UHF 대역 RFID 시스템용 Anti-collision Algorithm 개발

(최종 발표)

(2005. 03. 01 ~ 2005. 12. 31)

Apr. 14, 2006

Jae-Hyun Kim

Wireless Information & Network Engineering Research Lab.

School of Electrical Engineering AJOU University

Contents

연구 내용 및 결과

- 연구 목표
- 연구 내용
- 연구 결과 (Class 1)
- 성능 분석 결과 (Class 1)

EPC Gen 2 basics

- Gen 2 특징
- Gen 2 에서 태그 인식 과정
- 시뮬레이션 결과
- Gen 2 issues

Research contents and results

연구 목표

- UHF 대역 RFID 시스템용 Anti-collision Algorithm 개발

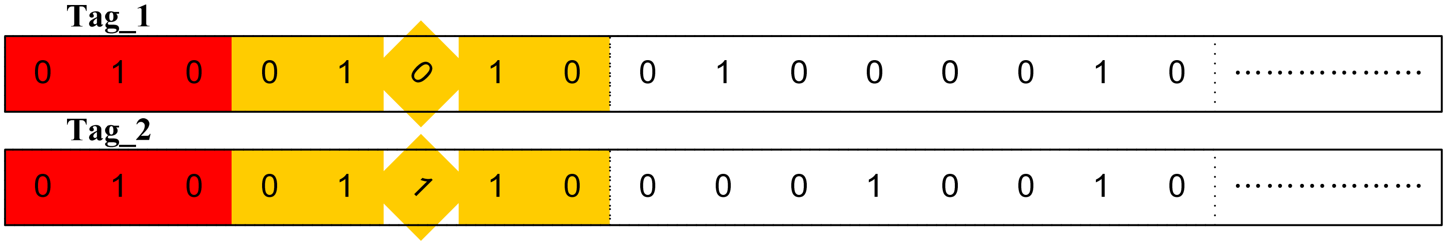
연구 내용

- 기존 Anti-collision Algorithm 조사 및 성능분석
- 새로운 UHF 대역 Anti-collision Algorithm 개발 및 성능 분석
- 알고리즘의 성능 분석 및 검증 위한 시뮬레이터 개발
- EPC CLASS 1 Gen 2로의 확장가능성 연구

Research results (1)

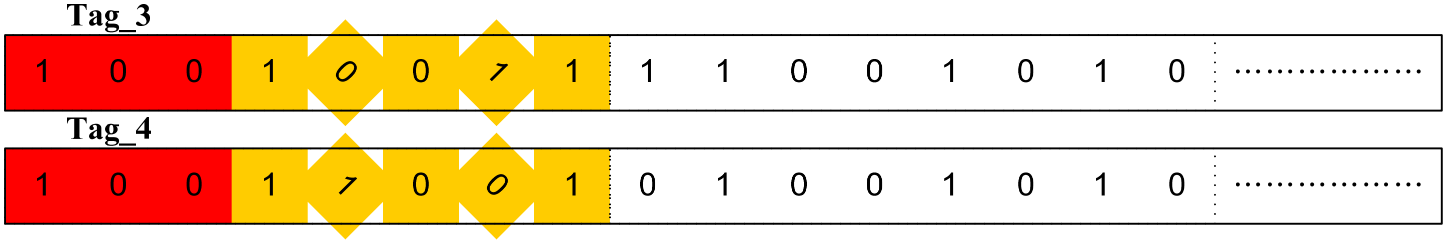
Bit 단위 충돌 정보를 이용하는 알고리즘(2005년 5월)

- 동작절차 :
  - 응답한 태그가 1개인 경우 : 리더는 응답한 태그를 인식
  - 충돌이 발생한 bit가 1개인 경우
    - ✓ 태그가 모두 2개가 있는 것이므로 리더는 ScrollID 명령을 사용하여 2개의 태그를 순차적으로 인식



Tag_1	0	1	0	0	1	0	1	0	0	1	0	0	0	0	1	0	…………………
Tag_2	0	1	0	0	1	1	1	0	0	0	0	1	0	0	1	0	…………………

  - 충돌이 발생한 bit가 2개 이상인 경우



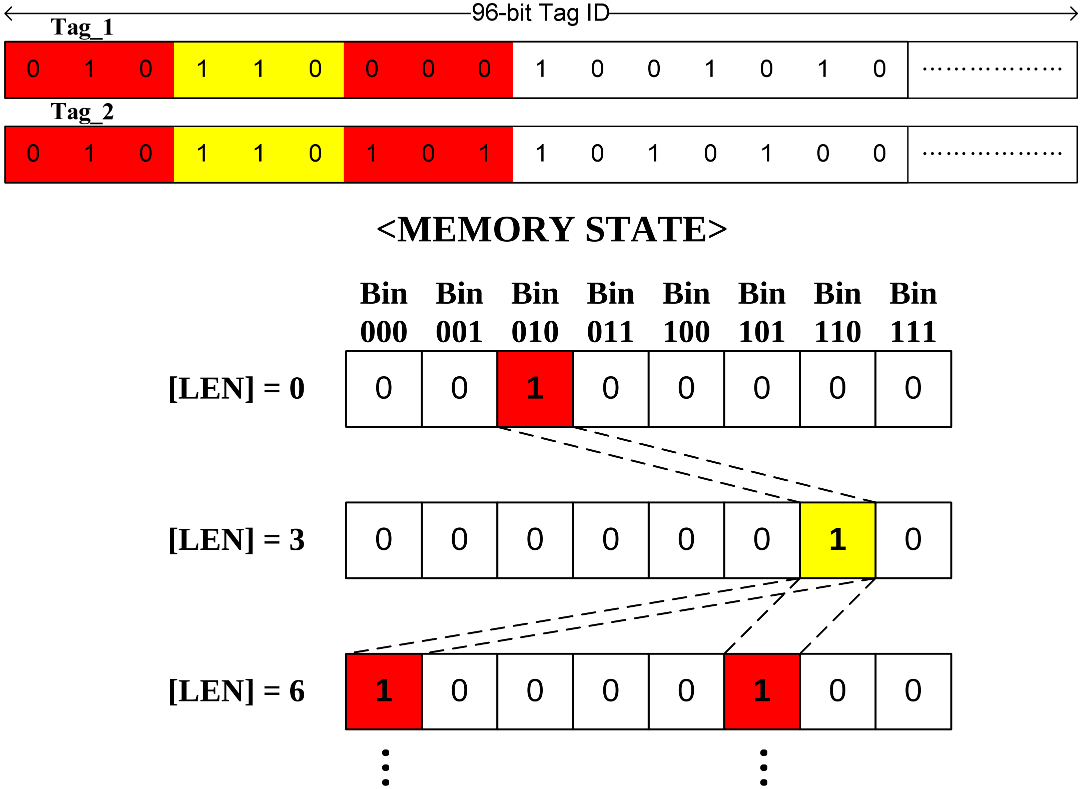
Tag_3	1	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	…………………
Tag_4	1	0	0	1	1	0	0	1	0	1	0	0	1	0	1	0	…………………

- 시뮬레이션 결과 초당 약 169.24개의 태그 인식 가능

Research results (2)

PingID 명령만을 사용하는 알고리즘 (2005년 11,12월)

- 동작절차 :
  - 어떤 충돌 정보도 이용하지 않고 트리 구조로 PingID 명령만을 이용하여 3-bit씩 태그의 ID를 인식하여 마지막 96 비트까지 인식



Tag_1	0	1	0	1	1	0	0	0	0	1	0	0	1	0	1	0	………………
Tag_2	0	1	0	1	1	0	1	0	1	1	0	1	0	1	0	0	………………

<MEMORY STATE>

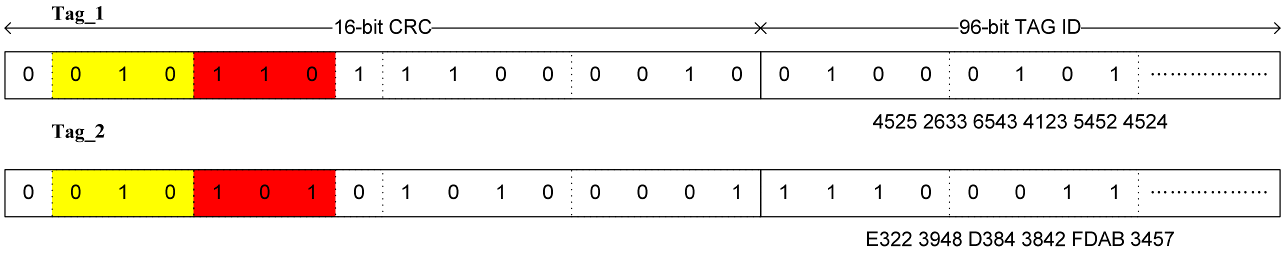
	Bin 000	Bin 001	Bin 010	Bin 011	Bin 100	Bin 101	Bin 110	Bin 111
[LEN] = 0	0	0	1	0	0	0	0	0
[LEN] = 3	0	0	0	0	0	0	1	0
[LEN] = 6	1	0	0	0	0	1	0	0
	⋮					⋮		

- 시뮬레이션 결과 초당 약 169.24개의 태그 인식 가능

Research results (3)

Alien 社 알고리즘 (2006년 1월)

- 동작 절차 :
  - 태그의 응답이 있을 경우 ScrollID 명령을 전송
    - ✓ 충돌이 있을 경우, [LEN]=[LEN]+3인 PingID 명령을 전송
    - ✓ 충돌이 없을 경우, 태그를 인식



