Application of reinforcement learning to medium access control for WSNs

Autonomous Networking

Master’s Degree in Computer Science

Simone Bianco (1986936)

Academic Year 2025/2026

SAPIENZA
UNIVERSITÀ DI ROMA

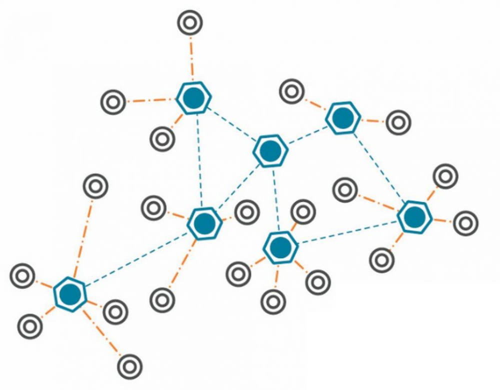
Table of Contents

Introduction

Wireless Sensor Networks

Wireless Sensor Network (WSN) are networks composed by distributed sensing devices used to monitor and record environmental conditions and events.

- Low-cost sensors
- Large number of nodes
- Multi-hop wireless communication
- Sink nodes on the edge of the network

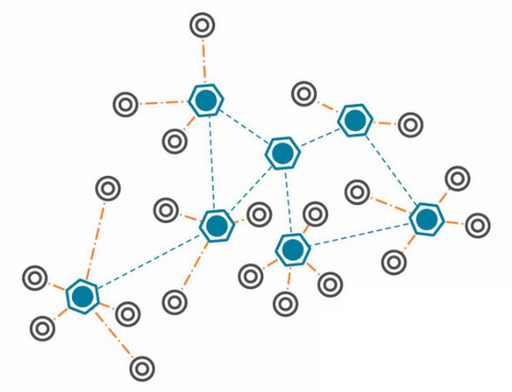


Introduction

Wireless Sensor Networks

Wireless Sensor Network (WSN) are networks composed by distributed sensing devices used to monitor and record environmental conditions and events.

- Low-cost sensors
- Large number of nodes
- Multi-hop wireless communication
- Sink nodes on the edge of the network

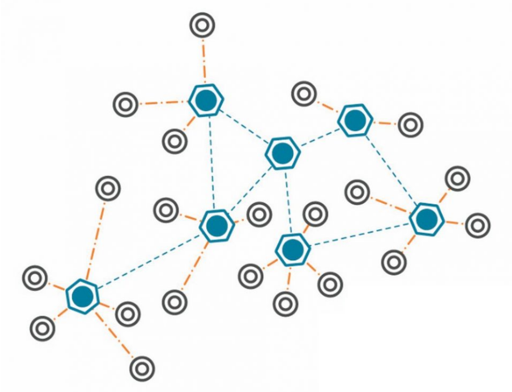


Introduction

Wireless Sensor Networks

Wireless Sensor Network (WSN) are networks composed by distributed sensing devices used to monitor and record environmental conditions and events.

- Low-cost sensors
- Large number of nodes
- Multi-hop wireless communication
- Sink nodes on the edge of the network

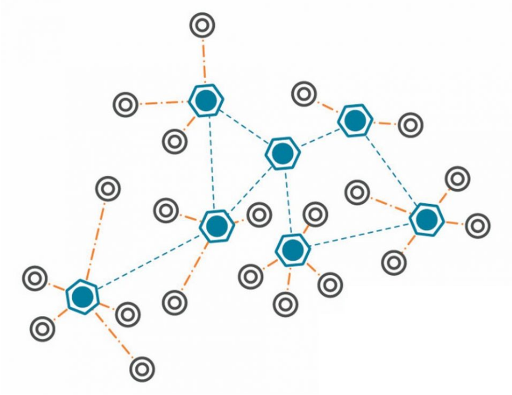


Introduction

Wireless Sensor Networks

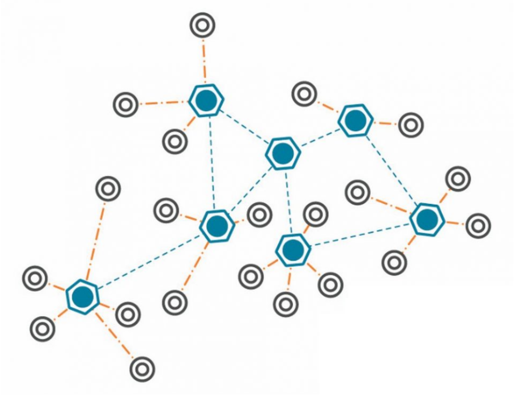
Wireless Sensor Network (WSN) are networks composed by distributed sensing devices used to monitor and record environmental conditions and events.

