Achievable Data Rates on Simultaneously Connected Bluetooth Low Energy Devices

Mirco Gysin, Andreas Müller, Andreas Rüst
Zurich University of Applied Science (ZHAW)
Institute of Embedded Systems (InES)
Winterthur, Switzerland
andreas.ruest@zhaw.ch

***Abstract* — Bluetooth Low Energy (BLE) has been optimized for low power and targets low data rate applications. However, there are applications where several peripherals are simultaneously connected to a central device, with the requirement of achieving moderate data rates for each connection. Available hardware and software stacks place limitations on the number of simultaneous connections as well as on the achievable data rates. This paper presents measurement results for different scenarios. On the central device side the explored options cover tablets based on Android and iOS as well as dongles that can be connected through a USB port. On the peripheral side several selected hardware set-ups (chips and modules) are measured and compared. Analysis of captured packets provides insight into why the reached data rates are often significantly lower than theoretically expected.**

Keywords — Bluetooth Smart, Bluetooth Low Energy, speed test, achievable data rates

I. INTRODUCTION

Applications such as the data logger in [1] would like to exploit moderate BLE data rates to read out memory areas from a peripheral to a central device. Based on the Bluetooth Low Energy standard [2] one could expect that the payload data rate from a peripheral to a central device could be as high as 230 kbit/s [3]. However when determining the maximum achievable data rate in a specific implementation, several additional limitations have to be considered. Depending on the implementation, hardware resource constraints and software performance restrictions can significantly lower the actually achieved data rate compared to the expected rate. Moreover the situation is exacerbated if several connections with the same central device are being simultaneously sustained.

This paper is structured accordingly. We begin by describing some theoretical background and calculations of payload bandwidths based on this theoretical understanding. We continue by describing our measurement setup and in the third section we list the devices upon which we perform our tests. In the forth section we present the results of our measurements and in the fifth section an analysis of the data of captured packets to analyze these data rates. We conclude with appropriate conclusions and suggestions for further work.

II. THEORETICAL BACKGROUND

With regard to the Link Layer, the BLE controller in a peripheral takes on the role of a slave, whereas the controller in a central device takes on the role of a master. The communication between the two devices is organized in connection events. Each connection event is started by a packet from the master to the slave followed by a packet sent in the opposite direction. Several further exchanges from the master to the slave and back to the master may follow. If one of the devices fails to uphold the sequence this will cause the connection event to terminate. The time period called the connection interval will define when the master will start the next connection event. Fig. 1 shows the data exchange between master and slave on the Link Layer.

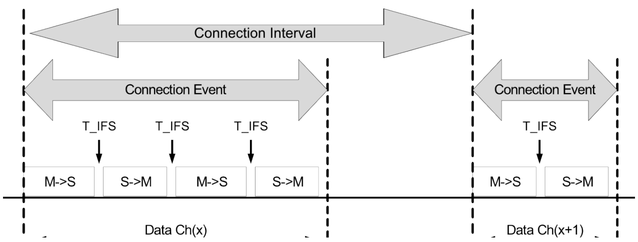

Fig. 1. Connection events and connection interval in [2].

The value of the connection interval is specific to each connection. The slave can request an update of the connection interval and propose a desired time range based on its requirements. The master may accept the request and choose an appropriate time period from the suggested range as a new connection interval.

The number of packet exchanges can be controlled by either the central device or the peripheral device. This can be used for flow control e.g. either due to lack of memory space or to give other applications a fair chance to transmit.

To push our payload data from the peripheral (slave) to the central (master) device we use notification messages in the Attribute Protocol (ATT) of BLE. They do not require a response or confirmation and can be sent by the peripheral at any time. Considering the limitations and overhead octets of the different protocol layers a packet/notification from the slave to the master has room for at most 20 octets of data payload.

In the tested implementations there are limitations on the minimum length of a connection interval as well as the number of notifications per connection event. The limitations can originate from the central device or from the peripherals, but in most cases it is the central device that limits these factors.

