



Protocolli di anti-collisione per sistemi RFID

Gaia Maselli
maselli@di.uniroma1.it

Tecnologia RFID

➤ **RFID - Radio Frequency Identification**

Tecnologia che consente l'identificazione automatica di oggetti.

➤ **La spesa oggi**

- I prodotti sono identificati da codici a barre che devono essere letti uno per volta alla cassa

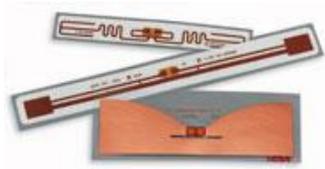
➤ **La spesa domani**

- Si può passare in cassa senza dover svuotare il carrello, avendo il conto in pochi istanti



➤ Non c'è necessità di visibilità diretta come nel caso dei codici a barre

Cosa è un sistema RFID?



RF Tags



Interrogators
and Antennas



Server
& Data repositories

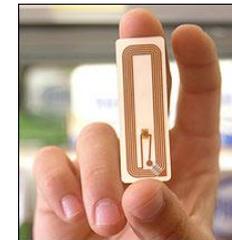
Le etichette a radio frequenza memorizzano un identificativo univoco (es.96 bits) e consistono di un microchip connesso a una piccola antenna. Vengono attaccati agli oggetti da identificare.

Il reader interroga i tag per acquisire informazioni (Identificativo)

Il server gestisce i dati ricevuti dal reader e li processa in base all'applicazione in esecuzione.

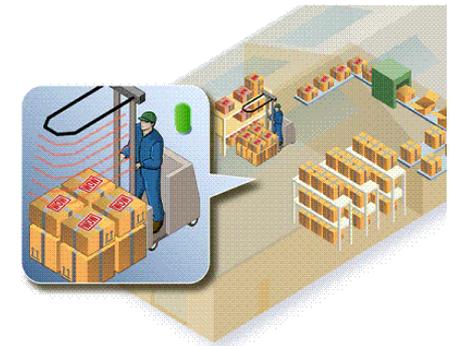
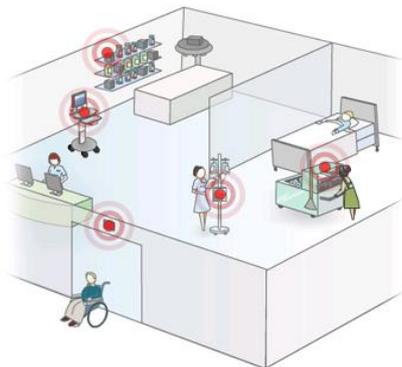
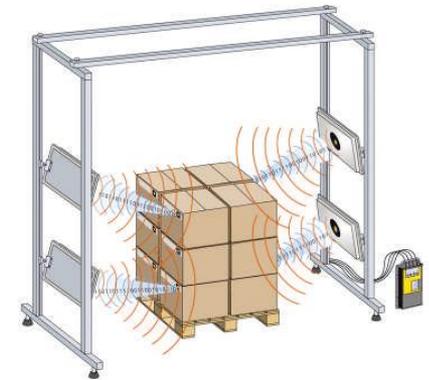
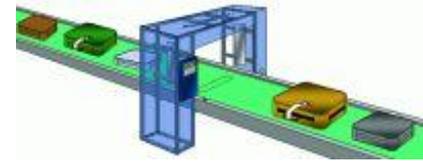
Sistemi RFID con tag passivi

- Molto piccoli, economici, duraturi
- Nessuna fonte di energia locale (batteria)
- Trasmissione mediante *back-scattering*:
 - Traggono la potenza necessaria ad attivare i circuiti dalle onde radio inviate dal reader che li interroga e induce una corrente nell'antenna.
- Tag attivi: sono alimentati da batterie, che offrono una maggiore portata al segnale radio e una maggiore distanza di lettura
 - Molto + costosi!



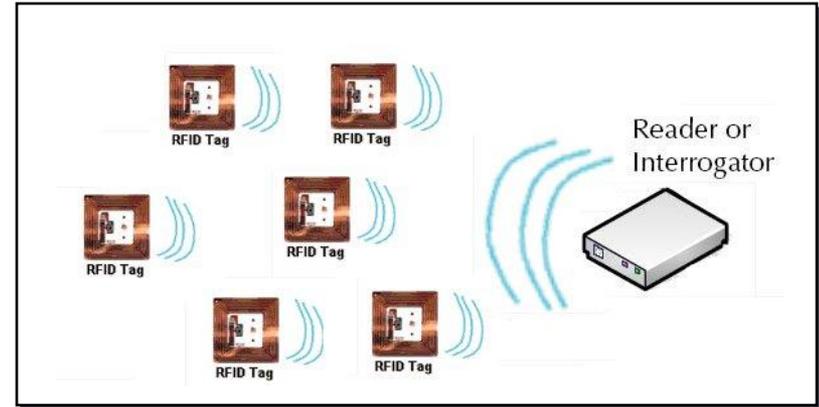
Applicazioni

- Inventari e logistica
- Controllo accessi e tracciamento prodotti
 - Biblioteche
 - Bagagli in aeroporto
- Domotica e Assisted Living
 - Elettrodomestici intelligenti
 - Assistenza giornaliera per categorie deboli (disabili)



Un sistema RFID è una rete

- Sistemi con un solo reader e tag passivi
- Comunicazione
 - Il reader interroga i tag
 - I tag rispondono al reader inviando il loro ID



Aspetti chiave

- I tag che rispondono simultaneamente causano collisione
- I tag non possono sentire la trasmissione degli altri tag



L'accesso al canale deve essere gestito dal reader

Identificare = evitare le collisioni

- Un protocollo di identificazione deve
 - interrogare i tag in modo da ottenere il più possibile risposte singole (identificazioni)
 - minimizzare le risposte simultanee (o collisioni che non consentono l'identificazione)
- Protocollo di identificazione \Rightarrow protocollo di anti-collisione o accesso al mezzo condiviso (MAC)

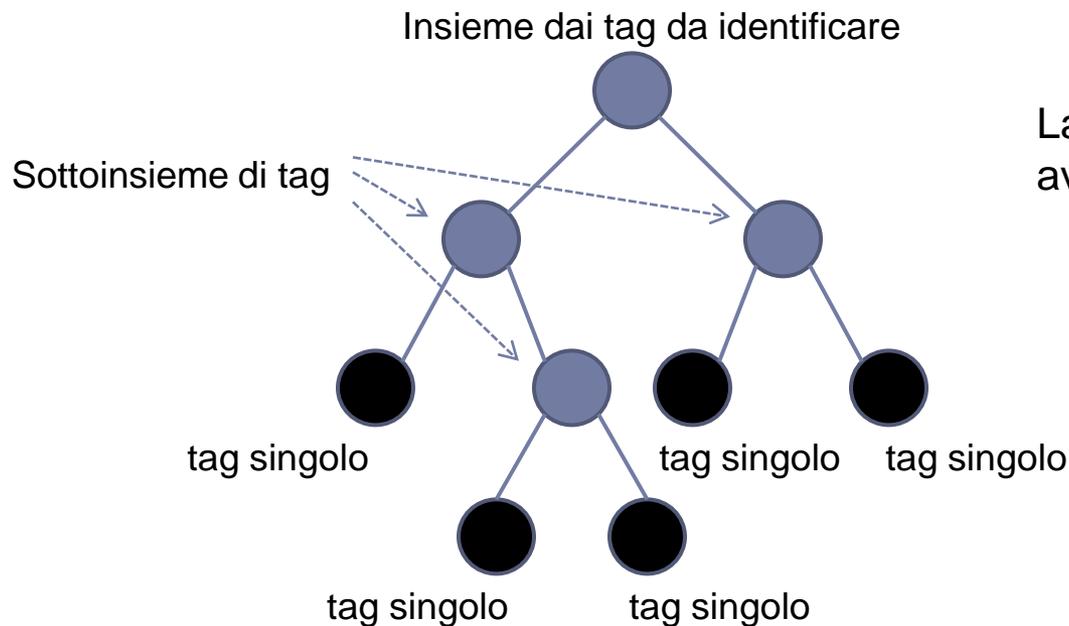
Goal \rightarrow Identificare in maniera efficace ed efficiente gli oggetti a cui i tag sono applicati

