Fast and Complete Unknown Tag Identification in Large RFID Systems

Xuan Liu, Shigeng Zhang, Kai Bu, Bin Xiao

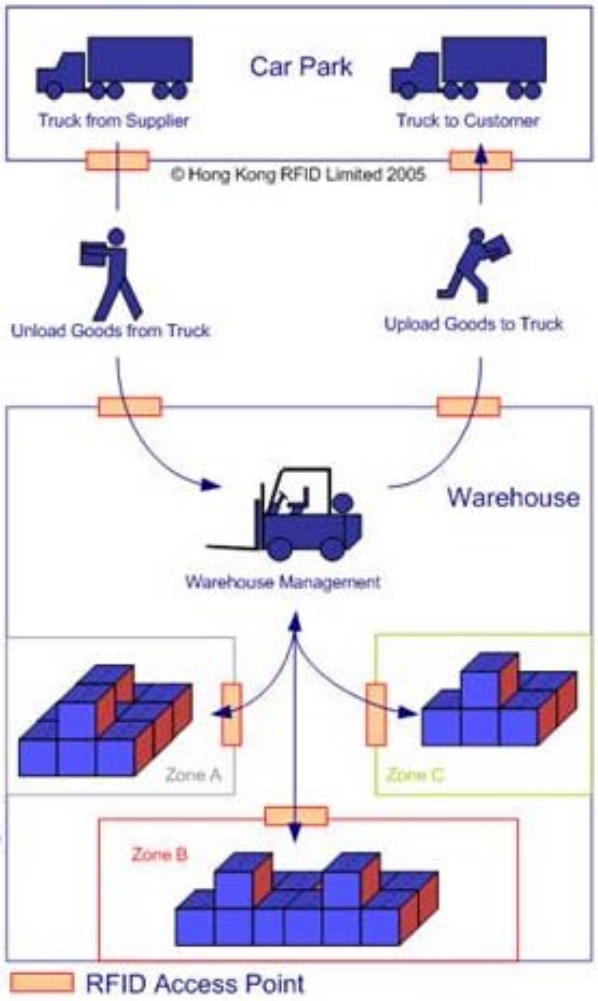
Supervisor: Dr. Bin Xiao
Department of Computing
The Hong Kong Polytechnic University

THE HONG KONG POLYTECHNIC UNIVERSITY

Outline

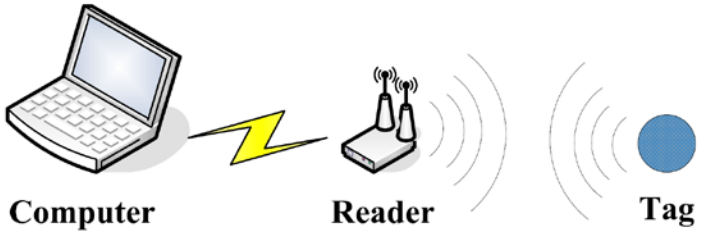
- RFID Background
- Problem Description
- Solutions
  - Baseline and Two Extended Protocols
- Simulation Results

Applications



Background: What is RFID?

- RFID is a technology that identifies objects remotely by radio waves
- An RFID system
  - RFID Readers
    - Connected with servers
    - Send out query to nearby RFID tags
    - Read data emitted from tags
  - RFID Tags
    - Passive tags reflect the RF signal
    - Active tags use battery to power the transceiver
    - Large-scale deployed, but cannot communicate with each other

