MIT 6.829: Computer Networks Fall 2017

Lecture 10: Backscatter

Scribe: Patrick Wahl, Harry Hsu

1 Overview

Topics covered:

- RFIDs
- Backscatter
- Gen2 Protocol
- Buzz

2 RFIDs

RFID, or radio frequency identification, is a communication scheme in which many nodes, or tags, respond to a powered node (or a reader/interrogator), generally through the use of backscatter communication. These tags can either be:

- *Passive*, using no battery to operate
- *Active*, using a small battery to enhance communication

Some benefits of RFIDs are that they are ultra low power and very cheap. The most costly part of most tags is the chip that stores the tag's data and controls backscattering, while the portion that generally takes up the most physical space is the large antenna needed to power the chip and communicate.

RFIDs store a 96-bit ID number, which can be large enough so that every existing tag can have a unique ID. The main value of RFID tags is that they can communicate this number to a reader, so that the ID number can be used (for example, in searching a database) to determine which item is attached to the tag.

Unlike WiFi, which generally operates at 2.4 GHz, or cellular, which operates over a wide frequency range, RFID communication occurs in two frequency bands:

- UHF (ultra high frequency): 860-960 MHz
- VHF (very high frequency): 13.56 MHz

UHF readers have a longer range (8-10 meters) and are used in product identification and item location, while VHF readers have a very short range and are used in things like door keys or transit passes.

3 Backscatter

Backscatter is the method in which RFID tags communicate with readers. Instead of generating and modulating their own waveforms, tags will instead alter the reflectivity of their antennas in order to modulate waves being transmitted by the reader. Therefore, there are only two states the tag needs to have: reflect, and don't reflect. This is called the on-off keying (OOK).

3.1 Pros of Backscatter

- Requires little to no power from tags - tags are powered by reader

3.2 Limitations of Backscatter

- RFIDs can't hear each other - unlike in WiFi's MAC protocol, which senses if another device is talking on the medium, RFID tags have no way of telling - how can they avoid collisions if they can't tell when someone else is transmitting?
- RFIDs don't do bit rate adaptation - although they can set their own bit rates, the protocol does not find the best one for communicating on a given channel. It would be inefficient to do this, since generally the bulk of the data being sent is only 96 bits.

4 Gen2 Protocol

This is the protocol which determines how RFID tags and interrogators communicate. It is based on Slotted Aloha, an early protocol. Gen2 operates as follows:

1. The reader picks a number of slots, Q, and transmits a query alerting the tags that there are Q slots.
2. Each tag that hears the query picks a random number between 0 and Q, not inclusive.
3. If a tag's number is 0 then it transmits an RN16 (16-bit random number) back to the reader. If there is no collision, the reader can ACK the tag by using the RN16, letting it know to transmit the ID.
4. The reader can send out successive queries, each one causing the tags to decrement their numbers between 0 and Q and transmit an RN16 if their number reaches 0.

4.1 Example

Suppose $Q = 4$. If there is no tag:

1. The reader transmits a query setting $Q = 4$.
2. The reader listens 4 times, once for each slot, but gets no responses.

Now, suppose there is one tag.

1. The reader transmits a query, setting $Q = 4$.
2. The tag receives the query and sets its random number to 2. Since its number does not equal 0, it doesn't transmit.
3. The reader does not get any responses in slot 0, so it sends out another query.
4. The tag receives the query and decrements its number to 1, so it still doesn't respond.
5. The reader again gets no responses and so sends out another query.
6. This time, the tag receives the query and sets its number to 0, so it is time to transmit. It generates an RN16 and sends that.
7. The reader receives the RN16, and sends an ACK back with the same RN16.
8. The tag receives the ACK, checks that the RN16 matches, and since it does, the tag transmits its 96-bit ID.
9. The reader, after processing the ID, sends out a final query, and, receiving no responses again, finishes the cycle.

4.2 Remarks

If Q is too small, many tags will pick the same number after the initial query, so there will be many collisions. However, if Q is too large, the reader will have to cycle through a number of empty slots in between the slots which actually have tags.

It turns out that the most efficient method is that if there are K tags, set $Q = K$. However, even if K is somehow known, the best efficiency which can be achieved is still

$$\frac{1}{e} \approx 0.37$$

5 Buzz

The core idea of Buzz is, instead of using divisional multiplexing scheme, we are able to let the senders collide, effectively creating a code across the virtual senders' bits. Moreover, with its decoding scheme, it adapts the transmission rate automatically by repeatedly boosting previous decoded sequences, making Buzz rateless.

5.1 Compress Sensing

The most general formulation of compress sensing is

$$\mathbf{y} = \mathbf{A}\mathbf{x},$$

where $\mathbf{y} \in \mathcal{R}^{M\times 1}$, $\mathbf{A} \in \mathcal{R}^{M\times N}$, and $\mathbf{x} \in \mathcal{R}^{N\times 1}$. Equivalently, we have M equations and N unknowns.

- If $M = N$ and $\mathbf{A}$ has rank N, $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$.
- If $M < N$, we have to impose more constraint on the solution. For example, the solution that has the least squared error is $\mathbf{x} = \mathbf{A}^{\star}\left(\mathbf{A}\mathbf{A}^{\star}\right)^{-1}\mathbf{y}$. However, this solution is not useful in our application since we want our solution to be sparse.
- If $\mathbf{x}$ has at most $K \ll N$ non-zero entries, we can recover $\mathbf{x}$ from $M \ll N$ measures, where $M = O\left(K \log \frac{N}{K}\right)$. In this case, $\mathbf{A}$ must satisfy restricted isometry property (RIP).

In reality, there will be a channel response from every sender, that is

$$\mathbf{y} = \mathbf{A}\mathbf{H}\mathbf{x},$$

where $\mathbf{H}$ is a diagonal matrix. Since the vector $\mathbf{z} = \mathbf{H}\mathbf{x}$ is sparse and has only K non-zero entries, this is again a compress sensing problem.

5.2 Belief Propagation Algorithm

The objective of the belief propagation algorithm is to find a vector $\hat{\mathbf{b}}$ that could have produced the received signal $\mathbf{y}$, or more concretely to find $\hat{\mathbf{b}}$ that minimizes the error:

$$\min_{\hat{\mathbf{b}}} \left\| \mathbf{D}\mathbf{H}\hat{\mathbf{b}} - \mathbf{y} \right\|_2$$

.

It works as follows:

1. $\hat{\mathbf{b}}$ is initialized to be random binary vector.
2. Let $\tilde{\mathbf{b}}_i = \hat{\mathbf{b}}$ with dimension i flipped, and calculate $G_i = \left\| \mathbf{D}\mathbf{H}\hat{\mathbf{b}} - \mathbf{y} \right\|_2 - \left\| \mathbf{D}\mathbf{H}\tilde{\mathbf{b}}_i - \mathbf{y} \right\|_2$.
3. Repeat step 2 on different dimensions, and find i that minimizes G_i.
4. Let $\hat{\mathbf{b}} = \tilde{\mathbf{b}}_i$ and repeat steps 2 - 3.