Estimating channel conditions at different rates

Purpose

The purpose of these tests is:

1. Measure drop rates on receivers, when the sender is moving randomly around (in the define space) and changing the multi cast data rate.
2. Figure out if there is any different between the multi cast data rates packet loss at a given time.

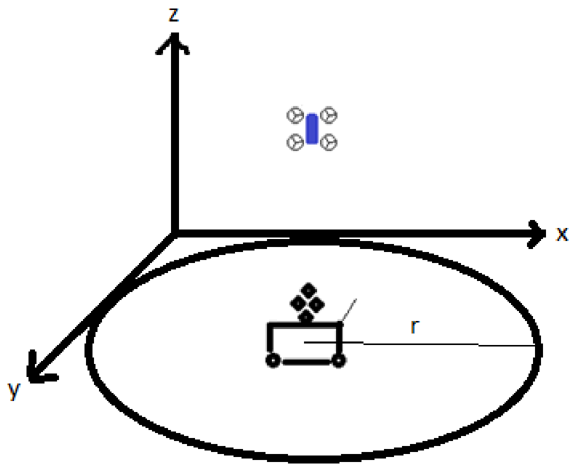
Method

Test parameters

To fulfill the purpose, we set up 5 Raspberry Pi 1 B as receivers and a Raspberry Pi 3 B mounted on a drone, which is the sender, as seen in the figure below. We conduct one test, varying the position and multi cast data rate of the drone. The mobility of the sender is bound by our scenario, which can be seen below.

Parameters to vary:

- The multi cast data rate of the sender.
- Position of the drone in the grid (x,y,z).
  - X and Y are defined in the ground plan and is bound by a circle with a radius of 100 meters, where the receiver is the center of the circle.
  - Z is the height, and have to be greater than 2 meter(higher than a person).



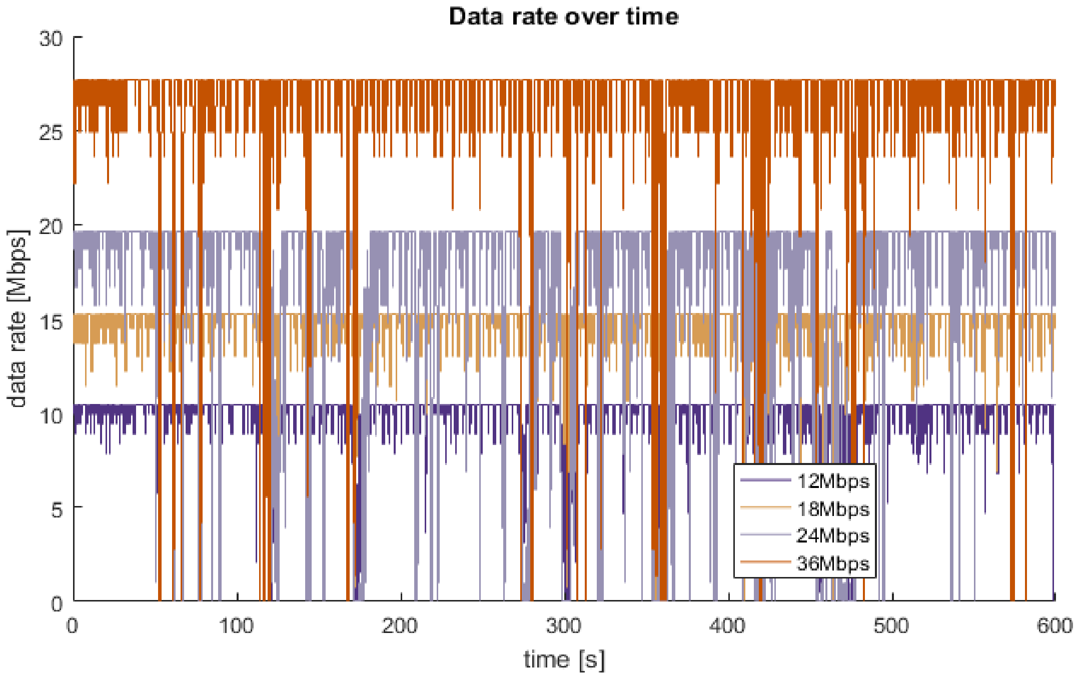
The receivers use a TP-link TL-WN722N USB Wi-Fi dongle and the sender used a RT5370 Wi-Fi dongle. We want to obtain a trace, which shows the packet loss at each of the data rates simultaneously. Since we are only able to transmit at one rate at a time, we use the following technique to estimate the packet loss at each data rate. The sender alternates between the four data rates 12 Mb/s, 18 Mb/s, 24Mb/s and 36Mb/s, sending 20 packets at each rate before switching. Since the data rate is changed from user space, we include a delay between switching data rate to ensure, that packets were send with the correct data rate.