- Low-cost sensors
- Large number of nodes
- Multi-hop wireless communication
- Sink nodes on the edge of the network



Wireless Sensor Network (WSN) are networks composed by distributed sensing devices used to monitor and record environmental conditions and events.

- Low-cost sensors
- Large number of nodes
- Multi-hop wireless communication
- Sink nodes on the edge of the network



Benefits

Wireless Sensor Networks

WSNs are employed ***everywhere*** there is a need for monitoring a physical space or using sensors for controlling a procedure.

- Simplicity
- Large-scale coverage
- Autonomous operations
- High scalability
- Real-time data

Benefits

Wireless Sensor Networks

WSNs are employed ***everywhere*** there is a need for monitoring a physical space or using sensors for controlling a procedure.

- Simplicity
- Large-scale coverage
- Autonomous operations
- High scalability
- Real-time data

Benefits

Wireless Sensor Networks

WSNs are employed ***everywhere*** there is a need for monitoring a physical space or using sensors for controlling a procedure.

- Simplicity
- Large-scale coverage
- Autonomous operations
- High scalability
- Real-time data

Benefits

Wireless Sensor Networks

WSNs are employed ***everywhere*** there is a need for monitoring a physical space or using sensors for controlling a procedure.

- Simplicity
- Large-scale coverage
- Autonomous operations
- High scalability
- Real-time data

Benefits

Wireless Sensor Networks

WSNs are employed ***everywhere*** there is a need for monitoring a physical space or using sensors for controlling a procedure.

- Simplicity
- Large-scale coverage
- Autonomous operations
- High scalability
- Real-time data

Benefits

Wireless Sensor Networks

WSNs are employed ***everywhere*** there is a need for monitoring a physical space or using sensors for controlling a procedure.

- Simplicity
- Large-scale coverage
- Autonomous operations
- High scalability
- Real-time data

"All that glisters is not gold!"

- The Merchant of Venice, William Shakespeare

Due to their nature, WSNs suffer from **critical** issues.

- Limited computation power
- Asymmetric flow of information
- **Energy consumption**

Issues

Wireless Sensor Networks

"All that glisters is not gold!"

- The Merchant of Venice, William Shakespeare

Due to their nature, WSNs suffer from **critical** issues.

- Limited computation power
- Asymmetric flow of information
- **Energy consumption**

Issues

Wireless Sensor Networks

"All that glisters is not gold!"

- The Merchant of Venice, William Shakespeare

Due to their nature, WSNs suffer from **critical** issues.

- Limited computation power
- Asymmetric flow of information
- **Energy consumption**

Issues

Wireless Sensor Networks

"All that glisters is not gold!"

- The Merchant of Venice, William Shakespeare

Due to their nature, WSNs suffer from **critical** issues.

- Limited computation power
- Asymmetric flow of information
- Energy consumption

Issues
Wireless Sensor Networks

"All that glisters is not gold!"

- The Merchant of Venice, William Shakespeare

Due to their nature, WSNs suffer from **critical** issues.

- Limited computation power
- Asymmetric flow of information
- **Energy consumption**

Any solutions?
Wireless Sensor Networks

Many protocols have been proposed to mitigate these issues.

- S-MAC, Z-MAC, ...

Some protocols even achieve great results, but they are too **complex**.

- Quorum-MAC (Q-MAC), Low-Energy Adaptive Clustering Hierarchy (LEACH), ...

New idea: Reinforcement Learning $\rightarrow$ Introducing **Aloha-Q**!

Any solutions?

Wireless Sensor Networks

Many protocols have been proposed to mitigate these issues.

- S-MAC, Z-MAC, ...

Some protocols even achieve great results, but they are too **complex**.

- Quorum-MAC (Q-MAC), Low-Energy Adaptive Clustering Hierarchy (LEACH), ...

New idea: Reinforcement Learning → Introducing **Aloha-Q**!

Any solutions?

Many protocols have been proposed to mitigate these issues.

- S-MAC, Z-MAC, ...

Some protocols even achieve great results, but they are too **complex**.

- Quorum-MAC (Q-MAC), Low-Energy Adaptive Clustering Hierarchy (LEACH), ...

New idea: Reinforcement Learning → Introducing **Aloha-Q**!

