专题1：RFID的识别、轮询与估算机制

谢磊 博士
南京大学计算机科学与技术系
主要内容：

一、RFID的基本通信原理

二、基于时隙ALOHA的标签识别机制（Identification）

三、RFID标签轮询机制（Polling）

四、RFID标签估算机制（Estimation）

五、开放性问题 (Open Problem)

六、参考文献
• Far-Field Propagation and Backscatter Principle

Fig. 1. Far-Field Propagation for RFID system
• Far-Field Propagation and Backscatter Principle
  
  – We denote the down-link communication from the reader to a tag as the *forward channel*, and denote the up-link communication from a tag to the reader as the *reverse channel*.
  
  – For successful reading of a passive tag with the backscatter scheme, there are two thresholds to meet the physical requirements. The first is the *tag power (sensitivity) threshold*, $P_{ts}$. It is the minimum received power necessary to turn on an RFID chip. The second is the *reader sensitivity threshold*, $P_{rs}$. It is the minimum level of the tag signal that the reader can detect and resolve.
  
  – Thus it must satisfy $P2 > P_{ts}$ for the tag to be powered up and resolve the received signal, and also $P4 > P_{rs}$ for the reader to detect and resolve the received signal.
• Tag inventory and access

Fig. 2. C1G2 protocol
• Tag inventory and access
  – The MAC protocol for the C1G2 system is based on *Slotted ALOHA*, where each frame has a number of slots and each active tag will reply in a randomly selected slot per frame.
  – When a reader (interrogator) wishes to read a set of tags, it first powers up and transmits a continuous wave (CW) to energize the tags. It then initiates a series of frames, varying the number of slots in each frame to best accommodate the number of tags. After all tags are read, the reader powers down. We refer to an individual frame as a *Query Round*, and the series of *Query Rounds* between power down periods as a *Query Cycle*. 
• Tag inventory and access
  – For each Query Round, the reader can optionally transmit a Select command which limits the number of active tags by providing a bit mask. Then a Query command is transmitted which contains the uplink frequency and data encoding, the $Q$ parameter determining the number of slots for the following frame, and a Target parameter.
  – When a tag receives a Query command, it chooses a random number in the range $(0, 2^Q - 1)$, where $Q$ is in the range $(0, 15)$, and the value is stored in the slot counter of the tag. If a tag stores a 0 in its slot counter, it will immediately backscatter a 16 bit random number, denoted by RN16. Upon receiving RN16, the reader will echo RN16 in an ACK command. If the tag successfully receives RN16, it will backscatter its ID information ($PC+EPC+CRC16$). Then a subsequent QueryRep command will be sent to the tag, signaling the end of the slot and toggling an inventoried flag in the tag to make it keep silent in the following rounds.
• Tag inventory and access
  – If the ID is not successfully received by the reader, an $NAK$ command is sent which resets the tag so as to keep the tag active in the next round.
  – Upon receiving the $QueryRep$ command, the remaining tags will decrement their slot counters, and respond with $RN16$ if their slot counters are set to 0. When the number of $QueryReps$ is equal to $2^Q$, the current $QueryRound$ ends.
Frame size $f=3$

RFID Reader

Query $f=3$

RN16

RN16

PC+EPC+CR C16

QueryRep

RN16

RN16

RN16

QueryRep

RN16

Tag 1 Slot=0

Tag 2 Slot=10

Tag 3 Slot=1

Query Round 1
Frame size $f=2$
• Efficiency Problem:
  – How to select an optimized frame size $f$ for each query round such that the overall scanning time can be minimized?

• Problem Formulation
  – The reader is essential to issue a number of *query rounds* to finish identifying all tags. Suppose in a certain *query round* $l$, the frame size is $f$, the number of remaining tags is $t$, the objective is how to set a frame size $f$ for each query round, such that the overall scanning time $T$ (or: overall number of slots) is minimized while finishing reading all tags.