Protocolli di anticollisione

- ▶ Si suddividono in 2 classi
 - ▶ Protocolli basati su *alberi (tree-based)*
 - ▶ *Domanda - risposta*
 - ▶ *Deterministici (in realtà uno di essi è randomico)*
 - ▶ Protocolli basati sul meccanismo aloha (*aloha-based*)
 - ▶ *Tempo slottizzato*
 - ▶ *Randomici*

Protocolli di anticollisione: tree-based

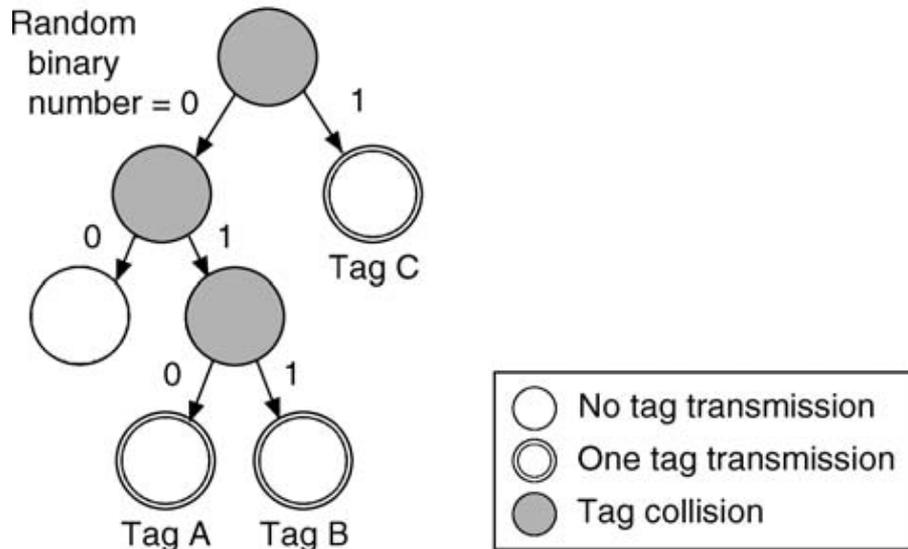
- ▶ Seguono una struttura ad albero binario per la ricerca degli identificativi univoci dei Tag (EPC)
 - Nodo radice , insieme dei Tag da identificare
 - Nodi intermedi, gruppi di Tag che generano collisione
 - Nodi foglia, Tag identificati



La suddivisione in sottoinsiemi avviene in base a meccanismi diversi

Tree-based: Binary splitting (BS)

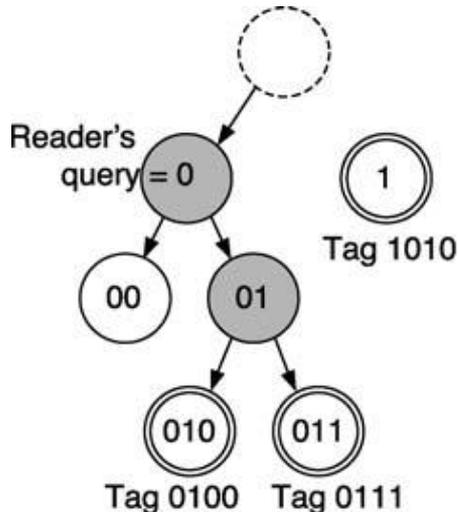
- ▶ I tag vengono suddivisi in sottoinsiemi in base alla generazione di un numero random interna al tag



- ▶ I tag hanno un contatore inizializzato a 0
- ▶ I tag trasmettono il loro ID quando il contatore è nullo
- ▶ Il reader informa i tag sull'esito della query (identificazione, collisione, nessuna risposta)
- ▶ I tag aggiornano il contatore in base all'esito della query:
 - ▶ Collisione: i tag che non hanno trasmesso incrementano di uno il contatore mentre quelli che hanno trasmesso generano un numero random $(0,1)$ e lo sommano al contatore
 - ▶ Non collisione (identificazione o nulla): tutti i tag decrementano di uno il contatore

Tree-based: Query Tree (QT)

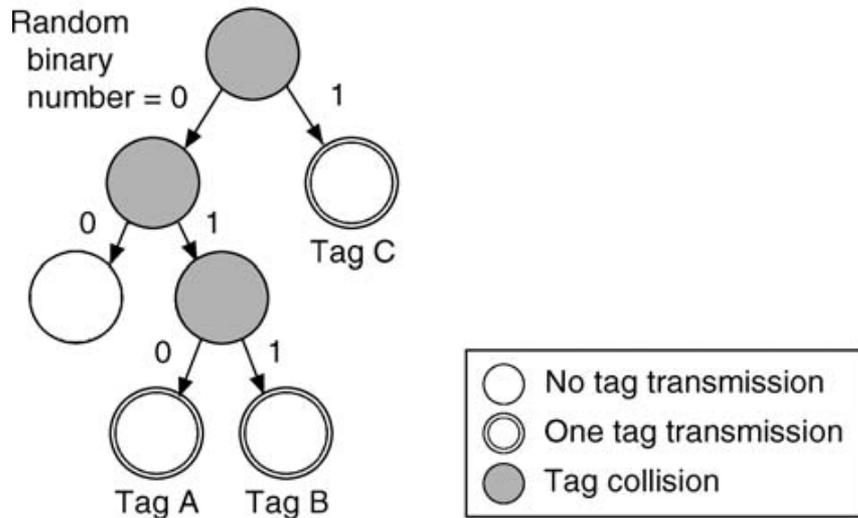
- ▶ I tag vengono suddivisi in sottoinsiemi in base alla struttura binaria del loro ID



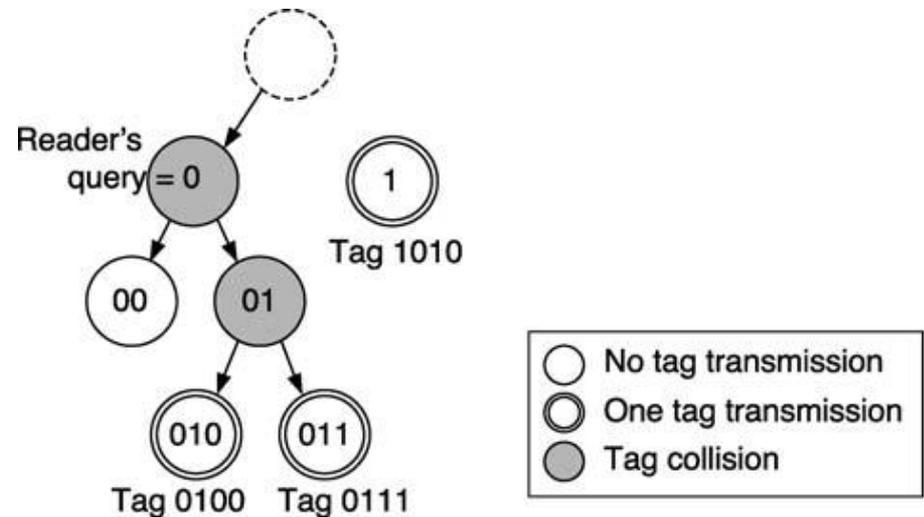
- ▶ Il reader invia una query che contiene una stringa di bit
- ▶ I tag che hanno i primi bit (prefisso) uguale alla stringa nella query rispondono inviando il loro ID
- ▶ Se c'è una collisione sulla stringa $q_1q_2\dots q_x$ ($q_i \in \{0,1\}$, $1 \leq x < b$, and b è il numero di bit nell'ID), il reader aggiunge un bit (0 e 1) alla stringa e invia due nuove query contenenti le stringhe: $q_1q_2\dots q_x0$ e $q_1q_2\dots q_x1$
- ▶ L'insieme dei tag che hanno colliso viene quindi suddiviso in due sottoinsiemi