Tag_1

0	0	1	0	1	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	1	0	1	………………

4525 2633 6543 4123 5452 4524

Tag_2

0	0	1	0	1	0	1	0	1	0	1	0	0	0	0	1	1	1	1	0	0	0	1	1	………………

E322 3948 D384 3842 FDAB 3457

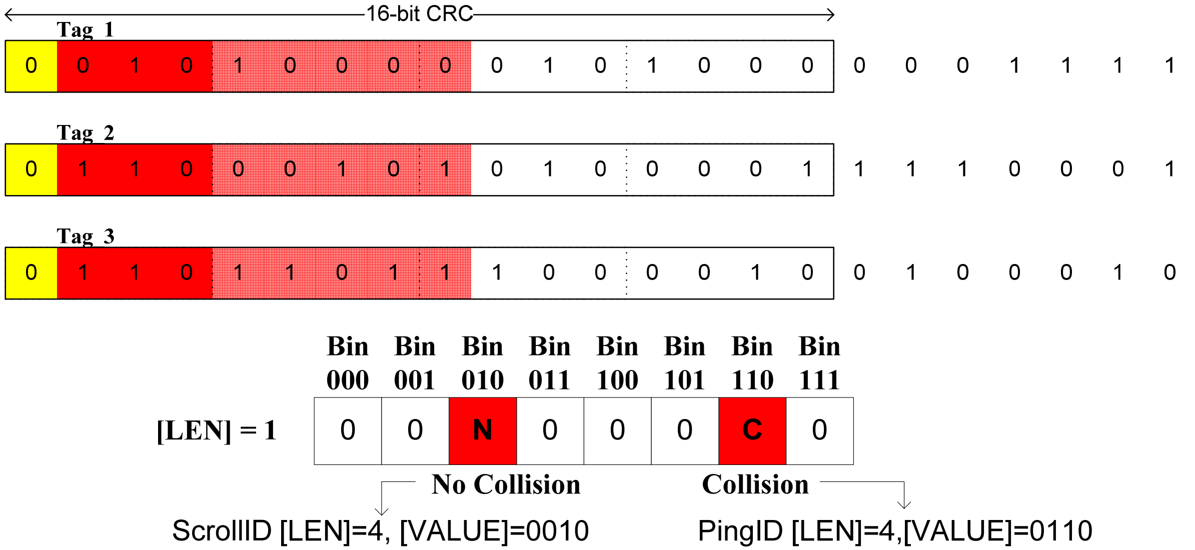
		Bin 000	Bin 001	Bin 010	Bin 011	Bin 100	Bin 101	Bin 110	Bin 111
ScrollAllID									
PingID [LEN]=1,[VALUE]=0	[LEN] = 1	0	0	1	0	0	0	0	0
ScrollID [LEN]=1,[VALUE]=0 → Collision									
PingID [LEN]=4,[VALUE]=0010	[LEN] = 4	0	0	0	0	0	1	1	0

- 시뮬레이션 결과 초당 약 58.88개의 태그 인식 가능

Research results (4)

Bin Slot 내의 충돌 有/無 정보를 이용하는 알고리즘 (2006년 2월)

- 동작 절차 :
  - Bin slot에 들어오는 8-bit 태그 응답 중 뒤의 5-bit에서 충돌 有/無 정보를 이용
    - ✓ 충돌이 있을 경우, [LEN]=[LEN]+3인 PingID 명령 전송
    - ✓ 충돌이 없을 경우, ScrollID 명령을 보냄
  - 태그의 개수가 500개일 때 ScrollID 명령어 개수가 약 200개 감소



16-bit CRC

Tag 1																						
0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	1
Tag 2																						
0	1	1	0	0	0	1	0	1	0	1	0	0	0	0	1	1	1	1	0	0	0	1
Tag 3																						
0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	1	0

	Bin 000	Bin 001	Bin 010	Bin 011	Bin 100	Bin 101	Bin 110	Bin 111
[LEN] = 1	0	0	N	0	0	0	C	0

No Collision

ScrollID [LEN]=4, [VALUE]=0010

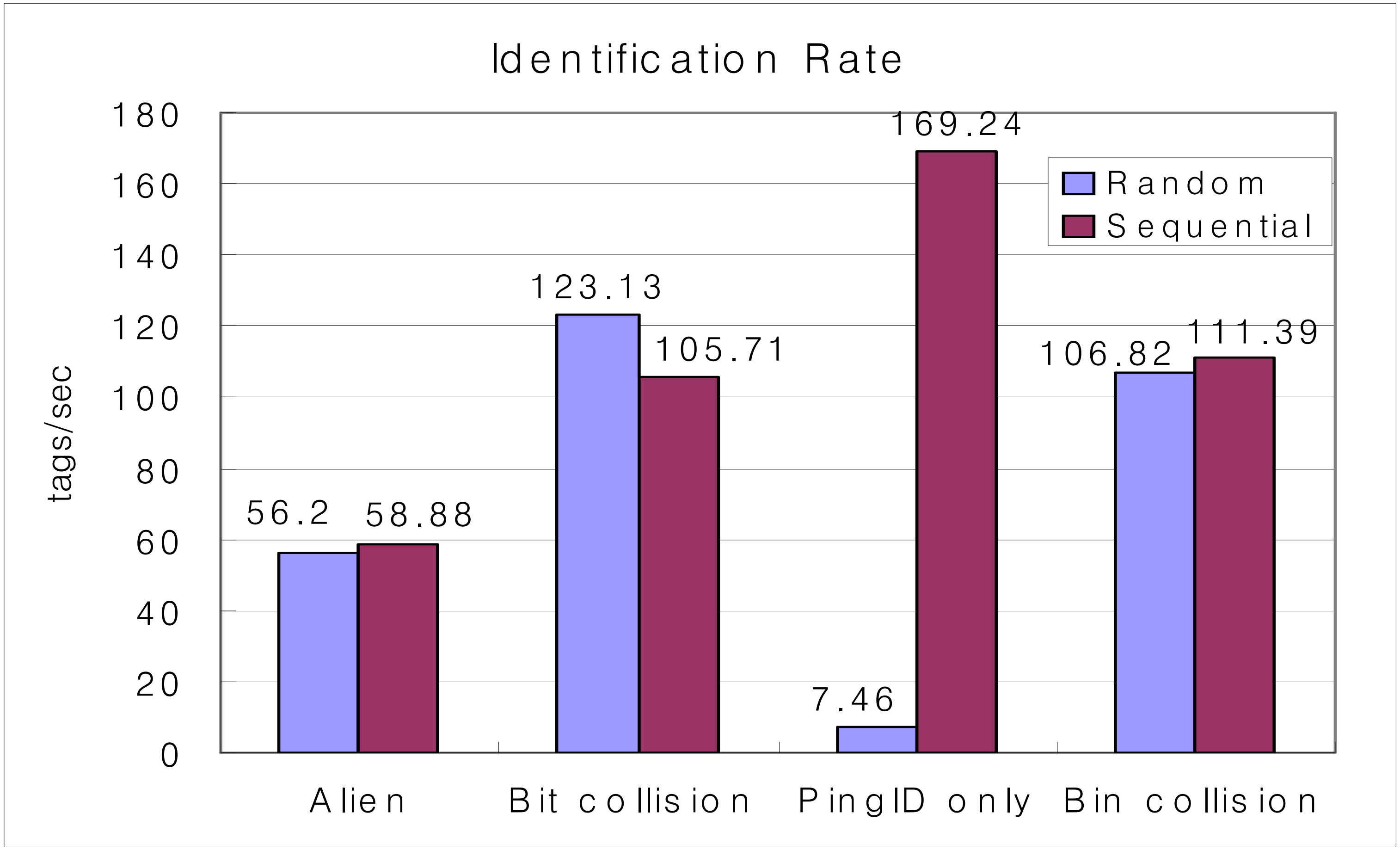
Collision

PingID [LEN]=4,[VALUE]=0110

- 시뮬레이션 결과 초당 약111.39개의 태그 인식 가능

Simulation results

- 알고리즘 별 인식률 비교 (한국 RFID 데이터 전송률 적용)



Gen 2 protocol basics

Gen2 features

- High data rate
  - R=>T : 26.7 to 128 kbps
  - T=>R : 40 to 640 kbps
- Proven air interface
  - Forward link : PIE ASK
  - Backscatter link : FM0 or Miller-modulated sub-carrier
- Access control and privacy
  - 32-bit kill and access passwords
- Flexible logical layer
  - 16-bit to 496-bit electronic product code (EPC)
  - Optional password-protected access control
  - Optional user memory

Gen2 features(con't)

Reliable operation

- Proven probabilistic/slotted anti-collision
- Adapt to rapidly changing tag populations
- Flexible selection masking
  - Select specific tags for identification

Session and inventoried flags

- Maximum 4 sessions supported
  - Pre-selection of tag groups
- Auto-inactivation by inventoried flags (A, B)

