Distributed Control of Multihop Wireless Networks with Quality-of-Service

A Thesis

Submitted for the Degree of

Doctor of Philosophy

in the **Faculty of Engineering**

by

Ashok Krishnan K. S.

Electrical Communication Engineering
Indian Institute of Science
Bangalore – 560 012 (INDIA)

FEBRUARY 2020

Signature of the Author: . . . .

Ashok Krishnan K. S.
Dept. of Electrical Communication Engineering
Indian Institute of Science, Bangalore

Signature of the Thesis Supervisor: . . . .

Vinod Sharma
Professor
Dept. of Electrical Communication Engineering
Indian Institute of Science, Bangalore

Acknowledgements

I would like to thank all those who made this work possible.

Firstly, I would like to thank my advisor, Prof. Vinod Sharma, who has been a major influence. His immense mathematical knowledge has furthered my understanding greatly. His words have continuously inspired me to explore newer territories in learning. I thank him for his support and intellectual input.

Secondly, I thank my labmates, for making life interesting. Deekshith, Santanu, Sahasranand, Jithin, Karthik, Sudheer, Gautam, Uday, Satya, KC, Shahid and all the others at PAL Lab have gifted me a lifetime of experiences.

Thirdly, I thank all my other friends at IISc for making life fun. I owe a lot to their support and encouragement. Thanks to Avishek for being there. And special thanks to my parents and my sister for their love.

Abstract

We consider a multihop wireless network. There are multiple flows in the network, moving from their respective sources to destinations, across multiple hops. These flows will have Quality-of-service (QoS) requirements as well, depending on the applications which generated them.

In the first part of the thesis, we will provide a joint power allocation, scheduling and routing policy, under the SINR model. This policy also has provisions for providing mean delay and hard deadline QoS guarantees, using a system of dynamic weights. The algorithm is implemented in a distributed manner using gossip algorithms. We show that the algorithm stabilizes a fraction of the capacity region. We also compare the performance of the algorithm with other existing algorithms by means of extensive simulations, and demonstrate its efficacy in providing QoS on demand.

In the second part, we solve the scheduling and routing problem for a network with graphical interference constraints. This model, although less general than the SINR model, is also widely used. Using the notions of Draining Time and Discrete Review from the theory of fluid limits of queues, an algorithm that meets end-to-end mean delay requirements of various flows in a network is constructed. The algorithm involves an optimization which is implemented in a cyclic distributed manner across nodes by using the technique of Iterative Gradient Descent, with minimal information exchange between nodes. The algorithm uses time varying weights to give priority to flows, and thus provids mean delay and hard deadline QoS. We also demonstrate that a modified version of the algorithm is throughput optimal, using Lyapunov drift analysis.

In the third part, we obtain the diffusion approximation of the above system under heavy traffic. This is done because the stationary distribution of the system is not tractable. We show that the stationary distribution of the scaled process of the network converges to that of the Brownian limit. Thus we obtain approximations for the mean queue length of the system under stationarity. This theoretically justifies the performance of the system, and simulations further verify our claims.

In the fourth part, we consider the problem of minimizing average age in a multihop wireless network. There are multiple source-destination pairs, transmitting data through multiple wireless channels, over multiple hops. We propose a network control policy which consists of a distributed scheduling algorithm, utilizing channel state information and queue lengths at each link, in combination with a packet dropping rule. Dropping of older packets locally at queues is seen to reduce the average age of flows, even below what can be achieved by Last Come First Served (LCFS) scheduling. The proposed scheduling policy obtains average age values close to a theoretical lower bound, and performs better than many existing algorithms in the literature.

Publications from the Thesis

1. Ashok Krishnan K. S. and Vinod Sharma. “A distributed algorithm for quality-of-service provisioning in multihop networks,” in the proceedings of the *Twenty-third National Conference on Communications (NCC)*, IIT Madras, Chennai, India, March 2-4, 2017.

2. Ashok Krishnan K. S. and Vinod Sharma. “A distributed scheduling algorithm to provide quality-of-service in multihop wireless networks,” in the proceedings of the *IEEE Global Communications Conference(GLOBECOM) 2017*, Singapore, December 4-8, 2017.

3. Ashok Krishnan K. S. and Vinod Sharma. “Distributed Control and Quality-of-Service in Multihop Wireless Networks,” in the proceedings of the *IEEE International Conference on Communications (ICC) 2018*, Kansas city, MO, USA, May 20-24, 2018.

4. Ashok Krishnan K. S. and Vinod Sharma. “Providing Quality-of-Service in Multihop Wireless Networks: Diffusion Approximation”, in the proceedings of the *International Conference on Advances in Applied Probability and Stochastic Processes (ICAAP&SP)*, CMS College, Kottayam, Kerala, India, 7-10 January 2019.

5. Ashok Krishnan K. S. and Vinod Sharma.“Minimizing Age of Information in a Multihop Wireless Network”, accepted for presentation at *IEEE International Conference on Communications (ICC) 2020* .

6. Ashok Krishnan K. S. and Vinod Sharma. “Quality-of-Service in Multihop Wireless Networks: Diffusion Approximation”, submitted to *IEEE Transactions on Wireless Communications.*

Contents

List of Figures

List of Tables

List of Symbols

$\exists$	there exists
$a.s.$	almost surely
$a \wedge b$	$\min(a, b)$
$a \vee b$	$\max(a, b)$
a^+	$\max(a, 0)$
$\lvert x \rvert$	modulus of x, if x is a real number; if x is a vector, its norm
$A \cap B$	intersection of sets A and B
$A \cup B$	union of sets A and B
$\overline{A}$	convex hull of set A
A_+	for a set $A \subset \mathbb{R}$, equals $A \cap \mathbb{R}_+$
A^c	complement of a set A
$\lvert A \rvert$	cardinality of a set A
$\mathbb{P}[A]$	probability of event A
$\mathbb{E}[X]$	expectation of random variable X
$\mathbb{E}_x[X(t)]$	$\mathbb{E}[X(t) \lvert X(0) = x]$
$\xrightarrow{\mathscr{L}}$	convergence in distribution
$\overset{\mathscr{L}}{=}$	equality of distribution of two random variables
$\approxeq$	approximately equal
$f(x)$ is $O(g(x))$	there exists M such that for all x large enough, $\lvert f(x) \rvert \leq Mg(x)$
$u.o.c.$	uniformly on compact sets
$\nabla f(x)$	gradient of function f at x
$\mathbb{R}$	set of all real numbers
$\mathbb{R}_+$	set of all non negative real numbers
$\mathbb{Z}$	set of all integers
$\mathbb{Z}_+$	set of all non negative integers

List of Notation

$\mathcal{V}$	Set of Nodes
$\mathcal{E}$	Set of Edges
$\mathcal{H}$	Set of Channel States
$\mathcal{F}$	Set of Flows
$\mathcal{S}$	Set of Schedules
Q_i^f	Queue Length of flow f at node i
A_i^f	Cumulative Exogenous Arrivals to Q_i^f
D_i^f	Cumulative Departures from Q_i^f
R_i^f	Cumulative Arrivals to Q_i^f by routing
S_{ij}^f	Cumulative number of packets of flow f served on link (i,j)
H_{ij}	Channel gain across link (i,j)
$\mu_{ij}^f(h,I)$	Rate to link (i,j) under channel state h, schedule I
E_h	Cumulative slots when channel gain was h
G_{ijf}^{hI}	Time with channel h, schedule I, flow f scheduled on (i,j)
Λ	Capacity region of the network

Chapter 1

Introduction

Over the last few decades, with the growth of wireless connectivity and mobile systems, wireless networks have become an important aspect of the communication landscape [77, 12, 74]. While telephonic systems and the early internet were predominantly systems with wired connections, the cellular revolution resulted in an accelerated growth of wireless connectivity. A number of wireless communication technologies and standards exist, such as LTE, WiFi, WiMAX, Bluetooth, RuBee, Z-Wave and ZigBee. These cover a variety of purposes, ranging from long range mobile communication to short range local area communication. Wireless networks support a wide variety of applications including voice (VoIP), text (email), video (live streaming, peer-to-peer), online gaming, cloud storage/data processing, home automation and remote healthcare. We now live in a world where mobile communication is the norm. Consequently, networks of devices connected over a wireless medium are of great practical interest. The control of such networks, tailored to meet the requirements of different applications, is an active area of research.

More recently, there has been a lot of interest in the Internet of Things (IoT) [4]. This envisages the coming together of different kinds of applications, using different physical methods of communication and standards, talking to each other [5]. Each application will have its own requirements. These will determine the Quality of Service (QoS) criteria for the packets corresponding to that application. Some applications require an end-to-end mean delay guarantee on the packets being transmitted. Some others, such as a live streaming video, may require all packets to satisfy a hard delay requirement. In some cases, the QoS constraint is a bandwidth requirement for the user. Services involving VoIP (Voice over IP) are sensitive to delay and delay variability in the network, and require preferential treatment over other packets [63]. Another service that requires QoS is remote health-care, which involves collection of data about a patient from a remote location and transmitting it elsewhere to be analysed [81]. Applications that involve live monitoring may require a low Age of Information. It becomes important to design network control policies that can service different flows arising from different applications, catering to heterogeneous requirements. Further, catering to different classes of customers, who have different requirements, and who will pay the service provider differently, will require the system to provide differential QoS.

The control of a wireless network consists of a number of aspects. Since a network will have a number of devices communicating to each other over a shared wireless medium, a number of questions arise. The primary question is how the devices share the network resources (time and bandwidth) amongst each other. The devices may be running applications, which require a certain level of communication performance. How to meet these performance requirements is another question. These questions are dealt with by a network controller. The implementation of the functionalities of the network controller is yet another design problem. In this thesis, we discuss these three questions from a theoretical perspective. Thus, we are dealing with the questions of scheduling, routing and power control in wireless networks. We want to formulate network control policies that do the above, while satisfying QoS requirements of different flows. In this work, we will consider the following forms of QoS: mean delay guarantees, hard deadline guarantees and (average) age of information. Furthermore, we seek to implement these control policies in a distributed manner, as well as study their performance theoretically.

While directly solving a QoS constrained network problem is not always feasible, owing to complexity, one may come up with appropriate approximate solutions. These are obtained using a variety of techniques. In some cases we replace the function being optimized with an approximation. In some other cases we relax the constraints involved. Often, insights may be obtained by studying the network behaviour in certain scaling regimes. The asymptotic behaviour of the network in these regimes often have a direct bearing on the actual performance of the network. In this thesis, we will be using some of these techniques to obtain useful insights into the performance of network control policies.

1.1 Related Work

The stability of a wireless network was studied in [94], where the capacity region of a network was defined as the set of all arrival rate vectors for which a stabilizing policy exists. The queue weighted maxweight algorithm was shown to be throughput optimal, i.e., it stabilized all points in the interior of the capacity region. In [71], a joint scheduling, routing and power control policy was obtained for a multihop wireless network, and was shown to be throughput optimal. This policy involves maximizing the sum of a rate-backpressure product over links. Further, a distributed policy was also proposed, which was less complex to implement than the actual algorithm (though it was not throughput optimal). While backpressure based algorithms offer good performance in terms of stability, they may not yield good delay performance [83], especially under light loads [26]. In [21] the authors propose a distributed scheme that is guaranteed to achieve at least one-third of the capacity region, by generating a maximal matching between the nodes. This work assumes a graphical interference model. In [104], the authors show that for a network with graphical interference constraints satisfying a condition known as *local pooling*, distributed algorithms can achieve maximal throughput. A scheme which maximizes the expected value of the rate-differential backlog metric was proposed in [53], under the SINR interference model.

A randomized scheduling policy that converges to the throughput optimal policy was proposed in [93]. This involves selecting a schedule randomly in every time slot, comparing with the performance in the previos time, and picking the better schedule. Building on this idea, a distributed network control scheme which stabilizes the network for a fraction of the capacity region was given in [59], under the SINR interference model. This algorithm used *gossip* algorithms [16, 30] to implement the optimization in every slot, in a decentralized manner. Gossip refers to randomized local communication (message passing) between neighbouring nodes. By means of such local exchanges, one can estimate global properties of the network, with probabilistic guarantees. These guarantees will depend on the network topology as well. See [78] for a comprehensive survey on gossip algorithms.

Another allocation rule which is known to be throughput optimal is the exponential rule, in [82]. In this work, throughput optimality is demonstrated using the technique of *fluid limits*. The use of fluid limit techniques to study networks goes back to works such as [76, 27, 87]. A fluid limit is a limit of the network process along a scaling regime, corresponding to the Functional Strong Law of Large Numbers (FSLLN) [98]. The fluid limit is called *stable* if the fluid queues reach the value zero in finite time. For many queueing models, stability of the fluid limit implies stability of the underlying stochastic system [2, 22] (positive recurrence of the underlying Markov Process). A generalized criterion for concluding the stability of a queueing (Markov) process from its fluid model is provided in [32]. The converse, i.e., stability of the stochastic model implying the stability of all associated fluid models is not true in general. In [18], a queueing system is presented, the fluid limit of which is not stable. Surprisingly, the underlying stochastic system is stable. In [66], it is shown that instability of the fluid model implies the transience of the stochastic model. A notion of L_2 stability of the fluid model is shown to be equivalent to various stability notions of the original stochastic system, in [56]. In [10], the authors propose a linear programming method to test the stability of fluid models in work conserving mutliclass queueing networks. Commonly, Lyapunov functions are used to test the stability of fluid network models. In [102], it is shown that, for a generic fluid network, the existence of a Lyapunov function is a necessary and sufficient condition for stability.

The fluid limit can also be used to obtain insights apart from stability. In [28], the authors provide sufficient conditions for obtaining bounds on the steady state moments of queue lengths in a multiclass queueing network. They also prove polynomial rates of convergence of mean queue length to its steady state value. These results are obtained combining fluid limit techniques and Markov chain theory [65]. Optimizing the fluid equivalent of a cost function is studied in [64]. The notion of fluid scale asymptotic optimality (FSAO) is used, and it is shown, that under certain conditions, the policy that is optimal with the given cost function, will also satisfy FSAO. The technique of discrete review, inspired by BIGSTEP policies in [39, 40], is used in [62]. Here, the network is viewed at certain review instants, and control decisions are taken till the next review instant using information from the current state. They also demonstrate FSAO.

In [23], an optimal infinite horizon fluid control policy is created by joining piecewise optimal policies, each of which is optimal for a period of time. In [43] a throughput optimal, per-queue based scheduling algorithm is presented. In [6], it is shown that the class of asymptotically optimal policies contains the class of time average optimal policies, and that the value function of the fluid model is a lower bound to the value function of the stochastic network. In [79], the authors study networks under multiplicative state space collapse, using a fluid scale analysis that does not assume complete resource pooling. A robust fluid model, obtained by adding stochastic variability to the conventional fluid model, is discussed in [11]. Another algorithm using per hop queue length information, with a low complexity approximation that stabilizes a fraction of the capacity region is given in [60]. A draining time based scheduling and routing algorithm to provide improved delay performance is given in [90]. The authors prove stability under this policy for a two node relay network. A comprehensive overview of different control techniques using fluid limits, and their analysis, is given in [65].

Another scaled approximation of networks is diffusion approximation. This is obtained by scaling network processes in the regime corresponding to the Functional Central Limit Theorem (FCLT) [13]. The networks are scaled while simultaneously increasing the traffic intensity to the boundary of capacity. This is called the Heavy Traffic regime [98]. Early work in this line includes [41] and [42]. A weak limit is obtained for a sequence of scaled processes. In many common systems this limit turns out to be a Reflected Brownian Motion (RBM) [38]. This limiting process provides approximations for different statistics, such as mean delay and queue length, of the queueing network. Sufficient conditions for the existence of a diffusion limit for multiclass queueing networks is given in [99]. This assumes a work conserving service policy, i.e., the queues are never idle when a customer is present. In [17], state space collapse is demonstrated, for diffusion scaled queueing networks with First In First out (FIFO) and Head of the Line processor sharing service disciplines. State space collapse is demonstrated for the fluid model first. Then, viewing the diffusion scaled paths as scaled and restarted fluid sample paths, the properties of the two are related.

In [88], the fluid limit of a maxweight scheduling policy, in a discrete time queueing network, is obtained. Here, work conservation holds only asymptotically. They use techniques from [17] to demonstrate state space collapse. The problem of routing arrivals to parallel resources is studied in [95], in the heavy traffic regime. In [47], fluid and diffusion approximation models are developed to study internet congestion control, operating under an α-fair bandwidth sharing policy. Approximations for the queue length of networks in heavy traffic is given in [31]. A recent concise survey of the use of diffusion approximation in queueing networks is provided in [68].

The diffusion limit has a stationary distribution, which is easier to calculate than the stationary distribution of the actual system. This provides an approximation for various system statistics of interest, such as mean queue length and delay. However, earlier papers on diffusion approximation did not provide convergence of stationary distributions. The first paper to do so seems to be [33] for general Jackson networks. They obtain convergence under the assumption that the inter-arrival and service times have exponential moments. In [20], convergence is shown under weaker assumptions. They use techniques refined from [28] to obtain sufficient conditions for convergence of the distributions. These limit exchange arguments require the Lipschitz continuity of an associated Skorohod map. The same problem, in the context of multiclass queueing networks, is solved in [48] and [103]. Sufficient conditions for the exchange of limits in multiclass networks is provided in [35]. These are conditions on the convergence rate of a fluid limit to an invariant manifold. In [37], the exchange of limits is proven in the case of stochastic fluid networks. The tightness of a sequence of diffusion scaled stationary distributions, in the Halfin-Whitt asymptotic regime, is given in [89]. In this regime, along with the service rate, the number of servers is also scaled up. This model finds application in call centre traffic analysis. In [19], the authors justify the heavy traffic diffusion approximation by showing convergence of moment generating functions (MGF) of the stationary distributions of diffusion scaled processes. To do so, they use the basic adjoint relationship to characterize the MGF. Using this method, they bypass the intermediate step of showing the existence of the diffusion process, in the spirit of [54], which provided a two moment approximation for the mean delay of a $GI/GI/1$ queue.

Providing different types of Quality-of-Service (QoS) to different flows has been explored using different models. In the network utility maximization (NUM) framework [51, 52], the network is modelled as a system of flows. One seeks to optimize a utility function of flow rates, subject to flow constraints. The choice of utility function would determine the fairness criterion involved in giving differential service to flows. The NUM problem may be considered a variant of the weighted sum-rate maximization problem [97]. As the name suggests, in such schemes, one optimizes a weighted sum of rates. Such a problem is in general quite computationally complex. For instance, maximizing the sum rate under the SINR rate model, is non-convex and NP-hard [61]. Approximate solutions to the sum rate maximization problem are provided in [92], which uses SINR approximations and a max-min weighted SINR optimization. A number of dual decomposition schemes, using primal/dual (sub)gradient methods, to solve the NUM problem in a distributed manner, are given in [72]. A weighted backpressure scheme to address various QoS requirements such as average delay and throughput is proposed in [86].

While explicitly providing QoS guarantees may not be easy, one may obtain approximate guarantees. In the large queue length regime, one approach to provide mean delays is to translate these requirements in terms of *effective bandwidth* and *effective delay* from Large Deviations theory [29], and obtain solutions in the physical layer. In [84], the authors use this technique for a K-user downlink scenario. Such techniques, however, cannot be applied easily in the multihop context, owing to the complex coupling between the queues, which makes it difficult to have a simplified one-to-one translation between delay requirements and control actions [25]. Using Markov Decision Processes (MDPs) [73] has been another approach to provide QoS [85]. In general settings, however, MDPs are not easy to handle owing to the huge size of the state space. Control based on Lyapunov optimization is quite popular in the multihop network setting. However, under general network models, it may be complex [34].

In general, it is not possible to design a high throughput, low complexity, low delay network control policy [80]. However, one may not need to meet all these requirements simultaneously, and for all flows. In [24], each node continuously keeps track of the minimum end-to-end delay, bandwidth and cost from that node to every other destination node. Given the QoS requirements for a flow, multiple paths are probed, from source to destination, and a feasible path is chosen using a scheme of forwarded 'tickets', which will collect the delay information along feasible paths. In [14], a one-to-one relationship is assumed between the given QoS constraints and the SINR. Thus, one can convert QoS constraints to SINR constraints. Under the additional assumption that the function mapping the feasible QoS set to the corresponding SINR values is log-convex, one can show that the feasible QoS region is a convex set. However, this additional assumption may not always hold. In [58, 57], the problem of minimizing power while providing mean and hard delay guarantees is studied. However the algorithm requires knowledge of system statistics and is not throughput optimal.

Age of Information (AoI) [50, 49] is a recently introduced and increasingly popular QoS metric. In [50], the problem of minimizing the average AoI for $M/M/1$, $D/M/1$ and $M/D/1$ queues, under the First Come First Served (FCFS) discipline, is studied and analytical expressions were obtained for Average AoI for the first two cases. However, obtaining explicit expressions for AoI may not be easy under other service disciplines or complex network assumptions. Later works looked at AoI for other single queue models, such as sharing of an $M/M/1$ FCFS queue by two traffic streams [100], an $M/M/1$ Last Come First Served (LCFS) queueing system with and without preemption [101], and an $M/M/2$ system [46]. In [45], the authors consider a single base station, with a number of nodes trying to communicate time-sensitive data to it. They propose three policies to minimize average AoI subject to throughput requirements. They further show that the AoI obtained in their policies is a multiplicative factor away from the optimal value. In [69], for a single queueing system the authors study the problem of giving preemptive priority to one flow over another. They obtain closed-form expressions for average age and average peak age.

In [7], the authors consider a multihop network with a single flow. Under the assumption that service times are exponentially distributed, they show that the (preemptive) Last Come First Served (LCFS) service discipline minimizes the age among all disciplines, in a stochastic ordering sense. In [91], the authors study distributed stationary policies that are not dependent on the channel state. Using these policies, they obtain tractable expressions for Average and Peak AoI, which are then optimized over this class of policies. However, this class of policies may be a small subset of all possible policies, and therefore not very likely to contain the policy that minimizes age among all possible policies. In [44], the authors propose an age based maxweight type scheduling policy that is throughput optimal, and further provide heavy traffic approximations for its performance. A concise survey covering diverse aspects of AoI, and giving a number of available AoI results for different system models, is [55].

1.2 System Model

In this work, we will be studying a multihop wireless network (see Fig 1.1). Such a system consists of a number of nodes communicating to each other over a wireless medium. The nodes represent communication devices. There will be flows, generated at some nodes (called source nodes), destined to some other nodes (called destination nodes). These flows consist of packets that have to be delivered. Some of these flows will also have service requirements. These are referred to as Quality-of-Service (QoS) requirements. These may be of different forms, such as an upper bound on the average end-to-end delay, or a minimum rate requirement. The type

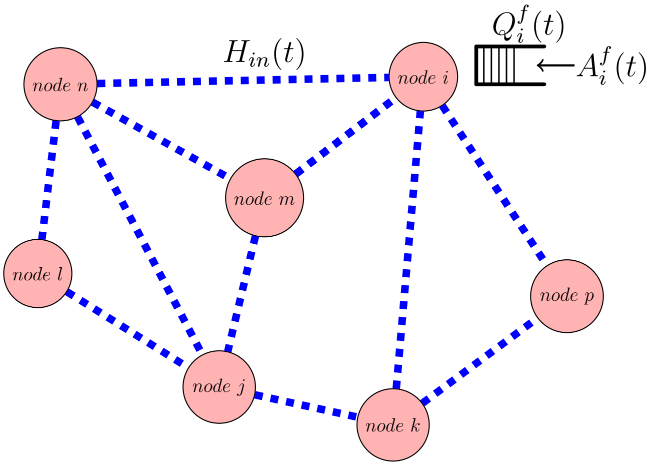

Figure 1.1: A simplified depiction of a multihop wireless network. Flow f starts from node i, and hence has $src(f) = i$.

of QoS required by a flow depends on the application that generates it. An application that live streams videos, for example, may have stringent delay requirements to be met by all the packets that go from the source to the destination.

The network (Fig 1.1) is modelled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes (vertices) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ being the set of edges (links) on $\mathcal{V}$. This network evolves in discrete time $t = 0, 1, 2, 3, \ldots$. There are multiple source nodes sending packets to destination nodes across the network. Each such stream of packets is called a *flow*. Denote the set of all flows by $\mathcal{F}$. For a flow $f \in \mathcal{F}$, we use $src(f)$ to denote its source node, and $des(f)$ to denote its destination. For a flow f, the discrete time random process denoting the arrival of packets to its source node is denoted by $A^f_{src(f)}(t)$. The nodes are connected by a time varying wireless channel. The channel gain of the wireless link between nodes i and j at time t will be denoted by $H_{ij}(t)$.

At each node, we will have multiple queues, one for each flow that passes through it. The queue length corresponding to flow f at node i will be denoted by $Q_i^f(t)$. At each time instant t, the system makes a control decision, as to how many packets from each flow are to be transmitted over which links. This decision could be done in a centralized or distributed manner. As a consequence of this decision, we obtain the scheduling and routing variables, $S_{ij}^f(t)$, which denotes the number of packets of flow f that are to be transmitted over link (i, j) at time t. As a consequence of the control decision, the queues evolve as,

$$Q_i^f(t+1) = Q_i^f(t) - \sum_{j \in \mathcal{V}} S_{ij}^f(t) + \sum_{k \in \mathcal{V}} S_{ki}^f(t), \tag{1.1}$$

for all nodes $i \neq des(f)$ (since at the destinations the packets of the corresponding flow are absorbed).

We will use $Q(t)$ to denote the vector $[Q_i^f(t)]_{i \in \mathcal{V}, f \in \mathcal{F}}$. Similarly we have $H(t) = [H_{ij}(t)]_{(i,j) \in \mathcal{E}}$, $A(t) = [A_i^f(t)]_{i \in \mathcal{V}, f \in \mathcal{F}}$ and $S(t) = [S_{ij}^f(t)]_{(i,j) \in \mathcal{E}, f \in \mathcal{F}}$. Under usual assumptions on the arrival and service, we will be able to show that the process $Q(t)$ evolves as a discrete time Markov chain. This Markov chain is said be *stable* if it is positive recurrent.

In this thesis, we will be considering two different channel models: the SINR model, and a graph based interference model. While the SINR model is more general, optimizing network metrics under this model can be quite complex, due to the highly non linear interaction between transmit rates of different links. Characterizing optimal control, therefore, is not easy. Graph based models, on the other hand, allow us more freedom in characterizing performance. One may use such models to gain insights about network performance. However, it must be noted that obtaining throughput optimal control policies for scheduling and routing under both models is generally an NP-hard problem [97].

In this thesis we will be providing different algorithms to design the $S(t)$ process so as to achieve flow requirements. These requirements include stability of the queue length process, the mean delay meeting a deadline, the delay meeting a hard deadline with high probability, and the age of the flow.

1.3 Contributions and Organization

In this thesis, we study the problem of wireless network control with QoS guarantees. To this end, we propose different algorithms under different channel (interference) models, and analyze their performance.

In Chapter 2, we study the problem of joint scheduling, routing and power control of a multihop wireless network under the SINR interference model. We obtain a randomized control policy for the same, which also contains provisions for mean delay and hard deadline guarantees. Flows are given higher priority using a system of dynamic weights, which depend on queue length as well as on whether the flow is meeting delay requirements at the destination node. This algorithm is implemented in a distributed manner using gossip algorithms. Theoretically, we show that the algorithm stabilizes a fraction of the capacity region. From simulations, we can see that the algorithm outperforms similar randomized or distributed algorithms in the literature.

In Chapter 3, we study the network scheduling and routing problem in a wireless network with graphical interference constraints. We propose an algorithm, inspired by the notion of draining time in fluid limits, to solve the control problem while giving QoS provisions. The QoS provided are mean delay and hard deadline guarantees. The network control follows a system of Discrete Review; here, control decisions are not made at every time slot. Instead, they are made at the beginning of review periods, and the decisions are used to operate the network till the beginning of the next review period. We also implement the algorithm in a distributed fashion using an Incremental Gradient Ascent scheme. It is shown that the distributed algorithm converges to the optimal value of the original centralized formulation. A modified version of the algorithm is shown to be throughput optimal, by means of fluid limit analysis. For this, we first obtain the fluid limit of the system state process, which is an ordinary differential equation (o.d.e.). The o.d.e. trajectory is shown to be stable by constructing a suitable Lyapunov function. The stability of the fluid limit o.d.e. implies the stability of the system. We also see that the system provides good QoS performance.

In Chapter 4, we obtain the Diffusion approximation of a wireless network under graphical interference constraints, for a control policy, similar to that of Chapter 3, under heavy traffic scaling. A scaled sequence of processes is shown to converge weakly to a Brownian motion with drift. This is done by decomposing the scaled workload process into two components, one which converges weakly to a Brownian motion with drift, and the other which converges to the regulating process corresponding to the Brownian motion. Consequently, the resulting process is a Reflected Brownian Motion (RBM) with drift. This RBM has a stationary distribution, which can be used as a proxy for the stationary distribution of the actual system. We show that this approximation is theoretically justified, by proving that the sequence of stationary distributions of the scaled systems converges to this distribution. We also verify this by means of simulations. To the best of our knowledge, this is the first work to provide a throughput optimal algorithm with a QoS provision.

