CS5430 Project 2 Covert Channel Construction

Section 1 [3 single-sided pages max] should explain your covert timing channel, giving English descriptions (and informal code sketches if needed) to depict (i) how P1 signals a 1 or 0 and (ii) how P2 detects a bit.

P1 and P2 need to collaborate and write to a file as their daily job. However, P1 has secret information and wants to share such information with P2 using the file without any other person noticing. The covert channel that they use to share the information is described below:

Both P1 (source principal) and P2 (destination principal) have read/write access to a file. P1 signals a 1 or 0 by opening and holding exclusive access to the file for different time spans. After transmitting a bit, P1 releases the hold and waits for a while to let P2 acquire the file lock and detect the gap between transmitted bits. P2 periodically checks if it can gain access to the file by trying to acquire the exclusive write lock and immediately releasing it, and evaluates how long P1 holds the file. Based on the time P1 holds the file, P2 then can convert the time spans back to bits.

In our implementation, we need to config the following times to make the channel working --
`zero_time`: the time period that P1 holds the file when it intended to transmit a 1.
`one_time`: the time period that P1 holds the file when it intended to transmit a 0.
`transmit_gap`: the time period that P1 releases the file after it transmits a bit but before transmitting the next bit.
`busy_check_time`: the time that P2 waits before it checks the file status for the next time.

Section 2 [2 single-sided pages max] should give a credible and interesting graph of raw channel bandwidth versus channel fidelity rate for executions of P1 and P2 on a shared computer. Section 2 should also give the justification for the number and range of X values and for the number of measurements in constructing a Y point.

The bandwidth X mainly depends on the time P1 sleeps when signaling bits. To conduct our experiments, we fixed our P1 buffer size as 20 (i.e. the size of the message sent from P1) and chose multiple combinations zero_time and one_time to adjust the bandwidth.

There is a trade-off between raw channel bandwidth and channel fidelity rate. According to our initial experiment results, a bandwidth as small as 1 bit/sec leads to 1.0 fidelity rate, and a bandwidth more than 100 bits/sec leads to a less optimal fidelity rate. Thus, the range of our X points is from 0 to 300. Our graph depicts around 20 raw channel bandwidth points to reflect the trend. The X values are much denser around 1 bit/sec, due to the fact that it is important to find the reflection point between flat and non-flat curve.

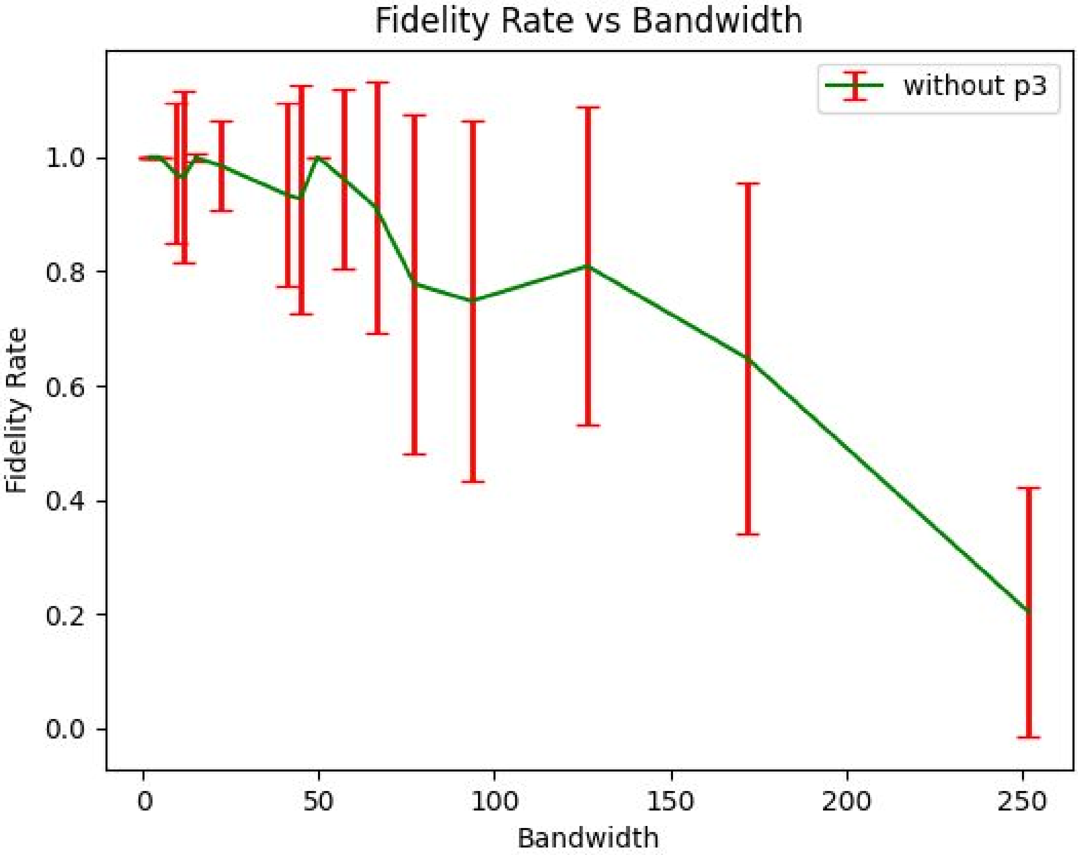
The zero_time and one_time (in seconds) we choose are

```
[(0.001, 0.002), (0.002, 0.004), (0.003, 0.006), (0.004, 0.008), (0.005, 0.01),
(0.006, 0.012), (0.007, 0.014), (0.008, 0.016), (0.009, 0.018), (0.01, 0.02), (0.02,
0.04), (0.03, 0.06), (0.04, 0.08), (0.05, 0.1), (0.1, 0.2), (0.15, 0.3), (0.2, 0.4),
(0.25, 0.5)]
```

