POLITECNICO
MILANO 1863

Internet of things

IoT Homework
Exercise 3

Authors:
Daniel Shala - 10710181
Jurij Diego Scandola - 10709931

Academic Year:
2024 - 2025

1 RFID

A RFID system based on Dynamic Frame ALOHA is composed of $N = 4$ tags

1. **Find the overall collision resolution efficiency** η in the different cases in which the initial frame size is set to $r1 = 1, 2, 3, 4, 5, 6$:
   - Assume that after the first frame, the frame size is correctly set to the current backlog size.
   - Assume as given the duration of the arbitration period with $N = 2, 3$ tags when $r = N$: $(L_2 = 4, \quad L_3 = \frac{51}{8})$
2. After computing the values of the efficiency with the different frame sizes, **produce a plot** with values of η over $r1$ (as in the figure below) and **add it in the report**.
3. **For what values of** $r1$ **do we obtain the maximum value for** η**? Comment.**

Dynamic Frame ALOHA Efficiency Simulation

Problem Description

We analyze a RFID system composed of $N = 4$ passive tags using the **Dynamic Frame ALOHA (DFA)** protocol for tag identification.
The goal is to evaluate the overall **collision resolution efficiency** $\eta = \frac{N}{L}$, where:

- N is the number of tags,
- L is the total number of time slots needed to identify all tags.

We perform this analysis for different values of initial frame size $r_1 = 1, 2, 3, 4, 5, 6$, assuming:

- After the first frame, the frame size is dynamically updated to match the current backlog (remaining tags),
- Each tag chooses a slot in the frame uniformly at random,
- The process continues until all tags are identified.

Simulation Approach

Analytical computation of $L(N)$ for Dynamic Frame ALOHA is complex due to the recursive dependency on probabilistic outcomes (successes, collisions, empty slots).
Therefore, we use a **Monte Carlo simulation** to estimate the efficiency:

1. For each r_1 in $\{1, 2, 3, 4, 5, 6\}$:
   - Simulate 10^5 executions of the DFA protocol.
   - In each trial:
     - Start with $N = 4$ tags and initial frame size r_1,
     - In each frame, each remaining tag selects a random slot,
     - Count how many slots have exactly one tag (successful),
     - Decrease the backlog accordingly,
     - Set the next frame size equal to the current backlog,
     - Accumulate the total number of slots used.
   - Compute the average number of slots L and efficiency $\eta = \frac{4}{L}$.

Simulation Implementation

To implement the Dynamic Frame ALOHA simulation, we used a Python script based on the Monte Carlo method. The simulation works as follows:

1. We define the total number of tags $N = 4$ and vary the initial frame size $r_1 \in \{1, 2, 3, 4, 5, 6\}$.
2. Each tag randomly chooses a slot from the current frame.
3. After all tags have selected a slot, the outcome of each slot is checked:

- A slot with exactly one tag is counted as a **success**,
- A slot with more than one tag indicates a **collision**,
- An empty slot is ignored.

4. The number of remaining (unidentified) tags is updated based on the number of successes.
5. The next frame size is set equal to the new backlog (as per Dynamic Frame ALOHA rules).
6. This process is repeated until all tags are identified.
7. The number of total slots used across the entire arbitration process is recorded.

This simulation is repeated 100,000 times for each initial frame size r_1, and the average number of slots L is computed. Efficiency is then derived as:

$$\eta = \frac{N}{\mathbb{E}[L]}$$

The full simulation loop is implemented in Python using standard libraries. A simplified version of the core logic is shown below:

```
def run_dfa_simulation(N, r1):
    total_slots = 0
    remaining_tags = N
    current_r = r1
    while remaining_tags > 0:
        slots = [0] * current_r
        for _ in range(remaining_tags):
            chosen = random.randint(0, current_r - 1)
            slots[chosen] += 1
        successes = sum(1 for s in slots if s == 1)
        remaining_tags -= successes
        total_slots += current_r
        current_r = remaining_tags
    return total_slots
```

This function is called repeatedly to build statistical confidence on the expected arbitration cost.

Simulation Results

Initial Frame Size r_1	Efficiency η
1	0.4076
2	0.4167
3	0.4476
4	**0.4540**
5	0.4420
6	0.4223

Table 1: Average efficiency over 100,000 simulations for each r_1

Efficiency Plot

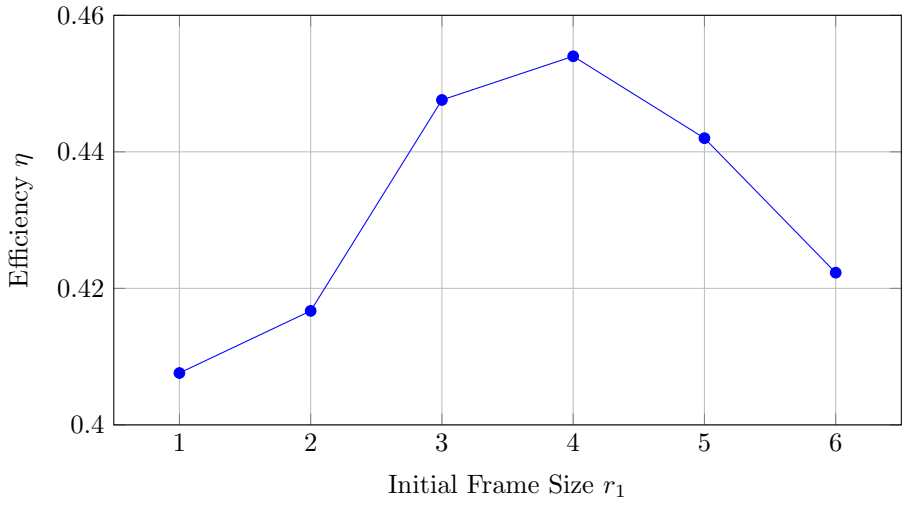

Figure 1: Efficiency vs. Initial Frame Size

Conclusion

The simulation shows that the best efficiency is obtained when the initial frame size is $r_1 = 4$, yielding an efficiency $\eta \approx 0.4540$.
This confirms that an appropriate choice of r_1 improves performance significantly. Choosing r_1 too small leads to too many collisions; too large results in many empty slots.