Distributed resource allocation algorithms for wireless networks

By

Swaroop Gopalam

Master of Research, Macquarie University, 2016.

Bachelor of Technology, IIT Bombay, 2014.

A thesis submitted to Macquarie University

for the degree of Doctor of Philosophy

School of Engineering

10 Dec 2021

MACQUARIE
University
SYDNEY·AUSTRALIA

Typeset in LaTeX 2_ε.

The work presented in this thesis was carried out at the School of Engineering, Macquarie University, Sydney, Australia, between March 2017 and December 2020. This work was principally supervised by Prof. Stephen Hanly and co-supervised by Prof. Philip Whiting.

Except where acknowledged in a customary manner, the material presented in this thesis is, to the best of my knowledge, original and has not been submitted in whole or part for a degree in any university.

Swaroop Gopalam

Acknowledgements

This thesis is the culmination of three years of my effort. This thesis would not have been possible without the support from my supervisors, Macquarie university, friends and family. I am immensely grateful for the help I have received through this period of my life. As the author, it is my duty to acknowledge these contributions.

First of all, I would like to thank Prof. Stephen Hanly for being an excellent supervisor. He has been the reliable guiding hand, always present to mentor me through this journey. He is a great teacher, and his methods have been instrumental in shaping my approach to research. He has an innate focus when it comes to writing in a manner engaging to the reader, and a gift for conveying complicated ideas in an intuitive manner. His guidance has greatly improved my writing skills, albeit I am still learning. As a matter of fact, the thesis in this shape or form would not have been possible without the careful revisions from Prof. Stephen Hanly. It has been a great pleasure and honour to work with him.

I would like to thank my co-supervisor Prof. Philip Whiting for his guidance over the past four years. Whenever I needed help with a new topic, he was always there to guide me towards the right reference material. He has provided me with excellent book and papers, which broadened my knowledge on several topics in this thesis. He has also greatly improved my mathematical skills. He has taught me to be careful with mathematical arguments; To approach any mathematical argument that I make with skepticism, which will expose any gaps that exist in the argument.

I would like to thank Macquarie university, and acknowledge all the support I received. The work in thesis was supported by a iMQRTP PhD scholarship from Macquarie University, and by a COVID 19 extension scholarship scheme from Macquarie University. It was also supported in part by the Australian Research Council under grant DP180103550.

Lastly, I would like to thank my mom, Lakshmi Kala Gopalam and my dad, Ramaswamy Gopalam for their unconditional love and unending support. I would like to thank my grand dad, Sambasiva

Rao Gopalam, who has always encouraged me to excel at life, and supported my interest in science during my childhood. I would like to thank Khushboo Singh for being a great friend and housemate. I would also like to thank Chunshan Liu for his help in introducing me to Heterogeneous Networks.

Abstract

In this thesis, we present distributed resource allocation algorithms for various wireless networks, which include Heterogenous Networks, mmWave Integrated Access and Backhaul networks. We consider the minimum time clearing objective for resource allocation in wireless networks. The minimum time clearing optimization corresponds to clearing a given set of files at the wireless nodes, in the minimum possible time subject to the scheduling constraints. The optimization is NP-complete in general. In this thesis we consider wireless network models with additional structure arising from the above applications. Based on this structure, we propose distributed resource allocation algorithms which only require local information and which do not suffer from a combinatorial explosion in complexity as the size of the network grows large.

We characterize the stability of the considered wireless networks under dynamic scenarios such as random traffic arrivals in time, and time varying channel conditions. Roughly speaking, we refer to the network as stable if the traffic does not build up indefinitely at the nodes in the network. We provide distributed scheduling and flow control algorithms for the wireless networks under dynamic scenarios. We show that a version of the minimum time clearing optimization can be formulated using the queue length information. The solution can then be used as a scheduling algorithm. It follows that the proposed scheduling algorithm can be implemented in a distributed and efficient manner, in topologies where the underlying structure allows for an efficient solution.

With the exception of Chapter 3, the scheduling algorithms proposed in the thesis are throughput optimal, i.e., stabilize the network for all arrival rates within the stability region. The greedy scheduling algorithm proposed in Chapter 3, only uses local information. We show that this proposed algorithm achieves the largest stability region among the class of scheduling policies which only use local information. We also provide conditions under which the greedy scheduling algorithm is throughput optimal.

Contents

Chapter 1

Introduction

1.1 Introduction

Resource allocation is a key component in the operation and control of wireless networks. It has been a topic of significant research interest over the years and was studied for various networks under different contexts. Broadly speaking, the purpose of resource allocation is to match the available resources (e.g., time slots, frequency channels) to the traffic demands (e.g., users, flows) of the network, so as to achieve desirable outcomes, such as fairness, maximizing the total/sum rate, or stability.

A general wireless network consists of several wireless nodes with each transmitter-receiver pair forming a link. There are constraints on which links can be activated simultaneously, such as half-duplex constraints, constraints due to interference. Many of the resource allocation problems that occur in general wireless networks are NP-hard. For example, finding the maximum weighted schedule is an NP-hard problem in general wireless networks [1]. However, there are several wireless networks where an underlying structure allows for efficient solutions to resource allocation problems. This thesis focuses on such wireless networks, and provides efficient resource allocation policies by exploiting the underlying structure.

The minimum clearing resource problem can be stated as the minimum amount of resource (such as time) required to clear a set of files (traffic) backlogged at the nodes in the network. This is also an NP-hard problem in general wireless networks [2]. The *minimum clearing resource problem* is a central topic of this thesis. The minimum clearing time optimization is closely related to the capacity of the network. With the exception of chapter 3, it features in every chapter. In this thesis, we present several networks and topologies where the minimum clearing time can be solved in polynomial time.

We also derive efficient resource allocation policies based on the solution. The presented policies are distributed in nature and hence practical in terms of implementation.

For the wireless networks in the thesis, we also consider stability under dynamic scenarios such as occur with random traffic arrivals in time, or time varying channel conditions. We refer to the network as stable if the traffic does not build up indefinitely at the nodes in the network. For several networks considered in this thesis, we will show that the value of the minimum clearing time problem (formulated using arrival rates) determines whether stability is possible, under any scheduling policy. In dynamic scenarios, we provide distributed scheduling policies which can be implemented using local information and message passing. With the exception of Chapter 3, the provided scheduling policies are throughput optimal, i.e., they stabilize the network for any arrival rate vector interior to the stability region. The proposed algorithm in Chapter 3 is a greedy scheduling algorithm which only uses local information. We show that the proposed algorithm has the largest stability region among the class of policies which use only local information. We also provide the conditions under which the proposed algorithm achieves 100% of the stability region (including of global policies).

In the following, we provide an overview of the wireless networks which will be considered in this thesis.

1.1.1 Heterogeneous Networks (HetNet)

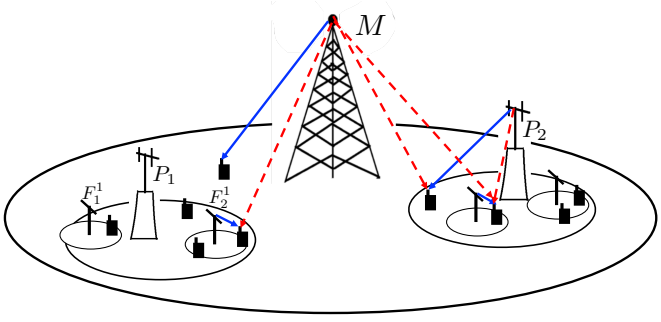


Figure 1.1: A three tier HetNet with macro, pico and femto tiers. The solid blue lines represent the wireless down-links and the dotted red lines represents the cross-tier interference.

The concept of HetNets originated from introduction of small cells such as picos and femtos to operate in the same region as the traditional macro cellular infrastructure [3]. HetNets provide several advantages. The small cells can be strategically deployed to traffic hotspots. They increase coverage

in blind spots of macro infrastructure, e.g., improve indoor coverage etc., More importantly, small cells offer increased network capacity due to increased spatial re-use of spectrum. Hence, HetNets and small cells have emerged as a potential solution to meet the increasing demand for wireless data. The trend of shrinking cell sizes is expected to continue in the future, e.g., ultra dense deployments of millimetre wave (mmWave) small cells. As cell sizes get smaller, there will be an increased number of tiers in the future wireless architectures. Future 5G networks are also expected to have wireless access simultaneously available from multiple access technologies such as satellite communication (SATCOM), aerial base stations (such as High Altitude Platforms (HAPs)), LTE, mmWave etc., Hence, HetNets are a key part of future wireless architectures.

In a HetNet environment, a user equipment (UE) can be within the range of multiple BSs of different tiers. The UE has to choose between several different BSs (of possibly different technologies) for association. Hence, cell association becomes a critical challenge in HetNets. The usual association scheme in cellular networks of choosing the BS with highest downlink signal to interference and noise ration (SINR) value has been shown to be sub-optimal for HetNets [4, 5]. In a LTE HetNet, macro BS has much higher transmit power compared to the small BSs. Hence, the max-SINR association leads to overloading of the macro cell, and under utilization of small cells. Several works have has shown the benefits of biased user association in HetNets. Biased SINR association via Cell Range Expansion scheme (CRE) was standardized as part of enhanced inter cell interference co-ordination (eICIC) in 3GPP to address this problem [6]. In Chapter 2, we consider the optimization of cell association in a three tier HetNet. We present novel algorithms that are more efficient than existing algorithms for three tier HetNets.

Another important challenge in HetNet resource allocation is interference management. Consider a LTE HetNet; the high transmit power of the macro can cause interference to the transmissions in the small cells. This interference is referred to as cross-tier interference and, unmitigated, it adversely affects the performance of small cells. Resource partitioning in the time domain via the Almost Blanking Subframes (ABS) scheme was introduced as part of enhanced inter cell interference co-ordination (eICIC) in 3GPP to address this [6]. Under ABS scheme, the macro is silent on the ABS subframes, which allows the small cells to transmit without the macro interference. In Chapter 2, we introduce a novel framework for joint-optimization of cell association and resource allocation (for interference avoidance) in a three tier HetNet. We derive new structural results for joint cell biasing and resource allocation, and propose novel algorithms based on the results. We provide example

networks to show that the framework can be applied in a wide variety of scenarios, such as three tier HetNets with mmWave femto cells, and three tier HetNets with HAPs.

In Chapter 4, we generalize the resource allocation framework to a K tier HetNet which can be applied for management of both cross-tier interference and co-tier interference (i.e., between the cells in the same tier). We present a novel resource allocation framework which is linearly scalable in the number of tiers. We also derive algorithms which are of linear complexity in the size of network, by exploiting the structure of the K tier HetNet model.

1.1.2 Millimetre wave Integrated Access and Backhaul (mmWave IAB)

mmWave cellular networks are expected to be a central part of the Next generation wireless communications (5G) [7]. mmWave technology is capable of delivering very high rates, due to the vast amount of spectrum available in the mmWave band. However, wireless communication at mmWave frequencies comes with two major challenges, namely 1) high isotropic propagation loss, and 2) sensitivity to blockage by the objects in the environment. To overcome the high propagation losses, directional communication using beam-forming is being considered for mmWave cellular. High beam-forming gains are achievable by implementing antenna arrays in a tiny area (large numbers of antennas are possible due to the small wavelengths). The mmWave cell sizes are expected to be small due to the high propagation loss and blocking, and ultra dense deployments of Next Generation Node Bases (gNBs) are being considered to provide universal coverage.

It is prohibitively expensive to provide fibre backhaul support to all the mmWave gNBS under dense deployments. Hence, there has been recent interest in multi-hop relaying (or self backhauling) in mmWave cellular networks as a potential solution. Notably, 3GPP has completed a recent study item on the potential solutions for efficient operation of integrated access and wireless backhaul (IAB) for New Radio (NR), as part of standardization [8]. The study emphasizes the joint consideration of radio-access and backhaul for mmWave cellular networks.

In [8], a multi-hop IAB network consists of two types of gNBs. A fraction of gNBs are deployed with dedicated fiber backhaul links, referred to as IAB donors [8]. The other gNBs (referred to as IAB nodes) relay their backhaul data over wireless mmWave links, possibly in multiple hops to an IAB donor. Dynamic resource allocation (or scheduling) is a key challenge in the control of multi-hop IAB networks [6, 9]. An IAB node establishes a link to a parent node (either another IAB node or a

donor). The central unit (CU) at the IAB donor establishes a forwarding route to the IAB node (via the parent). Therefore, traffic of a UE is forwarded along this established route from the IAB donor to the UE (on downlink).

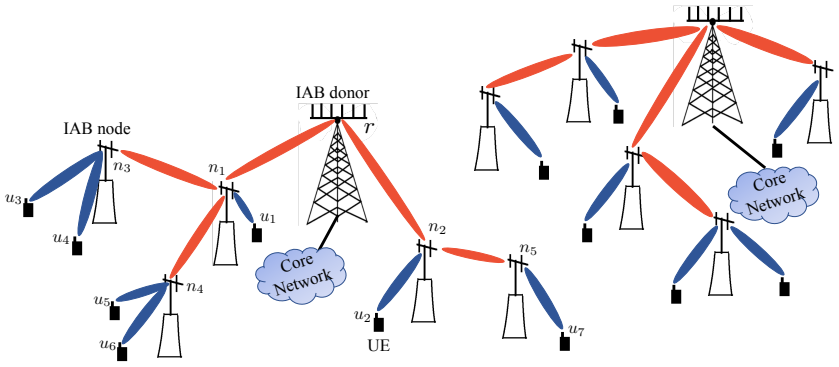


Figure 1.2: A mmWave Integrated Access and Backhaul Network.

The 3GPP study identified two topologies for the operation of mmWave IAB, 1) spanning tree (ST) and 2) directed acyclic graph (DAG) topology [8]. We primarily focus on the ST topology, where each IAB node has one parent node (either a IAB node or the IAB donor). e.g., see Figure 1.2.

Dynamic resource allocation (or scheduling) is a key challenge in the control of multi-hop IAB networks [6, 9]. Joint consideration of access and backhaul in resource allocation for IAB networks is emphasized in [8]. According to [8], it is critical to consider in-band backhauling (i.e., backhaul and access use the same frequencies) solutions that accommodate tighter interworking access and backhaul. In an in-band scenario, the half-duplex constraint imposes restrictions on the links that can be active simultaneously.

In chapter 3, we consider scheduling algortithms for the 3GPP mmWave IAB network model in an in-band IAB scenario. Under the model, the gNBs are allowed to have multiple RF chains. We consider a dynamic model where packets arrive as an exogenous process at the IAB donor node. We also consider time varying link rates to capture the short term variations (fading) in the mmWave channels. We consider the IAB network to be stable under a scheduling policy if the queue lengths do not grow indefinitely. We characterize the stability region of the IAB network, as set of the arrival rate vectors for which stability is possible under any scheduling policy.

We investigate a class of distributed scheduling policies which only require local information. We also characterize the stability region for this class of local policies. We show that stability region of the local policies is the same as that of the whole stability region, when the link rates are not varying

(i.e., constant) over time. We propose an optimal distributed local scheduling policy (from this class) for the IAB network which achieves the stability region of the class of local policies. Using numerical simulations, we show that the performance of the proposed local algorithm is very close to that of global policies.

1.1.3 Wireless Networks under conflict constraints

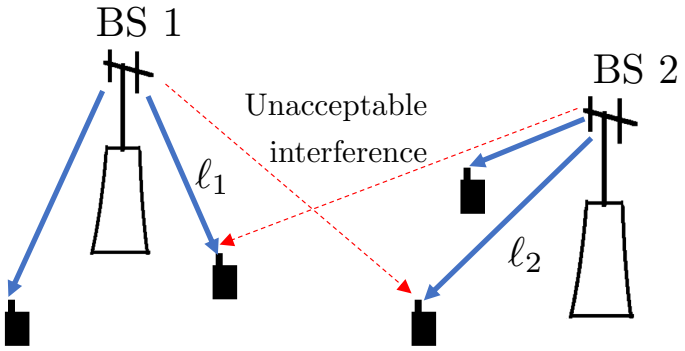


Figure 1.3: Example Network. The solid blue lines are wireless links and the red dotted lines represent the interference (or contention).

Consider a wireless network as a graph $G = (V, E)$, where V is the set of wireless nodes and E is the set of wireless links. In a wireless network, there can be constraints on simultaneous link activation for two links, say $\ell_1, \ell_2 \in E$. For example, 1) Suppose that receiver of link ℓ_1 is the transmitter of ℓ_2, then ℓ_1 and ℓ_2 cannot be activated simultaneously due to the half-duplex constraint. 2) Consider a Carrier Sense Multiple Access (CSMA) type wireless network [10, 11]. Suppose that ℓ_1 causes too much interference at the receiver of ℓ_2. Simultaneous activation of ℓ_1 and ℓ_2 together leads to collisions, and hence, not allowed. An example is provided in Figure. 1.3. We refer to the constraints of this kind, where simultaneous activation of two links are not allowed, as the *conflict constraints*. Conflict constraints have been widely used to model wireless networks in the wireless scheduling and queueing literature [10–19].

In Chapter 5, we propose a novel distributed greedy resource allocation scheme for a wireless network under conflict constraints. The algorithm only requires local information to make decisions. The greedy algorithm always produces feasible solutions to the minimum time clearing problem. We show that the algorithm has a monotonicity property that the objective value under the feasible solution

at time $t+1$ is less than or equal to the value at time t. We consider two topologies and show that the algorithm converges to the optimal solution under these topologies, due to the underlying structure of the topology.

In Chapter 6, we consider a wireless network under conflict constraints, in a dynamic scenario, with exogenous flow arrivals as a stochastic process. We consider flow control and resource allocation for this network, and characterize the stability region. We say the network is stable if the flow backlog in the network does not build up indefinitely. We propose a novel joint flow control and resource allocation policy, based on a minimum resource clearing optimization.

1.2 Overview of the thesis

1.2.1 Chapter 2

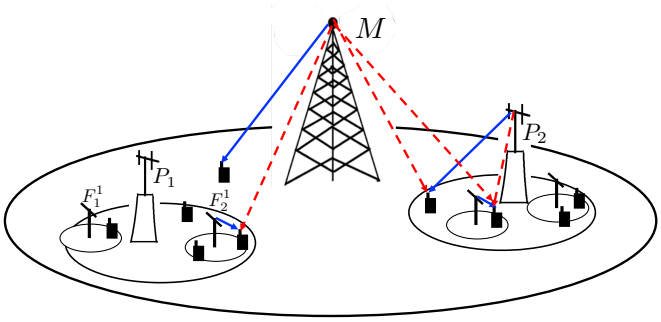
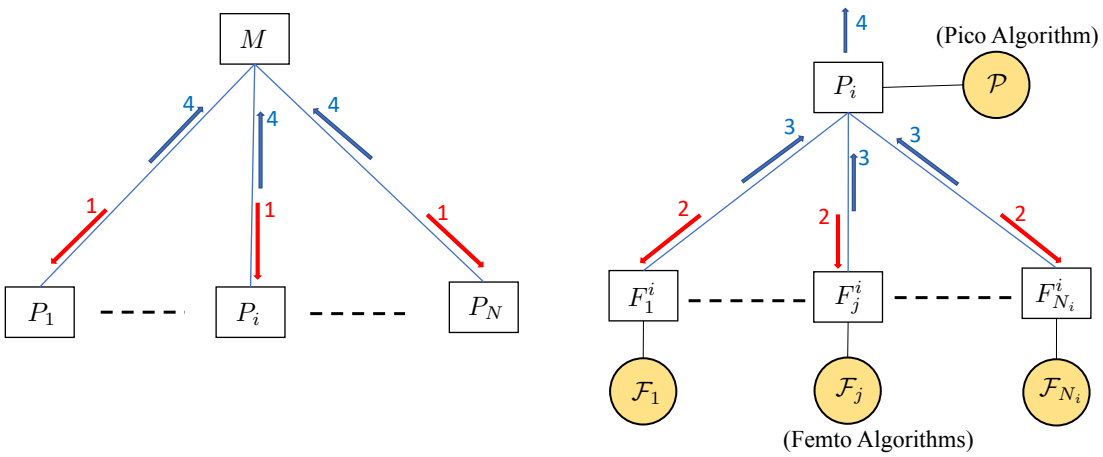


Figure 1.4: A three tier Heterogeneous Network with macro, pico and femto tiers. The solid blue lines represent the wireless down-links and the dotted red lines represents the cross-tier interference.

We present a novel distributed framework for optimization of resource allocation and cell association in three tier HetNets. We consider the problem of jointly optimizing user association and resource allocation in a three tier downlink HetNet. We refer to the tiers as macro, pico and femto tiers. We formulate the minimum time clearing problem as a linear program (LP). We show that by fixing the time allocated to small cells, the LP can be decomposed into several (equal to number of pico BSs) smaller independent LPs. It follows that significant parallelization can be achieved by solving these LPs simultaneously at the corresponding pico BSs. We then show that the user association is determined by a set of rate-bias multipliers, one multiplier per BS. The problem of finding the multipliers using conventional approaches leads to a high dimensional search, e.g.,[4, 20].

In contrast, we present new structural results which enable us to propose more efficient algorithms with reduced complexity. We show that each rate-bias value (corresponding to a BS) crucially only takes values from a discrete set. For a pico P_i, the size is less than $0.5|U_i|^2$, where $|U_i|$ is the number of UEs covered by the pico. The search space of multipliers is reduced to a small finite set which we characterize. We further show that the solution of the LP at each pico BS is determined by just two parameters: the femto time allocation and the pico rate-bias. We propose novel distributed resource allocation and cell association algorithms based on the structural results. Figure. 1.5 provides an illustration of message passing under the proposed framework.



(a) Macro level distributed scheme

(b) Pico level distributed scheme

Figure 1.5: Distributed computation of clearing time using message passing algorithms. The circles represent the allocation functions at the BSs, and the arrows represent the message exchanges. The numbering on the arrows is the order in which the message exchanges occur.

We apply the framework to a wide variety of three tier HetNet examples, which include three tier HetNet with mmWave femto cells, and three tier HetNet with HAPs. We also consider a dynamic three tier HetNet model, with stochastic flow arrivals. We show that the minimum time clearing optimization provides a stability characterization for the dynamic model. Hence, the three tier framework can be used for capacity planning under the dynamic model, or can be applied in real-time for optimal control of HetNets.

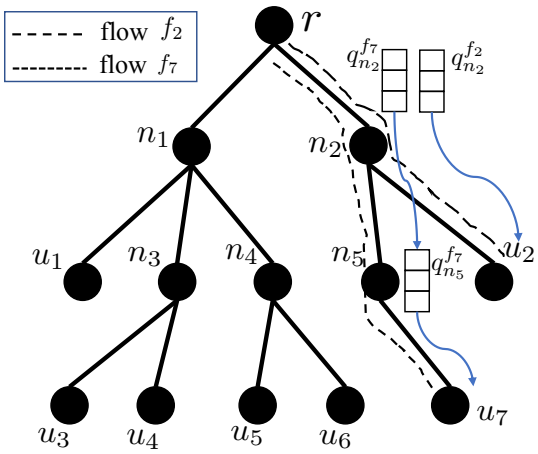


Figure 1.6: Graph representation of IAB network.

1.2.2 Chapter 3

We consider a mmWave IAB network under the tree topology in an in-band IAB scenario. We model it as a rooted tree. For an example, see Figure. 1.6. In Figure. 1.6, the root node r is IAB donor. In Figure. 1.6, n_1 provides wireless backhaul to IAB nodes n_3, n_4, and wireless access to UE u_1. In our model, we allow multiple RF chains.

We consider a dynamic scenario with exogenous packet arrivals and varying link rates. We consider the system to be stable if the queue lengths do not blow up to infinity. We characterize the stability region for the IAB network, as the set of arrival rate vectors for which stability is possible under some scheduling policy.

We present a novel distributed and local scheduling policy for the mmWave IAB network. The policy is a hierarchical scheduling algorithm, where a IAB node makes its scheduling decision (i.e., for its downlinks) based on the decision of its parent, as follows. If the backhaul link to the IAB node n is scheduled by its parent, no downlinks are scheduled at n. Otherwise, the IAB node n chooses its downlink schedule based on a local max-weight based rule which only requires queue information at n. We show that when the link states are reliable, i.e, unvarying, the proposed policy achieves the entire stability region (i.e., including that of global policies) for the mmWave IAB network.

We provide numerical simulation results for a IAB network in a realistic scenario with time-varying link states. We show that the proposed local policy performs very closely to the global max-weight and back-pressure policies (in terms of stability) in the considered IAB scenario, where the global policies have full access to all the queue lengths and link states information to make the scheduling

decision.

1.2.3 Chapter 4

We generalize the three tier resource allocation framework from Chapter 2 to K tiers.

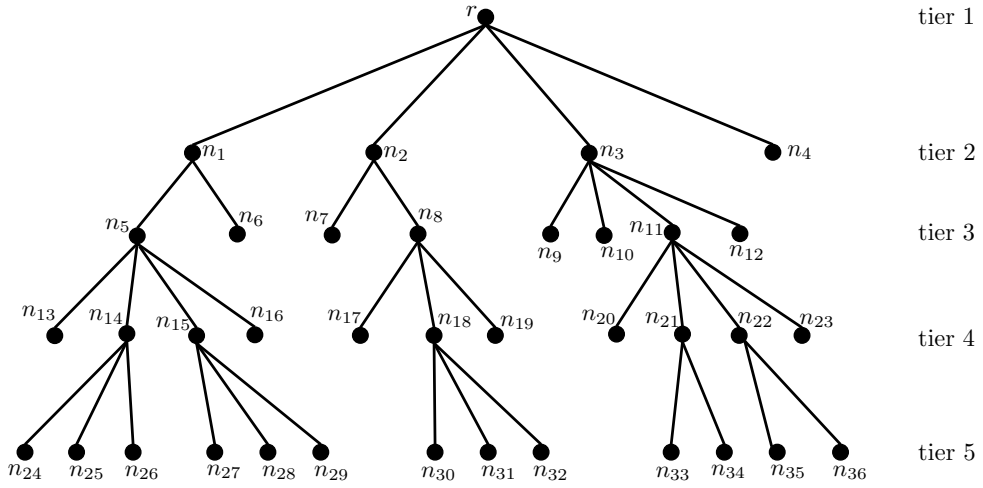


Figure 1.7: K tier HetNet.

We consider a K-tier HetNet, with a tier-1 BS r covering a wide area. There are several smaller BSs of different tiers operating in the coverage region of the tier 1 BS r. The BSs can be divided into tiers based on their coverage area. In general, the lower tier cells have BSs at higher altitudes, which have larger coverage areas. Several smaller cells (of higher tier) can operate in the coverage area of a lower tier cell. In Figure. 1.7, all the other BSs are operating in the coverage area of r, and n_1, n_2, n_3, n_4 are the tier 2 BSs under r. Similarly, n_7, n_8 are the tier 3 BSs operating under n_2.

Under the K tier HetNet model, a lower tier BS causes debilitating interference to the smaller cells (of higher tier) in its coverage area, if using same resources. Thus, a transmission from BS n of tier i causes interference to the transmissions in a tier j cell in the coverage area of n, where $j > i$, e.g., n_2 and n_{18} cause debilitating cross-tier interference to n_{30}, if scheduled on same resource as n_{30}. The other type of interference is the co-tier interference, which is the interference caused by transmissions of BSs in the same tier. Interference is avoided by resource partitioning, i.e., allocating separate resource to the interfering BSs. The K tier resource allocation framework can be applied for management of both cross-tier interference and co-tier interference (i.e., between the cells in same tier).

We introduce a novel graphical model for the K tier HetNet. For an illustration, see Figure. 1.8. In Figure. 1.8, node r is the tier 1 BS, which interferes with the rest of the BSs in the network. This is signified by the edge joining r to H_r. Here, H_r is analogous to the sub-network which is operating in the coverage area of r. It can be noted that there is a graph inside H_r, which models the interference constraints at tier 2. Here nodes n_1, n_2, n_3, n_4 are the tier 2 BSs. For co-tier interference, n_i is joined by an edge to n_j if there is co-tier interference between n_i and n_j. As with H_r, H_{n_i} is analogous to the sub-network which is operating in the coverage area of n_i. The edges connecting n_i to H_{n_j} represent the cross-tier interference caused to the lower tier BSs in the sub-network H_{n_j}.

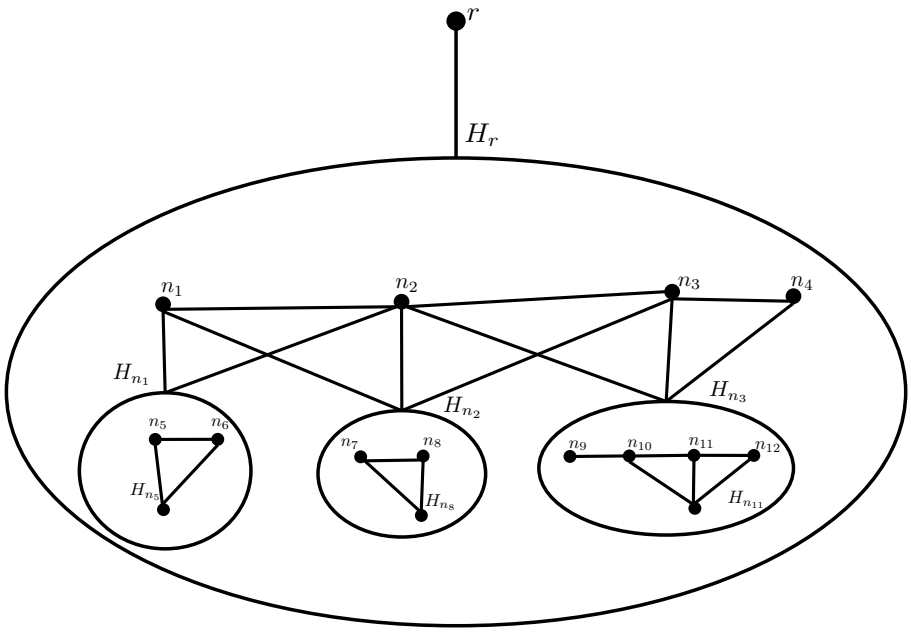


Figure 1.8: Example Graph.

We consider the minimum resource clearing problem for the K tier HetNet model, as a linear programming (LP) formulation. We propose a novel distributed method to solve the LP, which involves solving smaller linear programs at each tier (e.g., the formulation at r only involves $\{n_i\}_{i=1}^4$). The smaller LPs can be specified using only local information. We show that the overall LP can be solved by recursively solving the smaller LPs at each tier. Thus, the complexity of the proposed solution is the sum of complexities of the smaller LPs at each tier.

Based on the solution, we propose a novel distributed framework for K-tier resource allocation. The framework only requires knowledge of the interference relations at the tier level (or in the co-tier graph), e.g., r only needs to know the interference relations between $\{n_i\}_{i=1}^4$. We show that the

proposed resource allocation framework is scalable with the increase in tiers, i.e., complexity under the framework increases only linearly with the increase in number of tiers. When there is additional structure in the co-tier graph, we provide algorithms which are of linear complexity in the size of the co-tier graph. In such networks, the proposed framework only has a linear complexity in the size of the network.

1.2.4 Chapter 5

In Chapters 2-4, we present distributed resource allocation algorithms for various networks. The algorithms rely on the presence of a central node for implementation, (e.g., macro M in Chapter 2, and root node r in Chapter 3 and Chapter 4). In this chapter, we consider the minimum time clearing problem as a linear program for a wireless network under conflict constraints. We propose a novel distributed greedy resource allocation scheme for this network, which is more distributed than the algorithms in the preceding chapters. The algorithm is a *book ahead* slot reservation system, which only requires local information to make decisions. The scheme can be considered as a network-wide round robin scheduling algorithm.

We consider slot allocation for users $\{u_k\}_{k=1}^{N}$, where a user u_i cannot be scheduled in the same slot as any $u_j \in I(u_i)$. For the example in Figure. 1.9, for each link u_i, let $I(u_i)$ be set of all u_j such that u_i, u_j share a common node. Under the algorithm, u_i only needs to exchange information with its neighbors, i.e., users in $I(u_i)$.

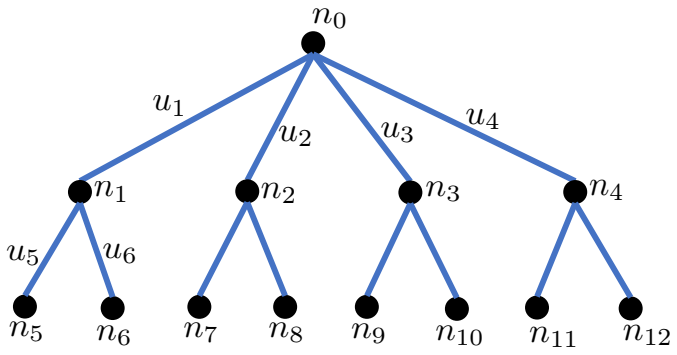


Figure 1.9: Example network under conflict constraints.

We show that the proposed greedy algorithm generates feasible solutions to the minimum time clearing linear program at each step. We show that the algorithm has a monotonic behaviour, that the

objective value under the feasible solution at time $t+1$ is less than or equal to the value at time t. In general, the greedy algorithm may only produce sub-optimal solutions of the minimum clearing time problem. We consider two topologies (with a special structure) and show that the algorithm always converges to the optimal solution under these topologies in finite time.

The two topologies are based on tree graphs. Topology 1 can be used to model wireless broadcast networks. Topology 2 can be used to model a wireless network with relays, e.g., IAB network, where IAB nodes have a single RF chain. Figure. 1.9 is an example of Topology 2. It can be used to represent the following relay network. Let n_0 be a BS with wired backhaul connection, and nodes $\{n_5, \ldots, n_{12}\}$ represent the mobile user equipments (UEs). The nodes $\{n_1, \ldots, n_4\}$ are relay BSs which forward the data from n_0 to the UEs. The links (which we call users) in Figure 5.3 correspond to the wireless links that occur in this network.

1.2.5 Chapter 6

We consider a wireless network under conflict constraints in a dynamic scenario. We consider a scenario with exogenous flow arrivals as a stochastic process. We consider the problem of flow control and resource allocation for this network. We consider the system to be stable under a policy if the backlogged flows (in the network) do not build up indefinitely. We characterize the stability region for the setup, as the set of arrival rate vectors for which stability is possible under any algorithm. We propose a novel joint flow control and resource allocation policy, based on a minimum resource clearing optimization. The proposed stationary policy only requires current flow backlog information to solve the optimization. The resource allocation is reconfigured only when the state changes, i.e., a when a flow arrival or a departure occurs in the network. We show that the proposed policy stabilizes the network for any arrival rate vector within the stability region. As an example, we present a detailed application of the proposed algorithm in a dynamic K tier HetNet model.

We make use of *Fluid limit* theory [21, 22] to establish the stability of the proposed policy. We derive the fluid limit model corresponding to the system under the proposed policy. We show that the fluid limit is stable (i.e., drains to zero state in a fixed time) for any arrival rate vector within the stability region. Once the stability of fluid limit is established, we follow the theory in [22] to establish the stability of the system under the proposed policy.

Chapter 2

Optimal User Association and Resource Allocation for Three tier HetNets

2.1 Introduction

Heterogeneous Networks (HetNet) consist of low power base stations (BS) such as pico cells and femto cells deployed to operate in the same region as traditional macro cellular infrastructure [7]. These small cells increase the capacity due to better spatial re-use of spectrum. Future 5G networks are expected to be even more heterogeneous with wireless access simultaneously available via multiple technologies, including new technologies such as mmWave and aerial BSs. As demand increases and cells get smaller, there will be an increased number of tiers in future HetNet architectures.

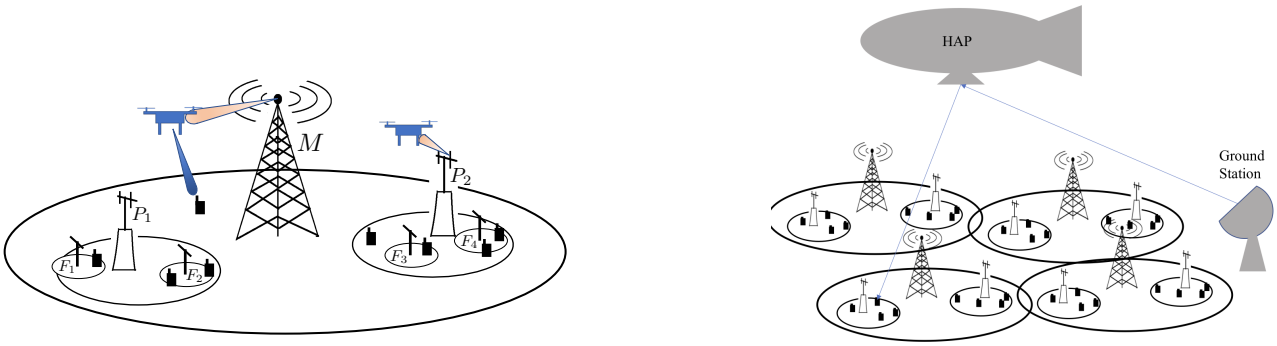
Stochastic geometry based approaches traditionally employed for studying HetNets provide analytical results on coverage and SINR distributions, but are not suitable for real-time control. The optimization based literature has focused on cell association and resource allocation for two tier HetNets, and to date, there is no complete analytical solution for HetNets with more than two tiers. With the increased complexity of 5G networks, there is a need for studying the problem in general cases with more than two tiers. This chapter provides a complete solution to the three tier problem. Results in this chapter allow for a wide range of new and emerging wireless networks to be analyzed within

---

A part of this chapter is published as: Gopalam, S., Hanly, S. V., & Whiting, P. (2020). "Distributed User Association and Resource Allocation Algorithms for Three Tier HetNets." *IEEE Transactions on Wireless Communications*, 19(12), 7913-7926.

a common framework. Examples of these networks include 1) mmWave small cell networks, 2) networks with aerial platforms and 3) multi-tier radio access technologies. In the following paragraphs, we discuss important future technologies which can be treated under this framework.

Ultra dense mmWave cellular networks are expected to play a key role in 5G [7, 23]. Wireless backhaul solutions are proposed to enable such a dense deployment [7]. Connectivity of mmWave links can be highly intermittent due to blocking by mobile objects [7, 23]. Multi-connectivity solutions (where a UE maintains connection to multiple BSs, so that there can be a fast handover in the event of blockage) are being studied to deal with blocking [23]. In section 2.5, we treat mmWave small cells under our optimization framework. We consider the effect of wireless backhaul and offloading of blocked users to microwave BSs and provide insights into system design and backhaul planning.



(a) Network with Drones

(b) HAP Network

Figure 2.1: HetNet with UAVs. (© 2020 IEEE)

In addition to terrestrial networks, wireless communication using aerial platforms is also being considered for future networks [24]. In low altitude platform applications, unmanned aerial vehicles (UAVs) are deployed to provide wireless access services as BSs, or can take the role of UEs requiring wireless access from the existing BSs [25]. In high altitude platform (HAP) applications, aircrafts or airships are deployed at altitudes of 17 to 22 km in the stratosphere to provide wireless connectivity over a large area [26]. HAPs have a very large coverage area, typically a few macro-sites, adding an extra tier at the top of the existing terrestrial network. These networks can be modelled as three tier HetNets as shown in Figure. 2.1.

Our framework has the following features which are common to all the above mentioned applications. 1) The BSs can be divided into tiers based on their coverage area. Generally, the higher tier

cells have BSs at higher altitudes and cover larger coverage areas. Several smaller cells can operate in the coverage area of a high tier cell. 2) A UE can potentially associate and get service from multiple different tier BSs. 3) Cross-tier interference - a higher tier BS causes debilitating interference to the smaller cells in its coverage area, if using same resources. From two to three tiers, there is an increase in the dimensionality of the joint optimization problem, e.g., two resource variables per UE to three per UE. Therefore, complexity of algorithms is a crucial consideration, which we address in this chapter.

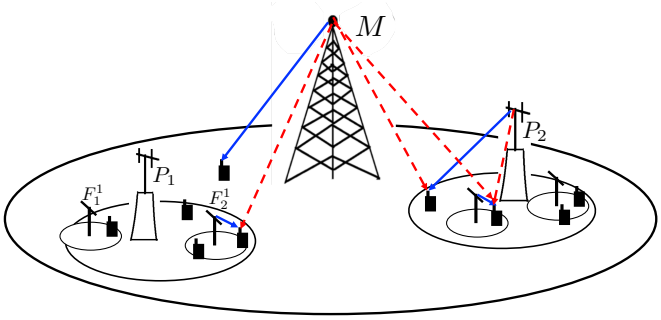


Figure 2.2: A general three tier HetNet. The blue lines depict the BS to UE links and the red lines depict the interference. (© 2020 IEEE)

Joint user association and resource allocation problems were studied for HetNet control in several works in the literature. In [4, 27–30], the approach was utility maximization. In [31–33], stochastic geometry was used to derive results. In [5, 34–39], optimization for flow-based models was considered. [40–43] considered utility maximization including power control.

Although some works modelled k-tier HetNets, they had drawbacks. In [4, 29, 35, 39] resource partitioning between tiers to avoid cross-tier interference was not considered. The solutions in [32, 33, 36] are not adaptive to the changes in traffic, and hence not suitable for real-time control. Also, the same bias value was applied to all the BSs in a tier, which is restrictive. Centralized solutions were proposed in [30, 40, 41, 43]. Only heuristic solutions were given in [40, 42, 43]. Also, the resource partitioning and user association results in [32] were derived using simulation and cannot be used for real-time control.

In this chapter, we consider the objective of clearing a given set of files in the network using minimum possible resources, and refer to it as the *minimum time clearing problem*. In our prior work [37, 38, 44], similar formulations were used to derive joint optimization results for two tier HetNets. The three tier problem was considered in an early investigation in [20]. However, the proposed solution

involved a high dimensional search (equal to number of femto BSs) to find the solution, which makes it prohibitive for real-time implementation.

We consider the problem of jointly optimizing user association and resource allocation in a three tier downlink HetNet. We refer to the tiers as macro, pico and femto tiers. We formulate the minimum time clearing problem as a linear program (LP). We show that by fixing the time allocated to small cells, the LP can be decomposed into several (equal to number of pico BSs) smaller independent LPs. It follows that significant parallelization can be achieved by solving these LPs simultaneously at the corresponding pico BSs. We then show that the user association is determined by a set of rate-bias multipliers, one multiplier per BS. The problem of finding the multipliers using conventional approaches leads to a high dimensional search, e.g., [4, 20]. In contrast, we present new structural results which enable us to propose more efficient algorithms with reduced complexity.

2.1.1 Contributions

- We provide a tractable framework for joint-optimization and cross-tier interference avoidance in three tier HetNets. The framework can be applied in a real-time manner for optimal control of HetNets, or can be used a offline tool for downlink capacity analysis.

- We present distributed algorithms to find the optimal solution under the proposed framework. The algorithms are highly efficient due to the new structural results we obtain in the chapter.

- We show that each rate-bias value (corresponding to a BS) crucially only takes values from a discrete set. For a pico P_i, the size is less than $0.5|U_i|^2$, where $|U_i|$ is the number of UEs covered by the pico. The search space of multipliers is reduced to a small finite set which we characterize.

- We further show that the solution of the LP at each pico BS is determined by just two parameters: the femto time allocation and the pico rate-bias.

We now present the outline of the chapter. In Section 2.2, we describe the system model and problem formulation. In Section 2.3, we present the main results of the chapter. In Section 2.4, we present the numerical results derived using simulations. In Section 2.5, we treat the HetNet with mmWave small cells and backhaul under the framework developed in the chapter.

2.2 System Model and Problem Formulation

Consider a 3-tier HetNet as shown in Figure. 2.2. There are N pico BSs labelled as $\{P_i\}_{i=1}^{N}$, operating in the coverage area of the macro BS M. There are N_i femto BSs operating in the coverage area of a pico P_i, labelled as $\{F_j^i\}_{j=1}^{N_i}$. Let U_j^i denote the set of UEs that are in the coverage area of F_j^i. A UE $u \in U_j^i$ can receive data from the BSs F_j^i, P_i and M as shown in Figure. 2.2. Let $U_i := \bigcup_{j=1}^{N_i} U_j^i$ denote the set of UEs that are covered by P_i.[1] Similarly, let $U := \bigcup_{i=1}^{N} U_i$ denote the set of all the UEs.

We consider time division duplexing (TDD) for resource partitioning. A high tier BS causes cross-tier interference to the smaller tier BSs in its coverage area, i.e., M causes interference to all the other BSs, and P_i causes interference to F_j^i. We consider significant cross-tier interference in our model (as it is the case in HetNets, e.g., [32]). Therefore, two interfering BSs such as P_i and F_j^i are not allowed to transmit at the same time under our model.

Let $\mathcal{B}_j^i$ denote the set of all the BSs excluding F_j^i, P_i and M. We consider the rate (in bits/sec) of the link between F_j^i and a UE $u \in U_j^i$ as follows

$$T_u = B \log_2(1 + p_{F_j^i} g_{F_j^i,u} / (\sigma^2 + I_{\mathcal{B}_j^i,u}))$$

where B is the transmission bandwidth, $g_{b,u}$ is channel gain between the BS b and the UE u. $g_{b,u}$ includes the antenna gain, path loss and shadowing loss. p_b is the transmit power of the BS b and σ^2 is the noise floor. The term $I_{\mathcal{B}_j^i,u}$ is the interference caused to the transmissions from F_j^i to u by the BSs in $\mathcal{B}_j^i$, i.e., BSs which are not covering F_j^i. We treat $I_{\mathcal{B}_j^i,u}$ as static interference which depends on $\mathcal{B}_j^i$ and u, i.e., as another noise source [2]. Similar assumptions are commonly adapated in the literature, e.g., [35, 39].

Therefore, T_u is rate of the link between F_j^i and u, provided the pico P_i and the macro M are muted. Similarly, let R_u (and S_u) denote the rate that a UE at user site u can receive from the macro M (and the pico P_i resp.), provided the interfering BSs are muted.

[1] For notational simplicity, we do not explicitly model the UEs that have no femto connectivity and only have coverage from a pico Pi and macro M. Such UEs can be treated as being in range of a virtual femto Fi which Ni+1 provides zero rate. For these UEs, some of the thresholds calculated in the chapter will be infinite, but this only means that the UEs do not associate with the virtual femto that provides zero rate.

[2] Strictly speaking, interference depends on the set of BSs in $\mathcal{B}_j^i$ transmitting in a given slot, and is upper-bounded by the worst-case interference $\sum_{b \in \mathcal{B}_j^i} p_b g_{b,u}$. However, fractional frequency re-use schemes are usually adopted to mitigate any significant co-tier interference between two near BSs of same tier [39], where the assumption is justified.

2.2.1 Minimum time clearing LP

A UE u has to download a file of size D_u bits. The file can be downloaded from any BS that is in range. A UE $u \in U_j^i$ can possibly download a part of the file from each of BSs M, P_i & F_j^i. For $u \in U_j^i$, let x_u (y_u & z_u) denote the amount of file (in bits) downloaded from the macro M (pico P_i & femto F_j^i resp.). For this setup, the minimum time (in sec) required to clear the traffic of all the UEs is formulated as LP (2.1-2.4).

$$\min_{x_u, y_y, z_u, \pi, \epsilon_i \geq 0} \pi + \sum_{u \in U} x_u / R_u \tag{2.1}$$

$$\text{s.t.} \sum_{u \in U_i} y_u / S_u \leq \pi - \epsilon_i, \ \forall i \in \{1, 2, \ldots N\} \tag{2.2}$$

$$\sum_{u \in U_j^i} z_u / T_u \leq \epsilon_i, \ \forall j \in \{1, 2, \ldots N_i\}, i \in \{1, 2, \ldots N\} \tag{2.3}$$

$$x_u + y_u + z_u = D_u, \ \forall u \in U \tag{2.4}$$

In (2.1), $\sum_{u \in U} x_u / R_u$ is the time used by the macro and π is the total time used by the small cell BSs. Out of time π, $\pi - \epsilon_i$ is used by a pico P_i in (3.3) and the rest ϵ_i is used by each of the femtos $\{F_j^i\}_{j=1}^{N_i}$ in (3.4). Note that the BSs M_i, P_i and F_j^i are allocated different times under the LP, thus avoiding cross-tier interference.

Fixing a value of π, LP (2.1-2.4) can be divided into N independent LPs, one for each pico P_i. The LP involving P_i is formulated as LP (2.5). Let $f_i(\pi)$ denote the optimal solution of LP (2.5) for a given π. The solution of LP (2.1-2.4) is given by $\min_{\pi \in [0, \infty)} \pi + \sum_{i=1}^{N} f_i(\pi)$.

$$\min_{x_u, y_u, z_u, \epsilon_i \geq 0} \sum_{u \in U_i} x_u / R_u$$

$$\alpha \text{ constraint :} \quad \text{s.t.} \sum_{u \in U_i} y_u / S_u \leq \pi - \epsilon_i$$

$$\beta_j \text{ constraint :} \quad \sum_{u \in U_j^i} z_u / T_u \leq \epsilon_i, \ \forall j \in \{1, 2, \ldots N_i\}$$

$$\gamma_u \text{ constraint :} \quad x_u + y_u + z_u = D_u, \ \forall u \in U_i \tag{2.5}$$

Note that $f_i(\pi)$ can be computed at pico P_i. Hence, the computation of clearing time $\pi + \sum_{i=1}^{N} f_i(\pi)$ can be parallelized and distributed over the picos. Fig 2.4(a) (in page 26) depicts a distributed scheme using message passing for such a computation. Due to the convex nature of the function $\pi + \sum_{i=1}^{N} f_i(\pi)$,

the optimal π and clearing time can be found by methods such as a line search. This is the strategy that we will follow to find the solution of LP (2.1-2.4).

In the rest of the chapter, we will focus on finding $f_i(\pi)$ at a pico P_i for an arbitrary $i \in \{1, \dots N\}$, i.e., we will focus on the solution of LP (2.5). The notation used in the chapter is summarised in the following table.

Table 2.1: Table of Notation. (© 2020 IEEE)

<table>
<tr><th>Notation</th><th colspan="2">Description</th></tr>
<tr><td>x_u; y_u; z_u</td><td colspan="2">Amount of file (in bits) allocated to $u \in U_j^i$ by macro M; pico P_i; femto F_j^i resp.</td></tr>
<tr><td>R_u; S_u; T_u</td><td colspan="2">Rate of the link (in bits/sec) between $u \in U_j^i$ and macro M; pico P_i; femto F_j^i resp.</td></tr>
<tr><td>$\pi - \epsilon_i$</td><td colspan="2">Time allocated for transmissions of pico P_i</td></tr>
<tr><td>ϵ_i</td><td colspan="2">Time allocated for simultaneous transmissions of femtos $\{F_j^i\}_{j=1}^{N_i}$</td></tr>
<tr><td>D_u</td><td colspan="2">Total file size of u (in bits)</td></tr>
<tr><td>ρ_u^α</td><td>$\min\{T_u/R_u, \alpha T_u/S_u\}$</td><td rowspan="3">$\alpha, \beta_j$ and ρ_u^α are
defined in Theorem 2.3.1
in the following section</td></tr>
<tr><td>1; α; β_j</td><td>Rate bias multiplier corresponding to macro M; pico P_i; femto F_j^i resp.</td></tr>
<tr><td>R_u; $\frac{S_u}{\alpha}$; $\frac{T_u}{\beta_j}$</td><td>Biased rate of $u \in U_j^i$ from macro M; pico P_i; femto F_j^i resp.</td></tr>
</table>

2.3 Main Results

For any $u \in U_i$, let x_u^*, y_u^* and z_u^* denote the value of x_u, y_u and z_u respectively under an optimal solution of LP (2.5). Let $\boldsymbol{x} := [x_u]_{u \in U_i}$, $\boldsymbol{y} := [y_u]_{u \in U_i}$ and $\boldsymbol{z} := [z_u]_{u \in U_i}$. We present the main results as the following Theorems. For proofs, refer to Appendix 2.7.1.

Theorem 2.3.1 (Rate biasing rule)**.** *There exist optimal rate-bias multipliers, $\alpha^* > 0$ corresponding to P_i and $\beta_j^* > 0$ corresponding to F_j^i, $\forall j \in \{1, \dots N_i\}$ which determine the user-association as follows. For any $u \in U_j^i$,*

1) $x_u^* > 0$, only if $R_u = 1/\gamma_u^*$

2) $y_u^* > 0$, only if $S_u/\alpha^* = 1/\gamma_u^*$

3) $z_u^* > 0$, *only if* $T_u/\beta_j^* = 1/\gamma_u^*$.

where $1/\gamma_u^* := \max\{R_u, S_u/\alpha^*, T_u/\beta_j^*\}$ *is the maximum biased rate. Further, define* $\rho_u^\alpha := \min\{\alpha T_u/S_u, T_u/R_u\}$. *For any* $u \in U_j^i$, *the statements of the flow chart in Fig 2.3 hold true.*

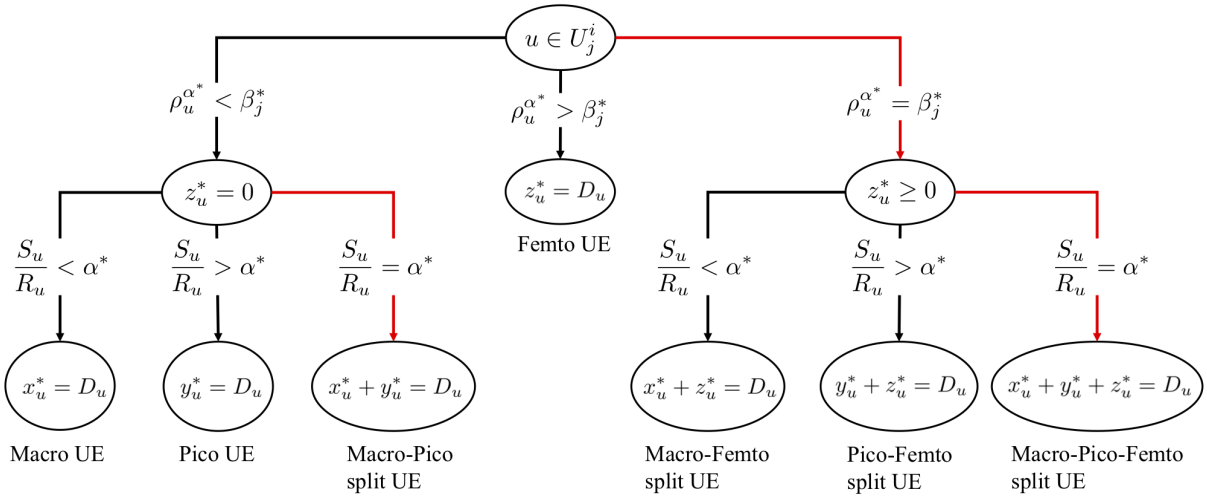


Figure 2.3: User Association flow chart. The conditions leading to split UE cases are colored in red. (© 2020 IEEE)

Theorem 2.3.1 states that the optimal user association is determined by a rate-biasing rule. The macro rate is not biased (or equivalently, the bias is 1). For a UE $u \in U_j^i$, the pico and femto biased rates are S_u/α^* and T_u/β_j^* respectively. A UE u associates with the BS (or BSs) providing the highest biased rate $1/\gamma_u^* = \max\{R_u, S_u/\alpha^*, T_u/\beta_j^*\}$. e.g., 1) If $R_u > \max\{S_u/\alpha^*, T_u/\beta_j^*\}$, the UE u associates with macro M (Macro UE case in Figure. 2.3), 2) If $R_u = S_u/\alpha^* > T_u/\beta_j^*$, the UE u associates with both the macro M and pico P_i (Macro-Pico split UE case in Figure. 2.3).

Figure. 2.3 presents the possible cases of allocation that can occur under the rate-biasing rule. Theorem 2.3.1 provides a partial characterization of the solution via Figure. 2.3. Given the optimal multipliers, Theorem 2.3.1 determines the allocation for non-split UEs (as given in Figure. 2.3). But, the split UEs may receive a part of the file from each associated BS which is not given here.

Algorithm 1 (on page 23) provides the full solution. To develop the algorithm, we first present the following two theorems concerning the optimal solution and the multipliers.

Theorem 2.3.2 (Finite set of rate-bias multipliers)**.** *Let* $\rho_u^\alpha := \min\{\alpha T_u/S_u, T_u/R_u\}$. *The rate bias multipliers* α^* *and* β_j^* *take values from finite sets A and* B_j *respectively, as follows*

(i) $\alpha^* \in A := \bigcup_{j=1}^{N_i} \{\frac{S_a T_b}{T_a R_b} : \frac{T_b}{R_b} \leq \frac{T_a}{R_a}, \frac{T_a}{S_a} \leq \frac{T_b}{S_b}\}_{(a,b)\in U_j^i \times U_j^i}$

(ii) $\beta_j^* \in B_j := \{\rho_u^\alpha : u \in U_j^i\}_{\alpha \in A}, \forall j \in \{1, \dots N_i\}$

Note that $|A| \leq \sum_{j=1}^{N_i} \frac{|U_j^i|^2 + |U_j^i|}{2}$.

There are only a finite set of possible values for the multipliers, which are given in Theorem 2.3.2. Naturally, one can be tempted to implement a discrete search to find the optimal multipliers. However as explained in the previous paragraphs concerning Theorem 2.3.1, such knowledge does not provide the solution for split UEs. The following theorem forms the basis of our algorithmic solution, which achieves two goals, 1) it provides the full allocation, including split UEs, and 2) it reduces the dimensionality of the problem to just 2.

Theorem 2.3.3 (Allocation function)**.** *There exists an allocation function* $\Theta : \mathbb{R}_+ \times [0, \pi] \to \mathbb{R}_+^{3|U_i|+N_i}$ *(defined in Algorithm 1), which provides a mapping from a pair of* $\{\alpha, \epsilon_i\} \in \mathbb{R}_+ \times [0, \pi]$ *to a solution* $[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}]$ *of LP (2.5) and the femto multipliers* $\boldsymbol{\beta} = [\beta_j]_{j=1}^{N_i} > 0$ *as follows.*

$$[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\beta}] = \Theta(\alpha, \epsilon_i)$$

Moreover, the function satisfies $[\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*, \boldsymbol{\beta}^*] = \Theta(\alpha^*, \epsilon_i^*)$*, where* α^* *is the optimal pico rate-bias multiplier and* ϵ_i^* *is the optimal femto time in LP (2.5).*

Theorem 2.3.3 provides the full solution of LP (2.5) as $\Theta(\alpha^*, \epsilon_i^*)$. It also shows that the solution is determined by just two variables - α^* and ϵ_i^*. Hence, a search over 2 parameters: discrete search for α^* over A and a continuous search for ϵ_i^* over $[0, \pi]$, can be implemented to solve LP (2.5) (in contrast to a high dimensional search for $N_i + 1$ multipliers e.g., [4, 20]).

Algorithm 1 Allocation Function $\Theta(\alpha, \epsilon_i)$

1: Run Algorithm 2 to obtain $\{\mathcal{F}_j(\alpha, \epsilon_i)\}_{j=1}^{N_i}$ and to evaluate $\boldsymbol{\beta}, \boldsymbol{z}, a, b, \delta$. // Femto allocation step

// A special case that can occur is when two split UEs, a femto-pico split UE a and a femto-macro split UE b are in U_j^i for some j (See step 5 in Algorithm 2). In this case, z_a, z_b will be determined by Algorithm 3 in the next step.

2: Run Algorithm 3 to obtain $\mathcal{P}(\alpha, \epsilon_i, \boldsymbol{z}, a, b, \delta)$, and to evaluate $\boldsymbol{y}, z_a, z_b$. // Pico allocation step

3: $x_u = D_u - y_u - z_u, \forall u \in U_i$ // Macro allocation step

4: **return** $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\beta}$

The allocation function $\Theta(\alpha, \epsilon_i)$ of Theorem 2.3.3 (given in Algorithm 1) defined using the femto allocation functions $\mathcal{F}_j(\alpha, \epsilon_i), j \in \{1, \ldots N_i\}$ given in Algorithm 2 (on page 24) and a pico allocation function $\mathcal{P}([\mathcal{F}_j(\alpha, \epsilon_i)]_{j=1}^{N_i})$ given in Algorithm 3 (on page 25).

The femto allocation function $\mathcal{F}_j(\alpha, \epsilon_i)$ determines the femto multiplier β_j and femto allocation $[z_u]_{u \in U_j^i}$ for the femto F_j^i. The pico function $\mathcal{P}([\mathcal{F}_j(\alpha, \epsilon_i)]_{j=1}^{N_i})$ takes the outputs from the femto functions and determines the pico allocation $[y_u]_{u \in U_i}$ for the pico P_i. The macro allocation $[x_u]_{u \in U_i}$ can be completed by line 3 in Algorithm 1. The individual steps in Algorithm 2 and Algorithm 3 are justified by the Lemmas mentioned in the corresponding steps.

Algorithm 2 Femto Allocation Algorithm $\mathcal{F}_j(\alpha, \epsilon_i)$

1: Initialize $flag_j = 0$ // This flag is used to note the occurrence of two split users case, and 0 by default.

2: Sort $u \in U_j^i$ in descending order of ρ_u^α such that $\rho_{u_1}^\alpha \geq \ldots \geq \rho_{u_K}^\alpha$. // where $K = |U_j^i|$

3: **if** $\sum_{k=1}^{K} D_{u_k}/T_{u_k} \leq \epsilon_i$ **then** // No split users case.

$$z_{u_k} = D_{u_k} \text{ for } 1 \leq k \leq K$$

$$\beta_j = \rho_{u_K}^\alpha \quad \text{(See Lemma 2.7.4 in Appendix 2.7.5 \& Appendix 2.7.7)}$$

4: **else if** $\exists l \leq K$ such that $\rho_{u_l}^\alpha \neq \rho_{u_k}^\alpha, \forall k \neq l$ and satisfying $\sum_{k=1}^{l-1} D_{u_k}/T_{u_k} \leq \epsilon_i < \sum_{k=1}^{l} D_{u_k}/T_{u_k}$ **then**
// One femto split user case

$$z_{u_k} = \begin{cases} D_{u_k} & \text{for } 1 \leq k \leq l-1 \\ T_{u_l}(\epsilon_i - \sum_{k'=1}^{l-1} D_{u_{k'}}/T_{u_{k'}}) & \text{for } k = l \\ 0 & \text{for } l+1 \leq k \leq K \end{cases}$$

$$\beta_j = \rho_{u_l}^\alpha \qquad \text{(See Lemma 2.7.5 in Appendix 2.7.5)}$$

5: **else if** $\exists l \leq K-1$ such that $\rho_{u_l}^\alpha = \rho_{u_{l+1}}^\alpha$ and $\sum_{k=1}^{l-1} D_{u_k}/T_{u_k} \leq \epsilon_i < \sum_{k=1}^{l+1} D_{u_k}/T_{u_k}$ **then** // Two femto split users case. We now set a flag to denote that this case occurred.

6: $flag_j = 1, a := u_l, b := u_{l+1}$ // femto-pico split UE is a, femto-macro split UE is b

$$z_{u_k} = \begin{cases} D_{u_k} & \text{for } 1 \leq k \leq l-1 \\ 0 & \text{for } l+2 \leq k \leq K \end{cases}$$

$$\beta_j = \rho_{u_l}^\alpha = \rho_{u_{l+1}}^\alpha \qquad \text{(See Lemma 2.7.6 in Appendix 2.7.5)}$$

7: $\delta := \epsilon_i - \sum_{k=1}^{l-1} D_{u_k}/T_{u_k}$. // δ is the time remaining for a and b. The allocation z_a and z_b is determined in step 7 of Algorithm 3

8: **end if**

9: **return** $\{z_u\}_{u \in U_j^i - \{a,b\}}, \beta_j, a, b, \delta$

Algorithm 3 Pico Allocation Algorithm $\mathcal{P}(\alpha, \epsilon_i, \boldsymbol{z}, a, b, \delta)$

1: W be the set of UEs $u \in U_i - \{a, b\}$ such that $D'_u > 0$, where $D'_u := D_u - z_u$ // not femto UEs

2: Sort $w_k \in W$ in descending order such that $S_{w_1}/R_{w_1} > \ldots > S_{w_{|W|}}/R_{w_{|W|}}$.

3: **if** $flag_j = 0, \forall j \in \{1, \ldots N_i\}$ **then** // Two split users case did not occur

4: Find $l \leq |W|$ such that $\sum_{k=1}^{l-1} D'_{w_k}/S_{w_k} < \pi - \epsilon_i \leq \sum_{k=1}^{l} D'_{w_k}/S_{w_k}$

$$y_{w_k} := \begin{cases} D'_{w_k} & \text{for } 1 \leq k \leq l-1 \\ S_{w_l}(\pi - \epsilon_i - \sum_{k'=1}^{l-1} D'_{w_{k'}}/S_{w_{k'}}) & \text{for } k = l \\ 0 & \text{for } l+1 \leq k \leq |W| \end{cases}$$

(See Lemma 2.7.7 in Appendix 2.7.6)

5: **else if** $flag_j = 1$, for one $j \in \{1, \ldots N_i\}$ **then** // Two femto split users case occurred in $\mathcal{F}_j^i$

6: Find $l \leq |W|$ satisfying $S_{w_l}/R_{w_l} > \alpha > S_{w_{l+1}}/R_{w_{l+1}}$

$$y_{w_k} = \begin{cases} D'_{w_k} & \text{for } 1 \leq k \leq l \\ 0 & \text{for } l+1 \leq k \leq |W| \end{cases}$$

$$y_a = S_a(\pi - \epsilon_i - \sum_{k=1}^{l} D'_{w_k}/S_{w_k})$$

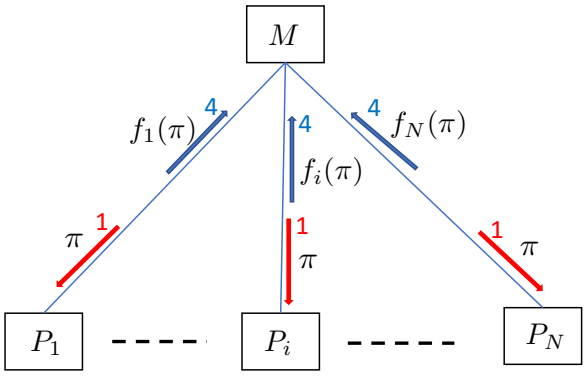
$$y_b = 0$$

7: $z_a = D_a - y_a$, $z_b = T_b(\delta - z_a/T_a)$ (See Lemma 2.7.8 in Appendix 2.7.6)

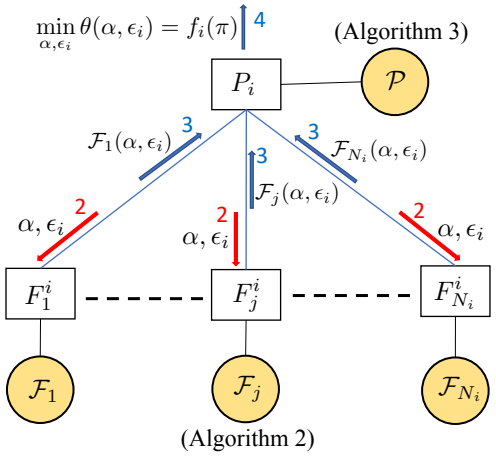
8: **end if**

9: **return** $\boldsymbol{y}, z_a, z_b$

2.3.1 Scalable and Distributed Implementation



(a) Macro level distributed scheme for computing clearing time.



(b) Pico level distributed scheme to evaluate $\Theta(\alpha,\epsilon_i)$.

Figure 2.4: Distributed computation of clearing time using message passing algorithms. The circles represent the allocation functions at the BSs, and the arrows represent the message exchanges. The numbering on the arrows is the order in which the message exchanges occur. (© 2020 IEEE)

Figure. 2.4(b) shows a distributed implemenation to evaluate the allocation function $\Theta(\alpha,\epsilon_i)$. The scheme can be implemented as follows. The pico broadcasts a message containing the values (α,ϵ_i) to the femtos $F_j^i, \forall j \in \{1,\dots N_i\}$. Then, each femto F_j^i runs the function $\mathcal{F}_j(\alpha,\epsilon_i)$ locally. Hence, the femto allocations $\{\mathcal{F}_j(\alpha,\epsilon_i)\}_{j=1}^{N_i}$ can be computed in parallel at the corresponding femtos. Following the computation, each femto F_j^i sends the evaluation $\mathcal{F}_j(\alpha,\epsilon_i)$ to the pico P_i. The pico P_i then computes $\mathcal{P}([\mathcal{F}_j(\alpha,\epsilon_i)]_{j=1}^{N_i})$, which completes the allocations $\boldsymbol{z},\boldsymbol{y}$. Line 3 of Algorithm 1 determines $\boldsymbol{x}$.

This implementation is scalable in number of femtos due to local nature of the functions $\mathcal{F}_j^i$. The only increase is in the number of passed messages to the pico which is equal to number of femtos N_i. A similar statement about scalability also holds true for macro-level process shown in Figure. 2.4(a). The worst case computational complexity of the function $\mathcal{F}_j$ is $O((|U_j^i|+1)\log|U_j^i|)$, and for function $\mathcal{P}$ it is $O((|U_i|+1)\log|U_i|)$.

The only thing left is the search procedure to find the optimal values(α^*,ϵ_i^*). Let $\theta(\alpha,\epsilon_i)$ be the value of the objective function $\sum_{u\in U_i} x_u/R_u$ under the solution given by $\Theta(\alpha,\epsilon_i)$. If the solution is infeasible, we take $\theta(\alpha,\epsilon_i)$ to be ∞. Now, $(\alpha^*,\epsilon_i^*) := \arg\min_{\alpha\in A,\epsilon_i\in[0,\pi]} \theta(\alpha,\epsilon_i)$, and the optimal value

of LP (2.5), $f_i(\pi)$ is given by

$$f_i(\pi) = \theta(\alpha^*, \epsilon_i^*) \tag{2.6}$$

The search algorithms and their convergence results are presented in the following section.

2.4 Numerical Results

To illustrate the results, we consider a three tier HetNet with a macro BS, 6 pico BSs and 4 femto BSs per pico site. The simulation parameters are given in the following tables. The BS parameters are in the order: macro, pico, femto.

Table 2.2: Simulation Parameters. (© 2020 IEEE)

BS parameters	**Values**
Transmit power	46, 30, 22 (in dBm)
Antenna gain	14, 5, 3 (in dBi)
Path-loss exponent n	3.76, 3.76, 3
Coverage radius	500, 150, 50 (in m)
Log-normal shadowing standard deviation	10, 6, 6 (in dB)

Parameter	**Value**
Transmission bandwidth	10 MHz
File size D_u	2.7 Mb
UE noise figure	10 dB
Noise power	-106 dBm
Minimum inter-BS distance	300 m (for pico tier) 90 m (for femto tier)

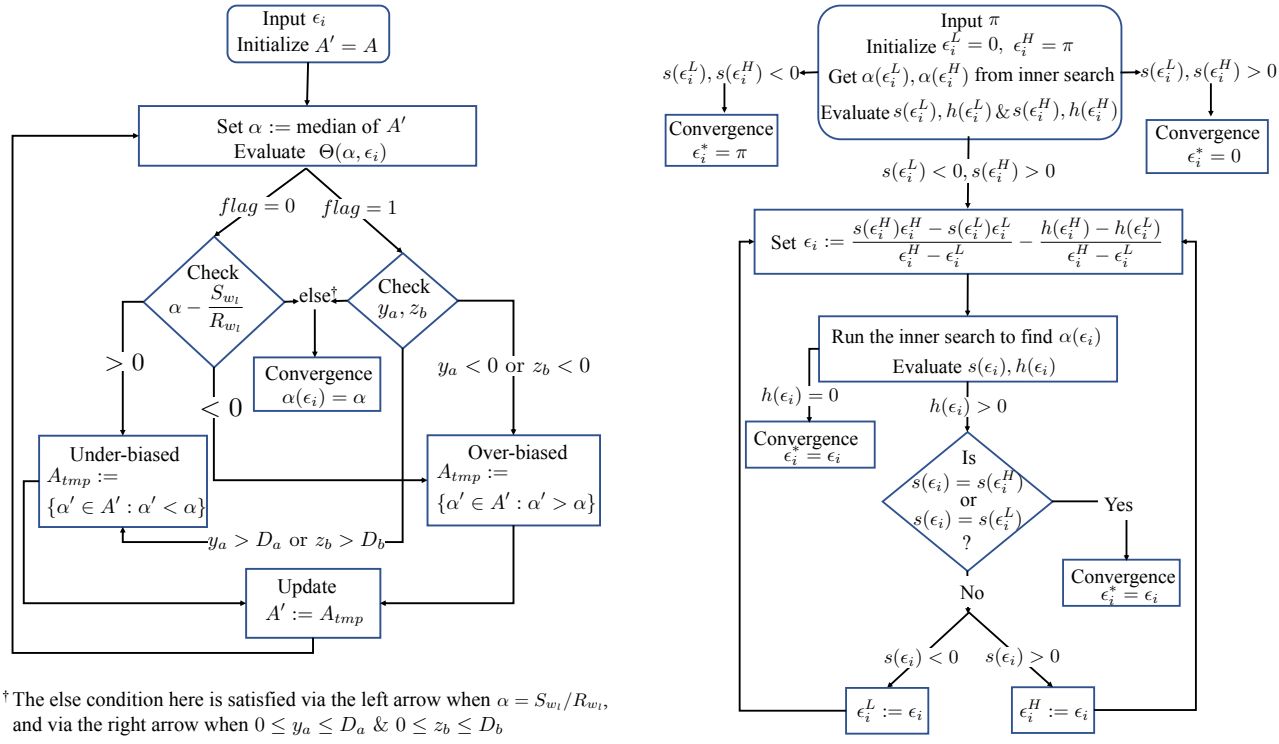
The macro BS is placed at the origin, the other BS locations are randomly realized in the macro coverage region such that inter-BS distances are greater than the specified values. We consider circular cells with the specified radii; a UE receives signal from a BS if within the coverage radius. UE placement is done in two stages, 5 UEs are uniformly scattered in each femto cell in the first stage, and

20 UEs are uniformly scattered in each pico cell in the second stage. The path-loss (in dB) formula is $128 + 10n\log_{10}(d/\text{km})$, where d is the BS-UE distance.