Table of Contents

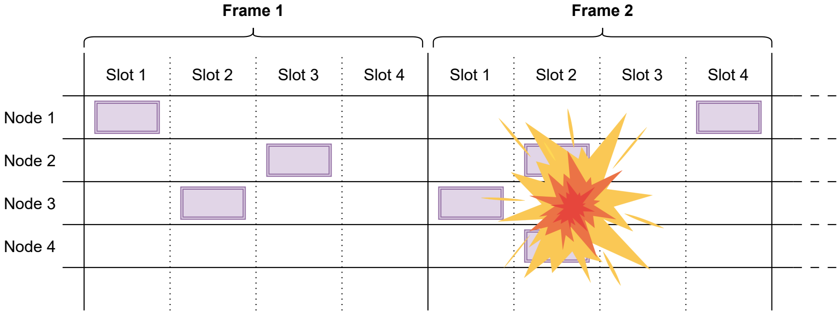
2 The Aloha-Q protocol

The main idea

The Aloha-Q protocol

Framed Slotted Aloha (FSA) with stateless **Q-Learning**.

- Learn from collisions!



The main idea

The Aloha-Q protocol

- *Frame size* M is approximately (and at least) equal to the number N of nodes
- Each node has individual **Q values** for every slot in the frame
  - Values are updated after each transmission
  - The largest value determines which slot is selected for the next transmission
- **Acknowledgement (ACK)** messages are sent when a message is received
- Nodes **wake up** only when they need to transmit and to receive ACKs
  - Synchronization times are embedded in ACK messages
- Only the sink nodes use *idle listening*

The main idea

The Aloha-Q protocol

- *Frame size* M is approximately (and at least) equal to the number N of nodes
- Each node has individual **Q values** for every slot in the frame
  - Values are updated after each transmission
  - The largest value determines which slot is selected for the next transmission
- **Acknowledgement (ACK)** messages are sent when a message is received
- Nodes **wake up** only when they need to transmit and to receive ACKs
  - Synchronization times are embedded in ACK messages
- Only the sink nodes use *idle listening*

The main idea

The Aloha-Q protocol

- *Frame size* M is approximately (and at least) equal to the number N of nodes
- Each node has individual **Q values** for every slot in the frame
  - Values are updated after each transmission
  - The largest value determines which slot is selected for the next transmission
- **Acknowledgement (ACK)** messages are sent when a message is received
- Nodes **wake up** only when they need to transmit and to receive ACKs
  - Synchronization times are embedded in ACK messages
- Only the sink nodes use *idle listening*

The main idea

The Aloha-Q protocol

- *Frame size* M is approximately (and at least) equal to the number N of nodes
- Each node has individual **Q values** for every slot in the frame
  - Values are updated after each transmission
  - The largest value determines which slot is selected for the next transmission
- **Acknowledgement (ACK)** messages are sent when a message is received
- Nodes **wake up** only when they need to transmit and to receive ACKs
  - Synchronization times are embedded in ACK messages
- Only the sink nodes use *idle listening*

The main idea

The Aloha-Q protocol

- *Frame size* M is approximately (and at least) equal to the number N of nodes
- Each node has individual **Q values** for every slot in the frame
  - Values are updated after each transmission
  - The largest value determines which slot is selected for the next transmission
- **Acknowledgement (ACK)** messages are sent when a message is received
- Nodes **wake up** only when they need to transmit and to receive ACKs
  - Synchronization times are embedded in ACK messages
- Only the sink nodes use *idle listening*

The main idea

The Aloha-Q protocol

- *Frame size* M is approximately (and at least) equal to the number N of nodes
- Each node has individual **Q values** for every slot in the frame
  - Values are updated after each transmission
  - The largest value determines which slot is selected for the next transmission
- **Acknowledgement (ACK)** messages are sent when a message is received
- Nodes **wake up** only when they need to transmit and to receive ACKs
  - Synchronization times are embedded in ACK messages
- Only the sink nodes use *idle listening*

The main idea

The Aloha-Q protocol

- *Frame size* M is approximately (and at least) equal to the number N of nodes
- Each node has individual **Q values** for every slot in the frame
  - Values are updated after each transmission
  - The largest value determines which slot is selected for the next transmission
- **Acknowledgement (ACK)** messages are sent when a message is received
- Nodes **wake up** only when they need to transmit and to receive ACKs
  - Synchronization times are embedded in ACK messages
- Only the sink nodes use *idle listening*

The main idea

The Aloha-Q protocol

- *Frame size* M is approximately (and at least) equal to the number N of nodes
- Each node has individual **Q values** for every slot in the frame
  - Values are updated after each transmission
  - The largest value determines which slot is selected for the next transmission
- **Acknowledgement (ACK)** messages are sent when a message is received
- Nodes **wake up** only when they need to transmit and to receive ACKs
  - Synchronization times are embedded in ACK messages
- Only the sink nodes use *idle listening*

