Retwork: Exploring Reader Network with a COTS RFID System

Jia Liu, Xingyu Chen, Shigang Chen, Wei Wang,
Dong Jiang, Lijun Chen

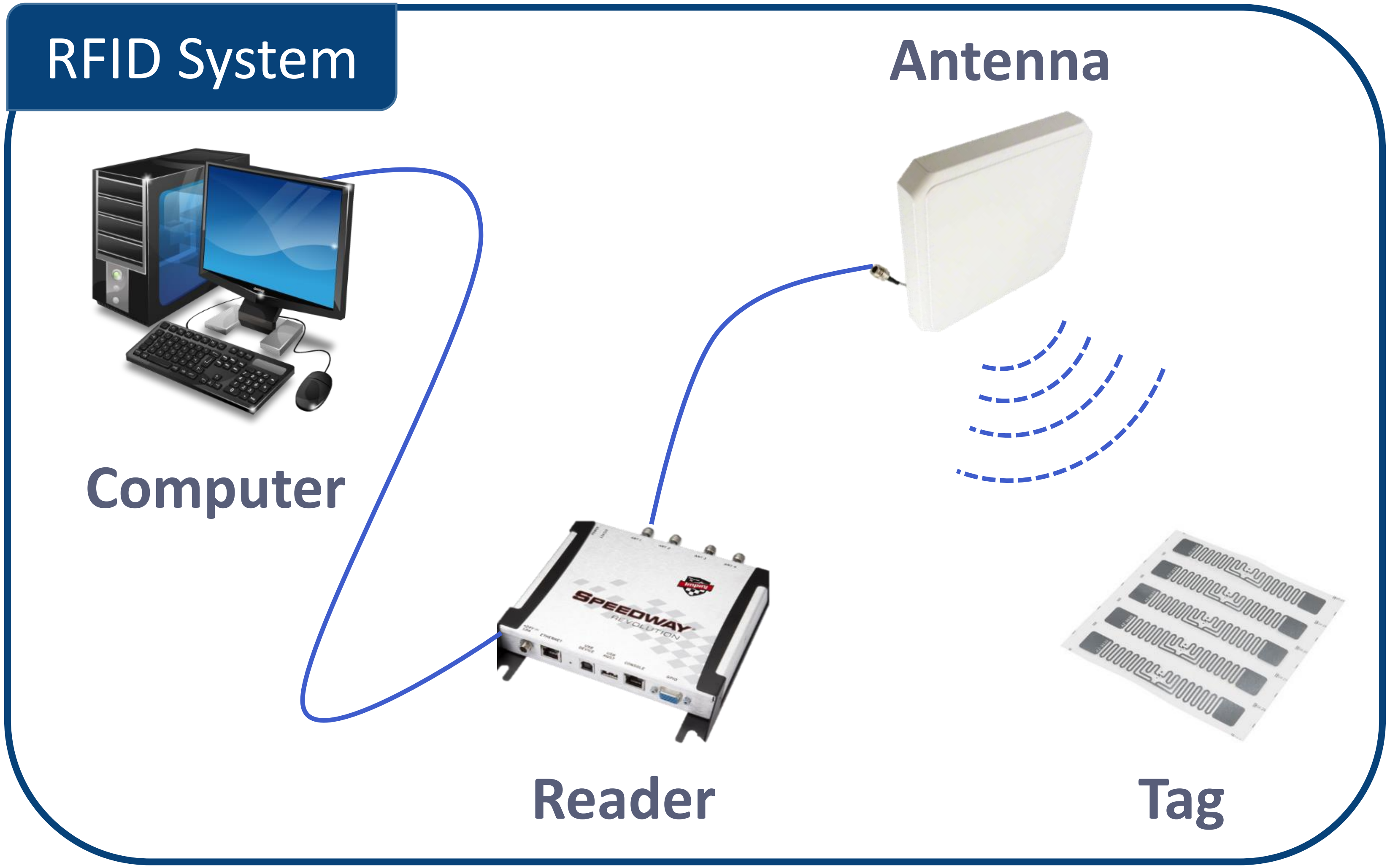
南京大學 NANJING UNIVERSITY

UF UNIVERSITY of FLORIDA

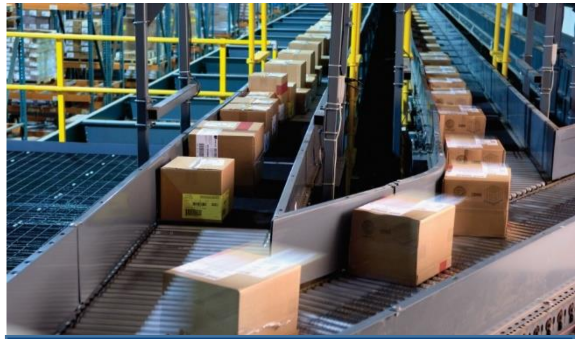
CONTENT

01 Background

What is RFID?