Binary Splitting vs Query Tree

- ▶ Gli alberi di identificazione erano simili
- ▶ La generazione del numero random interna al tag nel BS rispecchia la generazione random dei bit nell'ID su cui si basano le interrogazioni nel QT



Binary Splitting (BS)



Query Tree (QT)

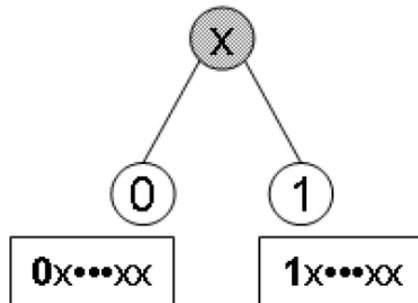
Query Tree Improved (QTI)

- ▶ Ottimizza il numero di interrogazioni, evitando quelle che sicuramente collidono
- ▶ Esempio:
 - ▶ La query con prefisso “*p*” produce una collisione
 - ▶ La query con prefisso “*p0*” non ottiene alcuna risposta
 - ▶ La query con prefisso “*p1*” viene saltata perché produrrà sicuramente collisione, mentre si eseguono direttamente “*p10*” e “*p11*”

Effetto della distribuzione degli ID sul QT (1/3)

- ▶ Caso migliore: minimo numero di collisioni e nessun idle (query senza risposta)

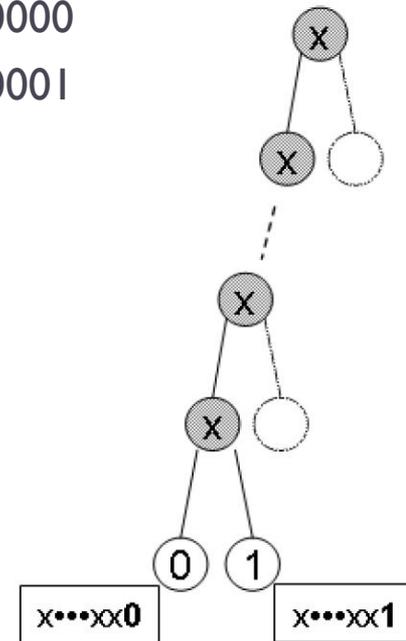
- ▶ 00000
- ▶ 10000



(b): Most significant bit

- ▶ Caso peggiore: massimo numero di collisioni

- ▶ 00000
- ▶ 00001



(a): Least significant bit

Il numero di nodi nell'albero fornisce una stima della complessità dell'algoritmo (più query → maggior tempo di identificazione)

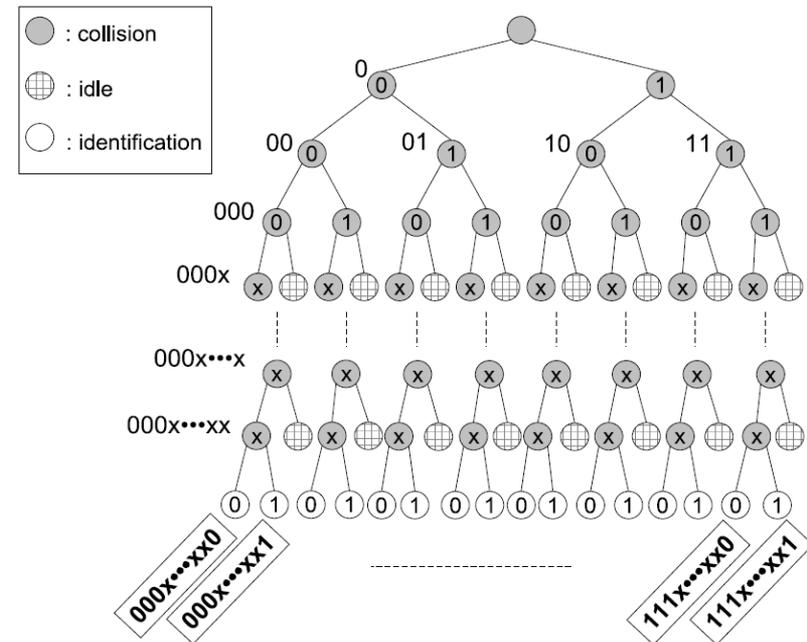
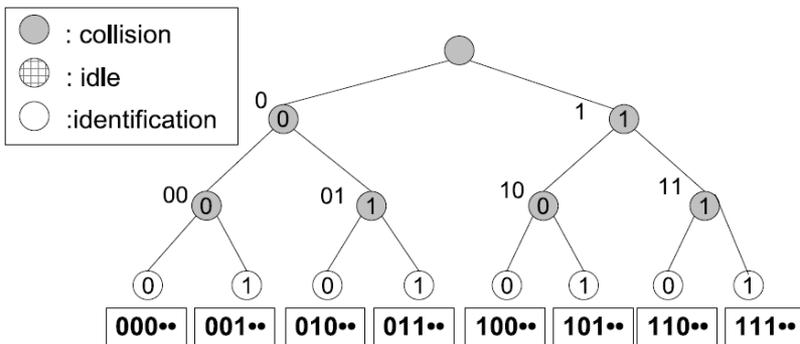
Effetto della distribuzione degli ID sul QT (2/3)

► Distribuzione ottima

- È unica?