Learning scheme

The Aloha-Q protocol

- Each node acts as an agent based on the **K-Armed Bandit**
  - State space: $\mathcal{S} = \{0\}$
  - Action space: $\mathcal{A} = \{0, \ldots, M-1\}$
- When the scheme converges, each node has an associated slot (recall $M \geq N$)
- Q values are described by a function $Q(x, k)$, where x is the node and k is the slot
- Each node x transmits in the slot k^* with the **highest Q value** for the node itself.

$$k^* \in \operatorname*{argmax}_{k \in \mathcal{A}} Q(x, k)$$

Learning scheme

The Aloha-Q protocol

- Each node acts as an agent based on the **K-Armed Bandit**
  - State space: $\mathcal{S} = \{0\}$
  - Action space: $\mathcal{A} = \{0, \ldots, M-1\}$
- When the scheme converges, each node has an associated slot (recall $M \geq N$)
- Q values are described by a function $Q(x, k)$, where x is the node and k is the slot
- Each node x transmits in the slot k^* with the **highest Q value** for the node itself.

$$k^* \in \operatorname*{argmax}_{k \in \mathcal{A}} Q(x, k)$$

Learning scheme

The Aloha-Q protocol

- Each node acts as an agent based on the **K-Armed Bandit**
  - State space: $\mathcal{S} = \{0\}$
  - Action space: $\mathcal{A} = \{0, \ldots, M-1\}$
- When the scheme converges, each node has an associated slot (recall $M \geq N$)
- Q values are described by a function $Q(x, k)$, where x is the node and k is the slot
- Each node x transmits in the slot k^* with the **highest Q value** for the node itself.

$$k^* \in \operatorname*{argmax}_{k \in \mathcal{A}} Q(x, k)$$

Learning scheme

The Aloha-Q protocol

- Each node acts as an agent based on the **K-Armed Bandit**
  - State space: $\mathcal{S} = \{0\}$
  - Action space: $\mathcal{A} = \{0, \ldots, M-1\}$
- When the scheme converges, each node has an associated slot (recall $M \geq N$)
- Q values are described by a function $Q(x, k)$, where x is the node and k is the slot
- Each node x transmits in the slot k^* with the **highest Q value** for the node itself.

$$k^* \in \operatorname*{argmax}_{k \in \mathcal{A}} Q(x, k)$$

Learning scheme

The Aloha-Q protocol

- Each node acts as an agent based on the **K-Armed Bandit**
  - State space: $\mathcal{S} = \{0\}$
  - Action space: $\mathcal{A} = \{0, \ldots, M-1\}$
- When the scheme converges, each node has an associated slot (recall $M \geq N$)
- Q values are described by a function $Q(x, k)$, where x is the node and k is the slot
- Each node x transmits in the slot k^* with the **highest Q value** for the node itself.

$$k^* \in \underset{k \in \mathcal{A}}{\operatorname{argmax}}\, Q(x, k)$$

Learning scheme

The Aloha-Q protocol

- Each node acts as an agent based on the **K-Armed Bandit**
  - State space: $\mathcal{S} = \{0\}$
  - Action space: $\mathcal{A} = \{0, \ldots, M-1\}$
- When the scheme converges, each node has an associated slot (recall $M \geq N$)
- Q values are described by a function $Q(x, k)$, where x is the node and k is the slot
- Each node x transmits in the slot k^* with the **highest Q value** for the node itself.

$$k^* \in \underset{k \in \mathcal{A}}{\operatorname{argmax}}\, Q(x, k)$$

K-Armed Bandit

The Aloha-Q protocol

- Each node starts with all Q values set to 0
  - Optimistic randomized start
- When a node x transmits in slot k, the Q value is updated:

$$Q_{t+1}(x,k) \leftarrow Q_t(x,k) + \alpha(r - Q_t(x,k))$$

where $\alpha \in [0,1]$ is the *learning rate* and r is the *reward*

K-Armed Bandit

The Aloha-Q protocol

- Each node starts with all Q values set to 0
  - **Optimistic randomized start**
- When a node x transmits in slot k, the Q value is updated:

$$Q_{t+1}(x,k) \leftarrow Q_t(x,k) + \alpha(r - Q_t(x,k))$$

where $\alpha \in [0,1]$ is the *learning rate* and r is the *reward*

- Each node starts with all Q values set to 0
  - **Optimistic randomized start**
- When a node x transmits in slot k, the Q value is updated:

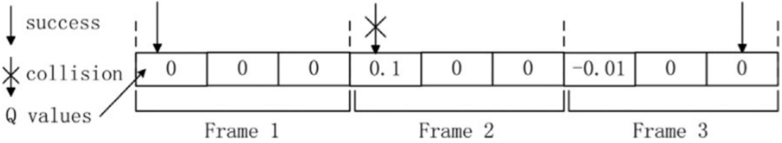
$$Q_{t+1}(x,k) \leftarrow Q_t(x,k) + \alpha(r - Q_t(x,k))$$

where $\alpha \in [0,1]$ is the *learning rate* and r is the *reward*

K-Armed Bandit

The Aloha-Q protocol

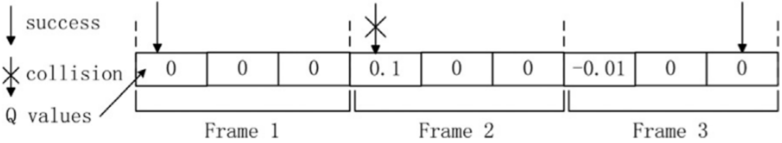
- α controls the *convergence speed*
  - Usually set to $\alpha = 0.1$ to mitigate node failures
- A reward $r = +1$ is given for **successes**, while $r = -1$ is given for **collisions**
  - Collisions are highly punished when the Q value is positive
  - Successes are highly rewarded when the Q value is negative



K-Armed Bandit

The Aloha-Q protocol

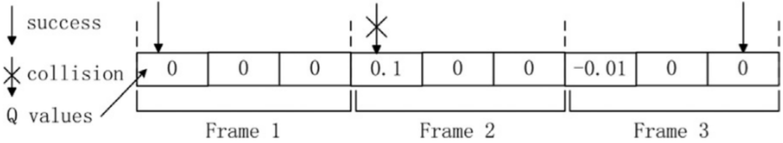
- α controls the *convergence speed*
  - Usually set to $\alpha = 0.1$ to mitigate node failures
- A reward $r = +1$ is given for **successes**, while $r = -1$ is given for **collisions**
  - Collisions are highly punished when the Q value is positive
  - Successes are highly rewarded when the Q value is negative



K-Armed Bandit

The Aloha-Q protocol

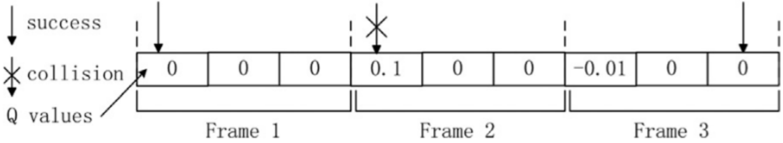
- α controls the *convergence speed*
  - Usually set to $\alpha = 0.1$ to mitigate node failures
- A reward $r = +1$ is given for **successes**, while $r = -1$ is given for **collisions**
  - Collisions are highly punished when the Q value is positive
  - Successes are highly rewarded when the Q value is negative



K-Armed Bandit

The Aloha-Q protocol

- α controls the *convergence speed*
  - Usually set to $\alpha = 0.1$ to mitigate node failures
- A reward $r = +1$ is given for **successes**, while $r = -1$ is given for **collisions**
  - Collisions are highly punished when the Q value is positive
  - Successes are highly rewarded when the Q value is negative



K-Armed Bandit

The Aloha-Q protocol

- α controls the *convergence speed*
  - Usually set to $\alpha = 0.1$ to mitigate node failures
- A reward $r = +1$ is given for **successes**, while $r = -1$ is given for **collisions**
  - Collisions are highly punished when the Q value is positive
  - Successes are highly rewarded when the Q value is negative

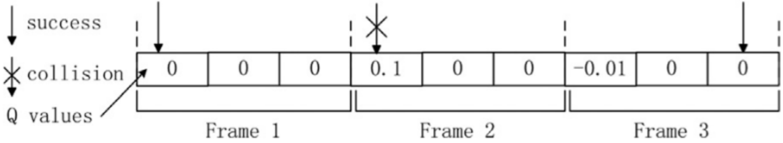


Table of Contents

Assumptions

Convergence of Aloha-Q

- Single-hop networks with N nodes and saturated traffic conditions
- Frame size equal to N
- Each node may trasmit only one packet per frame
- Learning rate set to $\alpha = 1$

$$Q_{t+1}(x,k) \leftarrow Q_t(x,k) + 1 \cdot (r - Q_t(x,k)) = r$$

Assumptions

Convergence of Aloha-Q

- Single-hop networks with N nodes and saturated traffic conditions
- Frame size equal to N
- Each node may trasmit only one packet per frame
- Learning rate set to $\alpha = 1$

