Adaptive Communication for Battery-Free Devices in Smart Homes

Scientific Paper by: Gaia Maselli, Mauro Piva, John A. Stankovic

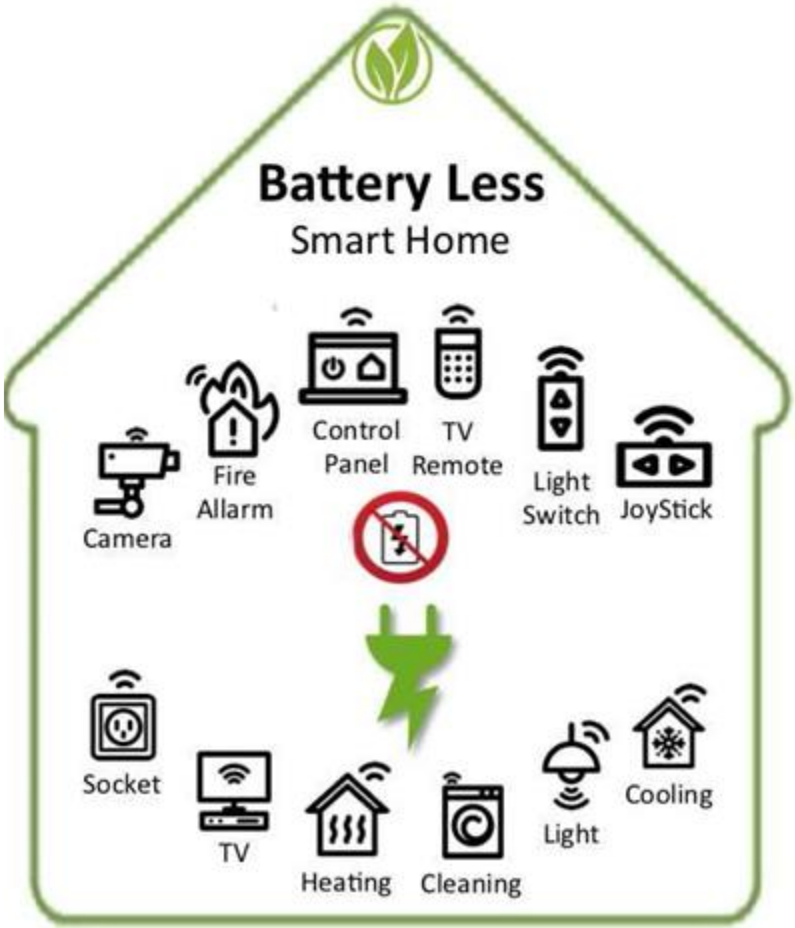
Presentation by: Matthias Bardonnet

Presentation's plan

- Context
- Addressed Problem
- Goal
- Methodology
- New MAC Protocol
- Performance Evaluation
- Conclusion