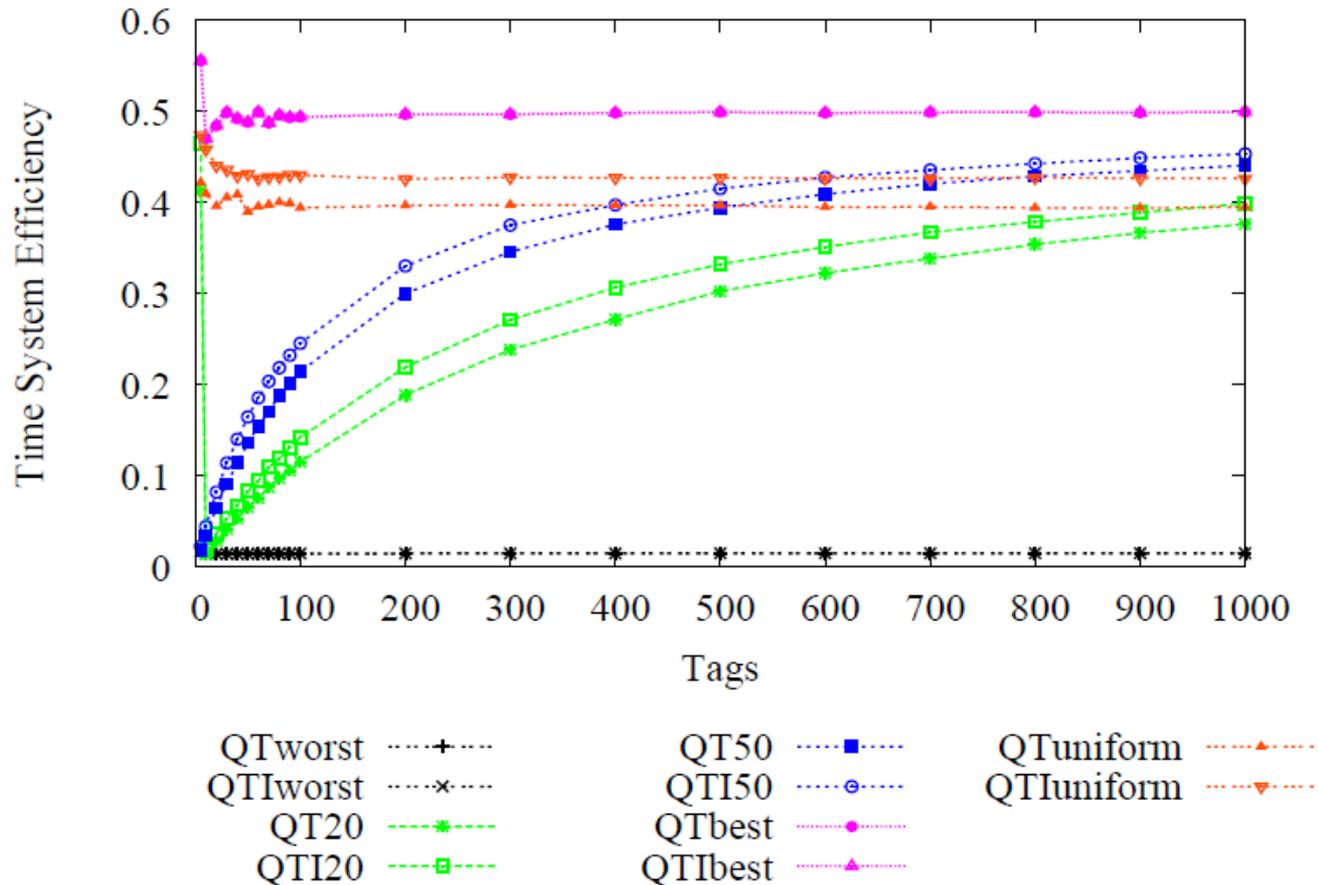
► Distribuzione pessima

- È unica?



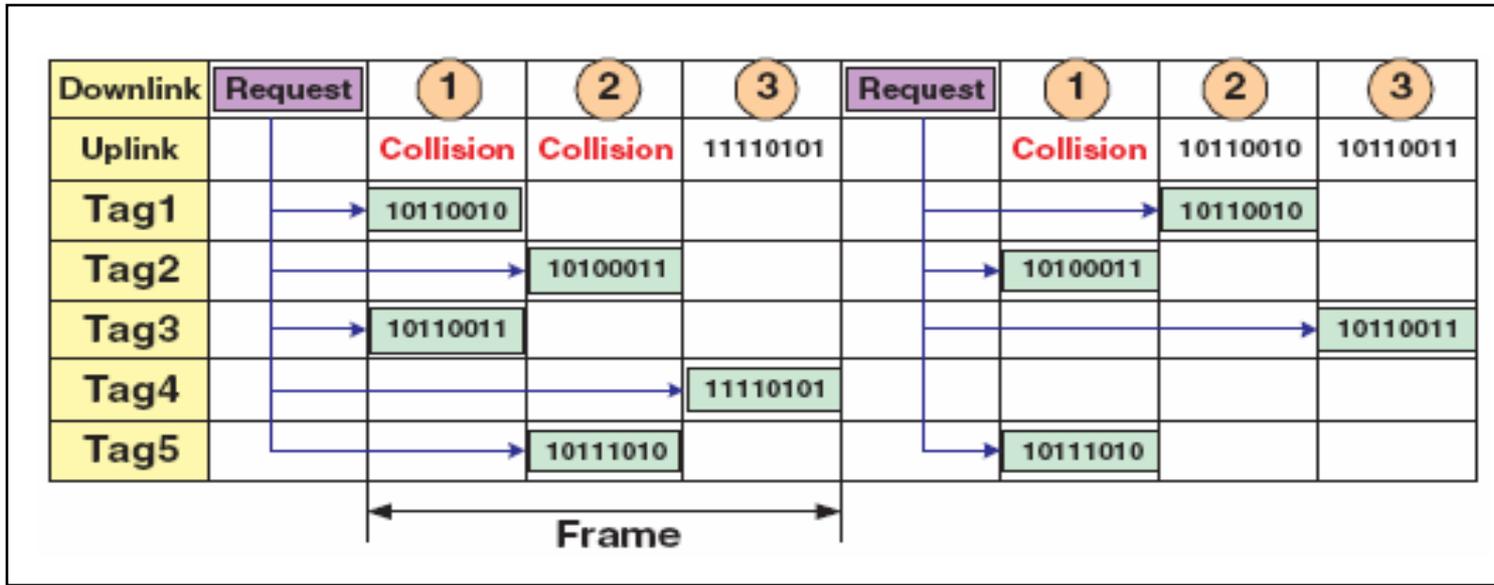
Effetto della distribuzione degli ID sul QT (3/3)

► Efficienza temporale con varie distribuzioni



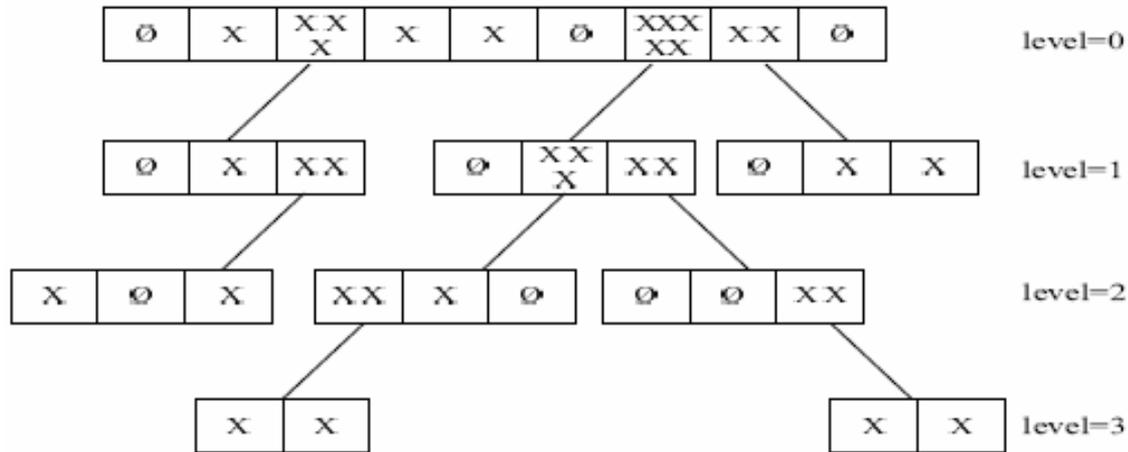
Framed Slotted Aloha

- Aloha slottizzato (selezione randomica dello slot)

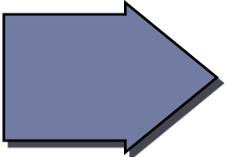


- 6 slot eseguiti: 3 collisioni + 3 identificazioni
- Efficienza del protocollo = # identificazioni / # slot eseguiti = 50%
- In generale
 - 37% di identificazioni (N.B. quando il frame ha tanti slot quanti sono i tag)
 - Il rimanente 63% è sprecato in collisioni e risposte nulle

