Fukuoka Institute of Technology
福岡工業大学
GRADUATE SCHOOL OF ENGINEERING

IoT Device Selection in Opportunistic Networks: Implementation and Performance Evaluation of Fuzzy-based Intelligent Systems and a Testbed

by

Miralda Cuka

PhD Supervisor

Prof. Leonard Barolli

Graduate School of Engineering

Fukuoka Institute of Technology
Japan

2020

Contents

List of Figures

List of Tables

Abstract

In Opportunistic Networks (OppNets) the contacts between Internet of Things (IoT) devices (nodes) are intermittent and links are highly variable. Upon receiving a message a device will store it in the buffer until another node comes in the transmission range or a forwarding opportunity exists. The IoT network consists of connected physical objects and devices with high mobility. By using the mobility of IoT devices, the OppNets provide a self-organizing network as a communication opportunity. The IoT devices generate and exchange a huge amount of data through heterogeneous networks and OppNets ease the concept of heterogeneity with their independence on decentralized infrastructure. The IoT network consists of different devices with different resource capabilities. When multiple IoT devices are deployed densely, there is a possibility that a node may reside in the coverage area of multiple devices. Thus, when a task requires an IoT device to complete it, it is very important to find the best device for that specific request. The IoT devices should be selected based on different parameters in order to achieve better network connectivity, stability and user coverage. In OppNets an end-to-end path between source and destination may not exist and network partitions occur often. Some of the most common issues for OppNets are energy consumption, storage constraint, limited contact opportunities and finding an optimal and robust topology of the network devices to support connectivity services to events. To deal with these issues many parameters should be considered which make the problem NP-Hard. Thus, the heuristic and intelligent algorithms are good solutions. In this research work, we consider IoT device selection in OppNets and propose new parameters and implement different intelligent systems based on Fuzzy Logic (FL). The proposed systems can be used in different environments and applications. We carried out many simulations and found that the performance of implemented systems is good. We observed that the complexity of the systems increases with the increase of the number of parameters. We also implemented a testbed and performed experiments in order to compare the simulation and experimental results. The experimental results

show that the implemented testbed makes a good selection of IoT devices. This thesis contributes in the research field as following: 1) Proposal of new parameters for IoT device selection in OppNets. 2) Proposal and implementation of intelligent systems based on FL for appropriate selection of IoT devices in OppNet. 3) Performance evaluation of implemented systems for different parameters and scenarios. 4) Comparison of implemented intelligent simulated systems. 5) Implementation of a testbed for OppNet and its application in a real scenario. 6) Give insights about future developments and application of OppNets and IoT as important technologies for wireless communications. This thesis is constructed with 9 chapters. In Chapter 1 is presented the background, motivation and thesis structure. Chapter 2 introduces the next generation wireless networks and describes in more details 5G cellular network technologies, Software-Defined Wide Area Network (SDWAN) and Mobile Ad-hoc Networks (MANETs). In Chapter 3, we introduce the architecture, challenges and applications of IoT and OppNets. In Chapter 4, we introduce Intelligent Algorithms. In Chapter 5, we present Fuzzy Logic. In Chapter 6, we introduce our proposed Fuzzy-based intelligent systems for IoT device selection in OppNets. In Chapter 7, are shown the performance evaluation results of proposed simulation systems. In Chapter 8, we present the testbed implementation and evaluation. In Chapter 9, we conclude this thesis and give the future work.

Chapter 1

Introduction

1.1 Background

Future communication systems are going to be increasingly complex, with thousands of heterogeneous nodes with diverse capabilities and different technologies interconnected with the aim of providing users ubiquitous access to information and advanced services at a high quality level at any place and time in a cost efficient manner. Opportunistic Networks (OppNets) provide an alternative way to support the diffusion of information in special locations within a city, particularly in crowded spaces where current wireless technologies can exhibit congestion issues. OppNets are an extension of Mobile Ad-hoc Networks (MANETs) so they face some issues similar to MANET such as limited bandwidth, disconnections and variable links. It starts as a seed network consisting of an original set of nodes and expands by growing to a larger network with new nodes consistently becoming part of the network. Nodes may be static or fixed so this makes the network easy to deploy since it's not dependent on infrastructure, making OppNets suitable for interplanetary communications, terrestrial wireless networks, mobile sensor networks and so on.

The OppNets are variants of Delay Tolerant Networks (DTNs) which is a class of networks that has emerged in the recent times. Routing is a very challenging task due to un-connected nature, sparse connectivity, limited resources and no infrastructure. Routing methods depend on schemes that utilize node mobility by having the node carry the message and wait for an opportunity to transfer the message to the next node rather than transmitting them over a fixed source-to-destination path.

Internet of Things (IoT) is a network of devices which collect and exchange data and than act on this information. IoT is the consistently growing network of objects which connect the physical world with the virtual one. These objects generate huge amount of data which travels through different networks.

1.2 Thesis Purpose and Contribution

In this thesis by considering the challenges of selecting an IoT device in OppNets, we proposed and implemented four simulation systems and one testbed based on Fuzzy Logic (FL). In an OppNet scenario, IoT devices will create a seed network and later add helpers when an event that needs action happens. Based on the resources each network device has, some will be better suited than others for acting upon the event. These resources are measured by the parameters each device has. To be able to provide effective help, these parameters are used as inputs on our proposed systems and decisions are made based on usability of each device.

In this work, we have proposed a meta-heuristic platform based on FL for choosing the best IoT device to perform a task based on specific parameters that apply to OppNets challenges. Parameter combination varies for each proposed and implemented system because we have consistently tried to consider many scenarios and optimize the proposed system by selecting new parameters.

The first system we have proposed is IoT Device Selection System 1 (IDSS1), which uses three input parameters: IoT Device Speed (IDS), IoT Device Remaining Energy (IDRE), IoT Device Distance from Task (IDDT). For our second system IoT Device Selection System 2 (IDSS2), we decided to increase its complexity by adding a fourth parameter IDS, IDDT, IDRE, IoT Device Storage (IDST). In our third system IoT Device Selection System 3 (IDSS3), we decided to use four parameters: IoT Device Waiting Time (IDWT), IoT Device Security (IDSC), IDRE, IDST. Different from our second system, in IDSS3 we have considered the waiting time and security of a device as two new parameters. In an OppNet scenario, where an event is taking place the effectiveness of completing said event is eased by having a device with many connections. These connections increase the chances of discovering new helpers which may become essential to complete the task. We have summarized this device property in one parameter, IoT Device Node Centrality (IDNC) which is included in the fourth system, IoT Device Selection System 4 (IDSS4). In IDSS4 we have four input parameters: IDNC, IDWT, IDRE, IDST.

Based on input parameters, our proposed systems give the possibilities of each device to be selected as our output parameter IoT Device Selection Decision (IDSD). Comparing complexity of IDSS1 with IDSS2, IDSS3 and IDSS4, the systems with four parameters are more complex than IDSS1 since they need more computational resources.

With simulation systems we are able to emulate different scenarios but to gain a further insight into a real life system, a testbed is the best choice. A simulation system only focuses on a subset of properties of the real system while the testbed tests a system behavior based on certain inputs and reflects a more realistic scenario. We have implemented a simulation system for selecting IoT nodes in Oppnets, IoT Node Selection System 1 (INSS1) with four input parameters: Node's Distance to Task, Node's Remaining Energy, Node's Buffer Occupancy, Node Inter Contact Time and Node Selection Decision as an output parameter. We implemented a testbed for further evaluation and compared the results obtained by both the simulation system and the tesbed and evaluate if they are comparable.

This thesis contributes in the research field as following:

1. Proposal of new parameters for IoT device selection in OppNets.
2. Proposal and implementation of intelligent systems based on FL for appropriate selection of IoT devices in OppNet.
3. Performance evaluation of implemented systems for different parameters and scenarios.
4. Comparison of implemented intelligent simulated systems.
5. Implementation of a testbed for OppNet and its application in a real scenario.
6. Give insights about future developments and application of OppNets and IoT as important technologies for wireless communications.

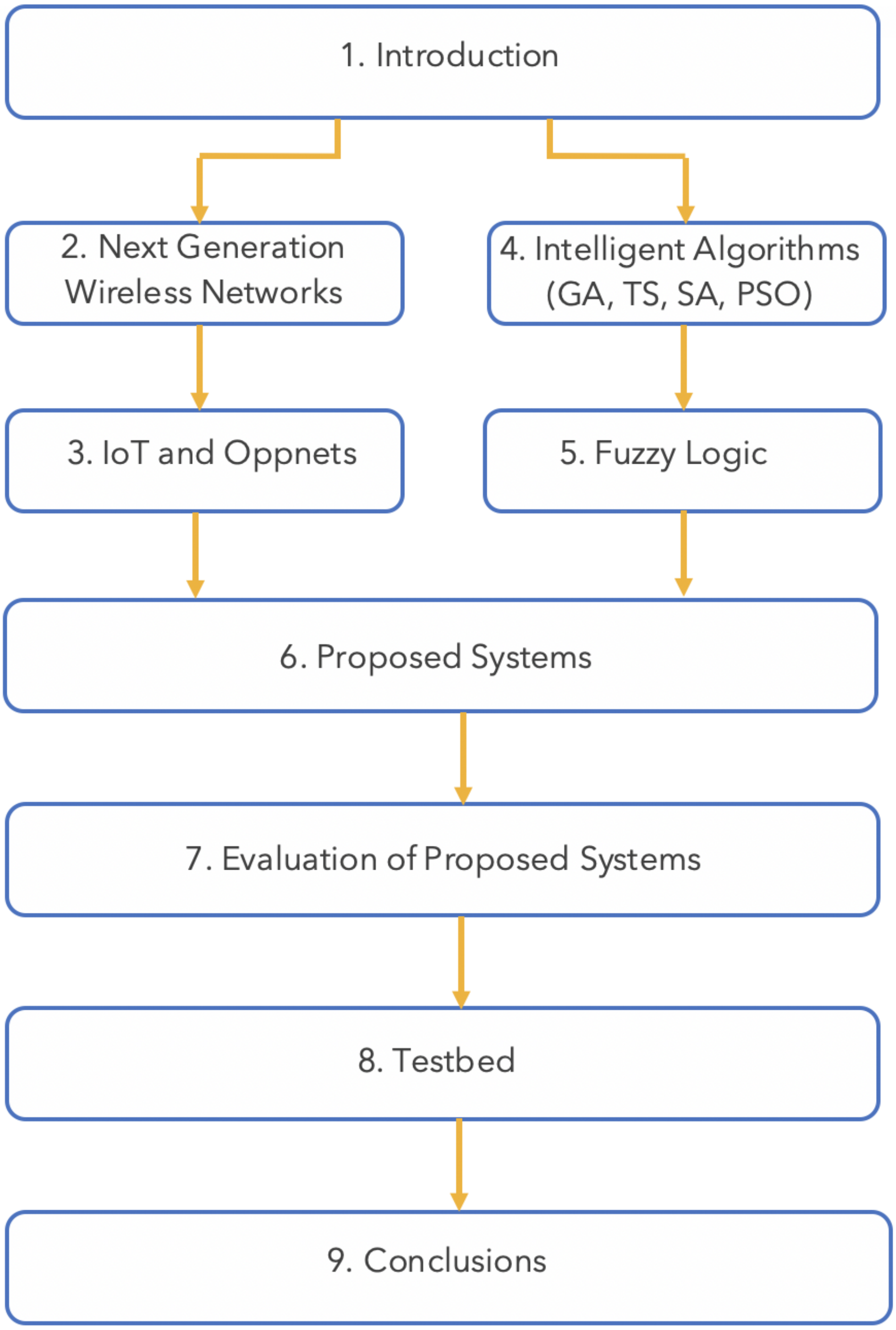


Figure 1.1: Thesis structure.

1.3 Thesis Outline

This thesis consists of 9 chapters and its structure is given in Fig.1.1.

The thesis is organized as follows.

Chapter 1 presents the background, motivation and thesis structure.

Chapter 2 introduces general aspects of wireless networks and describes Next Generation Wireless Networks (NGWN).

In Chapter 3, we introduce Internet of Things (IoT), Opportunistic Networks (OppNets) and its architecture.

In Chapter 4, we present Intelligent Algorithms (IA).

In Chapter 5, we present FL, Fuzzy sets and Fuzzy membership functions.

In Chapter 6, we present our proposed fuzzy-based simulation systems for IoT device selection in OppNets.

In Chapter 7, are shown the performance evaluation results of proposed simulation systems.

In Chapter 8, we show testbed implementation and evaluation.

In Chapter 9, we conclude this thesis and give the conclusions and future work.

Chapter 2

Next Generation Wireless Networks

2.1 Introduction

Technology companies and mobile carriers are working on overcoming the obstacles of existing wireless networks by adopting new wireless technologies that aim to define new global standards, handle the surge in data traffic and offer more spectrum bandwidth. Wireless IP-based Networks have evolved to overcome limitations and to support the rapidly increasing data. This rapid increase of the mobile nodes and the huge amount of data generated has created challenges for the wireless networks. The providers of services are facing bandwidth shortage hence the need for new technologies arises. The Next Generation Mobile Networks (NGMN) aim to offer ubiquitous connectivity in remote and challenging areas, lower latency and enable higher mobile speed. NGWN are bringing the vast array of Internet services and providing users with a successful platform for future mobile services. One of the most prominent next generation technologies is Fifth Generation (5G) networks. As the next iteration of 4G Long term Evolution (LTE), 5G will be able to support and scale the massiveness of IoT devices. The next generation 5G wireless communications will provide very high data rates, bring connection to remote and isolated areas, increase quality of service (QoS) and offer very low latency. To be able to accommodate the increasing demand for data that comes with the addition of new users, 5G technologies were developed [1].

2.2 Architecture of 5G

The 5G systems will support a wide range of services and applications by meeting the requirements of the fully connected and highly mobile societies. The spread of connected devices will pave the way to many services and will aid the communication needs of machine-to-human and machine-to-machine applications. The coexistence of multiple applications will impose many challenges that 5G has to overcome. To satisfy the demands, the concept of slicing has emerged as an efficient way for serving all the required services on a common infrastructure. Slicing has been conceptualized as a way of optimizing, simplifying and sharing the infrastructure between operators. With network slicing, new capabilities are brought to 5G infrastructures which bring flexibility in deployment and efficient resource utilization. With the many new services provided by 5G, in addition to enhanced Mobile Broadband (eMBB), two new mobile services: ultra-reliable and Low-latency Communications (uRLLC) and massive Machine Type Communications (mMTC) have to meet requirements. eMBB provides support for services with high bandwidth requirements such as Augmented Reality (AR), High Definition (HD), Virtual Reality (VR). uRLLC aims to support latency sensitive services such as remote management and assisted automated driving. mMTC will be able to support the massive amount of IoT devices expected to become part of the network and focuses on services that have high requirements for connection density requirements such as smart cities. In Fig.2.1 is shown the architecture of a 5G network for easy deployment of IoT devices since one of the key features of 5G is the support of the IoT applications.

As an extension of 4G broadband service, 5G allows an efficient scheduling of wireless resources to devices so no two devices access the same resource simultaneously. The key to 5G, is providing diversified services with the end-to-end network slicing and meet these services demands with Software-Defined Networking (SDN) and Network Functions Virtualization (NFV), which support the physical infrastructure and brings cloud closer to core network, transport and access.

2.2.1 Cloud RAN

The Radio Access Network (RAN) has always evolved with the coming generations of mobile communications. In a RAN, radio sites coordinate the management of resources and provide radio access. When a device is connected wirelessly to the core network, its signal will travel within the network traffic and be transited by RAN to different wireless

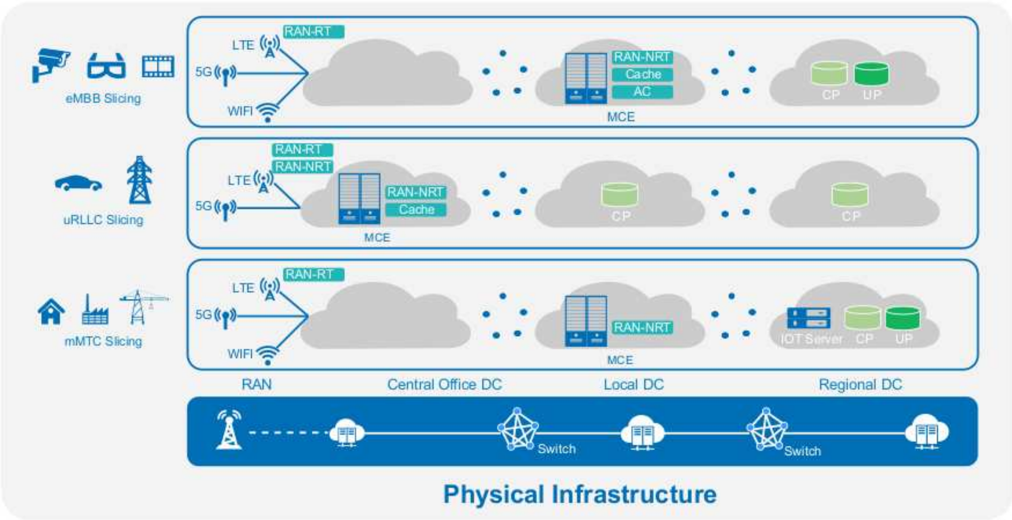


Figure 2.1: 5G Services Architecture Layer (*Figure from 5G Network Architecture, A High-Level Perspective, Huawei Technologies.*)

endpoints. The real time functions of RAN are power control, access network scheduling, retransmission, coding, interference coordination and link adaptation. All these functions require a high computer load and performance in real time. The non-real time functions of RAN include, cell selection, re-selection, inter-cell handover and multiple connection convergence. For the deployment of sites, dedicated hardware with high accelerator processing specifications and performance in close proximity to services, must be included. These functions require minimal real-time performance, latency requirements to dozens of milliseconds and are suitable for centralized deployment.

Multi-connectivity is gaining a reputation as an underlying fundamental construct for the deployment of the future network architecture. A huge leap in radio network deployment is that CRAN can be seamlessly deployed in a unified network architecture.

In current fragmented networks, increasing speed and reducing latency can improve user experience. Reliable high-speed data cannot depend on a single frequency band or standard connections. In heterogeneous networks, multi-connectivity helps provide an optimal user experience based on LTE and 5G capabilities, such as high bandwidth and rates of high frequency, network coverage and reliable mobility of low frequency, and accessible Wi-Fi resources.

In scenarios that require high bandwidth or continuity, a user requires multiple concurrent connections. To have high bandwidth, data aggregation of data from different multiple subscriptions like: LTE, Wi-Fi, 5G is required. After a user has accessed a 5G high-frequency small cell, a LTE network access is required to maintain continuity.

In scenarios that source multiple technologies, Cloud RAN serves as an anchor for data connection which noticeably reduces alternative transmission. In the traditional architecture integrating base stations as an anchor for data connection, LTE, 5G, and Wi-Fi data are aggregated into a non-real time processing module of a specific standard to be forwarded to each access point. In the Cloud RAN architecture, non-real time processing function modules in access points of different modes are integrated into the Mobile Cloud Engine (MCE), which serves as an anchor for data connection. Data flows are transmitted to each access point over the MCE, which prevents alternative transmission and reduces transmission cost by 15%, and latency by 10 ms.

2.2.2 5G Challenges

Even though 4G has not been around long, the widespread use of devices, capacity of data, and the emergence of IoT, demand a network that can handle the diversified needs. The principal goal of 5G is to overcome the limitations of 4G and satisfy the needs of services and applications that are always evolving. Before deploying a large scale network that will fulfill demands, several challenges must be met.

Massiveness of IoT: It is predicted that IoT will create a massive increase in the number of connections and devices across the network. The number of connected devices has already exceeded the population and many more are added to the network. Even though the amount of data generated by simple devices is relatively small, they still require infrastructure for managing the data and the physical connections. Previous wireless mobile networks limited the number of connected users on specific nodes with control mechanisms. But old access control mechanisms cannot handle the growth of IoT devices, so new scheduling and control mechanisms are required.

Big Data: One of the main reasons that caused the early development of 5G was the large volume of data. The amount of data generated and passing through mobile networks is increasing at about 25 to 50 percent a year, and it will continue to grow. This grow is not only because of the new applications that require high data rates, but also because of high screen resolution in 3D videos. Unlike 3G, 4G and 5G are all IP networks so the challenge of data capacity in an end to end network increases. Furthermore, this is not only for the air interface, but also for the access/ core network.

Capacity and Cost Ratio: Users are generating more and more data, but are unwilling to pay more. Therefore, increasing the network capacity without increasing the cost is very challenging. One way is to separate the user data from the distribution of control, to meet the data requirements. This can be achieved by using macro cells to provide control plane signaling to wide areas and small cells within the macro cell for user plane data without added complexity. By using existing sites and spectrum to increase capacity no significant extra cost is added.

Fast Deployment of 5G Architecture: Speed of deployment of 3G and 4G was restricted by the speed at which suitable backhaul network capacity could be provided to each new site, and the capacity/flexibility of the backhaul. 5G will be challenged to further develop CRAN as another evolution in network design, complementing the user and control plane separation in the move towards more flexible cloud based networks. In this concept, some functions of the RAN are moved from the cell site back into a consolidated baseband cloud service. This provides a solution to support scaling and economy, leading to deployment flexibility and easier reconfiguration, because the core signaling and intelligence is held within the cloud and the only localized physical elements are the RF transceivers to provide RF connection to users.

Uninterrupted Services for Emergency Situations Many services such as emergency services and medical monitoring require a high level of reliability and real time data. For example one area where wireless networks are being used is to provide a remote patient monitoring, care and access to its medical data so trained staff can provide remote support. Also, other emergency services such as police and ambulance services, need a always available high reliability link with call dropping or busy network issues. These dedicated services are offered by dedicated networks, but these networks do not have the resources for high bandwidth, high data capacity and reasonable coverage. Some of 5G requirements are for real time interaction and high data rates, as well as a fast service respond time. So 5G is supposed to offer "ultra reliable" services where the ability to connect and operate is independent on infrastructure. This is based on using device to device direct communication, ad-hoc backhaul, networking, and flexible re-configuration of networks.

Augmented Reality (AR): AR are being vastly deployed on personal and portable devices, increasing the demands for network capacity and performance. One key aspect is that to enable interactions between the real and virtual world, latency must be very small since human brain is very sensitive to time delays. Thus when processing visual data, VR services cannot be delivered unless the latency and delay are very small. The round trip between devices and servers must be extremely low and the communication link must be optimized. New signal/routing architectures will also be required as the overall latency required cannot be achieved using traditional centralized server architectures. So it is expected that critical low latency services will need infrastructure and architecture to locate the service/server close to the user, to ensure latency between user and service is minimized.

Machine-to-Machine: M2M have an important role in the overall IoT where one sector where this is being pushed forward is the automotive sector. There have already been developed and deployed automotive wireless connectivity applications, where the vehicles are used as a hub and cellular networks as the back-haul. Due to available battery power, vehicles are used as a base station or as a relay node within the network. Intelligent transport systems are creating demand for vehicle-to-vehicle and vehicle-to-infrastructure communications, as well as linking the vehicle to other devices. And an ultimate goal that now looks within reach is fully autonomous driving, but this will require secure and reliable communications for a widespread public deployment (beyond current trials and deployment scenarios that still use driver intervention as a back-up mode). 5G networks should be able to support this, if the network can deliver the capacity/coverage/ latency combination required by use of heterogeneous network technologies. The challenge is to go from concept to network architecture to technology, and meet these requirements with flexibility, but also satisfy the high reliability/availability demands of autonomous driving.

Device-to-Device: Device-to-Device communications have not been fully supported in previous cellular networks. In cellular networks the data goes from device to base station before going back to device, so direct links are not available. Such direct communications have some drawbacks as they have narrow spectrum bandwidth and limited capacity for transmitting data. In previous cellular networks, the technology of push-to-talk was implemented to deliver a similar experience, however, sufficient coverage for critical applications could not be guaranteed. In 5G networks the challenge is to support these types of communications and allow direct communication between devices.

Figure 2.2: A cloud based SD-WAN.

2.3 Software Defined Networking

Traditional network architecture is failing to meet with the requirements imposed by the huge amount of data generated from big enterprises. On account of this, Software Defined Networking (SDN) was developed to transform the existing networking architecture into a new one which supports all this change. In a SDN architecture, the network infrastructure is separated from data, control and application plane. This independency enables enterprises to have more control and build highly flexible and scalable networks.

- **Hierarchical architecture:**
  Conventional networks are hierarchical which served the client-server model, but such architectures do not adapt to the virtualization of servers, cloud services and the big data requirements of enterprises. Users are changing the traffic patterns by accessing different applications, databases, servers with any type of devices from anywhere, anytime.

- **Diversity of devices:**
  Devices are increasingly more diverse and in need of a consistent connection to the network. This puts the enterprises under pressure to accommodate all these devices in a seamless manner.

Securely accessing corporate resources requires mobile users to connect to a branch or HQ firewall VPN which could be very far from their location. This causes user experience issues, and encourages compliance violations (for example, direct access to Cloud services that bypasses corporate security policy). Ultimately, the mobile workforce is not effectively covered by the WAN. The Cloud-based, secure SD-WAN shown in Fig.2.2 is aiming to address these challenges.

2.4 Software Defined Wide Area Network

With the new digital innovations, businesses are eager to adopt new technologies to reduce cost and increase productivity. Traditional Wide Area Network (WAN) connects users to applications hosted in data centers. However, traditional WAN depend heavily on infrastructure which causes high latencies and single point failure. Furthermore, applications are moving from data centers into cloud and users progressively more mobile are using more and more devices.

Software-Defined Wide Area Network (SD-WAN) have evolved as a solution for these issues as it extends the benefits of cloud to applications from every location of the network. Basically with SD-WAN you get the most out of your network resources and existing infrastructure.

SD-WAN, handles traffic based on priority, quality of service and other business requirements by using software to intelligently steer the traffic across the WAN. Furthermore, SD-WAN enhances security by eliminating the dependency on other devices as security solutions are built into the EDGE routers which ensures user's traffic protection. Other key benefits are the increase of applications performance, reduces network complexity and cost, increasing bandwidth. They also simplify network management tasks and provide a programmable flexible interface for controlling the network.

2.5 MANET Characteristics

Mobile Ad-hoc NETworks (MANETs) are new paradigm of networks, offering unrestricted mobility without any underlying infrastructure. Basically, MANETs are a collection of nodes communicating with each other by forming a multi-hop network. In Fig.2.3 is shown the architecture of MANETs. In the following we show the characteristics of a MANET:

Dynamic Topologies: Nodes are free to move arbitrarily. The network topology may change randomly and have no restriction on their distance from other nodes. As a result of this random movement, the whole topology is changing in an unpredictable manner,

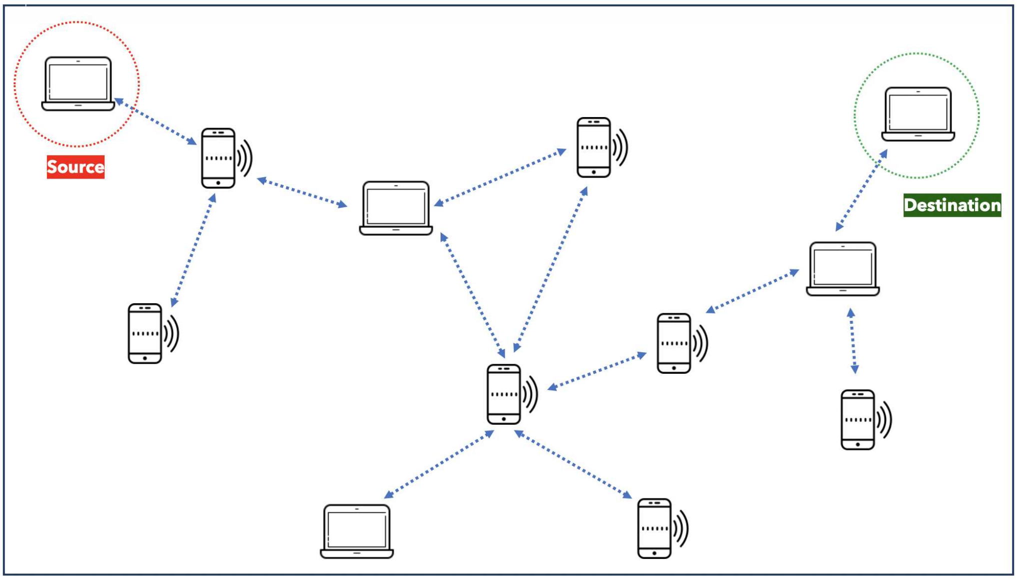


Figure 2.3: MANETs architecture.

which in turn gives rise to both directional as well as unidirectional links between the nodes.

Energy Constrained Operation: Almost all the nodes in an Ad-hoc network rely on batteries or other exhaustive means for their energy. The battery reduces due to extra work performed by the node in order to increase the lifetime of the network. Therefore, energy conservation is an important design optimization criterion.

Bandwidth Constraint: The capacity of wireless networks is significantly lower than the capacity of infrastructure based networks. Due to the effect of noise, interference, fading and multiple access, throughput of a wireless network is much lower. This results in an obstacle for bandwidth utilization.

Limited Physical Security: MANETs are distributed systems very vulnerable to physical security where physical threats are always present. As a result, there is an increased possibility of denial-of-service type attacks, intrusions, spoofing and masquerading.

2.5.1 MANET Challenges

The area of applications for MANETs varies, ranging from small static networks with constrained power sources, to highly dynamic large scale networks. Despite the application, MANET need distributed algorithms to determine routing, link scheduling and network organization. Due to their dynamic topologies, protocol design and delivery of messages in a decentralized environment is a complex task. In a static network, the shortest path from source to destination is calculated based on a cost function, but this idea is not extended in MANET due to variable wireless link quality, multiuser interference, power constraints, fading, and changes in topology. However, in some application such as military applications, latency, recovery from failure are significant concerns. Some challenges must still be solved since they make these networks very vulnerable.

Routing: Routing packets between two nodes with a network topology always changing, is a challenging task. Reactive routing protocols are better suited for these cases rather than proactive protocols. Due to the random movement of nodes in the network, multicast tree is not static, so multicast routing is always challenging. Also path from sender to receiver may contain multiple hops, which is more complex than single hop [2].

Security and Reliability: Like most of the wireless networks which have common security vulnerabilities, Ad-hoc networks have their particular security issues such as, malicious neighbors relaying packets. So, different authentication and key management schemes are required. Furthermore, because of the limited transmission range, mobility-induced packet losses and data transmission errors, wireless links may become unreliable.