• Solution:
  – Modeling
  – Compute the optimal frame size
• **Lemma 1.** Let \((N_0, N_1, N_c)\) represent the number of time slots with no transmissions, one transmission and collision respectively in a system with \(t\) tags and frame size \(f\). Let \(\rho = \frac{t}{f}\). Then

\[
\begin{align*}
E[N_0] &\approx f e^{-\rho} \\
E[N_1] &\approx f \rho e^{-\rho} \\
E[N_c] &\approx f(1 - (1 + \rho) e^{-\rho})
\end{align*}
\]
Proof:

\[
\lim_{x \to +\infty} \left(1 - \frac{1}{x}\right)^x = \frac{1}{e}
\]

• Slot \(j\) will be empty if none of the tags transmit in that slot. Therefore, \(Pr[X_j = 1] = (1 - \frac{1}{f})^t \approx e^{-\rho}\). This implies that

\[
E[N0] = \sum_{j=1}^{f} Pr[X_j = 1] \approx f e^{-\rho}.
\]

• Similarly, \(Pr[Y_j = 1] = t \frac{1}{f} (1 - \frac{1}{f})^{t-1} \approx \rho e^{-\rho}\), and

\[
E[N1] = \sum_{j=1}^{f} Pr[Y_j = 1] \approx f \rho e^{-\rho}.
\]

• Since \(X_j + Y_j + Y_j = 1\) for all \(j\),

\[
E[Nc] = \sum_{j=1}^{f} Pr[V_j = 1] \approx f - fe^{-\rho} - f \rho e^{-\rho}.
\]
• The portion of singleton slots inside each frame is $E[N1]/f \approx \rho e^{-\rho}$.

• Then in order to maximize the portion of singleton slots inside each frame, we compute $\frac{\partial E[N1]/f}{\partial f} = 0 \rightarrow f^* = t$. In this way, $E[N1]/f^* = \frac{1}{e}$.

• It infers that, during each *query round*, when the frame size $f$ is equal to the remaining number of tags $t$, the portion of singleton slots inside each frame is maximized. Then the local optimum is achieved.

• Furthermore, as the efficiency over all *query rounds* is $\frac{1}{e}$, which is the upper bound of efficiency among all *query rounds*, thus actually the global optimum is achieved.
• In the frame slotted ALOHA protocol, how do the tags randomly pick a slot inside a frame?
  – The procedure is actually pseudo-random.
  – For each query round, the reader broadcasts a random number $r$ to the tags.
  – Each active tag computes a slot number $s$ according to the following equation: $s = \text{hash}(id, r) \mod f$, where $id$ is the tag’s ID, $f$ is the current frame size.
  – Therefore, as long as the random number $r$ and the frame size $f$ is set, the position in the frame for any specified tag to appear can be predicted.
• Hash function
  – In order to keep the tag’s circuit simple, its hash value may be derived from a pool of pre-stored random bits: We use an offline random number generator with the ID of a tag as seed to generate a string of 200 random bits, which are then stored in the tag. Note that the random number generator is not executed by the tag. The bits form a logical ring.
  – H(id, r) returns a certain number of bits after the rth bit in the ring. 200 random bits provide 200 different hash values, which are sufficient for conventional purpose.
  – The above hash design does not place any restriction on the number of random bits, and a number larger than 200 can be chosen when necessary.
• Problem Formulation

– Consider a large RFID system of $N$ tags. Each tag carries a unique ID and has the capability of performing certain computations as well as communicating with the RFID reader wirelessly. The reader has a list of IDs of these $N$ tags. The problem is to design efficient protocols for the reader to exchange necessary information with the tags in order to identify the missing ones.