2.4.1 Search for α^*, ϵ_i^*

In this section, we focus on the search to find ϵ_i^* and α^*. We start by fixing $\pi = 0.4$ sec in LP (2.1-2.4), and solve LP (2.5) by finding ϵ_i^* and α^*. Recall from (2.6) that $f_i(\pi) = \theta(\alpha^*, \epsilon_i^*)$. [3]

For a given ϵ_i, define $\alpha(\epsilon_i) := \arg\min_{\alpha \in A} \theta(\alpha, \epsilon_i)$ as the α that minimizes the objective function. We consider a layered search over α, ϵ_i. In section 2.4.1.1, the inner search to find $\alpha(\epsilon_i)$ (shown in Figure. 2.5(a)). In section 2.4.1.2, the outer search for ϵ_i^* (shown in Figure. 2.5(b)). Note that $\alpha^* = \alpha(\epsilon_i^*)$, hence both α^* and ϵ_i^* are derived here.



(a) Inner search for $\alpha(\epsilon_i)$.

(b) Outer search for ϵ_i^*

Figure 2.5: Search algorithms to find α^*, ϵ_i^*. (© 2020 IEEE)

[3] There is macro-level search over π to minimize $\pi + \sum_{i=1}^{N} f_i(\pi)$ is presented in section 2.4.3. The value $\pi = 0.4 < \pi^*$ is chosen such that the constraints are tight, i.e., $f_i(\pi) > 0, \forall i$. The search is more straightforward when there is slackness.

2.4.1.1 Inner search for $\alpha(\epsilon_i)$

We find the $\alpha(\epsilon_i)$ for the given ϵ_i using inner search in Figure. 2.5(a). Recall that $flag_j = 1$ is used to denote two split users case in Algorithm 2. Define $flag := \max_{j=1}^{N} flag_j$. One of the following conditions will hold when the input $\alpha = \alpha(\epsilon_i)$

i) If $flag = 0$, then $\alpha = S_{w_l}/R_{w_l}$, where w_l is the split user in Algorithm 3. (See Lemma 2.7.7 in Appendix 2.7.6)

ii) If $flag = 1$, then $\alpha = S_a T_b / T_a R_b$, $0 \leq y_a \leq D_a$ and $0 \leq z_b \leq D_b$, where a, b are the two femto split users in Algorithms 2 & 3. (See Lemma 2.7.6 in Appendix 2.7.5)

When conditions i) and ii) do not hold, either $\alpha > \alpha(\epsilon_i)$ (over-biased) or $\alpha < \alpha(\epsilon_i)$ (under-biased). Figure. 2.5(a) provides the criteria to check this relation between α and $\alpha(\epsilon_i)$, depending on the value of $flag$. Using this property, Figure. 2.5(a) performs a binary search for $\alpha(\epsilon_i)$ over A. In each iteration, $|A_{tmp}| \leq |A'|/2$, since α is the median of set A'. Therefore, the set of possible α's is halved in size during the update $A' := A_{tmp}$. Hence, the convergence time (in steps) is at most $\log_2 |A|$.

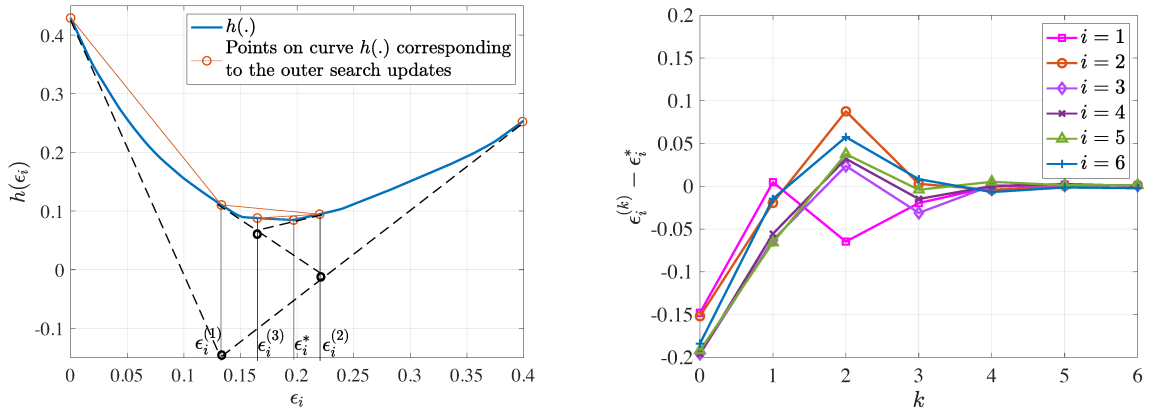
The average convergence times of inner search for the 6 picos are $[6.43, 5.12, 6.25, 4.25, 6.12, 6.12]$ steps respectively, where $|A|$ for the picos are $[82, 82, 80, 79, 76, 76]$ respectively. Here, the averages are calculated over the input ϵ_i's given by the outer search updates in Figure. 2.6(b).

2.4.1.2 Outer search for ϵ_i^*

Define $h(\epsilon_i) := \theta(\alpha(\epsilon_i), \epsilon_i)$. Note that $h(\epsilon_i)$ is the value of LP (2.5) for a fixed given ϵ_i. We use the convexity of $h(\cdot)$ to find $\epsilon_i^* := \arg\min_{\epsilon_i} h(\epsilon_i)$, using the outer search algorithm in Figure. 2.5(b).

Under the shadow-price interpretation of dual-variables, $s(\epsilon_i) := \partial h(\epsilon_i)/\partial \epsilon_i = \alpha' - \sum_{j=1}^{N_i} \beta'_j$, where $\alpha', \{\beta'_j\}_{j=1}^{N_i}$ are the dual-variables corresponding to the pico-time and femto-time constraints in LP (2.5). The rate bias multipliers $\alpha(\epsilon_i), \{\beta_j\}_{j=1}^{N_i}$ are equal to the corresponding dual-variables, provided the corresponding constraint is not slack. When a constraint is slack, the corresponding dual-variable is zero. (Refer to Appendix 2.7.7 for more details). Note that $\alpha(\epsilon_i)$ and $\Theta(\alpha(\epsilon_i), \epsilon_i)$ are evaluated by the inner search algorithm (in Figure. 2.5(a)). The gradient $s(\epsilon_i)$ can now be calculated since 1) $\alpha(\epsilon_i)$ is known, and 2) the allocation $[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}]$ (which determines slackness of constraints) and rate-multipliers $\boldsymbol{\beta}$ are given by $\Theta(\alpha(\epsilon_i), \epsilon_i)$.

$h(\cdot)$ is a piecewise linear function (blue curve in Figure. 2.6(a)). Figure. 2.5(b) provides a linear interpolation based search algorithm to find ϵ_i^* in a finite number of steps. We take a point ϵ_i^L with a



(a) Outer search updates for $i = 1$

(b) Convergence of outer search for all the picos

Figure 2.6: Outer search algorithm. Here k is the number of iterations, and $\epsilon_i^{(k)}$ is the value of ϵ_i in kth iteration. (© 2020 IEEE)

negative gradient and a point ϵ_i^H with a positive gradient and solve for a new ϵ_i as the ϵ_i-coordinate of the intersection of tangents (of curve $h(\cdot)$) at points $(\epsilon_i^L, h(\epsilon_i^L))$ and $(\epsilon_i^H, h(\epsilon_i^H))$. e.g., In Fig 2.6(a), $\epsilon_i^L = 0, \epsilon_i^H = 0.4$ during iteration 1, and $\epsilon_i^{(1)}$ is the new ϵ_i. Now, either $\epsilon_i = \epsilon_i^*$ or the point $(\epsilon_i, h(\epsilon_i))$ lies on a new line segment of the curve $h(\cdot)$ (See Figure. 2.6(a)). Due to convexity of $h(\cdot)$, ϵ_i is closer to the ϵ_i^* than at least one of $\epsilon_i^L, \epsilon_i^H$. Finally, either ϵ_i^L or ϵ_i^H is updated based on the slope $s(\epsilon_i)$. The convergence occurs in finite number of steps because the curve $h(\cdot)$ is composed of a finite number of line segments.

The convergence results can be seen in Fig 2.6. In Fig 2.6(a), ϵ_i^L is updated in iterations 1 and 3 (since the slope $s(\epsilon_i)$ is negative), and ϵ_i^H is updated in iteration 2. Figure. 2.6(b) shows the convergence times (in number of iterations or steps) for all the 6 picos.

2.4.2 Alternate approximate methods and convergence times

The search algorithms given in Figure. 2.5 in section 2.4.1 derive the exact solution (α^*, ϵ_i^*) in finite number of steps. The simulation results indicate convergence with in a small number of steps. However, in practical implementation, issues like delay may impose additional constraints on search time. In this case, the search can be truncated and last calculated feasible solution can be used, which lies within $\epsilon_i^H - \epsilon_i^L$ distance of the optimal value ϵ_i^*.

Alternatively, we now present an approximate scheme with bounded convergence time (in steps). Here, the parameters α, ϵ_i are allowed to take values from a predefined finite set, e.g., quantized levels

for parameters. Let S_α, S_{ϵ_i} denote the sets of values that α and ϵ_i can take respectively. We present the modified search algorithms as follows.

For the inner search, the algorithm in Figure. 2.5(a) can be applied with initialization $A' = S_\alpha$, and stopped when $|A'| = 1$. The convergence time is $\log_2 |S_\alpha|$. For the outer search, a binary search version of the algorithm in Figure. 2.5(b) can be applied, where the new $\epsilon_i \in S_{\epsilon_i}$ will be chosen as the median value between ϵ_i^L and ϵ_i^H (instead of the intersection of the tangents). Convergence occurs in $\log_2 |S_{\epsilon_i}|$ steps (when $\epsilon_i^L = \epsilon_i^H$). The total convergence time is $\leq \log_2 |S_\alpha| \log_2 |S_{\epsilon_i}|$.

Hence, for the approximate scheme with bounded convergence time, the quantization for the sets S_α, S_{ϵ_i} can be chosen based on the latency requirement. In other words, the sizes $|S_\alpha|$, $|S_{\epsilon_i}|$ can be chosen such that $\leq \log_2 |S_\alpha| \log_2 |S_{\epsilon_i}|$ is less than the latency requirement.

2.4.3 Performance Results of the Minimum Time Clearing Scheme

Recall that there is also a process at macro level (shown in Figure. 2.4(a)) to solve LP (2.1-2.4), i.e., to derive $\pi^* := \min_\pi \pi + \sum_{i=1}^N f_i(\pi)$. A similar search method to Fig 2.5(b) or traditional methods such as golden section search, line search can be used to find π^*. Since, our main focus is on LP (2.5), we have only presented the convergence results for finding $f_i(\pi)$. Now, we present the clearing time $\pi + \sum_{i=1}^N f_i(\pi)$ as a function of π (red curve A in Figure. 2.7(a)), and compare with alternative schemes.

For comparison, we consider schemes A-D given in the following table. Scheme A is the minimum time clearing scheme of this chapter, which uses full resource partitioning (FRP) between the tiers. Scheme D has no resource partitioning (No RP) between the tiers, and all the BSs are allowed to transmit simultaneously. For the other schemes B&C, we consider Almost Blanking Subframes (ABS) scheme of 3GPP. Under ABS, resource partitioning at macro tier is performed; the macro is silent during the small cell (or ABS) time. The picos and femtos are taken to use the entirety of small cell time for transmission. Scheme C uses SINR biased user association, which is equivalent to the Cell Range Expansion (CRE) scheme of 3GPP. The other schemes A,B&D use rate biased user association (as explained in the chapter).

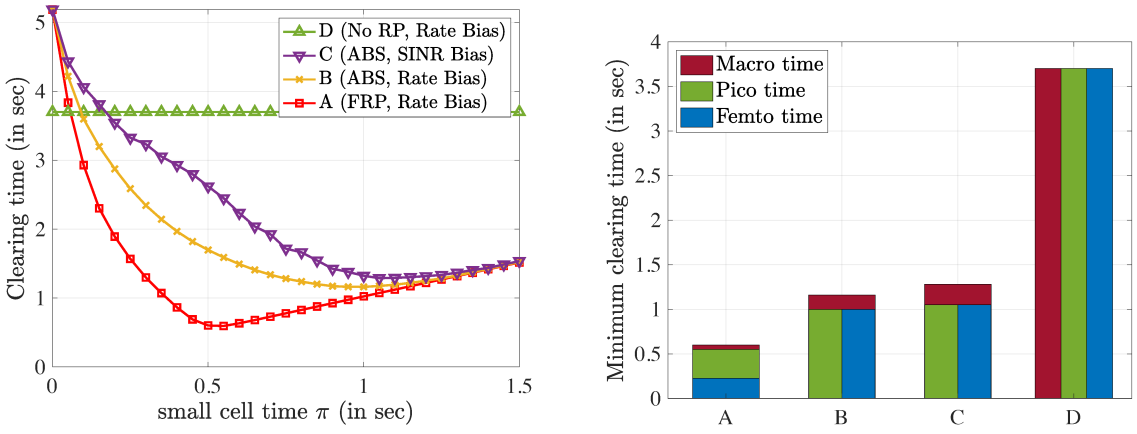
Note that the rates under No RP and ABS will be lower (than that of FRP) due to the cross-tier interference resulting from the simultaneous transmissions of different tiers.

We measure performance in terms of the time required to clear the files of a given set of UEs.

Table 2.3: Characteristics of HetNet control scheme. (© 2020 IEEE)

Scheme	Resource partitioning	UE association rule
A	FRP	Rate Bias
B; C	ABS	Rate Bias; SINR Bias resp.
D	No RP	Rate Bias

Note that smaller clearing time means higher capacity, since more files are transmitted per second. The schemes A-C are adaptive with respect to the small cell time π, and hence the clearing time is minimized over all possible choices of bias values for each π. Scheme D is fixed. It is optimized over all possible bias values and has a fixed small cell time π (given by optimal biasing). Therefore, the clearing times presented are the best possible for respective schemes. Note that the clearing time of C provides a lower-bound to the CRE and ABS schemes of 3GPP. D is a lower-bound to the rate-biased schemes in [4, 35, 36, 39].



(a) Clearing time comparison

(b) Macro and small cell times

Figure 2.7: Comparison of various user association and resource partitioning schemes. (© 2020 IEEE)

The results are presented in Figure. 2.7. It is clear that the minimum clearing scheme A performs better than the other schemes by definition. However, the difference is significant in the considered scenario, as can be observed from Figure. 2.7(a) and Figure. 2.7(b). It can also be observed that FRP (scheme A) provides significant gain over ABS (scheme B) for rate biased association, and the difference is even more significant between ABS (scheme B) and No RP (scheme D). This result

highlights the importance of resource partitioning in HetNets.

Figure. 2.7(b) shows the distribution of macro, pico and femto times across the considered schemes at their respective optima. Here, pico time (and femto time) is the time available to the picos (and the femto resp.). Under ABS (B&C), the small-cell time π is available to all the picos and the femtos. For schemes B&C, we illustrate this with two parallel bars (green & blue). Under FRP, recall that the time available to a F_j^i is ϵ_i, i.e., it depends on i. For scheme A, the stacked green and blue bars are the average pico time ($\pi - \sum_i \epsilon_i/N$) and femto time ($\sum_i \epsilon_i/N$) respectively. For D, the macro, pico and femto BSs are all transmitting at the same time, which is illustrated with three parallel bars (brown, green & blue). Scheme A has the smallest macro-cell load, followed by B & C. Lack of resource partitioning in D has resulted in a high macro-cell load.

2.5 Applications and Extensions

Thus far, the framework developed in the chapter has been used to obtain the minimum time clearing scheme A (FRP, Rate Bias) in section 2.4.3. However, the framework is more general and can be easily adapted to implement other three tier joint optimization schemes. As a more straightforward application, the framework can be used to optimize three tier user association under an ABS setup as follows.

2.5.1 Three HetNet under an ABS resource partitioning scheme

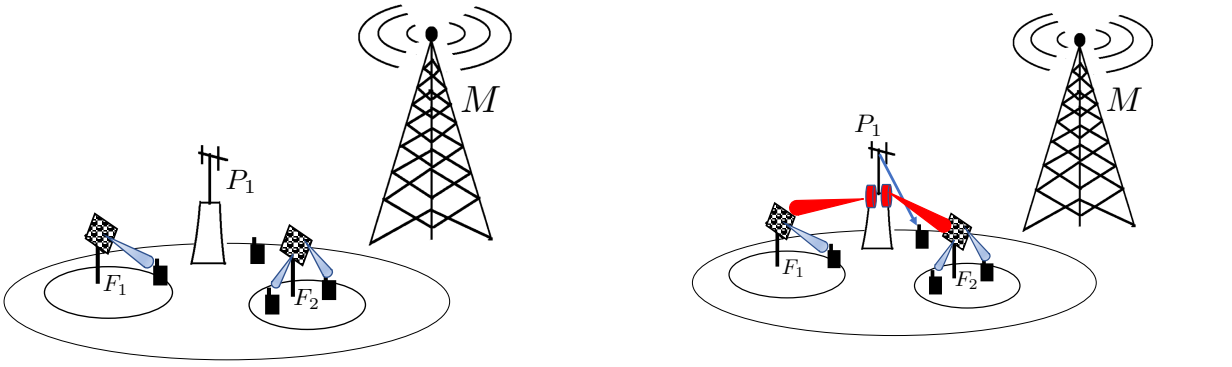
Consider the three tier HeNet (with BS setup described in section 2.2), now operating under the ABS resource partitioning scheme. Recall (from the section 2.4.3) that under ABS, the picos and femtos transmit simultaneously during the entirety of small cell time π. Consequently, UE SINRs (and rates) from femto F_j^i will include the cross-tier interference from pico P_i and vice versa. Let the rates R_u, S_u, T_u now denote the rates of user $u \in U_j^i$ from macro M, pico P_i and femto F_j^i calculated under the ABS scheme (i.e., by including the extra resulting interference in the rate equation.)

The minimum clearing time LP for this setup can be formulated as LP (2.1-2.4) with $\pi - \epsilon_i$ in (3.3) and ϵ_i in (3.4), replaced by π. Clearly, the framework can be adapted to this joint optimization of ABS and three tier user association. The optimization is simpler to solve since there is only one resource variable π (and no $\{\epsilon_i\}_{i=1}^N$). Hence, the decomposed LP (2.5) obtained by fixing π does not have ϵ_i

as a variable in this optimization. Hence, solution is given by just $\Theta(\pi, \alpha^*)$. The algorithms provided can be adapted and applied in the same manner to solve for $\Theta(\pi, \alpha^*)$.

In what follows in the section, we consider several such optimizations of three tier HetNets. Examples include HetNets involving mmWave femtos and mmWave wireless backhaul, HetNets involving HAPs. All the main results can be extended to these HetNets as we will show. The algorithms can be implemented with slight modifications. For the mmWave cells, we consider single stream MIMO beamforming. The results can be extended to networks with advanced techniques such as Space Division Multiplexing (SDMA), but are beyond the scope of this chapter.

2.5.2 Three tier HetNet with mmWave BSs



(a) mmWave femto cell

(b) mmWave femto cell with mmWave backhaul

Figure 2.8: mmWave Three tier HetNets. (© 2020 IEEE)

Consider a 3 tier HetNet with the femto BSs using mmWave frequencies. Hence, the femtos experience no interference from the macro or the pico BSs, and require no radio resources from these cells. Time only needs to be partitioned between the macro and pico tier to avoid cross-tier interference. We consider a setup where the allocation is performed periodically at the beginning of each frame. The frame length is Δ seconds. As before, let $\pi \leq \Delta$ denote the time allocated to the small cells, which will now be used exclusively by the pico BSs. Therefore, the pico time constraint for P_i will now be $\sum_{u \in U_i} y_u / S_u \leq \pi$.

2.5.2.1 mmWave link rates

We consider a setup where the femtos do not have a wired backhaul link. Pico BS P_i are equipped with special hardware to provide backhaul over a dedicated mmWave link to each F_j^i. The femto BSs employ beam-forming for serving UEs and for backhaul.

Consider a UE $u \in U_j^i$ in the range of femto F_j^i. Let B_m denote the mmWave bandwidth available for serving UEs. Let $P_{F_j^i}$ denote the transmit power of the femto BS, and $g_{F_j^i,u}$ denote the gain of the link between F_j^i and the UE u, including the beam-forming directivity gain of the BS and the UE, path loss and shadowing loss. σ^2 is the noise power. The rate of the link between femto F_j^i and UE u is $T_u = B_m \log_2(1 + P_{F_j^i} g_{F_j^i,u}/(\sum_{k\neq j} P_{F_k^i} g_{F_k^i,u} + \sigma^2))$. Due to the directional nature of the mmWave links, we assume that co-tier interference $\sum_{k\neq j} P_{F_k^i} g_{F_k^i,u} \ll \sigma^2$ as before. For the blocked UEs, we take T_u to be zero; these UEs have to associate with a pico or macro BS. We assume that the rate T_u remains constant for the duration of the frame.

Let S_j^i denote the rate of the backhaul link between the pico P_i and femto F_j^i, calculated similarly as T_u. The backhaul link has to carry all the traffic into the femto F_j^i, i.e., $\sum_{u\in U_j^i} z_u$. And, the femto cannot receive and transmit at the same time due to the half-duplex constraint. Therefore for a femto, the total time Δ has to be partitioned between the backhaul and UE transmissions. The femto time constraint for F_j^i will now be $\sum_{u\in U_j^i} z_u/S_j^i + \sum_{u\in U_j^i} z_u/T_u \leq \Delta$. This is equivalent to $\sum_{u\in U_j^i} z_u/T'_u \leq \Delta$, where

$$T'_u = T_u/(1 + T_u/S_j^i) \tag{2.7}$$

2.5.2.2 Rate requirements

Let a_u (in bits/s) represent the rate requirement (or target) of a UE $u \in U_i$. We assume that the rate requirements are set by a scheduler (based on some fairness criterion or a QOS requirement). The number of bits needed by u in the frame to meet the rate requirement is $a_u\Delta$. Now, the objective of

minimizing the clearing time of the microwave part of HetNet is formulated as LP (2.8).

$$\begin{aligned}\min_{x_u,y_u,z_u,\pi\geq 0}\ & \pi + \sum_{u\in\bigcup_{i=1}^{N} U_i} x_u/R_u \\ \text{s.t.}\ & \sum_{u\in U_i} y_u/S_u \leq \pi, \forall i \in \{1,2,\dots N\} \\ & \sum_{u\in U_j^i} z_u/T'_u \leq \Delta, \forall j \in \{1,2,\dots N_i\}, i \in \{1,2,\dots N\} \\ & x_u + y_u + z_u = a_u\Delta, \forall u \end{aligned} \tag{2.8}$$

Let $\bar{\Delta}$ be the value of LP (2.8). Note that when $\bar{\Delta} > \Delta$, the rate requirements cannot be met. We can take the solution $(\boldsymbol{x}^*, \boldsymbol{y}^*, \pi^*)$ and scale by $\Delta/\bar{\Delta}$ to get a feasible allocation for the current frame. Similarly when $\bar{\Delta} < \Delta$, scaling by $\Delta/\bar{\Delta}$ produces a maximal solution, so that no time is wasted in the frame.

2.5.2.3 Solution and Algorithms

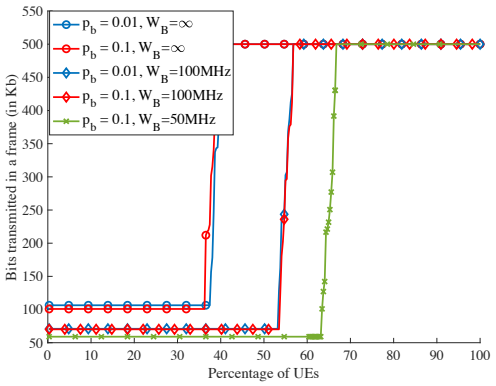
Note that LP (2.8) is simpler to solve than LP (2.1-2.4) since it has one less variable, ϵ_i (we have a constant Δ instead). As before, fixing π, the problem can be decomposed into N independent LPs. The solution of each LP can be found by a 1D search over parameter α at P_i.

To apply allocation algorithms in the chapter, firstly, it can be seen that T_u should be replaced with T'_u. Algorithm 2 and Algorithm 3 can now be applied by modifying the inputs. $\mathcal{F}_j(\alpha^*, \Delta)$ can be implemented as Algorithm 2 to solve for $\boldsymbol{z}^*$ and β_j^* at femto F_j^i. $\mathcal{P}(\alpha^*, 0, \boldsymbol{z}^*, a, b, \delta)$ can be implemented as Algorithm 3 to solve for $\boldsymbol{y}^*$.

2.5.2.4 Numerical Example

Due to the high rates of mmWave BSs, the value of S_j^i has a significant impact on T'_u (unlike the cases where $S_j^i \gg T_u$). To illustrate this effect, we consider the same BS setup as in section 2.4, with the femtos now using mmWave spectrum. 30 UEs are uniformly placed within 50m of each femto BS. The mmWave simulation parameters are given in Figure. 2.9(b). The results can be seen in Figure. 2.9(a). In each scenario, there are two throughput values obtained by the UEs. The high value, 500 Kb/frame corresponds to the UEs with the femtos (using mmWave band). The low values correspond to the UEs associated with the macro or picos (using the microwave band). The UEs which receive partial service from more than one BS receives a throughput value between the two.

UEs at the top, transmitting 500 Kb in a frame are the mmWave UEs, and the UEs at the bottom are the microwave UEs. Since there is not enough bandwidth to support 50 Mbps rate for all the UEs, the microwave UEs get scaled down rates. Here, the case $W_B = \infty$ corresponds to wired backhaul.



(a) Cumulative distribution of throughput

Parameter	Value
Femto BS and backhaul Tx power	22 dBm
Femto BS directivity gain	20 dB
UE directivity gain	10 dB
UE throughput requirement a_u	50 Mbps
mmWave pathloss	3GPP UMi Model
mmWave femto bandwidth	50 MHz
mmWave backhaul bandwidth	W_B
Link blockage probability	p_b
Frame size Δ	10 ms

(b) mmWave simulation parameters

Figure 2.9: Effect of backhaul bandwidth and blocking on mmWave HetNet capacity. (© 2020 IEEE)

From Figure. 2.9(a) [4], it can be observed that blocking does not significantly change the solution of LP (2.8). This is because some UEs need to be offloaded to the microwave BSs anyway, and blocking just affects which ones are offloaded. In this case, adapting the bias-values based on the state of blocking only has a minor impact. The proposed scheme can therefore be implemented on a slower-time scale, with the blocked UEs changing association to microwave BSs using the given bias-values, and the mmWave femto swaps the blocked UE with an unblocked microwave UE (the one with the highest ρ_u^{α}).

Secondly, it can be seen from Figure. 2.9(a) that the backhaul bandwidth has a significant impact on the traffic supported by the mmWave BSs. (2.7) shows that doubling S_j^i does not double T_u'. In Figure. 2.9(a), doubling W_B from 50 MHz to 100 MHz, only increased the number of femto UEs by $\approx$ 10%. We conclude that backhaul bandwidth needs to be accounted for in capacity planning, and there are diminishing returns from increasing it.

[4]The curve for $W_B = 50$ MHz, $p_b = 0.01$ is exactly the same as the green curve and so not depicted in Figure. 2.9(a)

2.5.3 Three tier HetNet under fixed resource partitioning - High Altitude Platforms

Consider a 3 tier HetNet where the BSs have fixed frequency assignments, and resource partitioning is not done in the time domain. This situation can arise when 1) cross-tier and co-tier interference are small enough that ABS scheme does not provide a major advantage, 2) the network operator chooses to do apriori fixed frequency partitioning among the BSs, 3) the BSs belong to different operators and resource sharing is not possible, or 4) There are three tiers of BSs belonging to different technologies, such as HetNet made up of HAP, LTE and mmWave tiers. In the following, we provide a detailed application of the framework by considering scenario 4).

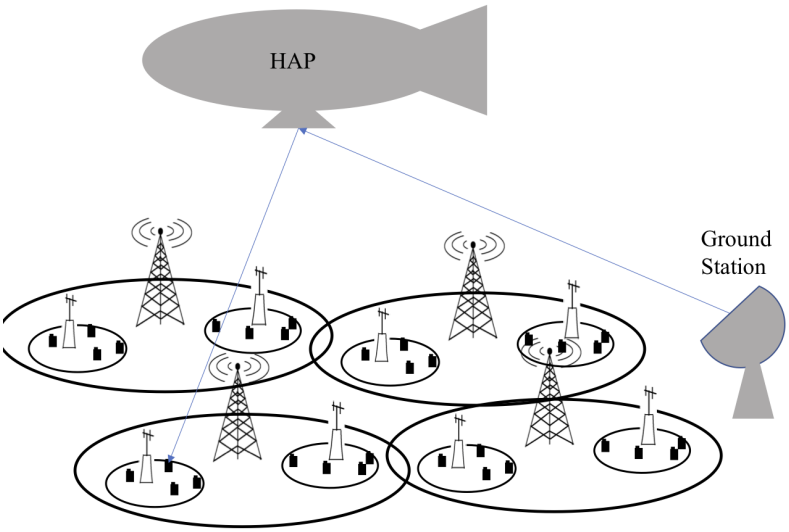


Figure 2.10: Three tier HetNet with HAP, LTE and mmWave tiers. (© 2020 IEEE)

We consider the 3 tier HetNet (shown in Figure. 2.10) with a HAP M at the top. The LTE BSs $\{P_i\}_{i=1}^{N}$ are operating in the coverage area of HAP M. There are N_i mmWave BSs $\{F_j^i\}_{j=1}^{N_i}$ operating in the coverage are of LTE BS P_i. As before, let U_j^i denote the set of UEs that are in the coverage area of F_j^i. A UE $u \in U_j^i$ can associate with and download from any of the BSs in $\{M, P_i, F_j^i\}$. Let $U_i := \bigcup_{j=1}^{N_i} U_j^i$ denote the set of UEs which are covered by P_i. Similarly, $U := \bigcup_{i=1}^{N} U_i$ is the set of all the UEs.

The different tiers HAP, LTE and mmWave are on separate frequency channels. Hence, no muting of BSs to avoid cross-tier interference is required, and the BSs transmit all the time. The rate of the

link between mmWave BS F_j^i and a UE at site u is given as

$$T_u = B\log_2(1 + P_{F_j^i} g_{F_j^i,u}/(\sigma^2 + \sum_{b\in\mathcal{I}_j^i} P_b g_{b,u})) \tag{2.9}$$

where B is the bandwidth available to mmWave BSs. P_b is the transmit power of BS b, $g_{b,u}$ is the gain of the link between b and the UE u, σ^2 is the noise power and $\mathcal{I}_j^i$ is the set of other mmWave BSs in the network which are on the same frequency band as F_j^i. Similarly, R_u (and S_u) is the rate at which a UE at site $u \in U_i^i$ can be served by the HAP M (and LTE BS P_i respectively).

We consider the user association problem using the minimum clearing time formulation as the following LP (2.10)

$$\begin{aligned}
&\min_{x_u,y_u,z_u T_c \geq 0} T_c \\
\text{s.t.} \quad &\sum_{u\in\bigcup_{i=1}^N U_i} x_u/R_u \leq T_c \\
&\sum_{u\in U_i} y_u/S_u \leq T_c, \forall i \in \{1,2,\ldots N\} \\
&\sum_{u\in U_j^i} z_u/T_u \leq T_c, \forall j \in \{1,2,\ldots N_i\}, i \in \{1,2,\ldots N\} \\
&x_u + y_u + z_u = \tau_u, \forall u
\end{aligned} \tag{2.10}$$

where T_c is the clearing time required to satisfy all the UEs. At first sight, LP (2.10) appears quite different to LP (2.1-2.4). However, similar techniques developed for solving LP (2.1-2.4) can be applied to decompose the problem here. Consider LP (2.11) formulated for a fixed value of clearing time $T_c > 0$.

$$\begin{aligned}
&\min_{x_u,y_u,z_u \geq 0} \sum_{u\in\bigcup_{i=1}^N U_i} x_u/R_u \\
&\text{s.t.} \\
&\sum_{u\in U_i} y_u/S_u \leq T_c, \forall i \in \{1,2,\ldots N\} \\
&\sum_{u\in U_j^i} z_u/T_u \leq T_c, \forall j \in \{1,2,\ldots N_i\}, i \in \{1,2,\ldots N\} \\
&x_u + y_u + z_u = \tau_u, \forall u \in U
\end{aligned} \tag{2.11}$$

Let $g(T_c)$ denote the value of LP (2.11) for a given T_c.

Lemma 2.5.1. *Suppose $g(0) > 0$ and let T_c^* be the optimal value of LP (2.10). Then, $g(T_c^*) = T_c^*$.*

Proof. First, we show that $g(T_c^*) \leq T_c^*$. Clearly, the optimal solution $[\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*, T_c^*]$ of LP (2.10), satisfies the constraints:

$$\sum_{u \in U} x_u^*/R_u \leq T_c^*$$
$$\sum_{u \in U_i} y_u^*/S_u \leq T_c^*, \forall i \in \{1, 2, \ldots N\}$$
$$\sum_{u \in U_j^i} z_u^*/T_u \leq T_c^*, \forall j \in \{1, 2, \ldots N_i\}, i \in \{1, 2, \ldots N\}$$
$$x_u^* + y_u^* + z_u^* = \tau_u, \forall u \in U$$

Thus the solution $[\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*, T_c^*]$ of LP (2.10) is a feasible solution of LP (2.11). Under this solution, the objective value of LP (2.11) equals T_c^*. Hence, $g(T_c^*) \leq T_c^*$

Suppose $g(T_c^*) < T_c^*$. Define $h(T_c) = g(T_c) - T_c, \forall T_c \geq 0$. Since $g(.)$ is a monotonically decreasing function, it follows that $h(.)$ is a monotonically decreasing function. Also note that $h(0) = g(0) > 0$ and $h(T_c^*) = g(T_c^*) - T_c^* < 0$. Since $h(.)$ is a continuous and strictly monotonically decreasing function, there must exist a unique $T_c' : 0 < T_c' < T_c^*$ such that $h(T_c') = 0$ (by intermediate value theorem and monotonicity of h(.)). Thus, $g(T_c') = T_c'$.

Let $[\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}']$ denote the optimal solution of LP (2.11) given T_c'. It follows that the solution $[\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}']$ satisfies

$$\sum_{u \in U} x_u'/R_u = T_c'$$
$$\sum_{u \in U_i} y_u'/S_u \leq T_c', \forall i \in \{1, 2, \ldots N\}$$
$$\sum_{u \in U_j^i} z_u'/T_u \leq T_c', \forall j \in \{1, 2, \ldots N_i\}, i \in \{1, 2, \ldots N\}$$
$$x_u' + y_u' + z_u' = \tau_u, \forall u \in U$$

Hence, it follows that the vector $[\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}', T_c']$ is a feasible solution of LP (2.10) with an objective value equal to T_c'. It follows that the optimal value of LP (2.10) $T_c^* \leq T_c'$, which is a contradiction (since $T_c' < T_c^*$). We conclude that $g(T_c^*) = T_c^*$. □

Note that $g(T_c)$ is a monotonically decreasing function on $[0, \infty)$, and T_c is a monotonically increasing function. It also follows that the optimal value T_c^* (from Lemma 2.5.1) occurs at the

intersection of the curves $y = T_c$ and $y = g(T_c)$. If $g(T_c) = T_c$, we have $T_c = T_c^*$. Otherwise, 1) if $g(T_c) > T_c$, we have $T_c < T_c^*$, and 2) if $g(T_c) < T_c$, we have $T_c > T_c^*$. Therefore, the optimal T_c^* can be found using a simple 1D search on T_c.

The only remaining challenge is evaluating $g(T_c)$, which is needed for the 1D search. In the following, we will show that $g(T_c)$ can be evaluated using the three tier framework developed in this chapter. Distributed user association and resource allocation can be performed using the same algorithms with slight modifications.

To evaluate the value of LP (2.11), $g(T_c)$, note that LP (2.11) can also be decomposed into N independent LPs. Consider the LP involving P_i and $\{F_j^i\}_{j=1}^{N_i}$ as follows

$$\min_{x_u, y_u, z_u \geq 0} \sum_{u \in U_i} x_u / R_u$$

$$\alpha \text{ constraint :} \quad \text{s.t.} \sum_{u \in U_i} y_u / S_u \leq T_c$$

$$\beta_j \text{ constraint :} \quad \sum_{u \in U_j^i} z_u / T_u \leq T_c, \ \forall j \in \{1, 2, \ldots N_i\}$$

$$\gamma_u \text{ constraint :} \quad x_u + y_u + z_u = D_u, \ \forall u \in U_i$$

i.e., it is LP (2.5) with the term $\pi - \epsilon_i$ replaced by T_c and the term ϵ_i replaced by T_c.

It is straightforward that the pico and femto allocation algorithms can be modified and applied here to find the solution of LP (2.11). The pico allocation algorithm in Algorithm 3, as $\mathcal{P}(\alpha, T_c, \boldsymbol{z}, a, b, \delta)$, can be applied at the LTE BS P_i. The femto allocation algorithm in Algorithm 2, as $\mathcal{F}_j(\alpha, T_c)$, can be applied at the mmWave BS F_j^i. Hence, the distributed scheme shown in Figure. 2.4 can be implemented to control the user association in the network.

2.5.4 Optimal SINR bias scheme in a three tier HetNet

The resource allocation and user association scheme resulting from the optimization LP (2.1-2.4) provides the optimal rate-bias scheme for three tier HetNets. However, the insights derived here can be used to adapt SINR-bias scheme for three tier HetNet.

For such an implementation, the ρ_u^α in Algorithm 2 (Femto allocation) should be defined based on SINR values in absolute scale (instead of rates), e.g., $\rho_u^\alpha = \min\{\alpha \Lambda_u^F / \Lambda_u^P, \Lambda_u^F / \Lambda_u^M\}$. Here Λ_u^M (Λ_u^P and Λ_u^F) is the SINR value of the signal from macro (pico and femto resp.) to UE u. Similarly, the ordering and ratios in Algorithm 3 (Pico allocation) should be done based on the SINR-ratio $\Lambda_u^P / \Lambda_u^M$

instead of the rate-ratio S_u/R_u. It can be easily verified that the resulting user association must satisfy the SINR-bias rule.

However, the inner and outer search algorithms provided in the previous section 2.4 cannot be used to find the optimal π^* and ϵ_i^*. This is because the clearing time function (i.e, the minimum clearing time for a given π, ϵ_i) is not guaranteed to be convex under the SINR bias rule. Hence, exhaustive search may be needed to find the optimal π^*, ϵ_i^*. However, in case of fixed resource partitioning, the pico and femto algorithms can be applied directly with the mentioned changes to thresholds and sorting procedures.

2.6 Relation to Capacity and Dynamic Model

In this section, we introduce a dynamic HetNet model with stochastic UE arrivals and file requests. We consider a packet queueing model and provide the criterion for stability of the model. The system is deemed to be stable if the queue lengths do not blow up to infinity. We define the capacity region to be the set of arrival rates for which it is possible to stabilize the system. We will show that the minimum clearing time LP provides a capacity characterization for the network.

2.6.1 System Model

Consider a dynamic model of the three tier HetNet, with the BS setup described in section 2.2. There are N pico BSs labelled as $\{P_i\}_{i=1}^N$, operating in the coverage area of the macro BS M. There are N_i femto BSs operating in the coverage area of a pico P_i, labelled as $\{F_j^i\}_{j=1}^{N_i}$.

We consider a discrete model for the UE locations in the network. The UEs arrive and request a file at one of the *user sites* (discrete locations) in the network. They depart from the network once the file is downloaded. The area covered by the network is divided into user sites as follows. U_j^i is the set of user sites which are covered by the femto F_j^i. The UEs at a site $u \in U_j^i$ can associate and download from M, P_i and F_j^i. Let $U_i := \bigcup_{j=1}^{N_i} U_j^i$ denote the set of user sites that are covered by P_i[5]. Similarly, let $U := \bigcup_{i=1}^{N} U_i$ denote the set of all user sites. The UEs arrive at a site $u \in U_j^i$ as a stochastic process as follows. We consider a slotted model, and $t \in \mathbb{N}$ denotes the slot. We assume that the number of

[5]Similar to section 2.2, we do not explicitly model the user sites that have no femto connectivity and only have coverage from a pico P_i and macro M. This is done for the sake of a cleaner treatment. However, the justification provided in footnote 1 also applies here.

arrivals in a slot (at a site u) is independently and identically distributed (i.i.d) as a poisson random variable with mean λ_u. Upon arrival at u, a UE requests a file download of size D_u packets. We assume that the requested file sizes at a site u are i.i.d with a mean $\bar{D}_u$ packets. Let $A_u(t)$ denote the number of packet arrivals at a site u in slot t. Further, we assume that the arrival processes $\{A_u(t)\}_u$ are independent across the user sites $u \in U$.

As in section 2.2, no two BSs in the set $\{M, P_i, F_j^i\}$ are allowed to transmit simultaneously due to the resulting cross-tier interference at receiving UEs. Similar to section 2.2, let R_u (in packets/slot) denote the rate of the link between macro M and UEs at site u, provided P_i and F_j^i are silent. Similarly, let S_u and T_u denote the pico and femto rates for UEs at site u. Here, $[R_u, S_u, T_u] \in \mathbb{Z}_+^3$, i.e., the rates (in packets/slot) are positive integers. (An alternate interpretation of user site u can be as a class of UEs which have the downlink rates given by the rate triplet $[R_u, S_u, T_u]$). A UE departs from the network when it downloads the file completely. As before, we allow for multi-connectivity and a UE can be served by any BS that is in its range, possibly download a part of the file from each available BS.

Let $Q_u(t) \in \mathbb{Z}_+$ denote the number of packets at site u at the beginning of slot t. We denote the state of the system at slot t as $\boldsymbol{Q}(t) = [Q_u(t)]_{u \in U}$. In a slot, we assume that a BS can serve at most one site u. At time t, a scheduling policy can allocate the slot to a set of non-interfering BS-UE pairs, such that the cross-tier interference constraint is not violated, i.e., no two BSs in $\{M, P_i, F_j^i\}$ can be active in the same slot for any $i \in \{1, 2, \ldots N\}, j \in \{1, 2, \ldots N_i\}$. Let $\boldsymbol{s}(t) = [s_u^{(1)}(t), s_u^{(2)}(t), s_u^{(3)}(t)]_{u \in U} \in \{0, 1\}^{3|U|}$ denote the schedule in slot t. For a user site $u \in U_j^i$,

$$s_u^{(1)}(t) = \begin{cases} 1 & \text{if macro } M \text{ is scheduled at site } u \text{ in slot } t. \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

$$s_u^{(2)}(t) = \begin{cases} 1 & \text{if macro } P_i \text{ is scheduled at site } u \text{ in slot } t. \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

$$s_u^{(3)}(t) = \begin{cases} 1 & \text{if macro } F_j^i \text{ is scheduled at site } u \text{ in slot } t. \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

To ensure feasibility i.e., such that the constraints hold, the scheduling policy must satisfy (2.15).

$$\sum_{u\in U} s_u^{(1)}(t) + \sum_{u\in U_i} s_u^{(2)}(t) + \sum_{u\in U_j^i} s_u^{(3)}(t) \leq 1, \forall i \in \{1,\ldots,N\}, j \in \{1,\ldots,N_i\} \tag{2.15}$$

Let $\mathcal{S} \in \{0,1\}^{3|U|}$ denote the set of all feasible schedules $\boldsymbol{s}$ such that the constraints in (2.15) hold. We consider *stationary* scheduling policies which chooses $\boldsymbol{s}(t)$ only based on the state of the system $\boldsymbol{Q}(t)$.

Definition 2.6.1 (Deterministic stationary scheduling policy)**.** *A deterministic scheduling policy $\theta : \mathbb{Z}_+^{|U|} \rightarrow \mathcal{S}$ is a mapping from the state $Q \in \mathcal{Q}$ to a feasible schedule $\boldsymbol{s} \in \mathcal{S}$.*

Definition 2.6.2 (Randomized stationary scheduling policy)**.** *Under a randomized scheduling policy, given the state $Q \in \mathcal{Q}$, the schedule is the output of a random variable $\mathcal{X}$ with a probability distribution $\mathcal{P}_Q$ on $\mathcal{S}$. The distribution $\mathcal{P}_Q$ depends only on the state Q. In each slot t, the choice of schedule is made independently.*

The queue evolution equation at site $u \in U_j^i$ is given as follows

$$Q_u(t+1) = \left(Q_u(t) + A_u(t) - R_u s_u^{(1)}(t) - S_u s_u^{(2)}(t) - T_u s_u^{(3)}(t)\right)^+ \tag{2.16}$$

where $(a)^+ = \max\{0, a\}$

Note that since $[R_u, S_u, T_u] \in \mathbb{Z}_+^3$, the state $\boldsymbol{Q}(t) \in \mathbb{Z}_+^{3|U|}$. Given the schedule $\boldsymbol{s}(t)$ and arrivals $\{A_u(t)\}_{u\in U}$ in slot t, the state $\boldsymbol{Q}(t+1)$ in slot $t+1$ is fully determined. It follows from the assumptions on arrival processes, that the state process $\{\boldsymbol{Q}(t)\}_{t=0}^{\infty}$ is a Markov chain under a stationary scheduling policy.

Definition 2.6.3. *The system is stable under a scheduling policy if and only if*

$$\lim_{T\rightarrow\infty} \sum_{t=0}^{T-1} \sum_{u\in U} E[Q_u(t)]/T < \infty \tag{2.17}$$

Definition 2.6.4. *The system is stabilizable if and only if there exists a resource allocation policy θ under which the system is stable.*

Let $\boldsymbol{\lambda} := [\lambda_u]_{u\in U}$ denote the arrival rate vector. Let Λ denote the set of all arrival vectors for which the system is stabilizable. In the following section, we will provide a minimum clearing LP formulation for a given $\boldsymbol{\lambda} \in {\mathbb{R}_+}^{|U|}$ as LP (2.18). We will show that whenever the value of LP (2.18)

is less than 1, $\boldsymbol{\lambda}$ is interior to Λ. For this case, we will also provide a randomized scheduling policy which will stabilize the system. We will also show that whenever the value of LP (2.18) is greater than 1, $\boldsymbol{\lambda}$ is exterior of Λ. Thus, we provide characterize the stability region Λ using minimum clearing LP (2.18).

2.6.2 Stationary randomized scheduling policy and LP formulation

Consider an equivalent formulation of LP (2.1-2.4) as the following LP (2.18).

$$\begin{aligned}
\min_{x_u, y_y, z_u, \pi, \epsilon_i \geq 0} \ & \pi + \sum_{u \in U} x_u / R_u \\
\text{s.t.} \ & \sum_{u \in U_i} y_u / S_u \leq \pi - \epsilon_i, \ \forall i \in \{1, 2, \ldots N\} \\
& \sum_{u \in U_j^i} z_u / T_u \leq \epsilon_i, \ \forall j \in \{1, 2, \ldots N_i\}, i \in \{1, 2, \ldots N\} \\
& x_u + y_u + z_u = \lambda_u \bar{D}_u, \ \forall u \in U
\end{aligned} \tag{2.18}$$

The variables in LP (2.18) have different units compared to the ones in LP (2.1-2.4). Here, λ_u is the mean arrival rate (in users/slot), $\bar{D}_u$ is the mean file request size (in packets/user). R_u, S_u, T_u are the macro, pico and femto rates (in packets/slot). Hence, x_u, y_u, z_u are in packets/slot. And, $\pi, \epsilon_i, x_u/R_u, y_u/S_u, z_u/T_u$ are unit-less quantities (which will be interpreted as probabilities in the following).

Let $[\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*]$ denote the optimal solution of LP (2.18) for a given $\boldsymbol{\lambda}$. Suppose that the optimal value $\pi^* + \sum_{u \in U} x_u^*/R_u < 1$. We will now propose a stabilizing randomized policy corresponding to the optimal solution. Consider the scaled feasible solution of LP (2.18) derived as follows.

$$\begin{aligned}
[x_u, y_u, z_u]_{u \in U} &= \frac{[x_u^*, y_u^*, z_u^*]_{u \in U}}{\pi^* + \sum_{u \in U} x_u^*/R_u} \\
\pi &= \frac{\pi^*}{\pi^* + \sum_{u \in U} x_u^*/R_u} \\
\epsilon_i &= \frac{\epsilon_i^*}{\pi^* + \sum_{u \in U} x_u^*/R_u}, i = 1, \ldots, N
\end{aligned}$$

From construction, the scaled feasible solution satisfies $\pi + \sum_{u \in U} x_u/R_u = 1$. A randomized scheduling policy can be constructed using this solution as follows. During a slot t, the macro is scheduled with probability $1 - \pi = \sum_{u \in U} x_u/R_u$. Suppose the macro is not scheduled in the slot, then

for each $i \in \{1, \dots N\}$, the pico P_i is scheduled with probability $(\pi - \epsilon_i)/\pi$ (independently across the picos). If both macro and pico P_i are not scheduled in slot t, then the femtos F_j^i are scheduled.

Given that the macro is scheduled in slot t, the random policy chooses site $u \in U$ with probability $(x_u/R_u)/(1-\pi)$, i.e., $\mathbb{P}[s_u^{(1)}(t) = 1|\text{macro is scheduled is slot } t] = (x_u/R_u)/(1-\pi)$. This is feasible since $\sum_{u\in U} x_u/R_u = 1 - \pi$ (because $[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \pi, \{\epsilon_i\}_{i=1}^N]$ is a feasible solution of LP (2.18) from construction).

Similarly, given that pico P_i is scheduled in slot t, the random policy chooses site $u \in U_i$ with probability $(y_u/S_u)/(\pi - \epsilon_i)$, i.e., $\mathbb{P}[s_u^{(2)}(t) = 1|\text{pico } P_i \text{ is scheduled is slot } t] = (y_u/S_u)/(\pi - \epsilon_i)$. This is feasible since $\sum_{u\in U} y_u/S_u \leq \pi - \epsilon_i$ (because $[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \pi, \{\epsilon_i\}_{i=1}^N]$ is a feasible solution of LP (2.18) from construction).

Similarly, given that femto F_j^i is scheduled in slot t, the random policy chooses site $u \in U_j^i$ with probability $(z_u/T_u)/\epsilon_i$, ie., $\mathbb{P}[s_u^{(3)}(t) = 1|\text{femto } F_j^i \text{ is scheduled is slot } t] = (z_u/T_u)/\epsilon_i$. This is feasible since $\sum_{u\in U} z_u/T_u \leq \epsilon_i$ (because $[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \pi, \{\epsilon_i\}_{i=1}^N]$ is a feasible solution of LP (2.18) from construction).

It can be noted that under the proposed randomized policy, probability that the queue Q_u at $u \in U_j^i$ gets served by macro M equals x_u/R_u. Similarly, probability that Q_u at $u \in U_j^i$ gets served by pico P_i (and femto F_j^i) equals y_u/S_u (and z_u/T_u respectively). Since the three events are mutually exclusive, it follows that in each slot t, the queue Q_u at $u \in U_j^i$ gets served at an expected rate of $x_u + y_u + z_u$ packets per slot. From construction, we have $x_u > x_u^*$, $y_u > y_u^*$, $z_u > z_u^*$, $\forall u \in U$. Hence, $x_u + y_u + z_u > \lambda_u \bar{D}_u$.

Therefore, the expected rate of service $x_u + y_u + z_u$ at each site u is greater than the arrival rate of packets $\lambda_u \bar{D}_u$. In Theorem 2.6.1, we will show that this condition implies that the system is stable under the proposed randomized policy.

Theorem 2.6.1. *Let $\pi^* + \sum_{u\in\bigcup_{i=1}^N} x_u^*/R_u$ denote the optimal solution of LP (2.18). Then,*
1) The system is stabilizable if $\pi^* + \sum_{u\in U} x_u^*/R_u < 1$
2) The system is not stabilizable if $\pi^* + \sum_{u\in U} x_u^*/R_u > 1$.

Proof. 1) We will show the stability of the proposed randomized policy. Recall that the under the randomized policy, the expected rate of service $x_u + y_u + z_u > \lambda_u \bar{D}_u$. It follows that

$$E[Q_u(t+1) - Q_u(t)|\boldsymbol{Q}(t) : Q_u(t) \geq \max\{R_u, S_u, T_u\}] \tag{2.19}$$

$$= E[A_u(t)] - R_u\mathbb{P}[s_u^{(1)}(t) = 1] - S_u\mathbb{P}[s_u^{(2)}(t) = 1] - T_u\mathbb{P}[s_u^{(3)}(t) = 1] \tag{2.20}$$

$$= \lambda_u \bar{D}_u - (x_u + y_u + z_u) < 0 \tag{2.21}$$

Define

$$-k_1 = \max_{u \in U} \lambda_u \bar{D}_u - (x_u + y_u + z_u) \tag{2.22}$$

Let the function $V(\boldsymbol{Q}(t)) := \sum_{u \in U} Q_u^2(t)$, and consider the drift

$$E[V(\boldsymbol{Q}(t+1)) - V(\boldsymbol{Q}(t))|\boldsymbol{Q}(t)] = \sum_{u \in U} E[(Q_u(t+1) - Q_u(t))^2 \,|\boldsymbol{Q}(t)] + 2 \sum_{u \in U} Q_u(t) E[Q_u(t+1) - Q_u(t)|\boldsymbol{Q}(t)] \tag{2.23}$$

Note that $E[(Q_u(t+1) - Q_u(t))^2 \,|\boldsymbol{Q}(t)] \leq E[A_u^2(t)] + (\max\{R_u, S_u, T_u\})^2$. Since the number of UE arrivals in a slot is a poisson random variable, and file size is bounded by $\bar{D}_u$, we have $E[A_u^2(t)] < \infty$. Hence, there exists $C > 0$ such that

$$E[V(\boldsymbol{Q}(t+1)) - V(\boldsymbol{Q}(t))|\boldsymbol{Q}(t)] \leq C + 2 \sum_{u \in U} Q_u(t) E[Q_u(t+1) - Q_u(t)|\boldsymbol{Q}(t)] \tag{2.24}$$

Consider the term $Q_u(t)E[Q_u(t+1) - Q_u(t)|\boldsymbol{Q}(t)]$. Let $B_u = \max\{R_u, S_u, T_u\}$. It follows from (2.21) and (2.22) that

$$Q_u(t) E[Q_u(t+1) - Q_u(t)|\boldsymbol{Q}(t) : Q_u(t) \geq B_u] \leq -k_1 Q_u(t) \tag{2.25}$$

$$< \lambda_u \bar{D}_u B_u + k_1 B_u - k_1 Q_u(t) \tag{2.26}$$

Note that $E[Q_u(t+1) - Q_u(t)|\boldsymbol{Q}(t)] \leq E[A_u(t)] = \lambda_u \bar{D}_u$. It follows that

$$Q_u(t) E[Q_u(t+1) - Q_u(t)|\boldsymbol{Q}(t) : Q_u(t) < B_u] \leq \lambda_u \bar{D}_u B_u \tag{2.27}$$

$$< \lambda_u \bar{D}_u B_u + k_1 B_u - k_1 Q_u(t) \tag{2.28}$$

Hence, it follows from (2.26) and (2.28) that $Q_u(t)E[Q_u(t+1) - Q_u(t)|\boldsymbol{Q}(t)] < \lambda_u \bar{D}_u B_u + k_1 B_u - k_1 Q_u(t)$. Substituting this in (2.24), it follows that,

$$E[V(\boldsymbol{Q}(t+1)) - V(\boldsymbol{Q}(t))|\boldsymbol{Q}(t)] \leq C + 2 \sum_{u \in U} Q_u(t) E[Q_u(t+1) - Q_u(t)|\boldsymbol{Q}(t)] \tag{2.29}$$

$$\leq C_1 - 2k_1 \sum_{u \in U} Q_u(t) \tag{2.30}$$

where $C_1 = C + 2\lambda_u \bar{D}_u B_u + 2k_1 B_u$.

Taking summation from $t = 0$ to $T-1$, we have $E[V(\boldsymbol{Q}(T)) - V(\boldsymbol{Q}(0))] \leq TC_1 - 2k_1 \sum_{t=0}^{T-1} \sum_{u \in U} E[Q_u(t)]$. Therefore, $\sum_{t=0}^{T-1} \sum_{u \in U} E[Q_u(t)]/T \leq C_1/2k_1 + E[V(\boldsymbol{Q}(0))]/2Tk_1$. Taking limit $T \to \infty$, we have

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{u \in U} E[Q_u(t)] \leq C_1/2k_1 \tag{2.31}$$

Hence, 1) is proved.

For 2), consider an arbitrary scheduling policy. Let $X_u(t)$ denote the total number of packets served by the macro M at site $u \in U_j^i$ until slot t. Similarly, let $Y_u(t)$ (and $Z_u(t)$) denote the total number of packets served by the pico P_i (and the femto F_j^i resp.) at site $u \in U_j^i$ until slot t.

Consider the macro M, a pico P_i and a femto F_j^i. No two of these BSs can simultaneously transmit during any slot. Hence, the $X_u(t)$ packets of macro, $Y_u(t)$ packets of pico and $Z_u(t)$ packets of femto must have been scheduled in separate slots in $\{1, \ldots, t\}$. Therefore,

$$\sum_{u \in U} X_u(t)/R_u + \sum_{u \in U_i} Y_u(t)/S_u + \sum_{u \in U_j^i} Z_u(t)/T_u \leq t, \forall i \in \{1, \ldots N\}, j \in \{1, \ldots N_i\} \tag{2.32}$$

$$\implies \sum_{u \in U} x_u(t)/R_u + \sum_{u \in U_i} y_u(t)/S_u + \sum_{u \in U_j^i} z_u(t)/T_u \leq 1, \forall i \in \{1, \ldots N\}, j \in \{1, \ldots N_i\} \tag{2.33}$$

where $x_u(t) := X_u(t)/t, y_u(t) := Y_u(t)/t$ and $z_u(t) := Z_u(t)/t$. Note that the inequalities (2.33) can be written as constraints and the objective of LP (2.18) by defining $\epsilon_i(t)$ and $\pi(t)$ as follows

$$\epsilon_i(t) := \max_{j=1}^{N_i} \sum_{u \in U_j^i} z_u(t)/T_u, \forall i \in \{1, \ldots N\} \tag{2.34}$$

$$\pi(t) := \sum_{u \in U_i} y_u(t)/S_u + \max_{i=1}^{N} \epsilon_i(t) \tag{2.35}$$

It follows from (2.33, 2.34, 2.35) that $[x_u(t), y_u(t), z_u(t)]_{u \in U}, \epsilon_i(t), \pi(t)$ satisfies

$$\begin{aligned} \pi(t) + \sum_{u \in U} x_u(t)/R_u &\leq 1 \\ \sum_{u \in U_i} y_u(t)/S_u &\leq \pi(t) - \epsilon_i(t),\ i \in \{1, \ldots, N\} \\ \sum_{u \in U_j^i} z_u(t)/T_u &\leq \epsilon_i(t),\ j \in \{1, \ldots, N_i\}, i \in \{1, \ldots, N\} \end{aligned}$$

Now consider $x_u(t) + y_u(t) + z_u(t)$. Suppose $x_u(t) + y_u(t) + z_u(t) \geq \lambda_u \bar{D}_u, \forall u \in U$. Let $\phi_u := \frac{\lambda_u \bar{D}_u}{x_u(t)+y_u(t)+z_u(t)} \leq 1, \forall u \in U$, which implies $\phi_u x_u(t) + \phi_u y_u(t) + \phi_u z_u(t) = \lambda_u \bar{D}_u, \forall u \in U$. Note that $[\phi_u x_u(t), \phi_u y_u(t), \phi_u z_u(t)]_{u \in U}, \epsilon_i(t), \pi(t)$ forms a feasible solution of LP (2.18) such that $\pi(t) + \sum_{u \in U} \phi_u x_u(t)/R_u \leq 1$. This is a contradiction since it is given that the optimal value of LP (2.18) is greater than 1.

Hence, for any t, $\exists u \in U$ such that $x_u(t) + y_u(t) + z_u(t) < \lambda_u \bar{D}_u$. Let $\mathcal{H}$ denote the set of all the

solutions $([x, y, z]_{u\in U}, \pi, \epsilon_i)$ which satisfy (2.36-2.38).

$$\pi + \sum_{u\in U} x_u/R_u \leq 1 \tag{2.36}$$

$$\sum_{u\in U_i} y_u/S_u \leq \pi - \epsilon_i, \forall i = 1, \ldots, N \tag{2.37}$$

$$\sum_{u\in U_j^i} z_u/T_u \leq \epsilon_i, \forall j \in \{1, \ldots N_i\}, i \in \{1, \ldots, N\} \tag{2.38}$$

Since it is given that the optimal value of LP (2.18) is greater than 1, it follows that for each solution in $\mathcal{H}$, $\exists u \in U$ such that $x_u + y_u + z_u < \lambda_u \bar{D}_u$. Define

$$-k := \sup_{\mathcal{H}_s} \inf_{u\in U} x_u + y_u + z_u - \lambda_u \bar{D}_u \tag{2.39}$$

Note that $[x_u(t), y_u(t), z_u(t)]_{u\in U}$ is a member of $\mathcal{H}$ for any sample path on arrival and file size processes. It follows that $[E[x_u(t)], E[y_u(t)], E[z_u(t)]]_{u\in U}$ is also a member of $\mathcal{H}$. Hence, there exists $u \in U$ such that $t\left(\lambda_u \bar{D}_u - E[x_u(t) + y_u(t) + z_u(t)]\right) > kt$. Note that $Q_u(t) = Q_u(0) + A_u(t) - t(x_u(t) + y_u(t) + z_u(t))$. Hence, $\exists u \in U$ such that

$$E[Q_u(t)] = E[Q_u(0)] + t(\lambda_u \bar{D}_u - E[x_u(t) + y_u(t) + z_u(t)]) \tag{2.40}$$

$$\geq E[Q_u(0)] + kt \tag{2.41}$$

Hence, $\sum_{u\in U} E[Q_u(t)] \geq kt, \forall t$. It follows that $\sum_{t=0}^{T-1} \sum_{u\in U} E[Q_u(t)] \geq kT(T-1)/2$. Hence,

$$\lim_{T\to\infty} \sum_{t=0}^{T-1} \sum_{u\in U} E[Q_u(t)]/T = \infty \tag{2.42}$$

Since the choice of policy was arbitrary, it follows that system cannot be stable under any policy, which proves 2). □

2.7 Theoretical Results

2.7.1 Proofs of Theorems 2.3.1-2.3.3

We develop the necessary theory in Appendices 2.7.2-2.7.7. The results will be used in the proofs here. The layout of other Appendices is as follows. In Appendix 2.7.2, we introduce the Lagrangian and the dual-variables corresponding to LP (2.5) and derive structural results of the optimal solution

using the KKT conditions. In Appendix 2.7.5, we present the femto allocation results using the KKT conditions. We justify Algorithm 2. In Appendix 2.7.6, we present the pico allocation results and justify Algortithm 3. The results of Appendices 2.7.5-2.7.6 were derived under the assumption that the dual variable $\alpha > 0$. In Appendix 2.7.7, we deal with the case when dual variable $\alpha = 0$.

Proof of Theorem 2.3.1. Suppose the dual-variables $\alpha, \beta_j > 0, \forall j \in \{1, \ldots N_i\}$ for $\epsilon_i = \epsilon_i^*$. The proof of (1-3) of Theorem 2.3.1 follows from (2.43-2.46) of Appendix 2.7.2. The results of Figure. 2.3 follow from Lemma 2.7.1 and Lemma 2.7.2 in Appendix 2.7.2.

Suppose one or more of the dual-variables α, β_j are zero for $\epsilon_i = \epsilon_i^*$. It follows from Appendix 2.7.7, that there exists α_m and β_j^m corresponding to the zero dual-variables such that (1-3) of Theorem 2.3.1 hold. Now, the results of Figure. 2.3 follow from Lemma 2.7.1 and Lemma 2.7.2 with α (and β_j) replaced with α_m (and β_j^m resp.). □

Proof of Theorem 2.3.2. Suppose the dual variable $\alpha > 0$ for $\epsilon_i = \epsilon_i^*$. If Pico allocation case 1 (in Appendix 2.7.6) holds, then $\alpha = S_{w_l}/R_{w_l}$ is an optimal dual-variable from Lemma 2.7.7. If Pico allocation case 2 (in Appendix 2.7.6) holds, then $\alpha = S_a T_b / T_a R_b$ from Lemma 2.7.8.

Suppose the dual variable $\alpha = 0$ for $\epsilon_i = \epsilon_i^*$. It follows from Appendix 2.7.7 that $\exists \alpha_m \in A$ such that $\theta(\alpha_m, \epsilon_i) = 0$. Here, $\alpha^* = \alpha_m$.

Similarly, suppose the dual-variable $\beta_j > 0$ for $\epsilon_i = \epsilon_i^*$. Then $\beta_j^* = \rho_u^\alpha$ from Appendix 2.7.5. Otherwise if dual-variable $\beta_j = 0$, $\beta_j^m = \min_{u \in U_j^i} \rho_u^\alpha$ is the rate-bias multiplier β_j^* from Appendix 2.7.7. □

Proof of Theorem 2.3.3. Suppose the dual variable $\alpha > 0$ for $\epsilon_i = \epsilon_i^*$. It follows from Lemmas 2.7.4-2.7.6 that Algorithm 2 determines $\boldsymbol{z}^*$ with α, ϵ_i^* as input (See Appendix 2.7.5). It follows from Lemmas 2.7.7-2.7.8 that Algorithm 3 determines $\boldsymbol{y}^*$. (See Appendix 2.7.6)

If $\alpha = 0$, the proof follows from Appendix 2.7.7. □

2.7.2 KKT conditions and Lagrangian minimization

For the given π, we start by fixing $\epsilon_i \in [0, \pi]$. We consider LP (2.5) for the given π, ϵ_i[6].

[6]Let $g(\pi, \epsilon_i)$ denote the solution of LP (2.5) for the given pair (π, ϵ_i). Note that $f_i(\pi) = \min_{\epsilon_i \in [0,\pi]} g(\pi, \epsilon_i)$, and $\epsilon_i^* = \arg\min_{\epsilon_i \in [0,\pi]} g(\pi, \epsilon_i)$.

Consider the Lagrangian L of LP (2.5), given as

$$L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \sum_{u \in U_i} x_u / R_u + \alpha \Big(\sum_{u \in U_i} y_u / S_u + \epsilon_i - \pi \Big)$$
$$+ \sum_{j=1}^{N_i} \beta_j \Big(\sum_{u \in U_j^i} z_u / T_u - \epsilon_i \Big) - \sum_{u \in U_i} \gamma_u (x_u + y_u + z_u - D_u)$$

where α, β_j and γ_u are the dual variables corresponding to the constraints of LP (2.5) (see page 20).

For a fixed (π, ϵ_i), LP (2.5) is equivalent to the Lagrangian minimization problem $\min_{x_u, y_u, z_u \geq 0} L$ with the optimal dual-variables. The KKT conditions provide sufficient conditions for optimality of the primal and dual variables.

2.7.3 Stationarity conditions

From the first order stationarity conditions of the KKT conditions, we must have

$$\partial L / \partial x_u = 1/R_u - \gamma_u \geq 0 \tag{2.43}$$

and $\gamma_u = 1/R_u$ if $x_u > 0$. i.e., minimum occurs either at a stationary point or at a point on the boundary. Similarly, we have

$$\partial L / \partial y_u = \alpha / S_u - \gamma_u \geq 0 \text{ and } \gamma_u = \alpha / S_u \text{ if } y_u > 0 \tag{2.44}$$

$$\partial L / \partial z_u = \beta_j / T_u - \gamma_u \geq 0 \text{ and } \gamma_u = \beta_j / T_u \text{ if } z_u > 0 \tag{2.45}$$

Using (2.43-2.45), $\gamma_u \leq \min\{1/R_u, \alpha/S_u, \beta_j/T_u\}$. Since $x_u + y_u + z_u = D_u > 0$, at least one of $x_u, y_u, z_u > 0$ and hence, at least one of the equality conditions of (2.43-2.45) must hold. Therefore,

$$\gamma_u = \min\{1/R_u, \alpha/S_u, \beta_j/T_u\} \geq 0 \tag{2.46}$$

Going forward, we take x_u, y_u, z_u to be the solution of LP (2.5) for the given fixed π, ϵ_i, and $\alpha, \beta_j, \gamma_u$ to be the optimal dual variables, i.e., KKT conditions hold for these values.

Assumption 1. *For any $u, v \in U_i$ and $u \neq v$, we assume that 1) $T_u/S_u \neq T_v/S_v$, 2) $S_u/R_u \neq S_v/R_v$ and 3) $T_u/R_u \neq T_v/R_v$. Furthermore, for any $(u_1, v_1) \neq (u_2, v_2) \in U_i \times U_i$, we assume that $S_{u_1} T_{v_1} / R_{v_1} T_{u_1} \neq S_{u_1} T_{v_1} / R_{v_1} T_{u_1}$.*

Note that the rates R_u, S_u, T_u are arbitrary real values, and Assumption 1 holds with probability 1. For the sake of brevity, we ignore the highly special cases where Assumption 1 does not hold, e.g., a case where two different UEs have exactly the same rate-ratios mentioned in Assumption 1.

2.7.4 Lemmas on relationship between primal and dual variables

Recall that $\rho_u^\alpha := \min\{T_u/R_u, \alpha T_u/S_u\}$ from Theorem 2.3.1. Proofs of the following three lemmas are direct consequences of the stationarity conditions (2.43-2.46).

Lemma 2.7.1. *Suppose the dual variable $\alpha > 0$. Then*
1) $z_u = D_u$, if $\rho_u^\alpha > \beta_j$ and 2) $z_u = 0$, if $\rho_u^\alpha < \beta_j$.

Proof. Suppose $\rho_u^\alpha > \beta_j$. This implies $\beta_j/T_u < \min\{1/R_u, \alpha/S_u\}$. From (2.46), we have $\gamma_u = \beta_j/T_u$ and $\gamma_u < \min\{1/R_u, \alpha/S_u\}$. Now from (2.43-2.44), we have $x_u = 0, y_u = 0$, and hence $z_u = D_u$. Therefore, $\rho_u^\alpha > \beta_j$ implies $z_u = D_u$.

Now suppose $\rho_u^\alpha < \beta_j$. This implies $\beta_j/T_u > \min\{1/R_u, \alpha/S_u\}$. From (2.46), $\beta_j/T_u > \gamma_u$. Now from (2.45), we have $z_u = 0$. Therefore, $\rho_u^\alpha < \beta_j$ implies $z_u = 0$. □

Lemma 2.7.2. *Suppose $D'_u := D_u - z_u > 0$ for some $u \in U_i$. Then*
1) $y_u = D'_u$, if $S_u/R_u > \alpha$ and 2) $y_u = 0$, if $S_u/R_u < \alpha$.

Proof. Suppose $S_u/R_u > \alpha$. This implies $1/R_u > \alpha/S_u$, and hence $\gamma_u < 1/R_u$ from (2.46). We have $x_u = 0$ from (2.43). Therefore, $y_u + z_u = D_u$. This proves 1).

For 2), suppose $S_u/R_u < \alpha$. This implies $1/R_u < \alpha/S_u$, and hence $\gamma_u < \alpha/S_u$ from (2.46). Therefore, $y_u = 0$ from (2.44). □

Lemma 2.7.3. *Consider a user $u \in U_j^i$. 1) If $x_u, y_u > 0$, then $\alpha = S_u/R_u$. 2) If $y_u, z_u > 0$, then $\beta_j = \rho_u^\alpha = \alpha T_u/S_u$ and 3) If $z_u, x_u > 0$, then $\beta_j = \rho_u^\alpha = T_u/R_u$.*

Proof. Suppose $x_u, y_u > 0$. From (2.43-2.45), we have $\gamma_u = 1/R_u$, $\gamma_u = \alpha/S_u$. Therefore, $\alpha = S_u/R_u$. This proves 1).

Suppose $y_u, z_u > 0$. From (2.43-2.45), we have $\gamma_u = \alpha/S_u$, $\gamma_u = \beta_j/T_u$. Therefore, $\beta_j = \alpha T_u/S_u$

3) can be proved using similar arguments. □

2.7.5 Femto Allocation

In this section, we present the femto allocation $[z_u]_{u \in U_j^i}$ for an arbitrary $j \in \{1, \ldots N_i\}$. We will show that Algorithm 2 determines the femto allocation. This is done under the assumption that the dual-variable $\alpha > 0$. The other case $\alpha = 0$ is done in Appendix 2.7.7.

Assume $\alpha > 0$. Recall that $\rho_u^\alpha := \min\{T_u/R_u, \alpha T_u/S_u\}$. Sort users u_k in U_j^i in descending order of ρ^α such that $\rho_{u_1}^\alpha \geq \rho_{u_2}^\alpha \geq \ldots \geq \rho_{u_K}^\alpha$. Here $K = |U_j^i|$. $\rho_{u_{k_1}}^\alpha = \rho_{u_{k_2}}^\alpha$ for at most one pair k_1, k_2 such that $1 \leq k_1 < k_2 \leq K$ (otherwise Assumption 1 is violated). Therefore, exactly one of the following three cases must hold

Case 1 (No split users case): Here (2.47) holds

$$\sum_{k=1}^{K} D_{u_k}/T_{u_k} \leq \epsilon_i \tag{2.47}$$

The femto allocation for this case is given in Lemma 2.7.4, which justifies step 3 of Algorithm 2.

