exercise power management by optimizing transmit power levels dynamically. A receiving device that is monitoring the RSSI may request a change in the transmit power level used by its peer in either direction. It may for example, ask the remote device to increase its transmit power level when the RSSI is getting lower or to reduce it when the RSSI is getting high and approaching the point at which saturation might be experienced. Transmitting devices may change their transmit power level autonomously and inform the other device that this has happened, along with various parameter values that include the new transmit power level.

The LE Power Control feature was introduced primarily as a means of allowing devices to minimise their use of power, by ensuring transmission power levels were never more than they needed to be. But there's a reliability benefit here as well. By keeping the RSSI within the range of levels that produce best performance from the receiver, the quality of the signal can be kept high and bit error rates low as the distance between connected devices changes.

4.2.7 Fast Acknowledgments and Fast Failure Detection

Link layer acknowledgements are generated almost at the very bottom of the Bluetooth® stack, which means that this happens very quickly. In technologies involving TCP/IP for example, acknowledgements are a function of a higher layer of the stack.

Similarly, if two CRC failures relating to the same packet occur in succession, the connection event is closed by the link layer, causing the next packet(s) to use a different radio channel. The link layer is able to detect problems and cause remedial action to be taken very quickly in this way.

Fast acknowledgements and failure detection help the system quickly recognise and respond to problems.

4.3 Reliability in Bluetooth Connectionless Communication

In this section, we'll examine reliability when Bluetooth technology is used for connectionless communication. Connectionless communication allows one device to communicate data to one or more receiving devices in each transmission. It can be a completely passive process, with no communication back from receivers to the transmitter. As such it is immensely scalable, with no limit to the number of devices that can be communicated with in this way.

4.3.1 Spread Spectrum in Connectionless Communication

The procedures by which connectionless communication is performed in Bluetooth technology are called the *advertising procedures*. Advertising may be performed in a number of different ways, falling into one of two categories called legacy advertising and *extended advertising*.

Connectionless communication involves one or two different spread spectrum techniques, depending on whether legacy advertising or extended advertising is used.

Legacy advertising involves transmitting a copy of each advertising packet, on up to three of the channels with channel index 37, 38 and 39, one channel at a time. The advertising channel selection algorithm picks channels in a random order, as depicted in Figure

14. Channel index will be explained in section 4.3.2.

Channels 37, 38 and 39 are called the *primary advertising channels*.

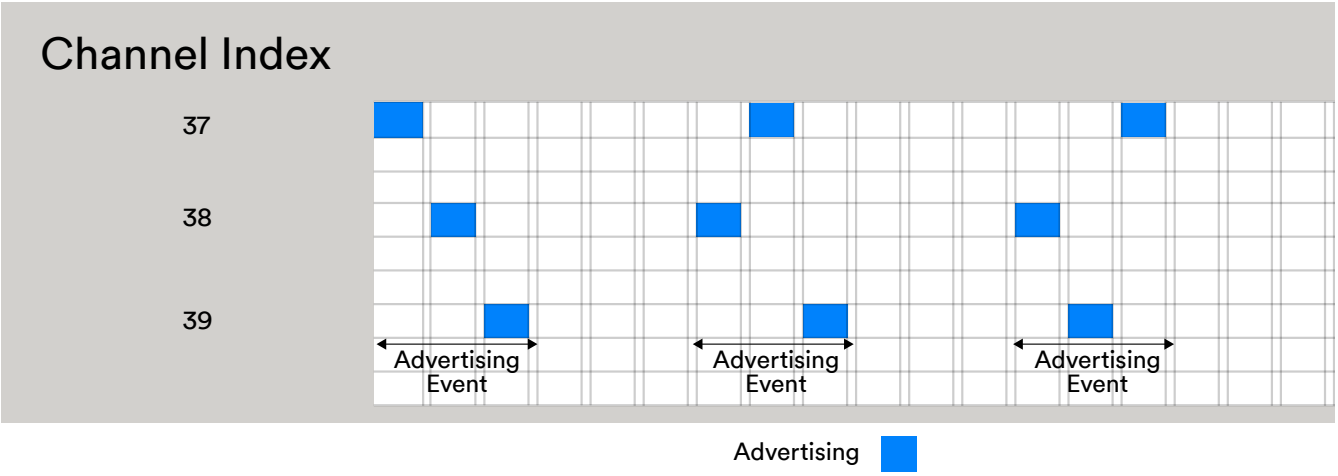

Figure 14 - Legacy advertising using the three advertising channels

Extended advertising may use all 40 channels, with one type of packet transmitted on the primary advertising channels, containing a pointer to the payload which is transmitted in a separate packet using one of the 37 general purpose channels. The algorithm used to select the channel index for extended advertising purposes is an implementation decision, but the Bluetooth® Core Specification does recommend that it result in *sufficient channel diversity to avoid collisions*.