$$Q_{t+1}(x, k) \leftarrow Q_t(x, k) + 1 \cdot (r - Q_t(x, k)) = r$$

Assumptions

Convergence of Aloha-Q

- Single-hop networks with N nodes and saturated traffic conditions
- Frame size equal to N
- Each node may trasmit only one packet per frame
- Learning rate set to $\alpha = 1$

$$Q_{t+1}(x, k) \leftarrow Q_t(x, k) + 1 \cdot (r - Q_t(x, k)) = r$$

Assumptions

Convergence of Aloha-Q

- Single-hop networks with N nodes and saturated traffic conditions
- Frame size equal to N
- Each node may trasmit only one packet per frame
- Learning rate set to $\alpha = 1$

$$Q_{t+1}(x,k) \leftarrow Q_t(x,k) + 1 \cdot (r - Q_t(x,k)) = r$$

Definitions

Convergence of Aloha-Q

- **Steady node**: a node with Q values set to -1 for all slots except for one set to $+1$
  - Otherwise, *hopping node*
- **Occupied slot**: a slot with Q values set to -1 for all nodes except for one set to $+1$
  - Otherwise, *unoccupied node*

Definitions

Convergence of Aloha-Q

- **Steady node**: a node with Q values set to -1 for all slots except for one set to $+1$
  - Otherwise, *hopping node*
- **Occupied slot**: a slot with Q values set to -1 for all nodes except for one set to $+1$
  - Otherwise, *unoccupied node*

Definitions

Convergence of Aloha-Q

- **Steady node**: a node with Q values set to -1 for all slots except for one set to $+1$
  - Otherwise, *hopping node*
- **Occupied slot**: a slot with Q values set to -1 for all nodes except for one set to $+1$
  - Otherwise, *unoccupied node*

Definitions

Convergence of Aloha-Q

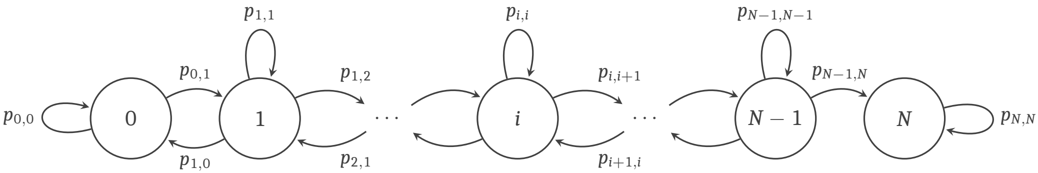
- **Steady node**: a node with Q values set to -1 for all slots except for one set to $+1$
  - Otherwise, *hopping node*
- **Occupied slot**: a slot with Q values set to -1 for all nodes except for one set to $+1$
  - Otherwise, *unoccupied node*

Markov model

Convergence of Aloha-Q

We consider a **Markov chain** with state space $\mathcal{I} = \{0, \ldots, N\}$

- Each state $i \in \mathcal{I}$ represents the number of steady nodes/occupied slots
- For each state $i \in \mathcal{I} - \{0, N\}$ we define three transitions: $p_{i,i}$, $p_{i,i-1}$ and $p_{i,i+1}$.
- State 0 has only two transitions: $p_{0,0}$ and $p_{0,1}$
- State N has only one transition: $p_{N,N}$

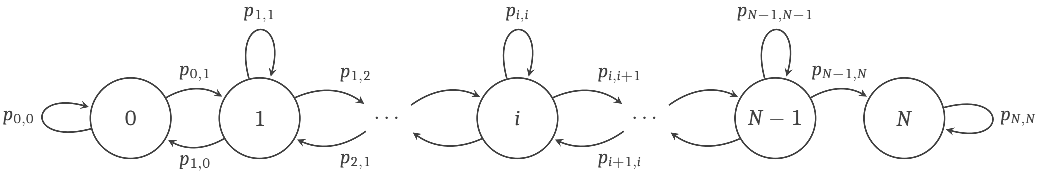


Markov model

Convergence of Aloha-Q

We consider a **Markov chain** with state space $\mathcal{I} = \{0, \ldots, N\}$

- Each state $i \in \mathcal{I}$ represents the number of steady nodes/occupied slots
- For each state $i \in \mathcal{I} - \{0, N\}$ we define three transitions: $p_{i,i}$, $p_{i,i-1}$ and $p_{i,i+1}$.
- State 0 has only two transitions: $p_{0,0}$ and $p_{0,1}$
- State N has only one transition: $p_{N,N}$