Based on the length of the connection interval (l_{CI}) and the number of notifications within a connection event (n_{CE}) we can calculate the expected maximum data rate from a peripheral to a central device as:

$$data\_rate\ (l_{CI},\ n_{CE}) = (20 * 8 * n_{CE})\ /\ l_{CI}\ \ [kbit/s] \qquad (1)$$

TABLE I. shows calculated data rates based on (1).

TABLE I. DATA RATES [KBIT/S] BASED ON (1)

l_{CI} [ms]	***Notifications per connection event (n_{CE})***						
	1	***2***	***3***	***4***	***5***	***6***	***7***
7.5	21.3	42.7	64.0	85.3	106.7	128.0	149.3
12.5	12.8	25.6	38.4	51.2	64.0	76.8	89.6
15.0	10.7	21.4	32.3	42.7	53.3	64.0	74.7
30.0	5.3	10.7	16.0	21.3	26.7	32.0	37.3
37.5	4.3	8.5	12.8	17.1	21.3	25.6	29.9

III. MEASUREMENT SETUP

The presented data rates have been measured with a dedicated application on the central device using a proprietary profile.

A. Profile

TABLE II. shows the applied profile and its characteristics.

TABLE II. CHARACTERISTICS OF SPEEDTEST PROFILE

Characteristic	**Description**	**Access**
CONNECTION_INTERVAL	Current value of the connection interval	Notify
TESTVALUE	Data transfer used to measure the data rate	Notify
COMMAND	Start, abort, terminate	Write
DESIRED_CONNECTION_INTERVAL	Minimum and maximum time of the connection interval (in units of 1.25 ms)	Write
BYTES_PER_NOTIFICATION	Number of payload octets per notification	Write
NROFPACKETS	Number of packets (notifications) sent by the peripheral through TESTVALUE.	Write

B. Measurement Procedure

The following steps describe the measurement procedure for each peripheral as seen from the central device:

1. Scan for peripheral advertising the speed test service and establish connection.
2. Discover the handles for the attributes of the speed test service.
3. Register/enable notifications for CONNECTION_INTERVAL and TESTVALUE.
4. Write BYTES_PER_NOTIFICATION and NROFPACKETS.
5. Write DESIRED_CONNECTION_INTERVAL.
6. Wait until a notification CONNECTION_INTERVAL is received from all peripherals.
7. Start timer and write the start command to COMMAND.
8. Peripheral delivers notifications with sequence numbers through TESTVALUE.
9. Wait until the notification with the lowest sequence number has been received, stop timer and calculate achieved data rate.

C. *Measurement Conditions*

The tests have been carried out with 1000 or more notifications sent from each peripheral. Each notification contains a payload of 20 octets, consisting of a two octet sequence number and the other octets set to zero.

As the achieved data rates of the individual peripherals vary, the peripherals will not complete their transmissions at the same point in time. However in our measurement procedure the peripherals remain connected even after completing their transmission. Analysis of recorded packets using a sniffer shows that the achieved data rates of ongoing transmissions remain constant even after other peripherals have completed their transmission. I.e. the unused bandwidth of peripherals completing early is not redistributed to other peripherals.

Packets that require retransmission are reflected in the results as they increase the measured time to transmit all the packets.

IV. TESTED DEVICES

A. *Central Device Side – Tablets*

On the central device side the following tablets have been evaluated:

- iPad 3 running iOS version 7.0.4
  The speed test application has been implemented in Objective-C.
- Nexus 7, generation 2013 running Android 4.4 KitKat. The speed test application has been implemented in Java.

The Wi-Fi and cellular functions have been turned off during the test. As far as possible no other applications were running on the device.