Gen2 process

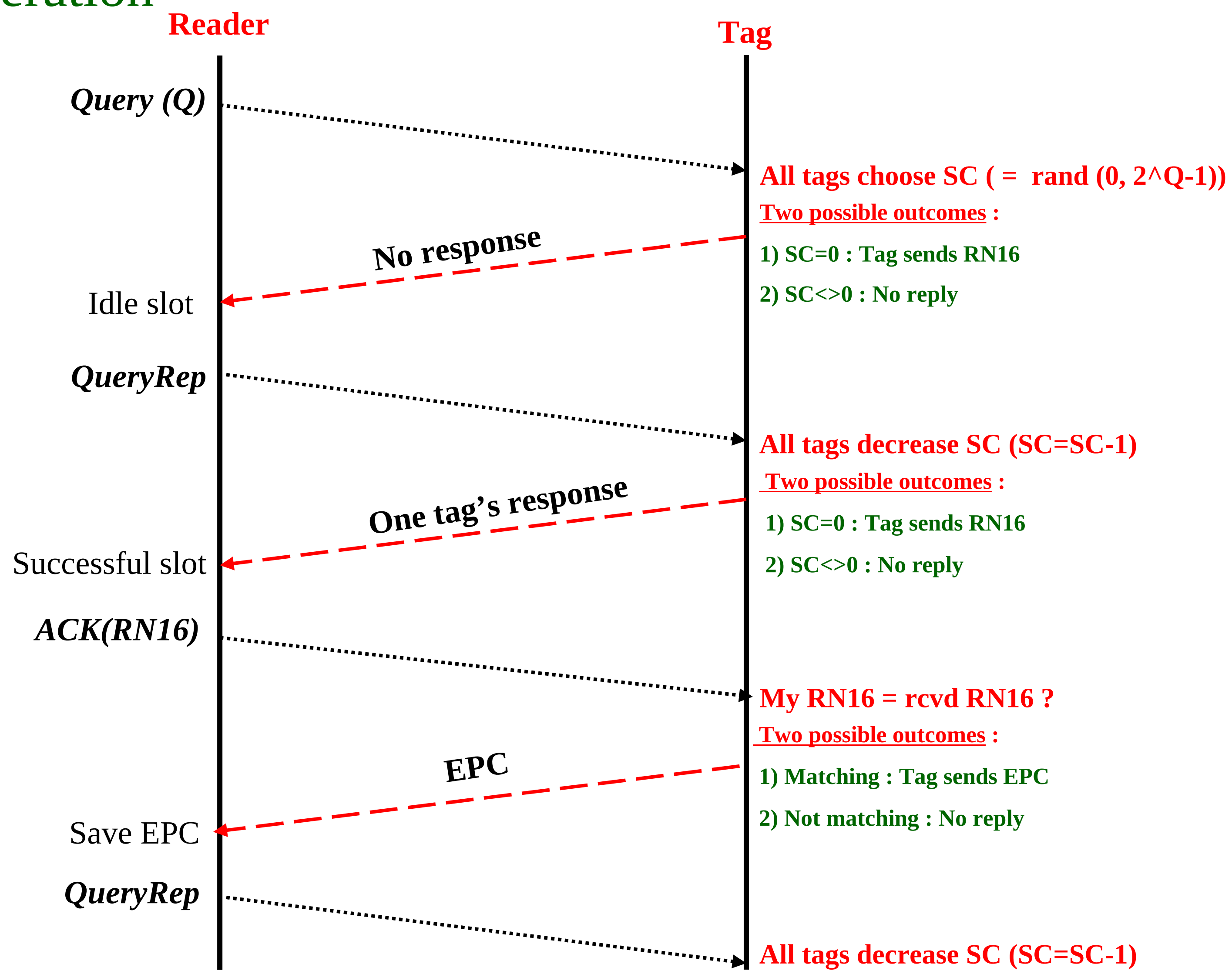
- Select process
  - The process by which an interrogator selects a Tag population for inventory and access. Interrogators may use one or more *Select* commands to select a particular Tag population prior to inventory.
- Inventory process
  - The process by which an Interrogator identifies Tags. An Interrogator begins and inventory round by transmitting a *Query* command in one of four sessions.
- Access process
  - The process by which an Interrogator transacts with (reads from or writes to) individual Tags.

Mandatory commands

- ***Select*** command
  - Select a particular tag population based on user-defined criteria
- ***Query*** command
  - Initiate an inventory round
  - Decide which tags participate in the round
  - Give the seed value, *Q*
- ***QueryAdjust*** command
  - Adjust *Q* (number of slots) without changing any other round parameters
  - Up and down *Q* value or not changing
- ***QueryRep*** command
  - Instruct tags to decrement their slot counters and, if slot=0 after decrementing, to backscatter and RN16 to the reader
- ***ACK*** command
  - Send and *ACK* to acknowledge a single tag.
  - *ACK* echoes the tag’s backscattered RN16
- ***NAK*** command
  - *NAK* shall return all Tags to the **arbitrate** state (initialization)

Inventory process

Basic operation



Reader

Tag

Query (Q)

All tags choose SC (= rand (0, 2^Q-1))

Two possible outcomes :

1) SC=0 : Tag sends RN16

2) SC<>0 : No reply

No response

Idle slot

QueryRep

All tags decrease SC (SC=SC-1)

Two possible outcomes :

1) SC=0 : Tag sends RN16

2) SC<>0 : No reply

One tag's response

Successful slot

ACK(RN16)

My RN16 = rcvd RN16 ?

Two possible outcomes :

1) Matching : Tag sends EPC

2) Not matching : No reply

EPC

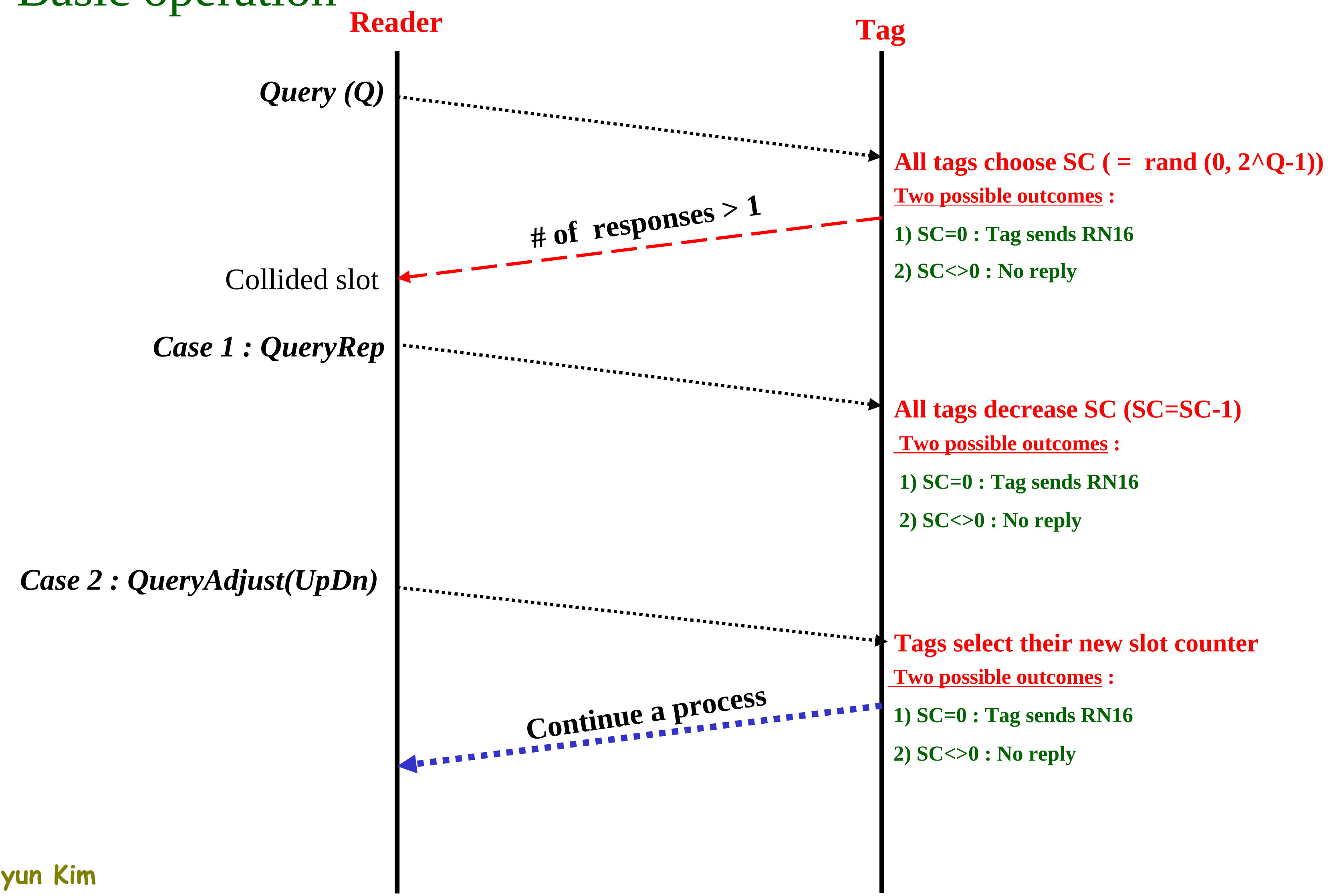
Save EPC

QueryRep

All tags decrease SC (SC=SC-1)

Inventory process

Basic operation



Reader

Tag

Query (Q)

All tags choose SC (= rand (0, 2^Q-1))

Two possible outcomes :

1) SC=0 : Tag sends RN16

2) SC<>0 : No reply

# of responses > 1

Collided slot

Case 1 : QueryRep

All tags decrease SC (SC=SC-1)

Two possible outcomes :

1) SC=0 : Tag sends RN16

2) SC<>0 : No reply

Case 2 : QueryAdjust(UpDn)

Tags select their new slot counter

Two possible outcomes :