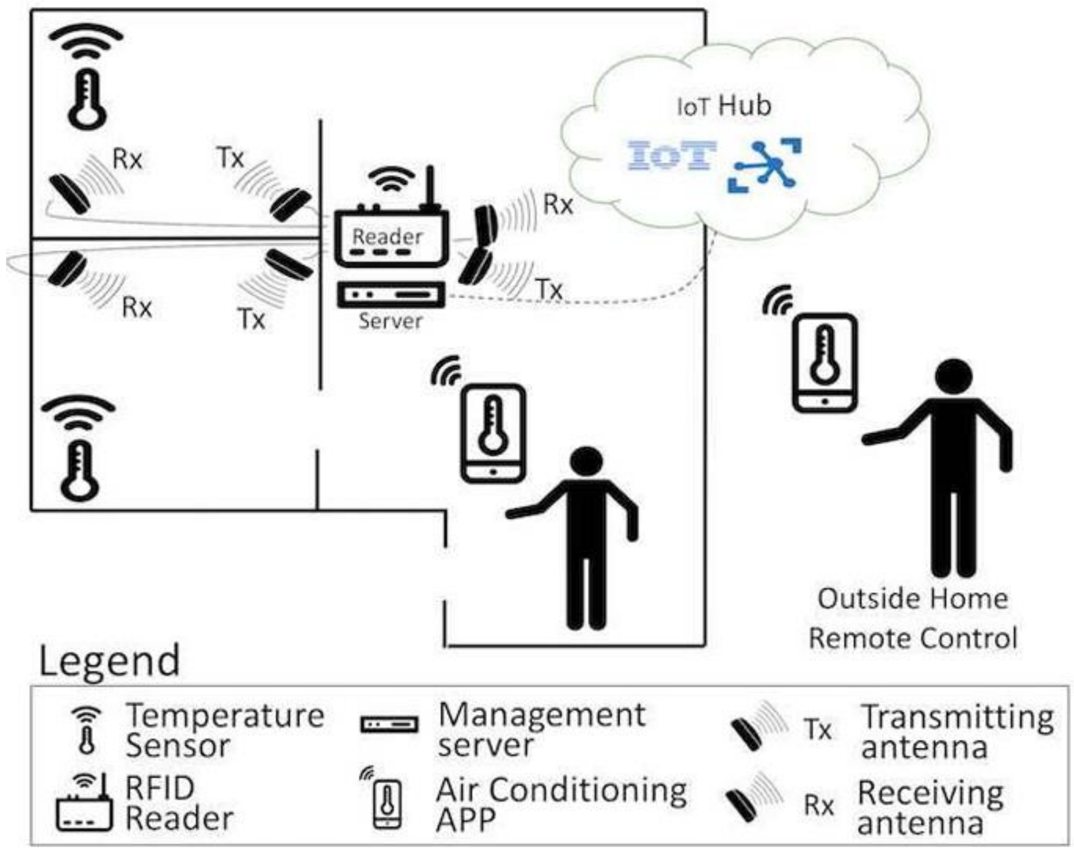
What Kind of Network?(1/2)

- **Type of Network:** Battery-free smart home network using **backscatter communication**.
- **Devices:** Includes **periodic sensors** (e.g., temperature), **event-based devices** (e.g., remotes), and **real-time devices**(e.g., joysticks and cameras).
- **Communication Protocol:** Centralized system with an **RFID reader** managing multiple devices through adaptive queries.



What Kind of Network? (2/2)

- **Sensor-augmented RFID tags:** RFID tags are equipped with on-board **sensors and/or actuators** to provide not only static information such as their ID but also dynamic and real time information about the **state of the tagged object** or the **environment where these objects reside.**



Main Characteristics

- **Power Source:** Devices powered via RF backscattering; **no batteries**.
- **Heterogeneity:** Devices have different transmission requirements, from **periodic updates** to **continuous real-time communication**.
- **Scalability:** Multiple devices simultaneously.

Main Challenges

- **Energy Efficiency:** Devices rely on energy harvested from RFID signals, limiting operational range and power.
- **Adaptability:** The protocol must dynamically handle varying device requirements without manual reconfiguration.
- **Fairness:** Ensuring all devices, especially low-demand ones, get queried appropriately to avoid starvation.

Addressed Problem

- **MAC:**
  - The problem involves selecting devices to query in a way that minimizes data loss, reduces delay, and balances device needs. We operate in a **dynamic environment** with heterogeneous, battery-free devices that have varying transmission requirements.
  - Efficient MAC is essential for managing communication between the RFID reader and devices while preventing collisions.
- **Adaptability:** The system needs to adapt to changes in device behavior (e.g., variations in data generation rates) and the addition of new devices to the network.

Goal

- **Primary Objective:** Efficiently manage communication in a smart home to ensure timely data delivery while minimizing data loss and energy consumption.
- **Secondary Objective:** Enable seamless operation without manual configuration, supporting a variety of devices.

Methodology

- Inventing a **new MAC Protocol**, called **APT-MAC**, to collect information from smart devices that improves **response time** and **data delivery.**
- **Approach:** Reinforcement Learning using a **multi-armed bandit** framework.
- **Reason for RL:** Dynamic adaptability to device behavior, making it ideal for heterogeneous and unpredictable environments.

