RFC 1: Collisions in Delay-Based ID Encoding Protocol

Margot Maxwell Lizzy Schoen Noah Strong Chloe Yugawa

October 28, 2017

Contents

1 Introduction

The protocol proposed for this project encodes an unique identification number in the transmitted signal by the delay between two consecutive pings. While this method is easy to implement in hardware, and works well in the case of only a signal transmitter, the addition of multiple transmitters within range of a receiver brings about the potential for collisions between different signals. In some cases, the collisions may be undetectable, leading to incorrect reporting of nearby crab IDs. Completely solving this problem will require a change to the underlying protocol.

However, the likelihood of such collisions may be low enough that we move forward with this known flaw in the protocol.

2 Original Protocol

In the current iteration of our detection and identification protocol, every transmitter will transmit at the exact same frequency, likely somewhere around 40kHz. This is ideal from a hardware perspective, as the piezoelectric equipment can be tuned to work on a single frequency with a high degree of accuracy.

Every transmitter will periodically send out two quick "pings," each separated by some delay d. The value of d will encode the ID of the transmitter. For example, $d = 42$ms may correspond to ID 30, and a delay of 50.5ms may correspond to an ID of 38. Note that these numbers are only examples. Because the receiving hardware can easily detect the two pings, it can measure the value of d by calculating the time difference between the rising edges of the consecutive signals. See Figure 1 for an illustration.

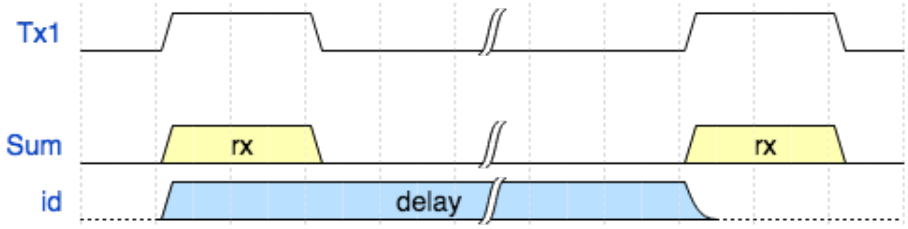

Figure 1: A single transmitter ($Tx1$), the input received by the receiver (Sum), and the calculated delay d based on the time measured between the rising edges of the two pings from $Tx1$.

Each transmission of an ID by a transmitter (that is, the sequence of a ping, a delay, and another ping) will happen regularly around some average interval. Because there is no synchronization between transmitters, the interval between each broadcast will vary randomly within a given range. For example, the delay may be 30 seconds, ± 5 seconds, with the random variation recalculated after every broadcast.

The motivation behind the randomly-varying schedule is to decrease the likelihood of simultaneous transmissions by two different receivers. However, the random interval only functions to reduce the likelihood of two *consecutive* overlapping transmissions. Without an inter-receiver collision-detection solution, there will always be the possibility that two transmissions overlap.

3 Possibility for Collisions

As discussed above, when two transmitters are present in the receiving range, and there is no synchronization between the two (i.e. they may transmit their IDs at any time, regardless of when the other transmits), it is possible for the two transmissions to be broadcast at similar times,

thereby overlapping. The signals may overlap in countless different ways depending on when both transmissions start and where the transmitters are in relation to each other and to the receiver.

In some cases, collisions may cause the data to be some so skewed that it will be easy to detect and throw away. However, in a more likely and more dangerous situation, the overlap of data will lead to inaccurate results and be extremely difficult to detect.

3.1 Example Scenarios

Below are a few possible scenarios that may occur in practice or in testing environments. Note that this is by no means an exhaustive list, but instead a selection of cases chosen to demonstrate potential issues.

We'll define some notation for the sake of convenience and clarity. If $Tx1$ is a transmitter, then d_1 is the delay time between the two pings for $Tx1$; that is, d_1 encodes the unique ID for $Tx1$. We'll also use B_1 to denote the first ping from $Tx1$ (the **B**eginning of the encoding) and E_1 to denote the second (**E**nd) ping.

3.1.1 Close Proximity, Similar Start

Situation: Two transmitters in close proximity to each other transmit at very nearly the same time. Their transmissions line up so that the first ping from $Tx2$ starts as the first ping from $Tx1$ is broadcasting. If the delay times happen to be very similar for the two transmitters, it is possible that the same effect is observed in reverse on the ending pings. That is, the second ping for $Tx2$ begins just before the second ping for $Tx1$. See Figure 2 for an illustration.