Case 2 (Single split user case): $\exists l \leq K$ such that

$$1) \sum_{k=1}^{l-1} D_{u_k}/T_{u_k} \leq \epsilon_i < \sum_{k=1}^{l} D_{u_k}/T_{u_k} \tag{2.48}$$

$$2) \rho_{u_l}^\alpha \neq \rho_{u_k}^\alpha, \forall k \in \{1, 2, \ldots K\} - \{l\} \tag{2.49}$$

The femto allocation for this case is given in Lemma 2.7.5, which justifies step 4 of Algorithm 2.

Case 3 (Two split users case): $\exists l \leq K - 1$ such that

$$1) \sum_{k=1}^{l-1} D_{u_k}/T_{u_k} \leq \epsilon_i < \sum_{k=1}^{l+1} D_{u_k}/T_{u_k} \tag{2.50}$$

$$2) \rho_{u_l}^\alpha = \rho_{u_{l+1}}^\alpha \tag{2.51}$$

For the two split users, w.l.o.g assume $\rho_{u_l}^\alpha = \alpha T_{u_l}/S_{u_l}$ and $\rho_{u_{l+1}}^\alpha = T_{u_{l+1}}/R_{u_{l+1}}$. Define $a := u_l$, $b := u_{l+1}$ and $\delta := \epsilon_i - \sum_{k=1}^{l-1} D_{u_k}/T_{u_k}$. Here, a is the pico-femto split user and b is the macro-femto split user.

The femto allocation for this case is given in Lemma 2.7.6, which justifies step 5 of Algorithm 2. Note that z_a, z_b are not given by Lemma 2.7.6, and will be given in Lemma 2.7.8.

Lemma 2.7.4. *Suppose $\alpha > 0$ and (2.47) holds. Then $z_{u_k} = D_{u_k}, \forall 1 \leq k \leq K$ and $\beta_j = 0$.*

Proof. We use proof by contradiction to show that $z_{u_k} = D_{u_k}, \forall 1 \leq k \leq K$. Suppose not and assume $z_{u_p} < D_{u_p}$ for some $1 \leq p \leq K$. Since $z_{u_k} \leq D_{u_k}, \forall k \neq p$, we have $\sum_{k=1}^{K} z_{u_k}/T_{u_k} < \sum_{k=1}^{K} D_{u_k}/T_{u_k} \leq \epsilon_i$ from (2.47). Therefore $\beta_j = 0$ from complementary slackness. If $\beta_j = 0$, we have $z_{u_p} = D_{u_p}$ from Lemma 2.7.1, which is a contradiction to the assumption. Hence, $z_{u_k} = D_{u_k}, \forall 1 \leq k \leq K$. Now if the inequality in (2.47) is strict, $\sum_{k=1}^{K} z_{u_k}/T_{u_k} < \epsilon_i$, and $\beta_j = 0$ from complementary slackness. Otherwise, if the equality in (2.47) holds, the KKT conditions hold for any $\beta_j \in [0, \rho_{u_K}^\alpha]$. □

Lemma 2.7.5. *Suppose $\alpha > 0$ and $\exists l \leq K$ such that (2.48),(2.49) hold. Then $\beta_j = \rho^{\alpha}_{u_l}$ and*

$$z_{u_k} = \begin{cases} D_{u_k} & \text{for } 1 \leq k \leq l-1 \\ T_{u_l}(\epsilon_i - \sum_{k'=1}^{l-1} D_{u_{k'}}/T_{u_{k'}}) & \text{for } k = l \\ 0 & \text{for } l+1 \leq k \leq K \end{cases}$$

Proof. 1) Firstly, we show that $z_{u_k} = D_{u_k}, \forall 1 \leq k \leq l-1$ using proof by contradiction.

Suppose not and assume $z_{u_p} < D_{u_p}$ for some $1 \leq p \leq l-1$. Note that $\rho^{\alpha}_p \leq \beta_j$ from Lemma 2.7.1. This implies $\beta_j \geq \rho^{\alpha}_p > \rho^{\alpha}_l$ from (2.49). Therefore, $z_{u_k} = 0, \forall l \leq k \leq K$ from Lemma 2.7.1. This implies $\sum_{k=1}^{K} z_{u_k}/T_{u_k} < \sum_{k=1}^{l-1} D_{u_k}/T_{u_k} \leq \epsilon_i$ from (2.48). Therefore, $\beta_j = 0$ from complementary slackness. Observe that $\beta_j = 0$ implies $z_{u_p} = D_{u_p}$ from Lemma 2.7.1, which is a contradiction to the assumption $z_{u_p} < D_{u_p}$. Hence, $z_{u_k} = D_{u_k}, \forall 1 \leq k \leq l-1$.

2) Now, we show that $z_{u_k} = 0, \forall l+1 \leq k \leq K$ using proof by contradiction.

Suppose not, and assume $z_{u_p} > 0$ for some $l+1 \leq p \leq K$. This implies $\beta_j \leq \rho^{\alpha}_p$. Therefore, $\rho^{\alpha}_{u_l} > \rho^{\alpha}_p \geq \beta_j$ from (2.49). Therefore, $z_{u_k} = D_{u_k}, \forall 1 \leq k \leq l$ from Lemma 2.7.1. This implies $\sum_{k=1}^{K} z_{u_k}/T_{u_k} \geq \sum_{k=1}^{l} D_{u_k}/T_{u_k} > \epsilon_i$ from (2.48). This violates the primal constraint that $\sum_{k=1}^{K} z_{u_k}/T_{u_k} \leq \epsilon_i$, which is a contradiction. Hence, $z_{u_k} = 0, \forall l+1 \leq k \leq K$.

3) We now show that $\beta_j > 0$ and determine z_{u_l}.

Suppose not, and assume $\beta_j = 0$. We have $z_{u_l} = D_{u_l}$ from Lemma 2.7.1, which implies $\sum_{k=1}^{K} z_{u_k}/T_{u_k} > \epsilon_i$ from (2.48). This is a contradiction since it violates the primal constraint that $\sum_{k=1}^{K} z_{u_k}/T_{u_k} \leq \epsilon_i$. Hence, $\beta_j > 0$, which implies $\sum_{k=1}^{K} z_{u_k}/T_{u_k} = \epsilon_i$ from complementary slackness. Substituting the other values, $z_{u_l} = T_{u_l}(\epsilon_i - \sum_{k=1}^{l-1} D_{u_k}/T_{u_k})$

Note that when the left inequality of (2.48) is strict, $0 < z_{u_l} < D_{u_l}$, which implies $\beta_j = \rho^{\alpha}_{u_l}$ from Lemma 2.7.3. Otherwise, if the equality holds in the left inequality of (2.48), the KKT conditions hold for any $\beta_j \in [\rho^{\alpha}_{u_l}, \rho^{\alpha}_{u_{l-1}}]$. □

Lemma 2.7.6. *Suppose $\alpha > 0$ and $\exists l \leq K-1$ such that (2.50),(2.51) hold. Let $a := u_l$, $b := u_{l+1}$ and $\delta := \epsilon_i - \sum_{k=1}^{l-1} D_{u_k}/T_{u_k}$. W.l.o.g, let $\rho^{\alpha}_a = \alpha T_a/S_a$ and $\rho^{\alpha}_b = T_b/R_b$. Then*

$$z_{u_k} = \begin{cases} D_{u_k} & \text{for } 1 \leq k \leq l-1 \\ 0 & \text{for } l+2 \leq k \leq K \end{cases}$$

$z_a/T_a + z_b/T_b = \delta$, $\alpha = S_a T_b/T_a R_b$, and $\beta_j = \rho^{\alpha}_a$.

Proof. It can proved that $z_{u_k} = D_{u_k}, \forall 1 \leq k \leq l-1$ and $z_{u_k} = 0, \forall l+2 \leq k \leq K$ using similar arguments as in the proof of Lemma 2.7.5.

Note that $\rho_a^\alpha = \rho_b^\alpha$ implies $\alpha T_a/S_a = T_b/R_b$. Hence $\alpha = S_a T_b/R_b T_a$.

It can proved that $\beta_j > 0$ using similar arguments as in the proof of Lemma 2.7.5. From complementary slackness, $\sum_{k=1}^{K} z_{u_k}/T_{u_k} = \epsilon_i$. Substituting other z values, we have $z_a/T_a + z_b/T_b = \delta$.

Note that $\delta < D_a/T_a + D_b/T_b$ from the right inequality of (2.50). This implies either $z_a < D_a$ or $z_b < D_b$. Therefore, $\beta_j \geq \rho_a^\alpha = \rho_b^\alpha$ from Lemma 2.7.1. Suppose the left inequality of (2.50) is strict, then $\delta > 0$. This implies at least one of $z_a, z_b > 0$, we have $\beta_j \leq \rho_a^\alpha = \rho_b^\alpha$ from Lemma 2.7.1. Therefore, $\beta_j = \rho_a^\alpha$ when the left inequality of (2.50) is strict. Otherwise, if the equality holds, the KKT conditions hold for any $\beta_j \in [\rho_a^\alpha, \rho_{l-1}^\alpha]$. □

2.7.6 Pico Allocation

We will present the pico allocation y_u for $u \in U_i$ under the assumption $\alpha > 0$. The other case $\alpha = 0$ is done in Appendix 2.7.7. Let $D'_u := D_u - z_u$ denote the residual file after the femto allocation for $u \in U_i - \{a, b\}$. Recall that a, b are the split users from (2.50 -2.51) in Appendix 2.7.5.

Let W denote the set of $u \in U_i - \{a, b\}$ such that $D'_u > 0$, i.e, positive residual file sizes after femto allocation. Sort the users $w_k \in W$ such that $S_{w_1}/R_{w_1} > \ldots > S_{w_{|W|}}/R_{w_|W|}$. We determine pico allocation y_{w_k} for $k \in \{1, \ldots, |W|\}$ as the following two cases.

2.7.6.1 Pico allocation case 1

Suppose conditions (2.50-2.51) do not hold for any $j \in \{1, \ldots N_i\}$, i.e., Case 3 (Two split users case) in Appendix 2.7.5 does not hold for any j. Here, $\{a, b\} = \phi$.

Since $\alpha > 0$, we have $\sum_{u \in U_i} y_u/S_u = \sum_{k=1}^{|W|} y_{w_k}/S_{w_k} = \pi - \epsilon_i$ from complementary slackness. And, $\exists l \leq |W|$ such that

$$\sum_{k=1}^{l-1} D'_{w_k}/S_{w_k} < \pi - \epsilon_i \leq \sum_{k=1}^{l} D'_{w_k}/S_{w_k} \tag{2.52}$$

The following lemma provides the pico allocation for this case and justifies step 5 of Algorithm 3.

Lemma 2.7.7. *Suppose $\alpha > 0$ and $\{a, b\} = \phi$. Also, suppose that (2.52) holds for $l \leq |W|$. Then*

$$y_{w_k} := \begin{cases} D'_{w_k} & \text{for } 1 \leq k \leq l-1 \\ S_{w_l}(\pi - \epsilon_i - \sum_{k'=1}^{l-1} D'_{w_{k'}}/S_{w_{k'}}) & \text{for } k = l \\ 0 & \text{for } l+1 \leq k \leq |W| \end{cases}$$

Moreover, $\alpha = S_{w_l}/R_{w_l}$.

Proof. Firstly, we show that $y_{w_k} = D'_{w_k}, \forall 1 \leq k \leq l-1$, using proof by contradiction.

Suppose not, and assume $y_{w_p} < D'_{w_p}$ for some $1 \leq p \leq l-1$. Note that this implies $S_{w_p}/R_{w_p} \leq \alpha$ from Lemma 2.7.2. Therefore, $S_{w_k}/R_{w_k} < \alpha, \forall p+1 \leq k \leq |W|$. Therefore, $y_{w_k} = 0, \forall l \leq k \leq |W|$. This implies $\sum_{k=1}^{|W|} y_{w_k}/S_{w_k} \leq \sum_{k=1}^{l-1} D'_{w_k}/S_{w_k} < \pi - \epsilon_i$ from (2.52). Therefore, $\alpha = 0$ from complementary slackness, which implies $y_{w_p} = D'_{w_p}$ from Lemma 2.7.2. This is a contradiction to the assumption $y_{w_p} < D'_{w_p}$. Hence, $y_{w_k} = D'_{w_k}, \forall 1 \leq k \leq l-1$.

Now, we show that $y_{w_k} = 0, \forall l+1 \leq k \leq |W|$, using proof by contradiction.

Suppose not, and assume $y_{w_p} > 0$ for some $l+1 \leq p \leq |W|$. Note that this implies $S_{w_p}/R_{w_p} \geq \alpha$ from Lemma 2.7.2. Therefore, $S_{w_k}/R_{w_k} > \alpha, \forall 1 \leq k \leq p-1$. Therefore, $y_{w_k} = D'_{w_k}, \forall 1 \leq k \leq l$ from Lemma 2.7.2, which implies $\sum_{k=1}^{|W|} y_{w_k}/S_{w_k} \geq \sum_{k=1}^{l} D'_{w_k}/S_{w_k} + y_{w_p}/S_{w_p} > \pi - \epsilon_i$ from (2.52). This violates the primal constraint that $\sum_{k=1}^{|W|} y_{w_k}/S_{w_k} \leq \pi - \epsilon_i$, which is a contradiction.

If the right inequality in (2.52) is strict, then $0 < y_{w_l} < D'_{w_l}$ and $x_{w_l} > 0$. Hence, $\alpha = S_{w_l}/R_{w_l}$ from Lemma 2.7.3. Otherwise, if the equality holds in (2.52), the KKT conditions hold for any $\alpha \in [S_{w_{l+1}}/R_{w_{l+1}}, S_{w_l}/R_{w_l}]$. □

2.7.6.2 Pico allocation case 2

Suppose conditions (2.50-2.51) of Case 3 (Two split users) hold for some $j \in \{1, \ldots N_i\}$ (See Appendix 2.7.5). Lemma 2.7.8 provides the pico allocation for this case, and justifies step 7 of Algorithm 3.

Lemma 2.7.8. *Suppose $\alpha > 0$ and conditions (2.50-2.51) of Case 3 (Two split users) hold for some*

$j \in \{1, \ldots N_i\}$ (See Appendix 2.7.5). Then $\alpha = S_a T_b / T_a R_b$ and

$$y_{w_k} = \begin{cases} D'_{w_k} & \text{for } 1 \leq k \leq l \\ 0 & \text{for } l+1 \leq k \leq |W| \end{cases}$$

$$y_a = S_a(\pi - \epsilon_i - \sum_{k=1}^{l} D'_{w_k}/S_{w_k})$$

$y_b = 0$, $z_a = D_a - y_a$ & $z_b = T_b(\delta - z_a/T_a)$. Further, (2.50-2.51) do not hold for any $j' \neq j$.

Proof. Note that $\alpha = S_a T_b / T_a R_b$ from Lemma 2.7.6. Due to Assumption 1, $S_{w_k}/R_{w_k} \neq \alpha, \forall 1 \leq k \leq |W|$. Therefore, $\exists l \leq |W|$ such that $S_{w_k}/R_{w_k} < \alpha, \forall 1 \leq k \leq l$ and $S_{w_k}/R_{w_k} > \alpha, \forall l+1 \leq k \leq |W|$. Therefore, $y_{w_k} = D'_{w_k}, \forall 1 \leq k \leq l$ and $y_{w_k} = 0, \forall l+1 \leq k \leq |W|$ from Lemma 2.7.2.

It remains to determine y_a, z_a, y_b, z_b. Recall from Lemma 2.7.6 that $\rho_b^\alpha = T_b/R_b < \alpha T_b/S_b$, which implies $S_b/R_b < \alpha$. Therefore, $y_b = 0$ from Lemma 2.7.2.

Since $\alpha > 0$, we have $\sum_{u \in U_i} y_u/S_u = \pi - \epsilon_i$ from complementary slackness. Substituting other y values, we get $y_a = S_a(\pi - \epsilon_i - \sum_{k=1}^{l} D'_{w_k}/S_{w_k})$.

For determining z_a, z_b, recall that $\rho_a^\alpha = \alpha T_a/S_a < T_a/R_a$ and hence $1/R_a > \gamma_a$ from (2.46). Therefore, $x_a = 0$ and $z_a = D_a - y_a$ from (2.43). Now z_b can be determined from $z_a/T_a + z_b/T_b = \delta$ (See Lemma 2.7.6 from Appendix 2.7.5).

Lastly, we prove that (2.50-2.51) do not hold for any $j' \neq j$. Suppose not and assume (2.50-2.51) holds for some $j' \in \{1, \ldots N_i\} - \{j\}$. It follows from Lemma 2.7.6 that $\exists (a', b') \neq (a, b)$ such that $\alpha = S_a T_b / T_a R_b = S_{a'} T_{b'} / T_{a'} R_{b'}$, which violates Assumption 1. □

2.7.7 Zero valued dual variables

In Appendices 2.7.2-2.7.6, we have established that $\Theta(\alpha, \epsilon_i)$ determines the solution of LP (2.5) for any π, ϵ_i, provided the dual variable $\alpha > 0$. Here, α is also the pico rate-multiplier.

In this Appendix, we will show that there exist positive rate multipliers (such that (2.43-2.46) hold) when the corresponding dual-variables are zero.

2.7.7.1 Existence of pico bias multiplier $\alpha_m > 0$ when the dual-variable $\alpha = 0$

When $\alpha = 0$, $\gamma_u = 0, \forall u \in U_i$ from (2.46). This implies $1/R_u > \gamma_u, \forall u \in U_i$. Now using (2.43), $x_u = 0, \forall u \in U_i$. Therefore, the value of the LP (2.5) is 0. This implies LP (2.53) must have a value $\leq \pi$.

Note that LP (2.53) is a two-tier LP, which was considered in [44]. The femto allocation z_u was derived by sorting the UEs in each U_j^i in descending order of T_u/S_u. The femto allocation was then determined by using up time ϵ_i by serving the files in the order of T_u/S_u. This is the structure of the solution of LP (2.53).

$$\begin{aligned} &\min_{y_u, z_u \geq 0} \sum_{u \in U_i} y_u/S_u \\ &\text{s.t.} \sum_{u \in U_j^i} z_u/T_u \leq \epsilon_i, \forall j \in \{1 \dots N_i\} \\ &y_u + z_u = D_u, \forall u \in U_i \end{aligned} \tag{2.53}$$

Define $\alpha_m := \min_{v \in U_i} S_v/R_v$. Note that $S_u/R_u \geq \alpha_m$ by definition. Therefore, $\alpha_m T_u/S_u \leq T_u/R_u, \forall u \in U_i$ which implies $\rho_u^{\alpha_m} = \alpha_m T_u/S_u, \forall u \in U_i$. Notice that when α_m is given as an input to $\Theta(\alpha_m, \epsilon_i)$, the UEs $u \in U_j^i$ will be sorted according to T_u/S_u in Algorithm 2 for each $j \in \{1, \dots N_i\}$. The allocation procedure in this case coincides with the optimal solution in [44]. Hence, $\Theta(\alpha_m, \epsilon_i)$ will produce an optimal allocation with $\theta(\alpha_m, \epsilon_i) = 0$.

2.7.7.2 Existence of femto bias multiplier $\beta_j^m > 0$ when dual variable $\beta_j = 0$

Recall that $\beta_j = 0$ when $\sum_{k=1}^{K} D_{u_k}/T_{u_k} < \epsilon_i$. This is a consequence of complementary slackness. Consider $\beta_j^m := \rho_{u_K}^{\alpha}$. Note that since $\rho_{u_k}^{\alpha} \geq \beta_j^m, \forall 1 \leq k \leq K$, the rate-bias rules (or the stationary conditions) (2.43-2.46) are still honored when β_j is replaced with β_j^m. Hence, β_j^m is a positive femto bias multiplier which adheres to the rate-bias rules when dual variable $\beta_j = 0$.

Chapter 3

Distributed Scheduling Algorithm for mmWave IAB networks

3.1 Introduction

mmWave cellular networks are expected to play a key role in the next generation wireless communications (5G) [7]. They are capable of delivering very high rates, due to the vast amount of spectrum available in the mmWave band. However, wireless communication at mmWave frequencies comes with two major challenges, including 1) high isotropic propagation loss, and 2) sensitivity to blockage by the objects in the environment. To overcome the high propagation losses, directional communication using beam-forming is being considered for mmWave cellular. High beam-forming gains are achievable by implementing large antenna arrays in a tiny area (which is possible due to the small wavelengths). The mmWave cell sizes are expected to be small due to the high propagation loss and blocking, and ultra dense deployments of Next Generation Node Bases (gNBs) are being considered to provide universal coverage.

It is prohibitively expensive to provide fibre backhaul support to all the gNBS under dense deployments. Hence, there has been recent interest in multi-hop relaying (or self backhauling) in mmWave cellular networks as a potential solution. Notably, as part of its standardization efforts, 3GPP has completed a recent study item on the potential solutions for efficient operation of integrated access and wireless backhaul (IAB) for NR [8]. The study emphasizes the joint consideration of radio-access and backhaul for mmWave cellular networks.

In this chapter, we consider a multi-hop IAB network, where a fraction of gNBs are deployed with dedicated fiber backhaul links, referred to as IAB donors [8]. The other gNBs (referred to as IAB nodes) relay their backhaul data over wireless mmWave links, possibly in multiple hops to an IAB donor. According to [8], an IAB node establishes a link to a parent node (either another IAB node or a donor) by following the same initial access procedure as a UE, and the central unit (CU) at the IAB donor establishes a forwarding route to the IAB node via the parent. Therefore, traffic of a UE is forwarded along this established route from the IAB donor to the UE (in downlink). The 3GPP study identified two topologies for the operation of mmWave IAB, 1) spanning tree (ST) and 2) directed acyclic graph (DAG) topology [8]. We primarily focus on the ST topology, where each IAB node has one parent node (either a IAB node or the IAB donor). An example IAB network can be seen in Figure. 3.1.

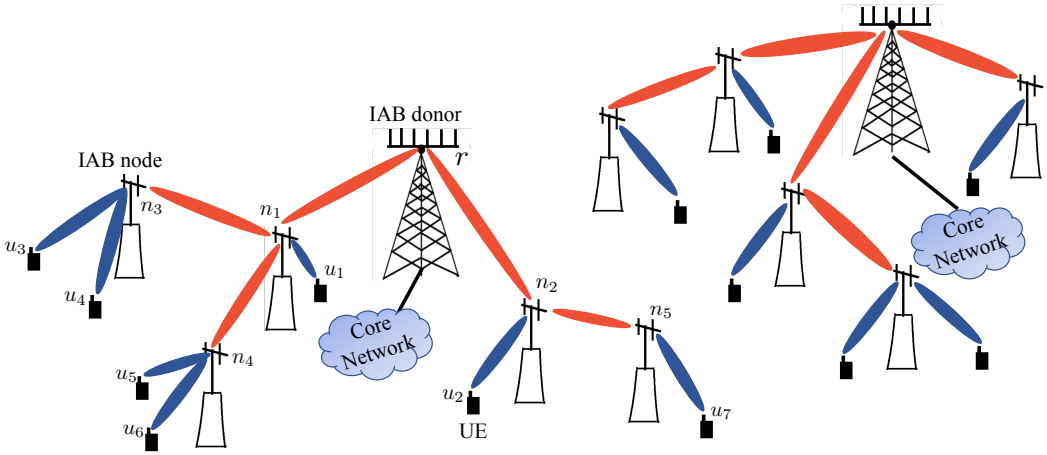


Figure 3.1: mmWave IAB network. The red links are mmWave backhaul links and the blue links are mmWave access links.

Dynamic resource allocation (or scheduling) is a key challenge in the control of multi-hop IAB networks [6, 9]. Joint consideration of access and backhaul in resource allocation for IAB networks is emphasized in [8]. According to [8], it is critical to consider in-band backhauling (i.e., backhaul and access use the same frequencies) solutions that accommodate tighter interworking access and backhaul. In an in-band scenario, the half-duplex constraint imposes restrictions on the links that can be active simultaneously. In this chapter, we characterize the capacity of 3GPP mmWave IAB networks in an in-band IAB scenario and multiple RF chains at the gNBs. We also propose a distributed scheduling algorithm for IAB networks. Our contributions are as follows.

Some works in the literature have studied joint routing and resource allocation for mmWave multi-hop networks as utility maximization problems [45–51]. Others considered queue based models [50, 52–55]. Utility maximization was used for path selection and scheduling in [50, 54, 55], and for congestion control and scheduling in [52, 53]. The solutions given in these works were centralized in nature. Moreover, some works only considered single stream downlink beamforming, and do not consider scheduling with multiple RF chains at a gNB [49, 51–54]. Dynamic path selection algorithms for topology management in mmWave networks were studied in [56–59], with [56] considering distributed schemes. Very few works have focused on distributed scheduling algorithms for mmWave networks.

Distributed scheduling for wireless networks in various setups have been studied in the literature [10–16]. Much of the work is focused on networks under a *primary interference constraint* [12, 13, 15, 16]. Under the primary interference constraint, any two links sharing a common node (either a transmitter or a receiver) are not allowed to be scheduled simultaneously. The other works considered more general *conflict constraints* [10, 11, 14]. Under the conflict constraint, a given link cannot be scheduled with any of the links in a predetermined set. In [12, 13], maximal scheduling based distributed algorithms were proposed, which were shown to achieve only a fraction of capacity in general. In [14], a distributed version of greedy maximal scheduling was proposed for a wireless network with time varying link rates. In [10, 11], Carrier Sense Multiple Access (CSMA) based distributed scheduling algorithms were proposed and shown to achieve full capacity. Pick-and-Compare (PaC) based distributed algorithms were proposed in [15, 16].

The results of the above mentioned papers cannot be directly applied to the mmWave IAB networks for the following reasons. 1) With the exception of [14], they did not consider time varying link rates, which is a key concern in mmWave networks due to blocking. 2) The RF chains impose a different type of constraint than the conflict constraint, e.g., Consider a gNB with 2 RF chains and serving 3 downlinks $\{\ell_1, \ell_2, \ell_3\}$. Even though no two links in $\{\ell_1, \ell_2, \ell_3\}$ conflict with each other, they cannot be scheduled simultaneously due to the limit on the RF chains. Only the links in the following sets $\{\ell_1, \ell_2\}, \{\ell_1, \ell_3\}, \{\ell_2, \ell_3\}$ can be scheduled simultaneously, which cannot be modelled by a conflict constraint. Hence, there is a need for designing new distributed algorithms for mmWave IAB networks. In this chapter, we provide a distributed and local scheduling algorithm for mmWave IAB networks.

- We propose a distributed and local version of the max-weight algorithm for mmWave IAB networks. The schedule at a gNB only depends on the local queue information, current link

rates and a one bit message from the parent node.

- We will show that the algorithm achieves 100% of the capacity region of the class of local policies which make decisions based on local information and information passed from the parent node, under some assumptions on the arrival and link rate processes.
- Using numerical simulations, we show that the performance (expected queue length) of the proposed algorithm is very close to that of centralized algorithms (using global information), such as the global max-weight and back-pressure in realistic scenarios.

3.2 System Model

3.2.1 SDMA Downlink Model

We consider M_n RF chains at a gNB n. The gNB n can beamform to M_n downstream nodes simultaneously. A downstream node can be a UE or a IAB node receiving backhaul. Consider a gNB n beamforming to downstream nodes $\{k_i\}_{i=1}^{M_n}$. We consider a slotted model with slots $t \in \mathbb{Z}_+$. The signal received by k_i in slot t is given as

$$y_i(t) = \boldsymbol{u}_i \boldsymbol{H}_i(t) \boldsymbol{w}_i x_i(t) + \sum_{j=1, j\neq i}^{M_n} \boldsymbol{u}_i \boldsymbol{H}_i(t) \boldsymbol{w}_j x_j(t) + z_i(t)$$

where $x_i(t), x_j(t)$ are the transmit symbols corresponding to nodes k_i and k_j resp. $z_i(t)$ is the noise at receiver k_i. $\boldsymbol{H}_i(t)$ is the channel matrix from n to k_i, which is assumed to be fixed for the slot duration. $\boldsymbol{H}_i(t) \in \mathbb{C}^{N_i^r \times N_n^t}$, where N_i^r (and N_n^t) is the number of antenna array elements at node k_i (and n resp.). $\boldsymbol{u}_i \in \mathbb{C}^{1\times N_i^r}$ (and $\boldsymbol{u}_j \in \mathbb{C}^{1\times N_j^r}$) is the receiver beam-forming vector at k_i (and k_j resp.). $\boldsymbol{w}_i \in \mathbb{C}^{N_n^t \times 1}$ (and $\boldsymbol{w}_j \in \mathbb{C}^{N_n^t \times 1}$) is transmit beam-forming vector corresponding to k_i (and k_j resp.).

The SINR of the received signal at k_i is given by

$$\mathrm{SINR}_{n,k_i}(t) = \frac{|\boldsymbol{u}_i \boldsymbol{H}_i(t) \boldsymbol{w}_i|^2}{\sum_{j=1, j\neq i}^{M_n} |\boldsymbol{u}_i \boldsymbol{H}_i(t) \boldsymbol{w}_j|^2 + \sigma^2} \tag{3.1}$$

where $\sigma^2 := \mathbb{E}[z^2(t)]$ is the noise power. The rate (in packets/slot) of the link between n and k_i in slot t is given by

$$\mu_{n,k_i}(t) = \frac{BT_s}{P} \log_2(1 + \mathrm{SINR}_{n,k_i}(t)) \tag{3.2}$$

where B is the transmission bandwidth (in Hz), T_s is the slot length (in sec) and P is the packet size (in bits).

3.2.2 Link scheduling constraints

We use a binary variable $s_\ell(t) \in \{0, 1\}$ to indicate the scheduled state of a link ℓ. $s_\ell(t) = 1$ indicates ℓ is scheduled in slot t, and $s_\ell(t) = 0$ indicates otherwise. There are two types of constraints on link scheduling - (3.3) due to the limit on number of RF chains at gNB n, and (3.4) due to the half-duplex constraint.

$$\sum_{\ell \in \mathcal{L}_n} s_\ell(t) \leq M_n \tag{3.3}$$

$$s_{b_n}(t) \times s_\ell(t) = 0, \forall \ell \in \mathcal{L}_n \tag{3.4}$$

where $\mathcal{L}_n$ is the set of downstream links of a gNB $n \in \mathcal{N}$, and b_n is a backhaul link from an upstream gNB to n.

3.2.3 Network and Queueing Model

We consider the spanning tree topology, and represent the IAB network as the rooted tree graph $G \equiv (\mathcal{U} \bigcup \mathcal{N}, \mathcal{L}, r)$ where $\mathcal{N}$ is the set of all the gNBs, $\mathcal{U}$ is the set of all the UEs and $\mathcal{L}$ denote the set of all the wireless links, i.e., the backhaul links and the access links. Here, IAB donor is the root node r. For an IAB node $n \in \mathcal{N} - \{r\}$, let $p(n)$ denote the upstream node of n in the path from n to r. We refer to $p(n)$ as the parent of node n. An IAB node $n \in \mathcal{N} - \{r\}$ gets backhaul data from node $p(n)$ via the backhaul link b_n connecting n and $p(n)$.

Assumption 2. *We abstract out the key features of a mmWave IAB network via the following modelling assumptions*

- *We assume that the graph G is a rooted tree, which models the spanning tree topology.*
- *A feasible schedule must satisfy (3.3) and (3.4), which models the half-duplex and RF chains constraints.*
- *For a mmWave access link ℓ between a gNB and a UE, we assume that $\mu_\ell(t)$ is a random variable taking values from $\{0, 1, \ldots, \mu_{\max}\}$. The effects of fading are modelled using the time-varying*

link rates. Here, $\mu_\ell(t) = 0$ corresponds to the small-scale outages[1] (due to tracking errors, beam mis-alignment etc.,).

- *For a gNB-gNB backhaul link b, we assume that $\mu_b(t)$ is a random variable taking values from $\{0, \bar{\mu}_b\}$. The backhaul links are highly directional LOS wireless links between two static gNBs. Therefore, the effect of fading is neglected. Small-scale outages are modelled using the 0 state.*

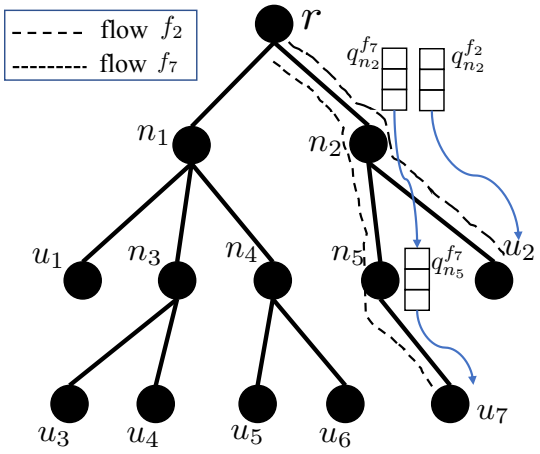


Figure 3.2: Graph representation of IAB network in Figure. 3.1.

The queuing model is as follows. Each source-destination pair $r - u$ is associated with a flow f, with root r as the source and the UE $u \in \mathcal{U}$ as the destination. The packets of flow f has to be routed from r to u via the path connecting r and u (see Figure. 3.2). Let $\mathcal{F}$ denote the set of all the flows in the network. A gNB n maintains a queue q_n^f corresponding to each flow f that passes through n. Let $q_n^f(t)$ denote the number of packets in the queue of flow f at the node n in slot t. Note that $q_n^f(t) = 0, \forall t \in \mathbb{Z}_+$ if the path of flow f does not include n.

The packet arrivals of each flow $f \in \mathcal{F}$ occur as a exogenous process at the root r. Let $a_r^f(t) \in \mathbb{Z}_+$ denote the number of packets of flow f arriving during slot t at node r, and let $d_n^f(t)$ denote the number of departures in slot t from flow f's queue at gNB n. For an IAB node $n \in \mathcal{N} - \{r\}$, packets arrive on the backhaul link from node $p(n)$ (see Figure. 3.2). Hence, arrivals into the queue q_n^f are the

[1]Note that this does not include blocking. The link outages caused due to blocking can last in the order of seconds, and a change in topology is necessary to address it. Once the new topology is established, the algorithm proposed in the chapter can be applied to stabilize the queues.

departures from $q^f_{p(n)}$. The queue evolution equations can be written as follows

$$q^f_r(t+1) = q^f_r(t) + a^f_r(t) - d^f_r(t)$$
$$q^f_n(t+1) = q^f_n(t) + d^f_{p(n)}(t) - d^f_n(t),\ \forall n \in \mathcal{N} - \{r\}$$

In the following, we present a table of important notation.

Table 3.1: Table of Notation.

Notation	Description
r	The root node in G and the IAB donor.
$p(n)$	The parent node of a gNB $n \in \mathcal{N} - \{r\}$.
b_n	The backhaul link connecting gNB $n \in \mathcal{N} - \{r\}$ to gNB $p(n)$
$\mathcal{L}_n$	The set of downlinks of gNB $n \in \mathcal{N}$.
$\mathcal{L}$	The set of all the links in G, $\mathcal{L} := \bigcup_{n\in\mathcal{N}} \mathcal{L}_n$.
$\mu_l(t)$	The link state (i.e., rate of the link in packets/slot) during slot t.
$s_l(t)$	The scheduled state of link l. $s_l(t) = 1$ if link l is scheduled in slot t, and 0 otherwise.
$q^f_n(t)$	The number of packets corresponding to flow f queued at gNB n in slot t

In the next section, we present the assumptions on the arrival and link rate processes, which are very general in nature and applicable to a wide range of examples. We define the stability criterion and characterize the stability region.

3.3 Stability

Following other works [17, 60, 61], we characterize stability under the following assumptions on the packet arrival and link rate processes.

Assumption 3. *1. The exogenous packet arrivals $\{a^f_r(t)\}_{t=0}^{\infty}$ of each flow $f \in \mathcal{F}$ is a stationary process, with a mean $\nu^f := \mathbb{E}[a^f_r(1)]$.*

2. Given any $\epsilon > 0$, there exists a positive integer M' such that $\forall M > M'$

$$\mathbb{E}\left[\left|\nu_i - \frac{1}{M}\sum_{t=k}^{k+M-1} a^f_r(t)\right|\right] < \epsilon,\ \forall f \in \mathcal{F}, k \in \mathbb{Z}_+$$

3. *Finally, the arrival process satisfies* $\lim_{A\to\infty} A^2\mathbb{P}[a_r^f(t) > A] = 0,\ \forall f \in \mathcal{F}$

Let $\boldsymbol{\mu}(t) := [\mu_l(t)]_{l\in\mathcal{L}}$ and let $\mathcal{M}$ denote the set of all possible link state vectors $\boldsymbol{\mu}$, i.e., $\boldsymbol{\mu}(t) \in \mathcal{M}, \forall t$.

Assumption 4. 1. *The link state process* $\{\boldsymbol{\mu}(t)\}_{t=0}^{\infty}$ *has a stationary distribution, where the stationary probability of being in a state* $\boldsymbol{\mu} \in \mathcal{M}$ *is denoted by* $\pi_{\boldsymbol{\mu}} > 0$.

2. *Given any* $\epsilon > 0$, *there exists a positive integer* M' *such that* $\forall M > M'$

$$\mathbb{E}\left[\left|\pi_{\boldsymbol{\mu}} - \frac{1}{M}\sum_{t=k}^{k+M-1}\mathbb{I}(\boldsymbol{\mu}(t) = \boldsymbol{\mu})\right|\right] < \epsilon,\ \forall l \in \mathcal{L}, k \in \mathbb{Z}_+$$

where $\mathbb{I}(.)$ *is the indicator function.*

3. *There exists a positive integer* $\mu_{\max}$ *such that* $\mu_l(t) \le \mu_{\max}, \forall l \in \mathcal{L}, t \in \mathbb{Z}_+$

3.3.1 Scheduling Policy

We consider stationary scheduling policies which make decisions based on the current state $[\boldsymbol{Q}(t), \boldsymbol{\mu}(t)]$. In each slot t, a scheduling policy chooses $\boldsymbol{s}(t) := [s_\ell(t)]_{\ell\in\mathcal{L}}$ only based on the current link rates $\boldsymbol{\mu}(t)$ and the queue lengths $\boldsymbol{Q}(t) = [q_n^f(t)]_{[f,n]\in\mathcal{F}\times\mathcal{N}}$, subject to the constraints (3.3) and (3.4). Let $\mathcal{S}$ denote the set of all the feasible schedules $\boldsymbol{s}$ such that constraints (3.3) and (3.4) hold. A deterministic scheduling policy provides a mapping from the $[\boldsymbol{q}, \boldsymbol{\mu}] \in \mathbb{Z}_+^{|\mathcal{F}|\times|\mathcal{N}|} \times \mathcal{M}$ to a schedule $\boldsymbol{s} \in \mathcal{S}$.

Given a state $[\boldsymbol{q}, \boldsymbol{\mu}] \in \mathbb{Z}_+^{|\mathcal{F}|\times|\mathcal{N}|} \times \mathcal{M}$, a randomized scheduling policy is the output of a random variable $Y_{\boldsymbol{q},\boldsymbol{\mu}}$ with the probability distribution $\mathcal{P}_{\boldsymbol{q},\boldsymbol{\mu}}$ on $\mathcal{S}$. The probability distribution $\mathcal{P}_{\boldsymbol{q},\boldsymbol{\mu}}$ only depends on the state $\{\boldsymbol{q}, \boldsymbol{\mu}\}$. At each time t, the schedule $s(t)$ is chosen independently according to the distribution $\mathcal{P}_{\boldsymbol{q}(t),\boldsymbol{\mu}(t)}$.

Definition 3.3.1. *We consider the system to be stable under a scheduling policy if and only if* $\limsup_{t\to\infty}\sum_{\tau=0}^{t-1}\mathbb{E}[\boldsymbol{Q}(\tau)]/t < \infty$.

Definition 3.3.2. *We consider the system to be stabilizable if and only if there exists a scheduling policy under which the system is stable.*

3.3.2 Stability region

We introduce the necessary terminology before characterizing the stability region. Let $\boldsymbol{\nu} := [\nu^f]_{f\in\mathcal{F}}$ denote the arrival rate vector. For $\ell \in \mathcal{L}$, Let $\mathcal{F}_\ell \subseteq \mathcal{F}$ denote the set of flows whose path includes ℓ, and $\nu_\ell := \sum_{f\in\mathcal{F}_\ell} \nu^f$ denotes the average arrival rate of packets into link ℓ.

For a given $\boldsymbol{\mu} \in \mathcal{M}, \boldsymbol{s} \in \mathcal{S}$, we denote the corresponding rate vector as $\boldsymbol{c_s} := \boldsymbol{\mu} \odot \boldsymbol{s}$, where $\odot$ is element-wise product. For a given link state $\boldsymbol{\mu}$, let $C_{\boldsymbol{\mu}} := \{\boldsymbol{c_s}\}_{s\in\mathcal{S}}$ denote the set of the rate vectors corresponding to the feasible schedules.

The stability region is given by the set Λ as stated in Lemma 3.3.1 and Lemma 3.3.2

$$\Lambda := \left\{ \boldsymbol{\nu} = [\nu^f]_{f\in\mathcal{F}} : [\nu_\ell]_{\ell\in\mathcal{L}} \in \sum_{\boldsymbol{\mu}\in\mathcal{M}} \pi_{\boldsymbol{\mu}} Conv(C_{\boldsymbol{\mu}}) \right\} \tag{3.5}$$

where $Conv(\cdot)$ is the convex hull of the given set.

Lemma 3.3.1. *The system is not stabilizable if $\boldsymbol{\nu} \notin \Lambda$.*

Proof. See proof of Lemma 3.3.1 in section 3.7. □

A point $\boldsymbol{\nu}$ is in interior of Λ if and only if there exists a $\delta > 0$ such that $\boldsymbol{\nu} + \delta\mathbf{1} \in \Lambda$, where $\mathbf{1} = [1]_{f\in\mathcal{F}}$.

Lemma 3.3.2. *Given a $\boldsymbol{\nu}$ in interior of Λ, there exists a stationary randomized policy which makes scheduling decisions based on the current link state $\boldsymbol{\mu}(t)$, which will stabilize the system.*

Proof. See proof of Lemma 3.3.2 in section 3.7. □

3.4 Local policies and their stability region

In this section, we consider a class of local scheduling policies where the scheduling decisions are made locally at the gNBs. We characterize the stability region of this class of policies, and propose a local algorithm which achieves stability for any arrival rate vector within capacity region for this class.

We make the following stronger assumptions on arrival and link processes for the analysis of the local policies. Note that under these assumptions, the state process $[\boldsymbol{Q}(t), \boldsymbol{\mu}(t)]_{t\in\mathbb{Z}_+}$ is a time homogeneous Markov chain under any stationary policy.

Assumption 5. *1. For each $f \in \mathcal{F}$, the arrival process $\{a_r^f(t)\}_{t=0}^{\infty}$ is an i.i.d sequence of random variables, satisfying $\mathbb{E}[(a_r^f(t))^2] < \infty$. Further, the arrival processes are independent across $f \in \mathcal{F}$.*

2. Let $\boldsymbol{\mu}_n(t) := [\mu_\ell(t)]_{\ell \in \mathcal{L}_n}$. For each $n \in \mathcal{N}$, $\{\boldsymbol{\mu}_n(t)\}_{t=0}^{\infty}$ is an i.i.d sequence of random variables. Further, the link processes $\boldsymbol{\mu}_n(t)$ are independent across $n \in \mathcal{N}$.

Consider the class $\mathcal{P}$ of local stationary scheduling policies which make scheduling decisions as follows. The decision process starts at the root, the root r makes a decision $\boldsymbol{s}_r(t) := [s_l(t)]_{l \in \mathcal{L}_r}$ based on the local information $\boldsymbol{\mu}_r(t), \boldsymbol{Q}_r(t) := [q_r^f(t)]_{f \in \mathcal{F}}$. For every other node n, the decision is made as follows

1. If $s_{b_n}(t) = 1$ (i.e., parent node $p(n)$ has decided to schedule backhaul link b_n), then the links in $\mathcal{L}_n$ are not scheduled i.e., $\boldsymbol{s}_n(t) = \mathbf{0}$

2. If $s_{b_n}(t) = 0$ (i.e., parent node $p(n)$ has decided to not schedule backhaul link b_n), then $\boldsymbol{s}_n(t) := [s_l(t)]_{l \in \mathcal{L}_n}$ is chosen such that $\sum_{l \in \mathcal{L}_n} s_l(t) \leq M_n$ based on the local information $\boldsymbol{\mu}_n(t), \boldsymbol{Q}_n(t) := [q_n^f(t)]_{f \in \mathcal{F}}$.

It can be noted that the scheduling policies in $\mathcal{P}$ do not violate the half-duplex and RF chains constraints.

Also, it can be noted that the policies in $\mathcal{P}$ satisfy the following property in common. (3.6) holds for policies in $\mathcal{P}$.

$$\mathbb{E}[s_{b_n}(t) | \boldsymbol{\mu}_n(t)] = \mathbb{E}[s_{b_n}(t)], \forall n \in \mathcal{N} - \{r\} \tag{3.6}$$

where $\boldsymbol{\mu}_n(t) := [\mu_l(t)]_{l \in \mathcal{L}_n}$. Since $s_{b_n}(t) \in \{0, 1\}$, (3.6) is equivalent to the statement that the scheduling decision of backhaul link b_n is made independently of the link states of the downstream links of gNB n.

The property (3.6) (along with Assumption 5) leads to a decomposition of stability region of $\mathcal{P}$ into individual local stability regions corresponding to each gNB. The characterization of local stability regions will be given in the following section.

3.4.1 Stability region of $\mathcal{P}$ and its decomposition

In this section, we will show that the stability region of class $\mathcal{P}$ can be decomposed into individual local stability regions corresponding to each gNB. The system is stable when the arrival rate vector is interior to each local stability region. We introduce the necessary notation for the formulation.

Consider the sub-network G_n formed using node n and its set of downstream links $\mathcal{L}_n$. We say that a local schedule $\boldsymbol{s}_n := [s_l]_{l\in\mathcal{L}_n} \in \{0,1\}^{|\mathcal{L}_n|}$ of sub-network G_n is *feasible* if and only if $\sum_{l\in\mathcal{L}_n} s_l \leq M_n$, i.e., number of activated links is less than the number of RF chains at gNB n. Let $\mathcal{S}_n$ denote the set of all the feasible local schedules $\boldsymbol{s}_n$. For a given link state $\boldsymbol{\mu}_n \in \mathcal{M}_n \subseteq \{0,\ldots,\mu_{\max}\}^{|\mathcal{L}_n|}$, we denote the rate vector corresponding to feasible schedule as $\boldsymbol{c}_{\boldsymbol{s}_n} := \boldsymbol{\mu}_n \odot \boldsymbol{s}_n$, where $\odot$ is element-wise product. For a given link state $\boldsymbol{\mu}_n$, $C_{\boldsymbol{\mu}_n} := \{\boldsymbol{c}_{\boldsymbol{s}_n}\}_{\boldsymbol{s}_n\in\mathcal{S}_n}$ is set of rate vectors corresponding to the feasible schedules. We define the local stability region Λ_n as

$$\Lambda_n := \left\{ \boldsymbol{\nu} = [\nu^f]_{f\in\mathcal{F}} : \frac{[\nu_l]_{l\in\mathcal{L}_n}}{1-\nu_{b_n}/\bar{\mu}_{b_n}} \in \sum_{\boldsymbol{\mu}_n\in\mathcal{M}_n} \pi_{\boldsymbol{\mu}_n} Conv(C_{\boldsymbol{\mu}_n}) \right\}, \tag{3.7}$$

where $\bar{\mu}_{b_n}$ is defined in Assumption 2. Recall that b_n is the backhaul link connecting n and its parent. Since the root node r has wired backhaul (and hence no parent), treat $\nu_{b_r}/\bar{\mu}_{b_r}$ as zero for Λ_r in (3.7). We define $\Lambda_{\mathcal{P}}$ as follows.

$$\Lambda_{\mathcal{P}} := \bigcap_n \Lambda_n \tag{3.8}$$

The stability region of the scheduling policies in class $\mathcal{P}$ is characterized by $\Lambda_{\mathcal{P}}$ as stated in Theorem 3.4.1.

Theorem 3.4.1. *Suppose Assumption 5 holds, then*

1. *If $\boldsymbol{\nu} \notin \Lambda_{\mathcal{P}}$, then the system is unstable under any policy in $\mathcal{P}$.*

2. *If $\boldsymbol{\nu} + \delta[1]_{f\in\mathcal{F}} \in \Lambda_{\mathcal{P}}$ for some $\delta > 0$, then system is stable under some policy in $\mathcal{P}$.*

Proof. For 1, the proof follows from Lemma 3.7.1 in section 3.7.

For 2, the proof follows from Lemma 3.7.2 in section section 3.7. □

Naturally, we have the following corollary, which states that the stability region of the local class must be superseded by the stability region of all stationary policies. In the following section, we will show that the two stability regions are the same, when the link state $\boldsymbol{\mu}(t)$ is constant for all t.

Corollary 1. $\Lambda_{\mathcal{P}} \subseteq \Lambda$

Proof. The class $\mathcal{P}$ of local policies is a subset of all the stationary scheduling policies. Hence, stability region of $\mathcal{P}$, must be included in the stability region of the class of all stationary policies.

It follows that Theorem 3.4.1 that $\Lambda_{\mathcal{P}}$ is the stability region for the class $\mathcal{P}$, and it follows from Lemma 3.3.1 and Lemma 3.3.2 that Λ is the stability region for the class of stationary policies. Hence, we have the result. □

3.4.2 Optimality of class $\mathcal{P}$ given fixed link states

In this section, we show that $\Lambda_{\mathcal{P}} = \Lambda$, provided the link state is not varying, i.e., $\boldsymbol{\mu}(t) = \boldsymbol{\mu}^d, \forall t \in \mathbb{Z}_+$. An intuitive explanation for this result is the following observation. The stability characterization of $\Lambda_{\mathcal{P}}$ results from property (3.6), (which is satisfied by all the policies in $\mathcal{P}$, under Assumption 5). If the link state is un-varying, then (3.6) holds for every stationary policy (and not just policies in class $\mathcal{P}$).

We prove the result more formally in the following Theorem 3.4.2.

Theorem 3.4.2. *Suppose that $\boldsymbol{\mu}(t) = \boldsymbol{\mu}^d, \forall t \in \mathbb{Z}_+$, where $\mu_l^d > 0, \forall l \in \mathcal{L}$. Then,*

$$\Lambda_{\mathcal{P}} = \Lambda \tag{3.9}$$

Proof. It follows from Corollary 3.4.1 that $\Lambda \supseteq \Lambda_{\mathcal{P}}$. Here, we will show that $\Lambda \subseteq \Lambda_{\mathcal{P}}$, which completes the proof.

Consider a $\boldsymbol{\nu} \in \Lambda$. Since it is given that $\boldsymbol{\mu}(t) = \boldsymbol{\mu}^d, \forall t$, it follows from the definition of Λ that $[\nu_l]_{l\in\mathcal{L}} \in Conv(C_{\boldsymbol{\mu}^d})$. Hence,

$$[\nu_l]_{l\in\mathcal{L}} \in Conv([\boldsymbol{\mu}^d \odot \boldsymbol{s}]_{s\in\mathcal{S}}) \tag{3.10}$$

where $\mathcal{S}$ is the set of all feasible schedules and $\odot$ is the element-wise product. Therefore, $\exists [p_s]_{s\in\mathcal{S}} \geq 0$ such that $[\nu_l]_{l\in\mathcal{L}} = \sum_{s\in\mathcal{S}} p_s \boldsymbol{\mu}^d \odot \boldsymbol{s}$ and $\sum_{s\in\mathcal{S}} p_s = 1$.

Consider some $n \in \mathcal{N}$. The set of feasible states $\mathcal{S}$ can be divided into two disjoint sets A_1 and A_2 as follows

$$A_1 := \{\boldsymbol{s} \in \mathcal{S} : s_{b_n} = 0\} \tag{3.11}$$

$$A_2 := \{\boldsymbol{s} \in \mathcal{S} : s_{b_n} = 1\} \tag{3.12}$$

Note that $\sum_{\boldsymbol{s}\in\mathcal{S}} p_{\boldsymbol{s}}\mu^d_{b_n} s_{b_n} = \nu_{b_n}$. It follows that $\sum_{\boldsymbol{s}\in A_2} p_{\boldsymbol{s}}\mu^d_{b_n} = \nu_{b_n}$. Therefore,

$$\sum_{\boldsymbol{s}\in A_2} p_{\boldsymbol{s}} = \nu_{b_n}/\mu^d_{b_n} \tag{3.13}$$

$$\sum_{\boldsymbol{s}\in A_1} p_{\boldsymbol{s}} = 1 - \nu_{b_n}/\mu^d_{b_n} \tag{3.14}$$

Consider the links $l \in \mathcal{L}_n$, we have

$$[\nu_l]_{l\in\mathcal{L}_n} = \sum_{\boldsymbol{s}\in\mathcal{S}} p_{\boldsymbol{s}}[\mu^d_l]_{l\in\mathcal{L}_n} \odot [s_l]_{l\in\mathcal{L}_n} \tag{3.15}$$

Let $\alpha_n(\boldsymbol{s})$ be the local feasible schedule $[s_l]_{l\in\mathcal{L}_n}$ corresponding to the feasible schedule $\boldsymbol{s}$. We define for each local schedule $\boldsymbol{s}_n \in \mathcal{S}_n$,

$$p'_{\boldsymbol{s}_n} = \sum_{\boldsymbol{s}\in A_1 : \alpha_n(\boldsymbol{s}) = \boldsymbol{s}_n} p_{\boldsymbol{s}} \tag{3.16}$$

Note that $\alpha_n(\boldsymbol{s}) = [0]_{l\in\mathcal{L}_n}, \forall \boldsymbol{s} \in A_2$. Therefore, it follows from (3.15) and (3.16) that

$$[\nu_l]_{l\in\mathcal{L}_n} = \sum_{\boldsymbol{s}_n\in\mathcal{S}_n} p'_{\boldsymbol{s}_n}[\mu^d_l]_{l\in\mathcal{L}_n} \odot \boldsymbol{s}_n \tag{3.17}$$

From construction, $\sum_{\boldsymbol{s}_n\in\mathcal{S}_n} p'_{\boldsymbol{s}_n} = \sum_{\boldsymbol{s}\in A_1} p_{\boldsymbol{s}}$. Therefore, $\sum_{\boldsymbol{s}_n\in\mathcal{S}_n} p'_{\boldsymbol{s}_n} = 1 - \nu_{b_n}/\mu^d_{b_n}$ (from (3.14)). Diving (3.17) on both sides by $(1 - \nu_{b_n}/\mu^d_{b_n})$, we have

$$\frac{[\nu_l]_{l\in\mathcal{L}_n}}{(1 - \nu_{b_n}/\mu^d_{b_n})} = \sum_{\boldsymbol{s}_n\in\mathcal{S}_n} \frac{p'_{\boldsymbol{s}_n}}{(1 - \nu_{b_n}/\mu^d_{b_n})}[\mu^d_l]_{l\in\mathcal{L}_n} \odot \boldsymbol{s}_n \tag{3.18}$$

Since, $\sum_{\boldsymbol{s}_n\in\mathcal{S}_n} \frac{p'_{\boldsymbol{s}_n}}{(1-\nu_{b_n}/\mu^d_{b_n})} = 1$, it follows that

$$\frac{[\nu_l]_{l\in\mathcal{L}_n}}{(1 - \nu_{b_n}/\mu^d_{b_n})} \in Conv(\boldsymbol{\mu}^d_n \odot \boldsymbol{s}_n) \tag{3.19}$$

where $\boldsymbol{\mu}^d_n = [\mu^d_l]_{l\in\mathcal{L}_n}$. Hence, $\boldsymbol{\nu} \in \Lambda_n$. Since the choice of n was arbitrary, it follows that $\boldsymbol{\nu} \in \Lambda_{\mathcal{P}}$. Hence, $\Lambda_{\mathcal{P}} \supseteq \Lambda$. □

3.5 Distributed and Local Max-weight Scheduling Algorithm

We present a distributed and local version of the max-weight scheduling algorithm which stabilizes the system. Recall that $\mathcal{L}_n$ is the set of downstream links of n, and b_n is the backhaul link connecting n

and its parent node $p(n)$. For a link $l \in \mathcal{L}_n$, define $q_n^l(t) := \sum_{f \in \mathcal{F}_l} q_n^f(t)$ as the total number of packets queued at n to be sent over link l.

Consider the set of links $\mathcal{L}'_n(t) \subset \mathcal{L}_n$ defined as follows; the set $\mathcal{L}'_n(t)$ contains a link $\ell \in \mathcal{L}_n$ iff either 1) ℓ is an access (gNB-UE) link such that $q_n^\ell(t) > 0$ or 2) ℓ is a backhaul (gNB-gNB) link such that $q_n^\ell(t) \geq \mu_\ell(t) > 0$. We propose the local scheduling rule as the following optimization

$$\begin{aligned} \max & \sum_{l \in \mathcal{L}_n} s_l(t)\mu_l(t)q_n^l(t) \\ \text{s.t.} & \sum_{l \in \mathcal{L}'_n(t)} s_l(t) \leq M_n \\ & s_l(t) \in \{0,1\}, \forall l \in \mathcal{L}'_n(t) \\ & s_l(t) = 0, \forall l \in \mathcal{L}_n - \mathcal{L}'_n(t) \end{aligned} \tag{3.20}$$

i.e., schedule the links with largest weights $w_l(t) := \mu_l(t)q_n^l(t)$ from the set $\mathcal{L}'_n(t)$ subject to the limit on RF chains.

3.5.1 Distributed Scheduling Policy

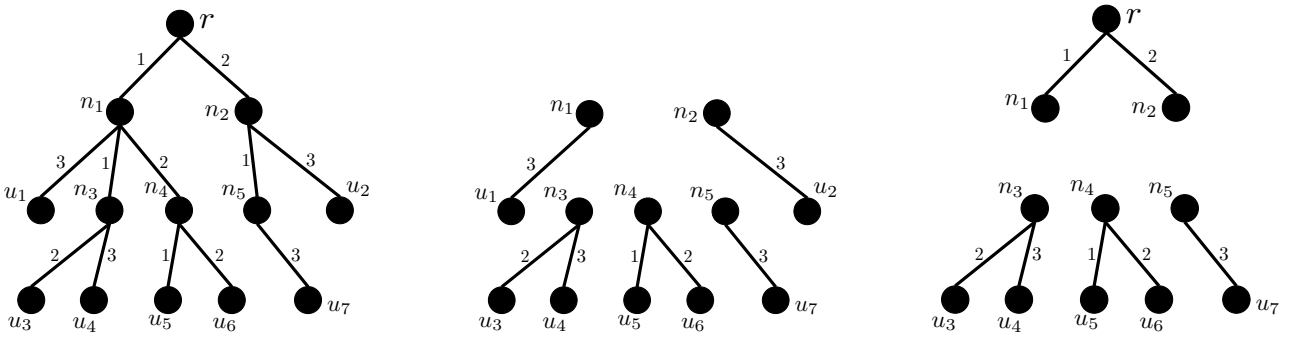
For the sake of convenience, let $\boldsymbol{s}_n(t) := [s_l(t)]_{l \in \mathcal{L}_n}$. In each slot t, the root r starts the decision process by choosing $\boldsymbol{s}_r(t)$ according to the local scheduling rule (3.20). Other nodes $n \in \mathcal{N} - \{r\}$ choose $\boldsymbol{s}_n(t)$ depending on the value of $s_{b_n}(t)$ as follows

1. If the backhaul link into n, (i.e., b_n) is scheduled in slot t and $s_{b_n}(t) = 1$, then gNB n does not transmit and $\boldsymbol{s}_n(t) = \mathbf{0}$.

2. Otherwise, $\boldsymbol{s}_n(t)$ is chosen according to the local scheduling rule (3.20).

It can be noted that the scheduling decision at $n \in \mathcal{N} - \{r\}$ depends only on the local information $\boldsymbol{\mu}_n(t), q_n^l(t)$ and the one bit information $s_{b_n}(t)$ (which is a part of $\boldsymbol{s}_{p(n)}(t)$) from the parent $p(n)$. Hence, it can be implemented in a distributed manner by down-stream passing on the tree G. It is clear that the algorithm is feasible, since it does not violate constraints (3.3)-(3.4).

We illustrate the difference between the traditional max-weight algorithm and the proposed algorithm with the example in Figure. 3.3. Consider the network and link weights $q_n^l(t)\mu_l(t)$ given in Figure. 3.3(a). Suppose each gNB node has 2 RF chains, i.e., $M_n = 2, \forall n \in \mathcal{N}$ and that



(a) Example network with weighted links

(b) Links scheduled under the traditional max-weight algorithm

(c) Links scheduled under the proposed local max-weight algorithm

Figure 3.3: Numerical example.

$q_n^l(t) \geq \mu_l(t), \forall l \in \mathcal{L}_n, n \in \mathcal{N}$. The links shown in Figure. 3.6(a) are scheduled under the traditional max-weight algorithm, with a total weight of 17. The links shown in Figure. 3.7(a) are scheduled under the proposed local max-weight algorithm, with a total weight of 14.

The following theorem characterizes the stabilizing properties of the local max-weight algorithm under Assumption 5

Theorem 3.5.1. *Given Assumption 5 holds, the system is stable under the proposed local max-weight algorithm for any $\boldsymbol{\nu}$ in the interior of $\Lambda(\mathcal{P})$.*

Proof. See proof of Theorem 3.5.1 in section 3.7. □

3.6 Numerical Results

We consider the gNB setup shown in Figure. 3.4. Here, gNB 1 is the IAB donor, and the other gNBs 2 – 5 are IAB nodes with the IAB topology shown in Figure. 3.4. The parameters for simulation are chosen as follows. For the gNB-gNB backhaul links, the distance is uniformly chosen between 340 and 440m. For the access gNB-UE links, the gNB-UE distance is chosen uniformly randomly between 0 and 200m. Following [62], we consider Rician fading for access links with K factor as 13 dB for LOS links and 6 dB for NLOS links. The fading realizations are generated independently in each slot. Following [63], we model outage of each link as an alternating renewal process. For access links, the outage periods are geometrically distributed with mean 5.56 slots, and the non-outage

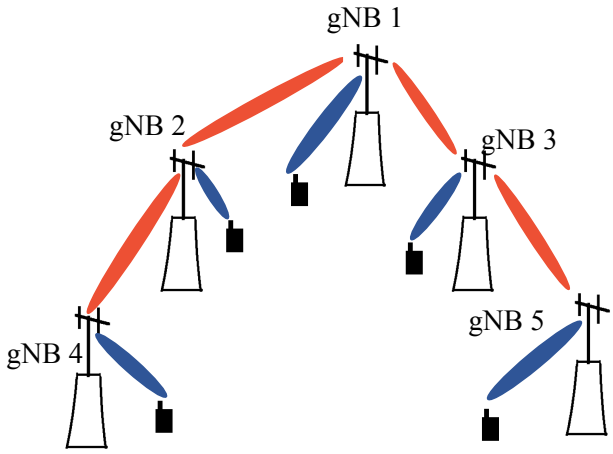


Figure 3.4: IAB Network Topology.

periods are geometrically distributed with mean 50 slots. Therefore, the stationary probability of an access link being in outage is 0.1. For backhaul links, the outage periods are are geometrically distributed with mean 1.01 slots, and the non-outage periods are geometrically distributed with mean 100 slots.Therefore, the stationary probability of a backhaul link being in outage is 0.01. We take the interference term in (3.1) to be zero.

The number of UEs associated at each gNB is chosen uniformly randomly between 4 and 11. For the arrival process, the number of packet arrivals in each slot, corresponding to each UE (or flow) is a i.i.d Poisson random variable. The mean is chosen to be the same for each UE. Other parameters are given in the following Table 3.2.

3.6.1 Comparison of scheduling policies

For comparison, we consider the following five scheduling policies.

1. *Proposed:* We consider the local algorithm proposed in the chapter.

2. *Maxweight:* We consider the traditional max weight algorithm which requires global information. The max weight algorithm maximizes the objective $\sum_{l\in\mathcal{L}} s_l(t)\mu_l(t)q_n^l(t)$ subject to the half-duplex and RF chains constraints (3.3-3.4).

3. *Backpressure:* We consider the back pressure algorithm which also requires global information. The back pressure algorithm maximizes the objective $\sum_{l\in\mathcal{L}} s_l(t)w_l^{bp}(t)$, where the back-pressure

Table 3.2: Simulation Parameters.

Paramter	Value
Carrier frequency	23 GHz
Bandwidth	1 GHz
Propagation model	3GPP Urban Micro
Slot duration	125 μs
Packet size	100 Kb
RF chains	4
Noise spectral density	-174 dBm/Hz
gNB transmit power	30 dBm
Beamforming gain	30 dB (for access), 40 dB (for backhaul)
Noise figure	5 dB (for gNB), 7 dB (for UE)

$w_l^{bp}(t) := \mu_l(t) q_n^l(t)$ for a UE link $l \in \mathcal{L}_n$, and for a backhaul link b_n from $p(n)$ to n, $w_{b_n}^{bp}(t) := \mu_{b_n}(t) \left(q_{p(n)}^{b_n}(t) - \sum_{l \in \mathcal{L}_n} q_n^l(t) \right)$.

4. *PropFair:* This is another algorithm from class $\mathcal{P}$. We consider a setup where proportional fairness algorithm is run locally at the gNBs following the hierarchy of IAB network. The proportional fairness algorithm is implemented at root r as follows. In each slot t, the root r schedules the 4 links in $\mathcal{L}_r$(since there are 4 RF chains at a gNB) with the highest ratios of instantaneous rate to average rate. The other gNBs n provide priority to the backhaul link b_n. If b_n is not scheduled in a slot t, then gNB n chooses 4 links in $\mathcal{L}_n$ with the highest ratios of instantaneous rate to average rate.

5. *LocalMaxweight:* This is another algorithm from class $\mathcal{P}$. We consider a setup where the max weight algorithm is run locally at the gNBs following the hierarchy of IAB network. Here, the max weight algorithm implemented at a gNB n maximizes $\sum_{l \in \mathcal{L}_n} \mu_l(t) q_n^l(t) s_l(t)$ provided the backhaul link b_n is not scheduled. There is a crucial difference between the proposed algorithm and this scheme. Here, all the links in $\mathcal{L}_n$ are considered for scheduling at a node n, whereas in the proposed scheme scheduling of the backhaul links in $\mathcal{L}_n$ with $q_n^l(t) < \mu_l(t)$ (i.e., small queue sizes) were avoided in favour of scheduling links at the downstream gNBs.

There are three types of UEs in the network in Figure. 3.4. 1) The UEs of gNB 1 are served by the root gNB 1. Hence packets of these UEs are not relayed over backhaul links. Here, the end-to-end delay for a UE is same as the scheduling delay (at gNB 1). 2) The UEs of gNB 2 and gNB 3. The packets of these UEs have to be relayed over 1 backhaul link. Here, the end-to-end delay is the sum of scheduling delay at gNB and backhaul delay (over 1 link). 3) The UEs of gNB 4 and gNB 5. The packets of these UEs have to be relayed over 2 backhaul links. Here, the end-to-end delay is the sum of scheduling delay at gNB and 2 backhaul delays (i.e., 2 hop). We present the average end-to-end delays of UEs of various gNBs w.r.t arrival rate in Figure. 3.5-Figure. 3.7.

Arrival rate (in packets/slot) is the expected rate of packet requests corresponding to each UE (or flow). The packet size (in Kb) and slot length (in μs) are presented in Table 3.2. In what follows, arrival rate is the expected rate of traffic (in Mb/s) corresponding to each UE.

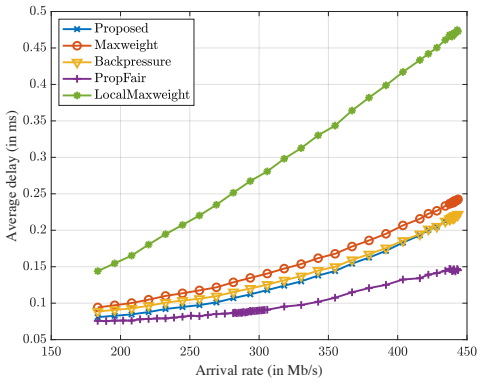
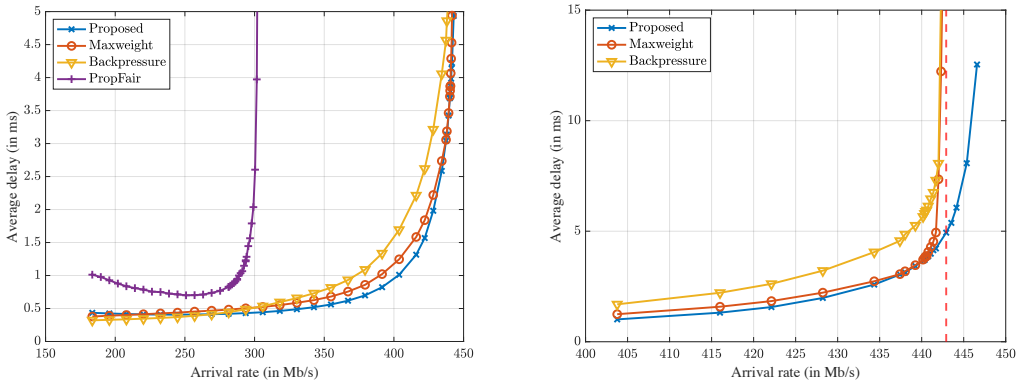


Figure 3.5: Average end-to-end delays of UEs at gNB 1.

The results for UEs at the root node gNB 1 are presented in Figure. 3.5. It can be observed that the average delays are much smaller (compared to the delays of UEs at other gNBs, given in Figure. 3.6 and Figure. 3.7) for all the considered algorithms. The PropFair algorithm has the best performance of all the schemes, with the difference being more significant at higher arrival rates. However, the following results will show that the PropFair algorithm has a smaller stability region compared to the other schemes.

The results for the UEs of gNBs 2&3 are presented in Figure. 3.6. The local max-weight algorithm is unstable for the considered arrival rates, and the end-to-end delays are unbounded. Hence, it is not plotted. It can be observed that the PropFair algorithm does not stabilize the network for arrival rates higher than 310 Mb/s. It can also be noted that the proposed algorithm has a comparable performance

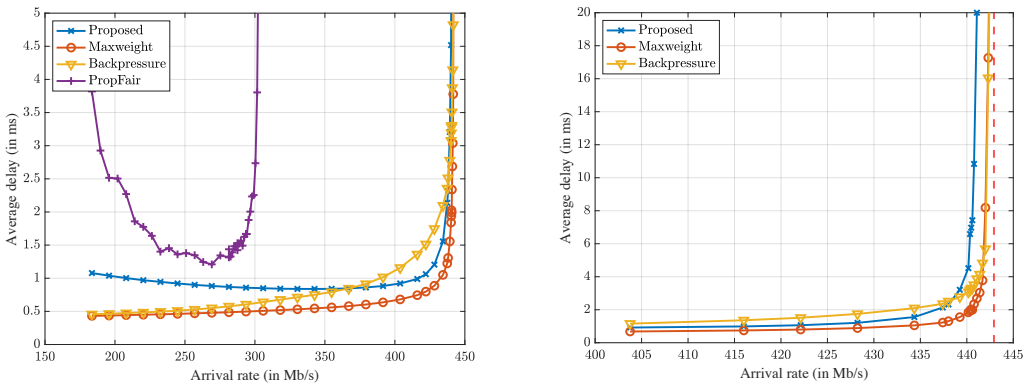


(a) Average delay vs. arrival rate.

(b) Asymptotic behaviour at high arrival rates.

Figure 3.6: Average end-to-end delays of UEs at gNBs 2&3.

to the global schemes, i.e., Maxweight and Backpressure. The asymptotic behaviour of the schemes is shown in Figure. 3.6(b). The dashed line is an upper-bound on the stability region. Here, it can be noted that the proposed algorithm has bounded delays for arrival rates beyond the stability region. However for the UEs of gNBs 4&5 (given in Figure. 3.7), the delays start blowing up at lower rates than the global schemes. The system is unstable under the proposed algorithm at arrival rates beyond 442 Mb/s (see Figure. 3.7), even though the delays are bounded for UEs at gNBs 1, 2&3. A possible explanation for this is the hierarchical nature of the proposed algorithm; the scheduling at an upstream node is given priority over the downstream nodes.



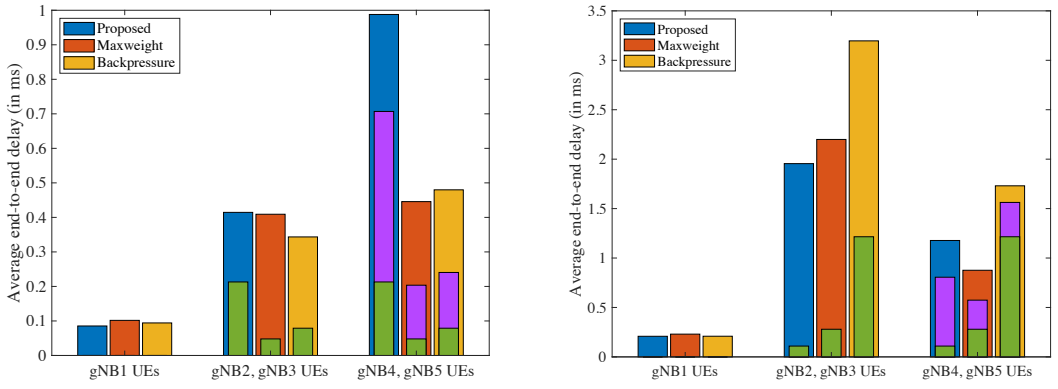
(a) Average delay vs. arrival rate.

(b) Asymptotic behaviour at high arrival rates.