Methodology

- **Multi-Armed Bandit Problem:**
  - The agent makes decisions on which actions to take.
  - Each action gives a reward.
  - The agent needs to maximize the cumulative reward over time.
  - Balances exploration and exploitation to maximize the cumulative reward over time.

- **Key Components:**
  - Agent: The RFID reader.
  - Actions: Querying a specific device.
  - Rewards: Positive for fresh data, negative for redundant queries.
  - Exploration: Querying less-used devices.
  - Exploitation: Prioritizing high-demand devices.

APT-MAC Protocol Workflow

- **Initialization:**
  - RFID reader identifies all devices and assigns unique IDs.
  - Devices' activity and reward parameters are initialized.
- **Adaptive Querying:**
  - Dynamically adjusts query frequency based on observed rewards.
  - Avoids redundant queries while ensuring fairness.

Key Mechanisms

- **Multi-Armed Bandit Algorithm:**
  - RFID reader (agent) queries devices (arms) to maximize cumulative rewards.
  - Balances exploration and exploitation using an **ε-greedy approach**.
  - **Probabilistic action selection**: Query device with **highest expected reward** most of the time ($p = 1 - \varepsilon$). Query another device **randomly** with a small probability ($\varepsilon = 0.1$).
- **Reward System:**
  - Reward = Bonus – Malus , with Bonus = 0.4 and Malus = 0.01 .
  - Positive (+0.39) for fresh data.
  - Negative (-0.01) for redundant queries.
- **Query Timing:**
  - **MinQD (Minimum Query Delay):** Prevents over-querying (50ms).
  - **MaxQD (Maximum Query Delay):** Ensures fairness by adjusting dynamically based on data loss (initially 2000ms).

RL Integration

- **Agent:** RFID reader.
- **Actions:** Querying specific devices.
- **Rewards:** Updated using the formula:

$$Q(a_i)(n+1) = Q(a_i)(n) + \boldsymbol{\alpha} \times \left(Reward - Q(a_i)(n)\right)$$

  - Time is slotted and each slot, also called epoch, involves a reader's action, (to query a tag). $\boldsymbol{n}$ = **number of epochs**.
  - Learning rate **($\boldsymbol{\alpha = 0.1}$).**
  - After each reward update, the **Softmax function** is applied to the Q vector, compressing its values into the range [0, 1], where all values sum to 1.

Pseudocode Summary

- **Initialize:** Assign initial rewards for each device.
- **Select Device:** Use softmax-based probabilities to choose the next query target.
- **Query Device:** Check MinQD and MaxQD constraints.
- **Update Rewards:** Adjust based on query results.

Performance Evaluation

- **Simulations** are used to evaluate the **performance** of the APT-MAC protocol.
- Protocol compared with:
  - A **TDMA** protocol that sequentially queries all tags.
  - ***Optimum*** : the optimal query strategy.

Performance Evaluation

- **Different Workload Scenarios**:
  - We can have n = 20, 30 or 40 devices. For each n devices, 4 cases.
  - Env. Sensors include temperature and presence sensors.

# of Sensors	Scenario	Joystick	Remote	Env. Sensors
20	Case 1	1	2	17
	Case 2	2	3	15
	Case 3	3	3	14
	Case 4	4	4	12
30	Case 1	1	2	27
	Case 2	2	3	25
	Case 3	3	3	24
	Case 4	4	4	22
40	Case 1	1	2	37
	Case 2	2	3	35
	Case 3	3	3	34
	Case 4	4	4	32

