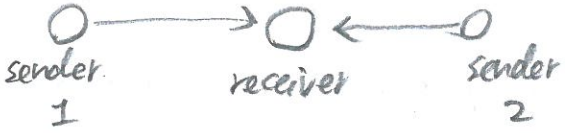
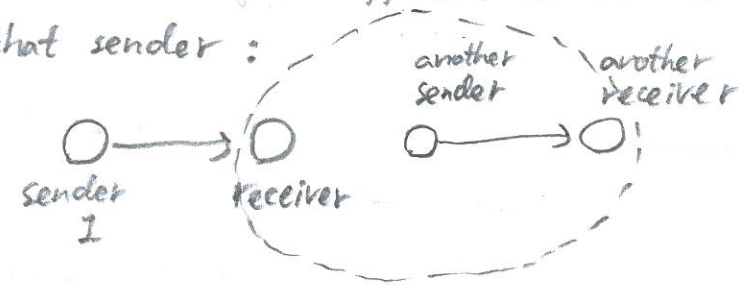
Name: Solution Student ID: CSC0056 Homework 5

(1.1) Primary interference refers to the case where there are multiple senders trying to send data to the same receiver at the same time:



(1.2) Secondary interference refers to the case where a receiver was interferred by a sender for someone else because the receiver happened to be within the transmission ranger of that sender:



(1.3) Graph coloring algorithms try to assign mutual colors to adjacent entities (depending on the context, the entities may be vertices, links, geographic regions, etc.). The requirement of a collision-free data communication schedule is to assign mutual time slots to wireless links that may suffer from primary/secondary interferences (because they are adjacent in the sense that at least one wireless node from each link is within the transmission range of another).

Name: _____ Student ID: _____

P2 (2.2) The contributions are summarized in the fourth paragraph:

① a holistic analysis for real-time reliable IoT edge computing;

② the ARREC architecture;

③ an implementation of ARREC and its empirical performance evaluation.

We may think in this way: an architecture needs to be based on some theoretical analysis that addresses the challenges we are to tackle; the analysis offers guidelines for us to design an architecture. An architecture needs to be implemented in a way that demonstrates the promised performance; the implementation and evaluation offer evidences for us to see the architecture's capability in dealing with the challenges mentioned.

(3.2.1) Here's a reason for the misleading result:

The data in e2e.delay.misleading contains some latency results obtained while the broker was still busying accepting new connections. Those results have larger latency. You may verify this by examining the first few lines in e2e.delay.misleading. Now, with a longer sampling duration we essentially average out these abnormal results, and therefore the averaged end-to-end latency is shorter.

An important lesson that we may learn here:

A software system often needs some time to "warm up", and thus we should avoid mixing the measurement obtained in the warm-up phase with that obtained in the later phase, to prevent some misinterpretation of the result.

ame: _____ Student ID: _____

(2.1) This is described in the related work section. With a higher workload it is likely that $T \uparrow$, and therefore from $N = \lambda T$ we see that $N \uparrow$, which implies (that) in average there will be more data in the system at any given point of time. So the system is more likely to have data losses should the primary IoT gateway fail to work.

(2.2) In scenarios such as temperature monitoring, provided that the sampling rate is high, we may use interpolation (內插法) to have a reasonable estimation of the lost data.

(2.3) In this case we have the assurance that the system can always recover some lost data so that there will be no two consecutive data missings, which by definition meets the application requirement "$L_i \geq 1$".

(3.2) @s prints the number of seconds since the Epoch.
@N prints the number of nanoseconds that have passed in the current second.

For the last question, the discrepancy could result from the fact that this public broker is very busy (people around the world are using it) and thus a message may need to wait for a while for the broker to complete processing those messages that have arrived earlier. #