Figure 3.7: Average end-to-end delays of UEs at gNBs 4&5.

The results for the third tier UEs, i.e., UEs of gNBs 4&5 are presented in Figure. 3.7. The local max-weight algorithm is unstable for the considered arrival rates, and hence not plotted. It can be observed that the delays under the PropFair algorithm blow up at approximately 300 Mb/s. It can be noted again that the proposed algorithm has a comparable performance to the global schemes, i.e., max-weight algorithm and the back-pressure algorithm. The asymptotic behaviour of the schemes is shown in Figure. 3.6(b). The dashed line is an upper-bound on the stability region. Here, it can be noted that the delays (under the proposed algorithm) start blowing up at lower rates than the Maxweight and Backpressure. This is the gap between the capacity achieved by the proposed algorithm and the global schemes (Maxweight and Backpressure).

For the considered simulation, the gap in capacity (between the global schemes and the proposed algorithm) is small. Lemma 3.4.2 provides an explanation. The variation (over time) in the mmWave link states is small in the considered IAB scenario. Hence, the proposed scheduling algorithm can be applied for in such scenarios (i.e., where the link variations are small) without a significant loss in capacity.



(a) Average delays for arrival rate of 214 Mb/s.

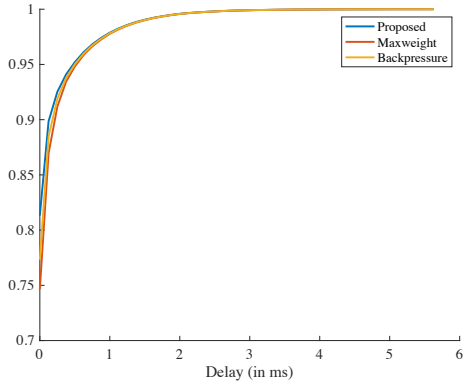
(b) Average delays for arrival rate of 428 Mb/s.

Figure 3.8: Average end-to-end delays.

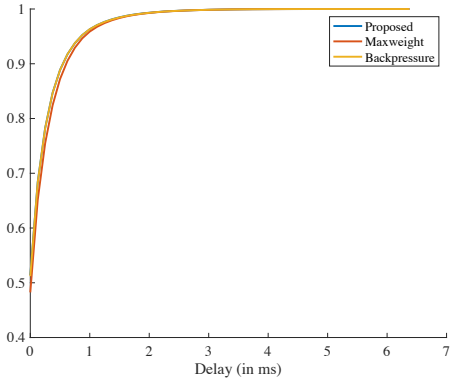
Figure. 3.8 presents the average delays of UEs at arrival rates 214 Mb/s (in Figure 3.8(a)) and 428 Mb/s (in Figure. 3.8(b)). As mentioned earlier, the end-to-end delays to UEs at gNBs $2-5$ are the sum of scheduling delay and backhaul delay. For the UEs of gNBs $i = 2,3$, the green bar represents the backhaul delay on the link connecting gNB 1 and gNB i. For the UEs of gNBs 4&5, the packets have to routed along two backhaul links. The green bar represents the delay on the first hop, and the pink bar represents the delay on the second hop.

It can observed that at low arrival rate 214 Mb/s, the proposed algorithm has much higher backhaul delays (compared to the other schemes). This is because the backhaul links b are only considered for scheduling when the criterion $q_b(t) \geq \mu_b(t)$ holds. Hence under the proposed algorithm, the packets are queued until the backhaul link capacity is reached before transmission is attempted (even though a scheduling resource, i.e., RF chain, might be available earlier). This leads to idling under the proposed algorithm at low arrival rates. At higher arrival rates such as 428 Mb/s, it can be observed that this phenomenon does not have a significant impact on end-to-end delay, since the queues build up quicker at higher arrival rates.

We now present the cumulative distribution of end-to-end delays under various schemes at arrival rates 214 and 428 Mb/s.



(a) End-to-end delays for UEs of gNB 1 for arrival rate of 214 Mb/s.



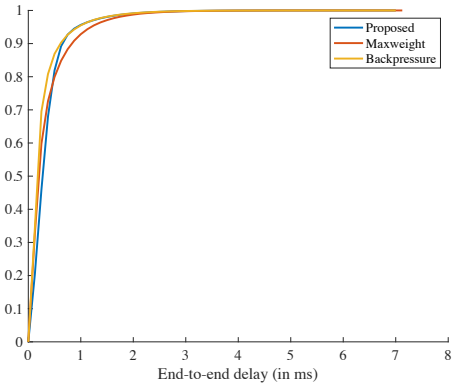
(b) End-to-end delays for UEs of gNB 1 for arrival rate of 428 Mb/s.

Figure 3.9: Cumulative distribution of end-to-end delays of UEs of gNB 1.

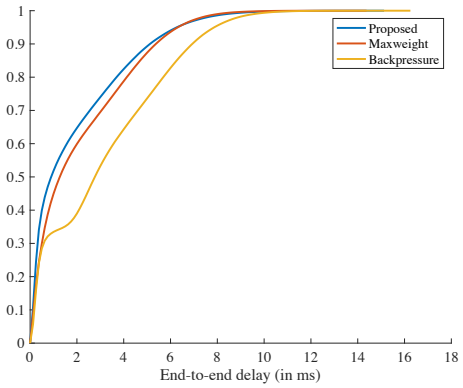
Figure. 3.9 presents the distribution of end-to-end delays for UEs of gNB 1. Here, it can be observed that the results of the proposed algorithm are very close the other schemes.

Figure. 3.10 presents the distribution for UEs of gNBs 2 and 3. The results of the proposed algorithm are comparable to the other schemes at arrival rate 214 Mb/s. At 428 Mb/s, the proposed algorithm has a better performance (we think due to the hierarchical preference given to UEs of gNBs 2 & 3 over the downstream UEs).

Figure. 3.11 presents the distribution for UEs of gNBs 4 and 5. The results of the proposed algorithm are worse compared to the other schemes at arrival rate 214 Mb/s (due to the earlier

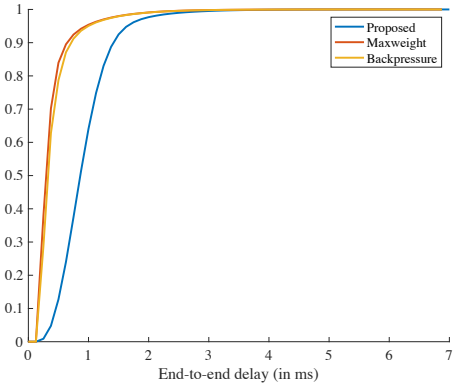


(a) End-to-end delays for UEs of gNBs 2&3 for arrival rate of 214 Mb/s.

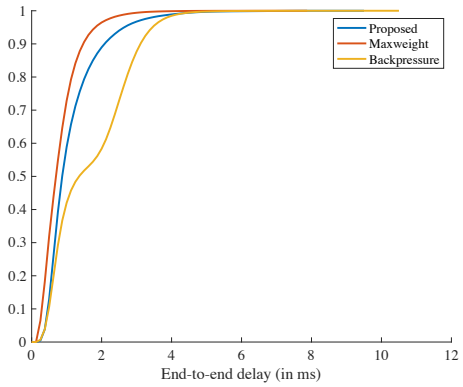


(b) End-to-end delays for UEs of gNBs 2&3 for arrival rate of 428 Mb/s.

Figure 3.10: Cumulative distribution of end-to-end delays of UEs of gNBs 2 & 3.



(a) End-to-end delays for UEs of gNBs 4&5 for arrival rate of 214 Mb/s.



(b) End-to-end delays for UEs of gNBs 4&5 for arrival rate of 428 Mb/s.

Figure 3.11: Cumulative distribution of end-to-end delays of UEs of gNB 4 & 5.

explained phenomenon of idling at low arrival rates). At 428 Mb/s, the proposed algorithm has a performance which lies between the Maxweight and the Backpressure algorithms.

3.7 Theoretical results

3.7.1 Telescoping equations

Consider P_n as the set of nodes in the path from n to r including n and r. We now derive telescoping equations of the aggregate queues flowing along this path, which will be used in the proofs. We first

introduce necessary terminology. Let

$$A_r^\ell(t) := \sum_{f \in \mathcal{F}_\ell} a_r^f(t), \forall \ell \in \mathcal{L} \tag{3.21}$$

$$D_n^\ell(t) := \sum_{f \in \mathcal{F}_\ell} d_n^f(t), \forall \ell \in \mathcal{L}, n \in \mathcal{N} \tag{3.22}$$

Also define,

$$Q_n^\ell(t) := \sum_{f \in \mathcal{F}_\ell} q_n^f(t), \forall \ell \in \mathcal{L},\ n \in \mathcal{N} \tag{3.23}$$

$$V_\ell(t) := \sum_{n' \in P_n} Q_{n'}^\ell(t), \forall \ell \in \mathcal{L}_n, n \in \mathcal{N} \tag{3.24}$$

Consider a link $\ell \in \mathcal{L}_n$. For any $n' \in P_n - \{r\}$, we have

$$Q_{n'}^\ell(t+1) = Q_{n'}^\ell(t) + D_{p(n')}^\ell(t) - D_{n'}^\ell(t) \tag{3.25}$$

For node r, we have

$$Q_r^\ell(t+1) = Q_r^\ell(t) + A_r^\ell(t) - D_r^\ell(t) \tag{3.26}$$

(3.26) and (3.25) for $n' \in P_n - \{r\}$ form a telescoping series of equations. Summing them yields

$$V_\ell(t+1) = V_\ell(t) + A_r^\ell(t) - D_n^\ell(t) \tag{3.27}$$

where $V_\ell(t) := \sum_{n' \in P_n} Q_{n'}^\ell(t)$ for any node $n \in \mathcal{N}$ and $\ell \in \mathcal{L}_n$.

3.7.2 Results for section 3.3

Proof of Lemma 3.3.1. From (3.27), for any $\ell \in \mathcal{L}$, we have

$$V_\ell(\tau+1) \geq V_\ell(\tau) + A_r^\ell(\tau) - \mu_\ell(\tau) s_\ell(\tau) \tag{3.28}$$

$$\implies \boldsymbol{V}(\tau+1) \geq \boldsymbol{V}(\tau) + \boldsymbol{a}(\tau) - \boldsymbol{\mu}(\tau) \odot \boldsymbol{s}(\tau) \tag{3.29}$$

where $\boldsymbol{V}(\tau) := [V_\ell(\tau)]_{\ell \in \mathcal{L}}$ and $\boldsymbol{a}(\tau) := [A_r^\ell(\tau)]_{\ell \in \mathcal{L}}$.

From Assumption 3 and Assumption 4, there exists M such that $\forall t \geq M$, we have $\mathbb{E}[\sum_{\tau=0}^{t-1} \boldsymbol{a}(\tau)] > t[\nu_l - \epsilon]_{l \in \mathcal{L}}$ and $\mathbb{E}[\sum_{\tau=0}^{t-1} \boldsymbol{\mu}(\tau) \odot \boldsymbol{s}(\tau)] < t\boldsymbol{c} + t\epsilon\mathbf{1}$ for some $\boldsymbol{c} \in \sum_{\mu \in \mathcal{M}} \pi_\mu Conv(C_\mu)$. Let $C := \sum_{\mu \in \mathcal{M}} \pi_\mu Conv(C_\mu)$.

Consider a $t \geq M$, summing (3.29) from $\tau = 0$ to $t-1$ and taking expectation, we have

$$\mathbb{E}[\boldsymbol{V}(t)] \geq \mathbb{E}[\boldsymbol{V}(0)] + \mathbb{E}[\sum_{\tau=0}^{t-1} \boldsymbol{a}(\tau)] - \mathbb{E}[\sum_{\tau=0}^{t-1} \boldsymbol{\mu}(\tau) \odot \boldsymbol{s}(\tau)] \tag{3.30}$$

$$\mathbb{E}[\boldsymbol{V}(t)] - \mathbb{E}[\boldsymbol{V}(0)] \geq t\,(\boldsymbol{\nu} - \boldsymbol{c}) - 2t\epsilon\mathbf{1} \tag{3.31}$$

Since $\boldsymbol{\nu} \notin \Lambda$ and $\boldsymbol{c} \in C$, there exists $\ell \in \mathcal{L}$ such that $\nu_\ell - c_\ell \geq \delta > 0$, where $\delta = \inf_{\boldsymbol{c}' \in C} \max_{l \in \mathcal{L}} \nu_l - c'_l$. Since the choice of ϵ is arbitrary, fix $\epsilon := \delta/4$. It follows that $\mathbb{E}[V_\ell(t)] - \mathbb{E}[V_\ell(0)] \geq t\delta/2$. It follows that

$$\mathbb{E}[\|\boldsymbol{V}(t)\|_1] \geq t\delta/2,\ \forall t \geq M \tag{3.32}$$

Therefore, $\sum_{\tau=M}^{t} \mathbb{E}[\|\boldsymbol{V}(\tau)\|_1] \geq \delta/2\,((t-M+1)M + (t-M)(t-M+1)/2)$. This implies

$$\limsup_{t\to\infty} \sum_{\tau=0}^{t} \mathbb{E}[\|\boldsymbol{V}(\tau)\|_1]/t = \infty$$

under any scheduling policy. Hence, the system cannot be stabilized. □

Proof of Lemma 3.3.2. By definition, $\exists \delta > 0$ such that $\boldsymbol{\nu'} := \boldsymbol{\nu} + \delta\mathbf{1} \in \Lambda$. From (3.5), there must exist $\{\boldsymbol{c}_{\boldsymbol{\mu}}\}_{\boldsymbol{\mu}\in\mathcal{M}}$ such that $[\nu'_\ell]_{\ell\in\mathcal{L}} = \sum_{\mu\in\mathcal{M}} \pi_\mu \boldsymbol{c}_{\boldsymbol{\mu}}$ and $\boldsymbol{c}_{\boldsymbol{\mu}} \in Conv(C_{\boldsymbol{\mu}})$. For any given $\boldsymbol{\mu}$, since $\boldsymbol{c}_{\boldsymbol{\mu}} \in Conv(C_{\boldsymbol{\mu}})$, there exists $\{p_{\boldsymbol{\mu},\boldsymbol{s}}\}_{\boldsymbol{s}\in\mathcal{S}}$ such that $\boldsymbol{c}_\mu = \sum_{\boldsymbol{s}\in\mathcal{S}} p_{\boldsymbol{\mu},\boldsymbol{s}}(\boldsymbol{\mu} \odot \boldsymbol{s})$ and $\sum_{\boldsymbol{s}\in\mathcal{S}} p_{\boldsymbol{\mu},\boldsymbol{s}} = 1$.

Now consider the stationary randomized policy which schedules a set $\boldsymbol{s} \in \mathcal{S}$ w.p. $p_{\boldsymbol{\mu},\boldsymbol{s}}$ provided the current link state is $\boldsymbol{\mu}$, i.e., $\mathbb{P}[\boldsymbol{s}(t) = \boldsymbol{s}|\boldsymbol{\mu}(t) = \boldsymbol{\mu}] = p_{\boldsymbol{\mu},\boldsymbol{s}}$. From construction, it follows that $\mathbb{E}[\mu_\ell(t)s_\ell(t)] = \nu'_\ell = \nu_\ell + \delta, \forall \ell \in \mathcal{L}$.

Consider $\epsilon := \delta/4$, it follows from Assumption 3 and Assumption 4 that there exists M such that for any $\tau \geq 0$ and $\ell \in \mathcal{L}$

$$\mathbb{E}[\sum_{t=\tau}^{\tau+M-1} A_r^\ell(t)] < M(\nu_\ell + \epsilon) \tag{3.33}$$

$$\mathbb{E}[\sum_{t=\tau}^{\tau+M-1} \mu_\ell(t)s_\ell(t)] > M(\nu'_\ell - \epsilon) \tag{3.34}$$

We now analyze the M-step drift under the randomized policy. Consider a link $\ell \in \mathcal{L}_r$ and the summation of (3.26) from $t = \tau$ to $t = \tau + M - 1$, we have

$$Q_r^\ell(\tau + M - 1) = Q_r^\ell(\tau) + \sum_{t=\tau}^{\tau+M-1} A_r^\ell(t) - \sum_{t=\tau}^{\tau+M-1} D_r^\ell(t) \tag{3.35}$$

$$Q_r^\ell(\tau + M - 1) \leq \max\left\{0, Q_r^\ell(\tau) - \sum_{t=\tau}^{\tau+M-1} \mu_\ell(t)s_\ell(t)\right\} + \sum_{t=\tau}^{\tau+M-1} A_r^\ell(t) \tag{3.36}$$

Define $A_\ell(\tau) := \sum_{t=\tau}^{\tau+M-1} A_r^\ell(t)$ and $D_\ell(\tau) := \sum_{t=\tau}^{\tau+M-1} \mu_\ell(t) s_\ell(t)$.

It follows that

$$\left(Q_r^\ell(\tau+M-1)\right)^2 \leq \left(Q_r^\ell(\tau) - D_\ell(\tau)\right)^2 + A_\ell^2(\tau) + 2A_\ell(\tau)\max\{0, Q_r^\ell(\tau) - D_\ell(\tau)\} \tag{3.37}$$

$$\leq \left(Q_r^\ell(\tau) - D_\ell(\tau)\right)^2 + A_\ell^2(\tau) + 2A_\ell(\tau)Q_r^\ell(\tau) \tag{3.38}$$

$$= \left(Q_r^\ell(\tau)\right)^2 + D_\ell^2(\tau) + A_\ell^2(\tau) + 2Q_r^\ell(\tau)(A_\ell(\tau) - D_\ell(\tau)) \tag{3.39}$$

Hence, we have $\mathbb{E}\left[\left(Q_r^\ell(\tau+M-1)\right)^2 - \left(Q_r^\ell(\tau)\right)^2 | \boldsymbol{Q}(\tau)\right]$

$$\leq \mathbb{E}[D_\ell^2(\tau)|\boldsymbol{Q}(\tau)] + \mathbb{E}[A_\ell^2(\tau)|\boldsymbol{Q}(\tau)] + 2Q_r^\ell(\tau)\mathbb{E}[A_\ell(\tau) - D_\ell(\tau)|\boldsymbol{Q}(\tau)] \tag{3.40}$$

$$= \mathbb{E}[D_\ell^2(\tau)] + \mathbb{E}[A_\ell^2(\tau)] + 2Q_r^\ell(\tau)\mathbb{E}[A_\ell(\tau) - D_\ell(\tau)] \tag{3.41}$$

From point 3 of Assumption 3, we have $\lim_{k\to\infty} k^2\mathbb{P}[A_\ell^2(\tau) > k] = 0$. Hence, we have $E[A_\ell^2(\tau)] < \infty$. From point 3 of Assumption 4, we have $D_\ell(\tau) \leq M\mu_{\max}$. Hence, for some positive constant K_ℓ, we have $\mathbb{E}[A_\ell^2(\tau)] + \mathbb{E}[D_\ell^2(\tau)] \leq K_\ell$. Also, $\mathbb{E}\left[A_\ell(\tau) - D_\ell(\tau)\right] < 2M\epsilon + M(\nu_\ell - \nu'_\ell) = -M\delta/2$ from (3.33),(3.34). Hence,

$$\mathbb{E}\left[\left(Q_r^\ell(\tau+M-1)\right)^2 - \left(Q_r^\ell(\tau)\right)^2 | \boldsymbol{Q}(\tau)\right] < K_\ell - M\delta Q_r^\ell(\tau) \tag{3.42}$$

Unconditioning (3.42) w.r.t the distribution of $\boldsymbol{Q}(\tau)$, we get

$$\mathbb{E}\left[\left(Q_r^\ell(\tau+M-1)\right)^2 - \left(Q_r^\ell(\tau)\right)^2\right] < K_\ell - M\delta\mathbb{E}[Q_r^\ell(\tau)] \tag{3.43}$$

Now, summing over τ from $\tau = 0$ to $T-1$, where $T > M$, we have

$$\sum_{\tau=0}^{T-1}\mathbb{E}\left[\left(Q_r^\ell(\tau+M-1)\right)^2\right] - \sum_{\tau=0}^{T-1}\mathbb{E}\left[\left(Q_r^\ell(\tau)\right)^2\right] < TK_\ell - M\delta\sum_{\tau=0}^{T-1}\mathbb{E}[Q_r^\ell(\tau)] \tag{3.44}$$

$$\tag{3.45}$$

Note that

$$\sum_{\tau=0}^{T-1}\mathbb{E}\left[\left(Q_r^\ell(\tau+M-1)\right)^2\right] - \sum_{\tau=0}^{T-1}\mathbb{E}\left[\left(Q_r^\ell(\tau)\right)^2\right] \tag{3.46}$$

$$= \sum_{\tau=M-1}^{T+M-2}\mathbb{E}\left[\left(Q_r^\ell(\tau)\right)^2\right] - \sum_{\tau=0}^{T-1}\mathbb{E}\left[\left(Q_r^\ell(\tau)\right)^2\right] \tag{3.47}$$

$$= \sum_{\tau=T}^{T+M-2}\mathbb{E}\left[\left(Q_r^\ell(\tau)\right)^2\right] - \sum_{\tau=0}^{M-2}\mathbb{E}\left[\left(Q_r^\ell(\tau)\right)^2\right] \tag{3.48}$$

$$= \sum_{m=0}^{M-2}\mathbb{E}\left[\left(Q_r^\ell(T+m)\right)^2\right] - \sum_{m=0}^{M-2}\mathbb{E}\left[\left(Q_r^\ell(m)\right)^2\right] \tag{3.49}$$

Hence, the telescoping series in (3.44) yields

$$\sum_{m=0}^{M-2} \mathbb{E}\left[\left(Q_r^{\ell}(T+m)\right)^2\right] - \sum_{m=0}^{M-2} \mathbb{E}\left[\left(Q_r^{\ell}(m)\right)^2\right] < TK_{\ell} - M\delta \sum_{\tau=0}^{T-1} \mathbb{E}[Q_r^{\ell}(\tau)] \tag{3.50}$$

$$\implies \sum_{\tau=0}^{T-1} \frac{\mathbb{E}[Q_r^{\ell}(\tau)]}{T} < \frac{K_{\ell}}{M\delta T} + \sum_{m=0}^{M-2} \frac{\mathbb{E}\left[\left(Q_r^{\ell}(m)\right)^2\right]}{M\delta T} \tag{3.51}$$

Hence, $\limsup_{T\to\infty} \sum_{\tau=0}^{T-1} \mathbb{E}[Q_r^{\ell}(\tau)]/T < \infty, \forall \ell \in \mathcal{L}_r$.

In the rest of the proof, we use induction to show that $\limsup_{T\to\infty} \sum_{\tau=0}^{T-1} \mathbb{E}[Q_n^{\ell}(\tau)]/T < \infty, \forall \ell \in \mathcal{L}_n$, for each $n \in \mathcal{N} - \{r\}$.

Suppose $\limsup_{T\to\infty} \sum_{\tau=0}^{T-1} \mathbb{E}[Q_{n'}^{\ell}(\tau)]/T < \infty, \forall \ell \in \mathcal{L}_{n'}, n' \in P_n$, where P_n is the set of nodes in the path from n to r excluding n. Consider a link $l \in \mathcal{L}_n$. We will now show that $\limsup_{T\to\infty} \sum_{\tau=0}^{T-1} \mathbb{E}[Q_n^{l}(\tau)]/T < \infty$.

Consider the drift $\mathbb{E}\left[V_l^2(\tau+M-1) - V_l^2(\tau)|\mathcal{Q}(\tau)\right]$

$$= \mathbb{E}\left[\left(V_l(\tau+M-1) - V_l(\tau)\right)^2 |\mathcal{Q}(\tau)\right] + 2V_l(\tau)\mathbb{E}\left[\left(V_l(\tau+M-1) - V_l(\tau)\right)|\mathcal{Q}(\tau)\right] \tag{3.52}$$

$$\leq K_l + 2\left(V_l(\tau) - Q_n^{l}(\tau)\right)\mathbb{E}\left[V_l(\tau+M-1) - V_l(\tau)|\mathcal{Q}(t)\right] + 2Q_n^{l}(\tau)\mathbb{E}\left[V_l(\tau+M-1) - V_l(\tau)|\mathcal{Q}(\tau)\right] \tag{3.53}$$

where, K_l is a positive constant such that $E[\sum_{t=\tau}^{\tau+M-1}(A_r^l(t))^2 + \sum_{t=\tau}^{\tau+M-1}(\mu_l(t)s_l(t))^2|\mathcal{Q}(\tau)] \leq K_l$. K_l exists from point 3 of Assumption 3 and Assumption 4, as argued earlier with K_{ℓ} in (3.41).

Since $V_l(\tau+M-1) - V_l(\tau) \leq \sum_{t=\tau}^{\tau+M-1} A_r^l(t)$, it follows from (3.33) that the second term in (3.53) can be upper-bounded by $2(V_l(\tau) - Q_n^l(\tau))(M\nu_l + \epsilon)$, which is a linear function of $\mathcal{Q}_{p(n)}(\tau) := \{q_{n'}^f(\tau)\}_{n'\in P_n, f\in\mathcal{F}}$. Let $g_l(.)$ denote the function. Now, we focus on manipulating the third term, $2Q_n^l(\tau)\mathbb{E}\left[V_l(\tau+M-1) - V_l(\tau)|\mathcal{Q}(\tau)\right]$, which equals

$$2Q_n^l(\tau)\mathbb{E}\left[A_l(\tau) - D_l(\tau)|\mathcal{Q}(\tau)\right] + 2Q_n^l(\tau)\mathbb{E}\left[D_l(\tau) - \sum_{t=\tau}^{\tau+M-1} D_n^l(t)|\mathcal{Q}(\tau)\right] \tag{3.54}$$

where $A_l(\tau) := \sum_{t=\tau}^{\tau+M-1} A_r^l(t)$ and $D_l(\tau) := \sum_{t=\tau}^{\tau+M-1} \mu_l(t)s_l(t)$. Observe that when $Q_n^l(\tau) \geq M\mu_{\max}$, the second term in (3.54) equals zero (because $D_n^l(t) = \mu_l(t)s_l(t)$ for $t = \tau$ to $\tau+M-1$). Otherwise, when $Q_n^l(\tau) < M\mu_{\max}$, it is upper-bounded by $2M^2\mu_{\max}^2$ (because $D_l(\tau) \leq \mu_{\max}M$). Hence, the second term in (3.54) can be upper-bounded by the constant $2M^2\mu_{\max}^2$. Therefore,

$$2Q_n^l(\tau)\mathbb{E}\left[V_l(\tau+M-1) - V_l(\tau)|\mathcal{Q}(\tau)\right] < 2M^2\mu_{\max}^2 + 2Q_n^l(\tau)\mathbb{E}\left[A_l(\tau) - D_l(\tau)\right] \tag{3.55}$$

$$< 2M^2\mu_{\max}^2 - M\delta Q_n^l(\tau) \tag{3.56}$$

since $\mathbb{E}[A_l(\tau) - D_l(\tau)] < -M\delta/2$ from (3.33) and (3.34).

Now, using (3.56) and (3.53) yields

$$\mathbb{E}\left[V_l^2(\tau + M - 1) - V_l^2(\tau)|\boldsymbol{Q}(\tau)\right] < K_l + 2M^2\mu_{\max}^2 + g_l(\boldsymbol{Q}_{p(n)}(\tau)) - M\delta Q_n^l(\tau) \tag{3.57}$$

Unconditioning w.r.t distribution of $\boldsymbol{Q}(\tau)$, we get

$$\mathbb{E}\left[V_l^2(\tau + M - 1) - V_l^2(\tau)\right] < K_l + 2M^2\mu_{\max}^2 + g_l(\boldsymbol{Q}_{p(n)}(\tau)) - M\delta\mathbb{E}[Q_n^l(\tau)] \tag{3.58}$$

Now, summing from $\tau = 0$ to $T - 1$ yields (3.59) (from similar arguments as in (3.46)-3.49)

$$\sum_{m=0}^{M-2}\mathbb{E}\left[V_l^2(T+m)\right] - \sum_{m=0}^{M-2}\mathbb{E}\left[V_l^2(m)\right] < T(K_l + 2M^2\mu_{\max}^2) + \sum_{\tau=0}^{T-1} g_l(\boldsymbol{Q}_{p(n)}(\tau)) - M\delta\sum_{\tau=0}^{T-1} Q_n^l(\tau) \tag{3.59}$$

$$\sum_{\tau=0}^{T-1}\frac{\mathbb{E}[Q_n^l(\tau)]}{T} < \frac{K_l + 2M^2\mu_{\max}^2}{M\delta} + \sum_{\tau=0}^{T-1}\frac{g_l(\boldsymbol{Q}_{p(n)}(\tau))}{M\delta T} + \sum_{m=0}^{M-2}\frac{\mathbb{E}\left[V_l^2(m)\right]}{M\delta T} \tag{3.60}$$

We have $\limsup_{T\to\infty} g_l(\boldsymbol{Q}_{p(n)}(\tau))/(M\delta T) < \infty$ from the supposition that $\limsup_{T\to\infty}\sum_{\tau=0}^{T-1}\mathbb{E}[Q_{n'}^{\ell}(\tau)]/T < \infty, \forall \ell \in \mathcal{L}_{n'}, n' \in P_n$. It follows that $\limsup_{T\to\infty}\mathbb{E}[Q_n^l(\tau)]/T < \infty$.

By principle of mathematical induction, we have $\limsup_{T\to\infty}\mathbb{E}[Q_n^l(\tau)]/T < \infty, \forall l \in \mathcal{L}$. Hence, the stationary randomized policy stabilizes the system. □

3.7.3 Results for section 3.4

Lemma 3.7.1. *If $[\nu_l]_{l\in\mathcal{L}_n} \notin \Lambda_n$, then the system is unstable under any policy in $\mathcal{P}$.*

Proof. Firstly, note that under Assumption 5, the state process $\{\boldsymbol{Q}(t), \boldsymbol{\mu}(t)\}_{t=0}^{\infty}$ is a time homogeneous Markov chain. By definition, for a stable system $\limsup_{t\to\infty}\sum_{\tau=0}^{t-1}\mathbb{E}[\boldsymbol{Q}(\tau)]/t$ exists. It follows that the Markov chain must be positive recurrent for a stable system. Also note that for a positive recurrent chain, $\lim_{t\to\infty}\sum_{\tau=0}^{t-1}\mathbb{E}[\boldsymbol{Q}(\tau)]/t$ exists and equal to the expectation taken over the stationary distribution.

Based on this knowledge, we provide a proof by contradiction. Suppose that the system is stable under a stationary policy for some $[\nu_l]_{l\in\mathcal{L}_n} \notin \Lambda_n$. Assume that the Markov chain $\{\boldsymbol{Q}(t)\}_{t=0}^{\infty}$ starts with the stationary distribution, i.e., the initial distribution is same as the stationary distribution of the Markov chain.

Since the Markov chain is in stationary distribution and the system is assumed to be stable, it follows that $\mathbb{E}[\mu_{b_n}(t)s_{b_n}(t)] \geq \nu_{b_n}$. Hence,

$$\mathbb{P}[s_{b_n}(t) = 1] \geq \nu_{b_n}/\bar{\mu}_{b_n} \tag{3.61}$$

$$\mathbb{P}[s_{b_n}(t) = 0] \leq 1 - \nu_{b_n}/\bar{\mu}_{b_n} \tag{3.62}$$

Consider the links $l \in \mathcal{L}_n$, $\forall t \geq 0$

$$\mathbb{E}[V_l(t+1) - V_l(t)] \geq \mathbb{E}[A_r^l(t)] - \mathbb{E}[\mu_l(t)s_l(t)] \tag{3.63}$$

$$= \nu_l - \mathbb{P}[s_{b_n}(t) = 0]\mathbb{E}[\mu_l(t)s_l(t)|s_{b_n}(t) = 0] \tag{3.64}$$

$$\geq \nu_l - (1 - \nu_{b_n}/\bar{\mu}_{b_n})\mathbb{E}[\mu_l(t)s_l(t)|s_{b_n}(t) = 0] \tag{3.65}$$

Let consider the term $\mathbb{E}[\mu_l(t)s_l(t)|s_{b_n}(t) = 0]$. Since $\mathbb{E}[s_{b_n}(t)|\boldsymbol{\mu}_n(t)] = \mathbb{E}[s_{b_n}(t)]$ for any policy in class $\mathcal{P}$, it follows that

$$\mathbb{E}[\mu_l(t)s_l(t)|s_{b_n}(t) = 0] = \sum_{\boldsymbol{\mu}_n \in \mathcal{M}_n} \mu_l \mathbb{P}[\boldsymbol{\mu}_n(t) = \boldsymbol{\mu}_n|s_{b_n}(t) = 0]\mathbb{E}[s_l(t)|s_{b_n}(t) = 0, \boldsymbol{\mu}_n(t) = \boldsymbol{\mu}_n] \tag{3.66}$$

$$= \sum_{\boldsymbol{\mu}_n \in \mathcal{M}_n} \mu_l \mathbb{P}[\boldsymbol{\mu}_n(t) = \boldsymbol{\mu}_n]\mathbb{E}[s_l(t)|s_{b_n}(t) = 0, \boldsymbol{\mu}_n(t) = \boldsymbol{\mu}_n] \tag{3.67}$$

$$= \sum_{\boldsymbol{\mu}_n \in \mathcal{M}_n} \mu_l \pi_{\boldsymbol{\mu}_n} \mathbb{P}[s_l(t) = 1|s_{b_n}(t) = 0, \boldsymbol{\mu}_n(t) = \boldsymbol{\mu}_n] \tag{3.68}$$

It follows from (3.68) that

$$\left[\mathbb{E}[\mu_l(t)s_l(t)|s_{b_n}(t) = 0]\right]_{l \in \mathcal{L}_n} \in \sum_{\boldsymbol{\mu}_n \in \mathcal{M}_n} \pi_{\boldsymbol{\mu}_n} Conv(C_{\boldsymbol{\mu}_n}) \tag{3.69}$$

Suppose $\frac{\nu_l}{1-\nu_{b_n}/\bar{\mu}_{b_n}} \leq \mathbb{E}[\mu_l(t)s_l(t)|s_{b_n}(t) = 0]$ for each $l \in \mathcal{L}_n$, then from (3.69), $[\nu_l]_{l \in \mathcal{L}_n} \in \Lambda_n$, which is a contradiction since it is given that $[\nu_l]_{l \in \mathcal{L}_n} \notin \Lambda_n$.

Hence, there must exist a $\ell \in \mathcal{L}_n$ such that $\nu_l > (1 - \nu_{b_n}/\bar{\mu}_{b_n})\mathbb{E}[\mu_\ell(t)s_\ell(t)|s_{b_n}(t) = 0]$. Then, $\mathbb{E}[V_\ell(t+1) - V_\ell(t)] > 0$ from (3.65). This is also a contradiction since the Markov chain is assumed to be in stationary distribution, which completes the proof. □

Lemma 3.7.2. *If $\boldsymbol{\nu} + \delta\mathbf{1} \in \Lambda_{\mathcal{P}}$ for some $\delta > 0$, then the system is stable under a policy in $\mathcal{P}$.*

Proof. **Randomized stationary policy $\hat{s}$ in $\mathcal{P}$**

Consider a stationary randomized policy $\hat{s}$ in $\mathcal{P}$ which makes decisions $\hat{\boldsymbol{s}}(t)$ based on $\boldsymbol{\mu}(t)$ defined as follows. The decision process starts at the root, the root r chooses $\hat{\boldsymbol{s}}_r(t) = \boldsymbol{s}_r \in \mathcal{S}_r$ w.p. $p_{\boldsymbol{\mu}_r(t),\boldsymbol{s}_r}$ given the current link state $\boldsymbol{\mu}_r(t) \in \mathcal{M}_r$. For every other node n, the decision is made as follows

1. If $\hat{s}_{b_n}(t) = 1$ (i.e., parent node $p(n)$ has decided to schedule backhaul link b_n), then the links in $\mathcal{L}_n$ are not scheduled i.e., $\hat{\boldsymbol{s}}_n(t) = \mathbf{0}$

2. If $\hat{s}_{b_n}(t) = 0$ (i.e., parent node $p(n)$ has decided to not schedule backhaul link b_n), then gNB n chooses $\hat{\boldsymbol{s}}_n(t) = \boldsymbol{s}_n \in \mathcal{S}_n$ w.p. $p_{\boldsymbol{\mu}_n(t),\boldsymbol{s}_n}$ given the current link state $\boldsymbol{\mu}_n(t) \in \mathcal{M}_n$.

The values $p_{\boldsymbol{\mu}_n,\boldsymbol{s}_n}$ for each $\boldsymbol{\mu}$, and $n \in \mathcal{N}$ are chosen according to the following procedure.

Since $\boldsymbol{\nu} + \delta\mathbf{1} \in \Lambda(\mathcal{P})$, it follows that $[\nu_l]_{l\in\mathcal{L}} + \delta\mathbf{1} \in \Lambda_n, \forall n \in \mathcal{N}$. From the definition of Λ_n,

$$\left[\frac{(\nu_l+\delta)}{(1-\nu_{b_n}/\bar{\mu}_{b_n})}\right]_{l\in\mathcal{L}_n} \in \sum_{\boldsymbol{\mu}_n\in\mathcal{M}_n} \pi_{\boldsymbol{\mu}_n} Conv(C_{\boldsymbol{\mu}_n}) \tag{3.70}$$

It can be observed that for any $\delta_{p(n)} < \frac{(\nu_l+\delta)\delta}{(1-\nu_{b_n}/\bar{\mu}_{b_n})}$, there exists a $\delta_n > 0$ such that

$$\frac{(\nu_l+\delta_n)}{(1-\nu_{b_n}/\bar{\mu}_{b_n}-\delta_{p(n)})} \leq \frac{(\nu_l+\delta)}{(1-\nu_{b_n}/\bar{\mu}_{b_n})} \tag{3.71}$$

It is immediate that for any $\delta_{p(n)} < \frac{(\min_{l\in\mathcal{L}_n}\nu_l+\delta)\delta}{(1-\nu_{b_n}/\bar{\mu}_{b_n})}$, there exists $\delta_n > 0$ such that

$$\left[\frac{(\nu_l+\delta_n)}{(1-\nu_{b_n}/\bar{\mu}_{b_n}-\delta_{p(n)})}\right]_{l\in\mathcal{L}_n} \in \sum_{\boldsymbol{\mu}_n\in\mathcal{M}_n} \pi_{\boldsymbol{\mu}_n} Conv(C_{\boldsymbol{\mu}_n}) \tag{3.72}$$

Hence, there must exist $\{\delta_n\}_{n\in\mathcal{L}_n} > 0$ such that for each $n \in \mathcal{N}$,

$$\left[\nu_l^{(1)}\right]_{l\in\mathcal{L}_n} \in \sum_{\boldsymbol{\mu}_n\in\mathcal{M}_n} \pi_{\boldsymbol{\mu}_n} Conv(C_{\boldsymbol{\mu}_n}) \tag{3.73}$$

where $\nu_l^{(1)} = (\nu_l+\delta_n)/(1-\nu_{b_n}/\bar{\mu}_{b_n}-\delta_{p(n)})$. By definition, there must exist $\{\boldsymbol{c}_{\boldsymbol{\mu}_n}\}_{\boldsymbol{\mu}_n\in\mathcal{M}_n}$ such that $\boldsymbol{c}_{\boldsymbol{\mu}_n} \in Conv(C_{\boldsymbol{\mu}_n})$ and $[\nu_l^{(1)}]_{l\in\mathcal{L}_n} = \sum_{\boldsymbol{\mu}_n\in\mathcal{M}_n} \pi_{\boldsymbol{\mu}_n}\boldsymbol{c}_{\boldsymbol{\mu}_n}$. Since $\boldsymbol{c}_{\boldsymbol{\mu}_n} \in Conv(C_{\boldsymbol{\mu}_n})$, it can be expressed as

$$\boldsymbol{c}_{\boldsymbol{\mu}_n} = \sum_{\boldsymbol{s}_n\in\mathcal{S}_n} p_{\boldsymbol{\mu}_n,\boldsymbol{s}_n}\boldsymbol{\mu}_n \odot \boldsymbol{s}_n \text{ such that } \sum_{\boldsymbol{s}_n\in\mathcal{S}_n} p_{\boldsymbol{\mu}_n,\boldsymbol{s}_n} = 1 \tag{3.74}$$

We now use proof by induction to show that the randomized policy stabilizes the system.

Negative drift of queues at gNB r under the randomized policy

Consider a link $l \in \mathcal{L}_r$, we have

$$Q_r^l(t+1) = Q_r^l(t) + A_r^l(t) - D_r^l(t) \tag{3.75}$$

$$= \max\{0, Q_r^l(t) - \mu_l(t)s_l(t)\} + A_r^l(t) \tag{3.76}$$

It follows that

$$\left(Q_r^l(t+1)\right)^2 \leq \left(Q_r^l(t) - \mu_l(t)s_l(t)\right)^2 + \left(A_r^l(t)\right)^2 + 2A_r^l(t)\max\{0, Q_r^l(t) - \mu_l(t)s_l(t)\} \tag{3.77}$$

$$\leq \left(Q_r^l(t) - \mu_l(t)s_l(t)\right)^2 + \left(A_r^l(t)\right)^2 + 2A_r^l(t)Q_r^l(t) \tag{3.78}$$

$$= \left(Q_r^l(t)\right)^2 + \mu_l^2(t)s_l^2(t) + \left(A_r^l(t)\right)^2 + 2Q_r^l(t)\left(A_r^l(t) - \mu_l(t)s_l(t)\right) \tag{3.79}$$

Consider the drift $\mathbb{E}[\left(Q_r^l(t+1)\right)^2 - \left(Q_r^l(t)\right)^2 | \boldsymbol{Q}(t)]$ under the policy $\hat{\boldsymbol{s}}$. It follows from (3.79) that

$$\mathbb{E}\left[\left(Q_r^l(t+1)\right)^2 - \left(Q_r^l(t)\right)^2 | \boldsymbol{Q}(t)\right] \leq \underbrace{\mathbb{E}[\mu_l^2(t) + \left(A_r^l(t)\right)^2]}_{\leq \mu_{\max}^2 + \mathbb{E}[\left(A_r^l(0)\right)^2]} + 2Q_r^l(t)\left(\nu_l - \mathbb{E}[\mu_l(t)\hat{s}_l(t)|\boldsymbol{Q}(t)]\right) \tag{3.80}$$

$$\leq K_l + 2Q_r^l(t)\left(\underbrace{\nu_l - \mathbb{E}[\mu_l(t)\hat{s}_l(t)]}_{\because\ \hat{\boldsymbol{s}}(t) \text{ only depends on } \boldsymbol{\mu}(t)}\right) \tag{3.81}$$

$$= K_l + 2Q_r^l(t)\left(\nu_l - \sum_{\boldsymbol{\mu}_n \in \mathcal{M}_n} \pi_{\boldsymbol{\mu}_r} p_{\boldsymbol{\mu}_r, s_r} \mu_l s_l\right) \tag{3.82}$$

$$= K_l - 2Q_r^l(t)\delta_r \tag{3.83}$$

where $K_l := \mathbb{E}[\mu_l^2(t) + \left(A_r^l(t)\right)^2]$, $\delta_r := \min_{\ell \in \mathcal{L}_r} \sum_{\boldsymbol{\mu}_n \in \mathcal{M}_n} \pi_{\boldsymbol{\mu}_r} p_{\boldsymbol{\mu}_r, s_r} \mu_\ell s_\ell - \nu_\ell$.

Unconditioning (3.83) w.r.t the distribution of $\boldsymbol{Q}(t)$, we get

$$\mathbb{E}\left[\left(Q_r^l(t+1)\right)^2 - \left(Q_r^l(t)\right)^2\right] \leq K_l - 2\delta_r \mathbb{E}[Q_r^l(t)] \tag{3.84}$$

Summing over from $t = 0$ to $T-1$, we get

$$\sum_{t=0}^{T-1} \mathbb{E}\left[\left(Q_r^l(t+1)\right)^2\right] - \sum_{t=0}^{T-1} \mathbb{E}\left[\left(Q_r^l(t)\right)^2\right] \leq TK_l - 2\delta_r \sum_{t=0}^{T-1} \mathbb{E}[Q_r^l(t)] \tag{3.85}$$

$$\mathbb{E}\left[\left(Q_r^l(T)\right)^2 - \left(Q_r^l(0)\right)^2\right] \leq TK_l - 2\delta_r \sum_{t=0}^{T-1} \mathbb{E}[Q_r^l(t)] \tag{3.86}$$

$$\implies \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[Q_r^l(t)] \leq K_l/2\delta_r + \mathbb{E}[\left(Q_r^l(0)\right)^2]/2T\delta_r \tag{3.87}$$

Hence, $\limsup_{T\to\infty} \sum_{t=0}^{T-1} \mathbb{E}[Q_r^l(t)]/T < \infty, \forall l \in \mathcal{L}_r$.

Negative drift of queues at gNB n under the randomized policy

Assume that $\limsup_{T\to\infty} \sum_{t=0}^{T-1} \mathbb{E}[Q_{n'}^l(t)]/T < \infty, \forall l \in \mathcal{L}_{n'}, \forall n' \in P_n$, where P_n is the set of nodes in the path from n to r excluding n. Consider a link $l \in \mathcal{L}_n$. We will now show that $\limsup_{T\to\infty} \sum_{t=0}^{T-1} \mathbb{E}[Q_n^l(t)]/T < \infty$

Consider the drift $\mathbb{E}[V_l^2(t+1) - V_l^2(t)|\boldsymbol{Q}(t)]$, which equals

$$\mathbb{E}[(V_l(t+1) - V_l(t))^2 | \boldsymbol{Q}(t)] + 2V_l(t)\mathbb{E}[V_l(t+1) - V_l(t)|\boldsymbol{Q}(t)] \tag{3.88}$$

$$\leq K_l + 2\left(V_l(t) - Q_n^l(t)\right)\mathbb{E}[V_l(t+1) - V_l(t)|\boldsymbol{Q}(t)] + 2Q_n^l(t)\mathbb{E}[V_l(t+1) - V_l(t)|\boldsymbol{Q}(t)] \tag{3.89}$$

where $K_l = \mathbb{E}[\mu_l^2(t) + \left(A_r^l(t)\right)^2]$. Since $V_l(t+1) - V_l(t) \leq A_r^l(t)$, the middle term in (3.89) is upper bounded by $2\nu_l(V_l(t) - Q_n^l(t))$, which is a linear function of $\boldsymbol{Q}_{p(n)}(t) := \{q_{n'}^f\}_{n' \in P_n, f \in \mathcal{F}}$. Let $g_l(.)$ denote this function. Now, we focus on the last term in (3.89), $2Q_n^l(t)\mathbb{E}[V_l(t+1) - V_l(t)|\boldsymbol{Q}(t)]$, which equals

$$2Q_n^l(t)\left(\mathbb{E}[A_r^l(t)] - \mathbb{E}[\mu_l(t)\hat{s}_l(t)]\right) + 2Q_n^l(t)\mathbb{E}[\mu_l(t)\hat{s}_l(t) - D_n^l(t)|\boldsymbol{Q}(t)] \tag{3.90}$$

$$= -2\delta_n Q_n^l(t) + 2Q_n^l(t)\mathbb{E}[\mu_l(t)\hat{s}_l(t) - D_n^l(t)|\boldsymbol{Q}(t)] \tag{3.91}$$

where $\delta_n := \min_{\ell \in \mathcal{L}_n} \sum_{\mu_n \in \mathcal{M}_n} \pi_{\mu_n} p_{\mu_n, s_n} \mu_\ell s_\ell - \nu_\ell$.

Note that $\mathbb{E}[\mu_l(t)\hat{s}_l(t) - D_n^l(t)|\boldsymbol{Q}(t)]$ is zero whenever $Q_n^l(t) \geq \mu_{\max}$ (since $D_n^l(t)$ equals $\mu_l(t)\hat{s}_l(t)$). Otherwise if $Q_n^l(t) < \mu_{\max}$, it is upper bounded by $\mu_{\max}^2$ (since $\mu_l(t)\hat{s}_l(t) < \mu_{\max}$). Hence, the second term in (3.91) is always upper bounded by $\mu_{\max}^2$. Hence, it follows from (3.91) and (3.89) that

$$\mathbb{E}[V_l^2(t+1) - V_l^2(t)|\boldsymbol{Q}(t)] \leq K_l + \mu_{\max}^2 + g_l(\boldsymbol{Q}_{p(n)}(t)) - 2\delta_n Q_n^l(t) \tag{3.92}$$

Unconditioning w.r.t distribution of $\boldsymbol{Q}(t)$, and summing from $t = 0$ to $T-1$ yields

$$\mathbb{E}[V_l^2(T)] - \mathbb{E}[V_l^2(0)] \leq T(K_l + \mu_{\max}^2) + \sum_{t=0}^{T-1} \mathbb{E}[g_l(\boldsymbol{Q}_{p(n)}(t))] - 2\delta_n \sum_{t=0}^{T-1} \mathbb{E}[Q_n^l(t)] \tag{3.93}$$

$$\implies \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[Q_n^l(t)] \leq \frac{K_l + \mu_{\max}^2}{2\delta_n} + \frac{\mathbb{E}[V_l^2(0)]}{2T\delta_n} + \frac{1}{2\delta_n}\sum_{t=0}^{T-1} \mathbb{E}[g_l(\boldsymbol{Q}_{p(n)}(t))]/T \tag{3.94}$$

Recall the assumption that $\limsup_{T\to\infty} \sum_{t=0}^{T-1} \mathbb{E}[Q_{n'}^l(t)]/T < \infty, \forall l \in \mathcal{L}_{n'}, \forall n' \in P_n$. Hence, it follows that $\limsup_{T\to\infty} \sum_{t=0}^{T-1} \mathbb{E}[Q_n^l(t)]/T < \infty$.

By principle of mathematical induction, we have $\limsup_{T\to\infty} \sum_{t=0}^{T-1} \mathbb{E}[Q_n^l(t)]/T < \infty, \forall l \in \mathcal{L}$. Hence, the stationary randomized policy $\hat{\boldsymbol{s}}$ stabilizes the system. □

3.7.4 Results for section 3.5

Lemma 3.7.3. *Given $\boldsymbol{\nu}$ is interior of $\Lambda_{\mathcal{P}}$, the queues at node r are stable under the proposed scheduling policy, i.e.,* $\limsup_{T\to\infty} \sum_{t=0}^{T} \sum_{f\in\mathcal{F}} \mathbb{E}[q_r^f(t)]/T < \infty$

Proof of Lemma 3.7.3. Define $V_r(\boldsymbol{Q}(t)) := \sum_{l\in\mathcal{L}_r}(Q_r^l(t))^2$. Let $A_r^l(t) := \sum_{f\in\mathcal{F}_l} a_r^f(t)$ and $D_r^l(t) :=$

$\sum_{f\in\mathcal{F}_l} d_r^f(t)$. With a slight abuse of notation, let $\Delta_r(\boldsymbol{Q}(t)) := \mathbb{E}[V_r(\boldsymbol{Q}(t+1)) - V_r(\boldsymbol{Q}(t))|\boldsymbol{Q}(t)]$

$$\Delta_r(\boldsymbol{Q}(t)) = \sum_{l\in\mathcal{L}_r} \mathbb{E}[\Big(Q_r^l(t+1) - Q_r^l(t)\Big)^2 |\boldsymbol{Q}(t)] + 2\sum_{l\in\mathcal{L}_r} Q_r^l(t)\mathbb{E}[Q_r^l(t+1) - Q_r^l(t)|\boldsymbol{Q}(t)] \tag{3.95}$$

$$= \sum_{l\in\mathcal{L}_r} \underbrace{\mathbb{E}[\Big(A_r^l(t) - D_r^l(t)\Big)^2 |\boldsymbol{Q}(t)]}_{\leq \mathbb{E}[(A_r^l(t))^2] + \mu_{\max}^2} + 2\sum_{l\in\mathcal{L}_r} Q_r^l(t)\mathbb{E}[A_r^l(t) - D_r^l(t)|\boldsymbol{Q}(t)] \tag{3.96}$$

$$\leq K + 2\sum_{l\in\mathcal{L}_r} Q_r^l(t)\mathbb{E}[A_r^l(t)|\boldsymbol{Q}(t)] - 2\mathbb{E}[\sum_{l\in\mathcal{L}_r} Q_r^l(t)d_r^l(t))|\boldsymbol{Q}(t)] \tag{3.97}$$

where $K = \sum_{l\in\mathcal{L}_r} \mathbb{E}[(A_r^l(t))^2] + |\mathcal{L}_r|\mu_{\max}^2$

Recall that $s_l(t) = 0, l \in \mathcal{L}_r - \mathcal{L}_r'(t)$, under the proposed policy. Consider the set $\mathcal{L}_r^{(1)}(t) \subset \mathcal{L}_r'(t)$ defined as the set of links $l \in \mathcal{L}_r$ such that $Q_r^l(t) \geq \mu_l(t) > 0$. Note that $Q_r^l(t)D_r^l(t) = Q_r^l(t)\mu_l(t)s_l(t)$ for the links $l \in \mathcal{L}_r^{(1)}(t)$. For links $l \in \mathcal{L}_r'(t) - \mathcal{L}_r^{(1)}(t)$, either $\mu_l(t) = 0$ or $Q_r^l(t) < \mu_l(t)$ by definition. This implies $Q_r^l(t)\mu_l(t)s_l(t) \leq \mu_l^2(t)s_l(t), \forall l \in \mathcal{L}_r'(t) - \mathcal{L}_r^{(1)}(t)$. Hence, we have

$$\sum_{l\in\mathcal{L}_r} Q_r^l(t)d_r^l(t) = \sum_{l\in\mathcal{L}_r^{(1)}(t)} Q_r^l(t)d_r^l(t) + \sum_{l\in\mathcal{L}_r'(t)-\mathcal{L}_r^{(1)}(t)} Q_r^l(t)d_r^l(t) \tag{3.98}$$

$$\geq \sum_{l\in\mathcal{L}_r^{(1)}(t)} Q_r^l(t)\mu_l(t)s_l(t) \tag{3.99}$$

$$= \sum_{l\in\mathcal{L}_r'(t)} Q_r^l(t)\mu_l(t)s_l(t) - \sum_{l\in\mathcal{L}_r'(t)-\mathcal{L}_r^{(1)}(t)} \underbrace{Q_r^l(t)\mu_l(t)s_l(t)}_{\leq \mu_l^2(t)} \tag{3.100}$$

$$\geq \sum_{l\in\mathcal{L}_r'(t)} Q_r^l(t)\mu_l(t)s_l(t) - |\mathcal{L}_r|\mu_{\max}^2 \tag{3.101}$$

It follows from (3.97) and (3.101) that

$$\Delta_r(\boldsymbol{Q}(t)) \leq K_1 + 2\sum_{l\in\mathcal{L}_r} Q_r^l(t)\nu_l - 2\mathbb{E}[\sum_{l\in\mathcal{L}_r'(t)} Q_r^l(t)\mu_l(t)s_l(t)|\boldsymbol{Q}(t)] \tag{3.102}$$

where $K_1 := K + 2|\mathcal{L}_r|\mu_{\max}^2$. Let $\boldsymbol{s}\hat{(t)} = [\hat{s}_l(t)]_{l\in\mathcal{L}}$ denote the schedule under the randomized scheduling policy in Lemma 3.7.2. Since the proposed algorithm maximizes $\sum_{l\in\mathcal{L}_r'(t)} \mu_l(t)Q_r^l(t)s_l(t)$ for any given $\boldsymbol{\mu}(t), \boldsymbol{Q}(t)$, it follows that $\sum_{l\in\mathcal{L}_r'(t)} \mu_l(t)Q_r^l(t)s_l(t) \geq \sum_{l\in\mathcal{L}_r'(t)} \mu_l(t)q_r^l(t)\hat{s}_l(t)$. Hence from (3.102),

$$\Delta_r(\boldsymbol{Q}(t)) \leq K_1 + 2\sum_{l\in\mathcal{L}_r} Q_r^l(t)\nu_l - 2\mathbb{E}[\sum_{l\in\mathcal{L}_r'(t)} Q_r^l(t)\mu_l(t)\hat{s}_l(t)|\boldsymbol{Q}(t)] \tag{3.103}$$

Note that $\forall l \in \mathcal{L}_r - \mathcal{L}_r'(t)$, either $\mu_l(t) = 0$ or $Q_r^l(t) \leq \mu_l(t)$. This implies $Q_r^l(t)\mu_l(t)\hat{s}_l(t) \leq \mu_l^2(t) \leq \mu_{\max}^2, \forall l \in \mathcal{L}_r - \mathcal{L}_r'(t)$. Therefore, $0 \leq 2\Big(|\mathcal{L}_r|\mu_{\max}^2 - \mathbb{E}[\sum_{l\in\mathcal{L}_r-\mathcal{L}_r'(t)} Q_r^l(t)\mu_l(t)\hat{s}_l(t)|\boldsymbol{Q}(t)]\Big)$. Adding

this to (3.103), we have

$$\Delta_r(\boldsymbol{Q}(t)) \leq K_2 + 2\sum_{l\in\mathcal{L}_r} Q_r^l(t)\nu_l - 2\sum_{l\in\mathcal{L}_r} Q_r^l(t)\mathbb{E}[\mu_l(t)\hat{s}_l(t)|\boldsymbol{Q}(t)] \tag{3.104}$$

where $K_r = K_1 + 2|\mathcal{L}_r|\mu_{\max}^2$. Since the randomized policy $\hat{\boldsymbol{s}}(t)$ in Lemma 3.7.2 only makes decisions based on the channel state $\boldsymbol{\mu}(t)$, we have $\mathbb{E}[\sum_{l\in\mathcal{L}_r} Q_r^l(t)\mu_l(t)\hat{s}_l(t)|\boldsymbol{Q}(t)] = \sum_{l\in\mathcal{L}_r} Q_r^l(t)\mathbb{E}[\mu_l(t)\hat{s}_l(t)]$. Moreover, since $\hat{\boldsymbol{s}}(t)$ is a stabilizing policy, $\exists\delta_r > 0$ such that $\mathbb{E}[\mu_l(t)\hat{s}_l(t)] = \nu_l + \delta_r, \forall l \in \mathcal{L}_r$. Hence, it follows from (3.104) that

$$\Delta_r(\boldsymbol{Q}(t)) \leq K_r + 2\sum_{l\in\mathcal{L}_r} Q_r^l(t)\nu_l - 2\sum_{l\in\mathcal{L}_r} Q_r^l(t)(\nu_l + \delta_r) \tag{3.105}$$

$$= K_r - 2\delta_r \sum_{l\in\mathcal{L}_r} Q_r^l(t) \tag{3.106}$$

Therefore, $\mathbb{E}[V_r(\boldsymbol{Q})(t+1)] - \mathbb{E}[V_r(\boldsymbol{Q})(t)] \leq K_r - 2\delta_r \sum_{l\in\mathcal{L}_r} \mathbb{E}[Q_r^l(t)]$. Summing from $t = 0$ to $T-1$, we have

$$\mathbb{E}[V_r(\boldsymbol{Q}(T))] - \mathbb{E}[V_r(\boldsymbol{Q}(0))] \leq TK_r - 2\delta_r \sum_{t=0}^{T-1}\sum_{l\in\mathcal{L}_r} \mathbb{E}[Q_r^l(t)] \tag{3.107}$$

$$\implies \sum_{t=0}^{T-1}\sum_{l\in\mathcal{L}_r} \frac{\mathbb{E}[Q_r^l(t)]}{T} \leq \frac{K_r}{2\delta_r} + \frac{\mathbb{E}[V_r(\boldsymbol{Q})(0)]}{2\delta_r T} \tag{3.108}$$

Since (3.108) holds for all $T \geq 0$, we have $\limsup_{T\to\infty} \sum_{t=0}^{T-1}\sum_{l\in\mathcal{L}_r} \mathbb{E}[Q_r^l(t)]/T \leq \frac{K_r}{2\delta_r}$. □

Lemma 3.7.4. *Given $\boldsymbol{\nu}$ is interior of $\Lambda(\mathcal{P})$, for $n \in \mathcal{N} - \{r\}$, $\exists K_n, \epsilon_n > 0$ such that*

$$\sum_{l\in\mathcal{L}_n} Q_n^l(t)\mathbb{E}[(W_l(t+1) - W_l(t))|\boldsymbol{Q}(t)] \leq K_n - \epsilon_n \sum_{l\in\mathcal{L}_n} Q_n^l(t)$$

under the proposed scheduling policy. Here, $W_l(t) := V_l(t) + \alpha_l V_{b_n}(t)$ for some $\alpha_l > 0$.

Proof of Lemma 3.7.4. We provide the proof in four parts. In the first part, we introduce a local randomized policy and derive some properties. We make use of these properties to complete proof.

Local randomized policy and its properties

Consider a local randomized policy which operates at n, and makes decisions $\hat{\boldsymbol{s}}^{loc}(t) := [\hat{s}_\ell^{loc}(t)]_{\ell\in\mathcal{L}_n}$, based on $\boldsymbol{\mu}_n(t) := [\mu_l(t)]_{l\in\mathcal{L}_n}$ and $s_{b_n}(t)$. We make use of the stabilizing policy $\hat{\boldsymbol{s}}(t)$ of Lemma 3.7.2

for this construction. We define the policy as follows; $\forall l \in \mathcal{L}_n$

$$\mathbb{P}[\hat{s}_l^{loc}(t) = 1 | \boldsymbol{\mu}_n(t), s_{b_n}(t) = 1] = 0 \tag{3.109}$$

$$\mathbb{P}[\hat{s}_l^{loc}(t) = 1 | \boldsymbol{\mu}_n(t), s_{b_n}(t) = 0] = \mathbb{P}[\hat{s}_l(t) = 1 | \boldsymbol{\mu}_n(t), \hat{s}_{b_n}(t) = 0] \tag{3.110}$$

Note that $s_{b_n}(t)$ is a function of $[\{q_{n'}^f(t)\}_{f\in\mathcal{F}}, \boldsymbol{\mu}_{n'}(t)]_{n'\in P_n - \{n\}}$. Therefore from Assumption 2, $\boldsymbol{\mu}_n(t)$ is independent of $s_{b_n}(t)$. Now using (3.110), we have

$$\mathbb{E}[\mu_l(t)\hat{s}_l^{loc}(t) | s_{b_n}(t) = 0] = \sum_{\boldsymbol{\mu}_n(t)} \mu_l(t)\mathbb{P}[\boldsymbol{\mu}_n(t)]\mathbb{P}[\hat{s}_l^{loc}(t) = 1 | \boldsymbol{\mu}_n(t), s_{b_n}(t) = 0] \tag{3.111}$$

$$= \sum_{\boldsymbol{\mu}_n(t)} \mu_l(t)\mathbb{P}[\boldsymbol{\mu}_n(t)]\mathbb{P}[\hat{s}_l(t) = 1 | \hat{s}_{b_n}(t) = 0, \boldsymbol{\mu}_n(t)] \tag{3.112}$$

$$= \mathbb{E}[\mu_l(t)\hat{s}_l(t) | \hat{s}_{b_n}(t) = 0] \tag{3.113}$$

Since the system is stable under the policy $\hat{\boldsymbol{s}}(t)$, the expected arrival rate of packets into link l is ν_l. The expected output rate is $\mathbb{E}[\mu_l(t)\hat{s}_l(t)]$. Hence,

$$\mathbb{E}[\mu_l(t)\hat{s}_l(t)] > \nu_l \tag{3.114}$$

$$\mathbb{P}[\hat{s}_{b_n}(t) = 0]\mathbb{E}[\mu_l(t)\hat{s}_l(t) | \hat{s}_{b_n}(t) = 0] > \nu_l \tag{3.115}$$

$$\mathbb{E}[\mu_l(t)\hat{s}_l(t) | \hat{s}_{b_n}(t) = 0] > \nu_l / \mathbb{P}[\hat{s}_{b_n}(t) = 0] \tag{3.116}$$

Since the queue at backhaul link b_n is stable under $\hat{\boldsymbol{s}}(t)$, we have $\mathbb{E}[\mu_{b_n}(t)s_{b_n}(t)] > \nu_{b_n}$, which implies $\mathbb{P}[\hat{s}_{b_n}(t) = 1] > \nu_{b_n}/\bar{\mu}_{b_n}$, and $\mathbb{P}[\hat{s}_{b_n}(t) = 0] < 1 - \nu_{b_n}/\bar{\mu}_{b_n}$. Hence, it follows from (3.113) and (3.116), that there exists $\delta > 0$

$$\mathbb{E}[\mu_l(t)\hat{s}_l^{loc}(t) | s_{b_n}(t) = 0] = (1 - \nu_{b_n}/\bar{\mu}_{b_n})^{-1}\nu_l + \delta, \ \forall l \in \mathcal{L}_n \tag{3.117}$$

Definition of $W_l(t)$ and the case $s_{b_n}(t) = 1$

For $l \in \mathcal{L}_n$, define $W_l(t) := V_l(t) + \alpha_l V_{b_n}(t)$, where $\alpha_l := \delta/2\nu_{b_n} + \nu_l/(\bar{\mu}_{b_n} - \nu_{b_n})$. From (3.27) for $l \in \mathcal{L}_n$ and b_n, we have

$$W_l(t+1) - W_l(t) = A_r^l(t) - D_n^l(t) + \alpha_l(A_r^{b_n}(t) - D_{p(n)}^{b_n}(t)) \tag{3.118}$$

Note that whenever $s_{b_n}(t) = 1$, we have $s_l(t) = 0, \forall l \in \mathcal{L}_n$. Moreover, b_n is scheduled only when

$\mu_{b_n}(t) = \bar{\mu}_{b_n}$ and $Q_n^l(t) \geq \bar{\mu}_{b_n}$. Hence from (3.118), for any $l \in \mathcal{L}_n$

$$\mathbb{E}[W_l(t+1) - W_l(t)|\boldsymbol{Q}(t), s_{b_n}(t) = 1] = \mathbb{E}[A_r^l(t)] + \alpha_l(\mathbb{E}[A_r^{b_n}(t)] - \bar{\mu}_{b_n}) \tag{3.119}$$

$$= \nu_l + \alpha_l(\nu_{b_n} - \bar{\mu}_{b_n}) \tag{3.120}$$

$$= \nu_l + \left(\delta/2\nu_{b_n} + \nu_l/(\bar{\mu}_{b_n} - \nu_{b_n})\right)(\nu_{b_n} - \bar{\mu}_{b_n}) \tag{3.121}$$

$$= -(\bar{\mu}_{b_n}/\nu_{b_n} - 1)\delta/2 < 0 \tag{3.122}$$

From (3.122), it follows that

$$\mathbb{E}[\sum_{l\in\mathcal{L}_n} Q_n^l(t)\left(W_l(t+1) - W_l(t)\right)|\boldsymbol{Q}(t), s_{b_n}(t) = 1] = -\delta' \sum_{l\in\mathcal{L}_n} Q_n^l(t) \tag{3.123}$$

where $\delta' := (\bar{\mu}_{b_n}/\nu_{b_n} - 1)\delta/2$.

For the case $s_{b_n}(t) = 0$

$$\mathbb{E}[\sum_{l\in\mathcal{L}_n} Q_n^l(t)\left(W_l(t+1) - W_l(t)\right)|\boldsymbol{Q}(t), s_{b_n}(t) = 0] \tag{3.124}$$

$$= \mathbb{E}[\sum_{l\in\mathcal{L}_n} Q_n^l(t)(\nu_l + \alpha_l\nu_{b_n} - D_n^l(t)|\boldsymbol{Q}(t), s_{b_n}(t) = 0] \tag{3.125}$$

$$= \sum_{l\in\mathcal{L}_n} Q_n^l(t)(\nu_l + \alpha_l\nu_{b_n}) - \underbrace{\sum_{l\in\mathcal{L}_n'(t)} Q_n^l(t)\mathbb{E}[D_n^l(t)|\boldsymbol{Q}(t), s_{b_n}(t) = 0]}_{\because\, s_l(t)=0, l\in\mathcal{L}_n - \mathcal{L}_n'(t)} \tag{3.126}$$

Consider the set $\mathcal{L}_n^{(1)}(t) \subset \mathcal{L}_n'(t)$ defined as the set of links $l \in \mathcal{L}_n$ such that $Q_n^l t) \geq \mu_l(t) > 0$. Note that $Q_n^l(t)D_n^l(t) = Q_n^l(t)\mu_l(t)s_l(t)$ for the links $l \in \mathcal{L}_n^{(1)}(t)$. Hence from (3.126), we have

$$\mathbb{E}[\sum_{l\in\mathcal{L}_n} Q_n^l(t)\left(W_l(t+1) - W_l(t)\right)|\boldsymbol{Q}(t), s_{b_n}(t) = 0] \tag{3.127}$$

$$\leq \sum_{l\in\mathcal{L}_n} Q_n^l(t)(\nu_l + \alpha_l\nu_{b_n}) - \sum_{l\in\mathcal{L}_n^{(1)}(t)} Q_n^l(t)\mathbb{E}[\mu_l(t)s_l(t)|\boldsymbol{Q}(t), s_{b_n}(t) = 0] \tag{3.128}$$

$$\begin{aligned} &= \sum_{l\in\mathcal{L}_n} Q_n^l(t)(\nu_l + \alpha_l\nu_{b_n}) - \sum_{l\in\mathcal{L}_n'(t)} Q_n^l(t)\mathbb{E}[\mu_l(t)s_l(t)|\boldsymbol{Q}(t), s_{b_n}(t) = 0] \\ &\quad + \sum_{l\in\mathcal{L}_n'(t)-\mathcal{L}_n^{(1)}(t)} Q_n^l(t)\mathbb{E}[\mu_l(t)s_l(t)|\boldsymbol{Q}(t), s_{b_n}(t) = 0] \end{aligned} \tag{3.129}$$

For the links $l \in \mathcal{L}'_n(t) - \mathcal{L}^{(1)}_n(t)$, either $\mu_l(t) = 0$ or $Q^l_n(t) < \mu_l(t)$. This implies $Q^l_n(t)\mu_l(t)s_l(t) \leq \mu^2_l(t)s_l(t), \forall l \in \mathcal{L}'_n(t) - \mathcal{L}^{(1)}_n(t)$. Hence from (3.129), we have

$$\mathbb{E}[\sum_{l\in\mathcal{L}_n} Q^l_n(t)\left(W_l(t+1) - W_l(t)\right)|\boldsymbol{Q}(t), s_{b_n}(t) = 0] \tag{3.130}$$

$$\leq \sum_{l\in\mathcal{L}_n} Q^l_n(t)(\nu_l + \alpha_l\nu_{b_n}) - \sum_{l\in\mathcal{L}'_n(t)} Q^l_n(t)\mathbb{E}[\mu_l(t)s_l(t)|\boldsymbol{Q}(t), s_{b_n}(t) = 0] + |\mathcal{L}_n|\mu^2_{\max} \tag{3.131}$$

Since the proposed policy maximizes $\sum_{l\in\mathcal{L}'_n(t)} Q^l_n(t)\mu_l(t)s_l(t)$ whenever $s_{b_n}(t) = 0$, we have

$$\mathbb{E}[\sum_{l\in\mathcal{L}'_n(t)} Q^l_n(t)\mu_l(t)s_l(t)|\boldsymbol{Q}(t), s_{b_n}(t) = 0] \geq \mathbb{E}[\sum_{l\in\mathcal{L}'_n(t)} Q^l_n(t)\mu_l(t)\hat{s}^{loc}_l(t)|s_{b_n}(t) = 0]$$

Hence, it follows from (3.131) that

$$\mathbb{E}[\sum_{l\in\mathcal{L}_n} Q^l_n(t)\left(W_l(t+1) - W_l(t)\right)|\boldsymbol{Q}(t), s_{b_n}(t) = 0] \tag{3.132}$$

$$\leq |\mathcal{L}_n|\mu^2_{\max} + \sum_{l\in\mathcal{L}_n} Q^l_n(t)(\nu_l + \alpha_l\nu_{b_n}) - \sum_{l\in\mathcal{L}'_n(t)} Q^l_n(t)\mathbb{E}[\mu_l(t)\hat{s}^{loc}_l(t)|s_{b_n}(t) = 0] \tag{3.133}$$

Note that $\forall l \in \mathcal{L}_n - \mathcal{L}'_n(t)$, either $\mu_l(t) = 0$ or $Q^l_n(t) \leq \mu_l(t)$. This implies $Q^l_n(t)\mu_l(t)\hat{s}_l(t) \leq \mu^2_l(t) \leq \mu^2_{\max}, \forall l \in \mathcal{L}_n - \mathcal{L}'_n(t)$. Therefore, $0 \leq |\mathcal{L}_n|\mu^2_{\max} - \mathbb{E}[\sum_{l\in\mathcal{L}_n-\mathcal{L}'_n(t)} Q^l_n(t)\mu_l(t)\hat{s}^{loc}_l(t)|s_{b_n}(t) = 0]$. Adding this to (3.133), we have

$$\mathbb{E}[\sum_{l\in\mathcal{L}_n} Q^l_n(t)\left(W_l(t+1) - W_l(t)\right)|\boldsymbol{Q}(t), s_{b_n}(t) = 0] \tag{3.134}$$

$$\leq 2|\mathcal{L}_n|\mu^2_{\max} + \sum_{l\in\mathcal{L}_n} Q^l_n(t)(\nu_l + \alpha_l\nu_{b_n}) - \sum_{l\in\mathcal{L}_n} Q^l_n(t)\mathbb{E}[\mu_l(t)\hat{s}^{loc}_l(t)|s_{b_n}(t) = 0] \tag{3.135}$$

It follows from (3.117) and (3.135) that

$$\mathbb{E}[\sum_{l\in\mathcal{L}_n} Q^l_n(t)\left(W_l(t+1) - W_l(t)\right)|\boldsymbol{Q}(t), s_{b_n}(t) = 0] \tag{3.136}$$

$$\leq 2|\mathcal{L}_n|\mu^2_{\max} + \sum_{l\in\mathcal{L}_n} Q^l_n(t)(\nu_l + \alpha_l\nu_{b_n}) - \sum_{l\in\mathcal{L}_n} Q^l_n(t)\left((1 - \nu_{b_n}/\bar{\mu}_{b_n})^{-1}\nu_l + \delta\right) \tag{3.137}$$

$$= 2|\mathcal{L}_n|\mu^2_{\max} - \underbrace{\delta/2\sum_{l\in\mathcal{L}_n} Q^l_n(t)}_{\because\ \alpha_l = \delta/2\nu_{b_n} + \nu_l/(\bar{\mu}_{b_n} - \nu_{b_n})} \tag{3.138}$$

Final steps of the proof

Note that

$$\begin{aligned}&\mathbb{E}[\sum_{l\in\mathcal{L}_n} Q_n^l(t)\left(W_l(t+1)-W_l(t)\right)|\boldsymbol{Q}(t)]\\&=\sum_{i\in\{0,1\}}\mathbb{P}[s_{b_n}(t)=i|\boldsymbol{Q}(t)]\mathbb{E}[\sum_{l\in\mathcal{L}_n} Q_n^l(t)\left(W_l(t+1)-W_l(t)\right)|\boldsymbol{Q}(t),s_{b_n}(t)=i]\end{aligned}\tag{3.139}$$

It follows from (3.138) and (3.123) that

$$\sum_{l\in\mathcal{L}_n} Q_n^l(t)\mathbb{E}[(W_l(t+1)-W_l(t))|\boldsymbol{Q}(t)]\le K_n-\epsilon_n\sum_{l\in\mathcal{L}_n} Q_n^l(t)\tag{3.140}$$

where $K_n=2|\mathcal{L}_n|\mu_{\max}^2$ and $\epsilon_n=\min\{\delta/2,\delta'\}$ □

Proof of Theorem 3.5.1. We use proof by induction. Firstly, $\limsup_{t\to\infty}\sum_{\tau=0}^t\sum_{f\in\mathcal{F}}\mathbb{E}[q_m^f(\tau)]/t<\infty$ is true for $m=r$ from Lemma 3.7.3.