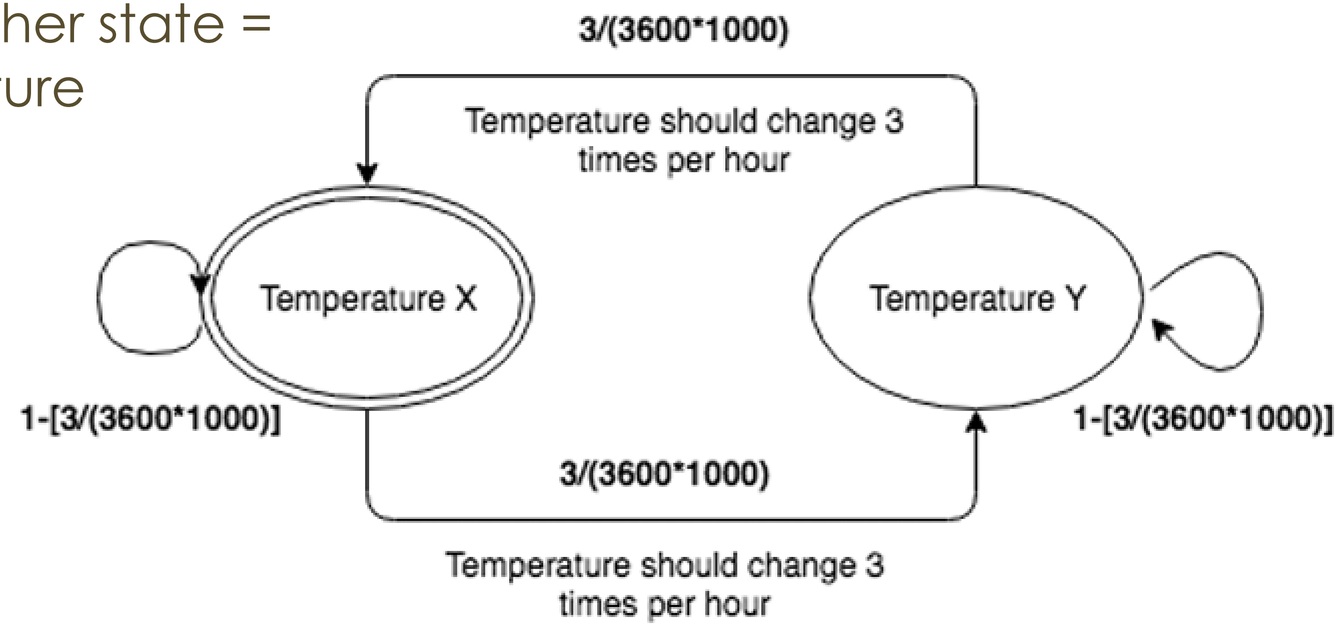
Performance Evaluation

- **Metrics** used in the performance evaluation:
  - **Packet Delay**: The **time** between the **generation of new sensor data** and its **delivery to the reader**.
  - **Data Loss**: The **amount of new data samples delivered to the reader** over the **amount of new data samples generated by the sensor**. Tags maintain a counter of changes that keeps the number of data updates performed since the last data transmission.

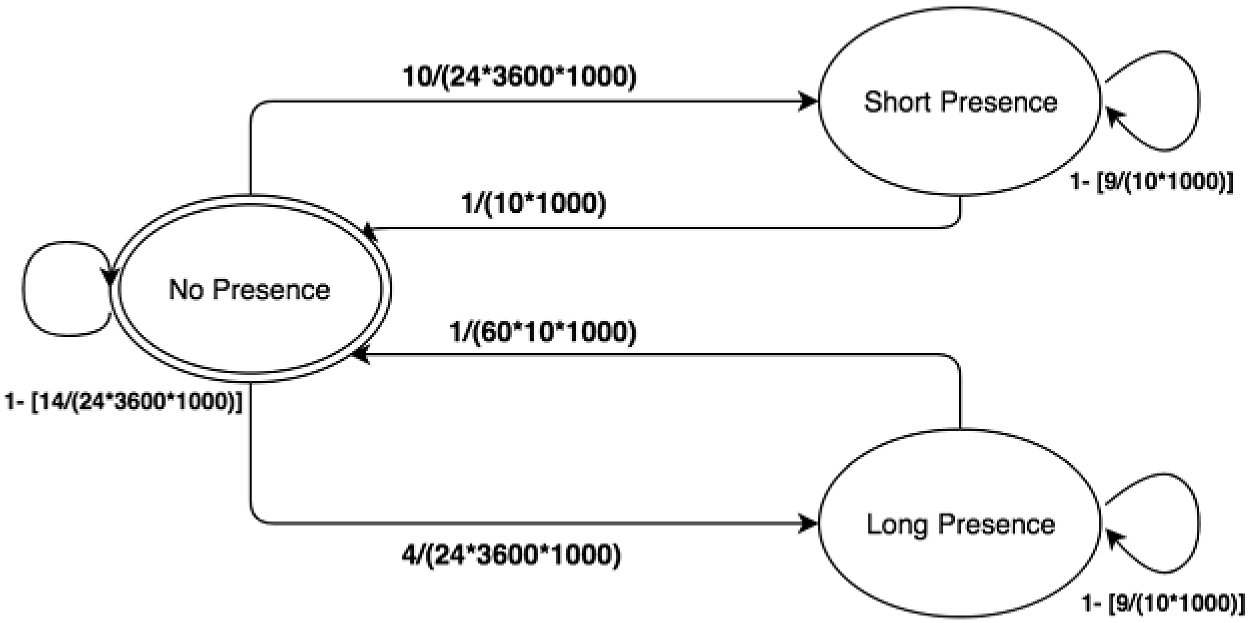
- **Device Model**
  - To build a realistic simulation environment, they modeled devices behavior through **Markov chains**. They are 4 different models.