– The objective is to minimize the overall scanning time while finishing identifying all missing tags.
• Problem Analysis

  – We classify the time slots based on their lengths: tag slots, long-response slots and short-response slots. The length of a tag slot is denoted as $t_{tag}$, which allows the transmission of a tag ID, either from the reader to the tags or from a tag to the reader. The length of a long-response slot is denoted as $t_l$, which allows the transmission of a long response carrying multi-bits information. The length of a short-response slot is denoted as $t_s$, which allows the transmission of a short response carrying only one bit information. Clearly, $t_s < t_l < t_{tag}$.

  – To design a time efficient protocol, we prefer the use of short-response slots over long-response slots or the use of long-response slots over tag slots.
• Baseline Protocol

- We observe that, since the RFID reader has access to the database of tag IDs, it does not have to read such information directly from the tags. Instead, it can broadcast these IDs one after another. After it transmits an ID, it waits for a short response from the tag that carries the ID. If it receives the response, the tag must be in the system; otherwise, the tag is missing.

- The verification of each tag’s existence takes $t_{tag} + ts$, and the total execution time is $N(t_{tag} + ts)$. This is called the baseline protocol. Comparing with the tag-collection protocols, it significantly reduces the execution time by eliminating the contention among the tags.
• Baseline Protocol

Reader ——— Short Response ——— Tags

<table>
<thead>
<tr>
<th>ID1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Response</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Response</td>
</tr>
</tbody>
</table>

………
• Two-Phase Protocol
  - The protocol consists of two phases: a frame phase and a polling phase.
  - The frame phase verifies the presence for a majority of the tags without any ID transmission. At the beginning of this phase, the RFID reader transmits a request \( <r, f> \), where \( r \) is a random number and \( f \) is the frame size. The frame consists of \( f \) short-response time slots right after the request. Each tag is pseudo-randomly mapped to a slot at index \( H(id, r) \), where \( id \) is the tag’s ID and \( H \) is a hash function whose range is \( [0..f-1] \). The tag transmits a short response at that slot.
- Two-Phase Protocol (frame phase)

Request \( <r, f> \)

- Expected State

- Reader

- Short Response

- Short Responses

- Tags

- Missing Tag
• Two-Phase Protocol
  – Because the reader knows the IDs of all tags, it knows which slot each tag is supposed to respond. Hence, it knows the locations of the empty, singleton and collision slots. If a slot is supposed to be singleton but the reader finds it to be empty, then the tag that is mapped to the slot must be missing. The frame phase can verify the existence of all tags that are mapped to the singleton slots. However, it cannot verify the existence of the tags that are mapped to the collision slots.
  – The polling phase performs the baseline protocol on the tags that are mapped to the collision slots in the frame phase. The reader broadcasts their IDs one after another. Upon receiving an ID, the tag that carries the same ID transmits a short response, allowing the reader to learn its presence.
Two-Phase Protocol with Tag Removal (Intuition)

- TPP can be further improved. Suppose two tags, x and y, are mapped to a collision slot in the frame phase. When the reader detects the slot is non-empty, it cannot determine whether both tags are present or only one of them is present.
- Hence, it has to broadcast both IDs in the polling phase. This approach is inefficient because the information carried in the collision slot is totally unused. To make the collision slot useful, we shall turn it into a singleton slot by removing one of the two tags from the frame phase.
• Two-Phase Protocol with Tag Removal
  – Our third protocol, TPP/TP, also has two phases, but the polling phase goes before the frame phase.
  – In the polling phase, a tag removal procedure is invoked to determine the set $S$ of tags that will not participate in the frame phase. In this procedure, the reader first maps the tags to the slots as what TPP does. For each $k$-collision slot, it randomly removes $k-1$ tags to turn the slot into a singleton. The removed tags are inserted in $S$. After all collision slots are turned into singletons, the reader broadcasts the IDs of the tags in $S$ one after another to verify their presence. When a tag receives its ID, it will transmit a short response and keep silent in the frame phase.
  – The frame phase is the same as in TPP except that the tags in $S$ do not participate.
RFID 标签轮询机制 (Polling) - 5