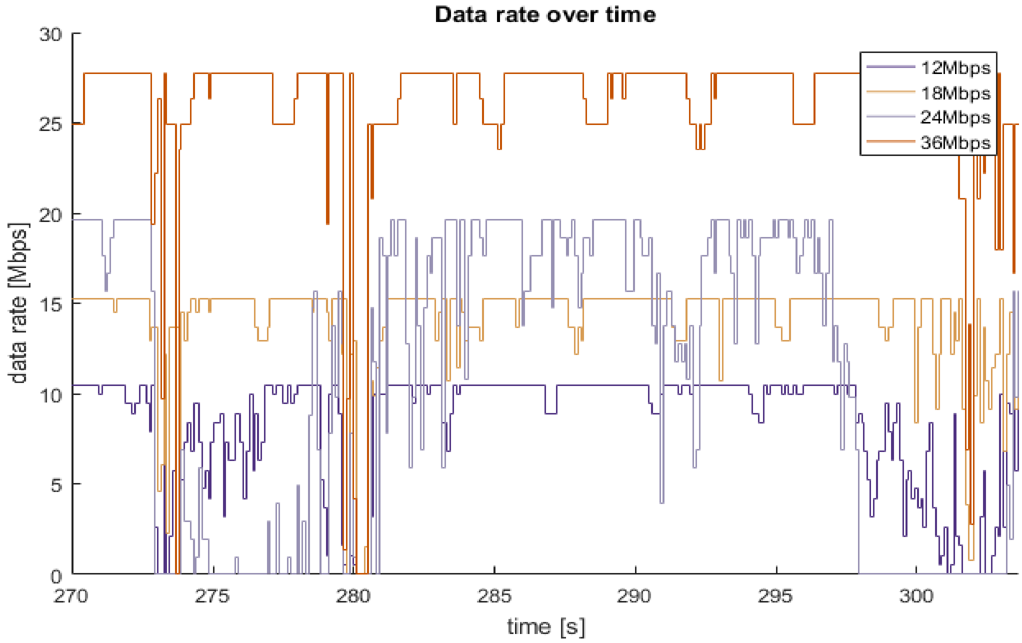
The lowest data rate we transmit at is 12 Mb/s, and considering packets of approx 1500 B, each packet takes approximately the following time to send:

$$t_{send} = \frac{Total\ data}{Transmission\ speed} = \frac{1500 * 8bit}{12 * 10^6\ bit/s} = 1ms$$

The practically obtainable data rate is lower since frame size is slightly larger and some parts always are transmitted with a lower rate. Furthermore when packets are sent in sequence other delays like a contention window between packets contribute to the fact that the actual time taken to transmit a packet is slightly larger than shown by equation above. Empirical experiments have shown that adding 25% guarantees with sufficiently large certainty, that the packet has been sent within this time. Therefore, we transmit 20 packets wait for 25 ms, switch data rate and repeat. This means that out of 100 ms, 1/4th of the time is used to transmit at one of the rates. We use the pattern we see in the 20 transmitted packets at each rate to estimate the behavior of the rate in the rest of the time interval. In the Figures below are the estimated traces. They show that the rates are strongly correlated which means that the errors we observe are not rate specific.

Result

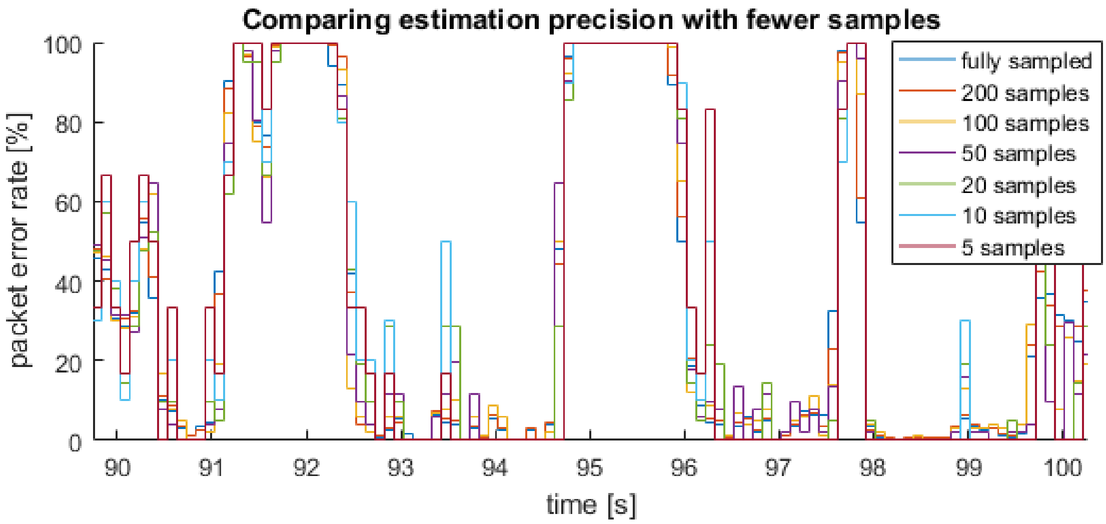




Evaluation of the estimation

We now evaluate how good an estimation we get by only using 5, 10 and 20 packets to estimate the loss behavior when all packets are used. This is done for the 36 Mb/s data since rate adaptation does not improve reliability in the scenario examined. We went out and collected a trace, where we continuously send packets at 36 Mb/s. From this trace we evaluate the estimation method.

