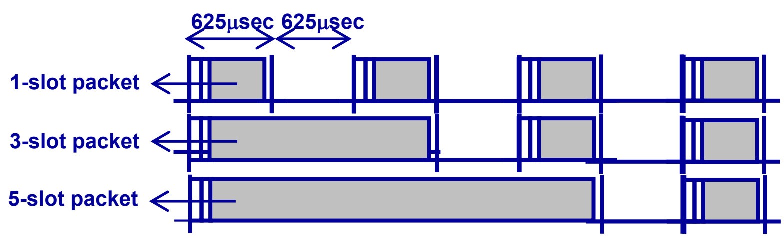
EECE 451 – Spring 2015 – 2016 **Solutions**

Assignment 1 **February 22, 2016**

Problem 1. Consider the table below that shows data about Bluetooth packets, and the accompanying diagram that shows the duration of each packet type (a 625μsec always separates two consecutive packets). Note that the rightmost three columns give the rate of transmission, and that (obviously) a Bluetooth packet contains header information.

Type	User Payload (bytes)	FEC	Symmetric (Kbps)	Assymetric (Kbps)	Assymetric(kbps)
DM1	0-17	Yes	108.0	108.0	108.0
DH1	0-27	No	172.8	172.8	172.8
DM3	0-121	Yes	256.0	384.0	54.4
DH3	0-183	No	384.0	576.0	86.4
DM5	0-224	Yes	286.7	477.8	36.3
DH5	0-339	No	432.6	721.0	57.6
HV1	0-10	Yes	64.0	-	-
HV3	0-20	Yes	128.0	-	-
HV5	0-30	No	192.0	-	-



In a piconet that consists of one Master, and four slaves, Slave S2 needs to send Slave S3 a 600KB MB audio file, while S3 needs to send Slave S4 a 768KB image file. All peers in the piconet have detected that the wireless channel is prone to errors (i.e., noisy).

1. The symmetric bitrate, shown in the table, is only applicable if the Master and one of the slaves are exchanging data in parallel
2. The Master continuously polls every slave in the Piconet. After slave S_i is polled, if it has data to send, it sends one data frame to the Master, else it sends a NAK.
3. If the Master has a data packet to send to slave S_i, and S_i has nothing to send, it just replies with an ACK; else S_i replies by sending its data packet
4. The POLL, NACK, and ACK packets have the same size as that of a DH1.

The channel is noisy and we have 5 total devices so we use DM3 packets. Each DM3 packet can carry up to 121 bytes (see table).

Question 1

S2 is sending a 600 KB file and S3 is sending a 768 KB file.

The number of packets being sent by S2 is $\frac{600\times2^{10}}{121} = 5078$ packets.

The number of packets being sent by S3 is $\frac{768\times2^{10}}{121} = 6500$ packets.

Each DM3 packet (data) takes 3×625µs + 1×625µs = 2.5 ms

Each DH1 packet (standalone poll, Ack, or Nak) takes 1×625µs + 1×625µs = 1.25 ms

The first cycle is as follows:

1. M polls S1. S1 replies with a NAK. => 2x1.25 = 2.5 ms.
2. M polls S2. S2 replies with Data (to be forwarded to S3). => 1.25 + 2.5 = 3.75 ms.
3. M sends S3 data (from S2) and a piggybacked poll. S3 replies with a data packet (to be forwarded to S4). => 2.5 + 2.5 = 5 ms.
4. M sends S4 data (from S3) and a piggybacked poll. S4 replies with an ACK.
   => 2.5 + 1.25 = 3.75 ms.
5. M polls S1. S1 replies with a NAK. =>2.5 ms
6. M sends S2 ACK (from S3) and a piggybacked poll. S2 replies with Data (to be forwarded to S3). => 1.25 + 2.5 = 3.75 ms
7. M sends S3 data (from S2), a piggybacked ACK (from S4), and a piggybacked poll. S3 replies with Data (to be forwarded to S4). => 2.5 + 2.5 = 5 ms
8. M sends S4 data (from S3) and a piggybacked poll. S4 replies with an ACK.
   => 2.5 + 1.25 = 3.75 ms
9. Steps 5 through 8 repeat until the audio file from S2 to S3 is completely sent.