In both Android and iOS the actually applied connection interval cannot be accessed by the application. To overcome this limitation the characteristic CONNECTION_INTERVAL has been included in the speed test profile. This allows the application to read the connection interval value established during the test.

iOS does not allow full control of the connection interval. The application programmer can only select a desired range. The minimum value may be no lower than 20 ms and the range must have an extent of at least 20 ms. iOS will then choose the connection interval within the selected range.

Android offers the application full control over the connection interval. As a result a connection interval as low as 7.5 ms can be selected. This allows significantly higher data rates compared to the iOS devices.

B. *Central Device Side – Dongles*

On the central device side the following dongles have been evaluated. The speed test application has been implemented in Python 3.4 and runs on a laptop with Ubuntu 14.04.

- TI dongle [4]
  CC2540Dongle, Rev. 1.0.1
  with firmware
  CC2540_USBdongle_HostTestRelease_All.hex
  Version 1.4.0
- Bluegiga dongle [5]
  BLED112 Bluetooth Smart Dongle, version 3
  with firmware 1.2 build 85

C. *Peripheral Side*

On the peripheral side the following devices have been evaluated:

- TI Keyfob [4]
  CC2540Keyfob, Rev 1.2.1
  with stack version 1.4.0
- Nordic
  PCA10001 V2.2.0
  with stack Softdevice s110 NRF51822 V7.0.0
- ST BlueNRG [6]
  STEVAL-IDB002V1 with stack DK-1.5.0.0

With the ST BlueNRG device only up to three simultaneous connections have been measured due to a limited number of available devices.

V. MEASUREMENT RESULTS

This section presents selected measurements of achieved data rates in cases where a single central device maintains simultaneous Bluetooth Low Energy connections with several peripheral devices.

A. *Tablets as Central Devices*

1) Android – Nexus 7

TABLE IV. summarizes the achieved data rates between Nexus 7 and different numbers and types of peripheral devices. The maximum data rates peak at around 60 kbit/s. This correlates to the calculated data rate of 64 kbit/s for a connection interval of 7.5 ms and three notifications per connection event. The variation between the different types of peripherals is negligible.

With two simultaneous connections both peripherals can sustain data rates around 60 kbit/s. With three simultaneous connections we observe an asymmetrical distribution of the available bandwidth. The first peripheral remains around 60 kbit/s whereas the others only achieve approximate data rates of 40 kbit/s and 24 kbit/s respectively.

With up to three simultaneous connections it is possible to use a connection interval of 7.5 ms. The use of more than three simultaneous connections requires an increase of the connection interval to 12.5 ms. Otherwise it is not possible to maintain stable connections in the measurement set-up.

In the case of four and five simultaneous connections a very unbalanced partitioning of the data rate can be observed. The first peripheral receives a major share of around 60 kbit/s whereas the others only achieve data rates around 12 kbit/s.

2) iOS – iPad 3

TABLE V. summarizes the maximum achieved data rates between the iPad 3 and different numbers and types of peripheral devices.

The achieved data rates are significantly lower than on Android. A peripheral can just barely reach 30 kbit/s. This is only about half the data rate of a peripheral in an Android system. However in contrast to the Android system the data rate achieved by a peripheral is independent of the number of simultaneously connected devices. Even with five peripherals each peripheral reaches the same data rate that it reaches if it is exclusively connected to the central device. The Nordic device achieves slightly lower data rates than the other devices.

Interestingly enough lowering the connection interval to 30.0 ms reduces the achieved data rates. TABLE VI. presents the measured values. The effect is caused by the central device allowing fewer notifications per connection event.

B. Dongles as Central Devices

TABLE VII. presents the measurement results for the Bluegiga dongle. The measurement results for the TI dongle can be found in TABLE VIII. All the measurement results in these tables use uniform connection intervals for all peripherals.

The Bluegiga dongle achieves peak data rates for an individual peripheral around 80 kbit/s whereas the peak data rates on the TI dongle reach only about half of that value, i.e. 40 kbit/s.