Quality of Service (QoS): In a constantly changing network, there exist different levels of QoS. The inherent stochastic feature of communications quality in a MANET makes it difficult to offer guaranteed services to a device, so an adaptive QoS to support multimedia services over the traditional networks must be implemented.

Inter-networking: In addition to the communication within an Ad-hoc network, inter-networking between MANETs and fixed networks (mainly IP based) is often expected in many cases. The coexistence of routing protocols in such environments, poses a challenge for mobility management [3].

Power Consumption and Conservation: For most of the light-weight mobile terminals, the communication related functions should be optimized for lean power consumption. Conservation of power and power-aware routing must be taken into consideration.

Chapter 3

IoT and Opportunistic Networks

3.1 Internet of Things (IoT)

Existing networks have already brought connectivity to a broad range of devices, such as mobiles, laptops, tablets, PC, etc. The Internet of Things (IoT) will extend the connectivity to devices beyond just mobile phones and laptops, but to buildings, wearables, cars, different things and objects. Considered as the next evolution of the Internet, IoT will enable devices to collect, analyze and exchange data. IoT is an intelligent network of enhanced and smart devices equipped with an IP address, which communicate with each other without the need of human interaction [4]. Multiple services will be enabled by connecting humans with devices and processes. A wide heterogeneous well connected network will benefit billions of people, economies, industries and potentially improve societies [5].

As shown in Fig.3.1, IoT architecture is made of Edge, Fog and Cloud layers which complement each other. Many enterprises are now mitigating toward a edge/fog/cloud infrastructure to increase the utilization of the end devices. IoT Edge layer is comprised of all smart IoT devices, sensors and embedded systems with varying operating systems, batteries, CPU Types, buffer sizes, which process the data directly or forward it to a node layer. The concept "fog computing" was introduced by Cisco company as an extension of the network's outer perimeter in which data generated from the IoT devices is preprocessed by mini data centered servers before getting uploaded to the cloud. Large scale IoT architectures have problems with latency, where edge devices have not the necessary resources to process all the data from end devices and sensors. Every device with a network connectivity, computing capabilities and storage can be a fog node. It is estimated

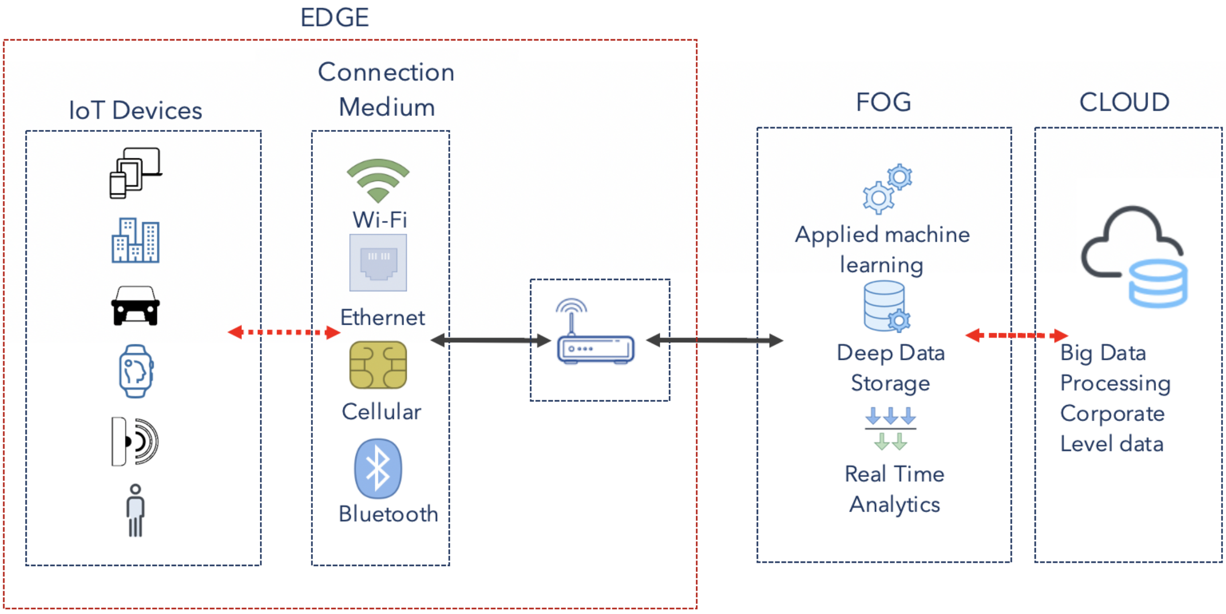


Figure 3.1: IoT Architecture Layers [7].

that the amount of the data generated that needs to be analyzed on devices is enormous. IoT solutions aim to minimize latency so analyzing data to where it is collected, helps by offloading huge amount of data from the network traffic [6].

This layer acts as an intermediary between IoT devices and the cloud. Fog nodes decide whether to process the data locally, or send it to the cloud for further analysis and processing.

The cloud platform, receives the aggregated data from multiple fog nodes and performs detailed analysis for deeper insights benefiting businesses. The difference between fog and cloud computing is how and when the data is processed. Cloud computing is made of centralized data centers with powerful resources such as back-end servers. Therefore, processes huge amount of data from multiple edge/fog nodes and performs high-order computations such as predictive analysis.

3.1.1 IoT Applications

With the current hype around IoT, more and more companies are getting involved in investing on prominent technologies.

Smart Home: A smart or automated home is an living environment equipped with smart objects which connect to the outside Internet via a gateway. Home automatization allows the homes to be fully connected and be internally or externally controlled. Smart homes started as simple systems for lighting and heating, but have evolved into smart technologies which include a broad range of devices and appliances. In Fig.3.2 is shown an estimation of the number of connected IoT devices. It is expected that the number of IoT devices will reach 75 billion by 2025.

Smart City: Implementation of information and communication technologies for urban areas will improve the quality of life. Due to many opportunities people encounter in urbanized areas, more and more people are living in these areas, with more than half estimated to be living by 2050. This migration of people toward urban areas has accelerated technological inventions to tackle with issues such as scarcity of resources, environmental changes and globalization. The unprecedented volume of data gathered each day has a tremendous cost but provides insights on how the demand patterns are changing. Smart technologies give faster, low costing solutions, optimize systems and create a more livable city. Smart cities have a wide area of applications such as security, health care, energy, economic development etc.

Smart Cars: Modern vehicles are equipped with sensors, services, enhanced computation systems and features that allow them to collect information. A smart car may be a self-driving car which senses its environment and performs actions with no human input, or a smart environment which connects with other cars and collect data in real-time to avoid traffic jams, pre-order parts that need replacing, avoid accidents and so on [8].

Wearables: Accessories are getting equipped with sensors and software which perform tasks same as mobile phones and laptops with added functionalities such as real time health monitoring. In case of wearables, sensors are placed closest to skin and continuously register vital parameters and movement. Using Bluetooth Low Energy (BLE) protocol they connect to other wearable devices such as smartphones, collect data and upload the gathered data to cloud for storing and deep analysis.

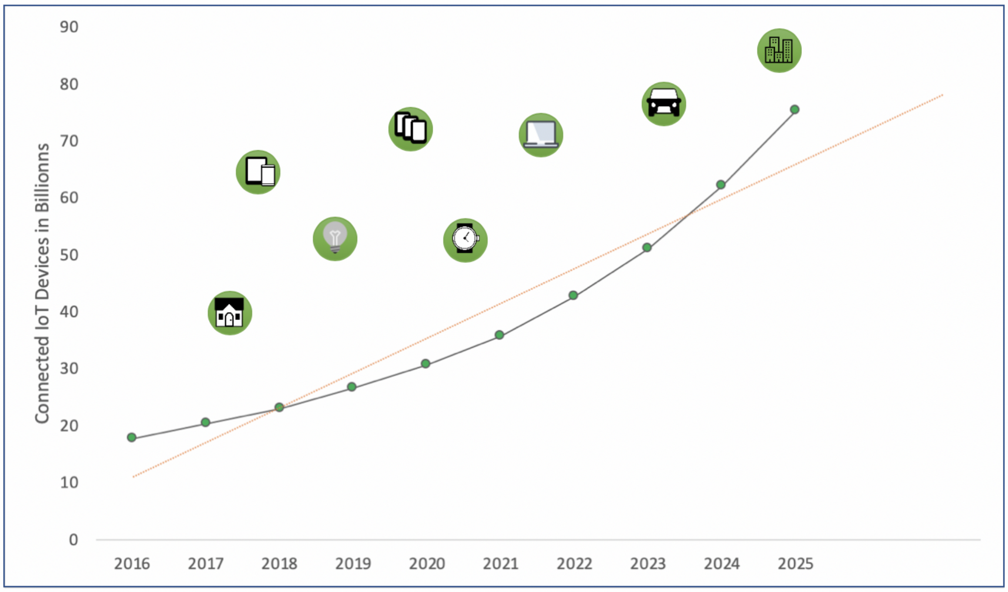


Figure 3.2: An estimation of the number of connected IoT devices (*Data from statista.com, Nov 27, 2016).*

3.1.2 IoT Challenges

As IoT evolves to a bigger network there are many challenges to overcome. IoT is changing the nature of Internet and as with many new technologies it will bring a broad range of benefits but also face many key challenges which are listed below:

Management Capabilities: As IoT networks evolves, it is becoming more and more complex. The number of devices will exponentially increase which means the amount of data to be generated and processed will also increase. IoT connected devices, range from small sensors to powerful devices and gateways which connect to each other. These different devices need to pass through a process of authentication, configuring, provisioning in order to manage the devices and have a high bandwidth and persistent connectivity.

Security: IoT platform offers a tremendous market for opportunities but it also poses a security risk. IoT consists of different networks so some may be vulnerable to attacks becoming a threat to devices in other networks. These compromised networks can be used as gateways to unsecured networks, allowing sensitive data to be extracted. An IoT device carries a vast amount of sensitive data which is personal to user, such as credit cards details, health information, medical history. Different from mobile devices, laptops, tablets where security has been considered during the design phase, for devices such as appliances and other objects security has not been a priority. Many IoT devices are resource-constraint and cannot compute security procedures such as advanced encryption or other measures.

Scalability: Scalability is a very big challenge since IoT devices need to have the ability to adapt to changes in the environment and not be affected by future changes. The addition or withdrawal of the devices should not affect the network. What makes scalability a challenge is the different aspects that make up an IoT platform. Product companies need to consider and take into account different aspects such as manufacturing devices with more capacities for data, make them more durable by putting security first.

Standardization: The number of objects connected to the Internet exceeds the world's population and each of them generates a very large volume of data which needs to be managed, processed and exchanged securely. Every company is aiming at an enterprise level functioning IoT platform and are building their strategies to accommodate their business interests. As a consequence many IoT device's users are required to download softwares and drivers to use existing technologies. Interoperability has been achieved through multi-protocol gateways, but scalability in IoT will lower the cost of data transfer, device manufacturing and reduce the gap between protocols.

3.2 Opportunistic Networks (OppNets)

Designed as a specialized ad hoc network suitable for applications such as emergency responses, OppNets are considered a sub-class of Delay-Tolerant Network (DTN) where communication opportunities (contacts) are intermittent, so an end-to-end path between the source and the destination may never exist. The network starts as a seed network made up of a small group of nodes and grows opportunistically during operation since nodes can join or leave the network at any given time. The link performance in an OppNets is highly variable. Therefore, TCP/IP protocol will break in this kind of environment because an end-to-end path between the source and the destination may only exist for a brief and unpredictable period of time.

Long propagation and variable queuing delays might be introduced and many Internet protocols which are designed to assume quick return of acknowledgements and data, can fail to work in such networks. One possible solution to resolve the above issues is to exploit node mobility and local forwarding in order to transfer data. Data can be stored and carried by taking advantage of node mobility and then forwarded during opportunistic

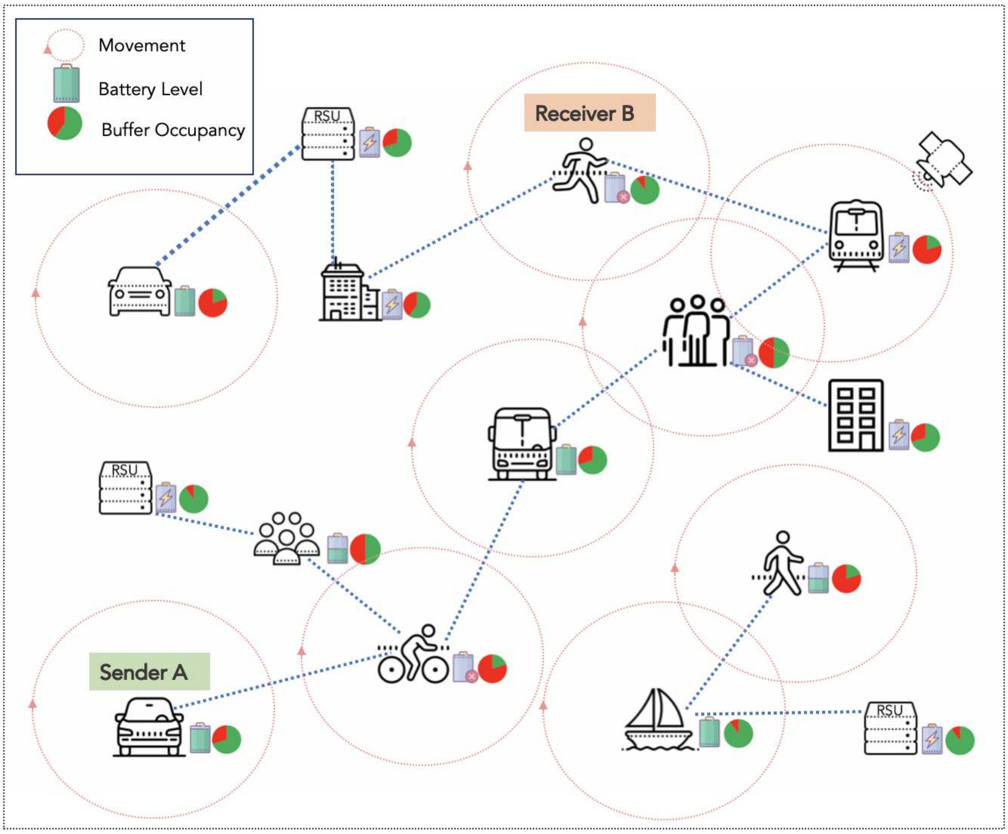


Figure 3.3: An example of OppNets.

contacts. Here entire chunks of message are transferred from one storage unit to a storage uit in another node along a path that is expected to reach the destination [9].

3.2.1 Architecture of OppNets

OppNets consist of nodes which can be anything from fixed devices, vehicles, pedestrians and so on. The data is sent from the source to destination using communication links created by opportunistic contacts that can be Wi-Fi, cellular technologies, Bluetooth or satellite links. These nodes can be IoT devices which roam and opportunistically encounter other IoT devices with which they perform data collection, exchanging or dissemination, as well as relay data between these networks enabling connectivity for other disconnected networks. In Fig.3.3 is shown an example of data forwarding in OppNets. In this example sender A, wants to send a message to receiver B. The message will go through many steps before reaching destination. Sender A in a car driving on the streets, will send the message to a cycler passing by in that area. The cycler passing through traffic, will use Bluetooth to forward the message to the bus passing nearby, which will forward it to the pedestrian. The latter will forward the message to the tram whenever there is a contact opportunity, which means a node has come in its range of communication. Now the message is carried by the other node and the same process is repeated until the destination has been reached. There can be one or many links between sender A and destination B and, links may be disrupt and the topology of the network can change. It can also be seen from the figure that at a given time, not all nodes have the same resources. Some have more storage than others, whilst some have better battery levels. If a node becomes a dead node due to its battery running low, it will store the message until it is activated again introducing delay [10].

3.2.2 OppNet Protocols

OppNets different from traditional networks, have asymmetrical data rates, limited transmission range and are characterized by intermittent connectivity. Therefore, network partitions happen all the time and to solve these problems OppNets uses Store-Carry-Forward mechanism where messages are stored in the nodes' buffer until forwarded to immediate nodes to reach the destination. The topology of the network consistently changes with links coming up and down due to node mobility. They have evolved from ad hoc networks but traditional ad hoc protocols do not work well as they require a fully connected end-to-end path, so other routing protocols must be used. Routing consists of forward decisions made based on predictions of future connectivities and node mobility information.

OppNets being a variant of DTN are challenged networks with different devices and routing environments. Since the conventional TCP model is not applicable in these networks because of the absence of an end-to-end path, a "bundle layer" is introduced between the transport and the application layer. This new layer provides an end-to-end data transfer for heterogeneous networks by allowing bundle protocols such as (Spray and Wait, Epidemic, MaxProp, PRoPHET) to interface with different transport protocols. In Fig.3.4 is shown the protocol architecture of DTN. As shown in Fig.3.3, the packets generated by the source node will go through different heterogeneous networks before reaching the destination. The transfer of these packets called "bundles" is provided by the bundle layer protocol through store-carry-forward mechanism. Different transport protocols can be used in different network segments [11, 12].

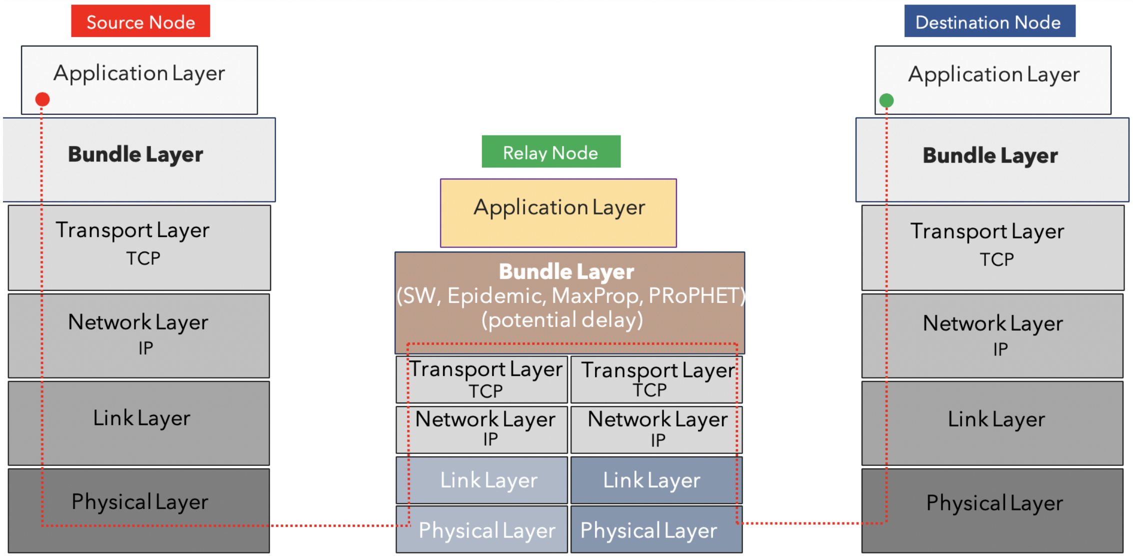


Figure 3.4: DTN Protocol Stack.

- ***Flooding based routing protocols:*** They spread the messages and their copies in the network.

  Epidemic Routing: Epidemic routing brings the concept of flooding in intermittent connected networks. Each node maintains a list of the all the messages which are waiting to be delivered. When it comes in contact with another node, they will exchange all the messages they do not have in common, making all nodes aware of the destination. Even though the maximum delivery probability is reached by spreading messages all over the network which eventually reach the destinations, it creates a lot of congestion and a large overhead. Furthermore, due to limited resources nodes tend to have, the messages may be dropped and/or retransmitted. Another drawback of epidemic routing is that nodes continue to propagate messages even if they have been successfully delivered to destination.

  Spray and Wait: The Spray and Wait protocol is an improvement of Epidemic Routing which limits the message forwarding by reducing the messages exchanged. The routing process consists of two phases: spray phase and wait phase. During the spray phase, one node will generate n copies of packets which will spread to the first nodes it encounters. After the spray phase, it goes to the second phase which is the wait phase where it waits for a confirmation that the message has been received by the destination. Once the data is delivered the destination will send an acknowledgment using the same two phases.

- ***Forward based routing protocols:*** These types of protocols use a more efficient mechanism to select relay nodes in order to increase the delivery probability, but lowering the overhead and limiting the resource usage.

  PRoPHET (Probabilistic Routing Protocol using history of Encounters and Transitivity): The PRoPHET protocol uses node's mobility pattern to predict the possibility that certain node will visit the location again, based on past history. In PRoPHET, a message is relayed to a contact node only if the delivery probability to destination node of the contact node, is higher than that of the transmitting node. By doing so, delivery probability is increased. However, since a node has to buffer the message until the destination delivery probability is met, much longer delays will be introduced and nodes may have insufficient buffer sizes.

  MaxProp: MaxProp protocol increases the delivery rate by using mechanisms that prevent retransmission and deletion of duplicate packets. When one node comes in contact with another one, it will not forward all the messages, but only the one the contact does not have. Each node will keep a list of all the stored packets ranked based on the cost which indicates the possibility of reaching the destination. MaxProp will assign priority values to packets and prevent storing the same packet twice. When a packet reaches the destination, an acknowledgement message will inform the other nodes to drop the delivered packets they are still holding.

3.2.3 OppNets Challenges

In this subsection we present the challenges in the use and development of OppNets:

Storage limitations: Nodes must have enough storage to store all messages for an unspecified period of time until there is a contact opportunity to transfer the messages. To deal with the storage limitations, buffer management and replication strategies must be considered. In OppNets, individual devices have variable capacities and the amount of traffic generated is not predictable. Therefore, devices with limited storage capabilities can severely affect QoS as transmission uses store-carry-forward mechanism.

Contact Opportunity: OppNets are networks which are formed when nodes come in contact with each other through physical proximity and share their services and resources by different communication media. However, due to node mobility and the dynamic wireless channels, a node can make contacts with other nodes at an unprecedented time. Since contacts between nodes are unpredictable, with every contact opportunity nodes try to relay messages so they reach destination.

Cooperation Level: It may be required that some nodes provide their own resources (buffer, battery, bandwidth) for other nodes to use without getting compensation. Differently from other traditional wireless networks, nodes are required to store the message in its own buffer for the other nodes, waisting both memory and battery resources. Some nodes don't cooperate with other nodes and don't participate in the routing process. Depending on the level of cooperation, nodes are categorized based on the reputation. Some nodes show selfish behaviors and stop forwarding data reducing packet delivery ratio or the messages might not get delivered at all.

Intermittent Connectivity: In a network of nodes with high mobility, frequent and lengthy disruptions mean that a path from source to destination hardly exists. The inconsistent connectivities between nodes due to mobility, is called intermittent connectivity. In such scenarios, links between two nodes are unpredictable and intermittent.

Energy: One of the main challenges of OppNets is energy consumption. Having a dynamic and time varying network topology, messages need to be replicated and relayed to nodes in their range. Continuously having to detect the environment for discovering nodes causes an energy consumption. What makes it even more challenging is that since OppNets are mostly deployed in challenging areas, there are restrictions with the battery recharging and replacing. Energy is essential in maximizing the network lifetime.

Security and Privacy: Always changing topology of the network presents many security and privacy issues with OppNets. Therefore is important to ensure node authentication, privacy protection, data integrity and confidentiality [13].

3.2.4 OppNets Applications

Different from traditional networks where nodes are all deployed together at the same time, in OppNets nodes constantly become part of the network continuously.

Emergency Scenarios: The network starts as a seed network which consists of nodes deployed in the initial phase of the network. If an event that needs an emergency response happens, the seed network will expand by adding helpers to the seed network. A seed network grows, when a node discovers potential helpers by a lookup in the directory or by scanning the spectrum for helpers. These helpers are invited or forced to join the network, based on the emergency of the situation. A helper node is required to assist and must offer its resources if it is within range of a sensitive event [14]. After an event is detected and the seed network is formed, new potential helpers are admitted and added to the network, such as ambulance, firefighter, police car, infrastructure sensors nearby.

Inter-Planetary Networks (IPNs): Another application of OppNets is establishment of a communication between Earth and satellites or other planets. In these environments the communication exhibits frequent disconnections so OppNets can be applied.

Opportunistic Vehicular Networking: The high mobility of vehicles causes short contacts between vehicles limiting the data transmitted making them a version of OppNets. Vehicles collect data from their own sensors, sensors installed in the city and distribute data to other vehicles

Chapter 4

Intelligent Algorithms

4.1 Introduction

With the huge amount of data generated, collected, processed and stored, Intelligent Algorithms (IA) have become prominent in order to retrieve, manipulate and interpret this data. IA, are designed to make decisions based on a variety of data, act on these data and make decisions, since classical logic is very limited in modeling all human reasoning.

So far, probability has been the only uncertainty with which mathematics has worked, but recently the uniqueness of probability theory as a model for capturing uncertainty and vagueness has been questioned. The uncertainty of probability generally relates to the occurrence of phenomena, as symbolized by the concept of randomness. Randomness and fuzziness differ in nature from probability being different aspects of uncertainty. The uncertainty lies in the meaning of the words, and since it is an essential characteristic of the words, it always follows them around to some extend.

Many attempts have been made, especially in this century, for augmenting the representational capabilities of logic, or for proposing non-additive models of uncertainty. One of the most radical and fruitful of these attempts was initiated by Prof. Lotfi Zadeh in 1965 with publication of his paper ”Fuzzy Sets” [15, 16, 17]. Fuzzy set theory has become accepted in the literature as a tool for dealing with certain forms of imprecision that frequently occur in decision making environments, but for which probability calculus is inadequate. Fuzzy theory use linguistic variables to describe the control parameters. By using relatively simple linguistic expressions is possible to describe and grasp very complex problems. A very important property of the linguistic variables is the capability of describing imprecise parameters.

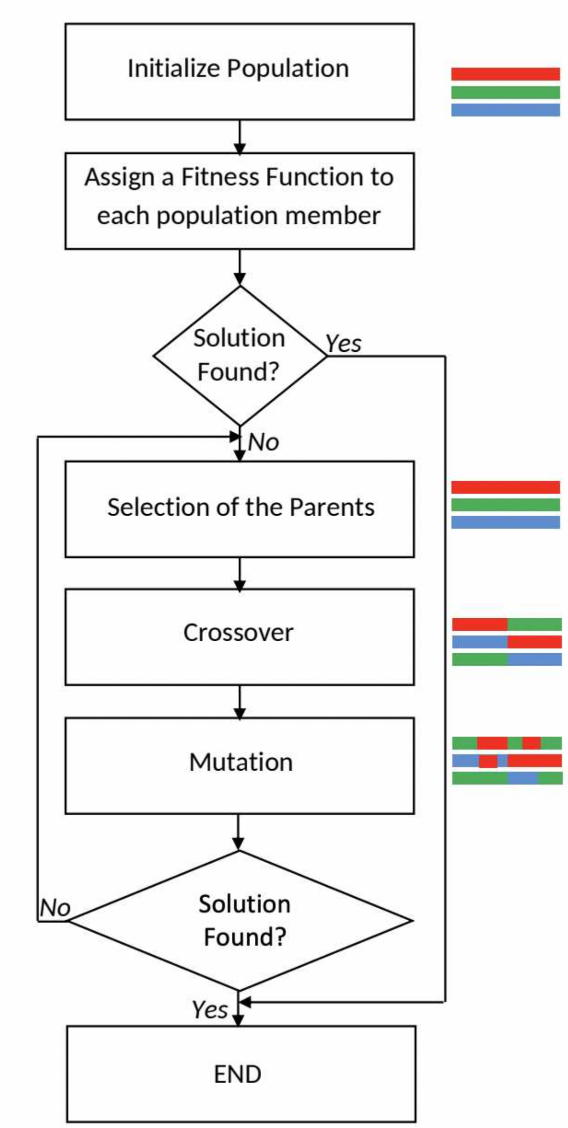


Figure 4.1: Genetic Algorithm Flowchart.

This chapter is about IA and the most commonly used IA. We will first introduce GA and its main concepts. Next we will give the basics of TS, SA, PSO.

4.2 Genetic Algorithm

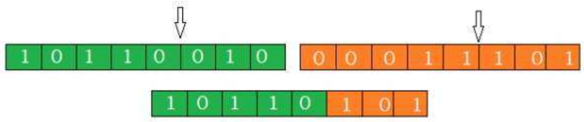
The Genetic Algorithms (GAs) are a meta-heuristic paradigm that can be implemented and applied in various problems including unconstrained and constrained optimization problems, nonlinear and stochastic programming and also node placement methods. GAs are a growing area of artificial intelligence and are inspired by Darwin's theory of biological evolution. Based on these evolutions, GAs can solve optimization problems. The heuristic search of GAs is based on Holland's scheme theorem. Genetic Algorithms, simulated annealing and evolutionary strategies are mainly used for probabilistic search mechanism directed toward decreasing cost or increasing payoff. GAs generate a sequence of populations by using a selection method and use crossover and mutation as search methods. One main difference between GAs and evolutionary strategies is that GAs uses crossover as a probabilistic mechanism and of useful data exchange to locate better solutions.

1. *Selection:* As selection operator, we use roulette-wheel selection [18, 19, 20]. In roulette-wheel selection, each individual in the population is assigned a roulette wheel slot sized in proportion to its fitness. That is, in the biased roulette wheel, good solutions have a larger slot size than the less fit solutions. The roulette wheel can obtain a reproduction candidate.

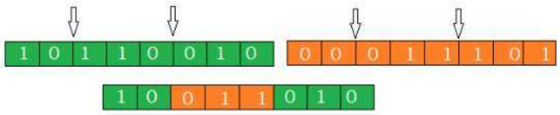
2. *Crossover:* The crossover operators are the most important ingredient of GAs. Indeed, by selecting individuals from the parental generation and interchanging their genes, new individuals (descendants) are obtained. The aim is to obtain descendants of better quality that will feed the next generation and enable the search to explore new regions of solution space not explored yet [21]. There exist many types of crossover operators explored in the evolutionary computing literature. Two of the most common types of crossover are shown in Fig. 4.2. It is very important to stress that crossover operators depend on the chromosome representation.

3. *Mutation:* Mutation operator is one of the GA ingredients. Unlike crossover operators, which achieve to transmit genetic information from parents to off-springs, mutation operators usually make some small local perturbation of the individuals, having thus less impact on newly generated individuals (see Fig. 4.3). Crossover is "a must" operator in GA and is usually applied with high probability, while mutation operators when implemented are applied with small probability. The rationale is that a large mutation rate would make the GA search to resemble a random search. Due to this, mutation operator is usually considered as a secondary operator.

GA is one of the most powerful heuristics for solving optimization problems that is based on natural selection. The GA repeatedly modifies a population of individual



(a) Single Point Crossover.



