Retwork: Exploring Reader Network with a COTS RFID System

Jia Liu, Xingyu Chen, Shigang Chen, Wei Wang, Dong Jiang, Lijun Chen
What is RFID?

- Access Control
- Supply Chain

RFID System

Computer

Antenna

Reader

Tag
Background

• Network graph: \( G = (V, E) \)
• Vertex set: \( V = \{V1, V2, V3\} \)
• Edge set: \( E = \{(V2,V3)\} \)

An edge exists if and only if at least one tag is located at the overlapped read zone.
Background

Usage: Anti-Collision

RFID Tags

Tags under collision cannot be read!
Our goal

**System design**

- **Time efficiency**
  The proposed system must be able to get the reader network as soon as possible.

**Gen2-Compatibility**

The proposed system must be Gen2-Compatible and work directly in commercial RFID systems.
With tag inventory, each reader can learn its neighbors by comparing its own tag list with others'.
System design

Basic idea

Phase 1: Over-the-air Writing

RFID Tag  📈  RFID Reader

Reader Vi

Reader Vj

Index i

Index j

Index i

Index j

Index i

Index j
Phase 2: Selective Reading

If the reader \( v_i \) detects any tags in the field-of-view, \( v_i \) and \( v_j \) are neighbors.
Challenge: One-to-many Write is out of Gen2

Phase 1: Over-the-air Writing

02
Related functions in EPCglobal Gen2 standard (Gen2) \cite{1}:

- F1: Inventoried flag.
- F2: Select Command.
- F3: Query Command.

\cite{1} GS1 EPCglobal. EPC radio-frequency identity protocols generation-2 UHF RFID version 2.0.1, 2015.
Writing a reader’s ID to all tags in its zone at once is not allowed.

A reader can set all tags’ inventory flag at once.

One to many write VS Inventoried flag
Select command

- **Fields of Select command:**

  - **Target = 1**
  - **Action = 0**
  - **MemBank = 1**
  - **Pointer = 2**
  - **Length = 2**
  - **Mask = 10**

Set **inventoried flags** to either **A** or **B**
Query command

Fields of Query command:

- **Cmd Code**
- **Target**
- **Sel**
- **Session**

Inventoried flag A (0) or B (1)

**Query**

- Target = 1
- Session = 1
- Sel = 0

The tags with **Target** state B will reply to the reader.
1. Vi: set all tags to A.
Detection of contention link

1. Vi: set all tags to A.

2. Vj: set all tags to B.

Flag = B : $S(2, 4, 1, 0, 0, 0)$
Detection of contention link

1. Vi: set all tags to A.
2. Vj: set all tags to B.

Vi queries B: \( Q(0, 2, 1) \)
System design

Detection of contention link

1. Vi: set all tags to A.

2. Vj: set all tags to B.


4. Check any two readers
System design

Gen2-Compatible Commands

\[
S(t, a, b, p, l, k),
\]

\[
v_i : S(2, 0, 1, 0, 0, 0)
\]

Target = 2: Select flag 2
Action = 0: Matching tag to A, nonmatching tag to B
Bitmask = (1,0,0,0): All tags are matching
System design

Gen2-Compatible Commands

\[ S( t, a, b, p, l, k ) , \]

Target = 2: Select flag 2
Action = 4: Matching tag to B, nonmatching tag to A
Bitmask = (1,0,0,0): All tags are matching
System design

Gen2-Compatible Commands

\[ S(t, a, b, p, l, k) \]
(Target, MemBank, Pointer, Length, Mask)

\[ Q(e, s, g) \]
(Sel, Session, Target)

01. \[ v_i : S(2, 0, 1, 0, 0, 0) \]
02. \[ v_j : S(2, 4, 1, 0, 0, 0) \]
03. \[ v_i : Q(0, 2, 1) \]

Sel = 0: Query according to Inventory flag
Session = 2: Choose flag 2
Target = 1: Tags with flag B reply
System design

Gen2-Compatible Commands

\[ S(t, a, b, p, l, k), \]
\[ Q(e, s, g). \]

For each pair of \( v_i \) and \( v_j \) Reader network

01 \( v_i : S(2,0,1,0,0,0) \)
02 \( v_j : S(2,4,1,0,0,0) \)
03 \( v_i : Q(0,2,1) \)
System design

Identification of reader network

\[(m-1)\text{ flag}\rightarrow A\]

\[1 \text{ flag}\rightarrow B\]

\[(m-1)\text{ query } B\]

<table>
<thead>
<tr>
<th>Methods</th>
<th>Basic Network</th>
<th>Enhanced Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of commands</td>
<td>(3(m-1) \approx 3m)</td>
<td>(2(m-1)+1 \approx 2m)</td>
</tr>
</tbody>
</table>
An RFID reader is redundant if all tags in its read zone are covered by other readers.

Redundancy detection is to identify which readers are redundant in a multi-reader RFID system.
System design

Case study: redundancy detection

01 Vi: set all tags to A.
   \[ v_i : S(2, 0, 1, 0, 0, 0) \]

02 V1-Vm: set all tags to B.
   \[ v_j : S(2, 4, 1, 0, 0, 0) \]

03 Vi: query A tags.
   \[ v_i : Q(0, 2, 0) \]
System design

Case study: redundancy detection

01 Vi: set all \textbf{tags to A}. \( v_i : S(2, 0, 1, 0, 0, 0) \)

02 V1-Vm: set all \textbf{tags to B}. \( v_j : S(2, 4, 1, 0, 0, 0) \)

03 Vi: query \textbf{A tags}. \( v_i : Q(0, 2, 0) \)
System design

Case study: redundancy detection

A reader is redundant if it cannot detect tag with flag A.

01 Vi: set all tags to A.

02 V1-Vm: set all tags to B.

03 Vi: query A tags.

Vi: set all tags to A.

$v_i : S(2, 0, 1, 0, 0, 0)$

V1-Vm: set all tags to B.

$v_j : S(2, 4, 1, 0, 0, 0)$

Vi: query A tags.

$v_i : Q(0, 2, 0)$
**Evaluation**

Figure 1: RFID readers and tags.

(a) Readers

(b) Tags

Figure 2: System deployment.
Table 1: Gen2-compatibility on Readers and Tags.

<table>
<thead>
<tr>
<th>Functions</th>
<th>ThingMagic</th>
<th>Impinj</th>
<th>Alien</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mercury6</td>
<td>R220</td>
<td>F800</td>
</tr>
<tr>
<td>Select</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Query</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SL</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Flag(A/B)</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>R420</th>
<th>9900+</th>
<th>Impinj Monza</th>
<th>Alien Higgs™</th>
</tr>
</thead>
<tbody>
<tr>
<td>Readers</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Tags</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Retwork functions properly on 4 reader models + **ALL** tag models
Evaluation

Retwork vs. Inventory

Retwork reduces the time from 55.9s to 3.9s, $14.7 \times$ performance gain

Figure 3: Time comparison between Retwork and tag inventory.
Evaluation

**Figure 4: Accuracy.**

- **FPR** < 3.5%
- **FNR** < 0.3%
Retwork reduces the time from 40.2s to 4.1s
An order of magnitude

Figure 5: Time Efficiency for redundancy detection.
Conclusion

01
We propose an efficient solution Retwork to the practically important problem of identifying the contention relationship among multiple readers in a large RFID system.

02
Our protocol exploits the flag-setting capability in Gen2. With a carefully-designed series of flag-flipping operations, it is able to check two readers are neighbors or not.

03
We implement a prototype of Retwork with 8,000 tags. Extensive experiments show that it can improve the time efficiency by an order of magnitude.