Enabling seamless multi-RAT networking for heterogeneous mesh networks

Ruben Smink
s1965891
r.b.smink@student.utwente.nl

***Abstract*—The rise of Internet of Things (IoT) and mesh networks has revolutionized wireless communication systems, offering resilience and adaptability for a variety of applications. However, mesh networks face a persistent challenge in their ability to cooperate with one another. One form of this lack of cooperation, is the inability of incompatible mesh networks to seamlessly leverage each other's network connections for the delivery of their packets. This thesis introduces a novel approach that utilizes multi-Radio Access Technology (multi-RAT) to address this issue. Multi-RAT, with its ability to allow nodes to communicate over various mesh networks concurrently, redefines the operation of mesh networks, enhancing connectivity, efficiency and adaptability. By using incompatible mesh networks as a bridge between two compatible mesh networks, it can enable communication between two separate, but compatible mesh networks where it was previously impossible. Furthermore, the multi-RAT nodes can enable more reliable and faster communication within an already fully connected mesh network. In the process, redundant nodes and inefficient paths within these mesh networks can be eliminated. The group of mesh networks that is connected through multi-RAT nodes, will henceforth be called multi-RAT networks. Within this multi-RAT network, its multi-RAT nodes will send packets through entire mesh networks instead of one connection between nodes. Discovering optimal routes within the multi-RAT network is done through an Ant Colony Optimization algorithm, as routing packets through entire networks introduces additional variables and considerations. This thesis reviews related work, discusses design choices, technical challenges and the results of implementing such a multi-RAT approach in the open-source discrete-event network simulator ns-3[1]. Additionally, potential future additions to this work are explored.**

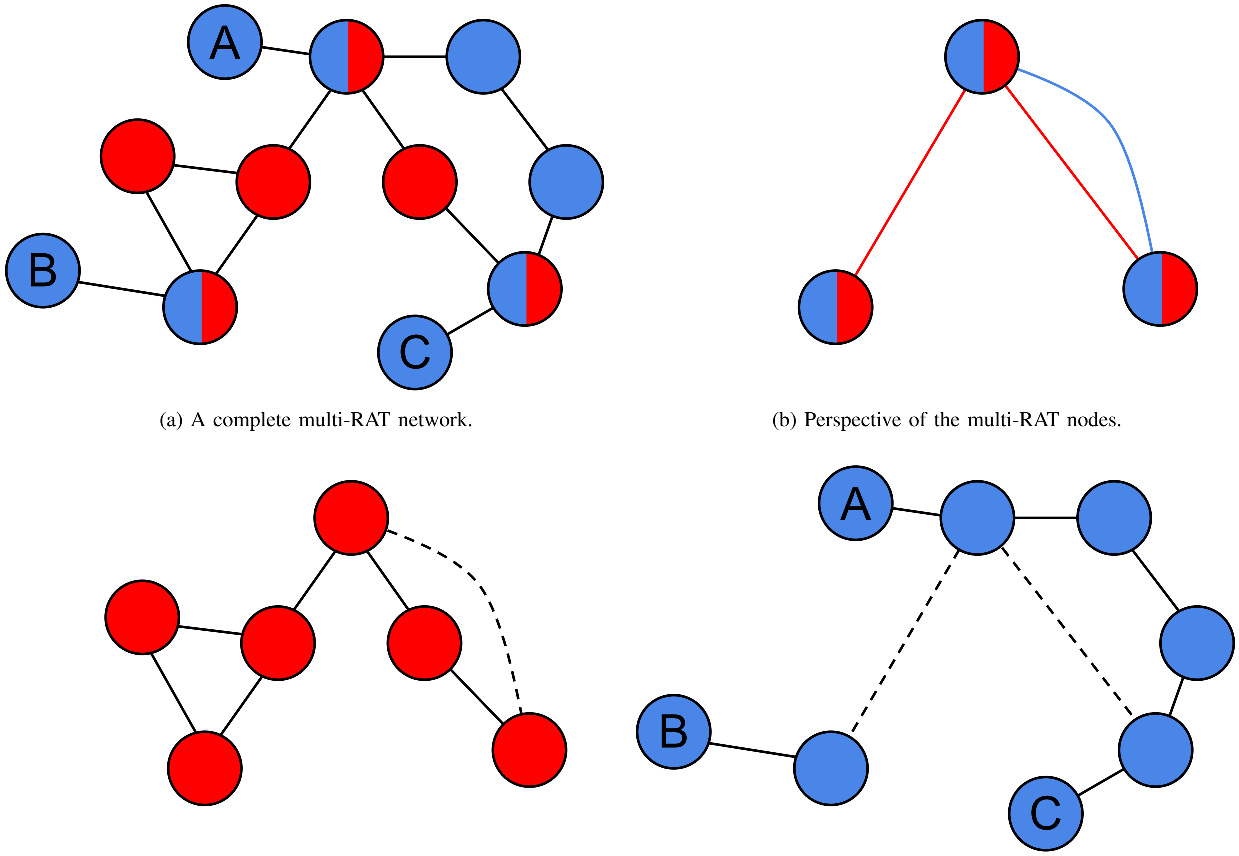
I. Introduction

The current landscape of wireless communication systems has been transformed by the rise of IoT and mesh networks. These technologies have revolutionized the way we collect and spread information. Mesh networks, inherently resilient and flexible, have become an interesting choice for various applications, from smart homes to industrial automation. However, despite their numerous advantages, mesh networks all essentially operate as isolated islands of connectivity. These isolated islands hinder the seamless exchange of data, leading to inefficiency and redundancy. In [1], it is suggested that managing these mesh networks in a multi-RAT scenario could improve their performance. This scenario would include mesh networks that primarily consist of nodes without multi-RAT capabilities. These nodes are unable to connect to different types of mesh networks due to factors such as varying protocols and frequency ranges, However, a small portion of the nodes in these mesh networks do possess multi-RAT capabilities. Due to the design of multi-RAT nodes. They are able to use multiple Radio Access Technologies, enabling connections to multiple different mesh networks. With this capability, they are able to function as bridges between compatible networks by facilitating the hand over of packets to and from incompatible mesh networks. An example of this type of network can be seen in Figure 1a which illustrates a network characterized by two distinct radio access technologies, each differentiated by their color. The multi-RAT nodes in this Figure are recognizable by their dual color scheme.

Through the management of these mesh networks, multi-RAT has the potential to completely change the way mesh networks operate and interconnect. Current implementations of multi-RAT often focus on end-user devices and the concurrent use of technologies such as WLAN and LTE, as discussed in [1]. However, we are going to be focusing on the unique features it offers that allows for increased connectivity, efficiency, and adaptability within mesh networks such as WLAN, BLE and Zigbee. This thesis introduces a novel approach where the ability of multi-RAT nodes to communicate to two or more different mesh networks simultaneously is harnessed. Using them to manage the mesh networks they are connected to and provide the ability for these networks to seamlessly utilize the full potential of multi-RAT networking without the mesh networks themselves having to be aware of it.

In the subsequent sections of this paper, we initially delve into the research problem and challenges in sections II and III. Here we discuss the problem we want to solve, the proposed solution and the challenges associated to it. Then we discuss the research questions section IV. In section V we discuss related work such as the similarly functioning routing for multi-modal transportation and the potential it holds for mesh network communication. Following this, we review the design choices for the creation of a multi-RAT network in section VI. Here, we are going to discuss the technical challenges and considerations that arise with creating the solution. In section VII we discuss the implementation of this solution in a simulation, ensuring a comprehensive understanding of the practical aspects of this concept. We then present the results and discuss them in sections VIII and IX. Finally, we discuss potential future additions in section X and provide a conclusion in section XI.

