# Q-Learning and NOMA Techniques for IoT-Satellite Terrestrial Relay Networks

*Thesis Submitted for Partial Fulfillment*

*of the Requirements for the Award of the Degree of*

### Bachelor of Technology in

**Electronics and Communication Engineering by**

### Anurag Baundwal

#### Roll Number: 1901038

**Under the Supervision of**

### Dr. Sudip Biswas and

### Dr. Shilpa Rao



**Department of Electronics and Communication Engineering Indian Institute of Information Technology Guwahati**

18*th November,* 2022

# Contents

1. Abstract 5
2. Reinforcement Learning 6
3. NOMA 9
4. Work Done 11
5. Conclusion and Future Work 13

[Bibliography](#_TOC_250003) 14

**Certificate of Approval**

This is to certify that the thesis entitled **“Title”** submitted by Anurag Baundwal to Indian Institute of Information Technology, Guwahati, is a record of bona fide research work under my supervision and I consider it worthy of consideration for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering at Indian Institute of Information Technology Guwahati.

[Dr. Sudip Biswas]

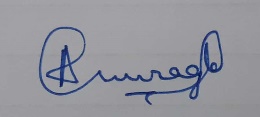
Assistant Professor

Department of Electronics and Communication Indian Institute of Information Technology, Guwahati Guwahati 781001, Assam, India.

# Declaration

I declare that

* 1. The work contained in the thesis is original and has been done by myself under the general supervision of my supervisor.
  2. The work has not been submitted to any other Institute for any degree or diploma.
  3. I have followed the guidelines provided by the Institute in writing the thesis.
  4. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
  5. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
  6. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.



Anurag Baundwal

# Abstract

*The term Q-Learning refers to a certain type of model-free Reinforcement Learning algorithms, having applications in a wide variety of fields. NOMA (Non-Orthogonal Multiple Access) is a powerful Multiple Access technique for resource allocation. Through this paper I have conducted a review of the literature in both these fields and discussed their application to IoT-Satellite Terrestrial Relay Networks (STRNs). Doing this helps provide better throughput compared to other techniques, as well reduce the resource requirements (channels and relays).*

# Chapter 1: Reinforcement Learning

# Introduction

# 10 Companies Using Machine Learning in Cool Ways

Reinforcement learning is a branch of Machine Learning. It deals with agents in an environment, which is typically modelled as a Markov Decision Process (MDP) and guides the agents on what actions to take maximize their rewards.

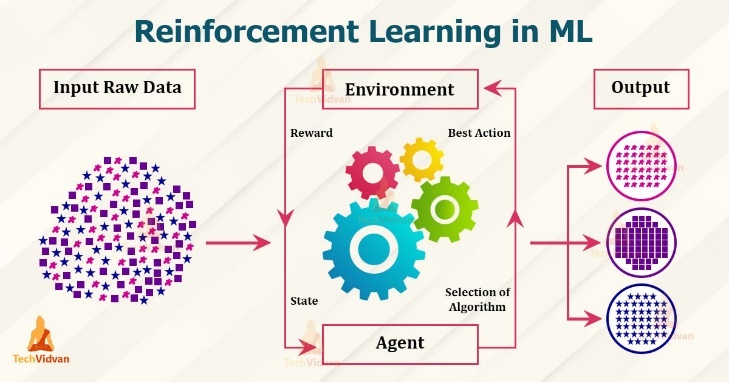


Figure 1: Reinforcement Learning

Applications: Reinforcement Learning has applications in many disciplines, such as game theory, control theory, operations research, information theory, simulation, multi-agent systems, statistics, etc.

Algorithms: There are several popular reinforcement learning algorithms, such as

* Monte Carlo
* Q- Learning
* SARSA
* DQN
* DDPG
* A3C
* NAF
* TRPO
* etc..

# Literature Review

Let us now look at some of the papers in the field of Reinforcement Learning, each discussing a different RL algorithm, in chronological order.

1989 – Q-Learning by Christopher Watkins

This was the first paper to introduce Q-Learning to the world. Wattkins proved that one-step Q-Learning method can converge to optimal value function and policy. Also discusses the estimation of q\* (now known as Q-Functions) with action-value functions. Defined. Q-Functions. Established the key idea of a Q-Table

1994 – On-Line Q-Learning using Connectionist Systems by G. A. Rummery & M. Niranjan

This paper introduced a new algorithm called Modified Connectionist Q-Learning (MCQ-L) which is also known as SARSA. This new algorithm was then compared with several other Q-Learning algorithms that existed at the time. The comparison was made by testing the algorithms on a robot navigation problem, where a simulated robot had to guide itself to a goal position in the presence of obstacles. The authors showed that MCQ-L and other on-line learning algorithms like it were more robust compared to standard Q-Learning algorithms.

1997 – Reinforcement Learning with Hierarchies of Machines by Ronald P. and Stuart Russel

This paper provided a description of the decomposition of higher-level activities into lower level activities. It also introduced the idea of HAMs, which are finite state machines/programs and work in a non-deterministic way. States that with HAMs, knowledge can be reused across different problems and that it only implies a recombination of component solutions to attack a larger more complex problem. Machines are associated with skills, like when finding a wall, the current machine can call a backoff machine or a follow-wall machine as a policy. The authors of HAM also propose HAMQ, a crossing between HAM and Q-Learning.

