Joint Optimization of Path Selection and Link Scheduling for 5G Millimeter Wave Transport Networks

A Master's Thesis submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Diego Andres Huerfano Torres

In partial fulfilment

of the requirements for the degree of

Master in Telecommunication Engineering

Supervisor: Ilker Demirkol

Barcelona, October 2016

Acknowledgments

Abstract

5G networks and its deployment face several issues already being addressed by the industry. A great amount of new connected users, different applications and services offered by a variety type of providers have increased the demand in an exponential manner and have called for a new logical and physical infrastructure that can support high amounts of variable traffic flows maintaining a high reliability.

New control strategies and technologies such as Software Defined Networking are allowing both operators, network controllers as well as content and service providers to agree on a framework in which a locally or globally placed controllers act as "network orchestrators", meaning that they will have full network state information and can enforce rules to each one of its controlled elements in order to meet certain performance specifications. This specifications come from different network functions necessary in order to achieve high performance with efficient use of resources.

Throughput this project, a multi-objective optimization approach to the Path Selection and Resource Scheduling Problem is going to be analyzed and evaluated with the help of a robust optimization solver, in order to show the importance of including the link and resource scheduling problem into future networks. This project will focus on Transport Networks due to their importance and key role in allowing 5G Networks to exist. This assesment will be done taking into account industry parameters and commercial realizations, as well as projected 5G and 4G traffic.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Fifth generation mobile telecommunications technology, one of the main advances for mobile communications, is in the verge of deployment and multiple research topics are being currently investigated. One of the main motivations and challenges that 5G networks has to deal with is the increase in data flowing through the network due to the increase in devices and applications that require high capacity links with minimum delay. It is expected that the increase of demand will be about 1,000 times more than what 4G/LTE supports as of today and the devices connected to the network will increase considerably.

More than 1000 Gbit/s/km2 area spectral capacity in dense urban environments, 10 times higher battery life time of connected devices, and 5 times reduced end-to-end (E2E) latency are anticipated in 5G systems [1]. This is due to the densification of low level cells like small or femto cells that cover small areas in order to increase coverage to limited areas and provide higher data rates. 5G networks have to cope with an extremely high variety of requirements and connections in which multiple vendors in all levels and multiple use cases coexist in the same network, which in turn asks for a programmable upper high level module capable of adapting and re-configuring the network based on operator policies and traffic profiles.

5G systems are envisioned to be highly flexible and scalable networks in which diverse functional splits, use cases and scenarios converge in a single deployment, which relies on this flexibility to efficiently increase coverage and number of connected users and devices, providing reliable communications in multiple scenarios. RAN domain, for example, will introduce different types of architecture in which different types of traffic will be generated on Macro cells, Remote Radio Heads and smaller scale cells, and all of this traffic should be carried efficiently by the network using almost the same resources.

The introduction of new Radio Access Technologies (RAT) and the over-densification of mobile networks and data, calls for a more flexible network in which also, the introduction of new and enhanced RAT such as Millimeter Wave (mmWave) and the increase of high level processing network elements, calls for a scalable and dynamic mobile network in order to include all of these new technologies in favor of optimizing network performance.

Thus it became necessary to have a flexible control and orchestration scheme in which a network controller or several network controllers are capable of re-programming, re-configuring the network and manage resources in an efficient manner.

To provide this flexibility and resource management, main efforts are focusing towards Software Defined Network (SDN) technology, in which a virtualized network controller has complete knowledge of the overall network and could enforce certain rules that nodes should follow to optimize overall functioning and management of resources in different segments of the network. Functions like link and resource scheduling, failure recovery, energy optimization and network re-configurability are some of the main objectives of controllers in an SDN environment. Providing efficient network management algorithms and network state abstraction thus becomes some of the most important features of 5G networks currently under investigation.

1.2 Objectives

One of the key issues raised by 5G's mobile systems comes from the fact that in a particular segment of the 5G network, some of the key functionalities of the signal processing and coding chain are either implemented on a centralized or distributed entity or a combination of both. By implementing this, network operators "split" baseband processing chain in order to either centralize or distribute certain network functionalities depending on their needs.

Fronthaul traffic (FH) in which baseband processing is done in a centralized entity and Backhaul traffic (BH) in which baseband processing is done entirely on site are two of the main traffic requirements of future transport networks. Millimeter Wave is one of the main wireless transport technologies that can be implemented in order to handle this amount of traffic efficiently.

This is why, the main general objective of this project is to research, develop and test a methodology in which we aim to achieve considerable performance improvements for 5G transport networks in the framework of resource utilization and network performance optimization.

The main objective of this project will be to develop a mathematical formulation for the link and resource scheduling problem applied to a 5G transport network based on mmWave technology. An optimization formulation will be derived taking into account the different requirements of future 5G transport networks regarding topology, network functioning and traffic requirements.

To assess the viability of this formulation we will make use of an optimization software which can be used to evaluate the optimization formulation to further develop possible algorithms and ways to approach our problem. Finally the results obtained from our evaluation will be derived and analyzed on different scenarios in order to analyze its performance.

This thesis will be part of the 5G-XHaul project whose input, ideas, assumptions and data have been gathered and thoroughly analyzed in order to derive a realistic scenario in which we can test our optimization formulation.

1.3 Structure of this thesis

This thesis is organized in different chapters. First, we can read in chapter 2 a through explanation of future 5G transport networks and mmWave technology as a key enabler to meet 5G requirements and its specifications given by the IEEE 802.11ad standard. Moreover, this thesis will try to explain software defined networking as a new technology that will be present on 5G mobile networks and that will be a basis for the problem formulation. In chapter 3 our optimization problem formulation will be explained and the optimization solver tool will also be presented. Chapter 4 includes main results and simulations. Conclusions and future work are explained in chapter 5.

Chapter 2

Background and Related Work

2.1 5G Network System Architecture

Future 5G Mobile systems impose a great challenge for the industry and for service providers. A broad range of different services and requirements regarding capacity, coverage, network usage and achievable data rates, specific for each case, poses a serious challenge in how 5G network architecture should be structured and how it should be managed.

5G is set to operate, as explained before, in a highly heterogeneous network in which different types of cells, technologies, layered architectures and vendors will be present. Breakthrough technologies such as software defined networking introduce capabilities like reconfigurability and architecture flexibility that enable this variable set of scenarios to coexist and in turn allows operators to develop and innovate in their own services in order to introduce them in a very cost-efficient manner on existing network architectures. This flexibility is possible by introducing multi-service and context aware network functions, control technologies such as software defined networking and joint optimization of resources on all network segments.

5G will operate in a highly heterogeneous environment characterized by the existence of multiple types of access technologies, multiple types of devices and a big amount of scenarios. One of the main developments made currently, that have a great impact on how the network will need to function is focused on the Radio

Access Networks (RAN). For a particular physical infraestructure for example, developments are being introduced in which the traditional wireless access network ruled by macro-cells is changing to a heterogeneous network where densely deployed small cells, femto cells and macrocells coexist. RAN as it is traditionally defined, is composed of radio units and baseband units located in the same site, which limits the flexibility of the network and increases the costs of dense areas covered by smaller cells. The idea of Centralized-RAN or Cloud-RAN (C-RAN) was introduced as a technology that can adapt itself to the new changing conditions and the increase of traffic. In C-RAN, the base band processing is made by a Base Band Unit (BBU) pool that sits in a different location than the access site and is in charge of processing and sending digitized radio signals to remote radio units known as Remote Radio Heads (RRH). Fronthaul network is then composed of transport links that connect BBU units with RRH and that need to carry considerably big amounts of data [7]. Although fronthaul networks are the rule, 5G networks will not only include fronthaul links but also backhaul links for traditional back-haul traffic.

This also brings challenges to the transport and Core Networks (CN) in the way that they will have to cope with both heavy and low flows of traffic depending on the underlying RAN architecture used. A possible simplified architecture for 5G networks is shown in Figure 2.1.

As seen in the 5G architecture, the service layer is in charge of orchestrating the supported services offered in the network, software Network layer is in charge of managing and orchestrating network functioning by virtualizing certain network functions like routing, link failure and energy optimization. Networking layer and resource abstraction layer are in charge of recollecting information from every element of the network for the goal of abstracting a model of the network's current state, in order to send it to the SDN layer to apply management algorithms.

The physical infrastructure is again, composed of RAN, Transport Network (TN) and Core Network (CN). Each segment can be based on several technologies and their elements should be able to communicate with the SDN controller. The main focus of this work is on the transport network segment of the 5G architecture because of its complexity and high traffic demand requirements.

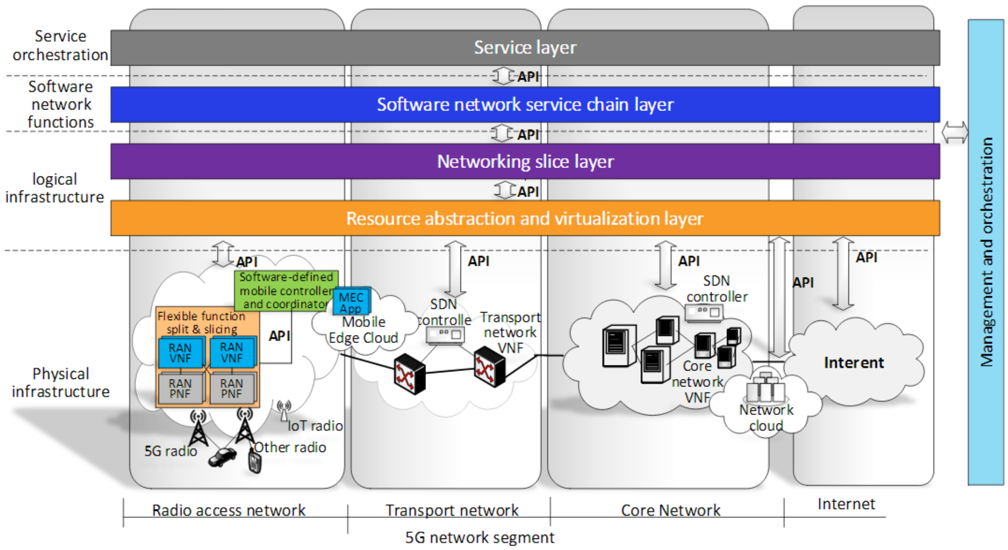

Figure 2.1: 5G Simplified Architecture

2.1.1 Support for Transport Classes

5G mobile networks as explained before are developed in a framework in which diverse use cases and scenarios are deployed, each with certain requirements regarding the traffic carried or generated. C-RAN deployments demand very high throughput traffic with strong latency requirements because all processing is done in a centralized base-band processing unit. On the other hand, traditional back-haul networks in which remote stations have the task of baseband processing, demand lower throughput demands with flexible latency requirements. Due to the high number of use cases and the dynamic nature of future 5G networks, with the inclusion of new technologies, it then becomes obvious that the design of RAN and transport networks should not account just for C-RAN deployments but also should include additional flexible functional splits in order to reduce the strong requirements of a scenario where only FH traffic is present. The C-RAN concept in a multi-node enviroment is shown in Figure 2.2[7].

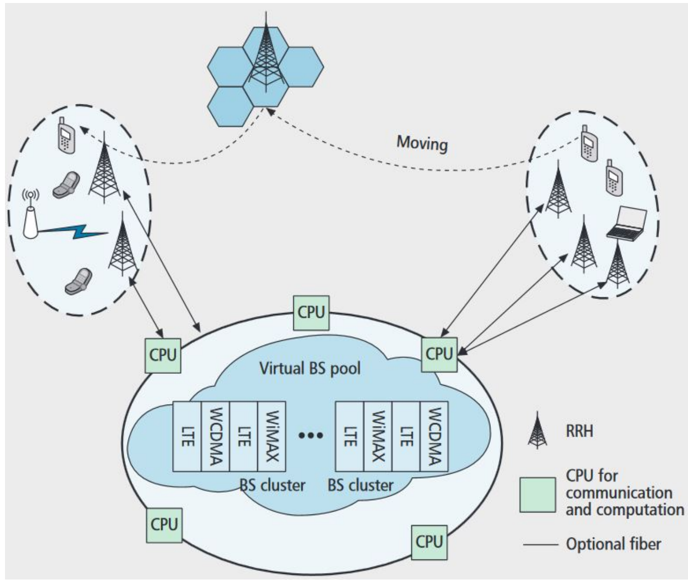

Figure 2.2: C-RAN in HetNet

For this matter, functional splits of the processing chain are being standardized in [8] in order to define a set of transport classes that account for both completely centralized RAN deployments and traditional distributed RAN. This task will be of key importance in order to fully dimension Transport segment.

It also will allow operators and controllers to design control functions that can organize and assign resources depending on traffic prioritization schemes, use cases and scenarios [8]. These functional split are shown in the Figure 2.3.

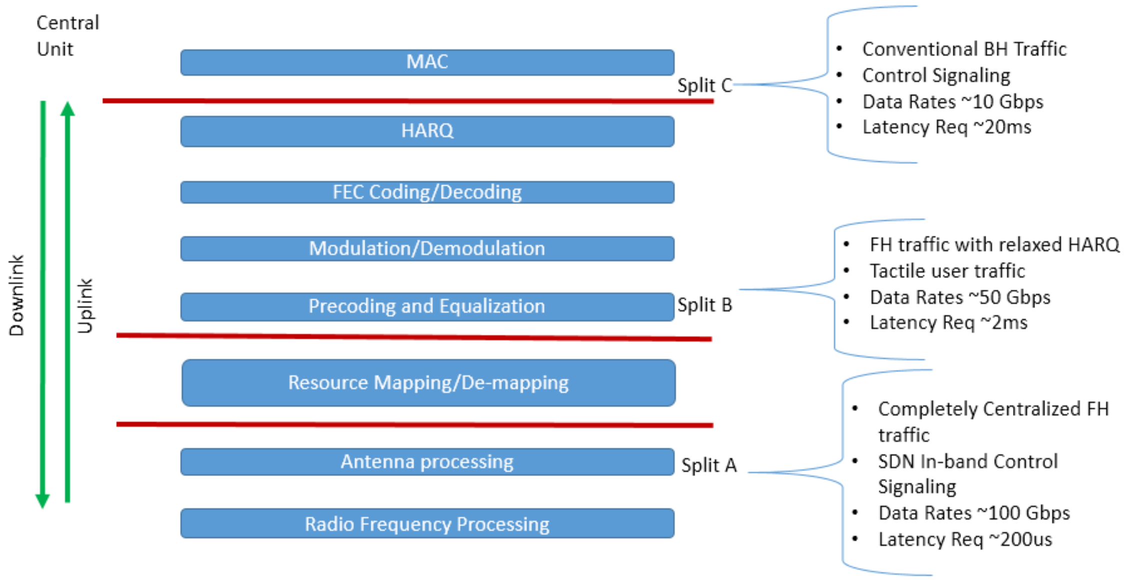

Figure 2.3: Functional Splits

Each functional split requires a certain throughput demand and latency requirement depending on which functions are centralized and which are distributed to the RRU . For example Split A, in which central unit is in charge of the whole baseband processing and complete time domain samples are sent to the RRU from the base band unit, entails having heavy flows in the transport network that require high data rates and stringent End-to-End delay requirements of at most 200 μs. In turn Split C, regarded as traditional back-haul traffic, demands lower data rates and has more flexible latency requirements. These splits can not only define BH or FH traffic but can be extended to certain services present in 5G networks. Therefore the transport network will need to be reconfigured and managed on a per-flow basis in which each flow will account for variable use cases, service and functional split. In order to organize the network accordingly, these variable flows can be bundled into comprehensive transport classes which are defined by certain QoS parameters that have to be fulfilled by the network (minimum throughput, delay, jitter etc.)

The projected amount of peak data rates for each traffic class that transport network will carry and the main requirements envisioned are shown in Table 2.1 [8].

Transport Class	Type of Traffic	Transport Latency	Typical Peak Data Rate
$TC0$	Synchronization	Low Variance	$10Mbps$
$TC1$	Split A Traffic	$\leq 200\mu s$	$100Gbps$
$TC2$	Split B Traffic	$\leq 2ms$	$50Gbps$
$TC3$	Split C Traffic	$\leq 20ms$	$10Gbps$

Table 2.1: 5G Supported Traffic Classes

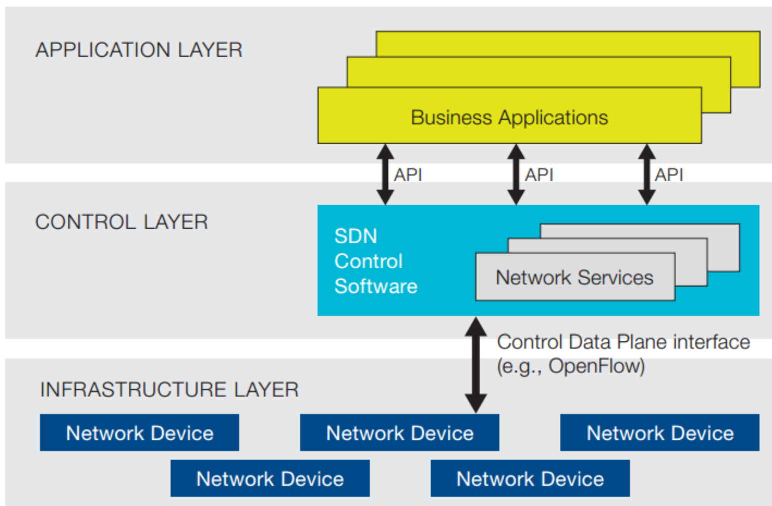

Figure 2.4: SDN Architecture [10]

2.2 Software Defined Networks

Software Defined Networks (SDN) is one of the key technologies that will enable 5G networks to be dynamic, flexible and fully scalable. The idea behind its applicability is that, using virtualization techniques and the high capability for processing and storage of data centers, network administrators can manage network services in an efficient manner by decoupling control plane (where network decisions are made) and data plane (underlying systems such as access or transport technologies). Network controllers and intelligence are logically centralized and thus maintains a global view of the underlying network. SDN architecture is shown in Figure 2.3

Main advantages of SDN based technologies are multiple. First SDN control software can control network devices from multiple vendors with the OpenFlow interface. This shifts the responsibility to network operators of defining their own network algorithms to account for their own performance metrics. This allows this controllers to deploy, configure and update devices across the entire network. On the other hand, OpenFlow interfaces allow the controllers to apply policies in a granular level. This means that each network manager can apply in an automated manner policies regarding session, user and application layers.

SDN is defined by three specific layers. The business layer defines a set of applications particular to each vendor or network operator. The business applications are tightly linked to vendor specific requirements and are independent between business operators. The second layer, which is the control layer is in charge of defining network controller's specific network services in order to meet business requirements. Functions like routing, resource scheduling, multi-cast, security, access control, bandwidth management, traffic engineering, quality of service, processor and storage optimization are some of the key services that network operators and controllers aim to provide. Through control data plane interfaces such as OpenFlow, network controllers can enforce rules on network devices in order to meet certain network performance goals. Through this interface, controllers are able to abstract network information and have complete knowledge of resources available and current network topology.

2.3 Millimeter Wave Communication

Millimeter Wave Communications have been investigated and are one of the key enabler technologies for future 5G networks to meet capacity and coverage requirements. Millimeter Wave comprises $3GHz-300GHz$ bands, however current research are focusing on $28GHz$ band, $36GHz$, $60GHz$ and the E-band $(71-72GHz)$. These bands have a great amount of available bandwidth that can provide solutions to the bottleneck presented on 5G networks. Exploiting the wide spectrum available, the small size antennas and the new beam-forming technologies that this technology provides allows us to achieve very high data rates with expanded coverage thus increasing support for multiple users with high data traffic.

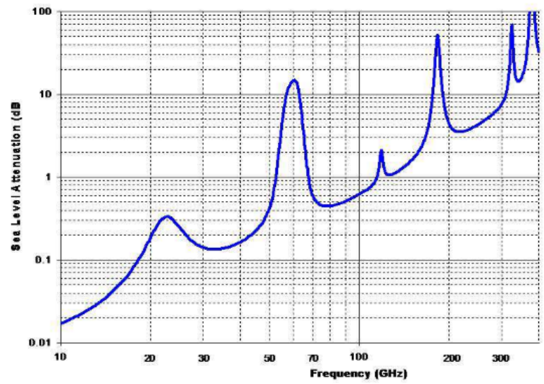

Figure 2.5: Millimeter Wave Atmospheric and Molecular Absorption [2]

However some issues regarding propagation characteristics, synchronization, the introduction of new physical layer technologies and interference coordination are currently being investigated in order to asses the reach and applicability of mmWave technologies in an Heterogeneous 5G Network.

2.3.1 Propagation Characteristics

Because of the high frequencies that mmWave technology uses, propagation of signals is rather difficult because of the effect of atmosphere, oxygen, fog and rain. This adverse conditions bring high atmospheric losses and limit the use of millimeter wave bands. For example oxygen absorption ranges from $0.04dB$ to $3.2dB$ and rain attenuation ranges from $0.9dB$ to $2.4dB$ for cell sites of 200 meters of radius. Atmospheric and molecular absorption characteristics are shown in Figure 2.5

For different frequency bands, measurements regarding the path loss exponent(PLE) and the atmospheric and oxygen absorption were made in order to define which are the most restrictive bands on mmWave and the effect of atmospheric absorption in an LOS and NLOS environment.

Frequency Band	PLE		Rain Attenuation@200 m		Oxygen Absorption @200 m
	LOS	NLOS	5 mm/h	25 mm/h	
28 GHz	1.8∼1.9	4.5∼4.6	0.18 dB	0.9 dB	0.04 dB
38 GHz	1.9 ∼2.0	2.7∼3.8	0.26 dB	1.4 dB	0.03 dB
60 GHz	2.23	4.19	0.44 dB	2 dB	3.2 dB
73 GHz	2	2.45∼2.69	0.6 dB	2.4 dB	0.09 dB

Figure 2.6: Propagation Characteristics in mmWave Bands [2]

Also NLOS environments provide a challenge because with small wavelenghts blockage of signal caused by large-sized objects becomes a significant obstacle to achieve high data rates and high link budgets. Maximum coverage distance is determined by the environment and the amount of obstructions present. For highly obstructed environments maximum coverage of up to 200 meters can be achieved. This fact limits the use of Millimeter Wave technology to LOS environments in order to fully exploit the advantages of using mmWave bands to provide high capacity links.