In some cases it has been observed that choosing non-uniform connection intervals among the peripherals yielded slightly higher aggregate data rates. However such specific choices can rarely be used in applications.

The TI dongle can support a maximum of three simultaneous connections. On the Bluegiga dongle it is possible to go up to eight simultaneous connections. However this requires writing a special firmware configuration to the dongle and increases the minimum time period for a connection interval.

The TI dongle loses connection with a single peripheral if a connection interval smaller than 16.25 ms is applied.

VI. Analysis of Results

A sniffer (TI CC2540 USB dongle with software „GENPacketSniffer 2.18.1.0“) has been used to record the packets exchanged between the central device and the peripherals. The recorded data is analyzed in Matlab and plotted. This yields further insight into the achieved data rates.

A. Nexus 7 with Three Peripherals

The following figures plot the recorded data for the measurement of the Nexus 7 with three peripherals of type ST BlueNRG.

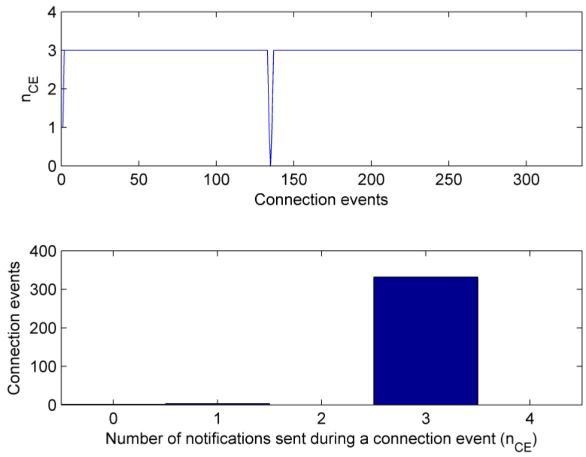

Fig. 2. Notifications per connection event (n_{CE}) for peripheral ST BlueNRG achieving highest data rate to Nexus 7.

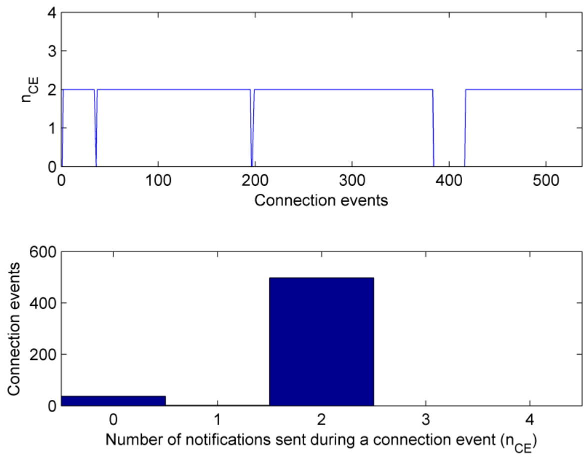

Fig. 3. Notifications per connection event (n_{CE}) for peripheral ST BlueNRG achieving intermediate data rate to Nexus 7.

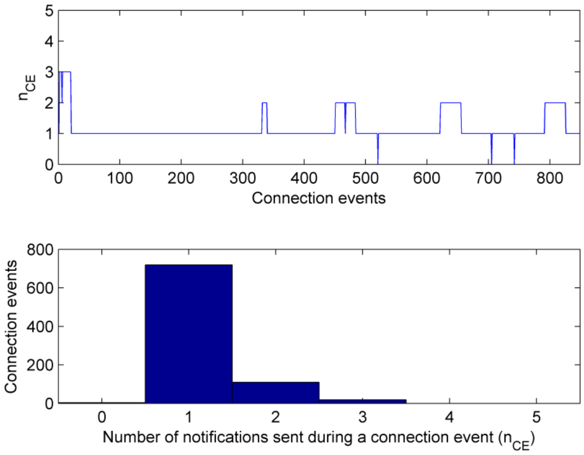

Fig. 4. Notifications per connection event (n_{CE}) for peripheral ST BlueNRG achieving lowest data rate to Nexus 7.

