ET4394 Wireless Networking - Wisent assignment

Sjors Nijhuis 4157478
Kevin van der Mark 4154509

This report describes the Wisent project required for the course Wireless Networking, ET 4394, at the Delft University of Technology.

Introduction

This project focuses on retrieving information on a backscatter signal. The devices used are described in the thesis "Backscatter Tag-to-Tag Network" by Michel Jansen. All code used can be found at `https://github.com/1424Bravo/Wireless/tree/master/RFID_sniffer`.

Goal

The goal is to decode the raw signal which was provided, as shown in figure 1, and deliver bit length information and frame length statistics.

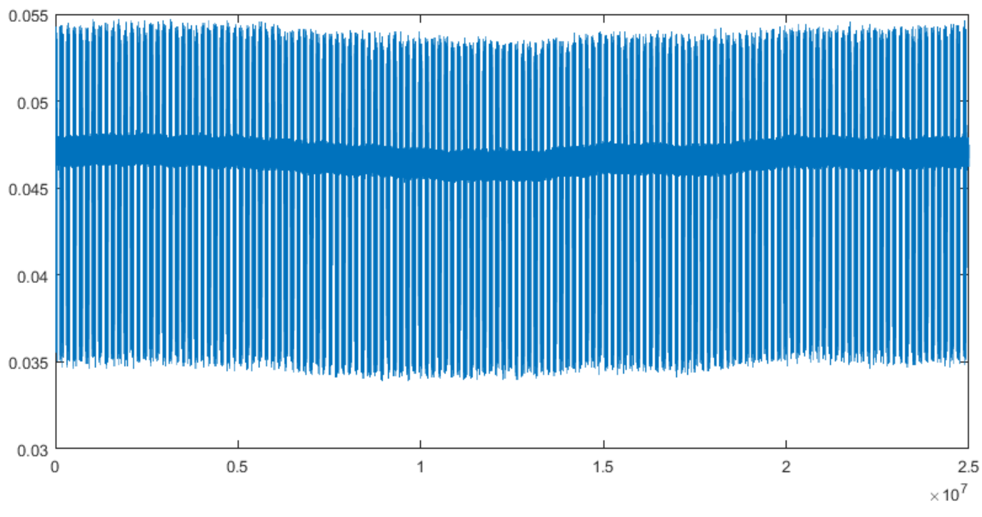

Figure 1: The provided raw signal

Obtaining the signal

The data files are obtained using GNU radio. The resulting .dat files can not be imported into Matlab but a Matlab function is provided, slice_signal.m. The signal obtained has some of noise, which is why a filter is implemented. A butterworth filter of order 6 is used with a cut off frequency of 9 kHz. One frame of the resulting signal is shown in figure 2.

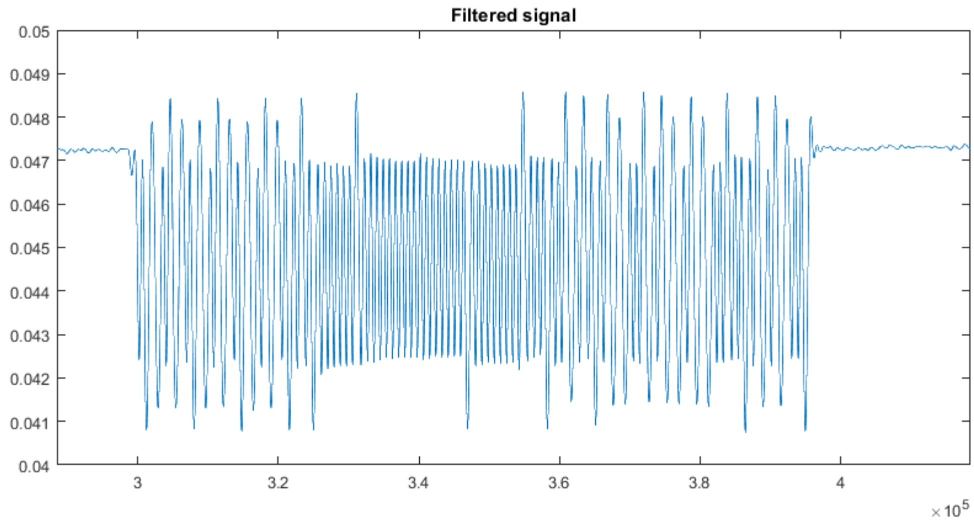

Figure 2: Filtered signal

From the filtered signal, an average signal value is calculated and multiplied by 0.95 to obtain a threshold just below the mean. This threshold is used to normalize the signal, setting all values above the threshold to one and below the threshold to zero. The result is shown in figure 3