[1] https://www.nsnam.org/



(a) A complete multi-RAT network.

(b) Perspective of the multi-RAT nodes.

(c) Perspective of the blue network.

(d) Perspective of the blue network.

Fig. 1: Different perspectives of a multi-RAT network where color determines the type of mesh network.

II. Problem statement

As it stands, mesh networks are often unable to cooperate with each other due to their incompatibilities. Even when adding multi-RAT nodes to mesh networks and thereby creating a multi-RAT network, the mesh networks would still not be able to make use of the advantages it provides without any help. In Figure 1a we have a multi-RAT network with three marked nodes. This is going to show us two important examples where multi-RAT nodes can provide improvements to the mesh networks. First of all, there is the connection between node A and B. These nodes are both part of the blue network. However, they are not connected and cannot send packets to each other. If were possible to utilize the red network, the two nodes would be able to send packets to each other. Secondly, there is the connection between node A and C. While these nodes are able to connect to each other entirely through the blue network, there is a better path available through the red network. These improvements are usually never made as the multi-RAT nodes have separate non routing protocols for the different mesh networks they connect to.

The solution we propose is focused around the central role of multi-RAT nodes in controlling the cooperative functioning of the surrounding networks, aiming to allow seamless cooperation among networks. The multi-RAT nodes determine whether a packet should be routed through its original mesh network or if a better performing route is available through utilizing multi-RAT, all without there being a necessity of an awareness for these mesh networks of their cooperation. Managing a multi-RAT network exclusively through its multi-RAT nodes primarily offers two advantages. Firstly, this approach allows the conversion of existing networks into multi-RAT configurations with relative ease, merely requiring the addition of multi-RAT nodes to the network. Secondly, it affords networks the freedom to continue utilizing their original functionalities and routing mechanisms, thereby preserving their unique functionality for the devices they serve, such as how BLE and Zigbee are specifically made to be energy efficient for battery-powered devices. However, it is important to recognize that adopting this approach to create a multi-RAT network also carries two significant drawbacks. Firstly, limited control over nodes within a network may occasionally lead to sub-optimal routing decisions within the network. Secondly, leaving the task of enabling the functionality of a multi-RAT network solely to the multi-RAT nodes is going to lead to complex multi-RAT nodes.

The solution is going to be realized by constructing a new graph of the multi-RAT network where multi-RAT nodes are the nodes and the mesh networks that connect them are their edges. This graph is given in Figure 1b. In this graph, information such as the characteristics of the mesh network and the number of hops between the two nodes can be added to the edges. If it is determined that a packet should be routed utilizing multi-RAT, this same graph is used by a routing algorithm to route the packet the multi-RAT network. Because this solution only requires the multi-RAT nodes to function, the red and blue networks do not need to be aware of it. This allows us to freely add multi-RAT nodes to existing mesh networks to enable multi-RAT capabilities. The perspective the red and blue mesh networks can be seen in figures 1c and 1d respectively. In these figures, striped edges are given for the connections that multi-RAT enables. These paths are only utilized when the multi-RAT nodes decide it improves a packet's route. For example, assuming both networks have the same characteristics, in Figure 1d, the striped edge on the right represents one red node, which is faster than crossing two blue nodes. Here the multi-RAT connection should be used. On the other hand, in Figure 1c, the striped line represents two blue nodes, which is slower than crossing one red node. Here the multi-RAT should not be utilized.

III. Challenges

The creation of a solution capable of facilitating multi-RAT networking presents quite the challenge, with various problems to be tackled to create a successful implementation. When looking at the current state of the art Multi-RAT routing, many of these works are primarily focused on the collaboration between LTE and WLAN technologies. The solutions often leverage the strengths of cellular networks and Wi-Fi for data transmission, ensuring seamless connectivity in various scenarios for the end users. An example of this is [2] where technologies such as LTE, WiMAX and Wi-Fi are mentioned. However, the focus of these papers does not fit as well when we shift our focus to IoT devices. For many of these types of devices it is simply not possible to use high power usage technologies such as LTE as they are often battery powered, requiring more efficient radio access technologies to preserve battery life. Another concern associated with the state of the art multi-RAT solutions is their primary emphasis on the end user experience within in the network. The scenarios often revolve around granting end users the ability to utilize multi-RAT capabilities without the networks themselves being able to utilize it. However, our primary objective centers on integration of multiple mesh networks, with a focus on the important role that will be played by the multi-RAT nodes that facilitate the connection between mesh networks. This shift from focusing on the network itself instead of the end user brings along some challenges. Here we list the ones that need to be solved to realize this solution.

A. Building a multi-RAT graph

To be able to build the multi-RAT graph as shown in figure 1b, multi-RAT nodes require the ability to periodically send each other updates on their status. This provides required information for the graph such as the utilization of a network.

B. Discovery

Multi-RAT nodes are not able to preemptively decide which packets will benefit from multi-RAT routing. The benefit is different for every combination of source and destination within the mesh network. Therefore, whenever discovery is initiated in a mesh network, the multi-RAT nodes have to perform their own discovery. This discovery determines whether packets from this source and destination combination should have their packets routed through multi-RAT. This includes determining to which multi-RAT node the packets should be sent for the best performance.

C. Controlling the network

Once a multi-RAT node determines that there is a better performing path for a packet by utilizing multi-RAT, it will need to actually route the packet through this new path. However, the goal is for mesh networks to not be aware of the activities of the multi-RAT nodes. Because of this, a method needs to be devised for the multi-RAT nodes to manipulate their mesh networks to send packets through the multi-RAT path.

D. Routing

In order to correctly route packets between multi-RAT nodes a routing protocol is needed. The main issue for routing within a multi-RAT network is that the options for paths within the multi-RAT graph grows rapidly as more mesh networks and multi-RAT nodes are added. This is because the multi-RAT graph is a Directed Multi graph where nodes can have as many edges between them as the amount of mesh networks types. Additionally, every edge holds multiple pieces of information about the network it represents. Using these pieces of information, the routing algorithm could end up having to optimize more than one value, such as the packet delivery time and the reliability of the connection.

IV. Research question

The goal of this thesis is to explore how effective a multi-RAT network controlled by its multi-RAT nodes is at enabling seamless networking between mesh networks. To fully answer this question, different configurations of mesh networks should be tested. This includes different layouts of the nodes in the mesh network, data rates, ranges and traffic.