S2 to finish transmitting the audio file:

Account for the initial steps (1 through 4) => first packet sent

+ loop (steps 5 through 8) => 5076 iterations to send 5076 packets

but last iteration stops at Step 6 and does not include the last idle slot => the last packet

=> Total of 5078 packets.

$$\begin{aligned}Transmission\ Time &= (2.5 + 3.75 + 5 + 3.75) \\ &\quad + 5076 * (2.5 + 3.75 + 5 + 3.75) \\ &\quad + (2.5 + 3.75) - 625\mu s \\ &= 76.160625\ seconds\end{aligned}$$

S3 to finish receiving the audio file:

$$= 76.160625\ seconds + 625\mu s + 2.5\ ms = 76.16375\ seconds.$$

After S3 has finished receiving its data, S3 still has to finish sending its file to S4 (its file is larger). So the loop of Steps 5 through 8 becomes:

5. M polls S1. S1 replies with a NAK. => 2.5 ms
6. M sends S2 a poll. S2 replies with a NAK. => 2.5 ms
7. M sends S3 an ACK (from S4) and a piggybacked poll. S3 replies with Data (to be forwarded to S4). => 3.75 ms
8. M sends S4 data (from S3) and a piggybacked poll. S4 replies with an ACK. => 3.75 ms
9. Steps 5 through 8 repeat until the image file from S3 to S4 is completely sent.

S4 to finish receiving the image file:

Remaining number of iterations/packets $= 6500 - 5078 = 1422\ packets.$

⇨ $S4\ reception\ time = 76.16375\ sec + (2.5 + 3.75)\text{ms} + 1422 * (2.5 + 2.5 + 3.75 + 3.75)ms - 1.25ms - 625\mu s$
$= 93.943125\ seconds$

Question 2

The Master has a ½ MB video file that it wants to send to all slaves, and the slaves have nothing to send, how long will it take for

M is sending a 0.5 MB, so the number of packets being sent by M is $\frac{0.5\times 2^{20}}{121} =$ 4333 packets.

The list of steps becomes:

1. M sends S1 data and piggybacks a poll. S1 replies with an ACK.
2. M sends S2 data and piggybacks a poll. S2 replies with an ACK.
3. M sends S3 data and piggybacks a poll. S3 replies with an ACK.
4. M sends S4 data and piggybacks a poll. S4 replies with an ACK.

Each of the steps takes 3.75 ms.

The Master to finish transmitting data

$$4333 * 4 * 3.75\ ms - 1.25\ ms\ (subtract\ the\ last\ ACK) - 625\mu\ (and\ the\ last\ idle\ slot)$$

$$= 64.993125\ seconds$$

The first slave to get the complete video file (i.e., earliest slave)