(b) Multi-point Crossover

Figure 4.2: Two of the most common types of crossovers.

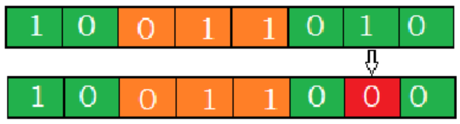


Figure 4.3: Chromosome before and after mutation.

solutions as shown in Fig. 4.1. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population "evolves" towards an optimal solution [22]. In Alg. 1 is shown the pseudo-code for a GA.

In our previous work, we have used GA for placement problems [23]. In IoT networks, when devices/nodes are deployed densely, there is a possibility that a node may reside in the coverage area of multiple different nodes. The goal is to find an optimal and robust topology for the network nodes that brings connectivity services to events.

4.3 Tabu Search (TS)

Tabu Search (TS) is a meta-heuristic method that provides optimal or very close optimal solutions to many combinatorial problems. Heuristic techniques have been used for NP-hard problems where it is very difficult to find an exact solution. The search methodology

Algorithm 1 Genetic Algorithm Pseudocode

t $\leftarrow$ **0**;

Initialize population $P(0)$

Assign a fitness function to each population member $P(0) of size\ \Theta$

while max_nr_of_generations_reached **do**

  Select the parent pool $P_p(t)\ of\ size\ \Phi$;

  Crossover pairs from parents pool $P_p(t) with\ probability\ p_c$; $P_c(t) = Crossover(P_p(t))$;

  Mutate individuals in $P_c(t)\ with\ probability\ p_m$; $P_m(t) = Mutate(P_c(t))$;

  Create new population with individuals from crossover and mutation; $P(t+1) = Individuals(P_c(t) \bigcup P_m(t))$;

  $t \leftarrow t+1$

end while

of TS is similar to neighborhood search. It starts from one point (solution) to another point until a termination criterion is satisfied. Search space of TS is the space of all the solutions that are considered during the search. Search space together with the neighborhood structure are the two basic elements of TS.

Each point starts as an initial solution in a search space, which by applying a series of local modifications called moves, improves to a solution which differs moderately from the previous one. The quality of solutions and computational time, depends on the complexity of the moves at each iteration. Whenever a local optimum has been reached, Local Search (LS) technique is applied which does not allow non-improving moves. This way going back to previously visited solutions is not allowed by the use of memories which are called *tabu*.

The final solution is called local optimum since it is better than the other solutions in the neighborhood but in most cases it will not be a global optimum [24]. While tabus are important as they prohibit moves that go back to non-improving solutions, they may also negatively affect the searching process.

When the *tabu* of a certain move is **0**, than the move can be accepted again. In Fig. 4.4 it can be seen that the algorithm can not go back to the local optimum so it has to search other regions in the search space. *Tabu* moves are saved in a *tabu list*, where for each iteration each *tabu* is decremented by one. There are cases where *tabu* moves are allowed. For example, when a *tabu* move allows a new global solution.

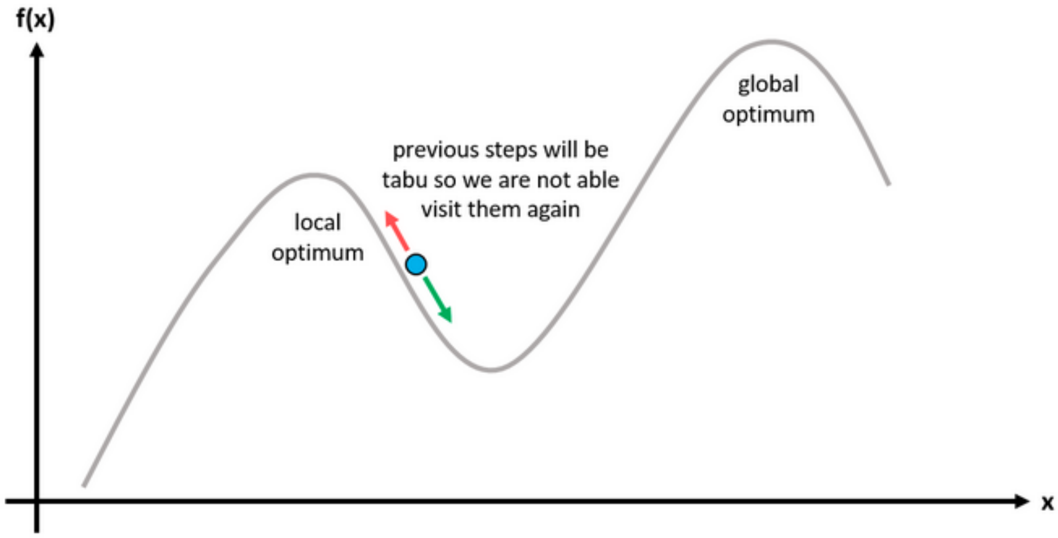


Figure 4.4: Illustration of a tabu move.

4.4 Simulated Annealing (SA)

Simulated Annealing (SA) is a probabilistic method that finds the global minimum of a function that may have several local minimum. It takes the name from the physical process where an object is heated to a temperature where atoms rearranged and than cooled down slowly until the object freezes into a regular structure. SA, uses the idea of annealing to find low cost solutions to combinatorial optimization problems. Many problems, as they become larger require many steps to reach a potential solution. Heuristic methods for optimization problems are important as they offer a solution in reasonable time, but they do not always guarantee the optimum one.

The current literature indicates that the use of simulated annealing algorithms broadens the solution space and results in a higher probability, though not guaranteed, of determination a global optimum rather than a local optimum. The traditional search methods rely on an iterative descent approach that performs well if the objective function has a convex continuous shaped function. In practice, SA algorithms yield a polynomial time solution to an exponential time problem.

One feature of SA, is that it does not get trapped at local minimum. The algorithm employs a random search, which not only accepts changes that decrease objective function, but also some changes that increase it. The latter are accepted with a probability.

The implementation of the SA algorithm depends on this annealing process structure and the process requires the following elements:

- a representation of possible solutions,
- a generator of random changes in solutions,
- a means of evaluating the problem functions.

Another significant component of an SA code is the random number generator, which is used both for generating random changes in the control variables and for the temperature dependent increase acceptance test [25]. SA algorithm always accepts a better solution based on the objective, but it also reduces the likelihood of the solution being trapped, by accepting a worse solution if an acceptance criterion value is greater than a selected random number.

4.4.1 Local Versus Global Search

For many problems, to find a solution that satisfies all the constraints, a large number of possible solutions must be searched. For example, consider the problem of composing a classroom schedule with constraints such as time constraints of students and lecturers and the availability of classrooms. Even for a problem with a relatively small number of constraints, it may be necessary to search through many possible schedules to find the one that satisfies all the constraints [26].

For certain cases, there have been found algorithms that solve large computational problems without having to search through the space of all possible solutions. However, sometimes such a large search cannot be avoided. The search space is referred as a combinatorial search where the best way to search it, depends on the problem to be solved. Inability to detect the unsolvability of a problem instance, is one of the main drawbacks of local search. When dealing with optimization problems it is difficult to determine whether the solution found is globally optimal. In Fig. 4.5 is shown the graph for the local and global minimum of SA method. In contrast with the local search methods, with the global search techniques, we are able to tell if no solution exists after all the search space has been explored and no solution is found.

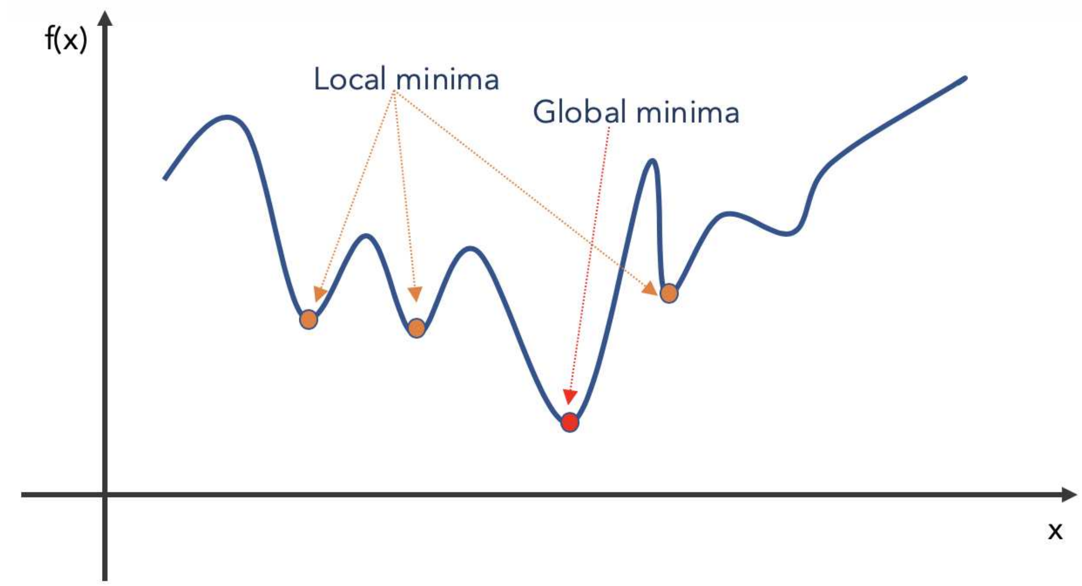


Figure 4.5: Local and global minimum for SA.

4.5 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is another heuristic optimization method based on swarm intelligence. It was introduced in 1995 by Kennedy and Eberhart and it is very popular due to its optimization performance. In PSO particles use their past experiences to find the optimum solution based on the experience of the swarm. PSO algorithm simulates the social behavior of birds in a flock which fly in synchrony with each other and regroup if suddenly change direction.

When searching for food, if the birds are scattered, they have a smaller chance of finding food than when together as a flock. When in a flock, one bird is always closer to the food and transmits this information to the other birds. Birds, or particles fly through out the search space with the aim of finding food or finding the optimum solution. To explain it in analogy with evolutionary paradigms, particle represents an individual in a population which is represented as a swarm. In Alg. 2 is shown the pseudo-code for PSO. Changes of the particle's position within this search space are influenced by the information or experience of its neighbors. In a population $P(0)$, each particle i has an

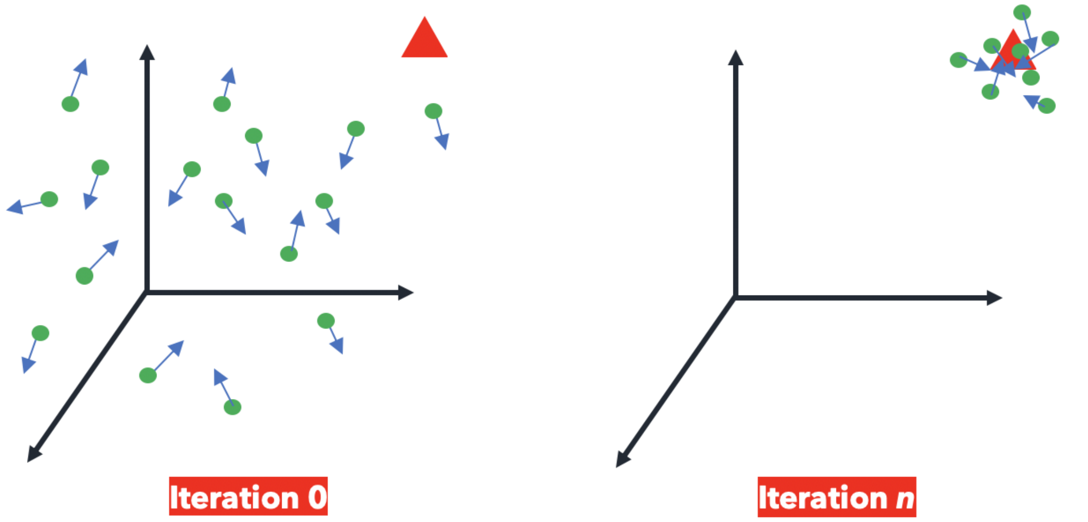


Figure 4.6: PSO particle movement.

initial position denoted as $x^i(k)$ at a discrete time step k. If a velocity $v^i(k+1)$ is added to a particle, its position will change to another position $x^i(k+1)$, i.e.

$$x^i(k+1) = x^i(k) + v^i(k+1) \tag{4.1}$$

The position of each particle changes based on the best position this individual particle has during its movement and its best position related to swarm position as shown in Fig. 4.6. In a population $P(0)$ of particles *i=1,.....,n*â, each particle will keep in its memory the best position of the search space for each iteration. The speed of the particles gets updated at each iteration. In Eq. 4.2 is shown the velocity of the particle i, at time step k.

$$v^i(k+1) = v^i(k) + c_1 r_1 (y^i(k) - x^i(k)) + c_2 r_2 (y^j(k) - x^i(k)) \tag{4.2}$$

Where:

- $x^i(k)$ - particle's position at time step k.
- $v^i(k)$ - particle's velocity.
- $v^i(k+1)$ - updated particle's velocity.
- c_1, c_2 - acceleration constants that represent cognitive and social components.
- r_1, r_2 - random numbers in the range [0, 1] for having a uniform distribution.

- $y^i(k)$ - particle's best individual position.
- $y^j(k)$ - particle's best swarm position.

Velocity of the particle represents the experiential and social information exchanged form particles' neighbor.

Algorithm 2 PSO Pseudocode

k $\leftarrow$ **0**;
Initialize: population $P(0)$
for each particle *i=1,....n* **do**
Initialize: particle's initial position $x^i(0)$
Calculate fitness function of each particle $f(x_i)$
if $f^i(x) <= f^i(y)$ *compare current fitness value with the best fitness value* **then then**
Set value $f(y^i)$ as the best
$y^i = x^i$ *global best position*
end if
end for
for each particle *i=1,....n* **do**
Eq. 4.1 $\leftarrow x^i(k+1)$;
Eq. 4.2 $\leftarrow v^i(k+1)$;
end for
Until Check if termination criteria is reached.

Chapter 5

Fuzzy Logic

5.1 Introduction

In this chapter we show how the fuzzy control systems can be used for real-world applications. These systems use heuristic logic and rules as opposed to conventional control approaches such as Proportional Integral Derivative (PID) where differential equations are used. Fuzzy Logic (FL) is the logic underlying modes of reasoning which are approximate rather then exact. The importance of FL derives from the fact that most modes of human reasoning and especially common sense reasoning are approximate in nature.

The essential characteristics of FL relate to the following.

- In FL, exact reasoning is viewed as a limiting case of approximate reasoning.
- In FL everything is a matter of degree.
- Any logic system can be fuzzified.
- In FL, knowledge is interpreted as a collection of elastic or, equivalently, fuzzy constraints on a collection of variables.
- Inference is viewed as a process of propagation of elastic constrains.

In a broad sense, FL is almost synonymous with fuzzy set theory. Fuzzy set theory, as its name suggests, is basically a theory of classes with unsharp boundaries. Fuzzy set theory is much broader than FL and contains the latter as one of its branches [27, 28, 29]. Among the other branches of fuzzy set theory are e.g., fuzzy arithmetic, fuzzy mathematical programming, fuzzy topology, fuzzy graph theory, and fuzzy data analysis.

What is important to recognize is that any crisp theory can be fuzzified by generalizing the concept of a set within that theory to the concept of a fuzzy set.

5.2 Fuzzy Logic Controller

Modeling and simulating a complex real world system is a challenging task. Fuzzy Control (FC) gives us a methodology for representing, manipulating and implementing heuristic knowledge to control a system in order to satisfy the necessary assumptions. FC consists of four components:

1. Fuzzification;
2. Inference Engine;
3. Rule-base;
4. Defuzzification;

Fuzzy logic can model nonlinear functions to a good degree of accuracy and map input and output data of systems. FC can be viewed as an artificial decision making system that operates by taking input data and then ensuring that the objectives are met. In fact, any kind of control law can be modeled by the FC methodology, provided that this law is expressible in terms of "if ... then ..." rules, just like in the case of expert systems. However, FL diverges from the standard expert system approach by providing an interpolation mechanism from several rules. In the contents of complex processes, it may turn out to be more practical to get knowledge from an expert operator than to calculate an optimal control, due to modeling costs or because a model is out of reach [30].

5.3 Fuzzification

Fuzzification is the process of conversing numerical input variables into fuzzy sets so they can be used by the inference engine. In real world, different hardwares such as devices and sensors, generate crisp data which are subject to a range of errors. Many of the quantities that we consider crisp, carry some uncertainty and fuzzification is widely used to handle imprecision of measurement. In a domain of numerical inputs $x_i \in X_i$, each numerical input x_i transforms into a fuzzy set A_i^{fuzz} which is defined on the universe of all possible fuzzy sets X_i^{fuzz}. The fuzzification $\mathscr{F}$ process begins with choosing the most suitable membership functions where each input variable has its own Membership Function (MF), where:

$$\mathscr{F}(x_i) = A_i^{fuzz} \tag{5.1}$$

In other words, input MFs, associate each numerical element with a number in the interval [0,1] as shown in Eq. 5.2.

$$\mu\ x_i :\in X_i \rightarrow [0,1] \tag{5.2}$$

There are two types of fuzzification methods: singelton and non-singleton fuzzifier. Singleton is the most used fuzzifier type because it is simpler and has lower computational requirements. Non-singleton fuzzifiers are used more successfully when noise is present in data processed by the system. The MFs for singleton and non-singleton fuzzifiers are defined in Eq. 5.3 and Eq. 5.4, respectively.

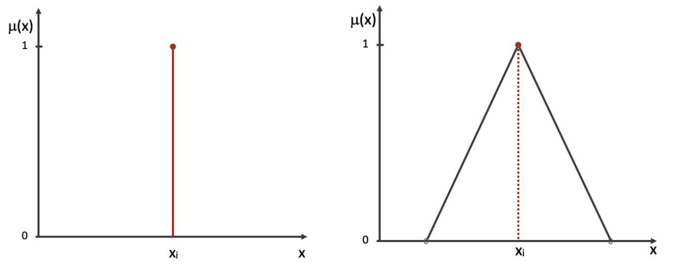
$$\mu\ (x) = \begin{cases} 1 & \text{if } x = x_i \\ 0 & \text{else } x \neq x_i \end{cases} \tag{5.3}$$

$$\mu\ (x) = \begin{cases} 1 & \text{if } x = x_i \\ other & \text{else } x \neq x_i \end{cases} \tag{5.4}$$

In Fig.5.1 are shown the pictures of MFs for each fuzzification type. As seen from the picture in the singleton fuzzification in Fig.5.1(a), $\mu\ (x_i)$ takes only the measured value, as there is no presence of noise, unlike the non-singleton fuzzification shown in Fig.5.1(b) where for values different from x_i, $\mu\ (x)$ takes into consideration the presence of noise [31]. Singleton fuzzification is preferred since it is used in practical scenarios, while non-singleton fuzzification adds complexity to the next process which is inference engine [32]. However, the fuzzification used is based on the noise level present and the type of the designed system.

5.3.1 Inputs and Outputs of FC

For any decision making system the main goal is to get the desired output for a given set of inputs. Before designing a control system, first a set of inputs based on the problem must



(a) Singleton Fuzzification (b) Non-Singleton Fuzzification

Figure 5.1: Fuzzification Types.

be identified. Deciding the information that will be used as the input in a decision making process can be difficult since the output is a direct function of the input parameters.

One of the challenges for non linear control systems is sampling input parameters in the big domain of all the possible inputs. For example, for a selection decision making system, input variables vary from each other, but they all affect the result in different degrees. The relative importance of each variable is taken into consideration when designing and implemented the system.

5.3.2 Linguistic Description of Parameters

In this section we have introduced the notion of linguistic description of variables. A linguistic variable is expressed in the natural or artificial language and its purpose is to approximate phenomena that are too complex to be expressed only as a crisp value. Each variable is described in linguistic values with associated degrees of membership. One important step in the process of decision making is the aggregation of information expressed in linguistic variable.

Every linguistic variable is aggregated in linguistic values or term sets. For a linguistic variable to be a useful analysis tool, it must be manipulated through different operations. One way of manipulating linguistic variables is by manipulating the MFs associated with them [33]. There are many ways of describing a linguistic variable in different term sets where each one represents one different level of quantity. However, the choice of linguistic variable or term sets does not affect the FC as it is just a simple way of approximately representing a parameter. Lets assume that we have a linguistic variable $\widehat{x_i}$ which is rep-

(a) Crisp Set (b) Fuzzy Set

Figure 5.2: Crisp and Fuzzy Set.

resented as a term set $\widehat{A}_i^j$ in the domain X_i where $j = 1\ ,\ 2...m$, and $\mu_{A_i^j}(x_i)$ is the MF associated with the fuzzy set A_i^j which maps X_i to [0,1].

Lets assume we have one parameter to which we assign a linguistic variable $x_1 =$ "*age*" and three linguistic values or term sets to this variable: $\widehat{A}_1^1 = young$, $\widehat{A}_1^2 = middleage$, $\widehat{A}_1^3 = old$. A_1^1, A_1^2, A_1^3 are fuzzy sets and their MF describes the degree of certainty that the numeric value of age, has the properties characterized by A_1^j. In this case we have decided to use three term sets per linguistic variable, but different levels of aggregation can be used depending on the problem.

5.4 Fuzzy Sets

Fuzzy sets theory is an extension of the classical theory of crisp logic which is based on two truth values: *true* or *false*. However, using only two values is not sufficient for human reasoning. FL uses the interval between **0** *(false)* and **1** *(true)* to give multiple values to variables. Fuzzy sets concept is first introduced by defining a MF and they heuristically quantify the linguistic variables, values and rules. Each member in a fuzzy set has different degrees of membership in the interval [0,1]. In Fig.5.2 are shown examples of a crisp set and fuzzy set. The input and output must be converted to linguistic variables whose values are words in natural or artificial language, however the original input and output must be crisp variables, but the intermediate process is a fuzzy inference process. Given a linguistic variable $\widehat{x}_i$ and a MF $\mu_{A_i^j}(x_i)$, a fuzzy set is defined as shown in Eq. 5.5.

$$A_i^j \;=\; (x_i,\ \mu_{A_i^j}(x_i)) :\ x_i \in X_i \tag{5.5}$$

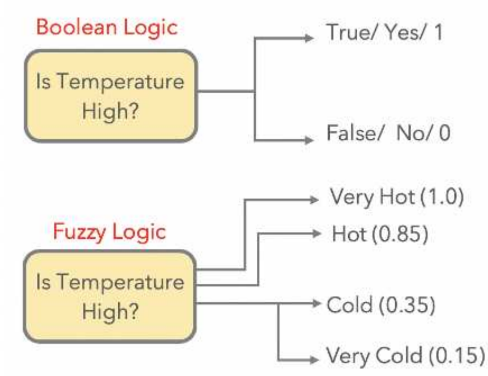
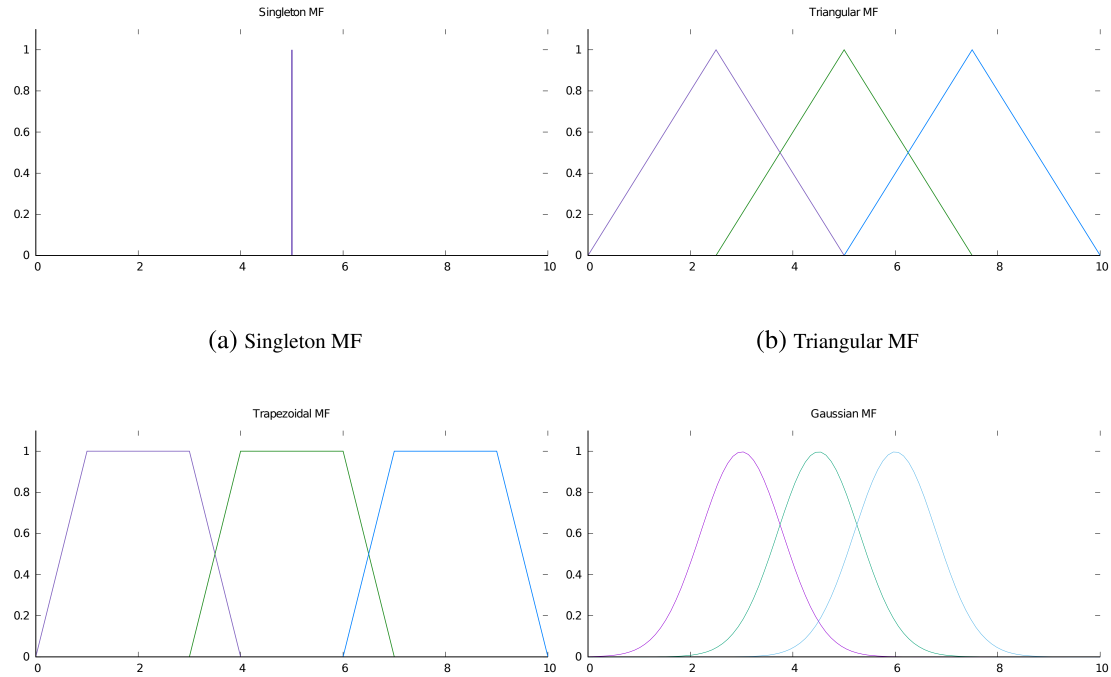


Figure 5.3: Crisp vs. Fuzzy Sets.

Previously, decision making systems were formulated based on the Boolean logic, where crisp values of *0* and *1* were used. However, human brain is not wired to think in "yes or no" logic, but it can be fuzzy, qualitative, uncertain in nature as shown in Fig.5.3. In FL, uncertainty does not refer to the lack of knowledge about the value of a parameter, rather than in the sens of vagueness. In the following are shown the essential notions of the fuzzy set theory. A fuzzy set can contain all the possible outcome from the interval [0,1].

- X_i domain of numerical input;
- x_i $i-th$ numerical input;
- $\widehat{x}_i$ linguistic variable for numerical input;
- A_i^j fuzzy set;
- $\widehat{A}_i^j$ $j-th$ linguistic value for $i-th$ linguistic variable;
- $i,\ j$ n-tuple where $n \geq 1$ and $m-tuple$ where $m \geq 1$;
- $[0,1]$ MF interval;
- $\mathscr{F}$ fuzzification operator;

A fuzzy set is an extension of a crisp set. Crisp sets only allow full membership or no membership at all, whereas fuzzy sets allow partial membership. In other words, an element may partially belong to a set.



(a) Singleton MF (b) Triangular MF

(c) Trapezoidal MF (d) Gaussian MF

Figure 5.4: Types of MF.

5.4.1 Membership Functions

A Membership Function (MF) is graph representation of the degree of participation of each input value to a interval [0,1]. The MF is usually denoted as μ_A and for each value x_i quantifies the degree of belongingness of the element x_i to the fuzzy set. With MF, we show how FL is used to quantify the meaning of each linguistic description so that we automate the control rules specified my the type of application. $\mu_{A_i^j}(x_i)$

In Fig.5.4 are shown the graphical representation on some of the most used types of MF. Triangular and trapezoidal are most commonly used due to their computational efficiency and their simplicity as they are formed with straight lines. The Gaussian MF unlike the other MF, has smooth curves but is not suited for application that require unsymmetrical MF. However, the types of MF are not limited only to the one showed in Fig.5.4, other types of MF can be used depending on the applications.

Lets assume that we have one input variable x_i which changes over time $x_i(t)$. We use the function μ to quantify at what certainty does $x_i(t)$ classify as a specific term set. Below we have shown a case analysis where we show how to interpret MF for different

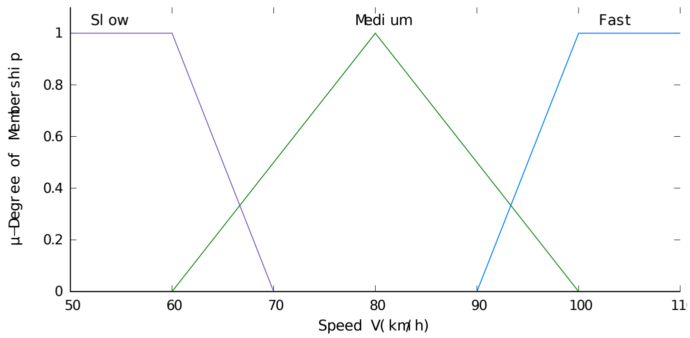


Figure 5.5: Membership Function for different linguistic values.

values of $x_i(t)$. In Fig.5.5 is shown a specific case where we have chosen *speed V(t)* as an input variable.

- If $V(t) = 50$, $\mu(50) = 1$ indicates that we are absolutely certain that $V(t) = 50$ is absolutely "*Slow*".
- If $V(t) = 70$, $\mu(70) = 0.5$ indicates that we are only halfway certain that $V(t) = 70$ "*Medium*". In terms of linguistic interpretation, this value is considered "gray area"

The types of MF used in Fig.5.5 are trapezoidal and triangular, but can be bell shaped or others. Also, very often all the MF for the input or output will be drawn in one graph with labels describing the meaning of their associated linguistic values. In this way we can easily specify the MF for all linguistic values.

5.4.2 FC Rules

FC describes the algorithm for process control, as a fuzzy relation between information about the conditions of the process to be controlled, x and y, and the output for the process z. The relationship between input and output is summarized in a form of rules in the control knowledge base or rule base. There are two main tasks in designing the control knowledge base. First, a set of linguistic variables must be selected which describe the values of the main control parameters of the process. Both the input and output parameters must be linguistically defined in this stage using proper term sets. The selection of the level of granularity of a term set for an input variable or an output variable plays an important role in the smoothness of control. Second, a control knowledge base must be developed which uses the above linguistic description of the input and output parameters.

The control algorithm is given in "if-then" expression, such as:

If x is small and y is big, then z is medium;
If x is big and y is medium, then z is big.

These rules are called *FC rules*. The "if" clause of the rules is called the antecedent and the "then" clause is called consequent. In general, variables x and y are called the input and z the output. "Small" and "big" are linguistic values for x and y, and they are expressed by fuzzy sets.

The rule base of a complex systems has different types of input to output ratio such as:

- Single Input Single Output (SISO)
- Multiple Input Single Output (MISO)
- Multiple Input Multiple Output (MIMO)

More than one FC rule can be fired at one time, because of the partial matching attribute of FC rules and the fact that the preconditions of the rules do overlap. The methodology which is used in deciding what control action should be taken as the result of the firing of several rules can be referred to as the process of "conflict resolution".

5.5 Defuzzification Methods

The defuzzification operation produces a non-FC action that best represent the MF of an inferred FC action. Several defuzification methods have been suggested in literature. Among them, four methods which have been applied most often are described in following [34].