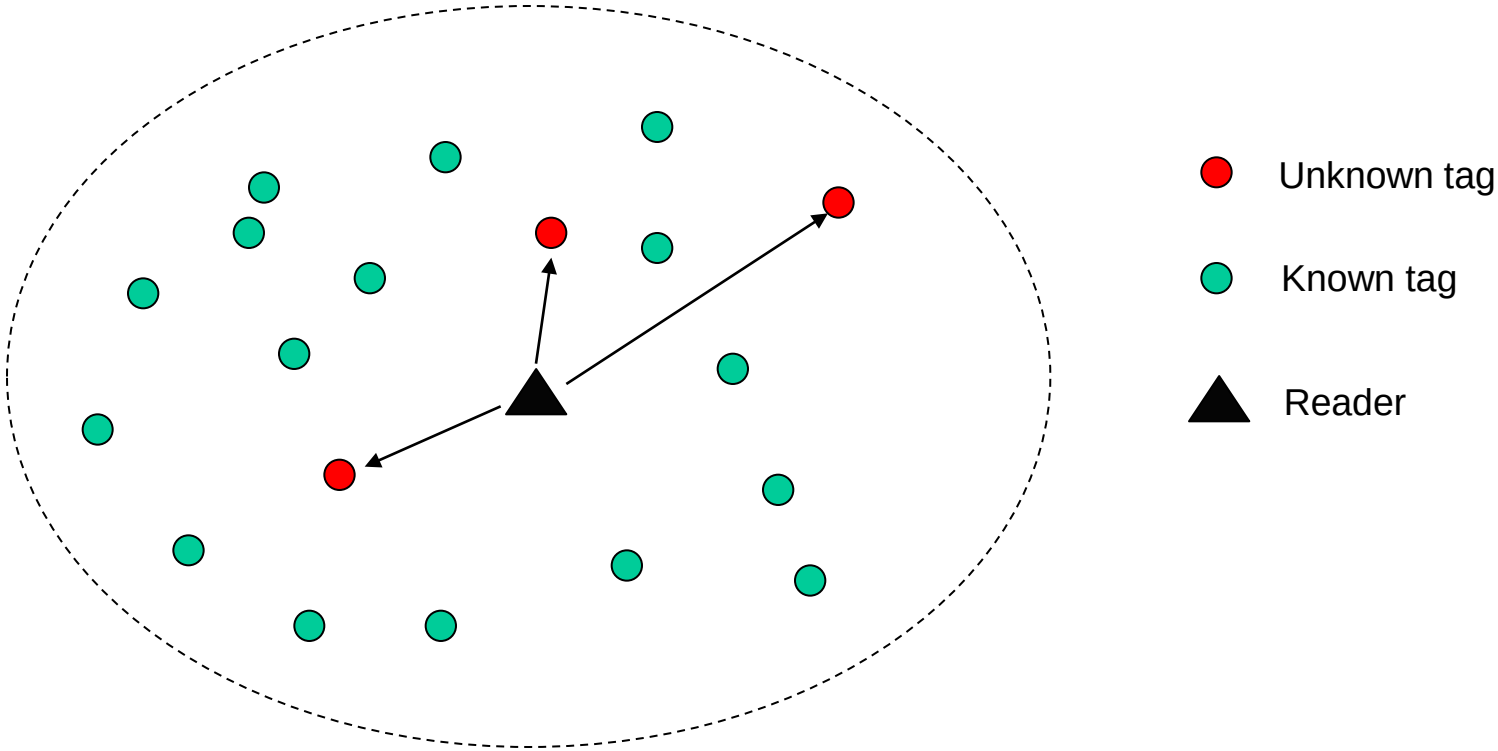


Unknown Tag

- The wrongly placed tags that are unknown to the readers covering their current appearance
  - In warehouse, the stevedores load products to wrong ships or trucks
  - In library, people sometimes put books into wrong categories casually

Problem

- Fast and complete identification of unknown tags in large RFID systems



Motivation

- A continuous scanning scheme identifies unknown tags with probability [INFOCOM 2010]
  - NOT complete
- Typical tag identification protocols collect all the IDs of tags to completely identify unknown tags
  - NOT fast

The Main Idea

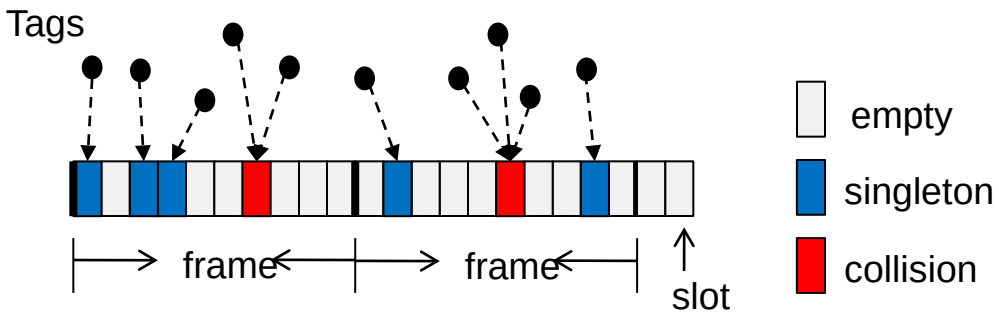
- The most efficient method: collect IDs of only unknown tags, however
  - All the tags would response the reader's query
- Only if a known tag has been acknowledged, it would not transmit when receiving the query from the reader
  - Typically, the reader acknowledges a tag when it receives the tag's ID
  - ID transmission is inefficient (i.e. 96bits)