Temperature sensor model

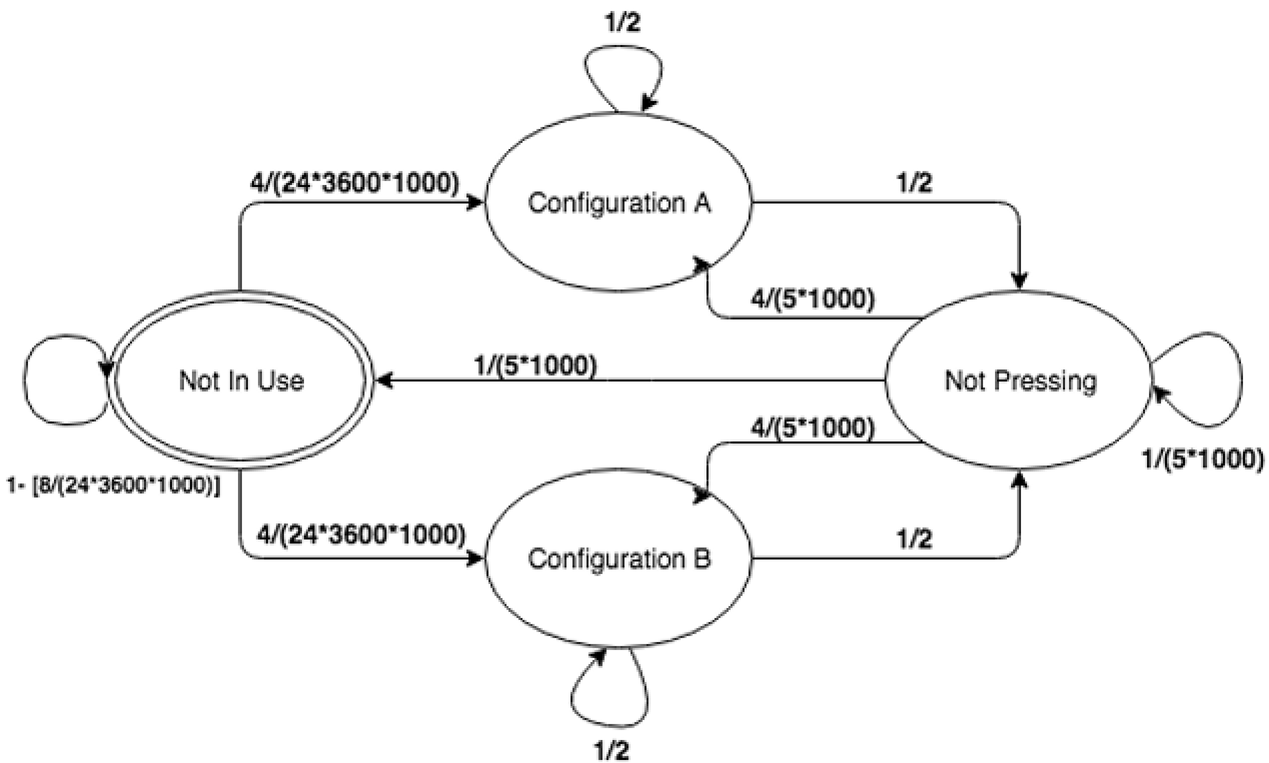
- **2 states**
- **Transition** to other state = **New** temperature



