BTM44X Enhanced Data Module – Throughput Analysis

BTM44x

Application Note *v2.0*

The BTM44x Enhanced Data 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 which sits on top of an underlying layer called L2CAP, which is packet based and, in turn, 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 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.

Given that RFCOMM, a streaming protocol, is sitting on top of a packet-based protocol, there is some logic between the streaming and packet-based layers. This includes a timer to ensure that, if data exists but does not fill an entire packet, it is still eventually sent. This raises latency and the unfilled packet creates non- optimal utilization of the baseband and hence less than optimal data throughput.

To enable optimization of latency versus data throughput, there is an adjustable parameter called 'RFCOMM framesize' which (range of 23 to 32767). This parameter is exposed via S Register 11 (or 9011 in AT mode).

This RFCOMM framesize is used by the data pump algorithm to transfer outgoing data. Analysis presented in this application note shows how throughput can vary in relation to the value set for RFCOMM framesize.

Further complicated, the module's limited memory (RAM) must be managed efficiently while connected; this is done by using CTS/RTS handshaking at the UART interface. Hence, although the Bluetooth specification allows the RFCOMM framesize to be up to 32767 bytes, in the BTM44x module it is limited to 4096 bytes.

Generally, a larger framesize improves throughput but raises latency. Given the module's RAM limit, analysis by the underlying Bluetooth stack vendor shows that little throughput is gained beyond a size of about 1000.

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 retries of packets, then the throughput can drop to almost zero before the connection is automatically dropped. The maximum throughput of 350 kbps is achieved during AT mode operation.

Figures 1-3 plot the actual throughput against the RFCOMM frame size for both AT and MP mode. They show that in MP mode, packet structure and inquiry/page scanning significantly impact throughput. Tests were performed at 460800 baud and show that in AT mode, the data throughput can be up to 350kbps.

In MP mode the UART host should optimize performance by sending data in the largest packets possible. The UART host should also set SReg4 and SReg5 = 0 to disable scanning. Tests show throughputs up to 325kbps.

Note: In MP mode, the host may bombard the module with the worst case scenario of three byte packets with just one data byte payload. In this case, if too many of these packets are sent and the framesize is large (about 64 and above), the module may reset and lose the connection. The module panics when it runs out of buffer memory and the watchdog kicks in. On the rare occasion that this happens, alleviate this issue by increasing SReg81. By default, this value is set to 30%. This signifies the percentage of heap blocks that must remain free before RTS is deasserted automatically

Ezurio testing shows that with a framesize larger than 64, sending a storm of three byte packets (with one byte payload) and the default value of 30% may panic the module into a reset. Setting SReg81 to 50% solves the problem.

Increasing this S Register has an impact on how many simultaneous SPP connections can be sustained.

Tune S Registers 7, 8, 9, 10, 11, 81 and MP packet sizes, when applicable, to ensure your desired throughput.