In Chapter 5, we obtain an control algorithm that deals with the Age of Information (AoI) problem in a multihop wireless network. We provide an algorithm that provides low average AoI for flows. This is achieved by combining packet drops at nodes along with a weighted control policy that uses queue and channel state information. This policy is motivated by our policy in Chapter 3. By comparing with a theoretical lower bound, we demonstrate that the policy is close to optimal. Using dynamic weights, we demonstrate how the average AoI of flows can be selectively reduced and brought close to the lower bound. Simulations also show that the algorithm performs better than standard algorithms in the literature. We also demonstrate how the control policy can be implemented in a distributed manner.

In Chapter 6 we conclude the thesis, and present directions for future research.

Chapter 2

Joint Power Allocation, Routing and Scheduling under the SINR model

In this chapter, we present a distributed algorithm for joint power control, routing and scheduling in multihop wireless networks. The algorithm also provides for Quality of Service (QoS) guarantees, namely, end-to-end mean delay guarantees and hard deadline guarantees, for different users. It is easily implementable and works by giving local dynamic priority to flows requiring QoS, the priority being a function of the queue length at the nodes. We prove that the algorithm stabilizes all arrival rates in a fraction of the capacity region. We also compare the performance of the algorithm with other existing algorithms by means of extensive simulations, and demonstrate its efficacy in providing QoS on demand.

2.1 System Model

We have a wireless multihop network (see Fig. 2.1) represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, evolving in discrete time. As described earlier, the set of flows is $\mathcal{F}$, the arrival process for flows is $A(t)$, the channel process is $H(t)$, and the control vector is $S(t)$. We will assume that the arrival process is independent and identically distributed (i.i.d.) across time, and also that it is independent across flows. The mean arrival rate is $\lambda_i^f = \mathbb{E}[A_i^f(t)]$, and the mean arrival rate vector is $\lambda = [\lambda_i^f]_{i \in \mathcal{V}, f \in \mathcal{F}}$. The channel process $H(t)$ takes values from a finite set $\mathcal{H}$ and evolves i.i.d. across time, with distribution γ.

Let us denote by $P_{ij}(t)$ the power used by node i to transmit to node j in time slot t. The vector $P(t)$ denotes $[P_{ij}(t)]_{(i,j) \in \mathcal{E}}$. The set of powers that are allowed will be denoted by $\mathcal{P} = [0, P_{max}]^{|\mathcal{E}|}$. This encapsulates constraints on the power. The rate of transmission from node i to node j at time t will be denoted by $\mu_{ij}(t)$, which is an achievable rate function,

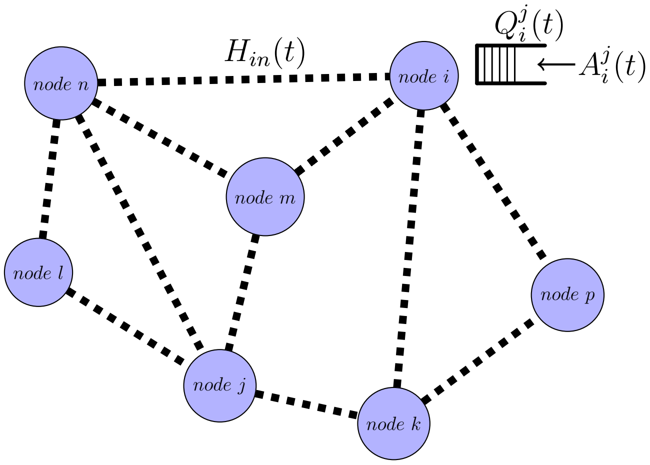

Figure 2.1: A simplified depiction of a Wireless Network

dependent on the channel state $H(t)$ and the power allocation $P(t)$. In this chapter we will be using the *SINR* (Signal to Interference plus Noise Ratio) rate function,

$$\mu_{ij}(P(t), H(t)) = \log_2\left(1 + \frac{P_{ij}(t)H_{ij}(t)}{N_j + \sum_{k\in\mathcal{V},k\neq i}\sum_{l\in\mathcal{V}} P_{kl}(t)H_{kj}(t)}\right), \tag{2.1}$$

with N_j denoting the noise power at node j. This rate may be allocated to packets in one or more of the flows in node i, to be transferred to the corresponding queue in node j. The rate vector is $\mu = [\mu_{ij}]_{(i,j)\in\mathcal{E}}$.

Let $S_{ij}^f(t)$ denote the number of packets of flow f transmitted on link (i, j) in time slot t. Then, we may write the queue evolution equation as,

$$Q_i^f(t+1) = Q_i^f(t) + A_i^f(t) + R_i^f(t) - D_i^f(t), \tag{2.2}$$

where,

$$R_i^f(t) = \sum_k S_{ki}^f(t) \text{ and } D_i^f(t) = \sum_j S_{ij}^f(t), \tag{2.3}$$

denote respectively, arrivals and departures by routing from the queue Q_i^f. If we denote the vectors $[Q_i^f(t)]_{i\in\mathcal{V},f\in\mathcal{F}}$, $[R_i^f(t)]_{i\in\mathcal{V},f\in\mathcal{F}}$, $[D_i^f(t)]_{i\in\mathcal{V},f\in\mathcal{F}}$ by $Q(t)$, $R(t)$ and $D(t)$, the queue evolution can be written as,

$$Q(t+1) = Q(t) + A(t) + R(t) - D(t). \tag{2.4}$$

We assume that the set of flows has a subset $\mathcal{F}_Q$, which are flows with QoS constraints. In this chapter, we will be considering hard deadline and mean delay constraints. Define,

$$\Delta_{ij}(t) = \max_f (Q_i^f(t) - Q_j^f(t))^+. \tag{2.5}$$

At any time t, the optimal power allocation $P^*(t)$ is defined to be the power allocation that optimizes the maxweight problem, i.e.,

$$P^*(t) = \arg_{P \in \mathcal{P}} \max \sum_{(i,j) \in \mathcal{E}} \Delta_{ij}(t) \mu_{ij}(P, H(t)). \tag{2.6}$$

To characterize the performance of network control under this power allocation, we first define the *capacity region.*

2.2 Capacity Region

Recall that the rate vector at time t is given by,

$$\mu(t) = \mu(P(t), H(t)) \tag{2.7}$$

where $P(t)$ is the power vector, and $H(t)$ is the channel state, at time t. Define,

$$\mathcal{M}_h = \{\mu(p, h) : p \in \mathcal{P}\}. \tag{2.8}$$

Let $\overline{\mathcal{M}}_h$ represent the convex hull of $\mathcal{M}_h$. Define,

$$\mathcal{M} = \sum_h \gamma_h \overline{\mathcal{M}}_h. \tag{2.9}$$

We will now define the capacity region of the network.

Definition 2.1 *The capacity region, Λ, is the set of all arrival rate vectors λ for which there exists a vector $\varpi = [\varpi_{ij}^f]_{(i,j) \in \mathcal{E}, f \in \mathcal{F}}$ which satisfies,*

$$\varpi_{ij}^f \geq 0, \ \forall i, j, f \tag{2.10}$$

$$\varpi_{ii}^f = 0, \ \forall i, f, \tag{2.11}$$

$$\varpi_{ij}^i = 0, \ \forall i, j, f, \tag{2.12}$$

$$\lambda_i^f \leq \sum_j \varpi_{ij}^f - \sum_k \varpi_{ki}^f, \ \forall i, f, \tag{2.13}$$

$$\sum_{f} \varpi_{ij}^{f} \leq m_{ij}, \ \textit{for some } m \in \mathcal{M}. \tag{2.14}$$

For stability, it is necessary that $\lambda \in \Lambda$,while $\lambda \in int(\Lambda)$ is sufficient, where $int(\Lambda)$ denotes the interior of Λ. An algorithm that stabilizes all $\lambda \in int(\Lambda)$ is called *throughput optimal.*

The following is a well known result.

Lemma 2.1 *The power allocation given by (2.6) is throughput optimal.*

A proof of this is provided in the appendix 2.A.

While the power allocation (2.6) is throughput optimal, it is not easy to solve the given optimization problem. It is in general NP-hard [97]. Hence, there is a need for low complexity algorithms that perform close to the benchmark provided by (2.6). A framework for obtaining such an algorithm was given in [59]. However, using queue state information, one can hugely improve its performance. Moreover, this scheme makes no provision for QoS. Since providing QoS is central to wireless networking systems, we are interested in developing policies that have both throughput guarantees and QoS provisions. Since providing throughput optimality itself is a hard problem, providing policies that have QoS guarantees could be much harder. We develop a low complexity policy that can give QoS, by trading off the network resources between QoS and non QoS flows. In the absence of QoS requirements, the algorithm stabilizes flows.

2.3 A Distributed Scheme Providing QoS

We propose a distributed algorithm (Algorithm 1) for joint scheduling, routing and power control, while also making provision for mean delay guarantees and hard deadline guarantees. This is an extension of the algorithm in [59]. However, it differs substantially from this algorithm on two counts: first, that it uses queue length information in the scheduling process, and second, that it makes provision for QoS as well. The use of queue length information is based on the intuitive idea of giving those nodes that have a higher queue length, a higher probability of becoming a transmitter. This should lead to improvement in performance. In this scheme, those links which have a high queue length at the transmitting side, and a low queue length at the receiving side, have a higher probability of being formed. This is a heuristic approach to backpressure.

In each time slot, we independently generate a random variable χ, where,

$$\chi = \begin{cases} 1, & w.p. \ \sigma, \\ 0, & w.p. \ 1-\sigma, \end{cases}$$

for some $\sigma \in (0, 1)$. This may be generated by one node and communicated to all others by signalling at the beginning of each time slot. Each node i computes,

$$Q_i = \sum_{f \in F} h^f(q_i^f), \tag{2.15}$$

where,

$$h^f(x) = \begin{cases} \theta x^2 \eta^f + x(1 - \eta^f), & f \in \mathcal{F}_Q, \\ x, & f \in \mathcal{F} \setminus \mathcal{F}_Q, \end{cases}$$

with $\theta > 1$. Here, η^f is one if the QoS constraint for flow f was met in the previous time slot, and is zero otherwise. Thus Q_i is a virtual queue length at node i, with extra weight being given to the backlogs of those flows whose QoS requirements were not met. The nodes now use Algorithm 2 (details of the working of Algorithm 2 are given in section 2.3.1), to compute, in a distributed manner, U^*, which is a surrogate for $U = \sum_i u_i$, where $u_i = \min(Q_i, B)$, with B chosen to be a very large number. Node i decides to be a transmitter with probability $\frac{u_i}{U^*}$; else, it becomes a receiver. As a result, nodes with a higher backlog of QoS packets have a higher probability of being a transmitter, and hence, pushing the packets out of itself. The queues with lesser backlog have a higher chance of being receivers. The algorithm thus dynamically moves packets from bigger queues to smaller queues.

Each transmitter tries to randomly pair up with one of its neighbours, and establishes a link if the neighbour chosen was neither a transmitter nor paired with any other node. Each transmitter also picks a random power level for transmission. Over each link thus formed, we schedule the flow that maximizes $(h^f(Q_i^f) - h^f(Q_j^f))^+$ if $\chi = 1$. Else, we choose the flow that maximizes $(Q_i^f - Q_j^f)^+$. During the slots where $\chi = 1$, this will prioritize flows to provide QoS. In other slots, this is needed for stability of the non-QoS flows. The variable χ captures the trade-off between stability and QoS. In a timeslot with $\chi = 1$, the system gives higher priority to QoS delivery, over the stability requirements of the system. The value of $\sigma = \mathbb{P}\{\chi = 1\}$ captures this.

Next, we compute the rate-differential backlog product over each link ij. Let $r_{ij}(t)$ denote the difference of rates under the chosen random power allocation and the power allocation in the previous slot, given by,

$$r_{ij}(t) = [\mu_{ij}(\tilde{P}(t), H(t)) - (1 - \alpha_2)\mu_{ij}(P(t-1), H(t))]. \tag{2.16}$$

The rate-backlog product difference between the random and previous power allocations is given by,

$$M_{ij} = \begin{cases} r_{ij}(t)\Delta_{ij} & \text{if } \chi = 0, \\ r_{ij}(t)(h^{f^*_{ij}}(Q_i^{f^*_{ij}}) - h^{f^*_{ij}}(Q_i^{f^*_{ij}}))^+ & \text{if } \chi = 1. \end{cases} \tag{2.17}$$

We obtain $\tilde{M}$, an estimate of $\sum M_{ij}$ using Algorithm 2. If $\tilde{M} \geq 0$, we use the power $\tilde{p}_i$ at node i; else we use the power used in the previous slot, as well as the corresponding scheduling. To ensure that each node has knowledge of the rate at which it can transmit, all nodes are required to send out signals of $\nu\tilde{P}_i(t)$ and $\nu P_i(t-1)$ (ν being sufficiently small) at the same time; as a result, each node may sense the power it receives, subtract the effect of its own power, and obtain its interference level without coming to know the entire channel state. This technique was used in [71].

The algorithm dynamically gives priority to the queues, depending on whether their QoS constraints have been met. The flows which fail to meet the QoS criterion are given higher weightage in the system, by means of the function h. We describe the working of Algorithm 2 below.

2.3.1 Gossip Algorithm

The gossip algorithm we use works on the following principle: Say we have K independent random variables distributed exponentially with parameters $y_1, y_2, ..y_K$. Then the minimum of these random variables is an exponential random variable with parameter $y_1 + y_2 + .. + y_K$. Hence, in order to compute the sum of K values, generate exponential random variables with these values as parameters, and compute their minimum. The inverse of this random variable is an estimate for their sum. One may generate a number of such random variables and compute the corresponding inverse of their average, for more accuracy.

A gossip algorithm on a graph operates by means of asynchronous exchange of information between neighbouring nodes. Consider the network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a *gossip matrix* P defined on $V \times V$. This matrix represents a *transition probability matrix* (t.p.m.) of a Markov chain associated with this graph. This matrix is assumed to be symmetric and doubly stochastic.

The gossip algorithm seeks to compute a sum. Let the nodes $i \in V$ have associated non-negative values v_i. We seek to find an estimate of $v = \sum_i v_i$ within a level of accuracy. Let each node i generate L random numbers, $\{x_i^\ell, 1 \leq \ell \leq L\}$. Each of these numbers is an independent sample, drawn from an exponential distribution with parameter v_i. Consider the quantity,

$$x_*^\ell = \min_{i=1,\ldots,|\mathcal{V}|} x_i^\ell. \tag{2.18}$$

Clearly, x_*^ℓ is sampled from an exponential distribution with parameter $\sum_i v_i$. Thus, the average $\frac{\sum_{\ell=1}^L x_*^\ell}{L}$ is an estimate of v.

To find x_*^ℓ, we do the following. For a sequence of times $\tau = 1, \ldots, T$, in each time slot, each node i contacts its neighbour j with probability $Q(i,j)$. If node i contacts node j, and if they have values x_i^ℓ and x_j^ℓ, they both update to $\min\{x_i^\ell, x_j^\ell\}$ for each $\ell \in \{1, \ldots, L\}$.

We have the following result for the performance of this algorithm.

Lemma 2.2 *Let $\epsilon, \delta \in (0, \frac{1}{2})$. Let $L = 3\delta^{-2}\log(4\epsilon^{-1})$. Assuming the gossiping matrix is complete, the gossiping algorithm computes an estimate $\tilde{S}$ of the sum S, with $\tilde{S} \in [(1-\delta)S, (1+\delta)S]$ for all nodes with probability greater than or equal to $1-\epsilon$ in time $T = O(\delta^{-2}\log N\epsilon^{-1}\delta^{-1})$.*

The proof is provided in [78]. For completeness, we provide it in the appendix 2.B.

The overall distributed algorithm is given below as Algorithm 1, which uses, in turn Algorithm 2 to compute sums.

Algorithm 1 Distributed Algorithm with provision for QoS

```
if $t = 0$ then $\eta^f \leftarrow 0 \; \forall f \in \mathcal{F}$
end if
while $t \geq 0$ do
    Generate $\chi$, with $\mathbb{P}\{\chi = 1\} = \sigma = 1 - \mathbb{P}\{\chi = 0\}$. Communicate its value to all nodes by signaling.
    At each node $i$ :
    Compute $Q_i = \sum_{c \in \mathcal{F}} h^f(Q_i^f)$.
    Generate $\{X_i^j\}_{j=1}^L$, i.i.d exponential with parameter $u_i = \min(Q_i, B)$.
    By gossiping (Algorithm 2) estimate $X_{min}^j = \min_i \{X_i^j\}_{j=1}^L$.
    Calculate $U^* = \left( \frac{1}{L} \sum_{j=1}^L X_{min}^j \right)^{-1}$.
    Generate $\phi \sim \mathcal{U}[0, 1]$.
    if $\phi < \frac{u_i}{U^*}$ then $i \leftarrow \textit{transmitter}$
    else $i \leftarrow \textit{receiver}$
    end if
    Each transmitter $i$ picks a power $p_i \sim \mathcal{U}[0, p_{max}]$. Pick a neighbour uniformly randomly and send a request to pair (RTP).
    Each receiver $j$ waits for an RTP, pairs up with the first transmitter that sends it an RTP.
    Over any link $(i, j)$ formed, schedule $f_{ij}^* = \arg_{f \in \mathcal{F}} \max \chi(h^f(Q_i^f) - h^f(Q_j^f))^+ + (1 - \chi)(Q_i^f - Q_j^f)^+$.
    Each paired transmitter $i$ beams $\nu \tilde{P}^i$ and $\nu P^i(t-1)$.
    if $\chi = 0$ then $M_{ij} \leftarrow \Delta_{ij} \mu_{ij}(t)$
    else $M_{ij} \leftarrow (h^{f_{ij}^*}(Q_i^{f_{ij}^*}) - h^{f_{ij}^*}(Q_i^{f_{ij}^*}))^+ \mu_{ij}(t)$
    end if
    Generate $\{Y_i^j\}_{j=1}^L$, i.i.d exponential with parameter $M_{ij}$.
    By gossiping (Algorithm 2) estimate $Y_{min}^j = \min_i \{Y_i^j\}_{j=1}^L$.
    Calculate $\tilde{M} = \left( \frac{1}{L} \sum_{j=1}^L Y_{min}^j \right)^{-1}$.
    If $\tilde{M} \geq 0$, use the power and scheduling generated in the current slot. Else, use the power allocation and scheduling from the previous slot.
    For each flow $f$:
    if QoS criterion was satisfied then $\eta^f \leftarrow 0$
    else $\eta^f \leftarrow 1$
    end if
    Update this information in the network using gossiping.
end while
```

Algorithm 2 Gossip Algorithm

```
Each node $i$ has $L$ numbers $Z_i^1, \dots Z_i^L$ with parameter $z_i$.
while $k = 0, 1, .., T$ do at each node
    Choose a neighbour with probability $1/N$. Call it $j$.
    $Z_i^l, Z_j^l \leftarrow \min(Z_i^l, Z_j^l)$ for each $l = 1, \dots, L$.
end while
```

2.4 Performance Analysis

We will obtain a bound on the stability region of Algorithm 1. To this end, we will need the following Lemma, which is a version of Theorem 1 in [59]. For a fixed channel state, let $\mu_{ij}(P)$ denote the rate across link ij under power allocation P. Denote the optimal rates in slot t by $\mu_{ij}(P^*(t))$, where $P^*(t)$ is given by (2.6).

Lemma 2.3 *Let an algorithm have power allocation sequence $\{P(t), t = 0, 1, 2, \dots\}$ and let the rate under its scheduling in time t be $\mu_{ij}(P(t))$, for each link $ij \in \mathcal{E}$, and at each time t. Let $\alpha_1, \alpha_2 \in (0, 1)$. Define the events,*

$$\mathcal{A}(t) := \left\{ \sum_{(i,j)\in\mathcal{E}} \Delta_{ij}(t)\mu_{ij}(P(t)) \geq (1-\alpha_1) \sum_{(i,j)\in\mathcal{E}} \Delta_{ij}(t)\mu_{ij}(P^*(t)) \right\}$$
$$\mathcal{B}(t) := \left\{ \sum_{(i,j)\in\mathcal{E}} \Delta_{ij}(t)\mu_{ij}(P(t)) \geq (1-\alpha_2) \sum_{(i,j)\in\mathcal{E}} \Delta_{ij}(t)\mu_{ij}(P(t-1)) \right\}$$

If there exist $\beta_1, \beta_2 \in (0, 1)$ such that for all t,

$$\mathbb{P}[\mathcal{A}(t)] \geq \beta_1, \mathbb{P}[\mathcal{B}(t)] \geq 1 - \beta_2,$$

the algorithm will stabilize the network for any arrival rate vector $\lambda \in \rho\Lambda$ where $\rho < 1 - (\alpha_1 + (1-\alpha_1)\alpha_2) - 2\sqrt{\frac{\beta_2}{\beta_1}}$.

The proof of this lemma is provided in the appendix 2.C.

While ρ may be a small number, the utility of this result lies in the fact that we can provide a stability result under very general rate models, including the SINR model, which is in general quite difficult to analyze.

From the following Lemmas (2.4 and 2.5) we verify that Algorithm 1 satisfies the two conditions of Lemma 2.3.

Lemma 2.4 *Let $\alpha_1 \in (0,1)$. Then, for Algorithm 1, at every time t, we have $\mathbb{P}[\mathcal{A}(t)] \geq \beta_1$, where $\beta_1 = (1-\beta_3)(\frac{\epsilon}{2(1-\alpha_3^2)N^{3.5}B^2})^N$, with $\beta_3 \in (0,1)$, $\alpha_3 \in (0, \frac{1}{2NB})$ and $\epsilon > 0$.*

Proof: In every slot, the probability of a node being a transmitter is u_i/U^*, where $u_i = \min(q_i, B)$ and U^* is the estimate of $U = \sum_{j \in V} u_j$ obtained by gossiping. Since each u_i is less than or equal to B, their sum, U, cannot exceed NB.

Pick $\alpha_3 \in (0, \frac{1}{2NB})$, and $\beta_3 \in (0,1)$. It follows from Lemma 2.2 that using Algorithm 2 for gossiping, and running for $O(\log(n{\beta_3}^{-1}{\alpha_3}^{-1})/{\alpha_3}^2)$ iterations, returns a value $U^* \in [(1-\alpha_3)U, (1+\alpha_3)U]$ with probability greater than or equal to $1-\beta_3$. We assume that the gossiping algorithm runs for this sufficient number of iterations. Conditioned on this event(which we will call $\mathcal{C}$), we have the probability of selecting any link ab, independent of other links, given by

$$\begin{aligned}\mathbb{P}(\text{link } ab|\mathcal{C}) &\geq \mathbb{P}(a \text{ is txr}|\mathcal{C})\frac{\mathbb{P}(b \text{ is a rxr } |\mathcal{C})}{(\text{no: of neighbours of } a)}\\ &\geq \frac{u_a}{U^*}\frac{1}{N}\left(1 - \frac{u_b}{U^*}\right).\end{aligned}$$

Since $\mathcal{C}$ implies that $(1-\alpha_3)U \leq U^* \leq (1+\alpha_3)U$, we have

$$\begin{aligned}\mathbb{P}(\text{link } ab|\mathcal{C}) &\geq \frac{u_a}{NU(1+\alpha_3)}\left(1 - \frac{u_b}{(1-\alpha_3)U}\right)\\ &= \frac{u_a}{NU(1-\alpha_3^2)}\left(\frac{U-u_b}{U} - \alpha_3\right).\end{aligned}$$

Since $U - u_b = \sum_{j\in V, j\neq b} u_j \geq u_a$, and $U \leq NB$,we have:

$$\begin{aligned}\mathbb{P}(\text{link } ab|\mathcal{C}) &\geq \frac{u_a}{N^2B(1-\alpha_3^2)}\left(\frac{u_a}{NB} - \alpha_3\right)\\ &\geq \frac{1}{N^2B(1-\alpha_3^2)}\left(\frac{1}{NB} - \alpha_3\right),\end{aligned}$$

where we have assumed $Q_a \geq 1$, since any node having total queue length equal to zero can be removed from the set of transmitters, without affecting the system's performance. With B being a large positive integer, $u_a = \min(Q_a, B) \geq 1$. Since we have chosen $\alpha_3 \in (0, \dfrac{1}{2NB})$, we find that:

$$\mathbb{P}(\text{link } ab|\mathcal{C}) \geq \frac{1}{2(1-\alpha_3^2)N^3B^2}.$$

Since the number of transmitter-receiver pairs (links) possible under our assumptions is less than N, the probability of choosing any particular configuration of links is bounded from below by $(\frac{1}{2(1-\alpha_3^2)N^3B^2})^N$. In particular, the probability of chosing the optimal link configuration is bounded below by this value. Since the power vector is chosen independent of the links, and is chosen uniformly randomly over the range $[0, p_{max}]^N$, the probability that the power vector is in an ϵ radius around the optimal power vector is bounded below by $(\frac{\epsilon}{N^{0.5}})^N$, assuming $P_{max} = 1$ (See Lemma 4 of [59] for details).

Since $\sum_{ij\in\mathcal{E}} \Delta_{ij}\mu_{ij}(P(t))$ is a continuous function of $P(t)$ for a fixed link configuration, for any $\alpha_1 \in (0,1)$, there exists $\epsilon > 0$ such that $\mathcal{A}$ is true for any $p(t)$ which satisfies the event $\{||P(t) - P^*(t)|| < \epsilon\}$. We have

$$\begin{aligned}\mathbb{P}[\mathcal{A}(t)|\mathcal{C}] &\geq \mathbb{P}[\mathcal{A}(t)|\mathcal{C},\mathcal{S}^*]\mathbb{P}[\mathcal{S}^*|\mathcal{C}] \\ &\geq \mathbb{P}[\{||P(t) - P^*(t)|| < \epsilon\}|\mathcal{C},\mathcal{S}^*]\mathbb{P}[\mathcal{S}^*|\mathcal{C}] \\ &\geq \left(\frac{\epsilon}{N^{0.5}}\right)^N \frac{1}{(2(1-\alpha_3^2)N^3B^2)^N},\end{aligned}$$

where $\mathcal{S}^*$ is the event corresponding to choosing the optimal link configuration. Using the identity $\mathbb{P}[\mathcal{A}(t)] \geq \mathbb{P}[\mathcal{A}(t)|\mathcal{C}]\mathbb{P}[\mathcal{C}]$, and since $\mathbb{P}[\mathcal{C}] = 1 - \beta_3$,the result follows. □

Lemma 2.5 *Let $\alpha_2, \beta \in (0,1)$. Then, for Algorithm 1, at every time t, $\mathbb{P}[\mathcal{B}(t)] \geq 1 - \beta_2$,where $\beta_2 = \beta + \sigma(1-\beta)$.*

Proof: Let $\mathcal{E}$ be the event $\{\chi = 0\}$. Conditioned on $\mathcal{E}$, at each transmitter i, we generate L exponential random variables, with parameter equal to $M_{ij} = \Delta_{ij}[\mu_{ij}(\tilde{P}(t)) - (1-\alpha_2)\mu_{ij}(P(t-1))]$. We need to estimate the sum $M = \sum_{ij} M_{ij}$, and if $M \geq 0$, we go with the power allocation $\tilde{P}(t)$, else we use $P(t-1)$.

Let $\alpha \in (0,1)$, and pick $L = 3(\alpha)^{-2}\ln(4/\beta)$. Then, assuming the Gossiping Algorithm runs for $T = O(\log(N\beta^{-1}\alpha^{-1})/\alpha^2)$ iterations, it follows from Lemma 2.2 that the estimate $\tilde{M} \in [(1-\alpha)M, (1+\alpha)M]$ with probability greater than or equal to $1-\beta$. Once these many iterations are complete, we have $\{M \geq 0\} \iff \{\tilde{M} \geq 0\}$. We can see that

$$\mathbb{P}[\mathcal{B}(t)|\mathcal{E}] = \mathbb{P}[M \geq 0] = \mathbb{P}[\tilde{M} \geq 0] \geq (1-\beta).$$

Since $\mathbb{P}[\mathcal{B}(t)] \geq \mathbb{P}[\mathcal{B}(t)|\mathcal{E}]\mathbb{P}[\mathcal{E}]$ and $\mathbb{P}[\mathcal{E}] = 1 - \sigma$,the result follows. □

Combining Lemmas 2.3, 2.4 and 2.5, we obtain the following stability result for Algorithm 1.

Theorem 2.1 *Algorithm 1 stabilizes the network for any arrival rate vector* $\lambda \in \rho\Lambda$ *where* $\rho < 1 - (\alpha_1 + (1-\alpha_1)\alpha_2) - 2\sqrt{\frac{\beta_2}{\beta_1}}$.

Proof: Follows from Lemmas 2.3, 2.4 and 2.5. □

Hence, we are guaranteed stability for all arrival rates in the region $\rho\Lambda$. Since δ_1 is decreasing as B increases, the guarantee that one can give in terms of achievable capacity region decreases as a consequence. However, in simulations below we will see that increasing B, or letting it go to infinity, does not reduce the stability region. The value of σ captures a trade-off between QoS and stability.

Comparing our algorithm with [59], we can see that for the same values of α_1 and α_2, we can obtain a better ρ by choosing corresponding values of σ and β. This is borne out by the simulations where we compare the performance of the algorithms in terms of stability region. Also, via extensive simulations we have seen that the algorithm actually provides a much larger stability region than what is dictated by ρ. Thus, it is in fact a practically useful distributed algorithm which provides end-to-end QoS in a multihop wireless network.