The use of multiple channels for the broadcasting of data, is a spread spectrum technique which helps mitigate the risk of collisions in busy radio environments, making connectionless communication more reliable.

4.3.2 Coexistence and Advertising Channels

Bluetooth LE radio channel numbers run from 0 to 39, ascending with the associated radio frequencies in the 2.4 GHz ISM band. The Bluetooth link layer selects channels by *channel index* rather than channel number, however and the sequence of channel indices, as the frequency increases, is not strictly linear.

Figure 15 - Advertising channel activity

Figure 10 shows channel indices alongside frequencies in the ISM band. Largely, the channel indices do ascend in a linear way, but three cases break that pattern and these are the primary advertising channels. Bluetooth® LE channel 37 sits at the very bottom of the band, centred on 2402 MHz. Channel 38 sits at 2426 MHz and channel 39 at the top end of the ISM band, at 2480 MHz. There are two reasons for this strange looking distribution of the advertising channels.

The first is that the three channels are widely separated from each other. This is to ensure that advertising continues to work, even in the event that a significant part of the ISM band is subject to powerful interference. Had the three advertising channels been placed close together, then this phenomenon, known as a *deep fade*, would have blocked advertising entirely.

The second reason is to avoid channels known to be used by Wi-Fi.

The careful assignment of the advertising channels to these distributed regions of the ISM band, helps make advertising reliable. Figure 15 shows the three advertising channels in use by a number of advertising devices.

4.3.3 Avoiding Persistent Collisions of Advertising Packets

The sequential use of the three reserved advertising channels brings with it a theoretical risk that several advertising devices in range of each other, could repeatedly select the same channel, broadcast at approximately the same time and get into a pattern where their packets were frequently and persistently colliding. This risk has been mitigated in two ways.

1. The scheduling of advertising is governed by a timing parameter called the *advertising interval* and a random delay of between 0 and 10ms. Each time this time period starts, we have what is referred to as an *advertising event* and the link layer broadcasts the appropriate advertising packet on one, two, or three of the advertising channels (depending on implementation choices), one at a time and in some sequence. The random delay, which is a factor in the scheduling of every advertising event, introduces a perturbation in time and helps ensure persistent collisions between advertising devices will not occur.
2. Advertising channels can be selected for use in any order at each advertising event. By effectively randomising the order in which advertising channels are used in each advertising event, the probability of collisions is further reduced.

4.3.4 Periodic Advertising

Connectionless communication has the potential to be less reliable than connection-oriented communication because receivers and transmitters are typically operating completely independently of each other. Therefore, data may be transmitted when one or more of the intended receiver devices are not listening, resulting in that data being lost from those devices' points of view. Figure 2 illustrates this.

To counter this issue and improve reliability, developers can program receiver devices to use a very high RX duty cycle, meaning that a very high proportion of the

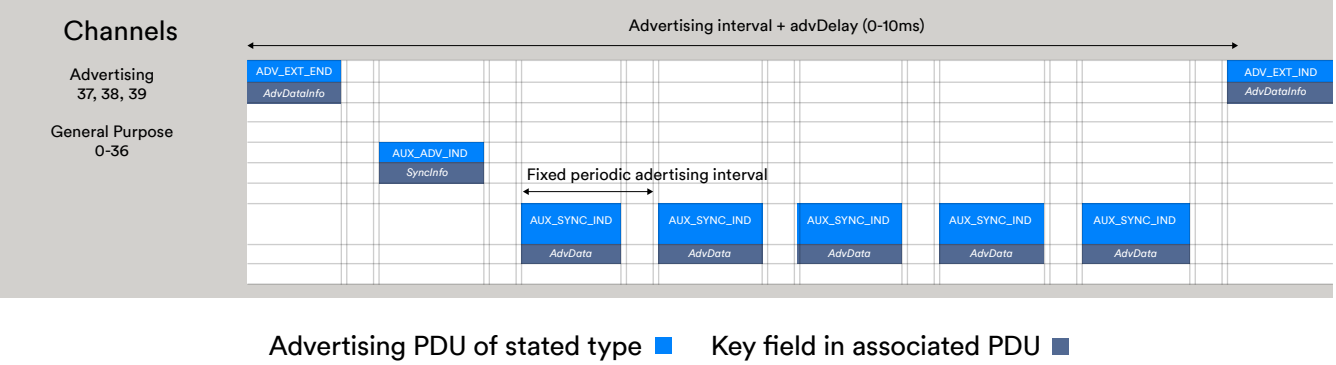

Figure 16 - Periodic Advertising

time, the radio is listening for transmitted data. This can dramatically reduce the chances of missing broadcast data, but it will also significantly increase power consumption.