Now suppose $\limsup_{t\to\infty}\sum_{\tau=0}^t\sum_{f\in\mathcal{F}}\mathbb{E}[q_m^f(\tau)]/t<\infty$ for all the nodes m in the path from n to r excluding n, i.e., $P_n-\{n\}$. Now we will show the same is true for $m=n$.

Consider the function $V_n(\boldsymbol{Q}(t)):=\sum_{l\in L_n}W_l^2(t)$, where $W_l(t)$ is defined in Lemma 3.7.4. We consider the conditional drift $\Delta_n(\boldsymbol{Q}(t)):=\mathbb{E}[V_n(\boldsymbol{Q}(t+1))-V_n(\boldsymbol{Q}(t))|\boldsymbol{Q}(t)]$,

$$\Delta_n(\boldsymbol{Q}(t))=\sum_{l\in L_n}\underbrace{\mathbb{E}[(W_l(t+1)-W_l(t))^2|\boldsymbol{Q}(t)]}_{\le\mathbb{E}[(A_r^l(t)+\alpha_l A_r^{b_n}(t))^2]+(1+\alpha_l^2)\mu_{\max}^2}+2\sum_{l\in\mathcal{L}_n}W_l(t)\mathbb{E}[(W_l(t+1)-W_l(t))|\boldsymbol{Q}(t)]\tag{3.141}$$

$$\le K_n'+2\sum_{l\in\mathcal{L}_n}\{W_l(t)-Q_n^l(t)\}\mathbb{E}[W_l(t+1)-W_l(t)|\boldsymbol{Q}(t)]+2\sum_{l\in\mathcal{L}_n}Q_n^l(t)\mathbb{E}[W_l(t+1)-W_l(t)|\boldsymbol{Q}(t)]\tag{3.142}$$

Since $W_l(t+1)-W_l(t)\le A_r^l(t)+\alpha_l A_r^{b_n}(t)$, the middle term is $\le 2\sum_{l\in\mathcal{L}_n}\{W_l(t)-Q_n^l(t)\}(\nu_l+\alpha_l\nu_{b_n})$, which is a linear function of $\boldsymbol{Q}_{p(n)}(t):=\{q_m^f(t)\}_{f\in\mathcal{F},m\in P_n-\{n\}}$. Let $g(\boldsymbol{Q}_{p(n)}(t)):=2\sum_{l\in\mathcal{L}_n}\{W_l(t)-Q_n^l(t)\}(\nu_l+\alpha_l\nu_{b_n})$. The final term is $\le 2K_n-2\epsilon_n\sum_{l\in\mathcal{L}_n}Q_n^l(t)$ from Lemma 3.7.4. Hence,

$$\Delta_n(\boldsymbol{Q}(t))\le K_n'+2K_n+g(\boldsymbol{Q}_{p(n)}(t))-2\epsilon_n\sum_{l\in\mathcal{L}_n}Q_n^l(t)\tag{3.143}$$

Let $K_n'':=K_n'+2K_n$. We have $\mathbb{E}[V_n(\boldsymbol{Q}(t+1))-V_n(\boldsymbol{Q}(t)]\le K_n''+\mathbb{E}[q(\boldsymbol{Q}_{p(n)}(t))]-2\epsilon_n\sum_{l\in\mathcal{L}_n}\mathbb{E}[Q_n^l(t)]$. Summing from $t=0$ to $T-1$,

$$\mathbb{E}[V_n(\boldsymbol{Q}(T))]-\mathbb{E}[V_n(\boldsymbol{Q}(0))]\le TK_n''+\sum_{t=0}^{T-1}\mathbb{E}[g(\boldsymbol{Q}_{p(n)}(t))]-2\epsilon_n\sum_{t=0}^{T-1}\sum_{l\in\mathcal{L}_n}\mathbb{E}[Q_n^l(t)]\tag{3.144}$$

$$\implies\frac{\sum_{t=0}^{T-1}\sum_{l\in\mathcal{L}_n}\mathbb{E}[Q_n^l(t)]}{T}\le\frac{K_n''}{2\epsilon_n}+\frac{\mathbb{E}[V_n(\boldsymbol{Q}(0))]}{2T\epsilon_n}+\frac{\sum_{t=0}^{T-1}\mathbb{E}[g(\boldsymbol{Q}_{p(n)}(t))]}{2T\epsilon_n}\tag{3.145}$$

Hence, we have $\limsup_{T\to\infty} \sum_{t=0}^{T-1} \sum_{l\in\mathcal{L}_n} \mathbb{E}[Q_n^l(t)] < \infty$. By principle of finite induction, this is true for all $n \in \mathcal{N}$. □

Chapter 4

Distributed Resource Allocation and Flow control Algorithms for k-tier HetNets

4.1 Introduction

The 3-tier HetNet model presented in chapter 2 has wide applicability as evident from the scenarios that are modelled under the framework. However, the strong trend in 5G towards increased number of tiers indicates that the future architectures may have more than 3 tiers. For example, integration of terrestrial cellular infrastructure with aerial communication technologies such as high altitudue platforms (HAP) and low altitude platforms (LAP) is being considered for future communication [26, 64]. Under such an integrated setup, a 4-tier network can be formed by HAP, LAP, LTE-macro and LTE-pico cell. Figure. 4.1 (from [64]) shows an example of a 5 tier HetNet with SATCOM at the top. (In practice, SATCOM uses a different spectrum than the other 4 tiers in Figure. 4.1.)

In this chapter, we introduce a K-tier HetNet model, which generalizes the key ideas behind the 3 tier framework introduced in Chapter 2. As in Chapter 2, all the tiers are assumed to share the same spectrum.

However, there are key differences in the models.

- In this chapter, we consider optimization of resource allocation but not user association (unlike Chapter 2 which considered both). The cell association is assumed to be given, e.g., fixed cell-bias values.

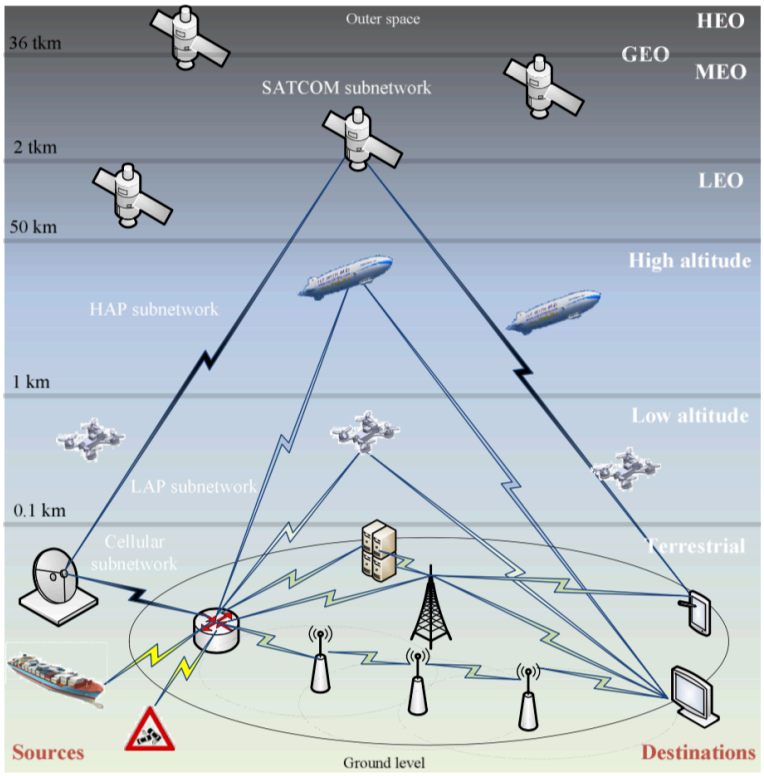


Figure 4.1: An example of a multi tier HetNet [64].

- We extend the framework so that co-tier interference (i.e., interference that occurs between two closely spaced cells in the same tier) can also be managed via resource partitioning, which was not present in Chapter 2.

- We propose a distributed dynamic flow control algorithm for the k-tier HetNet based on the structure of the resource allocation problem.

The main feature of k-tier framework is a novel interference graph model. For an illustration, see Figure. 4.2. In Figure. 4.2, node r is the tier 1 BS, which interferes with the rest of the BSs in the network. This is signified by the edge joining r to H_r. Here, H_r is analogous to the sub-network which is operating in the coverage area of r. It can be noted that there is a graph inside H_r, which models the interference constraints at tier 2. Here nodes n_1, n_2, n_3, n_4 are the tier 2 BSs. For co-tier interference, n_i is joined by an edge to n_j if there is co-tier interference between n_i and n_j. As with H_r, H_{n_i} is analogous to the sub-network which is operating in the coverage area of n_i. The edges connecting n_i to H_{n_j} represent the cross-tier interference caused to the lower tier BSs in the sub-network H_{n_j}.

The minimum resource clearing problem for the network is a LP formulation which involves all

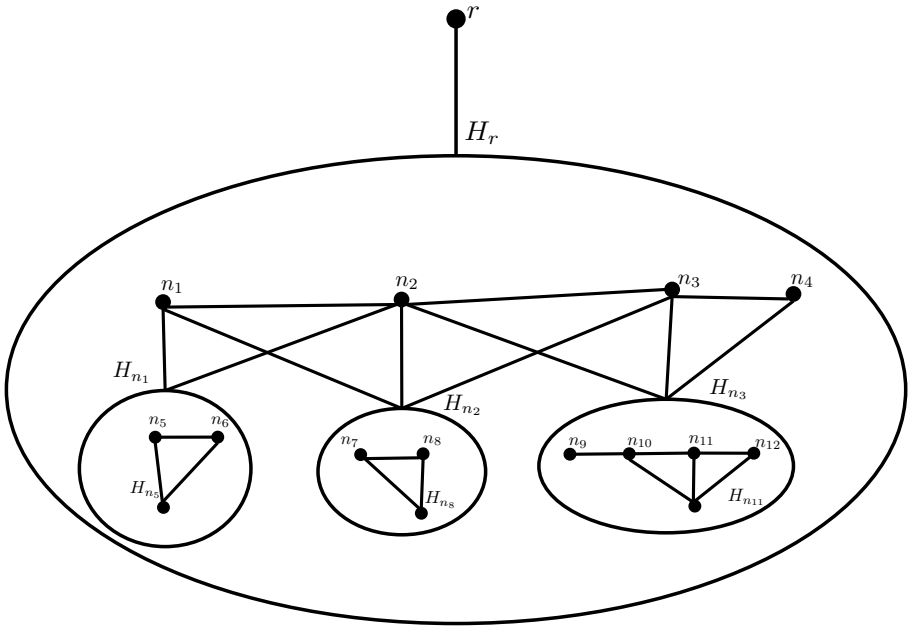


Figure 4.2: Example Graph.

the nodes. The new interference graph allows us to consider the clearing time problem as a series of recursive local formulations at each level. For example in Figure. 4.2, the formulation at r only involves nodes $\{n_1, n_2, n_3, n_4\}$. The formulation at a node only requires the interference relations at the tier level. We show that the framework is scalable in the number of tiers. Furthermore, when there is additional structure in the co-tier graph, we provide resource allocation algorithms which are linear complexity in the size of the network.

4.2 System Model

We consider a K-tier HetNet, with a tier-1 BS r covering a wide area. There are several smaller BSs of different tiers operating in the coverage region of the tier 1 BS r. We review the key abstractions from the 3-tier HeNet framework in chapter 2 in the following. Under the K tier model,

1. The BSs can be divided into tiers based on their coverage area. Generally, the lower tier cells have BSs at higher altitudes, which have larger coverage areas. Several smaller cells (of higher tier) can operate in the coverage area of a lower tier cell.

2. A UE can potentially associate and get service from multiple different tier BSs.

3. Cross-tier interference - a lower tier BS causes debilitating interference to the smaller cells (of higher tier) in its coverage area, if using same resources. i.e., a transmission from BS n of tier i causes interference to the transmissions in a tier j cell in the coverage area of n, where $j > i$. This interference can be avoided by resource partitioning, e.g., by using ABS (Almost Blanking Subframes) scheme in a two tier network.

$\mathcal{R}(r)$ is the set of BSs of tier 2 which are operating in the coverage area of BS r. In general, for a BS n of tier i, $\mathcal{R}(n)$ is the set of BSs of tier $i+1$ which are operating in the coverage area of BS n. For a BS n, $\mathcal{U}(n)$ denotes the set of users associated with n. We represent the HetNet with a rooted tree, (e.g., see Figure. 4.3). We represent the HetNet using the rooted tree $G = (V, E, r)$, where V is the set of all the BSs.

Before proceeding further, we introduce some terminology for rooted trees. Let (V, E, r) denote a rooted tree, where V is the set of nodes, E is the set of edges, and $r \in V$ is the root. Two nodes $n_1, n_2 \in V$ are *neighbors* iff there is an edge connecting n_1 and n_2. For a node $n \in V - \{r\}$, we refer to p as the *parent* of n iff p is the neighbor of n on the path from n to r. Similarly, we refer to n as the *child* of p. Note that the BSs in $\mathcal{R}(n)$ are children of the BS n in the graph representation. If a node a is in the path from n to r, then we refer to a as an *ancestor* of n. Similarly, we refer to n as a *descendant* of a. For $n \in V$, let $D(n)$ denote the set of all the descendants of n in G, and $A(n)$ denote the set of all the ancestors of n in G.

The root r is the tier 1 BS. All the other BSs are the descendants of r in the graph. Similarly, for a BS n in tier i, the set $D(n)$ is all the BSs of tier $j > i$ that are operating the coverage region of n.

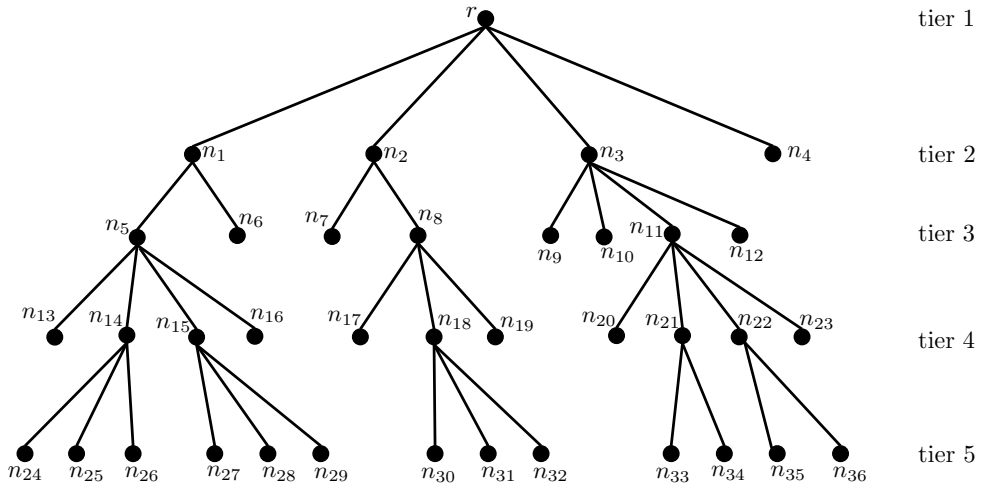


Figure 4.3: Graph Representation.

For example, Figure. 4.3 shows the rooted tree representation of a 5-tier HetNet. The BS r is the root. There are 4 second tier BSs $\{n_i\}_{i=1}^4$, all of which are in the coverage of r. There are 8 BSs in the third tier, each BS here is in the coverage of the parent BS (in second tier). Note that the rooted tree mirrors the hierarchical structure of the HetNet.

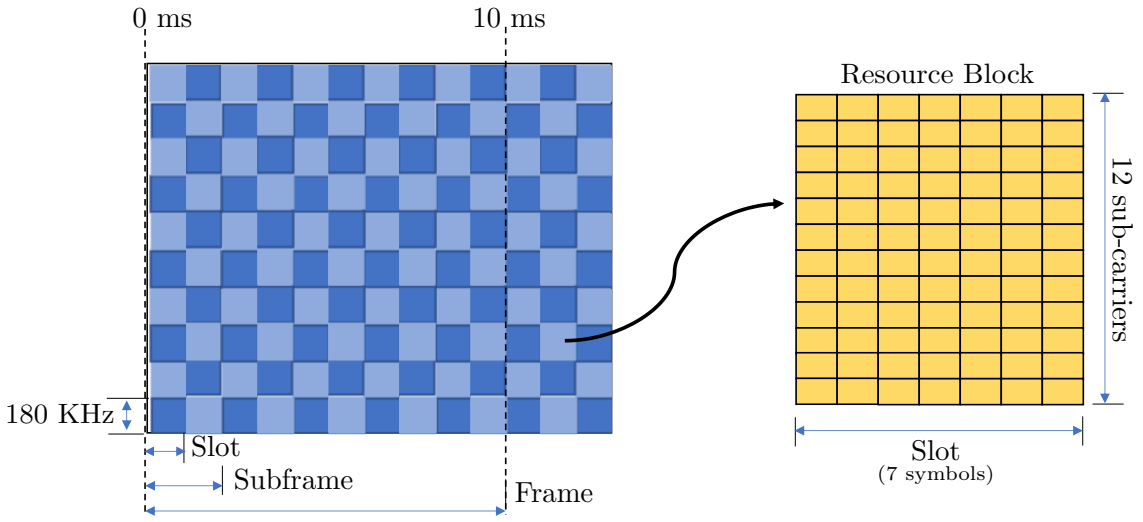


Figure 4.4: LTE Frame structure. Here, each blue square represents a resource block (RB).

We consider a orthogonal resource sharing scheme such as an OFDMA system, with a resource block structure similar to the one present in LTE (see Figure. 4.4). In LTE, each frame is composed of 10 slots, each slot with a length of 1 ms. The available bandwidth is divided into several carriers, each carrier has width 180 KHz. A resource block (RB) is the smallest unit of resource that can be allocated to a link. In LTE, a RB is 180 KHz wide (1 carrier) in frequency and 1ms (1 slot) long in time. Suppose there are N_c carriers available for transmission, then number of RBs in a LTE frame is $N_{RB} := 10N_c$. Under this setup, we consider resource allocation (i.e., allocating RBs) for a downlink HetNet.

4.2.1 Interference constraints

There are two kinds of interference that can occur in the K tier network. 1) Co-tier interference, which is the interference caused by transmissions of BSs in the same tier. 2) Cross-tier interference, which is the interference caused by transmissions of lower tier BSs (which are higher up in the tree G). When the interference is significant, it can cause severe rate degradation to the affected users. In the K

tier model, we consider resource partitioning to avoid simultaneous transmissions of two BSs which will result in significant interference for a user of either BSs. We model these constraints imposed by interference as follows.

1. Co-tier interference - Consider a BS m in tier i, where $i < K$. We model the co-tier interference constraints as follows. For a BS $n \in \mathcal{R}(m)$, $I_c(n) \subseteq \mathcal{R}(m)$ is the set of BSs in tier $i+1$ which cannot be scheduled with BS n on the same RB.

2. Cross-tier interference - We model the cross-tier interference constraints as follows. A BS in $D(n)$ (i.e, a descendant of n) cannot be scheduled with any BS in the set $I_c(n) \bigcup \{n\}$ on the same RB.

Consider the example in Figure. 4.3. We say two nodes are siblings in a rooted tree, if they have the same parent. In Figure. 4.3, $I_c(n_i)$ is the set of adjacent siblings of n_i, i.e., $I_c(n_i) = \{n_{i-1}, n_{i+1}\}$ if both n_{i-1}, n_{i+1} are siblings of n_i, or if n_{i-1} is a sibling of n_i but not n_{i+1}, then $I_c(n_i) = \{n_{i-1}\}$, or if n_{i+1} is a sibling of n_i but not n_{i-1}, then $I_c(n_i) = \{n_{i+1}\}$.

For this example, $I_c(n_1) = \{n_2\}$. Therefore, n_1 and n_2 cannot be scheduled on same RB due to co-tier interference. And, $n_5 \in D(n_1)$, hence n_5 cannot be scheduled along with either n_1 or n_2 due to cross-tier interference.

Under these interference constraints, a BS n cannot be scheduled with any BS in the set $I(n)$, where

$$I(n) := \underbrace{I_c(n)}_{\text{co-tier interference}} \bigcup \underbrace{D(n) \bigcup_{m \in I_c(n)} D(m)}_{\text{cross-tier interference from } n \text{ to higher tiers}} \bigcup \underbrace{A(n) \bigcup_{m \in A(n)} I_c(m)}_{\text{cross-tier interference from lower tiers to } n} \tag{4.1}$$

We say two BSs m, n interfere if they cannot be scheduled together under the interference constraints, i.e., if $n \in I(m)$, $m \in I(n)$.

Resource allocation for the K-tier HetNet model has to allocate RBs to a set of BSs such that the interference constraints are not violated. There can be exponentially many number of such sets with the increase in number of tiers and number of BSs. In this chapter, we provide distributed and scalable resource allocation algorithms for the K-tier HetNet. We consider the minimum clearing resource optimization for the K-tier HetNet model. We will propose distributed resource allocation algorithms based on the solution. We will show that resource allocation for the example in Figure. 4.3 can be found with linear complexity using the algorithms.

4.3 Problem Formulation

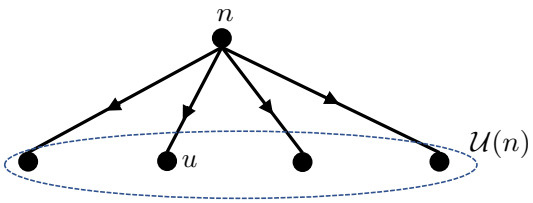


Figure 4.5: Graph showing BS n and its UEs. Here, each edge corresponds to the wireless link between n and a UE in $\mathcal{U}(n)$.

Given that the same RB cannot be allocated to interfering BSs, the rate of the link between BS n and UE $u \in \mathcal{U}(n)$ (given in the following) is calculated provided the BSs in $I(n)$ are muted. The rate of a link l between BS n and a UE $u \in \mathcal{U}(n)$ (in bits/RB) is

$$R_u = W \log_2 \left(1 + \frac{p_n g_{n,u}}{\sum_{m \in I'(n)} p_m g_{m,u} + \sigma^2}\right) \tag{4.2}$$

where $I'(n) := V - I(n)$, and W is the size of an RB in Hz $\times$ sec.

Each UE $u \in \mathcal{U}(n)$ has a throughput *demand* of α_u (in bits/frame). α_u can be interpreted in several ways. For example, it can be 1) The rate of data requests by u (average number of bits arriving per frame), 2) It can be a QOS metric that the user desires, e.g., A user may have a latency requirement of transmitting α_u bits in every frame, 3) It can be flow control metric, being adapted based on congestion in the network. This demand α_u has to be met by allocating the appropriate number of RBs, while adhering to the cross-tier interference constraints. Figure. 4.5 depicts a graph showing the BS n and the UEs in $\mathcal{U}(n)$. In Figure. 4.5, the edges represent wireless downlinks corresponding to the UEs[1].

Given the demand α_u, we can calculate the load of a UE u as $\tau_u := \alpha_u / R_u$ (in RBs/frame). τ_u is the number of RBs which need to be assigned to UE u in each frame. For this setup, we consider the *minimum resource clearing problem*, which is defined as the minimum number of RBs required to satisfy the load τ_u for all $u \in \bigcup_{n \in V} \mathcal{U}(n)$ such that no two cross-interfering BSs are using the same RB. We introduce some definitions and notation that will help with the formulation of the problem.

Definition 4.3.1. *A feasible set of a graph $G' \subset G$ is a set of BSs $S \subset G'$ such that $\forall n \in S, I(n) \bigcap S = \phi$.*

[1]We note that this graph is different from G, where each node of G is a BS, and the edges are used to represent the hierarchy of the different tiered BSs.

Definition 4.3.2. *A maximal feasible set is a feasible set that is not a subset of any other feasible set.*

Let $\mathcal{S}$ denote the set of all maximal feasible sets for the graph G. Let $\tau(n) := \sum_{u \in \mathcal{U}(n)} \tau_u$ denote the aggregate load of a BS $n \in V$. We formulate the minimum resource clearing problem as the following linear program (LP) (4.3).

$$\begin{aligned} &\min \sum_{S \in \mathcal{S}} f_S \\ &\text{s.t.} \\ &\sum_{S: n \in S} f_S \geq \tau(n), \forall n \in G; \\ &f_S \geq 0, \forall S \in \mathcal{S} \end{aligned} \tag{4.3}$$

where f_S is the number of RBs allocated to a feasible set S. By definition, the BSs in a feasible set S can use the allocated RBs simultaneously without violating the interference constraints. The constraints of LP (4.3) ensure that the load $\boldsymbol{\tau}$ (or the demands $\boldsymbol{\alpha}$) is met. It can also be noted that under any allocation, if a RB is allocated to a non-feasible set, the interference constraints will be violated by definition.

4.3.1 Feasibility of the demand vector

Let N_{RB} denote the total number of RBs in a frame, and let $\{f_S^*\}_{S \in \mathcal{S}}$ denote an optimal solution of the LP (4.3).

Definition 4.3.3. *The demand vector $\boldsymbol{\alpha}$ or a load vector $\boldsymbol{\tau}$ is feasible iff $\sum_{S \in \mathcal{S}} f_S^* \leq N_{RB}$*

Therefore, the solution of LP (4.3) can determine whether a given $\boldsymbol{\alpha}$ (or $\boldsymbol{\tau}$) is feasible, and also provides a scheme which satisfies the given demands.

4.4 Scalability of the solution in number of tiers

The direct approach to solving LP (4.3) using a LP solver is laborious and infeasible for large networks, since the number of feasible sets grow exponentially with the number of BSs. In this section, we provide a distributed algorithmic solution which will show that the complexity of the

problem does not grow exponentially in the number of tiers. We introduce necessary terminology for discussion.

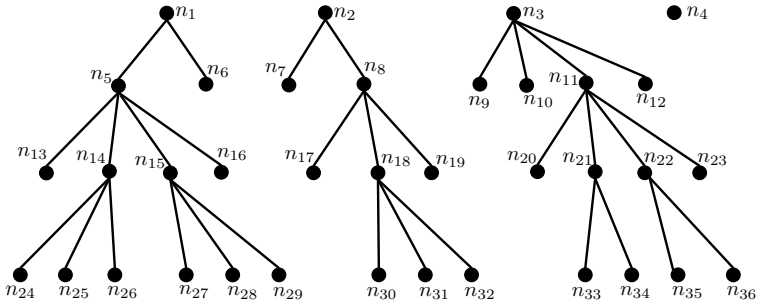


Figure 4.6: HetNet $H[r]$ for the example in Figure. 4.3.

For a BS $n \in V$, let $H[n]$ denote the induced sub-graph of G with the vertex set as the descendants of n (in graph G) i.e., $D(n)$. e.g., Figure. 4.6 shows $H[r]$ for the example considered in Figure. 4.3. $H[n]$ represents the smaller HetNet that is operating in the coverage area of n; it is the network formed by smaller cells in the coverage area of n. Let $\mathcal{S}_{H[n]}$ denote the set of all the maximal feasible sets for the graph $H[n]$. We can formulate the minimum resource clearing problem for the HetNet $H[n]$ as LP (4.4).

$$\begin{aligned} &\min \sum_{S \in \mathcal{S}_{H[n]}} f_S \\ &\text{s.t.} \\ &\sum_{S: m \in S} f_S \geq \tau(m), \forall m \in H[n]; \\ &f_S \geq 0, \forall S \in \mathcal{S}_{H[n]} \end{aligned} \tag{4.4}$$

where f_S is the number of RBs allocated to a feasible set S. Let $\gamma(H[n])$ denote the optimal value of LP (4.4) for the graph $H[n]$. Note that $\gamma(H[n])$ is minimum number of RBs required to satisfy the HetNet represented by $H[n]$.

Lemma 4.4.1. *Consider $p, q \in \mathcal{R}(n)$, where $p \neq q$.*

1. *Any node $p_1 \in H[p]$ does not interfere with any node $q_1 \in H[q]$.*
2. *Suppose q interferes (does not interfere, resp.) with some node $p_1 \in H[p]$, then q interferes (does not interfere, resp.) with each node in $H[p]$.*

Proof. For 1, note that the set of nodes in $H[n]$ that interfere with p_1 are given by $I(p_1) \bigcap D(n)$ (since H[n] is the induced subgraph of G with the vertex set $D(n)$). We will show that $I(p_1) \bigcap D(n)$ does not contain any nodes from $H[q]$.

Recall that for any $p_1 \in H[p]$,

$$I(p_1) = I_c(p_1) \bigcup D(p_1) \bigcup_{m \in I_c(p_1)} D(m) \bigcup A(p_1) \bigcup_{m \in A(p_1)} I_c(m) \tag{4.5}$$

Consider the example in Figure. 4.7. We say two nodes in a rooted tree are *siblings* if they have the same parent. For the example in Figure. 4.7, for each node m, we will take $I_c(m)$ to be the set of siblings of m. An illustration of $I(p_1)$ is given in the Figure. 4.7. Here, $I(p_1)$ is the set of all the circled nodes. The individual components of $I(p_1)$ (in (4.5)) are given in the legend in Figure. 4.7.

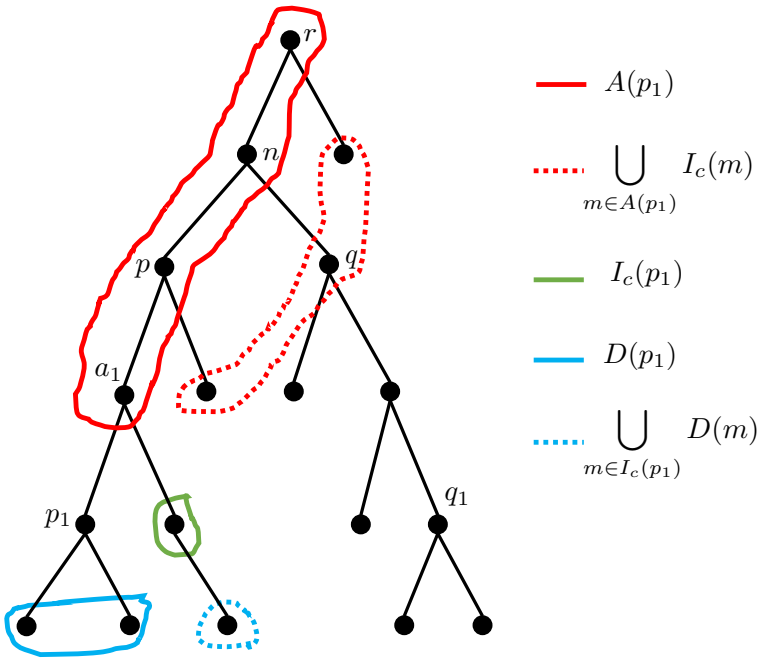


Figure 4.7: Illustration of $I(p_1)$.

Note that since $p_1 \in H[p]$, all the nodes in $I_c(p_1) \bigcup D(p_1) \bigcup_{m \in I_c(p_1)} D(m)$ (In Figure 4.7, these nodes are circled in blue and green) are contained in $D(p)$. Similarly, for all the ancestors a_1 of p_1 in $D(n) - \{p\}$, $\{a_1\} \bigcup I_c(a_1) \subset D(p)$ (In Figure 4.7, a_1 is the parent of p_1 and $I_c(a_1)$ is the sibling of a_1. It can be noted that both a_1 and its sibling are descendants of p). The only remaining ancestor of p_1

in $D(n)$ is p. Hence, it follows that

$$I(p_1) \bigcap D(n) \subseteq D(p) \bigcup \{p\} \bigcup I_c(p) \tag{4.6}$$

$$I(p_1) \bigcap D(n) \subseteq D(p) \bigcup \mathcal{R}(n) \tag{4.7}$$

Therefore, from the nodes in $H[n]$, $I(p_1)$ only contains nodes from $H[p]$ and $\mathcal{R}(n)$. This completes the proof of 1.

For 2, suppose $p_1 \in H[p]$ interferes with $q \in \mathcal{R}(n) - \{p\}$. This implies $q \in I(p_1)$. Note that p is an ancestor of all the nodes in $H[p]$, and since $q \in \mathcal{R}(n)$, it is in the same tier as p. It follows that $q \notin I_c(p_1) \bigcup D(p_1) \bigcup_{m \in I_c(p_1)} D(m)$, since q is in a lower tier than p_1. Similarly, $q \notin A(p_1)$ since p is the ancestor of p_1 in the tier corresponding to q. Hence, it only remains that $q \in I_c(m)$ for some $m \in A(p_1)$. Clearly, here the m must be p. Hence, $q \in I_c(p)$. Now since, p is an ancestor of all the nodes in $H[p]$ and $q \in I_c(p)$, it follows that $q \in I(p_2)$ for each node p_2 in $H[p]$.

Conversely, suppose $p_1 \in H[p]$ does not interfere with $q \in \mathcal{R}(n) - \{p\}$. It must follow that $q \notin I_c(p)$ (Otherwise, it leads to the contradiction that $q \in I(p_1)$ because, $q \in I_c(p)$ and $p \in A(p_1)$). Now, since p is an ancestor of all the nodes in $H[p]$ and $q \notin I_c(p)$, it follows that $q \notin I(p_2)$ for each node p_2 in $H[p]$. □

There is an underlying recursive structure to the K-tier model which will be used to derive a recursive solution to LP (4.4) and to LP (4.3) by extension. We now introduce the terminology and the notion of interference graphs necessary for the discussion.

4.4.1 Recursive Structure of HetNet

Definition 4.4.1. *In the interference graph notation, two nodes are joined by an edge if and only if they interfere.*

Definition 4.4.2. *A leaf node of a tree is a node such that it has no children.*

We use *interference graphs*[2] to illustrate the recursive structure of the HetNet. In the interference graph notation, two nodes are joined by an edge if and only if they interfere or conflict. The edges will be used to model the various interference constraints that occur in the HetNet.

[2]Interference graphs (a.k.a conflict graphs) have been used in the wireless scheduling literature (e.g., [11], [15]) to model the interference constraints that occur in link scheduling, where each node in the interference graph represents a link. Here, we use them in a slightly different manner. The nodes in our model correspond to the BSs in the network.

We define interference graphs H_n corresponding to each HetNet graph H[n]. Note that for a leaf node n in G, there are no descendants, i.e., $D(n) = \phi$. Hence, $H[n]$ does not exist for leaf nodes in G. Similarly, the interference graph H_n does not exist for the leaf nodes n. For the other nodes, we define the interference graphs H_n in (4.8), (4.9). We note that there are two roles for H_n s under this definition. Firstly, the graph definition for H_n is given in (4.8), (4.9). Secondly, $\{H_m\}_{m\in\mathcal{R}(n)}$ act as auxiliary nodes (i.e, nodes not present in G) in the graph definition of H_n (see (4.9)).

1. For a parent node n of leaf nodes (i.e., if $D(n) = \mathcal{R}(n)$),

$$H_n := (\mathcal{R}(n), \phi) \tag{4.8}$$

   where ϕ is the empty set.

2. For a node $n \in G$ such that $D(n) \supset \mathcal{R}(n)$ (i.e., has more than one generation of descendants),

$$H_n := (\mathcal{R}(n) \cup \{H_m\}_{m\in\mathcal{R}(n)}; E_n) \tag{4.9}$$

   where the edge set E_n is the set of edges connecting 1) m to nodes in $I_c(m)$ for each $m \in \mathcal{R}(n)$, 2) m to node H_m for each $m \in \mathcal{R}(n)$, and 3) m to nodes $\{H_k\}_{k\in I_c(m)}$ for each $m \in \mathcal{R}(n)$.

For a node n with at least two generations of descendants, (4.9) provides a recursive definition of graph H_n in which previously defined graphs $\{H_m\}_{m\in\mathcal{R}(n)}$ are nodes. The node H_m corresponds to the HetNet $H[m]$. There are two types of edges in the graph H_n. The first type is the set of edges connecting m to nodes in $I_c(m)$. These model the co-tier interference constraints of node m. The second type is the set of edges which connect the node m to the nodes $\{H_k\}_{k\in\{m\}\bigcup I_c(m)}$, for each node $m \in \mathcal{R}(n)$. These edges model the cross-tier interference from m to the BSs in $H[k]$ (i.e, descendants of k), for each $k \in \{m\} \bigcup I_c(m)$.

For example, consider the graph shown in Figure. 4.3 . The interference graph H_r is shown in Figure. 4.8.

The recursive graph structure shown in Figure. 4.9 captures all the interference constraints that occur in the network given in Figure. 4.3.

Definition 4.4.3. *An independent set J of a graph G' is set of nodes such that no two nodes in J are connected by an edge in G'.*

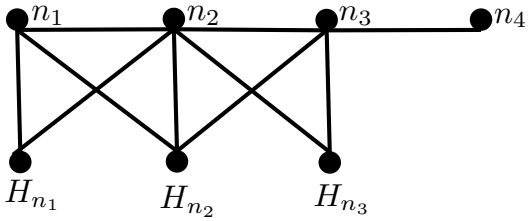


Figure 4.8: Interference graph H_r for the example in Figure. 4.3. $H[r]$ is shown in Figure. 4.6.

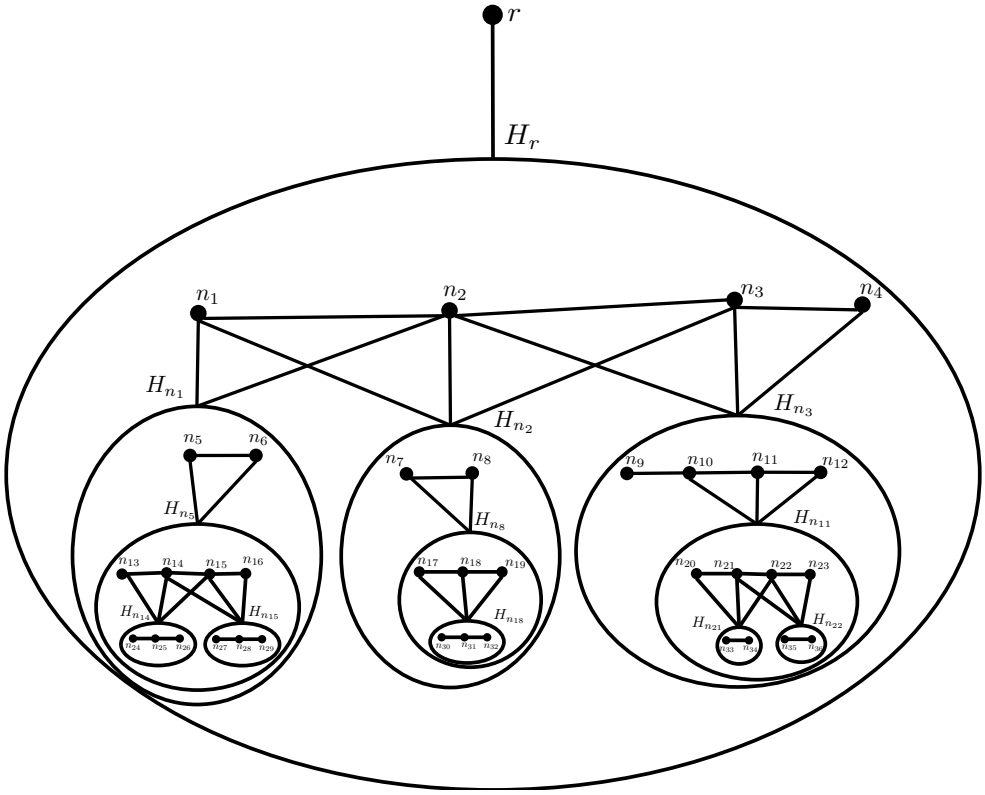


Figure 4.9: Figure illustrating all the interference graphs that occur in Figure. 4.3.

Definition 4.4.4. *A maximal independent set is an independent set that is not a subset of any independent set.*

Let $\mathcal{J}_{G'}$ denote the set of all the maximal independent sets of the graph G', where $G' \in \{H_n\}_{n \in V}$.

4.4.2 A mapping from $\mathcal{S}_{H[n]}$ to $\mathcal{J}_{H_n}$

Lemma 4.4.2. *Consider a maximal feasible S of graph $H[n]$, where $H[n]$ is the induced sub-graph of G with vertex set $D(n)$. Let $S_n := \mathcal{R}(n) \bigcap S$ and for a child m of n, i.e., $m \in \mathcal{R}(n)$, $S_m := H[m] \bigcap S$. The following statements hold true.*

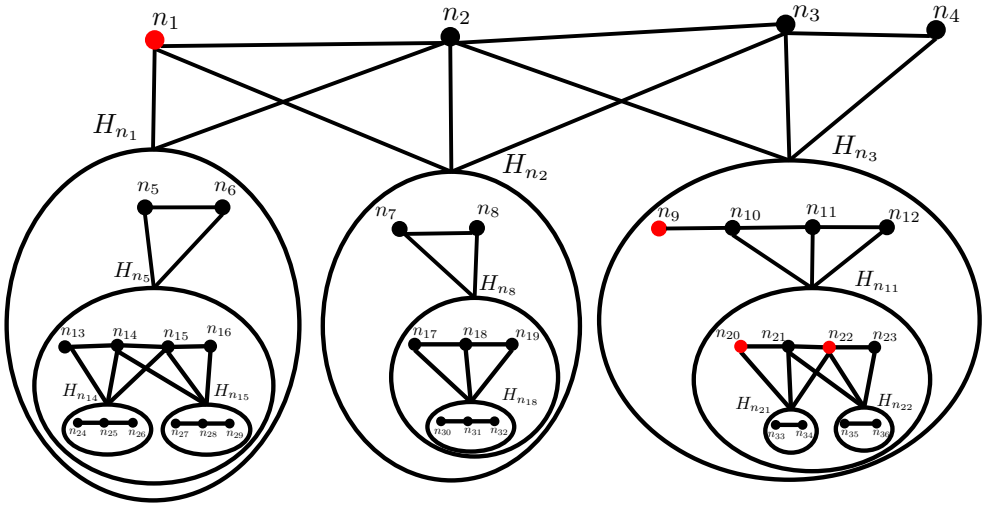
1. $S = S_n \bigcup_{m \in \mathcal{R}(n)} S_m$

2. *Suppose* $S_m \neq \phi$ *for some* $m \in \mathcal{R}(n)$*, then* S_m *is a maximal feasible set of* $H[m]$.

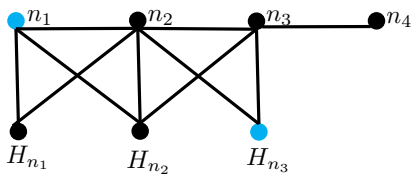
3. J_S *(defined in the following) is a maximal independent set of graph* H_n

$$J_S := S_n \bigcup_{m \in \mathcal{R}(n)} \{H_m : S_m \neq \phi\} \tag{4.10}$$

For an illustration of this Lemma, see Figure. 4.10.



(a) The nodes colored in red form a maximal feasible set S of graph $H[r]$. The graph $H[r]$ is shown in Figure. 4.6



(b) The nodes colored in blue form a maximal independent set J_S of graph H_r. The graph H_r is shown in Figure. 4.8

Figure 4.10: Figure illustrating maximal feasible set S of graph $H[r]$ and the corresponding maximal independent set J_S in graph H_r, for the example in Figure. 4.9. In this example, $S = \{n_1, n_9, n_{20}, n_{22}\}$ and $J_S = \{n_1, H_{n_3}\}$. Also, $S_r = \{n_1\}$, $S_{n_1} = \phi$, $S_{n_2} = \phi$, $S_{n_3} = \{n_9, n_{20}, n_{22}\}$, $S_{n_4} = \phi$.

Proof. **1.**

For 1, note that $H[n]$ is the induced subgraph of G with the vertex set $D(n)$ and also $D(n) = \mathcal{R}(n) \bigcup_{m \in \mathcal{R}(n)} D(m)$. Hence, each node in $H[n]$ is either contained in $\mathcal{R}(n)$ or some $\{H[m]\}_{m \in \mathcal{R}(n)}$. It follows that each node in S is either contained in S_n or in some $\{S_m\}_{m \in \mathcal{R}(n)}$.

2.

To show 2, we use proof by contradiction. Suppose that $S_q \neq \phi$ is not a maximal feasible set of $H[q]$ for some $q \in \mathcal{R}(n)$. Hence, $\exists p \in H[q]$ such that $S_q \bigcup \{p\}$ is a feasible set of $H[q]$.

It follows from Lemma 4.4.1 1. that p does not interfere with any node in S_m, for each $m \in \mathcal{R}(n) - \{q\}$ such that $S_m \neq \phi$. It follows from Lemma 4.4.1 2. that p does not interfere with any node in S_n. Hence, it follows that $S \bigcup \{p\}$ is a feasible set of $H[n]$, which is a contradiction to the assumption that S is a maximal feasible set of $H[n]$.

3.

We will first show that J_S is an independent set of graph H_n, using proof by contradiction. Suppose not, it follows that there exist $p, q \in \mathcal{R}(n)$ such that either a) $p, q \in J_S$, and p, q are connected by an edge. or b) $p, H_q \in J_S$, and p, H_q are connected by an edge.

Suppose a) is true. It follows that $p \in I_c(q)$ and $q \in I_c(p)$. Since $p, q \in S$, this is a contradiction because S is a feasible set.

Suppose the other case b) is true. It follows that $p \in I_c(q)$ and $S_q \neq \phi$. Since q is an ancestor of all the nodes in S_q and $p \in I_c(q)$, it follows that p interferes with all the nodes in S_q. This is a contradiction since $\{p\} \bigcup S_q \subseteq S$ is a feasible set.

Hence, it follows that J_S must be an independent set of H_n.

Now, we will show that J_S is a maximal independent set of H_n using proof by contradiction. Suppose not. There must exist either a $p \in \mathcal{R}(n)$ such that either c) $J_S \bigcup \{p\}$ or d) $J_S \bigcup \{H_p\}$ is an independent set of H_n.

Suppose c) is true. It follows that p is not connected to any of the nodes in S_n, which implies $p \notin I_c(m)$ for each $m \in S_n$. Also, p is not connected to any of the H_m nodes for which $S_m \neq \phi$. Hence, $p \notin I_c(m)$ for any m such that $S_m \neq \phi$ by construction of the interference graph. Therefore, it follows that p does not interfere with any node in $S_n \bigcup_{m \in \mathcal{R}(n)} S_m = S$, which implies that $S \bigcup \{p\}$ is a feasible

set of $H[n]$. This is a contradiction since S is given to be a maximal feasible set.

For the other case, suppose d) is true. It follows that H_p is not connected to any of the nodes in S_n. This implies $p \notin I_c(m)$ for each $m \in S_n$. Consider a maximal feasible set S_p of $H[p]$. Since, $p \notin I_c(m)$ for each m in S_n, any node in S_n does not interfere with any node in S_p. It follows from Lemma 4.4.1.2, that any node in S_p does not interfere with any node in $\bigcup_{m\in\mathcal{R}(n)} S_m$. Hence, it follows that any node in S_p does not interfere with any node in $S_n \bigcup_{m\in\mathcal{R}(n)} S_m = S$. Hence, $S \bigcup S_p$ is a feasible set of $H[n]$, which is a contradiction since S is given to be a maximal feasible set. □

Consider the maximal feasible sets $S_1 = \{n_1, n_9, n_{20}, n_{22}\}$, $S_2 = \{n_1, n_{10}, n_{12}\}$ and $S_3 = \{n_1, n_9, n_{20}, n_{35}\}$ of graph $H[r]$ (for the considered example given in Figure. 4.6). Under the construction given in Lemma 4.4.2, $J_{S_i} = \{n_1, H_{n_3}\}$ for $i = 1, 2, 3$, i.e., all the three maximal feasible sets correspond to the same maximal independent set $\{n_1, H_{n_3}\}$. For an illustration, see Figure. 4.11. Intuitively, the maximal feasible sets in $\mathcal{S}_{H[n]}$ can be grouped into sets corresponding to each maximal independent set in $\mathcal{J}_{H_n}$ using the construction in Lemma 4.4.2. In the following Lemma 4.4.3, we provide a mapping which performs this task. In what follows, we will propose a LP based on the maximal independent sets in $\mathcal{J}_{H_n}$. We will make use of the mapping in Lemma 4.4.3 to establish that this LP has the same optimal value as that of LP (4.4). The new LP is simpler to solve since there are fewer maximal independent sets of H_n than there are maximal feasible sets of $H[n]$.

Lemma 4.4.3. *Consider the mapping $Y : \mathcal{S}_{H[n]} \rightarrow \mathcal{J}_{H_n}$ such that $Y(S) = J_S$, where J_S is defined in (4.10) in Lemma 4.4.2). Let $\mathcal{S}_J \subseteq \mathcal{S}_{H[n]}$ denote the set of maximal feasible sets S which are mapped to J, i.e., $Y(S) = J, \forall S \in \mathcal{S}_J$. Then,*

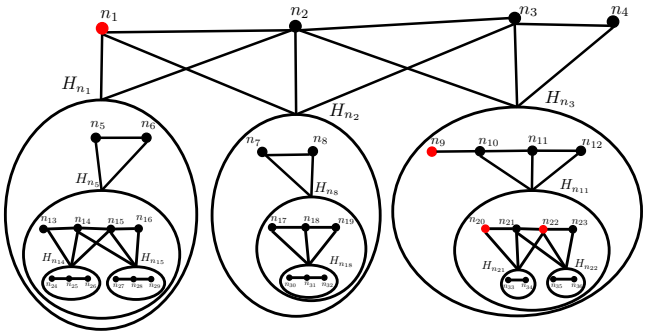
1. $\mathcal{S}_{J_1} \bigcap \mathcal{S}_{J_2} = \phi, \forall J_1 \neq J_2$.
2. $\bigcup_{J\in\mathcal{J}_{H_n}} \mathcal{S}_J = \mathcal{S}_{H[n]}$

Proof. Since $Y(S) = J_S$, it follows from the definition of J_S (4.10) in Lemma 4.4.2 that there exists a unique J_S for each $S \in \mathcal{S}_{H[n]}$. Hence, each S is only in exactly one $\mathcal{S}_J$. This proves 1.

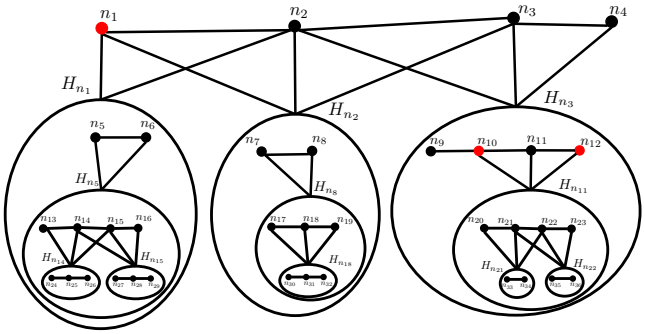
For 2, note that for every $S \in \mathcal{S}_{H[n]}$, it follows from (4.10) in Lemma 4.4.2) that $Y(S)$ exists. Hence, every S must be in a set $\mathcal{S}_J$ for some $J \in \mathcal{J}_{H_n}$. Hence,

$$\bigcup_{J\in\mathcal{J}_{H_n}} \mathcal{S}_J \supseteq \mathcal{S}_{H[n]} \tag{4.11}$$

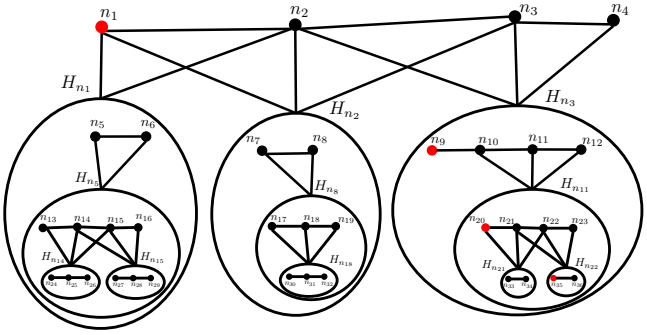
Also since $\mathcal{S}_J \subseteq \mathcal{S}_{H[n]}$ for each J, it follows that $\bigcup_{J\in\mathcal{J}_{H_n}} \mathcal{S}_J \subseteq \mathcal{S}_{H[n]}$. Hence, $\bigcup_{J\in\mathcal{J}_{H_n}} \mathcal{S}_J = \mathcal{S}_{H[n]}$ □



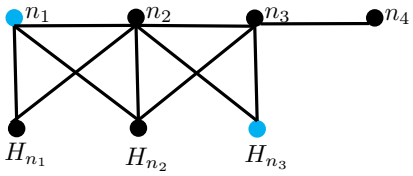
(a) The nodes $\{n_1, n_9, n_{20}, n_{22}\}$ colored in red form a maximal feasible set S_1 of graph $H[r]$.



(b) The nodes $\{n_1, n_{10}, n_{12}\}$ colored in red form a maximal feasible set S_2 of graph $H[r]$.



(c) The nodes $\{n_1, n_9, n_{20}, n_{35}\}$ colored in red form a maximal feasible set S_3 of graph $H[r]$.



(d) The nodes colored in blue form a maximal independent set J of graph H_r. The graph H_r is shown in Figure. 4.8.

Figure 4.11: Figure illustrating the mapping Y of Lemma 4.4.3. The graph $H[r]$ is shown in Figure. 4.6. The maximal feasible sets S_1, S_2, S_3 of $H[r]$ are shown in Figure. 4.11(a) - Figure. 4.11(c). The maximal independent set (of H_r) corresponding to each of the sets $\{S_i\}_{i=1}^{3}$ is the set J shown in Figure. 4.11(d). Hence, $Y(S_i) = J$, for each $i = 1, 2, 3$.

4.4.3 Recursive solution of LP (4.4) using LP (4.12)

Consider LP (4.12), which is a recursive formulation of the resource clearing problem using interference graphs.

$$\begin{aligned} &\min \sum_{J \in \mathcal{J}_{H_n}} f_J \\ &\text{s.t.} \\ &\sum_{J: m \in J} f_J \geq \tau(m), \forall m \in \mathcal{R}(n); \\ &\sum_{J: H_m \in J} f_J \geq \gamma(H[m]), \forall H_m \in H_n; \\ &f_J \geq 0, \forall J \in \mathcal{J}_{H_n} \end{aligned} \tag{4.12}$$

where f_J is the number of RBs allocated to a maximal independent set J, and $\gamma(H[m])$ is the optimal value of LP (4.4).

Theorem 4.4.4. *$\gamma(H[n])$ is the optimal value of LP (4.12) for graph H_n, where $\gamma(H[n])$ was defined to be the optimal value of LP (4.4)*

Proof. See section 4.6 at the end of the chapter. □

4.4.4 Distributed implementation of the recursive solution

Based on Theorem 4.4.4, the recursive solution described in Algorithm 4 can evaluate the optimal minimum resource clearing LP (4.3). Figure. 4.12 depicts a distributed message passing scheme which can be used to implement Algorithm 4.

4.4.5 Upstream Message Passing and Downstream Resource Allocation

4.4.5.1 Upstream message passing

The message passing scheme depicted in Figure 4.12 can be used to evaluate the optimal value of LP (4.3). Each node n in the tree G can obtain the knowledge of $\{\tau(m), \gamma(H[m])\}_{m \in \mathcal{R}(n)}$ via the messages received from the children in $\mathcal{R}(n)$. Using this information, the BS n can evaluate the LP (4.12). Therefore, node n can evaluate the optimal value $\gamma(H[n])$ and a solution which achieves it.

Algorithm 4 Calculating $\gamma(G)$

1: **for** a leaf node $n \in G$ **do**

2: $\gamma(H[n]) = 0$ // since H_n does not exist for a leaf node

3: **end for**

4: **for** a non-leaf node $n \in G$ **do**

5: Formulate LP (4.12) using the values $\{\tau(m)\}_{m\in\mathcal{R}(n)}, \{\gamma(H[m])\}_{m\in\mathcal{R}(n)}$. Solve LP (4.12) to evaluate $\gamma(H[n])$

6: **end for**

return $\gamma(G) = \gamma(H[r]) + \tau(r)$ // where r is the root of the graph G.

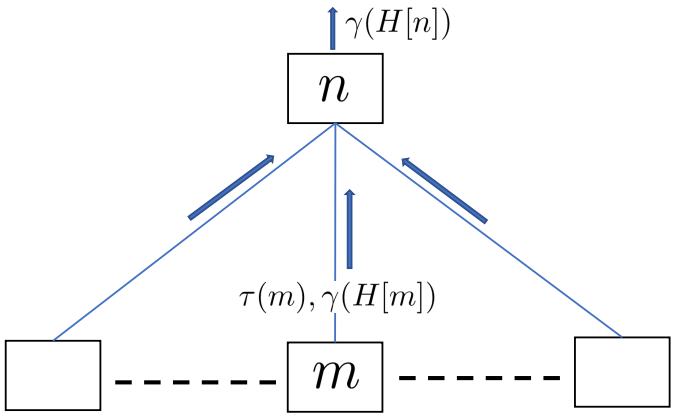


Figure 4.12: Distributed computation of $\gamma(H[n])$. Here, the upstream message to n are sent by the children $m \in \mathcal{R}(n)$.

Hence, minimum clearing time $\gamma(G)$ can be evaluated by applying Algorithm 4, in a distributed manner shown in Figure 4.12.

4.4.5.2 Downstream resource allocation

Given that the upstream message passing phase is completed, an optimal distributed resource allocation can be performed as follows.

The root BS r initiates the downstream allocation by allocating two sets of RBs to each child $m \in \mathcal{R}(r)$. The first set is $\tau(m)$ RBs to satisfy the demand of BS m, and the second set is a separate $\gamma(H[m])$ to satisfy the demand in sub-HetNet $H[m]$. These sets of RBs are allocated according to the solution of LP (4.12) for H_r. Note that the optimal solution of LP (4.12) is known (i.e., evaluated during the upstream phase), and hence the allocation can be done using the minimum number of RBs,

i.e., $\gamma(H[r])$ RBs.

Upon receiving the allocated RBs (from the parent node), a BS n can follow a similar procedure. Using the $H[n]$ RBs provided by its parent, n can allocate two sets of RBs to its children, according to the solution of LP (4.12) for H_n. This scheme is optimal since a total of $\gamma(G)$ RBs are used for allocating to all the nodes, which is optimal value of LP (4.3).

To attain the optimal solution, the RBs must be re-used (in an optimal manner) among the non-interfering BSs, which is done during the downstream allocation phase. The proposed algorithm achieves optimal re-use of RBs (i.e., using minimum possible RBs) based on local decisions, as follows. Based on the solution of LP (4.12), a BS n knows the which RBs to allocate to the BSs in $\mathcal{R}(n)$ and to the sub-HetNets $\{H[m]\}_{m\in\mathcal{R}(n)}$, (from the $\gamma(H[n])$ RBs allocated by the parent). In other words, the BS n knows the re-use at its tier, i.e., among its children and their corresponding sub-HetNets. However, the BS n does not know how the $\gamma(H[m])$ RBs allocated to sub-HetNet $H[m]$ will get used further down the tree. This resource allocation will occur at a later step in the algorithm, as it progresses downstream.

4.5 Complexity of LP (4.12)

So far in the chapter, we have shown that the minimum resource clearing LP (4.4) is scalable in the number of tiers. The distributed algorithm provided in the previous section shows that the complexity of the complete solution is the sum of complexities of the smaller LPs (4.12) (at various tiers).

The complexity of LP (4.12) can also be NP hard (in general). The number of independent sets of graph H_n can grow exponentially in the number of nodes. In this section, we focus on topologies of graph H_n with a certain structure. We will show that a greedy resource allocation algorithm is optimal (i.e., solves LP (4.12)) for these topologies.

We start with a simple example. Consider the special case where $I_c(m) = \phi, \forall m \in \mathcal{R}(n)$, i.e., there are no co-tier interference constraints among nodes in $\mathcal{R}(n)$. Note that now due to cross-tier interference, the graph H_n is made up disjoint sub-graphs. Each sub-graph corresponding to $m \in \mathcal{R}(n)$ is made up of an edge connecting m and H_m. Due to this special structure, we have the following lemma, which simplifies the evaluation of $\gamma(H[n])$.

Lemma 4.5.1. *Suppose $I_c(m) = \phi, \forall m \in \mathcal{R}(n)$, then*

$$\gamma(H[n]) = \max_{m \in \mathcal{R}(n)} (\tau(m) + \gamma(H[m])) \tag{4.13}$$

Proof. It follows from Theorem 4.4.4 that $\gamma(H[n])$ is the solution of LP (4.12). LP (4.12) corresponds to the minimum resource allocation on graph H_n, with load $\tau(m)$ on node m and $\gamma(H[m])$ on node H_m for each $m \in \mathcal{R}(n)$, such that same RB cannot be allocated to nodes joined by an edge. Let $A := \max_{m \in \mathcal{R}(n)}(\tau(m) + \gamma(H[m]))$.

Consider a BS $m \in \mathcal{R}(n)$. Since BS m and H_m are connected by an edge in H_n, they cannot use the same RB. Hence, $\gamma(H[n]) \geq \tau(m) + \gamma(H[m])$. Therefore, $\gamma(H[n]) \geq A$.

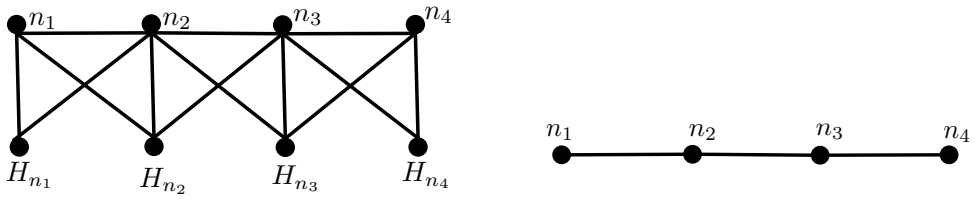
We now provide a feasible allocation for H_n using A RBs. Note that graph H_n is made of independent subgraphs of $\{m, H_m\}$ pairs. Allocate the same A RBs to every $\{m, H_m\}$ pair, and out of which m can get $\tau(m)$ RBs and H_m can get a separate $\gamma(H[m])$ RBS. Therefore, $A \geq \gamma(H[n])$. □

The main idea behind the proof of Lemma 4.5.1 is the following. Given $I_c(m) = \phi, \forall m \in \mathcal{R}(n)$, the graph H_n is composed of disjoint sub-graphs of $\{m, H_m\}$ pairs. Hence, the resource can be re-used in each sub-network, i.e., $\{m, H_m\}$ pair. The resource allocation within each sub-network can be done independently (in parallel). The minimum clearing resource allocation for the graph H_n equals the maximum among the sub-graphs.

The ideas here can be generalized to cases where the disjoint sub-graphs are not $\{m, H_m\}$ pairs. For example, suppose that the graph H_n is composed of two disjoint graphs, G_1 and G_2. The resource can be re-used in each of sub-graphs G_1 and G_2, and the resource allocation within each graph G_1 and G_2 can still be done independently. The minimum clearing resource allocation for the graph H_n equals the maximum among the clearing times for G_1, G_2.

In the following, we consider a more complex topology which has an underlying tree structure. We will show that a greedy resource allocation scheme (with linear complexity in number of nodes) is optimal for this topology. First, we introduce necessary terminology. For the sake of convenience, let $\tau(H_m) := \gamma(H[m])$ for the auxiliary nodes $\{H_m\}_{m \in \mathcal{R}(n)}$ in graph H_n. In this section, we will use $\tau(\cdot)$ to represent the load of a node, (for both a BS node and an auxiliary node). Further, we consider an ordered set of RBs labelled by $\{1, 2, \ldots\}$ for resource allocation.

4.5.1 Co-tier graph



(a) Example network for H_n

(b) H_n^B corresponding to H_n in Figure. 4.13(a)

Figure 4.13: Illustration showing H_n and the corresponding H_n^B.

Consider the induced sub-graph H_n^B of H_n with the vertex set $\mathcal{R}(n)$. The vertex set of graph H_n^B is the set of co-tier BSs $\mathcal{R}(n)$ (i.e., children of BS n). For the edges of H_n^B, each BS $m \in \mathcal{R}(n)$ is connected to each BS in $I_c(m) \subset \mathcal{R}(n)$, i.e, each BS m is connected to the co-tier interfering BSs. For an example, see Figure. 4.13

In this section, we will show that when the graph H_n^B is a tree, a greedy resource allocation algorithm is optimal. First, we introduce the necessary terminology.

We provide the following definition of a feasible resource allocation.

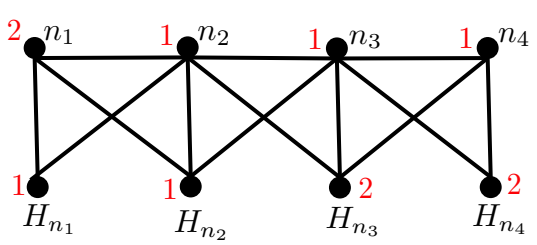
Definition 4.5.1. *Let $\{1,\ldots,T\}$ denote the set of RBs used for allocation, where T is a positive integer. A feasible resource allocation Δ on a graph $G' \in \{H_n\}_{n\in V}$ is a mapping from the set of nodes of G' (say V') to the power set of $\{1,\ldots,T\}$, i.e., $\Delta : V' \rightarrow 2^{\{1,\ldots T\}}$ satisfying the following properties*

1. *$\Delta(m) \subseteq \{1,\ldots,T\}$ is the set of RBs allocated to node m, such that $|\Delta(m)| \geq \tau(m)$ for each $m \in V'$, where $|.|$ denotes the cardinality of a set.*

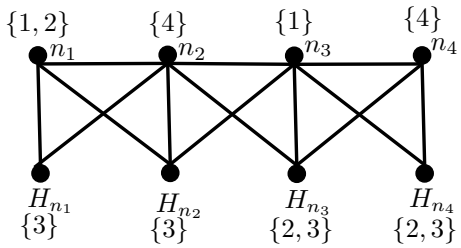
2. *$J_\Delta(t)$ is a independent set of G' for each $t \in \{1,\ldots T\}$, where $J_\Delta(t) := \{m \in V' : t \in \Delta(m)\}$.*

We refer to T as the length of the feasible allocation Δ.

Under a feasible resource allocation Δ, a set $\Delta(m)$ of RBs is allocated to each node $m \in V'$. Each node m gets at least $\tau(m)$ RBs, since $|\Delta(m)| \geq \tau(m)$. Any RB in $\{1,\ldots,T\}$ cannot be allocated to two connected nodes in V', under a feasible allocation, which follows from the condition that $J_\Delta(t)$ is an independent set. Note that a feasible resource allocation of graph H_n can be constructed using a



(a) Example allocation for H_n. The loads are given by the red numbers next to each node



(b) A feasible resource allocation for the example in Figure. 4.14(a). The RBs allocated is represented by the set of RBs next to each node

Figure 4.14: Feasible Resource allocation. The loads are given in Figure. 4.14(a) and the allocation is given in Figure. 4.14(b). Here, the set of $\{1,2,3,4\}$ RBs are used for allocation, i.e., the length of allocation is 4.

feasible solution of LP (4.12) and vice versa (since each RB is allocated to an independent set under a feasible allocation and a feasible solution).

Consider the example given in Figure. 4.14. Figure. 4.14(a) presents the loads on the nodes. Figure. 4.14(b) shows a feasible resource allocation for this setup. Note that under the feasible allocation RB 1 is allocated to nodes $\{n_1, n_3\}$, which is a maximal independent set of H_n. Similarly, RB 2 is allocated to maximal independent set $\{n_1, H_{n_3}, H_{n_4}\}$. RB 3 is allocated to maximal independent set $\{H_{n_i}\}_{i=1}^{4}$. RB 4 is allocated to maximal independent set $\{n_2, n_4\}$. Hence, a feasible solution of LP (4.12) can be constructed using these maximal independent sets. Here, the length of allocation (i.e., number of RBs used for allocation) is equal to the objective value of LP (4.12). A similar process (in reverse) can be applied to construct a feasible resource allocation from a feasible solution of LP (4.12).

Definition 4.5.2. *An optimal resource allocation Δ is a feasible resource allocation for which the length of allocation equals the value of the minimum resource clearing LP (4.12).*

It can be noted that an optimal resource allocation of graph H_n can be constructed using an optimal solution of LP (4.12) and vice versa.

Definition 4.5.3. *A clique C of a graph G' is a set of nodes of graph G' such that either C is a singleton set or a set with at least two nodes such that each node n in C is connected to all the other nodes $C - \{n\}$.*

Note that since every two nodes in a clique C are connected, no two nodes in C can be allocated in the same RB. It follows that at least $\sum_{m\in C}\tau(m)$ RBs are required for a feasible resource allocation. Hence, $\sum_{m\in C}\tau(m)$ provides a lower bound to the value of the minimum clearing LP (4.12) and the length of a feasible resource allocation. Let C denote the set of all the cliques, and $\tau_C^{\max} := \max_{C\in C}\sum_{m\in C}\tau(m)$. It follows that $\tau_C^{\max}$ provides a lower bound to the value of the minimum clearing LP (4.12) and the length of a feasible allocation. e.g., In Figure. 4.14(a), the set of nodes $\{n_1, H_{n_1}, n_2\}$ form a clique. Also, note that at least 4 RBs are required to allocate to the nodes in this clique, which provides a lower bound on the length of a feasible resource allocation.

In the following, we will show that when the graph H_n^B is a tree, the greedy algorithm (given in Algorithm 5) uses exactly $\tau_C^{\max}$ RBs and provides a feasible resource allocation for H_n. It follows that that the algorithm is optimal (since the lower bound given by the cliques is tight).

4.5.2 Greedy Allocation Algorithm

Firstly, we suppose that the graph H_n^B is a tree. We now introduce the necessary terminology and notation. Let $H_n^{B,m}$ denote the tree H_n^B rooted at a node $m \in H_n^B$. The choice of m is arbitrary. Let $\pi(p)$ denote the parent of a node p in the graph $H_n^{B,m}$. There are two types of cliques in graph H_n, corresponding to a node $p \in \mathcal{R}(n)$, 1) $C_p^{(1)}$, 2) $C_p^{(2)}$. These are defined in the following.

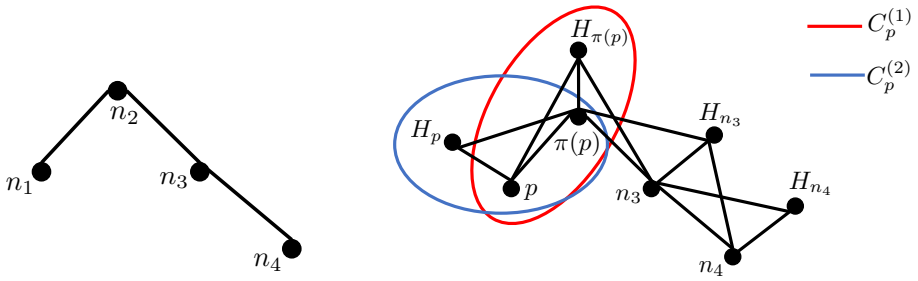
For $p \in \mathcal{R}(n) - \{m\}$, define

$$C_p^{(1)} := \begin{cases} \{\pi(p), H_{\pi(p)}, p\} & \text{if } \pi(p) \text{ is not a leaf node in } G \\ \{\pi(p), p\} & \text{otherwise, i.e., } H_{\pi(p)} \text{ does not exist.} \end{cases} \tag{4.14}$$

$$C_p^{(2)} := \begin{cases} \{\pi(p), p, H_p\} & \text{if } p \text{ is not a leaf node in } G \\ \{\pi(p), p\} & \text{otherwise, i.e., } H_p \text{ does not exist.} \end{cases} \tag{4.15}$$

Figure. 4.15(a) shows the rooted tree H_n^{B,n_2} for the network H_n in Figure. 4.13(a). We take p to be n_1 for illustration. Figure 4.15(b) shows the cliques $C_p^{(1)}$ and $C_p^{(2)}$ for this example. It is straightforwad that $C_p^{(1)}$ and $C_p^{(2)}$ are cliques of the graph H_n.