1) SC=0 : Tag sends RN16

2) SC<>0 : No reply

Continue a process

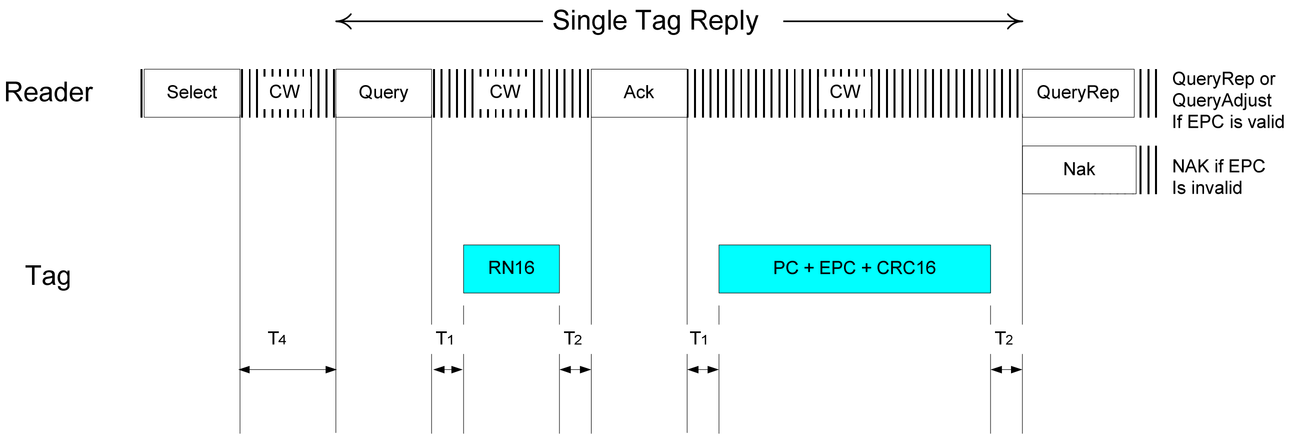
Inventory process

Basic operation

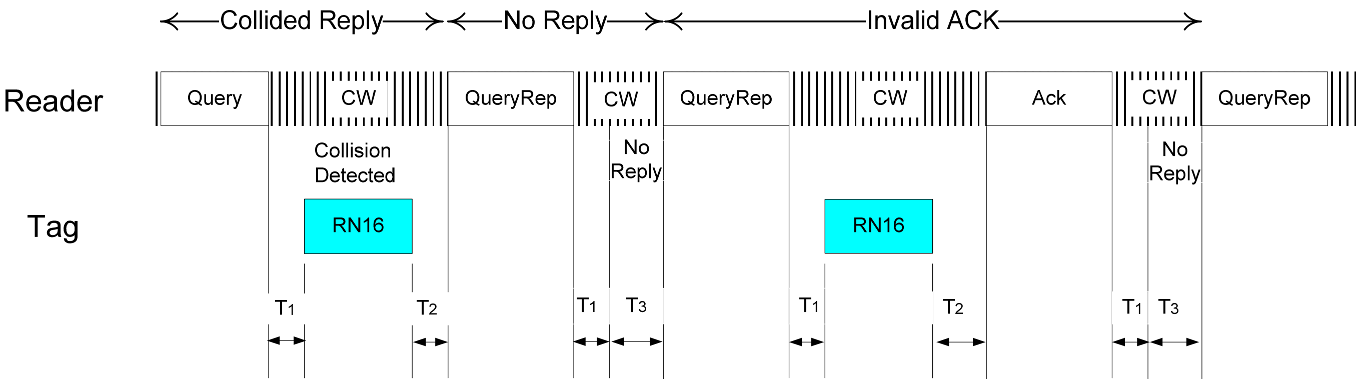
- Reader issues a *Query* command with a parameter Q
  - Starting the inventory round
- Tag generates SC(slot counter) using *Q* value [SC=rand(0,2^Q-1)]
  - If a tag loads a zero, it backscatters an RN16
- Reader acknowledges the tag by returning the RN16
- Acknowledged tag backscatters its EPC
- After the identification of a tag
  - If Reader issues a *QueryRep* command
    - ✓ Tag inverts its **inventoried** flag and leaves the round(inactivation)
    - ✓ All other tags decrement their slot counters
    - ✓ If any tag decrements to zero, it replies with an RN16
  - If Reader issues a *QueryAdjust* command
    - ✓ Reader adjusts Q value adaptively
    - ✓ Tags select their new RN16
    - ✓ If any tag loads to zero, it replies with an RN16

Inventory process

Link timing of Gen 2 protocol



Single tag reply

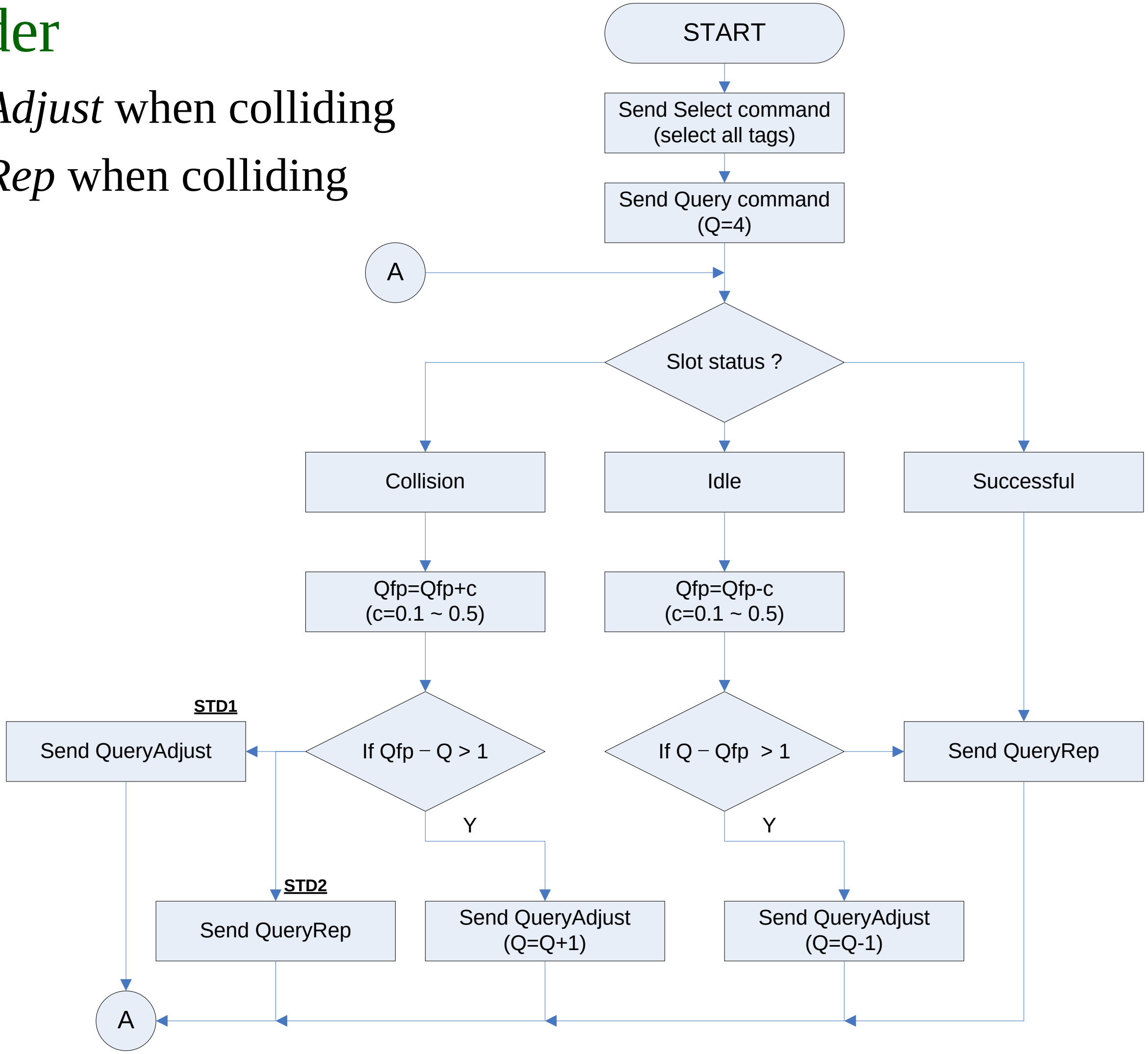


Collided tag reply

Inventory process

Flow chart of a reader

- STD 1 – Send *QueryAdjust* when colliding
- STD 2 – Send *QueryRep* when colliding



Where SC means slot counter which determines whether or not the tags send their RN16.