Tree Slotted Aloha (TSA)



- ▶ A new *child* frame is issued for each collision slot: only tags replying to the same slot participate
- ▶ Child frames should be sized properly according to the number of colliding tags



Estimating tag population to properly tune frame sizes

Tag estimation issues

- ▶ How to set the initial frame size (the number of tags is not known)
- ▶ How to estimate the number of tags that collide in the same slot) and properly tune the following frames
 - ▶ True in any aloha protocol

Estimating tag population

- ▶ The number of tags to be identified is not known
 - ▶ The initial frame size is set to a predefined value (i.e., 128)
 - ▶ The size of the following frames is estimated

$$\text{tags per collision slot} = \frac{(\text{estimated total num of tags}) - (\text{identified tags})}{\text{collision slots}}$$

- ▶ The total number of tags is estimated according to the outcome of the previous frame (based on Chebyshev's inequality)

$$\varepsilon(N, c_0, c_1, c_k) = \min_n \left| \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ a_k^{N,n} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right|$$

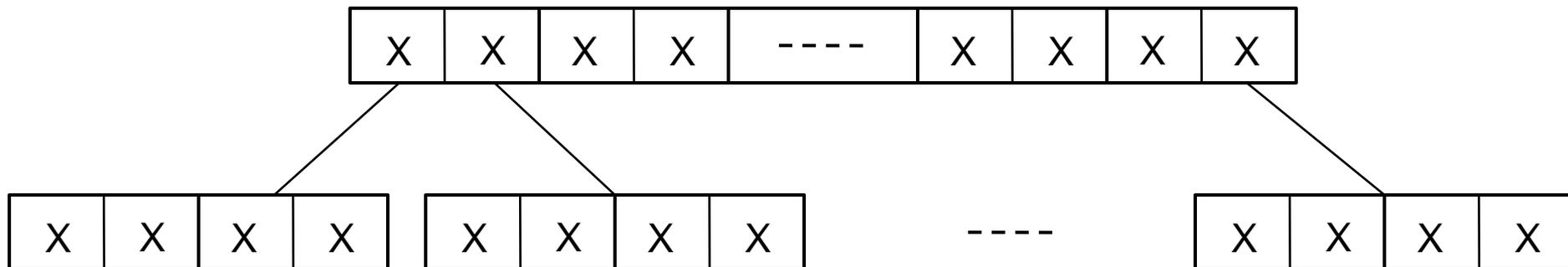
- ▶ N : size of completed frame
- ▶ $\langle c_0, c_1, c_k \rangle$ triple of observed values
- ▶ $\langle a_0, a_1, a_k \rangle$ triple of estimated values

- ▶ Given N and a possible value of n , the expected number of slots with r tags is estimated as

$$a_r^{N,n} = N \times \binom{n}{r} \left(\frac{1}{N} \right)^r \left(1 - \frac{1}{N} \right)^{n-r}$$

Inaccuracy of tag estimation for large networks

- ▶ The estimator does not capture the possibly high variance of the number of tags
- ▶ The minimum is computed over n ranging in $[c_1 + 2c_k, 2(c_1 + 2c_k)]$
- ▶ The upper bound $2(c_1 + 2c_k)$ is not adequate for network composed of thousands of nodes
 - ▶ Example: 5000 tags, $N=128$, it is highly likely that $c_1=0$
 n is estimated $2(c_1 + 2c_k) = 512$  definitively too small



Only 4 slots for an expected number of colliding tags around 40!

1° solution: Unbounded estimator

- ▶ Let's search for a better upper bound
- ▶ Let's not stop at $2(c_l + 2c_k)$
- ▶ For $N=128$ and $\langle c_0, c_l, c_k \rangle = \langle 0, 0, 128 \rangle$, the table shows the triple of estimated values and their distance from observed value by varying n

Varying n ↓

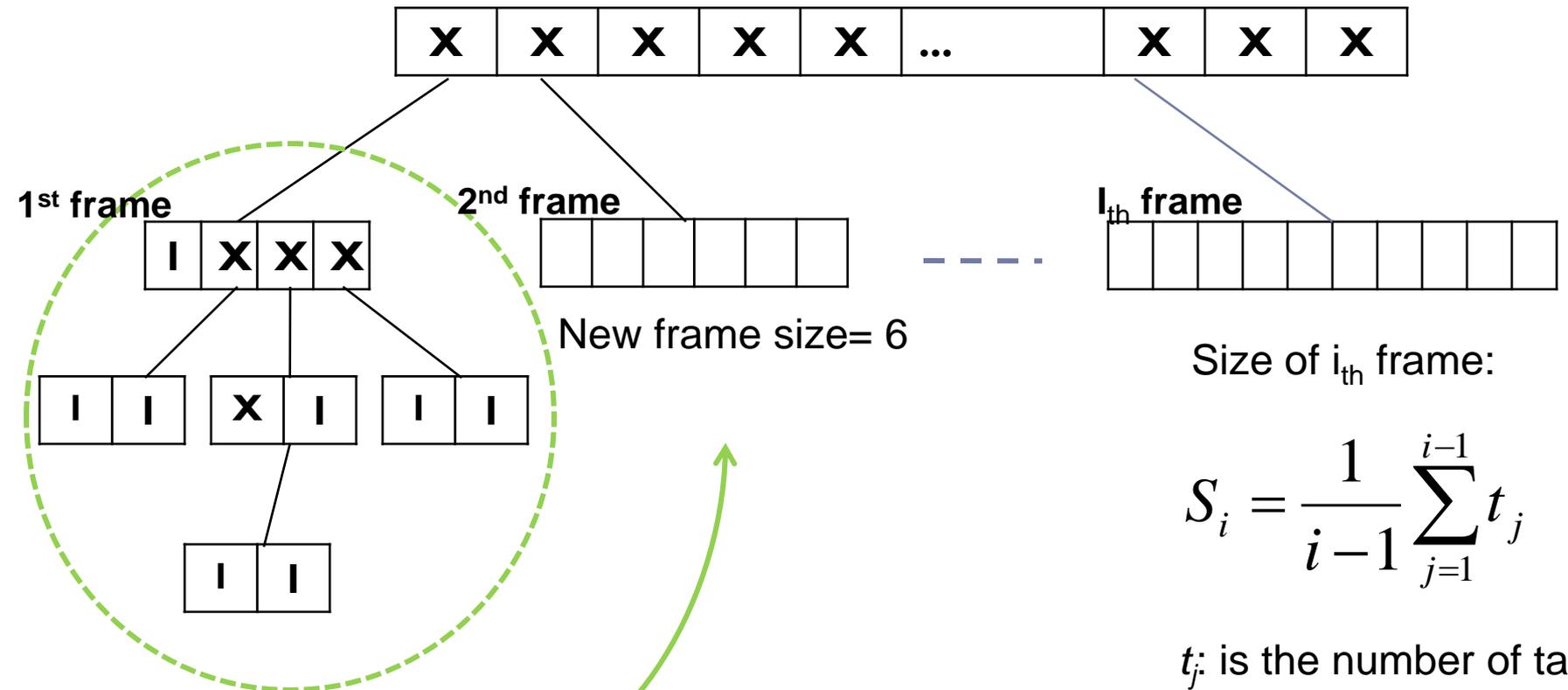
n	vect. distance	a_0	a_1	a_k
256	64.671	17.187	34.645	76.167
500	16.211	2.536	9.983	115.482
700	4.537	0.528	2.912	124.560
800	2.337	0.241	1.519	126.240
900	1.188	0.110	0.780	127.110
1000	0.598	0.050	0.396	127.554
1500	0.017	0.001	0.012	127.987
2000	0.0005	0.00002	0.0003	127.9997

→ still not accurate!