For a node $p \in H_n^{B,m}$, let $d_n^m(p)$ denote the number of nodes in the path from p to the root m, including m and p. We refer to $d_n^m(p)$ as the depth of node p. For example in Figure. 4.15(a), the depth of nodes n_1, n_3 is 2, and for n_4, the depth is 3. With a slight abuse of notation, let



(a) Rooted tree H_n^{B,n_2} for the example in Figure. 4.13(a).

(b) Figure illustrating cliques $C_p^{(1)}$ and $C_p^{(2)}$ of graph H_n. Here, we take p to be n_1

Figure 4.15: Illustration of cliques of H_n.

$d(H_n^{B,m}) := \max_{p \in H_n^{B,m}} d_n^m(p)$ be the depth of the tree $H_n^{B,m}$. Let V_i denote the set of nodes which are at depth i, i.e., V_i is the set of nodes $p \in H_n^{B,m}$ such that $d_n^m(p) = i$. We present the greedy resource allocation algorithm (in Algorithm 5) for graph H_n given the tree $H_n^{B,m}$.

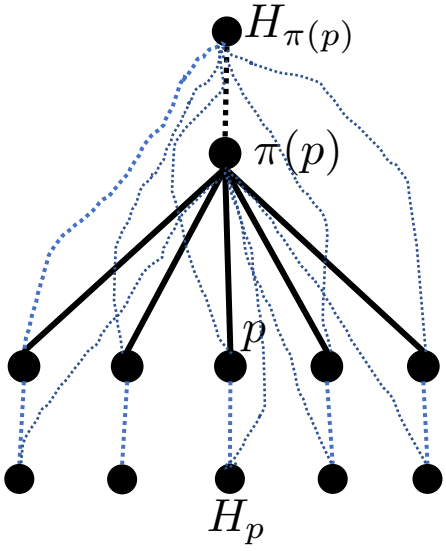


Figure 4.16: Illustration showing the connected nodes of p and H_p during an iteration of Algorithm 5. Here, the links in $H_n^{B,m}$ are shown using solid lines, and the links which are in H_n but not $H_n^{B,m}$ are shown using dotted lines. Both the solid and dotted lines represent interference constraints. Note that p is connected to $\{\pi(p), H_{\pi(p)}, H_p\}$ using solid or dotted lines, and H_p is connected to $p, \pi(p)$ using solid or dotted lines.

Algorithm 5 works as follows. We start at the root node m of tree $H_n^{B,m}$ and work our way down the tree. The resource allocation for the root m and auxiliary node H_m is done in step 1 and step 2 of

Algorithm 5. During iteration i, the algorithms considers the nodes V_i at depth i in $H_n^{B,m}$. Note that in graph $H_n^{B,m}$ a node p is connected to only its parent $\pi(p)$ and its children. We will denote the set of children of node p in $H_n^{B,m}$ by $X_p := \{p' \in H_n^{B,m} : \pi(p') = p\}$. Consequently, in graph H_n, p is only connected to node H_p and both the nodes $\{q, H_q\}$, for each $q \in \{\pi(p)\} \bigcup X_p$. Note that the children $X_p \subseteq V_{i+1}$ are at depth $i+1$, and hence the allocation for these nodes (and the corresponding auxiliary H nodes) is performed in a subsequent iteration. Hence, at step 7 of Algorithm 5, the allocation is available (i.e., done during a previous iteration) only for $\{\pi(p), H_{\pi(p)}\}$ (among the connected nodes). An illustration is given in Figure. 4.16. At step 7 of Algorithm 5, the algorithm avoids the RBs occupied by $\{\pi(p), H_{\pi(p)}\}$ and allocates from the unoccupied RBs. Since $|\Delta^*(q)| = |\tau(q)|$, (i.e, each node q is allocated exactly $\tau(q)$ RBs for $q \in \{\pi(p), H_{\pi(p)}\}$), it follows that there are at least $\tau(p)$ RBs in the set $\{1, \ldots, \sum_{q \in C_p^{(1)}} \tau(q)\} - \bigcup_{q \in \{\pi(p), H_{\pi(p)}\}} \Delta^*(q)$ at step 7 of Algorithm 5.

Similar arguments also apply to allocation for H_p. For H_p, the allocation is done from the set $\{1, \ldots, \sum_{q \in C_p^{(2)}} \tau(q)\} - \bigcup_{q \in \{\pi(p), p\}} \Delta^*(q)$, at step 9 of Algorithm 5. Note that at each step the allocation uses RBs with in the set $\{1, \ldots, \tau_C^{\max}\}$. In Lemma 4.5.2, we show that this property implies that Δ^* is an optimal resource allocation.

An example for allocation under Algorithm 5 is given in Figure. 4.17. Figure. 4.17(a) shows the loads on various nodes in the graph H_n. In Figure. 4.17(b), initialization of allocation (i.e., step 1 and step 3) for nodes n_2, H_{n_2} is given. The blue colored set is the resource block allocation for n_2 and red colored set is the resource block allocation for H_{n_2}. For $i = 2$ in Algorithm 5, $V_2 = \{n_1, n_3\}$. The allocation here is shown in Figure. 4.17(c). For node n_1, the RBs $\{3, 4\}$ are allocated to n_1 in step 7 and the RB $\{2\}$ is allocated to H_{n_1} in step 9 of Algorithm 5. For $i = 3$, the allocation is given in Figure. 4.17(d)

Lemma 4.5.2. *Given that H_n^B is a tree, Algorithm 5 produces an optimal allocation on tree H_n.*

Proof. Firstly, we will show that Δ^* is a feasible resource allocation. Since H_n^B is a tree, each node p in the rooted tree $H_n^{B,m}$ is only connected to its parent and its children (i.e., $I_c(p)$ is the set of parent and children of p in tree $H_n^{B,m}$). Note that the allocation in Algorithm 5 happens top-down (on tree $H_n^{B,m}$) starting from the root m and auxiliary node H_m (in step 1 and step 2 of Algorithm 5 respectively). Hence at step 7 of Algorithm 5, the allocation for the parent $\pi(p)$ and the auxiliary node $H_{\pi(p)}$ must have been completed in a previous iteration, and the allocation children of p (and its auxiliary nodes)

Algorithm 5 Allocation algorithm given a rooted tree $H_n^{B,m}$

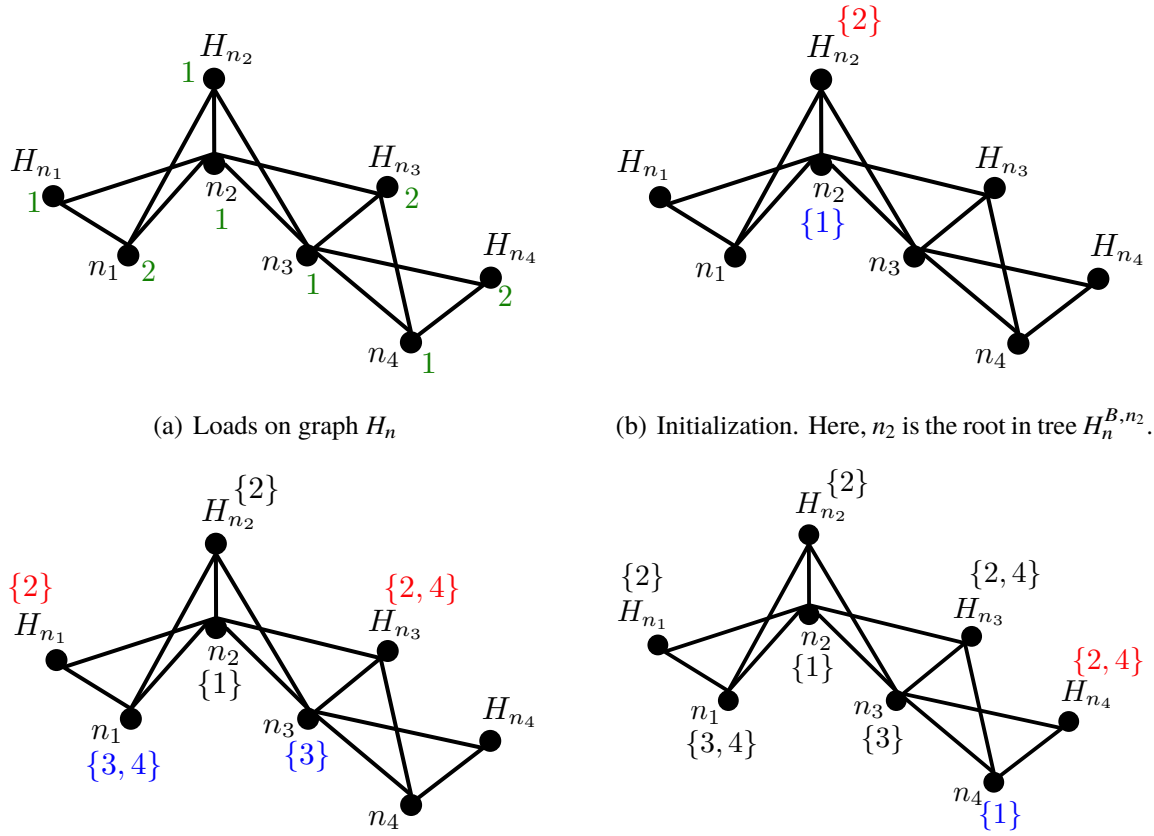
1: Consider a ordered set of RBs labelled using $\mathbb{N}$.
2: $\Delta^*(m) := \{1, \ldots \tau(m)\}$. // Initialization. RB allocation for the root m.
3: **if** m is not a leaf node of G **then** // i.e., if H_m exists,
4: $\Delta^*(H_m) := \{\tau(m)+1, \ldots, \tau(m)+\tau(H_m)\}$ // RB allocation for H_m.
5: **end if**
6: **for** $i = 2$ to $d(H_n^{B,m})$ **do**
7: **for** $p \in V_i$ **do** // V_i is the set of nodes at depth i in tree $H_n^{B,m}$.
8: $\Delta^*(p) :=$ the smallest $\tau(p)$ RBs from the set $\{1, \ldots, \sum_{q \in C_p^{(1)}} \tau(q)\} - \bigcup_{q \in \{\pi(p), H_{\pi(p)}\}} \Delta^*(q)$.
// Allocation for p. There are at least $\tau(p)$ RBs in the considered set since $|\Delta(q)| = \tau(q)$ for $q \in \{\pi(p), H_{\pi(p)}\}$.
9: **if** p is not a leaf node of G **then** // i.e., if H_p exists,
10: $\Delta^*(H_p) :=$ the smallest $\tau(H_p)$ RBs from the set $\{1, \ldots, \sum_{q \in C_p^{(2)}} \tau(q)\} - \bigcup_{q \in \{\pi(p), p\}} \Delta^*(q)$
// Allocation for H_p. There are at least $\tau(H_p)$ RBs in the considered set since $|\Delta(q)| = \tau(q)$ for $q \in \{\pi(p), p\}$.
11: **end if**
12: **end for**
13: **end for**
return Δ^*

is not yet complete (and will be done in a subsequent iteration).

Hence at step 7 of Algorithm 5, the only RBs that are occupied by connected nodes (in graph H_n), $I_c(p), \{H_q\}_{q \in I_c(p)}$ are the RBs allocated to the parent $\pi(p)$ and node $H_{\pi(p)}$, i.e., $\bigcup_{q \in \{\pi(p), H_{\pi(p)}\}} \Delta^*(q)$. Since $|\Delta^*(q)| = \tau(q), \forall q \in \{\pi(p), H_{\pi(p)}\}$ (because each node q is allocated exactly $\tau(q)$ RBs under the algorithm), it follows that there are at least $\tau(p)$ RBs in the set $\{1, \ldots, \sum_{q \in C_p^{(1)}} \tau(q)\} - \bigcup_{q \in \{\pi(p), H_{\pi(p)}\}} \Delta^*(q)$. (We note that there are exactly $\tau(p)$ RBs in the set, if $\bigcup_{q \in \{\pi(p), H_{\pi(p)}\}} \Delta^*(q) \subseteq \{1, \ldots, \sum_{q \in C_p^{(1)}} \tau(q)\}$. Otherwise, if $\Delta^*(q) \not\subseteq \{1, \ldots, \sum_{q \in C_p^{(1)}} \tau(q)\}$ for some $q \in \{\pi(p), H_{\pi(p)}\}$, then there are more than $\tau(p)$ free RBs.)

Similarly, H_p is connected to the nodes in $I_c(p) \bigcup \{p\}$. At step 8 of Algorithm 5, the allocation for only $p, \pi(p)$ (among the connected nodes) has been completed in a previous iteration. Since $|\Delta^*(q)| = \tau(q), \forall q \in \{\pi(p), p\}$ (because each node q is allocated exactly $\tau(q)$ RBs under the algorithm), it follows that there are at least $\tau(H_p)$ RBs in the set $\{1, \ldots, \sum_{q \in C_p^{(2)}} \tau(q)\} - \bigcup_{q \in \{\pi(p), p\}} \Delta^*(q)$.

Hence, each node p gets exactly $\tau(p)$ RBs which do not overlap with the allocation of the parent



(a) Loads on graph H_n

(b) Initialization. Here, n_2 is the root in tree H_n^{B,n_2}.

(c) Allocation for $i = 2$. Here, $V_2 = \{n_1, n_3\}$

(d) Allocation for $i = 3$. Here, $V_3 = \{n_4\}$

Figure 4.17: Figure illustrating resource allocation under Algorithm 5.

$\pi(p)$ or node $H_{\pi(p)}$. And, each node H_p gets $\tau(p)$ RBs which do not overlap with the allocation of $\{p, \pi(p)\}$. Since, this is true for every node p, it follows that condition 1 and condition 2 of Definition 4.5.1 are satisfied. Hence, Δ^* is a feasible resource allocation.

We will now show that Δ^* is an optimal allocation. It follows from step 1, step 3, step 7 and step 8 of Algorithm 5 that the allocated RBs lie in the set $\{1, \ldots, \tau_C^{\max}\}$ (since $\tau_C^{\max} \geq \max\{\sum_{q \in C_p^{(1)}} \tau(q), \sum_{q \in C_p^{(2)}} \tau(q)\}$ for each $p \in \mathcal{R}(n)$). Hence, the length of allocation of Δ^* is less than or equal to $\tau_C^{\max}$. Here,

$$\tau_C^{\max} := \max_{p \in \mathcal{R}(n)} \max_{i \in \{1,2\}} \sum_{q \in C_p^{(i)}} \tau(q)$$

Since $C_p^{(1)}$ and $C_p^{(2)}$ are cliques of graph H_n, it follows that length of any feasible allocation must be greater than or equal to $\max\{\sum_{q \in C_p^{(1)}} \tau(q), \sum_{q \in C_p^{(2)}} \tau(q)$ (because no two nodes in a clique set can be allocated in the same RB). Hence, the length of any feasible resource allocation must be greater than

or equal to $\tau_C^{\max}$. Therefore, Δ^* has the minimum length among all the feasible resource allocations. Hence, it follows that Δ^* is optimal. □

4.6 Theoretical Results

Proof of Theorem 4.4.4. Let $\alpha(H_n)$ denote the optimal value of LP (4.12) for H_n. We complete the proof in two parts. Firstly, we show that $\alpha(H_n) \geq \gamma(H[n])$, and next we show that $\alpha(H_n) \leq \gamma(H[n])$.

1) $\alpha(H_n) \geq \gamma(H[n])$

We will now propose a feasible solution to LP (4.4) for $H[n]$ using the solution of LP (4.12) for H_n.

Note that $\alpha(H_n)$ is the optimal value of LP (4.12). Consider the allocation corresponding to the solution of LP (4.12). Under the solution, $\alpha(H_n)$ RBs are used to allocate $\tau(m)$ RBs to each BS $m \in \mathcal{R}(n)$ and $\gamma(H[m])$ RBs to each auxiliary node H_m. For the BS nodes $m \in \mathcal{R}(n)$, the allocated RBs do not overlap with any of RBs allocated to the co-tier interfering BSs $I_c(m)$ (since the allocation happens over independent sets of graph H_n). Hence, the BS nodes $m \in \mathcal{R}(n)$ can use the allocated RBs without the co-tier interference. In the following paragraph, we will show that cross-tier constraints are also not violated under proposed solution.

Since $\gamma(H[m])$ is defined to be the optimal value of LP (4.4), the $\gamma(H[m])$ RBs allocated to auxiliary node H_m are sufficient for allocating to the BSs in $H[m]$ for each $m \in \mathcal{R}(n)$. We will now show that the RBs allocated to $H[m]$ do not overlap with any interfering BSs in $H[n]$. Among the BSs in $\mathcal{R}(n)$, note that BSs $p \in I_c(m) \subset \mathcal{R}(n)$ are the ones which cause cross-tier interference to BSs in $H[m]$. Since the allocation happens over independent sets of graph H_n, the $\gamma(H[m])$ RBs allocated to auxiliary node H_m do not overlap with the RBs allocated to interfering BSs $p \in I_c(m)$. For the BSs in $\{H[p]\}_{p\in\mathcal{R}(n)-\{m\}}$, it follows from Lemma 4.4.1 1 that any BS in $H[m]$ does not have a interfering BS in $H[p]$. Hence, no interference constraints are violated under the proposed RB allocation corresponding to the solution of LP (4.12).

2) $\alpha(H_n) \leq \gamma(H[n])$

To show $\alpha(H_n) \leq \gamma(H[n])$, we make use of the mapping $Y : \mathcal{S}_{H[n]} \rightarrow \mathcal{J}_{H_n}$ defined in Lemma 4.4.3 (which uses (4.10) in Lemma 4.4.2).

It follows from point 2) of Lemma 4.4.3 that $\mathcal{S}_{H[n]}$ can be expressed as a union of disjoint sets given by $\{\mathcal{S}_J\}_{J\in\mathcal{J}_{H_n}}$. Hence, LP (4.4) for $H[n]$ can be written as following LP (4.16)

$$\begin{aligned}
&\min \sum_{J\in\mathcal{J}_{H_n}} \sum_{S\in\mathcal{S}_J} f_S \\
&\text{s.t.} \\
&\sum_{J:m\in J} \sum_{S\in\mathcal{S}_J} f_S \geq \tau(m), \forall m \in \mathcal{R}(n) \\
&\sum_{S\in\mathcal{S}_{H[n]}:p\in S} f_S \geq \tau(p), \forall p \in H[m]; m \in \mathcal{R}(n) \\
&f_S \geq 0, \forall S \in \mathcal{S}_J, \forall J \in \mathcal{J}_{H_n}
\end{aligned} \tag{4.16}$$

Now consider the following LP (4.17)

$$\begin{aligned}
&\min \sum_{J\in\mathcal{J}_{H_n}} \sum_{S\in\mathcal{S}_J} f_S \\
&\text{s.t.} \\
&\sum_{J:m\in J} \sum_{S\in\mathcal{S}_J} f_S \geq \tau(m), \forall m \in \mathcal{R}(n); \\
&\sum_{J:H_m\in J} \sum_{S\in\mathcal{S}_J} f_S \geq \gamma(H[m]), \forall m \in \mathcal{R}(n); \\
&\sum_{S\in\mathcal{S}_J} f_S \geq 0, \forall J \in \mathcal{J}_{H_n}
\end{aligned} \tag{4.17}$$

Let $\boldsymbol{f} = [f_S]_{S\in\mathcal{S}_{H[n]}}$. Note that any $\boldsymbol{f} \in \mathbb{R}_+^{|\mathcal{S}_{H[n]}|}$ satisfying $f_S \geq 0, \forall S \in \mathcal{S}_{H[n]}$ also satisfies $\sum_{S\in\mathcal{S}_J} f_S \geq 0, \forall J \in \mathcal{J}_{H_n}$.

Since $\gamma(H[m])$ is defined to be the optimal value of LP (4.4) for $H[m]$, it follows that any feasible solution $\boldsymbol{f} \in \mathbb{R}_+^{|\mathcal{S}_{H[n]}|}$ satisfying

$$\sum_{S\in\mathcal{S}_{H[n]}:p\in S} f_S \geq \tau(p), \forall p \in H[m]$$

must also satisfy $\sum_{S\in\mathcal{S}_{H[n]}} f_S \geq \gamma(H[m])$. From Lemma 4.4.3, this is equivalent to

$$\sum_{J:H_m\in J} \sum_{S\in\mathcal{S}_J} f_S \geq \gamma(H[m])$$

Hence, any feasible solution of LP (4.16) (which is equivalent to LP (4.4)) is a feasible solution of LP (4.17). Therefore, the optimal value of LP (4.17) must be no larger than optimal value of LP

(4.16), which equals $\gamma(H[n])$. Note that replacing $\sum_{S \in \mathcal{S}_J} f_S$ with f_J in LP (4.17), immediately yields LP (4.12), i.e., they are equivalent. Hence, $\alpha(H_n) \leq \gamma(H[n])$. □

Chapter 5

Greedy iterative solution to the minimum clearing time problem

5.1 Introduction

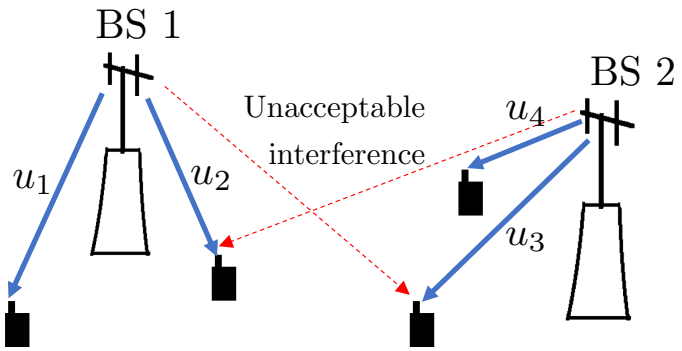


Figure 5.1: Example Network. The solid blue lines are wireless links and the red dotted lines represent the interference (or contention).

Consider the minimum clearing time problem for the following general setup. We consider a set of users $\{u_i\}_{i=1}^N$ with loads $\{\tau(u_i)\}_{i=1}^N \subset \mathbb{N}^N$. Each user u_i has to be allocated $\tau(u_i)$ slots. There are constraints on which users can be scheduled on the same slot defined as follows. A user u_i cannot be scheduled in the same slot as any of the users in the set $I(u_i) \subset \{u_i\}_{i=1}^N$. An example of a network which can be modelled by this setup is given in Figure. 5.1. Here, the users are wireless links $\{u_1, u_2, u_3, u_4\}$. Each BS can schedule one link at a time. There are also constraints due to the interference (shown

using red dotted lines). In Figure. 5.1, a transmission from BS 1 results in too much interference at the receiver of u_3, and a transmission from BS 2 results in too much interference at the receiver of u_2. Hence, in this example, $I(u_1) = \{u_2, u_3\}$, $I(u_2) = \{u_1, u_3, u_4\}$, $I(u_3) = \{u_1, u_2, u_4\}$, and $I(u_4) = \{u_3, u_2\}$.

We consider the minimum clearing time problem for this setup as the minimum number of slots required to satisfy the loads $\tau(u_i)$ for each user u_i such that the scheduling constraints are not violated in any slot. We can formulate the minimum clearing time problem (using the notion of feasible sets as was done in the previous chapters) as LP (5.1).

Definition 5.1.1. *A feasible set is a set of users $S \subset \{u_i\}_{i=1}^{N}$ such that $\forall i \in \{1, \ldots N\}, I(u_i) \bigcap S = \phi$.*

Definition 5.1.2. *A maximal feasible set is a feasible set that is not a subset of any other feasible set.*

$$\begin{aligned} &\min \sum_{S \in \mathcal{S}} f_S \\ &\text{s.t.} \\ &\sum_{S: u_i \in S} f_S \geq \tau(u_i), i \in \{1, \ldots N\}; \\ &f_S \geq 0, \forall S \in \mathcal{S} \end{aligned} \tag{5.1}$$

where $\mathcal{S}$ is the set of all maximal feasible sets, and f_S is the number of slots allocated to a maximal feasible set S.

In the previous chapters, we have provided efficient solutions to the minimum clearing problem for various networks with an underlying structure. The proposed algorithms (in those chapters) required the presence of a central root node (e.g., macro M in Chapter 2, and root node r in Chapter 4), and forward-backward message passing to the central node. In this chapter, we present a more decentralized greedy allocation algorithm which does not require the presence of a root node for implementation. Hence, the algorithm can be implemented in more general topologies (i.e., arbitrary $I(u_i)$ relations), and has a wider applicability. The proposed algorithm only requires local communication of u_i with the nodes in $I(u_i)$.

The algorithm can be described as a *book ahead* slot reservation system. Under the algorithm, a user u_i (updating at time t) is allowed to reserve $\tau(u_i)$ future slots (i.e., $> t$). The choice of slots (i.e., reserved by u_i) is made so that they do not overlap with reserved slots of users in $I(u_i)$. Such slot

reservation schemes for multiple access in wireless networks were considered for Packet Reservation Multiple Access in the literature [65].

We can consider the scheme as a network wide round robin scheduling algorithm, as will be clear when we describe the algorithm in section 5.2. The proposed greedy algorithm produces feasible solutions of LP (5.1) in general. However, we present the following two topologies where the algorithm converges to the optimal solution LP (5.1), in linear time. These are not the only topologies where the algorithm will be optimal. We have not characterized when the algorithm is optimal in this chapter. It is a topic for future research. The following topologies are simply two examples for which we will prove optimality in this chapter.

5.1.1 Topology 1

We consider a tree $G_1 = (V, E)$, where $V = \{u_i\}_{i=1}^{N}$ represents the set of users and E represents the set of edges. Each user $u_i \in V$ has a load of $\tau(u_i) \in \mathbb{N}$ slots. The conflict constraints are modelled as follows. A user u_i and u_j cannot be scheduled in the same slot if they are joined by an edge in the tree G_1. Figure. 5.2 gives an example with 13 users and 12 edges.

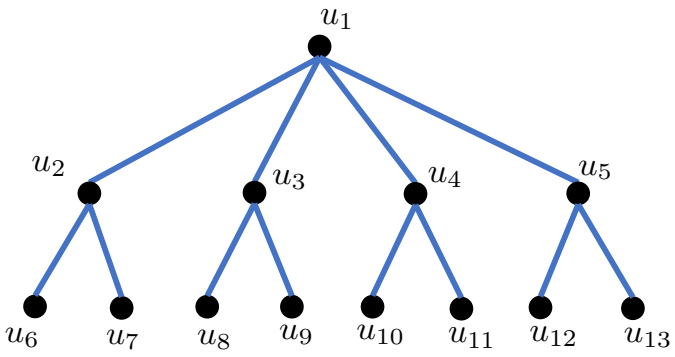


Figure 5.2: Example of Topology 1.

This topology can be used to model various wireless networks, e.g., 1) a tree network under primary interference constraints, 2) a wireless broadcast network where information is disseminated from root to the leaves.

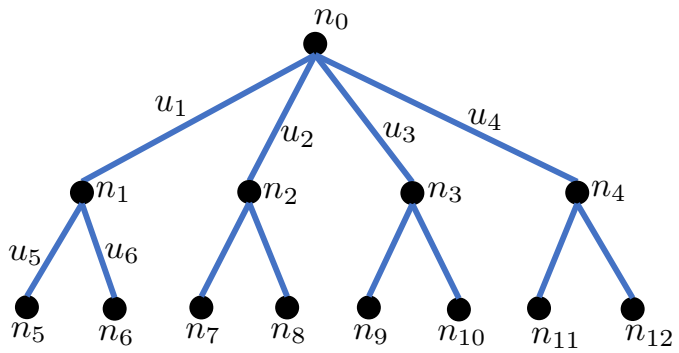


Figure 5.3: Example of Topology 2.

5.1.2 Topology 2

We consider a tree of links denoted by $G_2 = (\mathcal{V}, \mathcal{L})$, where each link corresponds to a user, e.g., see Figure. 5.3. Consider a load of $\tau(u_i) \in \mathbb{N}$ slots on each user $u_i \in \mathcal{L}$. The conflict constraints are modelled as follows. Two links sharing a common node cannot be scheduled together in any slots. Let $I(u_i)$ denotes the set of users that interfere with user u_i, e.g., In Figure. 5.3, $I(u_1) = \{u_2, u_3, u_4, u_5, u_6\}$.

This topology can be used to model a wireless network with relays under half-duplex constraints. For example, consider Figure 5.3. It can be used to represent the following relay network. Let n_0 be a BS with wired backhaul connection, and nodes $\{n_5, \ldots, n_{12}\}$ represent the mobile user equipments (UEs). The nodes $\{n_1, \ldots, n_4\}$ are relay BSs which forward the data from n_0 to the UEs. The links (which we call users) in Figure 5.3 correspond to the wireless links that occur in this network. A possible application of this setup is that of a mmWave IAB network with one RF chain.

Given the hardness of wireless scheduling in general, greedy approaches to scheduling have been considered in the literature. The Max-weight scheduling algorithm is well known to be throughput optimal [1]. However, the implementation involves finding the maximum weighted schedule, which is NP hard in general [1]. Several works in the literature have considered greedy approximations to the problem of finding maximum weighted schedule. The Greedy Maximal Scheduling (GMS) algorithms proposed in [66–69] construct a maximal schedule greedily in polynomial time. GMS was shown to be optimal in networks with special structure (e.g., tree networks under the K-hop interference model) [67].

Similar to how GMS is a greedy approximation to find the maximum weighted schedule, the algorithm that we provide in this chapter is a greedy method to solve the minimum clearing time problem. The algorithm only requires local information and decisions are made distributedly, unlike

in GMS where global knowledge is needed to find the greedy schedule. As mentioned earlier, the algorithm produces feasible solutions of LP (5.1) in general. The algorithm has a monotonic behaviour in the sense that the clearing time can never increase as time progresses. We will show convergence to the minimum clearing time in the considered topologies.

5.2 Algorithm

Let us consider an example with 7 users. Let the loads $[\tau(u_i)]_{i=1}^{N}$ be $[3,4,2,3,4,2,2]$. The constraints are given by $I(u_i) = \{u_{i-1}, u_{i+1}\}$ for $i = 2,\ldots,6$, and $I(u_1) = \{u_2\}, I(u_7) = \{u_6\}$. For this example, we present a feasible allocation at time t, in Table 5.1, where X marks an allocated slot. The algorithm generates feasible allocations of the kind given in Table 5.1. Note that each column in Table 5.1 forms a feasible set. e.g., $\{u_1, u_4, u_7\}$ is the feasible set in slot $t+1$, and $\{u_3, u_5\}$ is the feasible set in slot $t+5$. Before describing the algorithm, we formalize the definition of a feasible allocation and introduce the necessary notation.

Table 5.1: Feasible Allocation.

slot	$t+1$	$t+2$	$t+3$	$t+4$	$t+5$	$t+6$	$t+7$	$t+8$	$t+9$
u_1	X	X	X						
u_2						X	X	X	X
u_3				X	X				
u_4	X	X	X						
u_5					X	X	X	X	
u_6			X	X					
u_7	X	X							

Definition 5.2.1. *A feasible allocation $X(t)$ (at time t) assigns a subset $X_i(t) \subseteq \{t,\ldots t+K-1\}$ (for some positive integer K) of slots to each user u_i for $i = \{1,\ldots,N\}$, such that*

1. *$|X_i(t)| \geq \tau(u_i)$ for each $i \in \{1,\ldots,N\}$, where $|.|$ is the cardinality of a set.*

2. $X_i(t) \bigcap X_j(t) = \phi, \forall u_j \in I(u_i)$, $i \in \{1, \ldots, N\}$.

3. $\bigcup_{i=1}^{N} X_i(t) = \{t, \ldots t + K - 1\}$.

We refer to K as the length of allocation for $X(t)$.

Note that under a feasible allocation, (1) each user u_i gets $\tau(u_i)$ slots (since $|X_i(t)| = \tau(u_i)$). (2) No two conflicting users are scheduled in the same slot (since $X_i(t) \bigcap X_j(t) = \phi, \forall u_j \in I(u_i)$, i.e., there are no overlapping slots between conflicting users). Hence, each column in Table 5.1 is occupied by a feasible set of users. (3) Each slot in $\{t, \ldots, t + K - 1\}$ is allocated to at least one user (since $\bigcup_{i=1}^{N} X_i(t) = \{t, \ldots t + K - 1\}$).

Note that given a feasible allocation, a feasible solution to LP (5.1) can be constructed as follows. For the considered example, $f_S = 2$ for $S = \{u_1, u_4, u_7\}$ since slots $t + 1, t + 2$ are occupied by the maximal feasible set $\{u_1, u_4, u_7\}$. $f_S = 1$ for $S = \{u_1, u_4, u_6\}$ since slot $t + 3$ is occupied by $\{u_1, u_4, u_6\}$. Slot $t + 4$ is occupied by $\{u_3, u_6\}$, which is a feasible set but not a maximal feasible set. It can be made maximal by adding an extra user $\{u_1\}$. Hence, $f_S = 1$ for $S = \{u_1, u_3, u_6\}$ (since slot $t + 4$ are occupied by $\{u_3, u_6\}$ which is a subset of the maximal feasible set $\{u_1, u_3, u_6\}$). $f_S = 1$ for $S = \{u_1, u_3, u_5, u_7\}$, since $t + 5$ is occupied by $\{u_3, u_5\}$ which is a subset of maximal feasible set $\{u_1, u_3, u_5, u_7\}$. $f_S = 4$ for $S = \{u_2, u_5, u_7\}$, since slots $t + 6, t + 7, t + 8$ are occupied by $\{u_2, u_5\}$ and $t + 9$ is occupied by u_2, both of which are subsets of maximal feasible set $\{u_2, u_5, u_7\}$.

Under the above constructed feasible solution of LP (5.1), the objective value equals the length of allocation, which is 9 for this example. Later in the chapter, we define a feasible allocation to be optimal if its length equals the optimal value of LP (5.1). We will present an algorithm (Algorithm 7) that generates optimal allocations for the considered topologies.

later show that for the considered topologies, the algorithm generates optimal allocations.

Definition 5.2.2. *An optimal allocation $X^*(t)$ is a feasible allocation with length equal to the optimal value of LP (5.1).*

We will now describe the algorithm. It has two components, 1) Initialization and 2) Update rule. In initialization, we start with a feasible allocation satisfying certain properties (provided in the following). Once the initial allocation is provided, the update rule provides the slot allocations of each user u_i at each update. Given a initial allocation, the future allocations are fully determined by the update rule. The algorithm provides a map from $\{1, \ldots, \infty\}$ to the set of feasible sets, i.e., in each slot t, a feasible set is scheduled.

5.2.1 Initialization

An initial allocation is a feasible allocation of length $|X(0)|$ slots such that allocation $X_i(0)$ of each user u_i is made up of contiguous slots for each $i = 1, \ldots, N$. Consider the same example with 7 users, with loads $[3, 4, 2, 3, 4, 2, 2]$. Suppose $I(u_i) = \{u_{i-1}, u_{i+1}\}$ for $i = 2, \ldots, 6$, and $I(u_1) = \{u_2\}, I(u_7) = \{u_6\}$. We now present a valid initial allocation for this setup in the following Table 5.2. Here, the slots allocated to a user u_i are marked with X in the corresponding row.

Table 5.2: Initial Allocation.

slot	1	2	3	4	5	6	7	8	9
u_1	X	X	X						
u_2						X	X	X	X
u_3				X	X				
u_4	X	X	X						
u_5					X	X	X	X	
u_6			X	X					
u_7	X	X							

We present a simple rule to generate the initial allocations in Algorithm 6.

Algorithm 6 Initial Allocation

1: $X_i(0) \leftarrow \{0\}$ for $i = 1, \ldots, N$. // Initialization

2: Let $\{\sigma(i)\}_{i=1}^{N}$ be an arbitrary permutation on {1,. . . ,N}.

3: **for** $i = 1$ to N **do**

4: $\quad X_{\sigma(i)} \leftarrow \{t' + 1, \ldots, t' + \tau(u_{\sigma(i)})\}$, where $t' := \max\{\bigcup_{j \in I(u_{\sigma(i)})} X_j(0)\}$

5: **end for**

5.2.2 Update Rule

We present the update rule in Algorithm 7. It can be described as follows. The slots allocated to users expire as the time progresses and new slots have to be allocated. For example, in Table 5.2, u_1 is

allocated slots $\{1,2,3\}$. Hence, after slot 3, u_1 will have no slots remaining from the initial allocation. In the following Algorithm 7, $T_i(t)$ denotes the allocated slots (at time t) which are greater than or equal to t of user u_i. In Table 5.2, $T_1(1) = \{1,2,3\}$ and $T_1(2) = \{2,3\}$. When all the allocated slots expire, i.e., when $T_i(t) - \{t\} = \phi$, the allocation for user u_i is updated. During the update, a new set of slots $T_i(t+1)$ is allocated to user u_i in step 10 of Algorithm 7. The new allocation is chosen greedily as the earliest available contiguous block of slots which are not occupied by the conflicting users in the set $I(u_i)$.

Algorithm 7 Update Rule

1: $t = 1$
2: **for** $i = 1, \ldots N$ **do**
3: $\quad T_i(1) = X_i(0)$
4: **end for**
5: **while** $t \geq 1$ **do**
6: $\quad$ **for** $i = 1, \ldots N$ **do**
7: $\quad\quad$ **if** $T_i(t) - \{t\} \neq \phi$ **then**
8: $\quad\quad\quad T_i(t+1) = T_i(t) - \{t\}$
9: $\quad\quad$ **else** // In this case, t is an update slot for user u_i.
10: $\quad\quad\quad f_i(t) := \inf\{k \in \mathbb{N} : \ k > t, \{k, \ldots, k + \tau(u_i) - 1\} \bigcap \bigcup_{j:u_j \in I(u_i)} T_j(t) = \phi\}$ // We are searching for the earliest non-conflicted block of slots for user u_i starting from slot $t+1$
11: $\quad\quad\quad T_i(t+1) := \{f_i(t), \ldots, f_i(t) + \tau(u_i) - 1\}$
12: $\quad\quad$ **end if**
13: $\quad$ **end for**
14: $\quad t \leftarrow t + 1$
15: **end while**

Let $T(t) = \{T_i(t)\}_{i=1}^{N}$ denote the allocation state at time t. For the considered example in Table 5.2, Table 5.3 shows the allocation state at time $t = 1$. Table 5.4 shows the allocation state at time $t = 2$. Table 5.5 shows the allocation state at time $t = 3$. Note that $T_i(3) - \{3\} = \phi$ for $i = 1, 4$ in Table 5.5, hence $t = 3$ is an update slot for users u_1, u_4. The allocation state at $t = 4$ is shown in Table 5.6. As can be seen, u_1 and u_4 have been allocated new slots $\{10, 11, 12\}$ and $\{9, 10, 11\}$ respectively.

Table 5.3: Allocation state at $t = 1$.

slot	1	2	3	4	5	6	7	8	9
u_1	X	X	X						
u_2						X	X	X	X
u_3				X	X				
u_4	X	X	X						
u_5					X	X	X	X	
u_6			X	X					
u_7	X	X							

Table 5.4: Allocation state at $t = 2$.

slot	2	3	4	5	6	7	8	9
u_1	X	X						
u_2					X	X	X	X
u_3			X	X				
u_4	X	X						
u_5				X	X	X	X	
u_6		X	X					
u_7	X							

Table 5.5: Allocation state at $t = 3$.

slot	3	4	5	6	7	8	9
u_1	X						
u_2				X	X	X	X
u_3		X	X				
u_4	X						
u_5			X	X	X	X	
u_6	X	X					
u_7			X	X			

Table 5.6: Allocation state at $t = 4$.

slot	4	5	6	7	8	9	10	11	12
u_1							X	X	X
u_2			X	X	X	X			
u_3	X	X							
u_4						X	X	X	
u_5		X	X	X	X				
u_6	X								
u_7		X	X						

5.3 Performance and Optimality

It can be observed that a feasible allocation $X(t)$ generated by the Algorithm can be constructed at any time $t > |X(0)|$ by choosing a K large enough such that in the interval $\{t-K+1, \dots t\}$, each user u_i is allocated at least $\tau(u_i)$ slots in the interval. For the considered example, following Table 5.7 shows all the allocated slots in the interval $\{1, \dots, 17\}$. Suppose we want to construct a feasible allocation from the algorithm at time $t = 14$. Note that by choosing $K = 8$ each user u_i has at least $\tau(u_i)$ slots in the interval $\{7, \dots, 14\}$, and hence a feasible allocation can be constructed from the corresponding columns of Table 5.7. In this section, we formalize the concept of choosing K to construct feasible allocations from Algorithm 7.

Table 5.7: Allocation table showing all the allocated slots until $t = 17$.

slot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
u_1	X	X	X							X	X	X					
u_2						X	X	X	X					X	X	X	X
u_3				X	X							X	X				
u_4	X	X	X						X	X	X					X	X
u_5					X	X	X	X				X	X	X	X		
u_6			X	X					X	X						X	X
u_7	X	X			X	X	X	X			X	X	X	X			

We also provide the criterion for optimality of Algorithm 7 i.e., when the constructed feasible allocations are optimal. We explain the key intuition as follows. Recall that a feasible solution of LP (5.1) can be constructed using a feasible allocation, and the length of allocation equals the objective value of LP (5.1) under the constructed solution. For the feasible allocation formed using the columns in interval $\{t - K + 1, \ldots, t\}$, the objective value equals K. We consider the allocation to be optimal if K equals the optimal value of LP (5.1).

First, we introduce the notation necessary for analysis. We define $\alpha_i(t)$ to be the earliest update slot of u_i which is greater than or equal to t as follows.

$$\alpha_i(t) := \max(T_i(t)) \tag{5.2}$$

Recall from Algorithm 7 that for an update slot t, $T_i(t) = \{t\}$. Hence, if t is an update slot of u_i, then $\alpha_i(t) = t$. Otherwise, i.e., if t is not an update slot of u_i, then $\alpha_i(t) > t$.

For $t > |X_i(0)|$, we define $\beta_i(t)$, the latest update slot of u_i which is less than t as follows

$$\beta_i(t) := \sup\{k < t : \alpha_i(k) = k\} \tag{5.3}$$

Definition 5.3.1. *For $t > |X_i(0)|$, we define* inter-update *gap of user u_i as $g_i(t) := \alpha_i(t) - \beta_i(t)$.*

Lemma 5.3.1. *Given $t > |X_i(0)|$, under Algorithm 7, each user u_i gets at least $\tau(u_i)$ allocated slots in the interval $\{t - g_i(t) + 1, \ldots, t\}$, for $i = 1, \ldots, N$.*

Proof. The proof is given in section 5.8. □

From Lemma 5.3.1, each user u_i gets allocated exactly $\tau(u_i)$ slots in the interval $\{t-g_i(t)+1,\ldots,t\}$. It may seem like $g_i(t)$ can be used as a performance metric. However, note that 1) $g_i(t)$ can be different for different users u_i, and 2) $g_i(t)$ remains constant between updates ands changes its value abruptly immediately after an update slot. Due to these two issues, it is problematic to use $g_i(t)$ as a performance metric. We use $K(t)$ (defined in the following) as the performance metric which we use to determine optimality. $K(t)$ satisfies a monotonicity property as given in Lemma 5.3.2.

Definition 5.3.2. *We define $K(t)$ to be the minimum interval length $K \in \mathbb{N}$ such that each user u_i is allocated at least $\tau(u_i)$ slots in the interval $\{t-K+1,\ldots t\}$ under Algorithm 7.*

Note that it follows from Lemma 5.3.1 that

$$K(t) \leq \max_{i=1}^{N} g_i(t),\ \forall t > |X(0)| \tag{5.4}$$

The equality in (5.4) does not hold if there are at least $\tau(u_i)$ slots for each i in the interval $\{t - \max_{i=1}^{N} g_i(t)+2,\ldots,t\}$.

Lemma 5.3.2. *Under Algorithm 7, $K(t+1) \leq K(t)$, $\forall t > |X(0)|$.*

Proof. The proof is given in section 5.8. □

It follows from Lemma 5.3.2 that the allocations produced by the algorithm cannot increase in length as time passes. This property is a general result of Algorithm 7 and applies independently of the topology. Therefore, even though Algorithm 7 may only generate sub-optimal allocations for certain topologies, the solution at any slot t will be as good (in terms of objective value) as the solution at previous slot $t-1$. It follows that the asymptotic limit $K(\infty) = \lim_{t\to\infty} K(t)$ exists. In the asymptotic limit, either the algorithm converges to an optimal allocation, or it may cycle between feasible allocations of a length $K(\infty)$. This follows from there being only finite possibilities for feasible allocations of a given length $K(\infty)$. In the following, we define the criterion for optimality of the algorithm.

Definition 5.3.3. *The algorithm has converged to the optimal solution at time t if and only if $K(t) = \sum_{S\in\mathcal{S}} f_S^*$, where f_S^* is the optimal solution of LP (5.1).*

5.4 Properties of the algorithm

The following sections will show the convergence of Algorithm 7 in the two considered topologies. A common feature in these two topologies is that the *clique bound* is tight for them. In this section, we derive the necessary results regarding Algorithm 7 to show convergence in the considered topologies. We first provide the necessary definitions.

Definition 5.4.1. *A clique set $C \subset \{u_i\}_{i=1}^{N}$ is a set of users such that for any $u_j \in C$, $I(u_j) \supset C$.*

i.e., Every pair of users in a clique set conflict with each other.

Definition 5.4.2. *A maximal clique set is a clique set that is not a subset of any other clique set.*

Let C denote the set of all maximal clique sets, and $\tau_C^{\max} := \max_{C \in C}\{\sum_{u_i \in C} \tau(u_i)\}$. It is clear that since no two users in a clique set C can be scheduled in the same slot, we need at least $\sum_{u_i \in C} \tau(u_i)$ slots for scheduling users in C. Hence, it follows that $\sum_{S \in \mathcal{S}} f_S^* \geq \tau_C^{\max}$, i.e., the minimum clearing time can be bounded by the time to clear the cliques. We say the *clique bound* is tight if and only if $\sum_{S \in \mathcal{S}} f_S^* = \tau_C^{\max}$.

We will show that for the two topologies considered in the following sections, there exists T such that $g_i(t) \leq \tau_C^{\max}$ for each user u_i, $\forall t \geq T$. It follows from (5.4) that $K(T) \leq \tau_C^{\max}$. It follows from Lemma 5.3.1 that feasible allocations can be constructed by considering the interval $\{t-\tau_C^{\max}+1, \ldots, T\}$ for any $t \geq T$. The optimality of the allocations follows from the clique bound. It is also then immediate that the clique bound is tight for the considered topologies. Before proceeding further, we provide the following two lemmas regarding Algorithm 7 which will be pivotal in establishing the optimality of the algorithm for the topologies in the following sections.

Lemma 5.4.1. *Suppose that t is an update slot of an user u_i such that $t > |X_i(0)|$. Then one of the following conditions must hold*

1. *$t - \tau(u_i)$ is an update slot for a user $u_j \in I(u_i)$*
2. *$t - \tau(u_i)$ is an update slot for user u_i*

For an example of Lemma 5.4.1, consider the update slot 12 of u_1 in Table 5.7. Here $\tau(u_1) = 3$. It can be noted that slot $9 = 12 - 3$ is an update slot of $u_2 \in I(u_1)$.

Proof of Lemma 5.4.1. Let $t' = \beta_i(t)$, i.e., t' is the latest update slot of u_i before t. In the Algorithm 7, recall that $f_i(t') = \inf\{k \in \mathbb{N} : \ k > t', \{k, \dots k + \tau(u_i) - 1\} \bigcap \bigcup_{j \in I(u_i)} T_j(t') = \phi\}$ is the first slot of the conflict free block available to u_i. We consider the following two cases.

Case 1:

Suppose $f_i(t') = t' + 1$. During the update at t',

$$\{t' + 1, \dots t' + \tau(u_i)\} \bigcap \bigcup_{j \in I(u_i)} T_j(t') = \phi \tag{5.5}$$

It follows that $T_i(t') = \{t' + 1, \dots, t' + \tau(u_i)\}$, and that the next update slot $t = t' + \tau(u_i)$. This implies $t' = t - \tau(u_i)$. Therefore in this case, $t' = t - \tau(u_i)$ is an update slot for user u_i. Statement 2) of Lemma 5.4.1 holds in this case.

Case 2:

For the other case, suppose that $f_i(t') > t' + 1$. Hence, during the update at t', $f_i(t') - 1$ is a slot occupied by a user (say u_j) in $I(u_i)$. Observe that $f_i(t') + \tau(u_i) - 1 = t$, which implies slot $f_i(t') - 1 = t - \tau(u_i)$ is occupied by u_j but not $f_i(t')$ (since it is occupied by u_i). Hence, we can deduce that $f_i(t') - 1 = t - \tau(u_i)$ is an update slot of user u_j. Statement 1) of Lemma 5.4.1 holds in this case. □

The following Lemma 5.4.2 is the key result which will be used to establish the convergence in the considered topologies. The statement of Lemma 5.4.2 says that if u_i has a large gap (i.e, $> \tau_C^{\max}$) at time t, then there exists $u_j \in I(u_i)$ such that u_j has a large gap at time $t - \tau(u_i)$.

Lemma 5.4.2. *Suppose t is an update slot of user u_i such that $g_i(t) > \tau_C^{\max}$, and that t is sufficiently large. Then there exists $u_j \in I(u_i)$ such that*
a) $t - \tau(u_i)$ is an update slot of u_j.
b) $g_j(t - \tau(u_i)) > \tau_C^{\max}$
c) $t - \tau(u_i) - \tau(u_j)$ is not an update slot of u_i
d) $\exists u_k \in I(u_j) - \{u_i\}$ such that $t - \tau(u_i) - \tau(u_j)$ is an update slot of u_k.

Proof. The proof is given in Section 5.8. □

For an illustration of Lemma 5.4.2, consider the example in Table 5.7. Here, $\tau_C^{\max} = 7$. During the update slot 17 of u_2, the gap $g_2(17) = 17-9 = 8 > \tau_C^{\max}$. It can be noted that, for a), $13 = 17-\tau(u_2)$ is an update slot of user $u_3 \in I(u_2)$. For b), $g_3(13) = 13-5 = 8 > \tau_C^{\max}$. For c), $11 = 17-\tau(u_2)-\tau(u_3)$ is not an update slot of u_2. For d), $11 = 17-\tau(u_2)-\tau(u_3)$ is an update slot of $u_4 \in I(u_3) - \{u_2\}$.

The rest of the sections in the chapter will focus on establishing the convergence of Algoritm 7.We present the intuition behind how Lemma 5.4.2 is applied to prove convergence in G_1 in Topology 1. We use proof by contradiction. Suppose there is a large gap for some u_i at a sufficiently large time t. (The meaning of sufficiently large will be made clear in the following section.) Lemma 5.4.2 can be used to trace the large gap to a user $u_j \in I(u_i)$ at time $t-\tau(u_i)$. Repeating the argument again, the large gap can be traced back to a user $u_k \in I(u_j) - \{u_i\}$ at time $t-\tau(u_i)-\tau(u_j)$. Repeating this argument several times, the large gap can be traced back to a user u_m (along the a path $P_{i,l}$ connecting nodes $\{u_i, u_j, u_k, \ldots, u_l, u_m\}$ with large gaps) at $t' > 0$ such that u_m is a leaf node in G_1. Applying Lemma 5.4.2 for u_m will lead to a contradiction since $I(u_m) - \{u_l\}$ is an empty set.

The arguments in Topology 2 run along the same lines, albeit a bit more involved.

5.5 Convergence in Topology 1

We consider a tree $G_1 = (V, E)$, where $V = \{u_i\}_{i=1}^N$ represents the set of users and E represents the set of edges. Each user $u_i \in V$ has a load of $\tau(u_i) \in \mathbb{N}$ slots. The conflict constraints are modelled as follows. A user u_i and u_j cannot be scheduled in the same slot if they are joined by an edge in the tree G_1. Figure. 5.4 gives an example with 13 users and 12 edges.

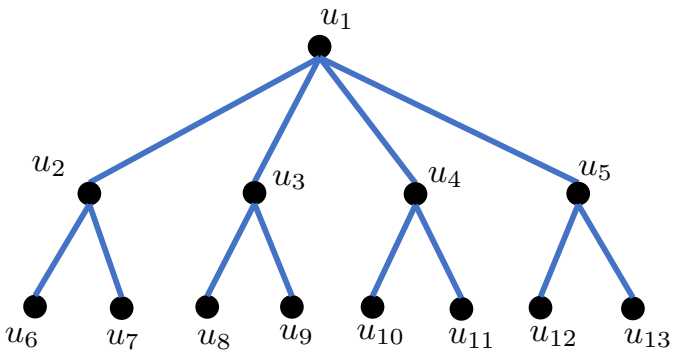
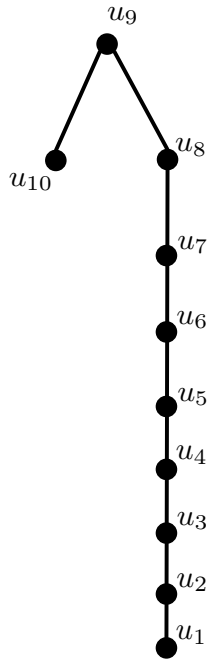


Figure 5.4: Example of Topology 1.

Let $P \subset V$ denote the set of users (i.e., nodes) in a path connecting two nodes of tree G_1. Let

$\tau_P := \sum_{u \in P} \tau(u)$, and $\mathcal{P}$ denote the set of all the paths of the tree. We provide an upper-bound on convergence time of Algorithm 7 (to the optimal solution) in the following Lemma 5.5.1.

An example showing the convergence of the algorithm is provided in Figure. 5.5. The graph is provided in Figure. 5.5(a) and the allocation is provided in Figure. 5.5(b). It can be noted that the length of initial allocation is 3, and the clique bound $\tau_C^{\max}$ equals 2 for the considered example. The algorithm converges after slot 9. It can be noted that $g_i(10) = 2 = \tau_C^{\max}, \forall i = 1, \ldots, 10$.



(a) Graph.

slot	1	2	3	4	5	6	7	8	9	10	11	12
u_1	X		X		X		X		X		X	
u_2		X		X		X		X		X		X
u_3			X		X		X		X		X	
u_4	X			X		X		X		X		X
u_5		X			X		X		X		X	
u_6			X			X		X		X		X
u_7	X			X			X		X		X	
u_8		X			X			X		X		X
u_9			X			X			X		X	
u_{10}	X	X		X	X		X	X		X		X

(b) Convergence.

Figure 5.5: Converge of Algorithm 7 in G_1. Figure. 5.5(b) presents the allocation from $t = 1$ to $t = 12$ for the example in Figure. 5.5(a). Here, $\tau(u_i) = 1$ for each $1 \leq i \leq 10$. The black colored X is used to represent the initial allocation for each u_i. For each u_i, the updated allocations (Xs) are marked with colors red, blue, green and violet alternatively.

Lemma 5.5.1. *For Algorithm 7 running on G_1, $g_i(t) \leq \tau_C^{\max}$ for each $i = 1, \ldots, N$, $\forall t > |X(0)| + \max_{P \in \mathcal{P}} \tau_P$.*

Proof. Lets suppose not, and assume there exists a user u_n and an update slot t such that $g_n(t) > \tau_C^{\max}$ for some $t > |X(0)| + \max_{P \in \mathcal{P}} \tau_P$. We use proof by contradiction in the following. We will apply Lemma 5.4.2, for the users in the path from u_n to a leaf node of G_1.

Consider the rooted version of tree G_1, G_1^n with root as node u_n. Let $c(k)$ denote a child of a node k in the tree G_1^n. For the sake of convenience, let $c^2(u_n) = c(c(u_n))$ denote a child of node $c(u_n)$. In general, let $c^m(u_n) = c(c^{m-1}(u_n))$ denote a child of node $c^{m-1}(u_n)$.

By applying $a), b)$ of Lemma 5.4.2 for user u_n at time t, we have $g_{c(u_n)}(t - \tau(u_i)) > \tau_C^{\max}$ and $t - \tau(u_n)$ is an update slot of $c(u_n)$. Here, $c(u_n)$ is the u_j in statement $a)$ of Lemma 5.4.2. It also follows from statement $c)$ Lemma 5.4.2 that $t - \tau(u_n) - \tau(c(u_n))$ is not an update slot of u_n.

For an illustration of the arguments here, consider the example given in Figure. 5.5. Note (from Figure. 5.5(b)) that at the update slot $t = 9$ of u_9, $g_9(9) = 3 > \tau_C^{\max} = 2$. In this example, u_n is u_9 and $c(u_n)$ is u_8. We note that time t considered in this lemma (which provides an upper-bound for convergence time) is greater than 13. In the example, convergence has occurred at time 10.

Now, by applying $a), b)$ of Lemma 5.4.2 for user $c(u_n)$ at time $t - \tau(u_n)$, we have

$$g_{c^2(u_i)}(t - \tau(u_n) - \tau(c(u_n))) > \tau_C^{\max} \tag{5.6}$$

for some child $c^2(u_n)$ of $c(u_n)$, and $t - \tau(u_n) - \tau(c(u_n))$ is an update slot of $c^2(u_n)$. Here, $c^2(u_n)$ is the u_j in statement $a)$ of Lemma 5.4.2. It also follows from statement $c)$ of Lemma 5.4.2 that $t - \tau(u_n) - \sum_{l=1}^{2} \tau(c^l(u_n))$ is not an update slot of $c(u_n)$. e.g., In Figure. 5.5, u_n is u_9, $c(u(n))$ is u_8 and $c^2(u(n))$ is u_7.

Repeating this process until $c^m(u_n)$ is a leaf node, we have $g_{c^m(u_n)}(t-\tau(u_n)-\sum_{l=1}^{m-1} \tau(c^l(u_n))) > \tau_C^{\max}$.

Now, if we try to apply statement $d)$ of Lemma 5.4.2 in this case, there must exist $u_k \in I(c^m(u_i)) - \{c^{m-1}(u_i)\}$ such that $t - \tau_i - \sum_{l=1}^{m} \tau(c^l(u_i))$ is an update slot of u_k. This is a contradiction since $I(c^m(u_i)) - \{c^{m-1}(u_i)\} = \phi$ for leaf node $c^m(u_i)$. □

5.6 Convergence in Topology 2

Consider a tree of links denoted by $G_2 = (\mathcal{V}, \mathcal{L})$, where each link corresponds to a user, e.g., see Figure. 5.6. Consider a load of $\tau(u_i) \in \mathbb{N}$ slots on each user $u_i \in \mathcal{L}$. The conflict constraints are modelled as follows. Two links sharing a common node cannot be scheduled together in any slots. Let $I(u_i)$ denotes the set of users that interfere with user u_i, e.g., In Figure. 5.6, $I(u_1) = \{u_2, u_3, u_4, u_5, u_6\}$.

Definition 5.6.1. *In G_2, let C_v be the set of all links connected to node $v \in \mathcal{V}$.*

For an example, consier Fig 5.6. Here, $C_{n_0} = \{u_1, u_2, u_3, u_4\}$. Observe that C_v is maximal clique set when node v is not a leaf of the tree. For a clique set C, define $\tau_C = \sum_{u \in C} \tau(u)$.

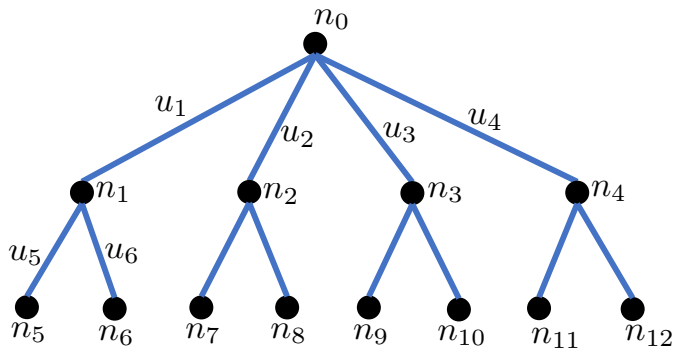


Figure 5.6: Example of Topology 2.

Lemma 5.6.1. *Suppose $g_i(t) > \tau_C^{\max}$ for an update slot $t > |X(0)| + \tau(u_i)$, and that $u_i \equiv (n,m)$ is the link connecting nodes n and m. Then at least one of the following must hold*

1. $\exists u_j \in C_n - \{u_i\} : g_j(t - \tau(u_i)) > \tau_C^{\max}$
2. $\exists u_j \in C_m - \{u_i\} : g_j(t - \tau(u_i)) > \tau_C^{\max}$

In either case, $t - \tau(u_i)$ is an update slot of u_j.

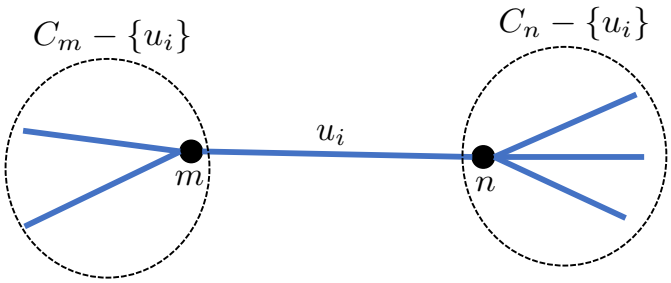


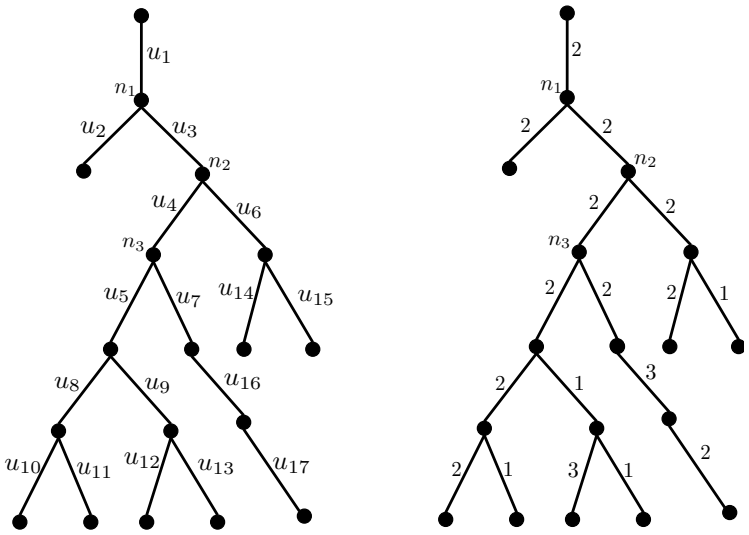
Figure 5.7: Example.

Proof. Observe that $I(u_i) \equiv C_n - \{u_i\} \bigcup C_m - \{u_i\}$ and $C_n - \{u_i\} \bigcap C_m - \{u_i\} = \phi$. (See Figure. 5.7)

By applying *a*), *b*) of Lemma 5.4.2 for user u_i at t, we obtain the result. □

In the following, we present the lemmas which will be used to show the convergence of Algorithm 7 in G_2. An example of convergence in G_2 is provided in Figure. 5.8.

Lemma 5.6.2. *Suppose for $t > \tau_{C_n} + |X(0)|$, there exist links u_i, u_j sharing a common node n such that a) t is an update slot of u_i and $t - \tau(u_i)$ is an update slot of u_j*



(a) Graph is shown on the left, and the loads $\tau(u_i)$ on the right.

slot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
u_1	X	X	X	X					X	X						X	X		
u_2							X	X						X	X				
u_3					X	X						X	X					X	X
u_4			X	X						X	X					X	X		
u_5	X	X						X	X					X	X				
u_6	X	X					X	X						X	X				
u_7						X	X					X	X					X	X
u_8			X	X						X	X					X	X		
u_9					X							X						X	
u_{10}						X	X					X	X					X	X
u_{11}	X				X			X	X					X	X				
u_{12}		X	X	X			X	X	X				X	X	X				X
u_{13}	X					X				X	X					X	X		
u_{14}				X	X				X	X		X	X				X	X	
u_{15}			X			X					X					X			X
u_{16}			X	X	X			X	X	X				X	X	X			
u_{17}	X	X				X	X				X	X					X	X	

(b) Allocation convergence. Here, (black colored) Xs are used to represent the initial allocation for each u_i. We note that the initial allocation is generated by Algorithm 6 using the permutation $[1, 5, 4, 3, 2, 6, 15, 14, 17, 16, 7, 8, 9, 11, 10, 13, 12]$ on users. For each u_i, the updated allocations (Xs) are colored red, blue and green alternatively. Algorithm 7 has converged at $t = 17$ since $K(17) = 6 = \tau_C^{\max}$. Also, note that $K(8) = 8$, $K(9) = 7$ and $K(16) = 7$.

Figure 5.8: Convergence of Algorithm 7 in G_2.

b) $g_i(t) > \tau_C^{\max}$

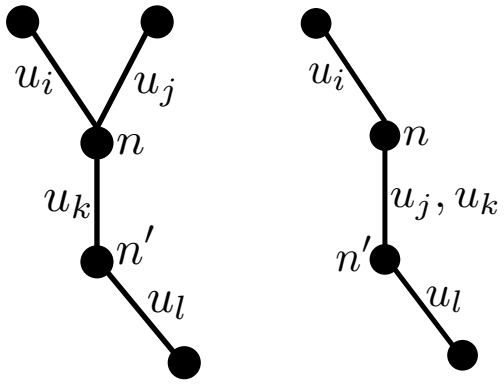
c) $g_j(t - \tau(u_i)) > \tau_C^{\max}$

Then, there exists time $t' \geq t - \tau_{C_n}$ *and links* u_k, u_l *sharing a common node* n' *such that*

i) $u_k \in C_n - \{u_i\} : g_k(t' + \tau(u_k)) > \tau_C^{\max}$

ii) $u_l \in C_{n'} - \{u_k\} : g_l(t') > \tau_C^{\max}$

iii) $(n, n') \equiv u_k$



(a) Scenario 1. (b) Scenario 2.

Figure 5.9: Possible scenarios that can occur in Lemma 5.6.2.

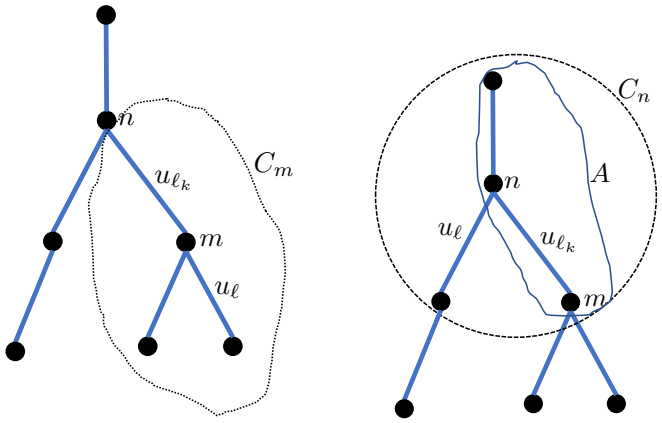
Before we provide the proof, we present an illustrative example of Lemma 5.6.2. Consider the example in Figure. 5.8. Here, $\tau_C^{\max} = 6$. Note that $g_1(t) = 7, g_2(t-2) = 7$ for u_1, u_2 at $t = 17$. Here, $i = 1, j = 2$ and $n = n_1$. It can be noted that $g_3(t' + 2) = 7, g_4(t') = 7$. Here, Lemma 5.6.2 holds for $k = 3, l = 4, t' = t - 6 = 11$ and $n' = n_2$. Here, the scenario shown in Figure. 5.9(a) occured.

For another example in Figure. 5.8, consider $g_3(t) = 7, g_4(t-2) = 7$ for u_3, u_4 at $t = 13$. Here, $i = 3, j = 4$ and $n = n_2$. Observe that $g_4(t' + 2) = 7, g_5(t') = 7$. Here, Lemma 5.6.2 holds for $k = j = 4, l = 5, t' = 9$ and $n' = n_3$. Here, the scenario shown in Figure. 5.9(b) occured.

Proof. Before starting the proof, we make some preliminary definitions. Let $\{\ell_i\}_{i=0}^{N-1}$ be a permutation of $\{1, \ldots N\}$.

Given $\ell_k : u_{\ell_k} \in C_n$ and time t', we say that $S_1(\ell_k, t')$ holds if $\exists u_\ell \in C_m - \{u_{\ell_k}\} : g_\ell(t') > \tau_C^{\max}$, where $u_{\ell_k} \equiv (n, m)$.

Given $A \subseteq C_n$ and time t', we say $S_2(A, t')$ hold if $\exists u_\ell \in C_n - A : g_\ell(t') > \tau_C^{\max}$.



(a) $S_1(\ell_k, t')$. (b) $S_2(A, t')$.

Figure 5.10: Illustration of $S_1(\ell_k, t'), S_2(A, t')$.

We now present the outline of the proof. In the following, we carry out the proof in steps. At each step r, there are two possibilities either $S_1(\ell_r, t_r)$ holds, or $S_2(A_r, t_r)$ holds. If $S_1(\ell_r, t_r)$ holds, the proof is completed and the Lemma holds. Otherwise, $S_2(A_r, t_r)$ holds, which leads to step $r+1$, where either $S_1(\ell_{r+1}, t_{r+1})$ holds, or $S_2(A_{r+1}, t_{r+1})$. The process terminates if $S_1(\ell_r, t_r)$ holds at any step r, and continues otherwise. In the following proof, we show that the process has to terminate in finite steps.

Step 1:

For initialization, let $u_{\ell_0} := u_i$, $u_{\ell_1} := u_j$ and $t_0 := t$. Define $A_1 := \{u_{\ell_0}, u_{\ell_1}\}, t_1 := t_0 - \tau(u_{\ell_0})$. Observe that it is given t_1 is the update slot of user u_{ℓ_1}, t_0 is an update slot of u_{ℓ_0}, and that $g_{\ell_0}(t_0) > \tau_C^{\max}$, $g_{\ell_1}(t_1) > \tau_C^{\max}$. It follows that u_{ℓ_0}, u_{ℓ_1} have no update slots in the interval $\{t_0 - \tau_{C_n}, \ldots, t_1\}$.

Now, using Lemma 5.6.1 for user u_{ℓ_1} (equivalent to say link (n, m)) at time t_1, we have that at least one of the following must hold.

1)$S_1(\ell_1, t_1 - \tau(u_{\ell_1}))$, i.e., $\exists u_\ell \in C_m - \{u_{\ell_1}\} : g_\ell(t_1 - \tau(u_{\ell_1})) > \tau_C^{\max}$, where $u_{\ell_1} \equiv (n, m)$. Here, m is the node m in Lemma 5.6.1.

2)$S_2(A_1, t_1 - \tau(u_{\ell_1}))$, i.e., $\exists u_\ell \in C_n - A_1 : g_\ell(t_1 - \tau(u_{\ell_1})) > \tau_C^{\max}$. Here, n is the node n in Lemma 5.6.1.

Let $t_2 = t_1 - \tau(u_{\ell_1})$. If $S_1(\ell_1, t_2)$ holds, observe that lemma is proved for $t' = t_2$ and $k = \ell_1$, $l = \ell$ (i.e., ℓ from $S_1(\ell_1, t_2)$). Suppose not, therefore $S_2(A_1, t_2)$ must hold. This implies $\exists u_{\ell_2} \in C_n - A_1 : g_{\ell_2}(t_2) > \tau_C^{\max}$, where ℓ_2 is the ℓ from $S_2(A_1, t_2)$.

For an illustration, consider the example in Figure. 5.8. Here, $\tau_C^{\max} = 6$. Note that $g_1(t_1) = 7, g_2(t_1 - 2) = 7$ for u_1, u_2 at $t_1 = 17$. Here, $\ell_0 = 1, \ell_1 = 2$ and $n = n_1$. In Figure. 5.8, we have

$g_3(t_1 - 4) = 7$. Observe that $S_2(A_1, t_2)$ holds here for $u_\ell = u_3 \in C_n - \{u_1, u_2\}$. Here, $\ell_2 = 3$.

Step 2:

Firstly, note that since $g_{\ell_2}(t_2) > \tau_C^{\max}$, it follows that u_{ℓ_2} has no update slot in the interval $\{t_0 - \tau_{C_n}, \ldots t_2 - \tau(u_{\ell_2})\}$.

Let $A_2 := A_1 \bigcup \{u_{\ell_2}\} = \{u_{\ell_0}, u_{\ell_1}, u_{\ell_2}\}$, $t_3 := t_2 - \tau(u_{\ell_2})$. Using Lemma 5.6.1 for user u_{ℓ_2} at time t_2, we have that at least one of the following must be true

1)$S_1(\ell_2, t_3)$

2)$S_2(A_2, t_3)$

If $S_1(\ell_2, t_3)$ is true, observe that lemma is proved for $t' = t_2$ and $k = \ell_2$, and $l = \ell$ (i.e., ℓ from $S_1(\ell_1, t_3)$). Suppose not, therefore $S_2(A_2, t_3)$ must hold. This implies $\exists u_{\ell_3} \in C_n - A_2$ such that $g_{\ell_3}(t_3) > \tau_C^{\max}$, where ℓ_3 is the ℓ from $S_2(A_2, t_3)$.

For an illustration, consider the example in Figure. 5.8. Note that $g_2(t_2 + 2) = 7, g_3(t_2) = 7$ for u_2, u_3 at $t_2 = 13$. Here, $\ell_1 = 2, \ell_2 = 3$ and $n = n_1$. In Figure. 5.8, we have $g_4(t_2 - 2) = 7$. Observe that $S_1(A_2, t_3)$ holds here for $u_\ell = u_4 \in C_{n_2} - \{u_3\}$. Here, the Lemma is proved for $k = \ell_2 = 3$, $l = \ell_3 = 4$ and $t' = t_3$.

Repeat this process r times until $A_r \equiv C_n$.