2013 – Playing Atari with Deep Reinforcement Learning by several authors from Deepmind Technologies (V. Minh, Koray K., David S., Alex G., Ioannis A. , Daan Wierstra, and Martin R.)

This paper deals with the problem of RL with high-dimensional sensory input, for example in vision and speech processing. Since the number of state-action pairs in these environments can be enormous, algorithms such as Q-Learning struggle because it becomes computationally infeasible to iteratively update the values in the Q-Table. So instead, the authors borrowed ideas from Deep Learning and used a Deep Neural Network to approximate the optimal Q-Function. After training, the network surpassed previous RL algorithms on all six of the games it trained on and surpassed an expert human player on three of them.

Figure : Hierarchy Of Multiple Access Schemes

# Chapter 2: 5G NOMA

# (Non-Orthogonal Multiple Access)

# Introduction

Non-orthogonal multiple access (NOMA) is a multiple access technique which has become an important principle for the design of radio access techniques for the fifth generation (5G) wireless networks. There are several 5G multiple access techniques (shown in the figure below under the NOMA branch), with power domain NOMA is the most popular. However in this paper we will focus on Single Carrier NOMA.

They key idea behind all these NOMA techniques is the same – serving more than one user per orthogonal resource block. In Power Domain NOMA, each user operates in the same band and at the same time where they are distinguished by their power levels. NOMA uses superposition coding at the transmitter and successive interference cancellation (SIC) at the receiver to decode the received signal.

Advantages of NOMA over OMA (Orthogonal Multiple Access):

* Higher spectral efficiency
* Higher connection density
* Enhanced user fairness
* Lower latency
* Supporting diverse QoS

Disadvantages:

* Increased receiver complexity

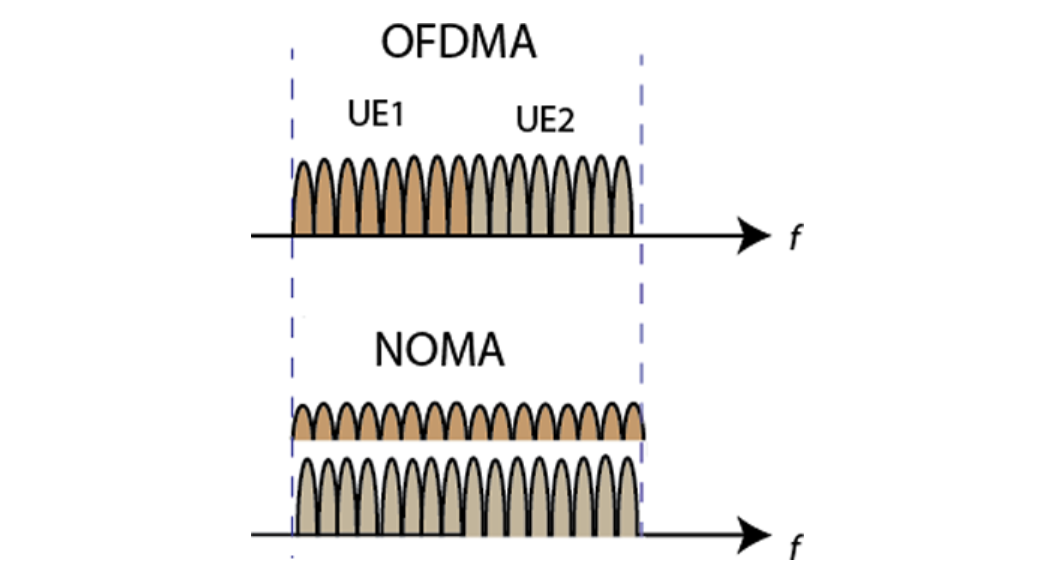


Figure Spectrum sharing for OFDMA and NOMA for two users.

# Literature Review

There have been several important papers discussing the application of NOMA to 5G networks. Some of them have been reviewed below:

Q-Learning NOMA Random Access for IoT-Satellite Terrestrial Relay Networks

by Douglas A. Tubiana, Jamil F., Glauber B., and Richard D. Souza

This paper discusses the existing IoT STRNs and the resource allocation techniques being used in them as well as techniques being considered by other researchers. Next the authors present their system model, proposed protocol, as well as numerical results. Key achievements – the proposed scheme was decentralized, had better normalized throughput than the other schemes that it was compared, and also required lesser resources (channels and relays)

A Tutorial on Nonorthogonal Multiple Access for 5G and Beyond

by Mahmoud Aldababsa, Mesut Toka, Selahattin Gokceli, GuneG Karabulut Kurt, and OLuz Kucur

This was a review article provides an in depth look at 5G NOMA Techniques. Starts by comparing NOMA with OMA in terms of key performance metrics such as spectral efficiency, user fairness, and compatibility. Next, the authors introduce the basic concepts and key terms in the field of NOMA, providing mathematical formulae where needed. Next, they discuss NOMA-MIMO and Cooperative NOMA, and finish off by discussing the practical implementation details for NOMA. The key strong point of this paper was that everything was discussed in depth and formulae were provided wherever required.