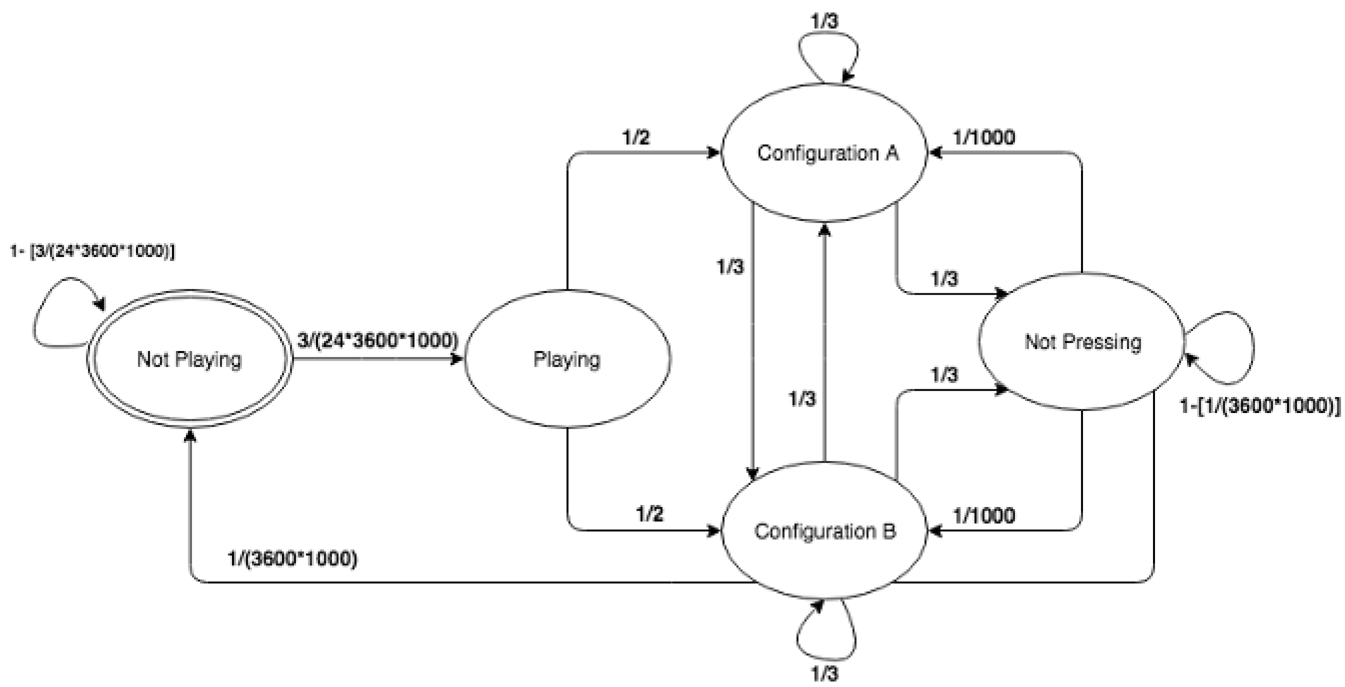
Presence sensor model



Tv remote model



Joystick model

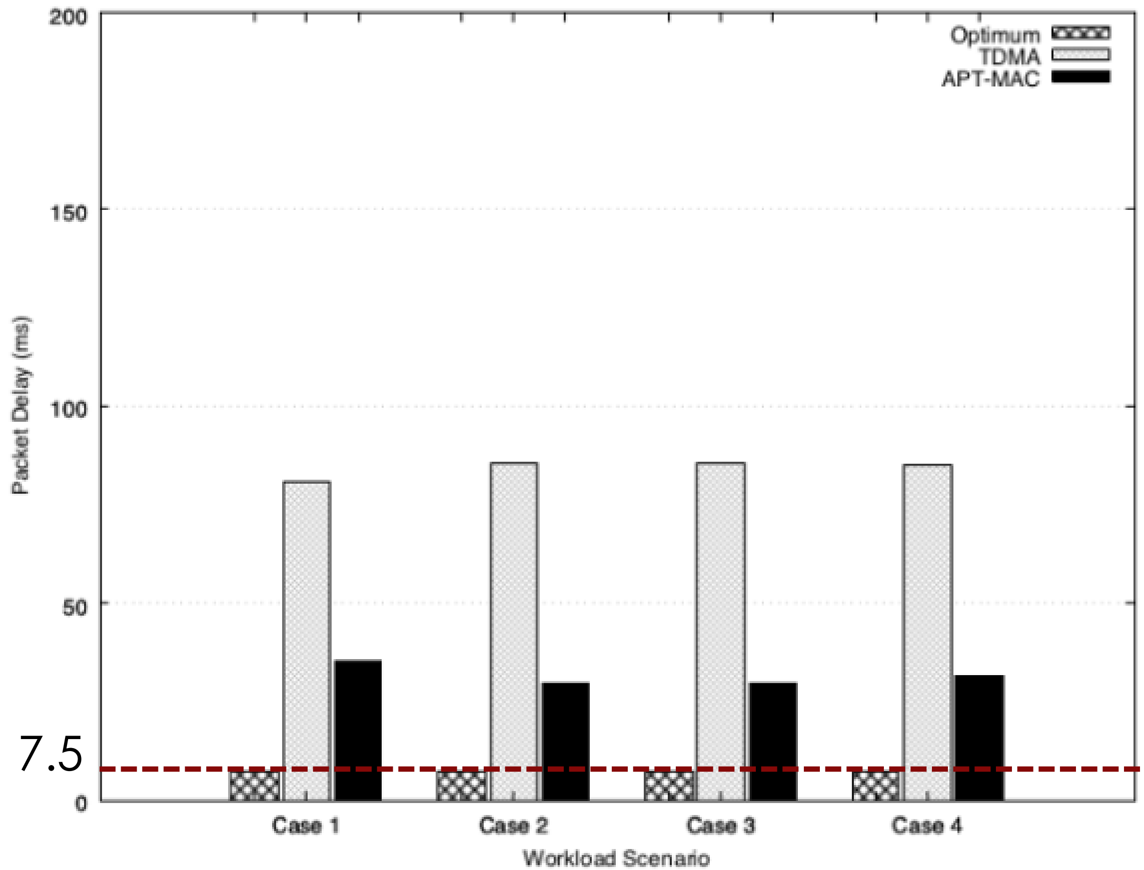


PE – Results (1/4)

- **Transient State:**
  - The model learns **transmission device requirements** and **minimize data losses** (tuning the **MaxQD** value).
  - The model stabilizes within ~12 hours of operation in simulations.
  - Early learning phase sees **higher losses** as the system adjusts to device behavior.