Step r:

$A_r := \{u_{\ell_0}, u_{\ell_1}, \ldots, u_{\ell_r}\} \equiv C_n$, $t_r := t_{r-1} - \tau(u_{\ell_{r-1}})$ and $t_{r+1} := t_r - \tau(u_{\ell_r})$. Using Lemma 5.6.1 for user i_r at time t_r, we have that at least one of the following must be true

1)$S_1(\ell_r, t_{r+1})$

2)$S_2(A_r, t_{r+1})$

If $S_1(\ell_r, t_{r+1})$ is true, observe that lemma is proved for $t' = t_{r+1}, k = \ell_r, l = \ell$ (i.e., ℓ from $S_1(\ell_r, t_{r+1})$). Suppose not, therefore $S_2(A_r, t_{r+1})$ must be true. This implies $\exists u_{\ell_{r+1}} \in C_n - A_r : g_{\ell_{r+1}}(t_{r+1}) > \tau_C^{\max}$. This is a contradiction since $C_n - A_r \equiv \phi$.

Therefore, $S_1(\ell_p, t_{p+1})$ must hold at some step $p \in \{1, 2, \ldots r\}$. Hence, Lemma is proved for $k = \ell_p$ and $l = \ell$ (i.e., ℓ from $S_1(\ell_p, t_{p+1})$), $t' = t_{p+1} = t - \sum_{\ell \in A_{p+1}} \tau(u_\ell)$. □

Lemma 5.6.3. $g_i(t) \leq \tau_C^{\max}, \forall i \in \{1, \ldots, N\}, t > \sum_{i=1}^{N} \tau(u_i) + |X(0)|$.

Proof. Lets suppose not, i.e., there exists a user u_i and an update slot t such that $g_i(t) > \tau_C^{\max}$ for some $t > \sum_{i=1}^{N} \tau(u_i) + |X(0)|$.

Consider the example in Figure. 5.8. We note that the time t considered in this lemma is greater than 32 (which is an upper bound on the convergence time). In Figure. 5.8(a), the algorithm has already converged at slot 19.

Using Lemma 5.6.1, there exists a user $u_j \in C_n$ such that $g_j(t - \tau(u_i)) > \tau_C^{\max}$, where n is the common node of u_i, u_j. For sake of convenience, let $i_0 := i, j_0 := j$ and $t_0 := t - \tau(u_{i_0})$

Consider the rooted version of tree G_2 as G_2^n with root as node n. Let $c(n)$ denote a child of node n in tree G_2^n. For the sake of convenience, let $c^2(n)$ denote a child of $c(n)$. In general, let $c^r(n)$ denote a child of $c^{r-1}(n)$.

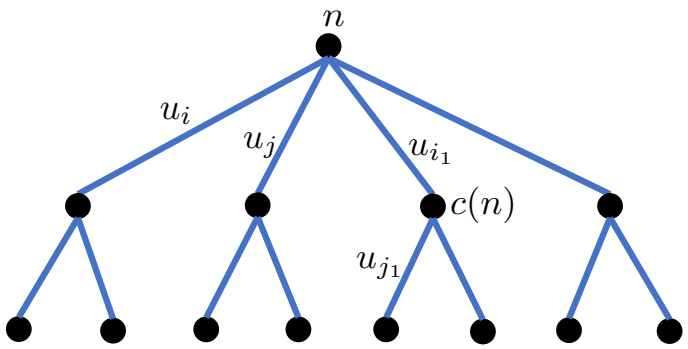


Figure 5.11: Example diagram at step 1.

We present the outline of the rest of the proof. In the following, we present the proof in steps. At each step we apply Lemma 5.6.2, which leads to a conclusion about a child node. Hence, at each step, we move one level down the tree. Eventually we reach a leaf node, where applying Lemma 5.6.2 results in a contradiction, hence completing the proof.

Step 1:

Using Lemma 5.6.2 for $u_i = u_{i_0}, u_j = u_{j_0}$ at time $t = t_0 + \tau(u_{i_0})$, we have that for some child $c(n)$, there exists $u_{i_1} := (n, c(n)) \neq u_{i_0}$ and $u_{j_1} \in C_{c(n)} - \{u_{i_1}\}$ and $t_1 \geq t_0 - \tau_{C_n}$ such that $g_{i_1}(t_1 + \tau(u_{i_1})) > \tau_C^{\max}$ and $g_{j_1}(t_1) > \tau_C^{\max}$. Here, i_1 is k, j_1 is l and $c(n)$ is n' from Lemma 5.6.2.

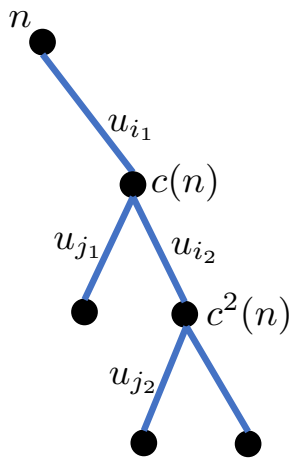


Figure 5.12: Example diagram at step 2.

Step 2:

Using Lemma 5.6.2 for $u_i = u_{i_1}, u_j = u_{j_1}$ at time $t = t_1 + \tau(u_{i_1})$, we have for some child $c^2(n)$, that there exists $u_{i_2} := (c(n), c^2(n))$, $u_{j_2} \in C_{c^2(n)} - \{u_{i_2}\}$ and $t_2 \geq t_1 - \tau_{C_{c^2(n)}}$ such that $g_{i_2}(t_2 + \tau(u_{i_2})) > \tau_C^{\max}$ and $g_{j_2}(t_2) > \tau_C^{\max}$. Here, i_2 is k, j_2 is l and $c^2(n)$ is n' from Lemma 5.6.2.

Repeat the process r times until eventually $c^r(n)$ is a leaf.

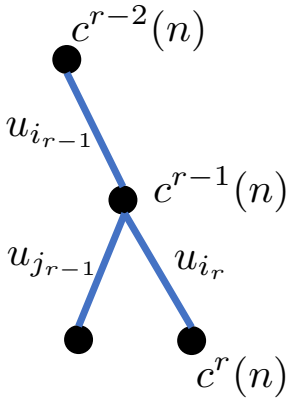


Figure 5.13: Example diagram at step r.

Step r:

Using Lemma 5.6.2 for $u_i = u_{i_{r-1}}, u_j = u_{j_{r-1}}$ at time $t = t_{r-1} + \tau(u_{i_{r-1}})$, we have for some child $c^r(n)$ of $c^{r-1}(n)$, that there exist $u_{i_r} := (c^{r-1}(n), c^r(n))$, $u_{j_r} \in C_{c^r(n)} - \{u_{i_r}\}$ and $t_r \geq t_{r-1} - \tau_{C_{c^r(n)}}$ such that $g_{i_r}(t_r + \tau(u_{i_r})) > \tau_C^{\max}$ and $g_{j_r}(t_r) > \tau_C^{\max}$. Here, i_r is k, j_r is l and $c^r(n)$ is n' from Lemma 5.6.2.

Since $c^r(n)$ is a leaf, $C_{c^r(n)} - \{u_{i_r}\} = \phi$. Therefore, the user u_{j_r} cannot exist, which is a contradiction. This completes the proof. □

5.7 Comparison with an alternate greedy approach

In the proposed algorithm, at an update for a user u_i, we insist on a contiguous group of free slots $\{f_i(t), \ldots, f_i(t) + \tau(u_i) - 1\}$ for any allocation. Intuitively, it may seem like the allocation is not efficient, since any free slots in the interval $\{t + 1, \ldots, f_i(t)\}$ are not allocated due to the contiguous property. In this section, we consider an alternate greedy approach (AG), where at each update slot t for u_i, $T_i(t + 1)$ is allocated as the first $\tau(u_i)$ free slots in the interval $\{t + 1, \ldots, \infty\}$, which can be considered to be a maximum packing strategy. Just as in the proposed algorithm, t is an update slot for u_i if and only $T_i(t) = \{t\}$. In the following, we provide an example showing the allocation under AG algorithm.

We consider an example in topology G_1 with 15 users $\{u_i\}_{i=1}^{15}$. Here, $I(u_1) = u_2$, $I(u_{15}) = u_{14}$, and $I(u_i) = \{u_{i-1}, u_{i+1}\}$ for $i = 2, \ldots, 14$. The loads $[\tau(u_i)]_{i=1}^{15}$ are given by $[4, 2, 4, 4, 1, 4, 4, 3, 2, 1, 3, 4, 1, 4, 4]$. An initial allocation for this setup is provided using black colored Xs in Table 5.8 (on page 153). The updated allocations under AG algorithm are marked using red, blue, green and violet colored Xs alternatively for each user u_i in Table 5.8 (on page 153) and Table 5.9 (on page 154).

The clique bound for this setup is given by 8. Note that $K(1) = 9$, $K(14) = 9$, $K(15) = 8$ and $K(16) = 9$. It may appear that the AG algorithm has converged to the optimal allocations. However upon closer inspection, one can note that $K(27) = 11$, $K(28) = 12$ and $K(29) = 13$ in Table 5.9 (on page 154). Hence, from this example, it is clear that monotonicity of $K(t)$ does not apply under the AG algorithm. Furthermore, the AG algorithm can go into sub-optimal allocations in the future, even though it is optimal now. Specifically for this example, even after 10^6 slots, $K(10^6) = 8$ and $K(10^6 + 7) > 8$, which indicates the AG algorithm may never converge to the optimal solution, but only cycles between feasible allocations (some optimal and some strictly sub-optimal).

Table 5.8: Alternative greedy algorithm.

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
u_1	X	X	X	X			X	X	X	X	X	X	X			X	
u_2					X	X								X	X		X
u_3	X	X	X	X						X	X	X	X				
u_4						X	X	X	X					X	X	X	X
u_5					X								X				
u_6	X	X	X	X					X	X	X	X					X
u_7					X	X	X	X					X	X	X	X	
u_8	X	X	X						X	X	X						X
u_9				X	X	X	X					X	X	X		X	
u_{10}	X	X	X					X	X	X	X				X		X
u_{11}					X	X	X					X	X	X			
u_{12}	X	X	X	X				X	X	X	X				X	X	X
u_{13}					X	X	X						X	X			
u_{14}	X	X	X	X					X	X	X	X					X
u_{15}					X	X	X	X					X	X	X	X	

Table 5.9: Alternative greedy algorithm (continued from Table 5.8).

t	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
u_1	X	X	X	X		X	X	X	X	X	X	X			X		X
u_2					X								X	X		X	
u_3	X	X	X	X					X	X	X	X					X
u_4					X	X	X	X					X	X	X	X	
u_5				X								X					
u_6	X	X	X					X	X	X	X					X	X
u_7				X	X	X	X					X	X	X	X		
u_8	X	X						X	X	X						X	X
u_9			X	X		X	X				X	X		X	X		
u_{10}	X				X			X					X			X	
u_{11}		X	X	X					X	X	X	X					X
u_{12}	X				X	X	X	X					X	X	X	X	
u_{13}				X								X					
u_{14}	X	X	X					X	X	X	X					X	X
u_{15}				X	X	X	X					X	X	X	X		

5.8 Theoretical Results

Proof of Lemma 5.3.1. Since at each update, user u_i gets allocated exactly $\tau(u_i)$ slots, it follows that $g_i(t) \geq \tau(u_i), \forall t > |X_i(0)|$. Note that since $\alpha_i(t)$ and $\beta_i(t)$ are update slots of u_i, it follows that u_i occupies the slots in the sets $\{\beta_i(t) - \tau(u_i) + 1, \ldots, \beta_i(t)\}$ and $\{\alpha_i(t) - \tau(u_i) + 1, \ldots, \alpha_i(t)\}$. Also note that u_i has no allocated slots in the interval $\{\beta_i(t) + 1, \ldots, \alpha_i(t) - \tau(u_i)\}$.

Suppose t is an update slot of u_i, i.e., $\alpha_i(t) = t$. It follows that slots $\{t - \tau(u_i) + 1, \ldots, t\}$ are occupied by u_i. Since $\alpha_i(t) = t$, it follows that $t - g_i(t) + 1 = \beta_i(t) + 1$ (because $g_i(t) = \alpha_i(t) - \beta_i(t)$). Therefore, $\{\beta_i(t) - \tau(u_i) + 1, \ldots, \beta_i(t)\} \bigcap \{t - g_i(t) + 1, \ldots, t\} = \phi$. The lemma holds for this case because the allocated slots $\{\alpha_i(t) - \tau(u_i) + 1, \ldots, \alpha_i(t)\} \subset \{t - g_i(t) + 1, \ldots, t\}$ are the only ones in the

interval.

For the other case, suppose that t is not an update slot of u_i. It follows that $\beta_i(t) < t < \alpha_i(t)$. We have the following two cases.

Case 1: Suppose $t < \alpha_i(t) - \tau(u_i) + 1$. Subtracting $g_i(t)$ from both sides of the inequality, it follows that $t - g_i(t) < \beta_i(t) - \tau(u_i) + 1$. Hence, it follows that $\{\beta_i(t) - \tau(u_i) + 1, \ldots, \beta_i(t)\} \subset \{t - g_i(t) + 1, \ldots, t\}$. Since $t < \alpha_i(t) - \tau(u_i) + 1$, we have $\{\alpha_i(t) - \tau(u_i) + 1, \ldots, \alpha_i(t)\} \bigcap \{t - g_i(t) + 1, \ldots, t\} = \phi$. The lemma holds for this case since $\{\beta_i(t) - \tau(u_i) + 1, \ldots, \beta_i(t)\} \subset \{t - g_i(t) + 1, \ldots, t\}$ are allocated slots of u_i, and also they are the only ones in the interval $\{t - g_i(t) + 1, \ldots, t\}$.

Case 2: Suppose $t \geq \alpha_i(t) - \tau(u_i) + 1$. Here, slots $S_1 := \{\alpha_i(t) - \tau(u_i) + 1, \ldots, t\}$ are occupied by u_i. Subtracting $g_i(t)$ from both sides of the inequality, it follows that $t - g_i(t) \geq \beta_i(t) - \tau(u_i) + 1$. Now since $t < \alpha_i(t)$, it also follows that $t - g_i(t) < \beta_i(t)$. Therefore, slots $S_2 := \{t - g_i(t) + 1, \ldots, \beta_i(t)\}$ are also occupied by u_i. Note that $|S_1| + |S_2| = \tau(u_i)$. Since $S_1 \bigcup S_2 \subset \{t - g_i(t) + 1, \ldots, t\}$, the lemma holds. This concludes the proof. □

Proof of Lemma 5.3.2. We show that for each $i = 1, \ldots, N$, there are at at least $\tau(u_i)$ allocated slots in the interval $\{t - K(t) + 2, \ldots, t + 1\}$, which is enough to prove the result. We divide the set $\{1, \ldots, N\}$ into mutually exclusive (and exhaustive) sets $U_1, U_2, U_{3,1}, U_{3,2}$. We show the claim for each of these sets in the following.

Let U_1 denote the set of $i \in \{1, \ldots, N\}$ such that there are at least $\tau(u_i) + 1$ allocated slots in the interval $\{t - K(t) + 1, \ldots, t\}$. It follows that for any $i \in U_1$, there are at at least $\tau(u_i)$ allocated slots in the interval $\{t - K(t) + 2, \ldots, t\} \subset \{t - K(t) + 2, \ldots, t + 1\}$.

Let U_2 denote the set of $i \in \{1, \ldots, N\}$ such that there are exactly $\tau(u_i)$ allocated slots in the interval $\{t - K(t) + 1, \ldots, t\}$, and $t - K(t) + 1$ is not an allocated slot of u_i. It follows that for any $i \in U_2$, there are at exactly $\tau(u_i)$ allocated slots in the interval $\{t - K(t) + 2, \ldots, t\}$, and at least $\tau(u_i)$ allocated slots in the interval $\{t - K(t) + 2, \ldots, t + 1\}$.

Let U_3 denote the set of $i \in \{1, \ldots, N\}$ such that there are exactly $\tau(u_i)$ allocated slots in the interval $\{t - K(t) + 1, \ldots, t\}$, and $t - K(t) + 1$ is an allocated slot of u_i. We consider the following two cases.

Let $U_{3,1}$ denote set of $i \in U_3$ such that slot t is allocated to u_i. Consider any $i \in U_{3,1}$. Since there are $\tau(u_i)$ slots of u_i in the interval $\{t - K(t) + 1, \ldots, t\}$, it follows that there is an update slot in the interval. We consider the following two cases for u_i

Case 1: Suppose that t is an update slot of u_i, it follows that slots $\{t-\tau(u_i)+1,\ldots,t\}$ are occupied by u_i. Since there are exactly allocated $\tau(u_i)$ slots of u_i in the interval $\{t-K(t)+1,\ldots,t\}$, it follows that $K(t)=\tau(u_i)$. Since all the other users u_j have allocated slots in the same interval $\{t-\tau(u_i)+1,\ldots,t\}$, it follows that $u_i \notin I(u_j), \forall j \in \{1,\ldots,N\}-\{i\}$. Hence, it follows that during the update at time t, u_i will occupy the immediate free slots $\{t+1,\ldots t+\tau(u_i)\}$. Therefore, there are $\tau(u_i)$ allocated slots in the interval $\{t-K(t)+2,\ldots,t+1\}$.

Case 2: For the other case, suppose that t is not an update slot of u_i. Since slots are allocated in a contiguous interval and since slot t occupied by u_i is not an update slot, it follows that $t+1$ is also occupied by u_i. Therefore, there are $\tau(u_i)$ allocated slots in the interval $\{t-K(t)+2,\ldots,t+1\}$.

For the final set of users, let $U_{3,2}$ denote set of $i \in U_3$ such that slot t is not allocated to u_i. Consider any $i \in U_{3,2}$. Since there are exactly $\tau(u_i)$ allocated slots in the interval $\{t-K(t)+1,\ldots,t\}$, it follows that there is one update slot for u_i in the interval. We claim that the update slot has to be $t-K(t)+\tau(u_i)$. We prove the claim in the following paragraph.

Let t' denote the update slot. 1) Suppose all of the allocated slots from update at t' lie inside the interval $\{t-K(t)+1,\ldots,t\}$. This implies the $\tau(u_i)$ slots (from the update) along with the slot t' are occupied by u_i. This is contradiction since there are exactly $\tau(u_i)$ allocated slots in the interval $\{t-K(t)+1,\ldots,t\}$. 2) Suppose some of the slots from the update at t' lie outside the interval $\{t-K(t)+1,\ldots,t\}$, and others lie inside. This implies slot t is occupied by u_i. This is a contradiction since u_i does not occupy slot t from the definition of $U_{3,2}$. 3) The only remaining option is that all the allocated slots from the update at t' lie outside the interval $\{t-K(t)+1,\ldots,t\}$. Since there are $\tau(u_i)$ allocated slots for u_i in the interval and since $t-K(t)+1$ is occupied by u_i, the only choice for t' must be $t-K(t)+\tau(u_i)$.

Hence, we have shown that for the arbitrarily considered $i \in U_{3,2}$, the slots $\{t-K(t)+1,\ldots,t-K(t)+\tau(u_i)\}$ are exactly the slots occupied by u_i in the given interval $\{t-K(t)+1,\ldots,t\}$. Note that for each $u_j \in I(u_i)$, there are at least $\tau(u_j)$ allocated slots in the interval $\{t-K(t)+1,\ldots,t\}$. Since the slots $\{t-K(t)+1,\ldots,t-K(t)+\tau(u_i)\}$ are occupied by u_i, there are at least $\tau(u_j)$ allocated slots in the interval $\{t-K(t)+\tau(u_i)+1,\ldots,t\}$ for each $u_j \in I(u_i)$. It follows that the occupied slots (by $u_j \in I(u_i)$) during the update for u_i at time $t-K(t)+\tau(u_i)$ must satisfy

$$\bigcup_{u_j \in I(u_i)} T_j(t-K(t)+\tau(u_i)) \subseteq \{t-K(t)+\tau(u_i)+1,\ldots,t\} \tag{5.7}$$

It follows that the immediate free slot $f_i(t-K(t)+\tau(u_i))$ at the update must be $t+1$. Hence, during

this update, u_i occupies the slots in interval $\{t+1,\ldots,t+\tau(u_i)-1\}$. Hence, it follows that for each $i \in U_{3,2}$, there are exactly $\tau(u_i)$ slots in the interval $\{t-K(t)+2,\ldots,t+1\}$. This concludes the proof. □

Proof of Lemma 5.4.2. It follows from Lemma 5.4.1 that $t-\tau(u_i)$ is an update slot for either user u_i or some user $u_m \in I(u_i)$. If $t-\tau(u_i)$ is an update slot for user u_i, then $g_i(t) = t-(t-\tau(u_i)) = \tau(u_i) \leq \tau_C^{\max}$, which is a contradiction (since it is given that $g_i(t) > \tau_C^{\max}$). Therefore, $t-\tau(u_i)$ is an update slot for some user $u_m \in I(u_i)$.

Let $t' = \beta_i(t)$, i.e., t' is the latest update slot of u_i before t. We established that during the update at t', $f_i(t')-1 = t-\tau(u_i)$ is already occupied by some user $u_m \in I(u_i)$. Let $I^* \subseteq I(u_i)$ denote the set of all the users in $I(u_i)$ that are occupying the slot $t-\tau(u_i)$. In the following, we will show the existence of a user $u_j \in I^*$ that satisfies properties $b), c)\&d)$ of Lemma 5.4.2, from which $a)$ follows (since $t-\tau(u_i)$ is an update slot for all users $u \in I^*$).

Using Lemma 5.4.1 for each $u_m \in I^*$ at time $t-\tau(u_i)$, we have $t-\tau(u_i)-\tau(u_m)$ is an update slot for a user in the set $\{u_m\} \bigcup I(u_m)$. Note that $u_i \in I(u_m)$ for each $u_m \in I^*$. Now consider the following lemma (Lemma A), which shows that $t-\tau(u_i)-\tau(u_m)$ is not an update slot of u_i for any $u_m \in I^*$.

Lemma (Lemma A)**.** *For each user $u_m \in I^*$, $t-\tau(u_i)-\tau(u_m)$ is an update slot of some user $u_k \in \{u_m\} \bigcup I(u_m) - \{u_i\}$.*

Proof. Suppose not. Assume that there exists a user $u_m \in I^*$, such that $t-\tau(u_i)-\tau(u_m)$ is an update slot of user u_i. It follows that

$$\begin{aligned} g_i(t) &\leq t-(t-\tau(u_i)-\tau(u_m)) \\ &= \tau(u_i)+\tau(u_m) \\ &\leq \tau_C^{\max} \end{aligned}$$

This is a contradiction since it is given that $g_i(t) > \tau_C^{\max}$. Therefore, $t-\tau(u_i)-\tau(u_m)$ is not an update slot of u_i for any $u_m \in I^*$. □

In the following, we will show that there exists a user $u_j \in I^*$ satisfying properties $b)$ and $d)$ of Lemma 5.4.2. It is immediate that $c)$ follows from the above Lemma A.

Lemma (Lemma B)**.** *There exists at least one user $u_j \in I^*$ such that there is no update slot of u_j in $\{t'+1,\ldots t-\tau(u_i)-\tau(u_j)\}$.*

Proof. Suppose not. Assume all users $u_m \in I^*$ have at least one update slot in the interval $\{t' + 1, \ldots t - \tau(u_i) - \tau(u_m)\}$.

This implies that every user $u_m \in I^*$ occupies the slot $t - \tau(u_i)$ during an update in $\{t' + 1, \ldots t - \tau(u_i) - \tau(u_m)\}$ and not before. Therefore, during user u_i's update at time t', $t - \tau(u_i) \notin T_m(t')$ for any $u_m \in I(u_i)$, i.e., $t - \tau(u_i)$ is not occupied by any of the users in $I(u_i)$. This implies that block $\{t - \tau(u_i), \ldots t - 1\}$ is conflict free during update at t', which implies $f_i(t') \leq t - \tau(u_i)$. This is a contradiction since we have already established that $f_i(t') = t - \tau(u_i) + 1$. □

Since $\{t' - \tau(u_i) + 1, \ldots t'\}$ are occupied by u_i, it follows from Lemma B that the latest update slot of u_j before the update slot $t - \tau(u_i)$ is $\beta_j(t - \tau(u_i)) \leq t' - \tau(u_i)$. Hence,

$$\begin{aligned} g_j(t - \tau(u_i)) &= t - \tau(u_i) - \beta_j(t - \tau(u_i)) \\ &\geq t - t' \\ &= g_i(t) \\ &> \tau_C^{\max} \end{aligned}$$

This proves property *b*) of Lemma 5.4.2. It also follows from Lemma B that slot $t - \tau(u_i) - \tau(u_j)$ is not an update slot of user u_j. Property *d*) now follows from Lemma A since $u_j \in I^*$. □

Chapter 6

Fluid Limit of Dynamic Resource Sharing based on Minimum clearing time formulation

6.1 Introduction

Consider the K tier HetNet model in Chapter 4. The key contribution of Chapter 4 is the distributed framework to solve the minimum resource clearing problem for K tier HetNet model. We have presented a optimal distributed resource allocation algorithm for the K tier HetNet model. The resource allocation in Chapter 4 was done for a given fixed set of users $\mathcal{U}$ and their demands $\boldsymbol{\tau}$ (or $\boldsymbol{\alpha}$ (in bits/frame)). In this chapter, we consider a dynamic flow based model which operates on a slower time scale compared to the RB scheduling time scale in Chapter 4. A flow can be described as a stream of packets associated with a UE or user file request. The file request can last several frames, e.g., a flow lasting 1 second equals 100 LTE frames. Under the flow model, the user file requests arrive as a stochastic process and depart once the service requirement is fulfilled. The network operator (or the BS) sets the rate at which a user file request (or flow) is served. The policy used to regulate the rate of service to a flow is known as the flow (or rate) control policy. The objective of a flow control policy can include congestion control, or ensuring QOS.

In this section, we introduce a dynamic flow control and resource allocation policy for a K tier HetNet. We consider a dynamic scenario with stochastic flow arrivals and departures. Each arriving user file request (or flow) requires a certain service demand (in bits) and departs once the service is completed. We present a joint flow control and resource allocation policy, which is based on the

distributed framework (developed in Chapter 4) to solve the minimum resource clearing problem. In the later sections of this chapter, we will consider a more general setup. We are introducing the K tier HetNet model first as a motivating example for the setup. We will show that the proposed algorithm will stabilize the network (for any arrival rate vector inside the stability region). We say stability in the sense that the backlogged file requests do not blow up to infinity.

Flow based models have been used to study internet congestion in the literature [70–72]. An utility optimization framework for rate (/congestion) control in wireline networks (i.e., internet) was introduced in [71]. In [72], α-fair bandwidth sharing algorithms were introduced for internet flow control and the impact of fairness on flow level stability was studied.

In multi-hop wireless networks, flow control (also known as congestion control) happens at the transport layer, e.g., TCP (Traffic Control Protocol). The transport layer controls the rate at which packets arrive into the network based on the congestion level. The arriving packets are scheduled at the MAC (Multiple Access Channel) layer. Following [71], the problem of cross-layer optimization for multi-hop wireless networks was considered using an utility optimization framework in [19]. The problem of cross-layer optimization for multi-hop wireless networks was also considered in [10, 19, 73]

With respect to HetNets, flow based models were considered in [36, 38, 39]. In [38], α fair utility optimization based flow control for two tier HetNets was considered. The results in [38] show that delay performance is improved by adapting the resource allocation (based on flows in the network) compared to a static allocation. Flow level stability (under optimized user association) was considered in [39].

6.1.1 Flow control for K tier HetNet

We consider a similar setup as in Chapter 4. We use graph G (see Chapter 4) to represent the K tier HetNet. We model the co-tier interference constraints as before. In contrast to the discrete resource block model in Chapter 4, here, we assume that the spectrum is infinitely divisible, and fractions of the spectrum can be allocated at any point in time. For a BS $m \in \mathcal{R}(n)$, $I_c(m) \subseteq \mathcal{R}(n)$ is the set of BSs in tier $i+1$ which cannot be scheduled with BS m on the same fraction of spectrum. Similarly with the cross-tier interference constraints. A BS in $D(n)$ (i.e, a descendant of n) cannot be scheduled with any BS in the set $I_c(n) \bigcup \{n\}$ on the same fraction of spectrum.

In general, a BS n cannot be scheduled with any BS in the set $I(n)$ on the same fraction of the

spectrum, where

$$I(n) := \underbrace{I_c(n)}_{\text{co-tier interference}} \bigcup \underbrace{D(n) \bigcup_{m \in I_c(n)} D(m)}_{\text{cross-tier interference from } n \text{ to higher tiers}} \bigcup \underbrace{A(n) \bigcup_{m \in A(n)} I_c(m)}_{\text{cross-tier interference from lower tiers to } n} \tag{6.1}$$

We consider dynamic arrivals at each BS n; the user file requests (or flows) arrive as an exogenous process. For the sake of convenience, we take the amount of spectrum available to be 1 (i.e., one unit). Let $\{\xi_n(i)\}_{i=1}^{\infty}$ denote the sequence of inter-arrival periods (in sec) of user file requests at BS n. The arriving user file requests have a service requirement (in sec $\times$ spectrum unit) from the BS and they depart once the service requirement is met. Suppose a UE file request u arriving at time t (seconds) has a service requirement of η_1, and it is allocated half of the spectrum from time t to $t+1$. Then the residual service requirement at time $t+1$ equals $\eta_1 - 0.5$. Let $\{\eta_n(i)\}_{i=1}^{\infty}$ denote the sequence of service requirements of flows arriving at BS n. We make the following assumption on the inter-arrival times and service requirements.

Assumption 6. *We assume that $\{\xi_n(i)\}_{i=0}^{\infty}$ and $\{\eta_n(i)\}_{i=0}^{\infty}$ are independent iid sequences of random variables $\forall\, n \in V$ such that $1/\bar{\xi}_n := \mathbb{E}[\xi_n(1)] < \infty$ and $1/\bar{\eta}_n := \mathbb{E}[\eta_n(1)] < \infty$. Further, the sequences are independent across the BSs $n \in V$.*

Under this model, let $q_n(t)$ denote the number of flows at BS n at time t, and let $Q(t) := \{q_n(t)\}_{n\in V}$ denote the queue lengths at time t. Consider that the flows in queue are served in FIFO order. At time t, let $v_n(t)$ denote the time remaining for the next arrival at BS n, and $w_n(t)$ denote the remaining service requirement of HoL (Head of the Line) flow at BS n. Let $V(t) := \{v_n(t)\}_{n\in V}$, $W(t) := \{w_n(t)\}_{n\in V}$. The state at time t is given by $X(t) := [Q(t), V(t), W(t)]$. A resource allocation policy assigns a fraction $z_S(t) \in [0,1]$ of available spectrum to a feasible set S at t such that $\sum_{S\in\mathcal{S}_G} z_S(t) \leq 1$. This definition of resource allocation based on feasible sets ensures that the interference constraints are not violated by the allocation policy. We consider a class of *stationary policies* which make the decision $Z(t) := \{z_S(t)\}_{S\in\mathcal{S}_G}$ based on the current state $X(t)$. Under a stationary policy, the process $\mathcal{X} = \{X(t)\}_{t=0}^{\infty}$ is a strong Markov process (in continuous time) with the state space $\mathbb{X} := \mathbb{Z}_+^{|V|} \times \mathbb{R}_+^{|V|} \times \mathbb{R}_+^{|V|}$.

6.1.2 Stabilizing stationary policy for K tier HetNet

Consider the following LP (6.2) formulated for a $Q \in \mathbb{Z}_+^{|V|}$. Note that this is same as LP (4.3) (from Chapter 4) with $\tau(n)$ replaced by $q_n/\bar{\eta}_n$. Hence, the optimal value of LP (6.2) can be derived using

the message passing algorithms described in Chapter 4.

$$\begin{aligned} &\min \sum_{S\in\mathcal{S}} f_S \\ &\text{s.t.} \\ &\sum_{S:n\in S} f_S \geq q_n/\bar{\eta}_n, \forall n \in G; \\ &f_S \geq 0, \forall S \in \mathcal{S} \end{aligned} \tag{6.2}$$

Let $L(Q)$ denote the optimal value of LP (6.2) given Q. We propose the stationary policy that resource (spectrum) allocation in state Q be done according to the scaled solution $[f_S^*]_{S\in\mathcal{S}}/L(Q)$. Under the policy, each BS $n \in G$ gets a fraction $(q_n/\bar{\eta}_n)/L(Q)$ of the spectrum in state Q. Under this approach, the resource allocation only has to be changed when the state of the system changes, i.e, when a new arrival or a departure occurs. The flow arrivals and departures happen over a much slower time scale than packets. Hence, less reconfiguration and calculations are required. Experiments in [74] show that flow-level resource allocation has faster convergence and outperforms packet based congestion control practices in terms of delay.

The policy can be implemented using the distributed framework (see Figure. 6.1) developed in Chapter 4. In what follows, $\gamma(H[m])$ is the solution of (6.2) formulated for the sub-graph $H[m]$ of G. Refer to Chapter 4 for definition of $H[m]$, $\gamma(H[m])$ etc.,

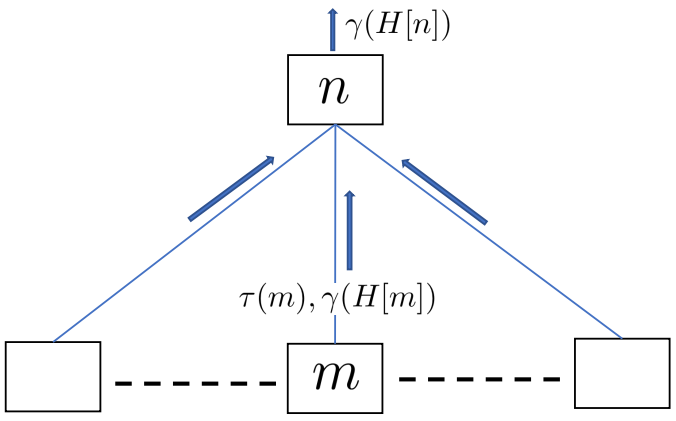


Figure 6.1: Distributed computation of $\gamma(H[n])$. Here, the upstream message to n are sent by the children $m \in \mathcal{R}(n)$.

6.1.2.1 Upstream Message passing

Given the current queue lenths $Q(t)$, $L(Q(t))$ can be derived using a distributed message passing algorithm (given in Chapter 4) as the solution of LP (6.2).

Now that the $L(Q(t))$ is known, we propose a downstream spectrum allocation according to the scaled solution of LP (6.2).

6.1.2.2 Downstream Resource allocation

The root BS r allocates $(q_r(t)/\bar{\eta}_r)/L(Q(t))$ for its own transmissions. It initiates the downstream allocation by allocating two fractions of spectrum to each child $m \in \mathcal{R}(n)$. The first fraction $(q_m(t)/\bar{\eta}_m)/L(Q(t))$ to BS m, and the second fraction of spectrum $\gamma(H[m])/L(Q(t))$ to $H[m]$ according to the scaled solution of LP (4.12). The allocations are feasible since $L(Q) = \gamma(H[r]) + q_r/\bar{\eta}_r$ which implies $(q_r/\bar{\eta}_r)/L(Q) + \gamma(H[r])/L(Q) = 1$.

Upon receiving the allocated spectrum, a BS n can follow a similar procedure. Using the $\gamma(H[n])/L(Q(t))$ fraction of RBs provided by its parent, n can allocate two fractions of spectrum to its children according to the solution of LP (4.12), scaled by $L(Q(t))$. The solution will be feasible since the scaled solution of LP (4.12) requires a fraction of $\gamma(H[n])/L(Q(t))$, which is provided by the parent.

In the following section, we introduce a more general setup and characterize the capacity region. We will propose a stationary flow control policy based on the solution of the minimum clearing problem, just as we did with K tier HetNet model here. The rest of the chapter will be focused on proving the stability under the proposed policy. We make use of the *Fluid limit* theory developed in [21, 22].

6.2 General Model Description

Consider a network of queues labelled as $\{1,\ldots,N\}$. Let $q_i(t)$ denote the number of flows in the queue i at time t. Let $Q(t) := \{q_i(t)\}_{i=1}^{N}$ denote the queue state at time t. The flow arrivals into queue i occur as an exogenous process. Let $\{\xi_i(n)\}_{n=1}^{\infty}$ denote the sequence of inter-arrival periods (in sec) of flows arriving into queue i.

We consider that one unit of resource (e.g., spectrum) is available, which has to be shared among

the queues. There are constraints on sharing the resources among the queues as follows. A queue i cannot be scheduled on the same fraction of resource along with any of the queues in the set $I(i) \subset \{1, \ldots, N\}$. Hence, any feasible resource allocation has to assign fractions of the resource to feasible sets of queues (defined in the following).

Definition 6.2.1. *A feasible set is a set of queues $S \subseteq \{1, \ldots, N\}$ such that $I(i) \bigcap S = \phi$, $\forall i \in S$.*

Definition 6.2.2. *A maximal feasible set is a feasible set which is not a subset of any other feasible set*

The sequence of service requirements (in sec $\times$ unit of resource) of the arriving flows at queue i are given by $\{\eta_i(n)\}_{n=1}^{\infty}$. For example, suppose a flow request arriving at time t (seconds) has a service requirement of η_1, and it is allocated half of the available resource from time t to $t+1$. Then the residual service requirement of the flow at time $t+1$ equals $\eta_1 - 0.5$. We make the following assumptions on the inter-arrival times and service requirements of flows.

Assumption 7. *We assume that $\{\xi_i(n)\}_{n=0}^{\infty}$ and $\{\eta_i(n)\}_{n=0}^{\infty}$ are independent iid sequences of random variables $\forall\ i \in \{1, \ldots N\}$ such that $1/\bar{\xi_i} := \mathbb{E}[\xi_i(1)] < \infty$ and $1/\bar{\eta_i} := \mathbb{E}[\eta_i(1)] < \infty$. Further, the sequences are independent across $i \in \{1, \ldots N\}$.*

Assumption 8.

$$\mathbb{E}[(\xi_i(1))^2] < \infty, \mathbb{E}[(\eta_i(1))^2] < \infty, \text{ for } i = 1, \ldots N \tag{6.3}$$

We consider HoL processing, where flows are served in the order of their arrival into the queues. At time t, let $u_i(t)$ denote the time remaining for the next arrival at queue i, and $v_i(t)$ denote the remaining service requirement of HoL flow in queue i. Let $U(t) := \{u_i(t)\}_{i=1}^{N}$, $V(t) := \{v_i(t)\}_{i=1}^{N}$. We consider the state at time t to be $X(t) := [Q(t), U(t), V(t)]$.

A resource allocation (or scheduling) policy assigns a fraction $z_S(t) \in [0, 1]$ of resource to a feasible set S at t such that $\sum_{S \in \mathcal{S}} z_S(t) \leq 1$, where $\mathcal{S}$ is the set of all the maximal feasible sets. We consider the class of stationary policies (see Definition 6.2.3) which make the decision $Z(t) := \{z_S(t)\}_{S \in \mathcal{S}}$ based on the current state $X(t)$, i.e., $\{z_S\}_{S \in \mathcal{S}}$ is a function of state X[1]. We note that the class of stationary

[1]In general, it is possible to consider classes of scheduling policies which do not satisfy this property. However, these other classes do not achieve more in terms of stability region. This fact can be shown using the same arguments used in the proof of Theorem 6.2.1

policies that we consider here also include the policies with knowledge of $U(t), V(t)$. In practice, it is not possible for policies have access to $U(t)$, which would require foresight of the next arrivals. Nevertheless, this knowledge does not provide any additional advantage to the policy in terms of stability, as it will become clear in the following Theorem 6.2.1.

Definition 6.2.3. *A stationary scheduling policy* $\theta : \mathbb{Z}_+^N \times \mathbb{R}_+^N \times \mathbb{R}_+^N \to [0,1]^{|\mathcal{S}|}$ *is a mapping from the state* X *to* $\{z_S(X)\}_{S\in\mathcal{S}}$ *such that* $\sum_{S\in\mathcal{S}} z_S(X) = 1$.

Table 6.1: State Notation.

Notation	Description
$X(t) := [Q(t), U(t), V(t)]$	The state of the system at time t.
$Q(t) := \{q_i(t)\}_{i=1}^N$	The queue lengths at time t. At time t, $q_i(t)$ is the number of backlogged flows in queue i at time t.
$U(t) := \{u_i(t)\}_{i=1}^N$	The residual arrival times at time t. At time t, $u_i(t)$ is the time remaining for next arrival into queue i.
$V(t) := \{v_i(t)\}_{i=1}^N$	The residual service requirements at time t. At time t, $v_i(t)$ is the remaining service requirement for HoL flow at queue i.

Under a stationary policy, the process $\mathcal{X} = \{X(t)\}_{t=0}^\infty$ is a strong (continuous time) Markov process with the state space $\mathbb{X} := \mathbb{Z}_+^N \times \mathbb{R}_+^N \times \mathbb{R}_+^N$. To establish the strong Markov property, we use the same line of argument given in [21]. Because of Assumption 7, we can check that X is Markov for a stationary policy. As time t increases, $u_i(t)$ and $v_i(t)$ decrease, while the remainder of the state $\{q_i(t)\}_{i=1}^N$ remains constant. A jump occurs for X when one of the residual processes $\{u_i(t)\}_{i=1}^N, \{v_i(t)\}_{i=1}^N$ reaches zero. Hence, $\{X(t)\}_{t\geq 0}$ is a piecewise deterministic Markov process, which satisfies Assumption 3.1 of [75]. The strong Markov property follows from [75] (page 362).

Remark. *Although we consider HoL processing for the queues, the results can be directly applied to processor sharing queues (where all the flows in a queue are served with equal effort) under the assumption that the service requirement distribution is exponential. This is possible since the state* $X(t) := [Q(t), U(t)]$ *under exponential distribution for service requirements.*

For general service distributions, the state X has to be redefined for processor sharing queues to include all the residual service requirements of the flows in the queue. Moreover, the fluid limit is also different in this case [21].

6.2.1 Stability Region

Definition 6.2.4. *We consider the network to be stable under a stationary policy if and only if*

$$\limsup_{t\to\infty} \frac{1}{t} \int_0^t \mathbb{E}\left[q_i(s)\right] ds < \infty, \text{ for } i = 1, \ldots, N \tag{6.4}$$

Definition 6.2.5. *The network is stabilizable if and only if there exists a stationary policy under which it is stable.*

Define

$$\rho_i := \bar{\xi}_i / \bar{\eta}_i, \ \forall i \in \{1, \ldots, N\}, \tag{6.5}$$

and $\boldsymbol{\rho} := [\rho_i]_{i=1}^N$.

Consider the following LP (6.6). We provide a capacity characterization using the optimal value of LP (6.6) in Theorem 6.2.1.

$$\begin{aligned} &\min \sum_{S\in\mathcal{S}} f_S \\ \text{s.t.} \ &\sum_{S:i\in S} f_S \geq \rho_i, \forall i \in \{1, \ldots N\}; \\ &f_S \geq 0, \forall S \in \mathcal{S} \end{aligned} \tag{6.6}$$

where f_S is the fraction of resource allocated to the feasible set S and $\mathcal{S}$ is the set of all maximal feasible sets.

Theorem 6.2.1. *Let $\{f_S^*\}_{S\in\mathcal{S}}$ denote the optimal solution of LP (6.6). The system is stabilizable if $\sum_{S\in\mathcal{S}} f_S^* < 1$, and not stabilizable if $\sum_{S\in\mathcal{S}} f_S^* > 1$.*

Proof. We provide the proof of instability here. The rest of the chapter is dedicated to proving the other case.

Suppose that $\sum_{S\in\mathcal{S}} f_S^* > 1$. Let $\mathcal{H}$ denote the set of all solutions $\{z_S\}_{S\in\mathcal{S}}$ such that $\sum_{s\in\mathcal{S}} z_S \leq 1$. For any $\{z_S\}_{S\in\mathcal{S}} \in \mathcal{H}$, we claim that $\sum_{S:i\in S} z_S < \rho_i$ for some $i \in \{1, \ldots, N\}$. The proof of the claim is given in the following paragraph.

Suppose not, and assume $\sum_{S:i\in S} z_S \geq \rho_i, \forall i \in \{1,\ldots,N\}$. It follows that $\{f_S\}_{S\in\mathcal{S}} := \{z_S\}_{S\in\mathcal{S}}$ is a feasible solution of LP (6.6). The value of objective function under this solution equals $\sum_{S\in\mathcal{S}} z_S \leq 1$, which is a contradiction since the optimal value $\sum_{S\in\mathcal{S}} f_S^* > 1$. Hence, the claim is proved.

Define

$$k_1 := \inf_{\mathcal{H}} \sup_{j=1,\ldots,N} \rho_j - \sum_{S:j\in S} z_S \tag{6.7}$$

Consider any arbitrary stationary scheduling policy. Let $Z_S(t) = \int_0^t z_S(\omega)d\omega$ denote the amount of resource allocated to a feasible set S until time t. Let $G_i(t) := \sum_{S:i\in S} Z_S(t)$ for each $i = 1,\ldots,N$. Since $\sum_{S\in\mathcal{S}} Z_S(t) \leq t$, it follows that $\{Z_S(t)/t\}_{S\in\mathcal{S}}$ is an element of $\mathcal{H}$ for any time $t \geq 0$. In the following, we consider the queue lengths under the chosen scheduling policy.

The counting processes corresponding to arrival process and departure process (at queue j) are given by the following equations.

$$a_j(t') = \max\{n : u_j(0) + \sum_{k=1}^{n-1} \xi_j(k) \leq t'\} \tag{6.8}$$

$$d_j(t') = \max\{n : v_j(0) + \sum_{k=1}^{n-1} \eta_j(k) \leq t'\} \tag{6.9}$$

The queue length of queue j at time t satisfies

$$q_j(t) \geq q_j(0) + a_j(t) - d_j(G_j(t)) \tag{6.10}$$

The inequality is since the queue may empty in between 0 and t, and departures are not possible when the queue is empty.

It follows from SLLN for renewal processes that

$$\lim_{t\to\infty} a_j(t)/t = \bar{\xi}_j \text{ a.s.} \tag{6.11}$$

for each $j = 1,\ldots,N$.

We also claim that for each $j = 1,\ldots,N$

$$\limsup_{t\to\infty} d_j(G_j(t))/t = \bar{\eta}_j \limsup_{t\to\infty} G_j(t)/t \text{ a.s.} \tag{6.12}$$

The proof of the claim is given as follows. Suppose first that $\limsup_{t\to\infty} G_j(t) < \infty$, it follows that almost surely $\limsup_{t\to\infty} d(G_j(t)) < \infty$. Hence, $\limsup_{t\to\infty} d_j(G_j(t))/t = 0 = \bar{\eta}_j \limsup_{t\to\infty} G_j(t)/t$ a.s.

For the other case, suppose $\limsup_{t\to\infty} G_j(t) = \infty$. It follows that

$$\limsup_{t\to\infty} d_j(G_j(t))/t = \limsup_{t\to\infty} \left(d(G_j(t))/G_j(t)\right)\left(G_j(t)/t\right) \tag{6.13}$$

By applying SLLN for renewal processes, we have $\limsup_{t\to\infty} d_j(G_j(t))/G_j(t) = \bar{\eta}_j$ almost surely. Hence, in either case, we have the result (6.12).

It follows from (6.10)-(6.12) that, for each $j = 1, \ldots, N$, almost surely,

$$\liminf_{t\to\infty} q_j(t)/t \geq \bar{\xi}_j - \bar{\eta}_j \limsup_{t\to\infty} G_j(t)/t \tag{6.14}$$

$$= \left(\bar{\xi}_j - \bar{\eta}_j \sum_{S:j\in S} \limsup_{t\to\infty} Z_S(t)/t\right) \tag{6.15}$$

Recall that $\{Z_S(t)/t\}_{S\in\mathcal{S}} \in \mathcal{H}$ for any $t > 0$. It follows that, given any $t > 0$, $\exists i \in \{1, \ldots, N\}$ such that

$$\left(\bar{\xi}_i - \bar{\eta}_i \sum_{S:i\in S} Z_S(t)/t\right) \geq \bar{\eta}_i k_1 \tag{6.16}$$

where k_1 is defined in (6.7). Since $q_j(t) \geq 0, \forall j, \forall t \geq 0$, it follows from (6.15) and (6.16) that

$$\liminf_{t\to\infty} \sum_{j=1}^{N} q_j(t)/t \geq k_1 \inf_{j\in\{1,\ldots,N\}} \bar{\eta}_j \text{ a.s.} \tag{6.17}$$

Let $k_2 := k_1 \inf_{j\in\{1,\ldots,N\}} \bar{\eta}_j$. From the definition of lim inf, it follows that $\exists T_1$ such that $\sum_{j=1}^{N} q_j(t) > k_2 t/2$, a.s.$\forall t \geq T_1$. Since $q_j(t)$ only takes values in non-negative integers, it follows that for $t \geq T_1$,

$$\sum_{j=1}^{N} E[q_j(t)] > k_2 t/2 \tag{6.18}$$

Hence, it follows that

$$\frac{1}{t}\int_0^t \sum_{j=1}^{N} E[q_j(\phi)]d\phi > \frac{1}{t}\int_{T_1}^t \frac{k_2}{2}\phi d\phi \tag{6.19}$$

$$= \frac{k_2}{4t}(t^2 - T_1^2) \tag{6.20}$$

It is immediate that

$$\limsup_{t\to\infty} \frac{1}{t}\int_0^t \sum_{j=1}^{N} E[q_j(\phi)]d\phi = \infty \tag{6.21}$$

Hence, the system is not stable under the chosen policy. Since the choice of policy is arbitrary, it follows that the system is not stabilizable. □

6.3 Stabilizing Stationary Policy

6.3.1 Linear Programming Formulation

Consider the following LP (6.22). Let $L(Q)$ denote the optimal value of LP (6.22) for a $Q \in \mathbb{Z}_+^N - \mathbf{0}$.

$$\begin{aligned} &\min \sum_{S \in \mathcal{S}} f_S \\ \text{s.t.} \quad &\sum_{S: n \in S} f_S \geq q_i/\bar{\eta}_i, \forall i \in \{1, \dots N\}; \\ &f_S \geq 0, \forall S \in \mathcal{S} \end{aligned} \tag{6.22}$$

We propose the scheduling policy which allocates $z_S = f_S^*/L(Q)$ fraction of resource to a feasible set S, where $\{f_S^*\}_{S \in \mathcal{S}}$ is the optimal solution of LP (6.22). The proposed policy is feasible since $\sum_{S \in \mathcal{S}} z_S = 1$ by construction. Define

$$f_i(Q) := \begin{cases} (q_i/\bar{\eta})/L(Q) & \text{if } L(Q) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{6.23}$$

Note that queue i gets at least $f_i(Q)$ fraction of the resource in state Q by construction. In what follows, we take $f_i(Q)$ to be exactly the fraction that is allocated to queue i under the proposed policy. We will show that allocating fraction $f_i(Q)$ is enough for stability.

6.3.2 Dual Program and Lyapunov Function

Consider the dual-program of LP (6.22) as LP (6.24)

$$\begin{aligned} &\max \sum_{i=1}^{N} e_i q_i/\bar{\eta}_i \\ \text{s.t.} \quad &\sum_{i: i \in S} e_i \leq 1, \forall S \in \mathcal{S}; \\ &e_i \geq 0, \forall i \in \{1, \dots N\} \end{aligned} \tag{6.24}$$

Let $\mathcal{E}$ denote the set of extreme points of the polytope defined by the constraints of LP (6.24). Let

$$L_{\boldsymbol{e}}(Q) = \sum_{i=1}^{N} e_i q_i/\bar{\eta}_i \tag{6.25}$$

for each $e \in \mathcal{E}$. It follows from strong duality theorem that

$$L(Q) = \max_{\boldsymbol{e} \in \mathcal{E}} \sum_{i=1}^{N} e_i q_i / \bar{\eta}_i \tag{6.26}$$

$$= \max_{\boldsymbol{e} \in \mathcal{E}} L_{\boldsymbol{e}}(Q) \tag{6.27}$$

Hence, $L(Q)$ is a positive real number for each $Q \in \boldsymbol{Z}_+^N - \{\mathbf{0}\}$, and is zero for $Q = \mathbf{0}$. In the following sections, we will introduce the fluid scaling model for the considered setup. We will establish the existence of a fluid limit. We will show that the function $L(.)$ serves as a Lyapunov function for the fluid limit model.

6.3.3 Alternative stability condition using dual interpretation

We present the following Lemma 6.3.1 which provides an equivalent statement of the stability condition in Theorem 6.2.1. This will be the stability condition which will be used in the following sections. In the rest of the chapter, we will show that the system is stable under the proposed policy, if (6.29) holds.

Lemma 6.3.1. *Let $\{f_S^*\}_{S \in \mathcal{S}}$ denote the optimal solution of LP (6.6), and $\boldsymbol{\rho} := [\bar{\xi}_i / \bar{\eta}_i]_{i=1}^N$. Then the following two statements (6.28), (6.29) are equivalent.*

$$\sum_{S \in \mathcal{S}} f_S^* < 1 \tag{6.28}$$

$$\sum_{i=1}^{N} e_i \rho_i < 1, \forall \boldsymbol{e} \in \mathcal{E} \tag{6.29}$$

where $\mathcal{E}$ denote the set of extreme points of the polytope defined by the constraints of LP (6.24)

Proof. Note that the dual program of LP (6.24) is equivalent to LP (6.6) with $q_i / \bar{\eta}_i$ replaced with ρ_i for each $i = 1, \ldots, N$. Hence, the optimal value of the dual program of LP (6.6) equals $\max_{\boldsymbol{e} \in \mathcal{E}} \sum_{i=1}^{N} e_i \rho_i$. The result is now immediate from the strong duality theorem. □

6.3.4 Queue Evolution

Assuming that the system starts in an initial state $X(0) = [Q(0), U(0), V(0)]$. The queue evolution equation is given in the following equation (6.30)

$$q_i(t) = q_i(0) + a_i(t) - d_i(g_i(t)) \tag{6.30}$$

where $a_i(t)$ is the number of exogenous arrivals at q_i until time t, $g_i(t)$ is the cumulative resource allocated to q_i until time t and $d_i(g_i(t))$ is the number of departures from q_i until time t. These quantities are defined as follows

$$g_i(t) = \int_0^t f_i(Q(s))ds \tag{6.31}$$

$$a_i(t) = \max\{n : u_i(0) + \sum_{j=1}^{n-1} \xi_i(j) \leq t\} \tag{6.32}$$

$$d_i(t) = \max\{n : v_i(0) + \sum_{j=1}^{n-1} \eta_i(j) \leq t\} \tag{6.33}$$

In what follows, with a slight abuse of notation, we use $f_i(s)$ to represent $f_i(Q(s))$ for the sake of brevity. Also, define $g_i(t) := \int_0^t f_i(s)ds$.

6.4 Fluid Scaled Model

Let $\{x_r\}_{r\in\mathbb{N}} \subset \mathcal{X}$ denote a sequence of states such that $|x_r| \to \infty$ as $r \to \infty$, where $|.|$ is the L1 norm. Consider the sequence of fluid scaled processes $\{X^{(r)}(t)\}_{r\in\mathbb{N}}$ with the corresponding initial states $\{x_r/|x_r|\}_{r\in\mathbb{N}}$, defined as follows

$$X^{(r)}(t) := X(|x_r|t)/|x_r|, \forall r \in \mathbb{N}, t \in \mathbb{R}_+ \tag{6.34}$$

i.e.,

$$q_i^{(r)}(t) := q_i(|x_r|t)/|x_r|, \tag{6.35}$$

$$u_i^{(r)}(t) := u_i(|x_r|t)/|x_r|, \tag{6.36}$$

$$v_i^{(r)}(t) := v_i(|x_r|t)/|x_r|, \forall i = 1, \ldots, N, \ \forall r \in \mathbb{N}, \ t \in \mathbb{R}_+ \tag{6.37}$$

Note that the sequence of scaled initial states satisfy $|X^{(r)}(0)| = 1$ for each r. Also, define

$$g_i^{(r)}(t) := g_i(|x_r|t)/|x_r|, \tag{6.38}$$

$$a_i^{(r)}(t) := a_i(|x_r|t)/|x_r|, \tag{6.39}$$

$$d_i^{(r)}(t) := d_i(|x_r|t)/|x_r|, \ \forall r \in \mathbb{N}, t \in \mathbb{R}_+, i \in \{1, \ldots N\} \tag{6.40}$$

and,

$$f_i^{(r)}(t) := f_i(|x_r|t), \ \forall r \in \mathbb{N}, t \in \mathbb{R}_+, i \in \{1, \ldots N\} \tag{6.41}$$

We are interested in the limit of the sequence of the fluid scaled processes $\{X^{(r)}(t)\}_{r\in\mathbb{N}}$. We establish the convergence to the fluid limit in the Skorokhod topology, i.e., the space of so called Cadlag functions which are right continuous and have left limits everywhere.

We now present the key ideas behind the Fluid limit model. In section 6.6, we will show that the fluid scaled processes converge (u.o.c) to the fluid limit in the Skorokhod topology almost surely. The fluid limit model is considered to be stable if $\exists T$ such that the fluid limit $\bar{X}(t) = \mathbf{0}, \forall t \geq T$. The fluid limits are absolutely continuous functions, and are hence differentiable almost everywhere. In section 6.7, we will show that fluid limit is stable (by considering the trajectory of $L(\bar{Q}(t))$ as the Lyapunov function), provided (6.29) holds.

The rest of the chapter is the application of the framework originally developed in [21, 22] to our model. In section 6.8, the existence of finite moments of the process $X(t)$ is then established using the theory developed in [22]. The stability of the fluid model is a prerequisite for application of the results in [22]. Hence, showing the stability of the fluid limit model is the heart of this chapter.

We present the layout of the rest of the chapter.

- In section 6.5, we present the preliminary results and definitions which are required for the theory that follows in the chapter.

- In section 6.6, we show that the scaled processes considered in (6.34)-(6.41) converge to a fluid limit.

- Assuming that (6.29) holds, in section 6.7, we show that the trajectory of the fluid limit converges to $\mathbf{0}$ by some time T, for any initial choice of initial state $\bar{X}(0)$. This result is referred to as the stability of fluid limit.

- [21, 22] consider a multi-class queueing network. Suppose that the fluid limit for a mulit-class queueing network is stable, and the second moments of the arrival and service processes are finite. In [22], it is shown that the expected queue lengths converge to a steady state value.

  In section 6.8, we adopt this framework to our resource allocation problem. We use two key theorems from [22] along with the fluid limit stability result to prove that the queueing system is stable under our proposed flow control policy.

6.5 Preliminary Results and Definitions

Before presenting the fluid limit results, we provide the definitions and preliminary results which will be used in the proofs. This section contains the necessary theory which will be referred to in the later sections. It does not contain the results of the work presented in the chapter, just preliminary necessities. Therefore, the reader may skip reading this section, and refer back if required.

Definition 6.5.1 (Point-wise convergence)**.** *A sequence of real valued functions $h_n : Y \to \mathbb{R}, n \in \mathbb{N}$ converges point-wise to a function h if, for any $y \in Y$ and $\epsilon > 0$, there exists $N_{y,\epsilon}$ such that*

$$|h_n(y) - h(y)| < \epsilon, \forall n \geq N_{y,\epsilon} \tag{6.42}$$

Point-wise convergence is denoted as $h_n \to h$.

Definition 6.5.2 (Uniform convergence on compact sets)**.** *Let $(Y, \mathcal{Y})$ be topological space. Consider a sequence of functions $h_n : Y \to \mathbb{R}, n \in \mathbb{N}$. The sequence h_n is said to converge to h uniformly over compact sets if, for every compact set $Y_c \subseteq Y$ and $\epsilon > 0$, there exists $N_{Y_c,\epsilon}$ such that*

$$|h_n(y) - h(y)| < \epsilon, \forall n \geq N_{Y_c,\epsilon}\,, \forall y \in Y_c \tag{6.43}$$

Uniform convergence on compact sets is denoted as $h_n \to h$ u.o.c.

The following lemma (taken from [21]) will be repeatedly used to obtain u.o.c convergence of the fluid scaled processes (to the fluid limit), which will be done in the following section.

Lemma 6.5.1 ([21], Lemma 4.1)**.** *Let $\{h_n\}$ be a sequence of non-decreasing functions on $\mathbb{R}_+$ and h be a continuous function on $\mathbb{R}_+$. Assume that $h_n(t) \to h(t)$ for all rational $t \geq 0$. Then $h_n \to h$ u.o.c.*

Proof. See proof of Lemma 4.1 from [21]. □

The following lemma provides an important result. It establishes that the Lyapunov function $L(\bar{Q}(t))$ of the fluid limit $\bar{Q}(t)$ (which will be derived in the following section) stays at zero after it decreases to zero from its initial value. A similar lemma (Lemma 5.2 in [21]) was given in [21] without proof. Here, we provide a proof for completeness.

Lemma 6.5.2 ([21], Lemma 5.2)**.** *Let $h : [\tau, \infty) \to [0, \infty)$ be a non-negative function that is absolutely continuous and $\kappa > 0$ be a constant. Suppose that almost everywhere on $[\tau, \infty)$ all regular points t (i.e., where derivative exists), $h'(t) \leq -\kappa$ whenever $h(t) > 0$. Then h is non-increasing on $[\tau, \infty)$ and $h(t) \equiv 0$, for $t \geq \tau + h(\tau)/\kappa$.*

Proof. Since $h(.)$ is a non-negative function, $h(\tau) \geq 0$. Let $\tau_0 := \inf\{t \geq \tau : h(t) = 0\}$. By definition of τ_0, we have $h(t) \geq 0, \forall \tau \leq t \leq \tau_0$. Since it was given that $h'(t) < -\kappa$ whenever $h(t) > 0$, it follows that that h is non-increasing in the interval (τ, τ_0), and that $\tau_0 \leq \tau + h(\tau)/\kappa$.

We will now show that $h(t) = 0, \forall t \geq \tau_0$. We use proof by contradiction.

Suppose not and assume $h(t) > 0$ for some $t = \tau_2 > \tau_0$. Consider $\tau_1 := \sup\{t < \tau_2 : h(t) = 0\}$. Note that $\tau_0 \leq \tau_1 < \tau_2$. Since $h(.)$ is continuous, it follows that $h(t) > 0, \forall t \in (\tau_1, \tau_2]$. Since $h(.)$ is absolutely continuous, it follows from fundamental theorem of calculus for Lebesgue integration that

$$\int_{\tau_1}^{\tau_2} h'(s)ds = h(\tau_2) - h(\tau_1) = h(\tau_2) > 0 \tag{6.44}$$

Note that since $h(t) > 0, \forall t \in (\tau_1, \tau_2)$ and it is given that $h'(t) < 0$ at all regular points such that $h(t) > 0$, it must be true that $\int_{\tau_1}^{\tau_2} h'(s)ds < 0$, which contradicts (6.44). Hence, $h(t) = 0, \forall t \geq \tau_0$ and h is non-increasing on $[\tau, \infty)$. □

The following two theorems are key results from Renewal theory. For proofs, refer to Theorem 5.5.2 from [76]. We note that $\mu = \infty$ is a possible value in the following renewal theorems ([76]).

Theorem 6.5.3 (Strong Law of Large Numbers for Renewal Process)**.** *Let $\{m(t)\}_{t\geq 0}$ denote a renewal counting process with a mean inter-arrival period of $\mu > 0$. Then almost surely*

$$\lim_{t\to\infty} m(t)/t = 1/\mu$$

Theorem 6.5.4 (Elementary Renewal Theorem)**.** *Let $\{m(t)\}_{t\geq 0}$ denote a renewal counting process with a mean inter-arrival period of $\mu > 0$. Then $\lim_{t\to\infty} \mathbb{E}[m(t)]/t = 1/\mu$*

The following lemma is another renewal theory result, taken from [22]. For a proof, see Theorem 5.1 (on page 57) in [77].

Lemma 6.5.5 ([22], Lemma 5.2)**.** *Let $\{\zeta(k) : k \in \mathbb{N}\}$ be an i.i.d sequence taking values in $(0, \infty)$, and let $m(t)$ denote the counting process $m(t) := \max\{n \geq 1 : \zeta(1) + \ldots + \zeta(n-1) \leq t\}$. If $E[\zeta(1)] < \infty$, then for any integer $r \geq 1$,*

$$\lim_{t\to\infty} E\left[\left(\frac{m(t)}{t}\right)^r\right] = \left(\frac{1}{E[\zeta(1)]}\right)^r$$

Hence, under these conditions,

(a) for any $\delta > 0$, $\sup_{t \geq \delta} E[(m(t)/t)^r] < \infty$.

(b) The random variables

$$\{(m(t)/t)^r : t \geq 1\}$$

are uniformly integrable.

The following proposition (taken from [22]) is a result concerning a general Markov process on $\mathcal{X}$. This result will be used to show the stability under the proposed algorithm.

Proposition 1 ([22], Proposition 5.4)**.** *Let X be a Borel right Markov process on* $\mathcal{X}$*, let* $f : X \to \mathbb{R}_+$*, and define for some* $\delta > 0$*, and a closed set* $C \subseteq \mathcal{X}$.

$$V(x) := E_x[\int_0^{\tau_C(\delta)} f(X(t))dt], x \in \mathcal{X}$$

If V is everywhere finite, and uniformly bounded on C, then there exists $\kappa < \infty$ *such that*

$$\frac{1}{t}E_x[V(X(t))] + \frac{1}{t}\int_0^t E_x[f(X(s))]ds \leq \frac{1}{t}V(x) + \kappa, \quad t > 0, x \in \mathcal{X}$$

where, $\tau_C(\delta) := \min(t \geq \delta : X(t) \in C)$.

6.6 Existence of a Fluid Limit

In this section, we establish the existence of fluid limits for the various processes considered in (6.34 - 6.41).

Lemma 6.6.1 establishes the convergence to a fluid limit for the scaled arrival and departure processes in (6.39),(6.40). This lemma is equivalent to Lemma 4.2 from [21]. The proof of Lemma 6.6.1 is identical to the one given in [21].

Lemma 6.6.1. *Consider a sequence of states* $\{x_r\}_{r \in \mathbb{N}} \subset \mathcal{X}$ *such that* $|x_r| \to \infty$ *as* $r \to \infty$*. Suppose that* $u_i^{(r)}(0) \to \bar{u}_i(0)$ *and* $v_i^{(r)}(0) \to \bar{v}_i(0)$ *as* $r \to \infty$ *for each* $i = 1, \ldots, N$*. Then almost surely,*

$$a_i^{(r)}(t) \to \bar{\xi}\,(t - \bar{u}_i(0))^+ \ \textit{u.o.c. as } r \to \infty \tag{6.45}$$

$$d_i^{(r)}(t) \to \bar{\eta}\,(t - \bar{v}_i(0))^+ \ \textit{u.o.c. as } r \to \infty \tag{6.46}$$

for each $i = 1, \ldots, N$, $\forall t \geq 0$ *where* $(.)^+ := \max\{0, .\}$

Proof. The proof is given in section 6.9. □

The following Lemma 6.6.1 establishes the convergence to a fluid limit for the scaled residual arrival time and residual service processes, which are part of the scaled state $X^{(r)}(t)$. This lemma is similar to Lemma 4.3 from [21]. The lemma is is slightly different from Lemma 4.3 in [21] due to our considered scheduling policy, specifically the processes $\{g_i^{(r)}(t)\}_{i=1}^N$. The proof here uses similar techniques as in [21].

Lemma 6.6.2. *Consider a sequence of states $\{x_r\}_{r\in\mathbb{N}} \subset \mathcal{X}$ such that $|x_r| \to \infty$ as $r \to \infty$. Suppose that $u_i^{(r)}(0) \to \bar{u}_i(0)$, $v_i^{(r)}(0) \to \bar{v}_i(0)$ as $r \to \infty$. Also, assume that $g_i^{(r)}(t) \to \bar{g}_i(t) \leq t$ almost surely u.o.c. as $r \to \infty$, $\forall t \geq 0$, where $\bar{g}_i(t)$ is an absolutely continuous function, for each $i = 1, \ldots, N$. Then for each $i = 1, \ldots, N$, almost surely,*

$$u_i^{(r)}(t) \to (\bar{u}_i(0) - t)^+ \text{ u.o.c. as } r \to \infty \tag{6.47}$$

$$v_i^{(r)}(t) \to (\bar{v}_i(0) - \bar{g}_i(t))^+ \text{ u.o.c. as } r \to \infty \tag{6.48}$$

$\forall t \geq 0$. Also, $\{u_i^{(r)}(t)\}_{r\in\mathbb{N}}$ and $\{v_i^{(r}(t)\}_{r\in\mathbb{N}}$ are uniformly integrable.

Proof. The proof is given in section 6.9. □

The following Lemma 6.6.1 establishes the convergence to a fluid limit for the scaled queue process $Q^{(r)}(t)$ and the scaled cumulative service process $\{g_i^{(r)}(t)\}_{i=1}^N$.

Lemma 6.6.3. *Consider a sequence of states $\{x_r\}_{r\in\mathbb{N}} \subset \mathcal{X}$ such that $|x_r| \to \infty$ as $r \to \infty$. There exists a subsequence $\{r_k\}_{k\in\mathbb{N}}$ such that $X^{(r_k)}(0) \to \bar{X}(0)$ as $k \to \infty$, i.e., $q_i^{(r_k)}(0) \to \bar{q}_i(0)$, $u_i^{(r_k)}(0) \to \bar{u}_i(0)$ and $v_i^{(r_k)}(0) \to \bar{v}_i(0)$ for each $i = 1, \ldots, N$. Also, as $k \to \infty$, almost surely,*

$$q_i^{(r_k)}(t) \to \bar{q}_i(t) \text{ u.o.c.} \tag{6.49}$$

$$g_i^{(r_k)}(t) \to \bar{g}_i(t) \text{ u.o.c.} \tag{6.50}$$

for each $i = 1, \ldots, N$ and $\forall t \geq 0$, where, $\bar{g}_i(t)$ is a Lipschitz continuous function (with a Lipschitz constant of 1), and,

$$\bar{q}_i(t) = \bar{q}_i(0) + \bar{\xi}_i(t - \bar{u}_i(0))^+ - \bar{\eta}_i(\bar{g}_i(t) - \bar{v}_i(0))^+ \tag{6.51}$$

Proof. The proof is given in section 6.9. □

Recall that $X^{(r)}(t) = [Q^{(r)}(t), U^{(r)}(t), V^{(r)}(t)]$ for each $r \in \mathbb{N}$ and $\forall t \geq 0$. Also, $Q^{(r)}(t) = [q_i^{(r)}(t)]_{i=1}^N$, $U^{(r)}(t) = [u_i^{(r)}(t)]_{i=1}^N$ and $V^{(r)}(t) = [v_i^{(r)}(t)]_{i=1}^N$. It follows from Lemma 6.6.3 that the sequence of scaled processes $\{Q^{(r)}(t)\}_{r\in\mathbb{N}}, \{[g_i^{(r)}]_{i=1}^N\}_{r\in\mathbb{N}}$ converge to a fluid limit $\bar{Q}(t), [\bar{g}_i(t)]_{i=1}^N$ u.o.c a.s. along the sub-sequence $\{r_k\}_{k\in\mathbb{N}}$. Also from Lemma 6.6.3, $X^{(r_k)}(0) \to \bar{X}(0)$ as $k \to \infty$.

Hence, the sequences $\{u_i^{(r_k)}(0), v_i^{(r_k)}(0)\}_{k\in\mathbb{N}}$ and $\{g_i^{(r_k)}(t)\}_{k\in\mathbb{N}} \to \bar{g}_i(t)$ satisfy the conditions necessary for Lemma 6.6.2. It follows from Lemma 6.6.2, that $u_i^{(r_k)}(t) \to \bar{u}_i(t)$ and $v_i^{(r_k)}(t) \to \bar{v}_i(t)$ u.o.c a.s. for each $i = 1, \ldots, N$. Hence, it follows that $U^{(r_k)}(t) \to \bar{U}(t)$ and $V^{(r_k)}(t) \to \bar{V}(t)$ u.o.c. a.s. as $k \to \infty$.

Therefore, the Lemmas in this section establish that the sequence of scaled processes $\{X^{(r)}(t)\}_{r\in\mathbb{N}}$, $\{[g_i^{(r)}]_{i=1}^N\}_{r\in\mathbb{N}}$ converge to a fluid limit $\bar{X}(t), [\bar{g}_i(t)]_{i=1}^N$ u.o.c a.s. along the sub-sequence $\{r_k\}_{k\in\mathbb{N}}$. The fluid limit satisfies the following equations

$$|\bar{X}(0)| = \sum_{i=1}^{N} \bar{q}_i(0) + \sum_{i=1}^{N} \bar{u}_i(0) + \sum_{i=1}^{N} \bar{v}_i(0) = 1 \tag{6.52}$$

and

$$\bar{u}_i(t) = (\bar{u}_i(0) - t)^+ \tag{6.53}$$

$$\bar{v}_i(t) = (\bar{v}_i(0) - \bar{g}_i(t))^+ \tag{6.54}$$

$$\bar{q}_i(t) = \bar{q}_i(0) + \bar{\xi}_i(t - \bar{u}_i(0))^+ - \bar{\eta}_i(\bar{g}_i(t) - \bar{v}_i(0))^+ \tag{6.55}$$

for each $i = 1, \ldots, N$ and $\forall t \geq 0$. In the following section, in Lemma 6.7.1, we will show that

$$\frac{d\bar{g}_i(t)}{dt} = \frac{\bar{q}_i(t)/\bar{\eta}_i}{L(\bar{Q}(t))} \tag{6.56}$$

for each $i = 1, \ldots, N$ and $t : \bar{Q}(t) \in \mathbb{Z}_+^N - \mathbf{0}$. The equations (6.52)- (6.56) define the trajectory of the fluid limit. We consider the fluid limit to be stable if the trajectory reaches $\mathbf{0}$ state for any initial state $\bar{X}(0)$. In the following section, we provide the precise definition of a stable fluid limit model.

6.7 Stability of the Fluid limit

In this section, we consider the derivatives of the fluid limit to study its trajectory. We show that the Lyapunov function $L(\bar{Q}(t))$ has a negative derivative at all regular points $t > \tau^*$ for some $\tau^* > 0$. Using this, we establish the stability of the fluid limit model (as defined in the following).