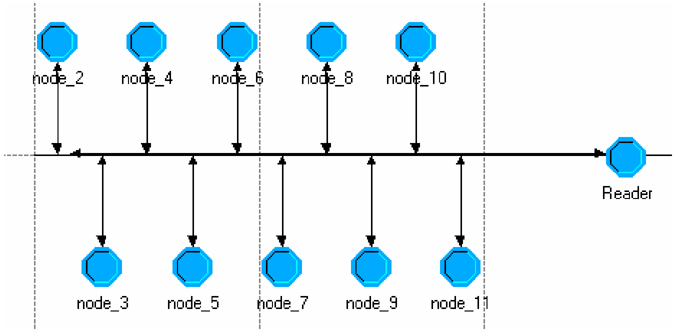
Simulation results

Simulation environments

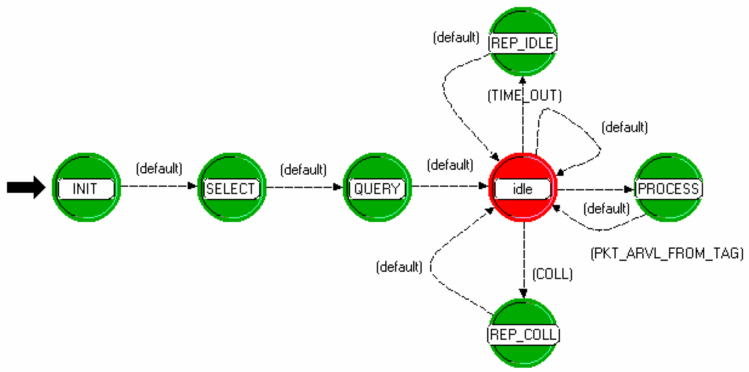
- Assumption
  - Use OPNET simulator
  - Do not consider errors in wireless channel
    - ✓ Actually bus topology used
  - All tags have 96 bits of ID length

- Parameters **[EPC Gen 2]**

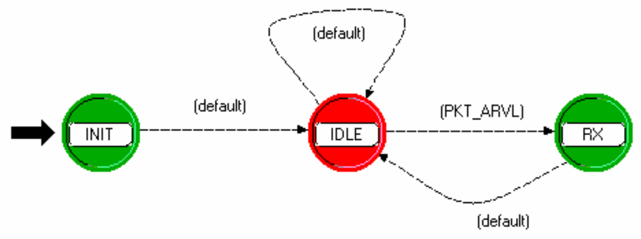
Air interface		Data rate	
Forward	**Backward**	**Forward**	**Backward**
PIE ASK	**FM0**	**40kb/s**	**640kb/s**



Network model



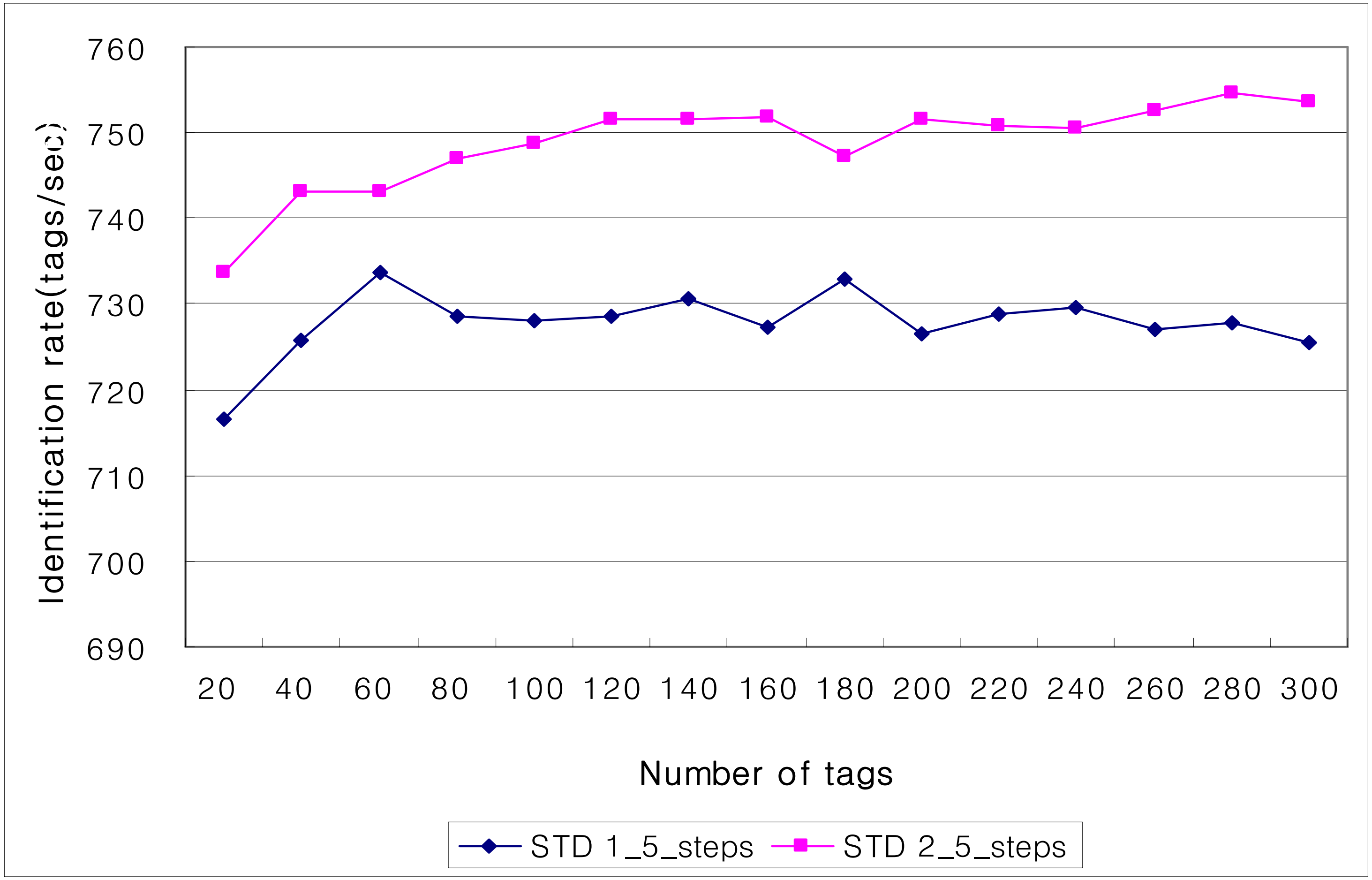
Process model (reader)



Process model (tag)

Simulation results

Identification rate vs. number of tags



- 태그가 **300**개일 때 **STD1** 이 **STD 2**보다 **5%** 정도 성능향상을 보임.

Conclusion (1/2)

- 연구 개발 품목 (계획서 기준)

	결과물명	규격	수량	확인
1	Anti-collision Algorithm 개발 설계서	CD/ Hard Copy	2	제출
2	Algorithm Simulator Tool	CD	2	제출
3	Algorithm Simulation Model 설계서	CD/ Hard Copy	2	제출
4	기존 알고리즘 성능 분석서	CD	2	제출
5	중간보고서	CD/ Hard Copy	2	제출
6	최종보고서	CD/ Hard Copy	2	금일 제출

Conclusion (2/2)

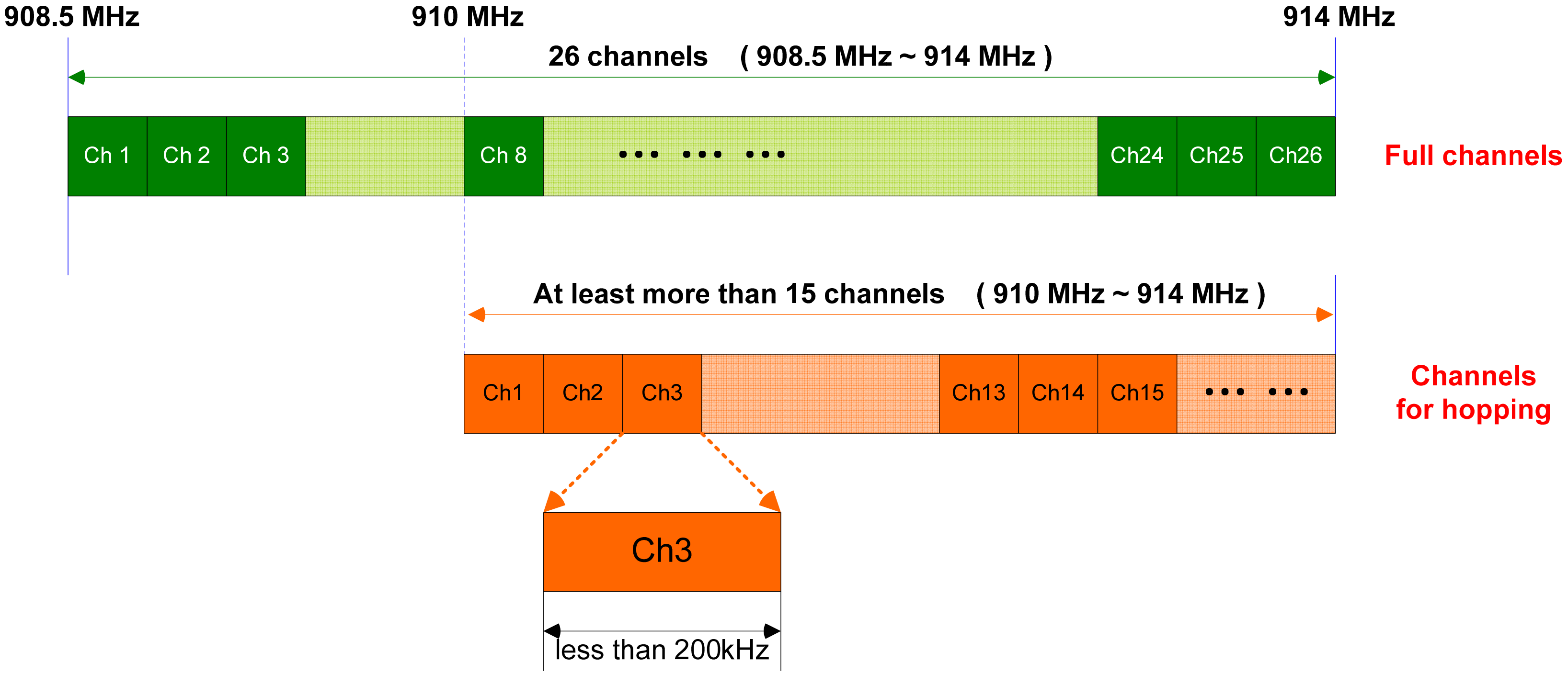
- 연구 실적
  - Alien 社 알고리즘 분석
    - Alien 社 리더의 출력 파형 분석을 통한 동작 과정 분석
    - 탈레스 리더 구현에 적용
  - Bin slot 정보를 이용한 알고리즘 제안
    - PingID 명령만을 사용하는 알고리즘
    - Bit 단위 충돌 응답 정보를 이용하는 알고리즘
    - Bin Slot 내의 충돌 有/無 정보를 이용하는 알고리즘
  - 시뮬레이터 개발
    - 각 알고리즘의 성능 평가를 위한 시뮬레이터 개발
    - 추 후 Gen 2 및 다른 알고리즘의 성능 평가에 확장 및 적용 가능
  - EPC Class 1 binary tree 논문 제출
    - TENCON 2005, EUC2005(LNCS)
- 추 후 예상 되는 결과물
  - 논문 2편
    - "Bin slot 정보를 이용한 EPC Class 1 용 알고리즘 – UCS 2006 (LNCS)
    - "Gen 2 프로토콜 성능 분석 및 응용" – HPCC 2006 (LNCS)