Even if the Gossip matrix is not complete, one may obtain the same result. However the number of timeslots in which one needs to operate the gossip algorithm will be much higher. Exact expressions may be calculated for these as well [78].

One may observe that since the algorithm guarantees stability for all arrival rate vectors contained in $\rho\Lambda$, it naturally provides for rate guarantees for any flow that generates packets at a constant rate within this region.

2.5 Simulation Results

For the simulations, we consider networks of 10, 15 and 20 nodes, with the nodes distributed randomly uniformly in a unit square. We assume Rayleigh fading between the nodes, as well as that packet arrivals are i.i.d across slots with Poisson distribution. The rate function, as mentioned earlier, will be the SINR rate function. For all the simulations we will use $\sigma = 0.999$ and $B = 10^5$. While these values reduce the theoretical value of ρ as given by Lemma 2.3, it is evident from the simulations that they enhance the performance.

We first compare the stability region that our algorithm offers, and compare it to two distributed algorithms: Lee [59] and Distributed DRPC [71]. For a network of 20 nodes we see that our algorithm outperforms both the others in terms of stability, when the number of flows is five (Fig. 2.2), as well as when it is fifteen (Fig. 2.3). We plot the change in total queue length as arrival rate at all nodes is increased uniformly. From the figures it is clear that our algorithm offers a huge improvement as far as stability is concerned.

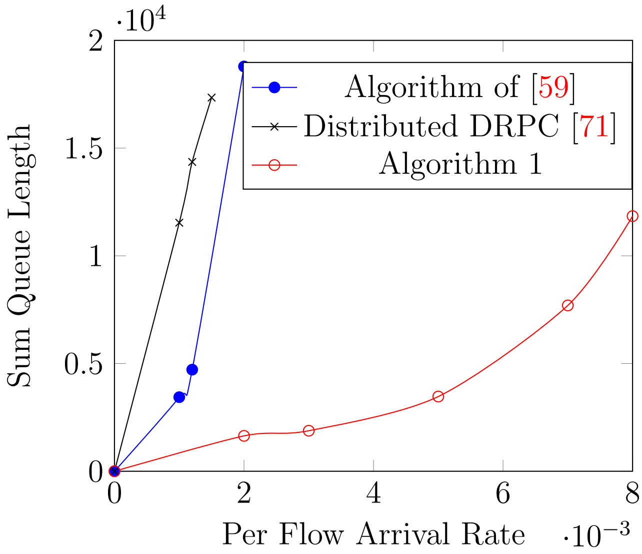

Figure 2.2: Stability Region for our algorithm for a network with 20 nodes and 5 flows

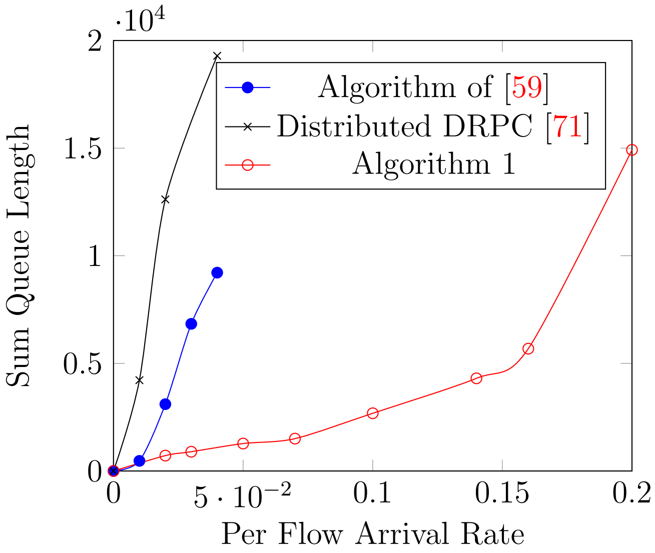

Figure 2.3: Stability Region for our algorithm for a network with 20 nodes and 15 flows

The first QoS parameter that we will consider is mean delay guarantee. For such a flow f, at its destination node, the mean end-to-end delay is computed empirically, by averaging over all packets of that flow that arrive at the destination. If this value is greater than the mean delay required by the flow, the corresponding $\eta^f(t)$ is set to 1. We present case studies of networks of 10 and 15 nodes, with the number of QoS flows being one or two. Each scenario is studied for a fixed value of the arrival rate vector, which is chosen within the capacity region of the network.

Table 2.1 gives the mean delay values of the QoS flow for two cases. Network 1 is a case of 10 nodes with 7 flows, of which one flow requires a mean delay guarantee. Network 2 is a case of 15 nodes with 10 flows of which one requires a mean delay guarantee. The value of the parameter θ used for giving priority, is 10 in both cases.

Table 2.2 corresponds to a network of 10 nodes with 7 flows, of which two flows are mean delay constrained flows, and $\theta = 5$. Table 2.3 is for 15 nodes with 7 flows, of which two flows require mean delay guarantees, and $\theta = 10$.

From the simulations it is evident that the value of θ may be increased in order to gain a better performance. Also, in the case of multiple flows with QoS requirements, the flows are likely to compete with each other as well, in order to have their share of the system resources. In Table 2.3, both QoS flows are given the same priority (as indicated by θ), one may also use different θ values corresponding to different flows. Due to the fact that the system is controlled in a distributed fashion, the number of QoS demands it can support simultaneously may not be huge. One also observes that the mean delay cannot always be brought down below a particular value. This in some sense is the limit of what the algorithm can achieve, given the network resources, for this particular form of the function h. This value is a function of the arrival rate vector.

The next QoS parameter is hard deadline guarantee. In this case the QoS is specified by two values, a delay deadline d and a dropping ratio r, and it is required that no more than r fraction of the packets have a delay more than d. The value of r is estimated empirically, and if this is greater than the required dropping ratio, the corresponding $\eta^f(t)$ is set to 1.

Table 2.4 gives the delay performance of a 10 node network with 8 flows, of which three are QoS flows: two have a mean delay requirement, and one has a hard deadline. To meet the hard deadline, the stability region has reduced. The hard deadline flow has to meet a delay deadline of 70. The mean delay flows have $h(x) = 10x^2$ and the hard deadline flow has $h(x) = 20x^2$. Note that the hard deadline is achieved for 94.9%, 97% and 98% of the packets, as required, with little impact on the mean delay performance. Note that running the algorithm of [59] results in a mean delay of 127 and 104 respectively, for flow 1 and 2 respectively; and the drop ratio for flow 3 is 52.7% (this is the fraction of packets that violates the end to end hard deadline). We see from simulations that we need to set the θ value for flows having hard deadline to be at least twice that for mean delay constrained flows.

Table 2.1: One flow with mean delay requirement

Network 1		Network 2	
Delay Target (slots)	Delay Achieved (slots)	Delay Target (slots)	Delay Achieved (slots)
200	202	350	353
180	181	300	292
150	152	230	236
120	121	200	212
100	100	180	193
80	83	150	160
60	61	120	149

Table 2.2: Two flows with mean delay requirement

Flow 1		Flow 2	
Delay Target (slots)	Delay Achieved (slots)	Delay Target (slots)	Delay Achieved (slots)
230	233	230	231
200	210	200	199
200	198	160	165
160	160	200	201
160	160	140	151
140	141	140	143
120	135	140	157

2.6 Conclusion

In this chapter, we have obtained a distributed algorithm for routing, power control and scheduling of links using queue length dependent cross-layer schemes under the SINR model of interference, while simultaneously providing mean delay guarantees and hard deadline guarantees. Distributed implementation of control was done using gossip algorithms. Simulations demonstrate that the scheme provides significant improvement over existing approaches, as well as its ability to provide delays close to what is demanded by the users. The stability region ex-

Table 2.3: Two flows with mean delay requirement

Flow 1		Flow 2	
Delay Target (slots)	Delay Achieved (slots)	Delay Target (slots)	Delay Achieved (slots)
300	308	300	330
250	248	250	256
200	210	250	270
150	169	200	202
180	182	180	189
160	185	160	179

Table 2.4: Two mean Delays and one hard deadline

Flow 1			Flow 2			Flow 3		
Delay Target (slots)	Delay Achieved (slots)	Mean Delay in [59] (slots)	Delay Target (slots)	Delay Achieved (slots)	Mean Delay in [59] (slots)	Drop ratio Target	Drop ratio Achieved	Drop Ratio in [59]
30	31	127	40	41	104	5%	5.1%	52.7%
30	31		40	41		3%	3%	
30	31		40	40		2%	2%	

pressions, as well as simulations indicate that asking for more QoS effectively diminishes the amount of traffic the system can support.

2.A Proof of Lemma 2.1

We show that the algorithm stabilizes all $\lambda \in int(\Lambda)$. Consider the Lyapunov function,

$$\mathcal{L}(Q(t)) = \sum_{i,f} (Q_i^f(t))^2. \tag{2.19}$$

Define the single step Lyapunov drift,

$$\Delta(t) = \mathbb{E}[\mathcal{L}(Q(t+1)) - \mathcal{L}(Q(t)) | Q(t)]. \tag{2.20}$$

It suffices to show that, for any $\lambda \in int(\Lambda)$, $\Delta(t) < 0$ for $Q(t)$ with values outside a compact set. Observe that, due to the queue evolution equation,

$$\Delta(t) = \sum_{i,f} \mathbb{E}\left[[A_i^f(t) + R_i^f(t) - D_i^f(t)]^2 + 2Q_i^f(t)[A_i^f(t) + R_i^f(t) - D_i^f(t)] | Q(t) \right]. \tag{2.21}$$

Note that the arrival process is independent of the queue state, and the arrival process has finite second moment. Further, we have assumed that the channel gains are bounded and power is chosen from a compact set, which would imply that the rates are bounded. Hence, we can see that there is a finite positive constant B such that,

$$\sum_{i,f} \mathbb{E}\left[[A_i^f(t) + R_i^f(t) - D_i^f(t)]^2 | Q(t) \right] < B. \tag{2.22}$$

We can write,

$$\sum_{i,f} \mathbb{E}\left[Q_i^f(t)[A_i^f(t) + R_i^f(t) - D_i^f(t)] | Q(t) \right] = \sum_{i,f} \mathbb{E}\left[Q_i^f(t)[\lambda_i^f + R_i^f(t) - D_i^f(t)] | Q(t) \right] \tag{2.23}$$

where we used the independence of $A(t$ and $Q(t)$. Since λ is in the interior of Λ, there exists a vector $\varpi = [\varpi]_{ij}^f$ and $\epsilon > 0$ satisfying ,

$$\lambda_i^f + \epsilon < \sum_j \varpi_{ij}^f - \sum_k \varpi_{ki}^f. \tag{2.24}$$

Hence, we may write,

$$\sum_{i,f} \mathbb{E}\left[Q_i^f(t)[\lambda_i^f + R_i^f(t) - D_i^f(t)]|Q(t)\right] \leq \sum_{i,f} \mathbb{E}\left[Q_i^f(t)[-\epsilon + \tilde{d}_i^f - \tilde{r}_i^f + R_i^f(t) - D_i^f(t)]|Q(t)\right], \tag{2.25}$$

where $\tilde{d}_i^f = \sum_j \varpi_{ij}^f$ and $\tilde{r}_i^f = \sum_k \varpi_{ki}^f$. It is easy to see that there exists a stationary policy with random rates $\tilde{R}_i^f = \sum_k \tilde{S}_{ki}^f$ and $\tilde{D}_i^f = \sum_j \tilde{S}_{ij}^f$ such that,

$$\mathbb{E}[\tilde{D}_i^f - \tilde{R}_i^f|Q(t)] = \tilde{d}_i^f - \tilde{r}_i^f. \tag{2.26}$$

Using this in (2.25), we see that,

$$\sum_{i,f} \mathbb{E}\left[Q_i^f(t)[\lambda_i^f + R_i^f(t) - D_i^f(t)]|Q(t)\right] \leq \sum_{i,f} \mathbb{E}\left[Q_i^f(t)[-\epsilon + \tilde{D}_i^f - \tilde{R}_i^f + R_i^f(t) - D_i^f(t)]|Q(t)\right], \tag{2.27}$$

$$= -\epsilon \sum_{i,f} Q_i^f(t) + \mathbb{E}[\sum_{i,j,f}(Q_i^f - Q_j^f)(\tilde{S}_{ij}^f - S_{ij}^f)|Q(t)]. \tag{2.28}$$

From the formulation of the policy, it can be seen that the last term is negative. Moreover, choosing $Q(t)$ to be outside a large enough compact set, we see that $\epsilon \sum_{i,f} Q_i^f(t)$ becomes larger than B. Consequently, the Lyapunov drift $\Delta(t)$ is negative, and the system is stable.

2.B Proof of Lemma 2.2

We are required to compute the time to calculate L different minima by gossiping, with a gossiping matrix P. Note that the time to calculate one minimum is not more than the time for all nodes to get one piece of information, which was initially with n arbitrary node. Hence, we first consider the problem of single piece information dissemination.

We consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $|\mathcal{V}| = n$. A single node has some information which needs to reach all other nodes. At each time t, a node i contacts its neighbour with probability P_{ij}, and once they come in contact, and if one of them has the information, at the end of the time slot, both will have the information. Let $P = [P_{ij}]_{i,j \in \mathcal{V}}$.

Let $\mathcal{I}(t)$ denote the set on nodes that have the information. Clearly, $|\mathcal{I}(0)| = 1$, and $|\mathcal{I}(t)|$ is non decreasing in t and bounded by N, the number of nodes.

Phase I

First we consider all times t such that $\mathcal{I}(t) \leq \frac{n}{2}$. In this phase we will be considerng the *push* aspect of information exchange, i.e., a node which has the information contacts a node without it, and transmits it. The time taken in this phase will be an upper bound for the case with both push and *pull*, the latter being information exchange when a node without information connects to a node with information. Let X_j be the event that a node j which is not in $\mathcal{I}(t)$ at time t receives the information. Then,

$$\begin{aligned} \mathbb{E}[X_j|\mathcal{I}(t)] &= 1 - \Pi_{i \in \mathcal{I}(t)}(1 - P_{ij}), && (2.29)\\ &\geq 1 - \Pi_{i \in \mathcal{I}(t)} \exp(-P_{ij}), && (2.30)\\ &= 1 - \exp\left(- \sum_{i \in \mathcal{I}(t)} P_{ij} \right), && (2.31)\\ &\geq \frac{\sum_{i \in \mathcal{I}(t)} P_{ij}}{2}, && (2.32) \end{aligned}$$

where in the last line we used the fact that for $x \in [0,1]$, $\exp(-x) \leq 1 - \frac{x}{2}$. We may write,

$$\mathbb{E}[|\mathcal{I}(t+1)| - |\mathcal{I}(t)||\mathcal{I}(t)] = \sum_{j \notin \mathcal{I}(t)} \mathbb{E}[X_j|\mathcal{I}(t)] \geq \frac{\sum_{i \in \mathcal{I}(t), j \notin \mathcal{I}(t)} P_{ij}}{2}. \quad (2.33)$$

For the matrix P, its *conductance* $\mho$ is defined as,

$$\mho = \min_{\mathcal{I} \subset \mathcal{V}, |\mathcal{I}| \leq \frac{n}{2}} \frac{\sum_{i \in \mathcal{I}, j \notin \mathcal{I}} P_{ij}}{|\mathcal{I}|}. \quad (2.34)$$

As we have assumed $|\mathfrak{I}(t)| \leq \frac{n}{2}$, it follows that,

$$\mathbb{E}[|\mathfrak{I}(t+1)| - |\mathfrak{I}(t)||\mathfrak{I}(t)] \geq \frac{|\mathfrak{I}(t)|\mho}{2}. \tag{2.35}$$

We seek to find a bound on the time that $|\mathfrak{I}(t)|$ exceeds $\frac{n}{2}$. Let us define,

$$\tau = \inf\{t : |\mathfrak{I}(t)| > \frac{n}{2}\}. \tag{2.36}$$

Define the process,

$$Z(t) = \frac{\exp\left(\frac{\mho t}{8}\right)}{|\mathfrak{I}(t)|}. \tag{2.37}$$

Next, we proceed to show that $Z(\tau \wedge t)$ is a supermartingale. In the case that $|\mathfrak{I}(t)| > \frac{n}{2}$, clearly $\tau \wedge t + 1 = \tau \wedge t + 1$, and hence,

$$\mathbb{E}[Z(\tau \wedge t+1)|\mathfrak{I}(\tau \wedge t)] = \mathbb{E}[Z(\tau \wedge t)|\mathfrak{I}(\tau \wedge t)] = Z(\tau \wedge t). \tag{2.38}$$

Now suppose $|\mathfrak{I}(t)| \leq \frac{n}{2}$. In this case, $\tau \wedge t + 1 = (\tau \wedge t) + 1$. Also, since the function $g(x) = \frac{1}{x}$ is convex for $x > 0$, we can see that, for positive x_1, x_2

$$\frac{1}{x_2} \geq \frac{1}{x_1} - \frac{x_2 - x_1}{(x_1)^2}. \tag{2.39}$$

Substituting $x_1 = |\mathfrak{I}(t+1)|$ and $x_2 = |\mathfrak{I}(t)|$, and noting that, since we are considering only the push effect, $|\mathfrak{I}(t+1)| \leq 2|\mathfrak{I}(t)|$, we obtain,

$$\frac{1}{|\mathfrak{I}(t+1)|} \leq \frac{1}{|\mathfrak{I}(t)|} - \frac{1}{4|\mathfrak{I}(t)|^2}(|\mathfrak{I}(t+1)| - |\mathfrak{I}(t)|). \tag{2.40}$$

Thus one obtains,

$$\mathbb{E}[\frac{1}{|\mathfrak{I}(t+1)|}|\mathfrak{I}(t)] \leq \frac{1}{|\mathfrak{I}(t)|}\exp\left(-\frac{\mho}{8}\right). \tag{2.41}$$

Thus, we have that,

$$\mathbb{E}[Z(\tau \wedge t+1)|\mathfrak{I}(\tau \wedge t)] = \mathbb{E}\left[\frac{\exp\left(\frac{\mho(\tau \wedge t+1)}{8}\right)}{|\mathfrak{I}(\tau \wedge t+1)|}|\mathfrak{I}(\tau \wedge t)\right], \tag{2.42}$$

$$= \exp \frac{\mho(\tau \wedge t)}{8} \exp \frac{\mho}{8} \mathbb{E}\left[\frac{1}{|\mathcal{I}(\tau \wedge t+1)|}|\mathcal{I}(\tau \wedge t)|\right] \leq Z(\tau \wedge t). \quad (2.43)$$

Thus $Z(\tau \wedge t)$ is a supermartingale. Thus, $\mathbb{E}[Z(\tau \wedge t)] \leq \mathbb{E}[Z(\tau \wedge 0)] = 1$. Also note that,

$$Z(\tau \wedge t) \geq \frac{\exp\left(\frac{\mho(\tau \wedge t)}{8}\right)}{n}. \quad (2.44)$$

Thus,

$$\mathbb{E}\left[\exp\left(\frac{\mho(\tau \wedge t)}{8}\right)\right] \leq n. \quad (2.45)$$

Now, $\exp(\frac{\mho(\tau \wedge t)}{8}) \to \exp(\frac{\mho(\tau)}{8})$ as $t \to \infty$. Hence, by the Monotone Convergence Theorem [3], we have,

$$\mathbb{E}\left[\exp\left(\frac{\mho\tau}{8}\right)\right] \leq n. \quad (2.46)$$

Hence,

$$\mathbb{P}[\tau > t] = \mathbb{P}\left[\exp\left(\frac{\mho\tau}{8}\right) > \exp\left(\frac{\mho t}{8}\right)\right] \quad (2.47)$$

$$\leq n \exp\left(-\frac{\mho t}{8}\right). \quad (2.48)$$

For $t = \frac{8}{\mho}\log(\frac{2n^2}{\epsilon})$,

$$\mathbb{P}[\tau > t] \leq \frac{\epsilon}{2n}. \quad (2.49)$$

Thus the time of Phase I is $O(\frac{\log n + \log \epsilon^{-1}}{\mho})$ with probability greater than $1 - \frac{\epsilon}{2n}$.

Phase II

In this phase, $\frac{n}{2} < |\mathcal{I}(t)| \leq n$. In this phase, we consider only the pull aspect of information transfer. Let node $j \in \mathcal{I}(t)^c$. Similar to what we had earlier, in this phase we can write,

$$\mathbb{E}[|\mathcal{I}(t)^c| - |\mathcal{I}(t+1)^c||\mathcal{I}(t)^c] \geq \sum_{j \in \mathcal{I}(t)^c, i \in \mathcal{I}(t)} P_{ji}, \quad (2.50)$$

which implies that,

$$\mathbb{E}[|\Im(t+1)^c||\Im(t)^c] \leq |\Im(t)^c|(1-\mho). \tag{2.51}$$

Now, observe that,

$$\begin{aligned}
\mathbb{E}[|\Im(t)^c|] &= \mathbb{E}[\mathbb{E}[|\Im(t)^c||\Im(t-1)^c]], && (2.52)\\
&\leq (1-\mho)\mathbb{E}[|\Im(t-1)^c|], && (2.53)\\
&\leq (1-\mho)^t\mathbb{E}[|\Im(0)^c|], && (2.54)\\
&\leq \exp(-\mho t)\frac{n}{2}. && (2.55)
\end{aligned}$$

For $t = \frac{\log n^2\epsilon^{-1}}{\mho}$, we see that,

$$\mathbb{P}[|\Im(t)^c| > 0] \leq \frac{\epsilon}{2n}. \tag{2.56}$$

Thus, single piece information dissemination with probability greater than $1 - \frac{\epsilon}{n}$ takes time $O(\frac{\log n + \log \epsilon^{-1}}{\mho})$. Thus, for L pieces of information, being shared in a round robin fashion, the time required for computation of minimums with probability greater than $1-\epsilon$ will be $O(\frac{L(\log n + \log L + \log \epsilon^{-1})}{\mho})$.

Let $X_1, X_2, \ldots, X_k$ be k i.i.d. exponential random variables with mean λ. Then, for any $\delta \in (0, \frac{1}{2})$, it can be shown [15] that,

$$\mathbb{P}\left(\left|\frac{1}{k}\sum_{i=1}^{k} X_i - \lambda\right| \geq \delta\lambda\right) \leq 2\exp\left(-\frac{k\delta^2}{3}\right). \tag{2.57}$$

Since we are using the symmetric matrix for gossiping, where $P_{ij} = \frac{1}{n}$ for all i, j, we have $\mho = O(1)$. Substituting these values for the given value of L yields the result.

2.C Proof of Lemma 2.3

We give a sketch of the proof (For detailed proof see [59]). Consider the Lyapunov function,

$$\mathcal{L}(Q(t)) = \sum_{i,f} (Q_i^f(t))^2. \tag{2.58}$$

Then, we can write the T-step Lyapunov drift as,

$$\Delta(T) = \mathbb{E}[\mathcal{L}(Q(t+T)) - \mathcal{L}(Q(t))|Q(t)], \tag{2.59}$$

$$= \sum_{\tau=0}^{T-1} \mathbb{E}[\mathcal{L}(Q(t+\tau)) - \mathcal{L}(Q(t+\tau-1))|Q(t)]. \tag{2.60}$$

Recall the queue evolution equation,

$$Q(t+1) = Q(t) + A(t) + R(t) - D(t), \tag{2.61}$$

and define $X(t) = D(t) - R(t)$. Observe that,

$$\mathbb{E}[\mathcal{L}(Q(t+\tau)) - \mathcal{L}(Q(t+\tau-1))|Q(t+\tau-1)] = \mathbb{E}[\sum_{i,f}(A_i^f(t) - X_i^f(t))^2 + 2Q_i^f(A_i^f(t) - X_i^f(t))]. \tag{2.62}$$

The first term can be bounded as follows. Assuming $\mathbb{E}[(A_i^f(t))^2] \le B_1$ and $\mu_{ij}(t) \le \sqrt{B_2}$, we obtain,

$$\mathbb{E}[\sum_{i,f}(A_i^f(t) - X_i^f(t))^2] \le 2\mathbb{E}[\sum_{i,f}(A_i^f(t))^2 + (X_i^f(t))^2] \le 2(B_1 + |\mathcal{V}|^2 B_2)(|\mathcal{V}||\mathcal{F}|) := C_1. \tag{2.63}$$

Using this in (2.60), and noting that the arrival process at time t is independent of the queue length at time t, we obtain,

$$\Delta(T) \le C_1 T + \sum_{\tau=0}^{T-1} 2\mathbb{E}[\langle Q(t+\tau-1), \lambda - X(t+\tau-1)\rangle|Q(t)]. \tag{2.64}$$

If $X^*(t)$ is the value of $X(t)$ corresponding to the optimal allocation, and we define,

$$\Upsilon(t) = \langle Q(t), X^*(t) - X(t)\rangle, \tag{2.65}$$

we can write,

$$\Delta(T) \leq C_1 T + 2\sum_{\tau=0}^{T-1} \mathbb{E}\left[\langle Q(t+\tau-1), \lambda - X^*(t+\tau-1)\rangle + \Upsilon(t+\tau-1) \middle| Q(t)\right]. \tag{2.66}$$

Define,

$$\tau_1 = \inf_{s\geq 0}\{\langle Q(t+s), X(t+s)\rangle \geq \alpha_1 \langle Q(t+s), X^*(t+s)\rangle\}, \tag{2.67}$$

$$\tau_2 = \inf_{s\geq 0}\{\mathcal{B}(t) \text{ fails at time } t+\tau_1+s\} - t. \tag{2.68}$$

For $\tau \leq \tau_1$ and $\tau \geq \tau_2$, we have,

$$\Upsilon(t+\tau) \leq \langle Q(t), X^*(t)\rangle + C_2. \tag{2.69}$$

Also, for $\tau_1 \leq \tau \leq \min\{T, \tau_2\}$, we have,

$$\Upsilon(t+\tau) \leq (1-(1-\alpha_1)(1-\alpha_2))\langle Q(t), X^*(t)\rangle - C_3. \tag{2.70}$$

Using the fact that $\mathbb{E}[\tau_1] \leq \frac{1}{\beta_1}$ and $\mathbb{E}[T - \min\{T, \tau_2\}] \leq \beta_2 T^2$, we obtain,

$$\sum_{\tau=0}^{T-1} \mathbb{E}[\Upsilon(t+\tau)|Q(t)] \leq T(1-(1-\alpha_1)(1-\alpha_2) + \frac{1}{\beta_1 T} + \beta_2 T)\langle Q(t), X^*(t)\rangle + C_4. \tag{2.71}$$

Let λ be in the interior of $\rho\Lambda$ for some $\rho > 0$. Then, there exist $0 < \epsilon < 1$ and variables $\tilde{X}$ such that, for all t,

$$\langle Q(t), \lambda + \mathbf{1}\epsilon\rangle \leq \rho\langle Q(t), \tilde{X}\rangle \leq \rho\langle Q(t), X^*(t)\rangle, \tag{2.72}$$

where $\mathbf{1}$ is the vector of all ones. Hence we obtain, for all t,

$$\langle Q(t), \lambda - X^*(t)\rangle \leq -\epsilon \sum_{i,f} Q_i^f(t) - (1-\rho)\langle Q(t), X^*(t)\rangle. \tag{2.73}$$

Combining (2.71), (2.73) and (2.66), and choosing $T = \sqrt{\frac{1}{\beta_1\beta_2}}$, we see that the T step Lyapunov drift is negative, for sufficiently large values of queue length, for any arrival rate vector $\lambda \in \rho\Lambda$ where $\rho < 1 - (\alpha_1 + (1-\alpha_1)\alpha_2) - 2\sqrt{\frac{\beta_2}{\beta_1}}$. This implies the result.

Chapter 3

A Distributed Draining Time Based Scheduling Algorithm with Graphical Interference Constraints

Using the notions of Draining Time and Discrete Review from the theory of fluid limits of queues, an algorithm that meets delay requirements to various flows in a network is constructed. The algorithm involves an optimization which is implemented in a cyclic distributed manner across nodes by using the technique of iterative gradient ascent, with minimal information exchange between nodes. The algorithm uses time varying weights to give priority to flows. The performance of the algorithm is studied in a network with interference modelled by independent sets. We modify the formulation to obtain an algorithm with similar performance, and is throughput optimal as well. The throughput optimality is demonstrated using fluid limits.