# Chapter 5: Work Done

Over the course of a week or so, I tried to implement the algorithm presented in the QL NOMA RA for IoT STRNs paper. However, it was an unsuccessful attempt. The Python code for the same is seen below. Or check the GitHub Link - https://gist.github.com/Anurag-Baundwal/8a2183a9756326266ad509372650b9ef

import numpy as np

from matplotlib import pyplot as plt

# from numpy import unravel\_index

# constants

D = 10              # no of devices

T = 100             # no of time slots

C = 1               # considering only 1 channel for now

R = 4               # no of relays

beta = 3            # for SINR equation

P = 180             # transmit power in dBm

episodes = 100  # no of simulation runs

frames = 50     # ...

alpha = 0.1         # learning rate

gamma = 0.5         # discount factor

for d in range(D):

    # initialise D Q-Tables (1 for each of the D devices) with zeros

    Q = []

    for i in range(d):

        Q[i] = np.zeros((T, C))

    for f in range(frames):

        feedback = [0 for \_ in range(T)] # T feedback bits (one for each timeslot)

        for t in range(T):

            transmitting\_devices = [0 for \_ in range(d)]

            # those iot devices will transmit:

            # for whom q value corresponding to timeslot t is maximum

            # ie, in the q table.. value for row t must be max

            # similarly find which channel c to transmit in

            for i in range(d): # for each iot device

                # check entry of row t in its q table

                max\_index = Q[d].argmax()

                my\_tuple = unravel\_index(Q[d].argmax(), Q[d].shape)

                if t == my\_tuple[0]:

                    transmitting\_devices[d] = 1

                # if it's the max among all rows then transmit

                # transmitting\_device[d] = 1 -> changed it from 0 to 1 to indicate that device d transmitted

            # now we know which devices transmitted

            # we can assign power and distance (dist will be random value btw 0 and 10 km)

            # we will get sinr using the formula

            # sort sinr values

            # keep decoding one by one till the  sinr value is below 2\*\*beta - 1

            # Relays employ SIC to decode possible superimposed messages transmitted at same channel

            sinr = # find SINR

            if sinr >= math.pow(2, beta) - 1:

                # transmission successful: set feedback bit to 1

            else:

                # transmission unsuccessful: set feeback bit to -1

    # now the relays transmit T feedback bits (one for each timeslot).

    # These feedback bits indicate whether the transmission was successful

    # Based on this feedback bit we will update the Q Tables

    for i in range(d):

        # update Q-Table of device i

        reward = #check feedback bit

        Q[i][timeslot][channel] = (1-alpha)\*(Q[i][timeslot][channel]) + alpha\*(reward + max(Q[]))

    # now find normalised throughput. formula:

    normalised\_throughput = beta\*(devices\*messages)/(T\*C\*total\_messages)

    # devices: no. of devices that successfully trans

    # enter this normalised throughput in an array of size D

    # plot the values in this array against the number of devices

    # normalised throughput on y axis and d on x axis

    throughputs = []

    throughputs[d] = normalised\_throughput

# plot

x = np.arrange(1, D+1)

y = throughputs[x]

plt.title("Throughput vs D, with R = 4 relays and C = 1 channels")

plt.xlabel("Number of Devices (D)")

plt.ylabel("Normalised Throughput [bps/Hz]")

plt.plot(x,y,"ob")

plt.show()

# Chapter 6: Conclusion & Future Work

In the future, I would like to finish the python code mentioned in Chapter 5 and verify the results of the QL NOMA RA paper. I would also like to learn more about Reinforcement Learning and Machine Learning in general.

# Bibliography

1. Q-Learning by Christopher Watkins, 1989 (<https://www.researchgate.net/publication/220344150_Technical_Note_Q-Learning>)
2. On-Line Q-Learning using Connectionist Systems by G. A. Rummery & M. Niranjan, 1994 (<http://mi.eng.cam.ac.uk/reports/svr-ftp/auto-pdf/rummery_tr166.pdf>)
3. Reinforcement Learning with Hierarchies of Machines by Ronald P. and Stuart Russel, 1997 (<https://proceedings.neurips.cc/paper/1997/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>)
4. Playing Atari with Deep Reinforcement Learning by several authors from Deepmind Technologies (V. Minh, Koray K., David S., Alex G., Ioannis A. , Daan Wierstra, and Martin R.), 2013 (<https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf>)
5. Q-Learning NOMA Random Access for IoT-Satellite Terrestrial Relay Networks, 2022 (<https://ieeexplore.ieee.org/document/9760513>)
6. A Tutorial on Nonorthogonal Multiple Access for 5G and Beyond

by Mahmoud Aldababsa, Mesut Toka, Selahattin Gokceli, GuneG Karabulut Kurt, and OLuz Kucur, 2019 (https://arxiv.org/abs/1902.08992)