2.3.2 PHY Layer enchancements

One of the key aspects to take into account in mmWave communications is the high directivity of mmWave links.

One is the great amount of unlicensed and available bandwidth that mmWave bands posses which would be useful for future mobile networks. About 10x more spectrum is available which allows larger channels to be managed, which in turn also allows greater spectrum reuse in order to cope with the increase and overdensification of small cells on urban environments.

Also, because of the small size antennas on mmWave bands, technologies like Massive MIMO allow us to have a set of steerable antenna arrays with thousands of elements in which each array is capable of directing its beams to the receiver by simply controlling phase of the electric signal feeding the antenna. These beams are highly directive in nature and allow transmitter and receiver to establish a link with high gain and low interference between adjacent beams, which eases interference coordination between cells, but in turn increases "deafness" issues arising from using highly directive communications.

Due to the advantages that highly directive links provide to obtain high data rates, industry is turning its efforts into providing control algorithms that allow network controllers to coordinate antenna elements of each of the mmWave nodes, providing reconfigurability in order to support variable data traffic. This in turn requires high computational complexity on precoders and bring out the need to have global Channel State Information (CSI), which is difficult due to the high directivity on mmWave links and synchronization issues.

Future developments of mmWave communications will allow to obtain further gains regarding multi-stream transmission and full-duplex links. For this matter IEEE 802.11ay standard, as opposed to the singles-stream transmission of IEEE 802.11ad standard [3], is investigating and aims to include multi-stream transmission in order to increase spectral efficiency and throughput. Potential data rates for mmWave communications up to 25Gbps are envisioned in the near future.

2.3.3 Standarization-IEEE 802.11ad

The IEEE 802.11ad standard [3] defines modifications to the MAC and PHY layers of Access Points (AP) and Stations (STA) to allow operation in the 60GHz band and achieve very high throughput.

Regarding the PHY layer, the 802.11ad standard defines three specific PHY layers: SC PHY, OFDM PHY and control PHY. Specifically it defines allowed Modulation and Coding schemes (MCS) for each layer in order to obtain data rates ranging from 27.5 Mbps to 6756.5 Mbps for Control PHY and OFDM PHY respectively. The maximum data rates for 60GHz mmWave based elements are shown in 2.2.

PHY Layer	**Maximum Modulation**	**PHY Rates**
Control PHY	$\pi/2 - DBPSK$	$27.5Mbps$
SC PHY	$\pi/2 - 16QAM$	$4620Mbps$
OFDM PHY	$\pi/2 - 16QAM$	$6756Mbps$

Table 2.2: IEEE 802.11ad Max PHY Data Rates

Depending on the modulation types for both control and data payloads, IEEE 802.11ad defines different packet structures with different preambles, headers and MCS configurations. The packet structures for the different physical layers are shown in Figure 2.7.

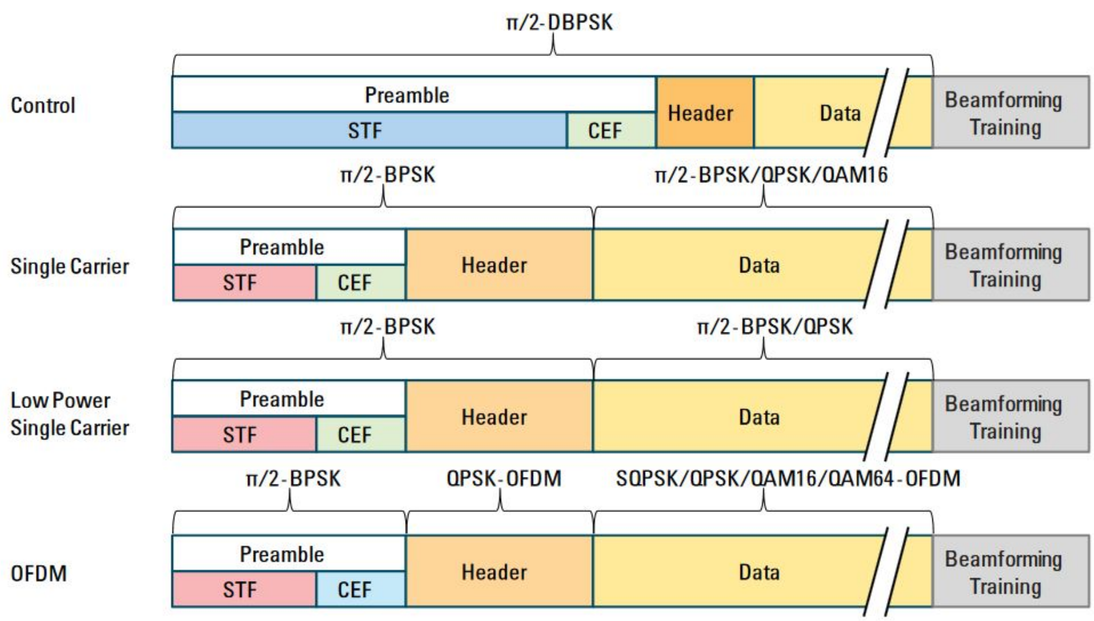

Figure 2.7: PHY Layer Packet Format

The main frame structure is composed of two main subframes, the Beacon Header Interval (BHI) and a Data Transmission Interval (DTI). The beacon header is used to exchange management information and network announcements, as well as beamforming training to take advantage of the high throughput available when using highly directional antennas. The BHI is followed by DTI in which actual nodes exchange information. In this interval there are either Contention-Based Access Periods (CBAP) in which stations contend for the use of the air interface and Service Periods (SP) in which two nodes exchange either data or extended beam-forming frames.

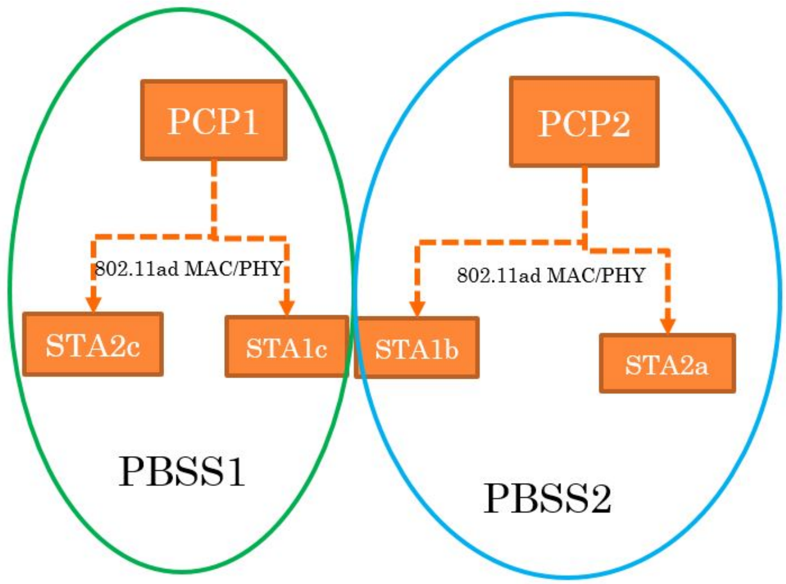

Figure 2.8: PBSS Configuration

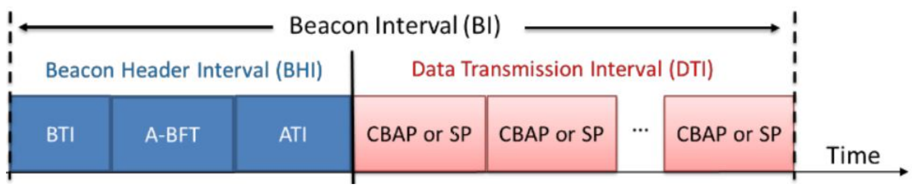

Figure 2.9: Air Interface frame 802.11ad

The DTI is composed by any combination of SP and CBAP allocations. The SP or CBAP scheduling procedures on each Beacon Interval (BI) are defined by the IEEE 802.11ad standard, each one imposing different delay, latency and overhead restrictions. It is worth nothing that the scheduling of resources is done by a network controller, which has complete network state information and thus is capable of enforcing rules to nodes on how to communicate with their peers. Only the network coordinator or coordinator station can organize the way in which nodes can communicate and is the one in charge of sending beacon frames so the nodes can know when they are allowed to transmit or receive traffic. The beacon interval structure as presented in the standard is shown in Figure 2.9 [5].

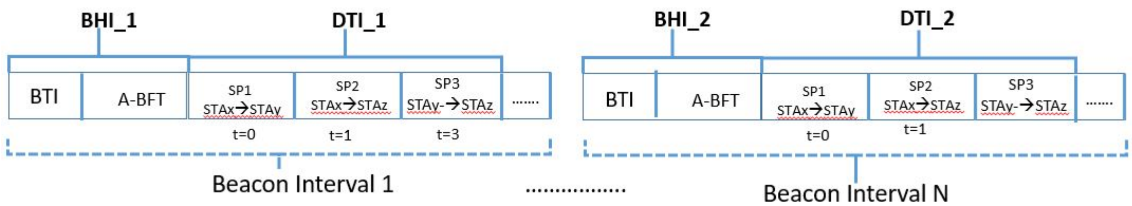

Figure 2.10: Pseudo Static Allocation

- Dynamic Scheduling: In dynamic scheduling, before each one of the SP, polling frames are issued to each one of the participating stations and the PCP organizes the time allocation following this grant periods. This comes across as an advantage because all stations are aware of the direction of incoming signals, either from the PCP or other peer STA. Dynamic allocation thus eliminates the deafness problem present on highly directional communications. Another advantage is that because time access allocation is done in a centralized manner, during each BI, the PCP can adapt to bursty traffic by changing parameters rapidly.

- Pseudo static Channel Allocation: In the pseudo-static allocation scheme, SP reoccur every BI and represents a frame exchange between two pair of stations. This scheduling is propagated by the PCP to all peer stations, which facilitates non scheduled stations to go into sleep mode thus reducing energy consumption. Each station in turn defines a traffic stream of MAC flows specifically delivered to a peer station with certain QoS parameters that have to be met. PCP then allocates transmission time according to these rules and restrictions. This scheme is shown in Figure 2.10.

2.4 Related Work

In this section we will show efforts done from researchers and academics regarding the use of mmWave communications as a technology that enables efficient scheduling and routing for future networks that can be implemented on an SDN controller in a layered architecture enviroment.

Millimeter wave scheduling has been introduced in the literature as a way to efficiently improve system performance (i.e. network throughput). The main advantage of exploiting the full potential of millimeter wave technologies is that it provides highly directive links and reduces in a certain way the possible interference with other nodes in the network. Also the amount of bandwidth available is one of the most attractive features of millimeter wave communications.

In the literature there are plenty of works that focus their efforts on explaining and introducing the advantages of using millimeter wave communications in WPAN and outdoor mesh networks. Most of them focus on converting scheduling and routing into an optimization problem in which the objective function is to maximize throughput, presenting with it some simulation and numerical results.

In [11] the concurrent transmission scheduling problem was introduced. Since millimeter wave communications provide highly directional antennas and great amount of bandwidth, there is a chance to exploit spatial time division multiple access to allow both interfering and non-interfering links to transmit simultaneously in the same time slot. Based on the SINR at each receiver based flow throughput was introduced in order to prioritize certain flows that need to be allocated above others.

The optimal scheduling problem was formulated in which there are transmission requests of data from the nodes to the controller. This last one is in charge of maximizing total throughput by determining which flows will be scheduled on each timeslot.

An heuristic algorithm was based on a slot by slot decision in which the idea is to try to schedule as many flows as possible in the network. To do this a hybrid multiple access of CSMA/CA and TDMA is defined, in which there is a superframe that consists of three phases: A beacon period for network synchronization and control messages, contention access period used to transmit requests to the controller and finally a Channel Time Allocation Period (CTAP) for data transmissions.

The CTAP period contains timeslots that are allocated to certain flows depending on the optimization results, so the controller makes scheduling decisions based on the maximization of network throughput.

Some other proposed works focused on scheduling schemes that take into account interference suppression and beam searching mechanisms in order to again, maximize network throughput. In [12] another proposed scheduling algorithm was developed to avoid interference by using optimization algorithms based on SINR and SNLR measurements. A scenario was defined in which a picostation schedules beams to each User Equipment (UE) on a given time slot. In the SINR based scheduling, the scheduler selects the beam with highest SINR in each iteration and computes the interference from other selected beams to the same user. In the same way, the SNLR based scheduling selects the highest SNLR at each step and computes the interference caused by this particular beam to other users. In the conventional priority factor scheduling (PF), each UE is scheduled to transmit depending on a priority factor that relates the instantaneous data rate and average data rate of user i.

Simulations showed that SNLR, SINR and PF scheduling schemes function in a better way than conventional Round Robin scheduling in which all beams associated to each UE are divided into groups and each group is assigned a time slot.

Other works like [13] adds to the optimization problem of concurrent transmissions the idea of beam-searching and the fact that a throughout search for highly directive beam alignment between receiver and transmitter adds alignment overhead which puts restrictions on the time needed to obtain a scheduling decision. Specifically this work defines the search taking into account sector-level and beam-level beamwidths which need to be correctly dimensioned and optimizes throughput, taking special care in not increasing alignment overhead too much (i.e. not so narrow beams). So, a joint beamwidth selection and transmission scheduling optimization is proposed in which the objective goal is to maximize system throughput needs to be resolved by the controller, restricting also time for beamwidth alignment.

Finally in [14] a more practical scenario the routing and scheduling using millimeter wave backhaul is addressed in which the objective is to select backhaul links and paths to maximize throughput and minimize delay for users in a network. This approach takes as a main objective to design a dynamic link scheduling to maximize backhaul capacity per given time window.

Here a Central Unit (CU) serves as a controller and traffic aggregator for dense

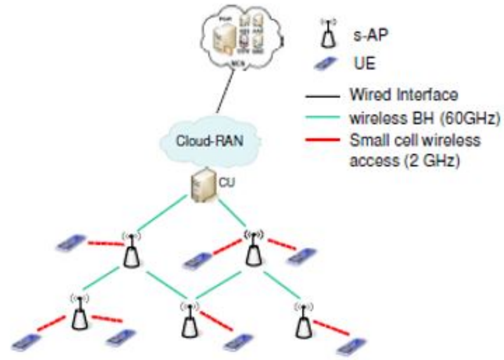

Figure 2.11: Scenario for path selection and scheduling algorithm

small cell network, access points (s-AP) have been provided with access link for backhaul based on 60GHz as seen in Figure 2.11. It is assumed that the channel knowledge is given in the CU.

The optimization problem is the minimization of the total number of time slots, defining the ratio of the demand over the backhaul link capacity towards an s-AP. The objective is to find the paths that traffic should follow and links to be activated to maximize system performance. The novelty of this solution is to propose a two stage problem using LP relaxation: a path selection algorithm and a packet scheduling problem.

After defining the time slot that each link is going to use, a scheduling algorithm of how to forward packets throughout the network with a minimum number of hops is considered. Packets are sent to their destinations and intermediate nodes store this packets in queues and in turn forward them to adjacent nodes.

Simulation results were shown with interesting discoveries. Regarding the path selection algorithm findings where that in long-distance links NLOS will impact performance and in the scenario of low number of paths between CU and s-APs, the short distance LOS links could increase throughput performance because these links can attain higher capacity.

Regarding the scheduling algorithm it was shown that the average time delay decreases with the number of paths created to each of the nodes. There is in fact a trade-off between maximizing throughput of the network and coping with maximum delay when more paths are created, because when more paths between CU and s-AP are defined, the time needed by the CU to send all traffic to all nodes increases.

Our particular formulation, which will be explained in the next chapter, provides a different perspective to the link and time-slot scheduling algorithms proposed previously. Our problem formulation focuses on providing a solution based on a multi-objective optimization in which both conflicting objectives such as time-slot scheduling and load balancing are optimized in a joint manner taking into account flow-delay based restrictions and a variable number of different flow-types. We aim to find a solution to both problems by defining a problem formulation that takes into account both decision variables without increasing computational efforts unjustifiably. This particular formulation also takes into account the amount of different flows with several different requirements.

Chapter 3

Routing and Link Scheduling Optimization

3.1 Overview and Motivation

Software defined networking will be the base of future 5G mobile network developments. As a network optimization enabler, SDN provides the necessary scalability and flexibility that both the transport and RAN networks demand. Having complete network state information and topology allows the orchestrator to adapt the network resources in order to meet certain demands and reconfigure the network in an automated way. Path selection and resource scheduling is one of the key control functions that network controllers will include in order to obtain such performance.

In an heterogeneous scenario like 5G networks, multiple sources of traffic are encountered even in small geographical areas due to network densification. Different types of services can be encountered on each cell, each with their own throughput and latency requirements. Taking into account that in future mobile networks, small cell densification and the introduction of new technologies such as C-RAN and further functional splits will be supported, a great amount of different traffic flows with diverse classifications and specifications will have to flow through the different parts of the physical infrastructure. This calls for a more robust and flexible control layer that can adapt to this increase in traffic demand by managing shared resources in an efficient manner.

Transport network is one of the scenarios in which such amount of traffic is encountered and where centralized control is a sufficiently scalable and flexible scenario. Different traffic will be aggregated and transported throughout the network to multiple different destinations (i.e. Core Network, Base Band Units, etc). How to overcome this challenge, also relies in a great way in the transport technology employed.

For example mmWave technologies can attain very high data rates employing massive amount of steerable and highly directive antennas, which makes this technology suitable for mesh and point-to-point topology and gives transport network the scalability and re-programmability it requires. By designing efficient scheduling algorithms in and SDN deployment, network topology can be reconfigured by redirecting a set of antennas of each transport node to communicate with different peers forming multiple paths for traffic to flow. Following this path selection, network controller should be able to enforce rules on transport nodes on how to handle traffic following certain design criteria.

In the next sections of this chapter our aim is to provide an optimization approach to the path selection and scheduling of resources in the transport network, involved in a SDN deployment with mmWave communications between the transport nodes. Path selection will be in charge of defining links between Transport Nodes (TN) in order to create paths for every source-destination pair. Resource scheduling is in charge of allocating air time to each participating link in order to meet latency and throughput requirements. The proposed optimization formulation takes into account the different types of flows bundled in traffic classes in a converged transport network

.

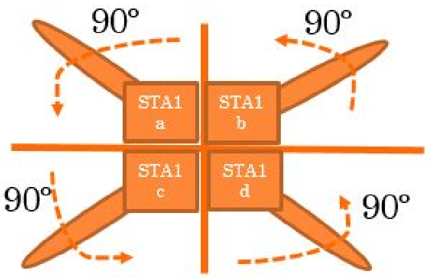

Figure 3.1: Transport Node Model

3.2 System Model and Assumptions

The system model assumptions that we took into account in order to define our network architecture and topology are the following:

- Transport nodes are based on mmWave technology and define their PHY and MAC layer functionalities based on IEEE 802.11ad standard.
- Following the 802.11ad standard, each transport node is composed of 4 STA (Figure 3.1), whereas each STA is composed of a 90° steerable antenna element, limiting number of possible links that can be scheduled on each transport node. This 4-STA based transport node model is based on standardized commercial products in the market.
- TDMA operation capabilities are assumed at the PCP, so this entity will issue Service Periods, each limited by a duration no more than one time slot in order for peer stations to send its frames. With this assumption, resource allocation is translated in time-slot allocation. This scheme is shown in Figure 3.3
- It is assumed that beamforming training and sector level sweeps are already defined and pose no effect on flow delay of each traffic flow. This means that we assume no information exhcange is present between STA's and PCP for scheduling of beamforming training.
- Pseudo-static allocation is assumed, which is explained in Chapter 2.

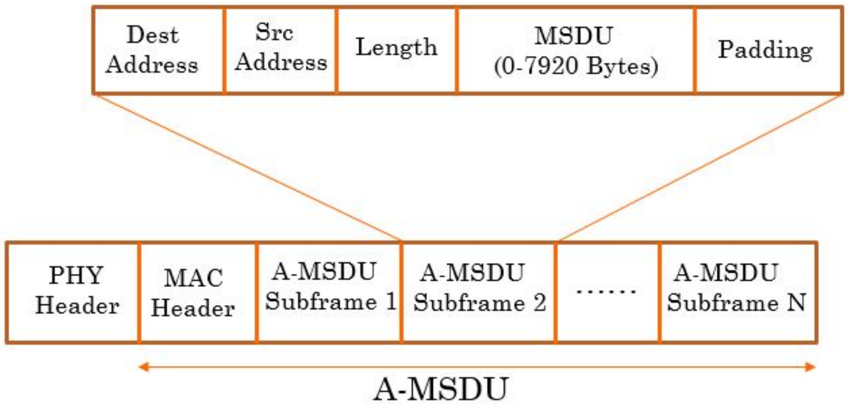

Figure 3.2: MAC A-MSDU aggregation

- A link based optimization is assumed. This is possible by allowing upper MAC layer to multiplex multiple MSDU's from different flows in the same air interface frame A-MPDU. A-MSDU is supported by IEEE 802.11ad standard, which defines maximum MSDU payload length and subsequent MAC headers. Figure 3.2 shows how this A-MSDU MAC aggregation scheme works.

- Due to limitations of mmWave and the high amount of traffic that functional split A (FH traffic) entails, our supported traffic classes for mmWave based transport network account for split B and split C traffic. This particular splits are load dependent and support statistical multiplexing to be performed. This allows us to define the demand of each flow based on real expected per-user traffic projected in 5G Networks.

- A LOS scenario is assumed for the signal attenuation and a path loss ABG model is used to determine path loss calculation. Atmospheric attenuation and interfering links are determined on an specific transmission time. Based on this interfering links, SINR is calculated on the receiving STA in order to determine if this SINR meets a certain threshold. If it does this link can be scheduled at time t.