3.1 System Model

We consider a multihop network (see Fig. 3.1), given by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, 2, .., N\}$ is the set of vertices and $\mathcal{E}$, the set of links on V. We have directional links, with link (i, j) from node i to node j having a time varying channel gain $H_{ij}(t)$ at time t. The channel vector is $H(t)$ at time t, and it takes values from a finite set $\mathcal{H}$, with distribution γ. The set of flows is $\mathcal{F}$ and the arrival process is $A(t)$ as before. In this chapter we assume that the flows have fixed paths, from source to destination. The paths can be chosen in an efficient way using routing algorithms (see [1] for a survey). We will assume that the links are sorted into M *interference sets* $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_M$. At any time, only one link from an interference set can be active. A link may belong to multiple interference sets. In this work we will assume that any

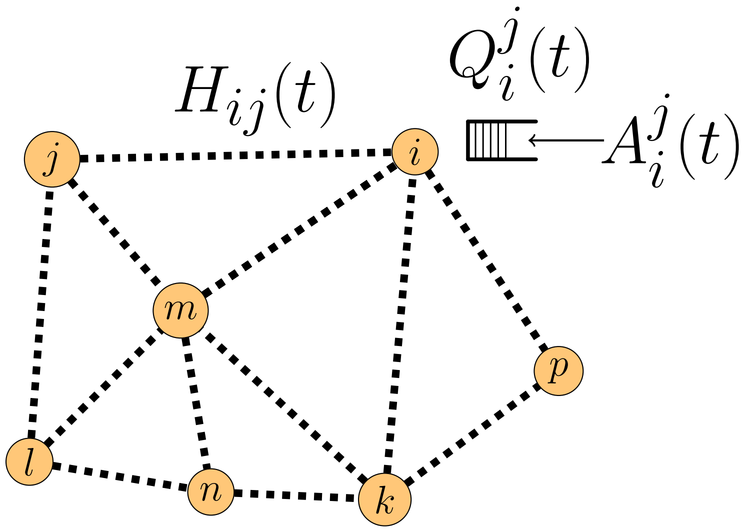

Figure 3.1: A simplified depiction of a Wireless Multihop Network

two links which share a common node will fall in the same interference set. The algorithm may be extended to a different interference model by an appropriate modification of the distributed projection step in Section 3.3.2. Note that this interference model is different from the one used in Chapter 2, where we had the SINR model, which was more general. Nevertheless we can approximate SINR type rates even with a graphical interference model, by choosing a suitable rate function that maps schedules to rates.

For a flow $f \in \mathcal{F}$, let $src(f)$ denote its source node, and $des(f)$ its destination.

Let $\mathcal{K}$ be the set of all link-flow pairs. A schedule s is a mapping $s : \mathcal{K} \to \{0, 1\}$. Let the set of all schedules be given by $\mathcal{S}$. For a channel state $h \in \mathcal{H}$ and a schedule $I \in \mathcal{S}$, we have a rate function,

$$\mu = \mu(H(t), I). \tag{3.1}$$

This will be some achievable rate function. Note that if two interfering links are present in a schedule s, $\mu_{ij}(h, s) = 0$ for all $h \in \mathcal{H}$ and all $(i, j) \in \mathcal{E}$. If none of the links interfere with each other, $\mu_{ij}^f(h, s) = f_{ij}(h)$, where f is some achievable rate function. Let $S_{ij}^f(t)$ denote the number of bits of flow f transmitted over link (i, j) at time t. Define,

$$R_i^f(t) = \sum_{k \neq i} S_{ki}^f(t),\ D_i^f(t) = \sum_{j \neq i} S_{ij}^f(t). \tag{3.2}$$

Let $R(t)$ and $D(t)$ denote the vectors $[R_i^f(t)]_{i \in \mathcal{V}, f \in \mathcal{F}}$ and $[D_i^f(t)]_{i \in \mathcal{V}, f \in \mathcal{F}}$ respectively. Then we

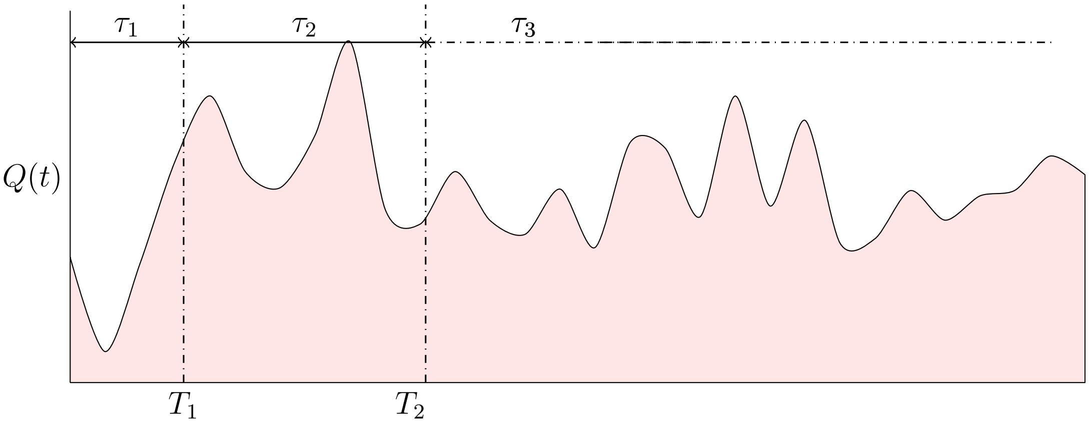

Figure 3.2: Review Times

have the queueing equation in vector notation,

$$Q(t+1) = Q(t) + A(t) + R(t) - D(t). \tag{3.3}$$

3.2 Discrete Review

While the system evolves in discrete time, $t \in \{0, 1, 2, ..\}$, making control decisions at all time slots may be expensive. We consider a system of Discrete Review (See [62], [65] for discussions). This involves an increasing sequence of times $0 \leq T_1 < T_2 \dots$ (see Fig. 3.2). At each T_i we make control decisions for the network, by solving an appropriate optimization problem. In the time frame $[T_i, T_{i+1})$, we will assume that the channel gains of different links are fixed (slow-fading), but drawn as an *i.i.d* sequence from a distribution γ on $\mathcal{H}$.

3.2.1 An Optimization based on Draining Time

What optimization do we perform at each review instant T_i? To formulate this problem, we will use some intuition from the idea of *draining time* for a fluid queue. Assume that we have a single queue evolving in continuous time. A flow arrives to the queue at a fixed rate c_1. This flow is served at a fixed rate of c_2. Then, the draining time, for a given initial condition, is defined to be the time for the queue to become empty. If the initial level of fluid in the queue was x, the fluid level at time t is given by (assuming $c_2 > c_1$),

$$q(t; x) = \max(x - (c_2 - c_1)t, 0).$$

The draining time in this case is given by $\tau = \dfrac{x}{c_2 - c_1}$ (see Figure 3.3). For a network of queues,

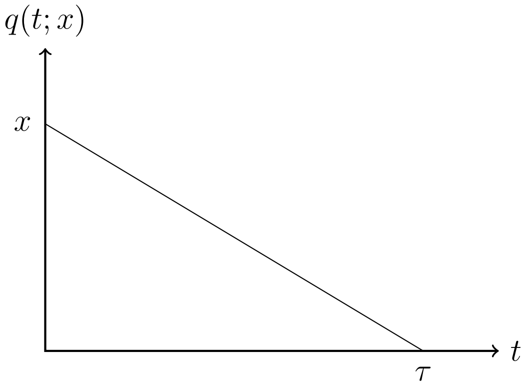

Figure 3.3: Draining Time

we can see that the rate of change of fluid in the queue can be written as,

$$\frac{d}{dt}q_i^f(t;x_i^f) = \lambda_i^f + \sum_{k\neq i}\zeta_{ki}^f(t)\mu_{ki} - \sum_{j\neq i}\zeta_{ij}^f(t)\mu_{ij}$$

where $q_i^f(t;x_i^f)$ denotes the amount of fluid in the queue at time t, starting from an initial condition x_i^f, $\frac{d}{dt}$ denotes the derivative, μ_{ij} denotes the rate available on link (i,j), and the variables ζ_{ij}^f are defined as the fraction of time flow f is scheduled on link (i,j). Clearly, they must satisfy,

$$\sum_f \zeta_{ij}^f(t) \leq 1 \ \forall (i,j) \in \mathcal{E}, \tag{3.4}$$

$$\zeta_{ij}^f(t) \geq 0 \ \forall (i,j) \in \mathcal{E}, f \in \mathcal{F}. \tag{3.5}$$

In unit time, the total amount of flow f passing through link (i,j) is less than or equal to $\zeta_{ij}^f\mu_{ij}$. Note that it is not always equal, because the queue may not have that many packets available to serve. Recall that in a review period, by our assumption, the service rate is fixed.

Draining time captures in some sense the delay associated with a flow. It is the time that an arrival at time $t = 0$ would have to wait before it gets served, assuming first-in-first-out service discipline. In a multihop network, the draining time will be given by the smallest τ that solves the equation

$$\sum_{j\neq i}\mu_{ij}\int_0^\tau \zeta_{ij}^f(t)dt - \sum_{k\neq i}\mu_{ki}\int_0^\tau \zeta_{ki}^f(t)dt - \lambda_i^f\tau = x_i^f.$$

Calculating this requires knowledge of arrival rates and the scheduling decisions of other nodes, and is not easy to obtain locally at a queue q_i^f. Therefore, we define a *pseudo draining time*, assuming further that $\zeta_{ij}^f(t) = \zeta_{ij}^f$,

$$D_i^f = \frac{x_i^f}{\sum_{j \neq i} \mu_{ij} \zeta_{ij}^f}.$$

This D_i^f is a lower bound to draining time. The draining time of the queue q_i^f would be D_i^f if the queue had no inflow, and was serving at constant rate $\sum_{j \neq i} \mu_{ij} \zeta_{ij}^f$. Consider the optimization

$$\max \sum_{i,f} \frac{w_i^f}{D_i^f},$$

where w_i^f is a weight corresponding to flow f on node i. Choosing $w_i^f = \theta^f (x_i^f)^2$, where θ^f is a positive constant, we obtain

$$\max \sum_{i,j,f} \theta^f x_i^f \zeta_{ij}^f \mu_{ij}, \tag{3.6}$$

$$s.t\ 0 \leq \zeta_{ij} := \sum_{f \in \mathcal{F}} \zeta_{ij}^f \leq 1\ \forall ij, \tag{3.7}$$

$$0 \leq \sum_{(i,j) \in I_m} \zeta_{ij} \leq 1,\ \forall m. \tag{3.8}$$

where the first constraint corresponds to the fact that only one flow can be scheduled across a link, and the second constraint corresponds to interference constraints on the links. This is also the standard weighted-rate maximization problem [97], with the weight given to rate μ_{ij} being $\sum_f \theta^f x_i^f \zeta_{ij}^f$.

3.2.2 Optimization at Review Times

At each T_i we make control decisions for the network, by solving the optimization problem (3.6)-(3.8), by choosing,

$$x_i^f = Q_i^f(T_i), \tag{3.9}$$

$$\mu_{ij} = \mu_{ij}(H(T_i), I_{ij}) \tag{3.10}$$

and I_{ij} corresponds to a schedule in which link (i,j) is on, and none of the links that interfere with it are on. We solve the fluid problem, and obtain scheduling variables corresponding to those fluid variables, at every review instant.

Each node transmits at a fixed power P. The rate function is chosen to be $\mu_{ij} = \log(1+\frac{H_{ij}P}{\sigma^2})$. Consider a packet of flow f which arrives at node i at the beginning of a review period. Such a packet observes a backlog of x_i^f in its queue. The total service allocated to flow f over link (i,j) in that period is $\zeta_{ij}^f(T_{i+1}-T_i)\mu_{ij}$ where $\zeta_{ij}^f \leq 1$. The times are chosen as

$$T_{i+1} - T_i = a_1 \log(1 + a_2 \sum_{i,f} Q_i^f(T_i)), \tag{3.11}$$

where a_1, a_2 are positive constants.

3.2.3 Providing Quality-of-Service

We will be solving the optimization problem defined by equations (3.6)-(3.8) at every discrete review instant. In order to incorporate QoS constraints, we will let θ^f vary dynamically. Let flow f_1 require its mean delay to be less than or equal to d_1. At the destination node of f_1, we estimate empirically its mean delay in the last review period. If it is greater than d_1, we set $\theta^{f_1} = \hat{\theta} > 1$; otherwise, $\theta^{f_1} = 1$. Thus the control variables corresponding to the flows that require QoS obtain higher weight in the optimization problem, if its QoS requirement was not met during the previous review period. For a hard deadline guarantee flow, we have two parameters, the hard deadline and the drop ratio, which is the percentage of packets we are willing to allow with delays larger than the hard deadline. At every review instant, at the destination of that flow, we check whether the percentage of packets that have arrived with delays larger than the deadline, exceeds the drop ratio. If yes, we set $\theta^f = \hat{\theta}$. In the next section we will provide a distributed algorithm for the optimization problem defined by equations (3.6)-(3.8).

3.3 Distributed Optimization

Let $\mathcal{K}$ be the set of all *link-flow* pairs $((i,j), f)$. For any $k \in \mathcal{K}$, there exists a link $(i(k), j(k))$ and a flow $f(k)$. A *schedule* is a vector $\mathbf{s}$ of length $|\mathcal{K}|$, with each element $\mathbf{s}(k)$ corresponding to the fraction of time link $(i(k), j(k))$ transmits flow $f(k)$. The *feasible set* $\mathcal{S}$ is the set of schedules that satisfy constraints (3.7) and (3.8); however, we remove the positivity constraint. Note that this changes the search space, but does not change the optimal value or the optimal point, since the quantity being maximized is a weighted sum of ζ_{ij}^c with positive weights. The set $\mathcal{S}$ will be a convex polytope, since it is generated by linear inequalities, and will be a closed subset of $\mathbb{R}^{|\mathcal{K}|}$.

We can rewrite equations (3.6) through (3.8) as

$$\max_{\mathbf{s}\in\mathcal{S}} \sum_{k\in\mathcal{K}} f_k(\mathbf{s}) \tag{3.12}$$

where,

$$f_k(\mathbf{s}) = w_k \mu_k \mathbf{s}(k),\ w_k = \theta^{c(k)} x_{i(k)}^{c(k)},\ \mu_k = \mu_{i(k)j(k)},\ \mathbf{s}(k) = \zeta_{i(k)j(k)}^{c(k)}. \tag{3.13}$$

3.3.1 Incremental Gradient Ascent

In order to optimize (3.12), we will use the incremental gradient method [8]. This involves the iteration

$$\mathbf{s}_{j+1} = \Pi_{\mathcal{S}}(\mathbf{s}_j + \alpha_j \nabla f_{k_j}(\mathbf{s}_j)), \tag{3.14}$$

with $k_j = j$ modulo $|\mathcal{K}|+1$, and $\Pi_{\mathcal{S}}$ denotes projection onto the set $\mathcal{S}$. Let $\mathbf{v}(r)$ denote a vector which is one only at its rth index and zero elsewhere. We can write

$$\nabla f_{k_j}(\mathbf{s}_j) = w_{k_j} \mu_{k_j} \mathbf{v}(k_j).$$

Hence we may rewrite equation (3.14) as

$$\mathbf{s}_{j+1} = \Pi_{\mathcal{S}}(\mathbf{s}_j + \alpha_j w_{k_j} \mu_{k_j} \mathbf{v}(k_j)). \tag{3.15}$$

3.3.2 Projection

Since interference exists between two links that share a node, an update of the optimization variables at a link affects those links which share a node with it. The constraint set $\mathcal{S}$ is defined by the intersection of half-spaces $\{\mathcal{H}_i\}_{i=1}^M$, where

$$\mathcal{H}_i = \{\mathbf{s} : \langle \mathbf{s}, \boldsymbol{\nu}^i \rangle \le \beta_i\},$$

where $\boldsymbol{\nu}^i$ is the unit normal vector to the plane, with $||\boldsymbol{\nu}^i||_2 = 1$. For example, the interference constraint

$$s_1 + s_2 + s_4 \le 1, \tag{3.16}$$

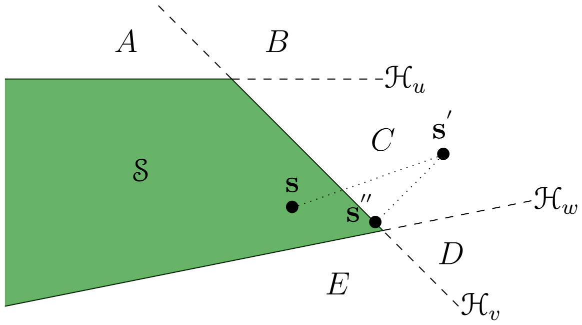

Figure 3.4: Single Step Projection

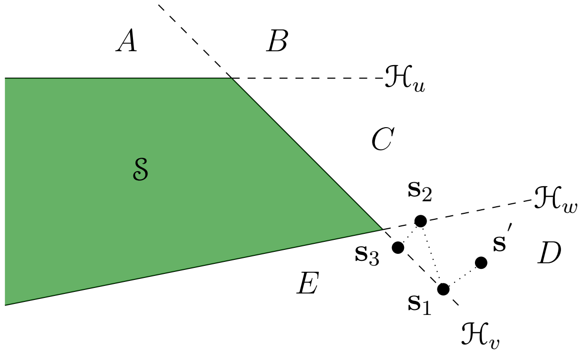

Figure 3.5: Multi Step Projection

can be represented by,

$$\langle \mathbf{s}, \boldsymbol{\nu}^j \rangle \leq \beta_j, \tag{3.17}$$

where

$$\boldsymbol{\nu}^j = \frac{1}{\sqrt{3}} \sum_{n=1,2,4} \mathbf{v}(n), \tag{3.18}$$

and $\beta_j = \frac{1}{\sqrt{3}}$. Due to the nature of our constraints, $\boldsymbol{\nu}^i$ will be non-negative. Each half-space corresponds to one constraint.

In the increment step, we update one component $\mathbf{s}(k)$ of $\mathbf{s}$, corresponding to a link flow pair $(i(k), j(k)), f(k)$. There are two half-space constraints, $\mathcal{H}_v$ and $\mathcal{H}_w$, corresponding to links connected to $i(k)$ and $j(k)$. If the point after update violates both constraints, projection is done repeatedly, first on $\mathcal{H}_v$ and then on $\mathcal{H}_w$, and so on. It can be shown [96, Theorem 13.7] that this iterative process converges to the projection of the point onto $\mathcal{H}_v \cap \mathcal{H}_w$. If a single hyperplane is violated, one step of projection suffices.

We will now obtain the analytical expressions for projecting a point onto a hyperplane. Let $\mathcal{H}_v$ be defined by

$$\langle \mathbf{s}, \boldsymbol{\nu}^v \rangle \leq \beta_v.$$

Let the point $\mathbf{s}^{'}$ be such that $\beta_v^* \triangleq \langle \mathbf{s}^{'}, \boldsymbol{\nu}^v \rangle > \beta_v$. Hence it lies outside $\mathcal{S}$. Let us define

$$\mathbf{s}^{''} = \mathbf{s}^{'} - (\beta_v^* - \beta_v)\boldsymbol{\nu}^v. \tag{3.19}$$

Observe that $\mathbf{s}^{''}$ is the orthogonal projection of $\mathbf{s}^{'}$ onto $\mathcal{H}_v$, since $\langle \mathbf{s}^{''}, \boldsymbol{\nu}^v \rangle = \beta_v$. Since $\mathbf{s}^{'} - \mathbf{s}^{''} = (\beta_v^* - \beta_v)\boldsymbol{\nu}^v$, and $\boldsymbol{\nu}^v$ is normal to the plane boundary of $\mathcal{H}_v$, the projection step projects the point perpendicularly onto $\mathcal{H}_v$. We show below that the projection does not break any additional constraints.

Proposition 3.1 *If* $\langle \boldsymbol{s}^{'}, \boldsymbol{\nu}^w \rangle \leq \beta_w$, *then* $\langle \boldsymbol{s}^{''}, \boldsymbol{\nu}^w \rangle \leq \beta_w$.

Proof:

$$\langle \mathbf{s}^{''}, \boldsymbol{\nu}^w \rangle = \langle \mathbf{s}^{'}, \boldsymbol{\nu}^w \rangle - (\beta_v^* - \beta_v)\langle \boldsymbol{\nu}^v, \boldsymbol{\nu}^w \rangle.$$

Since $\boldsymbol{\nu}^w$ and $\boldsymbol{\nu}^v$ are non-negative, and $\beta_v^* > \beta_v$, we have $(\beta_v^* - \beta_v)\langle \boldsymbol{\nu}^v, \boldsymbol{\nu}^w \rangle \geq 0$, and consequently, $\langle \mathbf{s}^{''}, \boldsymbol{\nu}^w \rangle \leq \beta_w$. □

Hence, if a point breaks exactly one hyperplane constraint, the projection step projects the point back on $\mathcal{S}$.

Consider an example. If the interference constraint is,

$$s_1 + s_2 + s_4 \leq 1, \tag{3.20}$$

the projection step (3.19) is equivalent to,

$$s_1 = s_1 - \frac{s-1}{3}, \; s_2 = s_2 - \frac{s-1}{3}, \; s_4 = s_4 - \frac{s-1}{3}, \tag{3.21}$$

where,

$$s = s_1 + s_2 + s_4. \tag{3.22}$$

The above discussion may be modified for other interference scenarios by noting that in such scenarios, the update step may break more than two interference constraints at once. In that case, we must first find out the closest hyperplane, or intersection of hyperplanes, and then do the projection.

3.3.3 Convergence

Let us define

$$f(\mathbf{s}) := \sum_{k \in \mathcal{K}} f_k(\mathbf{s}), \;\; f^* := \max_{\mathbf{s} \in \mathcal{S}} \sum_{k \in \mathcal{K}} f_k(\mathbf{s}).$$

We have the following theorem for the convergence of the distributed algorithm.

Theorem 3.1 *If* $\max_{i,j,c} \theta^c x_i^c \mu_{ij} \leq C_2$*, the algorithm defined by equation (3.15) results in a sequence of points* $\{\boldsymbol{s}_n\}$ *such that*

$$\lim_{j \to \infty} \sup f(\boldsymbol{s}_j) \geq f^* - C_3,$$

where $C_3 = \frac{\alpha \beta |\mathcal{K}|^2 C_2^2}{2}$ *with* $\beta = 4 + \frac{1}{|\mathcal{K}|}$*.*

Proof: See [8]. □

The time taken by the optimization to converge ε close to the optimum is of the order of $\frac{1}{\varepsilon}$. It is also inversely proportional to the step size α. We describe the algorithm below.

3.3.4 Algorithm Description

The algorithm proceeds in review cycles. At every slot t that is the beginning of a review cycle, the nodes calculate the number of slots till the next review slot by

$$T_{rev} = t + a_1 \log(1 + a_2 \sum_{i,f} Q_i^f(t)),$$

where a_1 and a_2 are constants. At the beginning of a review cycle, the nodes calculate the variables ζ_{ij}^c for all i, j and f, and use these till the end of the review cycle. We will now describe how the ζ_{ij}^c variables are calculated at each node.

The vector $\mathbf{s}$ is initialized to all ones. The calculation proceeds cyclically. The node which has the flow corresponding to the first component of the vector $\mathbf{s}$ will do the update

$$\mathbf{s}(1) = \mathbf{s}(1) + \alpha w(1) \mu(1). \tag{3.23}$$

Here $w(1) = \theta^{c(1)} x_{i(1)}^{c(1)}$, with $\theta = 1$ if the QoS constraint of flow $c(1)$ was satisfied in the previous review cycle; otherwise, it is set to be equal to a value $\hat{\theta}$. The node then calculates the inner products

$$\beta_1^* \triangleq \langle \mathbf{s}, \boldsymbol{\nu}^1 \rangle, \beta_2^* \triangleq \langle \mathbf{s}, \boldsymbol{\nu}^2 \rangle$$

where $\boldsymbol{\nu}^l, \boldsymbol{\nu}^2$, correspond to the two interference constraints that the update step may break. If one of these constraints is broken, the update can be projected back in a single step. If both are violated, we will have to go for the iterative projection method. For projection on a plane characterized by $\langle \mathbf{s}, \boldsymbol{\nu}^i \rangle = \beta_i$, the node calculates $\beta_{ex} = \frac{\beta_i^* - \beta_i}{N_v}$ where N_v is the number of links in that interference set. The node communicates this value to all links in its interference set. All these nodes, as well as the current node, update their values as

$$\mathbf{s}(k) = \mathbf{s}(k) - \beta_{ex}.$$

This is the projection step. Once the required number of projections is over, the node then passes its $s(1)$ to the node which has the next component of the vector $\mathbf{s}$, and that node updates its value of $\mathbf{s}(1)$. The next node now repeats the update and projection steps, and passes its update to its neighbour. This process is repeated cyclically, i.e, we repeat step (3.23) with 1 replaced by 2, and then by 3 and so on, across the nodes till a predetermined stopping time is reached. At the end of the stopping time, we set all the negative components of $\mathbf{s}$ to zero. For each interference set I, we check its constraint

$$\langle \mathbf{s}, \boldsymbol{\nu} \rangle \leq \beta.$$

If not, we apply the update

$$\mathbf{s}(k) = \frac{\mathbf{s}(k)}{\langle \mathbf{s}, \boldsymbol{\nu} \rangle}, \ k \in I.$$

This will ensure compliance with the constraints. The complete algorithm is given below, as Algorithm 3, which uses in turn, Algorithms 4, 5 and 6. The last algorithm creates the schedule by scheduling flows on a link for a fraction of time equal to the corresponding $\mathbf{s}(k)$.

Algorithm 3 Algorithm Q-Flo

```
$T_{rev} = 0$, $T_{prev} = 0$.
while $t \geq 0$ do
    if $t = T_{rev}$ then
        obtain variables $s_{ij}^{f}(T_{rev})$ using Algorithm 4
        $T_{prev} \leftarrow T_{rev}$
        $T_{rev} \leftarrow T_{rev} + a_1 \log(1 + a_2 \sum_{i,c} Q_i^f(T_{rev}))$
        Create $sched(i, j, f, t)$ from $t = T_{prev}$ to $t = T_{rev} - 1$ using Algorithm 6
    end if
    for all $i, j, f$ do
        if $Q_i^f(t) > 0$ and $sched(i, j, f, t) = 1$ then  schedule flow $c$ across link $(i, j)$
        end if
    end for
end while
```

Algorithm 4 Algorithm at node level

```
Stopping time $T_s$, $t^{'} = 0$, $s_{ij}^{f}(T_{rev}) = 0$ for all $i, j, f$
while $t^{'} < T_s$ do
    $k = t^{'}|\mathcal{K}| + 1$, $(i, j, c) \leftarrow (i(k), j(k), c(k))$
    If QoS criterion of $c$ satisfied, $\theta^c \leftarrow 2$; else $\theta^c \leftarrow 1$
    $w \leftarrow \theta^c Q_i^c(T_{rev})$, $\mu(k) \leftarrow \mu_{ij}$, $s_{ij}^c \leftarrow s_{ij}^c + \alpha w \mu(k)$
    Project $s_{ij}^c \leftarrow \Pi_{\mathcal{S}}(s_{ij}^c)$ using Algorithm 5
    $t^{'} \leftarrow t^{'} + 1$
end while
$s_{ij}^c \leftarrow \max(s_{ij}^c, 0)$
If $s := \sum_{j,c} s_{ij}^c + \sum_{j,c} s_{ji}^c > 1$, $s_{ij}^c \leftarrow \frac{s_{ij}^c}{s}$
$s_{ij}^c(T_{rev}) \leftarrow s_{ij}^c$
```

Algorithm 5 Algorithm for Projection

```
1: Link interference constraints $\langle \mathbf{s}, \boldsymbol{\nu}^1 \rangle \leq \beta_1, \langle \mathbf{s}, \boldsymbol{\nu}^2 \rangle \leq \beta_2$
2: Calculate $\beta_1^* \triangleq \langle \mathbf{s}, \boldsymbol{\nu}^1 \rangle, \beta_2^* \triangleq \langle \mathbf{s}, \boldsymbol{\nu}^2 \rangle$
3: if $\beta_i^* > \beta_i$ then and $\beta_j^* < \beta_j$
4:     $\beta_{ex} = \frac{\beta_i^* - \beta_i}{N_i + 1}$, $N_i$ = number of interferers.
5:     For all interferers and current link, update $\beta_{ij}^c - \beta_{ex}$.
6: end if
7: if $\beta_1^* > \beta_1$ and $\beta_2^* > \beta_2$ then
8:     Repeat steps 4 to 6 and 8 to 11 $N\_rep$ times
9: end if
```

Algorithm 6 Algorithm for Schedule Creation

```
1: Initialize $sched(i,j,f,t) = 0 \; \forall i,j,f,t$
2: for $k \in \{1, \ldots, |V|\}$ do
3:     Obtain $sched(i,j,c,t)$ for $i \leq k-1$
4:     Obtain $s_{kj}^c(T_{rev})$ for all $j, c$
5:     Set of links that interfere with node $k =: N_k$
6:     for $j \in N_k, c \in F, t \in [T_{prev}, T_{rev}]$ do
7:         if $\sum_{i \leq k-1} sched(i,j,c,t) = 0$ and $\sum_{i \in N_j} sched(j,i,c,t) = 0$ and
           $\sum_{t^o = T_{prev}}^{t} sched(k,j,c,t^o) < s_{kj}^c(T_{rev} - T_{prev})$ then
8:             $sched(k,j,c,t) = 1$
9:         end if
10:    end for
11: end for
```

3.4 Simulation Results

We consider a 10 node network, with connectivity as depicted in Fig. 3.6, on a unit area, and Rayleigh distributed channel gains with parameters proportional to the inverse of the square of the distance between the nodes. The source-destination pairs are from node 0 to node 9, node 1 to node 7, node 5 to node 7, node 2 to node 8 and node 4 to node 9 with fixed routes being $0 \to 1 \to 3 \to 7 \to 9$, $0 \to 4 \to 9$ and $0 \to 2 \to 6 \to 8 \to 9$ for the first flow, $1 \to 3 \to 7$ for the second, $5 \to 7$ for the third, $2 \to 6 \to 8$ for the fourth and $4 \to 9$ for the last. A packet is of size one bit. Nodes transmit with unit power. We first study the sensitivity of the algorithm to the number of iterations of the distributed algorithm. We fix $\alpha = 0.0001$, and the arrival process is Poisson with rate 3.3 corresponding to the flows from nodes 0 to 9, 1 to 7, 2 to 8,

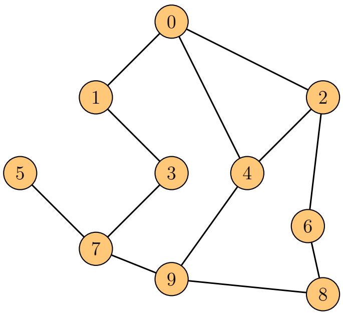

Figure 3.6: Sample Network

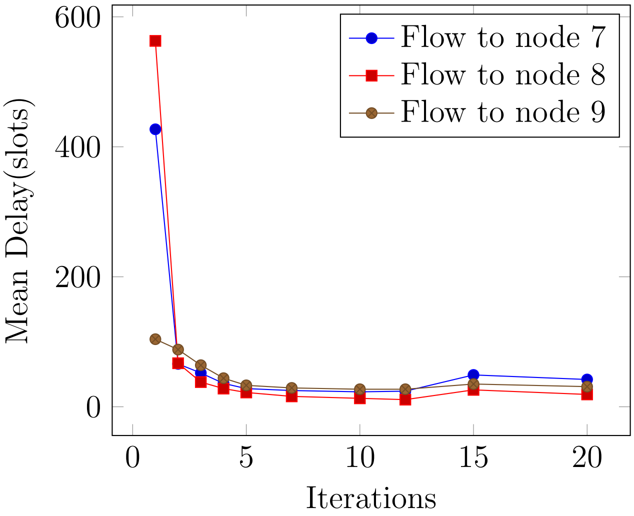

Figure 3.7: Number of Iterations versus Mean Delay

4 to 9, and 5 to 7 respectively. The simulation runs for 10^5 slots. The constants a_1 and a_2 in Algorithm 1 are set to 1.