Issues in Gen 2

Issues in Gen 2

Channels for RFID

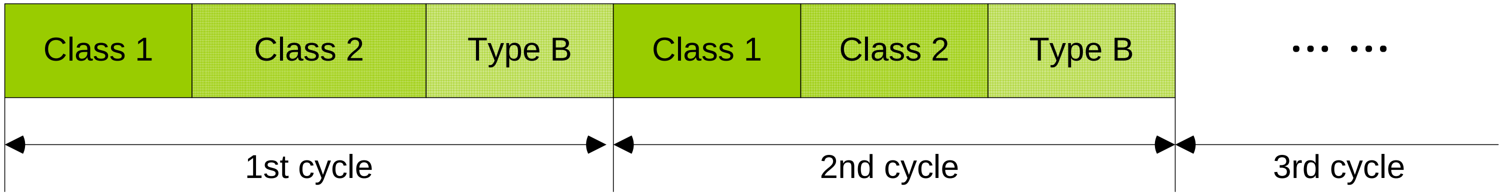
- 908.5 ~ 914 -> 26 channels
- 910 ~ 914 -> at least more than 15 channels when hopping



참고문헌 : 정보통신부고시 제 **2004-66**호 -
“방송.해상.항공.전기통신사업용외의기타업무용무선설비의기술기준 중 제 **5**조의 **2(RFID/USN**용 무선설비**)”**

Issues in Gen 2

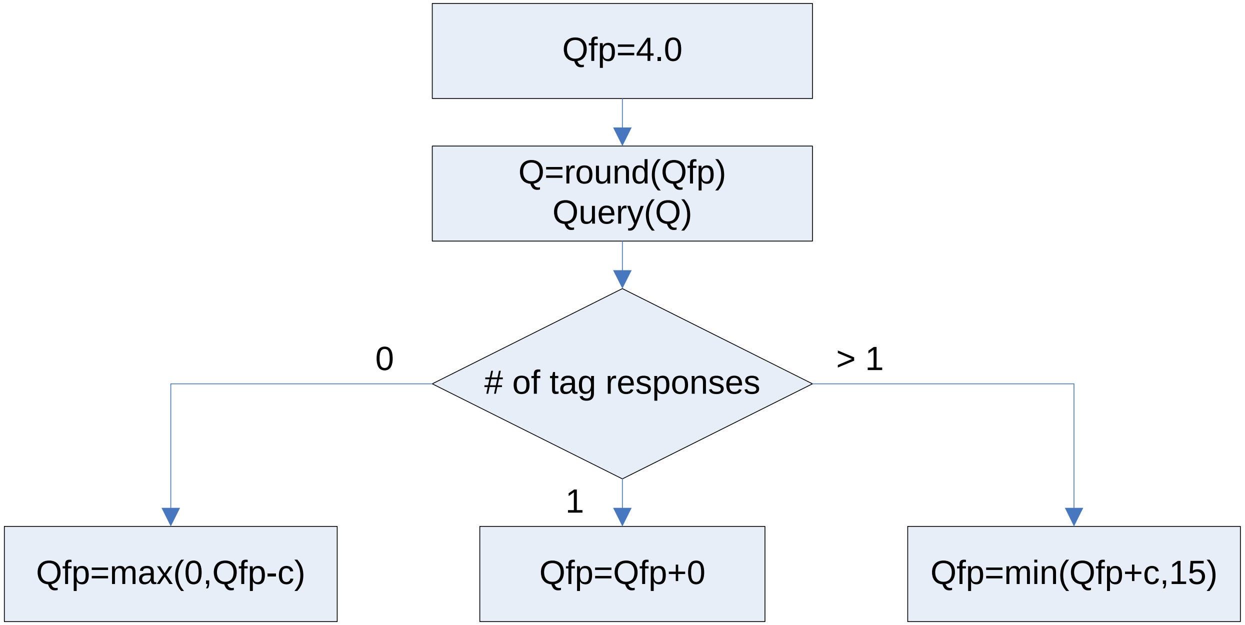
- Minimization of reader-to-reader interference
  - Need the optimal frequency hopping sequence
    - Frequency hopping duration : less than 0.4 s
  - Cell planning
    - Transmission power or distance
- Command combination
  - Which command should be used when colliding or idle !
    - *Query, QueryRep, and QueryAdjust*
- Multi-protocol supporting
  - Generally TDMA used



Gen2 protocol basics

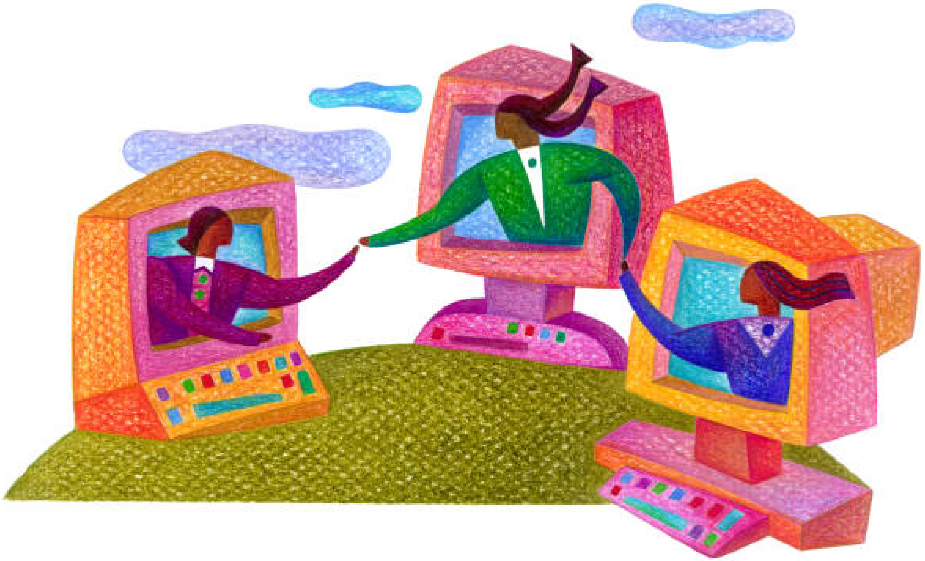
Issues in Gen 2(con't)

- Optimization of *Q*-selection algorithm
  - Need the optimal *c*(0.1~0.5) value according to the number of tags
  - Determination of optimal *Q* value needed



Typical values for C are 0.1<c<0.5. A reader typically uses small values of c when Q is large.

Thank you !!



E-mail : jkim@ajou.ac.kr

Home : Http://winner.ajou.ac.kr

References

[1] K. Finkenzeller, RFID Handbook ; Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition, John Wiley & Sons Ltd, pp.195-219, 2003.

[2] F. Zhou, D. Jin, C. Huang, and M. Hao, "Optimize the power consumption of Electronic Passive Tags for Anti-collision Schemes," IEEE, 2003.

[3] J. L. Massey, "Collision resolution algorithms and random-access communications," Univ. California, Los Angeles, Tech. Rep. UCLAENG -8016, Apr., 1980.

[4] ISO/IEC FDIS 18000-6:2003(E), Part 6: Parameters for air interface communications at 860-960 MHz, Nov. 26, 2003.

[5] M. Jacomet, A. Ehrsam, and U. Gehrig, "Contactless Identification Device With Anticollision algorithm," IEEE Computer Society CSCC’99, Jul. 4-8, Athens., 1999.

[6] 이수련, 이채우, "RFID 시스템에서 고속 인식을 위한 ALOHA 계열 충돌 방지 algorithm의 성능 분석, "JCCI2004, pp. 339, Apr. 28-30, 2004.

[7] Harald Vogt, "Efficient Object Identification with Passive RFID Tags," Pervasive2002, pp.98-113, 2002.

[8] C. S. Kim, K. L. Park, H. C. Kim and S. D. Kim, "An Efficient Stochastic Anti-collision algorithm using Bit-Slot Mechanism," PDPTA04, 2004.

[9] EPC Global, EPCTM Tag Data Standards Version 1.1 Rev.1.24, Apr. 2004.

[10] Auto-ID Center, 900 MHz ISM Band Class 1 Radio Frequency Identification Tag Interface Specification : Candidate Recommendation, Version 1.0.0, 2003.

[11] H. S. Choi, J. R. Cha and J. H. Kim, “Improved Bit-by-bit Binary Tree algorithm in Ubiquitous ID System,” in Proc. IEEE PCM 2004, Tokyo, Japan, Nov. 29 – Dec. 03, 2004, pp. 696-703.

[12] 정보통신부고시 제 2004-66호 , “*방송.해상.항공.전기통신사업용외의기타업무용무선설비의 기술기준 중 제5조의 2(RFID/USN용 무선설비)*”

[13] H. S. Choi and J. H. Kim, "Anti-collision algorithm using Bin slot in RFID System," in Proc. IEEE TENCON '05, Melbourne, Australia, Nov. 21-24, 2005, p.71.