$$4332 * 4 * 3.75ms\ +\ 3.75ms - 1.25ms - 625\mu s\ =\ 64.981875\ seconds$$

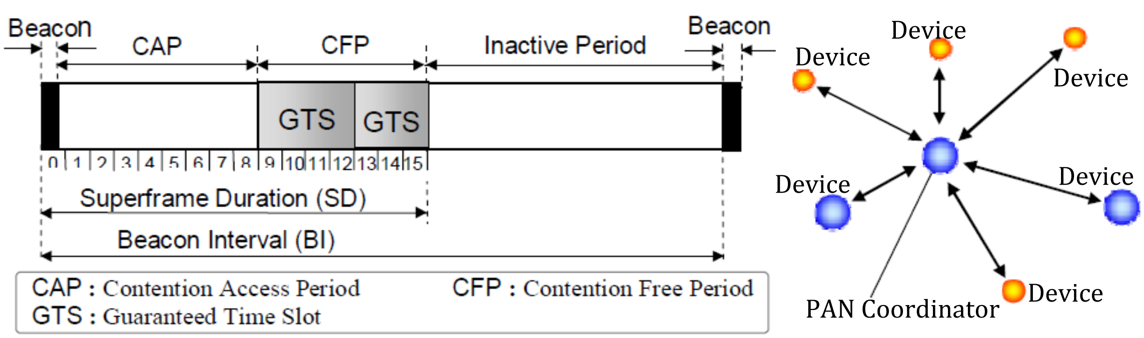
All slaves get the file

$$4333 * 4 * 3.75 - 1.25ms - 625\mu s = 64.993125\ seconds$$

Problem 2. The 802.15.4 standard supports device operations in CSMA/CA and in TDMA access modes. The TDMA mode of operation is supported via the Guaranteed Time Slot (GTS) feature. In a PAN, a device that wants to send to another device initiates the transmission by using CSMA/CA (in the CAP period) *if the data is not time-sensitive*, or requests from the PAN Coordinator GTS slots *if the data is time-sensitive*. The request of GTS slots takes place during the CAP period. The PAN coordinator assigns the GTS (in the CFP period) to devices on a first–come–first–served basis, in response to requests from devices. The coordinator indicates the assigned GTS in the beacon, so the concerned devices know when to transmit. Note that in 802.15.4, the Coordinator does not poll the devices in the PAN (unlike Bluetooth).

Now, consider the following data/information:

1. The Superframe is divided into 16 slots of time, plus an inactive period. The first one is occupied by the beacon.
2. Each slot is 5 milliseconds long.
3. The inactive period is 80 milliseconds long
4. The max MAC frame size is 118 bytes, out of which 9 bytes are overhead (i.e., header)
5. The data rate at the MAC layer is 250 Kbps.
6. As with any standard, a device can receive at *any time* while not transmitting



In the above figure, on the right, we show a star-topology PAN that includes 3 FFDs (including the PAN Coordinator) and 4 RFDs. Device 1, wants to send Device 6 a 4KB worth of data, while Device 5 wants to send Device 3 a 3KB data file. Both Device 1 and Device 5 have requested GTS to transmit their data. The Coordinator granted Device 1 a 4-slot GTS and Device 5 a 3-slot GTS, as illustrated above.

Device one has GTS duration of 4*tslot = 4*5ms = 0.02 s every superframe.

Device 5 has GTS duration of 3*tslot = 3*5ms = 0.015 s every superframe.

Time to send one MAC frame is $\frac{\textbf{bits in a MAC frame}}{\textbf{MAC Datarate}} = \frac{118 bytes*8}{250*1000}$ **= 3.8 ms.**

Time of one super frame = 16 slots * 5ms + 80ms = 0.16 s.

1. How long will it take Device 1 to completely transmit its data?

We can send $\frac{0.02s}{3.8ms}$ = 5.2632 or 5 MAC frames in one superframe from Device 1.

The number of MAC frames needed for Device 1 to send the whole file =
$\frac{\text{File Size}}{\text{Mac Payload Size}} = \frac{4\text{ KB}}{(118-9)\text{B}}$ **= 37.578 or 38 MAC frames.**

Number of superframes needed = $\frac{38}{5}$ = 7.6 => 8 superframes.

We need 7 *full* superframes, and until the end of the GTS of Dev1 in the 8th superframe.

Time to completely transmit data =7* 160ms+ 13 * 5ms = 1.185 s.

2. How long will it take for Device 6 to receive the entire data sent by Device 1?

7*160 + 13*5 + 4*5 = 1.205s.

3. How long will it take Device 5 to completely transmit its data?

We can send $\frac{0.015s}{3.8ms}$ = 3.9 or 3 MAC frames in one superframe from Device 5.

The number of MAC frames needed for Device 5 to send the whole file =
$\frac{\text{File Size}}{\text{Mac Payload Size}} = \frac{3\text{ KB}}{(118-9)\text{B}}$ **= 28.1835 or 29 MAC frames.**

Number of superframes needed = $\frac{29}{3}$ = 9.6667

Time to completely transmit data = 9*160ms + (1+8+4+3)*5ms = 1.52 s