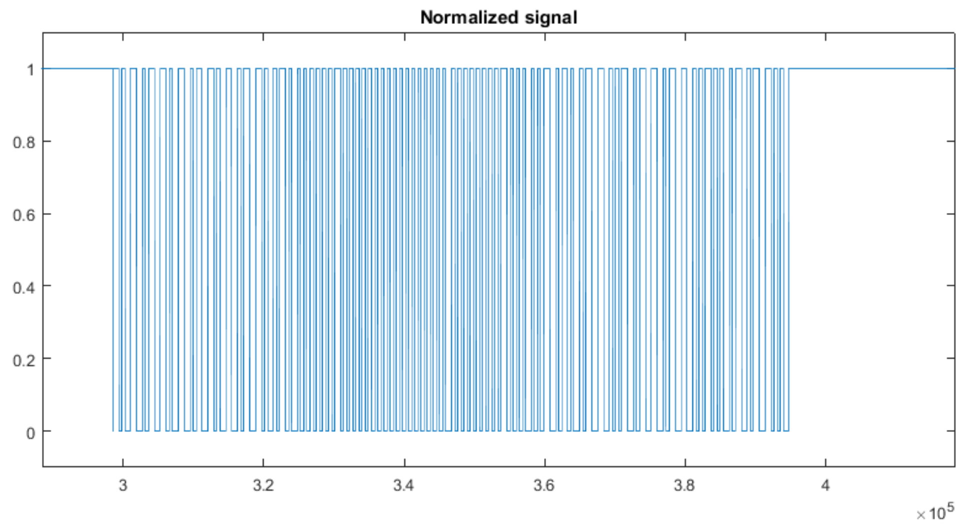

Figure 3: Normalized signal

Framing the signal

The provided signal hold many frames, which need to be split before it can be decoded. All frames are less then 100k samples, but to be sure the whole frame is captured, 10k guard samples are added before and after the frame. This results in frames of 120k samples, which are obtained using the function frames.m. For data validation, the start of the frame as a position of the whole sample is also an output of this function.

Decoding

The delivered data has 141 valid frames spread over 25 million samples. Each frame is processed using the decoder function, dec.m. The function starts by taking the differential of the normalized signal, resulting in +1 and -1 peaks for rising and falling edges of the signal, as shown in figure 4.

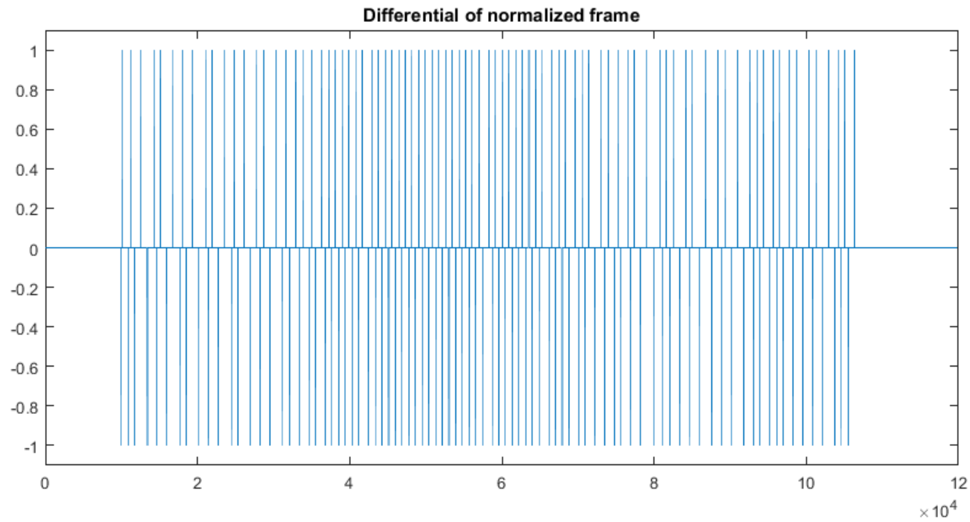

Figure 4: Differential of the normalized signal

Using this differential, all rising and falling edges are calculated and are all combined in one array. Using the difference between the rising and falling edges, the high and low periods are calculated. Since FM0 modulation is used, the period is calculated by multiplying the smallest high or low period with two. This gives a close approximation of the period of the signal, but is not perfect. The offset of this period calculation is compensated a few steps further.

To start decoding, the start variable is set to the first falling edge. The decoded then calculated the average value of the period $start$ to $start + period$. Since the signal is normalized and FM0 modulation is used, the mean of the period determines whether a '0' of a '1' was send. The start is moved by one period and the process is repeated.

Due to setting the period as as static value, a huge offset can occur at the end of the frame. To compensate for this, the new start is matched to the values in the array of rising and falling edges. If no match is found, the start is set to the nearest value in the array. This way the inaccuracy of the period is compensated and does not cause huge inaccuracies at the end of the frame.

Finally, the received bits are checked and frame data like bit lengths are send back to the main function. According to the thesis, the frame has the structure are shown in figure 5. The provided signal however was slightly different and has a preamble of 0xBBBBBB. The rest of the frame is the same and the main function splits the bits in their respective parts.