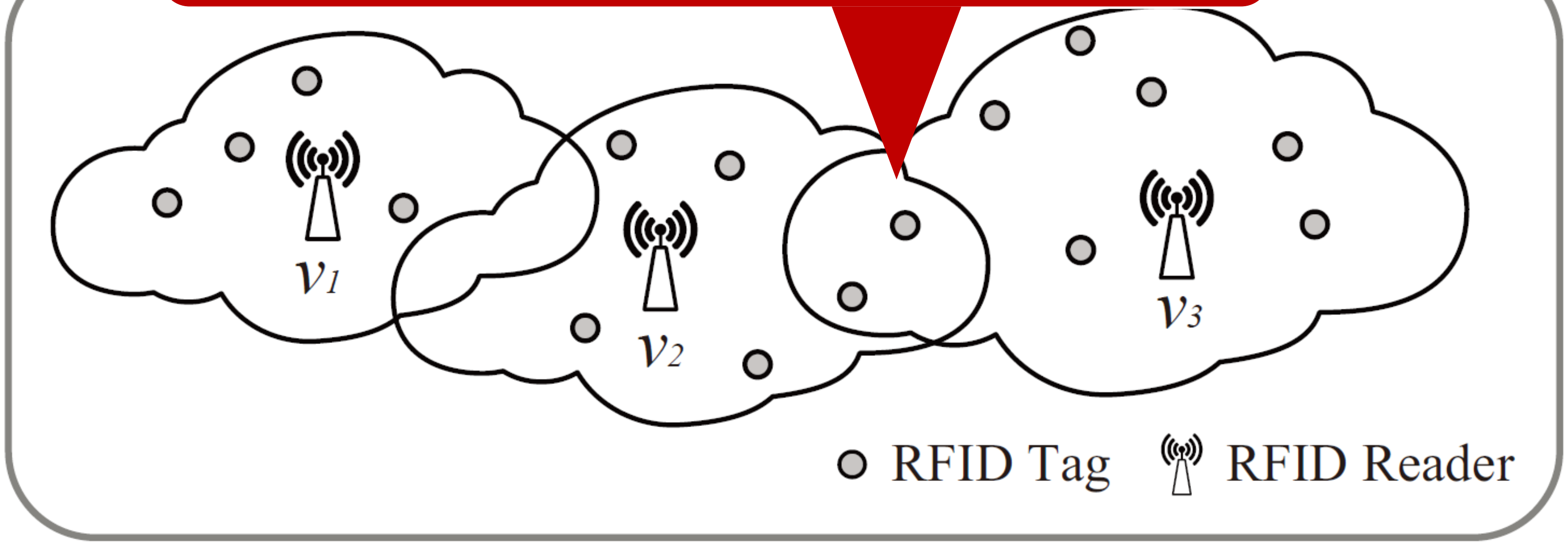

Access Control


Supply Chain

01 Background

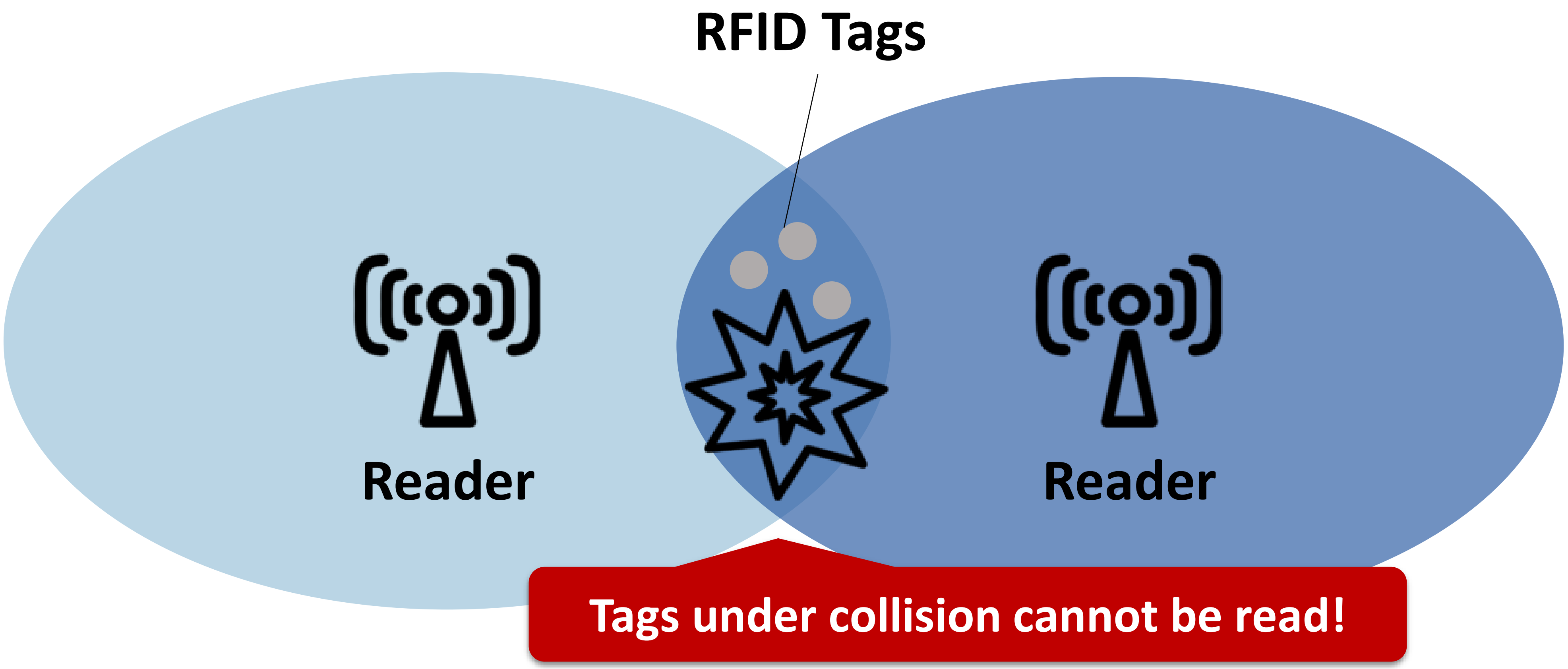
Reader Network

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



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

Usage: Anti-Collision



02

System design

Our goal

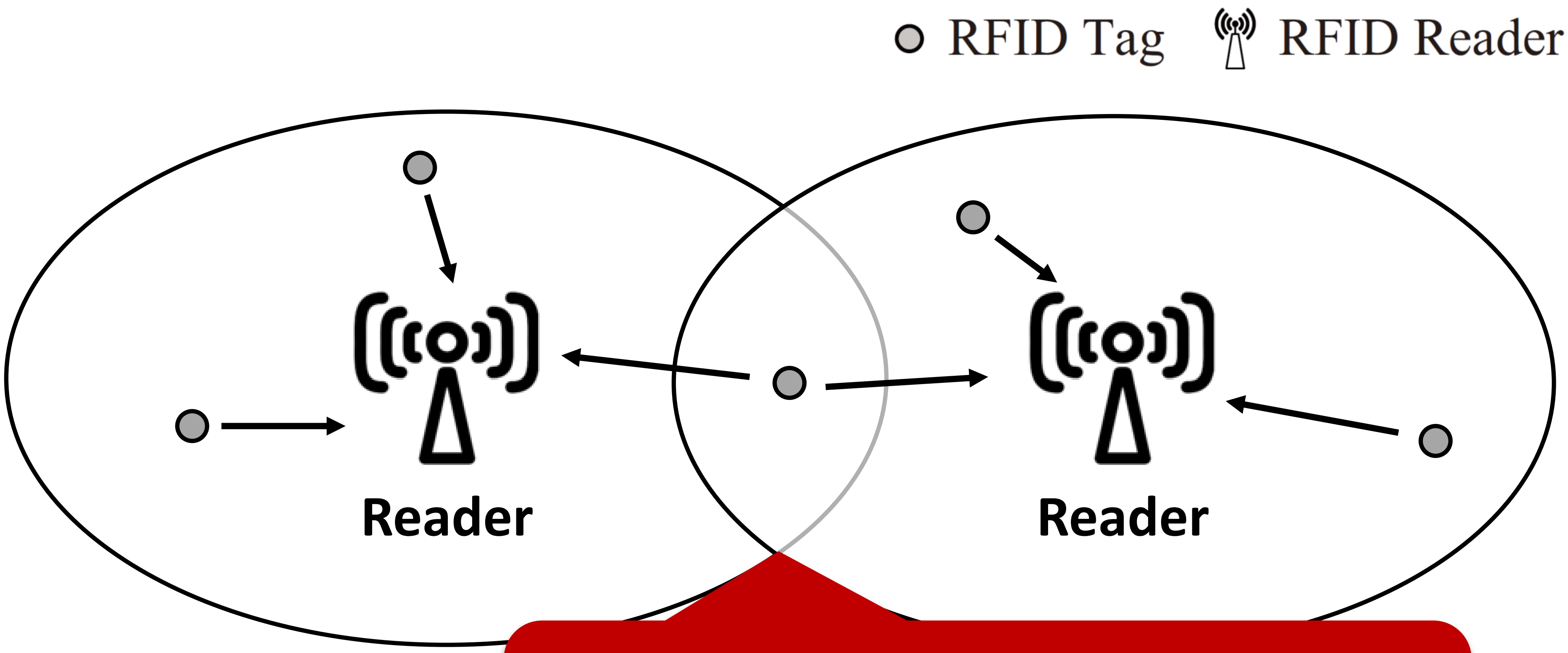
Time efficiency

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.

Naïve solution



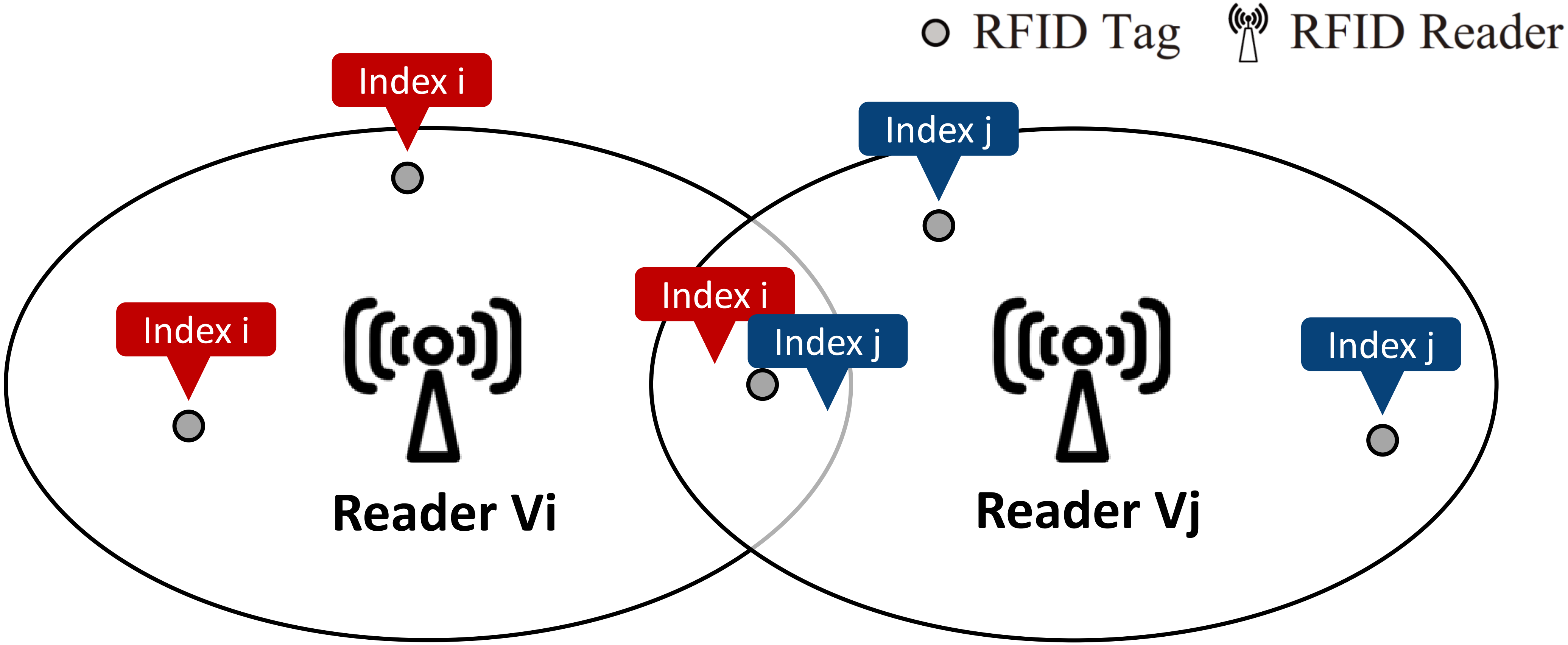
With tag inventory, each reader can learn its neighbors by comparing its own tag list with others'.

