SPP Throughput Analysis

BT900

Application Note *v1.1*

Introduction

The goal of this document includes the following:

1. To explain the configuration options for tweaking throughput for SPP connections.
2. To detail the testing procedure to verify throughput.
3. To detail Laird's testing results.

Overview

The BT900 module embeds a full Bluetooth stack up to and including the highest RFCOMM layer which is used in a Serial Port Profile (SPP) connection.

RFCOMM is a streaming protocol that sits on top of an underlying layer called L2CAP, which is packet based, and is used to manage the underlying packet-based radio baseband.

That means when data is received or sent by RFCOMM there is no guarantee that packet boundaries are preserved. For example, if "Hello" were to be transmitted, RFCOMM ensures that all 5 characters arrive at the peer RFCOMM layer in that same order, but for over-the-air, there is no guarantee that a single packet will carry all 5 bytes. There may be as many as 5 packets, each with a single byte payload.

Background

ACL Packets

The Following table shows the summary packet types exchanged between two basebands shown in the Bluetooth Specification 4.0 Vol 2, Part B, section 6.7.

Table 1: Type of ACL Packet Against Max Attainable Throughput Rates

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Symmetric Max. Rate (kb/s)	Asymmetric Max. Rate (kb/s)	
						Forward	Reverse
DM1	1	0-17	2/3	Yes	108.8	108.8	108.8
DH1	1	0-27	No	Yes	172.8	172.8	172.8
DM3	2	0-121	2/3	Yes	258.1	387.2	54.4
DH3	2	0-183	No	Yes	390.4	585.6	86.4

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Symmetric Max. Rate (kb/s)	Asymmetric Max. Rate (kb/s)	
						Forward	Reverse
DM5	2	0-224	2/3	Yes	286.7	477.8	36.3
DH5	2	0-339	No	Yes	433.9	723.2	57.6
AUX1	1	0-29	No	No	185.6	185.6	185.6
2-DH1	2	0-54	No	Yes	345.6	345.6	345.6
2-DH3	2	0-367	No	Yes	782.9	1174.4	172.8
2-DH5	2	0-679	No	Yes	869.1	1448.5	115.2
3-DH1	2	0-83	No	Yes	531.2	531.2	531.2
3-DH3	2	0-552	No	Yes	1177.6	1766.4	235.6
3-DH5	2	0-1021	No	Yes	1306.9	2178.1	177.1

The baseband will try to use the packet type which gives the greatest throughput for an ACL connection, so it is important to be aware of these boundaries when tuning for throughput. If, however, there is interference and a large number of re-transmissions the baseband will generally opt for smaller packets to sustain communication.

If the data queued exceeds the 3-DH5 packet size the transfer will be chunked into multiple transactions; when this occurs overall throughput gets degraded. For example between a payload of 1021 and 1024 the bitrate drops by ~25% due to the additional packet and ACK.

For this reason we set a maximum ACL payload of 1021 bytes.

L2CAP

The L2CAP packet format is described in the figure below, shown in the Bluetooth Specification 4.0 Vol 3, Part A, section 3.1.