1) how well can a multi-RAT network controlled by its multi-RAT nodes effectively enable seamless networking between mesh networks?
    a) To what extent can a multi-RAT network enable communication between nodes in two separate, but compatible mesh networks?

b) What impact does a multi-RAT network have upon communication between nodes within a fully connected mesh network?
   i) To what extent do different mesh network configurations impact the effect of multi-RAT networks
   ii) How well can multi-RAT networks respond to changes within mesh network configurations

In order to measure the impact of multi-RAT on a mesh network, two simulations are created with the exact same network configuration for its main network. However, the second simulation is going to have a second mesh network and several multi-RAT nodes which enable multi-RAT capabilities. Traffic is then generated in the main network, of which the reliability and delay are measured. For question 1a, we can make nodes send packets to each other that would not reach their destination in the normal mesh network. We can then connect these nodes through an additional mesh network and multi-RAT nodes and test it again in the second simulation. We then look at the reliability and delay of the traffic to see how well this new connection performs. For question 1(b)i, we generate two fully connected mesh networks with different configurations and add several multi-RAT nodes. This allows us to test the effect that the multi-RAT network has on traffic that can reach its destination in a normal mesh network. Traffic is again generated in the main network, of which the reliability and delay are measured. These metrics can then be used to determine the impact of multi-RAT to the generated mesh network configuration. Lastly, for 1(b)ii We create a multi-RAT network such as in 1a with two additional mesh networks instead of one. One of these mesh networks is going to have a higher data rate and is considered a better path for multi-RAT routing. We can then generate congestion on the better path to slow it down and lower its reliability. This allows us to measure how effective the multi-RAT network is at responding to these sorts of changes.

V. Related Work

While there is not much work directly related to this specific subject, there are subjects that are similar enough to our problem that we can learn from. We are going to go going over three subjects that hold relevant information which can be used to solve the challenges this solution faces.

A. Multi-modal transportation

Multi-modal transportation is the act of moving goods using at least two different methods of transportation to reach their destination. For example, a package could travel the first half of its journey through a container ship and the second half through a cargo truck. This is quite similar to the usages of multi-RAT to send a message from a source to its destination over multiple different mesh networks. This means that these studies could offer valuable insights and methodologies that can in turn be applied to the challenges faced within multi-RAT networks. Here, we review potentially interesting works in this domain.

One noteworthy study [3] has harnessed the power of deep neural network to improve upon the classic Dijkstra algorithm in route planning for multi-modal transport. This innovative work combines multiple transportation methods into a unified, overarching graph, wherein routes contain a multitude attributes that allow for more precise control of the planning in place of what would in most cases be a simple cost considerations. Beyond optimizing operational efficiency and storage capacity of shipments, the study introduces an interesting addition by introducing the concept of maximum delays. This addition could potentially be useful for a multi-RAT implementation as there are many applications that require timely delivery of messages and thus, could particularly benefit from a restriction like this.

In parallel, another contribution [4] adopts Q-learning, a model-free reinforcement learning methodology, to confront the challenges posed by multi-modal transportation. While the paper is based on the aspect of transportation, the core principles of this approach show large adaptability to the ever changing landscapes of dynamic network environments. One issue with a model-free methodology such as this one is the requirement of solving infinite loops. In this paper the deployment of directed graphs was used to ensure the trajectories of learning culminate at a terminal state. In scenarios where there are multiple connections between two nodes, fictitious intermediary nodes are introduced to facilitate the agent's decision-making process.

Our final paper pertaining to multi-modal transportation, is a study deploying an Ant Colony Optimization algorithm in order to solve the complex path finding problem posed by this subject in a timely manner [5]. Its ability to optimize both the total cost and delay of a route through multi-objective optimization aligns well with the objectives of optimizing network communication within a multi-RAT environments. The study shows that the ant colony algorithm outperforms traditional optimization techniques. An interesting addition to this study is the inclusion of data-driven techniques, reducing the need for running the simulation by storing previous routes and the utilization of Support Vector Regression in order to lighten the computational cost of the algorithm. This is a variant of a Support Vector Machine. However, instead of the usual binary classification that an SVM performs, it is used for regression purposes instead.

B. Multi-path routing

An interesting idea in the narrative of multi-path routing is discussed in [6] where Multi-Agent Reinforcement Learning is applied. The paper proposes to plan a path whereby every node along said path acts as its own separate agent, able to take decisions concerning the splitting of data flows. Interestingly, the method in which the agents are trained is through a Software-Defined Network. This allows them to train an individualized model for each network node. These nodes subsequently operate relatively autonomously, after this initial training, input from the SDN network is no longer required. However, in the case of this study, the SDN network still provides occasional network status updates to aid the agents in their decisions.

C. Hyper graphs

This model introduces the idea of multiple communication channels represented as layers, with nodes that can switch between those layers. This is shown in [7] where a network model based on multi-layer hyper-graphs is proposed to improve throughput in full-duplex cognitive radio networks. While this work primarily revolves around the usage of multiple channels, its concepts can be applied to multi-RAT networks. This approach accommodates the complexities of licensed channels and jamming, providing potential solutions to dynamic network conditions. In this hyper graph a simplifications is made to reduce the complexity of the network and in turn reduce the computation required by grouping up physically close nodes into a hyper node.

VI. Methodology

In order to manage a multi-RAT network, the multi-RAT nodes require the capability to interact with their mesh network and facilitate the delivery of packets over the multi-RAT network. Because this process is different for different types of mesh networks. We are focusing on the 802.11s standard for mesh networking using HWMP (Hybrid Wireless Mesh Protocol), which is a layer 2 routing protocol, in reactive mode [8]. This means that nodes will not attempt to discover destinations until they attempt to send packets to it.

We are starting off by walking through the steps required to enable the multi-RAT network. First, mesh networks with multi-RAT nodes are generated. The multi-RAT nodes in these mesh networks communicate with each in other to create a multi-RAT graph as shown in 1b, which holds information about the mesh networks and multi-RAT nodes. This graph is used for filling the routing table in each multi-RAT node using the routing algorithm. When all of this is finished, the multi-RAT nodes wait for ARP path requests. If a multi-RAT node determines there is a multi-RAT route available, it notifies the source node of the ARP path request using an ARP path reply. This causes the source to send its packets through the multi-RAT node. When a packet with a valid multi-RAT route is received by the multi-RAT node, it facilitates its delivery to the destination. In the rest of the chapter, these step expanded upon.

A. Generating networks

In order to actually generate random mesh networks to test the multi-RAT nodes with, we use Perlin noise [9]. This method of generation was chosen because of its ability to consistently create clusters of nodes with a lot of variety. There are some restrictions on the networks during creation to ensure a good outcome. Firstly, a main network is generated that will need to have 30 nodes or more. This is done to allow plenty of applications that are going to send packets over the network. Secondly, the second network and onwards will at least need to overlap with the main network the same amount of times as the amount of multi-RAT nodes to be added to the simulation. This is to ensure the multi-RAT nodes have valid locations to be placed down. When all the mesh networks are generated, the multi-RAT nodes are placed. This process picks random locations and combines all nodes at that location into a multi-RAT node. this multi-RAT node is compatible with all the mesh networks that the nodes were taken from. This is repeated until the desired amount of multi-RAT nodes is reached. Lastly, it is checked whether each individual mesh network is fully connected and if through the multi-RAT nodes mesh network together. This is to ensure we do not have situations where applications attempt to transmit to nodes that are not connected in the normal mesh network, but are connected in the multi-RAT network. If this were to happen it would skew reliability results towards the multi-RAT network because only the multi-RAT network would be able to successfully deliver packets in this situation. After generating the networks, we can simply select which multi-RAT networks are interesting by generating many of them and looking at their characteristics.