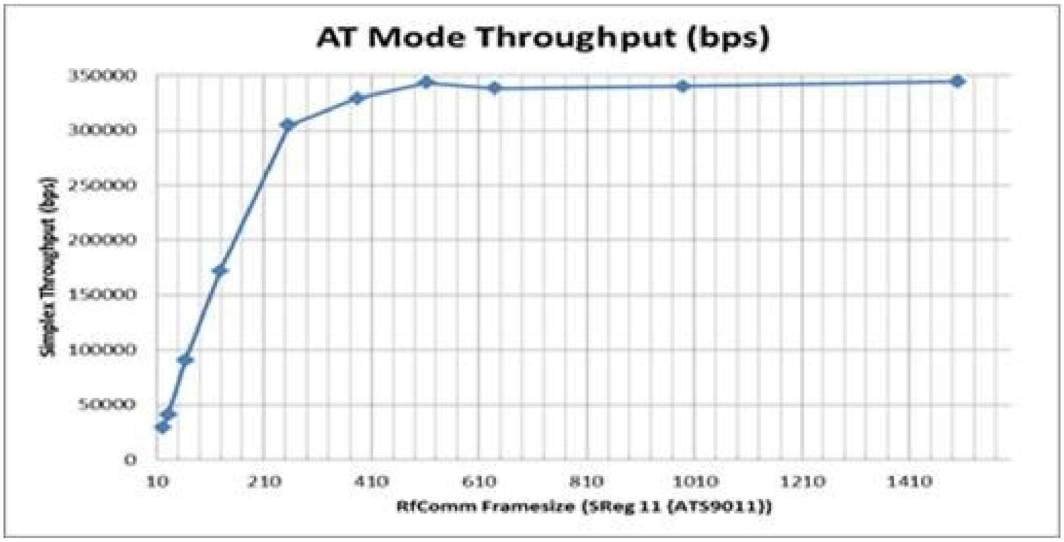

Figure 1: AT Mode Throughput

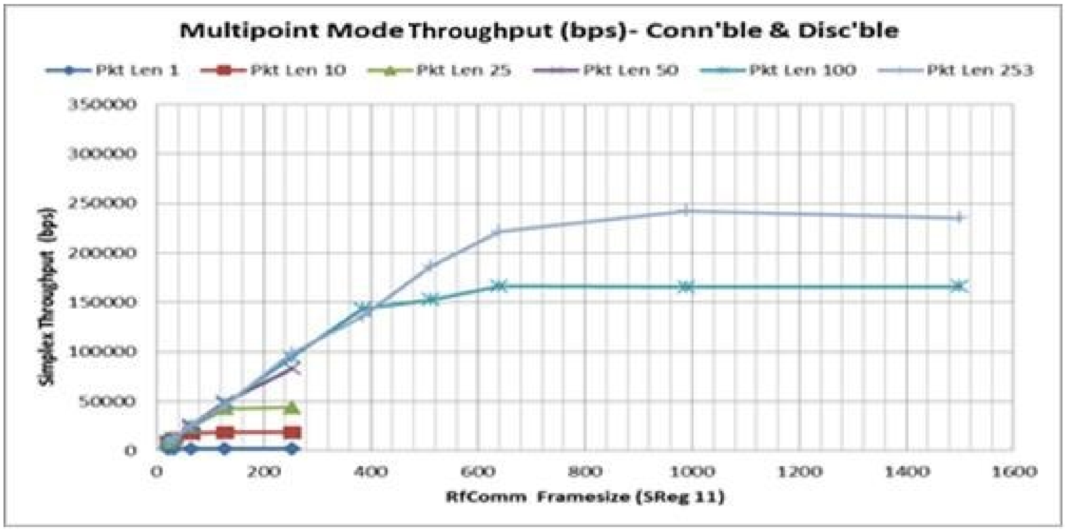

Figure 2: Multipoint Mode Throughput (Connectable & Discoverable

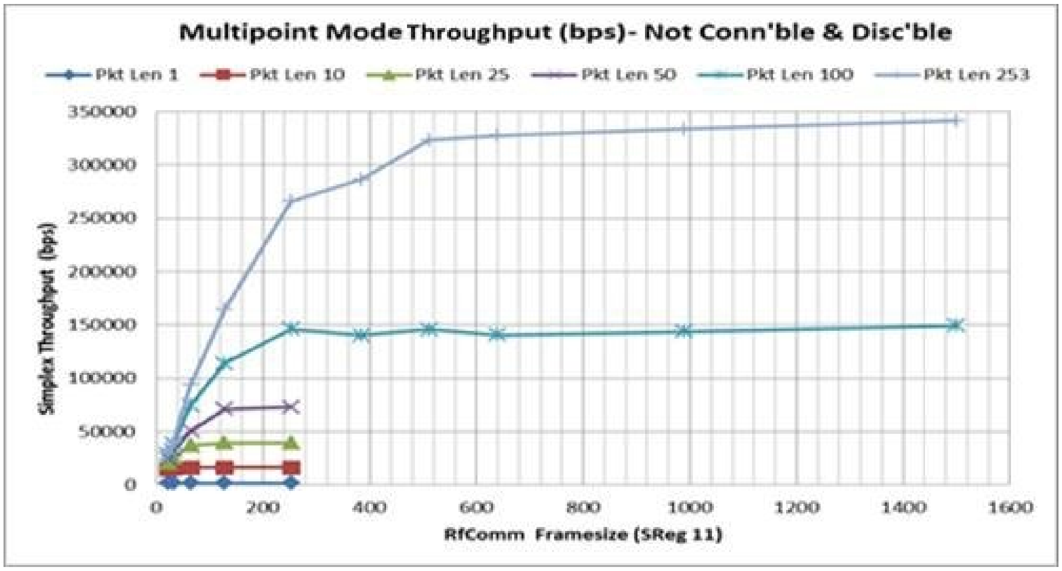

Figure 3: Multipoint Mode Throughput (Not Connectable & Discoverable)

Revision History

Version	Date	Notes	Contributor(s)	Approver
1.0	26 June 2012	Initial Release		Jonathan Kaye
1.1	19 June 2013	Grammatical edits		Sue White
1.2	22 Jan 2015	Added Revision History		Sue White
2.0	29 Apr 2025	Ezurio rebranding	Sue White	Dave Drogowski

Ezurio's products are subject to standard Terms & Conditions.