02 System design

Basic idea

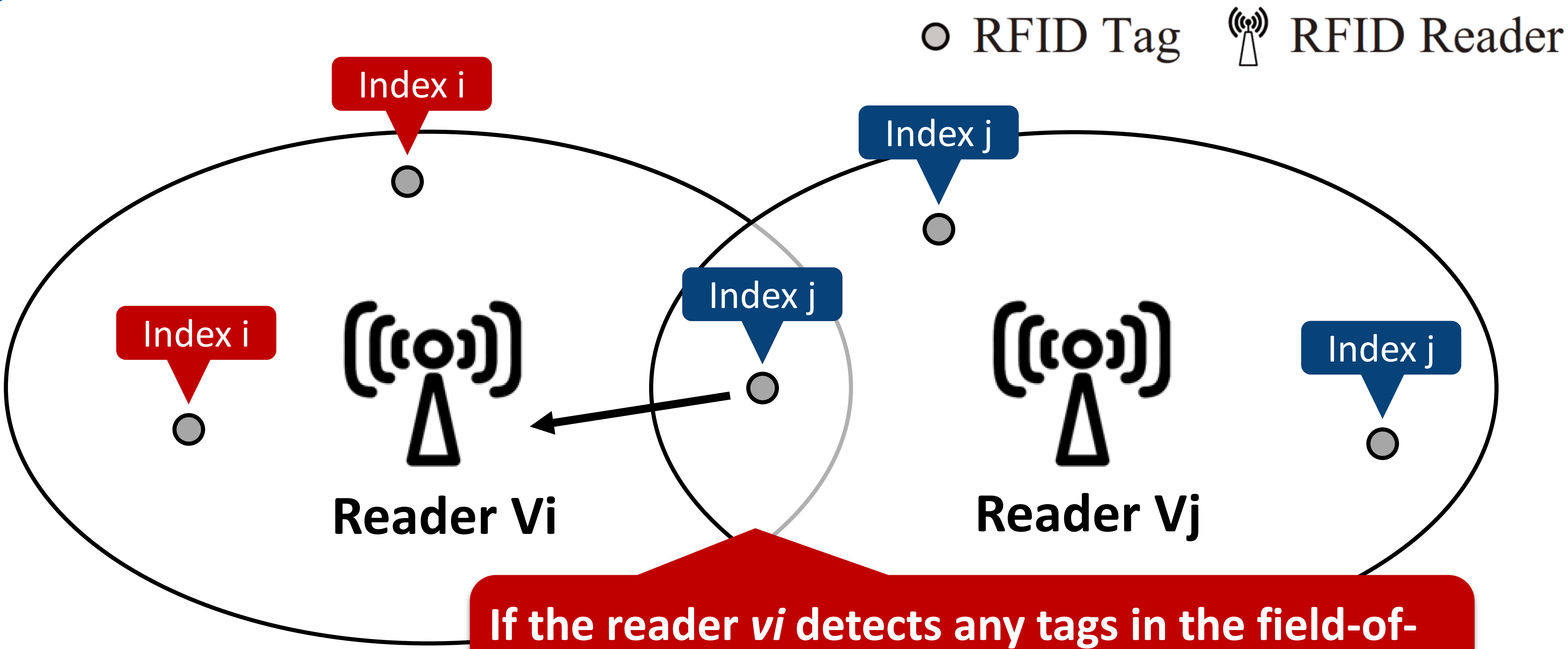
idea!!!

Phase 1 : Over-the-air Writing



Basic idea

Phase 2 : Selective Reading

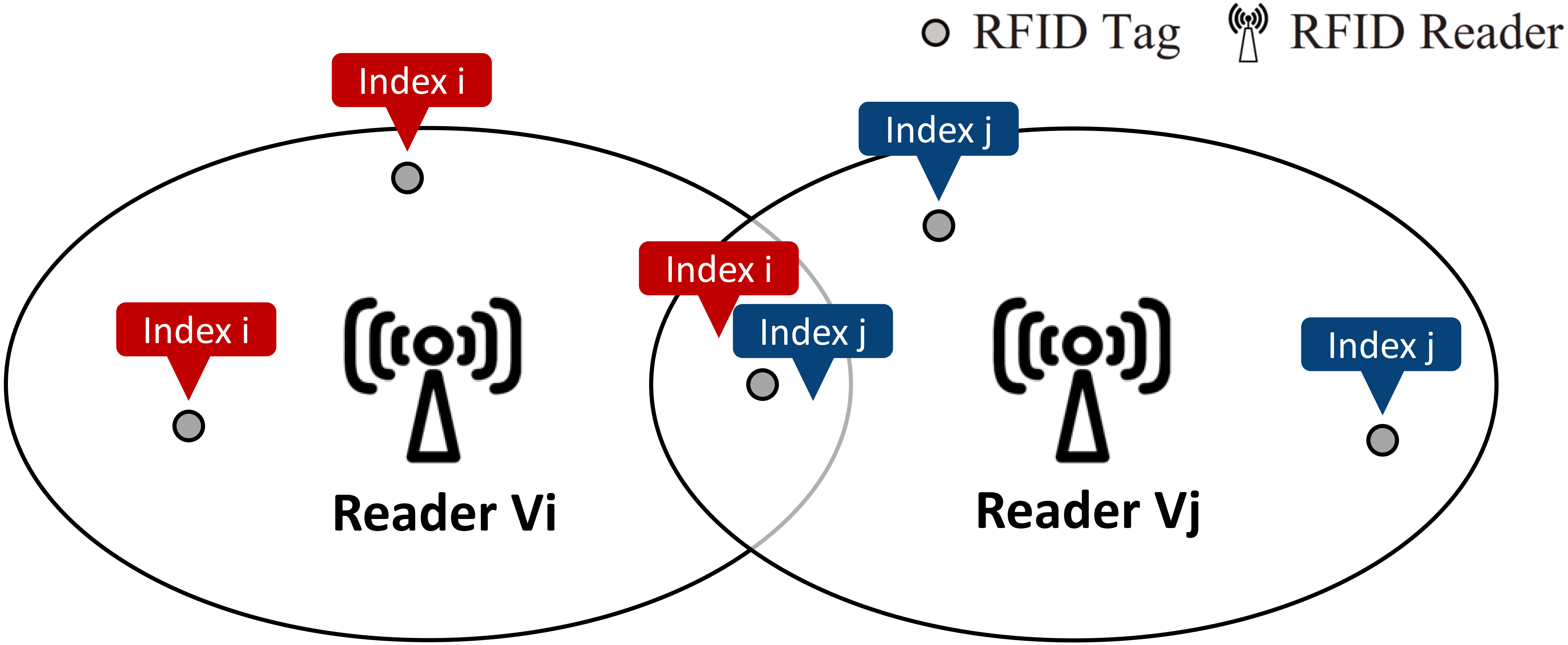


If the reader *vi* detects any tags in the field-of-view, *vi* and *vj* are neighbors.

02 Challenge: One-to-many Write is out of Gen2

idea!!!

Phase 1 : Over-the-air Writing



Gen2 protocol

- **Related functions in EPCglobal Gen2 standard (Gen2) [1] :**

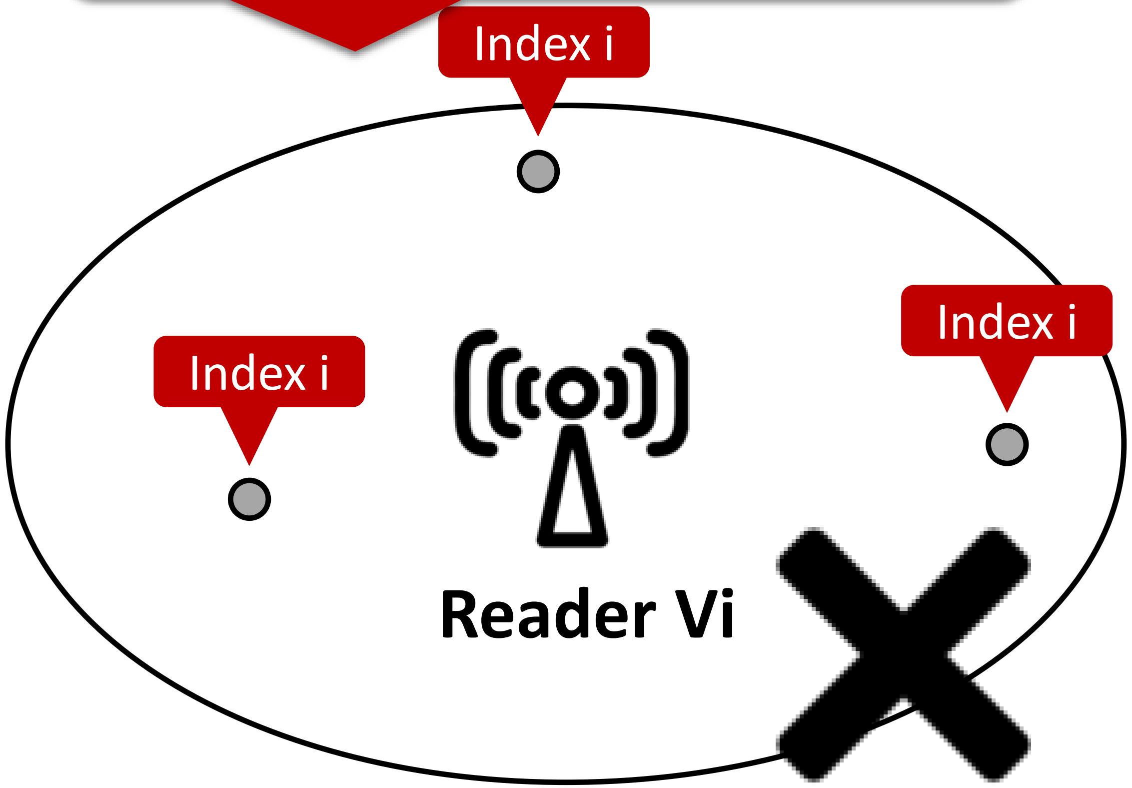
- F1: Inventoried flag.
- F2: Select Command.
- F3: Query Command.