- As for PBSS configuration, we assume that this PBSS's are already predefined. This relaxes the scope of our formulation because we assume PCP and STA

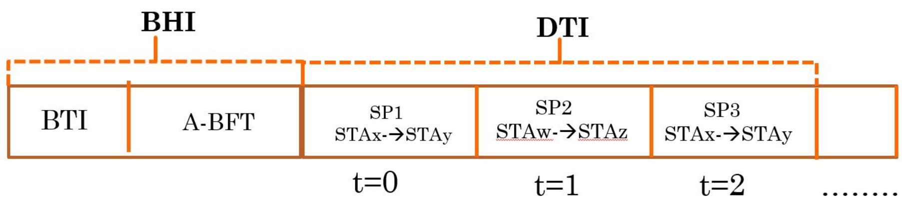

Figure 3.3: Time Slot Allocation Scheme

have already made their respective beam-forming sessions and every node is aware of its neighbors.

Our formulation, assumes that PCP has TDMA capabilities. That is, PCP is able to divide scheduling frame (DTI portion of BI) in time slots or transmission opportunities in which STA's can transmit frames to its peers. These time-slots are the basis for our time slot scheduling formulation. We assume that on each slot several different flows can be appended given certain thresholds like maximum allowed MAC payload, maximum allowed traffic on each slot and maximum slot duration. This time slot allocation is done during the DTI of each BI and is shown in Figure 3.3.

Given the amount of traffic demanded by each source and the traffic transmitted or sent by each transport node, allocated slots to each node are dynamic and adaptive, this means that air time is scheduled to each link if and only if it has data queued. If no data is queued on a link then no air time is going to be allocated.

3.3 Network Model

For the sake of simplicity for our mathematical formulation the complete network topology is characterized by a bipartite graph $G = (V, E)$ where $|V| = N$, and $|E| = M$ where V is the set of N transport nodes including source and destination nodes and the edges E are physical mmWave links between each pair of nodes. We denote $\mathscr{S}$ and $\mathscr{D}$ as the source and destination node set respectively and each flow per source-destination pair k_w as (s_k, d_k, w) where w denotes the flow number. Each flow from every source-destination pair is characterized by traffic specifications

(TSPEC) [3]. For the sake of simplicity in network optimization, diverse traffic flows can be bundled on certain traffic classes depending on their source and their traffic specifications. Since our main focus is directed to 5G Converged Transport Networks (CTN) where both FH and BH traffic flows will share same physical resources.

3.4 Problem Formulation

We define the path selection and time-slot scheduling problem as a multiobjective optimization in which two main objectives are defined. First we want to optimize load balancing, that is, distribute traffic flows optimally across the network by minimizing the traffic in the maximum utilized links. Given the limitations of the underlying transport technology used and the great amount of different traffic generated for example, in urban scenarios, balancing heavy and low flows could provide serious improvements on network performance while avoiding overloading of transmission links.

Second, we aim to minimize the number of timeslots needed to deliver each flow to each destination, taking into account delay and latency specifications. Each demand is characterized by its TSPEC (i.e. data rate, packet size, number of stream etc), origin and destination. Due to the limitations on the values that the variables can take and the multiple constraints, a complex integer programming optimization problem is represented in which decision variables are determined for each problem. First we define the links that are going to be used on path between each source-destination pair and then a resource scheduling optimization is performed in order to assign timeslots to the different flows taking into account constraints regarding maximum capacity, SINR, timing constraints and node capabilities. For the path selection we define the following variable:

$$x_{ij}^{kw} = \begin{cases} 1, & \text{if link (i,j) transmits flow } w \text{ from pair k} \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

The path selection problem finds the optimal path assignment for each demand, following a minimization of the maximum utilized link. The feasible solution is restricted by constraints of minimum flow demands, link delay and maximum capacity.

Next, timeslot scheduling optimization is in charge of allocating resources to STA's involved in each one of the paths that different flows will traverse. To accomplish this, a variable represented by u_{ij}^t states the following:

$$u_{ij}^t = \begin{cases} 1, \text{if link (i,j) is scheduled to transmit traffic in time t} \\ \\ 0, \text{otherwise} \end{cases} \tag{3.2}$$

In (3.2) we define that a timeslot can be used to send or receive traffic if and only if the link (i, j) is chosen to transmit the corresponding flow. This way we avoid scheduling timeslots to links that dont have data queued. The solution for the timeslot scheduling depends on whether link (i, j) is part of the feasible solution of the path selection problem. The variables that will be employed in the problem formulations are shown in Table 3.1.

3.5 Path Selection Formulation

First we will define mathematically the path selection problem in which our main goal is to achieve the minimization of the maximum link utilization which aims to provide load balancing to the network. The subsequent solution will find which links should be activated to create a path from each source and destination pair.

The mathematical formulation is the following:

$$min\left(\max_{\forall(i,j)\in E}\left\{\frac{\sum_{k=1}^{||\mathscr{K}||}\sum_{w=1}^{W_k} f_k^w \cdot x_{ij}^k}{c_{ij}}\right\}\right) \tag{3.3}$$

$$\sum_{k=1}^{||\mathscr{K}||}\sum_{w=1}^{W_k} f_k^w(x_{ij}^{kw} + x_{ji}^{kw}) \leq c_{ij}, \forall(i,j) \in E \tag{3.4}$$

Variables	**Description**
$\mathscr{K}$	Set of source-destination flow pair (s_k, d_k, w)
t	Slot number assigned to transmit through link i, j
T	Number of time slots available, represents total scheduling period.
f_k^w	Load of flow (w) from source-destination pair k
W_k	Set of flows per source-destination pair k
u_{ij}^t	Binary variable that states if link (i, j) is scheduled on time slot t
c_{ij}	Capacity of link (i,j)
ts	Switching delay
d_{tx}	Propagation delay
t_{over}	Overhead delay
$N(i)$	Set of neighbor nodes of i $\in V$
P_{ij}	Power radiated from node i to node j
MH_k^w	Maximum number of hops allowed for flow w of source-destination pair k
H_k^w	Number of hops that flow w from source destination pair k will go through
$PLij$	Path Loss from node i to node j
$SINR_{ij}$	SINR between node i and j
γ_{ij}	Minimum SINR for link (i, j)

Table 3.1: Optimization variables and constants

$$\sum_{j:N(i)} f_k^w x_{ij}^{kw} - \sum_{j:N(i)} f_k^w x_{ji}^{kw} = \begin{cases} f_k^w, i = s_k \in \mathscr{S} \\ -f_k^w, i = d_k \in \mathscr{D}, \\ 0, otherwise \end{cases} \quad \forall i \in V, \forall k \in \mathscr{K}, \forall w \in W_k \tag{3.5}$$

$$\sum_{j \in N(i)} x_{ij}^{kw} \leq 1, \; \forall k \in \mathscr{K}, \forall i \in V, \forall w \in W_k \tag{3.6}$$

$$\sum_{(i,j) \in E} x_{ij}^{kw} \leq MH_k^w, \; \forall k \in \mathscr{K}, \; \forall w \in W_k \tag{3.7}$$

$$x_{ij}^{kw} \in [0, 1] \;\; \forall (i, j) \in E, \;\; \forall w \in W_k, \; \forall k \in \mathscr{K}$$

Constraint (3.4) determines that for each link in the network, the summation of the flows that will be carried by the bidirectional link between two peer STA's must never be greater than the link capacity. Constraint (3.5) ensures that the minimum flow requirements are met and constraint (3.6) is defined in order to avoid having multiple paths per flow w of source destination pair k. Constraint (3.8) restricts the maximum number of hops that each flow will go through to reach its destination. This maximum number of hops MH_k^w are defined by each flow's TSPEC requirements.

3.6 Time Slot Scheduling

The main optimization goal that will be developed in this section is aimed to minimize the air interface time needed in order to send and receive all the traffic flows efficiently from source to destination. The variable u_{ij}^t, as explained before, will be a binary variable that states if timeslot number t is used to send traffic through link (i, j). The optimal solution that this optimization problem will give is the assignment of timeslots or air-time to each one of the links in order to deliver its queue traffic. The frame structure is based on the BI shown in Figure 3.3. On the DTI field of each beacon frame we assume a structure in which fixed time periods are assigned to non-intefering links according to SINR calculations, this wil allow us to assume a time-slot structure where each scheduled link is assigned time slots according to the flows that are scheduled.

Specifically in the DTI field, the optimization formulation will assign service periods to each one of the STAs on the transport nodes that will be part of each one of the paths of each individual or aggregated flows. Is assumed SP has fixed duration and available throughput.

From the path selection problem we define a set of ordered links from each source-destination pair k that each flow w will go through. This set of ordered links are represented as a vector with elements that represent the number of hops. This set of ordered links is part of the solution of the path selection problem.

$$l_k^w = [l_k^w(1), l_k^w(2), ..., l_k^w(H_k^w)] = [(s_k, j), (j, p)...., (r, d_k)] \tag{3.8}$$

In (3.8) each h-th element of the vector is a link that is part of the path between source-destination pair k for flow w. Each hop is characterized by a pair (i, j).

The objective function for the time slot scheduling problem is then as follows:

$$min \sum_{i \in V} \sum_{t=1}^{T} \sum_{j:N(i)} \left(u_{ij}^t - \frac{u_{ij}^t u_{ji}^t}{2} \right) \tag{3.9}$$

s.t.

$$\sum_{k=1}^{||\mathscr{K}||} \sum_{w=1}^{||W_k||} f_k^w (x_{ij}^k + x_{ji}^k) \leq \sum_{t=1}^{T} \frac{c_{ij}}{T} (u_{ij}^t + u_{ji}^t), \ \forall (i,j) \in E \tag{3.10}$$

$$||H_k^w||((t_s + d_{tx} + t_{over})) + \sum_{h=1}^{H_k^w} \Big(\max_{t \in [0,..,T]} (tu^t_{l_k^w(h+1)}) - \min_{t \in [0,..,T]} (tu^t_{l_k^w(h)}) \Big) d_t \leq \beta_{f_k^w}, \forall k \in \mathscr{K} \tag{3.11}$$

$$\sum_{j \in N(i)} u_{ij}^t + u_{ji}^t \leq 1, \ \forall i \in V, \ \forall t \in T, \tag{3.12}$$

$$SINR_{ij} = \frac{\frac{P_{i,j}}{L_{ij}}}{No + \sum_{\substack{k \neq i \\ k \in N(j)}} P_{kj} u_{ij}^t} \geq \gamma_e, \forall i,j \in E, \forall c \in C \forall t \in T \tag{3.13}$$

$$u_{ij}^t \in [0,1] \ \ \forall (i,j) \in E, \ \ \forall T$$

The objective function aims to minimize number of slots used by each one of the links in the network. The second term of the objective function is used to avoid counting one time slot twice if in the solution both uplink and downlink links are scheduled in the same time-slot.

Constraint (3.10) ensures that minimum throughput requirements for each one of the flows that go through each node are going to be met and that for each time slot, the scheduled flows cannot be greater than the maximum capacity of the link on a given time. In this constraints we allow flows to be scheduled on the same link providing that the duration is at most, the duration of one time slot and that TDD operation is possible on each link.

The per flow timing constraint is represented in (3.11). Here H_k represents the number of hops that each flow from source and destination pair k will go through. Each element h represents an arc (i, j) of the path defined in the path selection problem.

The delay introduced by the slot assignment is calculated by taking the difference between the time in which flow is received and the time in which it is sent in the next hop. That way we can define how much time it takes to process each traffic on each node. The other terms include the switching delay, air transmission delay (d_{t_x}) and overhead time from upper and MAC layers. The total overhead can be approximated as:

$$t_{over} = 2 * t_{SIFS} + t_{guard} + t_{PHY} \tag{3.14}$$

In (3.11) we assume that the time needed to wait for a block ACK from receiver station is negligible. The second term depends on the timeslot allocation. The constant d_t represents the scheduled air interface time that each STA is allocated in order to deliver traffic flows. The sum of both terms must be kept beneath a maximum latency threshold β_{f_k} for each type of flow. SIFS values as well as guard time and PHY header time for each frame sent by each station on each SP are based are given by the IEEE 802.11ad standard.

Constraint (3.12) exhibits the half-duplex limitations of each node. On any given time slot t, each node is only able to receive or send data through one of its links.

Initially in our problem we also define the maximum capacity of each link based on SINR measurements at a given time. The path loss calculation is made assuming a frequency range between 28GHz and 72GHz and a LOS urban environment. To calculate the path loss between each pair of nodes we use the following equation based on the alpha-beta-gamma (ABG) path loss model [16]:

$$PL_{ij}(fr, dn)[dB] = 10\,\alpha\,log_{10}\big(dn\big) + \theta + 10\Gamma\,log_{10}\big(f\big) + X_{\rho}^{ABG} \tag{3.15}$$

Where α and Γ are coefficients that show relationship between path loss and frequency, θ is an offset value for path loss in dB, f is the frequency in GHz, dn is the distance between node i and node j and X_{ρ}^{ABG} is the standard deviation that describes large-scale signal fluctuations about the mean path loss over distance.

Thus, to calculate the SINR we define the following equation:

$$SINR_{ij} = \frac{\frac{P_{ij}}{PL_{ij}}}{No + \sum_{(l,h)\in I_j} \frac{P_{lh}}{PL_{lh}} u_{lh}^{t}} \tag{3.16}$$

In (3.16) we have that the received power of node j is given by the transmission power from node i to j and the sum of the interference imposed by adjacent links scheduled in the same time-slot I_j and noise experienced by each node. The possible interferer links of pair (i, j) are predefined given the network topology and state.

However on the later results, due to some limitations regarding the software and the treatment of certain variables in the problem formulation, SINR calculations will only determine interfering links and will affect which links cannot transmit concurrently on the same time-slot. Capacity of each link will vary according to theoretical values expected for Millimeter Wave communications and this values will be stated following each case.

3.7 Joint Optimization

In the last subsection we described both sub-problems in which we solve path selection first and then find a solution for the optimal time-slot allocation. This approach finds an optimal solution for the path selection formulation and with these results, it allocates time-slots to each one of the links.

In this section we will define the joint optimization in which both objective functions are solved at the same time and the problem is treated as a multi-objective optimization.

The main constraints for both subproblems are defined in the same way, taking into account the constraint that relates the flows that are going to be scheduled on each link and the number of time slots each link will be allocated. For this matter, we combine both objective functions into a single scalar objective function that includes both path selection and timeslot allocation. Initially we include weights associated to each term of the single-objective function.

The joint problem is defined as follows:

$$min\,\gamma_1 \sum_{i\in V}\sum_{t=1}^{T}\sum_{j:N(i)}\left(u_{ij}^t - \frac{u_{ij}^t u_{ji}^t}{2}\right) + \gamma_2\left(\max_{\forall(i,j)\in E}\left\{\frac{\sum_{k=1}^{||\mathscr{K}||}\sum_{w=1}^{||W_k||} f_k^w \cdot x_{ij}^{kw}}{c_{ij}}\right\}\right) \tag{3.17}$$

$$\sum_{k=1}^{||\mathscr{K}||}\sum_{w=1}^{||W_k||} f_k^w x_{ij}^{kw} \leq \sum_{t=1}^{T}\frac{c_{ij}}{T}u_{ij}^t,\ \forall(i,j)\in E,\ \forall t\in T \tag{3.18}$$

$$u_{ij}^t \leq x_{ij}^{kw},\ \forall t\in T,\ k\in\mathscr{K},\ \forall w\in W_k,\ \forall(i,j)\in E \tag{3.19}$$

$$\sum_{j:N(i)} f_k^w x_{ij}^{kw} - \sum_{j:N(i)} f_k^w x_{ji}^{kw} = \begin{cases} f_k^w, i = s_k \in \mathscr{S} \\ -f_k^w, i = d_k \in \mathscr{D}, \quad \forall i\in V,\ \forall k\in\mathscr{K}, \forall w\in W_k \\ 0, otherwise \end{cases} \tag{3.20}$$

$$\sum_{j\in N(i)} x_{ij}^{kw} \leq 1,\ \forall k\in\mathscr{K}, \forall i\in V,\ \forall w\in W_k \tag{3.21}$$

$$\sum_{(i,j)\in E} x_{ij}^{kw} \leq MH_k^w,\ \forall k\in\mathscr{K},\ \forall w\in W_k \tag{3.22}$$

$$SINR_{ij} = \frac{\frac{P_{i,j}}{L_{ij}}}{No + \sum_{\substack{k\neq i \\ k\in N(j)}} P_{kj}u_{ij}^t} \geq \gamma_e, \forall i,j\in E, \forall c\in C \forall t\in T \tag{3.23}$$

$$x_{ij}^{kw}, u_{ij}^{t} \in [0,1] \ \forall (i,j) \in E \ \forall T, \ \forall w \in W_k$$

The benefit of multi-objective optimization, compared to single separate objective optimization as explained before is only found when multiple conflicting objectives are present. Solutions for this kind of problems face several issues when trying to find optimal feasible set of values for decision variables. When conflicting objectives are present there is a set of solutions that provide feasible and optimal outcomes for one objective goal while providing only feasible solutions for the other. This is regarded as Pareto Optimality [9]. In our case of study both minimizing maximum link utilization and minimizing the use of resource such as time-slots impose two objective goals which seem conflicting. On one hand, in order to minimize number of slots, MSDU aggregation in which we base our link-based scheduling tries to bundle flows and send them in the same SP providing this doesn't violate flow or capacity constraints on each link. However this comes at the expense of load balancing.

However, Multiobjective Optimization (MO) provides certain advantages that single objective optimization lacks. In the next chapter we will try to provide insights on how our problem formulation can be adapted to two-stage optimization and a joint optimization scheme and we will compare results obtained on both approaches on a 5G transport network scenario where multiple traffic classes are defined and scarce network resources play an important role in finding optimal solutions.

3.8 Shortest Path Heuristic Algorithm

In order to test the performance of our model, in that it provides feasible and optimal solutions to the link and resource scheduling function, we compared our results with a simple basic heuristic algorithm that assigns paths to each traffic flow and assign time slots to each one of these flows on each transport node that is part of their multihop path. We used the same scheme of MAC aggregation in which we multiplex multiple flows on one time-slot providing that they are sent to the same receiving node and within an already predefined time.

This algorithm will be regarded in the reminder of this work as the Shortest Path Heuristic Algorithm (SHPH) and it follows next the steps :

S1 Assign prioritization number to each flow depending on latency requirements and load demand. More latency restrictive flows are given highest priority number.

S2 Choose a flow with the highest priority traffic, calculate the minimum hop path for this flow.

S3 Update link scheduling for high priority traffic.

S4 Update cost and remaining capacity of already assigned links. This cost represents the number of slots already assigned to this link.

S5 Following step 5, for each remaining traffic flow determine least cost path to destination.

S6 Update cost and remaining capacity of links. Eliminate from any link who's maximum capacity has already been reached. Return to step 4

S7 After all flows have allocated paths to destination nodes, minimization of used slots for every link is applied by allowing MAC aggregation when possible.

S8 End after all flows have associated path and slot allocation.

3.9 Optimization Software

In order to asses the feasibility of our model an optimization solver was employed in order to program our mathematical model and run simulations to obtain feasible and optimal solutions in different network scenarios. Several considerations were taken in order to choose the proper optimization solver.

One of the main issues for choosing the the solver is the nature of our optimization problem. Given that in our problem we tend to discretize time in the form of time-slots and define an on/off scheme for links in the network, we constraint our problem to provide only to integer solutions. This means that decisions variables such as number of time slot and whether a link is used or not will only take binary values $(0, 1)$. This problem is regarded as an MIP problem [17]. In particular our problem is a [0,1] binary MIP problem.

MIP problems in nature are non-convex, which makes it difficult for solvers to find optimal solutions in a short period of time, specially when the number of variables increases considerably. Non-convex problems are problems in which objective and constraints are non-convex in nature. This type of problems have multiple feasible regions and multiple locally optimal points which makes scalable algorithms and search for global optimum challenging. This means that an exhaustive and systematic search has to be made to be able to find an optimal solution in the least amount of time possible. Consequently, our optimization solver must be able to do this exhaustive search with flexible algorithms, aided by a customization of search methods to meet certain performance requirements and reduce computation times.

Gurobi Optimizer is one of the fastest and most powerful solvers available in the industry. It supports linear programming, quadratic programming and mixed integer programming problems providing parallelism capabilities and cutting edge versions of basic heuristic algorithms to find feasible solutions. Gurobi Optimizer offers full range of programming and modeling language support and object oriented interfaces. On top of this it allows free academic licensing in order to model small to medium sized problems with exceptional benefits. Another main benefit of Gurobi Solver is that it allows the use of parallelization and distributed algorithms, thus enhancing performance with the use of servers and multiple processors to accelerate the optimization process. This means that, in order to obtain serious enhancements and reduce computation time for bigger problems with thousands of variables, multiple processors can be employed in order to accelerate exhaustive searches and try different algorithms and approaches to each problem. However this feature can only be obtained by using a commercial license, which for the sake and reach of this work wont be necessary.

Cutting planes and branch-bound algorithm are some of the key features that are available on the Gurobi interface for initial heuristic solutions. The parameters and reach of each heuristic method can be tuned to obtain more efficient and rapid solutions depending on the size and constraints of the network, although default settings for each one of the heuristic methods are regarded as optimal. Gurobi provides flexibility to the programmer to modify searching rules at any moment of the optimization process in favor of certain feasible solutions.

However one main drawback from the use of Gurobi Optimizer is that it limits the use of certain operations with some variables on MIP calculations. Since variables can take just 1 or 0 values, Gurobi does not allow operations like division due to possible issues of undetermined operations. Free licensing of Gurobi also prohibits the use of parallelization techniques to enhance search performance.

