A Hybrid Query Tree protocol for Tag Collision Arbitration in RFID systems

IEEE International conference on Communications 2007

Jiho Ryu, Hojin Lee, Yongho Seok, Taekyoung Kwon and Yanghee Choi
Seoul National University
Outline

• Introduction
• Hybrid Query Tree (HQT) protocol
• Performance Evaluation
• Conclusion
Introduction

- Passive RFID system
  - Tag does not have an internal power sources
  - Collisions between tags’ responses are critical issue

- Two types of tag anti-collision protocol
  - ALOHA-based
    - tag starvation problem
  - Tree-based
    - Query tree (QT)
      - simple, relatively need more query messages than BTS
    - Binary tree splitting (BTS)
      - more complex than QT

- Focused on the QT protocol, and try to reduce delay
Hybrid Query Tree protocol
Hybrid Query Tree (HQT) Protocol

- Three cycles
  - **Collision cycle**: Number of tags that respond to the interrogator (Reader) is more than one. The interrogator cannot identify the ID of tags
  - **Idle cycle**: No response. It is a waste time that should be reduced
  - **Success cycle**: Exactly one tag responds to the interrogator. The interrogator can identify the ID of the tag
Hybrid Query Tree (HQT) Protocol

• (1) 4-ary Search Tree Mechanism

Binary Query Tree

4-ary Query Tree

- Collision cycles can be reduced but idle cycles are increased!!!
Hybrid Query Tree (HQT) Protocol

- How can we reduce the number of the idle cycles?

- (2) Slotted Backoff Tag Response Mechanism
  - Assume that RFID interrogator has carrier sensing ability
  - A Query string from an interrogator
    - QT protocol
      - tag immediately responds to an interrogator
    - HQT protocol
      - defer its response by a backoff time
      - 2 bits which follow the prefix of tag ID identical to the query string

  - **By sensing the channel**, the RFID interrogator can estimate the minimum backoff time and the maximum backoff time of the collided tags → next query string
Hybrid Query Tree (HQT) Protocol

- **(2) Slotted Backoff Tag Response Mechanism**
  - Assume that RFID interrogator has carrier sensing ability
  - By sensing the channel, the RFID interrogator can estimate the minimum backoff time and the maximum backoff time of the collided tags

- No query with ‘010111’ string
  - reduce unnecessary idle cycles
• Example - How it works?
  - ID of tags: 0101, 0110, 1100, 1101

3 idles, 5 collisions, 4 successes
Simulation
Performance Evaluation

- Simulation Environment

  - **NS2 simulator**
    - Query Tree (binary, 4-ary), Hybrid Query Tree
    - 25~200 tags
    - **ID:** 128 bits (random generation)
    - Rate of Query and Response: 128Kbps
    - **Length of idle cycle:**
      - HQT: 80us (20us + max back-off time slot) + query tx time
      - others: 20us + query tx time
    - 20 times

  - Performance metrics: number of collision cycles, number of idle cycles, identification delay
    - **black line:** binary QT,  **blue line:** HQT,  **red line:** 4-ary QT
Performance Evaluation

- Simulation Result
  - Number of collision cycles

![Graph showing average number of collision cycles vs number of tags for binary QT, 4-ary QT, and HQT, with 50%标记 at a specific point on the graph.]

jhryu@mmlab.snu.ac.kr
Performance Evaluation

- Simulation Result
  - Number of idle cycles

![Graph showing average number of idle cycles versus number of tags for different QT methods: binary QT, 4-ary QT, HQT.](graph.png)
Performance Evaluation

- Simulation Result
  - Identification delay

![Graph showing identification delay vs number of tags for different QT types: binary QT, 4-ary QT, and HQT.](image)
Conclusion

• Hybrid Query Tree protocol
  - 4-ary search tree mechanism + slotted backoff tag response mechanism

• For implementation
  - Tag
    • Response deferring technique
    • Backoff timer, etc → little storages
  - RFID Interrogator (Reader)
    • Carrier sensing ability

• Future Work
  - Numerical analysis
Thank you!

Q&A