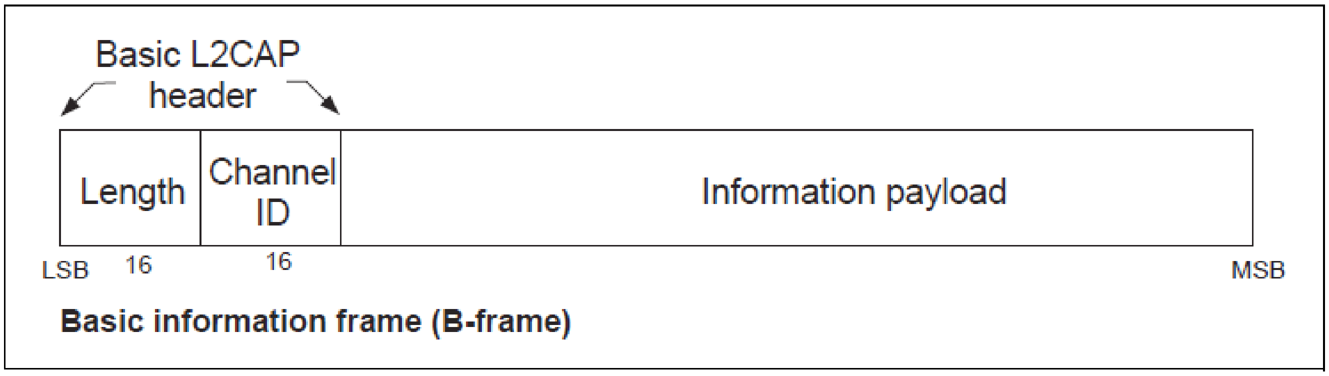


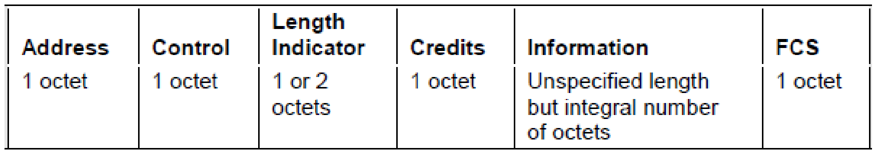
Figure 1: L2CAP PDU format in Basic L2CAP mode on connection-oriented channels (field sizes in bits)

The header spans 4 bytes so to ensure that the ACL payload length restrictions are adhered to the maximum L2CAP, we limit payload to 1017 bytes, (1021 – 4).

RFCOMM

The RFCOMM protocol is a credit based protocol; a credit represents a packet that a device can send or receive at a time. Each transmission uses a credit, and once the count reaches zero the receiver must provide more credits to the transmitter before it can receive more packets.

The packet structure changes based on the P/F bit setting but it is between 5 and 6 bytes and is described below in the RFCOMM Spec v12, Figure 6.1.



Address	Control	Length Indicator	Credits	Information	FCS
1 octet	1 octet	1 or 2 octets	1 octet	Unspecified length but integral number of octets	1 octet

Figure 2: Frame structure for Basic option, UIH frames with P/F-bit = 1 and credit based flow control used

To keep within the L2CAP payload, restrictions outlines above the RFCOMM payload is limited to 1011 bytes, (1017 – 6).

TUNING

We expose a series of controls to allow the tuning of the various layers, enabling users to tune the software for their use case. These are intended for advanced use and usually the defaults are sufficient for most people.

L2CAP

Internally there is a queue at the L2CAP layer. To achieve a high throughput it is important to try and keep the link saturated with data.

RFCOMM

One of the most important and the subject of this app note is the RFCOMM frame size. We impose an artificial limit on the frame size of 1011 bytes for reasons detailed above. Realistically there is little performance increase above ~256 bytes. As there are RAM limitations of the device this parameter should be tuned with care.

Affecting the incoming stream only is the number of available credits to the transmitting device. This value represents the size of an internal buffer (RFCOMM frame size * Credits) and should be tuned with care as it is replicated per open RFCOMM channel, thus easy to run out of memory.

The above tweaks are made available through the configuration function BtcSPPSetParams().

EXAMPLE

The following experiment shows the effects of tuning the RFCOMM frame size only on throughput with the following configuration:

- BT900 UART set to 921600bps
- BT900 paired with a Windows 7 (64bit) machine using at BT820
- BT900 set to bridge SPP port with UART using StreamBridge()
- BT900 once SPP established, connectable/discoverable turned off
- Windows set to non-discoverable
- Antennas connected using a simulated 40dB signal loss