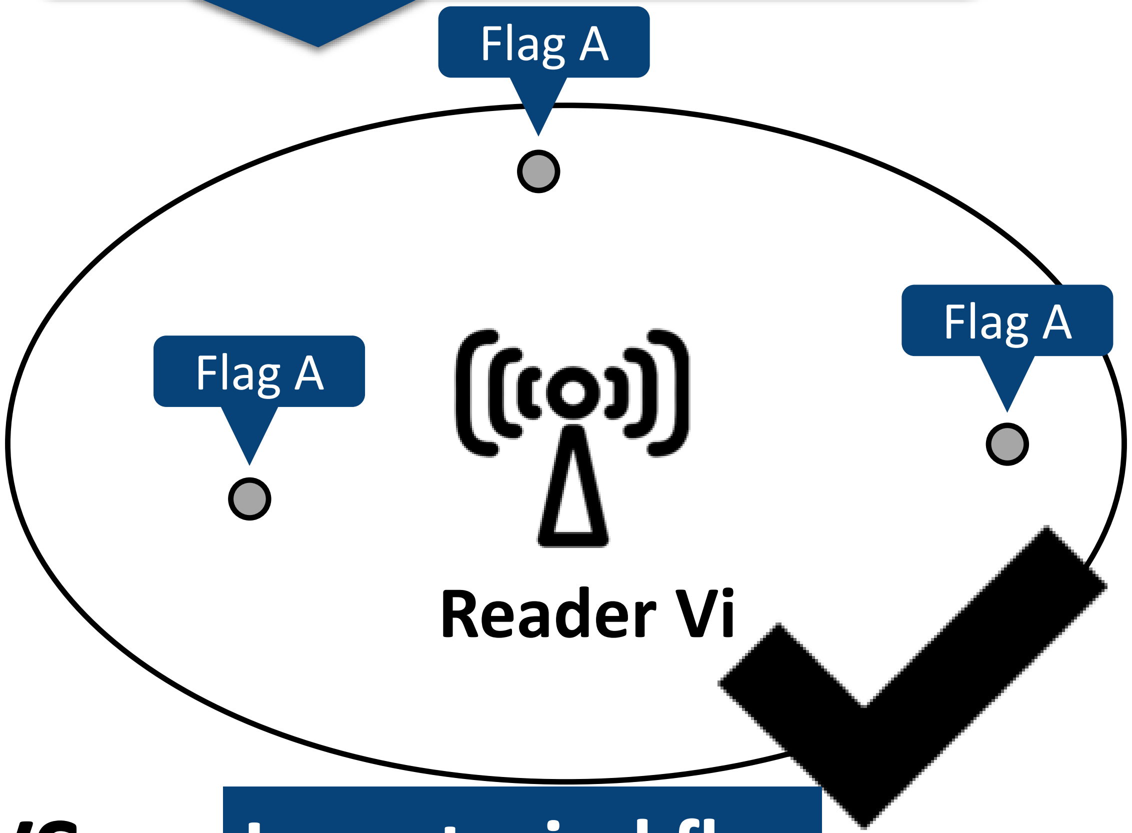
[1] GS1 EPCglobal. EPC radio-frequency identity protocols generation-2 UHF RFID version 2.0.1, 2015.

Inventoried flag

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.

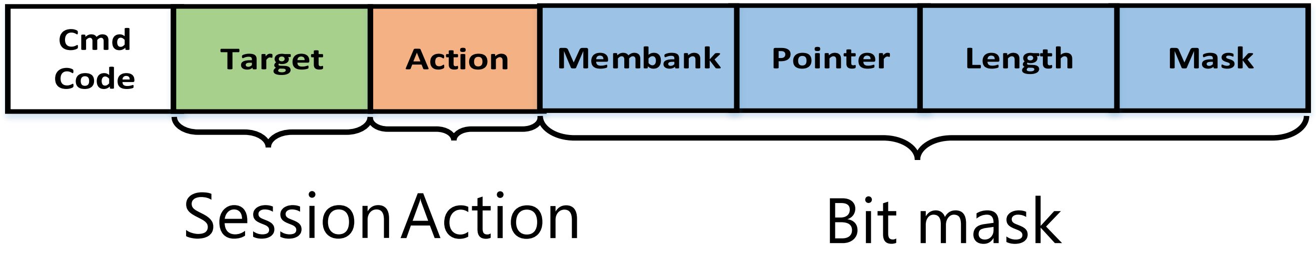




One to many write **VS** **Inventoried flag**

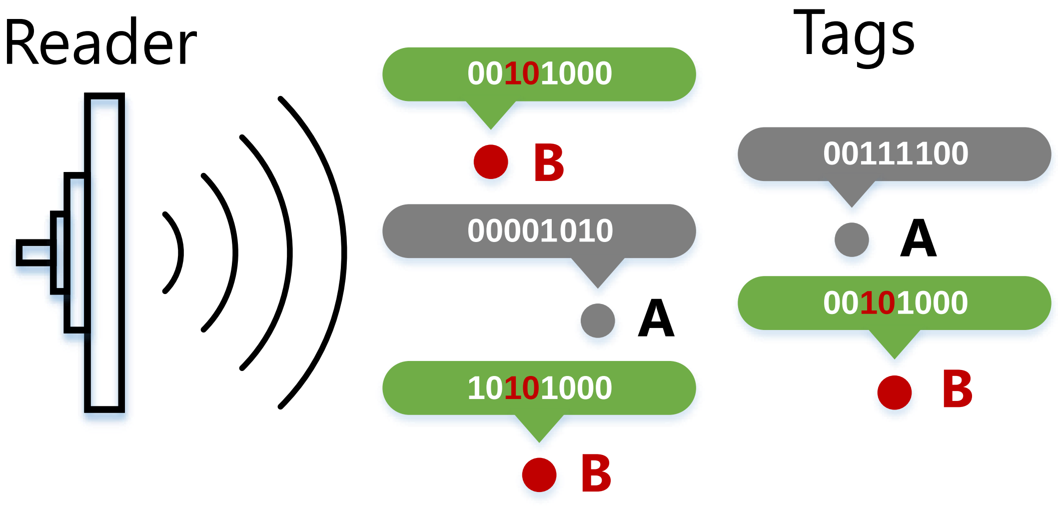
Select command

- **Fields of Select command:**



Select

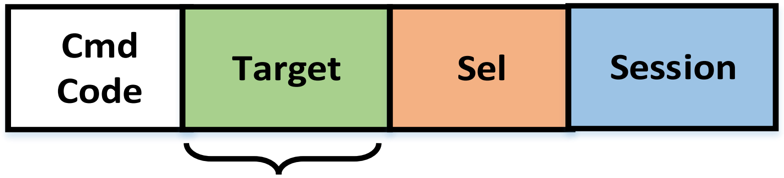
Target = 1
Action = 0
MemBank = 1
Pointer = 2
Length = 2
Mask = 10



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

Query command

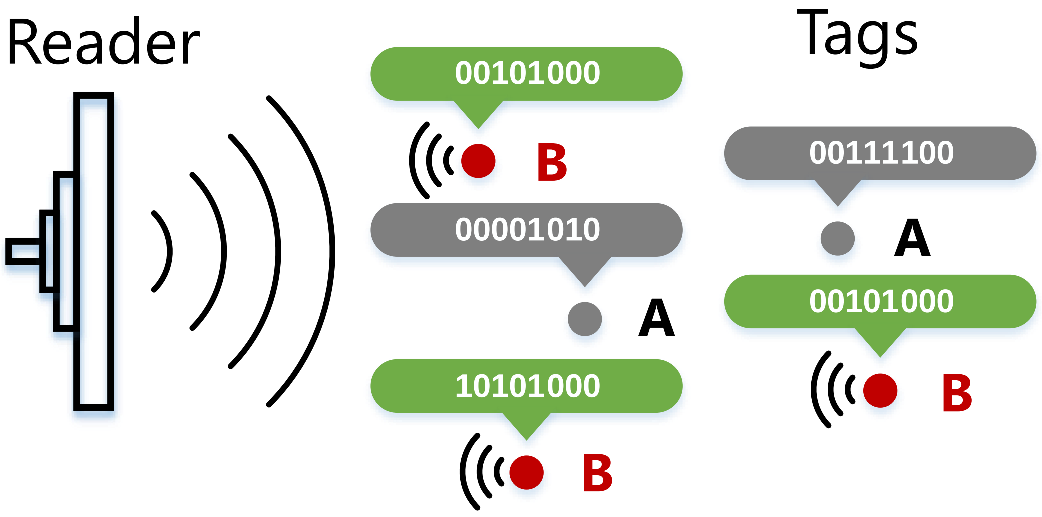
- **Fields of Query command:**



Inventoried flag A (0) or B (1)