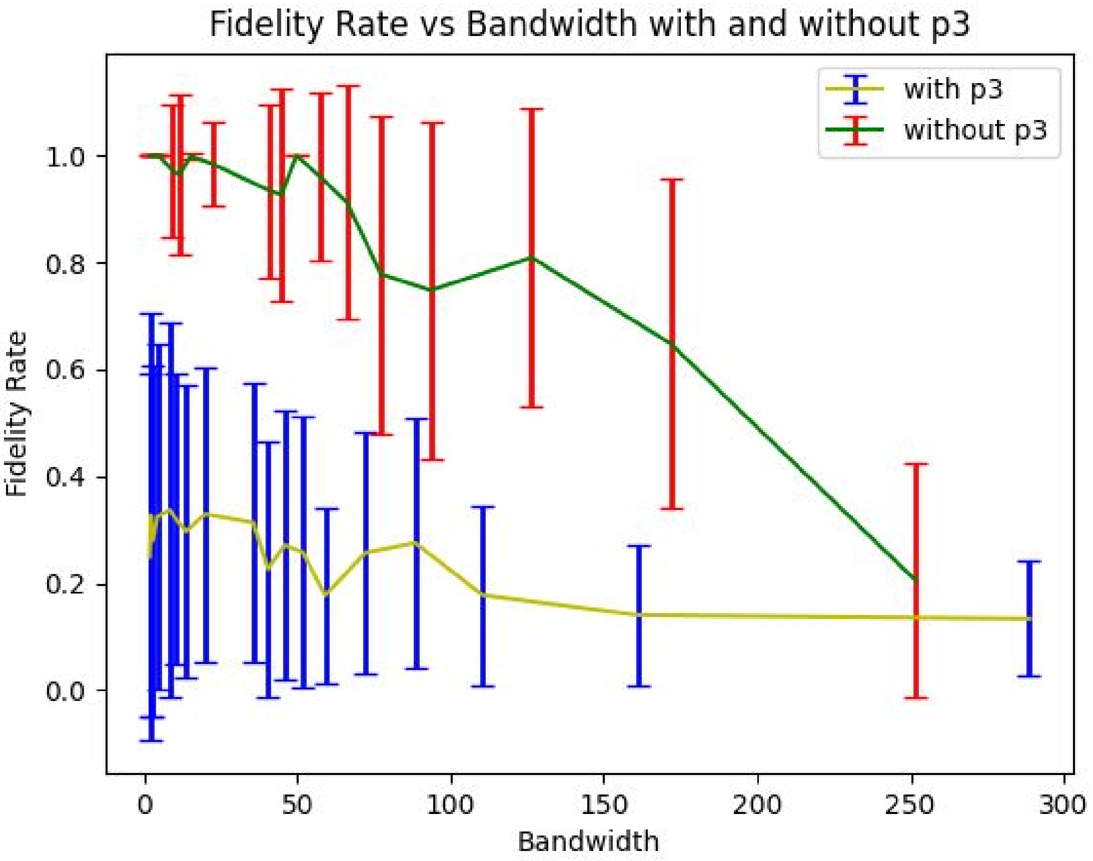
These 20 X values are mathematically sound, in terms of both the number and the range, since it concludes the reflection point of the channel, and well depicts the trend of large X values with a smooth curve. The range is reasonable because we can see that at the left end of the graph, the points converge at a fidelity rate of 1 so we do not need to explore any smaller bandwidth. At the right end of the graph, the fidelity rate drops below 0.5, which means the covert channel is no longer accurate enough to be useful so we do not need to explore any larger bandwidth.

This graph is drawn by measuring 50 times for each X value to obtain fidelity rate Y and the standard deviation of Y.

We test ran 5, 10, 20, 30, and 50 times to obtain different graphs, and found that, when measuring 5 times and 10 times for X, the standard deviations are relatively greater, and when measuring 30 times and 50 times for X, the standard deviations are reasonably close to each other. This accords with statistical knowledge that as the sample size increases, the standard deviation of the means decreases. Also, the central limit theorem establishes that as long as the sample is based on 30 or more observations, the sampling distribution of the mean is approximately normal.


Fidelity Rate vs Bandwidth
without p3
Fidelity Rate
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
250
Bandwidth

Extra Credit. Section 3 [2 single-sided pages max] should explain what P3 does and why that disrupts the covert timing channel, giving English descriptions (and an informal code sketch if useful). Section 3 should also give the graph described above in (3) for comparing channel performance with and without P3 execution.



Process 3 tries to access the file multiple times and when it gains the access to the file, it locks the file for a random period of time (long enough to confuse process 2) so that process 2 detects a gap between its two acquisition of the file lock and mistakenly views the gap as a message sent by process 1. Such a message might cause a misalignment in the bits sent later by process 1 and leads to a drop in the fidelity rate. In this way, the fidelity rate might drop while at the same time process 1 and 2 still can write to the file and do their normal work.

The graph above shows the fidelity rate vs bandwidth for the covert channel with and without p3. We can see from the graph that adding p3 causes a significant drop in the fidelity rate.