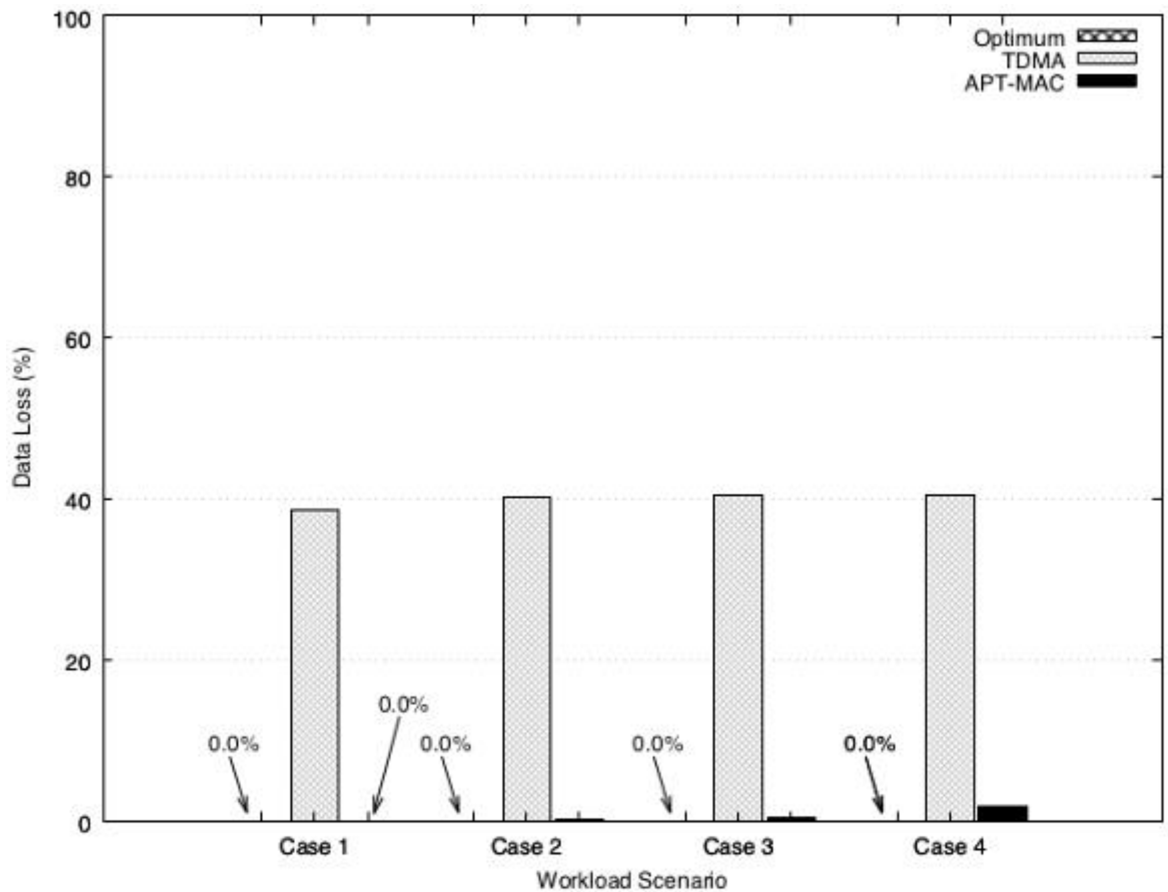
PE – Results (2/4)

- Packet Delay with 20 devices
- Optimum shows the minimum achievable delay
- APT-MAC between 29 and 35 ms.
- APT-MAC up to 2.8 times faster than TDMA



PE – Results (3/4)

- Data Loss with 20 devices
- APT-MAC does not lose more than 1.8% of new data.
- TDMA between 38.61% and 40.46%.
- Significant improvement by using APT-MAC



PE – Results (4/4)

- **Packet Delays** and **Data Loss** results with 30 and 40 devices follows the same trend as the results with 20 devices.
- APT-MAC is **always superior** to the TDMA.
- In the **worst case** (40 devices and case 4):
  - Packet Delays: APT-MAC is **4.59 times** faster than TDMA.
  - Data Loss: APT-MAC **slightly increases** (2.3%), while TDMA **significantly increases** (56.74%).

- APT-MAC maintains **low delay and minimal data loss** with increasing device count.
- TDMA struggles with scalability, leading to higher delay and loss.

Conclusion

- **Positives:**
  - **Innovative application of RL** in a challenging real-world context.
  - Demonstrates strong adaptability and scalability, with minimal data loss and delays.
  - Fairness mechanisms (MinQD/MaxQD) effectively prevent starvation.
- **Negatives:**
  - **Limited scalability** for environments with more demanding devices (e.g., **multiple cameras**).
  - Dependance on RFID technology restricts the **system's range** and **maximum data rate**.

Thank You!

- **Questions?**
- **Sources:** From the original Scientific paper.