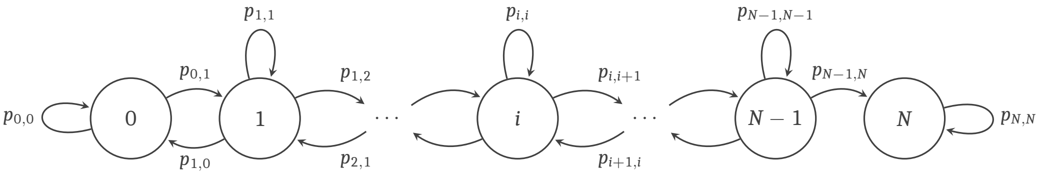


Markov model

Convergence of Aloha-Q

We consider a **Markov chain** with state space $\mathcal{I} = \{0, \ldots, N\}$

- Each state $i \in \mathcal{I}$ represents the number of steady nodes/occupied slots
- For each state $i \in \mathcal{I} - \{0, N\}$ we define three transitions: $p_{i,i}$, $p_{i,i-1}$ and $p_{i,i+1}$.
- State 0 has only two transitions: $p_{0,0}$ and $p_{0,1}$
- State N has only one transition: $p_{N,N}$

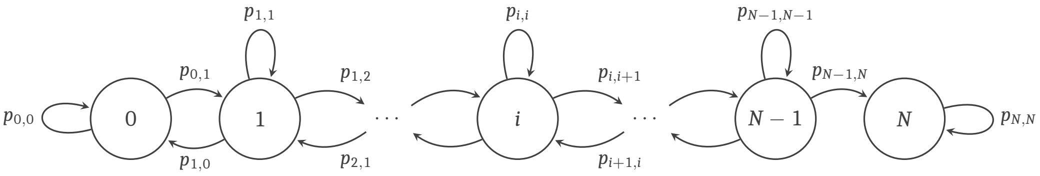


Markov model

Convergence of Aloha-Q

We consider a **Markov chain** with state space $\mathcal{I} = \{0, \ldots, N\}$

- Each state $i \in \mathcal{I}$ represents the number of steady nodes/occupied slots
- For each state $i \in \mathcal{I} - \{0, N\}$ we define three transitions: $p_{i,i}$, $p_{i,i-1}$ and $p_{i,i+1}$.
- State 0 has only two transitions: $p_{0,0}$ and $p_{0,1}$
- State N has only one transition: $p_{N,N}$



Markov model

Convergence of Aloha-Q

We move **up one state** when the current slot is unoccupied and only one hopping node transmits in it

$$p_{i,i+1} = \underbrace{\left(\frac{N-i}{N}\right)}_{\text{Pr. of unoccupied slot}} \cdot \underbrace{\left(\frac{N-i}{N}\right)\left(\frac{N-1}{N}\right)^{N-i-1}}_{\text{Pr. of } = 1 \text{ hopping node when unoccupied}}$$

Markov model

Convergence of Aloha-Q

We move **down one state** when the current slot is occupied and one or more hopping nodes transmits in it

$$p_{i,i-1} = \underbrace{\left(\frac{i}{N}\right)}_{\text{Pr. of occupied slot}} \cdot \underbrace{\left(1 - \left(\frac{N-1}{N}\right)^{N-i}\right)}_{\text{Pr. of } \geq 1 \text{ hopping nodes when occupied}}$$

Markov model

Convergence of Aloha-Q

We stay in the **same state** when:

- The current slot is occupied and no hopping nodes select the current slot.
- The current slot is unoccupied and two or more hopping nodes transmit packets in it.
- The current slot is unoccupied and there are no transmissions in it.

$$p_{i,i} = \frac{i}{N}\left(\frac{N-1}{N}\right)^{N-i} + \frac{N-i}{N}\left(1 - \frac{N-i}{N}\left(\frac{N-1}{N}\right)^{N-i-1}\right)$$

Limiting distribution

Convergence of Aloha-Q

Convergence is achieved when $\lim_{n \to +\infty} P^n_{i,N} = 1$ for all $i \in \{0, \ldots, N\}$

- P is the **Probability Transition Matrix (PTM)** of the Markov chain
- $P^n_{i,j} = \sum_{m=0}^{N} P^{n-1}_{i,m} P_{m,j}$

It can be proven that the above **limiting distribution** converges

Expected convergence time

Convergence of Aloha-Q

Expected number of visits to all states, except state N, across all n steps as $n \to +\infty$, starting from state 0.

