This Module's Topic: RFIDs

- RFIDs are passive, wireless devices
  - Power is harvested from the RF emitted by a reader
  - Communication/sensing is possible only from a few inches to perhaps a few meters
What Are They For?
What Are They For?

In Texas, 28,000 Students Test an Electronic Eye

Published: November 17, 2004

But on the morning Friger and Christopher shared a seat on bus No. 38, the district experienced one of the early technology hiccups. When the bus arrived at school, the system had not worked. On the Web site that includes the log of student movements, there was no record that any of the students on the bus had arrived.

It was just one of many headaches; the system had also made double entries for some students, and get arrive times and addresses wrong for others. “It’s early glitches,” said Brian Weiss, the head of transportation for the Spring district, adding that he expected to work out the problems.
The Inventorying Problem

Reader

Anyone there?

Yes!

Yes!

Yes!

Yes!

Yes!

Yes!

Yes!

Yes!

Yes!

Yes!

Yes!
Physical Constraints

• They have almost no memory
  – Memory is bit addressable!

• They have almost no compute
  – They’re hardware implementations of simple state machines, not von Neumann computers

• They have almost no transmit power
  – In fact, they have none – they backscatter a carrier transmitted by the reader
    • Low bandwidth, high bit error rate

• Result: communication is largely under control of the reader. (Tags never speak unless spoken to.)

A Few Specifics

• We’ll use the specific instance of the tags implemented for the next assignment.
  – They’re based on the spec for the Class 1 Generation 2 UHF RFID (860-960 MHz)

• Storage on the tag:
  – EPC: electronic product code (48-256 bits)
  – SL: selected bit (settable by reader)
  – INV: inventoried bit (settable by reader)

• Bandwidth is O(100Kbps)

• Bit error rate (BER) is (okay, no one knows for sure, but we’re saying) 0.1% - 1%
The Flavor of RFID Communication

Select | Directive from reader to conditionally modify SL or INV bit. The condition is a bit string that must match memory at a specified location.

Query | Reader supplies tags with a guard condition and a window size value. Tags meeting the guard choose a random slot. Any that choose slot 0 reply; others wait.

RN16 / ACK | Short temporary identifier supplied by tag, then used by reader to request the EPC.

Example

Reader

Tags

State

SL

INV

false true

false true

false

true

Arbitrate

Ready

Aked

Ready

Ready

Reply

if QSlot == 0

if rand_{rsent} == rand_{rcv}

if Q = 4

if Q = 4

QSlot = U[0,15]

if QSlot == 0

Reader

Select(set SL, mask = 0, len = 0 )

Query(set SL \land \neg INV, Q = 4 )

Ack(rand_{rsent})

RN16(rand_{rsent})

EPCFrame(EPC)

Query(set SL \land \neg INV, Q = 4 )
Protocol Issues

• What approach to collision resolution should be used?
  – Goal, say, is to obtain EPCs of all tags during the small time that the pallet is next to the reader

• What should be done to protect against bit errors?
  – What is the argument for transmitting error detection bits?
    • Against?

• Should you use ACKs and/or ARQ?
  – The spec defines the rules, and there are no ACKs. (Why?)
  – There are some situations where repeating a request is possible and makes sense.

Collision Resolution

• What the tags do in response to received frames is part of the spec
  – Not under software control

• Software decides what frames to send to them, though

• More on frames/tags in a second, but first let’s try to relate this to what we’ve seen before
**Ethernet vs. 802.11 vs. RFID Link Layers**

<table>
<thead>
<tr>
<th></th>
<th>Bandwidth</th>
<th>BER</th>
<th>Collision resolution runs in...</th>
<th>CR is part of spec?</th>
<th>Carrier Sense possible</th>
<th>Collision Detect possible</th>
<th>ACKs / ARQ</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet</td>
<td>High</td>
<td>Low</td>
<td>Sender</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>802.11</td>
<td>Medium</td>
<td>Medium</td>
<td>Sender</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RFID</td>
<td>Low</td>
<td>High</td>
<td>Reader (Receiver)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>?</td>
<td>Depends on frame</td>
</tr>
</tbody>
</table>

**RFID Collision Resolution Approaches**

- The reader needs to somehow distinguish (any) one tag from all the others

- To do this, it has to make use of something on a tag that distinguishes it from the others:
  - The tag’s EPC
  - The tag’s randomly selected slot number
A Slot-based Scheme (Appendix D of the Spec)

EPC-based Query Tree
Frames (Tag “Instructions”)

- **Select**
  - Set or invert SL or INV iff what is in a tag’s memory starting at a particular bit matches a (variable length) bit string in the Select frame

- **Query**
  - “Selects” tags with particular value of SL and INV
  - Provides a “backoff window” size
  - Tags pick a random slot in backoff window and respond if slot = 0

- **QueryRepeat**
  - Tags participating in the current round decrement their slot counter by 1
  - Respond if updated slot = 0

- **QueryAdjust**
  - Tags in the current round double, halve, or leave unchanged, the current backoff window
  - They then pick a new random slot and respond if slot = 0