An alternative approach is available with some devices. Bluetooth® technology has an optional feature known as *periodic advertising*. Periodic advertising allows advertising to take place at fixed intervals, with no random perturbation but using adaptive frequency hopping over 37 channels. Receivers can also discover information about the advertising device's periodic advertising schedule and then synchronise their scanning precisely with it. In this way, reliability can be increased without sacrificing power efficiency.

Periodic advertising is one feature of the Bluetooth LE *extended advertising* feature set and involves several PDU types, as illustrated in Figure 16.

4.3.5 Broadcast Isochronous Streams

One of the newest features to have been added to Bluetooth LE is that of *isochronous channels*. Isochronous channels are designed to allow time-bound data to be communicated to multiple devices such that the data is acted upon by those devices at exactly the same time.

Isochronous communication was primarily designed for use in audio products and systems. It provides the means by which audio, delivered from a source to multiple sinks, can be rendered at the same time, for properly synchronised playback. Audio data has a limited time during which it is valid after being generated at the source. If this time expires, the audio data is discarded so that it does not affect the listening experience at the sink(s).

Isochronous communication may be connectionless whereby data is delivered over a *broadcast isochronous stream (BIS)* to a potentially very large number of receivers.

Broadcast isochronous communication offers no means for receivers to acknowledge the receipt of packets. Instead, BIS reliability may be enhanced through the unconditional repeated transmission of identical packets ahead of time. Retransmissions are transmitted on different channels and selected channels must be at least 6 MHz from the last transmission. This strategy provides both frequency and time diversity and helps

mitigate potential packet loss due to interference on a particular channel or group of adjacent channels.

4.3.6 Connection-Oriented vs Connectionless

Whilst connectionless communication is always likely to be less reliable than connection-oriented communication, Bluetooth® provides mechanisms such as periodic advertising and retransmissions in broadcast isochronous streams which can be used to maximise reliability. Developers can also improve the reliability of connectionless communication by adjusting the parameters used. We discuss this further in section 5.

4.4 Reliability in Bluetooth Mesh Networks

The Bluetooth mesh protocol stack resides in the host part of the Bluetooth system architecture. It uses the Bluetooth LE controller and the standard Bluetooth LE air interface packet structure, as shown in Figures 3 and 4. As such, Bluetooth mesh benefits from each of the points about reliability that were made in section 4.1.

In this section, we'll explore those aspects of Bluetooth mesh that are designed to help ensure that communication in the network is reliable but will not revisit the underlying capabilities of the Bluetooth LE controller which have already been described.

One point sets the subject of reliability in the context of Bluetooth mesh apart from the other contexts in which the subject was examined in previous sections. A mesh network is a network. There are additional, network-related issues which require a different type of approach for reliability to be achieved.

4.4.1 Background

4.4.1.1 Bearers

Bluetooth mesh supports more than one method for using the Bluetooth LE controller to transport mesh PDUs, and these methods are called *bearers*. Both connectionless and connection-oriented approaches are supported, using the *advertising bearer* and the *GATT bearer* respectively. But use of the advertising bearer is much more common, with the GATT bearer usually only used to allow devices like smartphones to be part of the mesh network, via a special mesh node feature known as the *proxy feature*.

4.4.1.2 The Advertising Bearer

The Bluetooth mesh advertising bearer allows mesh PDUs to be encapsulated within a specific type of Bluetooth LE advertising packet called *ADV_NONCONN_IND*. A naming standard for link layer PDUs such as this one can be found in the Bluetooth Core Specification at Volume 1, Part E, section 3.2.1 and from this we can see that these are non-connectable advertising packets which are sent on the standard advertising channels (37,38 and 39) and for which no response PDU is defined.

In simple terms, a mesh message is sent or published by broadcasting an ADV_NONCONN_IND advertising packet which contains the mesh PDU within the advertising data. A mesh node receives mesh PDUs by scanning for the same type of advertising packets and passing them up the Bluetooth® mesh stack for further processing.

There are some reliability implications relating to the use of ADV_NONCONN_IND PDUs by the mesh advertising bearer. Only the three advertising channels are available for use and while this provides a basic spread spectrum technique, the more sophisticated and reliable adaptive frequency hopping does not apply. This is connectionless communication whose reliability issues were discussed in section 4.3.

Importantly, *periodic advertising* is not used as the bearer for mesh PDUs and so there is no synchronisation between the timing of advertising performed by mesh nodes when publishing messages and the timing of the scanning performed by nodes wishing to receive mesh messages.