- Two-Phase Protocol with Tag Removal (frame phase)
  request \( <r, f> \)

```
\begin{array}{cccc}
\text{Reader} & \text{Expected State} & \text{Tags} & \text{Missing Tag} \\
\hline
0 & 0 & & \\
1 & 1 & & \\
X & 1 & & \\
1 & 1 & & \\
0 & 0 & & \\
X & 1 & & \\
\end{array}
```

Short Response
• Three-Phase Protocol with Collision Sensitive Tag Removal (Intuition)
  – When \( f \) is reasonably large, most collision slots are 2-collision slots. Consider an arbitrary 2-collision slot. If the tags transmit short responses, the reader cannot distinguish the following two cases: (1) both tags are present and (2) only one tag is present. That is because in either case the reader detects the same nonempty slot. However, if the tags transmit long responses, the reader will observe a collision slot if both tags are present, and it will observe a singleton slot if only one tag is present.
  – Hence, observing an expected collision slot confirms that both tags are not missing, whereas observing an unexpected singleton slot means one of the tags is missing (but we do not know which one is missing). If an expected collision slot turns out to be empty, then both tags are missing.
• Three-Phase Protocol with Collision Sensitive Tag Removal
  
  – TPP/CSTR has three phases: a polling phase, a frame phase, and then another polling phase.
  
  – At the beginning of the first polling phase, TPP/CSTR executes a different tag removal procedure: The reader maps the tags to the slots in the same way as TPP does. For each \( k \)-collision slot with \( k \geq 3 \), it randomly removes \( k-2 \) tags to turn the slot into a 2-collision slot. The removed tags are inserted in \( S \). After all collision slots are turned into 2-collision slots, the reader broadcasts the IDs of the tags in \( S \) one after another to verify their presence. When a tag receives its ID, it will transmit a short response and keep silent in the frame phase.
- Three-Phase Protocol with Collision Sensitive Tag Removal (frame phase)

request \(<r, f>\)

<table>
<thead>
<tr>
<th>Reader</th>
<th>Expected State</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long Response</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Long Responses</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Long Response</td>
<td>X</td>
<td>2</td>
</tr>
<tr>
<td>Long Response</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Long Response</td>
<td>X</td>
<td>2</td>
</tr>
<tr>
<td>Long Response</td>
<td>X</td>
<td>2</td>
</tr>
</tbody>
</table>

Missing one tag, but cannot be verified.
• Three-Phase Protocol with Collision Sensitive Tag Removal
  – In the frame phase, the tags that are not in S transmit long responses. The reader records the slots that are expected to be 2-collision slots but turn out to be singletons. Only the tags that are mapped to these slots cannot be verified.
  – Hence, in the second polling phase that follows the frame phase, the reader broadcasts the IDs of these tags to verify their presence.
Iterative ID-free Protocol (Intuition)

- Transmitting the tag IDs in the polling phase is an expensive operation. In our final protocol (IIP), we remove the polling phase altogether. IIP iteratively performs the frame phase. Each frame verifies the presence of a portion of the tags. It repeats until short responses are received from all tags that are present. The tags whose responses are not received must be missing.
• Iterative ID-free Protocol

  - Let $S$ be the set of tags whose presence has been verified in the previous frames. Before a frame begins, the reader maps the tags not in $S$ to the slots of the frame in the same way as TPP does. When the reader sends the request $<r, f>$ to the tags, it also transmits a pre-frame vector, which consists of $f$ bits, each indicating the expected state of one slot, ‘0’ for empty or singleton and ‘1’ for collision.