Definition 6.7.1. *Any solution* $\{\bar{X}(t)\}_{t\geq 0} = [\bar{Q}(t), \bar{U}(t), \bar{V}(t)]_{t\geq 0}$ *satisfying the equations (6.52) - (6.56) is a fluid limit for our model. We say that the fluid limit (or the fluid model) is stable if and only if there exists a* $T > 0$ *(which depends only on* $\bar{\eta}_i$*,* $\bar{v}_i$*) such that* $\bar{X}(t) = \mathbf{0}, \forall t \geq T$*. The value of* T *must not depend on the choice of the initial state* $\bar{X}(0)$*.*

The function $\bar{g}_i(t)$ is the fluid limit of the cumulative service process. In Lemma 6.6.3, we have established that $\bar{g}_i(t)$ is a Lipschitz continuous function, which implies that it is an absolutely continuous function. It follows that $\bar{g}_i(t)$ is differentiable at almost every t. The following Lemma 6.7.1 provides the derivative of $\bar{g}_i(t)$ for all points t such that $L(\bar{Q}(t)) > 0$.

Lemma 6.7.1. *Suppose* $L(\bar{Q}(\tau)) > 0$*. The function* $\bar{g}_i(t)$ *is differentiable at* $t = \tau$ *and the derivative* $\bar{g}_i'(\tau)$ *is given by*

$$\bar{g}_i'(\tau) = \bar{f}_i(\tau) := \frac{\bar{q}_i(\tau)/\bar{\eta}_i}{L(\bar{Q}(\tau))}$$

Proof. The proof is given in section 6.9. □

The following lemma shows that the fluid limit processes $\bar{U}(t), \bar{V}(t)$ are zero for $t > \tau^*$ for some $\tau^* \geq 0$. This is a crucial component in establishing stability of the fluid limit.

Lemma 6.7.2. *There exists* $\tau^* \geq 0$ *which depends only on* $\{\bar{\xi}_i, \bar{\eta}_i\}_{i=1}^N$ *such that for each* $i \in \{1, \dots N\}$

$$t \geq \bar{u}_i(0) \text{ and } \bar{g}_i(t) \geq \bar{v}_i(0), \forall t \geq \tau^* \tag{6.57}$$

Proof. The proof is given in section 6.9. □

The following two lemmas are concerning the negative derivative of the Lyapunov function $L(\bar{Q}(t))$ whenever $\bar{Q}(t) \neq \mathbf{0}$. Recall from (6.26) that $L(\bar{Q}(t)) := \max_{\boldsymbol{e}\in\mathcal{E}} L_{\boldsymbol{e}}(\bar{Q}(t))$. Lemma 6.7.3 is a derivative result for $L_{\boldsymbol{e}}(\bar{Q}(t))$, and Lemma 6.7.4 is the negative derivative result for $L(\bar{Q}(t))$.

Lemma 6.7.3. *Suppose* $L(\bar{Q}(\tau)) > 0$ *for some* $\tau > \tau^*$*, where* τ^* *is defined in Lemma 6.7.2. Then*

$$\frac{dL_{\boldsymbol{e}}(\bar{Q}(t))}{dt}\Big|_{t=\tau} = \sum_{i=1}^{N} e_i\rho_i - \frac{L_{\boldsymbol{e}}(\bar{Q}(\tau))}{L(\bar{Q}(\tau))} \tag{6.58}$$

Lemma 6.7.4 (Negative gradient of Lyapunov function)**.** *Suppose $L(\bar{Q}(\tau)) > 0$ for some regular point (i.e., where the derivative exists) $\tau > \tau^*$, where τ^* is defined in Lemma 6.7.2. Then*

$$\frac{dL(\bar{Q}(t))}{dt}|_{t=\tau} \leq \max_{\boldsymbol{e}\in\mathcal{E}} \sum_{i=1}^{N} e_i\rho_i - 1 \tag{6.59}$$

Proof. Note that $\{\bar{q}_i(t)\}_{i=1}^N$ are absolutely continuous functions from Lemma 6.6.3. Since $L(\bar{Q}(t)) = \max_{\boldsymbol{e}\in\mathcal{E}} L_{\boldsymbol{e}}(\bar{Q}(t))$, it follows that $L(\bar{Q}(t))$ is also an absolutely continuous function. Hence, it is differentiable almost everywhere, and τ is given to be a regular point, where the derivative exists.

Let $\mathcal{E}_1$ denote the set of all $\boldsymbol{e} \in \mathcal{E}$ such that $L_{\boldsymbol{e}}(\bar{Q}(\tau)) = L(\bar{Q}(\tau))$. Note that $\frac{dL_{\boldsymbol{e}}(\bar{Q}(t))}{dt}|_{t=\tau} = \sum_{i=1}^N e_i\rho_i - 1$ for each $\boldsymbol{e} \in \mathcal{E}_1$ from Lemma 6.7.4. Since $L(\bar{Q}(t)) = \max_{\boldsymbol{e}\in\mathcal{E}} L_{\boldsymbol{e}}(\bar{Q}(t)), \forall t \geq 0$, and since τ is a regular point, it follows that the derivative

$$\frac{dL(\bar{Q}(t))}{dt}|_{t=\tau} = \sum_{i=1}^{N} e_i\rho_i - 1 \tag{6.60}$$

for some $\boldsymbol{e} \in \mathcal{E}_1$. Hence, the result follows.

□

In Lemma 6.7.2, we established that $\bar{U}(t), \bar{V}(t)$ are zero for $t > \tau^*$. The following Theorem 6.7.5 establishes that $\bar{X}(t) = \mathbf{0}$ for $t \geq T$ provided $\max_{\boldsymbol{e}\in\mathcal{E}} e_i\rho_i < 1$. It follows that the fluid limit is stable.

Theorem 6.7.5 (Stability of the fluid limit)**.** *Suppose* $\max_{\boldsymbol{e}\in\mathcal{E}} e_i\rho_i < 1$. *Then there exists $T(> \tau^*)$ which only depends on $\{\bar{\xi}_i, \bar{\eta}_i\}_{i=1}^N$, such that $\bar{X}(t) = \mathbf{0}, \forall t \geq T$, where τ^* is given in Lemma 6.7.2.*

Proof. Firstly, note that $\bar{q}_i(t) \leq \bar{q}_i(0) + \bar{\xi}_i t$ for each $t \geq 0$. Hence,

$$\bar{q}_i(\tau^*) \leq \bar{q}_i(0) + \bar{\xi}_i\tau^* \tag{6.61}$$

$$\leq 1 + \bar{\xi}_i\tau^* \tag{6.62}$$

where τ^* is from Lemma 6.7.2. Since $L(Q)$ satisfies (6.26), it follows that

$$L(\bar{Q}(\tau^*)) \leq \max_{\boldsymbol{e}\in\mathcal{E}} \sum_{i=1}^{N} e_i \left(1/\bar{\eta}_i + \rho_i\tau^*\right) \tag{6.63}$$

Since $\max_{\boldsymbol{e}\in\mathcal{E}} e_i\rho_i < 1$, it follows that $\kappa := 1 - \max_{\boldsymbol{e}\in\mathcal{E}} e_i\rho_i > 0$. It follows from Lemma 6.7.4 that for almost every $t > \tau^*$ such that $L(\bar{Q}(t)) > 0$

$$\frac{dL(\bar{Q}(t))}{dt} \leq -\kappa \tag{6.64}$$

It follows from (6.63) and Lemma 6.5.2 that $L(\bar{Q})(t) = 0, \forall t \geq \tau^* + \frac{1}{\kappa}\left(\max_{e\in\mathcal{E}} \sum_{i=1}^{N} e_i\,(1/\bar{\eta}_i + \rho_i\tau^*)\right)$. Since $L(\bar{Q}(t)) = 0$, only if $\bar{Q}(t) \equiv \mathbf{0}$, it follows that $\bar{Q}(t) = \mathbf{0}$ for $t \geq T$.

Since $T > \tau^*$, it follows from Lemma 6.7.2 that $\bar{u}_i(t) = 0, \bar{v}_i(t) = 0$ for each $i = 1, \ldots, N$ and $t \geq T$. Hence, $\bar{U}(t) = \mathbf{0}, \bar{V}(t) = \mathbf{0}$ for $t \geq T$, which completes the proof. □

6.8 Stability

We now have everything in place to apply the theory from [22]. Using the results of [22], we show that the expectation of the Markov process $X(t)$ converges to a finite value in the L1 norm as $t \to \infty$, whenever the fluid model is stable. This completes the proof of stability under the proposed algorithm.

To show the finite expectation result, we require the following two results from [22]. The proofs of the following propositions are identical to the proofs given in [21], with exception of a few additional steps. The departure process in our model is given by $d_i(g_i(t))$, and the equivalent departure process in [21] has the form $d_i(t)$. Hence, a few additional steps are required to address this change in the following manner. In our process, we can bound $d_i(g_i(t))$ by $d_i(t)$, since $g_i(t) \leq t$ (because $g_i(t) = \int_0^t f_i(s)ds$ and $f_i(s) \leq 1$), and this is sufficient to apply the following results. We use this bounding argument in (6.143) in proof of Proposition 2, and in (6.161) in proof of Proposition 3. The rest of the proofs is unchanged from [22].

Proposition 2. *Suppose Assumption 7 and Assumption 8 holds, and that the fluid model is stable. Then there exists $t_0 > 0$ such that*

$$\lim_{|x|\to\infty} \frac{1}{|x|^2} E[|X^x(t_0|x|)|^2] = 0 \tag{6.65}$$

where $\{X^{(x)}(t)\}_{t\geq 0}$ is the Markov process $\{X(t)\}_{t\geq 0}$ starting from the initial state $X(0) = x$, and $|.|$ is the L1 norm.

Proof. See section 6.9 □

Proposition 3. *Suppose that Assumption 7 and Assumption 8 are satisfied, and that the fluid model is stable. Then for some constant $c^* < \infty, \delta > 0$ and a compact set $C \subset \mathcal{X}$,*

$$E\left[\int_0^{\tau_C(\delta)} (1 + |X^x(t)|)dt\right] \leq c^*(|x|^2 + 1), x \in \mathcal{X}$$

where $\tau_C(\delta) := \min(t \geq \delta : X(t) \in C)$.

Proof. See section 6.9 □

Since Proposition 2 and Proposition 3 hold, the following theorem from[22] holds for $p = 1$ (which follows from Proposition 1 as given in [22]).

Theorem 5.5, [22]. *Suppose Assumption 7 and Assumption 8 hold, and that the fluid model is stable. Then there exists a constant $\kappa_p < \infty$ such that*

$$\frac{1}{t}\int_0^t E_x[|Q(s)|^p]ds \leq \kappa_p \left\{\frac{1}{t}|x|^{p+1} + 1\right\}, \quad t > 0, x \in \mathcal{X}$$

In particular, for each initial condition,

$$\limsup_{t\to\infty} \frac{1}{t}\int_0^t E_x[|Q(s)|^p]ds \leq \kappa_p.$$

where $\mathbb{E}_x[.]$ is the expectation, given the initial state is x.

Proof. See Theorem 5.5 from [22]. □

Theorem 6.8.1. *Suppose $\max_{\boldsymbol{e}\in\mathcal{E}} e_i\rho_i < 1$, the system is stable under the proposed scheduling policy.*

Proof. It follows from Theorem 6.7.5 that the fluid model is stable when $\max_{\boldsymbol{e}\in\mathcal{E}} e_i\rho_i < 1$. Now from Assumption 7 and Assumption 8, it can be observed that Propositions 1,2,3 hold. Hence, Theorem 5.5 of [22] holds with $p = 1$.

It follows from Theorem 5.5 of [22] with $p = 1$ that there exists $\kappa_1 > 0$ such that

$$\limsup_{t\to\infty} \frac{1}{t}\int_0^t \sum_{i=1}^{N} \mathbb{E}_x[q_i(s)]ds \leq \kappa_1, \text{ for each } x \in \mathcal{X} \tag{6.66}$$

where $\mathbb{E}_x[.]$ is the expectation, given the initial state is x.

Since (6.66) holds for each $x \in \mathcal{X}$, it follows that

$$\limsup_{t\to\infty} \frac{1}{t}\int_0^t \sum_{i=1}^{N} \mathbb{E}[q_i(s)]ds \leq \kappa_1 \tag{6.67}$$

Hence, the system is stable under the proposed scheduling policy. □

6.9 Theoretical Results

Proof of Lemma 6.6.1. Let $m(t)$ denote the renewal counting process corresponding to arrival process at queue i (i.e., inter-renewal periods are same as the inter-arrival periods at queue i).

$$m(t) := \max\{n : \sum_{j=1}^{n} \xi_i(j) \leq t\} \tag{6.68}$$

From SLLN for renewal processes (i.e., Theorem 6.5.3), it follows that

$$\frac{m(|x_r|t)}{|x_r|t} \to \bar{\xi}_i, \ \forall t > 0, \text{ a.s.} \tag{6.69}$$

Hence, almost surely $\forall t \geq 0$

$$m(|x_r|t)/|x_r| \to \bar{\xi}_i t \tag{6.70}$$

The first arrival at queue i does not happen before the residual time $|u_i(0)|$ has elapsed, hence $a_i(t) = 0, \forall t < u_i(0)$. For $t \geq u_i(0)$, the arrival process at q_i is a renewal counting process. It follows that

$$a_i(t) = \begin{cases} 0 & \text{if } t < u_i(0) \\ 1 + m((t - u_i(0))^+) & \text{if } t \geq u_i(0) \end{cases} \tag{6.71}$$

It follows that,

$$a_i(|x_r|t)/|x_r| = \begin{cases} 0 & \text{if } t < u_i^{(r)}(0) \\ \frac{1}{|x_r|}\left(1 + m(|x_r|(t - u_i^{(r)}(0))^+)\right) & \text{if } t \geq u_i^{(r)}(0) \end{cases} \tag{6.72}$$

Hence, $\forall t \geq 0$,

$$\frac{1}{|x_r|} m(|x_r|(t - u_i^{(r)}(0))^+) \leq a_i^{(r)}(t) \leq \frac{1}{|x_r|}\left(1 + m(|x_r|(t - u_i^{(r)}(0))^+)\right) \tag{6.73}$$

The right side inequality is immeditate from (6.72). For the left side inequality note that, $m(|x_r|(t - u_i^{(r)}(0))^+) = 0$ when $t < u_i^{(r)}(0)$, and, $m(|x_r|(t - u_i^{(r)}(0))^+)/|x_r| \leq a_i^{(r)}(t)$ when $t \geq u_i^{(r)}(0)$ (from (6.72)). Hence,

$$\lim_{r\to\infty} a_i^{(r)}(t) = \lim_{r\to\infty} \frac{1}{|x_r|} m(|x_r|(t - u_i^{(r)}(0))^+) \tag{6.74}$$

From (6.70) and (6.74), it follows that $a_i^{(r)}(t) \to \bar{\xi}_i(t - \bar{u}_i(0))^+$ almost surely as $r \to \infty$. $\{a_i^{(r)}(t)\}_l$ are a sequence of non-decreasing functions, and $\bar{\xi}_i(t - \bar{u}_i(0))^+$ is a continuous function. Hence, u.o.c convergence in (6.45) follows from Lemma 6.5.1.

Similarly, (6.46) follows from the same arguments. □

Proof of Lemma 6.6.2. Note that the residual time to arrival at time t for queue i, i.e., $u_i(t)$ can be written as

$$u_i(t) = u_i(0) + \sum_{j=1}^{a_i(t)+1} \xi_i(j) - t \tag{6.75}$$

From (6.45) of Lemma 6.6.1, it follows that $a_i(|x_r|t)/|x_r| \to \bar{\xi}_i(t - \bar{u}_i(0))^+$ almost surely. Hence for $t > 0$, $a_i(|x_r|t) \to \infty$ almost surely as $r \to \infty$. Hence, for each $t > 0$, by the SLLN, we have almost surely

$$\lim_{r\to\infty} \frac{1}{|x_r|} \sum_{j=1}^{a_i(|x_r|t)+1} \xi_i(j) = \lim_{r\to\infty} \frac{1}{\bar{\xi}_i} \frac{a_i(|x_r|t) + 1}{|x_r|} \tag{6.76}$$

$$= \lim_{r\to\infty} \frac{a_i^{(r)}(t)}{\bar{\xi}_i} \tag{6.77}$$

$$= (t - \bar{u}_i(0))^+ \tag{6.78}$$

$\frac{1}{|x_r|} \sum_{j=1}^{a_i(|x_r|t)+1} \xi_i(j)$ is non-decreasing function of t and $(t - \bar{u}_i(0))^+$ is a continuous function. It follows from Lemma 6.5.1 that almost surely for each $t \geq 0$

$$\frac{1}{|x_r|} \sum_{j=1}^{a_i(|x_r|t)+1} \xi_i(j) \to (t - \bar{u}_i(0))^+ \text{ u.o.c.} \tag{6.79}$$

From (6.75) and (6.79), it follows that $u_i^{(r)}(t) \to (\bar{u}_i(0) - t) + (t - \bar{u}_i(0))^+$u.o.c. a.s. as $r \to \infty$. Hence, (6.47) follows.

To show uniform integrability of $u^{(r)}(t)$, note that from (6.75)

$$E[u_i^{(r)}(t)] = u_i(0)/|x_r| + \frac{1}{|x_r|} E[\sum_{j=1}^{a_i(|x_r|t)+1} \xi_i(j)] - t \tag{6.80}$$

Using Wald's identity, we have

$$E[u_i^{(r)}(t)] = u_i(0)/|x_r| + \frac{1}{\bar{\xi}_i|x_r|} E[a_i(|x_r|t) + 1] - t \tag{6.81}$$

By elementary renewal theorem (Theorem 6.5.4), $E[u_i^{(r)}(t)] \to (\bar{u}_i(0) - t)^+$. The proof of uniform integrability now follows from Theorem 4.5.4 of [76], since $E[u_i^{(r)}(t)] \to (\bar{u}_i(0) - t)^+$ and $u_i^{(r)}(t) \to (\bar{u}_i(0) - t)^+$ a.s. as $r \to \infty$.

Similarly, the residual service requirement of HoL flow at queue i, i.e., $v_i(t)$ can be written as

$$v_i(t) = v_i(0) + \sum_{j=1}^{d_i(g_i(t))+1} \eta_i(j) - g_i(t) \tag{6.82}$$

Here, $g_i(t)$ is the cumulative service provided until time t, and $d_i(g_i(t))$ is the number of departures until time t.

From (6.48) of Lemma 6.6.1, and since $g_i^{(r)}(t) \to \bar{g}_i(t)$ a.s. u.o.c, it follows that $d_i^{(r)}(g_i^{(r)}(t)) \to \bar{\eta}_i(\bar{g}_i(t) - \bar{v}_i(0))^+$ a.s. u.o.c. as $r \to \infty$. Hence, $d_i(g_i(|x_r|t))/|x_r| \to \bar{\eta}_i(\bar{g}_i(t) - \bar{v}_i(0))^+$ a.s. u.o.c. as $r \to \infty$. Hence for $t > 0$, $d_i(g_i(|x_r|t)) \to \infty$ almost surely as $r \to \infty$ for any $t > 0$. Hence, for any $t > 0$, by the SLLN, we have almost surely

$$\lim_{r\to\infty} \frac{1}{|x_r|} \sum_{j=1}^{d_i(g_i(|x_r|t))+1} \eta_i(j) = \lim_{r\to\infty} \frac{1}{\bar{\eta}_i} \frac{d_i(g_i(|x_r|t)) + 1}{|x_r|} \tag{6.83}$$

$$= \lim_{r\to\infty} \frac{d_i^{(r)}(g_i^{(r)}(t))}{\bar{\eta}_i} \tag{6.84}$$

$$= (\bar{g}_i(t) - \bar{v}_i(0))^+ \tag{6.85}$$

$\frac{1}{|x_r|} \sum_{j=1}^{d_i(g_i(|x_r|t))+1} \eta_i(j)$ is non-decreasing function of t and $(\bar{g}_i(t) - \bar{v}_i(0))^+$ is a continuous function (since it is given that $\bar{g}_i(t)$ is continuous). It follows from Lemma 6.5.1 that almost surely for each $t \geq 0$

$$\frac{1}{|x_r|} \sum_{j=1}^{d_i(g_i(|x_r|t))+1} \eta_i(j) \to (\bar{g}_i(t) - \bar{v}_i(0))^+ \text{ u.o.c.} \tag{6.86}$$

From (6.82) and (6.86), it follows that $v_i^{(r)}(t) \to (\bar{v}_i(0) - \bar{g}_i(t)) + (\bar{g}_i(t) - \bar{v}_i(0))^+$u.o.c. a.s. as $r \to \infty$. Hence, (6.48) follows.

To show uniform integrability of $v^{(r)}(t)$, note that from (6.82)

$$E[v_i^{(r)}(t)] = v_i(0)/|x_r| + \frac{1}{|x_r|} E[\sum_{j=1}^{d_i(g_i(|x_r|t))+1} \eta_i(j)] - g_i^{(r)}(t) \tag{6.87}$$

Using Wald's identity, we have

$$E[v_i^{(r)}(t)] = \bar{v}_i(0)/|x_r| + \frac{1}{\bar{\eta}_i} E[d_i^{(r)}(g_i^{(r)}(t)) + 1/|x_r|] - g_i^{(r)}(t) \tag{6.88}$$

By elementary renewal theorem (Theorem 6.5.4), we have

$$\lim_{r\to\infty} E[d_i^{(r)}(g_i^{(r)}(t))] = \lim_{r\to\infty} \bar{\eta}_i (E[g_i^{(r)}(t)] - \bar{v}_i(0))^+ \tag{6.89}$$

By definition, $g_i^{(r)}(t) \leq t$, for each $r \in \mathbb{N}$. Since $0 \leq g_i^{(r)}(t) \leq t$ for each $r \in \mathbb{N}$ and since $\lim_{t\to\infty} g_i^{(r)}(t) = \bar{g}_i(t)$ almost surely, it follows that $\lim_{r\to\infty} E[g_i^{(r)}(t)] = \bar{g}_i(t)$. Therefore, $E[d_i^{(r)}(g_i^{(r)}(t))] \to \bar{\eta}_i(\bar{g}_i(t) - \bar{v}_i(0))^+$ and $E[g_i^{(r)}(t)] \to \bar{g}_i(t)$ as $r \to \infty$. Now from (6.88), it follows that $E[v_i^{(r)}(t)] \to \bar{v}_i(0) - \bar{g}_i(t) + (\bar{g}_i(t) - \bar{v}_i(0))^+$. The proof of uniform integrability now follows from Theorem 4.5.4 of [76], since $E[v_i^{(r)}(t)] \to (\bar{v}_i(0) - \bar{g}_i(t))^+$ and $v_i^{(r)}(t) \to (\bar{v}_i(0) - \bar{g}_i(t))^+$ a.s. □

Proof of Lemma 6.6.3. Since $|X^{(r)}(0)| = 1, \forall r \in \mathbb{N}$, it follows that $q_i^{(r)}(0) \leq 1, u_i^{(r)}(0), v_i^{(r)}(0) \forall r$. Hence, along some subsequence $\{r_l\}_{l\in\mathbb{N}} \subset \{r\}_{r\in\mathbb{N}}$,

$$q_i^{(r_l)}(0) \to \bar{q}_i(0) \tag{6.90}$$

$$u_i^{r_l}(0) \to \bar{u}_i(0) \tag{6.91}$$

$$v^{r_l}(0) \to \bar{v}_i(0) \tag{6.92}$$

for each i, as $l \to \infty$.

Note that $q_i^{(r)}(t) = q_i^{(r)}(0) + a_i^{(r)}(t) - d_i^{(r)}(g_i^{(r)}(t))$. We establish the convergence to limit of each term in the this equation.

Consider the sequence of processes $\{a_i^{(r_l)}(t)\}_{l\in\mathbb{N}}$, it follows from Lemma 6.6.1 that

$$a_i^{(r_l)}(t) \to \bar{\xi}\,(t - \bar{u}_i(0))^+ \text{ u.o.c. a.s.} \tag{6.93}$$

as $l \to \infty$.

Consider $\{g_i^{(r_l)}(t)\}_{l\to\infty}$. By definition,

$$g_i^{(r_l)}(t) = \frac{1}{|x_{r_l}|}\int_{s=0}^{|x_{r_l}|t} f_i(s)ds \tag{6.94}$$

Observe that since $f_i(t)$ is a fraction, $0 \leq f_i(t) \leq 1, \forall t \geq 0$. Therefore, for any $t, s \geq 0$

$$|g_i^{(r_l)}(s) - g_i^{(r_l)}(t)| = \frac{1}{|x_{r_l}|}\left|\int_0^{|x_{r_l}|s} f_i(\phi)d\phi - \int_0^{|x_{r_l}|t} f_i(\phi)d\phi\right| \tag{6.95}$$

$$= \frac{1}{|x_{r_l}|}\left|\int_{|x_{r_l}|t}^{|x_{r_l}|s} f_i(\phi)d\phi\right| \tag{6.96}$$

$$\leq |s - t| \tag{6.97}$$

Therefore, $\{g_i^{(r_l)}(t)\}_{l\in\mathbb{N}}$ form an equicontinuous family of functions. Using Arzela-Ascoli theorem, there must exist a subsequence $\{r_k\}_{k\in\mathbb{N}} \subset \{r_l\}_{l\in\mathbb{N}}$ such that $\{g_i^{(r_k)}(t)\}_{k\in\mathbb{N}}$ that converges u.o.c to $\bar{g}_i(t)$(say) for each $i = 1, \ldots, N$. It follows from Lemma 6.6.1 that

$$d_i^{r_k}(g_i^{r_k}(t)) \to \bar{\eta}(\bar{g}_i(t) - \bar{v}_i(0))^+ \text{a.s. u.o.c} \tag{6.98}$$

as $k \to \infty$.

Moreover, $\bar{g}_i(t)$ is also Lipschitz-continuous as can be seen from the following. Since $|g_i^{(r_k)}(s) - g_i^{(r_k)}(t)| \leq |s - t|, \forall k \in \mathbb{N}$, we have

$$\lim_{k\to\infty} |g_i^{(r_k)}(s) - g_i^{(r_k)}(t)| \leq |s - t| \tag{6.99}$$

$$|\bar{g}_i(s) - \bar{g}_i(t)| \leq |s - t| \tag{6.100}$$

Hence, the proof is completed. □

Proof of Lemma 6.7.1. By definition,

$$f_i^{(r_k)}(t) = \begin{cases} (q_i^{(r_k)}(t)/\bar{\eta}_i)/L(Q^{(r_k)}(t)) & \text{if } L(Q^{(r_k)}(t)) > 0 \\ 0 & \text{o.w.} \end{cases} \tag{6.101}$$

Since $\bar{g}_i(t)$ is Lipschitz continuous for each i, it follows that $\bar{Q}(t)$ are absolutely continuous functions from Lemma 6.6.3. Hence, $L(\bar{Q}(t))$ is an absolutely continuous function by the definition in (6.26). By continuity, it follows that there exists a $\delta > 0$, such that $L(\bar{Q}(t)) > 0$ for each $\tau - \delta \le t \le \tau + \delta$. Hence from Lemma 6.6.3 and (6.101), almost surely

$$f_i^{(r_k)}(t) \to (\bar{q}_i(t)/\bar{\eta}_i)/L(\bar{Q}(t)), \forall t \in [\tau - \delta, \tau + \delta] \tag{6.102}$$

Now for the derivative of $\bar{g}_i(t)$, note that by definition

$$\bar{g}_i(t) = \lim_{k\to\infty} \frac{\int_0^{|x_{r_k}|t} f_i(s)ds}{|x_{r_k}|} \tag{6.103}$$

$$= \lim_{k\to\infty} \int_0^t f_i^{(r_k)}(\phi)d\phi \tag{6.104}$$

Since the function $\bar{g}_i(t)$ is Lipschitz continuous, it is differentiable almost everywhere. At any regular point, the derivative can be written as

$$\bar{g}'(t) = \lim_{\delta_1\to 0} \lim_{k\to\infty} \frac{\int_t^{t+\delta_1} f_i^{(r_k)}(\phi)d\phi}{\delta_1} \tag{6.105}$$

Note that $f_i^{(r_k)}(t) \to \bar{f}_i(t) := (\bar{q}_i(t)/\bar{\eta}_i)/L(\bar{Q}(t))$ a.s. in the interval $[\tau - \delta, \tau + \delta]$ from (6.102), and also $f_i^{(r_k)}(t) \le 1, \forall k \in \mathbb{N}$. It follows from Lebesgue Dominated Convergence Theorem that

$$\bar{g}'(\tau) = \lim_{\delta_1\to 0} \frac{\int_t^{t+\delta_1} \bar{f}_i(\phi)d\phi}{\delta_1} \tag{6.106}$$

Since $L(\bar{Q}(t))$ and $\bar{q}_i(t)$ are continuous functions, and $L(\bar{Q}(t)) > 0$ in the interval $t \in [\tau - \delta, \tau + \delta]$, it follows $\bar{f}_i(t)$ is continuous in the interval $t \in [\tau - \delta, \tau + \delta]$. Hence, from Fundamental Theorem of Calculus

$$\lim_{\delta_1\to 0} \frac{\int_\tau^{\tau+\delta_1} \bar{f}_i(\phi)d\phi}{\delta_1} = \bar{f}_i(\tau) \tag{6.107}$$

$$\implies \bar{g}_i'(\tau) = \bar{f}_i(\tau) \tag{6.108}$$

□

Proof of Lemma 6.7.2. Since $|\bar{U}(0)| \leq |\bar{X}(0)| = 1$, it is clear that $t \geq \bar{u}_i(0)$ for each $t \geq 1$. We will now show $\bar{g}_i(t) \geq \bar{v}_i(0)$ for $t \geq \tau^*$ for each $i = 1, \ldots, N$.

Note that $\bar{g}_j(t)$ is a non-decreasing function, since the residual service process $g_i(t)$ is a non-decreasing function. Hence, if $\bar{g}_i(t') \geq \bar{v}_i(0)$ for some t', it must be true that $\bar{g}_i(t) \geq \bar{v}_i(0)$, for any $t \geq t'$. Now, suppose $\bar{g}_j(t) < \bar{v}_j(0)$ for some $j \in \{1, \ldots N\}$ at time $t = \tau_0 > 1$. It follows that $\bar{g}_j(t)\bar{v}_j(0), \forall t \leq \tau_0$. In the following, we will show that any such τ_0 must be less than a fixed value τ^*. It immediately follows that $\bar{g}_j(t) \geq \bar{v}_j(0)$, for each $t > \tau^*$.

Since, $\bar{g}_j(t) < \bar{v}_j(0), \forall t \in [1, \tau_0]$, it follows from Lemma 6.6.3 that

$$\bar{q}_j(t) = \bar{q}_j(0) + \bar{\xi}_j t - \bar{\xi}_j \bar{u}_j(0) > 0, \forall\, t \in [1, \tau_0] \tag{6.109}$$

Since $L(\bar{Q}(t)) \geq \bar{q}_j(t)/\bar{\eta} > 0, \forall\, t \in (1, \tau_0]$, it follows from Lemma 6.7.1 that

$$\bar{g}_j'(t) = (\bar{q}_j(t)/\bar{\eta}_j)/L(\bar{Q}(t)),\ \forall\, t \in (1, \tau_0] \tag{6.110}$$

Note that $\bar{q}_i(t) \leq \bar{q}_i(0) + \bar{\xi}_i t$ for each $i, \forall t \geq 0$. Also note that $u_i(0) \leq 1$, $q_i(0) \leq 1$ for each i. It follows that for $t \in (1, \tau_0]$,

$$(\bar{q}_j(t)/\bar{\eta}_j)/L(\bar{Q}(t) \geq \frac{\bar{q}_j(0)/\bar{\eta}_j + \rho_j t - \rho_j \bar{u}_j(0)}{\max_{\boldsymbol{e} \in \mathcal{E}} \sum_{i=1}^N e_i(\bar{q}_i(0)/\bar{\eta}_i + \rho_i t)} \tag{6.111}$$

$$\geq \frac{\rho_j t - \rho_j}{\max_{e \in \mathcal{E}} \sum_{i=1}^N e_i/\bar{\eta}_i + e_i \rho_i t} \tag{6.112}$$

$$\geq \frac{\rho_j t - \rho_j}{\sum_{i=1}^N \max_{e \in \mathcal{E}} e_i/\bar{\eta}_i + \max_{e \in \mathcal{E}} e_i \rho_i t} \tag{6.113}$$

Hence, it follows that for each $t \in (1, \tau_0]$

$$\bar{g}_j'(t) \geq \frac{t-1}{c_1 + c_2 t} \tag{6.114}$$

$$= \frac{1}{c_2}\left(\frac{c_2 t}{c_1 + c_2 t}\right) - \frac{1}{c_1 + c_2 t} \tag{6.115}$$

$$= \frac{1}{c_2}\left(1 - \frac{c_1}{c_1 + c_2 t}\right) - \frac{1}{c_1 + c_2 t} \tag{6.116}$$

$$= \frac{1}{c_2} - \frac{1 + c_1/c_2}{c_1 + c_2 t} \tag{6.117}$$

where $c_1 := \frac{1}{\rho_j} \sum_{i=1}^N \max_{e \in \mathcal{E}} e_i/\bar{\eta}_i$ and $c_2 := \frac{1}{\rho_j} \sum_{i=1}^N \max_{e \in \mathcal{E}} e_i \rho_i$.

Hence, we have

$$g_j(\tau_0) - g_j(1) \geq \frac{\tau_0 - 1}{c_2} - (1 + c_1/c_2)\frac{\log(c_1 + c_2 \tau_0)}{c_2} + (1 + c_1/c_2)\frac{\log(c_1 + c_2)}{c_2} \tag{6.118}$$

$$g_j(\tau_0) \geq \frac{\tau_0 - 1}{c_2} - (1 + c_1/c_2)\frac{\log(c_1 + c_2 \tau_0)}{c_2} + (1 + c_1/c_2)\frac{\log(c_1 + c_2)}{c_2} \tag{6.119}$$

The function on the right side in the preceding inequality goes to ∞ as τ_0 goes to ∞. Hence, $\bar{g}_j(t) \geq \bar{v}_j(0)$ for some τ^*. Since, c_1 and c_2 only depend on $\{\bar{\eta}_i, \bar{\xi}_i\}_{i=1}^N$, a fixed τ^* can also be found, depending only on these values. □

Proof of Lemma 6.7.3. It follows from Lemma 6.6.3 that for all $t \geq \tau^*$

$$\bar{q}_i(t) = \bar{q}_i(0) + \bar{\xi}_i t - \bar{\xi}_i \bar{u}_i(0) - \bar{\eta}_i \bar{g}_i(t) + \bar{\eta}_i \bar{v}_i(0) \tag{6.120}$$

Now since $L(\bar{Q}(\tau)) > 0$, we have $\bar{g}_i'(\tau) = (\bar{q}_i(\tau)/\bar{\eta}_i)/L(\bar{Q}(\tau))$ from Lemma 6.7.1. It follows that for each i,

$$\bar{q}_i'(\tau) = \bar{\xi}_i - \bar{\eta}_i \bar{g}'(\tau) \tag{6.121}$$

$$= \bar{\xi}_i - \bar{q}_i(\tau)/L(\bar{Q}(\tau)) \tag{6.122}$$

Hence from (6.25), we have

$$\frac{dL_{\boldsymbol{e}}(\bar{Q}(t))}{dt}|_{t=\tau} = \sum_{i=1}^{N} e_i \bar{\xi}_i/\bar{\eta}_i - \sum_{i=1}^{N} (\bar{q}_i(\tau)/\bar{\eta}_i)/L(\bar{Q}(\tau)) \tag{6.123}$$

$$= \sum_{i=1}^{N} e_i \rho_i - L_{\boldsymbol{e}}(\bar{Q}(\tau))/L(\bar{Q}(\tau)) \tag{6.124}$$

□

Proof of Proposition 2. Let $\{x_r\}_{r\in\mathbb{N}} \subset \mathcal{X}$ be any sequence of initial states with $|x_r| \to \infty$, where $|.|$ is the L1 norm. Consider the sequence of scaled processes $\{X^{(r)}(t)\}_{r\in\mathbb{N}}$, with the rth process starting from the initial state $x_r/|x_r|$. It follows that $|X^{(r)}(0)| = 1$ for each r.

It follows from Theorem 6.7.5 that there exists a sub-sequence $\{r_k\}_{k\in\mathbb{N}} \subseteq \{r\}_{r\in\mathbb{N}}$ such that $Q^{(r_k)}(t) \to \bar{Q}(t)$ a.s. u.o.c. It follows from Lemma 6.6.2, that $u_i^{(r_k)}(t) \to \bar{u}_i(t)$ and $v_i^{(r_k)}(t) \to \bar{v}_i(t)$ u.o.c a.s, for each $i = 1, \ldots, N$.

From Theorem 6.7.5, $\exists T > 0$ such that $\bar{Q}(t) = \mathbf{0}, \forall t \geq T$. Define $t_0 = \max 1, T$. Hence, we have,

$$\lim_{k\to\infty} |Q^{(r_k)}(t_0)| = \lim_{k\to\infty} \sum_{i=1}^{N} q_i^{(r_k)}(t_0) = 0 \text{ a.s.} \tag{6.125}$$

$$\lim_{k\to\infty} |Q^{x_{r_k}}(|x_{r_k}|t_0)|/|x_{r_k}| = 0 \text{ a.s.} \tag{6.126}$$

Consider an arbitrary queue $i \in \{1, \ldots, N\}$. Let $m_i(t)$ denote renewal counting process corresponding to the arrival process at queue i.

$$m_i(t) := \max\{n : \sum_{j=1}^{n} \xi_i(j) \leq t\} \tag{6.127}$$

It follows from (6.32) that

$$a_i(t) = \begin{cases} 0 & \text{if } t \le u_i(0) \\ 1 + m_i(t - u_i(0)) & \text{if } t > u_i(0) \end{cases} \tag{6.128}$$

Hence, $a_i(t) \le 1 + m_i(t - u_i(0))^+$. Since $t \le (t - u_i(0))^+, \forall t \ge 0$, it follows $a_i(t) \le 1 + m_i(t), \forall t \ge 0$. Hence for each $i = 1, \ldots, N$, $a^{(r_k)(t)} \le (1 + m_i(|x_{r_k}|t))/|x_{r_k}|$. Now since $q_i^{(r_k)}(t) \le q_i^{(r_k)}(0) + a_i^{(r_k)}(t)$, we have

$$q_i^{(r_k)}(t) \le q_i^{(r_k)}(0) + (1 + m_i(|x_{r_k}|t)/|x_{r_k}| \tag{6.129}$$

Since (6.129) holds for each $i = 1, \ldots, N$, we have

$$\sum_{i=1}^{N} q_i^{(r_k)}(t_0) \le \sum_{i=1}^{N} q_i^{(r_k)}(0) + \sum_{i=1}^{N}(1 + m_i(|x_{r_k}|t_0))/|x_{r_k}| \tag{6.130}$$

$$|Q^{(r_k)}(t_0)| \le |Q^{(r_k)}(0)| + \sum_{i=1}^{N}(1 + m_i(|x_{r_k}|t_0))/|x_{r_k}| \tag{6.131}$$

$$|Q^{x_{r_k}}(t_0)|/|x_{r_k}| \le 1 + \sum_{i=1}^{N}(1 + m_i(|x_{r_k}|t_0))/|x_{r_k}| \tag{6.132}$$

$$|Q^{x_{r_k}}(|x_{r_k}|t_0)|^2/(|x_{r_k}|t_0)^2 \le \left(1/t_0 + \sum_{i=1}^{N}(1 + m_i(|x_{r_k}|t_0))/(|x_{r_k}|t_0)\right)^2 \tag{6.133}$$

Since $t_0 \ge 1$, from Lemma 6.5.5, the sequence of rvs

$$\left\{\left(1/t_0 + \sum_{i=1}^{N}(1 + m_i(|x_{r_k}|t_0))/(|x_{r_k}|t_0)\right)^2\right\}_{k\in\mathbb{N}}$$

is uniformly integrable. Hence, from (6.133), $\{|Q^{x_{r_k}}(|x_{r_k}|t_0)|^2/(|x_{r_k}|t_0)^2\}_{k\in\mathbb{N}}$ is uniformly integrable. Now, it follows from (6.126) that

$$\lim_{k\to\infty} \frac{E[|Q^{x_{r_k}}(|x_{r_k}|t_0)|^2}{(|x_{r_k}|)^2} = 0 \tag{6.134}$$

It remains to show that

$$\lim_{k\to\infty} \frac{E[|U^{x_{r_k}}(|x_{r_k}|t_0)|^2}{(|x_{r_k}|)^2} = 0 \tag{6.135}$$

$$\lim_{k\to\infty} \frac{E[|V^{x_{r_k}}(|x_{r_k}|t_0)|^2}{(|x_{r_k}|)^2} = 0 \tag{6.136}$$

Since $t_0 \geq 1$ and $|x_{r_k}| = \sum_{i=1}^{N}(q_i(0) + u_i(0) + v_i(0))$, we have $t_0|x_{r_k}| \geq u_i(0)$. Hence, the residual arrival time at $|x_{r_k}|t_0$ satisfies $u_i(|x_{r_k}|t_0) \leq \xi_i(a_i(|x_{r_k}|t_0))$ for each $i = 1, \ldots, N$. Hence, for each $i = 1, \ldots, N$,

$$\frac{\left(u_i(|x_{r_k}|t_0)\right)^2}{|x_{r_k}|^2} \leq \frac{\left(\xi_i(a_i(|x_{r_k}|t_0))\right)^2}{|x|^2} \tag{6.137}$$

$$\leq \frac{1}{|x_{r_k}|^2} \sum_{j=1}^{a_i(|x_{r_k}|t_0)} (\xi_i(j))^2 \tag{6.138}$$

$$\leq \frac{1}{|x_{r_k}|^2} \sum_{j=1}^{m_i(|x_{r_k}|t_0)+1} (\xi_i(j))^2 \tag{6.139}$$

since $a_i(t) \leq 1 + m_i(t)$.

By Wald's identity,

$$\frac{1}{|x_{r_k}|^2} E\left[\sum_{j=1}^{m_i(|x_{r_k}|t_0)+2} (\xi_i(j))^2\right] = \frac{1}{|x_{r_k}|^2} E[m_i(|x_{r_k}|t_0) + 1]E[(\xi_i(1))^2] \tag{6.140}$$

Since $E[(\xi_i(1))^2] < \infty$ (from Assumption 8) and since $E[m_i(|x_{r_k}|t_0) + 1]/|x_{r_k}t_0| \to \bar{\xi}$ a.s. as $k \to \infty$ (from Elementary Renewal Theorem , i.e., Theorem 6.5.4), we have that the preceding equation converges to 0 as $k \to \infty$. Hence, from (6.139) and (6.140),

$$\lim_{k\to\infty} \frac{E[\left(u_i(|x_{r_k}|t_0)\right)^2]}{|x_{r_k}|^2} = 0 \tag{6.141}$$

for each $i = 1, \ldots, N$. Hence, $\lim_{k\to\infty} E[|U(|x_{r_k}|t_0)|^2]/|x_{r_k}|^2 = 0$.

Now consider $v_i(|x_{r_k}|t_0)$ for an arbitrary i. Note that $v_i(|x_{r_k}|t_0) \leq v_i(0)$, if $g_i(|x_{r_k}|t_0) < v_i(0)$. Otherwise, if $g_i(|x_{r_k}|t_0) \geq v_i(0)$, the residual service requirement satisfies $v_i(|x_{r_k}|t_0) \leq \eta_i(d_i(g_i(|x_{r_k}|t_0)))$. Hence, (proceeding similarly as with $u_i(|x_{r_k}|t_0)$ before) we have

$$\frac{\left(v_i(|x_{r_k}|t_0)\right)^2}{|x_{r_k}|^2} \leq \frac{\left(\eta_i(d_i(g_i(|x_{r_k}|t_0)))\right)^2}{|x|^2} \tag{6.142}$$

$$\leq \frac{\left(\eta_i(d_i(|x_{r_k}|t_0))\right)^2}{|x|^2} \tag{6.143}$$

(6.143) follows since $g_i(|x_{r_k}|t_0) \leq |x_{r_k}|t_0$, (because $g_i(t) = \int_0^t f_i(s)ds$, where $f_i(s) \leq 1, \forall s$). By repeating the earlier arguments, it follows that

$$\lim_{k\to\infty} \frac{E[\left(v_i(|x_{r_k}|t_0)\right)^2]}{|x_{r_k}|^2} = 0 \tag{6.144}$$

for each $i = 1, \ldots, N$. Hence, $\lim_{k\to\infty} E[|V^{x_{r_k}}(|x_{r_k}|t_0)|^2]/|x_{r_k}|^2 = 0$.

So far, we have shown that for any sequence $\{x_r\}_{r\in\mathbb{N}} \subset \mathcal{X}$ of initial states such that $|x_r| \to \infty$, there exists a subsequence $\{x_{r_k}\}_{k\in\mathbb{N}}$ and such that

$$\lim_{k\to\infty} \frac{E[|X^{x_{r_k}}(|x_{r_k}|t_0)|^2]}{|x_{r_k}|^2} = 0 \tag{6.145}$$

Since the choice of initial sequence is arbitrary, and since t_0 does not depend on the subsequence, we obtain the result. □

Proof of Proposition 3. It follows from Lemma 2 that there exists a compact set of the form $C := \{x \in \mathcal{X} : |x| \leq L\}$, (where $|.|$ is the L1 norm), such that for each $x \in C^c := \mathcal{X} - C$

$$E[|X^x(t_0|x|)|^2] \leq \frac{1}{2}|x|^2. \tag{6.146}$$

Let $P^t(x, A) = P(X^x(t) \in A)$ be the transition probability of X for $A \in \mathcal{B}_\mathcal{X}$, where $\mathcal{B}_\mathcal{X}$ is the Borel σ-field of $\mathcal{X}$. By letting $t(x) := t_0 \max\{L, |x|\}$ for each $x \in \mathcal{X}$, the preceding inequality can be written as

$$\int P^{t(x)}(x, dy)|y|^2 \leq \frac{1}{2}|x|^2 + b\mathbb{I}_C(x), \forall x \in \mathcal{X} \tag{6.147}$$

where $\mathbb{I}_C(.)$ is the indicator function, and b is a finite constant.

Define the sequence of stopping times $\sigma_0 = 0$, $\sigma_1 = t(x)$, and $\sigma_{k+1} = \sigma_k + \theta_{\sigma_k}\sigma_1$, $k \geq 1$, where θ is the shift operator on the sample space. The stochastic process $\hat{X}_k := \{X(\sigma_k)\}_{k\geq 0}$ is a Markov chain with transition kernel

$$\hat{P}(x, A) := P(X^x(t) \in A), \quad x \in X, A \in \mathcal{B}_\mathcal{X}, \tag{6.148}$$

and the bound (6.147) may be expressed

$$\int_\mathcal{X} \hat{P}(x, dy)U_2(y) \leq U_2(x) - \frac{1}{2}|x|^2 + b\mathbb{I}_C(x) \tag{6.149}$$

with $U_2(x) = |x|^2$. From the Comparison Theorem (Theorem 14.2.2 from [78]), we then have

$$E[\sum_{k=0}^{k_*-1} |X^x(\sigma_k)|^2] = E[\sum_{k=0}^{k_*-1} |\hat{X}_k^x|^2] \leq 2(|x|^2 + b\mathbb{I}_C(x)), \ \forall x \in \mathcal{X}, \tag{6.150}$$

where $k_* := \min\{k \geq 1 : \hat{X}_k \in C\}$.

To prove the proposition, we first show that for some constant c_0,

$$E\left[\int_{\sigma_k}^{\sigma_{k+1}} (1 + |X^x(t)|^2)dt|\mathcal{F}_{\sigma_k}\right] \leq c_0(|X^x(\sigma_k)|^{p+1} + 1), \ k \geq 0, x \in \mathcal{X}, \tag{6.151}$$

which by the strong Markov property amounts to

$$E[\int_0^{\sigma_1}(1+|X^x(t)|^p)dt] \leq c_0(|x|^{p+1}+1), x \in \mathcal{X} \tag{6.152}$$

Because $|X^x(t)| = |Q^x(t)| + |U^x(t)| + |V^x(t)|$. Let us first consider

$$E[\int_0^{\sigma_1}(u_i^x(s))^p ds] \tag{6.153}$$

Note that $u_i^x(t) \leq u_i(0)$ for $t < u_i(0)$ and $u_i^x(t) \leq \xi_i(a_i(t))$ for $t \geq u_i(0)$. Hence, we have

$$(u_i^x(s))^2 \leq (u_i(0))^2 + \sum_{j=1}^{a_i(s)+1}(\xi_i(j))^2 \tag{6.154}$$

$$\leq |x|^2 + \sum_{j=1}^{a_i(s)+1}(\xi_i(j))^2 \tag{6.155}$$

By Wald's identity and part (a) of Lemma 6.5.5,

$$E[(u_i^x(s))^2] \leq |x|^2 + E[\sum_{j=1}^{a_i^x(s)+1}(\xi_i(j))^2] \tag{6.156}$$

$$= |x|^2 + E[a_i^x(s)+1]E[(\xi_i(1))^2] \leq |x|^2 + c_1(s+1)(u_i^x(s))^2 \tag{6.157}$$

Thus,

$$E[\int_0^{\sigma_1}(u_i^x(s))^2 ds] \leq |x|^2\sigma_1 + c_1(\sigma_1 + (\sigma_1)^2/2)E[(\xi_i(1))^2] \tag{6.158}$$

$$\leq c_2(|x|^2+1) \tag{6.159}$$

Similarly, consider $E[\int_0^{\sigma_1}(v_i^x(s))^p ds]$. $v_i^x(t) \leq v_i(0)$ for $g_i(t) < v_i(0)$ and $v_i^x(t) \leq \eta_i(d_i^x(g_i(t)))$ for $g_i(t) \geq v_i(0)$. Hence, it follows that

$$(v_i^x(s))^2 \leq (v_i(0))^2 + \sum_{j=1}^{d_i^x(g_i(s))+1}(\eta_i(j))^2 \tag{6.160}$$

$$\leq |x|^2 + \sum_{j=1}^{d_i^x(s)+1}(\eta_i(j))^2 \tag{6.161}$$

since $g_i(s) \leq s$. Therefore, proceeding similarly, we have

$$E[\int_0^{\sigma_1}(v_i^x(s))^2 ds] \leq c_3(|x|^2+1) \tag{6.162}$$

So, it remains to bound the integral of $|q_i^x(t)|^2$. Note that $q_i^x(t) \leq q_i^x(0) + a_i^x(t)$. By part (a) of Lemma 6.5.5, there exists a constant c_4 such that

$$E[(a_i^x(t))^2] \leq c_4(t^2+1), t \geq 0 \tag{6.163}$$

and hence, for constants $c_5, c_6 < \infty$

$$E[\int_0^{\sigma_1} (q_i^x(t))^2] \leq c_5\sigma_1(|Q(0)|^p + (\sigma_1)^p) \tag{6.164}$$

$$\leq c_6(|x|^2+1), x \in \mathcal{X} \tag{6.165}$$

This together with (6.162) and (6.157), shows that (6.152) holds.

Substituting the equivalent bound (6.151) into (6.150), we have for some $c_7 < \infty$,

$$E\left[\sum_{k=0}^{\infty} E\left[\int_{\sigma_k}^{\sigma_{k+1}} (1+|X^x(t)|)dt|\mathcal{F}_{\sigma_k}\right]\mathbb{I}\{k < k^*\}\right] \leq c_7(|x|^2+1) \tag{6.166}$$

By Fubini's theorem and the smoothing property of the conditional expectation, LHS is precisely $E[\int_0^{\sigma_{k_*}}(1+|X(t)|)dt]$. Since $\sigma_{k_*} \geq \tau_C(t_0L)$, this establishes the proposition. □

Chapter 7

Conclusions and Future Work

Resource allocation problems in a general wireless network are well known for being NP-hard. Much of the complexity comes from the combinatorial explosion in the number of possible schedules with increase in the size of network. The main theme of the thesis is that additional structure can lead to tractable solutions to an otherwise hard resource allocation problems. We have considered the minimum resource clearing problem (which is NP-hard in general) as the key optimization problem in the thesis. We have presented several networks where the underlying structure led to efficient solutions to the optimization problem.

In the thesis, hierarchy is the key network structure which was exploited, by considering tree type graphs. We have presented resource allocation algorithms for various wireless networks, such as HetNets and mmWave IAB networks. The presented algorithms are distributed in nature and are of low computational complexity. There are potentially other networks and resource allocation problems, where a similar approach can be applied. Hence, for future work, one can investigate other networks and resource allocation problems, where the underlying structure yields efficient solutions. Alternatively, design of future network topologies can consider imposing structure as part of their design. As evident from the thesis, such an approach can lead to efficient and optimal multiple access protocols.

One can also investigate other types of structure, (e.g., ring graph), and investigate how that affects the hardness of the minimum resource clearing problem. Alternatively, one can consider other resource allocation problems (e.g., maximum rate problem), in the context of the networks in the thesis.

In the thesis, we have presented each chapter as a self contained piece of research, apart from the

noticeable theme of minimum clearing time problem. However, several chapters in the thesis are more closely associated than it may seem. In the following, we discuss these associations and future work which is specific to the chapter.

7.0.1 Chapter 2

In Chapter 2, we have derived novel structural results which enable an efficient solution to the minimum clearing time optimization in three tier HetNets. We have also demonstrated the wide applicability of the three tier framework, by considering several future hierarchical networks. For future work, the natural direction is to generalize the framework for $K(\geq 4)$ tiers. In Chapter 4, we generalize the resource partitioning framework for K tiers. However, the joint optimization of user association and resource partitioning is still an open problem for K tiers. As part of 5G, wireless communication using smaller cells, aerial platforms and satellites is being considered. Hence, future wireless architectures are predicted to be multi tiered. The optimal multi-tier user association is a key challenge in the area. Alternatively for future work in three tiers, co-tier interference mitigation as done in Chapter 4 can also be integrated into the three tier joint optimization.

7.0.2 Chapter 3

Chapter 3 considers the IAB tree network topology for mmWave multi-hop networks. We investigate optimal scheduling for the IAB network and derive the stability region, under a dynamic scenario with stochastic packet arrivals, and time-varying link rates. We investigate a class of local scheduling policies which only require local information for making scheduling decisions. We show that the stability region for the local class is same as that of global policies, provided the links are unvarying. We propose a local max-weight based policy which is optimal among the local class. Using numerical simulation, we show that the performance of the proposed local policy is comparable to the back-pressure policy under the considered IAB scenario.

An important problem for future work is to find a distributed implementation for an optimal global policy (such as the back-pressure policy). This would involve investigating the maximum weight optimization problem on graph G (in Chapter 3), subject to the half-duplex and RF chains constraints.

The simulation results indicate that for the mmWave links (assuming there is no blocking), there is not much variance in the rates from slow-scale fading. Hence for future work, one can consider a

deterministic problem (with fixed link rates and mean arrival rates) for the IAB topology. Assuming there is only one RF chain at each gNB, this formulation would be equivalent to the minimum clearing time problem for Topology 2 in Chapter 5. Hence, the main challenge is to extend the algorithm in Chapter 5 to include multiple RF chains.

7.0.3 Chapter 4

In Chapter 4, we introduce the K tier HetNet model, which generalizes the key ideas behind the three tier framework of Chapter 2. There are two main differences between the two frameworks. 1) Chapter 2 considers the joint optimization of user-association and resource partitioning, whereas in Chapter 4, we consider that user association is given. 2) Chapter 4 consider co-tier interference mitigation which was not considered in Chapter 2.

We introduce a novel graphical model to model the interference in the K-tier HetNet. We use this model to develop a recursive formulation for minimum resource clearing optimization. The recursive formulation only requires local information, i.e, the information relations at the tier level. We present a forward-backward scheme to solve the global minimum resource clearing optimization, using the local recursive optimizations.

For future work, the K tier framework can be generalized by jointly considering user-association as part of the optimization. The K tier framework is theoretical in nature. Hence, based on the application, some of the assumptions made here may not apply. Hence, modifications of the recursive graph structure based on individual applications can be considered for future work. A similar exercise was done for three tier framework in section 2.5 on applications.

7.0.4 Chapter 5

Chapter 5 provides an iterative local update rule to find the solution of the minimum clearing time problem. We have shown that the scheme always converges, and converges to optimal solution in the topologies presented in Chapter 5. The formulation here considers deterministic loads $\tau(u_i)$ on each user u_i. For the considered topologies, in the long-term, each user u_i gets a fraction $\tau(u_i)/\tau_C^{\max}$ of time allocated. We also note that $\tau_C^{\max}$ is the optimal value of minimum time clearing LP in Chapter 5.

For a moment, replace $\tau(u_i)$ by $q_i/\bar{\eta}_i$, and $\tau_C^{\max}$ by $L(\bar{Q})$. We get the expression for $f_i(Q)$ from (6.23) in Chapter 6. In Chapter 6, q_i is the queue length at queue i, and $L(Q)$ is the optimal value of

minimum clearing LP formulated using queue lengths. The implication is the following.

Consider a queueing analogue of the network in Chapter 5, where each user/node u_i maintains a queue q_i which has exogenous flow arrivals. The inter-arrival period and service requirements obey the same assumptions as in Chapter 6. For this queueing system, each user u_i can apply the algorithm in Chapter 5, with $\tau(u_i) = q_i/\bar{\eta}_i$. Assuming that the Chapter 5's algorithm runs on a much faster time scale compared to flow arrivals, the time fraction achieved by u_i equals $[f_i(Q)]_i$ from (6.23) in Chapter 6. This means that applying the algorithm of Chapter 5 with $\tau(u_i) = q_i/\bar{\eta}_i$ stabilizes this queueing network (for all the arrival rates within the stability region).

7.0.5 Chapter 6

The work in Chapter 6 is a very general framework which can applied in many queueing networks. The framework provides a flow control policy which is based on minimum resource clearing LP formulation. Broadly speaking, it provides a queueing analogue of the minimum clearing LP. The immediate consequence is the following. If an efficient scheme to solve minimum clearing LP exists, then the same solver can be used to implement a flow control policy on the queueing analog of the network. The introduction of Chapter 6 starts with such an application to the k tier HetNet (of Chapter 4). However, it can also be applied to the networks in Chapters 2 and 5, which also consider minimum clearing formulations.

References

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *29th IEEE Conference on Decision and Control*, 1990, pp. 2130–2132 vol.4.

[2] E. Arikan, "Some complexity results about packet radio networks (Corresp.)," *IEEE Transactions on Information Theory*, vol. 30, no. 4, pp. 681–685, 1984.

[3] S.-P. Yeh, S. Talwar, G. Wu, N. Himayat, and K. Johnsson, "Capacity and coverage enhancement in heterogeneous networks," *IEEE Wireless Communications*, vol. 18, no. 3, pp. 32–38, 2011.

[4] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2706–2716, 2013.

[5] D. Fooladivanda and C. Rosenberg, "Joint resource allocation and user association for heterogeneous wireless cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 1, pp. 248–257, 2012.

[6] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRAN); Overall description; Stage 2," 3rd Generation Partnership Project (3GPP), Tech. Rep., September 2013.

[7] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5G be?" *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[8] 3GPP, "Study on Integrated Access and Backhaul," 3rd Generation Partnership Project (3GPP), TR 38.874 (Rel 16), 2019.

[9] M. Polese, M. Giordani, T. Zugno, A. Roy, S. Goyal, D. Castor, and M. Zorzi, "Integrated Access and Backhaul in 5G mmWave Networks: Potentials and Challenges," *arXiv preprint arXiv:1906.01099v1*, 06 2019.

[10] L. Jiang and J. Walrand, "A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 960–972, 2010.

[11] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: Queue-Length Based CSMA/CA Algorithms for Achieving Maximum Throughput and Low Delay in Wireless Networks," in *2010 Proceedings IEEE INFOCOM*, 2010, pp. 1–5.

[12] G. Sharma, C. Joo, N. B. Shroff, and R. R. Mazumdar, "Joint Congestion Control and Distributed Scheduling for Throughput Guarantees in Wireless Networks," *ACM Trans. Model. Comput. Simul.*, vol. 21, no. 1, Dec. 2010. [Online]. Available: https://doi.org/10.1145/1870085.1870090

[13] A. Gupta, X. Lin, and R. Srikant, "Low-Complexity Distributed Scheduling Algorithms for Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, pp. 1846–1859, 2009.

[14] C. Joo, X. Lin, J. Ryu, and N. B. Shroff, "Distributed Greedy Approximation to Maximum Weighted Independent Set for Scheduling With Fading Channels," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1476–1488, 2016.

[15] L. X. Bui, S. Sanghavi, and R. Srikant, "Distributed Link Scheduling With Constant Overhead," *IEEE/ACM Transactions on Networking*, vol. 17, no. 5, pp. 1467–1480, 2009.

[16] E. Modiano, D. Shah, and G. Zussman, "Maximizing Throughput in Wireless Networks via Gossiping," in *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '06/Performance '06. New York, NY, USA: Association for Computing Machinery, 2006, pp. 27–38. [Online]. Available: https://doi.org/10.1145/1140277.1140283

[17] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource allocation and cross-layer control in wireless networks*. Now Publishers Inc, 2006.

[18] C. Joo, X. Lin, and N. B. Shroff, "Greedy maximal matching: Performance limits for arbitrary network graphs under the node-exclusive interference model," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2734–2744, 2009.

[19] A. Eryilmaz, A. Ozdaglar, and E. Modiano, "Polynomial complexity algorithms for full utilization of multi-hop wireless networks," in *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*. IEEE, 2007, pp. 499–507.

[20] C. Liu, P. Whiting, and S. V. Hanly, "Joint resource allocation and user association in downlink three-tier heterogeneous networks," in *2014 IEEE Global Communications Conference*. IEEE, 2014, pp. 4232–4238.

[21] J. G. Dai, "On positive Harris recurrence of multiclass queueing networks: A unified approach via fluid limit models," in *Annals of Applied Probability*, vol. 5, 1995, pp. 49–77.

[22] J. G. Dai and S. P. Meyn, "Stability and convergence of moments for multiclass queueing networks via fluid limit models," *IEEE Transactions on Automatic Control*, vol. 40, no. 11, pp. 1889–1904, 1995.

[23] A. Ghosh, T. A. Thomas, M. C. Cudak, R. Ratasuk, P. Moorut, F. W. Vook, T. S. Rappaport, G. R. MacCartney, S. Sun, and S. Nie, "Millimeter-wave enhanced local area systems: A high-data-rate approach for future wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1152–1163, 2014.

[24] S. Chandrasekharan, K. Gomez, A. Al-Hourani, S. Kandeepan, T. Rasheed, L. Goratti, L. Reynaud, D. Grace, I. Bucaille, T. Wirth *et al.*, "Designing and implementing future aerial communication networks," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 26–34, 2016.

[25] M. M. Azari, F. Rosas, A. Chiumento, and S. Pollin, "Coexistence of terrestrial and aerial users in cellular networks," in *2017 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2017, pp. 1–6.

[26] A. Mohammed, A. Mehmood, F.-N. Pavlidou, and M. Mohorcic, "The role of high-altitude platforms (HAPs) in the global wireless connectivity," *Proceedings of the IEEE*, vol. 99, no. 11, pp. 1939–1953, 2011.

[27] S. Borst, S. Hanly, and P. Whiting, "Throughput Utility Optimization in HetNets," in *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*, June 2013, pp. 1–5.

[28] C. Liu, M. Li, S. V. Hanly, and P. Whiting, "Joint downlink user association and interference management in two-tier HetNets with dynamic resource partitioning," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1365–1378, 2016.

[29] X. Ge, X. Li, H. Jin, and J. Cheng, "Joint User Association and User Scheduling for Load Balancing in Heterogeneous Networks," *IEEE Transactions on Wireless Communications*, vol. 17, pp. 3211–3225, 02 2018.

[30] Q. Kuang, W. Utschick, and A. Dotzler, "Optimal Joint User Association and Multi-Pattern Resource Allocation in Heterogeneous Networks," *IEEE Transactions on Signal Processing*, vol. 64, pp. 3388–3401, 06 2016.

[31] W. C. Cheung, T. Q. Quek, and M. Kountouris, "Throughput optimization, spectrum allocation, and access control in two-tier femtocell networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 3, pp. 561–574, 2012.

[32] S. Singh and J. G. Andrews, "Joint resource partitioning and offloading in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 2, pp. 888–901, 2013.

[33] Y. Lin, W. Bao, W. Yu, and B. Liang, "Optimizing User Association and Spectrum Allocation in HetNets: A Utility Perspective," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 6, pp. 1025–1039, June 2015.

[34] J. Ghimire and C. Rosenberg, "Resource allocation, transmission coordination and user association in heterogeneous networks: A flow-based unified approach," *IEEE Transactions on Wireless Communications*, vol. 12, no. 3, pp. 1340–1351, 2013.

[35] N. Sapountzis, T. Spyropoulos, N. Nikaein, and U. Salim, "Joint Optimization of User Association and Dynamic TDD for Ultra-Dense Networks," in *IEEE INFOCOM 2018*, April 2018, pp. 2681–2689.

[36] G. Arvanitakis, T. Spyropoulos, and F. Kaltenberger, "An Analytical Model for Flow-Level

Performance in Heterogeneous Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 17, pp. 1–1, 12 2017.

[37] S. V. Hanly, C. Liu, and P. Whiting, "Capacity and stable scheduling in heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 6, pp. 1266–1279, 2015.

[38] S. Borst, H. Bakker, M. Gruber, S. Klein, and P. Whiting, "Flow-level capacity and performance in hetnets," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. IEEE, 2015, pp. 1–6.

[39] H. Kim, G. De Veciana, X. Yang, and M. Venkatachalam, "Distributed α-optimal user association and cell load balancing in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 177–190, 2011.

[40] F. Wang, W. Chen, H. Tang, and Q. Wu, "Joint Optimization of User Association, Subchannel Allocation, and Power Allocation in Multi-Cell Multi-Association OFDMA Heterogeneous Networks," *IEEE Transactions on Communications*, vol. 65, pp. 1–1, 03 2017.

[41] L. P. Qian, Y. J. A. Zhang, Y. Wu, and J. Chen, "Joint Base Station Association and Power Control via Benders' Decomposition," *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1651–1665, April 2013.

[42] Y. Chen, J. Li, W. Chen, Z. Lin, and B. Vucetic, "Joint User Association and Resource Allocation in the Downlink of Heterogeneous Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 7, pp. 5701–5706, July 2016.

[43] R. Sun, M. Hong, and Z.-Q. Luo, "Joint Downlink Base Station Association and Power Control for Max-Min Fairness: Computation and Complexity," *Selected Areas in Communications, IEEE Journal on*, vol. 33, 07 2014.

[44] S. Borst, S. Hanly, and P. Whiting, "Optimal resource allocation in HetNets," in *2013 IEEE International Conference on Communications (ICC)*, June 2013, pp. 5437–5441.

[45] R. Ford, F. Gómez-Cuba, M. Mezzavilla, and S. Rangan, "Dynamic time-domain duplexing for

self-backhauled millimeter wave cellular networks," in *2015 IEEE International Conference on Communication Workshop (ICCW)*. IEEE, 2015, pp. 13–18.

[46] D. Yuan, H.-Y. Lin, J. Widmer, and M. Hollick, "Optimal joint routing and scheduling in millimeter-wave cellular networks," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1205–1213.

[47] M. Eslami Rasekh, D. Guo, and U. Madhow, "Joint Routing and Resource Allocation for Millimeter Wave Picocellular Backhaul," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 783–794, Feb 2020.

[48] Y. Li, J. Luo, R. Stirling-Gallacher, and G. Caire, "Integrated Access and Backhaul Optimization for Millimeter Wave Heterogeneous Networks," *arXiv preprint arXiv:1901.04959v1*, 01 2019.

[49] Y. Niu, "Exploiting Multi-Hop Relaying to Overcome Blockage in Directional mmWave Small Cells," *Journal of Communications and Networks*, 04 2015.

[50] B. Sahoo, C.-H. Yao, and H.-y. Wei, "Millimeter-Wave Multi-Hop Wireless Backhauling for 5G Cellular Networks," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*. IEEE, 06 2017, pp. 1–5.

[51] Y. Yao, H. Tian, G. Nie, H. Wu, and J. Jin, "Multi-path Routing Based QoS-aware Fairness Backhaul-Access Scheduling in mmWave UDN," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 09 2018, pp. 1–7.

[52] J. García-Rois, F. Gómez-Cuba, M. R. Akdeniz, F. J. González-Castaño, J. C. Burguillo, S. Rangan, and B. Lorenzo, "On the analysis of scheduling in dynamic duplex multihop mmWave cellular systems," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6028–6042, 2015.

[53] F. Gomez-Cuba and M. Zorzi, "Optimal link scheduling in millimeter wave multi-hop networks with space division multiple access," in *2016 Information Theory and Applications Workshop (ITA)*. IEEE, 2016, pp. 1–9.

[54] J. Garcia-Rois, R. Banirazi, F. Gonzalez-Castano, B. Lorenzo, and J. Burguillo, "Delay-Aware Optimization Framework for Proportional Flow Delay Differentiation in Millimeter-Wave Backhaul Cellular Networks," *IEEE Transactions on Communications*, vol. PP, pp. 1–1, 01 2018.

[55] K. Vu, C.-F. Liu, M. Bennis, M. Debbah, and M. Latva-aho, "Path Selection and Rate Allocation in Self-Backhauled mmWave Networks," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 04 2018, pp. 1–6.

[56] M. Polese, M. Giordani, A. Roy, D. Castor, and M. Zorzi, "Distributed path selection strategies for integrated access and backhaul at mmWaves," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[57] Q. Hu and D. Blough, "Relay Selection and Scheduling for Millimeter Wave Backhaul in Urban Environments," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 10 2017, pp. 206–214.

[58] Y. Liu, Q. Hu, and D. M. Blough, "Joint Link-Level and Network-Level Reconfiguration for MmWave Backhaul Survivability in Urban Environments," in *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWIM '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 143–151. [Online]. Available: https://doi.org/10.1145/3345768.3355913

[59] Y.-H. Chiang and W. Liao, "Mw-HierBack: A Cost-Effective and Robust Millimeter Wave Hierarchical Backhaul Solution for HetNets," *IEEE Transactions on Mobile Computing*, vol. PP, pp. 1–1, 04 2017.

[60] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable Scheduling Policies for Fading Wireless Channels," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411–424, Apr. 2005. [Online]. Available: https://doi.org/10.1109/TNET.2004.842226

[61] M. J. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," Ph.D. dissertation, Massachusetts Institute of Technology, 2003.

[62] M. K. Samimi, G. R. MacCartney, S. Sun, and T. S. Rappaport, "28 GHz Millimeter-Wave Ultrawideband Small-Scale Fading Models in Wireless Channels," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, pp. 1–6.

[63] M. Gapeyenko, A. Samuylov, M. Gerasimenko, D. Moltchanov, S. Singh, M. R. Akdeniz, E. Aryafar, N. Himayat, S. Andreev, and Y. Koucheryavy, "On the Temporal Effects of Mobile Blockers in Urban Millimeter-Wave Cellular Scenarios," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 124–10 138, 2017.

[64] A. Anttonen, P. Ruuska, and M. Kiviranta, *3GPP nonterrestrial networks: A concise review and look ahead*, ser. VTT Research Report. Finland: VTT Technical Research Centre of Finland, 2019, no. VTT-R-00079-19, vTT-R-00079-19.

[65] G. Bianchi, F. Borgonovo, L. Fratta, L. Musumeci, and M. Zorzi, "C-PRMA: a centralized packet reservation multiple access for local wireless communications," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 2, pp. 422–436, 1997.

[66] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar, "Throughput and Fairness Guarantees Through Maximal Scheduling in Wireless Networks," in *IEEE transactions on Information Theory*, vol. 54, no. 2, 2008, pp. 572–594.

[67] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," in *IEEE/ACM transactions on networking*, vol. 17, no. 4, 2009.

[68] A. Dimakis and J. Walrand, "Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits," *Advances in Applied probability*, pp. 505–521, 2006.

[69] M. Leconte, J. Ni, and R. Srikant, "Improved bounds on throughput efficiency of greedy maximal scheduling in wireless networks," in *IEEE/ACM transactions on networking*, vol. 19, no. 3, 2011, pp. 709–720.

[70] S. B. Fred, T. Bonald, A. Proutiere, G. Régnié, and J. W. Roberts, "Statistical bandwidth sharing: a study of congestion at flow level," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 111–122, 2001.

[71] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, vol. 49, no. 3, pp. 237–252, 1998.

[72] T. Bonald and L. Massoulié, "Impact of fairness on Internet performance," in *Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2001, pp. 82–91.

[73] G. Sharma, C. Joo, N. B. Shroff, and R. R. Mazumdar, "Joint congestion control and distributed scheduling for throughput guarantees in wireless networks," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 21, no. 1, pp. 1–25, 2010.

[74] J. Perry, H. Balakrishnan, and D. Shah, "Flowtune: Flowlet control for datacenter networks," in *14th USENIX Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, 2017, pp. 421–435.

[75] M. H. Davis, "Piecewise-deterministic Markov processes: a general class of non-diffusion stochastic models," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 46, no. 3, pp. 353–376, 1984.

[76] K. Chung, M. Chung, and K. M. R. Collection, *A Course in Probability Theory*, ser. Probability and mathematical statistics. Academic Press, 1974. [Online]. Available: https://books.google.com.au/books?id=e2IPAQAAMAAJ

[77] A. Gut, *Stopped random walks limit theorems and applications*, 2nd ed., ser. Springer series in operations research and financial engineering. New York ;: Springer, 2008.

[78] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.