IoT: lecture

Bardh Prenkaj
Dept. of Computer Science

Internet of Things A.A. 17-18

EPCglobal Class-1 Generation-2 Anti-collision Protocol

Anti-collision in most RFID Systems

- Use time division duplex (**TDD**) to implement the two way communication between **reader** and **tags**

- Adopt time division multiple access (**TDMA**) to simultaneously read the tags

- Anti-collision scheme becomes a strategy to solve the **multiple-access** problem

- **Efficiency** problems when the tag number is large

EPCglobal

- Non-profit standard organization working to commercialize the use of electronic product code (**EPC**)

- Developed a standard called **EPCglobal Class-1 Generation-2** which:

  - Meets the **performance** for reading a **dense** tag number

  - Is an interrogator-talks-first (**ITF**) system operating in the 860 MHz-960 MHz frequency range (UHF)

  - Can support reading speed up to hundreds of tags per second

# EPC Gen2 Anti-Collision Q-Algorithm

Q-Algorithm Overview

- Based on **Dynamic FSA** that allows readers to **adjust** frame length at any slotted time

- Improves performance when the number of tags is much less or much more than the frame length

- In literature there are different solutions to improved **schema** of the **Q-Algorithm** (more on this in the next slides)

Q-Algorithm Procedures

■ Dynamic Frame Slotted ALOHA generates an **adaptive** frame size to the next reading round according to what happened in the **previous one**

■ **Three** procedures which comprise one or more commands each:

- **Select**

- **Inventory**

- **Access**
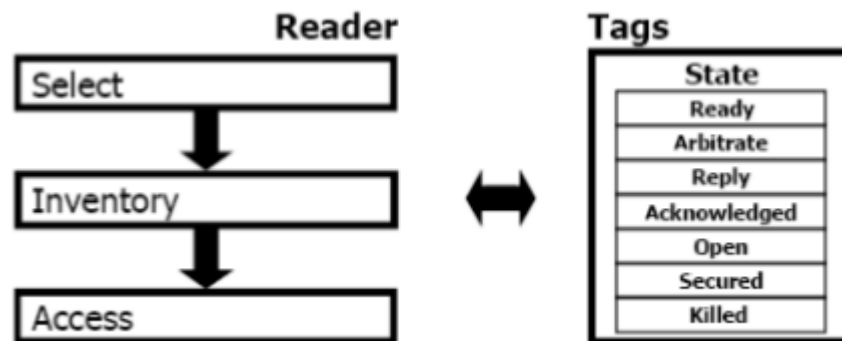
Q-Algorithm
Select & Access

■ **Select**:

- An interrogator selects a tag population for inventory and access

- May do more select commands to grab the particular tag population

■ **Access**:

- An interrogator transacts with (reads from or writes to) the individual tags

- An individual tag with an EPC must be uniquely identified prior to access

Q-Algorithm Inventory (I)

■ **Inventory:**

- An interrogator identifies all tags to be accessed

- A round begins by transmitting a **Query** command (with a Q-value) to which one or more tags may reply

- The interrogator detects a **single** tag reply and identifies this single tag

- A **series** of Query commands is **repeated** for the possible **backlog** of tags

■ Each inventory round includes **several time-slots**

- The number of time-slots (frame size) is required to be determined by the DFSA

Q-Algorithm Inventory (II)

- After the Query command is sent with a **Q value** all the receiving tags generate a 16 bit random number (**RN16**) and get a random number from the last Q-bits of the RN16 $\in$ [0, (2^Q)-1]

- The tags will count down their slot numbers under the control of the reader and send their own RN16 when the **counter decreases to 0**

- There are three possible outcomes for tag responses: single tag reply (**ACK** is sent to the tags), collided reply (**no response** is sent or **NACK** based on the chosen strategy), and no reply

Q-Algorithm Inventory (III)