Fig. 2 shows that the peripheral achieving the highest data rate sustains three notifications per connection event. The only exception occurs after about a third of the time where a single connection event with only one notification occurs. The exception is probably caused by a lost packet due to a transmission error.

The peripheral achieving the intermediate data rate reaches an average of almost two notifications per connection event (Fig. 3) whereas the peripheral with the lowest data rate only achieves slightly above one notification per connection event (Fig. 4). Consequentially the peripherals with fewer notifications per connection event require a higher number of connection events to transmit all their data. The data is summarized in TABLE III.

TABLE III. SUMMARY OF RECORDED RESULTS

	Peripheral		
	1	***2***	***3***
achieved data rate [kbit/s]	63.0	38.4	24.6
number of connection events	336	537	849
average # of notifications per connection event	2.97	1.86	1.17

Normally the peripheral which is the first to receive the start command achieves the highest data rate.

B. iPad 3 with Peripherals

Based on (1) a reduction of the length of a connection interval (l_{CI}) should lead to an increase of the achieved data rate. However if a shorter connection interval is used a central device may allow a lower number of notifications per connection event. In some cases this may overcompensate for a shorter connection interval and result in lower data rates. This effect can be seen when comparing the iPad 3 results in TABLE V. and TABLE VI.

As can be seen in the sniffer data shown in Fig. 5 the iPad 3 will allow seven notifications per connection event in case a connection interval of 37.5 ms is used. Using TABLE I. this should result in a data rate of 29.9 kbit/s. This is in accordance with the measured data rates for the TI Keyfob and the ST BlueNRG.

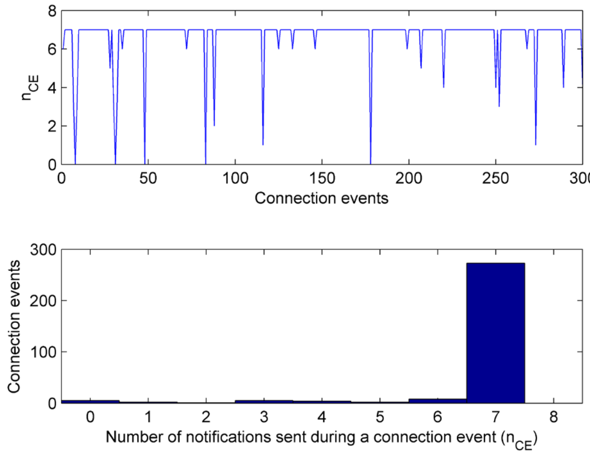

Fig. 5. Notifications for a single ST BlueNRG connected to iPad 3 with l_{CI} = 37.5 ms.

If the connection interval is lowered to 30 ms the number of notifications per connection event is reduced to only four. This is clearly shown in Fig. 6. Using (1) this results in an expected data rate of 21.3 kbit/s. The measured values in TABLE VI. are only slightly below the calculated values.

Interestingly enough for a connection interval of 37.5 ms the Nordic peripheral achieves a lower data rate than the other devices: 25.5 kbit/s versus 29.6 kbit/s. Inspection of the sniffer data shows that in contrast to the other devices the Nordic device sets the 'More Data' flag to zero after six notifications. This informs the central device that it wants to end the connection event. So it seems that the Nordic device allows at most six notifications per connection event. The expected data rate for a connection interval of 37.5 ms and six notifications is 25.6 kbit/s. This again is very close to the measured values.