- **Tsukamoto's Defuzzification Method**
  If monotonic MFs are used, then a crisp control action can be calculated by:

$$Z^* = \frac{\sum_{i=1}^{n} \omega_i x_i}{\sum_{i=1}^{n} \omega_i} \tag{5.6}$$

  where n is the number of rules with firing strength (ω_i) greater then 0 and x_i is the amount of control action recommended by rule i.

- **The Center Of Area (COA) Method**
  Assuming that a control action with a pointwise MF μ_C has been produced. The COA method calculates the center of gravity of the distribution for the control action. Assuming a discrete universe of discourse, we have:

$$Z^* = \frac{\sum_{j=1}^{q} z_j \mu_C(z_j)}{\sum_{j=1}^{q} \mu_C(z_j)} \tag{5.7}$$

  where q is the number of quantization levels of the output, z_j is the amount of control output at the quantization level j and $\mu_C(z_j)$ represents its MF value in C.

- **The Mean Of Maximum(MOM) Method**
  The MOM method generates a crisp control action by averaging the support values which their membership values reach the maximum. For a discrete universe of discourse, this is calculated by:

$$Z^* = \sum_{j=1}^{l} \frac{z_j}{l} \tag{5.8}$$

  where l is the number of quantized z values which reach their maximum memberships.

- **Defuzzification when Output of Rules are Function of Their Inputs**
  FC rules may be written as a function of their inputs. For example,

  Rule i: If X is A_i and Y is B_i then Z is $f_i(X,Y)$;

  assuming that α_i is the firing strength of the rule i, then:

$$Z^* = \frac{\sum_{i=1}^{n} \alpha_i f_i(x_i, y_i)}{\sum_{j=1}^{n} \alpha_i}. \tag{5.9}$$

Chapter 6

IoT Device Selection Systems based on Fuzzy Logic

In this chapter, we present four of the proposed fuzzy-based systems.

6.1 Problem Description

Due to high diversity, an IoT network consists of different nodes with different resource capabilities. When multiple IoT nodes are deployed densely, there is a possibility that a node may reside in the coverage area of multiple different nodes. When a specific task request requires an IoT node to complete it, it is challenging to determine which is the best one for that specific request. First, the IoT networks are heterogeneous rather than homogeneous, which consists of many diverse IoT nodes which have largely different demands on data traffic and data processing [35]. To maintain network quality and to have better resource allocation, IoT nodes are selected based on different parameters or based on event coverage. IoT node selection proves useful in mitigating common IoT-related issues like resource allocation, network lifetime, and the confidence in the collected data, by having the right IoT nodes active at a given time. IoT node selection helps in saving and better managing resources by choosing the right subset of nodes to be active depending on the task requirements [36].

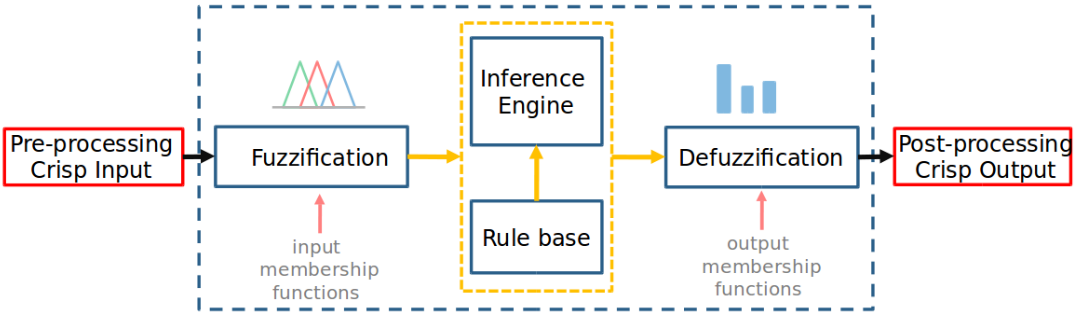


Figure 6.1: Fuzzy Logic Controller

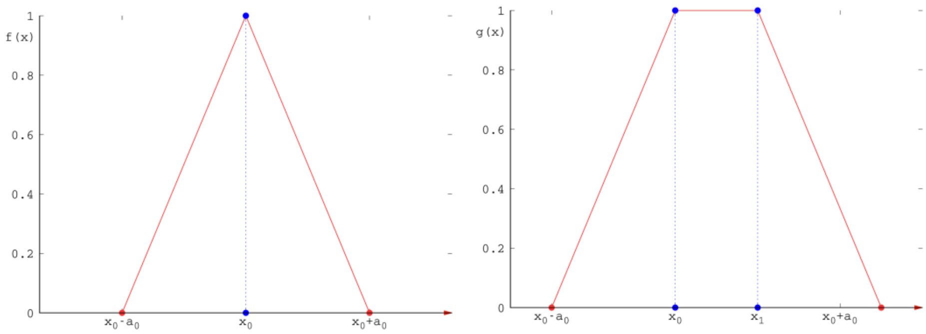


Figure 6.2: Triangular and trapezoidal MF.

6.2 System Parameters

Based on Oppnets characteristics and challenges, we consider the following parameters for implementation of our proposed system.

IoT Device Speed (IDS): There are different types of IoT devices in Oppnets scenarios such as: mobile phone terminals, computers, cars, trains, plains, robots and so on. Considering that high speed IoT devices can transfer the information faster, they will be selected with high probability.

IoT Device Distance from Task (IDDT): when an IoT device is called for action near an event, the distance of the device from the event varies for different scenarios. Depending on three distance levels, our system takes decisions based on the availability of the IoT device node.

IoT Device Remaining Energy (IDRE): IoT devices in Oppnets are active and can perform tasks and exchange data in different ways from each other. Consequently, some IoT devices may have a lot of remaining power and other may have very little, when an event occurs.

IoT Device Storage (IDST): In DTNs data is carried by the IoT device until a communication opportunity is available. Considering different IoT devices have different storage capabilities, the selection desicion is made based on the storage capacity.

IoT Device Waiting Time for sending data (IDWT): Considering network congestion, some IoT devices wait longer and some wait less for sending data. The IoT devices that have been waiting longer have a high possibility to be selected.

IoT Device Security (IDSC): Security measures against an illegal request should be considered. For establishing a secure IoT network, we consider three levels of SC for secure IoT device selection.

IoT Device Node Centrality (IDNC): In Oppnets, finding the most suitable node is challenging. Central nodes can be seen as good candidates to be relay nodes. Centrality is the quantitative measure of the importance of the IoT device, in relation with other IoT devices in the network. The middle node has three advantages over the other nodes: it has more ties, it can reach all the others more quickly, and it controls the flow between the others.

IoT Device Selection Decision (IDSD): The proposed system considers the following levels for IoT device selection:

- Extremely Low Selection Possibility (ELSP) - The IoT device will have an extremely low probability to be selected.
- Very Low Selection Possibility (VLSP) - The IoT device will have very low probability to be selected.
- Low Selection Possibility (LSP) - There might be other IoT devices which can do the job better.
- Medium Selection Possibility (MSP) - The IoT device is ready to be assigned a task, but is not the 'chosen' one.
- High Selection Possibility (HSP) - The IoT device takes responsibility of completing the task.
- Very High Selection Possibility (VHSP) - The IoT device has almost all the required information and potential to be selected and then allocated in an appropriate position to carry out a job.

- Extremely High Selection Possibility (EHSP) - The IoT device has all the required information and the possibility of an IoT device to be selected is extremely high.

The abbrevitaions for the input and output parameters are as follows:

- **Input parameters:**
- IoT Device Speed (IDS)
- IoT Device Distance from Task (IDDT)
- IoT Device Remaining Energy (IDRE)
- IoT Device Storage (IDST)
- IoT Device Waiting Time for sending data (IDWT)
- IoT Device Security (IDSC)
- IoT Device Node Centrality (IDNC)
- **Output parameter:**
- IoT Device Selection Decision (IDSD)

6.3 Systems Implementation

There are several steps for implementing a system based on FL.

Parameter Selection: The selection of parameters depends on the challenges that Opp-Nets face or a particular situation. Choosing the best parameters is a challenging task. Special scenarios favor some parameters to others, making the choosing task more difficult. Furthermore, the number of parameters greatly affects the systems performance and its resources.

Assign linguistic values to each parameter: When choosing linguistic values, we have to choose the granularity of each linguistic term sets. Linguistic values are chosen so they are as short as possible but still accurately represents the input and output parameters. Find MFs: Based on the problem, it is important to select the appropriate input and output MF which can be selected for the list of predefined ones, or be specifically designed.

When implementing a system based on fuzzy, the first issue to address is designing the FC. The FC has four components: 1) The fuzzification process which modifies the input so it can be compared to the rules in the Rule Base (RB). 2) The inference engine which evaluates the control rules and decides what will the output be. 3) Construction of the RB as a set of rules that shows how to control the system. 4) Defuzzifying the outputs of the inference engine to crisp output.

Fuzzy systems are used to represent knowledge that is imprecise. Humans exhibit indecisiveness in their decision making due to continuing variations. Uncertainty in decision making is introduced in fuzzy MF by using a range of membership values associated with input values.

The main part of our selection system is shown in Fig. 6.1. It consists of one Fuzzy Logic Controller (FLC), which is the main part of our system and its basic elements which are the fuzzifier, inference engine, FRB and defuzzifier [37].

In Fig. 6.2, are shown the membership functions we have used for our systems. We have use polygonal type MF such as triangular and trapezoidal as they are less complex, more flexible when splitting different term sets values. A triangular function is defined by a lower limit $x_0 - a_0$, an upper limit $x_0 + a_0$ and a central value x_0, $x_0 - a_0 < x_0 < x_0 + a_0$. While trapezoidal functions are defined by a lower limit $x_0 - a_0$, a lower support limit x_0, an upper limit $x_1 + a_0$ and an upper support limit x_1, $x_0 - a_0 < x_0 < x_1 + a_0 < x_1$. Also the use of triangular and trapezoidal MF enables us to use unsymmetrical MF (see Fig. 6.7(d)). Note that MFs may be different for the same parameters, but for different systems. MF are constructed based on the scenario you wish to implement the system, or the parameter specifics. However, there are parameters which limit the design process. For example, IDSC is a very sensitive parameter, where a device is considered to have a high security mechanism only for values above 90% (see Fig. 6.7(d)).

6.3.1 Description of IDSS1

The implemented system is shown in Fig. 6.3. We use three input parameters for FLC of IDSS1:

- IoT Device Speed (IDS);
- IoT Device Remaining Energy (IDRE);
- IoT Device Distance from Task (IDDT);

Output Parameter:

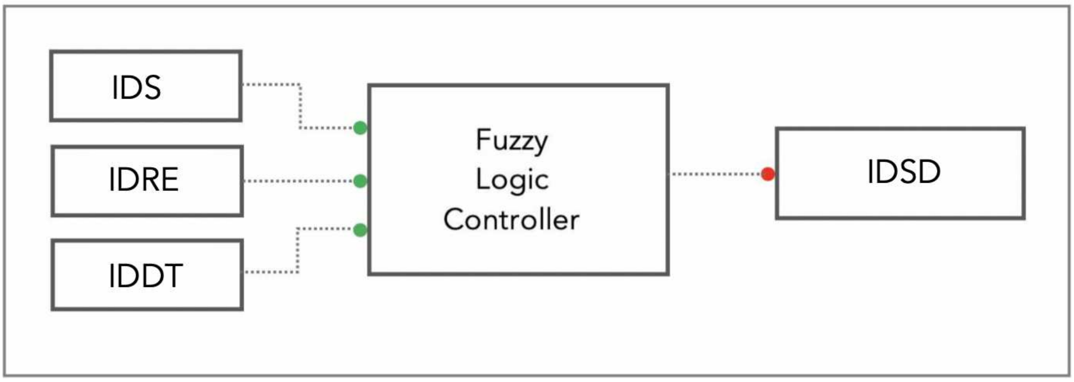


Figure 6.3: Proposed Implemented System IDSS1.

- **IoT Device Selection Decision (IDSD)**;

The term sets for each input linguistic parameter are defined respectively as shown below.

$$\begin{aligned} T(IDS) &= \{Slow(Sl), Medium(Md), Fast(Fa)\} \\ T(IDRE) &= \{Low(Lw), Medium(Mdm), High(Hg)\} \\ T(IDDT) &= \{Near(Ne), Middle(Mi), Far(Fr)\} \end{aligned}$$

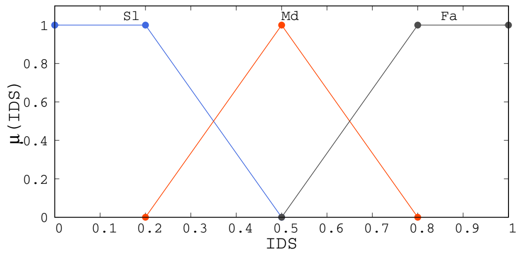
The term sets of IDSD are defined as follows:

$$\begin{aligned} &\{Very\ Low\ Selection\ Possibility\ (VLSP), \\ &Low\ Selection\ Possibility\ (LSP), \\ &Middle\ Selection\ Possibility\ (MSP), \\ &High\ Selection\ Possibility\ (HSP), \\ &Very\ High\ Selection\ Possibility\ (VHSP)\}. \end{aligned}$$

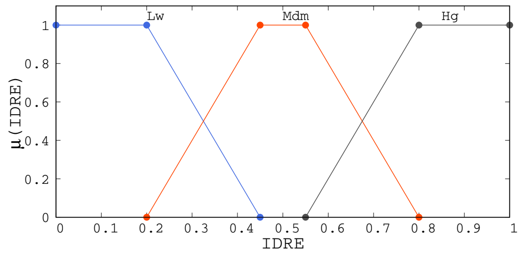
In Fig. 6.4 are shown the MFs of IDSS1 and the FRB of IDSS1 are shown as formulated rules of the parameter space in Table 6.1. The FRB forms a fuzzy set of dimensions $|\mu(IDS)| \times |\mu(IDRE)| \times |\mu(IDDT)|$, where $|\mu(x)|$ is the number of terms on $\mu(x)$. The FRB of IDSS1 has 27 rules which are linguistic IF-THEN conditions that have the form "IF A THEN B", where A and B are propositions with linguistic variables.

6.3.2 Description of IDSS2

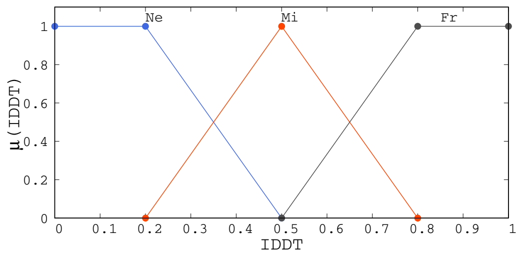
We consider four input parameters for FLC of IDSS2:



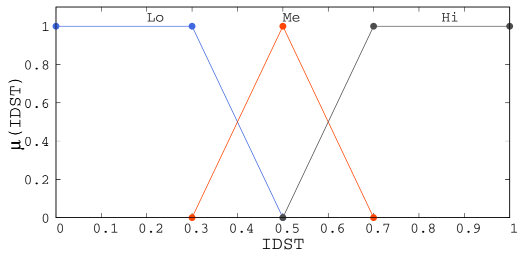
(a) IoT Device Speed



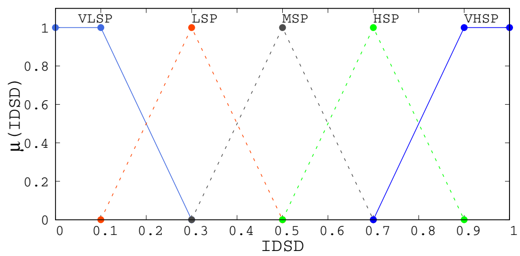
(b) IoT Device Remaining Energy



(c) IoT Device Distance from Task



(d) IoT Device Storage



(e) IoT Device Selection Decision

Figure 6.4: Fuzzy MFs of IDSS1 and IDSS2, (a) IDS (b) IDDT (c) IDRE (d) IDST (e) IDSD.

- IoT Device Speed (IDS);
- IoT Device Remaining Energy (IDRE);
- IoT Device Distance from Task (IDDT);
- ***IoT Device Storage (IDST)***.

Output Parameter:

- **IoT Device Selection Decision (IDSD)**;

Table 6.1: FRB of IDSS1.

No.	IDS	IDRE	IDDT	IDSD
1	Sl	Lw	Ne	LSP
2	Sl	Mdm	Ne	MSP
3	Sl	Hg	Ne	VHSP
4	Sl	Lw	Mi	VLSP
5	Sl	Mdm	Mi	LSP
6	Sl	Hg	Mi	HSP
7	Sl	Lw	Fr	VLSP
8	Sl	Mdm	Fr	VLSP
9	Sl	Hg	Fr	LSP
10	Md	Lw	Ne	MSP
11	Md	Mdm	Ne	HSP
12	Md	Hg	Ne	VHSP
13	Md	Lw	Mi	LSP
14	Md	Mdm	Mi	MSP
15	Md	Hg	Mi	HSP
16	Md	Lw	Fr	VLSP
17	Md	Mdm	Fr	LSP
18	Md	Hg	Fr	MSP
19	Fa	Lw	Ne	HSP
20	Fa	Mdm	Ne	VHSP
21	Fa	Hg	Ne	VHSP
22	Fa	Lw	Mi	MSP
23	Fa	Mdm	Mi	HSP
24	Fa	Hg	Mi	VHSP
25	Fa	Lw	Fr	LSP
26	Fa	Mdm	Fr	MSP
27	Fa	Hg	Fr	HSP

In IDSS2, we have considered storage (IDST) as a new parameter due to its importance in data availability and durability. The success of the network is the ability to achieve an end-to-end data delivery where relay nodes need to have enough storage to keep the data without dropping until it reaches destination. The structure of IDSS2 is shown in Fig. 6.5.

The term sets for each input linguistic parameter are defined respectively as shown below.

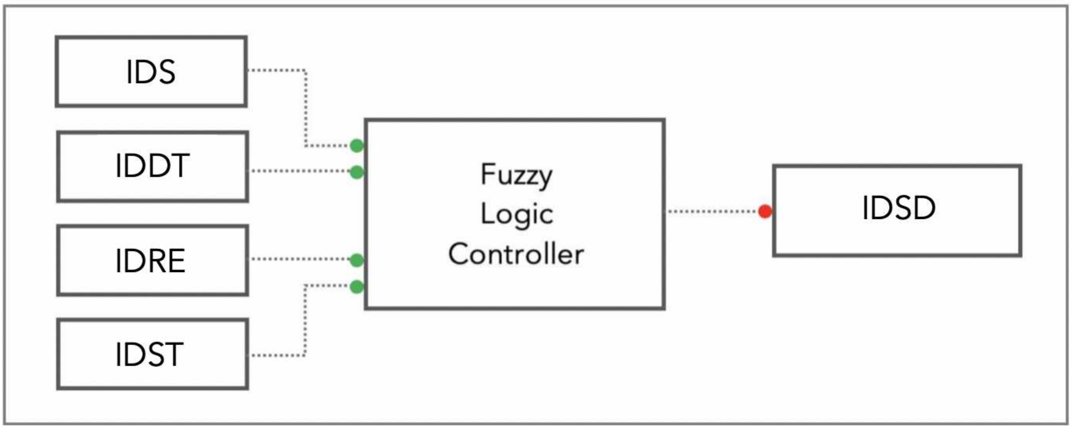


Figure 6.5: Proposed Implemented System IDSS2.

$$
\begin{aligned}
T(IDS) &= \{Slow(Sl), Medium(Md), Fast(Fa)\} \\
T(IDDT) &= \{Near(Ne), Middle(Mi), Far(Fr)\} \\
T(IDRE) &= \{Low(Lw), Medium(Mdm), High(Hg)\} \\
T(IDST) &= \{Low(Lo), Medium(Me), High(Hi)\}
\end{aligned}
$$

We define the term set of IDSD as follows:

$$
\begin{aligned}
\{&Extremely\ Low\ Selection\ Possibility, \\
&Very\ Low\ Selection\ Possibility\ (VLSP), \\
&Low\ Selection\ Possibility\ (LSP), \\
&Middle\ Selection\ Possibility\ (MSP), \\
&High\ Selection\ Possibility\ (HSP), \\
&Very\ High\ Selection\ Possibility\ (VHSP), \\
&Extremely\ High\ Selection\ Possibility\ (EHSP)\}.
\end{aligned}
$$

The MFs are shown in Fig. 6.4 and the FRB of IDSS2 is shown in Table 8.2.

6.3.3 Description of IDSS3

For IDSS3, we decided to keep two of the former IDSS1 and IDSS2 systems parameters (IDST and IDRE) and added two new parameters: IoT Device Waiting Time (IDWT) and IoT Device Security (IDSC).

Table 6.2: FRB of IDSS2.

No.	IDS	IDDT	IDRE	IDST	IDSD	N o.	IDS	IDDT	IDRE	IDST	IDSD
1	Sl	Ne	Lw	Lo	VLSP	41	Md	Mi	Mdm	Me	HSP
2	Sl	Ne	Lw	Me	LSP	42	Md	Mi	Mdm	Hi	MSP
3	Sl	Ne	Lw	Hi	MSP	43	Md	Mi	Hg	Lo	HSP
4	Sl	Ne	Mdm	Lo	LSP	44	Md	Mi	Hg	Me	VHSP
5	Sl	Ne	Mdm	Me	MSP	45	Md	Mi	Hg	Hi	VLSP
6	Sl	Ne	Mdm	Hi	MSP	46	Md	Fr	Lw	Lo	LSP
7	Sl	Ne	Hg	Lo	MSP	47	Md	Fr	Lw	Me	VLSP
8	Sl	Ne	Hg	Me	HSP	48	Md	Fr	Lw	Hi	LSP
9	Sl	Ne	Hg	Hi	VHSP	49	Md	Fr	Mdm	Lo	MSP
10	Sl	Mi	Lw	Lo	VLSP	50	Md	Fr	Mdm	Me	LSP
11	Sl	Mi	Lw	Me	VLSP	51	Md	Fr	Mdm	Hi	MSP
12	Sl	Mi	Lw	Hi	LSP	52	Md	Fr	Hg	Lo	LSP
13	Sl	Mi	Mdm	Lo	VLSP	53	Md	Fr	Hg	Me	MSP
14	Sl	Mi	Mdm	Me	LSP	54	Md	Fr	Hg	Hi	HSP
15	Sl	Mi	Mdm	Hi	MSP	55	Fa	Ne	Lw	Lo	MSP
16	Sl	Mi	Hg	Lo	LSP	56	Fa	Ne	Lw	Me	HSP
17	Sl	Mi	Hg	Me	MSP	57	Fa	Ne	Lw	Hi	VHSP
18	Sl	Mi	Hg	Hi	VHSP	58	Fa	Ne	Mdm	Lo	VHSP
19	Sl	Fr	Lw	Lo	VLSP	59	Fa	Ne	Mdm	Me	VHSP
20	Sl	Fr	Lw	Me	VLSP	60	Fa	Ne	Mdm	Hi	VHSP
21	Sl	Fr	Lw	Hi	VLSP	61	Fa	Ne	Hg	Lo	VHSP
22	Sl	Fr	Mdm	Lo	VLSP	62	Fa	Ne	Hg	Me	VHSP
23	Sl	Fr	Mdm	Me	VLSP	63	Fa	Ne	Hg	Hi	VHSP
24	Sl	Fr	Mdm	Hi	LSP	64	Fa	Mi	Lw	Lo	LSP
25	Sl	Fr	Hg	Lo	VLSP	65	Fa	Mi	Lw	Me	MSP
26	Sl	Fr	Hg	Me	LSP	66	Fa	Mi	Lw	Hi	VHSP
27	Sl	Fr	Hg	Hi	HSP	67	Fa	Mi	Mdm	Lo	HSP
28	Md	Ne	Lw	Lo	LSP	68	Fa	Mi	Mdm	Me	VHSP
29	Md	Ne	Lw	Me	LSP	69	Fa	Mi	Mdm	Hi	VHSP
30	Md	Ne	Lw	Hi	HSP	70	Fa	Mi	Hg	Lo	VHSP
31	Md	Ne	Mdm	Lo	MSP	71	Fa	Mi	Hg	Me	VHSP
32	Md	Ne	Mdm	Me	MSP	72	Fa	Mi	Hg	Hi	VHSP
33	Md	Ne	Mdm	Hi	VHSP	73	Fa	Fr	Lw	Lo	LSP
34	Md	Ne	Hg	Lo	HSP	74	Fa	Fr	Lw	Me	LSP
35	Md	Ne	Hg	Me	VHSP	75	Fa	Fr	Lw	Hi	HSP
36	Md	Ne	Hg	Hi	VHSP	76	Fa	Fr	Mdm	Lo	MSP
37	Md	Mi	Lw	Lo	VLSP	77	Fa	Fr	Mdm	Me	HSP
38	Md	Mi	Lw	Me	MSP	78	Fa	Fr	Mdm	Hi	VHSP
39	Md	Mi	Lw	Hi	LSP	79	Fa	Fr	Hg	Lo	HSP
40	Md	Mi	Mdm	Lo	MSP	80	Fa	Fr	Hg	Me	VHSP
						81	Fa	Fr	Hg	Hi	VHSP

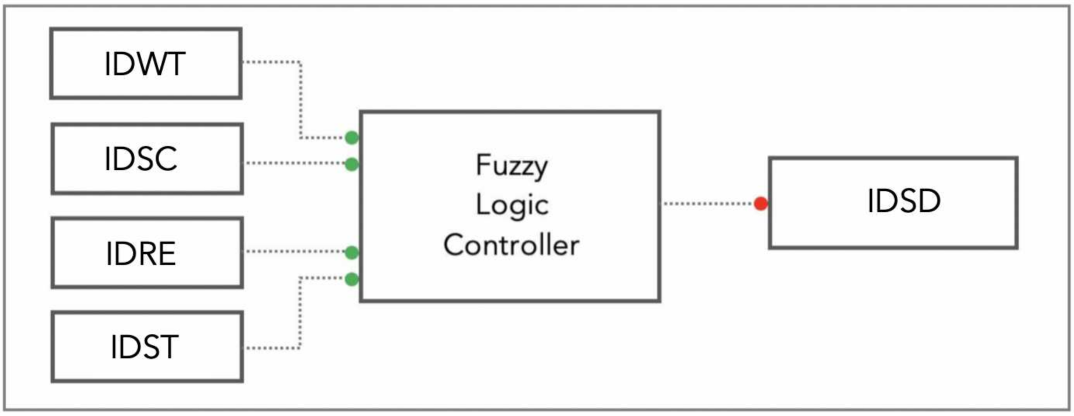


Figure 6.6: Proposed Implemented System IDSS3.

For IDSS3 as shown Fig. 6.6, we have used again four parameters but have added IDSC and IDWT as two new parameters.

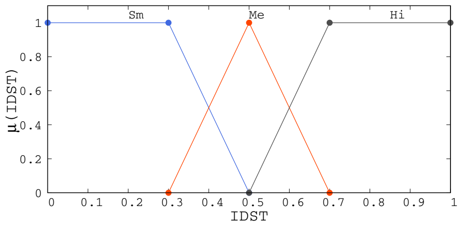
- IoT Device Storage (IDST);
- ***IoT Device Waiting Time (IDWT)***;
- IoT Device Remaining Energy (IDRE);
- ***IoT Device Security (IDSC)***;

Output Parameter:

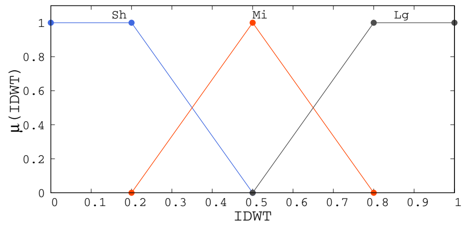
- **IoT Device Selection Decision (IDSD)**;

The term sets for each input linguistic parameter are defined respectively as shown below.

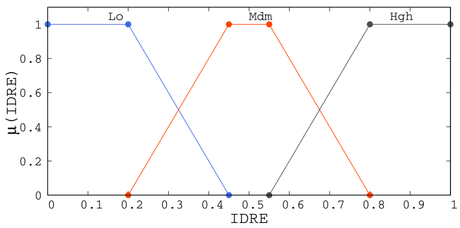
$$\begin{aligned} T(IDWT) &= \{Short(Sh), Medium(Mi), Long(Lg)\} \\ T(IDSC) &= \{Weak(We), Moderate(Mo), Strong(St)\} \\ T(IDRE) &= \{Low(Lo), Medium(Mdm), High(Hgh)\} \\ T(IDST) &= \{Small(Sm), Medium(Me), High(Hi)\} \end{aligned}$$



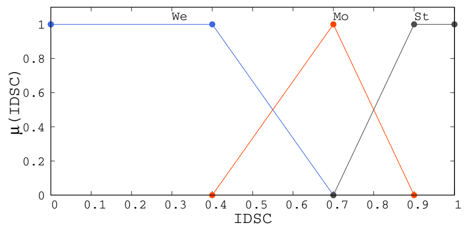
(a) IoT Device Storage



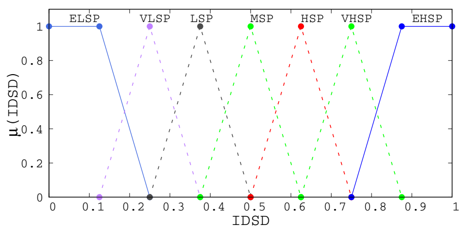
(b) IoT Device Waiting Time



(c) IoT Device Remaining Energy



(d) IoT Device Security



(e) IoT Device Selection Decision

Figure 6.7: Fuzzy MFs of IDSS3.