4.4.1.3 Stochastic Behaviours

There are a number of random and unpredictable factors governing communication in a Bluetooth mesh network. For example:

- When a node publishes a message, will all of the destination nodes be listening at just the right time and on the right advertising channel to receive it?
- Will a path through the network, along which a message needs to be relayed be available at that time?
- Could high volumes of mesh messages in the parts of the network that the message will be relayed through cause collisions?

A Bluetooth mesh network should be thought of as a stochastic system, with inherent randomness and consequential unpredictability. On its own, these factors could lead to very poor reliability, but Bluetooth mesh includes a number of mechanisms which make the message-oriented communication between nodes in the network achieve ***fit for purpose*** reliability.

Given its stochastic nature, to understand reliability in Bluetooth mesh networks it can be helpful to think in terms of probabilities rather than in terms of deterministic sequences of cause and effect.

4.4.1.4 RX Duty Cycle

When performing connectionless communication, the RX duty cycle (as discussed in section 3.3) is an important parameter that affects the probability that a transmitted packet will be received. For this reason, the Bluetooth mesh profile specification recommends that "*a device supporting only the advertising bearer should perform passive scanning with a* ***duty cycle as close to 100 percent as possible*** *in order to avoid missing any incoming mesh messages or Provisioning PDUs*".

Up to three different advertising channels may be used by the advertising bearer, per any use of Bluetooth advertising. The advertising bearer transmits a copy of each

mesh network PDU on each of the advertising channels that are in use. Using all three channels, reduces the probability of collisions and therefore would be generally recommended.

For receivers though, using more channels effectively reduces the duty cycle, viewed on a per channel basis. If all three advertising channels are in use then each channel can be scanned for less than one third of the available time. Why less than a third? Because the radio hardware will take some time to switch channels and when switching, the radio is effectively off.

Note: In special cases where a very high RX duty cycle of as close to 100% as possible on all three channels is required, it is technically possible to include multiple radios in a product, each dedicated to one channel.

It should also be noted that when the advertising bearer is used by Bluetooth® mesh, the specification does not mandate exactly when advertising channels should be switched or how long one channel should be used for, so there is flexibility for an implementation to make use of the three advertising channels in a way which is optimal for the device and its environment.

Consider Figure *17*.

Figure 17 illustrates what could happen in two Bluetooth mesh devices in their use of the

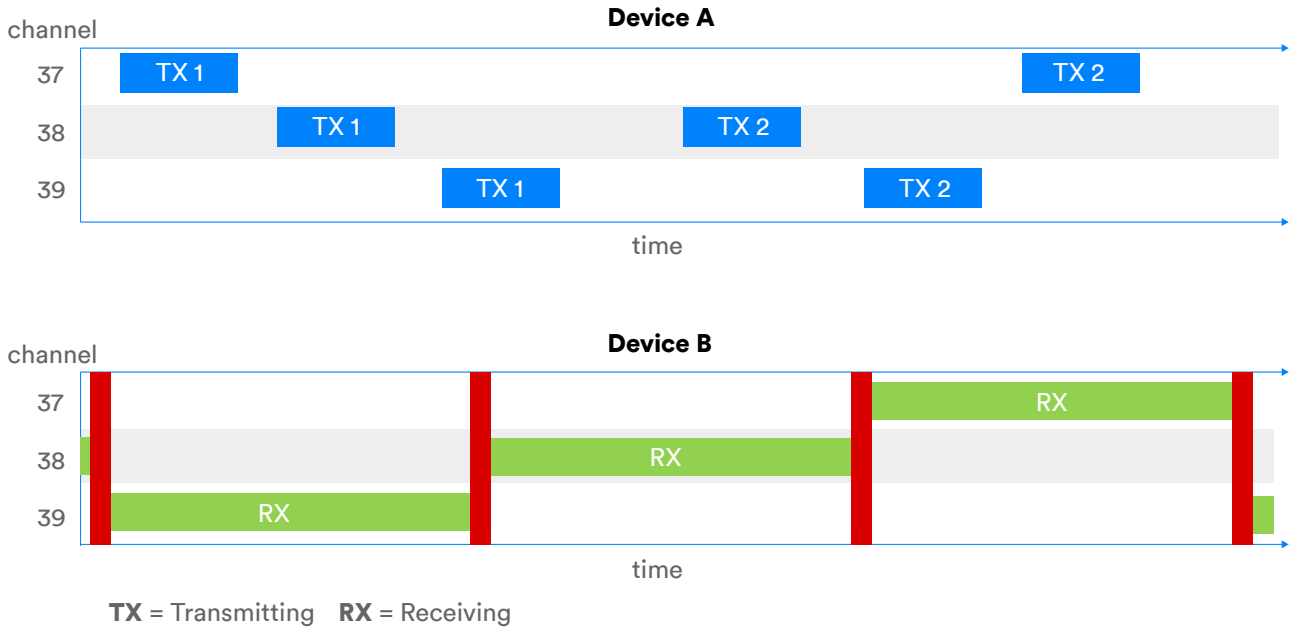

Figure 17 - Unsynchronised Advertising and Scanning

advertising bearer.