Testing Procedure

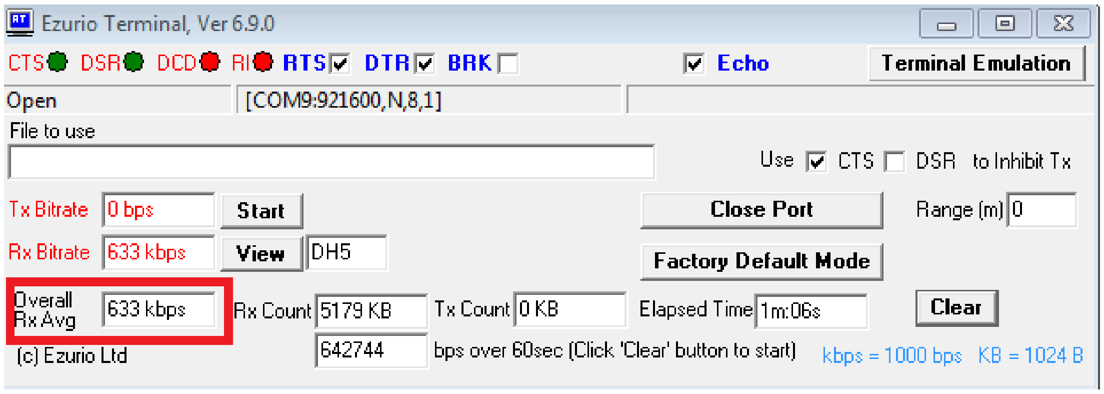
To test the BT900 throughput, we used the following procedure:

1. Modify the following sections of the BT900 $autorun$.SPP.UART.bridge.incoming.sb program:
   - #define UART_BAUDRATE (921600)
   - #define MAX_FRAME_SIZE 256 *or* 512 *or* 1011 (choose one)
   - #define RX_CREDITS 8
2. XCompile/Load/Run the script at 115200 baud rate in BT900 UWTerminal (Host A).
3. Once the script is running, change baud rate to 921600 in *Config* Tab and click **OK**.
4. Pair the BT900 (Host A) to Host B by entering “pair <MAC address of host B>” (no quotes or carets).

Note: In our case, Host B is a Windows 7 PC with a BT820 dongle attached.

5. Follow the on-screen prompts to accept the pairing. Once paired, COM ports should be created on Host B.
6. When the COM ports are created on Host B, click **Close Port** in UWTerminal on Host A.
7. Open an instance of Ezurio Terminal on each host.
8. On the Host A Ezurio Terminal, choose the appropriate COM port and Baud Rate (921600) and confirm CTS light is green.
9. On the Host B Ezurio Terminal, choose the appropriate COM Port and Baud Rate (921600). All lights should be red until connected via SPP to the BT900.
10. Enter “connect <MAC address of Host B>” (no quotes or carets) in the BT900 Ezurio Terminal. The other Ezurio Terminal should show the CTS and DSR icons green.
11. Confirm you can terminal chat between both hosts.
12. Once the hosts are communicating, click **Data Transfer Test** button in top right hand corner on both terminals.
13. Clear everything before testing and click **Start** button to begin testing.

Test results are shown in the Ezurio Terminal as shown below.


Ezurio Terminal, Ver 6.9.0
CTS● DSR● DCD● RI● RTS ☑ DTR ☑ BRK ☐ ☑ Echo
Open [COM9:921600,N,8,1]
File to use
Use ☑ CTS ☐ DSR to Inhibit Tx
Tx Bitrate 0 bps
Rx Bitrate 633 kbps
View DH5
Range (m) 0
Overall Rx Avg 633 kbps
Rx Count 5179 KB
Tx Count 0 KB
Elapsed Time 1m:06s
642744 bps over 60sec (Click 'Clear' button to start)
kbps = 1000 bps KB = 1024 B
(c) Ezurio Ltd

Testing Results

During testing we found that disabling discoverability freed up significant speed as the radio wasn’t busy dedicating time slots to listening. As our labs are very busy with Bluetooth modules and Wi-Fi we chose to attach the radios with a wire simulating a 40dB link loss, effectively screening the devices from all other interference. We found a speed increase of 80Kbps when shielded in this way from noise, so the environment of operation is a very real consideration.