Dynamic Tree Slotted Aloha (Dy_TSA)

- ▶ **Dynamic** tag estimation that exploits the knowledge gained during previously completed frames
- ▶ Assumption: tags are uniformly distributed among all slots
 - ▶ The expected number of tags in a slot is $E[X] = \frac{n}{N}$
 - ▶ Satisfied for when $n \gg N$

Dy_TSA: dynamic tag estimation



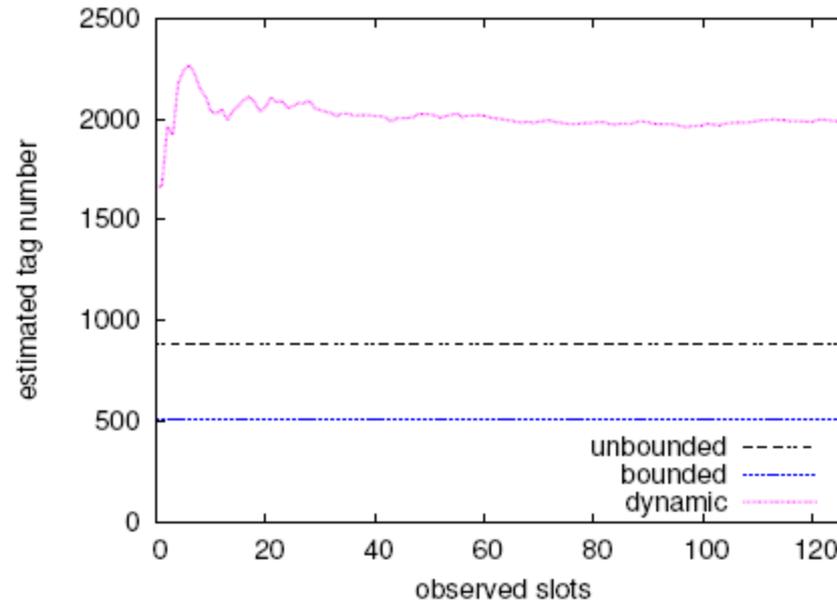
$$S_i = \frac{1}{i-1} \sum_{j=1}^{i-1} t_j$$

t_j is the number of tags that participated to frame j

8 tags found!

- ▶ As TSA proceeds in depth-first order, the estimation method can be recursively applied on deeper levels of the tree

Accuracy of dynamic tag estimation



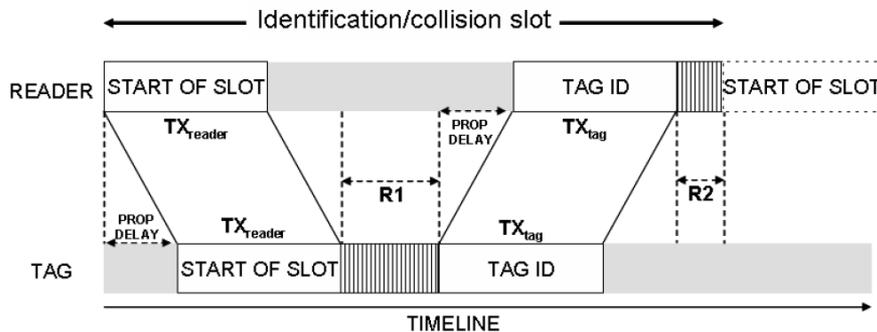
Estimated number of tags as slots of the first frame are resolved (n=2000)

Performance evaluation

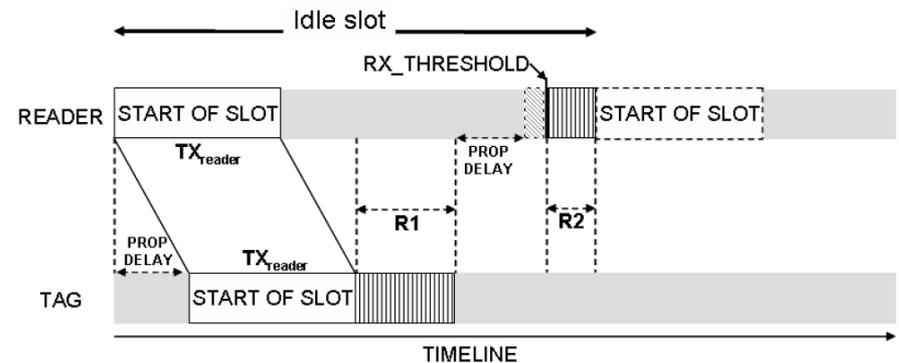
- ▶ Implementation of RFID framework within Network Simulator *ns2* (v. 2.30)
- ▶ Simulated protocols: QTI, BS, TSA, Dy_TSA
- ▶ Metrics
 - ▶ **Latency:** protocol execution time defined as the time (in seconds) for identifying all tags.
 - ▶ **System efficiency:** the fraction of rounds or time spent by the various protocols identifying tags.
 - ▶ In terms of *rounds* $SE_r = R_{id}/R_{tot}$
where R_{id} is the amount of identification rounds (which is equal to the number of tags), and R_{tot} is the total number of rounds.
 - ▶ In terms of *time* $SE_t = T_{id}/T_{tot}$
where T_{id} is the time spent in identifying tags, and T_{tot} is the total protocol execution time.

Transmission time model

- ▶ Derived from EPCglobal Specification Class 1 Gen 2



(a) Identification or collision slot.



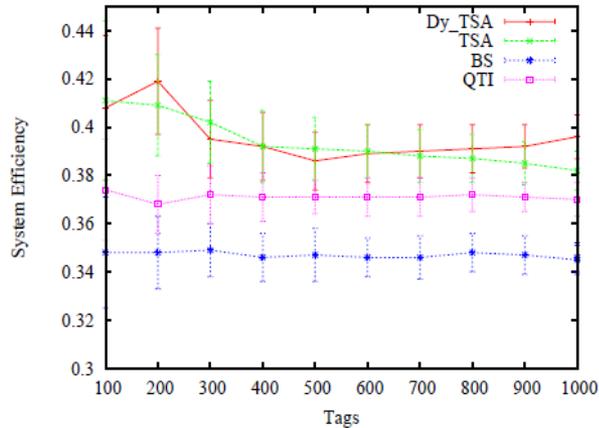
(b) Idle slot.