- **Single tag reply**: the acknowledged tag sends the Protocol Counter (PC), its EPC and the CRC16 (Cyclic Redundant Check) to the interrogator
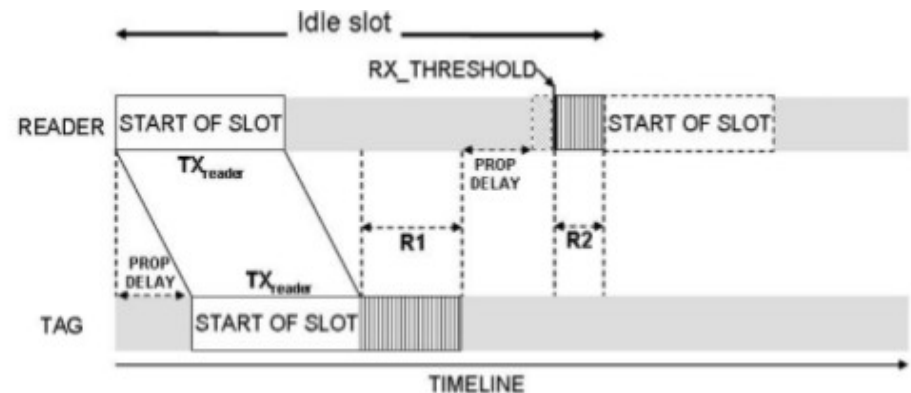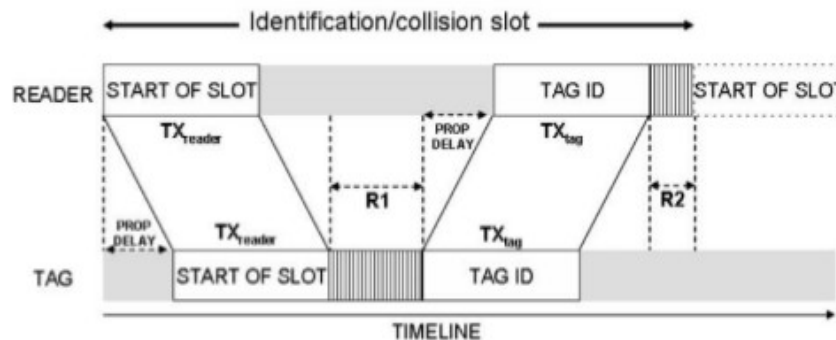
- **No reply**: the interrogator will realize this after some time T1+T2 whereupon the slot gets automatically closed

Q-Algorithm QueryRep & QueryAdj

- After a Query command, the interrogator typically sends one or more **QueryRep**/**QueryAdj** commands.

  - QueryRep repeats a previous query **without** changing any parameters

  - QueryAdj repeats a previous query and may **increase** or **decrease** the Q value by one → adjust the frame length at each time slot

- The interrogator maintains a floating-point parameter Q_{fp} and a constant **C** for the frame length adjustment

Q_{fp} and C adjustments (I)

■ **Single identification slot**:

- Q_{fp} remains unchanged

- If the round value Q_{fp} is different from the current Q value, the interrogator sends **QueryAdj** → too many idle or colliding slots
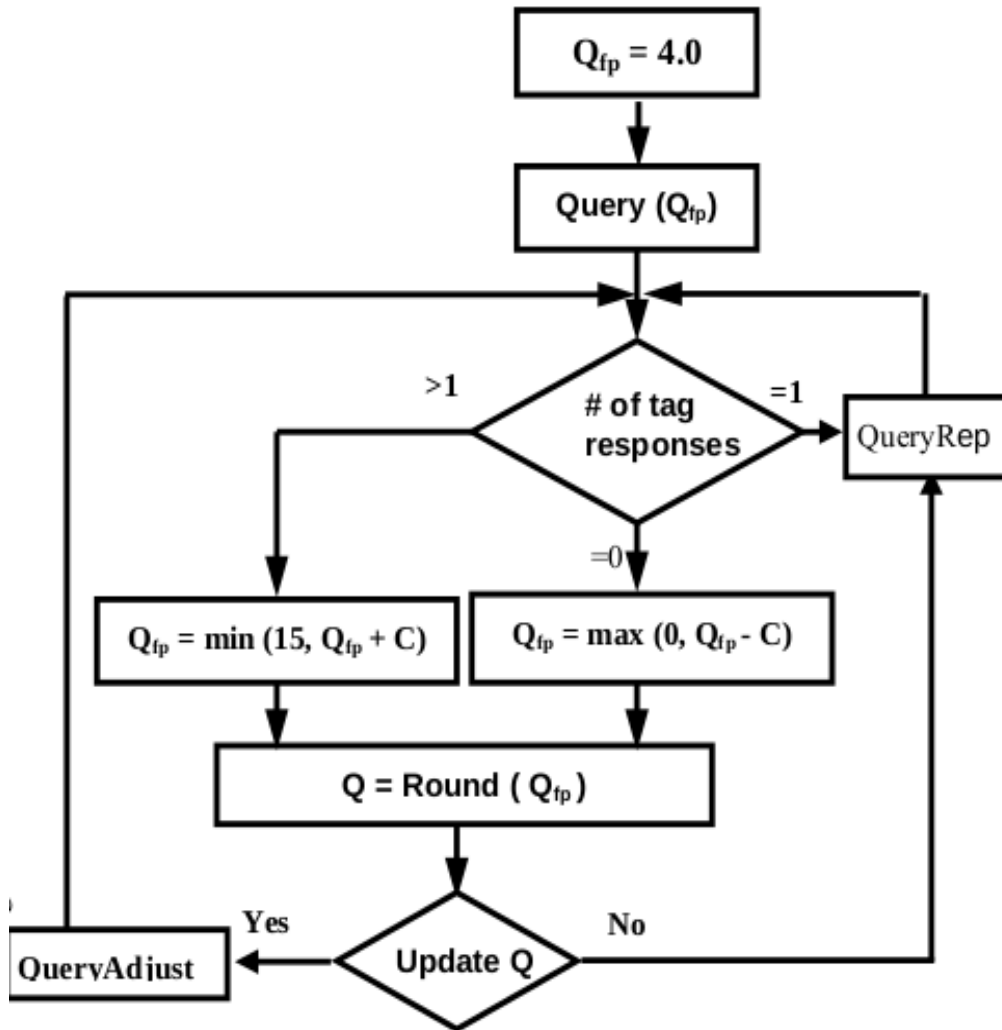
■ **Collision slot:**