  - Recall that a tag is mapped to the slot of index $H(id, r)$ in the frame. Since the reader knows all tags, it has the full knowledge of which are the collision slots. If a tag learns that it is mapped to a collision slot (i.e., the bit at index $H(id, r)$ in the preframe vector is ‘1’), it will decide with 50% probability to not participate in the current frame.
• Iterative ID-free Protocol
  request \(<r, f>\)

<table>
<thead>
<tr>
<th>Pre-frame vector</th>
<th>Expected State</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 1 0 0 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Short Response</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Short Response</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>Short Response</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Short Response</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>Short Response</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Post-frame vector</td>
<td>0 1 1 1 0 1</td>
<td></td>
</tr>
</tbody>
</table>
- **Iterative ID-free Protocol**
  - More specifically, the tag performs another hashing $H'(id, r)$ whose result is either ‘0’ or ‘1’. Only when the hashing result is ‘1’, it will participate in the current frame by transmitting a short response at slot $H(id, r)$. Since half of the tags mapped to collision slots will not participate, it helps resolve some collision slots and turn them into singletons.
  - Knowing all the IDs, the reader can also determine which tags will not participate, which collision slots will be turned into singletons, and which other tags will respond in those singletons and thus be verified.
• Iterative ID-free Protocol
  
  – After the tags respond in the slots of the frame, the reader measures the state of the slots and constructs a *post-frame vector*, consisting of \( f \) bits, each indicating the actual state of one slot, ‘0’ for empty or collision and ‘1’ for singleton. The presence of the tags that respond in the singleton slots is successfully verified, and the reader inserts them into \( S \). It then transmits the post-frame vector.

  – If a tag sees that its slot is a singleton (i.e., the bit at index \( H(\text{id}, r) \) in the post-frame vector is ‘1’), it will not participate further in the protocol execution. After transmitting the post-frame vector, the reader starts the next frame with a reduced size because there are fewer tags left to respond.
• Iterative ID-free Protocol
  
  – When no tag responds in a frame, the reader will repeat the same frame with a pre-frame vector of all zeros, which essentially requires all remaining tags, if there are any, to respond. If still no tag responds, the protocol terminates.
One of the common problems that arise in any RFID deployment is the problem of **quick estimation of the number of tags in the field up to a desired level of accuracy**.

Prior work in this area has focused on the identification of tags, which needs more time, and is unsuitable for many situations, especially where the tag set is dense.

In the frame slotted ALOHA protocol, how to fast estimate the cardinality of tags based on the number of empty /singleton /collision slots, while achieving the required accuracy requirement?
RFID标签估算机制（Estimation）

- Problem formulation
  - Given a set of $t$ tags in the system, the reader has to estimate the number of tags in the system with an confidence interval of width $\beta$, i.e., we want to obtain an estimate $t'$ such that
    
    $\frac{t'}{t} \in \left(1 - \frac{\beta}{2}, 1 + \frac{\beta}{2}\right)$

    with probability greater than $\alpha$. In other words, we need maximum error to be at most $\pm \frac{\beta t}{2}$ with probability greater than $\alpha$.

  - A sample problem would be to estimate the number of tags within $\pm 1\%$ of the actual number of tags with probability greater than 99.99\%. 
Lemma 1. Let \((N_0, N_1, N_c)\) represent the number of time slots with no transmissions, one transmission and collision respectively in a system with \(t\) tags and frame size \(f\). Let \(\rho = t/f\).

Then

\[
\begin{align*}
E[N_0] & \approx f e^{-\rho} \\
E[N_1] & \approx f \rho e^{-\rho} \\
E[N_c] & \approx f(1 - (1 + \rho) e^{-\rho})
\end{align*}
\]
• The reader measures \((n_0, n_1, nc)\). From Lemma 1, we know that the expected number of empty slots is \(f e^{-\rho}\), or the fraction of empty slots is \(e^{-\rho}\). From the current measurement the reader observes that the fraction of empty slots is \(n_0/f\). Equating the expected value and the observed value, the reader now determines \(\rho_0\) that solves \(e^{-\rho_0} = n_0/f\) and sets \(t_0 = f\rho_0\). Similarly, the reader can get estimates for \(t\) from the singleton slots as well as the collisions. We show the three estimates in Table I.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Problem to be Solved</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZE: Zero Estimator (t_0)</td>
<td>(e^{-(t_0/f)} = n_0/f)</td>
</tr>
<tr>
<td>SE: Singleton Estimator (t_1)</td>
<td>((t_1/f) e^{-(t_1/f)} = n_1/f)</td>
</tr>
<tr>
<td>CE: Collision Estimator (t_c)</td>
<td>(1 - (1 + (t_c/f)) e^{-(t_c/f)} = n_c/f)</td>
</tr>
</tbody>
</table>