- ▶ $R1$: tag reaction time
- ▶ $R2$: reader reaction time
- ▶ $RX_threshold$: time at which the reader should receive the first bit of tag transmission

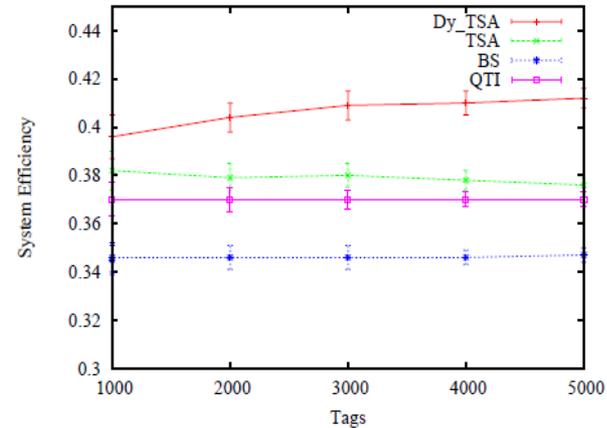
Scenarios

- ▶ Network size $n = 100, \dots, 5000$ tags
- ▶ Channel data rate: 40 Kbps
- ▶ Tag ID length: 96 bits
- ▶ Initial frame size for aloha-based protocols is set to 128 slots
- ▶ Uniform distribution of tag IDs
- ▶ Results have been obtained by averaging over 100 runs

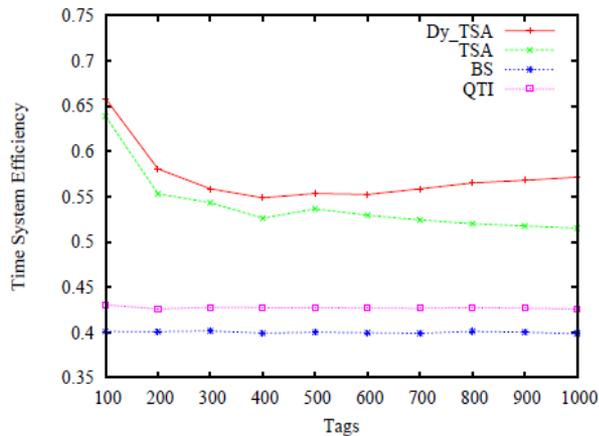
Results: Round vs. Time System Efficiency



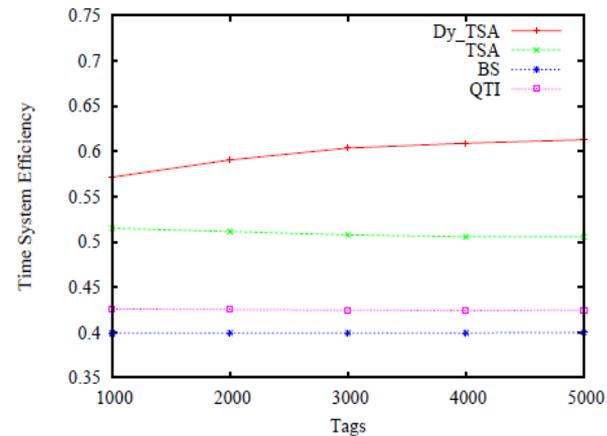
(a) System efficiency: 100-1000 nodes.



(b) System efficiency: 1000-5000 nodes.

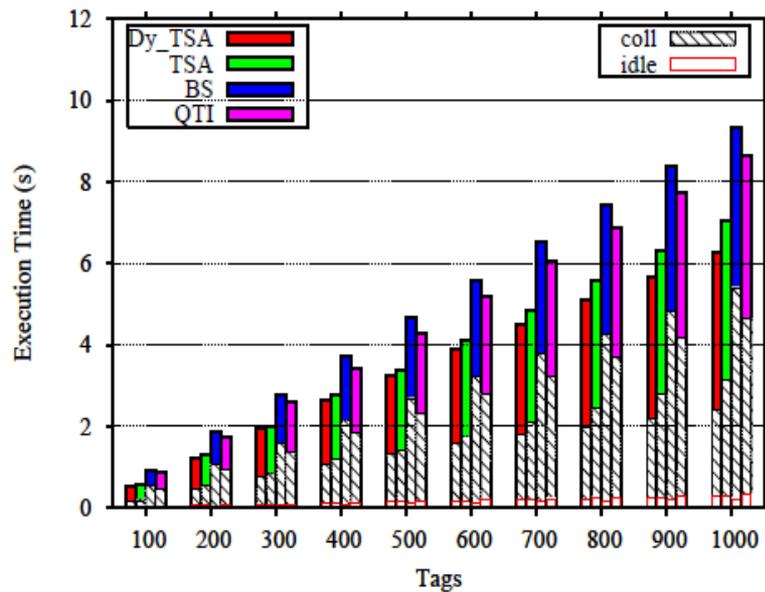


(c) Time system efficiency: 100-1000 nodes.

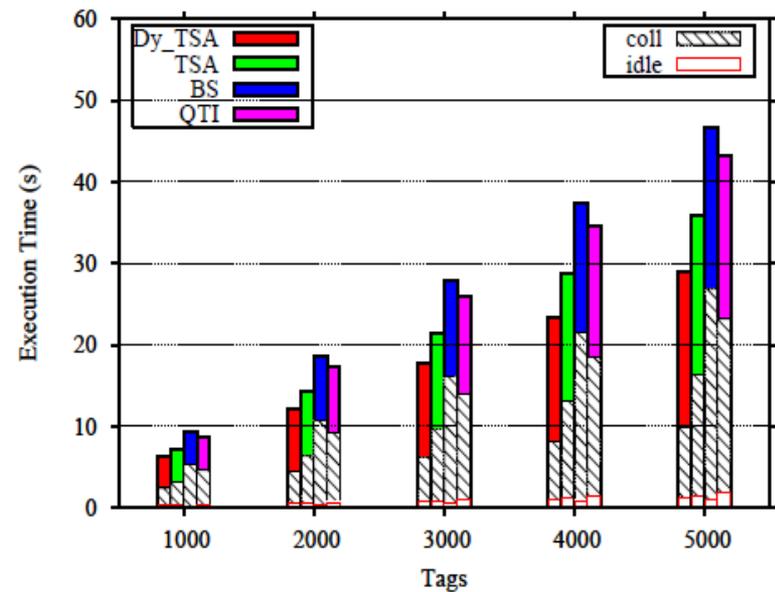


(d) Time system efficiency: 1000-5000 nodes.

Results: latency



(a) Execution time: 100-1000 nodes.



(b) Execution time: 1000-5000 nodes.

2° solution: Binary Splitting Tree Slotted Aloha (BSTSA)

- ▶ **Combination of BS and TSA**
 - ▶ BS is used to divide tags into groups whose size can be easily estimated
 - ▶ TSA is use to identify tags
- ▶ **Optimal frame sizing is adopted for each frame**
 - ▶ We derive (and use for sizing each frame) the frame size which maximizes the time system efficiency of Framed Slotted Aloha protocols

Time system efficiency

- ▶ Let R_{ident} , R_{coll} , and R_{idle} be the number of identification, collision and idle rounds during the tag identification process
- ▶ In Framed Slotted Aloha protocols in which n tags randomly select the slot to answer among N slots the probability that r tags answer in the same slot is given by the binomial distribution
- ▶ $R_{idle} = N \times (1 - 1/N)^n$
- ▶ $R_{ident} = n \times (1 - 1/N)^{n-1}$
- ▶ $R_{coll} = N - R_{idle} - R_{ident}$
- ▶ System efficiency in case of rounds of the same duration (weight) is 36%
- ▶ If idle rounds last a β fraction of identification and collision round:

$$Time_SE = \frac{R_{ident}}{\beta R_{idle} + R_{ident} + R_{coll}} = \frac{n \left(1 - \frac{1}{N}\right)^{n-1}}{(\beta - 1)N \left(1 - \frac{1}{N}\right)^n + N}$$