[14] J. R. Cha and J. H. Kim, "Dynamic Framed Slotted ALOHA Algorithm using Fast Tag Estimation method for RFID System," in Proc. CCNC2006, Las Vegas, USA, Jan. 8-10, 2006.

Back-up slides

Select process

Select command

- Select a particular Tag population based on user-defined criteria
- Enable union, intersection, and negation.
- Only one command, *Select*, allowed
- Fields
  - Target and Action indicate whether and how a *Select* modifies a Tag's **SL** or **inventoried** flag.
  - MemBank specifies if the mask applies to EPC, TID, or User memory.
  - Pointer, Length, and Mask : Pointer and Length describe a memory range. Mask, which must be Length bits long, contains a bit string that a Tag compares against the specified memory range.
  - Truncate specifies whether a Tag backscatters its entire EPC, or only that portion of the EPC immediately following Mask.

Selection process

Select command

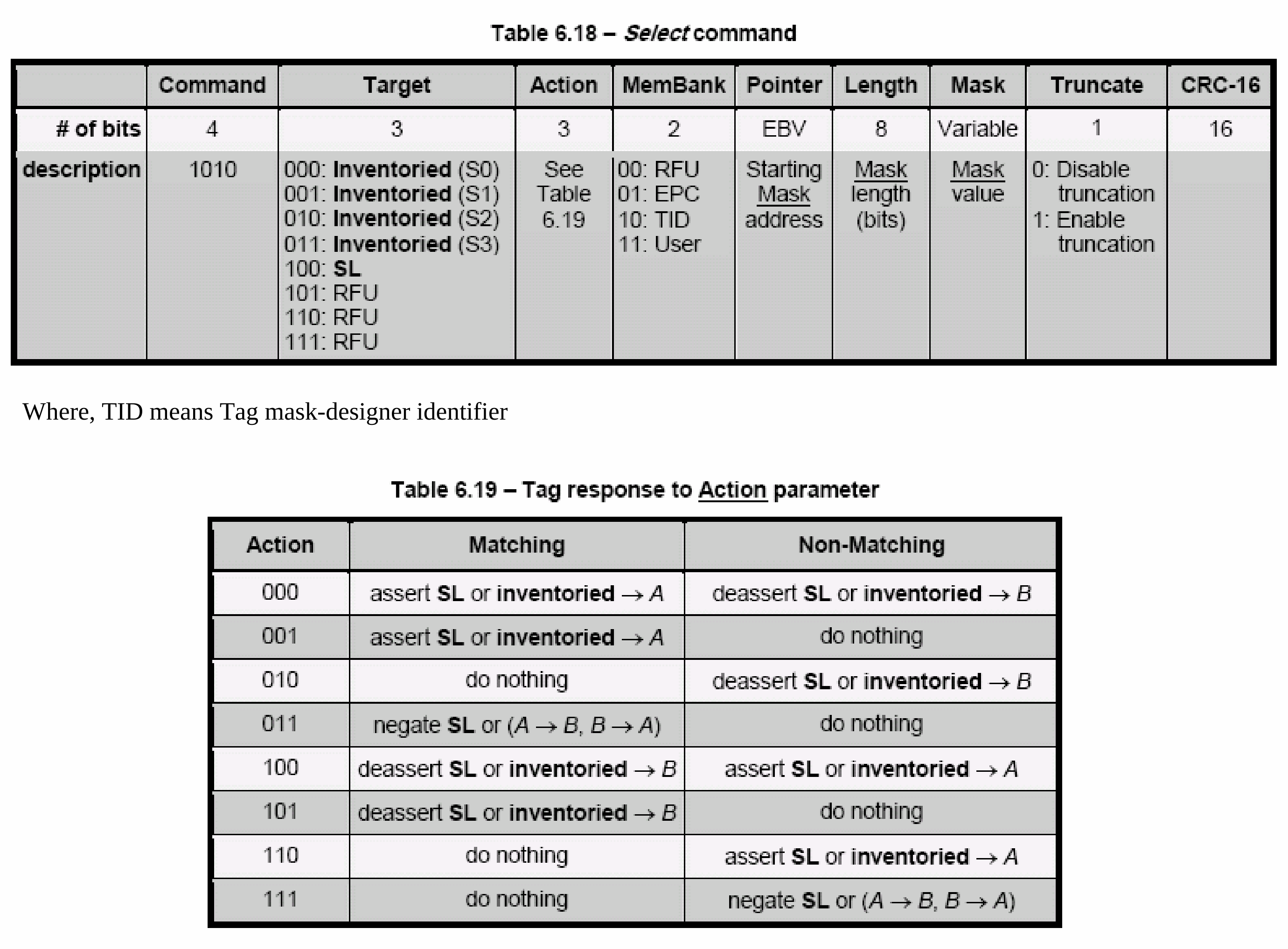


Table 6.18 – *Select* command

	Command	Target	Action	MemBank	Pointer	Length	Mask	Truncate	CRC-16
# of bits	4	3	3	2	EBV	8	Variable	1	16
description	1010	000: **Inventoried** (S0) 001: **Inventoried** (S1) 010: **Inventoried** (S2) 011: **Inventoried** (S3) 100: **SL** 101: RFU 110: RFU 111: RFU	See Table 6.19	00: RFU 01: EPC 10: TID 11: User	Starting Mask address	Mask length (bits)	Mask value	0: Disable truncation 1: Enable truncation	

Where, TID means Tag mask-designer identifier

Table 6.19 – Tag response to Action parameter

Action	Matching	Non-Matching
000	assert **SL** or **inventoried** → *A*	deassert **SL** or **inventoried** → *B*
001	assert **SL** or **inventoried** → *A*	do nothing
010	do nothing	deassert **SL** or **inventoried** → *B*
011	negate **SL** or (*A* → *B*, *B* → *A*)	do nothing
100	deassert **SL** or **inventoried** → *B*	assert **SL** or **inventoried** → *A*
101	deassert **SL** or **inventoried** → *B*	do nothing
110	do nothing	assert **SL** or **inventoried** → *A*
111	do nothing	negate **SL** or (*A* → *B*, *B* → *A*)

Inventory process

Query command

- Initiate an inventory round and decide which Tags participate in the round (where "inventory round" is defined as the period between successive *Query* commands.)
- Fields
  - DR (Trcal divide ratio) sets the T=>R link frequency
  - M (cycles per symbol) sets the T=>R data rate and modulation format
  - TRext chooses whether the T=>R preamble is prepended with a pilot tone
  - Sel chooses which Tags respond to the Query
  - Session chooses a session for the inventory round
  - Target selects whether Tags whose inventories flag is A or B participate in the inventory round. Tags may change their inventoried flag from A to B (or vice versa) as a result of being singulated.
  - Q sets the number of slots in the round

Inventory process

Query command

- Upon receiving a *Query*, Tags with matching Sel and Target pick a random value in the range (0, 2^Q -1), and load this value into their slot counter. If a Tag, in response to the *Query*, loads its slot counter with zero, then its reply to a *Query* shall be as shown in Table 6.21(next slide) ;otherwise the Tag shall remain silent.

- A *Query* may initiate an inventory round in a new session, or in the prior session. If a Tag in the **acknowledged**, **open**, or **secured** states receives a Query whose session parameter matches the prior session, it shall invert its **inventoried** flag (i.e. A->B or B->A) for the session.

Inventory process

Query command

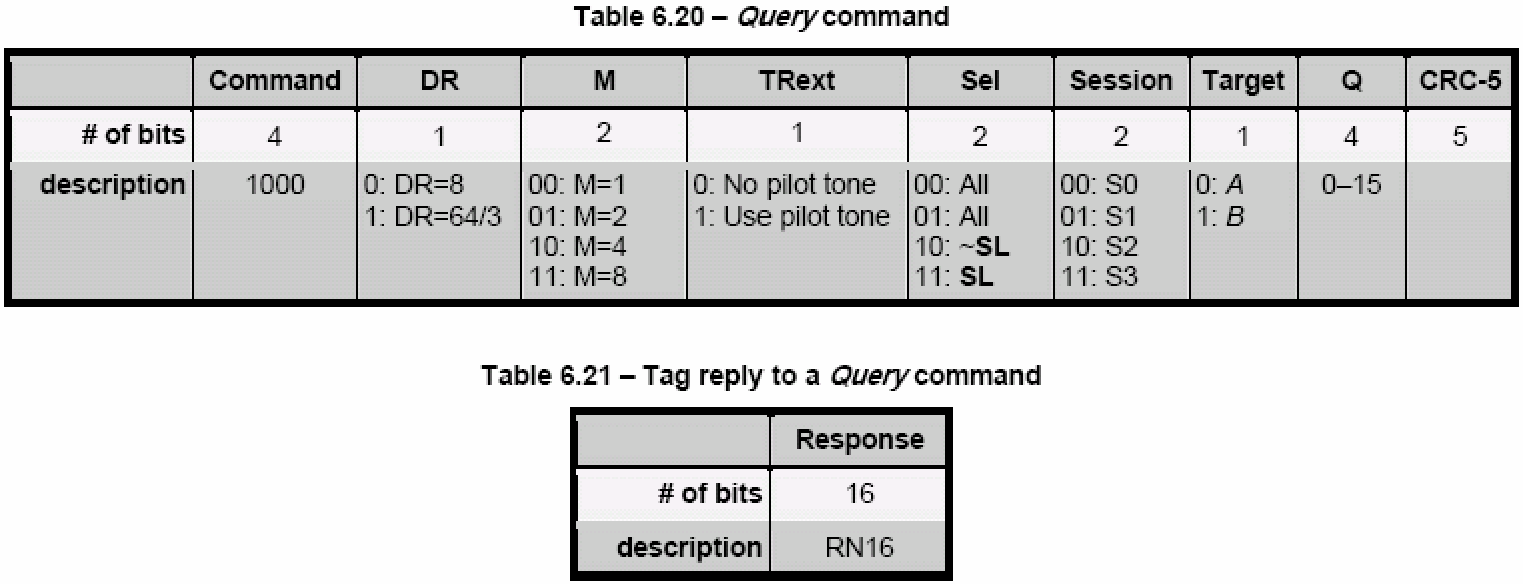


Table 6.20 – *Query* command

	Command	DR	M	TRext	Sel	Session	Target	Q	CRC-5
# of bits	4	1	2	1	2	2	1	4	5
description	1000	0: DR=8 1: DR=64/3	00: M=1 01: M=2 10: M=4 11: M=8	0: No pilot tone 1: Use pilot tone	00: All 01: All 10: ~**SL** 11: **SL**	00: S0 01: S1 10: S2 11: S3	0: *A* 1: *B*	0–15	