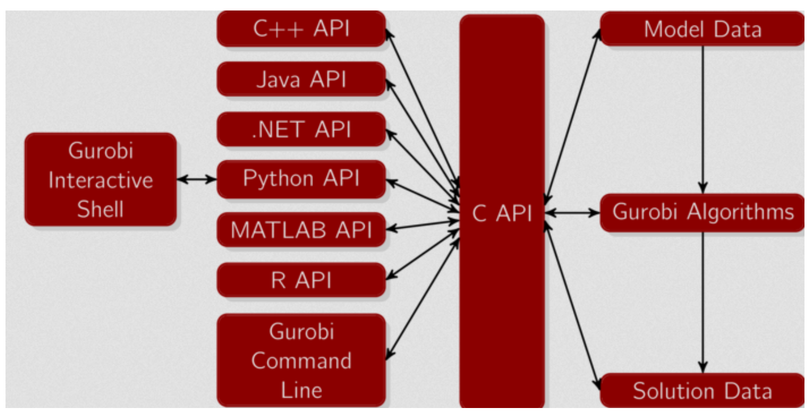

Figure 3.4: Gurobi Optimizer Framework

Main functioning and supported languages are shown in Figure 3.4. Because it provides different programming languages support and flexibility in algorithms and modeling of data, Gurobi's functioning can be extended to any size of network and a great amount of variables can be included on the models and can also be used jointly with other software to process solutions and data.

Chapter 4

Optimization Results

In this chapter we focus on applying our optimization formulation to a scenario that can be easily scaled to a real life scenario for 5G transport networks. We will optimize and test our mathematical formulation to evaluate its feasibility and accuracy in depicting how future mmWave transport networks will be organized and how mmWave technology will support SDN based deployments. For that matter real measured traffic data with real latency requirements will be used as inputs for our problem formulation and our subsequent optimization results will show the effect that large amounts of traffic will have on future mmWave based transport networks. Two-Stage Optimization (TSO), Joint Optimization (JO) and the SHPH approaches programmed and their different results analyzed in terms of functionality, flexibility, computational resource utilization and optimality. Figure4.1 shows an insight on the evaluation methodology.

4.1 Network Parameters

To analyze the performance in realistic networks of our optimization approach an initial network scenario was considered. Due to the few transport network scenarios because of the preliminary state of 5G networks we used the assumptions made in previous chapters regarding how transport networks will be organized, how the transport nodes are going to organize themselves and what amount of BH and FH

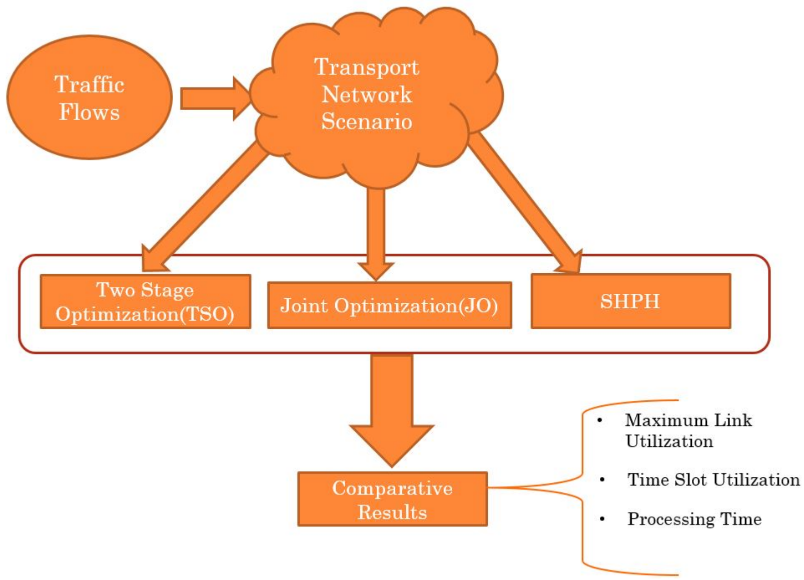

Figure 4.1: Evaluation Methodology

traffic will be generated, for example, in dense urban deployments. Mesh topologies in which both fronthaul as well as backhaul traffic is present is most likely a scenario encountered on 5G transport networks. This type of topologies are chosen because for our formulation we assume mmWave links between transport nodes, and as explained in previous chapter, features like beamforming training and sector level sweeps, allow mesh nodes to reconfigure paths easily depending on the traffic demands imposed on the network.

The terminology used for transport nodes, gateways, source and destination nodes comply with the terminology used in [18]. Edge Transport Nodes or ETN connect tenant VNF to the transport network and encapsulates tenant traffic into specific transport tunnels. These transport tunnels are mostly defined by throughput and latency requirements, source and destination addresses. Inter Area Transport Nodes (IATN) in turn support interconnection of different technologies such as wireless, optical, etc., and finally transport networks provide forwarding services and configure themselves according to the rules exposed by network controllers.

Regarding our link capacity calculations, if not stated otherwise, we assume a fixed capacity based on projections for mmWave technology with the inclusion of

Parameter	**Value**
t_{SIFS}	$3\mu s$
d_{tx}	$3\mu s$
t_s	$10\mu s$
t_{guard}	$3\mu s$
t_{PHY}	$4.79\mu s$
PHY Layer	OFDM
Link Capacity 5G	$25Gbps$
Link Capacity 4G	$4.69Gbps$
MAC aggregation	A-MSDU
d	$100m$
Frequency Band	$60GHz$
EIRP	$40dBm$
Antenna Gain	$8.5dBi$
Rain Att	$0.44dB$
Oxygen Abs	$2.3dB$

Table 4.1: Timing and Transport Node Parameters

multi-spatial streams and the use of Massive MIMO.

Link capacity for projected mmWave communications was based from projected values in IEEE 802.11ay draft [19]. This value is possible with the implementation of Massive MIMO and multi-stream transmission in the near future. Furthermore, Ericsson has claimed to achieve $25Gbps$ of mmWave capacity using this $MU-MIMO$ technology with beam-tracking on downlink [4].

For the link capacity used in 4G, the value of 4.59Gbps is chosen based on the IEEE 802.11ad standard using an MCS of 12 in an SC PHY layer implementation [8]. As for timing and power parameters used, IEEE 802.11ad provides us with base parameters which we can employ on our network simulation in any of the scenarios. These and other evaluation parameters are shown in Table 4.1.

Based on our assumptions made on previous chapters, even with the increased capacity of mmWave links in the near future, this particular technology cannot support complete centralized processing. This means that traffic classes with high data rates, as it is in split A at first will not be taken into account for our problem formulation. Rather, split B, which accounts for a more flexible and relaxed centralized processing with considerable high data rates, can be supported by mmWave technology in order to carry heavy FH flows. More lenient and flexible splits regarding data rates and distributed base band processing can also be supported. Split C for example, which represents conventional backhaul traffic, is one of the traffic classes in which mmWave communications play an important role. This type of flows can be carried efficiently with low latency and high throughput even on scenarios where there is a strong presence of heavy FH flows.

One of the main features of split C and split B flows is their dependence on the actual load of the network and cell usage, which allows to use statistical multiplexing to aggregate traffic and reduce traffic requirements. Given the dependence of these splits on actual load generated in the RAN the use of average busy hour loads to dimension 5G transport networks becomes an accurate assumption, this means that transport network should be dimensioned based on average busy hour loads rather than peak data rates shown in Table 2.1.

For LTE traffic the data rate requirements we will use are based on the Next Generation Fronthaul Interface (NGFI) [21] and Next Generation Mobile Networks (NGMN) Alliance [22]. Its calculations are based on a single 20HMz LTE carrier with 8 antenna ports and the highest MCS attainable for DL (64QAM) and UL (16QAM). Depending on the functional split chosen, data rates can vary and also latency requirements will depend on the added functionality of Base Band Units. Functional splits 1 and 4 as stated in the NGFI White Paper [21] that account for an aggregated total of 0.247 *Gbps* and 3.2 *Gbps* respectively. Functional Split 4 is based on assuming functions like Channel Estimation and Layer Mapping to be done in the RRH, which resembles functional split B on [8]. Conversely, split 1 resembles Split C in [8] in that it leaves latency restrictive functions to RRH and leaves Higher MAC functionality to BBU. For 5G traffic data rate requirements, 5G-Xhaul project forecasts are used. Table 4.2 shows the different maximum traffic demand of each class that Transport Network will handle based on (NGFI) real measurements for both projected LTE and projected 5G traffic per AP.

In order to asses the performance of our formulation we will determine traffic

Traffic Flow	**Peak Data Rate per AP**	**Latency Requirements**
5G FH	$50Gbps$	2 ms
5G BH	$10Gbps$	20 ms
LTE-A FH	$3.2Gbps$	2 ms
LTE-A BH	$0.247Gbps$	40 ms

Table 4.2: Traffic Profile

flows from each ETN that are characterized by their destination node, source node, throughput and latency requirements. For this matter, projected 5G traffic as well as projected 4G-LTE traffic are going to act as input for our evaluation. Each flow is based on the values of Table 4.2 and the rules and requirements will vary depending on the number of users and the number of flows envisioned on each source node.

4.2 Capacity and Traffic Demand Analysis

For this section, evaluations were made in which demand load of each flow increases following a load profile ranging from a low (10%) to a high (100%) load for each one of the flows generated per ETN based on the maximum values of Table 4.2.

These results were analyzed on different case scenarios in which different amount of flows are present. Following this set of tests and assuming a 60 % loaded network, link capacity was modified on each iteration to see its impact on the decision making of the optimization formulation. Comparative graphs will show the three approaches covered in this work.

The following assumptions were made regarding traffic class in order to define amount of load of each flow that is going to be scheduled:

- Maximum Data rate per AP (per source node) is given by Table 4.2.

- Each flow is defined by TSPEC requirements of its corresponding traffic (FH/BH, 4G/5G).

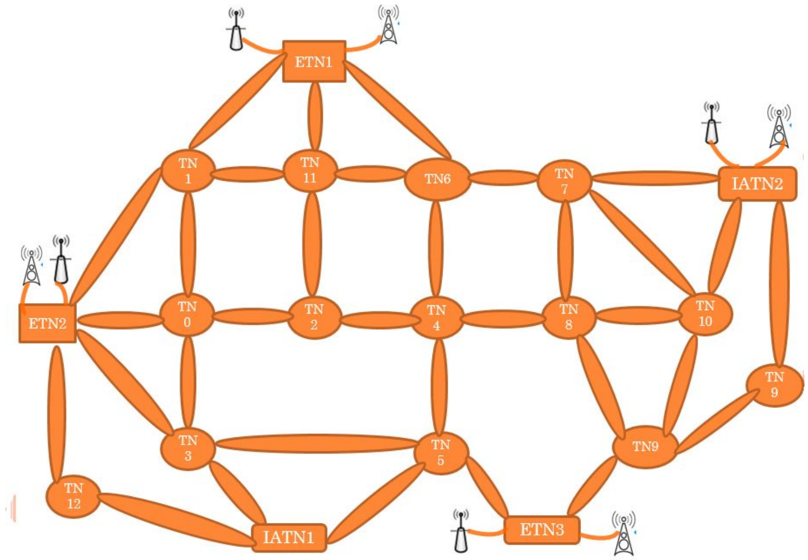

Figure 4.2: Evaluated Scenario

4.3 Network Scenario

Our evaluation scenario is composed by 12 Millimeter Wave Transport Nodes, 3 sources of traffic and 2 different destination nodes and is show in Figure 4.2. Each edge transport node will carry different flows to the destination nodes depending on their traffic class.

4.3.1 Symmetric Flows

For the first scenario, a symmetric amount of flow loads were defined per source node. This means that every source node generates the same amount of flows per traffic class to all of the different destinations. The different flows and their load for projected 5G and 4G data are shown in Tables 4.3 and 4.4.

At first we assume that each ETN has a total of 8 different flows scheduled. Four flows can be bundled into Split B traffic and 4 flows are regarded as being part of Split C for the purpose of our problem formulation. Its important to notice that the values calculated for each flow take into account the peak data rate of Table 4.2.

For visualization purposes on how the optimization approach acts and which are the links and time-slots scheduled for the optimal solution, we assume a symmetric flow distribution of 8 flows per ETN and a data rate per ETN of $30Gbps$ for FH and $10Gbps$ for BH traffic. In this case, node IATN1 will receive 4 flows from each source node representing FH traffic and IATN2 will receive 4 flows from each source representing representing Backhaul traffic.

An actual load of 60% is applied to each flow's demand. Applying our optimization formulation for this case we obtain the following network graph representation:

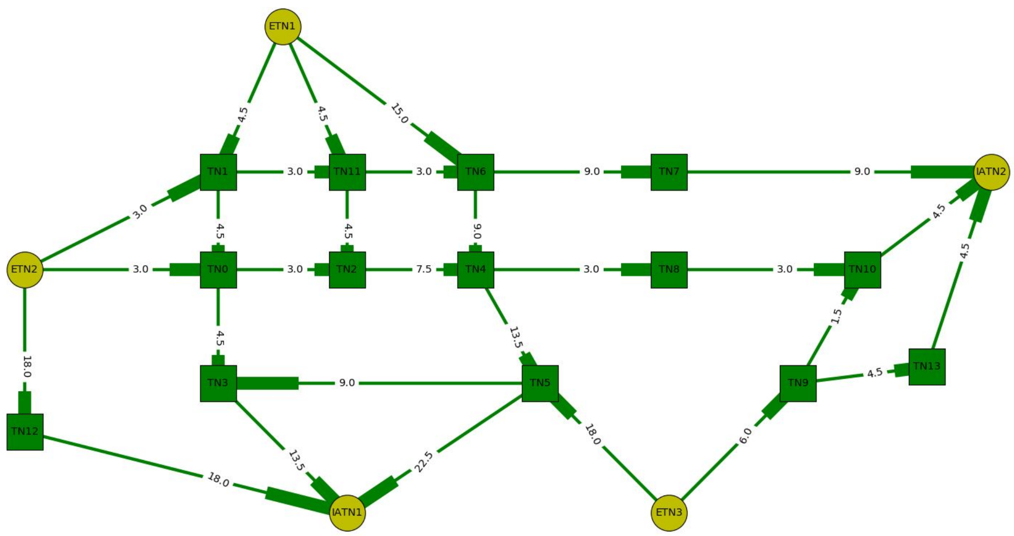

Figure 4.3: MLU-5G Symmetric Flows

The Maximum Link Utilization in this case is 90% and it is found on link between TN5 and Destination IATN1. Each one of the destination nodes receives their scheduled traffic while trying to minimize link utilization, in turn every source node distributes their flows throughout their different interfaces in order to balance load throughout all of the links.

However, because time slot scheduling is also a part of the objective optimization goal, a tradeoff is found between the two optimal solutions. Thus time slot scheduling, following the different constraints imposed on the problem formulation explained in the latter chapter, yields the distribution shown in Figure 4.4.

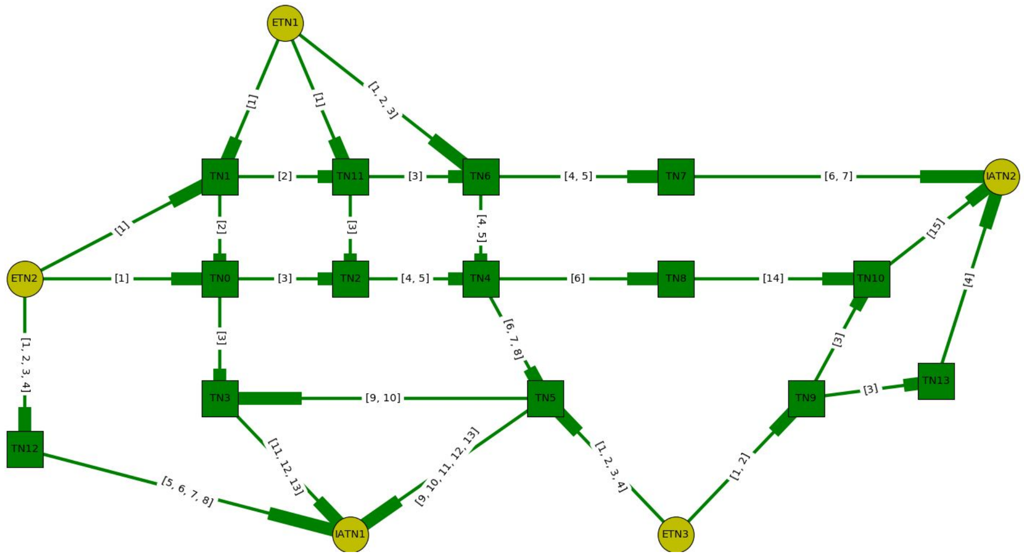

Figure 4.4: Time Slot Scheduling

S-D Pair	**Number of Flows**	**Per flow Demand**	**Type of Traffic**
ETN1,IATN1	4	12.5 Gbps	FH
ETN1,IATN2	4	2.5 Gbps	BH
ETN2,IATN1	4	12.5 Gbps	FH
ETN2,IATN2	4	2.5 Gbps	BH
ETN3,IATN1	4	12.5 Gbps	FH
ETN3,IATN2	4	2.5 Gbps	BH

Table 4.3: Symmetric flow loads-5G

S-D Pair	**Number of Flows**	**Per flow Demand**	**Type of Traffic**
ETN1,IATN1	4	800 Mbps	FH
ETN1,IATN2	4	68.5 Mbps	BH
ETN2,IATN1	4	800 Mbps	FH
ETN2,IATN2	4	68.5 Mbps	BH
ETN3,IATN1	4	800 Mbps	FH
ETN3,IATN2	4	68.5 Mbps	BH

Table 4.4: Symmetric flow loads-4G

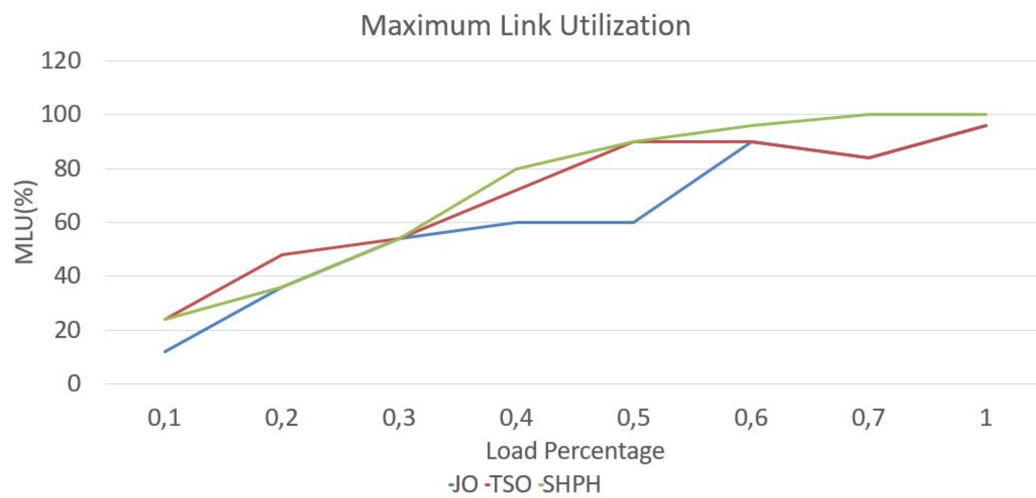

Figure 4.5: MLU-5G Symmetric Flows

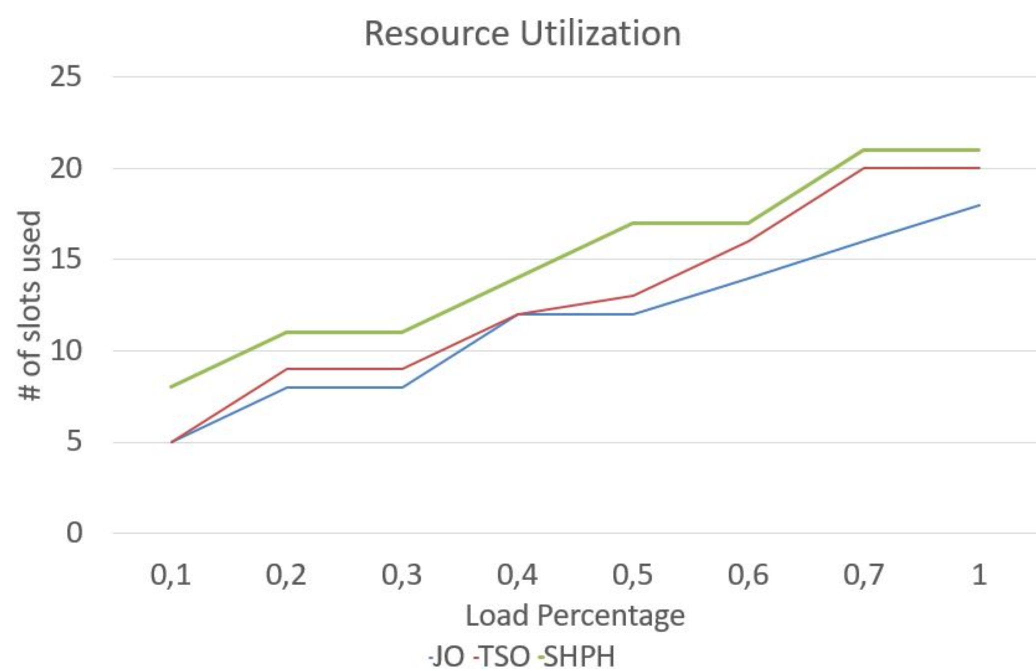

Figure 4.6: Resource Utilization-5G Symmetric Flows

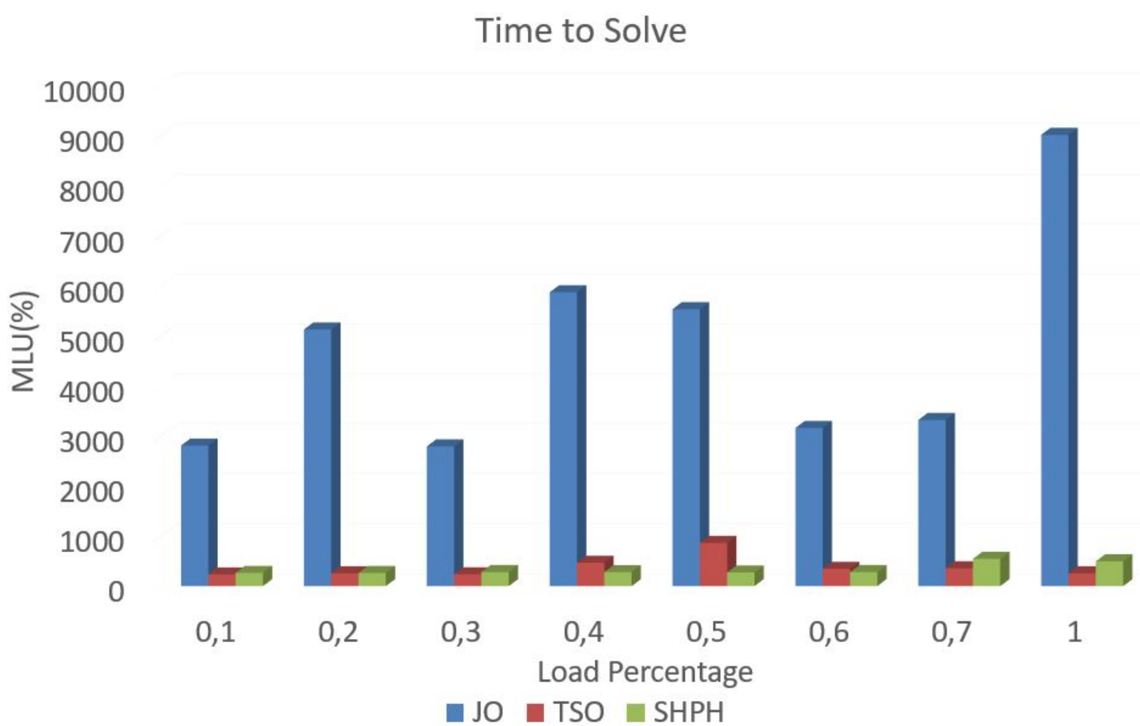

Figure 4.7: Time to Solve-5G Symmetric Flows

Results of varying load percentages

For the case in which load varies between 10%-100%, and given this network and timing specifications, Figures 4.5 - 4.10 show maximum link utilization, resource utilization, per traffic class latency and computation time evolution results as traffic load increases.