Challenges

- How to acknowledge all the known tags ?
  - Without transmitting their IDs
  - Guarantee the time efficiency with a large number of known tags in the system

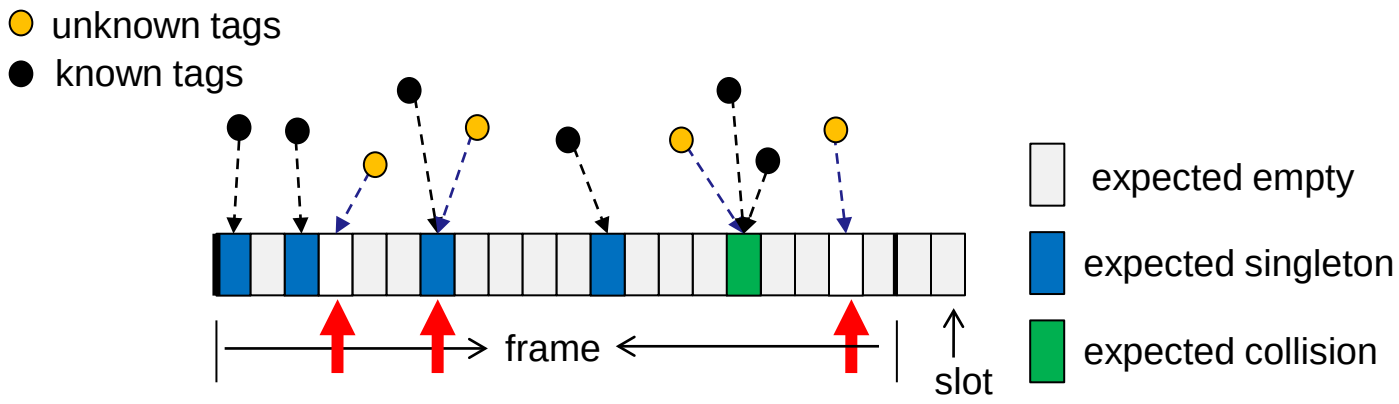
Framed-slotted ALOHA Protocol

- Reader-talks-first model
  - The reader issues commands $< f; r >$
  - Tags reply to the reader in slot $H(ID||r) \bmod f$
- Three kinds of slots
  - Empty slot: no tag transmits in this slot
  - Singleton slot: just one tag transmits in this slot
  - Collision slot: multiple tags transmits in this slot



