Estimating tag population for intermediate frames

- 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

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

- Can we calculate this formula?
- We know number of identified tags and number of collision slots
- But we do not know the total number of tags!

Estimating tag population for intermediate frames

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

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

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

- N: size of completed frame
- <c_0,c_1,c_k> triple of observed values
- <a_0,a_1,a_k> triple of estimated values

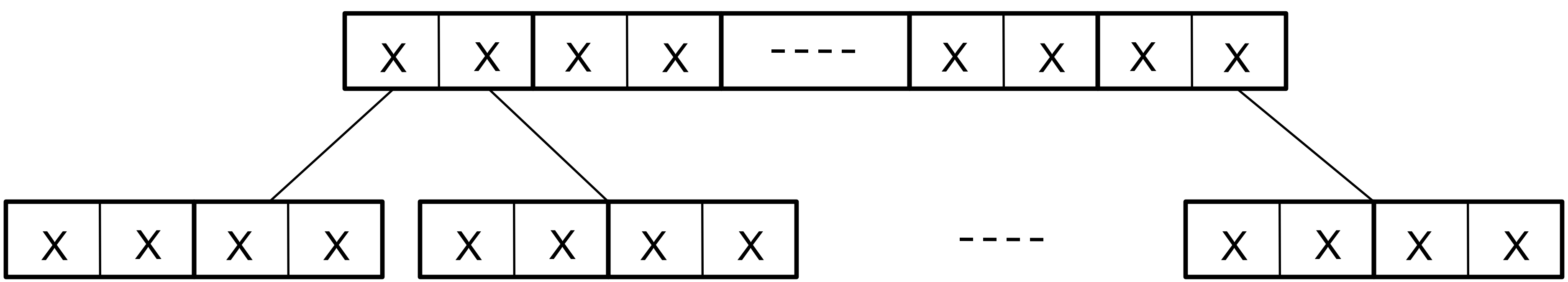
- Given *N* and a possible value of *n*, the expected number of slots with *r* tags is estimated as

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

Inaccuracy of tag estimation for large networks

- The estimator does not capture the possibly high variance of the number of tags
- The minimum distance 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!

Unbounded estimator

- Let us search for a better upper bound
- Let us not stop at $2(c_1+2c_k)$
- For N=128 and $<c_0,c_1,c_k> = <0,0,128>$, 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!

Can we find a better solution?

Starting with a proper frame size leads to better estimation also for intermediate frames

How do we estimate the initial tag population?

Estimating initial tag population

We need to estimate the **initial tag population** to properly set the size of the **first** frame

