Radio-frequency identification (RFID) is a technology that uses communication via electromagnetic waves to exchange data between a terminal and an object such as a product, animal, or person for the purpose of identification and tracking.

Some tags can be read from several meters away and beyond the line of sight of the reader.

An RFID tag used for electronic toll collection.
RFID Systems

- Involves *interrogators* (also known as readers), and *tags* (also known as labels)
- Three types of RFID tags:
  - *Passive RFID tags* (no power source; require an external electromagnetic field to initiate a signal transmission)
  - *Active RFID tags* (contain a battery and can transmit signals once an external source 'Interrogator' has been successfully identified)
  - *Battery Assisted Passive (BAP) RFID tags* (require an external source to wake up but have significant higher forward link capability providing greater range)
RFID Systems (Performance)

- Desirable Application (reading protocol)
  - Fast
  - Energy efficient

- Problems
  - Increasing number of tags
  - **Collisions** due to the tags replying simultaneously to a reader
    - Wasting bandwidth
    - Wasting energy
    - Increase in identification delay
RFID Systems (Anti-Collision Protocols)

- Collision Problem
RFID Systems (Anti-Collision Protocols)

Multiple Access / Anti-Collision Protocols

- SDMA
- TDMA
- FDMA
- CDMA

Tag Driven (Asynchronous)
- Pure Aloha and its variants

Reader Driven (Synchronous)
- Slotted Aloha and its variants
- Framed Slotted Aloha (FSA)

Tree Protocols
- Basic FSA and its variants
- Dynamic FSA and its variants
- Query tree (QT) based protocols
- Tree splitting (TS) based protocols
- Binary Search (BS) based protocols
- Bitwise arbitration (BTA) algorithm
RFID Systems (Anti-Collision Protocols)

- **SDMA (Space Division Multiple Access)**
  - Separate the channel using directional antennas or multiple readers
  - Price + Special antenna design

- **FDMA (Frequency Division Multiple Access)**
  - tags transmitting in one of several predefined frequency channels
  - Complex receiver at the reader

- **CDMA (Code Division Multiple Access)**
  - tags multiply their ID with a pseudo-random sequence (PN) before transmission
  - Price + Power hungry

- **TDMA (Time Division Multiple Access)**
  - Reader-Talk-First (RTF)
  - Tag-Talk-First (TTF)
RFID Systems (Anti-Collision Protocols)

- TDMA (Time Division Multiple Access)
  - Reader-Talk-First (RTF)
  - Tag-Talk-First (TTF)

[ALSO]

- Aloha-based
- Tree-based
Aloha-based Protocols

- Aloha based Protocols
  - Pure Aloha (PA)
  - Slotted Aloha (SA)
  - Framed Slotted Aloha (FSA)
    a) Basic framed slotted Aloha (BFSA)
    b) Dynamic framed slotted Aloha (DFSA)
    c) Enhanced Dynamic framed slotted Aloha (EDFSA)
Pure Aloha (PA)

- A tag responds with its ID randomly after being energized by a reader.
- Waits for the reader to reply with:
  1. ACK indicating its ID has been received correctly
  2. a negative acknowledgment (NACK), meaning a collision has occurred

Pure ALOHA protocol. Boxes indicate frames. Shaded boxes indicate frames which have collided. In Pure ALOHA, only about 18.4% of the time is used for successful transmissions.
Pure Aloha (PA)

**PA with Muting:** The number of tags is reduced after each successful tag response. Hence, *muting* has the effect of reducing the offered load to the reader after each successful identification.

**PA with Slow Down:** Instead of being muted, a tag can be instructed using a *slow down* command to reduce its rate of transmissions, hence decreasing the probability of collision.

**PA with Fast Mode:** A *silence* command is sent by the reader once it has detected the start of a tag transmission. This command has the effect of stopping other tags from transmitting. Tags are allowed to transmit again after the reader has sent an ACK command or until their waiting timer expires.
Pure Aloha (PA) with Muting

Tag 1

Tag 2

Tag 3

Reader

Shared Medium

Time

Tag Transmission

Collision

Mute
Pure Aloha (PA) with Slow Down
Pure Aloha (PA) with Fast Mode

- Tag 1
- Transmission begins
- Tag 2
- Tag Transmission
- Collision
- Tag 3
- Reader
- Silence Command
- Shared Medium
- Time
Pure Aloha (PA) with Fast Mode and Muting

[Diagram showing the timeline and states of different tags (Tag 1, Tag 2, Tag 3) with annotations for Silence, Mute, and Collision.]
Pure Aloha (PA) with Fast Mode and Slow Down

Tag 1
Silence
Slow

Tag 2
Transmission begins
Slow

Tag 3
Silence
Silence

Shared Medium

Reader

Time

Tag Transmission
Collision
Slotted Aloha (SA)

- An improvement to the original ALOHA was Slotted ALOHA, which introduced discrete timeslots and increased the maximum throughput.
- A station can send only at the beginning of a timeslot, and thus collisions are reduced.