It is clear to see that Joint Optimization, at the expense of greater computing time brings a better solution for the link and resource scheduling problem. For 4G traffic, a load percentage of less than 20%, the three approaches give almost the same performance because the complete set of feasible solutions is smaller and per link flow aggregation does not have much effect on either formulation. However for higher loaded networks, the amount of feasible solutions broadens because more combinations for flow aggregation are possible. There is a better performance for the Joint Optimization in the complete spectrum of solutions. However, as it is expected, having higher loaded networks affects considerably the computation time for Joint Optimization approach. However for some cases, in which two problems are jointly optimized, one feasible solution for one problem will in fact affect the outcome of another one providing that feasibility is not compromised.

As for projections on 4G traffic it is safe to say that 802.11ad based STA'S can carry either back-haul or fronthaul traffic without heavily loading links either on low load as well as high loaded transport networks, thus supporting multi-hop paths for traffic without strong latency requirements.

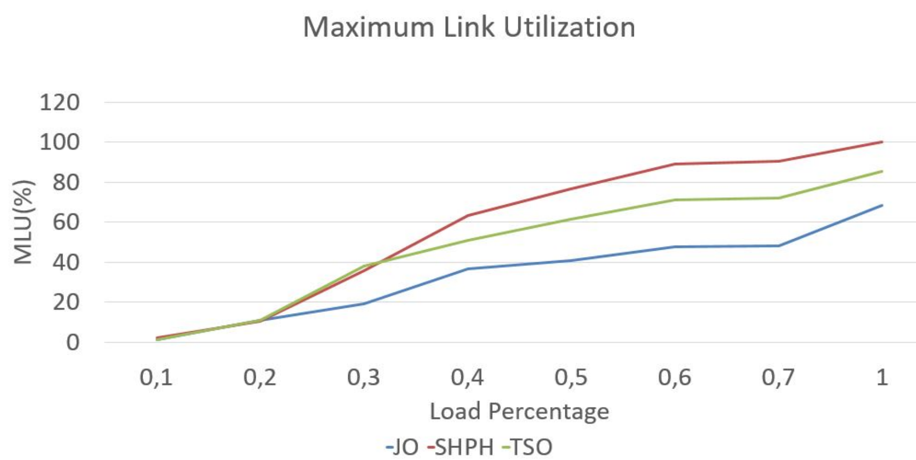

Figure 4.8: MLU-4G Symmetric Flows

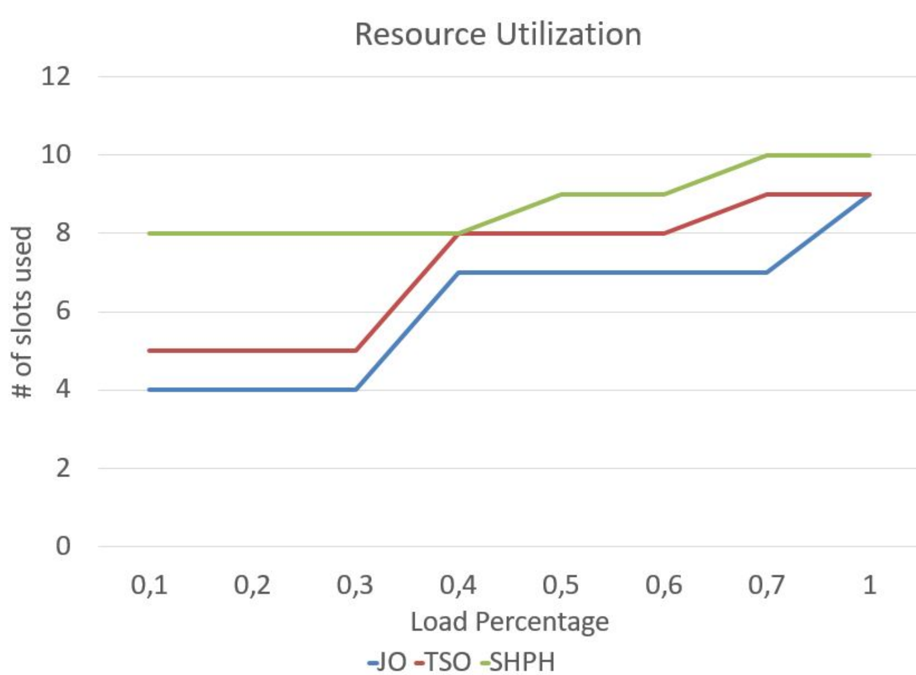

Figure 4.9: Resource Utilization-4G Symmetric Flows

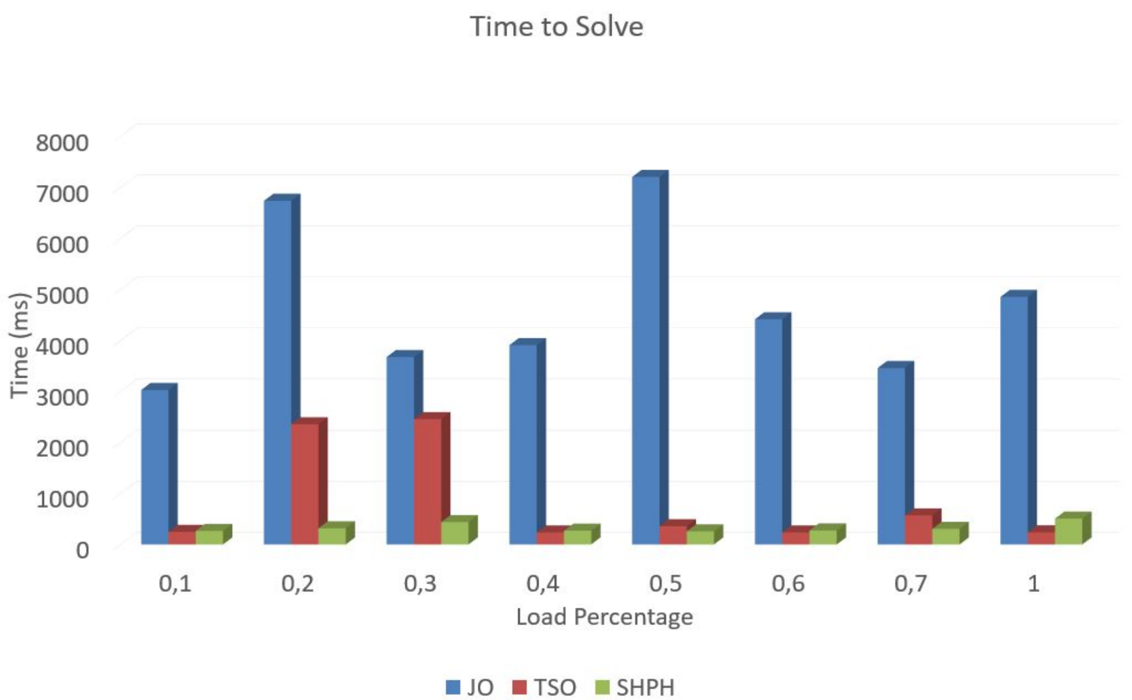

Figure 4.10: Time to Solve-4G Symmetric Flows

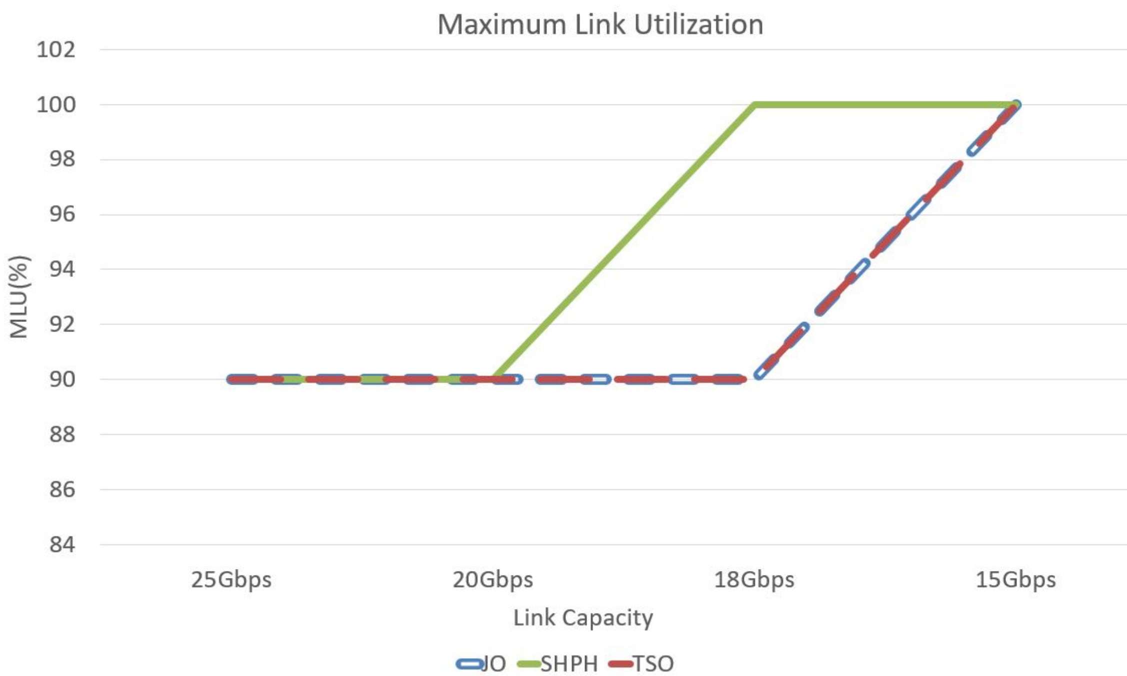

Figure 4.11: MLU-5G Symmetric Flows - Variable Capacity

Results of varying link capacities

For our capacity analysis, flows were fixed and the capacity of the links vary between $15Gbps - 25Gbps$ for 5G traffic and $4.69Gbps - 10Gbps$ for 4G Traffic. Simulations were made for both 5G based symmetric flows and 4G based symmetric flow cases.

Varying the capacity of the Millimeter Wave links in this case has small effect on the performance of the algorithm although Joint Optimization approach produces better solutions both on load balancing and resource utilization. Link capacity however is a bottleneck for Millimeter Wave Transport Networks in the way that for capacities lower than $10Gbps$, just a few heavy load flows can be scheduled efficiently. For 4G based traffic, simulations show the same behavior regarding the performance of Joint Optimization in comparison with the other two approaches. One fact we should notice is that for a Single Carrier PHY layer of IEEE 802.11ad standard with highest possible MCS (MCS=12), link utilization is rather low which shows that for projected 4G traffic including FH, Multi-Gigabit Communications could carry heavy 4G traffic flows without serious overloading or diminishing network performance.

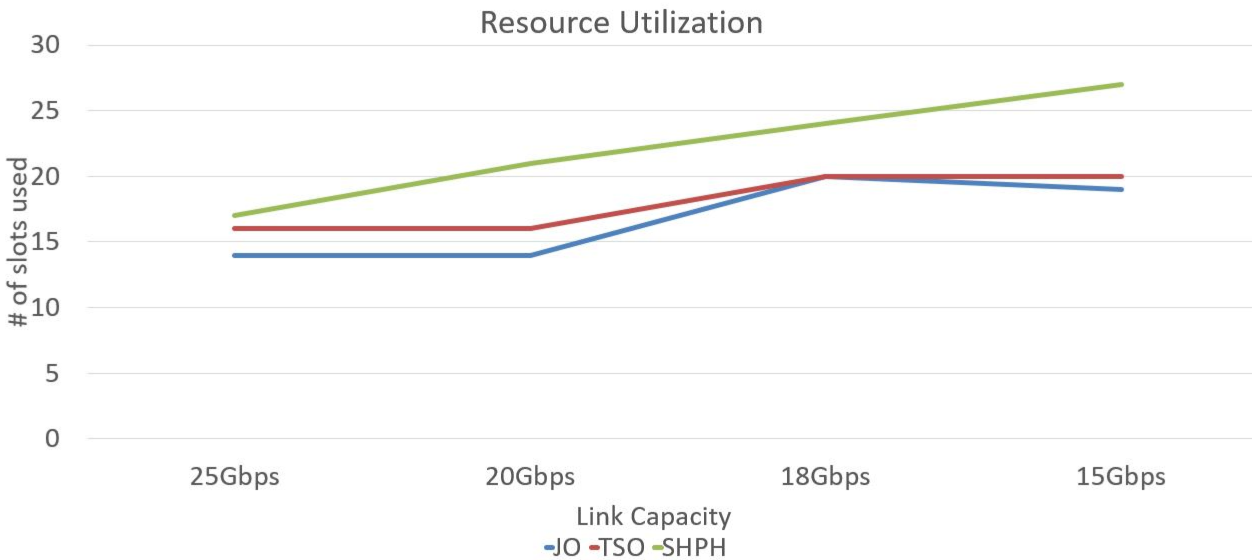

Figure 4.12: Resource Utilization-5G Symmetric Flows - Variable Capacity

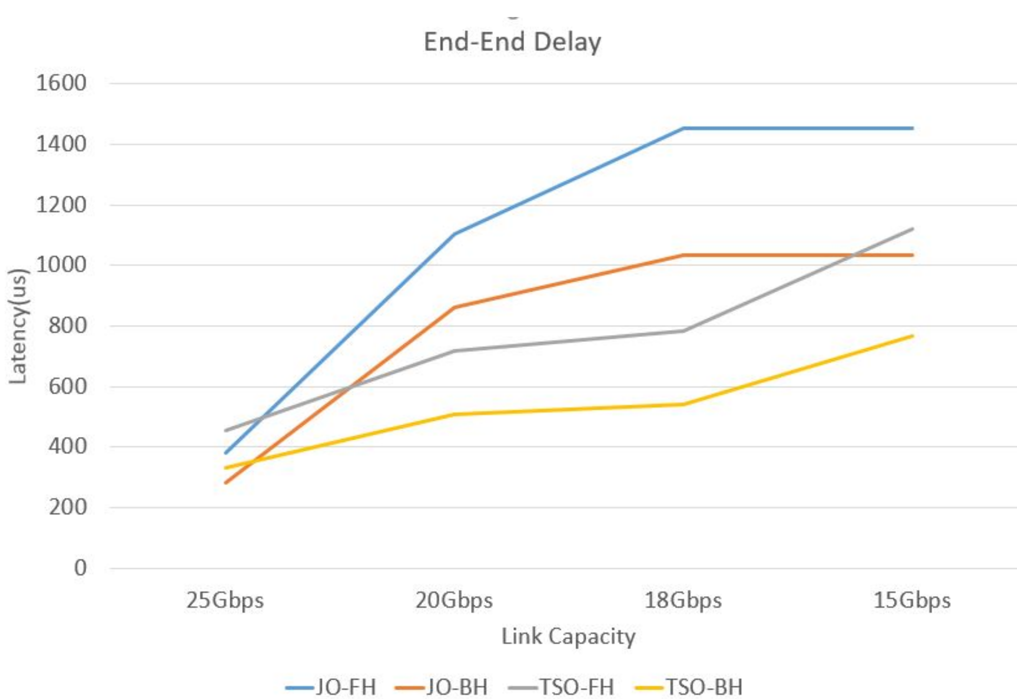

Figure 4.13: BH/FH Latency-5G Symmetric Flows

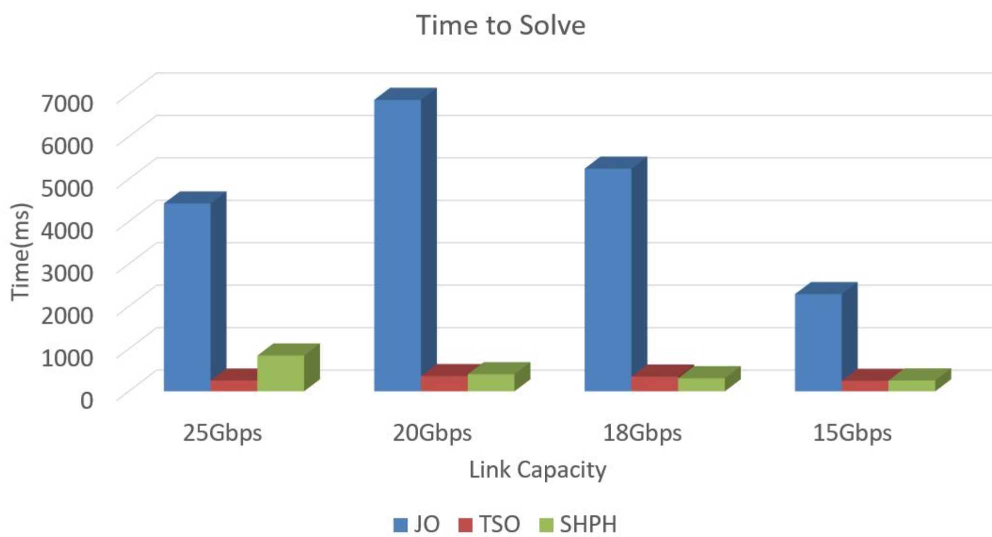

Figure 4.14: Time to Solve-5G Symmetric Flows

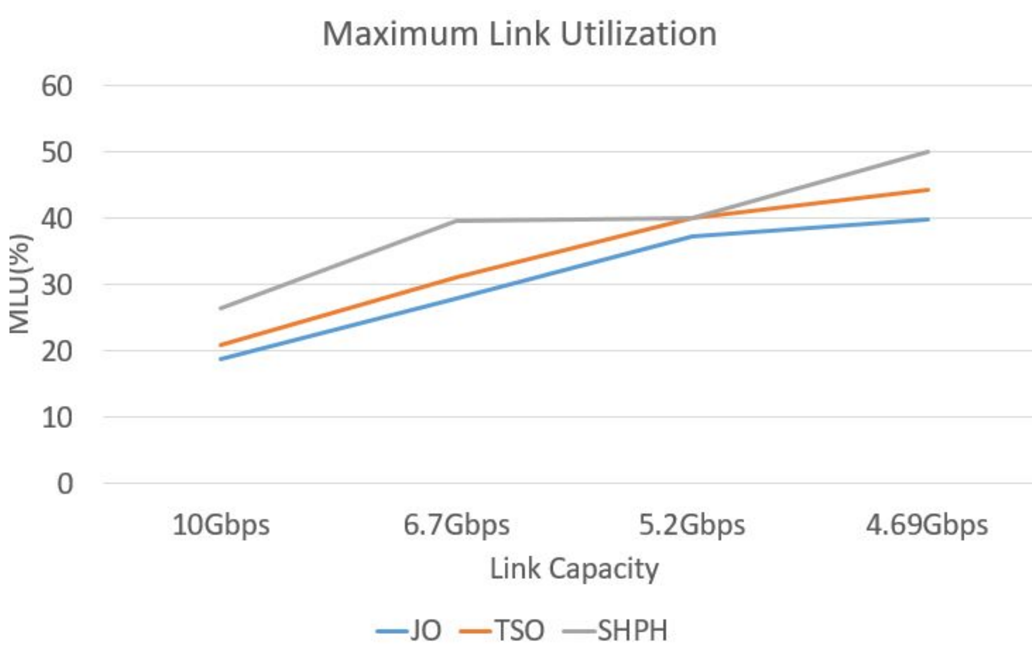

Figure 4.15: MLU-4G Variable Capacity-Symmetric Flows

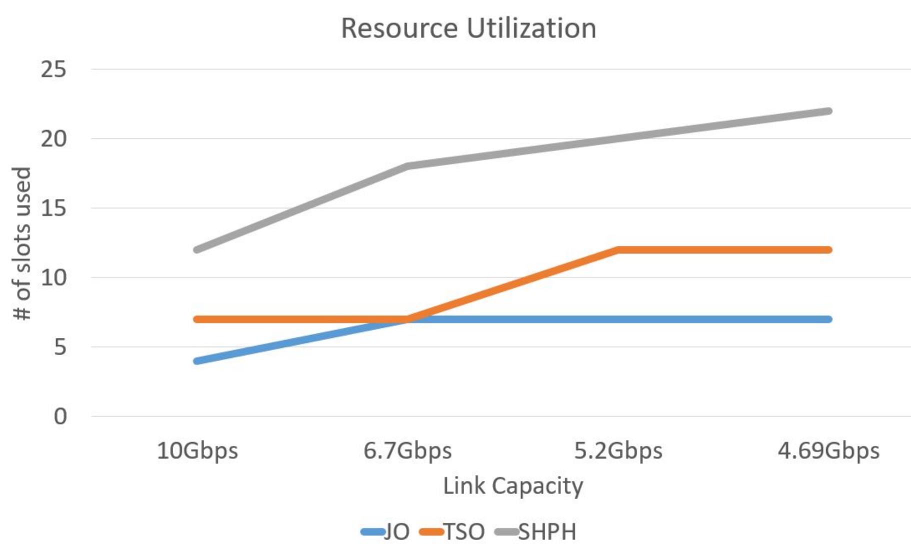

Figure 4.16: Slots-4G Variable Capacity-Symmetric Flows

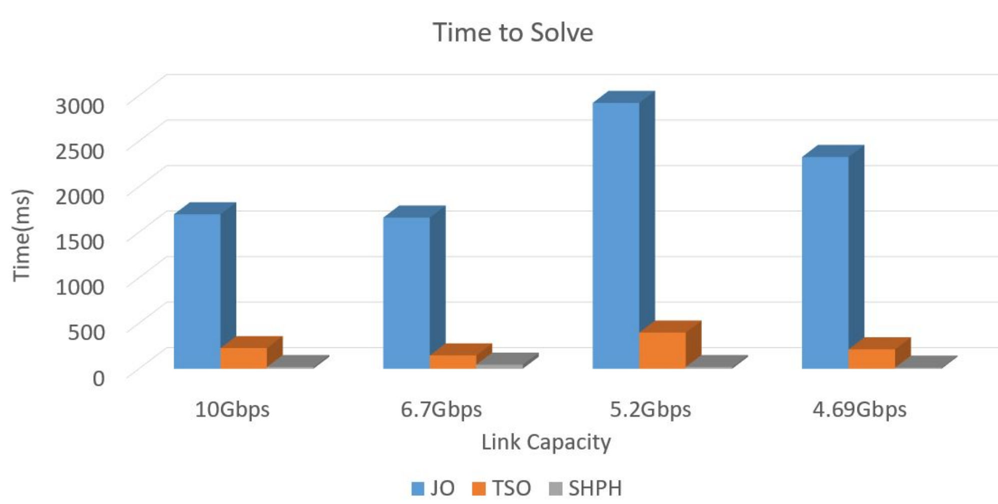

Figure 4.17: Time to Solve-4G Variable Capacity-Symmetric Flows

4.3.2 Asymmetric Flows

Now we assume that on each ETN , a non-symmetric number of flows per source node are generated. Again as the latter example, per AP or ETN, the peak data rate per ETN is based on Table 4.2. The traffic flows assumed for the asymmetric scenario are shown in Tables 4.5 and 4.6.