Device A sends two PDUs which Device B is intended to receive. Copies of each of the two PDUs are broadcast on each of the three advertising channels, one at a time in rapid succession.

Device B is listening, by scanning at intervals and for a given duration on each of the three channels in turn. The red regions indicate the short period during which channel switching is taking place and data cannot be received.

The first mesh PDU, broadcast in the three packets labelled “TX1” will not be received. Device B is never scanning on the channel that Device A is broadcasting on for the full duration of that broadcast. The packet labelled “TX2” will be received on channel 38.

This is a normal and expected scenario in a Bluetooth® mesh network. There is always a certain probability that a broadcast packet will not be received by a destination node, since the RX duty cycle per channel will never be more than around 33%.

The next sections explain how Bluetooth mesh increases the probability of successful message delivery.

4.4.2 Achieving Reliability in a Bluetooth Mesh Network

4.4.2.1 Efficient Network Utilisation

The probability of persistent, repeated collisions occurring, and ultimately causing the failure of an operation in the network, depends on how the radio spectrum is being used to support the logical operations represented by published messages.

Bluetooth mesh is designed to make efficient use of the shared radio spectrum so that the risk of collisions is reduced. For example:

- PDUs are at most 29 octets in length. Commonly used message types like those used to switch devices on or off are only 22 octets in length. There’s some additional data in the Bluetooth LE packet which wraps the mesh PDU, but only about another 18 octets.
- Bluetooth LE offers the fastest radio of the low power wireless communications technologies, with a symbol rate used by Bluetooth mesh of 1 Msym/s.
- The TTL field allows the number of times a message is relayed to be controlled so that spectrum use is limited to relevant parts of the network.
- Bluetooth mesh uses a decentralised architecture for lighting control with control logic implemented in software inside lighting nodes rather than in physically separate, dedicated control units. This has a dramatic effect on network utilisation, sometimes producing as little as 1% of the traffic that a centralised architecture would generate.

Small packets transmitted by a fast radio mean that each operation requires the radio spectrum to be used for the briefest time and therefore with a much lower probability of collision.

An article on Bluetooth mesh and scalability, published on the Bluetooth SIG website explores this topic at length. See bluetooth.com/mesh-scalability

4.4.2.2 Network Layer Retransmissions

Figure 3b shows the layers of the Bluetooth® mesh stack. Sitting above the bearer layer is the *network layer*. The network layer can be configured to automatically retransmit copies of PDUs multiple times and at specified intervals.

Network retransmissions increase the probability of a mesh message being received by a destination node. Imagine the probability that a mesh network PDU, transmitted on the three channels will not be received is 10 percent. If two copies of the PDU are transmitted, in rapid succession, then the probability that neither is received is 1 percent. If a third transmission is performed, then the probability of message loss becomes 0.1% or to put it another way, we have achieved a success rate of 99.9%. Retransmissions rapidly and dramatically reduce the probability of message loss and conversely, increase the probability of success.

Network layer retransmission parameters exist as configurable states that must be present in the primary element of every node. Two composite states are defined, the first of which defines the network retransmission parameters for the node when it is the originator of a message (Network Transmit) and the second which defines the network retransmission behaviour when the node is relaying a message (Relay Retransmit).

Typically, Network Transmit is configured so that more retransmissions are performed by a node when it is the originator of a message than when it is a relay.

Network designers must consider the value to assign to the network and relay retransmission configuration parameters so as to achieve the required reliability, whilst avoiding overuse of the radio spectrum.

4.4.2.3 Model Publication, Retransmissions and Synchronisation

Mesh models may publish messages in response to external events such as a button being pressed by the user or automatically, at configured intervals. Retransmissions with which to increase reliability may be configured for model publication of messages and are performed at the application layer rather than at the network layer, as was described in 5.4.5.

Each model supported by a node has an associated composite state called Model Publication. Included within this state are states such as Publish Retransmission Count and Publish Retransmit Interval Steps. These states allow the configuration of different retransmission behaviours for messages published by each type of model.

In lighting systems, it is common to want groups of lights that are controlled by the same switch or sensor, to be perceived by human observers to have responded to messages at exactly the same time. Bluetooth mesh allows this to be achieved and avoids the appearance of jitter across the group of lights, sometimes known as the *popcorn effect*.

Most Bluetooth mesh messages include a *delay* field, which specifies a number of milliseconds that a node receiving the message should wait before acting upon it.

This can be exploited in conjunction with model retransmissions to achieve the required synchronised behaviour across groups of nodes to which a message is addressed.