The MFs for input parameters of FLC are defined as:

$$
\begin{aligned}
\mu_{Sh}(IDWT) &= g(IDWT; Sh_0, Sh_1, Sh_{w0}, Sh_{w1}) \\
\mu_{Mi}(IDWT) &= f(IDWT; Mi_0, Mi_{w0}, Mi_{w1}) \\
\mu_{Lg}(IDWT) &= g(IDWT; Lg_0, Lg_1, Lg_{w0}, Lg_{w1}) \\
\mu_{We}(IDSC) &= g(IDSC; We_0, We_1, We_{w0}, We_{w1}) \\
\mu_{Mo}(IDSC) &= f(IDSC; Mo_0, Mo_{w0}, Mo_{w1}) \\
\mu_{St}(IDSC) &= g(IDSC; St_0, St_1, St_{w0}, St_{w1}) \\
\mu_{Lo}(IDRE) &= g(IDRE; Lo_0, Lo_{w0}, Lo_{w1}) \\
\mu_{Mdm}(IDRE) &= g(IDRE; Mdm_0, Mdm_1, Mdm_{w0}, Mdm_{w1}) \\
\mu_{Hgh}(IDRE) &= g(IDRE; Hgh_0, Hgh_0, Hgh_{w0}, Hgh_{w1}) \\
\mu_{Sm}(IDST) &= g(IDST; Sm_0, Sm_1, Sm_{w0}, Sm_{w1}) \\
\mu_{Me}(IDST) &= f(IDST; Me_0, Me_{w0}, Me_{w1}) \\
\mu_{Hi}(IDST) &= g(IDST; Hi_0, Hi_1, Hi_{w0}, Hi_{w1})
\end{aligned}
$$

The output linguistic parameter is the Actor Selection Decision (IDSD). The term sets of IDSD are defined as follows:

$$\begin{aligned}
&\{Extremely\ Low\ Selection\ Possibility,\\
&Very\ Low\ Selection\ Possibility\ (VLSP),\\
&Low\ Selection\ Possibility\ (LSP),\\
&Middle\ Selection\ Possibility\ (MSP),\\
&High\ Selection\ Possibility\ (HSP),\\
&Very\ High\ Selection\ Possibility\ (VHSP),\\
&Extremely\ High\ Selection\ Possibility\ (EHSP)\}.
\end{aligned}$$

The MFs for the output parameter *IDSD* are defined as:

$$\begin{aligned}
\mu_{ELSP}(IDSD) &= g(IDSD; ELSP_0, ELSP_1, ELSP_{w0}, ELSP_{w1})\\
\mu_{VLSP}(IDSD) &= f(IDSD; VLSP_0, VLSP_{w0}, VLSP_{w1})\\
\mu_{LSP}(IDSD) &= f(IDSD; LSP_0, LSP_{w0}, LSP_{w1})\\
\mu_{MSP}(IDSD) &= f(IDSD; MSP_0, MSP_{w0}, MSP_{w1})\\
\mu_{HSP}(IDSD) &= f(IDSD; HSP_0, HSP_{w0}, HSP_{w1})\\
\mu_{VHSP}(IDSD) &= f(IDSD; VHSP_0, VHSP_{w0}, VHSP_{w1}).\\
\mu_{EHSP}(IDSD) &= g(IDSD; EHSP_0, EHSP_1, EHSP_{w0}, EHSP_{w1}).
\end{aligned}$$

The MFs are shown in Fig. 6.7 and the FRB for IDSS3 are shown in Table 6.3.

6.3.4 Description of IDSS4

For IDSS4 we have considered the importance of one device in the network in terms of its connections and summarized this as a new parameter. The implemented system is shown in Fig. 6.8. The parameters used are as follows:

- ***IoT Device Node Centrality (IDNC)***;
- IoT Device Storage (IDST);
- IoT Device Waiting Time (IDWT);
- IoT Device Remaing Energy (IDRE);

Output Parameter:

Table 6.3: FRB of IDSS3.

No.	IDWT	IDSC	IDRE	IDST	IDSD	No.	IDWT	IDSC	IDRE	IDST	IDSD
1	Sh	We	Lo	Sm	ELSP	41	Mi	Mo	Mdm	Me	LSP
2	Sh	We	Lo	Me	ELSP	42	Mi	Mo	Mdm	Hi	HSP
3	Sh	We	Lo	Hi	VLSP	43	Mi	Mo	Hgh	Sm	MSP
4	Sh	We	Mdm	Sm	ELSP	44	Mi	Mo	Hgh	Me	HSP
5	Sh	We	Mdm	Me	VLSP	45	Mi	Mo	Hgh	Hi	VHSP
6	Sh	We	Mdm	Hi	LSP	46	Mi	St	Lo	Sm	LSP
7	Sh	We	Hgh	Sm	VLSP	47	Mi	St	Lo	Me	MSP
8	Sh	We	Hgh	Me	LSP	48	Mi	St	Lo	Hi	VHSP
9	Sh	We	Hgh	Hi	HSP	49	Mi	St	Mdm	Sm	MSP
10	Sh	Mo	Lo	Sm	ELSP	50	Mi	St	Mdm	Me	VHSP
11	Sh	Mo	Lo	Me	ELSP	51	Mi	St	Mdm	Hi	EHSP
12	Sh	Mo	Lo	Hi	VLSP	52	Mi	St	Hgh	Sm	VHSP
13	Sh	Mo	Mdm	Sm	ELSP	53	Mi	St	Hgh	Me	EHSP
14	Sh	Mo	Mdm	Me	VLSP	54	Mi	St	Hgh	Hi	EHSP
15	Sh	Mo	Mdm	Hi	MSP	55	Lg	We	Lo	Sm	ELSP
16	Sh	Mo	Hgh	Sm	LSP	56	Lg	We	Lo	Me	VLSP
17	Sh	Mo	Hgh	Me	MSP	57	Lg	We	Lo	Hi	LSP
18	Sh	Mo	Hgh	Hi	VHSP	58	Lg	We	Mdm	Sm	VLSP
19	Sh	St	Lo	Sm	VLSP	59	Lg	We	Mdm	Me	MSP
20	Sh	St	Lo	Me	LSP	60	Lg	We	Mdm	Hi	HSP
21	Sh	St	Lo	Hi	HSP	61	Lg	We	Hgh	Sm	MSP
22	Sh	St	Mdm	Sm	LSP	62	Lg	We	Hgh	Me	HSP
23	Sh	St	Mdm	Me	HSP	63	Lg	We	Hgh	Hi	EHSP
24	Sh	St	Mdm	Hi	VHSP	64	Lg	Mo	Lo	Sm	ELSP
25	Sh	St	Hgh	Sm	HSP	65	Lg	Mo	Lo	Me	LSP
26	Sh	St	Hgh	Me	VHSP	66	Lg	Mo	Lo	Hi	MSP
27	Sh	St	Hgh	Hi	EHSP	67	Lg	Mo	Mdm	Sm	LSP
28	Mi	We	Lo	Sm	ELSP	68	Lg	Mo	Mdm	Me	MSP
29	Mi	We	Lo	Me	ELSP	69	Lg	Mo	Mdm	Hi	VHSP
30	Mi	We	Lo	Hi	VLSP	70	Lg	Mo	Hgh	Sm	HSP
31	Mi	We	Mdm	Sm	ELSP	71	Lg	Mo	Hgh	Me	VHSP
32	Mi	We	Mdm	Me	VLSP	72	Lg	Mo	Hgh	Hi	EHSP
33	Mi	We	Mdm	Hi	MSP	73	Lg	St	Lo	Sm	MSP
34	Mi	We	Hgh	Sm	LSP	74	Lg	St	Lo	Me	HSP
35	Mi	We	Hgh	Me	MSP	75	Lg	St	Lo	Hi	VHSP
36	Mi	We	Hgh	Hi	VHSP	76	Lg	St	Mdm	Sm	HSP
37	Mi	Mo	Lo	Sm	ELSP	77	Lg	St	Mdm	Me	EHSP
38	Mi	Mo	Lo	Me	VLSP	78	Lg	St	Mdm	Hi	EHSP
39	Mi	Mo	Lo	Hi	LSP	79	Lg	St	Hgh	Sm	EHSP
40	Mi	Mo	Mdm	Sm	VLSP	80	Lg	St	Hgh	Me	EHSP
						81	Lg	St	Hgh	Hi	EHSP

- **IoT Device Selection Decision (IDSD)**;

The term sets for each input linguistic parameter are defined as follows.

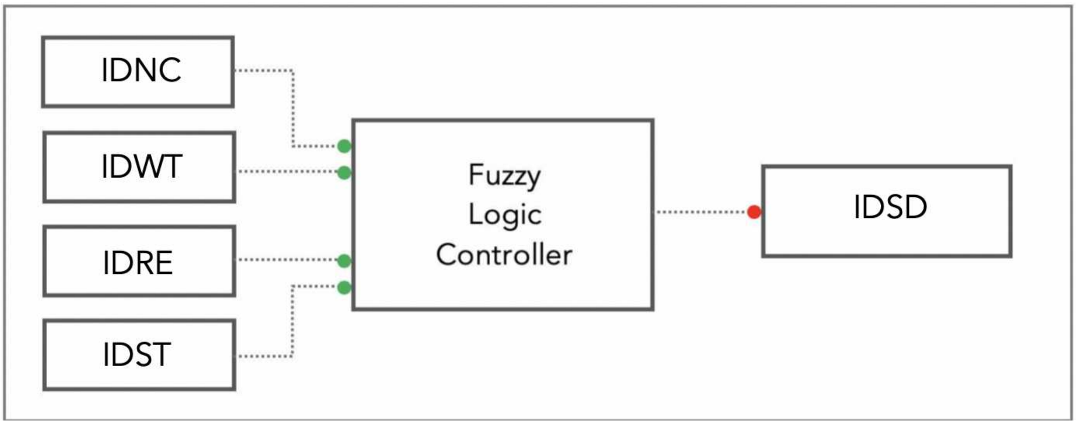
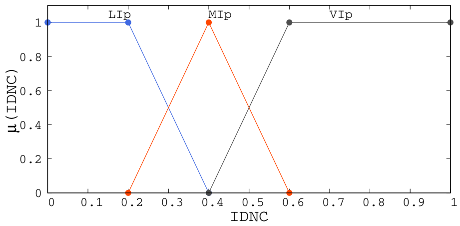
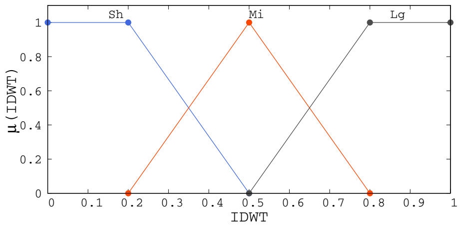


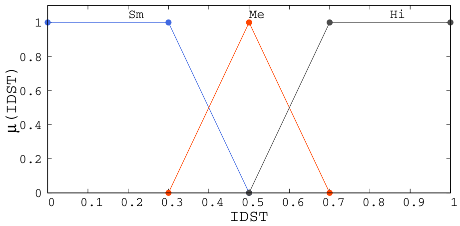
Figure 6.8: Proposed Implemented System IDSS4.



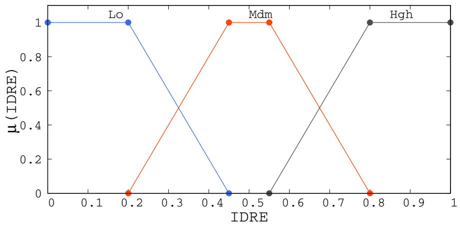
(a) IoT Device Node Centrality



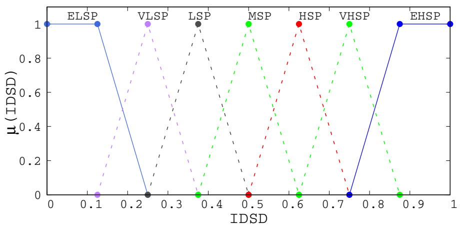
(b) IoT Device Waiting Time



(c) IoT Device Storage



(d) IoT Device Remaining Energy



(e) IoT Device Selection Decision

Figure 6.9: Fuzzy MFs for IDSS4.

Table 6.4: FRB of IDSS4.

No.	IDNC	IDWT	IDRE	IDST	IDSD	No.	IDNC	IDWT	IDRE	IDST	IDSD
1	LIp	Sho	Lo	Sm	ELSP	41	MIp	Mi	Mdm	Me	MSP
2	LIp	Sho	Lo	Me	ELSP	42	MIp	Mi	Mdm	Hi	HSP
3	LIp	Sho	Lo	Hi	ELSP	43	MIp	Mi	Hgh	Sm	MSP
4	LIp	Sho	Mdm	Sm	ELSP	44	MIp	Mi	Hgh	Me	VHSP
5	LIp	Sho	Mdm	Me	ELSP	45	MIp	Mi	Hgh	Hi	EHSP
6	LIp	Sho	Mdm	Hi	LSP	46	MIp	Lg	Lo	Sm	ELSP
7	LIp	Sho	Hgh	Sm	VLSP	47	MIp	Lg	Lo	Me	MSP
8	LIp	Sho	Hgh	Me	LSP	48	MIp	Lg	Lo	Hi	VHSP
9	LIp	Sho	Hgh	Hi	HSP	49	MIp	Lg	Mdm	Sm	MSP
10	LIp	Mi	Lo	Sm	ELSP	50	MIp	Lg	Mdm	Me	VHSP
11	LIp	Mi	Lo	Me	ELSP	51	MIp	Lg	Mdm	Hi	EHSP
12	LIp	Mi	Lo	Hi	VLSP	52	MIp	Lg	Hgh	Sm	VHSP
13	LIp	Mi	Mdm	Sm	ELSP	53	MIp	Lg	Hgh	Me	EHSP
14	LIp	Mi	Mdm	Me	VLSP	54	MIp	Lg	Hgh	Hi	EHSP
15	LIp	Mi	Mdm	Hi	LSP	55	VIp	Sho	Lo	Sm	LSP
16	LIp	Mi	Hgh	Sm	VLSP	56	VIp	Sho	Lo	Me	VLSP
17	LIp	Mi	Hgh	Me	MSP	57	VIp	Sho	Lo	Hi	LSP
18	LIp	Mi	Hgh	Hi	HSP	58	VIp	Sho	Mdm	Sm	VLSP
19	LIp	Lg	Lo	Sm	ELSP	59	VIp	Sho	Mdm	Me	MSP
20	LIp	Lg	Lo	Me	VLSP	60	VIp	Sho	Mdm	Hi	HSP
21	LIp	Lg	Lo	Hi	MSP	61	VIp	Sho	Hgh	Sm	MSP
22	LIp	Lg	Mdm	Sm	LSP	62	VIp	Sho	Hgh	Me	VHSP
23	LIp	Lg	Mdm	Me	MSP	63	VIp	Sho	Hgh	Hi	EHSP
24	LIp	Lg	Mdm	Hi	VHSP	64	VIp	Mi	Lo	Sm	ELSP
25	LIp	Lg	Hgh	Sm	HSP	65	VIp	Mi	Lo	Me	LSP
26	LIp	Lg	Hgh	Me	VHSP	66	VIp	Mi	Lo	Hi	MSP
27	LIp	Lg	Hgh	Hi	EHSP	67	VIp	Mi	Mdm	Sm	LSP
28	MIp	Sho	Lo	Sm	LSP	68	VIp	Mi	Mdm	Me	MSP
29	MIp	Sho	Lo	Me	VLSP	69	VIp	Mi	Mdm	Hi	VHSP
30	MIp	Sho	Lo	Hi	LSP	70	VIp	Mi	Hgh	Sm	HSP
31	MIp	Sho	Mdm	Sm	VLSP	71	VIp	Mi	Hgh	Me	VHSP
32	MIp	Sho	Mdm	Me	LSP	72	VIp	Mi	Hgh	Hi	EHSP
33	MIp	Sho	Mdm	Hi	HSP	73	VIp	Lg	Lo	Sm	VLSP
34	MIp	Sho	Hgh	Sm	LSP	74	VIp	Lg	Lo	Me	HSP
35	MIp	Sho	Hgh	Me	HSP	75	VIp	Lg	Lo	Hi	VHSP
36	MIp	Sho	Hgh	Hi	VHSP	76	VIp	Lg	Mdm	Sm	HSP
37	MIp	Mi	Lo	Sm	ELSP	77	VIp	Lg	Mdm	Me	VHSP
38	MIp	Mi	Lo	Me	VLSP	78	VIp	Lg	Mdm	Hi	EHSP
39	MIp	Mi	Lo	Hi	MSP	79	VIp	Lg	Hgh	Sm	EHSP
40	MIp	Mi	Mdm	Sm	VLSP	80	VIp	Lg	Hgh	Me	EHSP
						81	VIp	Lg	Hgh	Hi	EHSP

$$
\begin{aligned}
T(IDNC) &= \{LittleImportant(LIp), MediumImportant(MIp), VeryImportant(VIp)\} \\
T(IDWT) &= \{Short(Sho), Medium(Mi), Long(Lg)\} \\
T(IDRE) &= \{Low(Lo), Medium(Mdm), High(Hgh)\} \\
T(IDST) &= \{Small(Sm), Medium(Me), High(Hi)\}
\end{aligned}
$$

The MFs for input parameters are defined as follows.

$$\begin{aligned}
\mu_{LIp}(IDNC) &= g(IDNC; LIp_0, LIp_1, LIp_{w0}, LIp_{w1}) \\
\mu_{MIp}(IDNC) &= f(IDNC; MIp_0, MIp_{w0}, MIp_{w1}) \\
\mu_{VIp}(IDNC) &= g(IDNC; VIp_0, VIp_1, VIp_{w0}, VIp_{w1}) \\
\mu_{Sho}(IDWT) &= g(IDWT; Sho_0, Sho_1, Sho_{w0}, Sho_{w1}) \\
\mu_{Mi}(IDWT) &= f(IDWT; Mi_0, Mi_{w0}, Mi_{w1}) \\
\mu_{Lg}(IDWT) &= g(IDWT; Lg_0, Lg_1, Lg_{w0}, Lg_{w1}) \\
\mu_{Lo}(IDRE) &= g(IDRE; Lo_0, Lo_1, Lo_{w0}, Lo_{w1}) \\
\mu_{Mdm}(IDRE) &= g(IDRE; Mdm_0, Mdm_1, Mdm_{w0}, Mdm_{w1}) \\
\mu_{Hgh}(IDRE) &= g(IDRE; Hgh_0, Hgh_1, Hgh_{w0}, Hgh_{w1}) \\
\mu_{Sm}(IDST) &= g(IDST; Sm_0, Sm_1, Sm_{w0}, Sm_{w1}) \\
\mu_{Me}(IDST) &= f(IDST; Me_0, Me_{w0}, Me_{w1}) \\
\mu_{Hi}(IDST) &= g(IDST; Hi_0, Hi_1, Hi_{w0}, Hi_{w1})
\end{aligned}$$

The small letters *w0* and *w1* mean left width and right width, respectively.

The output linguistic parameter is the IoT Device Selection Decision (IDSD). We define the term set of IDSD as:

$$\begin{aligned}
&\{Extremely\ Low\ Selection\ Possibility, \\
&Very\ Low\ Selection\ Possibility\ (VLSP), \\
&Low\ Selection\ Possibility\ (LSP), \\
&Middle\ Selection\ Possibility\ (MSP), \\
&High\ Selection\ Possibility\ (HSP), \\
&Very\ High\ Selection\ Possibility\ (VHSP), \\
&Extremely\ High\ Selection\ Possibility\ (EHSP)\}.
\end{aligned}$$

The MFs for the output parameter *IDSD* are defined as:

$$\begin{aligned}
\mu_{VLSP}(IDSD) &= g(IDSD; VLSP_0, VLSP_1, VLSP_{w0}, VLSP_{w1}) \\
\mu_{LSP}(IDSD) &= g(IDSD; LSP_0, LSP_1, LSP_{w0}, LSP_{w1}) \\
\mu_{MSP}(IDSD) &= g(IDSD; MSP_0, MSP_1, MSP_{w0}, MSP_{w1}) \\
\mu_{HSP}(IDSD) &= g(IDSD; HSP_0, HSP_1, HSP_{w0}, HSP_{w1}) \\
\mu_{VHSP}(IDSD) &= g(IDSD; VHSP_0, VHSP_1, VHSP_{w0}, VHSP_{w1}).
\end{aligned}$$

The MFs are shown in Fig. 6.9. The FRB has 81 rules.

Chapter 7

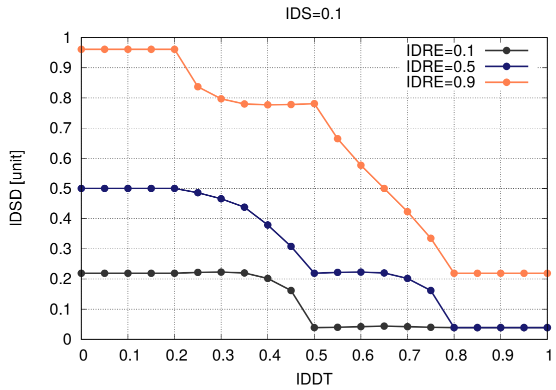
Evaluation of Proposed Systems

After showing the implemented systems in Chapter 6, in this chapter we show their performance evaluation. System evaluation is the process of assessing the performance of the system based on input variations. We used FL, which consists of three stages: input phase, processing stage and output stage. The processing stage is done by the inference engine which simplifies the design procedure with options to add new parameters, variables, membership functions and rules. In this chapter, we present the details of our systems, the parameter selection, explanation and the MF selection. The simulations aim to explore the performance of the proposed systems and evaluate each parameter effect on the output value. The systems were evaluated by computer simulations. The simulations were carried out in a Linux Ubuntu 16.04 LTS OS computer with these specifications: RAM (10.6GB), CPU i5 (3.2 GHz x 4) and SSD (300 GB). For simulation, we used our implemented FuzzyC system [38, 39, 40, 41, 42, 44, 45, 46, 47, 48, 49, 50, 51]. The FuzzyC is a simulation system written in C language and equipped with Fuzzy library.

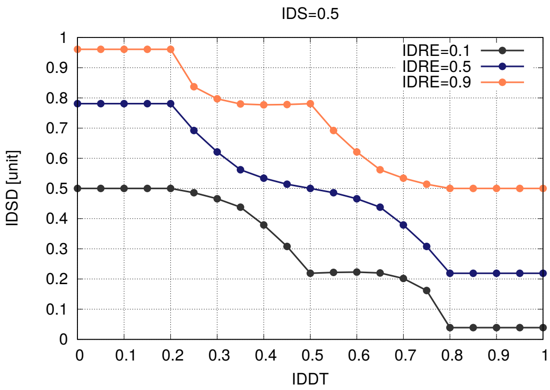
7.1 Simulation Results for IDSS1

The simulations of our first proposed system are shown in Fig.7.1. In this figure we show the relation of IDSD versus the IDS, IDDT and IDRE and evaluate the effect of each parameter on the IDSD.

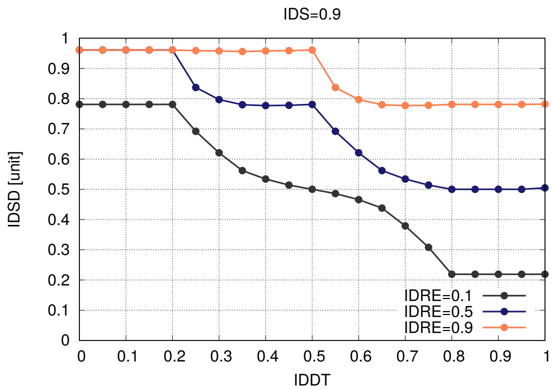
In OppNets due to the lack of a path from source to destination, IoT devices depend on the new contacts they make to deliver the data. Some devices are static, therefore not so active. They depend on other devices mobility speed to make contact with them so they can exchange and receive data. However being dependent on other nodes, greatly reduces



(a) IDDT=0.1



(b) IDDT=0.5



(c) IDDT=0.9

Figure 7.1: Simulation results of IDSS1.

their importance in the network. The effect of speed is shown by comparing Fig.7.1(a) with Fig.7.1(b) and Fig.7.1(b) with Fig.7.1(c), for IDRE=0.1 and IDDT=0.2. When IDS goes from 0.1 to 0.5 and from 0.5 to 0.9 we have an increase of 29% and 28% in IDSD, respectively. The reason for this increase is that mobile devices which move across the network, make new connections. So the number of new contacts is greatly affected by the speed of the devices.

With the results obtained from the simulations, it should be noted that beyond IDDT=0.8 there is not much difference in the output. The reason is that different parameteres are weighted differently based on their importance in the network. For example, residual energy of one IoT device is essential to network longevity. Some IoT devices may be fully resourced, but lack energy therefore are useless. For that reason, IDRE has a very strong effect on IDSD. For example, in Fig.7.1(c), for IDDT=0.5, when IDRE goes from 0.1 to 0.5 and from 0.1 to 0.9, we see an increase of 38% and 47% in IDSD, respectively.

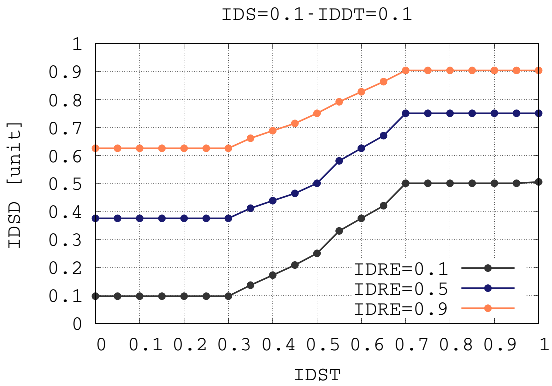
The distance between the IoT device and the task is measured in order to be able to asses whether the said IoT device has the necessary resources for reaching the task and then completing it. The distance between devices and the event is calculated for each device with sensors mounted on them (see Chapter. 8) and then the nearest device to event is favored. In Fig.7.1(b), for IDS=0.5 and IDRE=0.9, we study the effect of IDDT in IDSD. When IDSD increases from 0.2 to 0.5, IDSD decreases 17%. If IDSD increases from 0.5 to 0.9, IDSD decreases even further by 29%. Distance greatly affects the IDSD as it makes IoT devices waste valuable resources. However, a long distance doesn't only affect the device's resources, but also the time the IoT device needs to complete the task. In emergency scenarios that need prompt action, this is intolerable.

7.2 Simulation Results for IDSS2

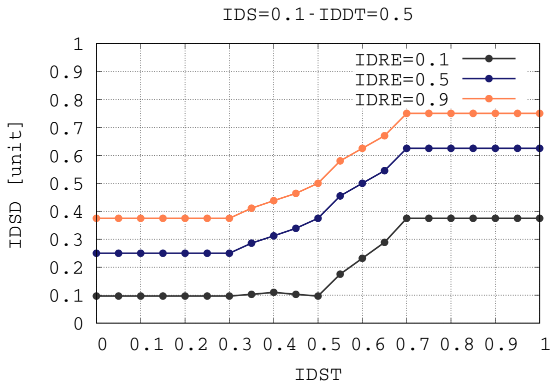
In Fig.7.2, Fig.7.3 and Fig.7.4 are shown the simulation result of IDSS2. In IDSS2, we have added a new parameter, IDST. Adding a fourth parameter to a system, greatly increases its complexity and computational time.

Storage is one of the most important parameters in OppNets considering they use store carry forward mechanism. We have used this parameter in most of our proposed systems due to its importance. Routing in OppNets is very challenging task since there are no pre-established routes between two devices and data delivery relies on the storage capabilities of devices. The effects of varying storage can be observed in Fig.7.2(a), for IDRE=0.9. When IDST of a device, increases from 0.3 to 0.5 and from 0.5 to 0.9, IDSD increases 13% and 15%, respectively. Due to the fact that devices are heterogeneous, we have selected this parameter to include all the scenarios where devices have limited or adequate storage. When one IoT device receives a message for forwarding, it stores it in its buffer and eventually forwards it to the destination device if they ever meet, or to another device.

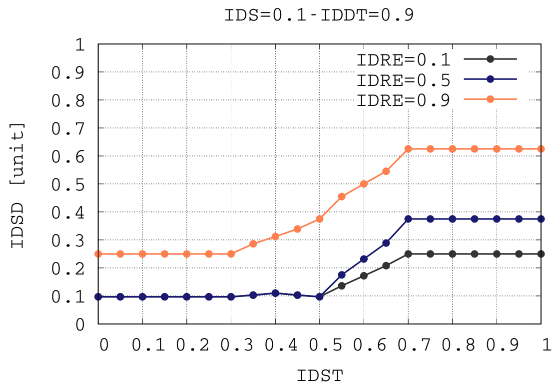
The other three parameters are the same parameters as previously used in IDSS1. In Fig.7.3, for IDST=0.3 and IDRE=0.5, we observe that when IDDT increases from 0.1 to 0.9 and from 0.5 to 0.9, IDSD decreases 25% and 15%, respectively. We compare Fig.7.2(a) with Fig.7.3(a) and Fig.7.2(a) with Fig.7.4(a) to evaluate the effect of IDS on



(a) IDS=0.1-IDDT=0.1



(b) IDS=0.1-IDDT=0.5

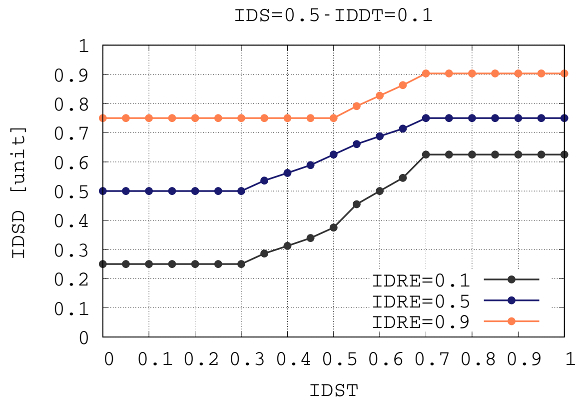


(c) IDS=0.1-IDDT=0.9

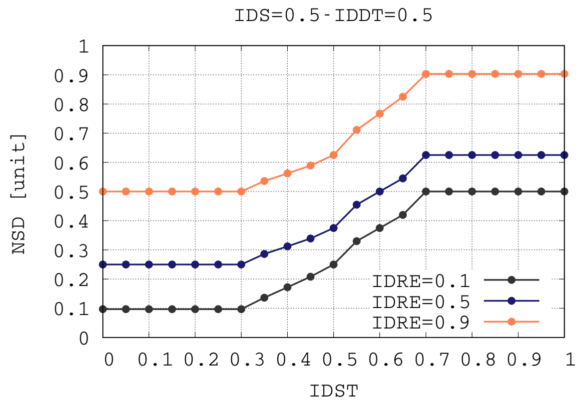
Figure 7.2: Simulation results of IDSS2 for $IDS = 0.1$.

IDSD for IDSS2. When IDS goes from 0.1 to 0.5 and from 0.1 to 0.9 for IDRE=0.9 and IDST=0.3, IDST increases 13% and 28%, respectively.

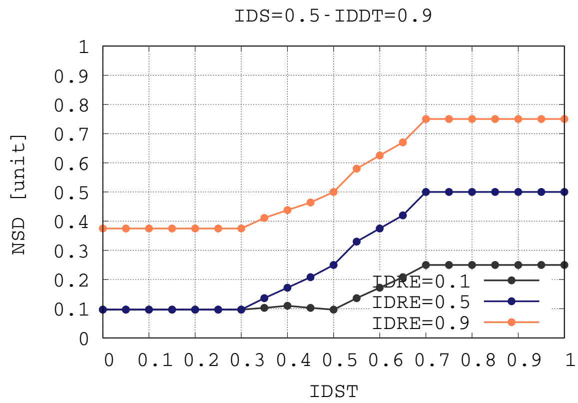
IDSS2 system uses three parameters same to IDSS1, with only one new parameter. However, these systems are significantly different due to the way how these parameters are weighted and assessed. The significance of a parameter is assessed in respect to others. Each one of them is weighted differently based on which parameter should be emphasized.