S-D Pair	**Number of Flows**	**Per flow Demand**	**Type of Traffic**
ETN1,IATN1	5	6 Gbps	FH
ETN1,IATN2	4	2.5 Gbps	BH
ETN2,IATN1	4	7.5 Gbps	FH
ETN2,IATN2	3	3.3 Gbps	BH
ETN3,IATN1	3	3.3Gbps	BH
ETN3,IATN2	4	7.5 Gbps	FH

Table 4.5: Asymmetric flows-5G

For example, for Source ETN2 we have a total of 30 Gbps of FH and 10 Gbps of BH traffic. For pair ETN2-IATN1, we have 4 flows of FH traffic which means each flow will have $f_{FH} = 30Gbps/4 = 7,5Gbps$ and $f_{BH} = 10Gbps/3 = 3,3Gbps$. Figures 4.18 - 4.26 shows the results obtained by changing the amount of load of the network and also varying the capacity of the links.

S-D Pair	**Number of Flows**	**Per flow Demand**	**Type of Traffic**
ETN1,IATN1	5	0.64 Gbps	FH
ETN1,IATN2	4	91 Mbps	BH
ETN2,IATN1	4	1.6 Gbps	FH
ETN2,IATN2	3	45 Mbps	BH
ETN3,IATN1	3	54 Mbps	FH
ETN3,IATN2	4	0.137 Gbps	BH

Table 4.6: Asymmetric flow loads-4G

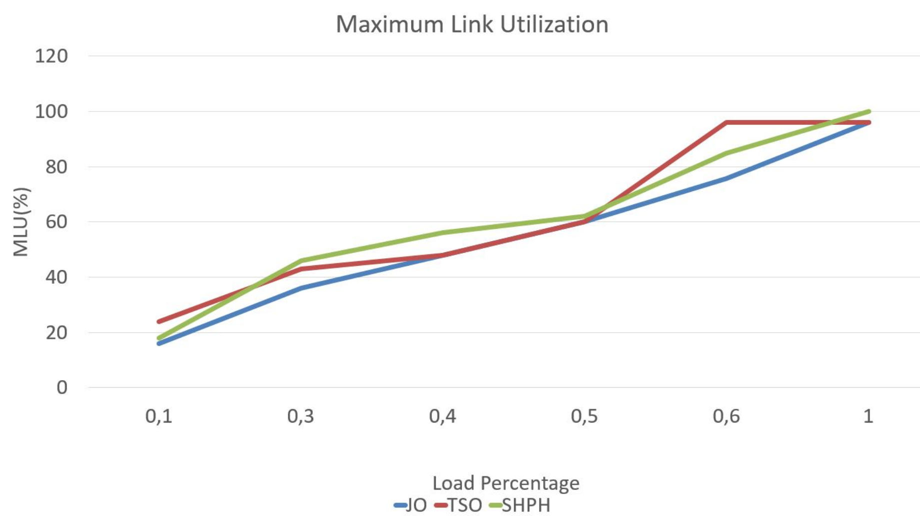

Figure 4.18: MLU-5G Asymmetric Flows

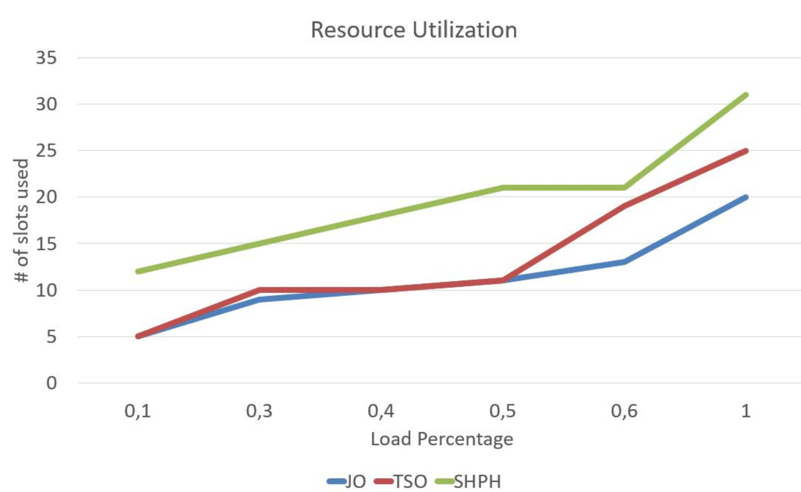

Figure 4.19: Resource Utilization of Asymmetric Flows-5G

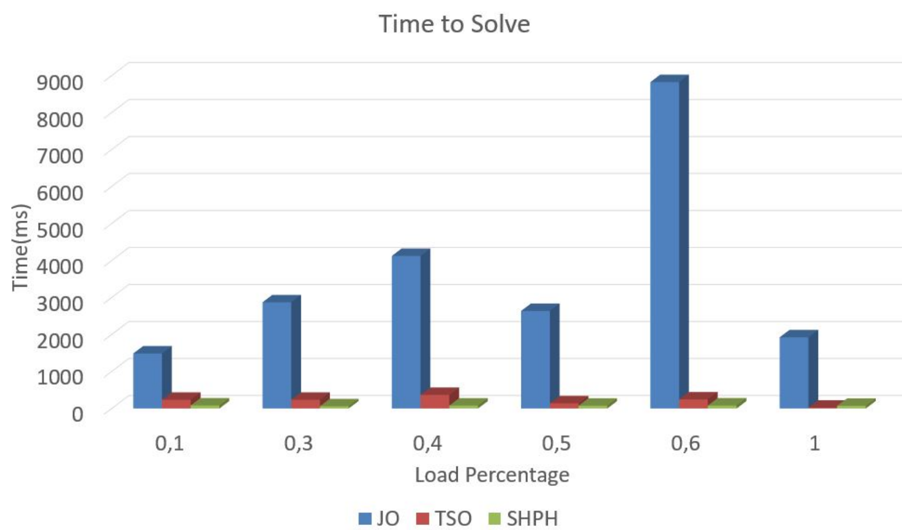

Figure 4.20: Time to Solve-5G Asymmetric Flows

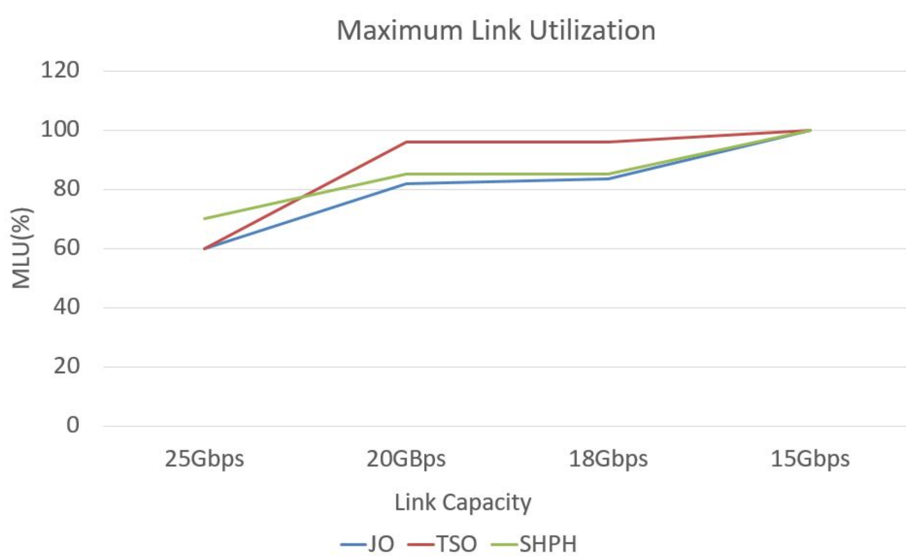

Figure 4.21: MLU-5G Asymmetric Flows-Variable Capacity

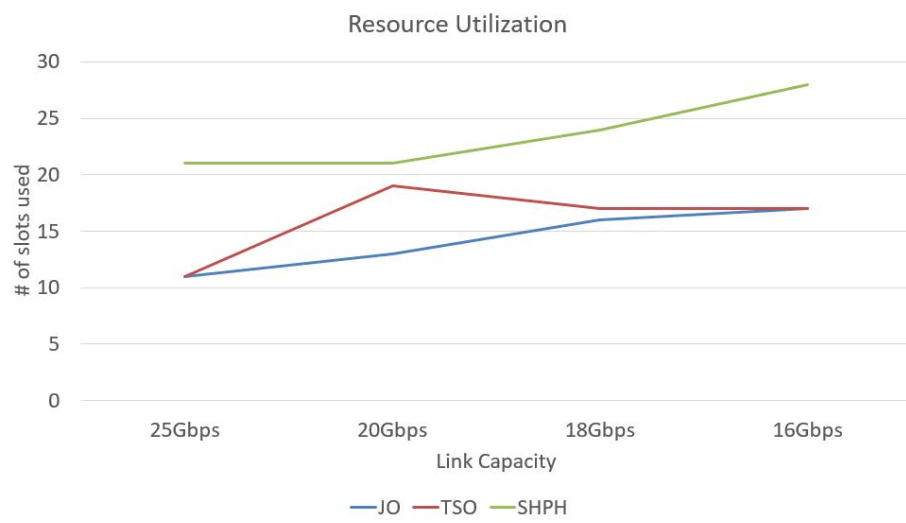

Figure 4.22: Slots-5G Asymmetric Flows-Variable Capacity

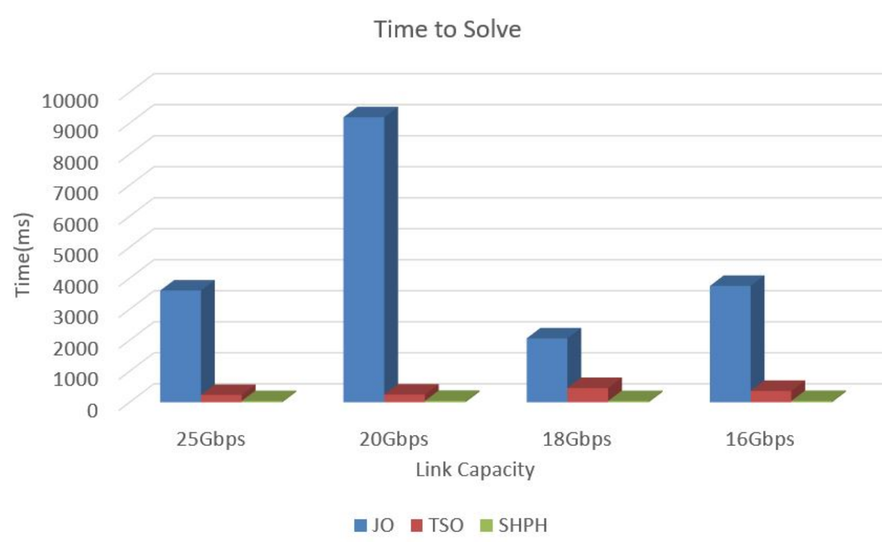

Figure 4.23: Time to Solve-5G Asymmetric Flow loads-Variable Capacity

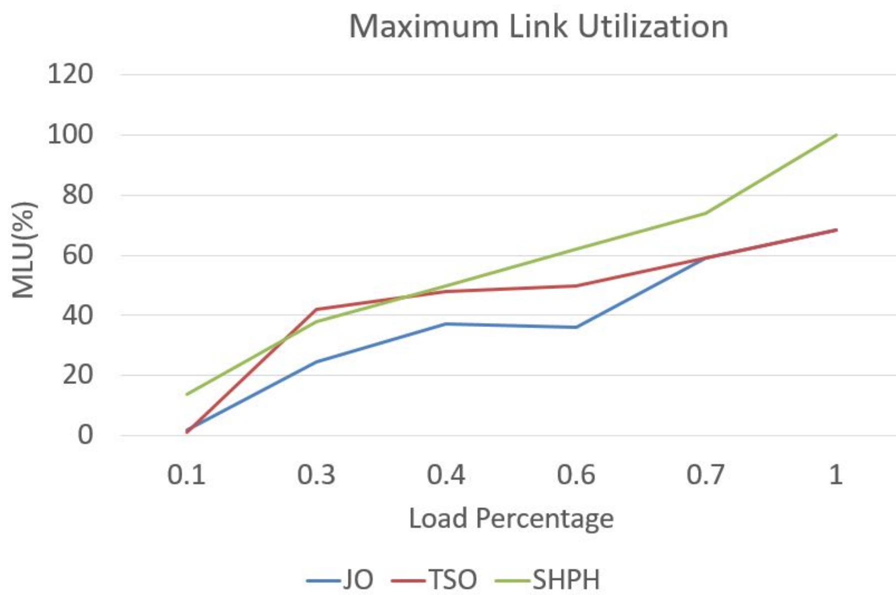

Figure 4.24: MLU-4G Asymmetric Flow

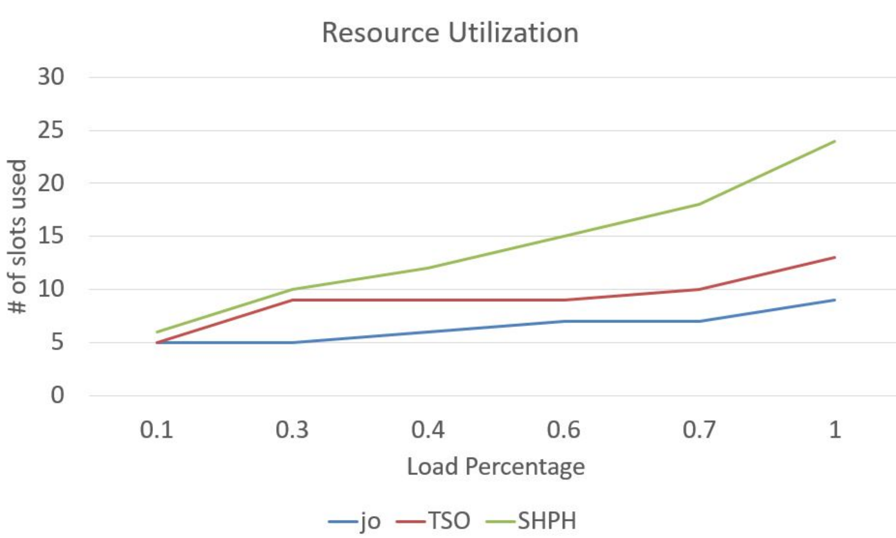

Figure 4.25: Resource Utilization-4G Asymmetric Flow loads

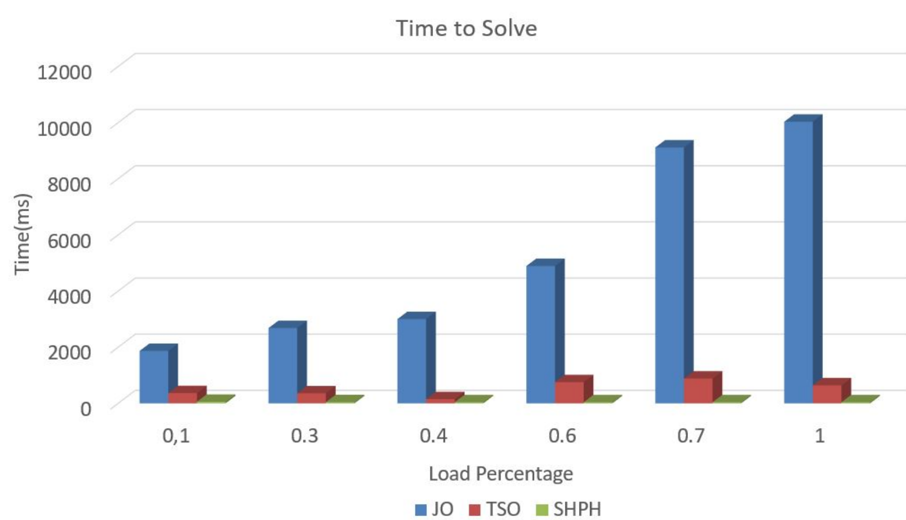

Figure 4.26: Time to Solve-4G Asymmetric Flow loads

Results of varying load percentages

It is clear that having asymmetric flows compared to symmetric flows in our scenario shows similar results regarding maximum link utilization due to the similar loads of flows from both cases. However a small gain is obtained. At 40% of peak data rate per AP almost 45% is the maximum link utilization, as opposed to our symmetric flow's case scenario where at 40% of load, maximum link utilization is approximately 60%. However we can see a performance enhancement in Resource Utilization coming from our formulation because allowing MSDU multiplex reduces the number of time-slots or service periods needed in order for STAs to send or receive frames regardless on how variable traffic flow is.

Results of varying link capacities

Varying the capacity of the links has almost the same effect compared to the case where symmetric flows were same loaded flows were present. However for the case of Asymmetric flows, when capacity is reduced, and because its reduction is not considerable, flows can be aggregated and feasible solutions increase if there are two or more flows that can be aggregated on same air interface frame. Regardless on aggregation, Joint Optimization performs better than SHPH or Two-stage optimization in most of the cases except in computation time measurements. Gains are truly visualized on the total resource allocation for a certain capacity.

4.3.3 Variable Demand

Finally for this same scenario, random flow demands where generated per source node. The load of each flow follows a random distribution, in our case with maximum rate given by the peak data rates in Table 4.2 with statistical multiplexing. A total of 10 Realizations were made and the results for MLU and Resource Utilization, for both 5G and 4G based traffic are shown in Figures 4.27-4.29 respectively.