Table 1: Estimators for \(t\)
• It is easy to solve for the estimator $t_0$ in closed form but the other two estimators involve solving a non-linear equation in one variable.

• A simple bisection search or Newton’s method can be used to solve the equation, since the estimation functions shown above are well behaved and therefore both these methods converge very quickly.

• We can also use the fact that the estimate has to be an integer to terminate the search once we know the interval of uncertainty is less than one.
The three estimators have very different characteristics.

In Figure 1 we plot the normalized expected values, $E[N_0]/f$, $E[N_1]/f$ and $E[N_c]/f$ as functions of the load factor $\rho$. Note that the curves for empty slots and collision slots are monotonic in $\rho$ but singleton slots is non-monotonic.
RFID标签估算机制（Estimation）

- Accuracy of the Estimators

**Theorem 2.** Let \( t, f \to \infty \) while maintaining \( t/f = \rho \).

Then

\[
[g_0 (N_0) - g_0 (\mu_0(t))] \sim \mathcal{N} [0, \delta_0]
\]

and

\[
[g_c (N_c) - g_c (\mu_c(t))] \sim \mathcal{N} [0, \delta_c]
\]

where

\[
\delta_0 = t \left( e^\rho - (1 + \rho) \right) \frac{1}{\rho}
\]  \hspace{1cm} (1)

and

\[
\delta_c = t \left( e^\rho (1 + \rho) - (1 + 2\rho + \rho^2 + \rho^3) \right) \frac{1}{\rho^3}
\]  \hspace{1cm} (2)
• Compare the variances of the two estimators
• Reducing the Variance of the Simple Estimators
  
  – If we want the variance to be less than $\sigma^2$, for a given estimate $\hat{t}$, we first set

  $$\hat{t} \left( e^\rho - (1 + \rho) \right) / \rho \leq \sigma^2$$

  and solve for $\rho$. We can then set $f \geq \hat{t} \rho$.

  – In practice, all systems have some maximum frame size restriction. Therefore, if the frame size computation above leads to a size larger than the maximum permitted, then we use the maximum permitted frame size instead. This implies that we may have to perform multiple experiments in order to reduce the variance.
• **Reducing the Variance of the Simple Estimators**
  
  – A straightforward way of reducing the variance of an estimator is to repeat the experiment multiple times and take the average of the estimates.

  – If the final estimate is the average of $m$ independent experiments each with an estimator variance of $\sigma^2$, then the variance of the average is $\sigma^2/m$. 
• Computing the Estimate using Combined Simple Estimators

  - Based on our earlier observation that the two estimators are complementary to each other, we can devise a unified Simple Estimation Algorithm, which uses both estimators, depending on the frame size and the estimated number of tags.

ESTIMATION PROCEDURE

1. Compute the desired variance, \( \sigma^2 = \frac{z^2}{\beta^2} \).
2. Compute the initial frame size \( f \) by solving \( fe^{-\frac{t}{f}} = 5 \).
3. Energize the tags and get \( n_0 \) and \( n_c \).
4. Compute \( t_0 \) as in Table 1 and the variance of this estimate \( \delta_0 \) using Equation 1.
5. Compute \( t_c \) as in Table 1 and the variance of this estimate \( \delta_c \) using Equation 2.
6. If \( \delta_0 < \delta_c \) then set \( \hat{t} \leftarrow t_0 \) else \( \hat{t} \leftarrow t_c \).
• Estimate the cardinality of tags based on posteriori probability
  