(a) IDS=0.5-IDDT=0.1

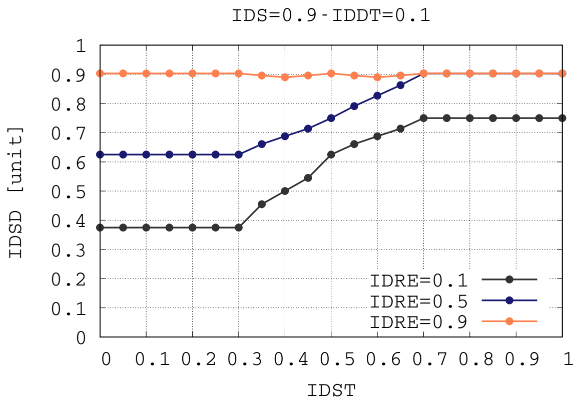


(b) IDS=0.5-IDDT=0.5

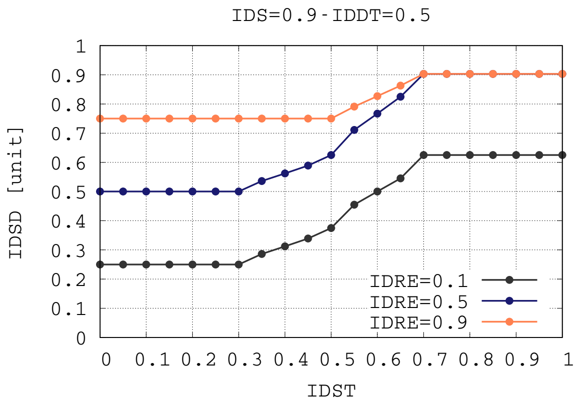


(c) IDS=0.5-IDDT=0.9

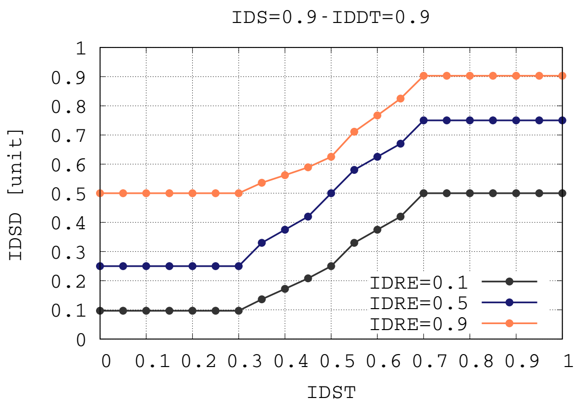
Figure 7.3: Simulation results of IDSS2 for $IDS = 0.5$.



(a) IDS=0.9-IDDT=0.1



(b) IDS=0.9-IDDT=0.5



(c) IDS=0.9-IDDT=0.9

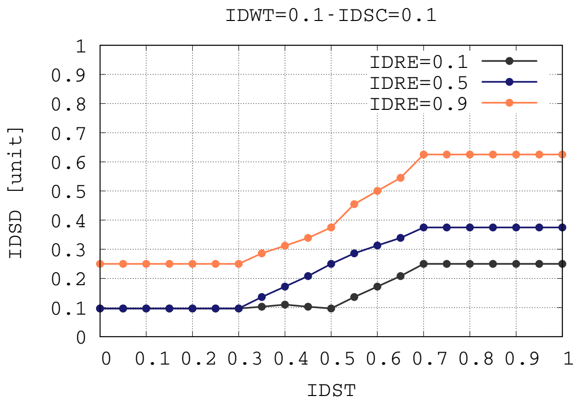
Figure 7.4: Simulation results of IDSS2 for $IDS = 0.9$.

7.3 Simulation Results for IDSS3

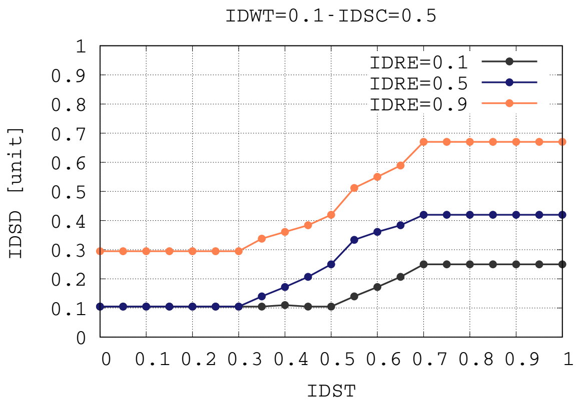
Our next set of results is shown in Fig.7.5, Fig.7.6 and Fig.7.7. Same as IDSS3, in IDSS4 we used four input parameters, but added IDSC and IDWT as two new parameters together with IDRE and IDST. In the previous systems we saw the effect of IDRE and IDST on IDSD.

In OppNets, in the absence of a global infrastructure, all devices are expected to participate in the forwarding process to increase the communication opportunities. However, even though the overall throughput of the network is increased, having all types of devices take part in the forwarding process, poses a security threat. Different from infrastructure based networks, security is a very big challenge in OppNets. The lack of a stable topology makes it difficult to implement device authentication with centralized authentication servers. When two devices come in contact with each other there are three possible scenarios: Device A and Device B do not trust each other since it is the first time they come in contact. Device A fully trusts Device B having previously met. Device A doesn't fully trust Device B. So based on the level of trust that devices have for one another and the level of security mechanisms they have, we have used IDSC parameter. In Figure 7.5, for IDRE=0.1 and IDST=0.7, when IDSC increases from 0.1 to 0.5 and from 0.5 to 0.9, IDSD increases 6% and 22%, respectively. It can be seen that when IDSC increases from 0.1 to 0.5, IDSD has a very low increase. Devices joining or quitting the network have different security mechanisms. Security is something that cannot be compromised, especially in a network which depends on new devices being added to the network. So devices with very high security mechanisms are preferred to others.

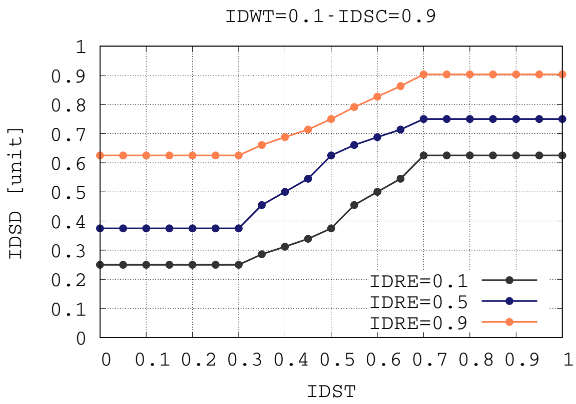
Devices have random waiting time based on many factors, such as mobility patterns, speed, work load and so on. In case of an event, devices are asked to cooperate and selflessly offer their resources based on the network needs. However, many devices show selfish behavior and don't wait around an event to help complete it, but move away. These devices are considered uncooperative devices and are not preferable to be selected. We observe the effect of IDWT by comparing Fig.7.5(a) with Fig.7.6(a), and Fig.7.6(a) with Fig.7.7(a), for IDRE=0.9 and IDST=0.3. When IDWT increases from 0.1 to 0.5 and from 0.5 to 0.9, IDSD increases 14% and 11%, respectively. Some devices may be busier than others, therefore the ones that wait longer are more likely to be selected.



(a) IDWT=0.1-IDSC=0.1

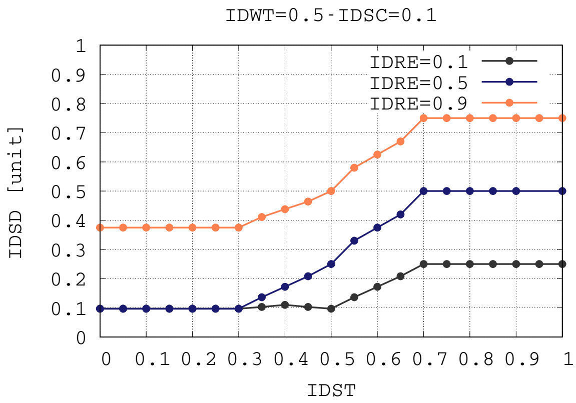


(b) IDWT=0.1-IDSC=0.5

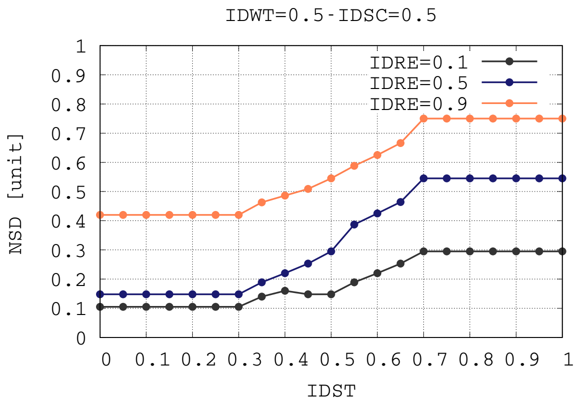


(c) IDWT=0.1-IDSC=0.9

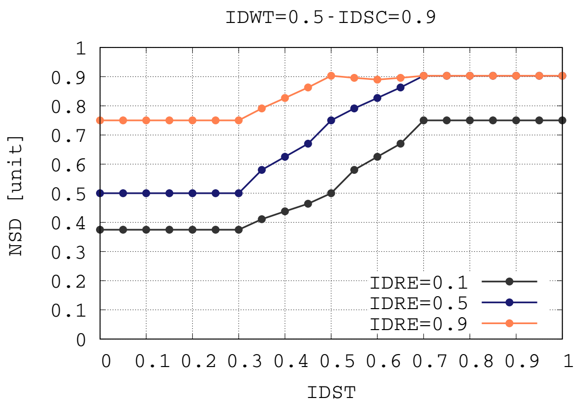
Figure 7.5: Simulation results of IDSS3 for $IDWT = 0.1$.



(a) IDWT=0.5-IDSC=0.1

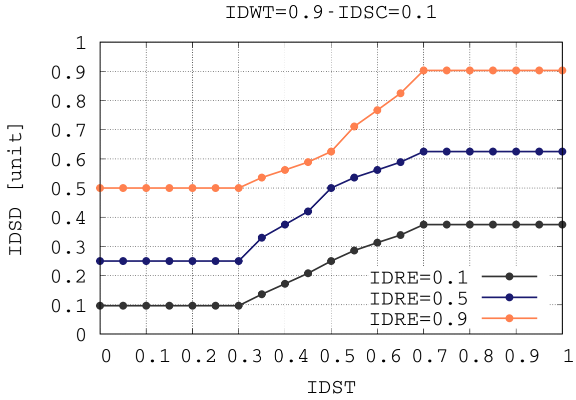


(b) IDWT=0.5-IDSC=0.5

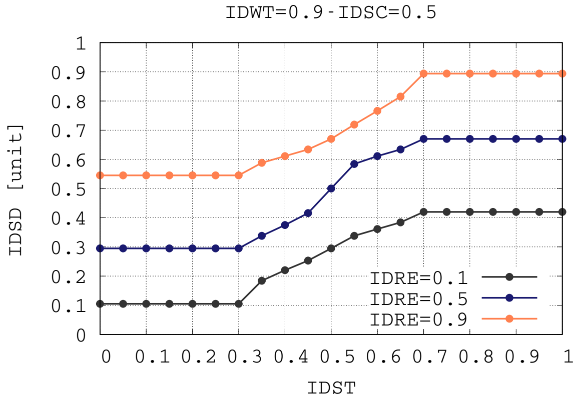


(c) IDWT=0.5-IDSC=0.9

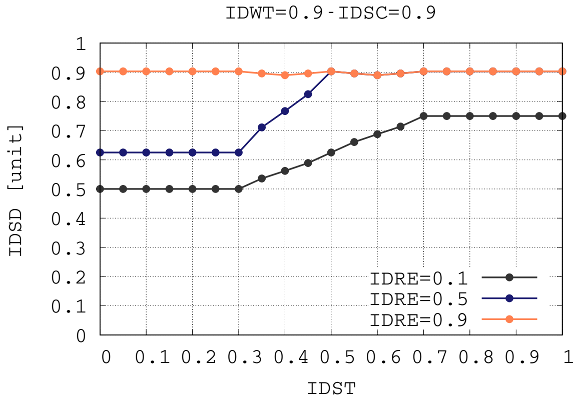
Figure 7.6: Simulation results of IDSS3 for $IDWT = 0.5$.



(a) IDWT=0.9-IDSC=0.1

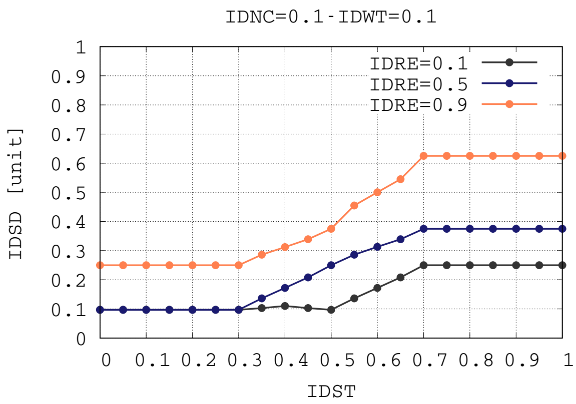


(b) IDWT=0.9-IDSC=0.5

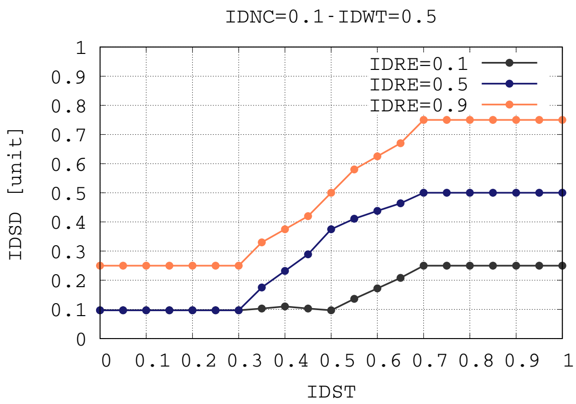


(c) IDWT=0.9-IDSC=0.9

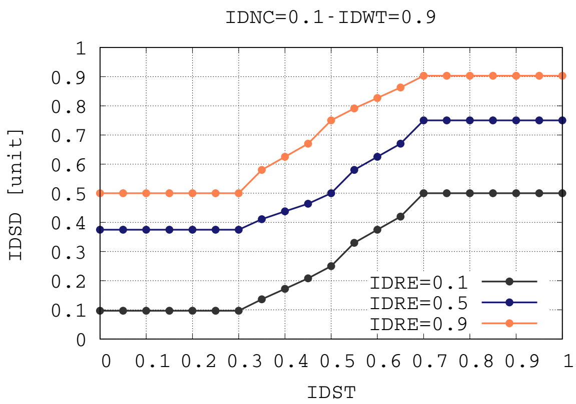
Figure 7.7: Simulation results of IDSS3 for $IDWT = 0.9$.



(a) IDNC=0.1-IDWT=0.1

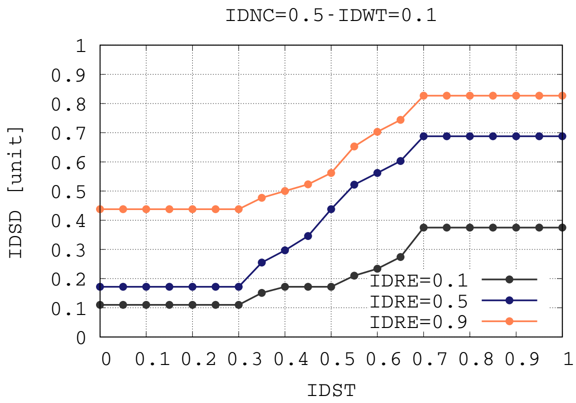


(b) IDNC=0.1-IDWT=0.5

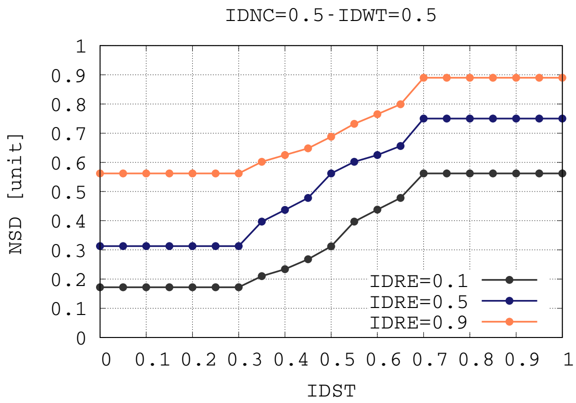


(c) IDNC=0.1-IDWT=0.9

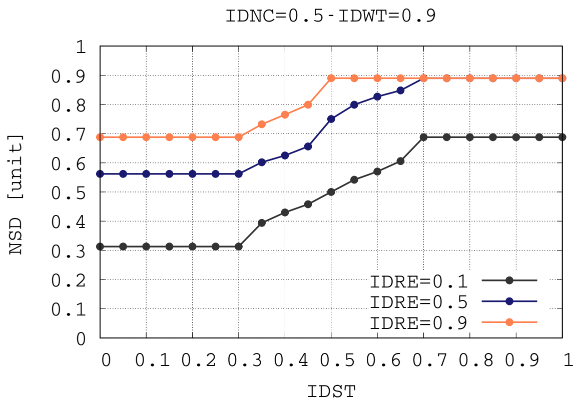
Figure 7.8: Simulation results of IDSS4 for $IDNC = 0.1$.



(a) IDNC=0.5-IDWT=0.1

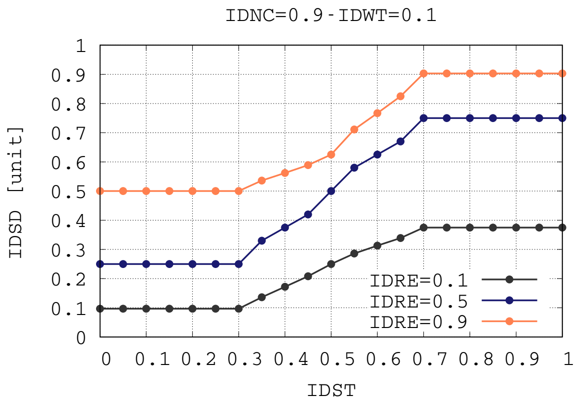


(b) IDNC=0.5-IDWT=0.5

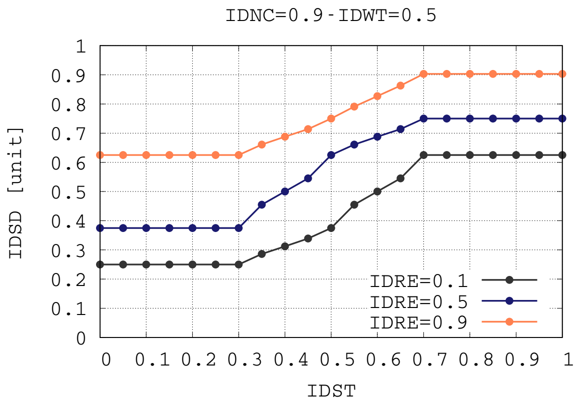


(c) IDNC=0.5-IDWT=0.9

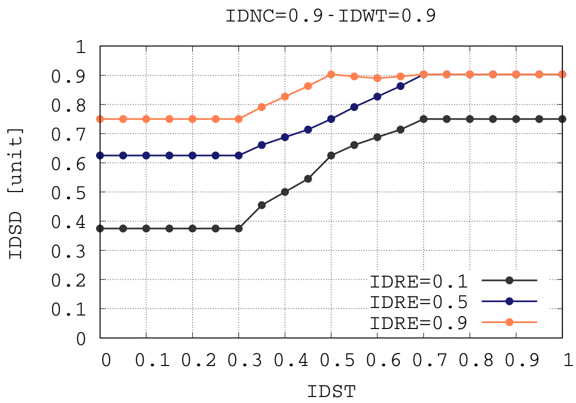
Figure 7.9: Simulation results of IDSS4 for $IDNC = 0.5$.



(a) IDNC=0.9-IDWT=0.1



(b) IDNC=0.9-IDWT=0.5



(c) IDNC=0.9-IDWT=0.9

Figure 7.10: Simulation results of IDSS4 for $IDNC = 0.9$.

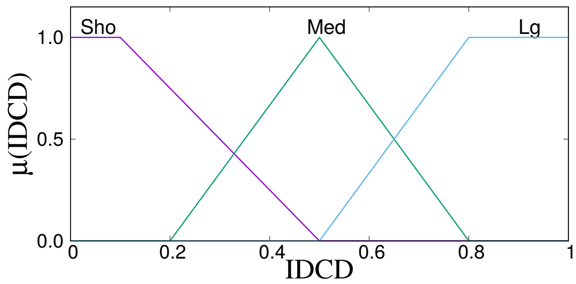
7.4 Simulation Results for IDSS4

The simulation results of IDSS4 are shown in Figures Fig.7.8, Fig.7.9, and Fig.7.10. We used four input parameters and observe the effect of a new parameter: IDNC. Centrality in OppNets, differs from networks were topology of the network is known by the devices. However identification of central devices is very important for the diffusion of the data across the network. In OppNets due to their specific nature, devices do not have global knowledge about the network, so the centrality we have referred to is local. So a high IDNC means that devices are influential and important to the network.

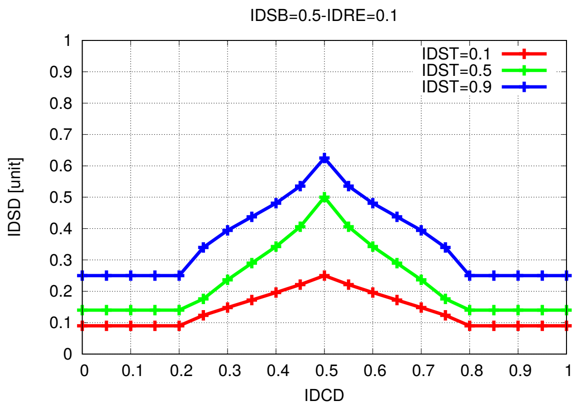
To see the effect of IDNC on IDSD, we compare figures Fig.7.8(a) with Fig.7.9(a) and Fig.7.9(a) with Fig.7.10(a), for IDST=0.3 and IDRE=0.9. When IDNC increases from 0.1 to 0.5 and from 0.5 to 0.9, IDSD increases 12% and 25%, respectively. A high IDNC means that devices are more influential, have more connections and can propagate the information easier than other devices. Even though the connection between devices change over time, centrality is a concept that helps us understand the network connectivity at certain times. Using IDNC as a parameter helps us spread information across the network by using the number of connections a device has.

7.5 Other Systems

To study the OppNets scenarios more extensively, we have implemented many other systems in our previous published works. Our goal was to run simulations to include as many scenarios as possible. Attempting to be more inclusive we have broadened our parameters pool. In one of our previous papers, we introduce IDCD (IoT Device Contact Duration) as a new parameter [43]. Its MFs are shown in Fig. 7.11(a). Contact duration between two devices should be long enough so the whole message is transferred without interruption. However, contact duration is affected by the node's transmission range and its mobility. In some cases more than one contact may be needed to transmit the whole message. In an OppNet, devices with high mobility create contacts with a very short duration time, while less mobile devices will stay in contact with each other for a longer time, since they will not go out of range of each other. In this case, the opportunity of the whole message to be transmitted to the next node is increased, but the network mobility will be affected since it will be limited to a small confined area.



(a) IoT Device Contact Duration



(b) IDSB=0.5, IDRE=0.1

Figure 7.11: IDCD and its MFs.

Contact duration between two devices has a major impact on selection decision. In Fig. 7.11(b), for IDST 0.9, when IDCD is increased from 0.2 to 0.5, IDSD is increased 36%. But when IDCD further increases from 0.5 to 0.8, IDSD is decreased 36%. This is because a short time of contact means that two devices may not have enough time to establish a connection. In realistic OppNet environments, contact durations between IoT nodes are very short due to high node mobility. For example, high speed IoT nodes such as vehicles, create many contacts with very short durations. For these cases to compensate the high speed, IoT nodes communicate via long range communications media WiFi (802.11g). While hand-held devices communicate via Bluetooth which have shorter communication range, but fairly lower speeds than vehicles. When IoT nodes encounter each other, they stay in contact with other IoT nodes depending on their transmission range or speed. However, contact duration must be long enough in order that the message is transferred during a single contact.

7.6 Summary and Discussion

In this chapter we presented the simulation results for our four proposed systems, measuring the performance based on the different parameters used. We used Fuzzy C, which performs the processing stage of FL. The systems use three and four parameters selected based on the IoT devices' properties. IDSS1, is the only system that uses three input parameters. Comparing it with IDSS2, IDSS3, IDSS4, we note that while IDSS1 is much simpler to implement and requires significantly less resources as it has fewer rules, its performance lacks. However, by using three parameters instead of four, we trade parameter inclusion with the computational time.

We used IDRE in all of our systems since most of the devices are battery powered with some having very limited battery capacity. However devices need sufficient battery to be able to perform basic routing tasks, perform more advanced ones if necessary and also be able to support Fuzzy C for the selection process since each device must complete the selection process within itself. A high battery level or remaining energy, increased the possibility of one device to be selected up to 47%. Because this resource is scarce, devices preserve energy by going into idle mode while not busy, or in less preferred scenarios, they exhibit selfish behavior and don't offer their resources as a way of preserving battery.

Other parameters used, such as IDS, affected IDSD by increasing it up to 29%. While a high IDSC increases IDSD 22%. Security is a great concern in OppNets since it is hard to quantify the security level of devices. Other parameters such as IDST affect IDSD by increasing it 25%. Devices have to carry data for an undefined amount of time so without enough storage they are useless in OppNets. Devices are scattered across the network some being more remote than others. This remoteness is translated in nodes being separated from the others, therefore do not make enough contacts to spread the network connectivity. That is why a high IDNC increases IDSD up to 25%. What is clear is that different parameters affect IDSD differently, based on their importance or specific scenarios.

Chapter 8

Testbed Implementation

8.1 Testbed Settings

In this chapter, we propose and implement a Fuzy-based system for selection of IoT nodes in OppNets: INSS1. INSS1 considers four input parameters: Node's Distance from Task (NDT), Node's Remaining Energy (NRE), Node's Buffer Occupancy (NBO) and Node Inter Contact Time (NICT). To better evaluate the proposed system, we implemented a testbed and compared experimental results with the simulation results. The testbed setup shown in Fig. 8.1, consists of the hardware and software part. Different data sensing sensors are mounted on Arduino Uno via IoT Tab Shield 4. This sensed data is collected by a processing device which is connected to Arduino Uno via USB cable. The processing device consists of Raspberry Pi 3 model B+ which operates on an optimized Debian based system, or a Mac OS laptop [54]. We have implemented the Testbed as shown in Fig. 8.2 to test the proposed simulation system. For the software part, we used Arduino IDE to collect the sensed data, Processing language to read this data and FuzzyC to evaluate which of the nodes based on the data is more likely to be selected for a certain task. The hardware is mounted on different IoT nodes to imitate a real life scenario. In Fig. 8.2(a) and Fig. 8.2(b) are shown static and mobile IoT nodes, respectively. In static IoT nodes, the data is sensed by the sensor mounted in Arduino with IoT Tab Shield 4, read and processed using the laptop. For mobile IoT nodes, we use Raspberry Pi 3 model B+ for data reading and processing, which is power supplied by a 24000mAh battery with a lcd display for battery level reading.

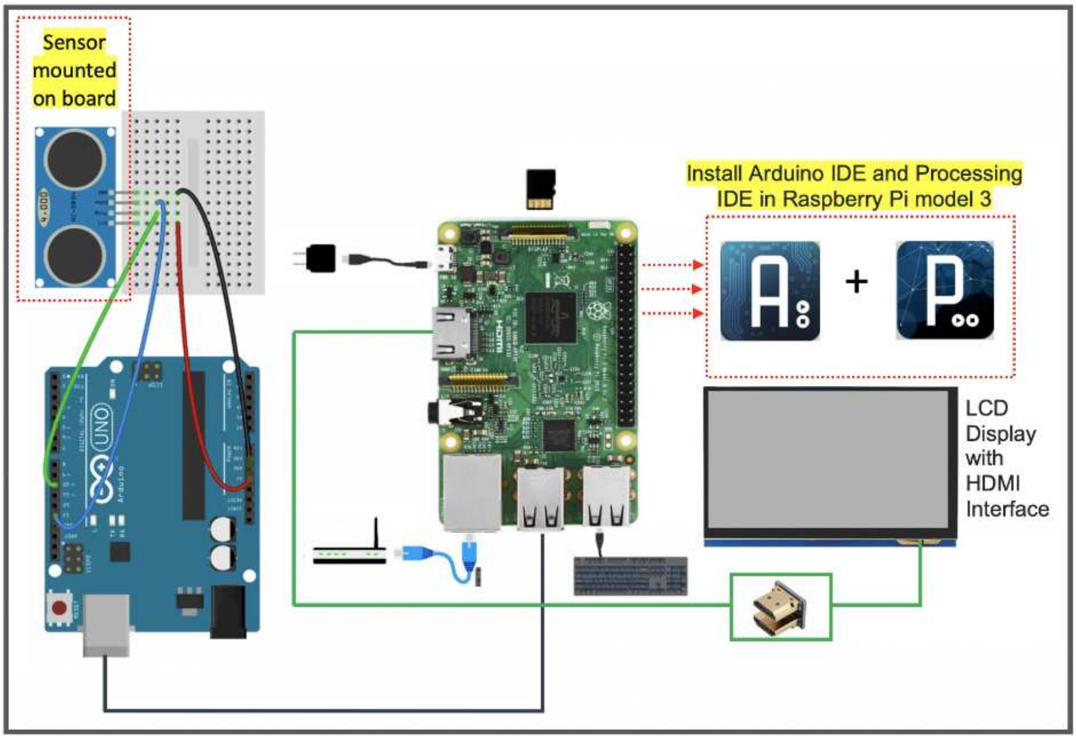
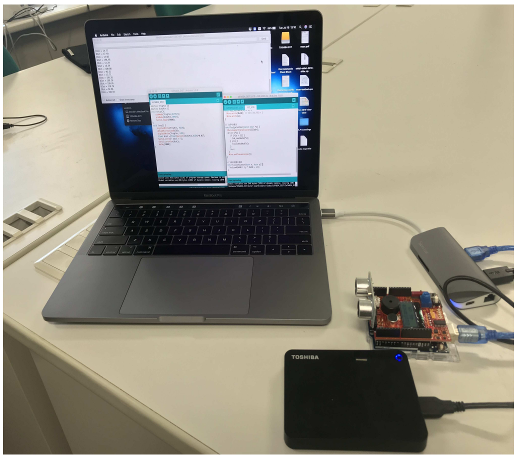
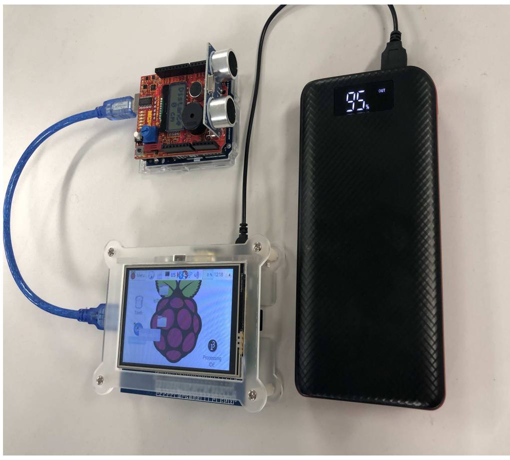


Figure 8.1: Testbed Setup.