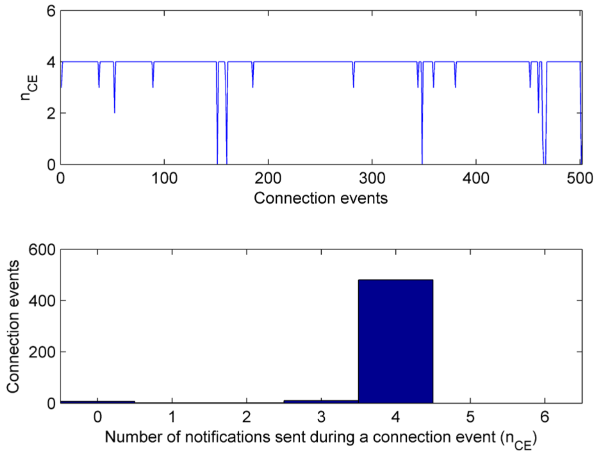

Fig. 6. Notifications for a single ST BlueNRG connected to iPad 3 with $l_{CI} = 30$ ms.

C. *Bluegiga Dongle*

The three different types of peripherals achieve different peak data rates with the Bluegiga dongle. Fig. 7 shows that the TI Keyfob often reaches five notifications per connection event. On the other hand a considerable amount of connection events contain no notifications at all resulting in a lower achieved data rate.

Looking at the captured packet data shows that the Bluegiga dongle reports "Unexpected Next Sequence Number" events which lead to a termination of the connection event and a retry on the next connection event.

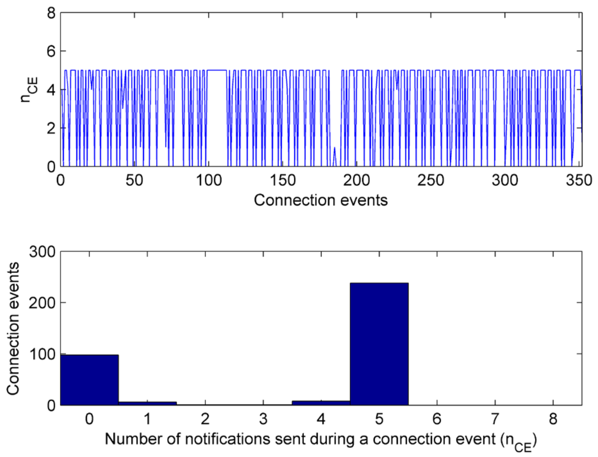

Fig. 7. Single TI Keyfob connected to Bluegiga dongle ($l_{CI} = 7.5$ ms).

The ST BlueNRG device reaches between three and four notifications per connection event (Fig. 8). The ST BlueNRG ends the connection events early by setting the 'More Data' flag to zero after three or four notifications. The probable reason for this is flow control.

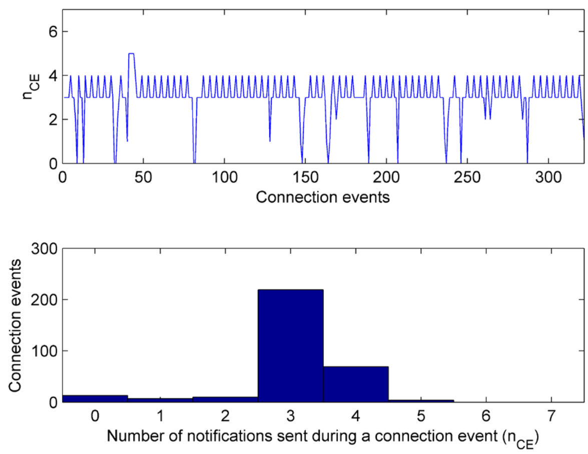

Fig. 8. Single ST BlueNRG connected to Bluegiga dongle ($l_{CI} = 7.5$ ms).

Finally Fig. 9 indicates that the Nordic device is able to sustain an average of almost five notifications per connection event. Examining the captured data uncovers several "Unexpected Next Sequence Number" events but it seems that the Nordic device recovers faster from such conditions than the TI Keyfob and has much fewer connection events with zero notifications. Consequently the achieved data rate is significantly higher.