B. Communication with the network

In order to successfully send a packet through a mesh network, there are four layers of communication that are important: physical, data link, network and transport. First, the packet is handled by the transport layer, which adds its header to the packet and sends it to the network layer. The network layer is handled by ARP (Address Resolution Protocol), which attempts to find the corresponding MAC address for the destination IP address of the packet using an ARP path request packet. The difference in a mesh network compared to usual is that the data link layer has its own routing protocol, the HWMP protocol. This protocol is able to simplify the mesh network to the layers above it by making it seem to ARP that all the nodes within the mesh network are connected to one switch.

When adding multi-RAT nodes to a mesh network, we can make use of this disconnect between the ARP and the lower layers to send our own custom ARP messages through the network. We can even add additional headers as long as we ensure that these packets are not sent to upper layers. We can avoid this by using control messages such as a path request or path reply.

C. Discovery

In order to start any kind of routing within the multi-RAT network, discovery is going to have to be performed by the multi-RAT nodes to collect information about its connection to other multi-RAT nodes. Periodically, a broadcast packet is sent by the multi-RAT node on each of its mesh networks. This packet is an ARP path request for the broadcast IP of its network. As this is the broadcast IP, non of the nodes within the mesh network will have this IP and it will be ignored. The IP address of the sender and its MAC address are already given fields in the ARP request. However, an additional header is added to store the node ID of the source and the time of sending. From this packet, several pieces of information can be extracted. Such as, the time to live, which gives information about the amount of hops between the two multi-RAT nodes. The reliability of the connection can be measured through a moving average. Since we know that the interval of the discovery packet is 5 seconds, we can update the reliability based on how much extra delay there is between two discovery packets. If no discovery packets are received for an extended amount of time, the reliability of the connection is gradually reduced after a period of 10 seconds. Using this information, we are able to construct the multi-RAT graph. This is done through accessing all the multi-RAT nodes directly and taking their status information on every connection with another multi-RAT node. This only works within simulations. However, it greatly reduces the amount of work required to make a fully fledged routing algorithm.

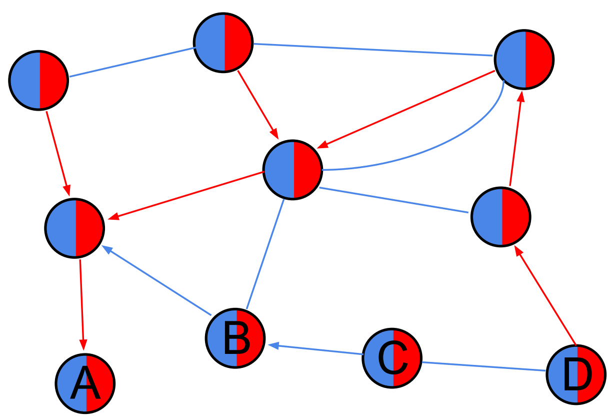


Fig. 2: Result of an ant networking creating routes towards node 1.

D. Routing

Multi-RAT routing introduces added complexities compared to routing within a single network, primarily due to the differences between the characteristics of the mesh networks such as the aforementioned ranges and data rates. For instance, certain radio access technologies, such as ZigBee, are ill-suited for the transportation of substantial data loads, given their low data rate. In light of these diverse mesh network characteristics, it is important to implement a routing protocol made for accommodating the various characteristics present within the edges of our multi-RAT graph. As discussed in Section V, routing protocols employed in multi-modal transportation systems tackle exactly this issue. Consequently, we propose a solution inspired by [5].

We leverage an Ant Colony Optimization algorithm to establish optimal routes from any multi-RAT node in the multi-RAT network to any destination multi-RAT node. The optimal route within the multi-RAT network towards a multi-RAT node will not only depend on the characteristics of the mesh networks themselves, but also the connections between the nodes in the network that facilitate the connection between the two multi-RAT nodes. When the Ant Colony Optimization algorithm is finished, routes emerge from each multi-RAT node to a predetermined destination. In [10] the time complexity of this algorithm is given as $O(wkn^2)$, where n = the number of nodes, k = the number of ants and w = the the rate of the Ant Colony Optimization algorithm implementation's convergence. However, since we are using a simple implementation, both k and w are constants at k = 100000 and w = 1. As an example of the algorithm, in figure 2, whereas before an edge represents an entire mesh network. The figure shows each edge as an arrow if it used in the route and a line if it is an unused connection.

Here, utilizing the blue network is in most cases avoided because of its low data rate. This is especially apparent in the route from node D to node A. However, for nodes B and C it becomes optimal to use blue networks as there is either no way to reach node A without utilizing blue networks or the choice of using one or more blue networks to send the message becomes optimal when too many red networks are required to be crossed to avoid using a blue network.

For the calculation of edge cost, we are using the amount of hops the discovery packet has made to reach the destination multi-RAT node divided by the data rate in Mbps of the mesh network this edge belongs to. This will give us a good measurement of how well an edge compares to other edges in performance. This number is then divided by the reliability squared of the discovery packets arriving at its destination. This is done to avoid badly performing networks due to either high traffic or simply a bad connection between multi-RAT nodes. We square this value because mesh networks with very bad reliability should only be taken as a measure of last resort. Lastly, to avoid taking an unnecessarily long path and represent queue times, an additional constant α is added. This value is left at α = 0.1 during all the simulations in this thesis. The calculation is shown in formula 1.

$$cost = \frac{\frac{hops}{datarate}}{reliability^2} + \alpha \quad (1)$$

E. Delivering packets

When a multi-RAT node has the required information to be able to deliver packets over the multi-RAT network, it starts listening for ARP path requests. Whenever it receives one, it initiates a multi-RAT path request. It iterates over all known multi-RAT nodes and sends out a multi-RAT path request if they are compatible with the mesh network that the original ARP path request originates from. These multi-RAT path requests are routed using the routes calculated by the Ant Colony Optimization algorithm. Both the compatible multi-RAT nodes and the paths to them are known ahead of time from the discovery procedure. Because we do not have any control over other nodes in the original mesh network, the multi-RAT path request has to compete with the original ARP path request. This is difficult feat as ARP path requests are broadcast to nearby nodes without much delay. In order to keep a multi-RAT path request competitive with the ARP path request with the original network, we use the original ARP path request in combination with an additional header for the node ID of the source and the node ID of the destination. This added information causes the multi-RAT path request to be slightly slower than the original ARP request if the total amount of hops and the characteristics of both mesh networks are the same.

While reusing ARP requests in the second network significantly increases speed, it assumes there are no nodes with the same IP address in the second network. While this is usually not the case, it is possible that the multi-RAT path goes through the original network and causes the final destination node to send an ARP path reply. However, this is not really a problem, other than causing slightly more overhead due to the extra ARP path reply. When a multi-RAT node receives a path request from another multi-RAT node, it is detectable by the packet size being larger than the usual ARP request. This allows the multi-RAT node know that the packet is a multi-RAT path request and causes the node to read the additional header for the destination node ID and forward it based on its routing table.

When the multi-RAT path request is received by the destination multi-RAT node, the request is sent out into the compatible network. However, in order to receive an answer from the destination node, the source MAC address is changed to that of the multi-RAT node and the additional header is removed. To maintain the required information for sending a multi-RAT path reply to the initial multi-RAT node, information about the request is stored for a certain amount of time or until a reply arrives.

When an ARP path reply is received by a multi-RAT node, we first check the stored request information to ensure that this multi-RAT node sent an ARP path request for it. When this information is confirmed, the multi-RAT node sends a multi-RAT path reply to the multi-RAT node that originally sent the request. The procedure is similar to the multi-RAT path request, where an additional header is added for the source node ID and destination node ID for the multi-RAT nodes. The forwarding procedure is also the same, as ARP path replies are detected as being from a multi-RAT node through their increased packet size.