If there is a collision, tags retransmit after a random delay. The collision occurs at slots boundary only, hence there are no partial collisions. The maximum throughput is approximately 0.368 frames per frame-time, or 36.8%.

Slotted ALOHA protocol (shaded slots indicate collision)
Slotted Aloha (SA)

**SA with Muting/Slow Down:** The principle operation is similar to PA with muting/slow down, but operates in a slotted manner.

**SA with Early End:** If no transmission is detected at the beginning of a slot, the reader closes the slot early. Two commands are used: *start-of-frame* (SOF) and *end-of frame* (EOF).

**SA with Early End and Muting:** The reader sends a *mute* command whenever it successfully identifies a tag.

**SA with Slow Down and Early End:** This combines *slow down* with the *early end* feature.
Slotted Aloha (SA) with Early End

Diagram showing the concept of Slotted Aloha (SA) with Early End. The diagram illustrates the behavior of tags over time, with different states indicated by grey boxes. "Idle slots terminated early" is noted at the top. The diagram also includes symbols for collision and tag transmission.
Frame Slotted Aloha (FSA)

- FSA protocols mandates that each tag responds only once per frame

**Basic Frame Slotted Aloha (BFSA):** the term *basic* refers to the frame size being fixed throughout the reading process

**Dynamic Frame Slotted Aloha (DFSA):** FSA protocols with variable frame size. In each read round, the reader uses a *tag estimation function* to vary its frame size. The optimal frame size is one which promises the maximum system efficiency and minimum identification delay. Theoretically, the optimal frame size is equal to the number of tags

*No prior knowledge of the number of tags!*
Basic Frame Slotted Aloha (BFSA)

- BFSA has four variants:
  1) BFSA-non muting
  2) BFSA-muting
  3) BFSA-non-muting-early-end
  4) BFSA-muting-early end
Choosing the right *frame size* is a crucial issue to determine the performance

**Optimal frame sizes for a given tag range.**

<table>
<thead>
<tr>
<th>Frame Size (N)</th>
<th>Low (n)</th>
<th>High (n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>32</td>
<td>10</td>
<td>27</td>
</tr>
<tr>
<td>64</td>
<td>17</td>
<td>56</td>
</tr>
<tr>
<td>128</td>
<td>51</td>
<td>129</td>
</tr>
<tr>
<td>256</td>
<td>112</td>
<td>∞</td>
</tr>
</tbody>
</table>
Comparison of Aloha-based Protocols

(Delay – Pure/Slotted)

Highest

Delay to read a tag successfully from n tags

Lowest
Comparison of Aloha-based Protocols

(Energy Consumption - Pure/Slotted)

Energy Consumed to read a tag successfully from n tags
Comparison of Aloha-based Protocols
(Delay - Framed Slotted)
Comparison of Aloha-based Protocols
(Energy Consumption – Frame Slotted)
Tree-based Protocols

- Tree based Protocols
  - **Tree Splitting (TS)**
    Splitting responding tags into multiple subsets using a random number generator
  - **Query Tree (QT)**
    Storing tree construction information at the reader, and tags only need to have a prefix matching circuit
  - **Binary Search (BS)**
    Involves the reader transmitting a *serial number* to tags, which they then compare against their ID. Those tags with ID equal to or lower than the serial number respond
  - **Bitwise Arbitration (BTA)**
    Requesting tags to respond bit by bit from the most significant bit (MSB) to the least significant bit (LSB) of their ID
Tree Splitting

**Basic Tree Splitting (BTS)**: the reader informs the tags about the last time-slot resulted in collision/single/no response

**Tag’s counter in the BTS algorithm.**

<table>
<thead>
<tr>
<th>Time slots</th>
<th>Feedback</th>
<th>Tag A</th>
<th>Tag B</th>
<th>Tag C</th>
<th>Tag D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Collision</td>
<td>0 (Transmit)</td>
<td>0 (Transmit)</td>
<td>0 (Transmit)</td>
<td>0 (Transmit)</td>
</tr>
<tr>
<td>2</td>
<td>Collision</td>
<td>0 (Transmit)</td>
<td>0 (Transmit)</td>
<td>1 (Wait)</td>
<td>0 (Transmit)</td>
</tr>
<tr>
<td>3</td>
<td>Identified</td>
<td>0 (Transmit)</td>
<td>1 (Wait)</td>
<td>2 (Wait)***</td>
<td>1 (Wait)</td>
</tr>
<tr>
<td>4</td>
<td>Collision</td>
<td>—</td>
<td>0 (Transmit)</td>
<td>1 (Wait)***</td>
<td>0 (Transmit)</td>
</tr>
<tr>
<td>5</td>
<td>Collision</td>
<td>—</td>
<td>0 (Transmit)</td>
<td>2 (Wait)</td>
<td>0 (Transmit)</td>
</tr>
<tr>
<td>6</td>
<td>Identified</td>
<td>—</td>
<td>1 (Wait)</td>
<td>3 (Wait)</td>
<td>0 (Transmit)</td>
</tr>
<tr>
<td>7</td>
<td>Identified</td>
<td>—</td>
<td>0 (Transmit)</td>
<td>2 (Wait)</td>
<td>—</td>
</tr>
<tr>
<td>8</td>
<td>Idle</td>
<td>—</td>
<td>—</td>
<td>1 (Wait)</td>
<td>—</td>
</tr>
<tr>
<td>9</td>
<td>Identified</td>
<td>—</td>
<td>—</td>
<td>0 (Transmit)***</td>
<td>—</td>
</tr>
</tbody>
</table>

*Tags in the wait state increment their counter by one because of collision.