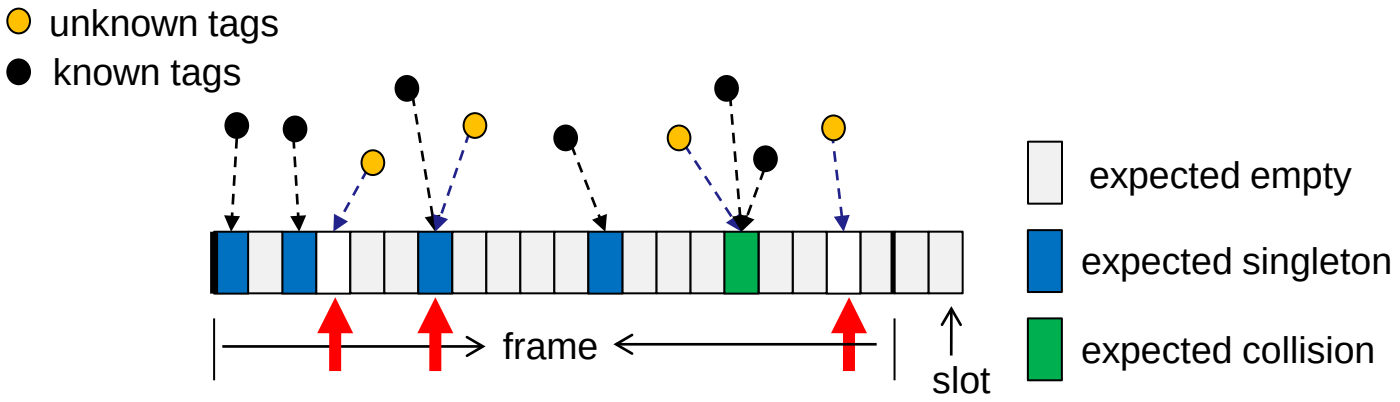
Assumption

- The reader knows all the IDs of known tags stored in a database
  - The reader could predict the slot that a known tag would choose with *H(ID||r) mod f*
  - The reader knows the expected state of each slot
    - Expected slots are concerned about only the known tags
    - The reader cannot predict the slots that unknown tags would choose



BUIP: **B**asic **U**nknown tag **I**dentification **P**rotocol

- Use expected singleton slots to acknowledge known tags
  - If only one tag replies in an expected singleton slot, the reader acknowledges the known tag with an ACKs
  - Tags only need to transmit a short response for collision detection instead of ID
    - $t_I << t_{ID}$



Known Tag Acknowledgment

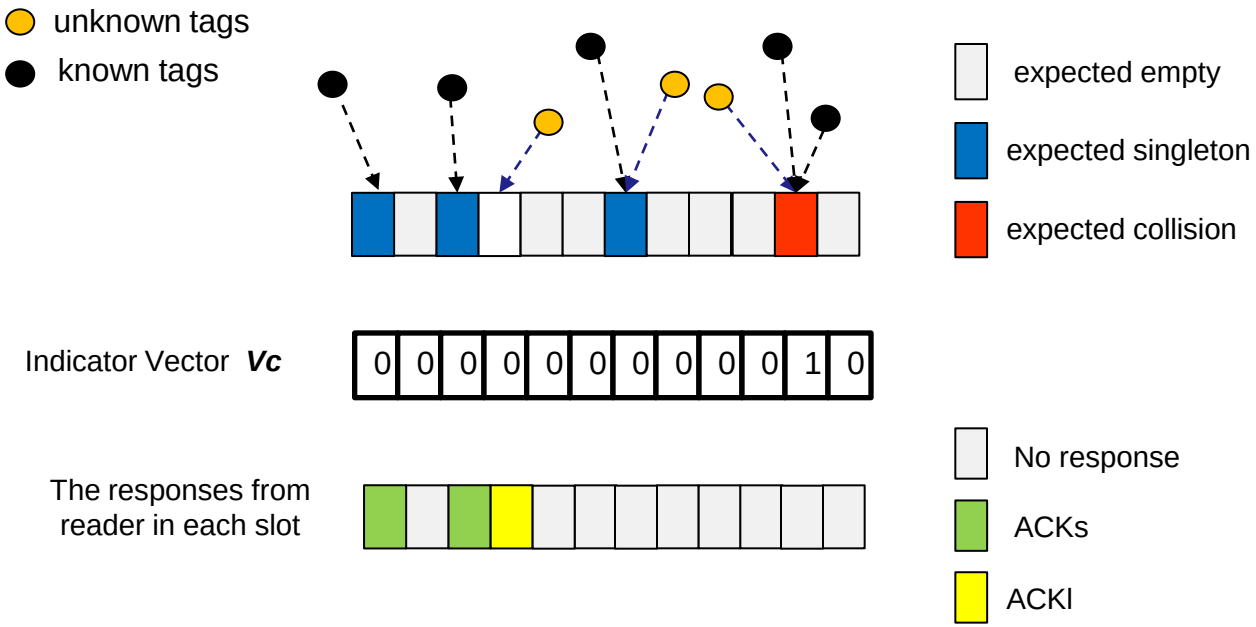
- Not all known tags can be acknowledged in one round
  - Those reply in expected collision slots
    - Interference from other known tags
  - Those reply in expected singleton slots but a collision detected
    - Interference from unknown tags

BUIP Design

- Two phases
  - Known tag inactivation phase
    - Consists of multiple rounds
    - In each round, the reader broadcasts Query< f; r >,
    - And an indicator vector Vc
      - Prohibit transmission in expected collision slots
  - Unknown tag collection phase
    - Collect IDs of unknown tags with typical ALOHA protocol