In Fig. 3.7 we plot the sensitivity of mean delay of three flows in the network to the number of iterations of the distributed algorithm. One iteration is equivalent to the completion of the update and project step at all the nodes. By Little's Law, since mean delay is directly proportional to mean queue length, it is evident from Fig 3.7 that as the number of distributed iterations increases, the system has a lower mean queue length. This can be attributed to the fact that as the number of iterations of the distributed optimization increase, we come closer to the actual optimal value of the control parameters. From the simulations, around 5 rounds of iterations seem to be sufficient, and there is no major improvement in mean delay after that. There is a marginal increase in the delay when the iterations increase to around 15. This is probably owing to the error accumulation as a result of the finite truncation of the iterative steps. Another parameter of interest is the number of rounds of iterative projection, N_rep. From simulations, it seems that 2 to 4 rounds are sufficient, since average delays (and consequently average queue lengths) seem to stabilize after these many rounds of iterations.

We consider the case where we are trying to provide end-to-end mean delay guarantees to two flows: those destined to nodes 7 and 8 (Table 3.1), with flow 9 receiving no delay guarantee. The arrival rate is 3.3 packets/slot for all arrivals. We study two cases, with $\hat{\theta}$ equal to 6 and 7. Using a higher weight $\hat{\theta}$, we are able to give tighter delay guarantees. Also, we see that as the delay constraint becomes tighter, the delay of the non QoS flow decreases. This is because while a given priority weight θ^f reserves resources for a QoS flow, if the delay required is smaller, the flow will have a smaller mean queue length, which will result in higher weight being given to non QoS flows in review periods where the delay criterion is satisfied, since the optimization function (3.6) is proportional to the queue length. Consequently, giving higher weights $\hat{\theta}$ to QoS flows does not negatively impact the non QoS flows as would have been expected. Here $T_s = 8$ and $N\_rep = 10$.

In Table 3.2, we demonstrate how to provide hard delay guarantee for flow 7 and mean delay guarantee for flow 8. Flow 9 receives no delay guarantee. The weights $\hat{\theta}$ for flows 7 and 8 are 2 and 1.5. Note that these weights are lower than those used in Table 3.1, and hence, the reduction possible in the mean delay of flow 8 is lower in this case. For flow 7, the packet is dropped at the destination if its deadline is not met. We have set a target of 2% for such packets. We see that packets of flow 7 meet this target for the different deadlines fixed. The mean delay requirements of flow 8 are also met.

Table 3.1: Two Flows with mean delay requirement

Mean Delay(slots)							
Flow to node 7			Flow to node 8			Flow to node 9	
Target Mean Delay	Achieved		Target Mean Delay	Achieved		Delay with $\hat{\theta} = 6$	Delay with $\hat{\theta} = 7$
	with $\hat{\theta} = 6$	with $\hat{\theta} = 7$		with $\hat{\theta} = 6$	with $\hat{\theta} = 7$		
50	51	51	30	32	33	318	275
40	40	40	25	26	28	253	196
30	32	30	20	22	21	172	165
25	30	26	15	18	15	145	147

Table 3.2: One mean delay and one hard deadline

Flow to node 7		Flow to node 8		Flow to node 9
Hard Delay Target(slots), Drop Ratio Target	Drop Ratio Achieved	Mean Delay Target (slots)	Mean Delay Achieved (slots)	Mean Delay (slots)
180,2%	2%	50	51	136
180,2%	2%	40	43	100
180,2%	2%	35	36	89
160,2%	2%	45	45	88
140,2%	2%	30	33	91
120,2%	2%	35	37	94

3.5 Throughput Optimal Algorithm

While the algorithm proposed in (3.6)-(3.8) has good performance in terms of mean delay and hard deadline QoS, it does not seem possible to show that it is throughput optimal. Hence, we propose another algorithm, closely related to the optimization (3.6)-(3.8). We call it Queue Weighted Discrete Review (QWDR). Consider the following optimization to be solved at the beginning of every review period, T_i.

$$\max \sum_{i,j,f} \alpha(Q^f(T_i), \overline{Q}^f) Q^f_{ij}(T_i) \zeta^f_{ij} \mu_{ij}, \tag{3.24}$$

$$s.t \; 0 \leq \zeta_{ij} := \sum_{f \in F} \zeta^f_{ij} \leq 1 \; \forall ij, \tag{3.25}$$

$$0 \leq \zeta_{ij} + \zeta_{kl} \leq 1, \;\; \forall (i,j), (k,l) \in I_m, \forall m, \tag{3.26}$$

where $Q^f_{ij} = \max(Q^f_i - Q^f_j, 0), Q^f(t) = \sum_i Q^f_i(t)$. This optimization is done assuming $Q^f_{ij} > 0$ for at least one link flow pair $(i, j), f$. If all Q^f_{ij} are zero, we define the solution to be $\zeta^f_{ij} = 0$ for all i, j, f. The first constraint corresponds to the fact that flows cannot simultaneously be scheduled on a link, and the second constraint corresponds to interference constraints. In (3.24), we optimize the sum of rates weighted by the function α as well as the queue lengths. More weight may be given to flows with larger backlogs, while the α function captures the delay requirement of the flow. These are chosen such that flows requiring a lower mean delay would have a higher weight compared to flows needing a higher mean delay. Also, flows whose mean delay requirements are not met should get priority over flows whose requirements have been met. The weights α therefore are functions of the state, and $\overline{Q}^f$ denotes a desired value for the queue length of flow f. We use the function

$$\alpha(x, \overline{x}) = 1 + \frac{a_1}{1 + \exp(-a_2(x - \overline{x}))}. \tag{3.27}$$

Thus α is close to $1 + a_1$ when x is larger than $\overline{x}$, and reduces to 1 as x reduces. Thus, delays which are above certain thresholds obtain higher weights in the optimization function. We seek to regulate the queue lengths using α with a careful selection of $\overline{Q}^f$, and thereby control the delays. For any flow, the $\overline{Q}^f$ are chosen in the following manner. If the required end-to-end mean delay of the flow with arrival rate λ is $\overline{D}$, we choose $\overline{Q}^f = \lambda\overline{D}$. In some sense, we are taking the queue length equivalent to the required delay using Little's Law and using it as a threshold that determines the scheduling process. Note that we may also suppress the $\bar{x}$ for convenience, and write $\alpha(x, \bar{x})$ as $\alpha(x)$, where necessary.

The proposed optimization differs from (3.6) in that we have replaced Q^f_i by $(Q^f_i - Q^f_j)^+$. Further, instead of the discontinuous function θ, we have a continuous function α. Note that the distributed implementation does not change in implementation, but only in function value.

3.5.1 An Alternate Representation

We will now rewrite the optimization (3.24)-(3.26) in a different manner to simplify the exposition. Let us consider all non negative vectors $(\hat{\mu}^f_{ij})_{(i,j)\in\mathcal{E}, f\in\mathcal{F}}$ which satisfy,

$$\sum_{f\in\mathcal{F}} \hat{\mu}^f_{ij} \le \mu_{ij}, \ \forall (i, j) \in \mathcal{E}. \tag{3.28}$$

This represents a feasible allocation to different flows across the link (i, j). Observe that this new function $\hat{\mu}$ is a mapping,

$$\hat{\mu}_{ij}^f = \hat{\mu}_{ij}^f(H(t), I), \tag{3.29}$$

where $H(t)$ is the current channel state, and I is a feasible schedule belonging to $\mathcal{S}$. Using this notation, the optimization (3.24)-(3.26) may be rewritten as,

$$\max_{I \in \mathcal{S}} \sum_{i,j,f} \alpha(Q^f(T_i), \overline{Q}^f) Q_{ij}^f(T_i) \hat{\mu}_{ij}^f(H(T_i), I). \tag{3.30}$$

3.6 Capacity Region and Rate Region

The capacity region for this network model is defined similar to what was done in the previous chapter, but with one modification. We define the rate vector at time t as,

$$\mu(t) = \mu(H(t), I), \tag{3.31}$$

where $H(t)$ is the channel state at time t, and I is a schedule. Hence, we define,

$$\mathcal{M}_h = \{\mu(h, I) : i \in \mathcal{S}\}, \tag{3.32}$$

where $\mathcal{S}$ is the set of all feasible schedules. Let $\overline{\mathcal{M}_h}$ denote the convex hull of $\mathcal{M}_h$. Then we define,

$$\mathcal{M} = \sum_{h \in \mathcal{H}} \gamma_h \overline{\mathcal{M}}_h, \tag{3.33}$$

and define Λ as follows.

Definition 3.1 *The capacity region, Λ, is the set of all arrival rate vectors λ for which there exists a vector $\varpi = [\varpi_{ij}^f]_{(i,j) \in \mathcal{E}, f \in \mathcal{F}}$ which satisfies,*

$$\varpi_{ij}^f \geq 0, \ \forall i, j, f \tag{3.34}$$

$$\varpi_{ii}^f = 0, \ \forall i, f, \tag{3.35}$$

$$\varpi_{ij}^i = 0, \ \forall i, j, f, \tag{3.36}$$

$$\lambda_i^f \leq \sum_j \varpi_{ij}^f - \sum_k \varpi_{ki}^f, \ \forall i, f, \tag{3.37}$$

$$\sum_f \varpi_{ij}^f \le m_{ij}, \ \textit{for some } m \in \mathcal{M}. \tag{3.38}$$

We define the *rate region* $\mathcal{W}$ as follows.

$$\mathcal{W} = \{[\varpi_{ij}^f]_{(i,j)\in\mathcal{E}, f\in\mathcal{F}} = \varpi : \exists m \in \mathcal{M} \ s.t. \ \sum_f \varpi_{ij}^f \le m_{ij}, \ \forall i, j\}. \tag{3.39}$$

The rate region $\mathcal{W}$ is the set of all feasible rate vectors, i.e, all rate vectors ϖ for which there exists a schedule that can support it.

In the next section we show that the present algorithm is throughput optimal in the sense that if there is any other algorithm that will stabilise the network for given traffic and channel statistics, then this algorithm will. In literature there are some algorithms available, e.g., back pressure [71], which are throughput optimal. But, often the delays associated with these algorithms are high and they do not ensure any QoS. We will show that we are able to provide the required end-to-end mean delays and hard deadline constraints with the present algorithm. However, the present algorithm has the limitation that the routing has already been fixed via some other algorithm while back pressure has routing as part of the algorithm.

To show that our algorithm is throughput optimal, we first show that the scaled process for our system converges to a fluid limit.

3.7 Fluid Limit

To obtain the fluid limit, we first need to define a few processes. Define,

$$\hat{A}_i^f(t) = \sum_{\tau=1}^{t} A_i^f(\tau), \ \forall i, f, t. \tag{3.40}$$

This is the cumulative number of packets of flow f that have arrived exogenously at node i. Denote the vector $[\hat{A}_i^f(t)]_{i\in\mathcal{V}, f\in\mathcal{F}}$ by $\hat{A}(t)$. Let $E_h(t)$ denote the number of slots till time t that the channel state was $h \in \mathcal{H}$. The vector $[E_h(t)]_{h\in\mathcal{H}}$ will be denoted by $E_h(t)$. Recall that $\hat{\mu} = [\hat{\mu}_{ij}]_{(i,j)\in\mathcal{E}, f\in\mathcal{F}}$ denotes an allocation of the rate vector $\mu(h, I)$ for $h \in \mathcal{H}, I \in \mathcal{S}$. Let $G_{\hat{\mu}}^{hI}(t)$ denote the cumulative number of slots till time t when channel state was h, the schedule chosen was I and the rate allocation was the vector $\hat{\mu}$. It will be assumed that the possible allocations $\hat{\mu}$ forms a finite set. It follows that,

$$\sum_{\hat{\mu}, I} G_{\hat{\mu}}^{hI}(t) = E_h(t). \tag{3.41}$$

Let $\hat{S}_{ij}^f(t)$, $\hat{R}_i^f(t)$ and $\hat{D}_i^f(t)$ be defined as,

$$\hat{S}_{ij}^f(t) = \sum_{\tau=1}^{t} S_{ij}^f(\tau), \tag{3.42}$$

$$\hat{R}_i^f(t) = \sum_{\tau=1}^{t} R_i^f(\tau), \tag{3.43}$$

$$\hat{D}_i^f(t) = \sum_{\tau=1}^{t} D_i^f(\tau), \tag{3.44}$$

and hence,

$$\hat{R}_i^f(t) = \sum_k \hat{S}_{ki}^f(t), \;\; \hat{D}_i^f(t) = \sum_j \hat{S}_{ij}^f(t). \tag{3.45}$$

Using this notation, the queueing equation can be written as,

$$Q(t) = Q(0) + \hat{A}(t) + \hat{R}(t) - \hat{D}(t). \tag{3.46}$$

Define the system state to be

$$Y(t) = (Q(t), \tilde{Q}(t), \tilde{S}(t)),$$

with the process $\tilde{Q}(t) = Q(T)$ with $T = \sup\{s \leq t : s = T_i \text{ for some } i\}$, representing the queue values at the last review instant, and $\tilde{S}_{ij}^f(t) = \hat{S}_{ij}^f(t) - \hat{S}_{ij}^f(T)$ representing the cumulative allocation vector from the last review instant to the current time. From the queue evolution (3.3) and the allocation, it is clear that the system $Y(t)$ evolves as a discrete time countable state Markov chain, since at any time t the next state may be computed by solving the optimization (3.24) with Q replaced by $\tilde{Q}$, and using the cumulative allocation process $\tilde{S}$ to determine how allocation must be done in the next slot to satisfy the solution of (3.24). The associated norm is $||Y(t)|| = \sum_{i,f}(Q_i^f + \tilde{Q}_i^f) + \sum_{i,j,f} \tilde{S}_{ij}^f$. Positive recurrence of this Markov chain would imply stability. We will show the positive recurrence of this Markov process via its fluid limit.

Define the process $Z(t)$ as,

$$Z(t) = (\hat{A}(t), E(t), G(t), \hat{D}(t), \hat{R}(t), \hat{S}(t), \tilde{S}(t), Q(t), \tilde{Q}(t)). \tag{3.47}$$

Let $Z = \{Z(t), t \geq 0\}$ and $Y = \{Y(t), t \geq 0\}$. The process Y is a projection of Z. For the components of the process $Z(t)$, define the corresponding scaled (continuous time) processes indexed by n, for $t \geq 0$,

$$a^n(t) = \frac{\hat{A}(\lfloor nt \rfloor)}{n}, \tag{3.48}$$

$$e^n(t) = \frac{E(\lfloor nt \rfloor)}{n}, \tag{3.49}$$

$$g^n(t) = \frac{G(\lfloor nt \rfloor)}{n}, \tag{3.50}$$

$$d^n(t) = \frac{\hat{D}(\lfloor nt \rfloor)}{n}, \tag{3.51}$$

$$r^n(t) = \frac{\hat{R}(\lfloor nt \rfloor)}{n}, \tag{3.52}$$

$$s^n(t) = \frac{\hat{S}(\lfloor nt \rfloor)}{n}, \tag{3.53}$$

$$\tilde{s}^n(t) = \frac{\tilde{S}(\lfloor nt \rfloor)}{n}, \tag{3.54}$$

$$q^n(t) = \frac{Q(\lfloor nt \rfloor)}{n}, \tag{3.55}$$

$$\tilde{q}^n(t) = \frac{\tilde{Q}(\lfloor nt \rfloor)}{n}. \tag{3.56}$$

Thus we obtain the process,

$$z^n(t) = (a^n(t), e^n(t), g^n(t), d^n(t), r^n(t), s^n(t), \tilde{s}^n(t), q^n(t), \tilde{q}^n(t)). \tag{3.57}$$

Let z^n denote the process $\{z^n(t), t \geq 0\}$. Note that,

$$z^n = (a^n, e^n, g^n, d^n, r^n, s^n, \tilde{s}^n, q^n, \tilde{q}^n). \tag{3.58}$$

The term fluid limit denotes the limits obtained as we scale $n \to \infty$ for this process.

We assume that the rates satisfy $\mu_{ij}(t) \leq \mu_{max}$. This will happen since the channel gains are assumed bounded and transmit power is fixed.

We will use the following definition.

Definition 3.2 *A sequence of functions ξ^n is said to converge uniformly on compact sets (u.o.c) if $\xi^n \to \xi$ uniformly on every compact subset of the domain.*

We will also require the following theorem; for a proof see [27].

Lemma 3.1 *Let $\xi_n : [0, \infty) \to \mathbb{R}$ be a sequence of monotonically increasing functions. Let $\xi_n(x) \to \xi(x)$ for all rational x. If $\xi(x)$ is continuous, the convergence of ξ_n to ξ is u.o.c..*

We will also use the following well known result [70]. It is stated without proof.

Lemma 3.2 (Helly's Selection Theorem) *Let ξ_n be a sequence of monotonically increasing functions on $\mathbb{R}$, such that $0 \leq \xi_n(x) \leq B < \infty$, for all x and n. Then, there is a function ξ and a subsequence $\{n_k\}$ such that,*

$$\xi(x) = \lim_{n_k \to \infty} \xi_{n_k}(x). \tag{3.59}$$

We obtain the following result for z^n.

Theorem 3.2 *Consider a sequence of scaled systems $\{z^n, n \geq 0\}$ such that the initial condition $||Q(0)|| = n$ in the n-th system. Then, for almost every sample path ω, there exists a subsequence $n_k(\omega) \to \infty$ such that, along this subsequence,*

$$z^n \to z, \tag{3.60}$$

where $z = (a, e, g, d, r, s, \tilde{s}, q, \tilde{q})$. The component functions of z^n converge to the respective component functions of z u.o.c. as well. The limiting functions are also Lipschitz continuous, and hence almost everywhere differentiable. The limiting functions satisfy the following properties for all $t \geq 0$.

$$a(t) = \lambda t, \qquad e(t) = \gamma t, \tag{3.61}$$

$$r_i^f(t) = \sum_{k \neq i} s_{ki}^f(t), \qquad d_i^f(t) = \sum_{j \neq i} s_{ij}^f(t), \tag{3.62}$$

$$q_i^f(t) = q_i^f(0) + a_i^f(t) + r_i^f(t) - d_i^f(t), \tag{3.63}$$

$$\dot{q}_i^f(t) = \lambda_i^f + \dot{r}_i^f(t) - \dot{d}_i^f(t), \tag{3.64}$$

$$\sum_{I,\hat{\mu}} g_{\hat{\mu}}^{hI}(t) = e_h(t), \quad ||q(0)|| \leq 1, \tag{3.65}$$

$$\tilde{s}(t) = 0, \qquad \tilde{q}(t) = q(t), \tag{3.66}$$

$$s_{ij}^f(t) = \int_0^t \dot{s}_{ij}^f(\tau)d\tau, \tag{3.67}$$

where $\dot{s}(t)$ satisfies

$$\sum_{i,j,f} \alpha(q^f(t))q_{ij}^f(t)\dot{s}_{ij}^f(t) = \max_{\varpi \in \mathcal{W}} \sum_{i,j,f} \alpha(q^f(t))q_{ij}^f(t)\varpi_{ij}^f, \tag{3.68}$$

where the dot indicates derivative, at regular t (the points where the function is differentiable) and $\mathcal{W}$ is defined by (3.39).

Proof: The Strong Law of Large Numbers (SLLN) implies

$$\frac{\hat{A}_i^f(nt)}{n} = \frac{\sum_{\tau=1}^{nt} A_i^f(\tau)}{n} = t\frac{\sum_{\tau=1}^{nt} A_i^f(\tau)}{nt} \to \lambda t, \text{as } n \to \infty.$$

This, in conjunction with Lemma 3.1 gives the first part of (3.61). The convergence of e^n to γt u.o.c. also follows from the SLLN and Lemma 3.1.

The family of functions $\{\frac{1}{n}\hat{S}_{ij}^f(nt)\}$ is a family of monotone increasing functions. Moreover,

$$\frac{\hat{S}_{ij}^f(nt)}{n} \leq \frac{n\mu_{max}t}{n} = \mu_{max}t. \tag{3.69}$$

Using Helly's selection theorem (Lemma 3.2), one can obtain a convergent subsequence as follows. Consider intervals of the form $[0, m]$. Let s_m^n denote the function s^m restricted to $[0, m]$. Consider the family $\{s_1^n, n \geq 1\}$. This family is bounded by μ_{max} by (3.69), and hence, by Helly's selection theorem, we can obtain a convergent subsequence $\{s_{1*}^n, n \geq 1\}$. Now consider this subsequence of functions restricted to $[0, 2]$, and observe that these are uniformly bounded by $2\mu_{max}$. Applying Lemma 3.2 again, we obtain a further convergent subsequence, $\{s_{2*}^n, n \geq 1\}$. Proceed iteratively over m. Then, the convergent subsequence is given by the limit,

$$s = \lim_{m\to\infty} s_{m*}^m. \tag{3.70}$$

Thus, we obtain a subsequential limit s of s^n. Along this subsequence, $r_n \to r$ and $d_n \to d$ satisfying (3.62), due to (3.2) and (3.45).

Since the rates are bounded, it follows that $\hat{S}_{ij}^f(t) \leq \mu_{max}t$. Therefore, for $0 \leq t_1 \leq t_2$, we have

$$\hat{S}_{ij}^f(nt_2) - \hat{S}_{ij}^f(nt_1) \leq n\mu_{max}(t_2 - t_1),$$

and hence,

$$\frac{\hat{S}_{ij}^f(nt_2)}{n} - \frac{\hat{S}_{ij}^f(nt_1)}{n} \leq \mu_{max}(t_2 - t_1). \tag{3.71}$$

Taking the limit along the subsequence along which $s^n \to s$, we obtain,

$$s_{ij}^f(t_2) - s_{ij}^f(t_1) \leq \mu_{max}(t_2 - t_1). \tag{3.72}$$

It follows that s_{ij}^f is Lipschitz continuous, and hence so is s, and consequently r and d are Lipschitz as well. Hence, from Lemma 3.1, we obtain u.o.c. convergence for s^n, r^n and d^n along the chosen subsequence. Since s is Lipschitz and hence almost everywhere differentiable, (3.67) follows.

From the queueing equation (3.46), we can see that,

$$Q(nt) = Q(0) + \hat{A}(nt) + \hat{R}(nt) - \hat{D}(nt). \tag{3.73}$$

Dividing by n on both sides and taking $n \to \infty$ along the chosen subsequence yields the convergence,

$$q^n \to q, \tag{3.74}$$

with $q(t)$ defined by (3.63).

Since a, r and d are Lipschitz, q will also be Lipschitz, making it differentiable almost everywhere. At points where it is differentiable, we obtain (3.64) by differentiating (3.63).

The functions $G_{ijf}^{hI}(t)$ are also a monotone family, bounded uniformly on each compact interval. Hence, we can apply Helly's selection theorem again, as we did in the case of s^n, to obtain a subsequence along which $g^n \to g$. As is the case of s, observe that,

$$\frac{1}{n}(G_{\hat{\mu}}^{hI}(nt_2) - G_{\hat{\mu}}^{hI}(nt_1)) \leq t_2 - t_1, \tag{3.75}$$

for $t_2 > t_1$. This shows that g is Lipschitz continuous, and consequently along this new subsequence $g_n \to g$ u.o.c. as well.

Before characterizing the allocation process s, it must be pointed out that we do not distinguish between the actual and the ideal allocation, since they converge to the same limit. Ideal allocation is the allocation assigned in each review period to a flow f over a link (i, j). If the channel gain is μ_{ij} and the review period has length $\hat{T}$, the ideal allocation in that review period is $\zeta_{ij}^f \mu_{ij} \hat{T}$. However, the actual allocation may be slightly different, owing to roundoff errors (since service can only be in integer bits).

Let the actual (cumulative) allocation be $\bar{S}_{ij}^f(t)$. The actual allocation differs from the ideal allocation due to round-off errors. At a time nt, let $m = \max\{i : T_i \leq nt\}$. Bounding possible errors in each review period we get,

$$|\hat{S}_{ij}^f(nt) - \bar{S}_{ij}^f(nt)| \leq \mu_{max} \hat{T}_m + m\mu_{max}.$$

The last term follows by summing up round-off errors in review periods upto m, and observing that in any review period $\hat{T}$, errors are of the form $\mu_{ij}|x - \lfloor x \rfloor|$, where $x = \zeta_{ij}^f \hat{T}$. Since $m \leq \frac{nt}{T}$, where $T = \min_{i<m}\{\hat{T}_i\}$, we get

$$\frac{1}{n}|\hat{S}_{ij}^f(nt) - \bar{S}_{ij}^f(nt)| \leq \mu_{max} \left\{ \frac{\hat{T}_m}{n} + \frac{t}{T} \right\}.$$

Since $\hat{T}_i$ are $\max(1, \log(1 + k_0||Q||))$ and $\lim_{n\to\infty} ||Q|| = \infty$, we have $\lim_{n\to\infty} T = \infty$ and $\lim_{n\to\infty} \frac{\hat{T}_m}{n} = 0$, and hence, the fluid limits of $\hat{S}$ and $\bar{S}$ are equal.