Query

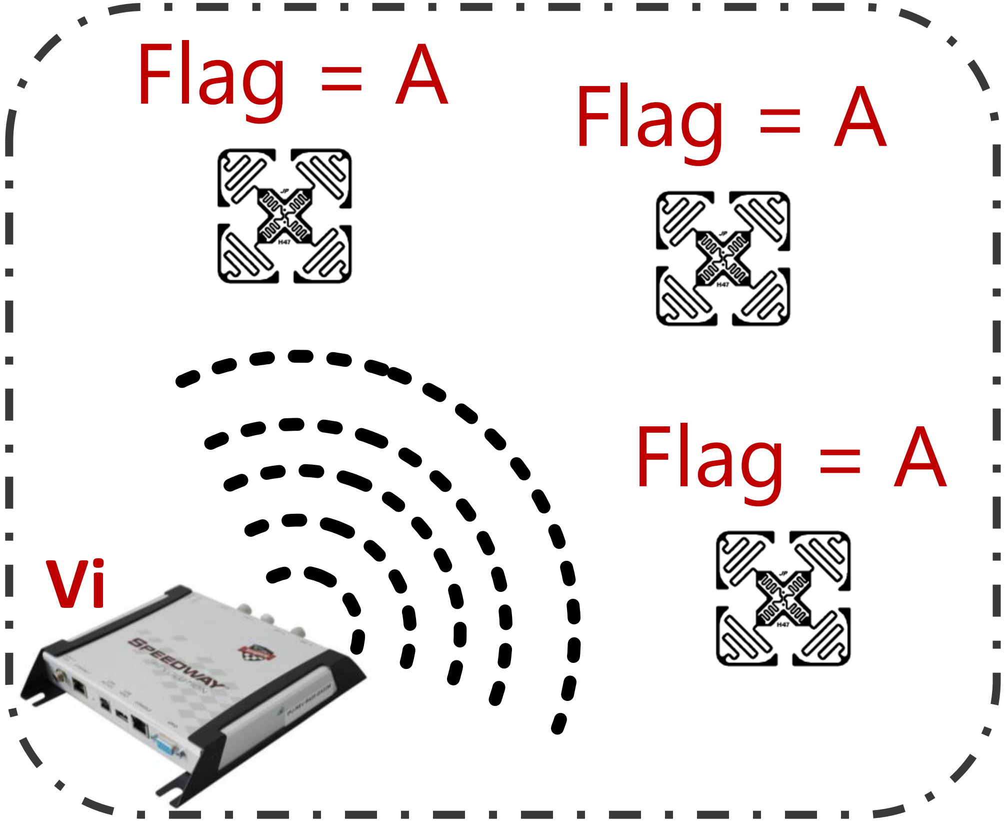
Target = 1
Session = 1
Sel = 0



The tags with **Target** state **B** will reply to the reader

02 System design

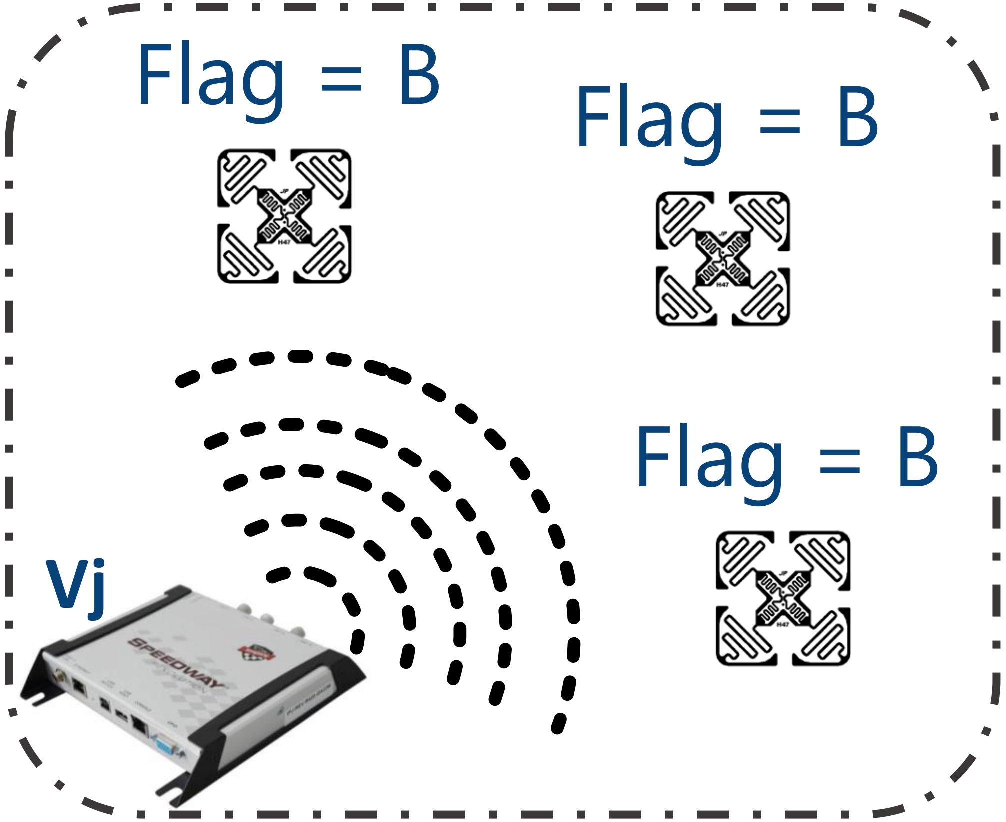
Detection of contention link



$$Flag = A : \mathcal{S}(2,0,1,0,0,0)$$

1. Vi: set all tags to A.

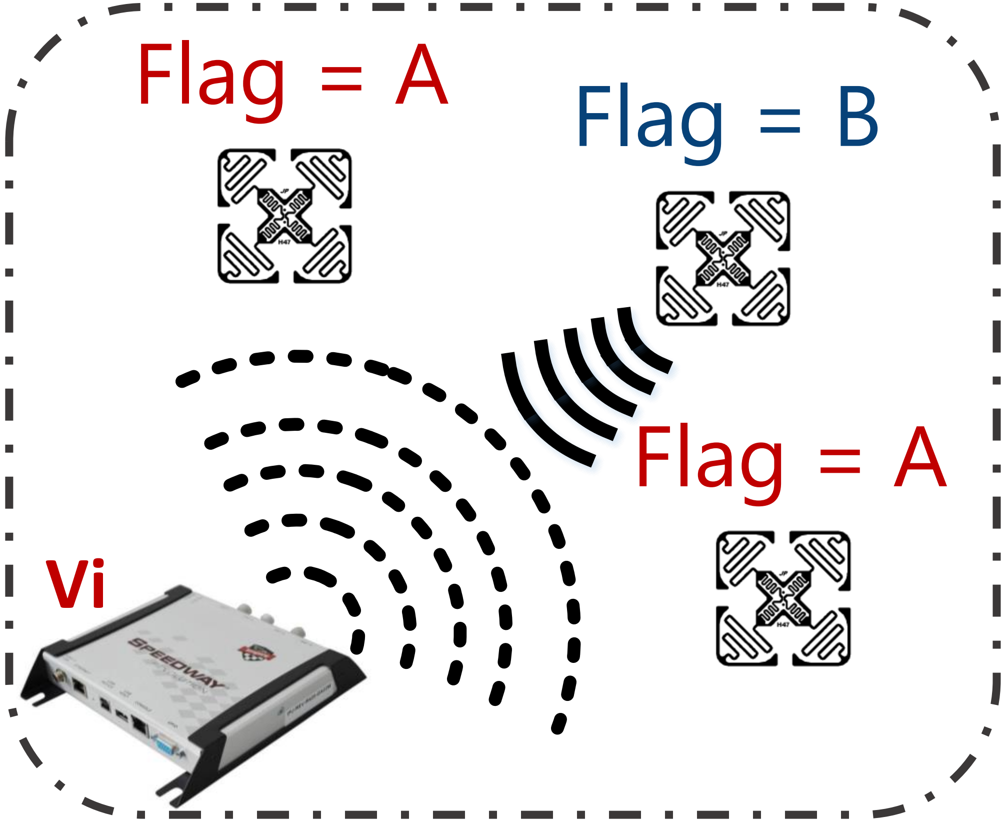
Detection of contention link



$$Flag = B : S(2,4,1,0,0,0)$$

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

Detection of contention link

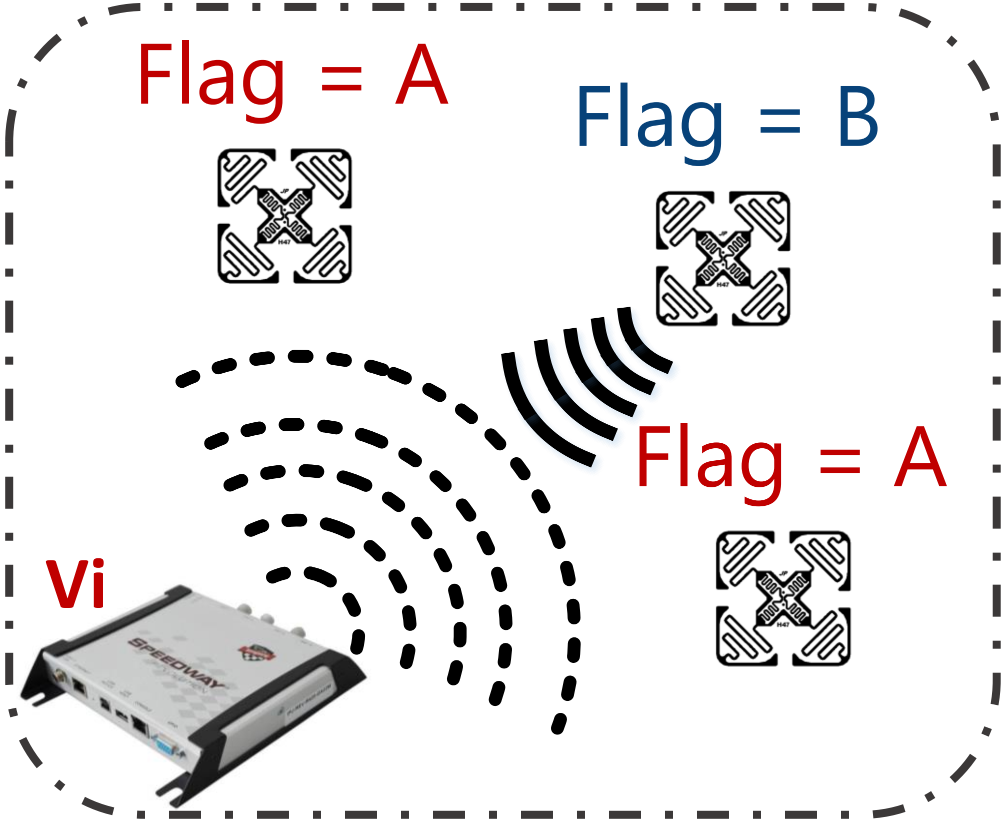


Vi queries B: $Q(0, 2, 1)$

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

02 System design

Detection of contention link



Reader network is formed.