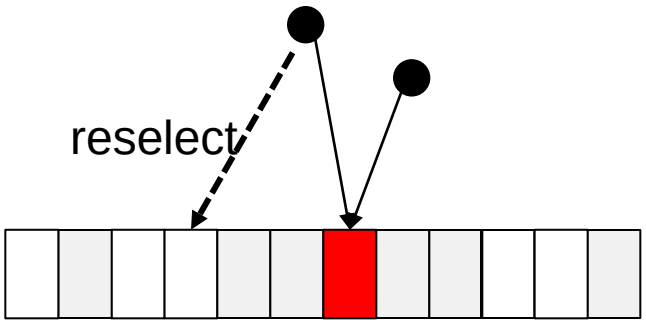
Indicator Vector Illustration

- Consisting of ***f*** elements
- One element for one slot: ‘1’ for expected collision; ‘0’ for others
- Prohibit tag transmissions in expected collision slots



BUIP extension I: Collision-empty slots pairing

- Motivation
  - Known tags can be inactivated only in *expected singleton slots*
  - Slot *reselection* ----- may turn an *expected collision slot* into *an singleton slot*
    - Multiple tags select the collision slot originally
    - Some of them select other empty slots

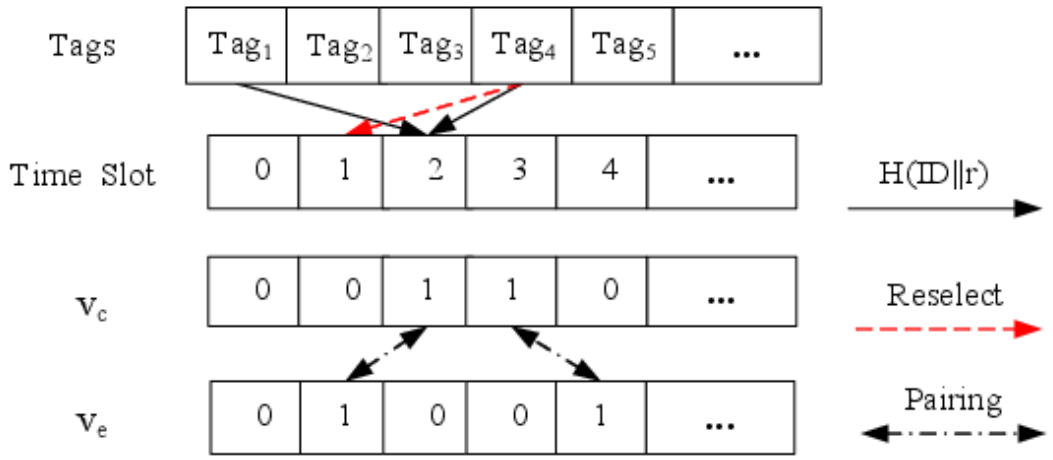


BUIP extension I: Collision-empty slots pairing

- Challenge:
  - How to assign an empty slot for each collision slot
  - How to notice tags the index of the empty slot