Figure 18 depicts a light switch which controls six lighting nodes. The switch implements the *generic on off client model* and the light nodes implement the *generic on off server model*. The client model has been configured to transmit three copies of the *generic on off set unacknowledged* message that it sends, at intervals of 50ms. It sets the *delay* field of the first message to 100ms and then reduces this value by 50ms at each of the two retransmissions. This produces a synchronised response to the act of pressing the light switch, regardless of which of the three message transmissions each light first receives.

In the scenario depicted in Figure 18, four of the six lights successfully receive the first transmission, but wait for the specified 100ms delay before acting upon it. 50ms after sending the first copy of the message, a second copy is transmitted but this time the delay field contains a value of only 50ms. Lighting node #5 receives this message and waits for the specified 50ms. Finally, the third transmission occurs, 100ms after the first was sent and this time with a delay field value of zero. Lighting node #2 receives this message and immediately acts upon it, at exactly the same time as node #5, which had waited 50ms before responding to message #2 and the other nodes which received message #1 and waited for 100ms. The net effect is that the user who pressed the light switch observes all 6 lights coming on at the same time and with an imperceptible delay.

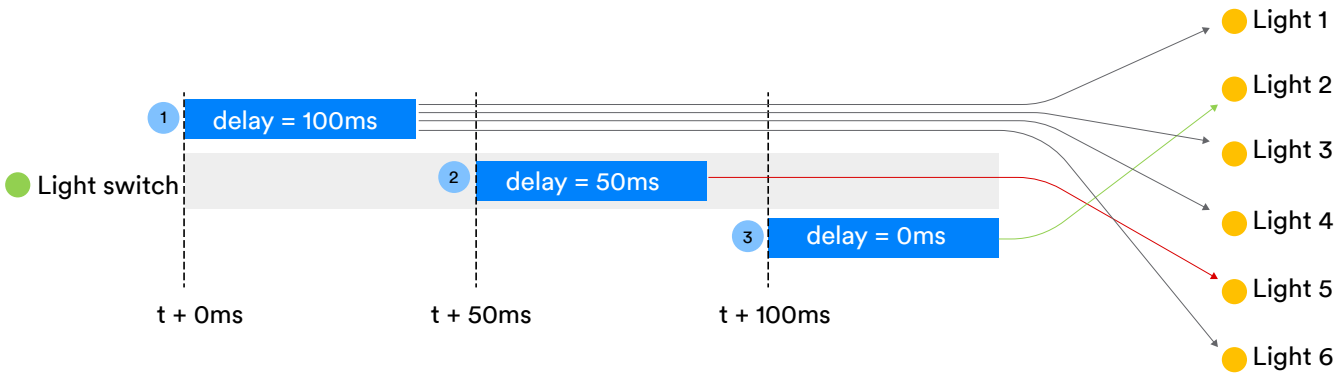

Figure 18 - Using delayed execution to synchronise node message responses

4.4.2.4 Eliminating Single Points of Failure

When a Bluetooth® mesh message is published by a node, it may travel a considerable distance across the network to its destination(s), significantly beyond the direct radio range. This is accomplished through a process called *relaying*.

Relaying involves the retransmission of a received mesh network PDU by a node which has had the *relay feature* enabled, known as a *relay node*. This allows messages to hop across the network, from relay to relay, until a destination node is reached. The number of hops a message might take can be limited by a message parameter called Time To Live

(TTL) so that messages do not hop further than is necessary.

In addition to allowing communication with nodes anywhere in the network no matter how far away they are, relays also allow multiple delivery paths to be created. Consider Figure 19.

The green circle represents a Bluetooth® mesh light switch and the yellow circle is a single lighting node which the switch controls. Each of the red nodes is a node which is configured to act as a relay.