The CRC checksum is checked by calculating the correct CRC from the received data. It is then compared with the received CRC checksum. If the checksums are the same, the packet is considered valid. If the CRC checksums differ, the packet is ignored; its data is not valid.

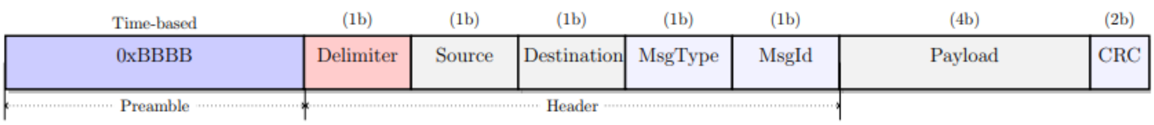

Figure 5: Frame structure of protocol [1]

[1] Michel Jansen, Backscatter Tag-to-Tag Network Embedded Software Master thesis, 11th December 2017

Results

The data is correctly decoded and divided among the different fields. Because the decoder works with rising and falling edges, obtaining timing data is trivial. In figure 6 multiple plots can be found including the average, min and max bit lengths in number of samples. As recommended, the data is presented in histograms.

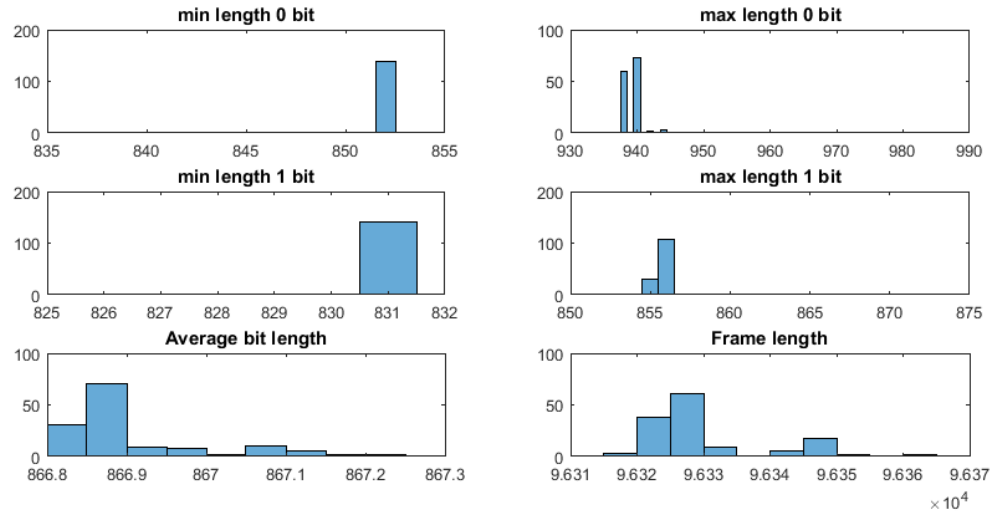

Figure 6: Bit length in number of samples in different cases

The results show very little deviation in the bit length in the provided sample. Between a ’0’ and a ’1’ however is a difference. Probably due to the shifts required for a ’0’, the extremes for a ’0’ are higher then for a ’1’. Between the frames and average bit length however is only 0.06% variance.

Using the frame length and the previously obtained start of each frame, it is possible to plot each frame at each stage with the start and end noted by the red points.

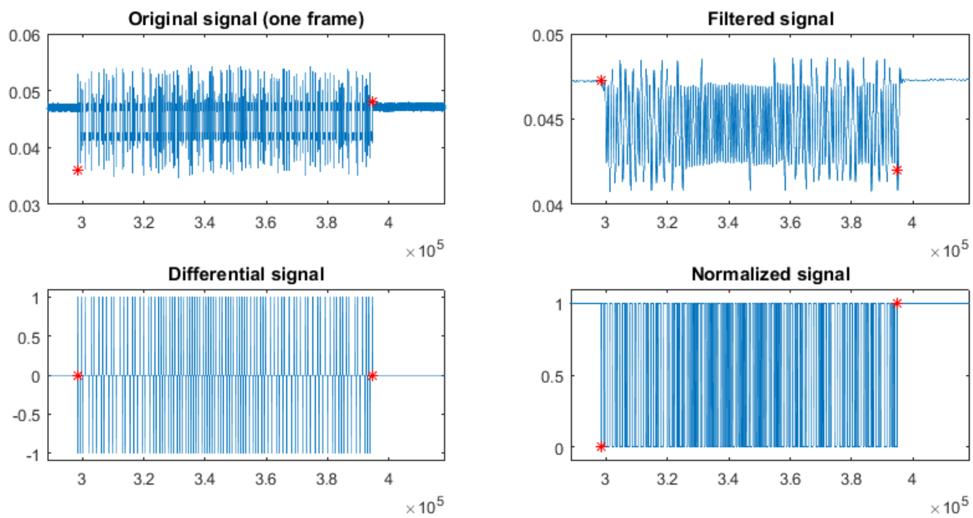

Figure 7: The original, filtered, differential and normalized signal with start and end