Finally, when the multi-RAT path reply is received by the destination multi-RAT node, the source multi-rat node ID is saved together with the IP address of the destination, this way the multi-RAT node is able to send any future packets with data to the correct multi-RAT node. The ARP response is then modified to include the MAC address of the multi-RAT node as the source address to ensure future packets are sent to the multi-RAT node and the additional header is removed. Sending out this ARP path reply triggers the original node to start sending its packets through the multi-RAT connection. These packets receive the same additional header with the multi-RAT node source ID and destination ID and are sent from multi-RAT node to multi-RAT node using the same path as the ARP packets until they reach their destination multi-RAT node. The additional header is then removed and the packet is sent to its destination node.

VII. Implementation

The open-source discrete-event network simulator ns-3 plays a fundamental role within our research methodology. This tool has shown to be versatile for creating a wide range of network simulations, serving an important role in the development, evaluation, and analysis of a variety of networking protocols, algorithms, and systems. This simulator, which is primarily written in C++, offers a flexible and extensible platform for the simulation, modeling, and exploration of complex network scenarios. It is used to simulate and assess the performance of the multi-RAT design. However, it is important to note that the simulator, in its current state, lacks the capability to simulate a multi-RAT network as described earlier. Therefore, it requires extensions to tailor the simulator to our specific requirements.

A. Nodes

In order to create a network in ns-3, four main components are necessary. Nodes, Applications, Net Devices and channels. In ns-3, nodes represent the abstraction of basic computing entities and could be viewed as nodes inside of a network graph. While these nodes are not going to be of any use on their own, they serve as containers for applications and Net Devices, which in turn will add functionality to these nodes. In Figure 3 we can see a depiction of the structure of these nodes.

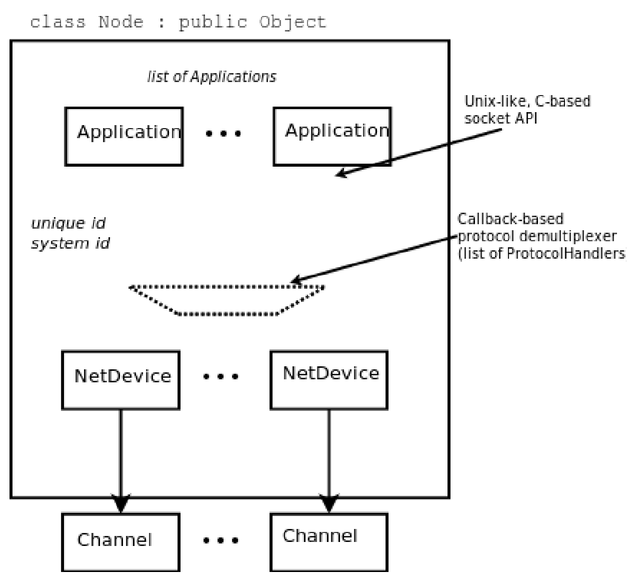
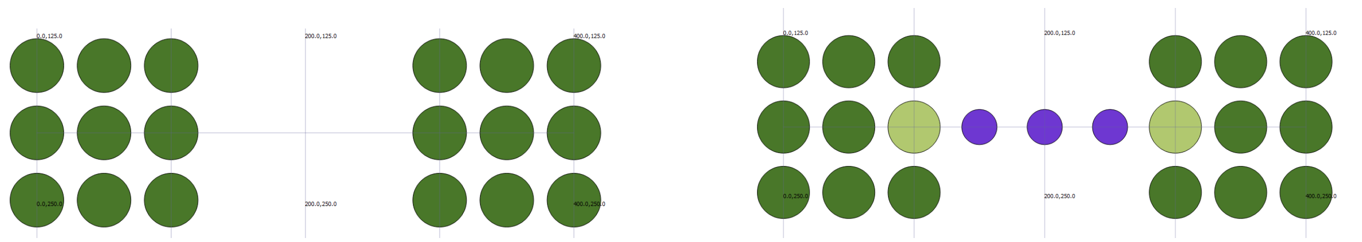


Fig. 3: Visual representation of a node within ns-3[2].

B. Net Devices

Net Devices serve as the link between a node and a network, similar to the role of applications. A node can house multiple Net Devices as seen in Figure 3. However, it is important to note that for connecting a multi-RAT node to multiple

[2]https://www.nsnam.org/docs/release/3.10/manual/html/nodes-and-devices-overview.html



(a) Two clusters of nodes that cannot reach each other.

(b) Network from 4a with an additional network and multi-RAT nodes.

Fig. 4: An example of how the multi-RAT solution is able to connect two networks together.

networks, at least two or more Net Devices are required. Each Net Device can hold a channel object. These objects facilitate communication between Net Devices of neighboring nodes, assuming they share the same channel object. Therefore, we can create distinct networks that are isolated from one another by using several channel objects in a simulation.

C. Applications

Every node in ns-3 has the capacity to accommodate one or more applications, as seen in Figure 3. These are required for executing tasks within the network. These applications allow nodes to perform a diverse variety of actions and could be seen as the actors inside of the simulation. There are already several applications that come with ns-3. For example, an echo client and server, that are able to send and receive messages through the Net Devices that are connected to the node.

Importantly, we are implementing a custom application that is designed to perform the brunt of the work in making the simulation align with our design. This custom multi-RAT applications is then installed on several nodes within the network, turning them into multi-RAT nodes which regulate the communication over several networks in the multi-RAT network

D. Network

While ns-3 does not officially support common IoT radio access technologies such as: BLE and Zigbee. It does provide an implementation of the 802.11s standard for mesh networking [11]. This standard operates on the second layer and ensures all nodes on the mesh can communicate with each other. Additionally, it provides substantial flexibility for configuring networks to mimic the characteristics of other technologies. This configurability enables us to create mesh networks with a variety of characteristics, which can then be connected together using multi-RAT nodes. However, while the standard allows various routing protocols, for simplicity's sake, we are using the default implementation of the HWMP routing protocol. In order to still create networks with various characteristics, we adjust the data rate and range of nodes within these networks.

VIII. Results

A. Results for research question 1a

Using the given method, we can now deploy multi-RAT nodes within mesh networks in the simulation to test their effectiveness. To start off, in Figure 4a we can see two clusters of node that are all the same heterogeneous mesh network. However, the nodes on the left are too far from the nodes on the right to effectively communicate with each other.

We can now create a bridge between these two clusters using a second network and multi-RAT nodes. In Figure 4b we add a second mesh network using purple nodes, which forms a bridge between the two clusters. These two networks are then connected to each other through multi-RAT nodes indicated by their light green color. This newly created multi-RAT network now allows the left and right clusters to communicate as if they are in the same mesh network by sending their packets through the purple network.