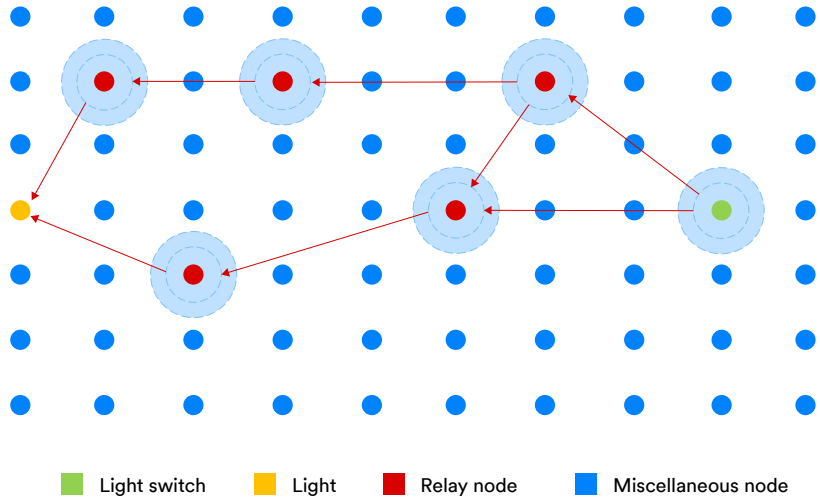

Figure 19 - Multipath delivery using relays

Pressing the light switch causes a message to be broadcast. Those relay nodes that are in direct radio range, retransmit it and this process repeats until the message reaches the light. As a consequence of the network's design, particularly the choices made regarding which nodes to use as relays, we can see that there are three paths along which messages can travel from the light switch to the light under control. When the switch is pressed, a copy of the message travels concurrently along each path. Delivery of copies may be staggered due to differing path lengths and node processing times and the first copy to arrive at the light will be acted upon, with later arrivals recognised as duplicates and discarded.

The use of relays in this way, to create multiple delivery paths for messages, introduces *redundancy* to the network and eliminates single points of failure. If one path is momentarily not available, perhaps because a relay on the path is switching channels at that moment, the message will very probably be successful over one of the other paths. And don't forget, the light switch will use network retransmissions so that the same message is transmitted multiple times in rapid succession and a copy of each transmission is then relayed along different paths.

The combination of network retransmissions and multi-path delivery allows Bluetooth mesh to deliver excellent reliability in the network.

4.4.2.5 Acknowledged vs Unacknowledged Messages

Bluetooth mesh uses a system of standard messages to obtain or change the state of nodes in the network. All messages have a source (*SRC*) and destination (*DST*) address.

All independent parts or *elements* of every node have a unique, 16-bit unicast address but there are also two types of address by which collections of devices can be

addressed by a single message. These are known as *group addresses* and *virtual addresses*. Most messages are sent to one of these address types and this is known as *multicast messaging* because one transmitted message is logically addressed to multiple destination elements. The ability for a single message, consisting of a quite small number of bytes to reference potentially thousands of target devices in this way, makes highly efficient use of the available radio spectrum and so is a very scalable way of supporting the common, one-to-many communication uses cases in typical mesh networking scenarios such as in
smart buildings.

Bluetooth® mesh does not require a distributed database containing lists of group and virtual addresses and the nodes that are members of the logical collections that they identify to be maintained in the network. Instead, a *publish/subscribe* system is used. When initially set up, part of the configuration of a node is to indicate to it the destination addresses to which it must react. This is called *subscribing*. Nodes will ignore any messages that their radios receive at the bottom of the stack and whose DST addresses it has not subscribed to.

Messages which change state in destination devices are called *set messages*. There are two types of set message. Set messages may either be *acknowledged* or *unacknowledged*.

There are a great many protocols in the world which use a system of request and response PDU pairs, where a request must result in a response which indicates the outcome of the request and must be returned within a time period known as the *timeout*. HTTP is a good example of a protocol which is designed in this way.

Acknowledged set messages in Bluetooth mesh are comparable to request/response PDUs in other protocols. A set *acknowledged* message of some type, will result in each destination node that receives and processes the set message replying with a *status* message which acts as
an acknowledgement.

Unacknowledged set messages are not responded to by target nodes.

In a typical Bluetooth mesh network, unacknowledged messages are used far more often than acknowledged messages, which are only ever used in special situations, such as when configuring a single node directly. The reason for this perhaps counterintuitive choice is as follows.

Consider the following scenario. Imagine an open plan office space containing 50 desks, illuminated by a total of 200 LED lighting fixtures, each of which is a Bluetooth mesh node. A light switch at the entrance to the floor can be used to switch on all 200 lights. On pressing the light switch, if acknowledged messages were to be used, something like this might happen:

1. The switch sends its on/off message, addressed to the group address that all lights in the office have subscribed to. All nodes in direct range receive the message and act upon it. Those that act as relays, retransmit the message.

2. Nodes in range of the first set of relays, receive and respond to the message. Those that act as relays, retransmit the message.
3. The actions described in (1) and (2) are repeated as copies of the original message travel across the network until all destination nodes have been reached.

At this stage, the behaviour of the system as described is exactly how Bluetooth® mesh **does** work. Under normal circumstances, the original message sent by the switch would have been an unacknowledged message, the lights would be on and the scenario would have ended here. But if an acknowledged message had been sent though, this is what happens next:

Each of the 200 nodes that had subscribed to the group address used by the switch, on receiving and acting upon the message, immediately reply with a status message, which acts as an acknowledgement. This causes a large spike in network utilisation, as 200 messages are broadcast by each of the lighting nodes, in a very short time period. These messages get received and repeated by relays and some but sometimes not all of them, arrive back at the light switch.