1. Vi: set all tags to A.
2. Vj: set all tags to B.
3. Vi: queries B tags.
4. Check any two readers

Gen2-Compatible Commands

$$\mathcal{S}(\underbrace{t}_{\text{Target}}, \overbrace{a}^{\text{Action}}, \underbrace{b}_{\text{MemBank}}, \overbrace{p}^{\text{Pointer}}, \underbrace{l}_{\text{Length}}, \overbrace{k}^{\text{Mask}}),$$

01 $v_i: \ \mathcal{S}(2,0,1,0,0,0)$

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

02 System design

Gen2-Compatible Commands

$$\mathcal{S}(\underbrace{t}_{\text{Target}}, \overbrace{a}^{\text{Action}}, \underbrace{b}_{\text{MemBank}}, \overbrace{p}^{\text{Pointer}}, \underbrace{l}_{\text{Length}}, \overbrace{k}^{\text{Mask}}),$$

01 $v_i: \ \mathcal{S}(2,0,1,0,0,0)$

02 $v_j: \ \mathcal{S}(2,4,1,0,0,0)$

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

Gen2-Compatible Commands

$$\mathcal{S}(\underbrace{t}_{\texttt{Target}},\overbrace{a}^{\texttt{Action}},\underbrace{b}_{\texttt{MemBank}},\overbrace{p}^{\texttt{Pointer}},\underbrace{l}_{\texttt{Length}},\overbrace{k}^{\texttt{Mask}}),\quad Q(\underbrace{e}_{\texttt{Sel}},\overbrace{s}^{\texttt{Session}},\underbrace{g}_{\texttt{Target}}).$$

01 $v_i:\ \mathcal{S}(2,0,1,0,0,0)$

02 $v_j:\ \mathcal{S}(2,4,1,0,0,0)$

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**

02 System design

Gen2-Compatible Commands

$$\mathcal{S}(\underbrace{t}_{\text{Target}}, \overbrace{a}^{\text{Action}}, \underbrace{b}_{\text{MemBank}}, \overbrace{p}^{\text{Pointer}}, \underbrace{l}_{\text{Length}}, \overbrace{k}^{\text{Mask}}), \quad \mathcal{Q}(\underbrace{e}_{\text{Sel}}, \overbrace{s}^{\text{Session}}, \underbrace{g}_{\text{Target}}).$$

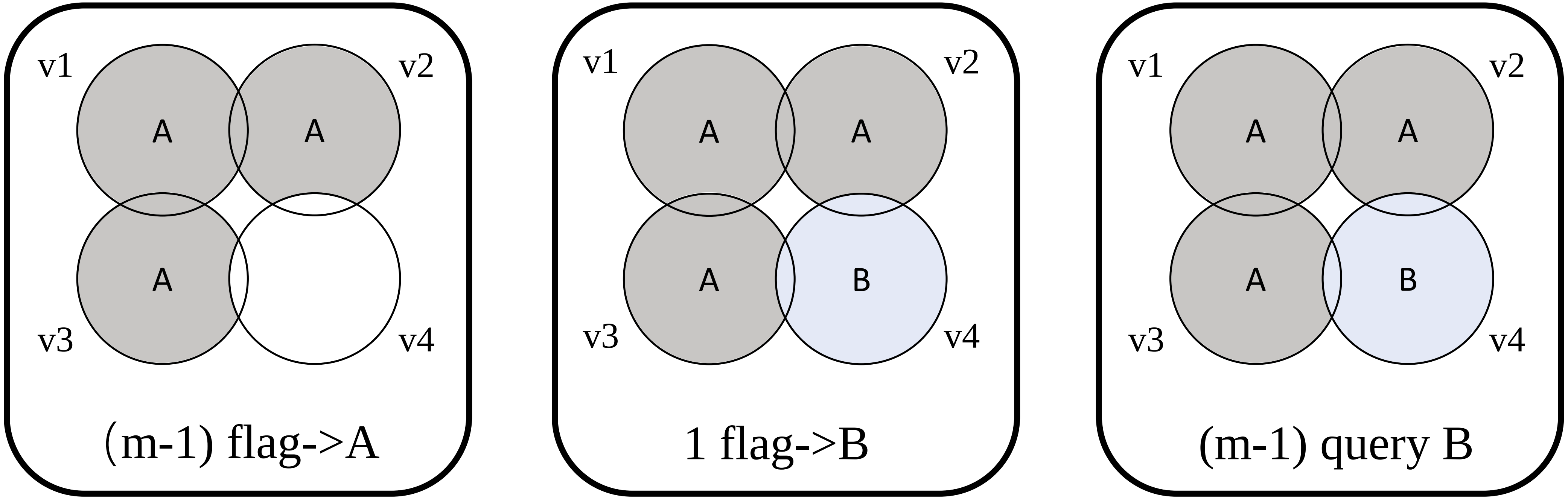
01 $v_i:\ \mathcal{S}(2,0,1,0,0,0)$

02 $v_j:\ \mathcal{S}(2,4,1,0,0,0)$

03 $v_i:\ \mathcal{Q}(0,2,1)$

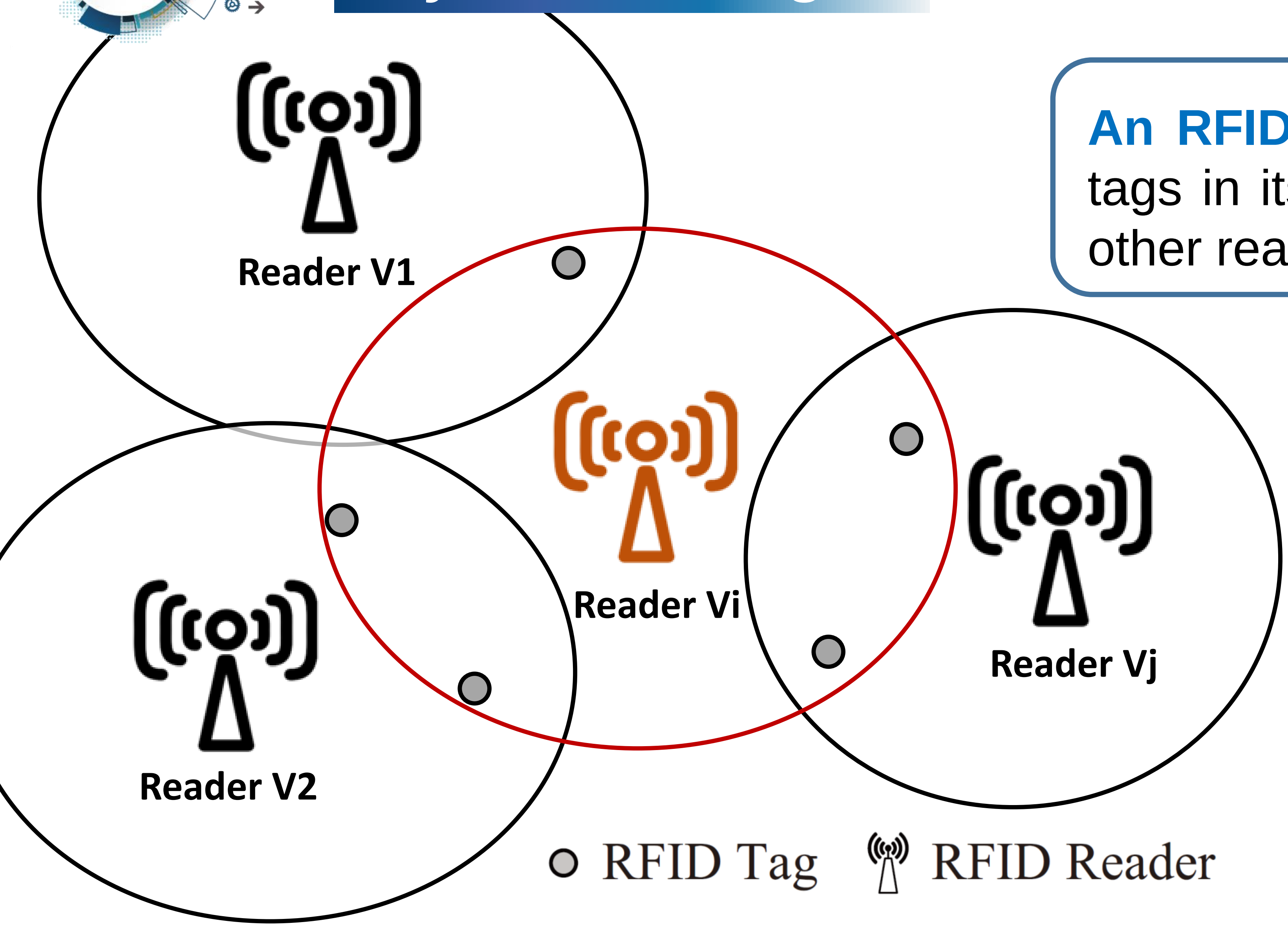
For each pair of vi and vj → Reader network

Identification of reader network



Methods	Basic Retwork	Enhanced Retwork
Number of commands	**3(m-1) ≈ 3m**	2(m-1)+1 ≈ **2m**