To show (3.68), observe that,

$$S_{ij}^f(t) = \sum_{h,I,\hat{\mu}} G_{\hat{\mu}}^{hI}(t) \hat{\mu}_{ij}^f(h, I). \tag{3.76}$$

Hence, we have,

$$S_{ij}^f(nt_2) - S_{ij}^f(nt_1) = \sum_{h,I,\hat{\mu}} (G_{\hat{\mu}}^{hI}(nt_2) - G_{ijf}^{hI}(nt_1)) \hat{\mu}_{ij}^f(h, I).$$

Multiplying LHS and RHS by $\alpha(\frac{Q^f(nt_1)}{n}) \frac{Q_{ij}^f(nt_1)}{n} \frac{1}{n}$, summing over i, j, f, and taking $n \to \infty$, the LHS becomes

$$\sum_{i,j,f} \alpha(q^f(t_1)) q_{ij}^f(t_1) [s_{ij}^f(t_2) - s_{ij}^f(t_1)], \tag{3.77}$$

where $q_{ij}^f(t) = \max(q_i^f(t) - q_j^f(t), 0)$ and $q^f(t_1) = \lim_{n\to\infty} \frac{Q^f(nt_1)}{n} = \sum_i q_i^f(t)$. The RHS becomes,

$$\sum_{i,j,f} \alpha \left(\frac{Q^f(nt_1)}{n} \right) \frac{Q_{ij}^f(nt_1)}{n} \sum_{h,I,\hat{\mu}} \left(\frac{G_{\hat{\mu}}^{hI}(nt_2)}{n} - \frac{G_{\hat{\mu}}^{hI}(nt_1)}{n} \right) \hat{\mu}_{ij}^f(h, I). \tag{3.78}$$

The allocation satisfies

$$\sum_{i,j,f} \alpha\left(\frac{Q^f(nt^{'})}{n}\right)\frac{Q^f_{ij}(nt^{'})}{n}\hat{\mu}^f_{ij}(h,I) = \max_I \sum_{i,j,f} \alpha\left(\frac{Q^f(nt^{'})}{n}\right)\frac{Q^f_{ij}(nt^{'})}{n}\hat{\mu}^f_{ij}(h,I), \tag{3.79}$$

where $nt^{'}$ was the previous review point with $nt_1 = nt^{'} + T$. Going along the subsequence along which $q^n \to q$, we obtain,

$$\sum_{i,j,f} \alpha(q^f(t^{'}))q^f_{ij}(t^{'})\hat{\mu}^f_{ij}(h,I) = \max_I \sum_{i,j,f} \alpha(q^f(t^{'}))q^f_{ij}\hat{\mu}^f_{ij}(h,I). \tag{3.80}$$

Along the same subsequence, (3.78) becomes,

$$\sum_{i,j,f} \alpha(q^f(t_1))q^f_{ij}(t_1)\sum_{h,I,\hat{\mu}}(g^{hI}_{\hat{\mu}}(t_2) - g^{hI}_{\hat{\mu}}(t_2))\hat{\mu}^f_{ij}(h,I). \tag{3.81}$$

Since $0 \leq \frac{T}{n} \leq \frac{\hat{T}}{n} \to 0$, we can write $q^f_i(t_1)$ as

$$q^f_i(t_1) = \lim_{n\to\infty} \frac{Q^f_i(nt_1)}{n} = \lim_{n\to\infty} \frac{1}{n}Q^f_i(n(t^{'} + \frac{T}{n})) = q^f_i(t^{'}). \tag{3.82}$$

Using this fact in combination with (3.80), we see that (3.81) becomes,

$$\sum_{h,I,\hat{\mu}}(g^{hI}_{ijf}(t_2) - g^{hI}_{ijf}(t_2)) \max_I \sum_{\hat{i},\hat{j},\hat{f}} \alpha(q^{\hat{f}}(t_1))q^{\hat{f}}_{\hat{i}\hat{j}}(t_1)\hat{\mu}^{\hat{f}}_{\hat{i}\hat{j}}(h,I) \tag{3.83}$$

Using (3.65), (3.82) and (3.61), this becomes

$$\sum_h [e_h(t_2) - e_h(t_1)] \max_I \sum_{\hat{i},\hat{j},\hat{f}} \alpha(q^{\hat{f}}(t_1))q^{\hat{f}}_{\hat{i}\hat{j}}(t_1)\hat{\mu}^{\hat{f}}_{\hat{i}\hat{j}}(h,I), \tag{3.84}$$

$$=(t_2 - t_1)\sum_h \gamma_h \max_I \sum_{\hat{i},\hat{j},\hat{f}} \alpha(q^{\hat{f}}(t_1))q^{\hat{f}}_{\hat{i}\hat{j}}(t_1)\hat{\mu}^{\hat{f}}_{\hat{i}\hat{j}}(h,I). \tag{3.85}$$

Dividing (3.77) and (3.85) by $t_2 - t_1$, equating, and taking $t_2 \to t_1$, we obtain,

$$\sum_{i,j,f} \alpha(q^f(t_1))q^f_{ij}(t_1)\dot{s}^f_{ij}(t_1) = \sum_h \gamma_h \max_I \sum_{\hat{i},\hat{j},\hat{f}} \alpha(q^{\hat{f}}(t_1))q^{\hat{f}}_{\hat{i}\hat{j}}(t_1)\hat{\mu}^{\hat{f}}_{\hat{i}\hat{j}}(h,I). \tag{3.86}$$

Now observe that an element $\varpi \in \mathcal{W}$ satisfies,

$$\varpi_{ij}^f = \sum_h \gamma_h \sum_I \nu_h(I)\hat{\mu}_{ij}^f(h, I), \tag{3.87}$$

where $\nu_h(I)$ is a probability distribution over $\mathcal{S}$ and $\hat{\mu}_{ij}^f(h, I)$ satisfies,

$$\sum_f \hat{\mu}_{ij}^f(h, I) \leq \mu_{ij}(h, I), \tag{3.88}$$

for a some achievable rate μ and for all h, I. Hence the RHS of (3.86) can be written as,

$$\max_{\varpi \in \mathcal{W}} \sum_{i,j,f} \alpha(q^f(t_1)q_{ij}^f(t_1)\varpi_{ij}^f. \tag{3.89}$$

Thus, we obtain,

$$\sum_{i,j,f} \alpha(q^f(t_1))q_{ij}^f(t_1)\dot{s}_{ij}^f(t_1) = \max_{\varpi \in \mathcal{W}} \sum_{i,j,f} \alpha(q^f(t_1)q_{ij}^f(t_1)\varpi_{ij}^f, \tag{3.90}$$

with $\mathcal{W}$ defined by (3.39). Thus we obtain (3.68).

To obtain the first part of (3.66), observe that

$$0 \leq \tilde{S}_{ij}^f(n, t) \leq \mu_{max}\frac{\hat{T}}{n},$$

with $\hat{T}$ being a review period. Taking $n \to \infty$, we see that,

$$\tilde{s}(t) = 0. \tag{3.91}$$

The second part of (3.66) follows from (3.82). The first part of (3.65) follows by applying the fluid scaling to (3.41). From the assumption that $||Q(0)|| = n$ for the n-th system, the second part of (3.65) follows. □

Denote the vector of all $q_i^f(t)$ by $q(t)$. We will use the following result to establish the stability of the network.

Theorem 3.3 *(Theorem 4 of [2]) Let Y be a Markov Process with $||Y(.)||$ denoting its norm. If there exist $\alpha > 0$ and a time $T > 0$ such that for a scaled sequence of processes $\{Y^n, n =$*

$0, 1, 2, ..\}$, *we have*

$$\lim_{n\to\infty} \sup \mathbb{E}[||Y(n,T)||] \leq 1-\alpha,$$

then the process Y is stable (positive recurrent).

Using this result, we will establish stability of the network under our algorithm and show that it is throughput optimal.

Theorem 3.4 *The policy QWDR, as defined in (3.30), stabilizes the process $\{Q(t), t \geq 0\}$ for all arrivals in the interior of Λ.*

Proof: Pick an arrival rate $\lambda = \{\lambda_i^f\} \in int(\Lambda)$. Consider the Lyapunov function,

$$\mathcal{L}_1(q(t)) = -\int_t^\infty \exp(t-\tau) \sum_{i,f} \alpha(q^f(\tau)) q_i^f(\tau) \dot{q}_i^f(\tau) d\tau, \tag{3.92}$$

where the dot indicates the derivative. This is a continuous function of $q(t)$, with $L(0) = 0$. We can write the (time) derivative,

$$\dot{\mathcal{L}}_1(q(t)) = \sum_{i,f} \alpha(q^f) q_i^f \dot{q}_i^f = \sum_{i,f} \alpha(q^f) q_i^f (\lambda_i^f + \sum_m \dot{s}_{mi}^f(t) - \sum_n \dot{s}_{in}^f(t)). \tag{3.93}$$

This follows from (3.63).

Recall the definition of Λ (3.34)-(3.38). Since λ is in the interior of Λ, there exists a non negative vector $[\varpi_{ij}^f]_{(i,j)\in\mathcal{E}, f\in\mathcal{F}}$ for which,

$$\lambda_i^f + \epsilon < \sum_j \varpi_{ij}^f - \sum_k \varpi_{ki}^f \;\forall i, f, \tag{3.94}$$

and there exists $m \in \mathcal{M}$ such that,

$$\sum_f \varpi_{ij}^f \leq m_{ij}. \tag{3.95}$$

Substituting this in the previous equation, we can write,

$$\begin{aligned}\dot{\mathcal{L}}_1(q(t)) < -\epsilon \sum_{i,f} \alpha(q^f) q_i^f + \\ \sum_{i,f} \alpha(q^f) q_i^f (\sum_n \varpi_{in}^f - \sum_m \varpi_{mi}^f + \sum_m \dot{s}_{mi}^f(t) - \sum_n \dot{s}_{in}^f(t)).\end{aligned}$$

Observing that

$$\sum_{i,f} \alpha(q^f) q_i^f (\sum_n \varpi_{in}^f - \sum_m \varpi_{mi}^f) = \sum_{i,j,f} \alpha(q^f) \varpi_{ij}^f (q_i^f - q_j^f),$$

and that a similar equation holds for ϖ replaced by $\dot{s}$, it follows that if we show

$$\sum_{i,j,f} \alpha(q^f) \varpi_{ij}^f (q_i^f - q_j^f) \leq \sum_{i,j,f} \alpha(q^f) \dot{s}_{ij}^f (q_i^f - q_j^f), \tag{3.96}$$

it will imply $\dot{\mathcal{L}}_1(q(t)) < 0$. We have

$$\sum_{i,j,f} \alpha(q^f) \varpi_{ij}^f (q_i^f - q_j^f) \leq \sum_{i,j,f} \alpha(q^f) \varpi_{ij}^f q_{ij}^f \leq \sum_{i,j,f} \alpha(q^f) \dot{s}_{ij}^f q_{ij}^f,$$

where the first inequality follows from the fact that $q_{ij}^f = (q_i^f - q_j^f)^+$, and the second follows from (3.68).

Now, if we show that $\dot{s}_{ij}^f = 0$ whenever $q_{ij}^f = 0$, (3.96) will follow. To see this, assume that at some t, $\dot{s}_{ij}^f = \delta^1 > 0$ and $q_{ij}^f = 0$. This would mean that for large enough n, there is a time s sufficiently close to t such that, for $\delta = \frac{\delta^1}{2}$,

$$S_{ij}^f(nt) - S_{ij}^f(ns) > n\delta(t-s).$$

This implies that at a time $t_1 \in (s,t)$ with $Q_i^f(nt_1) - Q_j^f(nt_1) \leq 0$ the queue Q_i^f was served. This would mean that the optimization resulted in a positive μ_{ij}^f. This cannot when all Q_{ij}^f are zero, since in that state, by definition, all μ_{ij}^f are set to zero. Hence there exists k, l, m such that $Q_{kl}^m > 0$. If μ_{ij}^f is added to μ_{kl}^m, the value of the summand in (3.30) would only increase, thus contradicting its optimality. It follows that $\dot{s}_{ij}^f = 0$ whenever $q_{ij}^f = 0$, and hence, (3.96) is true.

Thus, $\dot{\mathcal{L}}_1(q(t)) < -\epsilon \sum_{i,f} \alpha(q^f) q_i^f$, and hence, from (3.92) and (3.93), we see that $\mathcal{L}_1(q(t)) > 0$ whenever $q(t) \neq 0$. Fix $\delta_1 < 1$. Then, there exists $T \leq T_1 = \frac{\mathcal{L}_1(q(0))}{\epsilon\delta_1} + \delta_1$ such that $\sum_{i,f} q_i^f \leq \delta_1$. To see this, assume otherwise, that $\sum_{i,f} q_i^f(t) > \delta_1$ for $t \in [0, T_1]$. Now,

$$\mathcal{L}_1(q(t)) = \mathcal{L}_1(q(0)) + \int_0^t \dot{\mathcal{L}}_1(q(\tau)) d\tau.$$

Since q is Lipschitz, $\dot{q}$ will be bounded. It is easy to see that $\mathcal{L}(q(0))$ is finite. Since $w(q^f) \geq 1$,

$$\mathcal{L}_1(q(t)) \leq \mathcal{L}_1(q(0)) - \epsilon\delta_1 t,$$

for $t \in [0, T_1]$, and by choosing $t = T_1$, we obtain $\mathcal{L}_1(q(T_1)) < 0$, which is a contradiction. Hence, $\sum_{i,f} q_i^f(T) \leq \delta_1$. Since the fluid queue is a deterministic process following the trajectory defined by equations (3.61)-(3.68), it follows that, almost surely,

$$\lim_{n\to\infty} \sup ||Q^n(T)|| = \sum_{i,j,f} q(T) \leq \delta_1 < 1.$$

From the definition of Q, we have that

$$||Q^n(T)|| \leq [1 + \sum_{i,f} A_i^{f,n}(T) + T \sum_{i,j,f} \mu_{max}].$$

Since $\mathbb{E}[\sum_{i,f} A_i^{f,n}(T)] = T(\sum_{i,f} \lambda_i^f) < \infty$, we can use the Dominated Convergence Theorem [3] to see that Theorem 3.3 holds for Q with $\alpha = 1 - \delta_1$. The result follows. □

3.8 Simulation Results

For simulation we consider a fifteen node network with seven flows, with connectivity as depicted in Fig 3.8, over a unit area. We will be trying to provide mean delay QoS for three of these flows. The channel gains are Rayleigh distributed with parameters inversely proportional to the square of the distance between nodes, and the arrival distribution is Poisson. The flows are $F10 : 7 \rightarrow 9 \rightarrow 10$, $F4 : 7 \rightarrow 8 \rightarrow 2 \rightarrow 4$, $F11 : 1 \rightarrow 2 \rightarrow 4 \rightarrow 11$, $F13 : 9 \rightarrow 10 \rightarrow 13$, $F12 : 1 \rightarrow 3 \rightarrow 6 \rightarrow 12$, $F15 : 5 \rightarrow 14 \rightarrow 15$ and $F6 : 5 \rightarrow 3 \rightarrow 6$. The constant $k_0 = 0.01$ in (3.2), and in the distributed optimization, the algorithm runs 15 cycles over the set of nodes with $\alpha = 0.0001$ and the initial state is zero. The simulation runs for 10^5 slots, with $a_1 = 0.2$ and $a_2 = 2$ in (3.27). The arrival rates are 3.8 for $F6$, 3.74 for $F10$, and 2.5 for the others. These are chosen to take the queues to the edge of the stability region, where delays are larger, and the control of the algorithm in providing QoS will be more evident. The values are shown in Table 3.3, with flows $F10$, $F11$ and $F6$ having mean delay requirements, which are translated to $\lambda\bar{Q}^f$ in the w function. The delays are rounded to the nearest integer.

The first row represents delays of the flows when $w = 1$ for all flows, i.e., no priority is given. In the other rows, the values in brackets are of the form (target delay, achieved delay). The flows seem to respond very well to the target, often coming much lower than what is desired, since the weights tend to push the queue lengths to below these threshold values. In all cases, the delays can be brought down to less than 50% of their unweighed values. Another effect is that giving QoS to one flow does not adversely affect the delay of the other flows. In fact, it can substantially reduce the mean delays of the other flows as well. Since the algorithm uses backpressure values, this is not surprising, and the weight function can be thought of as fine tuning the delay behaviour of the network.

The modified algorithm, QWDR, though throughput optimal, was not able to provide hard deadline guarantees. This suggests that the modified weight function α, is not sufficient to provide the necessary priority.

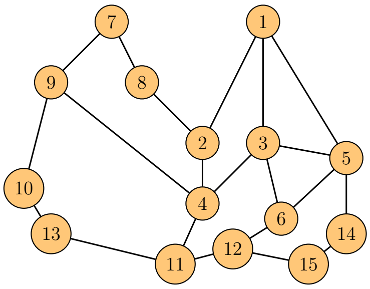

Figure 3.8: Sample Network

Table 3.3: Simulation for example in Fig 3.8. Three Flows with mean delay requirements, network of fifteen nodes. Entries of the form (a,b) indicate delay target a, delay achieved b.

Mean Delay(slots) for each flow						
F10	F4	F11	F13	F12	F15	F6
318	68	499	233	642	25	111
(200,188)	61	(350,304)	163	403	23	(70,70)
(150,96)	60	(300,265)	85	362	22	(60,65)
(150,67)	56	(150,148)	61	235	22	(45,55)
(200,136)	56	(130,134)	119	220	22	(50,55)

3.9 Conclusion

In this chapter, we have developed a distributed algorithm to provide Quality-of-Service requirements in terms of end-to-end mean delay guarantees and hard deadline guarantees to flows in a multihop wireless network. The algorithm uses discrete review to solve an optimization problem at review instants, and uses a control policy based on the solution of an optimization problem. The algorithm optimizes, in a distributed fashion, a function with distributed weights given to pseudo draining times, with the weights varied dynamically to provide priority for flows in the network, and consequently, meeting their respective delay constraints. We use iterative gradient ascent and distributed iterative projection methods in order to compute the optimal point in a distributed manner. By means of simulations we establish the efficacy of the algorithm in providing the required delay demands. We study the convergence properties of the algorithm and also see via simulations that the algorithm converges quickly. We also demonstrate throughput optimality of a modified version algorithm by a theoretical analysis. This used the technique of fluid limits. The modified algorithm also meets mean delay constraints. Surprisingly, the algorithm not only reduces the mean delays of the targeted flows, it reduces mean delays of other flows as well. While a smoother weight function α is necessary for the throughput optimal algorithm, it is not sufficient to provide for hard delay guarantees.

Chapter 4

Diffusion Approximation and Convergence of Stationary Distributions

In the previous chapter we developed a new scheduling algorithm which provides QoS and is also thoughput optimal. The QoS was confirmed via simulations. It is of interest to study its performance theoretically. In particular it will be very useful to compute the end-to-end mean delay and more generally the distribution under stationarity. However, it is intractable to compute these quantities for such a complicated system. Thus, we consider approximations.

In this chapter, we obtain a diffusion approximation of the network in the heavy traffic regime. This is done by taking the limit of scaled system processes, where the scaling corresponds to the Functional Central Limit Theorem. The limiting process is a reflected Brownian motion with drift. Furthermore, we also show that the stationary distribution of the scaled process of the network converges to that of the Brownian limit, providing an approximation to the stationary distribution under heavy traffic. This will provide approximations of stationary end-to-end mean delay and distribution even under moderately heavy load, which are of main interest in practice, because under low load, the QoS of different flows will anyway generally be met. Finally simulations further verify our claims.

4.1 System Model

We continue with the system model that was used in the previous chapter. We consider a multihop wireless network (Fig. 4.1). The network is modelled as a connected directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{1, 2, \ldots, N\}$ being the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ being the set of links.

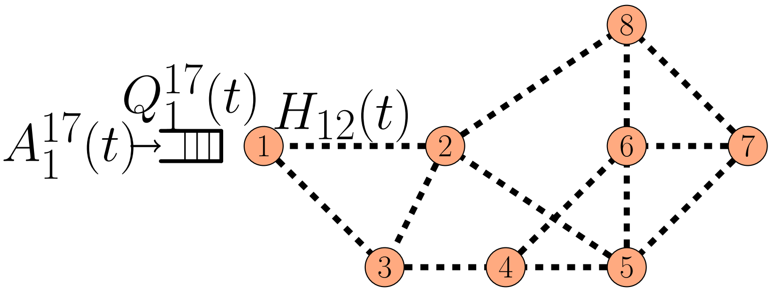

Figure 4.1: A simplified depiction of a Wireless Multihop Network. The flow 17 corresponds to the source 1 and destination 7.

We consider a discrete time system, with time slots of length 1. The links are directed, with link (i, j) from node i to node j having a time varying channel gain $H_{ij}(t)$ at time t (it stays constant over one time slot). The channel gain vector, $H(t) = (H_{ij}(t))_{(i,j)\in\mathcal{E}}$, evolves as an independent and identically distributed (i.i.d.) process across slots with distribution γ over a finite set $\mathcal{H}$. Let $E_h(t)$ denote the cumulative number of slots in $[0, t]$ when the channel state was $h \in \mathcal{H}$. The vector $(E_h(t))_{h\in\mathcal{H}}$ is denoted by $E(t)$. We assume that the links are sorted into M *interference sets* $\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_M$. At any time, only one link from an interference set can be active. A link may belong to multiple interference sets.

There are multiple *flows* in the network, each corresponding to a source-destination pair. The set of all flows is denoted by $\mathcal{F}$. We assume that every flow has a fixed path to follow from source to destination. Denote the set of links on the path corresponding to flow f by $\mathcal{R}_f$. Denote the source node of flow f by $src(f)$, and destination by $des(f)$. At any node $i = src_(f)$, $A_i^f(t)$ denotes the process of exogenous arrival of packets corresponding to flow f. The packets arrive as an i.i.d sequence across slots, with mean arrival rate λ_i^f and variance σ_i^f. Let λ denote the vector of all λ_i^f. Define the cumulative arrival process,

$$\hat{A}(t) = \sum_{\tau=1}^{t} A(\tau). \tag{4.1}$$

At each node there are queues, with $Q_i^f(t)$ denoting the queue length at node i corresponding to flow $f \in \mathcal{F}$ at time t. Define the set of all link-flow pairs to be $\mathcal{K} = \{((i, j), f) : (i, j) \in \mathcal{E}, f \in \mathcal{F}\}$. A schedule is a mapping from $\mathcal{K}$ to $[0, 1]$. The value of the schedule vector for link-flow element k corresponds to the fraction of time that flow is scheduled over that link. The set of feasible schedules is denoted by $\mathcal{S}$. The elements of $\mathcal{S}$ are determined by the interference constraints of the network. For a given channel state $h \in \mathcal{H}$ and schedule $I \in \mathcal{S}$, there is a rate vector, $\mu = (\mu_{ij})_{(i,j)\in\mathcal{E}}$ defined as,

$$\mu_{ij} = \mu_{ij}(h, I). \tag{4.2}$$

This will be an achievable rate function. Corresponding to any rate vector there will be allocation vectors, $\hat{\mu} = [\hat{\mu}_{ij}^{f}]_{(i,j)\in\mathcal{E},f\in\mathcal{F}}$. These are non negative vectors that satisfy,

$$\sum_{f\in\mathcal{F}} \hat{\mu}_{ij}^{f} \leq \mu_{ij}, \tag{4.3}$$

for some rate vector $\mu(h, I)$. We also assume that $\mu_{ii}^{f} = 0$ for all i, f and $\mu_{i,j}^{f} = 0$ for all $i = des(f)$. Also, $\mu_{ij}^{f} = 0$ if $(i, j) \notin \mathcal{R}_f$. Define,

$$\mathcal{M}_h = \{\mu(h, I) : I \in \mathcal{S}\} \tag{4.4}$$

The number of bits of flow f transmitted from node i to node j in time slot t is denoted by $S_{ij}^{f}(t)$. The vector $[S_{ij}^{f}(t)]_{(i,j)\in\mathcal{E},f\in\mathcal{F}}$ is denoted by $S(t)$. Denote by $\hat{S}(t)$ the cumulative process $\sum_{\tau=1}^{t} S(\tau)$.

For a queue Q_i^f with $i \neq f$, we have the queue evolution given by,

$$Q_i^f(t) = Q_i^f(0) + \hat{A}_i^f(t) + \hat{R}_i^f(t) - \hat{D}_i^f(t), \tag{4.5}$$

where $R_i^f(t)$ is the cumulative arrival of packets by routing (i.e., arrivals from other nodes), and $D_i^f(t)$ is the cumulative departure of packets, given by,

$$R_i^f(t) = \sum_{k\neq i} \hat{S}_{ki}^{f}(t), \text{ and } D_i^f(t) = \sum_{j\neq i} \hat{S}_{ij}^{f}(t), \tag{4.6}$$

and $Q_i^f(0)$ is the initial queue length, at time 0. The vector of queues at time t is denoted by $Q(t)$. Similarly we have the vectors $\hat{A}(t)$, $\hat{R}(t)$, $\hat{D}(t)$ and $\hat{S}(t)$.

We want to develop scheduling policies such that the different flows obtain their end-to-end mean delay deadline guarantees. Define $Q_{ij}^f = \max(Q_i^f - Q_j^f, 0), Q^f(t) = \sum_i Q_i^f(t)$. Our network control policy is the same as in Chapter 3. The only difference is that unlike in Chapter 3, now the control is exercised after each slot, i.e., the discrete review interval is one slot. We obtain the optimal allocation $\bar{\mu}(t) = \hat{\mu}(H(t), I^*(t))$, where,

$$I^*(t) = \arg_{I\in\mathcal{S}} \max \sum_{i,j,f} \alpha(Q^f(t), \overline{Q}^f) Q_{ij}^f(t) \hat{\mu}_{ij}^{f}(H(t), I), \tag{4.7}$$

assuming $Q_{ij}^f > 0$ for at least one link flow pair $(i,j), f$. If all Q_{ij}^f are zero, we define the solution to be $\bar{\mu}(t) = 0$. The number of bits of flow f transmitted over link (i,j) is given by,

$$\bar{S}_{ij}^f(t) = \min(Q_i^f(t), \sum_j \bar{\mu}_{ij}^f(t)). \tag{4.8}$$

In (4.7), we optimize a weighted sum of rates, with more weight given to flows with larger backlogs, with α capturing the delay requirement of the flow. The weights α are functions of $Q^f(t)$, and $\overline{Q}^f$ denotes a desired value for the queue length of flow f, which is determined by the end-to-end mean delay requirement of flow f. We use,

$$\alpha(x, \overline{x}) = 1 + \frac{a_1}{1 + \exp(-a_2(x - \overline{x}))}. \tag{4.9}$$

Thus, flows requiring a lower mean delay would have a higher weight compared to flows needing a higher mean delay. Flows whose mean delay requirements are not met should get priority over the other flows.The $\overline{Q}^f$ are chosen, using Little's Law, as $\overline{Q}^f = \lambda^f \tau_{delay}^f$, where τ_{delay}^f is the target end to end mean delay and λ^f is the arrival rate of flow f. Note that we will often use $\alpha(x)$ instead of $\alpha(x, \bar{x})$ for simplicity of notation.

Let $G_{\hat{\mu}}^{hI}(t)$ be the number of slots till time t, in which channel state was h, the schedule was I and the rate function chosen was $\hat{\mu}$. Denote the vector of all $G_{ijf}^{hI}(t)$ by $G(t)$. Define the process,

$$Z = (A, E, G, D, R, S, Q), \tag{4.10}$$

where we have $A = \{A(t), t \geq 0\}$ (and likewise for the other processes). This process describes the evolution of the system. The state of the system at time t is $Q(t)$, which takes values in a state space $\mathcal{Q}$. The capacity region Λ and rate region $\mathcal{W}$ of the network is defined as in Chapter 3, (3.34)-(3.38).

4.2 Fluid Limit and Stability

We first establish the throughput optimality of the system under the control policy given by (4.7). Towards this, we first define the fluid scaling and fluid limit of the system as in Chapter 3. For the process $Z(t)$, define the sequence of scaled processes, given by,

$$z^n(t) = \frac{Z(\lfloor nt \rfloor)}{n}, \tag{4.11}$$

where $n \in \mathcal{N}$. This is called fluid scaling of the process Z. Denote by z^n the process $\{z^n(t), t \geq 0\}$. Clearly,

$$z = (a, e, g, d, r, s, q). \tag{4.12}$$

Then, we have the following result.

Lemma 4.1 *1 Let $\mathcal{N}$ be a sequence of positive integers increasing to infinity. Then, there exists a subsequence $\mathcal{N}_1 \subseteq \mathcal{N}$, such that, as $n \to \infty$ along $\mathcal{N}_1$, we have, almost surely (a.s.),*

$$z^n \to z, \tag{4.13}$$

where $z = (a, e, g, d, r, s, q)$, is called a fluid limit*, and the convergence of the processes is u.o.c. The limiting functions are also Lipschitz continuous, and hence almost everywhere differentiable. The points t at which these are differentiable are called regular points. In addition, the limiting functions satisfy,*

$$a(t) = \lambda t, \qquad e(t) = \gamma t, \tag{4.14}$$

$$r_i^f(t) = \sum_j s_{ji}^f(t), \qquad d_i^f(t) = \sum_j s_{ij}^f(t), \tag{4.15}$$

$$q(t) = q(0) + a(t) + r(t) - d(t), \tag{4.16}$$

$$\dot{q}(t) = \lambda + \dot{r}(t) - \dot{d}(t), \tag{4.17}$$

$$\sum_{I,\hat{\mu}} g_{\hat{\mu}}^{hI}(t) = e_h(t), \quad s_{ij}^f(t) = \int_0^t \dot{s}_{ij}^f(\tau) d\tau, \tag{4.18}$$

where $\dot{s}(t)$ satisfies

$$\sum_{i,j,f} \alpha(q^f(t)) q_{ij}^f(t) \dot{s}_{ij}^f(t) = \max_{\varpi \in \mathcal{W}} \sum_{i,j,f} \alpha(q^f(t)) q_{ij}^f(t) \varpi_{ij}^f, \tag{4.19}$$

and the dot indicates derivative, at regular t.

The proof of this lemma is similar to the proof of Theorem 3.2. Note that the fluid limit processes obtained here are the same as those obtained in Theorem 3.2. Thus, both systems have the same evolution as far as the fluid queue is concerned. Since the proof is quite close to that of Theorem 3.2, we skip it.

Similarly, we also obtain the following stability result.

Lemma 4.2 *The policy given in (4.7) stabilizes the process* $\{Q(t), t \geq 0\}$ *for all arrivals in the interior of* Λ*.*

Again, the proof is the same as for Theorem 3.4. Since the fluid queue has the same evolution in both cases, we can use the same Lyapunov function, given by,

$$\mathcal{L}_1(q(t)) = -\int_t^\infty \exp(t-\tau) \sum_{i,f} \alpha(q^f(\tau)) q_i^f(\tau) \dot{q}_i^f(\tau) d\tau. \tag{4.20}$$

This function is positive, and has negative drift when $\lambda \in int(\Lambda)$.

4.2.1 Draining Time

An important parameter that we obtain from the fluid limit is the draining time τ_{drain}, defined as the time t by which the fluid queue $q(t)$ has norm zero. We have the following result, which relates the draining time to the time T obtained in the proof of Lemma 3.4.