(a) Statically deployed IoT Nodes.



(b) Mobile IoT Nodes.

Figure 8.2: Testbed Implementation.

8.2 Testbed Parameters

The structure of the proposed system for the IoT node selection is shown in Fig. 8.3. Based on the OppNets challenges, we have considered the following parameters for the implementation of INSS1:

Node's Distance to Task (NDT): The distance of a node from the task is an important parameter. An IoT node will be selected to carry out a task with high possibility if the node is close to the task.

Node's Remaining Energy (NRE): Depending on the need and the difficulty of the task, different IoT nodes have different levels of remaining energy. Some IoT nodes have limited energy during a certain time which affects how long will this node operate. This is a very important parameter as it affects the longevity and the topology of the network, since in Oppnet each node is a combination of source, influencer node or a destination.

Node's Buffer Occupancy (NBO): In an network that consists of diverse IoT nodes with different resources, buffer occupancy at a certain time is very important. Some IoT nodes are in more advantageous position than others, making them more likely to deliver messages thus making them busier than others. Due to high amount of traffic, these nodes's buffer may overflow affecting the average throughput and the dropping ratio.

Node Inter Contact Time (NICT): The inter-contact time measures the time between the end of previous contact and the beginning of a new one between two IoT nodes. Shorter inter-contact time means having more opportunities to forward the message to the next IoT node.

The output parameter for INSS1 is NSD:

Node Selection Decision (NSD): When an task requires an IoT node to complete it, an evaluation of all the IoT nodes which are part of the network is made in order to choose the best one, but also to manage resources efficiently.

8.3 Fuzzy-based Testbed

Fuzzy sets and FL [52] have been developed to manage vagueness and uncertainty in a reasoning process of an intelligent system such as a knowledge based system, an expert system or a logic control system. In this chapter, we use fuzzy logic system called FuzzyC [53] to implement the proposed fuzzy-based testbed.

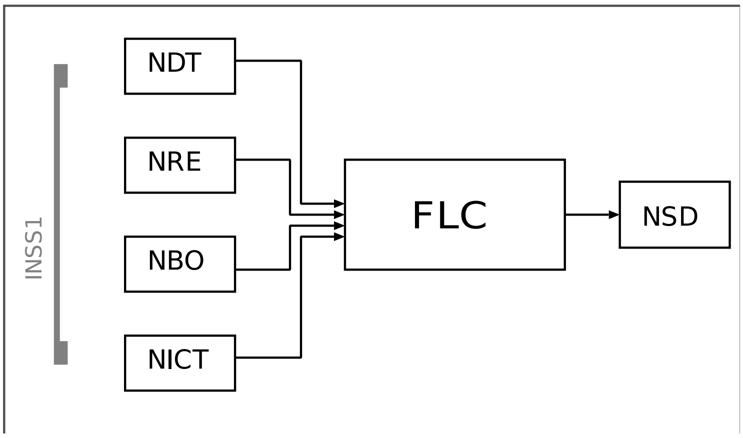


Figure 8.3: Proposed fuzzy-based testbed model.

Table 8.1: Parameters and their term sets for FLC.

Parameters	**Term Sets**
Node's Distance to Task *(NDT)*	Near *(Nr)*, Close *(Cl)*, Far *(Fr)*
Node's Remaining Energy *(NRE)*	Low *(Lo)*, Medium *(Md)*, High *(Hg)*
Node's Buffer Occupancy *(NBO)*	Minimum *(Min)*, Medium *(Med)*, Maximum *(Max)*
Node Inter Contact Time *(NICT)*	Short *(Sh)*, Medium *(Mdm)*, Long *(Lng)*
Node Selection Decision *(NSD)*	Extremely Low Selection Possibility *(ELSP)*, Very Low Selection Possibility *(VLSP)*,Low Selection Possibility *(LSP)*, Medium Selection Possibility *(MSP)*, High Selection Possibility *(HSP)*, Very High Selection Possibility *(VHSP)*, Extremely High Selection Possibility *(EHSP)*

The structure of the proposed system is shown in Fig. 8.3. The term sets for these parameters are shown in Table 8.1. When an task requires an IoT node to complete it, an evaluation of all the IoT nodes which are part of the network is made in order to choose the best one, but also to manage resources efficiently. Since we have used four input parameters for the INSS1 our systems has 81 rules. These parameters will be represented from numerical form into linguistic variables. We use fuzzy membership functions to quantify the linguistic term. The fuzzy membership functions of our system our shown in Fig. 8.4. We use triangular and trapezoidal membership functions for FLC, because they are suitable for real-time operations.

Table 8.2: FRB for INSS1.

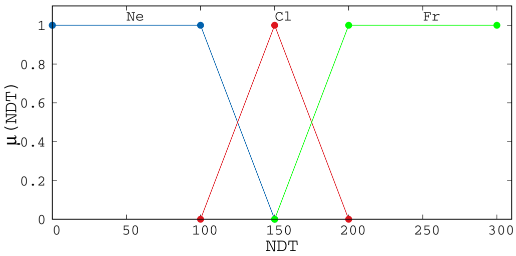
No.	NDT	NRE	NBO	NICT	NSD	No.	NDT	NRE	NBO	NICT	NSD	No.	NDT	NRE	NBO	NICT	NSD
1	Nr	Lo	Min	Sh	EHSP	28	Cl	Lo	Min	Sh	VHSP	55	Fr	Lo	Min	Sh	VHSP
2	Nr	Lo	Min	Mdm	VHSP	29	Cl	Lo	Min	Mdm	MSP	56	Fr	Lo	Min	Mdm	LSP
3	Nr	Lo	Min	Lng	VHSP	30	Cl	Lo	Min	Lng	MSP	57	Fr	Lo	Min	Lng	LSP
4	Nr	Lo	Med	Sh	EHSP	31	Cl	Lo	Med	Sh	HSP	58	Fr	Lo	Med	Sh	MSP
5	Nr	Lo	Med	Mdm	HSP	32	Cl	Lo	Med	Mdm	VLSP	59	Fr	Lo	Med	Mdm	VLSP
6	Nr	Lo	Med	Lng	HSP	33	Cl	Lo	Med	Lng	VLSP	60	Fr	Lo	Med	Lng	VLSP
7	Nr	Lo	Max	Sh	HSP	34	Cl	Lo	Max	Sh	LSP	61	Fr	Lo	Max	Sh	VLSP
8	Nr	Lo	Max	Mdm	LSP	35	Cl	Lo	Max	Mdm	ELSP	62	Fr	Lo	Max	Mdm	ELSP
9	Nr	Lo	Max	Lng	LSP	36	Cl	Lo	Max	Lng	ELSP	63	Fr	Lo	Max	Lng	ELSP
10	Nr	Md	Min	Sh	EHSP	37	Cl	Md	Min	Sh	EHSP	64	Fr	Md	Min	Sh	VHSP
11	Nr	Md	Min	Mdm	EHSP	38	Cl	Md	Min	Mdm	HSP	65	Fr	Md	Min	Mdm	MSP
12	Nr	Md	Min	Lng	EHSP	39	Cl	Md	Min	Lng	HSP	66	Fr	Md	Min	Lng	MSP
13	Nr	Md	Med	Sh	EHSP	40	Cl	Md	Med	Sh	VHSP	67	Fr	Md	Med	Sh	HSP
14	Nr	Md	Med	Mdm	HSP	41	Cl	Md	Med	Mdm	LSP	68	Fr	Md	Med	Mdm	VLSP
15	Nr	Md	Med	Lng	HSP	42	Cl	Md	Med	Lng	LSP	69	Fr	Md	Med	Lng	VLSP
16	Nr	Md	Max	Sh	VHSP	43	Cl	Md	Max	Sh	MSP	70	Fr	Md	Max	Sh	LSP
17	Nr	Md	Max	Mdm	MSP	44	Cl	Md	Max	Mdm	VLSP	71	Fr	Md	Max	Mdm	ELSP
18	Nr	Md	Max	Lng	MSP	45	Cl	Md	Max	Lng	VLSP	72	Fr	Md	Max	Lng	ELSP
19	Nr	Hg	Min	Sh	EHSP	46	Cl	Hg	Min	Sh	EHSP	73	Fr	Hg	Min	Sh	EHSP
20	Nr	Hg	Min	Mdm	EHSP	47	Cl	Hg	Min	Mdm	EHSP	74	Fr	Hg	Min	Mdm	VHSP
21	Nr	Hg	Min	Lng	EHSP	48	Cl	Hg	Min	Lng	EHSP	75	Fr	Hg	Min	Lng	VHSP
22	Nr	Hg	Med	Sh	EHSP	49	Cl	Hg	Med	Sh	EHSP	76	Fr	Hg	Med	Sh	EHSP
23	Nr	Hg	Med	Mdm	EHSP	50	Cl	Hg	Med	Mdm	VHSP	77	Fr	Hg	Med	Mdm	HSP
24	Nr	Hg	Med	Lng	EHSP	51	Cl	Hg	Med	Lng	VHSP	78	Fr	Hg	Med	Lng	HSP
25	Nr	Hg	Max	Sh	EHSP	52	Cl	Hg	Max	Sh	EHSP	79	Fr	Hg	Max	Sh	VHSP
26	Nr	Hg	Max	Mdm	VHSP	53	Cl	Hg	Max	Mdm	MSP	80	Fr	Hg	Max	Mdm	LSP
27	Nr	Hg	Max	Lng	VHSP	54	Cl	Hg	Max	Lng	MSP	81	Fr	Hg	Max	Lng	LSP

8.4 Evaluation Results

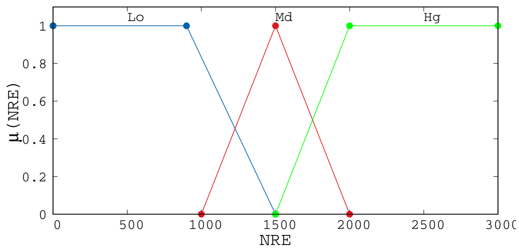
In this subsection, we will present the simulation results of the system. The results for the INSS1 are shown in figure Fig. 8.5, Fig. 8.6, Fig. 8.7. We show the relation between the possibility of an IoT node to be selected (NSD) to carry out a task, versus NDT, NRE, NBO and NICT.

In Fig. 8.5(a), Fig. 8.5(b) and Fig. 8.5(c), are shown the figures for three different values of energy, from lowest to highest. To show how remaining energy affects the selection of an IoT node, we compare Fig. 8.5(a) with Fig. 8.5(b) and Fig. 8.5(b) with Fig. 8.5(c), for NICT=0.4 and NBO=0.9. For NRE=0.1 to NRE=0.5, NSD is increased 12% and 25% for NRE=0.5 to NRE=0.9. IoT nodes who have higher residual energy have better odds of staying connected to the network.

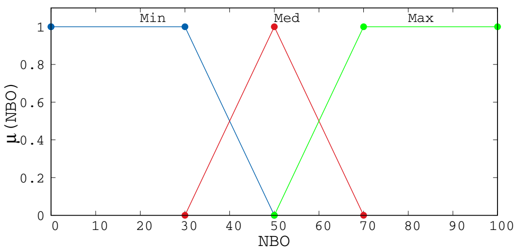
In Fig. 8.6(a), Fig. 8.6(b) and Fig. 8.6(c) are shown the simulation results for NDT=0.5. Comparing Fig. 8.6(a) with Fig. 8.5(a), when NICT=0.4 and NBO=0.1, we see that tha



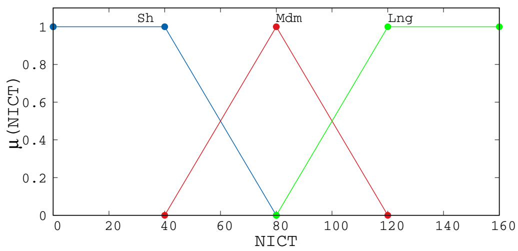
(a) Node's Distance from Task



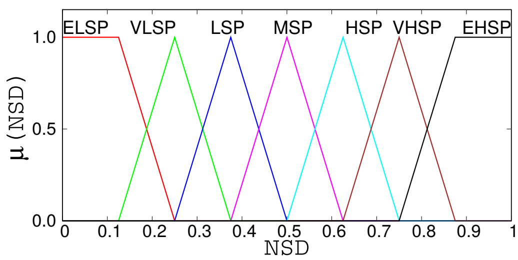
(b) Node's Remaining Energy



(c) Node's Buffer Occupancy



(d) Node Inter Contact Time



(e) Node Selection Decision

Figure 8.4: Fuzzy membership functions.

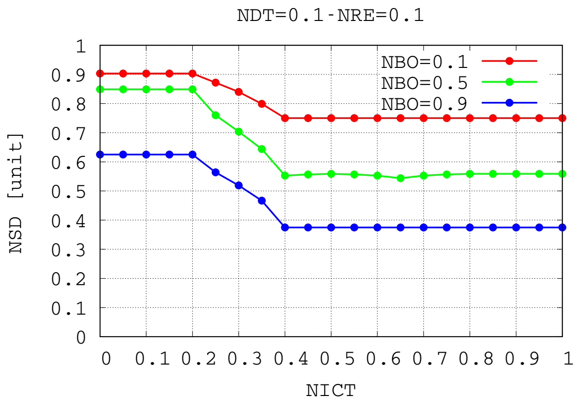
NDS is decreased 16%. This means that nodes which are far from task, are less likely to be selected since these IoT nodes will need more resources to reach this task.

In Fig. 8.7(a) and Fig. 8.7(c), the NDT is increased to 0.9. We have a further decrease of NSD with the increase of NDT. In Fig. 8.7(a), for NICT=0.2 to NICT=0.4 and NBO=0.1, we see that NSD is decreased 38%. IoT nodes that take a longer time to come in contact with other nodes will create less connections, thus the possibility that the IoT node be selected decreases. To see the effect that buffer occupancy has on NSD, we take NICT=0.4 for NBO=0.9 and NBO=0.1 in Fig. 8.7(c). We see that NSD is increased 40% with the decrease of NBO from NBO=0.9 to NBO=0.1. The buffer of some IoT nodes may be occupied or fully occupied. Since these networks use store-carry-forward mechanism, an occupied buffer will cause a congestion due to buffer overflow.

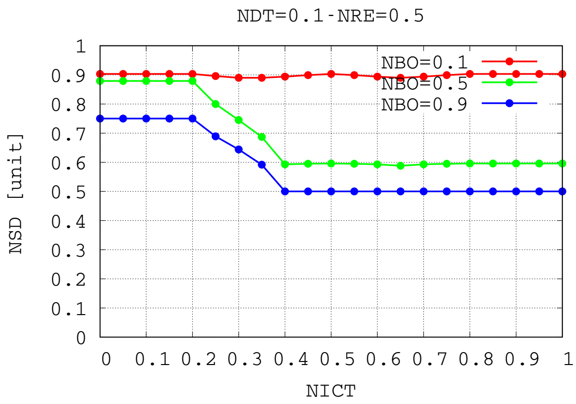
8.4.1 Experimental Results

The experimental results are shown in Fig. 8.8, Fig. 8.9, Fig. 8.10. In Fig. 8.8(a) and Fig. 8.8(c) are shown the results for NDT=Near, NRE=Low and NDT=Near, NRE=High, respectively. In the paragraph above we showed the result of the simulation system. Since results obtained from emulators are more accurate than those obtained from simulators, we implemented a testbed to further evaluate our proposed systems. During the testbed implementation we gathered a lot of data from the sensors.

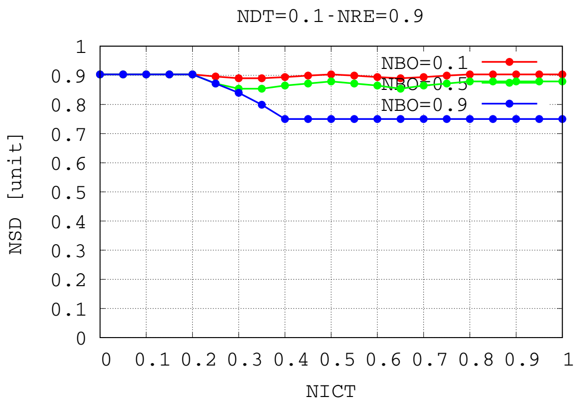
The simulation results in Fig. 8.5(a), Fig. 8.5(b) and Fig. 8.5(c) are close with experimental results in Fig. 8.8(a), Fig. 8.8(b) and Fig. 8.8(c). However, there are some variations from point to point which represent the different outside factors that affect experimental results. In Fig. 8.9(a), Fig. 8.9(b) and Fig. 8.9(c) are shown results for NDT=Close, NRE=Low, NDT=Close, NRE=Medium and NDT=Close, NRE=Long. In Fig. 8.10(a). Fig. 8.10(b) and Fig. 8.10(c), are shown results for NDT=Far, NRE=Low, NDT=Far, NRE= Medium and NDT=Far, NRE=High. For all the above results, we can see that the simulation results are close to the experimental results.



(a) NDT=0.1, NRE=0.1

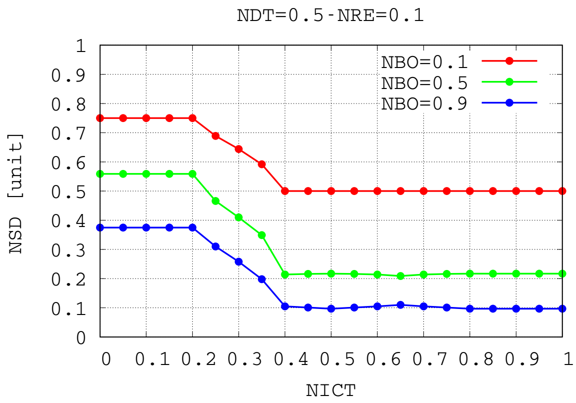


(b) NDT=0.1, NRE=0.5

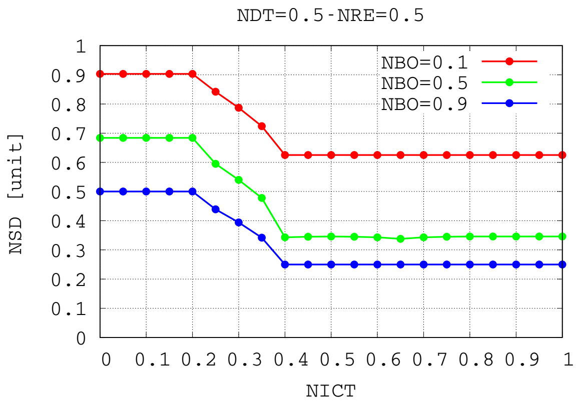


(c) NDT=0.1, NRE=0.9

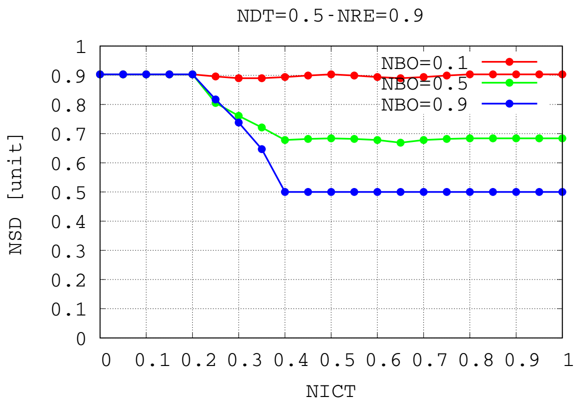
Figure 8.5: Simulation Results for NDT=0.1.



(a) NDT=0.5, NRE=0.1

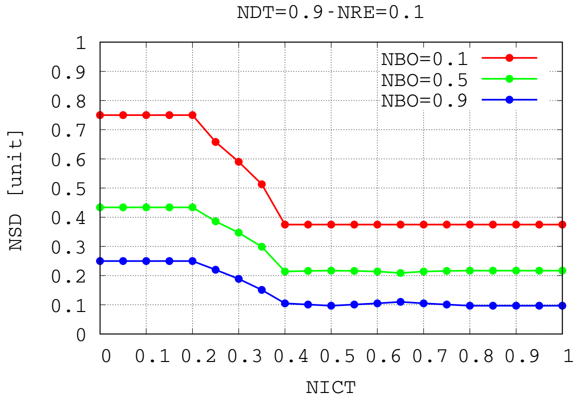


(b) NDT=0.5, NRE=0.5

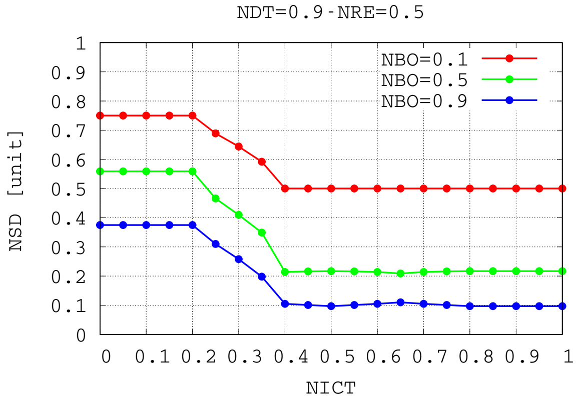


(c) NDT=0.5, NRE=0.9

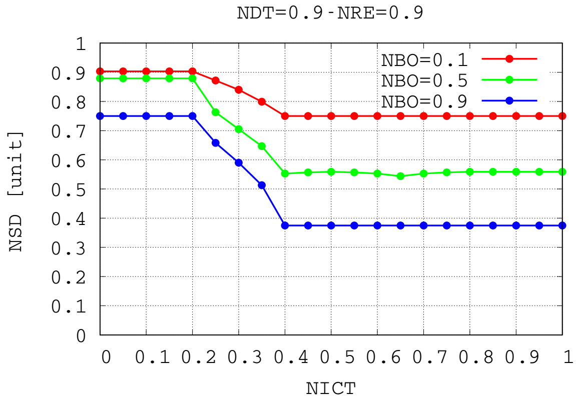
Figure 8.6: Simulation Results for NDT=0.5.



(a) NDT=0.9, NRE=0.1

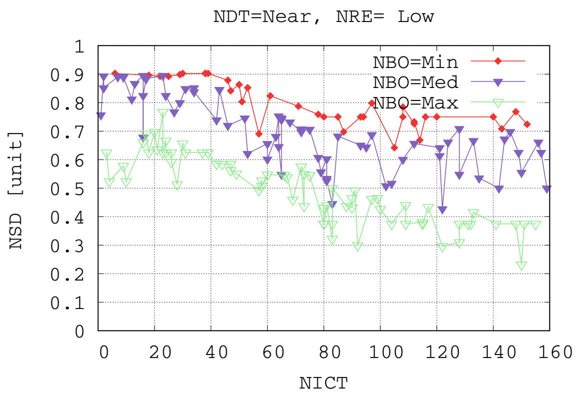


(b) NDT=0.9, NRE=0.5

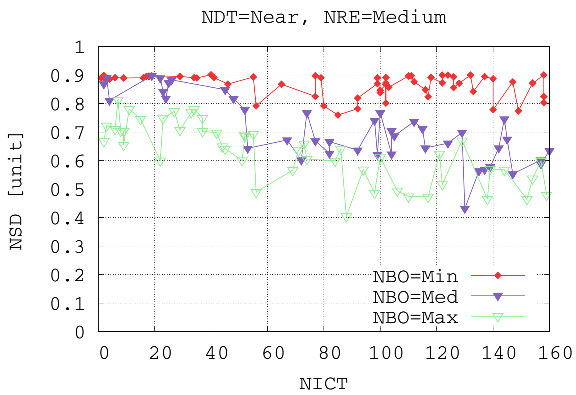


(c) NDT=0.9, NRE=0.9

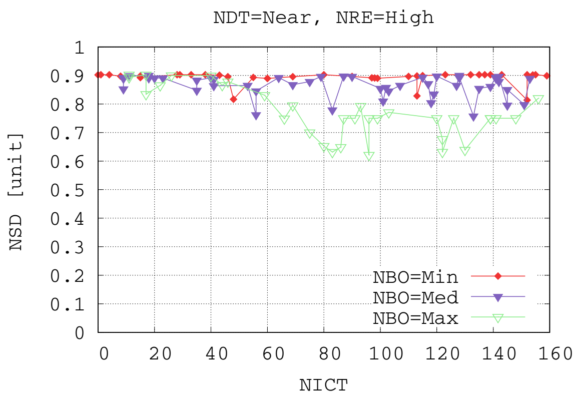
Figure 8.7: Simulation Results for NDT=0.9.



(a) NDT= Near, NRE=Low

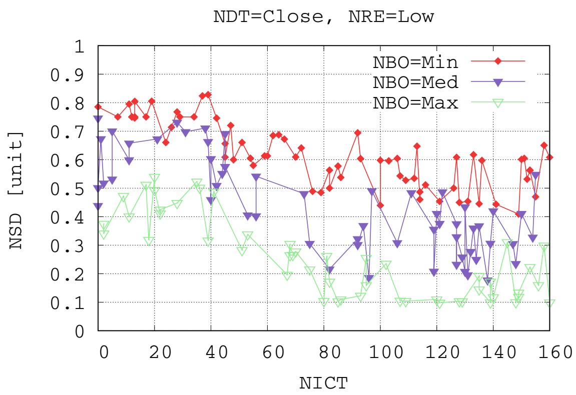


(b) NDT= Near, NRE=Medium

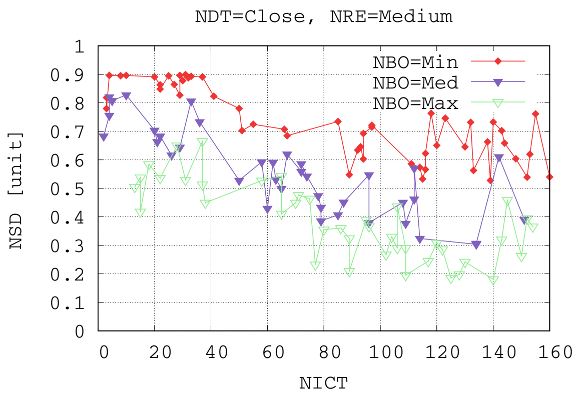


(c) NDT= Near, NRE=High

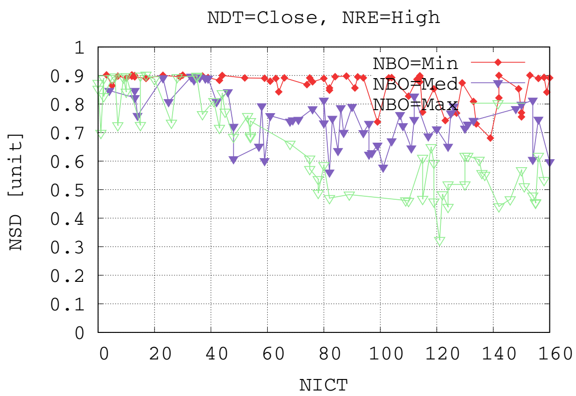
Figure 8.8: Simulation Results for NDT=Near.



(a) NDT= Close, NRE=Low

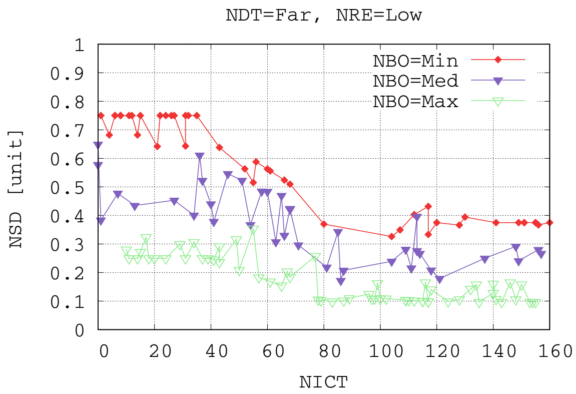


(b) NDT= Close, NRE=Medium

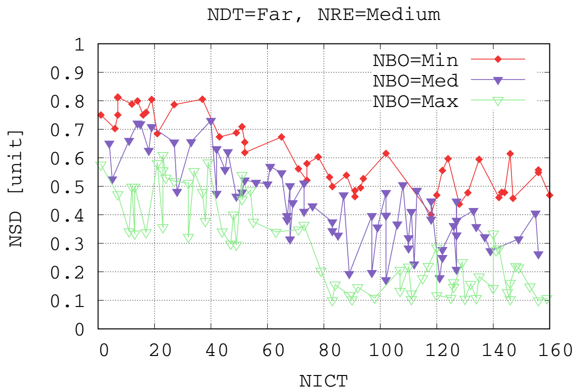


(c) NDT= Close, NRE=High

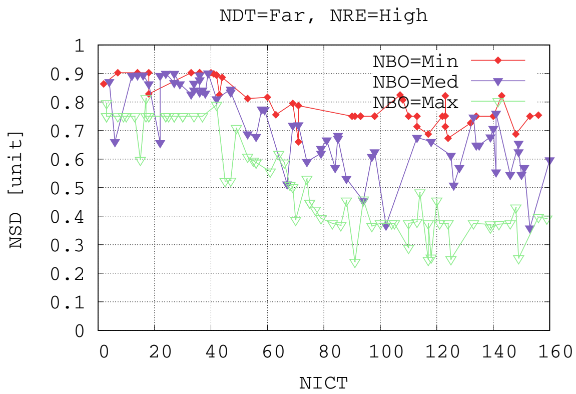
Figure 8.9: Simulation Results for NDT=Close.



(a) NDT= Far, NRE=Low



(b) NDT= Far, NRE=Medium



(c) NDT= Far, NRE=High

Figure 8.10: Simulation Results for NDT=Far.