We examine the 36 Mb/s trace and calculate the PER over the same amount of time, used for the other trace collections i.e. a 100ms period. In this period, the sender can transmit 239 packets. The “true” PER is based on the 239 packets. We calculate PER based on the 5, 10, 20, 50, 100 and 200 first packets of every period. We examine the correlation between the different PER traces by looking at the mean squared error. Next we do a moving average, averaging over a number of previous samples, for the different traces and take a mean squared error of these traces.

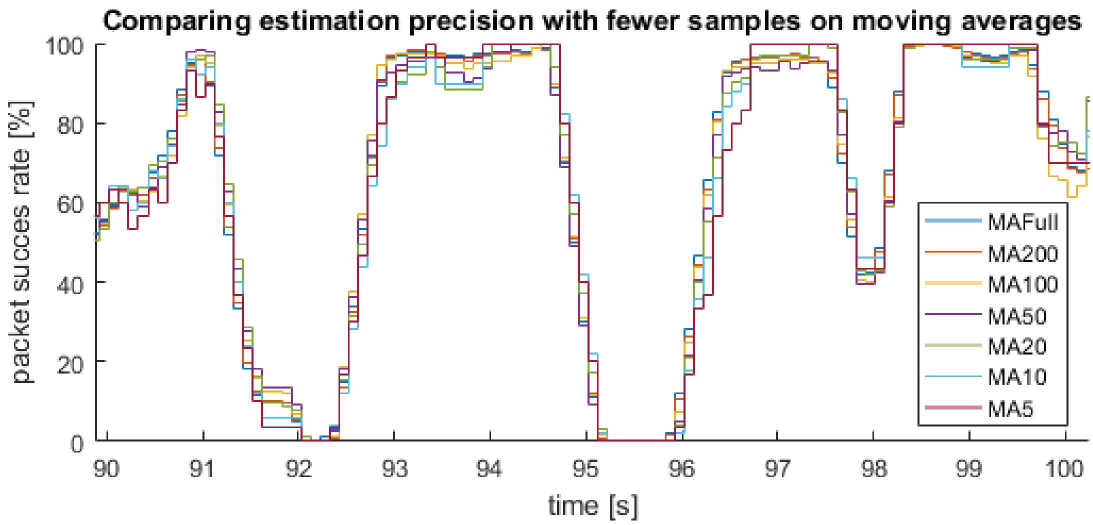


The figure above shows a segment of the packet error rate of the true and several estimations of the PER. We see, that while there is a clear trend throughout the data, all estimations deviate to some extent from the full trace. In the table below we show the mean error (ME) for the different estimations of PER have an insignificant error – 0.01% for the 20 sample estimation. This means that the average packet error rate of the entire trace is well estimated by all estimations.

Samples	5	10	20	50	100	200
ME [%]	0.02	0.01	0.01	0.03	0.02	-0.03
MSE [%^{2}]	223.6	179.5	123.2	62.3	29.9	2.9
MA (5) MSE [%^{2}]	47.2	37.9	25.9	13.7	6.7	0.7
MA (10) MSE [%^{2}]	24.2	19.3	12.9	7	3.3	0.3
MA (25) MSE [%^{2}]	9.4	7.5	4.8	2.6	1.2	0.1

Next, we look at the correlation on the single PER measurement level. This is done by examining the mean square error (MSE). The MSE for the 20 samples estimation has a MSE of 123.2 meaning the average aberration is $\sqrt{123.2} \approx 11\%$. This seems like a quite high average error. However, the high correlation between the real trace and estimation in PER over time is not included in the MSE. To account for the time correlation, we average the PER traces over 5, 10 and 25 samples, and calculate the MSE. MA (x) denotes the moving average, averaging over x samples including the current value and the previous $x - 1$samples.

As the moving average encompass a larger number of samples, the MSE becomes lower. The MSE of the MA (5) PER is approximately a factor 5 lower than the non-averaged MSE. This linearity confirms that the PER is highly correlated when observed over a period. Plotting the MA (25) PERs, see figure below, we see a high correlation as expected.



The mean error(ME) is calculated as:$ME = \frac{\sum(error)}{n}$

The mean square error(MSE) is calculated as:$MSE = \frac{(error_1^2 + error_2^2 + \cdots + error_n^2)}{n}$