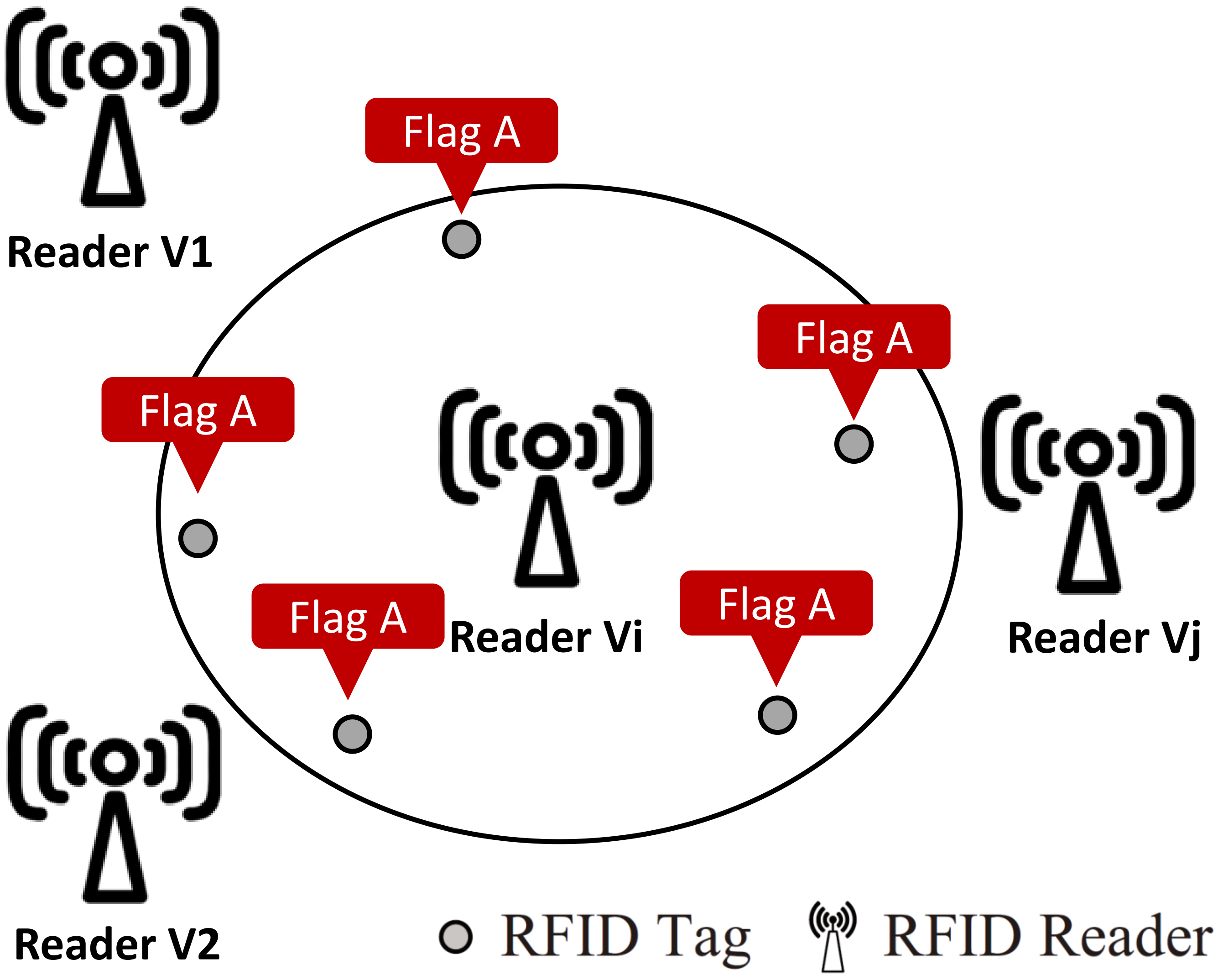
Case study: redundancy detection



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.

Case study: redundancy detection



01 Vi: set all **tags to A**.

$v_i: \mathcal{S}(2,0,1,0,0,0)$

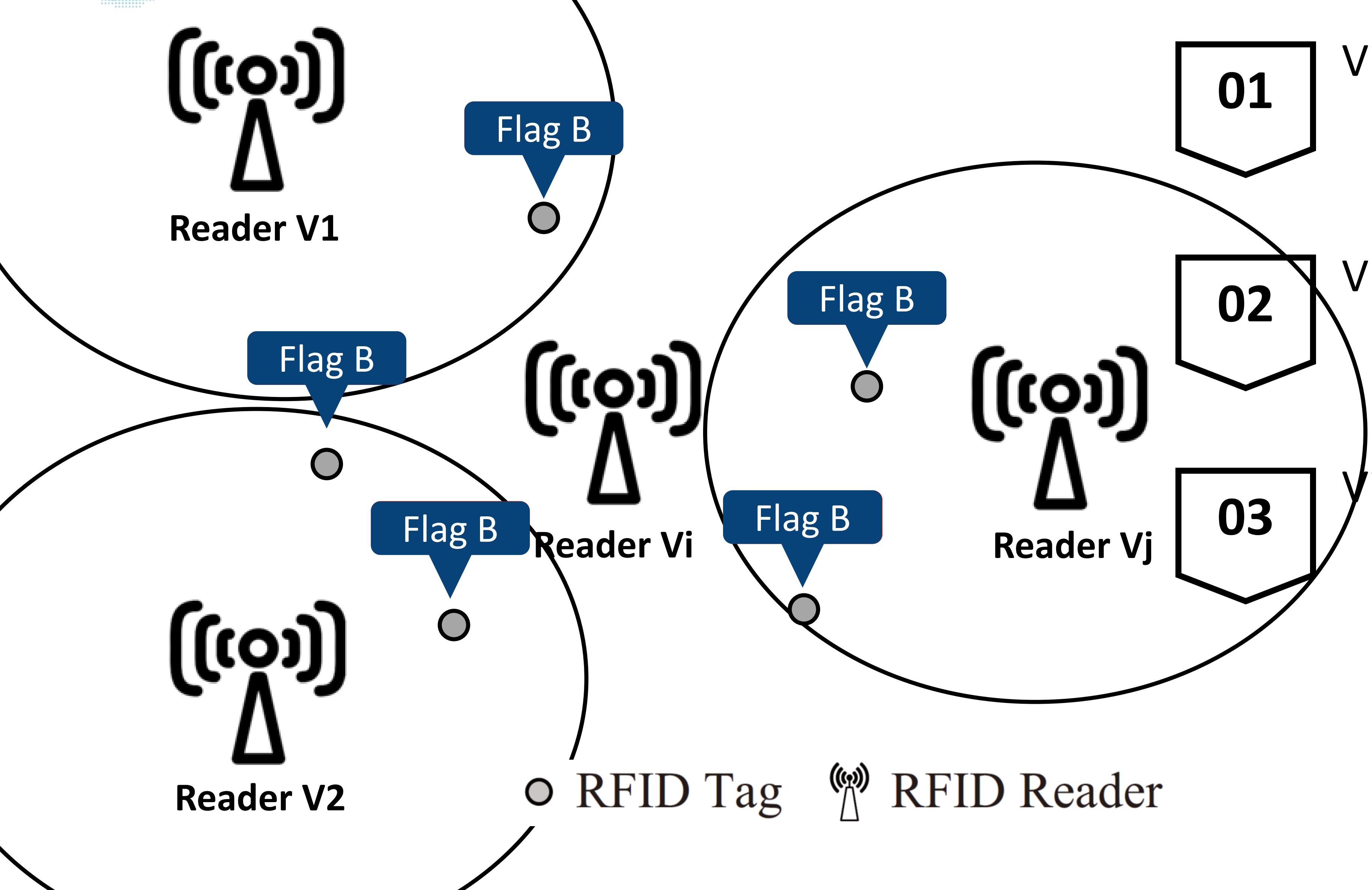
02 V1-Vm: set all **tags to B**.