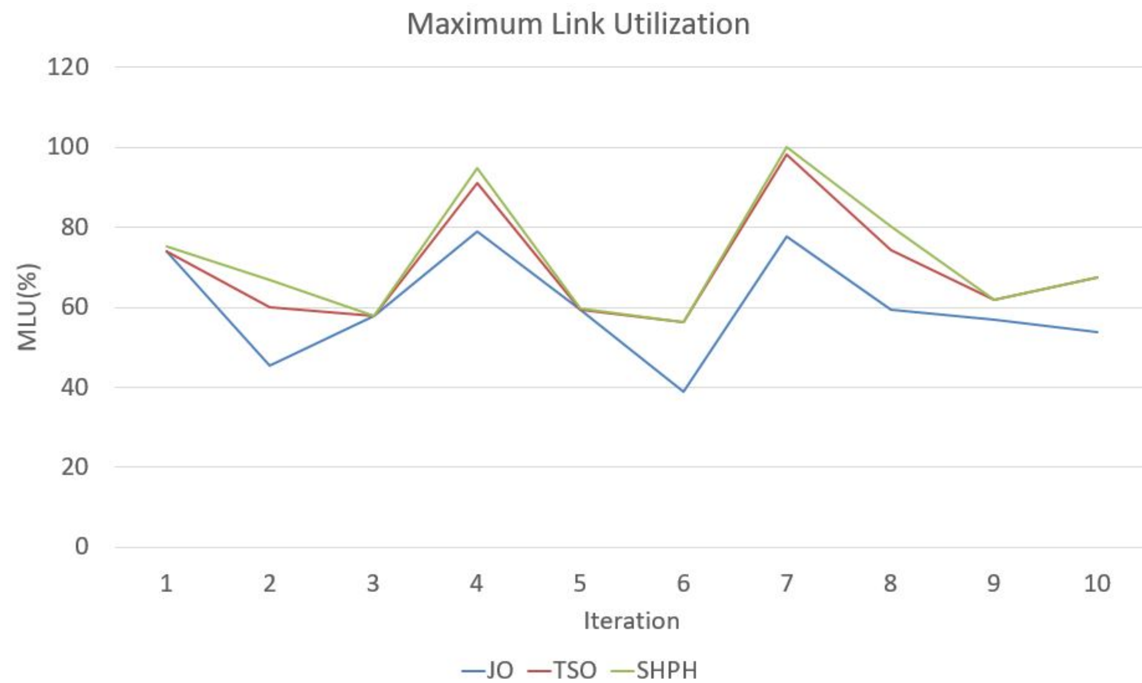

Figure 4.27: 5G MLU Variable Data Rate

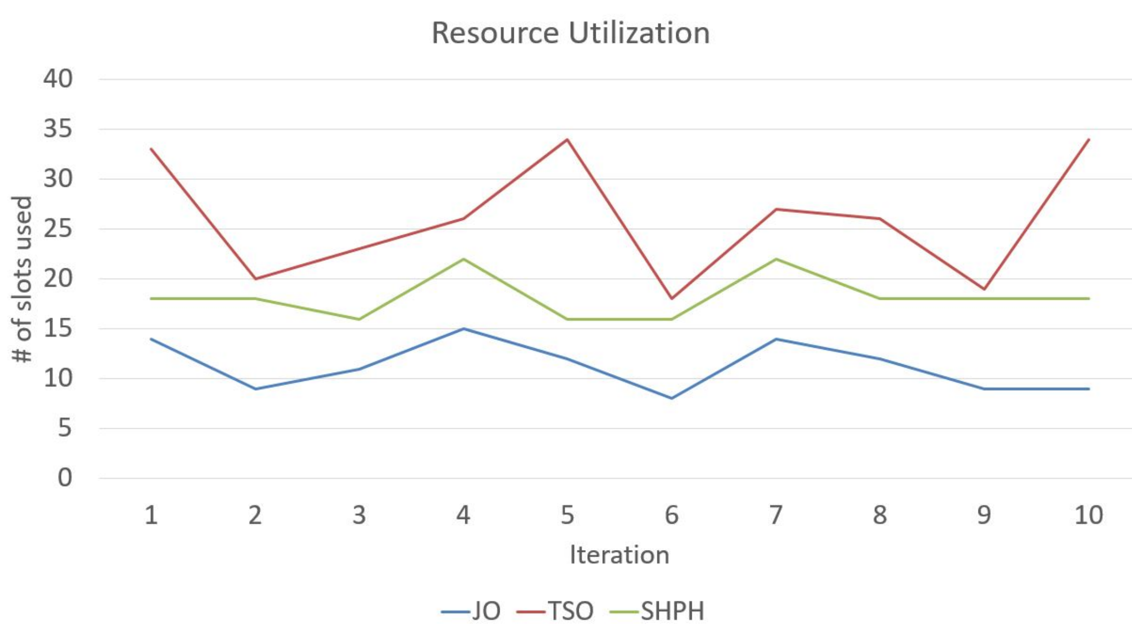

Figure 4.28: 5G Timeslots Used-Variable Data Rate

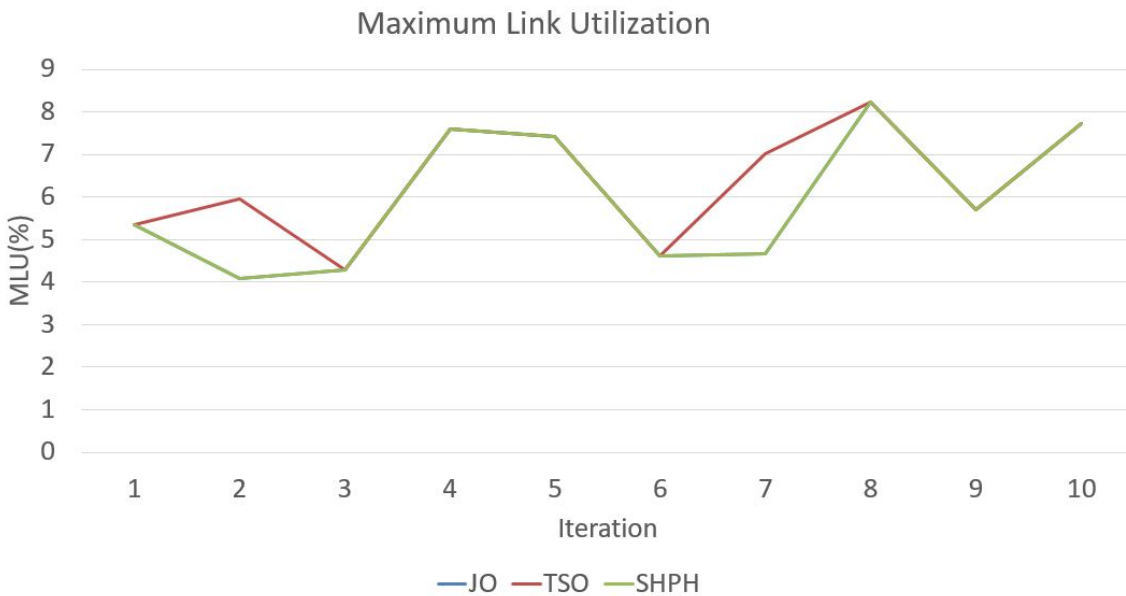

Figure 4.29: 4G MLU Variable Data Rate

Again, Joint Optimization of link scheduling and time-slot allocation does perform better than both the completely heuristic approach and a Two-Stage optimization. This however comes at the expenses of increasing computation times and the need for a more robust search algorithm that aims to find feasible optimal solutions that could more robust search algorithm. Compared to TSO and SHPH algorithms, Joint Optimization has a 60% maximum gain with respect to the Two stage Approach and 25% with respect to the SHPH algorithm. In a few cases, joint optimization and two stage optimization have the same value for both variables but due to the through search done on Joint Optimization by the optimization solver, allows the solution to be the optimal one. This again comes at the expense of losing computational efficiency.

Chapter 5

Future Work and Conclusions

As viewed in the early Chapters of this work and the results obtained from the evaluation methodology proposed we can draw some considerable conclusions and considerations, can be taken into account to further the advance on this topic and provide enhancements in order to obtain higher performance and scalability.

First, millimeter Wave is a technology that can act as an enabler for future 5G transport networks in terms of underlying transport technology used. Reconfigurability and flexibility of the network and how it is organized and how each one of the elements cooperate ,is one key features that have to be taken into consideration for further advances and designs for 5G Networks. Robust control, which can manage the network according to certain rules and service requirements is one of the main challenges but also potential opportunities in order to achieve said flexibility. Diverse sources of different traffic encountered throughout all segments of the 5G network demand underlying transport technologies that can carry heavy flows of data while meeting QoS specifications in an optimal manner for every traffic flow. Performance enhancements can be obtained by defining a set of network performance goals, such as, load balancing, minimize energy consumption and optimize network reconfiguration in case of element failure are some of these network functions necessary for all networks, specially 5G Networks. Thus adopting an optimization formulation approach to provide simplicity and computational scalability using mathematical formulations, proves to be an optimal solution. Moreover, defining and switching between multiple objectives according to the use case and network current state information can aid network controller to manage resources more efficiently.

In this work we aimed to address two optimization objective or criteria, in order to evaluate them on a possible scenario on Transport Networks in 5G Networks in which Millimeter Wave acts as the underlying technology and is in charge of communicating transport nodes. Some conclusions can be made regarding our findings:

- Millimeter Wave technology, as a transport network physical layer technology, with projected enhancements in achievable data rates, is one technology that can provide flexibility in terms of network topology due to the reconfigurable nature of their links through the use of highly directive antennas. Although beam-forming delay and sector level sweeping was not taken into account in our problem formulation, the delay issued by beam-forming is negligible compared to scheduling delay, switching delay and per hop delay intrdouced by the network.

- Link scheduling and resource scheduling as shown by the results, can be jointly optimized, achieving optimal results at the expense of utilizing more computational time to find a solution to the MOP problem. However, given aggregation of flows to one timeslot with the use of flow aggregation on each transport node, further increaese in performance is attained. However on a more realistic network, synchronization is a main issue that has to be taken into account when flow aggregation is addressed and where multiple paths are defined. Having different PBSS within a network provides a challenge to STAs in order to communicate with their peers or PBSS controller without significant loss of information. However, when attained, flow aggregation could provide the necessary flexibility to support multiple different traffic flows on each scheduling interval.

- Optimization software, as any kind of limited processing limited operation, has scalability issues when bigger problems with more variables are applied. Being able to separate a big problem into several smaller problems and searching for a common solution, can boost the performance and reduce the computational burden on the optimizer. Concurrent optimization as well as distributed optimization approaches, are of great importance and should differentiate which optimization solvers are suitable for certain types of problems. In this work a simple approach of an exhaustive search method with tunable parameters and search engines gives an insight on the potential for distributed processing when bigger scenarios are presented. As insight, in Annex A-C the python based optimization of the Link and Resource Scheduling is presented.

- The use of multiobjective schemes with highly exhaustive searches could pose a problem due to the high volatility of future mobile networks and their need to be quickly adaptable and highly reliable. However having multiple objectives functions allows to have flexibility on how the search for the optimal point is made depending on certain controllers criteria or optimization performance requirements.

Appendix A

Python Code

A.1 Joint Optimization

```
################# Initialization of Model Formulation#############
z=Model('Transport')

############ Initialize Link  Scheduling Variable#######

x={}
for w in w:
   for s,d in sdpair:
    for r in nodes:
        for j in Neigh.get(r):
            if r not in DN or j not in SN:
                x[r,j,s,d,w]=z.addVar(vtype=GRB.BINARY,name='x'+'_'+
                str(r)+'_'+str(j)+'('+str(s)+'_'+str(d)+')')
z.update

############ Initialize Time-slot allocation variable#######
u={}
for ts in range(1,Tmax+1):
   for w in W:
         for i,j in arcs:
             u[i,j,w,ts]=z.addVar(vtype=GRB.BINARY,name='u'+'_'+'N'+
             str(i)+'_'+'N'+str(j)+'_'+str(ts))

############ Initialize Time-slot allocation variable#######
#
```

```
for r in SN:
    for d in DN:
        if (r,d) in sdpair and demand[r,d,w]!=0:
            for w in W:
              for j in Neigh.get(r):
               z.addConstr(quicksum(u[r,j,t]
               for t in range(1,Tmax+1))>=1)
z.update()


for r in DN:
    for s in SN:
        if (s,r) in sdpair and demand[s,r,w]!=0:
            for j in Neigh.get(r):
                for w in W:
                    z.addConstr(quicksum(u[j,r,t]
                    for t in range(1,Tmax+1))>=1)
z.update()


  ###Per flow conservation CONSTRAINTS#####
for s,d in sdpair:
    for w in W:
        for r in IN:
           z.addConstr(quicksum(demand[s,d,w]*x[i,j,s,d,w]
           for i,j in arcs.select('*',r)) ==
           quicksum(demand[s,d,w]*x[j,k,s,d,w]
           for j,k in arcs.select(r,'*')))

for s,d in sdpair:
   for r in nodes:
       for w in W:
           if r==s:
             z.addConstr(quicksum(demand[s,d,w]*x[r,j,s,d,w]
             for j in Neigh.get(r)  ) -

             quicksum(demand[s,d,w]*x[l,r,s,d,w]
             for l in Neigh.get(r) )==demand[s,d,w])

           elif r==d:
             z.addConstr(quicksum(demand[s,d,w]*x[r,j,s,d,w]
             for j in Neigh.get(r) ) -
             quicksum(demand[s,d,w]*x[l,r,s,d,w]
```

```
                for l in Neigh.get(r) )==-demand[s,d,w])

#
    for s,d in sdpair:
        for w in w:
           for r in SN:
               if r!=s:
                  z.addConstr(quicksum(x[r,j,s,d,w]*demand[s,d,w]
                    for j in Neigh.get(r))==0)
    z.update()


    for s,d in sdpair:
        for r in DN:
             for w in w:
                 if r!=d:
                     z.addConstr(quicksum(x[j,r,s,d,w]*demand[s,d,w]
                     for j in Neigh.get(r))==0)


    z.update()



 #######Minimum Throughput Constraint######## ###
for r in nodes:
        for j in Neigh.get(r):
            z.addConstr(quicksum(demand[s,d,w]*x[r,j,s,d,w]
            for s,d in sdpair)<= quicksum((u[r,j,t])*(capacity[r,j]/5)
            for t in range(1,Tmax+1)))
z.update()
 #######Minimum Throughput Constraint######## ###


 #######Link Capacity Constraint####### ###
    for r in nodes:
            for j in Neigh.get(r):
                   z.addConstr(quicksum(demand[s,d,w]*x[r,j,s,d,w]
                   for s,d in sdpair for w in W)<= capacityIn)
    z.update()

    for t in range(1,Tmax+1):
        for ts in range(1,t+1):
```

```
            for r in IN:
                for j in Neigh.get(r):
                    for k in Neigh.get(r):
                        if j!=k:
                            z.addConstr(t*u[j,r,t]+ts*u[r,k,ts]<=t)


#########Minimum SINR Constraint##################
for r in nodes:
  A=[]
  for j in Neigh.get(r):
     A=[(h,g) for (h,g) in arcs if (h,g)!=(r,j) ]
     for t in range(1,Tmax+1):
       IntPower=0
       for (k,l) in A:
          IntPower+=(arcpower[k,l]-pathLoss(float(distance[k,l])))*u[k,l,t
       RecvSign=powerTX-pathLoss(distance[r,j])
       SINRTot=RecvSign-(NoiseFig+IntPower)

       z.addConstr(RecvSign-(NoiseFig+IntPower)>=MinRecv[r])
#########Minimum SINR Constraint##################

 #############Per Flow Delay Constraints##################
for s,d in sdpair:
 TotDelay=0
 AA=find_all_paths(SN,DN,Graph,s,d)
 if len(AA)!=1:
   for e in range(1,len(AA)):
       hops=AA[e]
       Delay=0
       TotDelay=0
       for w in range(1,len(hops)-1):
           i=hops[w-1]
           j=hops[w]
           h=hops[w+1]

           TotDelay+=quicksum(deltaTime*ts*u[j,h,ts] for ts
           in range(1,Tmax+1))-quicksum(deltaTime*ts*u[i,j,ts]
           for ts in range(1,Tmax+1))

       for p in range(1,len(hops)):
           i=hops[p-1]
           j=hops[p]
```

```
           Delay+=quicksum(u[i,j,ts] for ts in range(1,Tmax+1))*(tover)

       SWDelay=(len(AA)-1)*(tsw)
       z.addConstr(TotDelay+SWDelay+Delay<=delayCon[s,d])
 #############Per Flow Delay Constraints#################

 #############Maximum Number of Hops#################
for s,d in sdpair:
    TotDelay=0
    AA=find_all_paths(SN,DN,Graph,s,d)
    if len(AA)!=1:
       for e in range(1,len(AA)):
           hops=AA[e]
           NumHops=0
           for w in range(1,len(hops)-1):
               i=hops[w-1]
               j=hops[w]
               h=hops[w+1]
               NumHops+=x[i,j,s,d,w]
           z.addConstr(NumHops<=MaxHops[s,d])
 #############Maximum Number of Hops#################


 #############Gurobi's Optimization Search Parameters#################
z.params.Threads=8
z.params.MIPGap=branch

z.optimize()
```