The BT900 is loaded with the '$autorun$.SPP.UART.bridge.incoming.sb' application available from the Laird GitHub account, here, with the define RX_CREDITS equal to 8 and varying MAX_FRAME_SIZE.

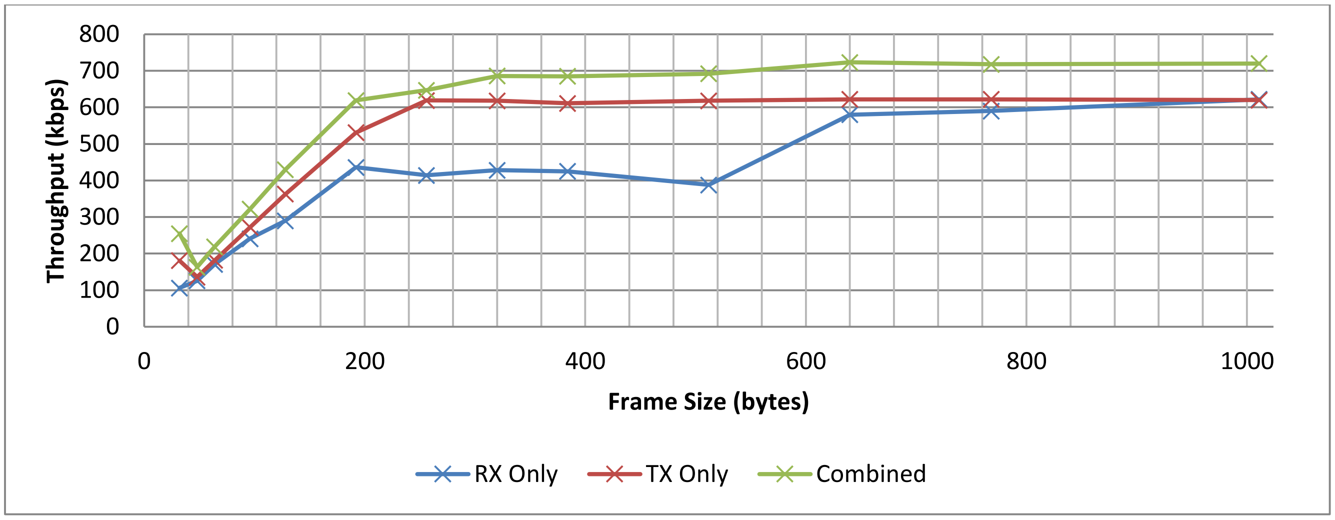


Figure 3: Bridge Mode throughput vs. frame size

CONSIDERATIONS

In addition, there are various other factors that affect the overall throughput:

Baud Rate

The UART baud rate limits the throughput and is a theoretical maximum of 80% of the baud rate if no parity and one stop bit are used. That theoretical maximum reduces to around 67% if parity is enabled along with two stop bits. In the latter case, four bits are required to convey eight bits.

Radio Utilization

At any time, up to three non-transient operations can be active. The radio can 1) be servicing ongoing connections, 2) be scanning for inquiries, and 3) be scanning for incoming connections. For the latter two, the scanning operation has a duty cycle; the worst case of 100% will have a major impact on the throughput because the single radio's time is being shared between connections and scanning tasks.

Radio Connection Quality

If the radio connection quality is bad and there are many packet retries, the throughput can drop to almost zero before the connection is automatically dropped. Having many Bluetooth devices in the vicinity will degrade performance as the potential for collisions increases, likewise for Wi-Fi as they cover the same spectrum.

REVISION HISTORY

Version	Date	Notes	Approver
1.0	4 Sep 2015	Initial Release	Jonathan Kaye
1.1	15 Sep 2015	Updated with Testing Procedure	Jonathan Kaye