$v_j: \mathcal{S}(2,4,1,0,0,0)$

03 Vi: query **A tags**.

$v_i: \mathcal{Q}(0,2,0)$

Case study: redundancy detection



01 Vi: set all **tags to A**.

$v_i: \mathcal{S}(2,0,1,0,0,0)$

02 V1-Vm: set all **tags to B**.

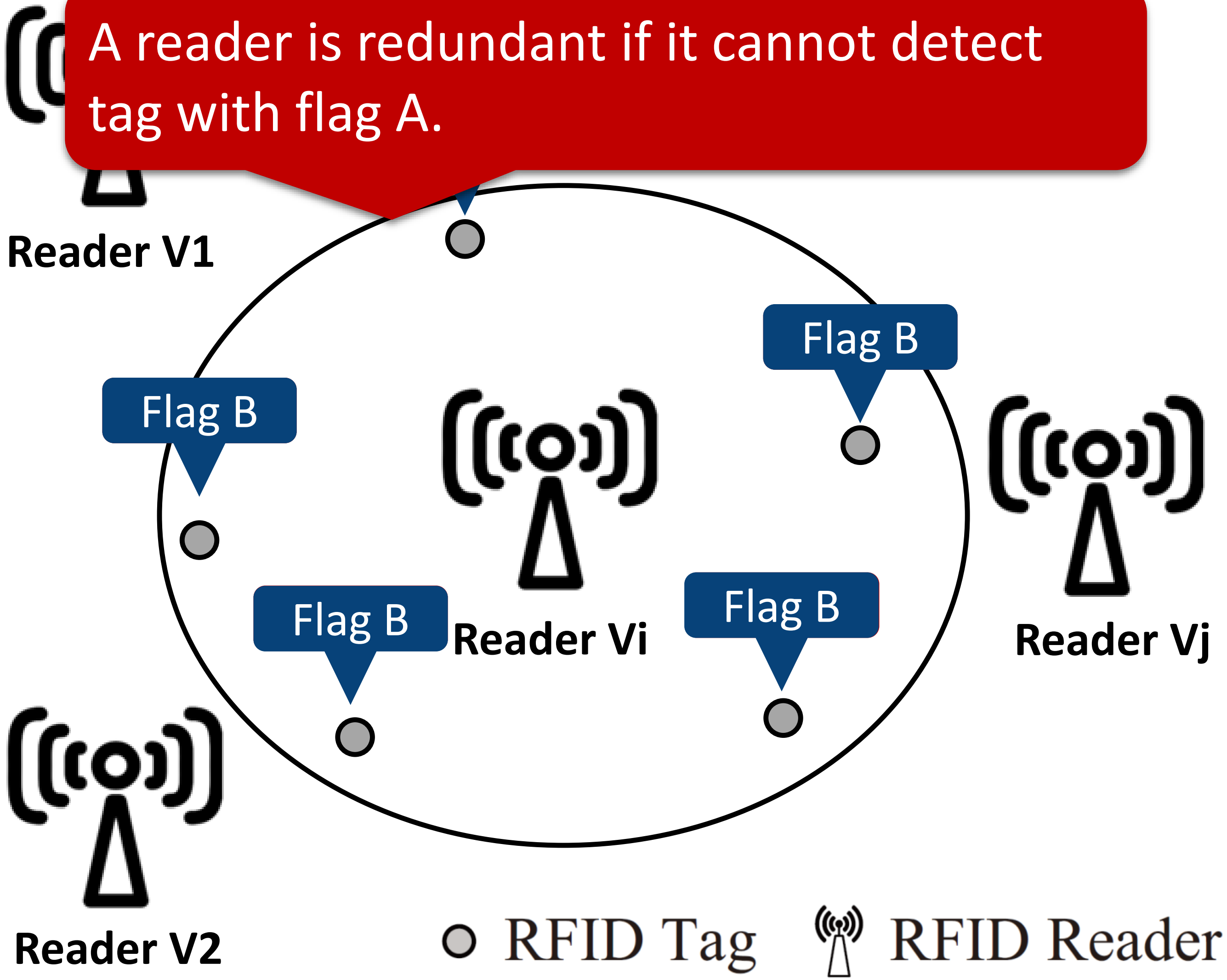
$v_j: \mathcal{S}(2,4,1,0,0,0)$

03 Vi: query **A tags**.

$v_i: \mathcal{Q}(0,2,0)$

Case study: redundancy detection

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



01 Vi: set all **tags to A**.

$$v_i: \ \mathcal{S}(2,0,1,0,0,0)$$

02 V1-Vm: set all **tags to B**.

$$v_j: \ \mathcal{S}(2,4,1,0,0,0)$$

03 Vi: query **A tags**.

$$v_i: \ Q(0,2,0)$$

Experimental setup

(a) Readers (b) Tags

Figure 1: RFID readers and tags.

Figure 2: System deployment.

03

Evaluation

Compatibility

Table 1: Gen2-compatibility on Readers and Tags.

Functions	Readers						Tags							
	ThingMagic		**Impinj**		**Alien**		**Impinj Monza**					**Alien Higgs™**		
	Mercury6	M6e	R220	R420	F800	9900+	4	5	R6	R6-P	R6-C	4	5	EC
Select	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Query	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SL	✓	✓	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Flag(A/B)	✓	✓	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

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

Retwork vs. Inventory

Retwork reduces the time from 55.9s to 3.9s

14.7× performance gain

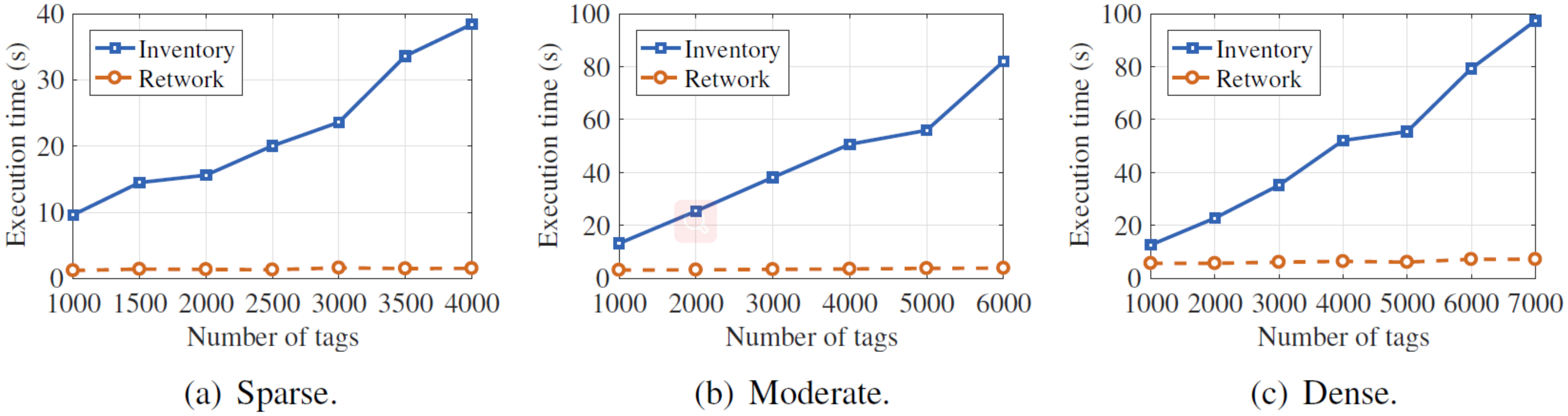


Figure 3: Time comparison between Retwork and tag inventory.

Accuracy

FPR < 3.5%   FNR<0.3%

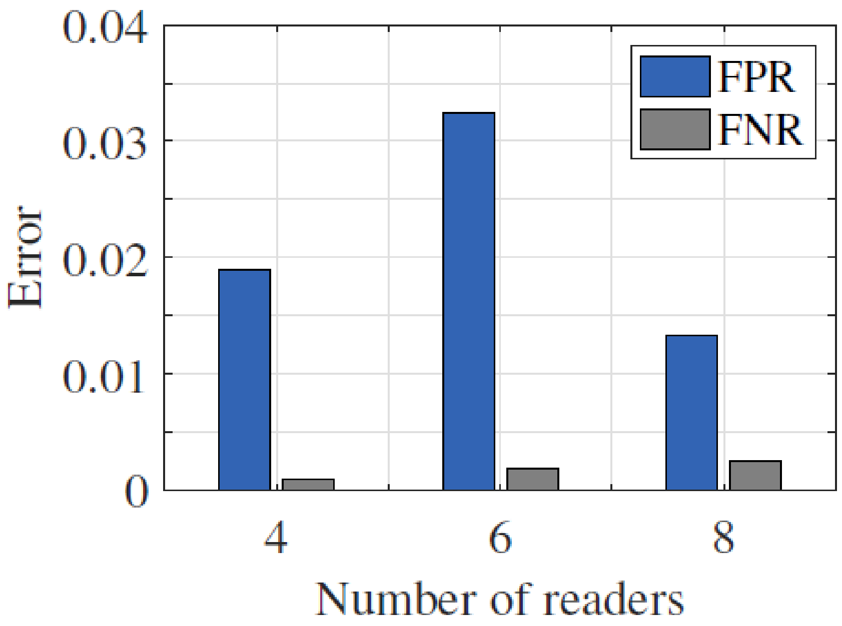


Figure 4: Accuracy.

Reader Redundancy

Retwork reduces the time from 40.2s to 4.1s

An order of magnitude

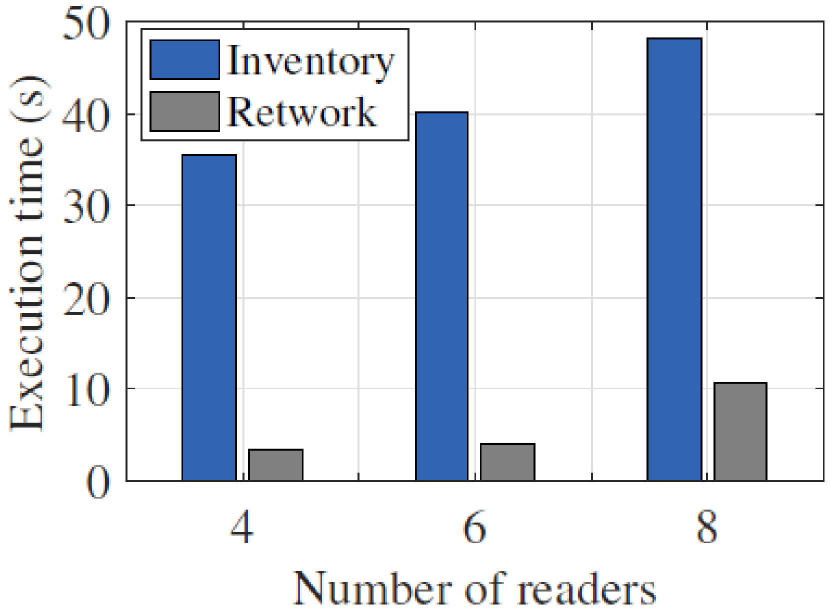


Figure 5: Time Efficiency for redundancy detection.

04

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.

THANK YOU

jialiu@nju.edu.cn