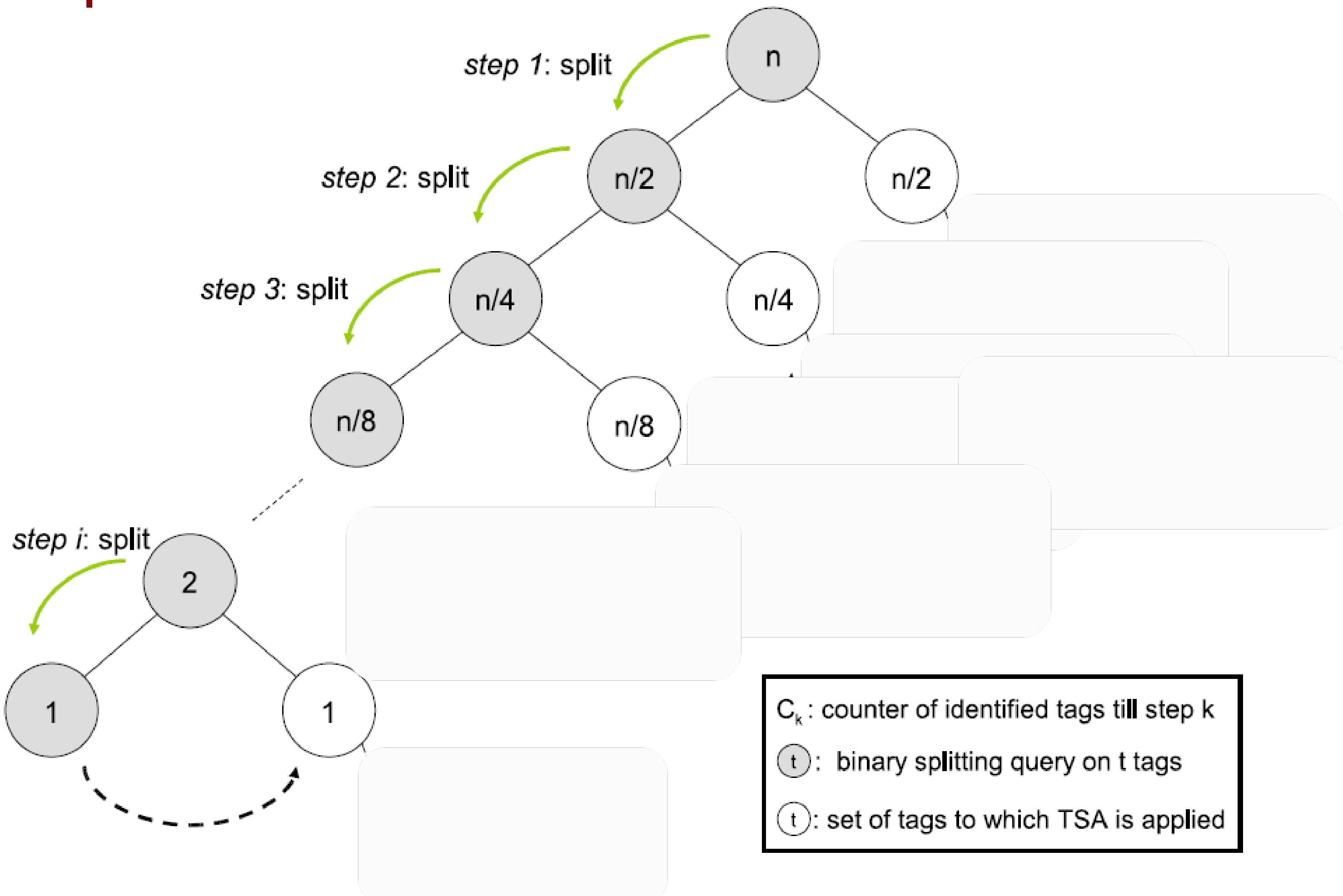
Two solutions

Dy_TSA protocol	**BSTSA protocol**

Binary Splitting Tree Slotted Aloha (BSTSA)