But things are not over yet because handling acknowledged messages from groups of devices can get complicated very quickly and here's why:

Bluetooth mesh does **not** define a way of using acknowledged messages to track the outcome of messages sent to group addresses, so this is something which would need to be implemented as a custom behaviour (and for that reason alone is not recommended). One way this could work is for the sending node to contain a list of all the unicast addresses of all destination nodes that have subscribed to the group address. As acknowledgements are received, each with the unicast address of the sending node as its SRC address, it could use this list to determine which nodes have acknowledged and which have not. The need to maintain this list is itself an issue because it must be maintained across every node which will ever send a message to a group address. But that's not the only problem.

The switch will then need to wait for a period of time. Examining its acknowledgement tracking table, it notes a number of nodes from which acknowledgements have not been received. This does not necessarily mean that the original set message was not delivered to the associated light. It means the acknowledgement was not sent by that light or it was sent but not received and there is no way of knowing which of these two conditions the lack of acknowledgement signifies. Consequently, the switch retransmits its message to the group address or sends multiple individual messages, each addressed to the unicast address of those nodes from which no acknowledgement has been received. The process then repeats, however many times the switch wants to keep trying, with the on/off message resent and acknowledgement receipt tracked until all nodes have replied or an overall timeout or retry limit is reached.

To be clear, **this is not how Bluetooth mesh works**. But if it did, as you can see it would not work well. The huge spike in network traffic would cause congestion and probably impact other operations in the network taking place at that time. And the

complexity of the network and the individual nodes required to be able to track acknowledgements in this way, would be an order of magnitude greater than it needs to be.

Multicast messaging in wireless communication systems is notoriously difficult to make scalable and reliable and that's why ***Bluetooth® mesh takes a different approach****.*

Unacknowledged messages with network layer retransmissions and multiple paths providing redundancy work extremely well and suffer from none of the complexity, capacity and reliability problems that acknowledged messages do when used with group addresses.

4.4.2.6 Bluetooth Mesh and Reliable Lighting Systems

A Bluetooth SIG paper entitled Building a Sensor-Driven Lighting Control System Based on Bluetooth Mesh is available and offers further recommendations for building effective and reliable Bluetooth mesh networks.

5.0 Getting the Best out of Bluetooth® Reliability

Bluetooth technology incorporates features that are designed to enable reliable communication, distributed throughout the stack, starting at the very bottom where the radio is put to work in a smart and effective way, with a deliberately chosen modulation scheme, spread spectrum techniques and error detection and correction capabilities. Much of the reliability exhibited by Bluetooth technology happens automatically therefore. But there are ways in which product and application designers and developers can use Bluetooth technology so that reliability is maximised.

A selection of some of the key considerations are summarised in Table 1.

Tip	Applicable Usage Type	Additional Comments
Think about the physical environment and device placement	all	Environmental issues can make a difference to the reliability of communication. Where possible, consider issues such as device location density and the possible impact of physical barriers.
Use connection-oriented communication where possible	connection-oriented	By its nature, connectionless communication presents more reliability challenges than connection-oriented communication, and therefore additional measures are required.
Use ATT transactions	connection-oriented	If reliability is a prime concern, and the attribute protocol (ATT) is to be used, then request/response and indication/confirmation transactions will deliver the best reliability.
Use EATT	connection-oriented	Use the enhanced attribute protocol (EATT) instead of ATT if possible to achieve better reliability through the use of flow control.

Tip	Applicable Usage Type	Additional Comments
Use a message authentication check	connection-oriented, mesh	Encryption will add a message authentication check (MIC) to all packets and prevent deliberate, malicious changes to packets from being possible without detection.
Make sure buffer sizes are large enough to handle busy environments without packet loss due to overflow	connection-oriented, connectionless, mesh	
Use a high RX duty cycle to ensure broadcast data is received	connectionless, mesh	This may need to be balanced against power consumption goals and constraints.
Use periodic advertising if possible to synchronise scanning with advertising	connectionless	
Create redundant paths through the network by placing relays thoughtfully.	mesh	
Configure network retransmissions in each node, balancing reliability requirements with network utilisation goals	mesh	
Use hardware which has a fast radio channel switching time	mesh	Fast switching results in very short periods where the radio is unable to receive.

Table 1 - Tips for Optimising Reliability

6.0 In Conclusion

Bluetooth® technology is capable of achieving highly reliable communication in even the most challenging circumstances. Much of the system was designed with reliability in mind, from features like adaptive frequency hopping through to flow control and transactions in the enhanced attribute protocol. Designers and developers can optimise the reliability of their products and applications in a number of ways, too.

Reliability in Bluetooth technology is more than just the sum of the reliability of its parts.

Bluetooth technology is reliable by design. ■