Chapter 9

Concluding Remarks

9.1 Conclusions and Future Work

In this thesis, we proposed and implemented four fuzzy-based simulation systems and one testbed that decide whether one IoT device will be selected for a specific task based on the device's parameters. The thesis is organized as follows:

In Chapter 1, we presented the introduction, background of the thesis and its content. Also the outline of this thesis is included in this chapter. In Chapter 2 we descibe some of the wireless networks such as NGWN, SDN, SDWN and MANET and their characteristic.

Chapter 3, we provided an introduction of IoT and OppNets. It described their main characteristic and applications, OppNets protocol stack, architecture and challenges. In Chapter 4 we introduced the concept of IA and described some of the most commonly used algorithms. Chapter 5 presented fuzzy logic. It discussed the meaning and basics of fuzzy theory and its principles such as linguistic variables, FC rules, fuzzification and defuzification methods. In Chapter 6, we explain in detail the design and implementation of our fuzzy-based simulation systems. In Chapter 7, we evaluated the performance of the proposed simulation systems.

In Chapter 8, we present a simulation system and a testbed for IoT device selection, and compared and evaluate both systems. The main goal of the proposed systems is to find an IoT device that is more likely to finish a task. To chose the best device many parameters were considered based on the challenges that devices in an OppNets face. Deployed in different scenarios with different characteristic each device has different parameters at a certain moment making them more or less likely to finish a task. To consider as many scenarios as possible, we implemented four systems for IoT device selection with different combinations of parameters.

From the simulation results of IDSS1, we conclude as follows:

- When an IoT device has a higher energy level its importance in the network is significant so IDSD increases.
- Devices that are closer to the event have more advantage than those further away so they are more likely to be selected.
- When devices move fast they have better chances of being closer to an event so a high IDS increases IDSD and their response rate to an emergency situations.

From the simulation results of IDSS2, we found the following results:

- Adding a fourth parameter increases the computational time.
- Considering the architecture of OppNets where nodes have to carry the message for an undefined amount of time, we added IDST as a new parameter and noticed that an IoT device with a bigger buffer size will keep the message longer without dropping it so with IDSD is increased with the increase of IDST.

From the simulation results of IDSS3, we conclude as follows:

- As in IDSS2 we used four input parameters, but added IDWT and IDSC as two new parameters.
- Since OppNets consist of new devices/ helpers being added constantly, devices with higher security mechanisms do not compromise the network and are more favorable.
- Some IoT devices will wait for a longer time to complete a task so they are more likely to be selected than others.

For IDSS4, we observed from simulation results that by adding IDNC as a new parameter, some IoT devices are more central than others and have more connections, so they are more likely to get selected as they increase the message delivery.

From the evaluation of systems, we saw the effect of different parameters on the selection of an IoT device. Our proposed systems gave us an insight on which devices are better than others based on their individual characteristic. We further evaluated our system by implementing a testbed.

From comparing simulation system INSS1 with the testbed we found that the simulation results and experimental results are close, but in the experiment there are some variations.

In the future work, we will consider different parameters combination to evaluate a wider range of scenarios and we will make extensive simulations to evaluate the proposed systems.

References

[1] Salah Eddine Elayoubi, Sana Ben Jemaa, Zwi Altman, Ana Galindo-Serrano, ”5G RAN slicing for verticals: Enablers and challenges”, IEEE Communications Magazine, Vol. 57, No. 1, pp. 28-34, 2019.

[2] Yih Chun Hu, David B. Johnson, ”Caching Strategies in On-demand Routing Protocols for Wireless Ad Hoc Networks”, Proc. of IEEE/ACM MobiCom, 2000.

[3] Mehran Abolhasan, Tadeusz Wysocki, Eryk Dutkiewicz, ”A Review of Routing Protocols for Mobile Ad hoc Networks”, Ad Hoc Networks, Vol. 2, No. 1, pp. 1-22, 2004.

[4] Svitlana Popereshnyak, Olha Suprun, Oleh Suprun, Tadeusz Wieckowski, ”IoT application testing features based on the modeling network”, IEEE, pp. 127–131, 2018.

[5] GSM Association, ”Understanding the Internet of Things (IoT)”, Connected Living, July 2014.

[6] Nanxi Chen, Yang Yang, Jin Li, Tao Zhang, ”A Fog-based service enablement architecture for cross-domain IoT applications”, Fog World Congress (FWC), pp. 1–6, July 2017.

[7] Icon8, ”https://icons8.com/”, Accessed on October, 2019.

[8] Ruhi Kiran Bajaj, Madhuri Rao, Himanshu Agrawal, ”Internet Of Things (IoT) In The Smart Automotive Sector: A Review”, IOSR Journal of Computer Engineering (IOSR-JCE), 2278-0661, p-ISSN: 2278-8727, No. 1, pp. 36-44, 2018.

[9] Chung-Ming Huang, Kun-chan Lan, Chang-Zhou Tsai, ”A survey of opportunistic networks”, 22nd International Conference on Advanced Information Networking and Applications-Workshops, pp. 1672–1677, 2008.

[10] Navneet Kaur, Gauri Mathur, ”Opportunistic networks: A review”, IOSR Journal of Computer Engineering (IOSR-ICE), Vol. 18, No. 2, pp. 20-26, 2016.

[11] Carlo Caini, Rosario Firrincieli, Marco Livini, ”DTN Bundle Layer over TCP: Retransmission Algorithms in the Presence of Channel Disruptions”, JCM, Vol. 5, No. 2, pp. 106-116, 2010.

[12] Carlo Caini, and Rosario Firrincieli, and Marco Livini, ”Neighbor discovery for opportunistic networking in Internet of things scenarios: A survey”, IEEE Access, Vol. 3, pp. 1101–1131, 2015.

[13] Leszek Lilien, Zille Huma Kamal, Ajay Gupta, ”Opportunistic networks: Challenges in specializing the p2p paradigm”, 17th International Workshop on Database and Expert Systems Applications (DEXA'06), pp. 722–726, 2006.

[14] Leszek Lilien, and Ajay Gupta, and Zijian Yang, ”Opportunistic networks for emergency applications and their standard implementation framework”, 2007 IEEE International Performance, Computing, and Communications Conference, pp. 588-593, 2007.

[15] Lotfi Zadeh,”Fuzzy Sets”, Information Control Journal, Elsevier, Vol. 8, No. 3, DOI:10.1016/S0019-9958(65)90241-X, pp. 338-353, June 1965.

[16] Lotfi Zadeh,”Fuzzy Logic, Neural Networks and Soft Computing”, Communications of the ACM, Vol. 37, No. 3, pp. 77-84, March 1994.

[17] Hans-Jurgen Zimmermann,”Fuzzy Set Theory and Its Applications”, Springer, Dordrecht, pp. 203-240, 1991.

[18] John H. Holland, ”Adaptation in natural and artificial systems”, Ann Arbor: University of Michigan Press, 1975.

[19] David E Goldberg, John H Holland, ”Genetic algorithms and machine learning”, Machine learning, Vol. 3, No. 2, pp. 95–99, 1988.

[20] Kumara Sastry, David Goldberg, Graham Kendall, ”Genetic algorithms”, Springer, 97–125, 2005.

[21] Fatos Xhafa, Christian Sanchez, Leonard Barolli, Evjola Spaho, ”Evaluation of genetic algorithms for mesh router nodes placement in wireless mesh networks”,

Journal of Ambient Intelligence and Humanized Computing, Vol. 1, No. 4, 271–282, 2010.

[22] Amol P. Bhondekar, Renu Vig, Madan Lal Singla, C. Ghanshyam, Pawan Kapur, ”Genetic algorithm based node placement methodology for wireless sensor networks”, Proceedings of the international multiconference of engineers and computer scientists, Vol. 1, pp. 18–20, 2010.

[23] Miralda Cuka, Donald Elmazi, Makoto Ikeda, Keita Matsuo and Leonard Barolli, ”IoT Node Selection and Placement: A New Approach Based on Fuzzy Logic and Genetic Algorithm”, Proc. Of. CISIS-2019, Sydney, Australia, pp. 22-35, 2019.

[24] Ali Akbar Abbasi, Muhammad Younis, Kemal Akkaya, ”Movement-Assisted Connectivity Restoration in Wireless Sensor and Actor Networks”, IEEE Transactions on Parallel and Distributed Systems, Vol. 20, No.9, pp. 1366–1379, 2009.

[25] ÖzdaĞoĞlu, Güzin, ”A Simulated Annealing Application on Flowshop Sequencing Problem: A Comparative Case Study”, IEEE Transactions on Fuzzy Systems, Vol. 22, No.2, 2008.

[26] Bart Selman, Carla P. Gomes, ”Hill-climbing Search”, Encyclopedia of Cognitive Science, 2006

[27] Didier Dubois, Henri Prade, Ronald Yager, ”Fuzzy Sets for Intelligent Systems”, Morgan Kaufman Publishers Inc., 1993.

[28] Abraham Kandel, Gideon Langhholz, ”Fuzzy Control Systems”, CRC Press, September 1994.

[29] Tom Procyk, Ebrahim Mamdani, ”A Linguistic Self-organizing Process Controller”, Automatica, Elsevier, Vol. 15, No. 1, pp. 15-30, 1979.

[30] Jan Jantzen, ”Design of fuzzy controllers”, Automatica, Elsevier, Vol. 326, pp. 362–367, 1998.

[31] Kevin M Passino, Stephen Yurkovich, Michael Reinfrank, ”Fuzzy control”, Citeseer, Vol. 42, 1998.

[32] George C. Mouzouris, Jerry M. Mendel, ”Nonsingleton fuzzy logic systems: theory and application”, IEEE Transactions on Fuzzy Systems, Vol. 5, No.1, pp. 56–71, 1997.

[33] Arfi Badredine , ”Fuzzy decision making in politics: A linguistic fuzzy-set approach (LFSA)”, IEEE Transactions on Fuzzy Systems, Vol. 13, No.1, pp. 23–56, 2005.

[34] Timothy J. Ross, ”Properties of membership functions, fuzzification, and defuzzification”, Fuzzy logic with engineering applications, pp. 89–116, 2010.

[35] Zhiwei Zhao, Geyong Min, Weifeng Gao, Yulei Wu, Hancong Duan, Qiang Ni, ”Deploying edge computing nodes for large-scale IoT: A diversity aware approach”, IEEE Internet of Things Journal, Vol. 5, No.5, pp. 3606–3614, 2018.

[36] Ahmed Alagha, Shakti Singh, Rabeb Mizouni, Anis Ouali, Hadi Otrok, ”Data-Driven Dynamic Active Node Selection for Event Localization in IoT Applications - A Case Study of Radiation Localization”, IEEE Access, Vol. 7, pp. 16168-16183, 2019.

[37] Jerry M. Mendel, ”Fuzzy Logic Systems for Engineering: a Tutorial”, Proc. of the IEEE, Vol. 83, No. 3, pp. 345-377, 1995.

[38] Takaaki Inaba, Shinji Sakamoto, Vladi Kolici, Gjergji Mino, Leonard Barolli, ”A CAC Scheme Based on Fuzzy Logic for Cellular Networks Considering Security and Priority Parameters”, Proc. of the 9-th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA-2014), pp. 340-346, November 2014.

[39] Evjola Spaho, Shinji Sakamoto, Leonard Barolli, Fatos Xhafa, Valbona Barolli, Joel Iwashige, ” A Fuzzy-Based System for Peer Reliability in JXTA-Overlay P2P, Considering Number of Interactions”, The 16th International Conference on Network-Based Information Systems, pp. 156–161, 2013.

[40] Keita Matsuo, Donald Elmazi, Yi Liu, Shinji Sakamoto, Gjergji Mino, Leonard Barolli, ” FACS-MP: A Fuzzy Admission Control System with Many Priorities for Wireless Cellular Networks and Its Performance Evaluation”, Journal of High Speed Networks, Vol. 21, No. 1, pp. 1–4, 2015.

[41] Michel Grabisch, " The Application of Fuzzy Integrals in Multicriteria Decision Making", European journal of operational research, Vol. 89, No. 3, pp. 445–456, 1996.

[42] Takaaki Inaba, Donald Elmazi, Yi Liu, Shinji Sakamoto, Leonard Barolli, Kazunori Uchida, "Integrating Wireless Cellular and Ad-Hoc Networks Using Fuzzy Logic Considering Node Mobility and Security", Proc. of the 29th IEEE International Conference on Advanced Information Networking and Applications Workshops, pp. 54-60, March 2015.

[43] Miralda Cuka, Donald Elmazi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, Makoto Takizawa, "Selection of IoT Devices in Opportunistic Networks: A Fuzzy-based Approach Considering IoT Device's Selfish Behaviour", Prof. Of. AINA-2019, Matsue, Japan, pp. 251-264, March 2019.

[44] Elis Kulla, Gjergji Mino, Shinji Sakamoto, Makoto Ikeda, Santi Caballé, Leonard Barolli, "FBMIS: A Fuzzy-Based Multi-interface System for Cellular and Ad Hoc Networks", International Conference on Advanced Information Networking and Applications, pp. 180–185, March 2014.

[45] Donald Elmazi, Elis Kulla, Tesuya Oda, Evjola Spaho, Shinji Sakamoto, Leonard Barolli, "A Comparison Study of Two Fuzzy-based Systems for Selection of Actor Node in Wireless Sensor Actor Networks", Journal of Ambient Intelligence and Humanized Computing, Vol. 6, No. 5, pp. 635–645, 2015.

[46] Evjola Spaho, Shinji Sakamoto, Leonard Barolli, Fatos Xhafa, Makoto Ikeda, "Trustworthiness in P2P: Performance Behaviour of Two Fuzzy-based Systems for JXTA-overlay Platform", Soft Computing, Vol. 18, No. 9, pp. 1783–1793, 2014.

[47] Takaaki Inaba, Shinji Sakamoto, ElisKulla, Santi Caballe, Makoto Ikeda, Leonard Barolli, "An Integrated System for Wireless Cellular and Ad-Hoc Networks Using Fuzzy Logic", International Conference on Intelligent Networking and Collaborative Systems, pp. 157–162, 2014.

[48] Keita Matsuo, Donald Elmazi, Yi Liu, Shinji Sakamoto, Leonard Barolli, "A Multimodal Simulation System for Wireless Sensor Networks: A Comparison Study Considering Stationary and Mobile Sink and Event", Journal of Ambient Intelligence and Humanized Computing, Vol. 6, No. 4, pp. 519–529, 2015.

[49] Vladi Kolici, Takaaki Inaba, Algenti Lala, Gjergji Mino, Shinji Sakamoto, Leonard Barolli, ”A Fuzzy-Based CAC Scheme for Cellular Networks Considering Security”, International Conference on Network-Based Information Systems, pp. 368–373, 2014.

[50] Yi Liu, Shinji Sakamoto, Keita Matsuo, Makoto Ikeda, Leonard Barolli, Fatos Xhafa, ”A Comparison Study for Two Fuzzy-based Systems: Improving Reliability and Security of JXTA-overlay P2P Platform”, Soft Computing, Vol. 20, No. 7, pp. 2677–2687, 2015.

[51] Keita Matsuo, Donald Elmazi, Yi Liu, Shinji Sakamoto, Gjergji Mino, Leonard Barolli, ”FACS-MP: A Fuzzy Admission Control System with Many Priorities for Wireless Cellular Networks and Its Performance Evaluation”, Journal of High Speed Network, Vol. 21, No. 1, pp. 1–14, 2015

[52] Lotfi Zadeh, ”Fuzzy logic, neural networks, and soft computing”, ACM Communications Journal, Vol. 37, No.3, pp. 77â84, 1994.

[53] Takaaki Inaba, Shinji Sakamoto, Tetsuya Oda, Leonard Barolli, Makoto Takizawa, ”A new FACS for cellular wireless networks considering QoS: A comparison study of FuzzyC with MATLAB”, Proc. of the 18-th International Conference on Network-Based Information Systems (NBiS-2015), pp. 338–344, September 2015.

[54] Raspbian (2017), ”https://www.raspbian.org/”, Accessed on September 15, 2017.

List of Papers

Journals Papers

1. Masafumi Yamada, **Miralda Cuka**, Yi Liu, Tetsuya Oda, Keita Matsuo, Leonard Barolli, "Evaluation of an IoT-Based E-Learning Testbed: Performance of OLSR Protocol in a NLoS Environment and Mean-Shift Clustering Approach Considering Electroencephalogram Data", International Journal of Web and Information Systems (IJWIS), Emerald Publishing, Vol. 13. No. 1, pp. 2-13, DOI: 10.1108/IJWIS-12-2016-0072, 18 April 2017.

2. Ryoichiro Obukata, **Miralda Cuka**, Donald Elmazi, Tetsuya Oda, Makoto Ikeda, Leonard Barolli, "Design and Evaluation of an Ambient Intelligence Testbed for Improving Quality of Life", International Journal of Space-Based and Situated Computing (IJSSC), Inderscience, Vol. 7, No. 1, No. 1, pp. 8-15, DOI: 10.1504 IJSSC.2017.084119, March 2017.

3. **Miralda Cuka**, Donald Elmazi, Takaaki Inaba, Tetsuya Oda, Makoto Ikeda, Leonard Barolli, "An Integrated Fuzzy-Based System for Cluster-Head Selection and Sensor Speed Control in Wireless Sensor Networks", International Journal of Distributed Systems and Technologies (IJDST), IGI Global Publishers, Vol. 8, No. 2, pp. 1-14, DOI: 10.4018 IJDST.2017040101, April-June 2017.

4. Keita Matsuo, **Miralda Cuka**, Takaaki Inaba, Tetsuya Oda, Leonard Barolli, "Performance Analysis of Two WMN Architectures by WMN-GA Simulation System Considering Different Distributions and Transmission Rates", International Journal of Grid and Utility Computing (IJGUC), Inderscience, Vol.9 No.1, pp.75 - 82, DOI: 10.1504 IJGUC.2018.090229, 2018.

5. Donald Elmazi, **Miralda Cuka**, Kevin Bylykbashi, Evjola Spaho, Makoto Ikeda, Leonard Barolli, "Implementation of Intelligent Fuzzy-based Systems for Actor

Node Selection in WSANs: A Comparison Study Considering Effect of Actor Congestion Situation", Accepted, Journal of High Speed Networks", IOS Press, Vol. 24, No. 3, pp. 187-199, DOI: 10.3233 JHS-180590, 2018.

6. Donald Elmazi, **Miralda Cuka**, Elis Kulla, Tetsuya Oda, Makoto Ikeda, Leonard Barolli, "Implementation and Comparison of Two Intelligent Systems Based on Fuzzy Logic for Actor Selection in WSANs: Effect of Node Density on Actor Selection", Accepted, International Journal of Space-Based and Situated Computing (IJSSC), Inderscience, Vol. 7, No. 4, pp. 229-238, DOI: 10.1504 IJSSC.2017.089885, 2017.

7. **Miralda Cuka**, Donald Elmazi, Elis Kulla, Tetsuya Oda, Makoto Ikeda, Leonard Barolli, "Implementation of two fuzzy-based systems for IoT device selection in opportunistic networks: effect of storage parameter on IoT device selection", International Journal Communication Networks and Distributed Systems (IJCNDS), Vol. 21, No. 1, pp. 95- 114, DOI: 10.1504 IJCNDS.2018.093400, 18 June 2018.

8. **Miralda Cuka**, Donald Elmazi, Kevin Bylykbashi, Evjola Spaho, Makoto Ikeda, Leonard Barolli, "Implementation and Performance Evaluation of Two Fuzzy-based Systems for Selection of IoT Devices in Opportunistic Networks", Journal of Ambient Intelligence and Humanized Computing, Springer, Vol. 10, No. 2, pp. 1-11, DOI: 10.1007 s12652-017-0676-0, 2018.

9. **Miralda Cuka**, Donald Elmazi, Kevin Bylykbashi, Evjola Spaho, Makoto Ikeda, Leonard Barolli, "Effect of node centrality for IoT device selection in Opportunistic Networks: A comparison study", Concurrency and Computation Practice and Experience, Vol. 30, No. 21, pp. e4790, DOI: 10.1002 cpe.4790, 24 August 2018.

10. **Miralda Cuka**, Donald Elmazi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "IoT node selection in Opportunistic Networks: Implementation of fuzzy-based simulation systems and testbed", Internet of Things, Vol. 8, DOI: 10.1016 j.iot-2019-100105, December 2019.

International Conference Papers

1. **Miralda Cuka**, Kouke Ozera, Ryoichiro Obukata, Donald Elmazi, Tetsuya Oda, Leonard Barolli, "Implementation of a GA-based Simulation System for Placement

of IoT Devices: Evaluation for a WSAN Scenario", Proc. of EIDWT-2017, Wuhan, China, pp. 34-42, June 2017.

2. Tetsuya Oda, **Miralda Cuka**, Ryoichiro Obukata, Makoto Ikeda, Leonard Barolli, "A User Prediction and Identification System for Tor Networks Using ARIMA Model", Proc. of EIDWT-2017, Wuhan, China, pp. 89-97, June 2017.

3. Ryoichiro Obukata, **Miralda Cuka**, Donald Elmazi, Tetsuya Oda, Leonard Barolli, "Implementation of Actor Node in an Ambient Intelligence Testbed: Evaluation and Effects of Actor Node on Human Sleeping Conditions", Proc. of EIDWT-2017, Wuhan, China, pp. 98-106, June 2017.

4. **Miralda Cuka**, Donald Elmazi, Tetsuya Oda, Elis Kulla, Makoto Ikeda, Leonard Barolli, "A Delay-aware Fuzzy-based System for Selection of IoT Devices in Opportunistic Networks", Proc. of CISIS-2017, Torino, Italy, pp. 3-13, July 2017.

5. Donald Elmazi, **Miralda Cuka**, Tetsuya Oda, Elis Kulla, Makoto Ikeda, Leonard Barolli, "Selection of Actor Nodes in Wireless Sensor and Actor Networks: A Fuzzy-based System Considering Packet Error Rate as a New Parameter", Proc. of CISIS-2017, Torino, Italy, pp. 43-55, July 2017.

6. Masafumi Yamada, **Miralda Cuka**, Yi Liu, Tetsuya Oda, Keita Matsuo, Leonard Barolli, "Design of a Smart Desk for an IoT Testbed: Improving Learning Efficiency and System Security", Proc. of IMIS-2017, Torino, Italy, pp. 27-35, July 2017.

7. Tetsuya Oda, Elis Kulla, **Miralda Cuka**, Donald Elmazi, Makoto Ikeda, Leonard Barolli, "Performance Evaluation of a Deep Q-Network Based Simulation System for Actor Node Mobility Control in Wireless Sensor and Actor Networks Considering Different Distributions of Events", Proc. of IMIS-2017, Torino, Italy, pp. 36-49, July 2017.

8. Donald Elmazi, **Miralda Cuka**, Tetsuya Oda, Elis Kulla, Makoto Ikeda, Leonard Barolli, "Effect of Packet Error Rate on Selection of Actor Nodes in WSANs: A Comparison Study of Two Fuzzy-Based Systems", Proc. of NBiS-2017, Toronto, Canada, pp. 114-126, August 2017.

9. **Miralda Cuka**, Donald Elmazi, Tetsuya Oda, Elis Kulla, Makoto Ikeda, Leonard Barolli, "A Fuzzy-Based System for Selection of IoT Devices in Opportunistic Networks Considering IoT Device Speed, Storage and Remaining Energy Parameters", Proc. of INCoS-2017, Toronto, Canada, pp. 6-27, August 2017.

10. Tetsuya Oda, Donald Elmazi, **Miralda Cuka**, Elis Kulla, Makoto Ikeda, Leonard Barolli, "Performance Evaluation of a Deep Q-Network Based Simulation System for Actor Node Mobility Control in Wireless Sensor and Actor Networks Considering Three-Dimensional Environment", Proc. of INCoS-2017, Toronto, Canada, pp. 41-52, August 2017.

11. Masafumi Yamada, **Miralda Cuka**, Yi Liu, Tetsuya Oda, Keita Matsuo, Leonard Barolli, "Performance Evaluation of an IoT-Based E-Learning Testbed Using Mean-Shift Clustering Approach Considering Theta Type of Brain Waves, Proc. of INCoS-2017, Toronto, Canada, pp. 62-72, August 2017.

12. Ryoichiro Obukata, **Miralda Cuka**, Donald Elmazi, Tetsuya Oda, Keita Matsuo, Leonard Barolli, "Implementation of an Actor Node for an Ambient Intelligence Testbed Considering Bed Temperature and Room Lighting: Its Effects on Human Sleeping Condition", Proc. of INCoS-2017, Toronto, Canada, pp. 73-81, August 2017.

13. Donald Elmazi, **Miralda Cuka**, Kevin Bylykbashi, Evjola Spaho, Makoto Ikeda, Leonard Barolli, "Selection of Actor Nodes in Wireless Sensor and Actor Networks Considering Actor-Sensor Coordination Quality Parameter", Proc. of BWCCA-2017, Barcelona, Spain, pp. 87-99, 2017.

14. **Miralda Cuka**, Donald Elmazi, Tetsuya Oda, Elis Kulla, Makoto Ikeda, Leonard Barolli, "Effect of Storage Size on IoT Device Selection in Opportunistic Networks: A Comparison Study of Two Fuzzy-Based Systems", Proc. of BWCCA-2017, Barcelona, Spain, pp. 100-113, 2017.

15. Masafumi Yamada, **Miralda Cuka**, Yi liu, Kevin Bylykbashi, Keita Matsuo, Leonard Barolli, "Performance Evaluation of an IoT-Based E-Learning Testbed Using Mean-Shift Clustering Approach Considering Gamma Type of Brain Waves", Proc. of BWCCA-2017, Barcelona, Spain, pp. 671-681, 2017.

16. **Miralda Cuka**, Donald Elmazi, Kevin Bylykbashi, Evjola Spaho. Makoto Ikeda, Leonard Barolli, "A Fuzzy-Based System for Selection of IoT Devices in Opportunistic Networks Considering IoT Device Storage, Waiting Time and Security Parameters", Proc. Of EIDWT-2018, Tirana, Albania, pp. 94-105, 2018.

17. **Miralda Cuka**, Donald Elmazi, Kevin Bylykbashi, Evjola Spaho, Makoto Ikeda, Leonard Barolli, "A Fuzzy-based System for Selection of IoT Devices in Opportunistic Networks Considering IoT Device Storage, Waiting Time and Node Centrality Parameters", Proc. Of AINA-2018, Krakow, Poland, pp. 710-716, 2018.

18. **Miralda Cuka**, Donald Elmazi, Keita Matsuo, Makoto Ikeda, Leonard Barolli, "A Fuzzy-Based System for Selection of IoT Devices in Opportunistic Networks Considering IoT Device Contact Duration, Storage and Remaining Energy" Proc. Of. IMIS-2018, Matsue, Japan, pp. 74-85, 2018.

19. **Miralda Cuka**, Donald Elmazi, Keita Matsuo, Makoto Ikeda, Leonard Barolli, "A Delay-Aware Fuzzy-based System for Selection of IoT Devices in Opportunistic Networks", Proc. Of NbiS-2018, Bratislava, Slovakia, pp. 16-29, 2018.

20. Donald Elmazi, **Miralda Cuka**, Makoto Ikeda, Leonard Barolli, "A Fuzzy-based System for Actor Node Selection in WSANs for Improving Network Connectivity and Increasing Number of Covered Sensors", Proc. Of NbiS-2018, Bratislava, Slovakia, pp. 3-15, 2018.

21. **Miralda Cuka**, Donald Elmazi, Kevin Bylykbashi, Keita Matsuo, Makoto Ikeda, Leonard Barolli "A Fuzzy-Based System for Selection of IoT Devices in Opportunistic Networks Considering Number of Past Encounters", Proc. Of 3PGCIC-2018, Taichung, Taiwan, pp. 223-237, 2018.

22. Donald Elmazi, **Miralda Cuka**, Makoto Ikeda, Leonard Barolli, "A Fuzzy-Based System for Actor Node Selection in WSANs Considering Load Balancing of Actors", Proc, Of BWCCA-2018, Taichung, Taiwan, pp. 97-109, 2018.

23. **Miralda Cuka**, Donald Elmazi, Keita Matsuo, Makoto Ikeda, Leonard Barolli and Makoto Takizawa, "IoT Device Selection in Opportunistic Networks: A Fuzzy Approach Considering IoT Device Failure Rate", Prof. Of. EIDWT-2019, Fujairah, UAE, pp. 39-52, 2019.

24. Donald Elmazi, **Miralda Cuka**, Makoto Ikeda, Leonard Barolli and Makoto Takizawa, "A Fuzzy-Based System for Selection of Actor Nodes in WSANs Considering Actor Reliability and Load Distribution", Prof. Of. EIDWT-2019, Fujairah, UAE, pp. 25-38, 2019.

25. **Miralda Cuka**, Donald Elmazi, Makoto Ikeda, Keita Matsuo, Leonard Barolli and Makoto Takizawa, "Selection of IoT Devices in Opportunistic Networks: A Fuzzy-Based Approach Considering IoT Device's Selfish Behaviour", Prof. Of. AINA-2019, Matsue, Japan, pp. 251-264, 2019.

26. Donald Elmazi, **Miralda Cuka**, Makoto Ikeda, Leonard Barolli and Makoto Takizawa, "A Fuzzy-Based System for Actor Node Selection in WSANs Considering Level of Received Signal", Proc. Of. AINA-2019, Matsue, Japan, pp. 238-250, 2019.

27. **Miralda Cuka**, Donald Elmazi, Makoto Ikeda, Keita Matsuo and Leonard Barolli, "IoT Node Selection and Placement: A New Approach Based on Fuzzy Logic and Genetic Algorithm", Proc. Of. CISIS-2019, Sydney, Australia, pp. 22-35, 2019.

28. Donald Elmazi, **Miralda Cuka**, Makoto Ikeda, Keita Matsuo and Leonard Barolli, "A Fuzzy-Based System for Actor Node Selection in WSANS: Simulation and Experimental Results", Proc. Of. CISIS-2019, Sydney, Australia, pp. 11-24, 2019.

29. **Miralda Cuka**, Donald Elmazi, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "A Fuzzy-based Simulation System for IoT Node Selection in Opportunistic Networks and Testbed Implementation", Proc. of. BWCCA-2019, Antwerp, Belgium, pp. 32-43, 2019.

30. Donald Elmazi, **Miralda Cuka**, Makoto Ikeda, Keita Matsuo, Leonard Barolli, "Effect of Degree of Centrality Parameter on Actor Selection in WSANs: A Fuzzy-based Simulation System and Its Performance Evaluation", Proc. of. 3PGCIC-2019, Antwerp, Belgium, pp. 35-46, 2019.