Optimal frame tuning

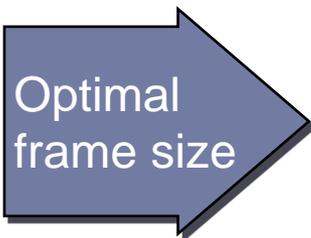
- ▶ To obtain the optimal frame size N for a given number of tags, we compute the maximum value of *Time SE* by deriving it, and posing

$$\frac{\delta Time_SE}{\delta N} = 0$$

- ▶ The maximum is achieved when

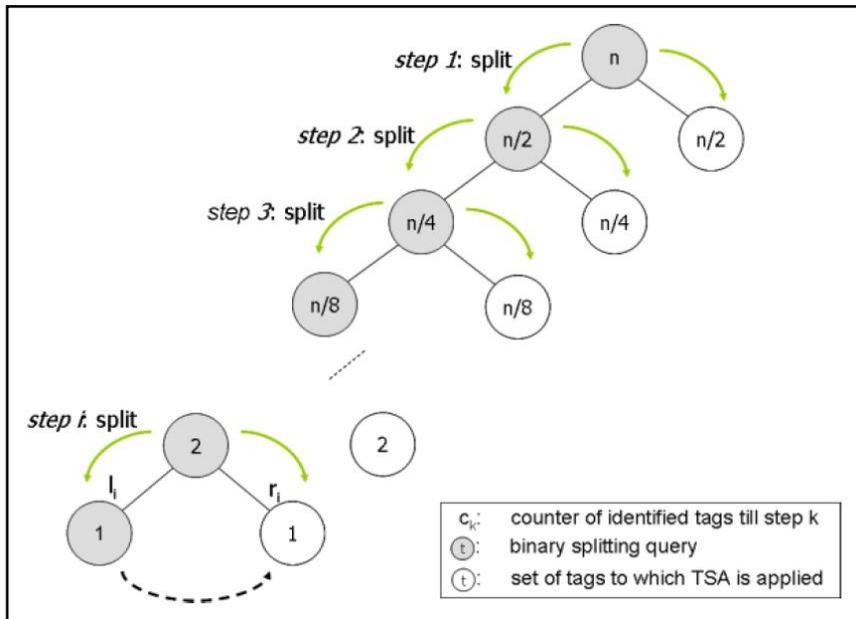
$$(\beta - 1) \left(1 - \frac{1}{N}\right)^n + 1 = \frac{n}{N}$$

- ▶ Studying the two functions we have found that the *Time_SE* is maximum (upper bounded by 80%) when

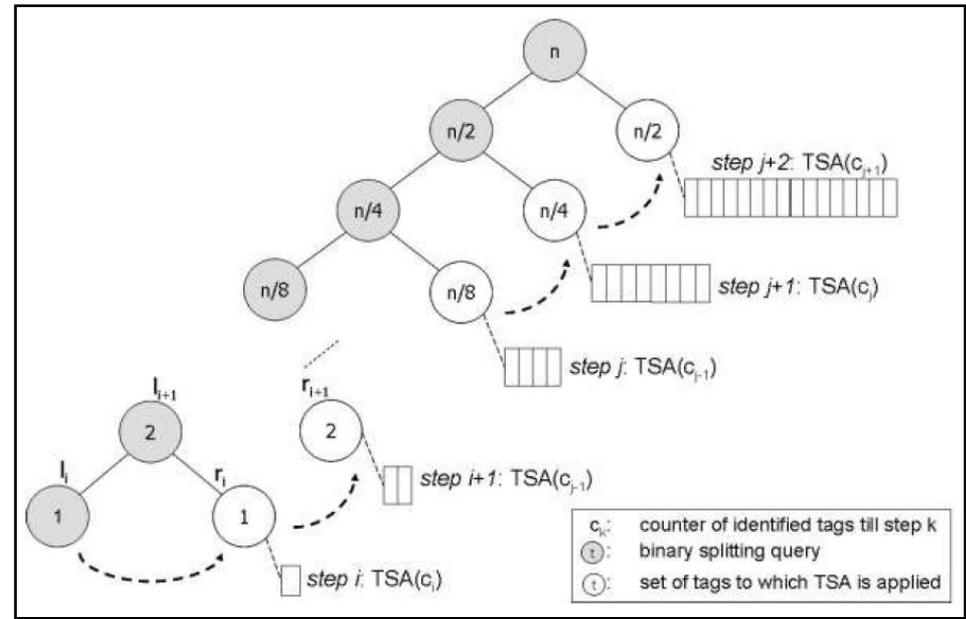


$$N = 4.406 \times n - 1$$

BSTSA protocol description

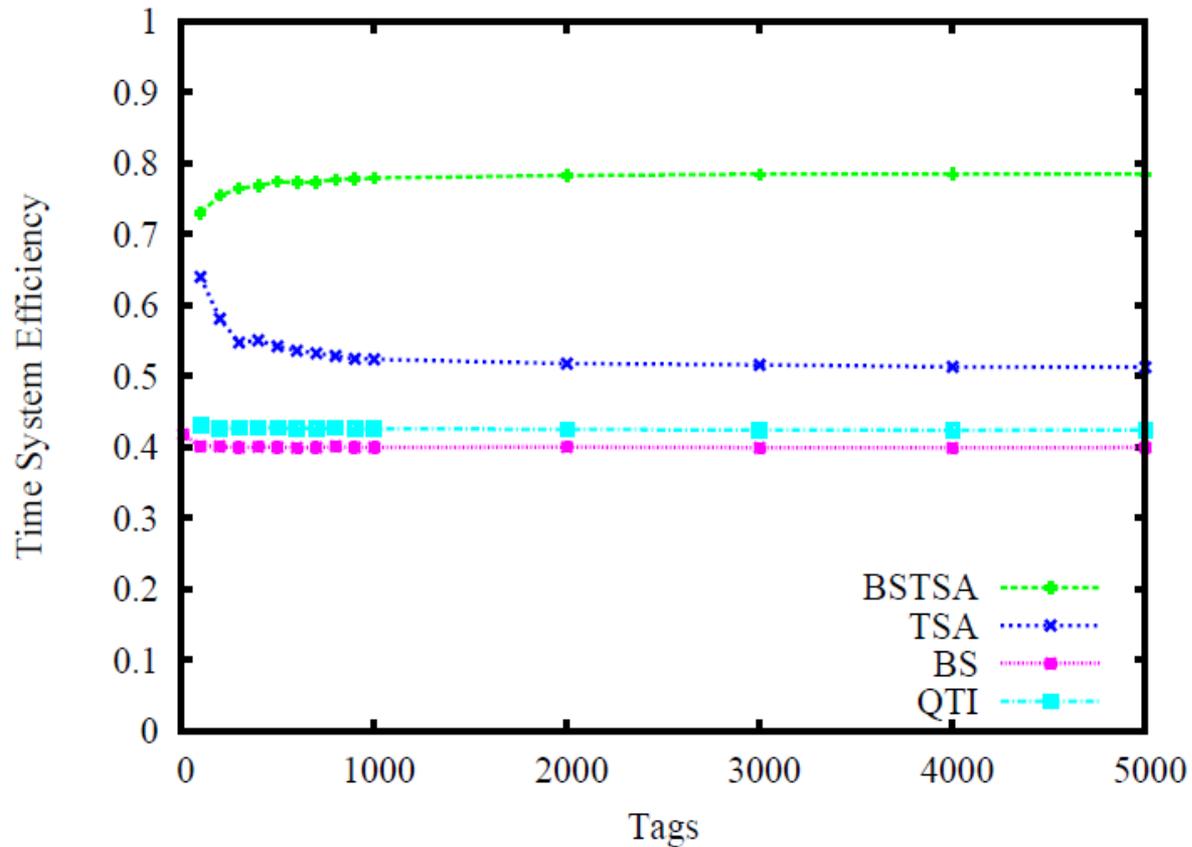


Binary Splitting phase



Tree Slotted Aloha phase

Results: Time system efficiency



Results: Latency