$$\mathbb{E}[T] = \sum_{n=1}^{+\infty} \sum_{j=0}^{N-1} P_{0,j}^{n}$$

Requires intensive computations due to no closed form

Table of Contents

Simulations

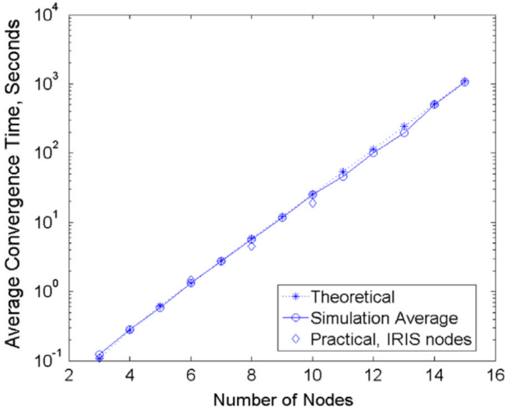
Performance of Aloha-Q

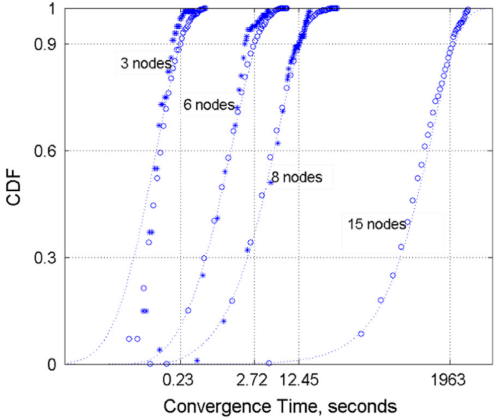
$100 - 200$ simulations, $50 - 100$ practical trials, various network sizes

Parameters	Values
Channel bit rate	250 kbits/s
Data packet length (simulation)	1044 bits
Data packet length (practical)	935 bits
ACK packet length (simulation)	20 bits
ACK packet length (practical)	144 bits
Slot length	1100 bits

Convergence time

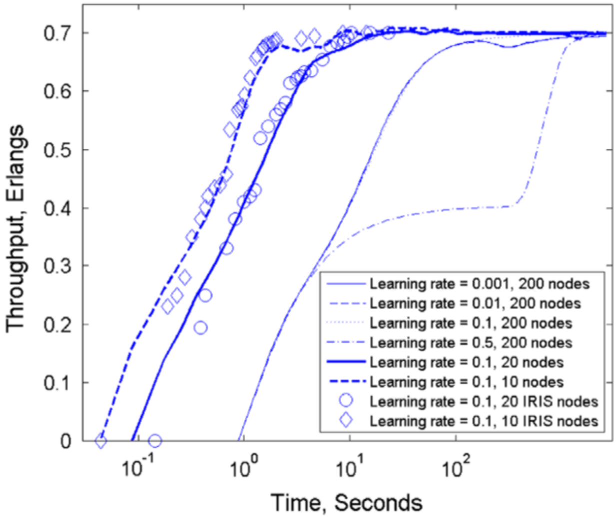
Performance of Aloha-Q





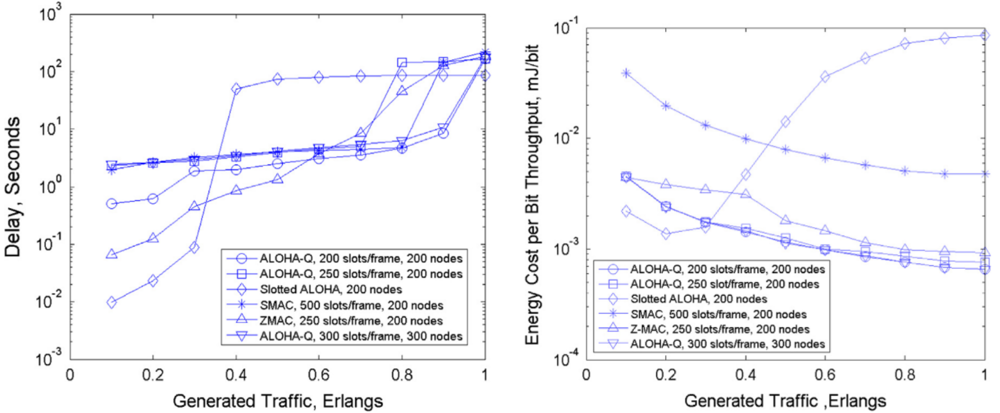
Throughput

Performance of Aloha-Q



Delay and Energy consumption

Performance of Aloha-Q



Conclusions

Performance of Aloha-Q

Summary of **Aloha-Q** performance analysis:

- Simulations and practical trials are close to theoretical limits
- Convergence is reached in short time (relative to lifespan of the WSN)
- Performance is better than S-MAC and very close to Z-MAC (when converged)
- Way less over overhead than already existing protocols

Thank you for listening!
Any questions?