  Consider tags are to be read and a read cycle with a frame length of $f$ time slots. Given one of the time slots, the number of tags allocated in the slot is a binomial distribution with $n$ Bernoulli experiments and $1/f$ occupied probability. The probability of finding $r$ tags in the slot is therefore given by

$$B(r) = \binom{n}{r} \left(\frac{1}{f}\right)^r \left(1 - \frac{1}{f}\right)^{n-r}$$

  Accordingly, we obtain the probabilities of empty, successful transmission, and collision for the slot as $p_e = B(0)$, $p_s = B(1)$, $p_c = 1 - p_e - p_s$. 
• Estimate the cardinality of tags based on posteriori probability
  
  – We need to derive the probability of finding the exact $N_0$ empty slots, $N_1$ singly occupied slots, and $N_c$ collision slots if there are $f$ slots. The problem can be modeled as a multinomial distribution with repeated independent $f$ trials, where each trial has one of three outcomes: empty, successful, or collision.
  
  – Suppose that the possible outcome in each trial is $p_e$ for empty, $p_s$ for successful, and $p_c$ for collision. In general, the probability is subject to the condition $p_e + p_s + p_c = 1$.
  
  – The probability that in $f$ trials, empty outcome occurs $N_0$ times, successful outcome occurs $N_1$ times, and collision outcome occurs $N_c$ times is $P(N_0, N_1, N_c) = \frac{f!}{N_0!N_1!N_c!} (p_e)^{N_0} (p_s)^{N_1} (p_c)^{N_c}$
• **Estimate the cardinality of tags based on posteriori probability**

  Therefore, for a read cycle with frame length, we have a posteriori probability for the number of tags when empty slots, singly occupied slots, and collision slots are observed, as follows:

  \[ P(n, N0, N1, Nc) \]

  \[
  = \frac{f!}{N0! N1! Nc!} \times \left( 1 - \frac{1}{f} \right)^n \left( \frac{n}{f} \left( 1 - \frac{1}{f} \right)^{n-1} \right)^{N1} \\
  \times \left( 1 - \left( 1 - \frac{1}{f} \right)^n - \frac{n}{f} \left( 1 - \frac{1}{f} \right)^{n-1} \right)^{Nc}
  \]
• **Estimate the cardinality of tags based on posteriori probability**
  
  - Based on the a posteriori probability distribution, we attempt to determine tag quantity such that the probability is maximized. Hence, the decision rule of our proposed tag estimate method is as follows:
  
  - Set the tag estimate \( n' = n \) if \( P(n, N0, N1, Nc) \) is maximum.
  
  - This decision rule can be referred to as the maximum a posteriori probability rule.
Fig. 1. Reading the moving tags on the conveyor.
Lei Xie, Bo Sheng, Chiu Tan, Hao Han, Qun Li, Daoxu Chen. Efficient Tag Identification in Mobile RFID Systems. In Proceeding of IEEE International Conference INFOCOM 2010.
Bo Sheng, Qun Li, and Weizhen Mao. **Efficient Continuous Scanning in RFID Systems.** IEEE Infocom, San Diego, CA, Mar. 15-19, 2010
Authentication over RFID tags

Lei Yang, Jinsong Han, Yong Qi, Yunhao Liu. **Identification-Free Batch Authentication for RFID Tags.** In *Proc. of ICNP*, 2010.
RFID Tag Identification


RFID Tag Polling


RFID Tag Size Estimation

课程作业1（读书报告）

- 可选择其中之一的课题对相关论文进行阅读，并完成读书报告。要求对论文中的算法、协议以及相关内容进行总结、归纳。鼓励提出创新的研究思路和解决方案。
  - RFID标签识别机制-冲突以及防冲突算法研究
  - RFID标签轮询机制研究
  - RFID标签数目估算机制研究
- 具体要求与论文列表详见课程主页
  - http://cs.nju.edu.cn/lxie/IOT.htm