Lemma 4.3 *For the fluid limit* z *defined by Lemma 4.1,*

$$\tau_{drain} \leq \frac{T}{1-\delta_1}. \tag{4.21}$$

Proof: The proof is a consequence of the scaling properties of the fluid limit functions. Observe that, for any positive δ, as $n \to \infty$ along $\mathcal{N}_1$,

$$q(t) = \lim_{n\to\infty} \frac{Q(\lfloor \delta n t \rfloor)}{\delta n} = \frac{1}{\delta} q(\delta t). \tag{4.22}$$

Hence, a fluid limit path $q(t)$ is equivalent to a fluid path $\frac{q(\delta t)}{\delta}$. Let us define a fluid path $q^{'}(t) = q(t+T)$ for $t \geq 0$. This is a fluid path with initial condition,

$$|q^{'}(0)| \leq \delta_1. \tag{4.23}$$

Observe that, by (4.22),

$$q^{'}(t) = q(t+T) = \frac{1}{\delta_1^{-1}} q(\delta_1^{-1}(t+T)). \tag{4.24}$$

If T_1 is the time for the path $q^{'}(t)$ to reach the level $(\delta_1)^2$, we have,

$$|q(\delta_1^{-1}(T_1+T))| = \delta_1. \tag{4.25}$$

However, $|q(t)|$ reaches δ_1 in time T. Hence $|q(\delta_1^{-1}t)|$ reaches δ_1 in time $t = \delta_1 T$. Hence, $T_1 \leq \delta_1 T$. Continuing in this line, we can bound the time to reach δ_1, $(\delta_2)^2$, and so on by T_1, T_2, etc., where

$$T_n \leq (\delta_1)^n T. \tag{4.26}$$

Hence, the time for the queue to reach level zero is bounded by,

$$T + \delta_1 T + (\delta_1)^2 T + \cdots = \frac{T}{1-\delta_1}. \tag{4.27}$$

□

Studying the fluid limit gives us insights into the stability properties of the system. However, it only proves the existence of a stationary distribution. In order to predict the behaviour of the system, one needs the stationary distribution, or some approximation to the same. However, explicitly computing the stationary distribution for our system is not feasible. Thus we define the heavy traffic regime, and the associated diffusion scaling, below. We will also show that the stationary distribution of our system process converges to that of the limiting Brownian network. This will provide us an approximation of the stationary distribution of ours system under heavy traffic, the scenario of most practical interest.

4.3 Diffusion Scaling and Heavy Traffic Limit

Now we consider a new sequence of scaled systems, Z^n. The n-th process is the above system but with arrival rate vector λ^n and standard deviation σ^n. The λ^n are chosen such that, as $n \to \infty$, $\lambda^n \to \lambda^*$, and,

$$\lim_{n\to\infty} n\langle \psi, \lambda^n - \lambda^* \rangle = b^* \in \mathbb{R}, \tag{4.28}$$

where λ^* is a point on the boundary of Λ, and ψ denotes the outer normal vector to Λ at the point λ^*. We will also assume that λ^* falls in the relative interior of one of the faces of the boundary of Λ (This is the *resource pooling* condition). For this sequence of systems, we define the diffusion scaling, given by,

$$\hat{z}^n(t) = \frac{Z^n(\lfloor n^2 t \rfloor)}{n}. \tag{4.29}$$

Let $\hat{z}^n$ denote the process $(\hat{z}^n(t), t \geq 0)$. As before, we have,

$$\hat{z}^n = (\hat{a}^n, \hat{e}^n, \hat{g}^n, \hat{d}^n, \hat{r}^n, \hat{s}^n, \hat{q}^n).$$

Define the system workload $W^n(t)$ in the direction ψ,

$$W^n(t) = \langle \psi, Q^n(t) \rangle, \tag{4.30}$$

and,

$$\hat{w}^n(t) = \frac{W(\lfloor n^2 t \rfloor)}{n}.$$

Denote $\hat{w}^n = \{\hat{w}^n(t), t \geq 0\}$. We will use $\mathscr{D}[0, \infty)$ to denote the space of all functions from $[0, \infty)$ to $\mathbb{R}$, that are right continuous with left hand limits (RCLL, also called cadlag).

Define an *invariant point* to be a vector ϕ that satisfies, for some $k > 0$,

$$\alpha(\phi)\phi = k\psi, \tag{4.31}$$

where $\alpha(\phi)$ is the vector of all $\alpha(\phi_j)$, with α defined in (4.9). Assume that,

$$\sigma^n \to \sigma, \tag{4.32}$$

as $n \to \infty$. Assume that the arrival process $A^n(t)$ satisfies, for all i, f,

$$\lim_{x \to \infty} \sup_{n \geq 1} \mathbb{E}[(A_i^{f,n}(1))^2 \mathbf{1}_{\{A_i^{f,n}(1) \geq x\}}] = 0. \tag{4.33}$$

This is a sufficient condition for Donsker's Theorem to hold for the arrival process [13]. A sufficient condition for the above condition is,

$$\sup_{n\geq 1} \mathbb{E}[A_i^{f,n}(1)]^{2+\delta} < \infty, \tag{4.34}$$

for some $\delta > 0$. Under these assumptions, we have the following result, which characterizes the weak convergence of the diffusion scaled processes.

Theorem 4.1 *Consider $\{\hat{z}^n, n \in \mathcal{N}\}$, under heavy traffic scaling satisfying (4.28),and $\mathcal{N}$ a sequence of positive integers n increasing to infinity. Assume that the arrival process satisfies (4.33). Further, assume that,*

$$\hat{q}^n(0) \xrightarrow{\mathscr{L}} c\phi, \tag{4.35}$$

where c is a non negative real number. Then, the sequence $\{\hat{w}^n, n \in \mathcal{N}\}$ converges weakly to a reflected Brownian motion $\hat{w}$ as $n \to \infty$ in $\mathscr{D}[0, \infty)$. Further, $\{\hat{q}^n, n \in \mathcal{N}\}$ converges weakly to $\phi\hat{w}$.

The proof of this Theorem proceeds in the following manner. The process $\hat{w}^n$ is decomposed into two parts. The first of these parts converges to a Brownian motion. The second converges to the *unique regulator* corresponding to the Brownian motion. Together, they add up and form a *reflected* Brownian motion. First, we decompose $\hat{w}^n$.

Towards this, first we define, $\mathcal{W}_h$ to be,

$$\mathcal{W}_h = \{[\varpi_{ij}^f]_{(i,j)\in\mathcal{E}, f\in\mathcal{F}} = \varpi : \exists m \in \mathcal{M}_h \ s.t. \sum_f \varpi_{ij}^f \leq m_{ij}, \ \forall i, j\}. \tag{4.36}$$

For a vector $\varpi = [\varpi_{ij}^f]_{(i,j)\in\mathcal{E}, f\in\mathcal{F}}$, define the transformation ζ by,

$$\zeta_i^f(\varpi) = \sum_j \varpi_{ij}^f - \sum_k \varpi_{ki}^f. \tag{4.37}$$

Applied to a rate vector, this shows the net outflow by routing. Define the set,

$$\zeta(\mathcal{W}_h) = \{\zeta(\varpi) : \varpi \in \mathcal{M}_h\}. \tag{4.38}$$

Let us denote the maximum allocation in the direction ψ, when the channel is in state h, by

$$\rho_h = \max_{\rho\in\zeta(\mathcal{W}_h)} \langle \psi, \rho \rangle, \ h \in \mathcal{H}. \tag{4.39}$$

Define the vectors,

$$\rho = [\rho_h]_{h \in \mathcal{H}}, \tag{4.40}$$

$$\hat{\rho} = [(\rho_h)^2]_{h \in \mathcal{H}}. \tag{4.41}$$

Define the random variables,

$$X_\mu(t) = \tilde{\mu}_{H(t)},\ t \geq 1.$$

The random variables $\{X_\mu(t), t \geq 0\}$ are i.i.d, with mean and variance given by,

$$\hat{\nu} = \langle \rho, \gamma \rangle,\ \hat{\sigma}^2 = \mathbb{E}[(X_\mu(1) - \hat{\nu})^2] = \langle \hat{\rho}, \gamma \rangle - \hat{\nu}^2 \geq 0.$$

Define the cumulative process,

$$X(t) = \sum_{k=1}^{t} X_\mu(k). \tag{4.42}$$

This is the cumulative maximum possible service in the direction ψ. We can write,

$$U(t) = W(0) + \langle \psi, A(t) \rangle - X(t), \tag{4.43}$$

$$V(t) = X(t) + \langle \psi, R(t) \rangle - \langle \psi, D(t) \rangle, \tag{4.44}$$

and, consequently, we can decompose the workload as,

$$W(t) = U(t) + V(t). \tag{4.45}$$

Consequently,

$$W^n(n^2t) = U^n(n^2t) + V^n(n^2t). \tag{4.46}$$

Define,

$$\hat{u}^n(t) = \frac{U^n(\lfloor n^2t \rfloor)}{n},\ \hat{v}^n(t) = \frac{V^n(\lfloor n^2t \rfloor)}{n}.$$

Thus we have,

$$\hat{w}^n(t) = \hat{u}^n(t) + \hat{v}^n(t). \tag{4.47}$$

Let us denote $\hat{w}^n = \{\hat{w}^n(t), t \geq 0\}$, $\hat{u}^n = \{\hat{u}^n(t), t \geq 0\}$ and $\hat{v}^n = \{\hat{v}^n(t), t \geq 0\}$.

Thus, we have decomposed the $\hat{w}^n$ process. Now we look at convergence of the constituent processes.

4.3.1 Convergence of $\hat{u}^n$

The following theorem tells us about the convergence of the $\hat{u}^n$ component.

Lemma 4.4 *Assuming that the initial condition converges weakly to an invariant point, i.e,*

$$\hat{w}^n(0) \xrightarrow{\mathscr{L}} \hat{w}(0), \tag{4.48}$$

as $n \to \infty$ along $\mathcal{N}$, where $\alpha(\hat{w}(0))\hat{w}(0) = \psi$. Then, it follows that,

$$\hat{u}^n \xrightarrow{\mathscr{L}} \hat{u},$$

in $\mathscr{D}[0,\infty)$ as $n \to \infty$ along $\mathcal{N}$, where $\hat{u} = (\hat{u}(t), t \geq 0)$ is a Brownian motion with drift, given by,

$$\hat{u}(t) = \hat{w}(0) + b^* t + \sigma \mathscr{B}(t), \tag{4.49}$$

where $\mathscr{B}(t)$ is a standard Brownian motion, $\sigma^2 = \sum_{i,f} (\sigma_i^f)^2 + \hat{\sigma}^2$, and b^* is given by (4.28).

Proof: We can write $\hat{u}^n$ as,

$$\begin{aligned}
\hat{u}^n(t) &= \frac{U^n(n^2 t)}{n} \\
&= \hat{w}^n(0) + \langle \psi, \hat{a}^n(t) \rangle - \hat{x}^n(t), \\
&= \hat{w}^n(0) + \langle \psi, \hat{a}^n(t) - \lambda^n nt \rangle - (\hat{x}^n(t) - \hat{\nu} nt) + (\langle \psi, \lambda^n \rangle - \hat{\nu}) nt.
\end{aligned}$$

Since $\hat{\nu} = \langle \rho, \gamma \rangle$, we can see that,

$$\hat{\nu} = \sum_{h \in \mathcal{H}} \gamma_h \rho_h = \sum_{h \in \mathcal{H}} \gamma_h \max_{\rho \in \zeta(\mathcal{M}_h)} \langle \psi, \rho \rangle = \max_{\tilde{\rho} \in \sum_h \gamma_h \zeta(\mathcal{M}_h)} \langle \psi, \tilde{\rho} \rangle = \langle \psi, \lambda^* \rangle, \tag{4.50}$$

where the last equality holds since λ^* is at the boundary of the capacity region and $\tilde{\rho} \in \sum_h \gamma_h \zeta(\mathcal{M}_h)$ represents service rate in the system,whose inner product with ψ is maximized when it takes the value λ^*. From assumption (4.28), it follows that,

$$(\langle \psi, \lambda^n \rangle - \hat{\nu})nt \to b^* t.$$

The convergence of the processes $(\langle \psi, \hat{a}^n(t) - \lambda^n nt \rangle, t \geq 0)$ and $(\hat{x}^n(t) - \hat{\nu} nt, t \geq 0)$ to independent Brownian motions follows by Donsker's theorem [13]. This implies the result. □

Thus, we have established the weak convergence of the processes $\{\hat{u}^n(t), t \geq 0\}$. An almost sure version of the same convergence can be established by using the following result [98].

Lemma 4.5 (Skorohod Representation) *Let $\{X_n, n \geq 1\}$, X be random variables taking values in a separable metric space $(\mathscr{X}, d_{\mathscr{X}})$. If,*

$$X_n \xrightarrow{\mathscr{L}} X, \tag{4.51}$$

there exist $(\mathscr{X}, d_{\mathscr{X}})$ valued random variables $\{\tilde{X}_n, n \geq 1\}$, $\tilde{X}$ defined on a common underlying probability space such that,

$$\tilde{X} \overset{\mathscr{L}}{=} X, \tag{4.52}$$

$$\tilde{X}_n \overset{\mathscr{L}}{=} X_n, \ \forall \ n, \tag{4.53}$$

$$\tilde{X}_n \to \tilde{X} \ a.s., \tag{4.54}$$

where $X \overset{\mathscr{L}}{=} Y$ means that X and Y have the same distribution.

Using Lemma 4.5, one can construct a probability space where we have $\mathscr{D}[0, \infty)$ valued processes $\hat{u}^n_S$ and $\hat{u}_S$, such that, almost surely,

$$\hat{u}^n_S \to \hat{u}_S \ u.o.c.,$$

where $\hat{u}^n_S$ and $\hat{u}_S$ are identical in distribution to $\hat{u}^n$ and $\hat{u}$. Thus $\hat{u}_S$ is the Brownian motion given in (4.49). We augment this probability space to include the other components of Z as well. On this probability space, we will have the functions $\hat{v}^n$ and $\hat{w}^n$ as before. In this augmented probability space, we will prove the convergence of the $\hat{v}^n$ processes.

4.3.2 Convergence of $\hat{v}^n$

To prove the convergence of $\hat{v}^n$, we will require the following result, which is derived in [88] from the *weak estimates* of [17].

Lemma 4.6 *Let $z^n = (a^n, e^n, g^n, d^n, r^n, s^n, q^n)$ be the fluid scaled process, with components $a^n = (a_i^{f,n})_{i,f}$ and $e^n = (e_h^n)_{h\in\mathcal{H}}$. Let $\mathcal{N}_1$ be an arbitrary subsequence of $\mathcal{N}$. Then, there exists a further subsequence $\mathcal{N}_2$ of $\mathcal{N}_1$, such that almost surely, as $n \to \infty$ along $\mathcal{N}_2$, the fluid scaled process satisfies, for any $T > 0$, for all i, j, f, $c \in \mathcal{H}$,*

$$\max_{0\leq\ell\leq nT}\sup_{0\leq\epsilon\leq1} |a_i^{f,n}(\ell+\epsilon) - a_i^{f,n}(\ell) - \lambda_i^f\epsilon| \to 0, \tag{4.55}$$

$$\max_{0\leq\ell\leq nT}\sup_{0\leq\epsilon\leq1} |e_c^n(\ell+\epsilon) - e_c^n(\ell) - \gamma_c\epsilon| \to 0. \tag{4.56}$$

The proof is provided in the appendix 4.A.

Next, we will show that the limit of the processes $\{v^n(t), t \geq 0\}$ has a limit which satisfies certain conditions necessary for it to be the unique *regulator* corresponding to the Brownian motion $\hat{u}$. The relationship between a one dimensional Brownian motion and its regulator is given by the following result.

Lemma 4.7 (One dimensional Skorohod Problem) *Let $\xi \in \mathscr{D}[0,\infty)$, such that ξ is continuous, and $\xi(0) \geq 0$. Then there exists a unique pair of functions ξ_1, ξ_2, both in $\mathscr{D}[0,\infty)$ such that,*

1. *$\xi_1(t) = \xi(t) + \xi_2(t)$ for all $t \geq 0$,*
2. *$\xi_1(t) \geq 0$ for all $t \geq 0$,*
3. *$\xi_2(0) = 0$,*
4. *$\xi_2(t)$ is non negative, non decreasing and continuous,*
5. *for any $t \geq 0$, if $\xi_1(t) > 0$, then it is not a point of increase of $\xi_2(t)$.*

Further, this pair is given by,

$$\xi_2(t) = \sup_{0\leq\tau\leq t}(-\xi(\tau))^+,\ \xi_1(t) = \xi(t) + \xi_2(t),\ t \geq 0. \tag{4.57}$$

The proof of this result is given in the appendix 4.B.

If the process $\xi(t)$ is a sample path of a Brownian motion, $\xi_2(t)$ is called its regulator, and $\xi_1(t)$ is called the reflected (regulated) Brownian motion. It is clear that the proof of convergence of the processes $\{\hat{w}^n, t \geq 0\}$ to the reflected Brownian motion corresponding to the Brownian motion $\{\hat{u}(t), t \geq 0\}$ would involve showing the limit of the processes $\{\hat{v}^n, t \geq 0\}$ as ξ_2 satisfies property (4.57) with ξ being $\hat{u}$. This is done in the following theorem.

Theorem 4.2 *For any subsequence $\mathbb{N}_1$ of $\mathbb{N}$ as given in Theorem 4.1, there is a further subsequence $\mathbb{N}_2$ along which the processes $\{\hat{v}^n, t \geq 0\}$ has a limit $\hat{v} = \{\hat{v}, \geq 0\}$, which satisfies,*

1. *$\hat{v}(t)$ is continuous.*

2. *$\hat{v}(t)$ is finite for $t \in [0, \infty)$*

3. *$\hat{v}(0) = 0$*

4. *If $\hat{w}(t) > 0$, then t is not a point of increase of $\hat{v}$.*

The proof of this result is provided in the appendix 4.C.

Now we outline the proof of Theorem 4.1.

Proof: [Proof of Theorem 4.1] As explained in the previous section, from Lemma 4.4, using Lemma 4.5, one can construct a probability space where we have $\mathscr{D}[0, \infty)$ valued processes $\hat{u}_S^n$ and $\hat{u}_S$, such that, almost surely,

$$\hat{u}_S^n \to \hat{u}_S \ u.o.c.,$$

where $\hat{u}_S^n$ and $\hat{u}_S$ are identical in distribution to $\hat{u}^n$ and $\hat{u}$. Thus $\hat{u}_S$ is the Brownian motion given in (4.49). We augment this probability space to include the other components of Z as well. On this probability space, we will have the functions $\hat{v}^n$ and $\hat{w}^n$ as before.

Using Theorems 4.2 in combination with 4.7, we can see that $\hat{v}$ is the unique regulator corresponding to $\hat{u}$. Consequently, we see that the process $\hat{w}$ converges to a reflected Brownian motion.

What remains to be shown is that $\{\hat{q}^n, n \in \mathbb{N}\}$ converges weakly to $\phi\hat{w}$. This will follow if $\hat{q}^n$ converges to $\phi\hat{w}$ u.o.c.. For this, it suffices to show that for any $t \geq 0$ and $\epsilon > 0$, there exists a $\delta > 0$ such that,

$$\limsup_{n\to\infty} \sup_{\tau \in [t-\delta, t+\delta]_+} |\hat{q}^n(\tau) - \phi\hat{w}^n(\tau)| < \epsilon. \tag{4.58}$$

If this were true,the u.o.c. convergence can be obtained as follows. Let $\mathbb{C}$ be a compact set. Let ϵ be fixed. Then, for every $t \in \mathbb{C}$, there exists a δ_t such that (4.58) holds. Consider all sets of the form $(t - \frac{\delta}{2}, t + \frac{\delta}{2})$. These form an open cover for $\mathbb{C}$. Since the set is compact, there exists a finite subcover [75]. Therefore, there exists some finite number K such that, we have numbers $t_1, \ldots, t_K$ all from $\mathbb{C}$, such that,

$$\mathbb{C} \subset \cup_{i=1}^{K} \left(t_i - \frac{\delta_{t_i}}{2}, t_i + \frac{\delta_{t_i}}{2} \right). \tag{4.59}$$

Using this in combination with (4.58), the result follows. The result (4.143) implies (4.58). □

Now that we have established the existence of a limiting Brownian motion, we proceed to demonstrate that the stationary distributions of the scaled systems converge to the stationary distribution of the Brownian motion, in the next section.

4.4 Convergence of Stationary Distributions

We have the following result.

Theorem 4.3 *As $n \to \infty$,*

$$\hat{q}^n(\infty) \xrightarrow{\mathscr{L}} \phi \hat{w}(\infty), \tag{4.60}$$

where the time argument being infinity denotes the respective stationary distributions.

To prove this result, we first define a new set of fluid limit processes, given by,

$$\bar{z}^{n,r}(t) = \frac{Z^n(\lfloor rt \rfloor)}{r}. \tag{4.61}$$

Let $\bar{z}^{n,r} = (\bar{a}^{n,r}, \bar{e}^{n,r}, \bar{g}^{n,r}, \bar{d}^{n,r}, \bar{r}^{n,r}, \bar{s}^{n,r}, \bar{q}^{n,r})$, denote the process $(\bar{z}^{n,r}(t), t \geq 0)$, and $\bar{z}^n$ the fluid limit process obtained, for each n, by taking the limit $r \to \infty$. This limit exists just as in the previous section. For each Z^n, let π_n denote the stationary distribution of the corresponding network. These exist because for each n, the system Q^n is stable. The draining time (time for all queues to reach level zero) for the n-th fluid system will be denoted by τ^n_{drain}. We can see (from Sec. 4.2.1) that τ^n_{drain} is inversely proportional to the distance from the boundary of the capacity region Λ. It is also easy to see that, due to (4.28), the distance to the boundary of the capacity region, which is the plane whose normal vector is ψ, decreases as $\frac{1}{n}$. Hence,

$$\tau^n_{drain} \leq nT_1, \tag{4.62}$$

for some finite T_1, assuming that the initial fluid level is unity.

We will first state a result from [67].

Lemma 4.8 *Let $\{X_k, k \geq 1\}$ be a Markov chain with transition matrix P. Suppose there exists non negative functions $\Phi_1(x)$, $\Phi_2(x)$ and $\Phi_3(x)$ that satisfy, for all x,*

$$\int_x P(x, dy)\Phi_1(y) \leq \Phi_1(x) - \Phi_2(x) + \Phi_3(x), \tag{4.63}$$

then, for any stopping time $\mathcal{T}$,

$$\mathbb{E}_x[\sum_{k=0}^{\mathcal{T}-1} \Phi_2(X_k)] \leq \Phi_1(x) + \mathbb{E}_x[\sum_{k=0}^{\mathcal{T}-1} \Phi_3(X_k)]. \tag{4.64}$$

Now, we state a sufficient condition for the sequence $\{\pi_n, n \geq 0\}$ to be tight. Note that by writing $\hat{q}_x^n(\cdot)$ we indicate that the initial condition of the queue is x.

Lemma 4.9 *Assume that, for all nodes i, j, flows f, for any $n \geq 1$, $t \geq 0$, we have, for some $B < \infty$,*

$$\mathbb{E}[\sup_{0\leq k\leq t} |A_i^{f,n}(k) - \bar{a}_i^{f,n}(k)|^2] \leq Bt, \tag{4.65}$$

$$\mathbb{E}[\sup_{0\leq k\leq t} |R_i^{f,n}(k) - \bar{r}_i^{f,n}(k)|^2] \leq Bt, \tag{4.66}$$

$$\mathbb{E}[\sup_{0\leq k\leq t} |D_i^{f,n}(k) - \bar{d}_i^{f,n}(k)|^2] \leq Bt. \tag{4.67}$$

Further, assume that there exists T such that for all $t \geq T$, we have,

$$\lim_{|x|\to\infty} \sup_n \frac{1}{|x|^2}\mathbb{E}|\hat{q}_x^n(t|x|)|^2 = 0. \tag{4.68}$$

Then the sequence of distributions $\{\pi_n\}$ is tight.

The result is an adaptation of the techniques in [20] to our case. We give an outline of the proof below.

Proof: From (4.68), it follows that there exists M, $0 < M < \infty$, such that, with $\mathbb{D} = \{x : |x| < M\}$, for all $x \notin \mathbb{D}$,

$$\sup_n \mathbb{E}||\hat{q}_x^n(T|x|)|^2 \leq \frac{|x|^2}{2}. \tag{4.69}$$

Define $\delta = TM$ and $\tau^n(\delta) = \inf\{t \geq \delta : |\hat{q}_x^n(t)| \leq M\}$. Define a sequence of stopping times,

$$\mathcal{T}_0 = 0,\ \mathcal{T}_m = \mathcal{T}_{m-1} + T\max(|\hat{q}_x^n(\mathcal{T}_{m-1})|, M). \tag{4.70}$$

Define,

$$m_n^* = \min\{m \geq 1 : |\hat{q}_x^n(\mathcal{T}_m)| \leq M\}. \tag{4.71}$$

Define,

$$\hat{V}_n(x) = \mathbb{E}[\int_0^{\tau^n(\delta)} (1 + |\hat{q}_x^n(t)|)dt], \tag{4.72}$$

where It follows that,

$$\hat{V}_n(x) \leq \mathbb{E}[\int_0^{\mathcal{T}_{m_n^*}} (1 + |\hat{q}_x^n(t)|)dt] = \sum_{k=0}^{\infty} \mathbb{E}[\int_{\mathcal{T}_k}^{\mathcal{T}_{k+1}} (1 + |\hat{q}_x^n(t)|)dt \mathbf{1}_{\{k<m_n^*\}}]. \tag{4.73}$$

Define the filtration $\mathscr{F}_t$ as the sigma algebra generated by $\{\hat{q}_x^n(s) < 0 \leq s \leq t\}$. It can be shown that (see appendix 4.D for proof) there exists a finite non negative constant c_0 such that, for all n, k, x, we have,

$$\mathbb{E}[\int_{\mathcal{T}_k}^{\mathcal{T}_{k+1}} (1 + |\hat{q}_x^n(t)|)dt|\mathscr{F}_{\mathcal{T}_k}]\mathbf{1}_{\{k<m_n^*\}} \leq c_0(1 + |\hat{q}_x^n(\mathcal{T}_k)|^2)\mathbf{1}_{\{k<m_n^*\}}. \tag{4.74}$$

Using this, one obtains the estimate,

$$\sup_n \hat{V}_n(x) \leq c_0 \sup_n \mathbb{E}[\sum_{k=0}^{m_n^*-1} (1 + |\hat{q}_x^n(\mathcal{T}_k)|^2)]. \tag{4.75}$$

Observe that the Markov chain $\{\hat{q}_x^n(\mathcal{T}_m), m \geq 1\}$ has the single step transition kernel,

$$P_n(x, A) = \hat{P}_n^{T \max(|x|,M)}(x, A), \tag{4.76}$$

where $\hat{P}_n^t$ was the transition kernel of $\hat{q}^n$. Using (4.68) and (4.69), we can write, for some finite positive B,

$$sup_n \int_x P_n(x, dy)|y|^2 \leq |x|^2 - \frac{|x|^2}{2} + B\mathbf{1}_{[1,M]}(|x|). \tag{4.77}$$

Using this in Lemma 4.8, and plugging in the bound obtained in (4.75), we see that, for all x,

$$\sup_n \int_0^{\tau^n(\delta)} (1 + |\hat{q}_x^n(t)|)dt \leq c(1 + |x|^2). \tag{4.78}$$

It can be shown (for proof see the appendix 4.E) that, we see that there exists a positive $\kappa < \infty$ such that, for all t, x and n,

$$\frac{\mathbb{E}[\hat{V}_n(\hat{q}^n_x(t))]}{t} + \frac{\int_0^t \mathbb{E}(1 + |\hat{q}^n_x(s)|)ds}{t} \leq \frac{\hat{V}_n(x)}{t} + \kappa. \tag{4.79}$$

Define the functions,

$$V^k_n(x) = \min(\hat{V}_n(x), k), \tag{4.80}$$

$$\Gamma^k_n(x) = \frac{1}{t}(V^k_n(x) - \mathbb{E}[V^k_n(\hat{q}^n_x(t))]), \tag{4.81}$$

$$\Gamma_n(x) = \frac{1}{t}(\hat{V}_n(x) - \mathbb{E}[\hat{V}_n(\hat{q}^n_x(t))]). \tag{4.82}$$

Now, $\Gamma^k_n(x) \to \Gamma_n(x)$ as $k \to \infty$, by the monotone convergence theorem. Also, since π_n is the invariant measure of the n-th system, we have,

$$\int_x \Gamma^k_n(x)\pi_n(dx) = 0. \tag{4.83}$$

By an application of Fatou's Lemma, we can see that,

$$\int_x \Gamma_n(x)\pi_n(dx) \leq \liminf_{k\to\infty} \int_x \Gamma^k_n(x)\pi_n(dx) = 0. \tag{4.84}$$

If $\hat{V}_n(x) \leq k$, from (4.79), we know that,

$$\Gamma^k_n(x) \geq -\kappa. \tag{4.85}$$

Ih $\hat{V}_n(x) > k$, we have,

$$\Gamma^k_n(x) \geq 0. \tag{4.86}$$

Hence, $\Gamma^k_n(x) \geq -\kappa$ for all x. From (4.79), we can see that,

$$\Gamma_n(x) \geq \frac{\int_0^t \mathbb{E}(1 + |\hat{q}^n_x(s)|)ds}{t} - \kappa. \tag{4.87}$$

Thus we obtain the bound,

$$\int_x \Gamma_n(x)\pi_n(dx) \geq \frac{\int_0^t \int_x \mathbb{E}(1+|\hat{q}_x^n(s)|)\pi_n(dx)ds}{t} - \kappa. \tag{4.88}$$

Combining with (4.84), and noting that the systems are assumed to be stationary, we obtain,

$$\int_x \mathbb{E}(1+|\hat{q}_x^n(t)|)\pi_n(dx) \leq \kappa. \tag{4.89}$$

Since π_n is the invariant measure for the n-th system, this is equivalent to,

$$\int_x (1+|x|)\pi_n(dx) \leq \kappa. \tag{4.90}$$

Let ϵ be fixed. Let $\mathbb{M} = \{x : |x| \leq M\}$, for some $M > \frac{\kappa}{\epsilon} - 1$. Then,

$$\int_{x \notin \mathbb{M}} (1+|x|)\pi_n(dx) \geq (1+M)\pi_n(\mathbb{M}^c). \tag{4.91}$$

Using (4.90), we have that,

$$\pi_n(\mathbb{M}^c) \leq \frac{\kappa}{1+M} < \epsilon, \tag{4.92}$$

by our choice of M. Since this is true for all n, it implies that the sequence of probability measures $\{\pi_n, n \geq 1\}$ is tight. □

Lemma 4.10 *In our system model, conditions (4.65)-(4.67) hold. Further, there exists T such that (4.68) holds. Consequently, the sequence $\{\pi_n\}$ is tight.*

Proof: Since the process $\{A_i^{f,n}(t) - a_i^{f,n}(t), t \geq 0\}$ is a martingale, we can use Doob's inequality [3] to obtain,

$$\begin{aligned}
\mathbb{E}[\sup_{0\leq k\leq t} |A_i^{f,n}(s) - \bar{a}_i^{f,n}(s)|^2] &\leq B_1^{'}\mathbb{E}|A_i^{f,n}(t) - \bar{a}_i^{f,n}(t)|^2, \\
&\leq B_1^{'} t\mathbb{E}|A_i^{f,n}(1) - \bar{a}_i^{f,n}(1)|^2, \\
&= B_1 t,
\end{aligned}$$

where the second inequality follows from the i.i.d nature of the arrival process [36]. Hence, (4.65) holds.