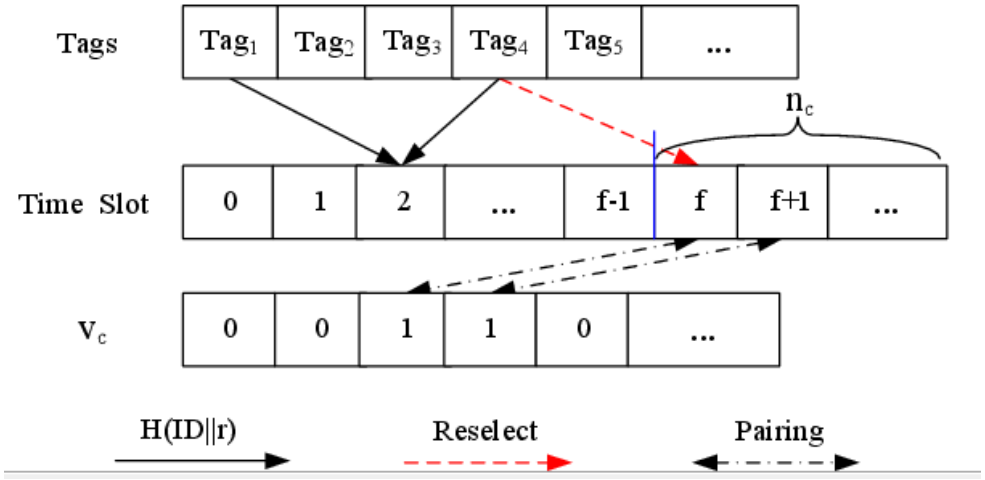
Collision-Empty Slots Paring

- The i-th collision slot pairs with the i-th empty slot
- The reader sends two vectors *Vc* and *Ve* for tags to determine pairing empty slots
  - Vc shows the index of collision slots
  - Ve shows the index of expected empty slots
- Reselection with a probability of 0.5



BUIP extension II: Collision-Fresh slots pairing

- Motivation：
  - BUIP-CE Pairs collision slots with expected empty slots
    - These slots may be non-empty --- unknown tags may choose them
    - Use fresh slots in the extended frame which can be guaranteed empty



BUIP-CF performance

- Probability that known tags are inactivated with different number of reselection seeds

TABLE II
THE VALUE OF p_d WITH DIFFERENT k ($n = 10,000, m = 2,000$).

k	1	2	3	4	5	6	7
p_d	.575	.671	.717	.740	.752	.758	.761

Execution time in known tag inactivation

TABLE III
EXECUTION TIME IN KNOWN TAG INACTIVATION PHASE WHEN m CHANGES ($n = 10000$).

Alg. Name	Execution time in known tag inactivation phase (s)									
	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
CU-All	41.61	52.80	57.25	65.14	71.25	78.14	80.70	92.82	89.25	93.50
CU($R_U = 0.01m$)	35.17	33.66	36.06	38.46	40.86	43.26	45.66	47.76	44.47	46.25
CU($R_U = 0.05m$)	23.45	21.04	22.54	24.04	25.54	27.04	28.54	30.04	31.54	33.04
CU($R_U = 0.1m$)	15.63	16.83	18.03	19.23	20.43	21.63	22.83	24.03	25.23	20.48
BUIP	24.73	26.60	28.34	29.93	31.44	32.84	34.11	35.36	36.57	37.65
BUIP-CE	18.48	21.05	23.28	25.33	27.26	29.05	30.67	32.10	33.59	34.99
BUIP-CF	14.80	16.15	17.53	18.59	19.70	20.64	21.62	22.44	23.24	23.90

Total execution time

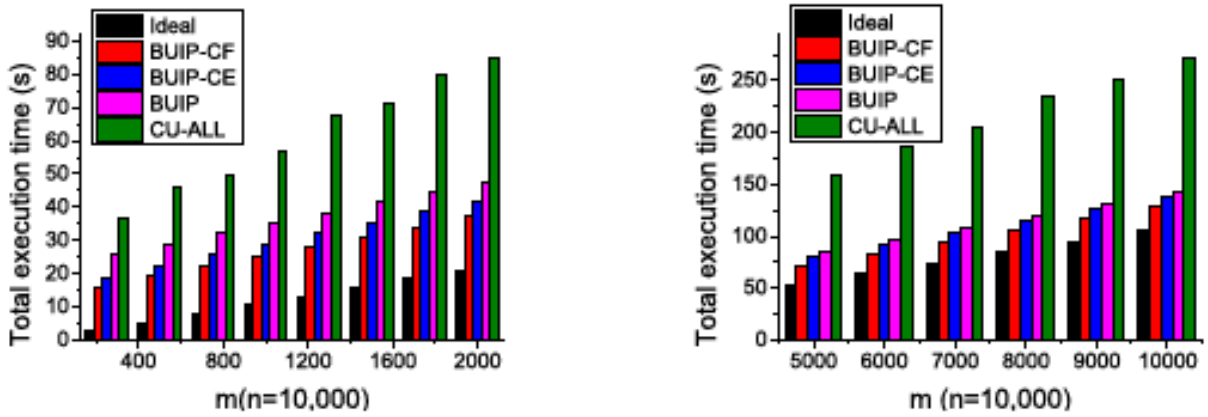


Fig. 5. Total execution time of different protocols: (a) small m; and (b) large m.

Question and Answer