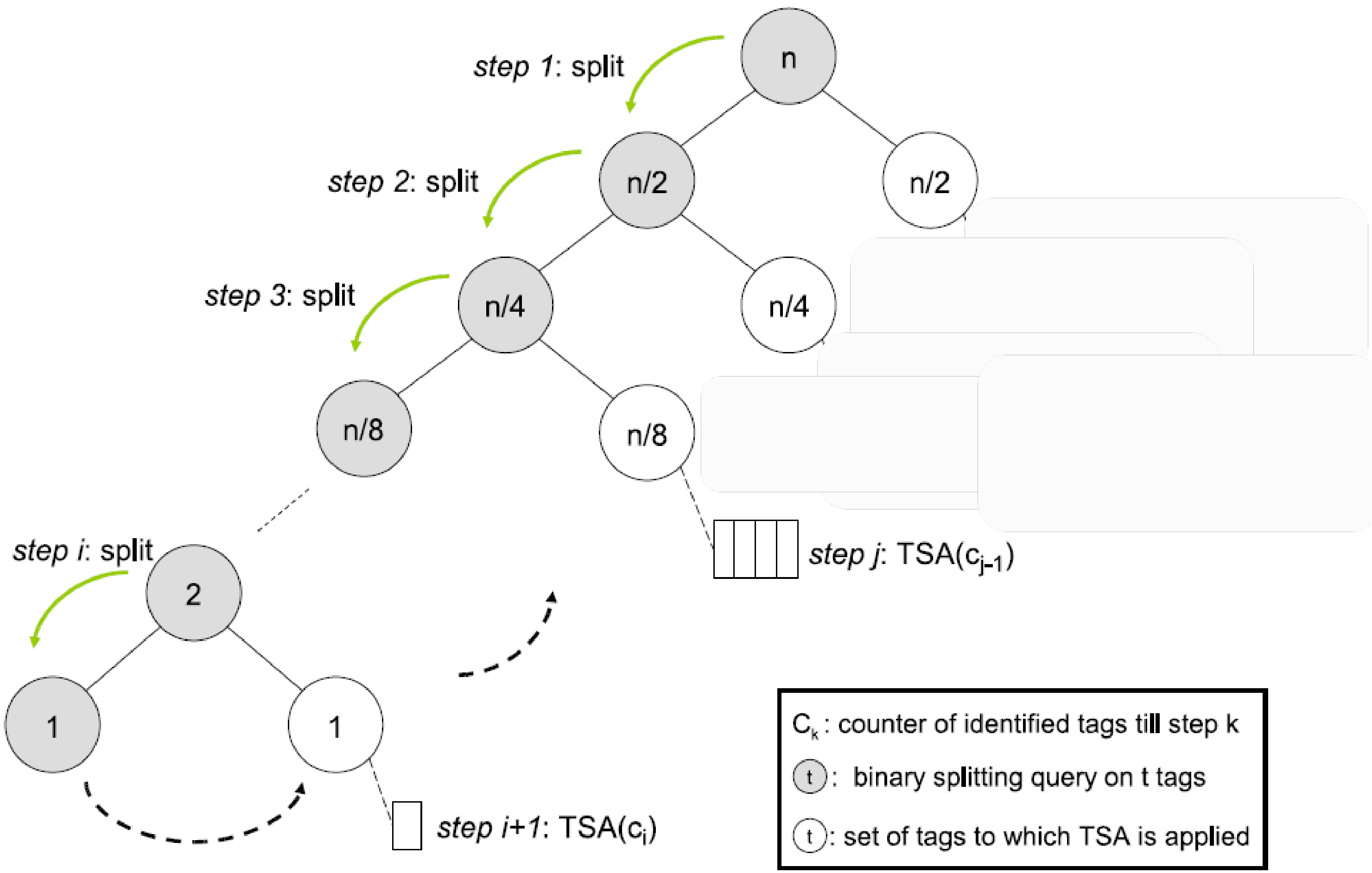
- ***Basic principle*: any large group of elements randomly split into two groups of *almost the same size***
- BS randomly splits tags
- BSTSA: Combination of BS and TSA
  - BS is used to divide tags into groups whose size can be easily estimated
  - TSA is used to identify tags