Potential Effect: Should this happen, the receiver will register two pings, just as it may if only one transmitter broadcasts. However, the measured delay $d_{measured}$ will not equal d_1 or d_2. Thus the software will report the ID corresponding to $d_{measured}$, and not the ID of either of the transmitters actually broadcasting. Worse still, there is no way to detect when this happens.

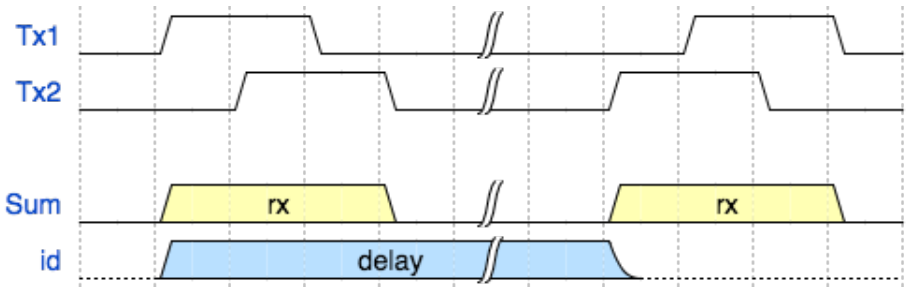

Figure 2: An example collision. The computed delay d is measured as the time difference between the rising edge of $Tx1$'s first ping and $Tx2$'s second ping.

3.1.2 Close Proximity, Offset Broadcasts

Situation: As in the previous example, suppose there are two transmitters in close proximity to each other. However, in this situation, suppose the two transmissions are broadcast at similar times so that they are interleaved. That is, the receiver will see B_1, B_2, E_1, E_2, in that order. See Figure 2 for an illustration.

Potential Effect: In this case, the receiver will detect two different delay times, which may seem normal. However, these delays are once again wrong – they do not correspond to any transmitter actually transmitting. The software will then report that there are two signals coming from a specific direction, but the IDs that it thinks these correspond to will be incorrect.

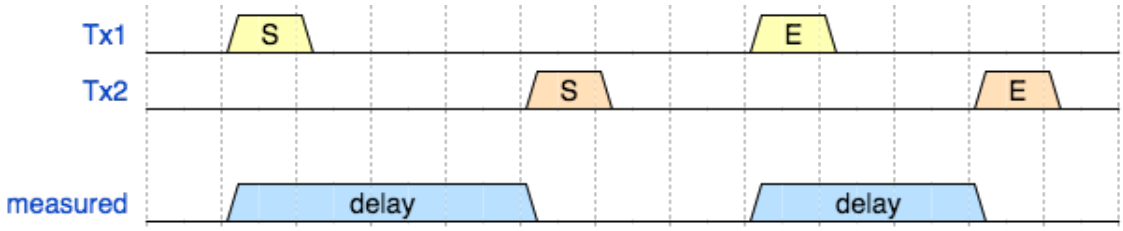

Figure 3: Another example collision. Note that in this situation, two distinct delays are measured. While it is possible for them to both correlate to valid IDs, neither delay matches the delays actually encoded by $Tx1$ or $Tx2$.

4 Potential Solutions

4.1 Use Two Frequencies To Differentiate Pings

One proposed solution is for every transmitter to have the ability to broadcast at two different frequencies, and use one for the first ping and another for the second. For example, the first ping may be 40 kHz and the second one could be 42 kHz.

Pros:

- With some added computation, we could compensate for some overlaps, such as the one described in example 3.1.2. Though some extreme cases, such as 3.1.1, would still lead to undetected incorrect results, the majority of overlap cases could be fixed.

Cons:

- This method would not scale to > 2 transmitters overlapping. Situations analogous to 3.1.2 but with 3 or more transmitters would still cause problems. However, the likelihood of such events occurring is extremely low in almost any reasonable configuration.

4.2 Attempt to Detect Anomalies Through Statistics

Another proposal is to make the software smart enough to remember where a signal comes from every time a transmission is received. It keeps track of where it thinks all the known transmitters are and compares that data with incoming data whenever a transmitter is detected.

Further investigation into this suggestion is needed.

5 Statistical Probability of Collisions

6 Discussion