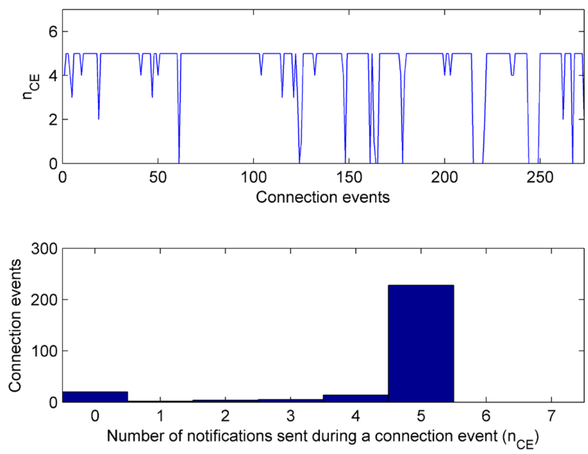

Fig. 9. Single Nordic connected to Bluegiga dongle ($l_{CI} = 7.5$ ms).

VII. CONCLUSIONS

This paper presents measurement results for achieved data rates when simultaneously transmitting data from several BLE peripherals to a central device. While the Nexus 7 under Android allows high peak data rates for individual peripherals, the iPad 3 under iOS allows a very uniform and scalable distribution of data rates even for five peripherals. Furthermore analysis using a sniffer confirms that the achieved data rates indeed depend on the size of the used connection interval as well as on the number of notifications per connection event. In the majority of cases it is the central device that is limiting the achievable data rates. Other mobile platforms like Windows Phone will be studied in future work.

References

[1] MSR145WD Wireless Data Logger with BLE, Display & MSR SmartCloud, Datasheet, MSR Electronics GmbH, 2013

[2] Specification of the Bluetooth System Version 4.1, Bluetooth SIG, December 2013

[3] A. Rüst, M. Gysin, A. Müller, and M. Würms, “Simultaneously connecting devices through Bluetooth Smart”, Embedded World Conference, February 2014

[4] Bluetooth Low Energy; CC2540/41 Mini Development Kit; User’s Guide. Document Number: SWRU270C; Document Version: 1.2; Development Kit Part Number: CC2540DK-MINI; Texas Instruments, January 2013

[5] BLED112, Data Sheet, Version 1.1, Bluegiga Technologies, April 2014

[6] UM1686; User manual; BlueNRG development kits Document Number: DocID025464 Rev 2, STMicroelectronics, April 2014

TABLE IV. ACHIEVED DATA RATES [KBIT/S] ON NEXUS 7 WITH N PERIPHERALS

	TI Keyfob					ST BlueNRG			Nordic				
	N=1	*N=2*	*N=3*	*N=4*	*N=5*	*N=1*	*N=2*	*N=3*	*N=1*	*N=2*	*N=3*	*N=4*	*N=5*
l_{CI} [ms]	7.5	7.5	7.5	12.5	12.5	7.5	7.5	7.5	7.5	7.5	7.5	12.5	12.5
Nexus 7 [kbit/s]	59.2	60.1	59.4	46.0	61.4	61.5	63.6	63.7	63.0	63.3	63.5	63.3	63.3
	−	59.0	38.5	12.0	13.0	−	61.5	38.1	−	63.5	40.2	13.4	12.1
	−	−	23.5	12.9	11.8	−	−	24.4	−	−	23.0	13.4	13.5
	−	−	−	12.1	11.1	−	−	−	−	−	−	12.1	12.1
	−	−	−	−	12.0	−	−	−	−	−	−	−	12.1
aggregate [kbit/s]	59.2	119.1	121.4	83.0	109.8	61.5	125.1	126.2	63.0	126.8	126.7	102.2	113.1