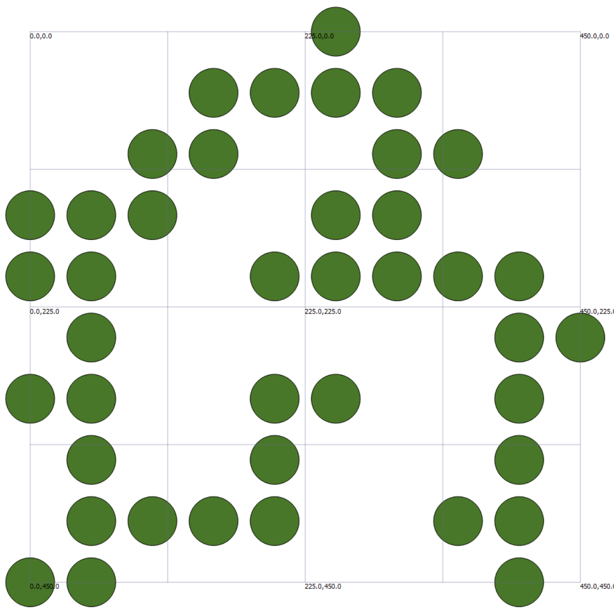
1) Performance:

B. Results for research question 1(b)i

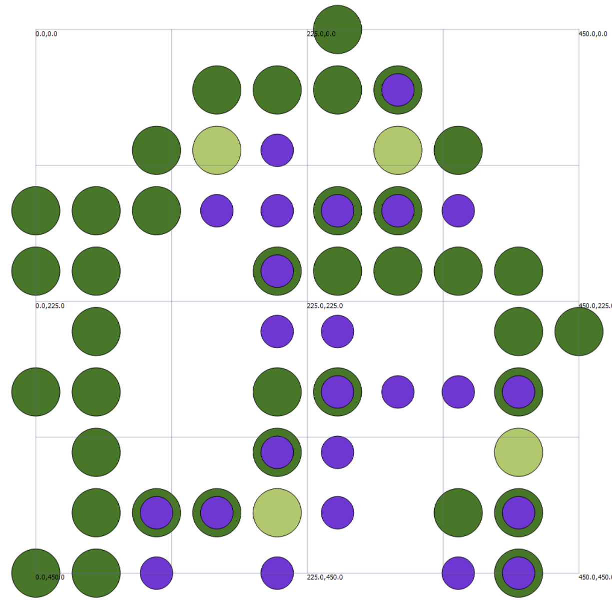
The same principle as in the previous mesh network can be applied to randomly generated networks. We take a generated network such as in Figure 5a and combine it with a second mesh network. Lastly, we replace a certain amount of nodes with multi-RAT nodes. In this case we add four multi-RAT nodes. The resulting network can be seen in Figure 5b. During a simulation, several multi-RAT node locations are cycled through.

In this network we randomly place a UDP client and server pair only on the green network and have them continuously send packets. From these packets we measure the reliability of the connection by counting the amount of sent and received packets. We also attach a tag to the packets with the time of sending. This tag is separate from the packet and does not affect the size of the packet. When the packet is delivered the time is read and used in the calculation of the average packet delivery time.

Using this method, the average packet delivery time of the network in Figure 5a comes out to be 16.7 milliseconds. In Table I we can see the performance of the multi-RAT network with different characteristics for the purple network. The green network has the same data rate and range in all of these simulations of 9Mbps and 60 meters respectively. We

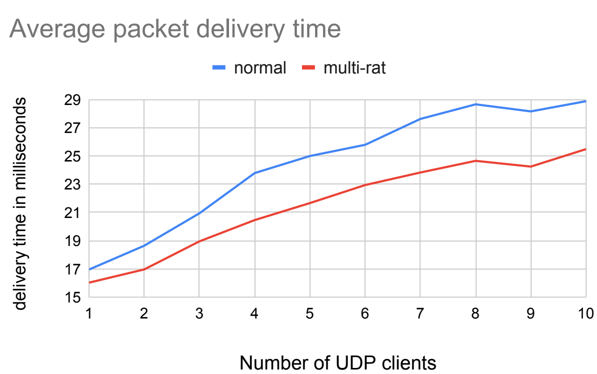


(a) A network containing nodes with a low average degree.

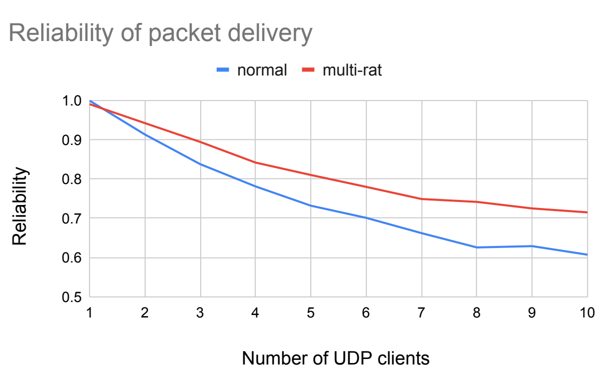


(b) Network from 5a with an additional network and multi-RAT nodes.

Fig. 5: Low average degree network and its multi-RAT counterpart.



(a) Average packet delivery time.



(b) Reliability of packet delivery.

Fig. 6: Measurement results of the low average degree network from Figure 5.

can see in the Table that even when the purple network has the same characteristics as the green network, it can decrease the average packet delivery time to 15.2 milliseconds which means almost 10% less time is needed on average to deliver a packet. The reliability is slightly negatively affected. Mainly decreasing when increasing the range of the nodes in the purple network. This is a fairly small loss in reliability and likely caused by the increased noise from more nodes being inside of each other's range.

data rate	range in meters		
	60	110	160
9Mbps	15.2	13.3	13.3
12Mbps	14.2	13.4	13.2
18Mbps	14.7	13.3	12.9
24Mbps	14.1	13.3	12.4

TABLE I: Average packet delivery time in milliseconds.

data rate	range in meters		
	60	110	160
9Mbps	99.15%	97.39%	97.24%
12Mbps	98.88%	96.79%	96.92%
18Mbps	98.18%	98.65%	95.94%
24Mbps	99.06%	98.55%	98.03%

TABLE II: Packet reliability.

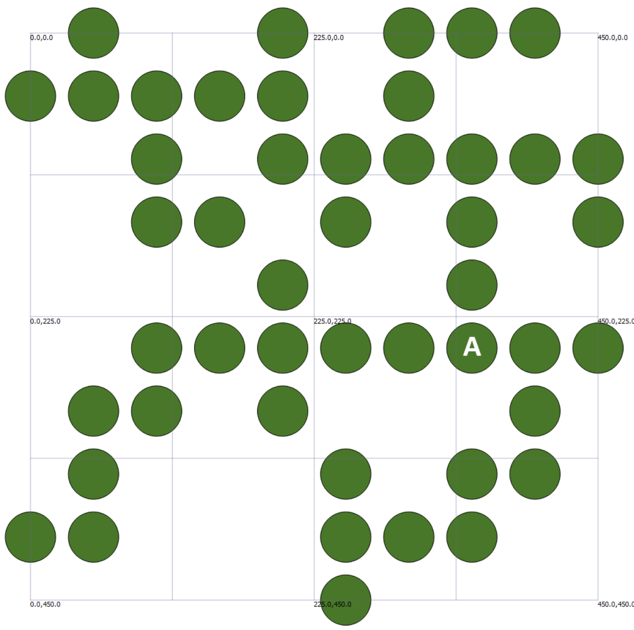


Fig. 7: Scale free network with a choke point at node A.