** Tags in the wait state decrement their counter by one because of identified tag.

*** Tags in the wait state decrement their counter by one because of idle response.
Tree Splitting

Basic Tree Splitting (BTS)
Tree-based Protocols

- **Tree Splitting (TS)**
  Splitting responding tags into multiple subsets using a random number generator

- **Query Tree (QT)**
  storing tree construction information at the reader, and tags only need to have a prefix matching circuit

- **Binary Search (BS)**
  involves the reader transmitting a *serial number* to tags, which they then compare against their ID. Those tags with ID equal to or lower than the serial number respond

- **Bitwise Arbitration (BTA)**
  requesting tags to respond bit by bit from the most significant bit (MSB) to the least significant bit (LSB) of their ID
Query Tree (QT)

Each tag has a *prefix matching* circuit. The reader transmits a query \( q \), and tags with a matching prefix reply to the reader. Collision occurs when multiple tags have the same prefix. In this case, the reader forms a new query by appending \( q \) with a binary 0 or 1. The reader then repeats the reading process using the augmented query.
**Shortcutting:** This extension reduces QT’s identification delay by removing redundant queries. The reader transmits a query \( q \), and if there was a collision, the reader appends \( q \) with 0 and 1, and pushes \( q0 \) and \( q1 \) onto the stack. The reader first transmits the query \( q0 \). If there was no response, the reader infers that at least two tags have the prefix \( q1 \). Thus, if the reader transmits \( q1 \), a collision will occur. Therefore, the reader removes the query \( q1 \) from the stack and pushes \( q10 \) and \( q11 \) onto the stack instead.

**QT-Shortcutting Algorithm**
Tree-based Protocols

- **Tree based Protocols**

- **Tree Splitting (TS)**
  Splitting responding tags into multiple subsets using a random number generator

- **Query Tree (QT)**
  Storing tree construction information at the reader, and tags only need to have a prefix matching circuit

- **Binary Search (BS)**
  Involves the reader transmitting a *serial number* to tags, which they then compare against their ID. Those tags with ID equal to or lower than the serial number respond

- **Bitwise Arbitration (BTA)**
  Requesting tags to respond bit by bit from the most significant bit (MSB) to the least significant bit (LSB) of their ID
Binary Search (BS)

Round 1
Collision (010, 011, 100, 110)
Reply {XXX}

Round 2
Collision (010, 011)
Reply {01X}

Round 3
Tag (010)

Round 4
Collision (011, 100, 110)
Reply {XXX}

Round 5
Tag (011)

Restart
ID=111

Round 6
Collision (100, 110)
Reply {1X0}

Round 7
Tag (100)

Restart
ID=111

Round 8
Tag (110)

Collision
Identified
Tree-based Protocols

- **Tree Splitting (TS)**
  Splitting responding tags into multiple subsets using a random number generator

- **Query Tree (QT)**
  Storing tree construction information at the reader, and tags only need to have a prefix matching circuit

- **Binary Search (BS)**
  Involves the reader transmitting a *serial number* to tags, which they then compare against their ID. Those tags with ID equal to or lower than the serial number respond

- **Bitwise Arbitration (BTA)**
  Requesting tags to respond bit by bit from the most significant bit (MSB) to the least significant bit (LSB) of their ID
Bitwise Arbitration (BTA)

- Bit replies are synchronized, meaning multiple tags responses of the same bit value result in no collision. A collision is observed only if two tags respond with different bit values. Moreover, the reader has to specify the bit position it wants to read.

- **Bit query (BQ):** A reader transmits a bit query \( q \) to tags. Tags with their prefix matching the query \( q \) respond with the bit that is adjacent to the requested prefix. Other tags deactivate themselves. If the reader receives a tag’s bit response successfully, that bit is sent as the next query. However, if there is a collision, the reader uses bit zero as the next query.
RFID Applications

- A RFID system consists of three key components:
  - RFID tag or transponder
  - Reader/writer
  - Application computer or processor

- Invented in 1948 --> Lab. experiments before 1950 --> Application trial in 1960 --> rapid growth in 1990s --> Total revenue of 5.6 B$ earned by RFID products in 2009
RFID Applications (e.g. Animal Management)
RFID Applications (e.g. Transportation and logistics)
RFID Applications (e.g. Asset management and retail sales)
RFID Applications (e.g. Passports)

The contactless chip found in British passports
RFID Applications (e.g. Race Timing)
RFID Applications (e.g. Human Implants)
Thank You
Questions?