Table 6.21 – Tag reply to a *Query* command

	Response
# of bits	16
description	RN16

Inventory process

QueryAdjust command

- QueryAdjust adjusts Q (i.e. the number of slots in an inventory round) without changing any other round parameters.
- Fields
  - Session corroborates the session number for the inventory round.
  - UpDn determines whether and how the Tag adjusts Q, as follows:
    - ✓ 110 : Increment Q (i.e. Q=Q+1)
    - ✓ 000 : No change to Q.
    - ✓ 011 : Decrement Q (i.e. Q=Q-1)
- Upon receiving a *QueryAdjust* Tags first update Q, then pick a random value in the range (0. 2^Q-1), and load this value into their slot counter.
- Tags shall respond to a *QueryAdjust* only if they received a prior *Query*.

Inventory process

- *QueryAdjust* command

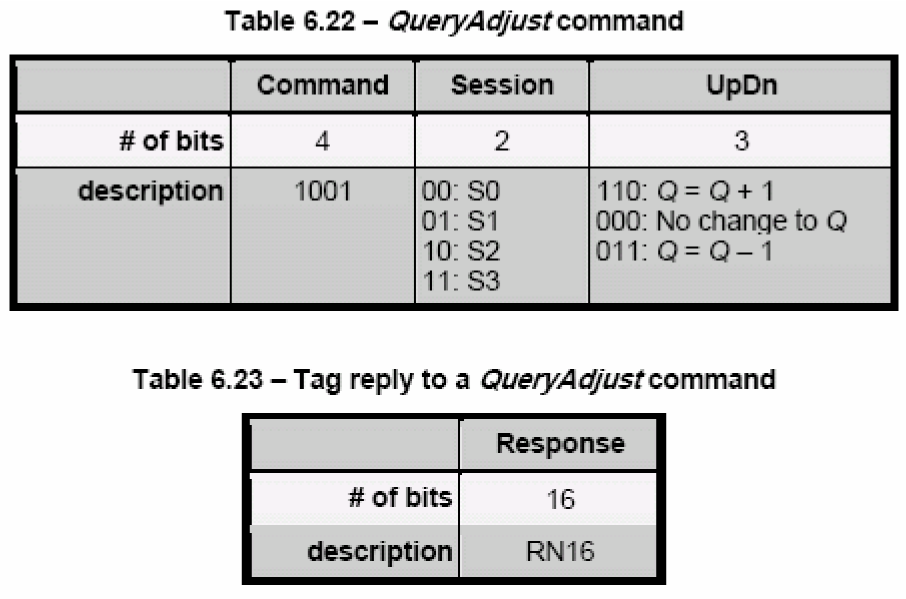


Table 6.22 – *QueryAdjust* command

	Command	Session	UpDn
# of bits	4	2	3
description	1001	00: S0 01: S1 10: S2 11: S3	110: *Q* = *Q* + 1 000: No change to *Q* 011: *Q* = *Q* – 1

Table 6.23 – Tag reply to a *QueryAdjust* command

	Response
# of bits	16
description	RN16

Inventory process

QueryRep command

- *QueryRep* instructs Tags to decrement their slot counters and, if slot=0 after decrementing, to backscatter and RN16 to the Interrogator.
- Fileds
  - Session corroborates the session number for the inventory round.
    - ✓ If a Tag receives a *QueryRep* whose session number is different from the session number in the *Query* that initiated the round it shall ignore the command.
- If a Tag, in response to QueryRep, decrements its slot counter and the decremented slot value is zero, then its reply to QueryRep shall be as shown in Table 6.25(next slide) ; otherwise the Tag shall remain silent.
- Tags in the **acknowledged**, **open** or **secured** states that receive a *QueryRep* invert their **inventoried** flag (i.e. A->B or B->A, as appropriate) for the current session and transition to **ready**.

Inventory process

QueryRep command

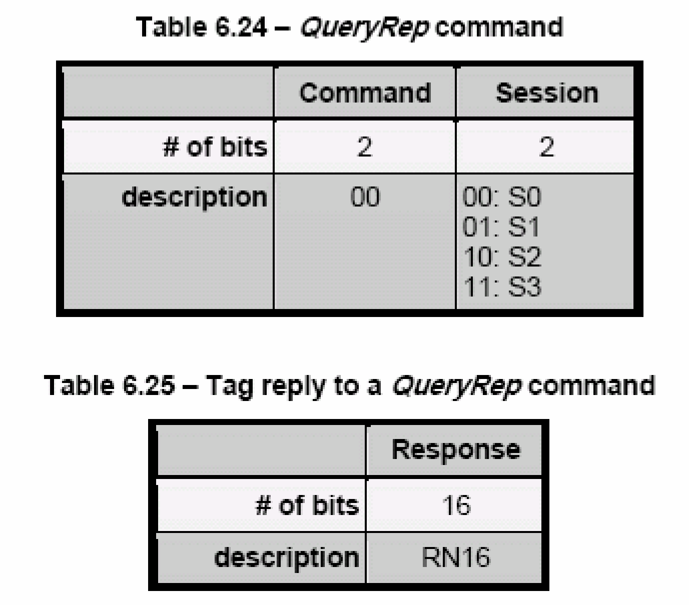


Table 6.24 – *QueryRep* command

	Command	Session
# of bits	2	2
description	00	00: S0 01: S1 10: S2 11: S3

Table 6.25 – Tag reply to a *QueryRep* command

	Response
# of bits	16
description	RN16

Inventory process

ACK command

- An Interrogator sends and *ACK* to acknowledge a single Tag.
- *ACK* echoes the Tag’s backscattered RN16.
- An Interrogator issues an *ACK* to a Tag
  - If the received RN16 is equal to my RN 16
    - ✓ Send EPC
  - If the received RN16 is not equal to my RN 16
    - ✓ Do nothing

Inventory process

ACK command

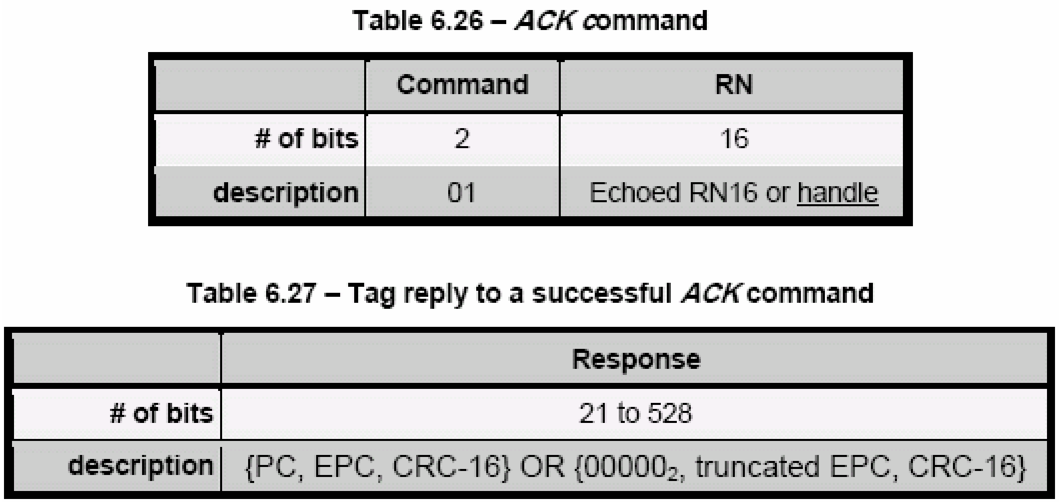


Table 6.26 – *ACK* command

	Command	RN
# of bits	2	16
description	01	Echoed RN16 or handle

Table 6.27 – Tag reply to a successful *ACK* command

	Response
# of bits	21 to 528
description	{PC, EPC, CRC-16} OR {00000_2, truncated EPC, CRC-16}

Inventory process

NAK (mandatory)

- *NAK* shall return all Tags to the **arbitrate** state unless they are in **ready** or **killed**.
- Tags shall not reply to a *NAK*.

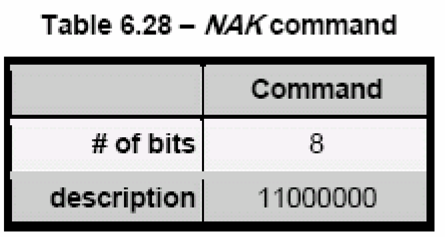


Table 6.28 – *NAK* command

	Command
# of bits	8
description	11000000