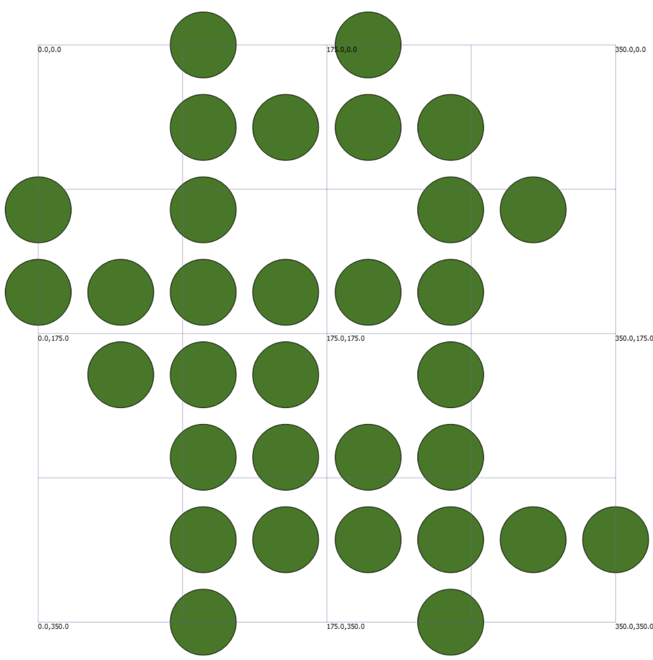
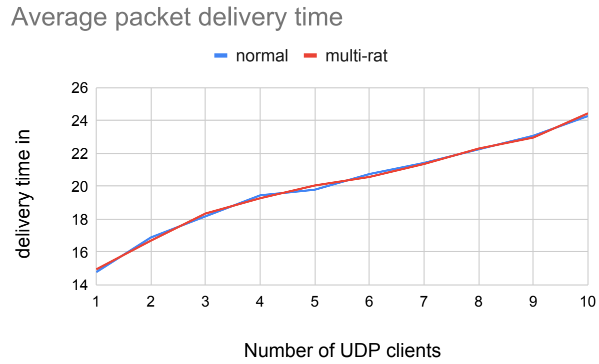
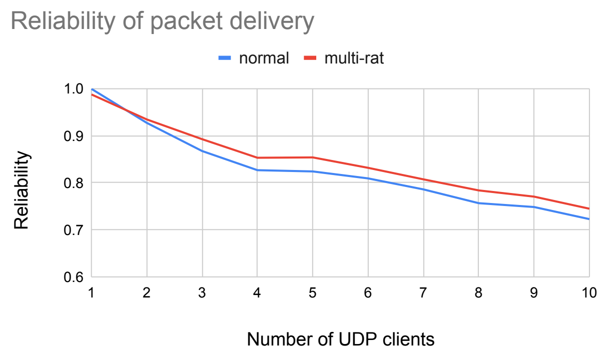


Fig. 8: High average degree network.

1) congestion: In order to measure the effect of congestion on reliability and delivery time of the normal network and the multi-RAT network, we can add more UDP client and server pairs to generate more packets inside of the green mesh network. We then use the same method of counting sent and received packets to determine the overall performance of the network using a certain amount of clients. In this simulation both mesh networks have the same characteristics. We can



(a) Average packet delivery time.



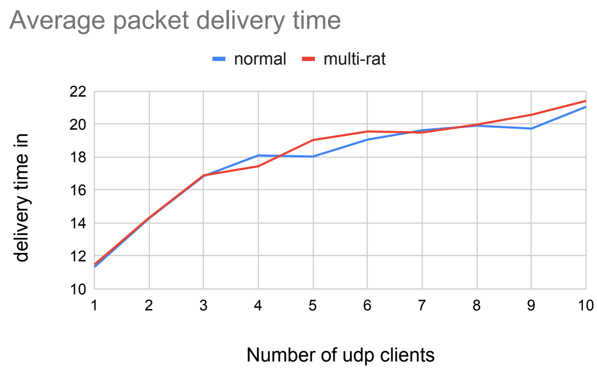
(b) Reliability of packet delivery.

Fig. 9: Measurement results of the scale free network from Figure 7.

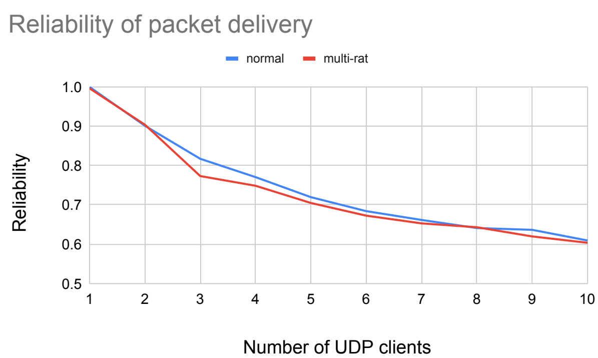
see in Figure 6a that the gap between the average packet delivery time in the normal mesh network and the multi-RAT network only increases as the number of UDP pairs increases. Additionally, in Figure 6b it is shown that when increasing the amount of UDP pairs, the reliability of the multi-RAT network goes from slightly below the normal network to as much as 10% better than the normal network.

2) varying topology: First, we look at a scale free mesh network with several choke points. The network in Figure 7 has multiple choke points, one the biggest of them being the node marked with A. The multi-RAT version of this network gives very little opportunity to deliver packets faster. However, it does allow connections to avoid the choke point by going through a different mesh network instead. This is visible in figures 9a and 9b, where the average packet delivery time is not really affected by the multi-RAT version of the network, while the reliability slightly improves

Lastly, we look at the effect of multi-RAT on a network with very high average degree. The network given in Figure 8 has a very high average degree, which means it is also very well-connected. This is generally not very good for multi-RAT as the discovery packets used to route within the multi-RAT are of a bigger size than the usual routing packets used by the networks. The results can be seen in figures 10a and 10b, where the multi-RAT version of the network not only does not increase the reliability of the network. It actually reduces the reliability.



(a) Average packet delivery time.



(b) Reliability of packet delivery.

Fig. 10: Measurement results of high average degree network from Figure 8.

C. *Results for research question 1(b)ii*

Because we have more exact control over the network as soon packets reach their first multi-RAT node, the multi-RAT nodes are able dynamically change the path they use based on changes happening within the mesh networks that it uses for its routes. For example, a network that would usually be a good path as it is faster, suddenly becomes congested. This is then detected by the discovery process and reacted upon when the routes are updated.

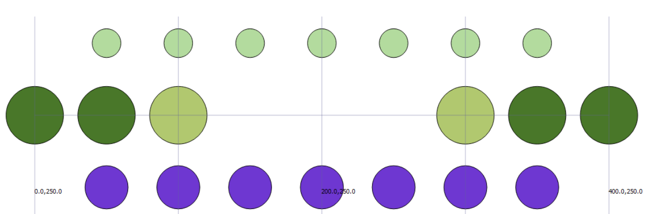


Fig. 11: Multi-RAT network with three networks to test dynamic routing.

In Figure 11 there is a third network added to the multi-RAT network that has a slightly higher data rate than the second purple network. Additionally, a UDP client and server are installed on the far left and right node of the main green network respectively. During the simulation, the third network is congested for a certain percentage of the time, causing the multi-RAT network to start using a path through the purple mesh network instead. In Table III, we can see this ability of the multi-RAT network to adapt to changes caused by congestion. Additionally, we can see the effect of congestion when the multi-RAT network does not change its route during the congestion.