BSTSA protocol description

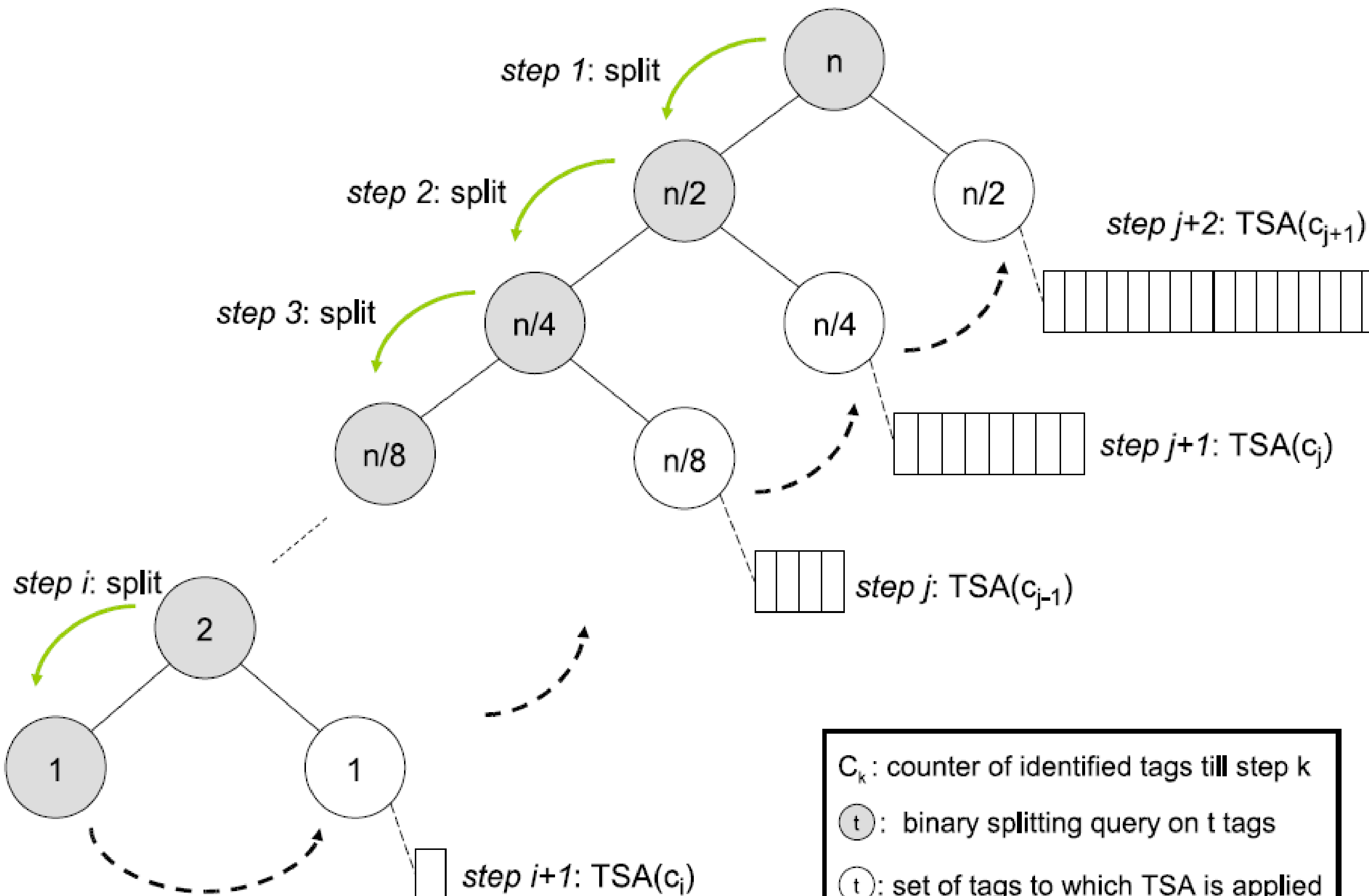


When the splitting process reaches a single-tag group (i.e., the left leaf on the tree), the protocol starts identifying the right siblings on the tree.