TABLE V. ACHIEVED DATA RATES ON IPAD 3 WITH N PERIPHERALS AND CI = 37.5 MS

	TI Keyfob					ST BlueNRG			Nordic				
	N=1	*N=2*	*N=3*	*N=4*	*N=5*	*N=1*	*N=2*	*N=3*	*N=1*	*N=2*	*N=3*	*N=4*	*N=5*
l_{CI} [ms]	37.5	37.5	37.5	37.5	37.5	37.5	37.5	37.5	37.5	37.5	37.5	37.5	37.5
iPad 3 [kbit/s]	29.8	29.7	29.6	29.5	29.6	29.7	29.4	29.6	25.5	25.4	25.5	25.3	25.4
	−	29.6	29.7	29.7	29.6	−	29.7	29.4	−	25.5	25.4	25.4	25.5
	−	−	29.6	29.5	29.7	−	−	29.6	−	−	25.5	25.5	25.3
	−	−	−	29.7	29.4	−	−	−	−	−	−	25.5	25.4
	−	−	−	−	29.6	−	−	−	−	−	−	−	25.3
aggregate [kbit/s]	29.8	59.3	88.9	118.4	147.9	29.7	59.1	88.6	25.5	50.9	76.4	101.7	126.9

TABLE VI. ACHIEVED DATA RATES ON IPAD 3 WITH N PERIPHERALS AND CI = 30.0 MS

	TI Keyfob					**ST BlueNRG**			**Nordic**				
	N=1	***N=2***	***N=3***	***N=4***	***N=5***	***N=1***	***N=2***	***N=3***	***N=1***	***N=2***	***N=3***	***N=4***	***N=5***
l_{CI} [ms]	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
iPad 3 [kbit/s]	21.1	21.2	21.2	21.3	21.2	21.2	21.3	21.3	21.1	21.3	21.1	21.1	21.1
	−	21.2	21.2	21.2	21.0	−	21.2	21.2	−	21.2	21.2	21.3	21.1
	−	−	21.2	21.1	21.3	−	−	21.2	−	−	21.2	21.3	21.1
	−	−	−	21.1	21.1	−	−	−	−	−	−	21.2	20.9
	−	−	−	−	21.2	−	−	−	−	−	−	−	21.0
aggregate [kbit/s]	21.1	42.4	63.6	84.7	105.8	21.1	42.5	63.7	21.1	42.5	63.5	84.9	105.2

TABLE VII. ACHIEVED DATA RATES ON BLUEGIGA DONGLE WITH N PERIPHERALS

	TI Keyfob			**ST BlueNRG**			**Nordic**		
	N=1	***N=2***	***N=3***	***N=1***	***N=2***	***N=3***	***N=1***	***N=2***	***N=3***
l_{CI} [ms]	7.5	7.5	12.5	7.5	7.5	7.5	7.5	7.5	7.5
Bluegiga Dongle [kbit/s]	64.2	62.1	28.1	69.0	68.4	21.1	81.8	84.4	21.2
	−	21.1	7.1	−	21.3	21.3	−	21.4	21.0
	−	−	28.2	−	−	21.3	−	−	21.2
aggregate [kbit/s]	64.2	124.2	63.4	69.0	89.7	63.7	81.8	105.8	63.4

TABLE VIII. ACHIEVED DATA RATES ON TI DONGLE WITH N PERIPHERALS

	TI Keyfob			**ST BlueNRG**			**Nordic**		
	N=1	***N=2***	***N=3***	***N=1***	***N=2***	***N=3***	***N=1***	***N=2***	***N=3***
l_{CI} [ms]	16.25	10.0	20.0	16.25	10.0	20.0	16.25	10.0	20.0
TI Dongle [kbit/s]	38.9	15.8	8.0	39.4	16.0	7.9	39.4	15.9	8.0
	−	15.8	8.0	−	15.9	8.0	−	15.9	7.9
	−	−	8.0	−	−	8.0	−	−	7.9
aggregate [kbit/s]	38.9	31.6	24.0	39.4	31.9	23.9	39.4	31.8	23.8