Congestion time	network used			no updates
	2nd	3rd	total	
0%	0	1000	1000	1000
25%	230	765	995	876
50%	480	520	1000	770
75%	720	278	998	589
100%	969	29	998	523

TABLE III: Which network was to transfer packets over by the multi-RAT network from Figure 11.

IX. Discussion

A. *Research question 1a*

The multi-RAT nodes successfully enable communication between the two clusters of nodes as expected and the connection it provides performs similarly to what it would be if both clusters were connected with nodes of their own green network.

B. *Research question 1(b)i*

While the effect of adding multi-RAT capabilities to the low average degree and scale free mesh networks went as expected on both their reliability and average packet delivery time, it did not behave as expected on the high average degree mesh network. The low performance of the high degree network is caused by two main problems. Firstly, the multi-RAT aforementioned additional overhead that multi-RAT nodes generate with their routing packets negatively impact the already congested network. This causes slightly lower performance compared to the normal network. The reason why the situation in the main mesh network does not improve overall, is because the control packets that are used to determine which path will be taken, are prioritized by the mesh network over data packets. This causes the ARP path requests to travel towards their destination without delay, even though the main mesh network is already congested. Meanwhile, the multi-RAT path request that is delivered through the second mesh network is slightly delayed due to its larger size. This problem causes the multi-RAT nodes to not contribute as much as expected to a mesh network such as this. This is not a large concern for the scale free mesh network as its choke points are so congested that the control packets are also delayed.

C. Research question 1(b)ii

The routing algorithm of the multi-RAT nodes is able to correctly detect digestion in the networks it routes its packets through and respond accordingly, allowing many more packets to reach their destination than simply ignoring the congestion. It will still however, let some packets be dropped as the routing algorithm only updates its paths every 10 seconds in these simulations as well as a 10 second delay before degrading inactive connections. This could be made stricter, but with the trade-off that one missed discovery packet could easily lead to the mesh network no longer being used by the multi-RAT network.

D. Ant colony optimization

While the Ant colony generally reaches the correct solution for the given multi-RAT networks, multi-RAT networks with only four multi-RAT nodes are very simple, even when there can be multiple routes between the same nodes. As a result, the Ant colony is slower than a simple A* search algorithm, as the simple implementation we used still deploys a significant amount of ants to calculate optimal routes, even in simple networks.

X. Future work

For the future there are many improvements that could still be made to improve upon the existing multi-RAT node design. At the moment, when there are many multi-RAT nodes with compatible channels, a multi-RAT path request will send a request message to each node individually. This creates a lot of overhead. In order to successfully add many multi-RAT nodes to a simulation without it causing additional overhead, this problem should be solved.

Secondly, the current routing protocol simply collects information about multi-RAT nodes on the network and uses the information of all the multi-RAT nodes in the simulation to create routes. While this method works inside of a simulation, this does work in reality. That is why it would be preferable to modify it so it shares information between multi-RAT nodes through packets.

Lastly, the current routing algorithm calls upon a python script to generate its routes. This is done through writing the multi-RAT node status into a status file, running the route generator and then reading the routes file. This makes it so the route generator using Ant Colony Optimization can easily be swapped out for another one and thus different methods could be tested for generating routes for the multi-RAT nodes to use.

XI. Conclusion

From the results we have been able to achieve, we can say that we have successfully implemented multi-RAT. They are able to enable communication between nodes in two separate, but compatible mesh networks as if the nodes were in the same mesh network. They were also able to improve upon communication between nodes within the same mesh network. While looking at to what extent different mesh network configurations impact the effect of multi-RAT networks. We have seen that the effectiveness of the addition of multi-RAT is very dependant on the network it is deployed in. A well suited network for multi-RAT such as one with a low average degree has many opportunities for a second mesh network to provide improved packet delivery times and reliability for the main mesh network to use. Meanwhile, for a scale free network we have seen that even when there are no shorter paths available, the multi-RAT network is able provide improved reliability to the main mesh network by creating paths that avoid the choke points of the main mesh network. Lastly, we have seen a high average degree mesh network that did not benefit from the addition of multi-RAT due the way the mesh network prioritizes control messages. The final point we looked at was the ability for multi-RAT networks to respond to changes within mesh network configurations. Here, the multi-RAT network was able to correctly respond to congestion in one of the mesh networks and route its packets through a different mesh network instead.

References

[1] K. Zia, A. Chiumento, and P. J. M. Havinga, “AI-Enabled Reliable QoS in Multi-RAT Wireless IoT Networks: Prospects, Challenges, and Future Directions,” en, *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1906–1929, 2022. DOI: 10.1109/OJCOMS.2022.3215731.

[2] J. Li, X. Zhang, S. Wang, and W. Wang, “Context-aware multi-RAT connection with bi-level decision in 5G heterogeneous networks,” en, in *2017 IEEE/CIC International Conference on Communications in China (ICCC)*, Qingdao, China: IEEE, Oct. 2017, pp. 1–6. DOI: 10.1109/ICCChina.2017.8330398.

[3] P. Wang, J. Qin, J. Li, M. Wu, S. Zhou, and L. Feng, “Optimal Transshipment Route Planning Method Based on Deep Learning for Multimodal Transport Scenarios,” en, *Electronics*, vol. 12, no. 2, p. 417, Jan. 2023. DOI: 10.3390/electronics12020417.

[4] A. P. Badetskii and O. A. Medved, “Choice of cargo delivery option in multimodal connection based on reinforcement learning,” en, *Journal of Physics: Conference Series*, vol. 2131, no. 3, p. 032 103, Dec. 2021. DOI: 10.1088/1742-6596/2131/3/032103.

[5] Y. Peng, S. H. Gao, D. Yu, Y. P. Xiao, and Y. J. Luo, “Multi-objective optimization for multimodal transportation routing problem with stochastic transportation time based on data-driven approaches,” en, *RAIRO - Operations Research*, vol. 57, no. 4, pp. 1745–1765, Jul. 2023. DOI: 10.1051/ro/2023090.

[6] X. Cai and Y. Chen, “Multipath Routing for Traffic Engineering with Hypergraph Attention Enhanced Multi-Agent Reinforcement Learning,” en, in *2022 31st Wireless and Optical Communications Conference (WOCC)*, Shenzhen, China: IEEE, Aug. 2022, pp. 103–108. DOI: 10.1109/WOCC55104.2022.9880574.

[7] S. H. Alnabelsi, H. Bany Salameh, R. R. Saifan, and K. A. Darabkh, "A multi-layer hyper-graph routing with jamming-awareness for improved throughput in full-duplex cognitive radio networks," en, *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 5318–5332, Sep. 2022. DOI: 10.1016/j.jksuci.2022.01.003.

[8] G. R. Hiertz, D. Denteneer, S. Max, *et al.*, "IEEE 802.11s: The WLAN Mesh Standard," en, *IEEE Wireless Communications*, vol. 17, no. 1, pp. 104–111, Feb. 2010. DOI: 10.1109/MWC.2010.5416357.

[9] K. Perlin, "An image synthesizer," en, vol. 19, no. 3, 1985.

[10] B. May, E. Greer, G. Holt, and K. Vargas, "Ant Colony Optimization: An Advanced Approach to the Traveling Salesman Problem," en,

[11] K. Andreev and P. Boyko, "IEEE 802.11s Mesh Networking NS-3 Model," en,