BSTSA protocol description



BSTSA protocol description

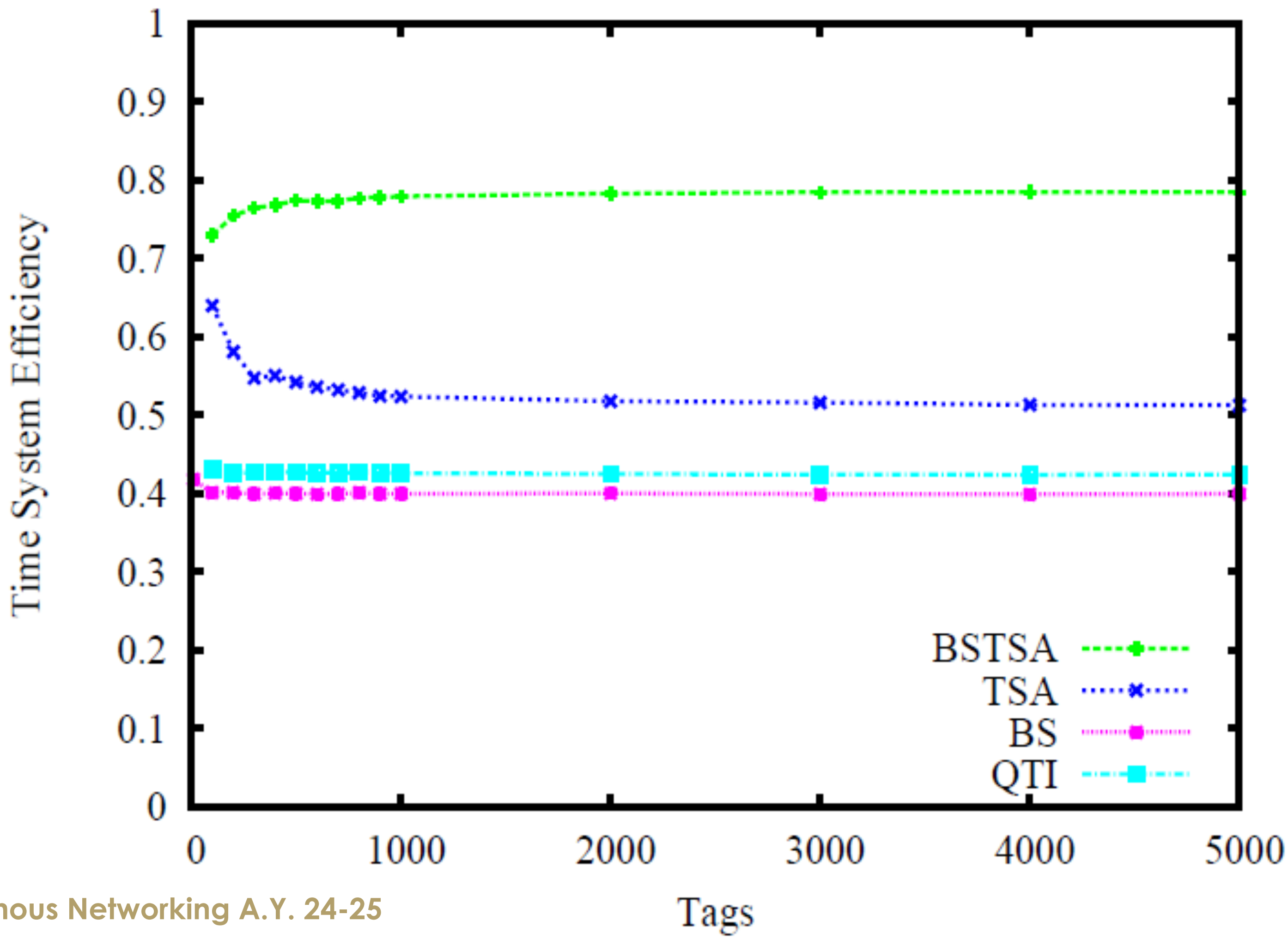


BSTSA performance

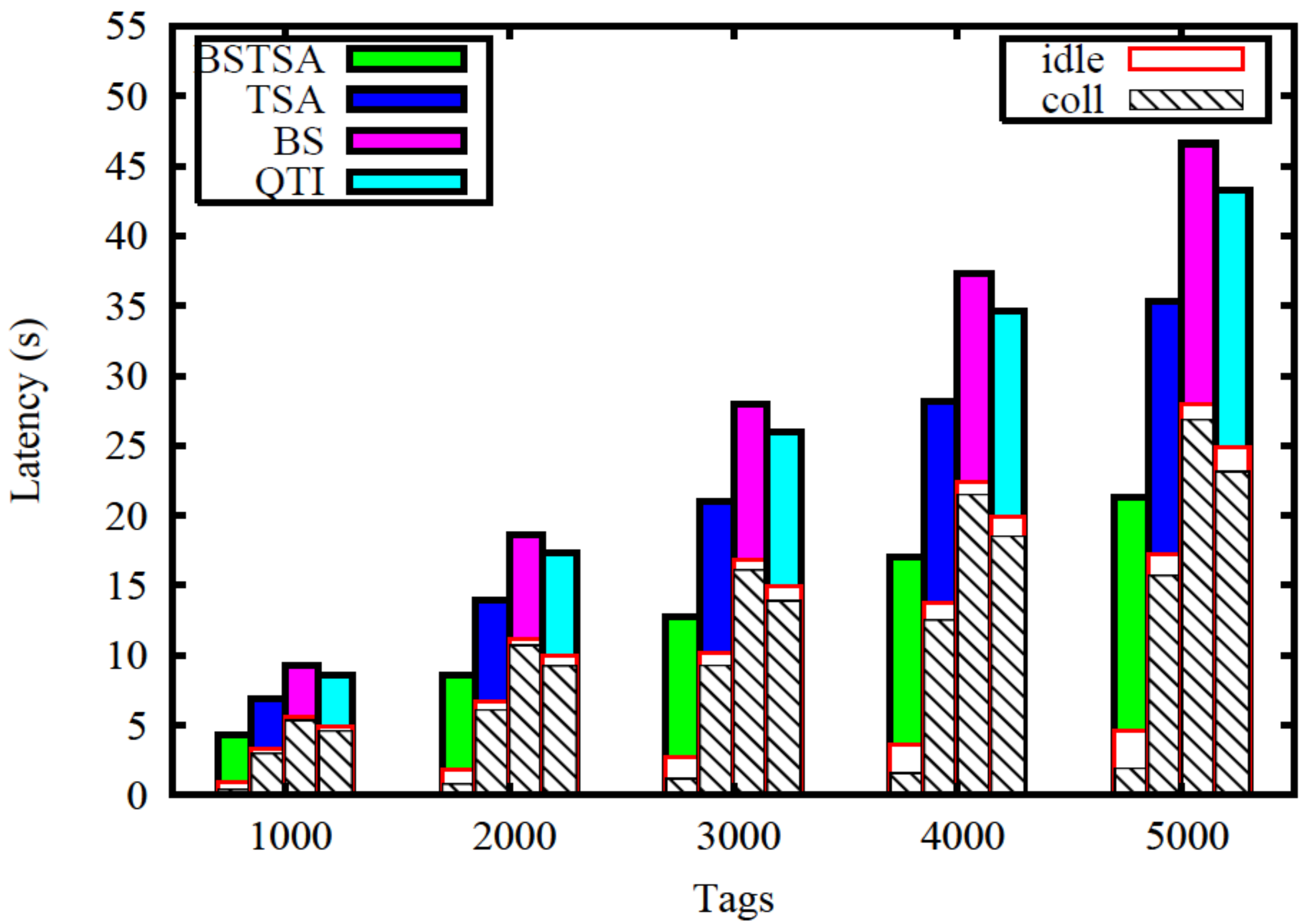
- To evaluate BSTSA performance
  - BS performace up to the last split
  - TSA performance for each group

- Optimal frame tuning is considered (overestimating frame size to allow for more idle slots than collision slots)

Results:
Time system efficiency



Results: Latency



Readings

- Paper available on IEEE digital library:
- T.F. La Porta, G. Maselli, C. Petrioli, "**Anti-collision Protocols for Single-Reader RFID Systems: Temporal Analysis and Optimization**", *IEEE Transactions on Mobile Computing*, vol.10, no.2, pp.267,279, Feb. 2011.

Questions?