- Sum C to Q_{fp}

■ **Idle slot:**

- Q_{fp} decreases by C

Q_{fp} and C adjustments (II)

- Typical values for C are between 0.1 and 0.5

- An interrogator adopts small values of C when Q is large and adopts large values of C when Q is small

- **Optimal** value **C = 0.2**

Chen and Kao's Q-Algorithm

Assumptions

- **Idle** conditions occupy **less** time interval than the collided ones

- To speed up the read procedure for all tags, expect **more idle slots** than collision slots

  - The decrement of Q_{fp} should be less than its increment under the consideration of read performance

- Need of **different C** values (C_1 and C_2) for the adjustment of Q_{fp} value → a new sch eme for the Q-algorithm is proposed

  - $0.1 < C_1 < C_2 < 0.5$

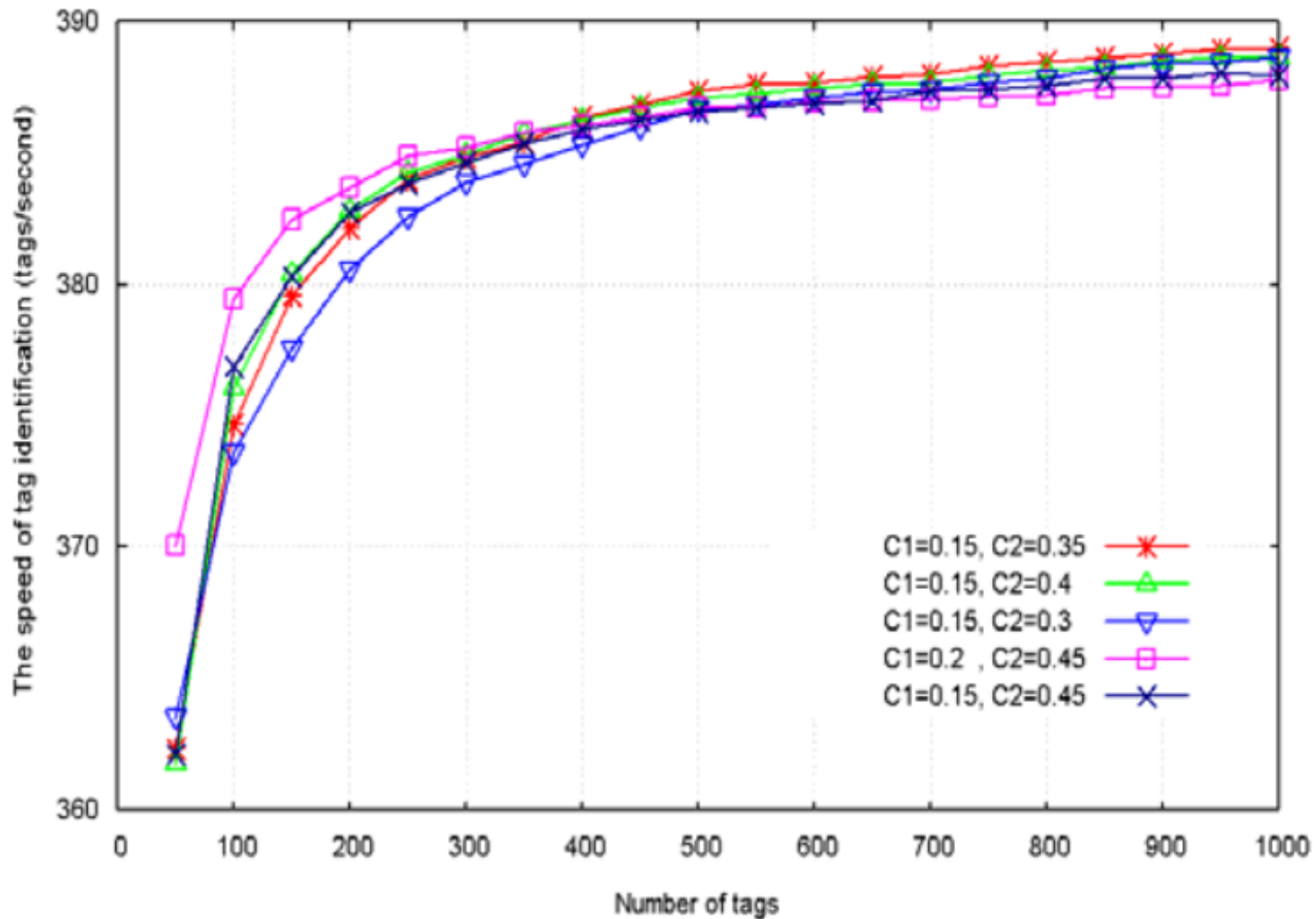 - These two values effect on the total time spent in reading tags
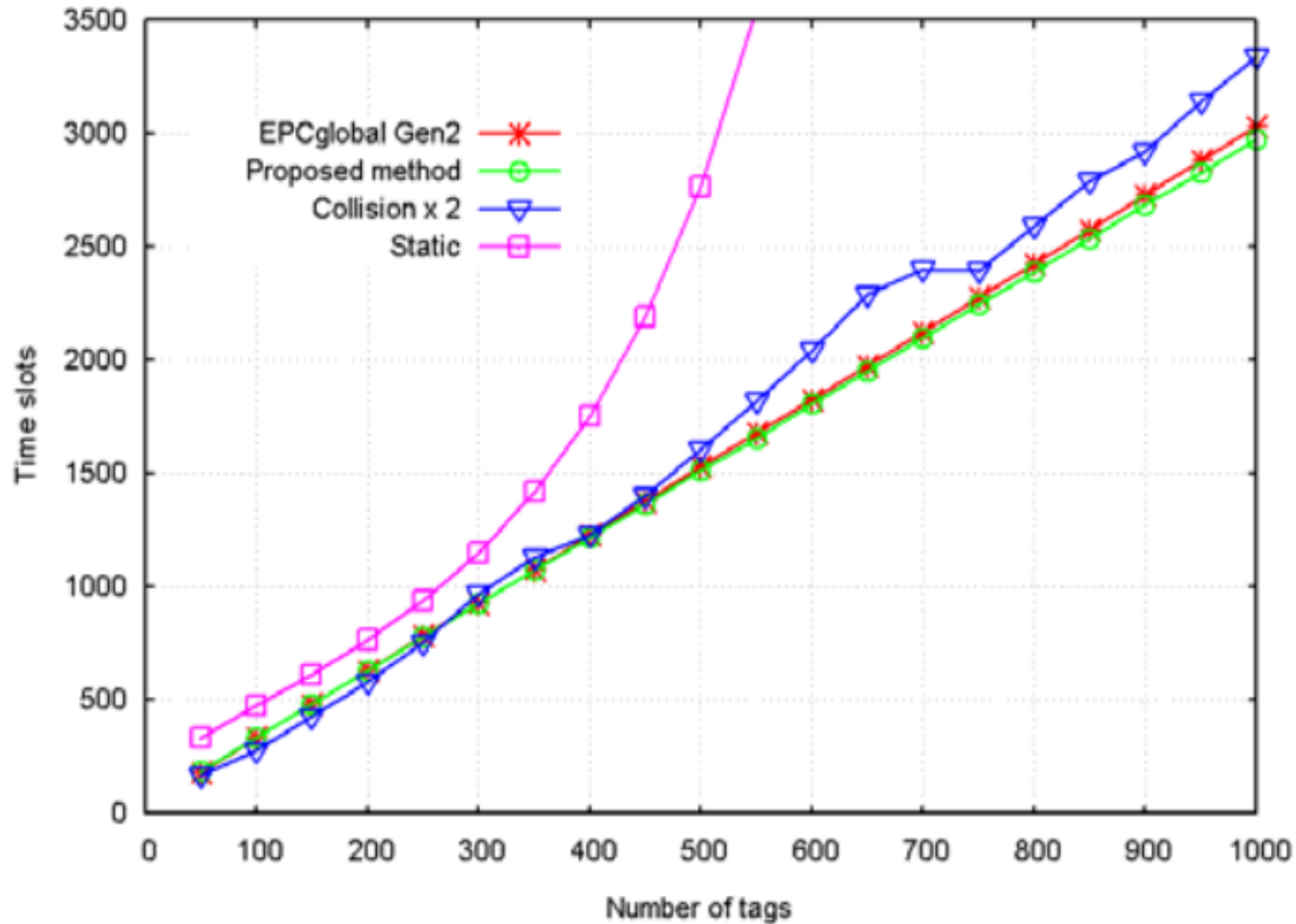
Modified Q-Algorithm flow
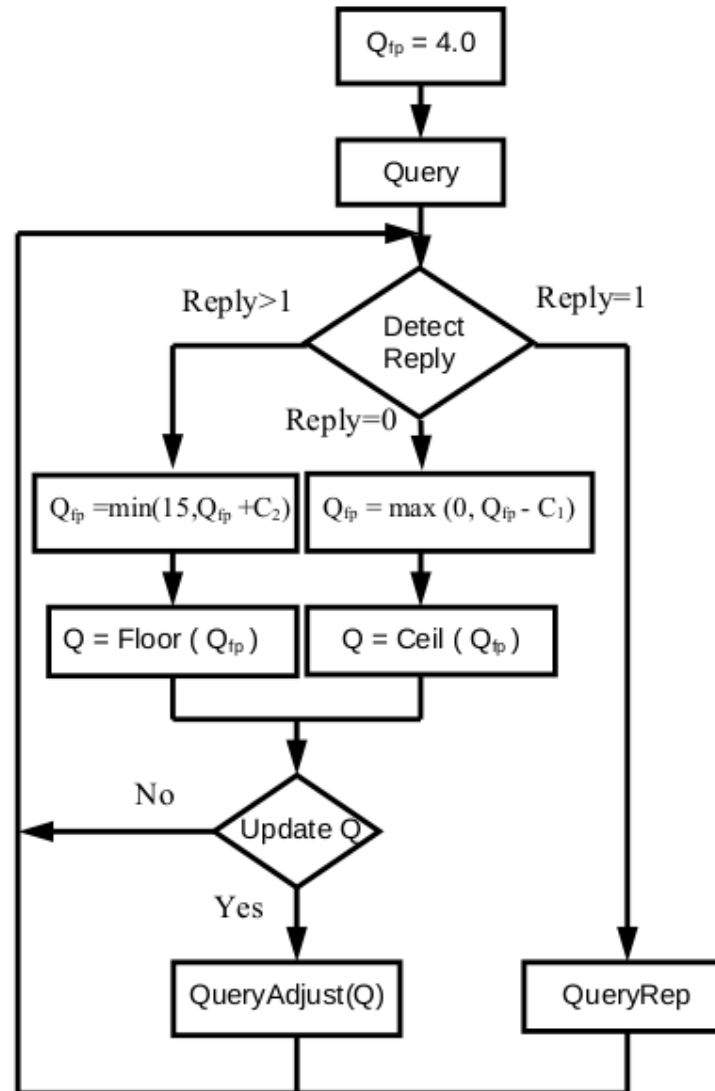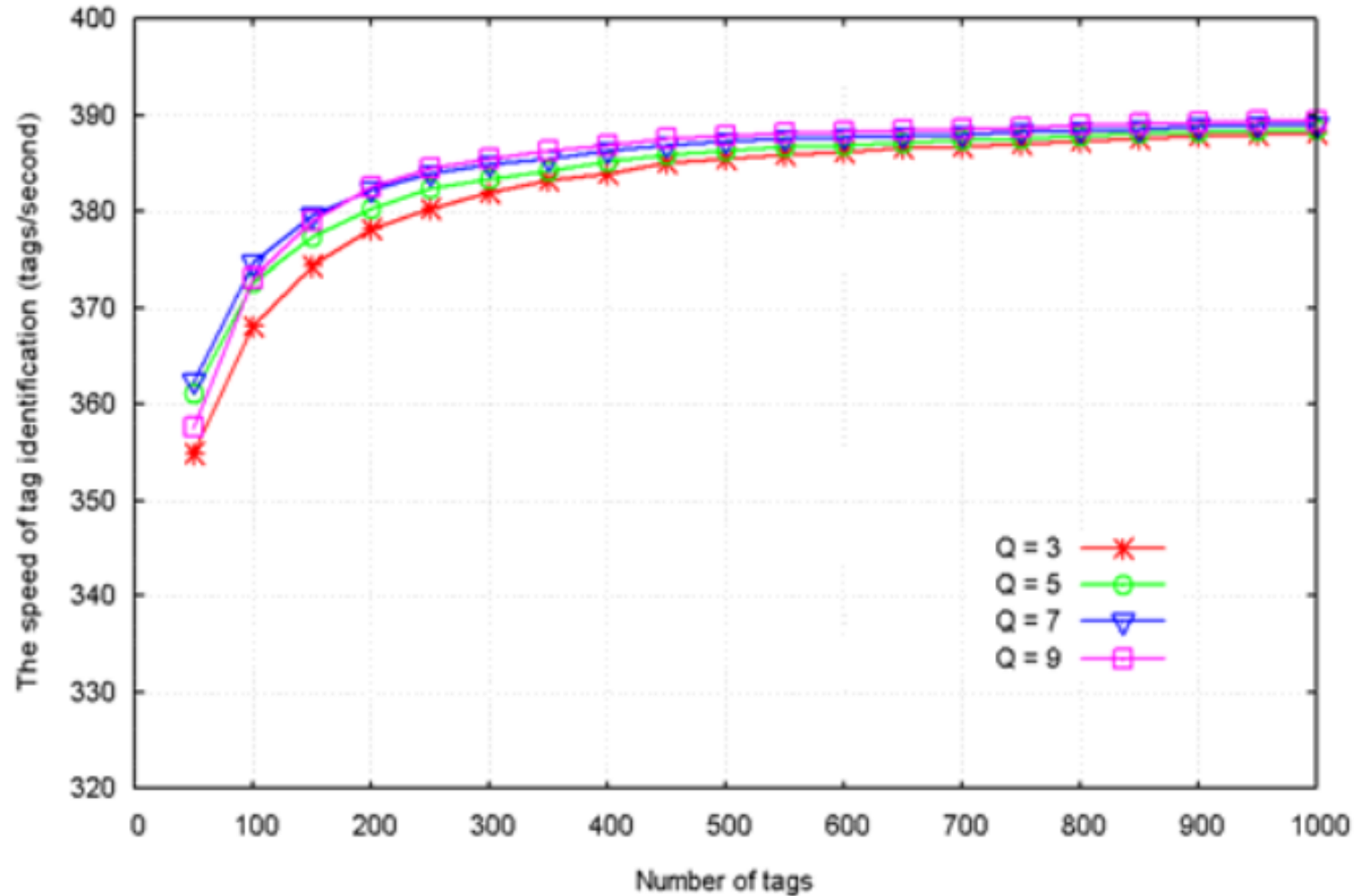
Tag identification speed

Time slot number for tag identification

2^Q-sized frame read speed effect

Readings

- W. T. Chen, W. B. Kao, **A novel Q-algorithm for EPCglobal class-1 generation-2 anti-collision protocol**, in World Academy of Science, Engineering and Technology, 78, 801-804, 2011.

- W. T. Chen, **A New RFID Anti-collision Algorithm for the EPCglobal UHF Class-1 Generation-2 Standard**, in UIC-ATC '12 Proceedings of the 2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing, 811-815, September 2012.