A.2 Two Stage Optimization

```
################# Initialization of Model Formulation#############
z=Model('Transport')

############ Initialize Link  Scheduling Variable#######

x={}
for tc in TC:
   for s,d in sdpair:
    for r in nodes:
        for j in Neigh.get(r):
            if r not in DN or j not in SN:
```

```
                x[r,j,s,d,tc]=z.addVar(vtype=GRB.BINARY,name='x'+'_'+
                str(r)+'_'+str(j)+'('+str(s)+'_'+str(d)+')')
z.update


u={}
for ts in range(1,Tmax+1):
   for tc in TC:
         for i,j in arcs:
             u[i,j,tc,ts]=z.addVar(vtype=GRB.BINARY,name='u'+'_'+'N'+
             str(i)+'_'+'N'+str(j)+'_'+str(ts))


###Joint Optimization Objective Function####
SumaLinks2=0
for (i,j) in arcs:
  if i not in SN and j not in DN:
     SumaLinks2+=quicksum(demand[s,d,tc]*x[i,j,s,d,tc]+
     demand[k,l,tc]*x[j,i,k,l,tc] for s,d in sdpair for k,l in sdpair
     for tc in TC)/capacity[i,j]
z.update()

SumaLinks=0
for i in nodes:
    if i in IN or SN:
      aa=quicksum((u[i,j,tc,t]-(u[i,j,tc,t]*u[j,i,tc,t])/2) for t
      in range(1,Tmax+1) for j in  Neigh.get(i)  )
      SumaLinks+=aa

#####Set Multiobjective Goal for Link and Resource Scheduling#####
z.setObjective(y1*SumaLinks+y2*SumaLinks2,GRB.MINIMIZE)
z.update()
#####Set Multiobjective Goal for Link and Resource Scheduling#####


## Optimization  Objective Definition (Minimize Maximum Link Utilization)

#############Flow Constraints################
for s,d in sdpair:
  for r in IN:
     m.addConstr(quicksum(demand[s,d,tc]*x[i,j,s,d,tc] for i,j in
     arcs.select('*',r) ) ==  quicksum(demand[s,d,tc]*x[j,k,s,d,tc]
     for j,k in arcs.select(r,'*') ) )
```

```

for s,d in sdpair:
   for r in nodes:
       if r==s:
         m.addConstr(quicksum(demand[s,d,tc]*x[r,j,s,d,tc] for j
         in Neigh.get(r)  ) - quicksum(demand[s,d,tc]*x[l,r,s,d,tc]
         for l in Neigh.get(r) )==demand[s,d,tc])
       elif r==d:
         m.addConstr(quicksum(demand[s,d,tc]*x[r,j,s,d,tc] for j
         in Neigh.get(r) ) - quicksum(demand[s,d,tc]*x[l,r,s,d,tc]
         for l in Neigh.get(r) )==-demand[s,d,tc])

########################Flow Constraints#############


################Avoid Unwanted link assignments############
for s,d in sdpair:
    for r in SN:
       for tc in TC:
          if r!=s:
             m.addConstr(quicksum(x[r,j,s,d,tc]*demand[s,d,tc]
             for j in Neigh.get(r))==0)
m.update()

for s,d in sdpair:
  for r in DN:
    for tc in TC:
      if r!=d:
        m.addConstr(quicksum(x[j,r,s,d,tc]*demand[s,d,tc]
        for j in Neigh.get(r))==0)

m.update()

for r in SN:
    for s,d in sdpair:
        for tc in TC:
            for j in Neigh.get(r):
                   m.addConstr(x[j,r,s,d,tc]==0)
m.update()

for r in DN:
    for s,d in sdpair:
        for tc in TC:
```

```
                for j in Neigh.get(r):
                        m.addConstr(x[r,j,s,d,tc]==0)
m.update()
###########Avoid Unwanted link assignments###########


#######Minimum Assignment of Slots per Demand ############
for r in SN:
  for d in DN:
    if (r,d) in sdpair and demand[r,d,tc]!=0:
        for j in Neigh.get(r):
           z.addConstr(quicksum(u[r,j,tc,t] for t in range(1,Tmax+1))>=1)
z.update()

for r in DN:
  for s in SN:
     if (s,r) in sdpair and demand[s,r,tc]!=0:
       for j in Neigh.get(r):
          z.addConstr(quicksum(u[j,r,tc,t] for t in range(1,Tmax+1))>=1)
z.update()
#######Minimum Assignment of Slots per Demand ############


#############Minimum Throughput Constraint##################
for r in nodes:
  for j in Neigh.get(r):
      z.addConstr(quicksum(demand[s,d,tc]*x[r,j,s,d,tc] for
          s,d in sdpair)<= quicksum((u[r,j,tc,t])*(capacity[r,j]/5)
          for t in range(1,Tmax+1)))
z.update()
#########Maximum Capacity Constraint###############


########Maximum Hop Count per S-D Pair#############
for s,d in sdpair:
   m.addConstr(quicksum(x[r,j,s,d,tc]
   for i,j in arcs)<=maxHops[s,d,tc])
#####Maximum Hop Count per S-D Pair###############



#########  Optimize Model #################
m.optimize()
```

```
###########  Optimize Model ###############



########Define Scheduled Links per S-D Pair#################

solution2=m.getAttr('x',x)########Get Optimal Solution###########
arcSolS={}
arcpowerS={}
arcLossS={}
for s,d in sdpair:
    for i,j in arcs:
            if solution2[i,j,s,d,tc]!=0:
                temp={(i,j):capacity[i,j]}
                temp1={(i,j):arcpower[i,j]}
                temp2={(i,j):arcLoss[i,j]}
                arcSolS.update(temp)
                arcpowerS.update(temp1)
                arcLossS.update(temp2)
arcsD={}
arcSol=arcSolS.keys()#######Get Arcs###########
arcSol=tuplelist(arcSol)

############Arc Demand ##############
labeldemand={}
for i,j in arcSol:
        SumaFlujo=0
        for s,d in sdpair:
                if solution2[i,j,s,d,tc]!=0:
                    SumaFlujo+=solution2[i,j,s,d,tc]*demand[s,d,tc]
        temp={(i,j):SumaFlujo}
        labeldemand2.update(temp)
#############Arc Demand #############


#########Determine Maximum Link Utilization##########
maxLink=list(labeldemand2.values())
maximumA=max(maxLink)
arreglo=(float(maximumA)/capacityIn)*100
################Determine Maximum Link Utilization#######

b=m.RunTime*1000#######Define Time to Solve##########
demandas={}
```

```
demandas=solution2#####Define Demand per arc#####

######Input to Time Slot Allocation

##########Define Model For Time Slot Allocation Variables######
z=Model('TmeSlot')
z.reset()

u={}
for ts in range(1,Tmax+1):
   for tc in TC:
      for i,j in arcSol:
        u[i,j,tc,ts]=z.addVar(vtype=GRB.BINARY,name='u'+'_'+ 'N'+
        str(i)+'_'+'N'+str(j)+'_'+str(ts))
        u[j,i,tc,ts]=z.addVar(vtype=GRB.BINARY,name='u'+'_'+ 'N'+
        str(j)+'_'+'N'+str(i)+'_'+str(ts))
z.update()
############Define Model For Time Slot Allocation Variables##

##Determine Objective Function (Minimize Time Slot Allocation)##
SumaLinks=0
for i in nodes:
  if i in SN or IN:
   aa=quicksum((u[i,j,tc,t]-(u[i,j,tc,t]*u[j,i,tc,t])/2) for j
   in Neigh.get(i) for t in range(1,Tmax+1)
   if (i,j) in arcSol and (i,j) in arcSol)
   SumaLinks+=aa
z.setObjective(SumaLinks,GRB.MINIMIZE)
z.update()
###Determine Objective Function (Minimize Time Slot Allocation)##


#########Minimum Throughput Requirement Constraint##############
for i,j in arcSol:
   z.addConstr(quicksum(float(demandas[i,j,s,d,tc]*demand[s,d,tc])
   for s,d in arcsD.get((i,j)))<=quicksum(u[i,j,tc,t]*(capacity[i,j]/5)
   for t in range(1,Tmax+1)))
z.update()
#########Minimum Throughput Requirement Constraint#####


############Maximum Bidirectional Slot Allocation Constraint#####
for r in nodes:
```

```
    for i,j in arcSol.select(r,'*'):
        z.addConstr(quicksum(u[i,j,tc,t]+u[j,i,tc,t]
        for t in range(1,Tmax+1)) <=Tmax)
z.update()

for r in nodes:
        for i,j in arcSol.select(r,'*'):
            z.addConstr(quicksum(u[i,j,tc,t]+u[j,i,tc,t]
            for t in range(1,Tmax+1)) <=Tmax)
z.update()


#########Maximum Bidirectional Slot Allocation Constraint##########
for t in range(1,Tmax+1):
    for ts in range(1,t+1):
       for r in IN:
          for j in Neigh.get(r):
              for k in Neigh.get(r):
                  if j!=k :
                     z.addConstr(t*u[j,r,tc,t]+ts*u[r,k,tc,ts]<=t)
######Maximum Bidirectional Slot Allocation Constraint##########


######## Maximum End-End Delay Constraint per Flow############
for s,d in sdpair:
  TotDelay=0
  DelayOver=0
  w=0
  if demand[s,d,tc]!=0:
    AA=tuplelist(newflows.get((s,d)))#Define Paths from Src to Dst
    if len(AA)!=1:
      for w in range(1,len(AA)-1):
      i=AA[w-1]
      j=AA[w]
      h=AA[w+1]
      TotDelay+=quicksum(deltaTime*ts*u[j,h,tc,ts] for ts in
      range(1,Tmax+1))-quicksum(deltaTime*ts*u[i,j,tc,ts]
      for ts in range(1,Tmax+1))##AMPDU Air Interface Transmission#####
      TotDelay2=quicksum(deltaTime*ts*u[i,j,tc,ts]
      for ts in range(1,Tmax+1)):

      for p in range(1,len(AA))
         i=AA[p-1]
```

```
            j=AA[p]
            DelayOver+=quicksum(u[i,j,tc,ts] for ts
            in range(1,Tmax+1))*(tover)
            ######OverHead Delay per TXOP############
        SWDelay=(len(AA)-1)*(tsw) ##########Per hop Switching Delay

        z.addConstr((DelayOver+SWDelay+TotDelay)<=delayCon[s,d])######

######### Maximum End-End Delay Constraint per Flow#########


################## Minimum Link SINR Constraint ###########
for i,j in arcSol:
  if (i,j) in arcPBSS1:
    A=[(h,g) for (h,g) in arcSol if (h,g) in arcPBSS1 ]
  elif (i,j) in arcPBSS2:
    A=[(h,g) for (h,g) in arcSol if (h,g) in arcPBSS2]

  elif (i,j) in arcPBSS3:
     A=[(h,g) for (h,g) in arcSol if (h,g) in arcPBSS3]

  for t in range(1,Tmax+1):
     PL=0
     for h,g in A:
        PL+=(math.pow(10,(arcpower[h,g])/10))/(math.pow(
        10,(pathLoss(distance[h,g])/10)))*u[h,g,tc,t]#Path Loss
     PathLossTX=math.pow(10,(pathLoss(distance[i,j]))/10)
     powerTX=math.pow(10,(arcpower[i,j]/10))
     SINRarc=powerTX/PathLossTX
     SINRTot=PL/SINRarc#######SINR Calculation################
     z.addConstr(SINRTot >= (1/4)*u[i,j,tc,t])
################ Minimum Link SINR Constraint ############


############### Maximum Hop Constraint ############
for s,d in sdpair:
    TotDelay=0
    AA=find_all_paths(SN,DN,Graph,s,d)
    if len(AA)!=1:
       for e in range(1,len(AA)):
           hops=AA[e]
           NumHops=0
           for w in range(1,len(hops)-1):
```

```
                i=hops[w-1]
                j=hops[w]
                h=hops[w+1]
                NumHops+=x[i,j,s,d,tc]
            z.addConstr(NumHops<=MaxHops[s,d])
############## Maximum Hop Constraint ############


######### Tuning Parameters for MIP models #################

z.params.Threads=4 ###Define Number of Threads used for solution search
z.params.MIPGap=0.05#####Define Gap between Best Objective Bound

########### Tuning Parameters for MIP models ##############


z.optimize()#####Solve Optimization ##########
solutio5=z.getAttr('x',u)
############# Calculation of End-End Delay per Traffic Class############

DatosDelay={}
for s,d in sdpair:
  if demand[s,d,tc]!=0:
     TotDelayB=0
     DelayOverB=0
     AA=newflows.get((s,d))
     for x in range(1,len(AA)-1):
       i=AA[x-1]
       j=AA[x]
       h=AA[x+1]

       TotDelayA=quicksum(deltaTime*ts*solutio5[j,h,tc,ts] for ts in
       range(1,Tmax+1))-quicksum(deltaTime*ts*solutio5[i,j,tc,ts]
       for ts in range(1,Tmax+1))

       TotDelayB+=abs(TotDelayA.getValue())
       for p in range(1,len(AA))
         i=AA[p-1]
         j=AA[p]
         qqa=quicksum(solutio5[i,j,tc,ts] for ts
         in range(1,Tmax+1))*(tover)
         DelayOverB+=abs(qqa.getValue())
       SWDelay=(len(AA)-1)*(tsw)
```

```
       aa=TotDelayB+DelayOverB+SWDelay
       nn=aa*1e6
       nn=abs(nn)
       temp={(s,d):nn}
       DatosDelay.update(temp)
       print ('Pair (%s,%s) Total Delay=%f %s' %
                        (s,d,nn,'us'))

########### Calculation of End-End Delay per Traffic Class###########


###########Define Time slots assigned to each node###################
Rsend={}
Rrec={}
NumSlotsRec=0
NumSlotsSend=0
for r in nodes:
     te=[]
     tr=[]
     pair=[]
     a=arcSol.select(r,'*')
     for i,j in a:
         for t in range(1,Tmax+1):
          for tc in TC:
             if solutio5[i,j,tc,t]==1:
                 te.append('A'+'('+str(i)+'_'+str(j)+')'+','+str(t))
                 NumSlotsRec+=1
     temp={r:['S',sorted(te)]}
     Rsend.update(temp)
     aa=arcSol.select('*',r)
     for i,j in aa:
         for t in range(1,Tmax+1):
          for tc in TC:
                if solutio5[i,j,tc,t]==1:
                    tr.append('A'+'('+str(i)+'_'+str(j)+')'+','+str(t))
                    NumSlotsSend+=1

     temp1={r:['R',sorted(tr)]}
     Rrec.update(temp1)

###############Define Time slots assigned to each node################
a=z.RunTime*1000  ####Determine Time To Solve Time-Slot allocation
```

```
solution2=z.getAttr('x',x)
solution3=z.getAttr('x',u)

arcSolS={}
arcpowerS={}
arcLossS={}
for s,d in sdpair:
    for i,j in arcs:
        if solution2[i,j,s,d,tc]!=0:

          temp={(i,j):capacity[i,j]}

          temp1={(i,j):arcpower[i,j]}
          temp2={(i,j):arcLoss[i,j]}
          arcSolS.update(temp)
          arcpowerS.update(temp1)
          arcLossS.update(temp2)
arcsD={}
arcSol=arcSolS.keys()
arcSol=tuplelist(arcSol)



###############Define Traffic Carried through each Link############3
for i,j in arcs:
    dm=[]
    for s,d in sdpair:
           if solution2[i,j,s,d,tc]!=0:
               dm.append((s,d))
               arcos={(i,j):set(dm)}
               arcsD.update(arcos)


flows={}
for s,d in sdpair:
    dm=[]
    for i,j in arcSol:
    #           print arcSol
           if solution2[i,j,s,d,tc]!=0:
               dm.append((i,j))
               a=set(dm)
               arcos={(s,d):sorted(a)}
               flows.update(arcos)
```

```
##    print flows
newflows={}
for s,d in sdpair:
    if demand[s,d,tc]!=0:
        arco=[]
        A=find_all_paths(SN,DN,Graph,s,d)
        Q=sorted(tuplelist(flows.get((s,d))))
        Q=[h for (h,g) in Q]
        Q.append(d)
#             print Q,s,d
        for e in range(0,len(A)):
         iff=sequences_contain_same_items(A[e],Q)
         if iff==True:
             arco={(s,d):A[e]}
             newflows.update(arco)


labelarc={}
for i,j in arcSol:
    sd=[]
    for s,d in sdpair:
            if solution2[i,j,s,d,tc]!=0:
                sd.append(str(s)+'_'+str(d))

    temp={(i,j):sorted(sd)}
#    print temp
    labelarc.update(temp)
#    print labelarc

labeldemand={}
for i,j in arcSol:
    if i not in SN and j not in SN and i not in DN and j not in DN:
        SumaFlujo=0
        for s,d in sdpair:
                if solution2[i,j,s,d,tc]!=0:
                    SumaFlujo+=solution2[i,j,s,d,tc]*demand[s,d,tc]
        temp={(i,j):SumaFlujo}
        labeldemand.update(temp)

#    Suma2=0
#    for i,j in labeldemand.keys():
#        Suma2+=labeldemand.get((i,j))/100
#    print Suma2
```

```
#
Suma2=0
for i,j in labeldemand.keys():
        Suma2+=labeldemand.get((i,j))
Suma2=Suma2/len(labeldemand.keys())
#     print labeldemand
#     print Suma2


labeldemand2={}
for i,j in arcSol:
      SumaFlujo=0
      for s,d in sdpair:
            if solution2[i,j,s,d,tc]!=0:
                  SumaFlujo+=solution2[i,j,s,d,tc]*(demand[s,d,tc]/1000)
      temp={(i,j):str(SumaFlujo)+'Gbps'}
      labeldemand2.update(temp)
#     print labeldemand
maxLink=list(labeldemand.values())

#b=m.RunTime*1000
maximumA=max(maxLink)

Rsend={}
Rrec={}
NumSlotsRec=0
NumSlotsSend=0
for r in nodes:
      te=[]
      tr=[]
      pair=[]
      a=arcSol.select(r,'*')
      for i,j in a:
            for t in range(1,Tmax+1):
                if solution3[i,j,tc,t]==1:
                        te.append('('+str(i)+'_'+str(j)+')'+'S'+str(t))
                        NumSlotsRec+=1
          temp={r:['S',sorted(te)]}
          Rsend.update(temp)
          aa=arcSol.select('*',r)
          for i,j in aa:
                  for t in range(1,Tmax+1):
                          if solution3[i,j,tc,t]==1:
```

```
                    tr.append('('+str(i)+'_'+str(j)+')'+'S'+str(t))
                    NumSlotsSend+=1
      temp1={r:['R',sorted(tr)]}
      Rrec.update(temp1)

DatosDelay={}
#     print newflows
for s,d in sdpair:
    if demand[s,d,tc]!=0 :
     TotDelay=0
     AA=newflows.get((s,d))
     Delay=0
     for x in range(1,len(AA)-1):
        i=AA[x-1]
        j=AA[x]
        h=AA[x+1]
        qq=quicksum(deltaTime*ts*solution3[j,h,tc,ts]for ts
        in range(1,Tmax+1))-quicksum(deltaTime*ts*solution3[i,j,tc,ts]
        for ts in range(1,Tmax+1))
        TotDelay+=abs(qq.getValue())

        for p in range(1,len(AA)):
           i=AA[p-1]
           j=AA[p]
           qqa=quicksum(solution3[i,j,tc,ts] for ts in
           range(1,Tmax+1))*(tover)
           Delay+=abs(qqa.getValue())
        SWDelay=(len(AA)-1)*(tsw)
        aa=SWDelay+TotDelay+Delay
        nn=aa*1e6

        temp={(s,d):nn}
        DatosDelay.update(temp)


b=z.RunTime*1000##### Time To Solve Joint Optimization
arreglo=(float(maximumA)/capacityIn)*100
slots=len(ta)
```

A.3 Shortest Path Algorithm

```
D = nx.MultiDiGraph(day="Slots")
```

```
D.add_nodes_from(nodes)
D.add_edges_from(arcs)

def SHPH(arcs,demand,capacity,pThresold):

    for (i,j) in arcs:
        D.add_edge(i,j,weight=capacity[i,j])
    caminocorto={}
    pVect={}#####Define Prioritization Vector

    for s,d in sdpair:
        vect=[]
        if d in DTC2:
            temp1={(s,d):[2]}###High Priority Flows
            pVect.update(temp1)
        elif d not in DTC2:
            temp2={(s,d):[1]}
            pVect.update(temp2)###Low Priority Flows

    CaminoSol={}
    slotsAsign={}
    pVect2=pVect

    while len(pVect2.keys())!=0:#Iterate until all Traffic Flows###
    #####have assigned links

        ###Choose Flow with Highest Priority Value###
        inverse = [(value,key) for key, value in pVect2.items()]
        yy=max(inverse)[1]
        demandA=demand[yy[0],yy[1],tc]
        ###Choose Flow with Highest Priority Value###

        ###Calculate Shortest Path for high priority flow###
        for (i,j) in arcs:
           D.add_edge(i,j,weight=costoarc[i,j])
        w=nx.dshortest_path(D,yy[0],yy[1])
        ###Calculate Shortest Path######

        ####Update Scheduled Links and Link Capacity######
        arcoSol=[]
        for y in range(1,len(w)):
           e=w[y-1]
           l=w[y]
```

```
           costoActual=costoarc.get((e,l))
           temp={(e,l):demandA+costoActual}
           costoarc.update(temp)
           arcoSol.append((e,l))
         ####Update Scheduled Links and Link Capacity######

        arcoTemporal={(yy[0],yy[1]):arcoSol}
        CaminoSol.update(arcoTemporal)
        xx=(yy[0],yy[1])
        pVect2=removekey(pVect2,xx)###Update Priority Vector


    while len(pVect.keys())!=0:##Iterate Until all flows
    #have timeslots assigned

        ######Choose Flow with Highest Priority Value###
        inverse = [(value,key) for key, value in pVect.items()]
        yy=max(inverse)[1]
        ee=CaminoSol.get((yy[0],yy[1]))
        ###Choose Flow with Highest Priority Value###


        tIni=1
        for y in range(1,len(ee)+1):
            e=ee[y-1][0]
            l=ee[y-1][1]
            arcodemand=costoarc.get((e,l))
    ##Define Number of Slots per Link with Aggregation if Possible###
            if arcodemand<=(float(capacityIn)/5):##Define Number of Slots
                demandPair=1                     #Necessary per Link
            else:
             demandPair=int(math.ceil(float(arcodemand)/(capacityIn/5)))

            te=[t for t in range(tIni,tIni+demandPair+1) ]
            temp={(e,l):te}

            slotsAsign.update(temp)
            tIni=max(te)+1
        xx=(yy[0],yy[1])
        pVect=removekey(pVect,xx)
    ###Define Maximum Link Utilization#####
    labeldemandAltern={}
    for i,j in slotsAsign.keys():
```

```
        e=0
        for s,d in sdpair:
             ty=CaminoSol.get((s,d))
             if (i,j) in ty:
                  e+=demand[s,d,tc]
             temp={(i,j):e}
             labeldemandAltern.update(temp)

    maxLink=list(labeldemandAltern.values())
    maximumAltern=max(maxLink)/capacityIn

    ###Define Number of Slots Used#####
    maxNumSlot=0
    for (i,j) in slotsAsign.keys():
        maxim=slotsAsign.get((i,j))
        e=max(maxim)
        if maxNumSlot<=e:
            maxNumSlot=max(maxim)
        else:
            maxNumSlot=maxNumSlot
```

Bibliography

[1] M. Peng, Y. Li, Z. Zhao and C. Wang *System architecture and key technologies for 5G heterogeneous cloud radio access networks*, in IEEE Network, vol. 29, no. 2, pp. 6-14, March-April 2015.

[2] Niu, Yong Li, Yong Jin, Depeng Su,Li Vasilakos, Athanasios V *A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges*, Wireless Networks vol. 21, no. 8, pp. 2657–2676, 2015.

[3] IEEE 802.11 working group *IEEE 802.11ad, Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band*, December 2012.

[4] Mary-Ann Russon *5G: The most exciting innovations that could transform the future of mobile internet.* International Business Times. February 2016. Article available online at: $http : //www.ibtimes.co.uk/5g - most - exciting - innovations - that - could - transform - future - mobile - internet - 1546301$

[5] T. Nitsche, C. Cordeiro, Adriana F., Edward W. Knightly, E. Perahia and J. C. Widme *IEEE 802.11ad: directional 60 GHz communication for multi Gigabit-per-second Wi-Fi* in IEEE Communications Magazine, vol. 52, no. 12, pp. 132-141, December 2014.

[6] J. Wu, Z. Zhang, Y. Hong and Y. Wen *Cloud radio access network (C-RAN): a primer*, in IEEE Network, vol. 29, no. 1, pp. 35-41, Jan.-Feb. 2015.

[7] ss *Cloud RAN for Mobile Networks - a Technology Overview* IEEE Communications Surveys and Tutorials, 17(1), p 405-426.

[8] J. Bartelt, P. Rost, D. Wubben, J. Lessmann, B. Melis and G. Fettweis *Fronthaul and backhaul requirements of flexibly centralized radio access networks* in IEEE Wireless Communications, vol. 22, no. 5, pp. 105-111, October 2015.

[9] P. Ngatchou A. Zarei and M. A. El-Sharkawi *Pareto multi objective optimization* Proc. 13th Int. Conf. Intell. Syst. App. Power Syst pp. 84

[10] Huawei H., Peng Li, Song Guo, and Weihua Z , *Software-Defined Wireless Mesh Networks: Architecture and Traffic Orchestration* , IEEE Network, vol. 29, no. 4, pp. 24-30, July-August 2015.

[11] Jian Q., Lin X. Cai‡ , Xuemin S. and Jon W. Mark†, *STDMA-based Scheduling Algorithm for Concurrent Transmissions in Directional Millimeter-Wave Networks*, Communications (ICC), 2012 IEEE International Conference on, Ottawa, ON, 2012, pp. 5221-5225.

[12] Hong Li, Lei Huang and Yi Wang *Scheduling Schemes for Interference Suppression in Millimeter-Wave Cellular Network*, 2015 IEEE 26th International Symposium on Personal, Indoor and Mobile Radio Communications, pg. 2244-2248, 2015.

[13] H. Shokri-Ghadikolaei, L. Gkatzikis and C. Fischione, *Beam-searching and transmission scheduling in millimeter wave communications*, Communications (ICC), 2015 IEEE International Conference on, London, 2015, pp. 1292-1297.

[14] E. Pateromichelakis, M. Shariat, A. ul Quddus and R. Tafazolli, *Joint routing and scheduling in dense small cell networks using 60 GHz backhaul*, IEEE Communication Workshop (ICCW), 2015 IEEE International Conference on, London, 2015, pp. 2732-2737.

[15] Jyoti Kolap, Shoba Krishnan, Ninad Shaha, *Frame Aggregation Mechanism For High Throughput 802.11 WLANs*

[16] Sun, S., Rappaport, T. S., Rangan, S *Propagation Path Loss Models for 5G Urban Micro- and Macro-Cellular Scenarios*, 2016 IEEE 83rd Vehicular Technology Conference (VTC2016-Spring), May 2016

[17] Wolsey, L. A. *Mixed Integer Programming. Wiley Encyclopedia of Computer Science and Engineering* 15 September 2008. 1–10.

[18] Jesus Gutierrez, Nebojsa Maletic, Daniel Camps-Mur, Eduard Garcia, Ignacio Berberana, Markos Anastasopoulos, Anna Tzanakaki, Vaia Kalokidou, Paris Flegkas, Dimitris Syrivelis ,Thanasis Korakis, Peter Legg, Dusan Markovic, George Limperopoulos, Jens Bartelt, Jay Kant Chaudhary, Michael Grieger , Nikola Vucic, Jim Zou and Eckhard Grass *5G-XHaul: A Converged Optical and Wireless Solution for 5G Transport Networks* TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGIES, 2016.

[19] IEEE P802.11 - Task Group ay *IEEE 802.11ay* available online at $http://www.ieee802.org/11/Reports/tgay_update.html$

[20] J. Bartelt1, A. Author2 and G. Fettweis *The Impact of 5G Radio Access Technologies on a Converged Fronthaul and Back-haul Network*, Technische Universität Dresden, Germany, 2016

[21] China Mobile *White Paper of Next Generation Fronthaul Interface*, White paper,Version 1,October 2014.

[22] NGMN Alliance *Guidelines for LTE Backhaul Traffic Estimation*, White paper, July 2011.