The bounds for R and D would hold if a corresponding bound holds for the S_{ij}^f processes.

Define the slotwise allocation process $\bar{S}^f_{ij}$, where,

$$S^f_{ij}(t) = \sum_{t^{'}=1}^{t} \bar{S}^f_{ij}(Q(t^{'}), H(t^{'})),$$

since $\bar{S}^f_{ij}$ depends on both the queue state at time t, and the channel state at time t. Let $\mathcal{S}$ be the set of possible values $S(t)$ can take. Since $\mathcal{H}$ is finite (and consequently, $\mathcal{S}$), there are only a finite set of mappings from $\mathcal{H}$ to $\mathcal{S}$. This set of mappings will be denoted by $\{\mathbb{F}_1, \ldots, \mathbb{F}_{K_1}\}$. Each $S(Q(t), H(t))$ will take the value of one of these functions. It is easy to see that the state space of queues can be partitioned as,

$$\mathcal{Q} = \cup_{m=1,\ldots,K_1} \mathcal{Q}_m, \tag{4.93}$$

where, if $Q(t) \in \mathcal{Q}_m$, we have $S(Q(t), H(t)) = \mathbb{F}_m(H(t))$, and the $\mathcal{Q}_m$ are disjoint. Now we can write,

$$S^f_{ij}(t) = \sum_{t^{'}=1}^{t} \sum_{m=1}^{K_1} \mathbb{F}_m(H(t)) \mathbf{1}_{\{Q(t)=m\}}, \tag{4.94}$$

where $\mathbf{1}$ is the indicator function. Rewrite this as,

$$S^f_{ij}(t) = \sum_{m=1}^{K_1} \sum_{k \in \hat{T}_m(t)} \mathbb{F}_m(H(k)), \tag{4.95}$$

where $\hat{T}_m(t)$ is the set of time slots till t when the queue state was in $\mathcal{Q}_m$. Since the system is stationary, we can also obtain,

$$s^f_{ij}(t) = \mathbb{E}[S^f_{ij}(t)]. \tag{4.96}$$

Thus, we may write, with $\bar{\mathbb{F}}_m = \mathbb{E}[\mathbb{F}_m(H(1))]$,

$$|S^f_{ij}(t) - s^f_{ij}(t)|^2 \leq B_2^{'} \sum_{m=1}^{K_1} \left| \sum_{k \in \hat{T}_m(t)} \mathbb{F}_m(H(k)) - \bar{\mathbb{F}}_m \right|^2,$$

where $B_2^{'}$ depends only on K_1. For any m, along $k \in \hat{T}_m(t)$, $\mathbb{F}_m(H(k))$ is an i.i.d sequence.

Therefore, proceeding similar to what was done for A, we now obtain,

$$\mathbb{E}[\sup_{0\leq k\leq t}|S_{ij}^{f}(k)-s_{ij}^{f}(k)|^{2}]\leq B_{2}\mathbb{E}[\sum_{m}|\hat{T}_{m}(t)|]=B_{2}t,$$

where the equality follows, since $\sum_{m}|\hat{T}_{m}(t)|=t$. Hence the bounds hold for R and D as well. Hence (4.65)-(4.67) hold, choosing $B=\max\{B_{1},B_{2}\}$.

To show (4.68), observe that, for a particular queue Q_{i}^{f}, it follows from the queueing equation that,

$$\begin{aligned}n\hat{q}_{i}^{f,n}(t)&=Q_{i}^{f,n}(n^{2}t),\\&=Q_{i}^{f,n}(0)+A_{i}^{f,n}(n^{2}t)+R_{i}^{f,n}(n^{2}t)-D_{i}^{f,n}(n^{2}t).\end{aligned}$$

Subtracting on either side with the corresponding fluid queue $q_{i}^{f,n}(t^{'})$ at time $t^{'}=n^{2}t$, we obtain,

$$\begin{aligned}Q_{i}^{f,n}(n^{2}t)-\bar{q}_{i}^{f,n}(n^{2}t)=Q_{i}^{f,n}(0)-\bar{q}_{i}^{f,n}(0)+A_{i}^{f,n}(n^{2}t)\\-\bar{a}_{i}^{f,n}(n^{2}t)+R_{i}^{f,n}(n^{2}t)-r_{i}^{f,n}(n^{2}t)\\-D_{i}^{f,n}(n^{2}t)+\bar{d}_{i}^{f,n}(n^{2}t).\end{aligned}$$

Hence, we have,

$$\begin{aligned}|Q_{i}^{f,n}(n^{2}t)-\bar{q}_{i}^{f,n}(n^{2}t)|^{2}\leq C(&|Q_{i}^{f,n}(0)-\bar{q}_{i}^{f,n}(0)|^{2}\\&+|A_{i}^{f,n}(n^{2}t)-\bar{a}_{i}^{f,n}(n^{2}t)|^{2}\\&+|R_{i}^{f,n}(n^{2}t)-\bar{r}_{i}^{f,n}(n^{2}t)|^{2}\\&+|D_{i}^{f,n}(n^{2}t)-\bar{d}_{i}^{f,n}(n^{2}t)|^{2}).\end{aligned}$$

Choosing $Q_{i}^{f,n}(0)=\bar{q}_{i}^{f,n}(0)$, we obtain, using (4.65)-(4.67),

$$\mathbb{E}|Q_{i}^{f,n}(n^{2}t)-\bar{q}_{i}^{f,n}(n^{2}t)|^{2}\leq C_{2}n^{2}t, \tag{4.97}$$

and hence it follows for the vector process Q, with a higher constant $C_{2}^{'}$,

$$\mathbb{E}|Q^{n}(n^{2}t)-\bar{q}^{n}(n^{2}t)|^{2}\leq C_{2}^{'}n^{2}t. \tag{4.98}$$

From (4.62), since the draining time of the fluid system $\bar{q}^{n}$ with initial condition equal to one,

$\tau^n_{drain} \leq nT_1$, the fluid system with initial condition x, will be zero at any time greater than $\tau^n_{drain}|x|$. Setting $t \geq T_1|x|$, and dividing by n^2, we get,

$$\mathbb{E}|\hat{q}^n_x(t|x|)|^2 \leq C_2^{'} t|x|. \tag{4.99}$$

Since the bound is uniform over n, dividing by $|x|^2$ and taking $|x| \to \infty$ gives the result. □
With this result, we are ready to prove Theorem 4.3.

Proof: [Proof of Theorem 4.3] Since the π_n are tight, any subsequence of π_n has a convergent subsequence. Let such a limit point be π^*. On the convergent subsequence, assume that the initial conditions $\hat{Z}^n(0)$ are distributed as π_n. Since the systems $\hat{Z}^n$ converge to a reflected Brownian motion (RBM), the initial condition of the RBM $\hat{w}$ will have distribution π^*. Also, we have shown that finite dimensional distributions of $\hat{z}^n$ also converge to that of $\hat{w}$. In particular, $\hat{z}^n(t)$ weakly converges to $\hat{w}(t)$ for any $t \geq 0$. But the distribution of $\hat{z}^n(t)$ is π_n. Thus distribution of $\hat{w}(t)$ is π^* for each t. Hence π^* is the stationary distribution of $\hat{w}$. □

The Brownian motion $\hat{w}$ obtained as the limit of $\hat{w}^n$ is a unidimensional reflected Brownian motion, having drift $b^* < 0$. If $\hat{w}(\infty)$ has the stationary distribution of $\hat{w}$, from [41],

$$\mathbb{P}[\hat{w}(\infty) < y] = 1 - \exp(2b^* y/\sigma^2). \tag{4.100}$$

4.5 Numerical Simulations

We will consider two example networks.

Example 1. Consider a star network topology (Figure 4.2). There are two Poisson distributed arrival processes, one arriving at node 1, with node 4 as its destination. The other arrives at node 2, with node 5 as destination. We will also assume that two links which share a common node interfere with each other. We assume that the channels are independent and identically distributed, with the distribution being uniform over the set $\{0, 1, 2, 3\}$. We consider the arrival vector $(\lambda_1, \lambda_2) = (\lambda, \lambda)$, i.e., increasing along the line of unit slope. In this case $\lambda^* = (0.65, 0.65)$. From the diffusion approximation and (4.100), we can see that the mean of the Brownian motion corresponding to the queue can be approximated by the vector $\phi \frac{\sigma^2}{2b^*}$. The Brownian motion is a limit of the scaled process of the form $\frac{Q(n^2 t)}{n}$. For a large n, we may approximately write,

$$Q(n^2 t) \approxeq n\phi \frac{\sigma^2}{2b^*}.$$

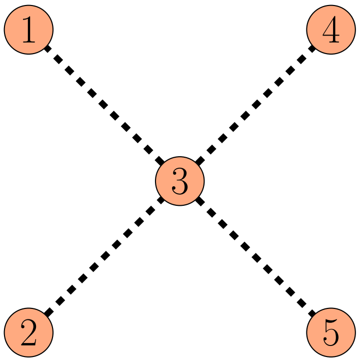

Figure 4.2: Example 1: The Network

If we run the simulations for a time n, we may further also approximately write $b^* = n|\lambda - \lambda^*|$. Hence, we have the approximation,

$$Q(\infty) \approxeq \phi \frac{\sigma^2}{2|\lambda - \lambda^*|}. \tag{4.101}$$

We will be looking at the total queue length of the flow $1 \to 3 \to 4$. The value of σ^2 is $2\lambda + \hat{\sigma}^2$. The vector ϕ is approximately $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ (The value of $\bar{Q}$ for both queues is set at 100). We take $\hat{\sigma}^2 \approxeq 8$. The values of the total queue length of the flow $1 \to 3 \to 5$ are listed in Table 4.1 (owing to symmetry both queue lengths are same), for simulation runs of length 10^5, averaged over 20 simulations. It can be seen that the approximations follow the queue length closely.

In order to demonstrate that the algorithm can satisfy different QoS requirements, we simulate the network at three points in the interior of the capacity region. The mean queue length asked from the flows is 250 and 100 respectively. We also pick a_2 in the expression of α for the second flow to be 4, since it requires a tighter constraint to be met. In Table 4.2, the

Arrival Rate λ	Mean Queue Length	Approximation
0.64	233	232
0.641	263	258
0.642	319	290
0.643	367	332
0.644	381	387
0.645	479	465
0.646	517	581
0.647	568	775

Table 4.1: Approximation of Queues. The mean queue length of the flow $1 \to 3 \to 5$ corresponding to various arrival rates is displayed, along with the numerical approximation.

first column gives the arrival rate, the second shows the target queue length for the two flows, and the final column shows the queue length obtained. We see that the end-to-end mean queue length requirement is met for both the flows till rate 0.64. The capacity boundary is at 0.65. Thus, our algorithm can provide QoS under heavy traffic as well.

λ	Mean Queue Length Asked	Queue Length Obtained
0.63	(250,100)	(213,98)
0.64	(250,100)	(264,110)
0.641	(250,100)	(292,120)

Table 4.2: Mean Queue Length Target and Obtained, for both flows.

Example 2. Consider the network in Figure 4.3. The arrival process, channel state distribution and interference constraints are the same as in Example 1. There are three flows, $1 \to 3 \to 4 \to 6 \to 8$, $2 \to 3 \to 4 \to 5$ and $7 \to 4 \to 6 \to 9$. They will be called Flow 8, Flow 5 and Flow 9. The boundary of the capacity region, $\lambda^* \approx (0.59, 0.59, 0.01)$. We take arrival rates close to this point and show the values of total queue length of Flow 8 obtained by simulations and the numerical approximations (using (4.101)), in Table 4.3. For calculating the approximation, we use $\hat{\sigma}^2 \approx 9$. In this case also, the approximations track the queue lengths well. Just as in the previous case, we provide an example to show how the queue length values meet targets, in Table 4.4. These are simulated at the arrival rate $(0.55.0.55, 0.01)$, which is in the interior of the capacity region. In the weight function α, we use $a_1 = 5, a_2 = 1$ to give weights to flows.Since flows 8 and 5 are competing for network resources; delays of both cannot be reduced simultaneously. This is also clear from the simulations.

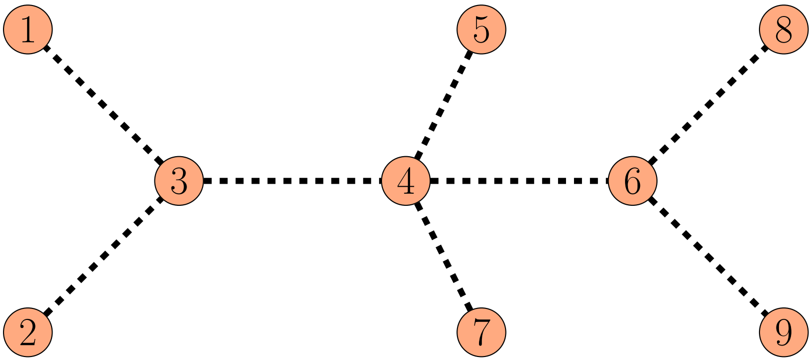

Figure 4.3: Example 2: The Network

Table 4.3: Entries of the form (a,b) indicate delay target a, delay achieved b.

Arrival Rate λ	Mean Queue Length	Approximation
0.5	21	26
0.54	52	47
0.56	99	79
0.57	119	144
0.58	253	239
0.582	331	299
0.584	403	399
0.585	457	479

Table 4.4: Entries of the form (a,b) indicate delay target a, delay achieved b. Arrival rate is $(0.55.0.55, 0.01)$.

Mean Delay(slots) for each flow		
Flow 8	Flow 5	Flow 9
(50,52)	(100,112)	9
(40,46)	(100,114)	9
(100,139)	(50,53)	21

4.6 Conclusion

We have presented an algorithm, similar to that of Chapter 3, for scheduling in multihop wireless networks that guarantees end-to-end mean delays of the packets transmitted in the network. The algorithm is throughput optimal. Using diffusion scaling, we obtain the Brownian approximation of the algorithm. We also prove theoretically that the stationary distribution of the limiting Brownian motion is the limit of stationary distributions of a sequence of scaled systems, and is consequently a good approximation for the stationary distribution of the original system. Using these relations, we obtain an approximation for queue lengths, and demonstrate via simulations that these are accurate.

4.A Proof of Lemma 4.6

It suffices to show the result for the a process; the result for e follows similarly.

First, we establish a Lemma due to [17].

Lemma 4.11 *Let $\{X_n, n \geq 1\}$ be a sequence of i.i.d. random variables with mean μ, such that they satisfy the tail condition,*

$$\mathbb{E}[(X_1)^2 \mathbf{1}_{\{|X|>x\}}] \leq \phi(x), \tag{4.102}$$

where,

$$\phi(x) \to 0 \text{ as } x \to \infty. \tag{4.103}$$

Define $S_n = \sum_{i=1}^n X_i$. Then, for fixed $\delta > 0$ and large enough n,

$$\mathbb{P}(\max_{1\leq i\leq n} |S_i - i\mu| \geq \delta n) \leq \frac{\delta}{n}. \tag{4.104}$$

Proof: For each n, write X_n as,

$$X_n = Y_n + Z_n, \tag{4.105}$$

where, $\{Y_n.n \geq 1\}$ and $\{Z_n.n \geq 1\}$ are i.i.d., given by,

$$Y_n = X_n \mathbf{1}_{\{|X_n|\leq M\}}, \tag{4.106}$$

$$Z_n = X_n \mathbf{1}_{\{|X_n|> M\}}, \tag{4.107}$$

for some finite $M > 0$. Let $\mu_Y = \mathbb{E}[Y_1]$, $\mu_Z = \mathbb{E}[Z_1]$. Thus, we have,

$$S_n = S_n^Y + S_n^Z, \tag{4.108}$$

where $S_n^Y = \sum_{i=1}^n Y_i$ and $S_n^Z = \sum_{i=1}^n Z_i$. We can write,

$$\mathbb{P}(|S_i - i\mu| \geq \delta n) = \mathbb{P}(|S_i^Y + S_i^Z - i\mu_Y - i\mu_Z| \geq \delta n), \tag{4.109}$$

$$\leq \mathbb{P}(|S_i^Y - i\mu_Y| \geq \frac{\delta n}{2}) + \mathbb{P}(|S_i^Z - i\mu_Z| \geq \frac{\delta n}{2}). \tag{4.110}$$

Using Chebyshev's inequality, one obtains the bounds,

$$\mathbb{P}(|S_i^Y - i\mu_Y| \geq \frac{\delta n}{2}) \leq 2^4 \frac{\mathbb{E}[|S_i^Y - i\mu_Y|^4]}{\delta^4 n^4} \leq 2^4 \frac{i^2 2^4 M^4}{\delta^4 n^4} \leq 2^8 M^4 \delta^{-4} n^{-2}, \tag{4.111}$$

$$\mathbb{P}(|S_i^Z - i\mu_Z| \geq \frac{\delta n}{2}) \leq 2^2 \frac{\mathbb{E}[|S_i^Z - i\mu_Z|^2]}{\delta^2 n^2} \leq 2^2 \frac{\phi(M)}{\delta^2 n^2} = 2^2 \phi(M) \delta^{-2} n^{-2}, \tag{4.112}$$

where the second line of inequalities used the tail condition (4.103). Choosing $M = n^{\frac{1}{8}}$, we obtain,

$$\mathbb{P}(|S_i - i\mu| \geq \delta n) \leq \frac{1}{n}(2^8 \delta^{-4} n^{-\frac{1}{2}} + 2^2 \phi(n^{\frac{1}{8}}) \delta^{-2} n^{-2}). \tag{4.113}$$

Choosing n large enough, we obtain,

$$\mathbb{P}(|S_i - i\mu| \geq \delta n) \leq \frac{\delta}{n}. \tag{4.114}$$

The result follows from this. □

Now, consider $A_i^f(n(\ell + \epsilon)) - A_i^f(n\ell)$, for $\epsilon \in (0, 1]$. This process has i.i.d. increments at times when $n(\ell + \epsilon)$ takes integer values. Let L_ℓ denote the corresponding values of ϵ. Using Lemma 4.11, we may write,

$$\mathbb{P}(\max_{\epsilon \in L_\ell} |A_i^f(n(\ell + \epsilon)) - A_i^f(n\ell) - \lambda_i^f m_\epsilon| \geq \delta \lceil n \rceil) \leq \frac{\delta}{\lceil n \rceil}, \tag{4.115}$$

where m_ϵ is the size of the set of all elements of L_ℓ smaller than or equal to ϵ. This is equivalent to,

$$\mathbb{P}(\max_{\epsilon \in L_\ell} |a_i^{f,n}(\ell + \epsilon) - a_i^{f,n}(\ell) - \lambda_i^f \frac{m_\epsilon}{n}| \geq \delta \frac{\lceil n \rceil}{n}) \leq \frac{\delta}{\lceil n \rceil}. \tag{4.116}$$

Since $\frac{m_\epsilon}{n}$ on $\epsilon \in (0, 1]$ converges uniformly to ϵ, we can write the above as,

$$\mathbb{P}(\sup_{0 \leq \epsilon \leq 1} |a_i^{f,n}(\ell + \epsilon) - a_i^{f,n}(\ell) - \lambda_i^f \epsilon| \geq \delta') \leq \frac{\delta}{\lceil n \rceil} < \frac{\delta'}{n}, \tag{4.117}$$

where $\delta' = \delta \frac{\lceil n \rceil}{n}$. By means of a union bound, we can se that,

$$\mathbb{P}(\max_{0 \leq \ell \leq nT} \sup_{0 \leq \epsilon \leq 1} |a_i^{f,n}(\ell + \epsilon) - a_i^{f,n}(\ell) - \lambda_i^f \epsilon| \geq \delta') < (nT + n) \frac{\delta'}{n} = (T + 1)\delta'. \tag{4.118}$$

Given a subsequence $\mathcal{N}_1$ of $\mathcal{N}$, choose a further subsequence $\mathcal{N}_2$ along which, there is a sequence $T(n)$ and $\delta(n)$ such that, (4.118) is satisfied with $T = T(n)$ and $\delta^{'} = \delta(n)$ and,

$$\sum_{n=1}^{\infty}(T(n)+1)\delta(n) < \infty. \tag{4.119}$$

From the Borel-Cantelli Lemma [3], it follows that almost surely, as $n \to \infty$ along $\mathcal{N}_2$,

$$\max_{0\leq\ell\leq nT}\sup_{0\leq\epsilon\leq 1}|a_i^{f,n}(\ell+\epsilon) - a_i^{f,n}(\ell) - \lambda_i^f\epsilon| \to 0. \tag{4.120}$$

4.B Proof of Lemma 4.7

Define

$$\xi_2(t) = \sup_{0 \le \tau \le t} (-\xi(\tau))^+. \tag{4.121}$$

It is easy to see that $\xi_2(t)$ is continuous, since $\xi(t)$ is continuous. Since

$$(-\xi(t))^+ \ge 0 \ \forall t, \tag{4.122}$$

we can see that $\xi_2(t) \ge 0$ for all t. Also, from the definition of ξ_2, it is easy to see that $\xi_2(t)$ is a non decreasing function and that $\xi_1(t) = \xi(t) + \xi_1(t) \ge 0$ for all t. Thus we obtain properties $1-4$ of Lemma 4.7. What remains to be shown is property 5, and that ξ_2 is unique (give ξ).

Assume that $\xi_1(t) > 0$. Observe that, if t is a point of increase of $\xi_2(t)$, it would imply that,

$$\arg_{0 \le \tau \le t} sup(-\xi(\tau))^+ = t, \tag{4.123}$$

and hence, $\xi_2(t) = -\xi(t)$. Consequently,

$$\xi_1(t) = \xi(t) + \xi_2(t) = 0, \tag{4.124}$$

which is a contradiction. Hence property 5 of Lemma 4.7 follows.

To show uniqueness, assume there exists a pair of functions $\tilde{\xi}_1$ and $\tilde{\xi}_2$ that satisfy conditions $1-5$ of Lemma 4.7. Then, define,

$$\xi_0(t) = \xi_1(t) - \tilde{\xi}_1(t) = \xi_2(t) - \tilde{\xi}_2(t). \tag{4.125}$$

Clearly, $\xi_0(t)$ is the difference of two non decreasing functions. Thus, it is differentiable almost everywhere [70]. We can write,

$$\begin{aligned}
0 \le \frac{1}{2}(\xi_1(t) - \tilde{\xi}_1(t))^2 &= \int_0^t (\xi_1(\tau) - \tilde{\xi}_1(\tau)) d(\xi_1(\tau) - \tilde{\xi}_1(\tau)) && (4.126)\\
&= \int_0^t (\xi_1(\tau) - \tilde{\xi}_1(\tau)) d(\xi_1(\tau) - \tilde{\xi}_1(\tau)), && (4.127)\\
&= \int_0^t (\xi_1(\tau) - \tilde{\xi}_1(\tau)) d(\xi_2(\tau) - \tilde{\xi}_2(\tau)), && (4.128)\\
&= \int_0^t \xi_1(\tau) d\xi_2(\tau) + \tilde{\xi}_1(\tau) d\tilde{\xi}_2(\tau) - \xi_1(\tau) d\tilde{\xi}_2(\tau) - \tilde{\xi}_1(\tau) d\xi_2(\tau). && (4.129)
\end{aligned}$$

By property 5, we have that,

$$\xi_1(t)d\xi_2(t) = 0, \tag{4.130}$$

$$\tilde{\xi_1}(t)d\tilde{\xi_2}(t) = 0. \tag{4.131}$$

Substituting in (4.129), and noting that $\xi_1(t)$, $\tilde{\xi_1}(t)$, $d\tilde{\xi_2}(t)$ and $d\xi_2(t)$are all non negative, we see that,

$$0 \leq \frac{1}{2}(\xi_1(t) - \tilde{\xi_1}(t))^2 \leq 0. \tag{4.132}$$

Hence ξ_1 and $\tilde{\xi_1}$ (and consequently ξ_2 and $\tilde{\xi_2}$) are identical.

4.C Proof of Theorem 4.2

We need to show that, along the subsequence $\mathcal{N}_2$, we have a limit $\hat{v}$ of $\hat{v}^n$, which has the properties:

1. $\hat{v}(t)$ is continuous.
2. $\hat{v}(t)$ is finite for $t \in [0, \infty)$
3. $\hat{v}(0) = 0$
4. If $\hat{w}(t) > 0$, then t is not a point of increase of $\hat{v}$.

To prove these properties, we need to study a set of fluid sample paths.

Rescaled Fluid Paths

To study diffusion properties on an interval $[t_n, t_n + \delta]$ for $\delta > 0$, we look at fluid paths on the time $[nt_n, nt_n + n\delta]$. We consider the following family of fluid paths, started at a time T apart from each other. For a time evolving process $f(t)$, define the operator $\Theta(\tau)$ as the shift, corresponding to the process started at time τ.

Consider the fluid scaled process z^n. Consider a shifted form of these processes,

$$\tilde{z}^{m,l} = \Theta(mt_m + Tl)z^m, \tag{4.133}$$

where $\Theta(x)\xi$ denotes the function ξ started at x. Define the family of processes,

$$\mathcal{Z} = \{\tilde{z}^{m,l(m)}, m \in \mathcal{N}_3\}, \tag{4.134}$$

where the index set $\mathcal{N}_3$ has the property that as $m \to \infty$ along $\mathcal{N}_3$, $t_m \to t$. Using these fluid paths we can obtain properties of the diffusion scaled process, since an interval of time $[mt, mt + m\delta]$ on the diffusion scale corresponds to a time $[t, t + \delta]$ on the diffusion scale.

If $t_m \to t$, and $l(m) \in [0, 2\delta m/T - 1]$, a time $s \in [0, T]$ for the path $\tilde{z}(m, l(m))$, for m large enough, corresponds to a time,

$$s^{'} = t_m + l(m)T/m + s/m \in [t - 3\delta, t + 3\delta]^+. \tag{4.135}$$

We have the following results regarding the behaviour of the fluid sample paths, from [88]. The first is presented without proof.

Lemma 4.12 *Consider the family $\mathcal{Z}$ with an associated sequence t_m, constants T and δ, both positive. Assume that $|\tilde{q}^{m,l(m)}| \in [c_1, c_2]$, with $0 \leq c_1 \leq c_2 < \infty$, and $l(m) \in [0, 2\delta m/T - 1] \cap \mathbb{Z}$. Then, there is a subsequence m_k along which, $\tilde{z}^{m,l(m)}$ converges to a fluid limit z, u.o.c, with $|q(0)| \in [c_1, c_2]$.*

Recall the Lyapunov function $\mathcal{L}_1(q(t))$ defined in the proof of Lemma 3.4. This function is non negative, finite and its time derivative is negative. If, along $q(t)$, if $\lim_{t\to\infty} \mathcal{L}_1(q(t)) = 0$, define $\mathcal{L}_2 = \mathcal{L}_1$. Else, if $\lim_{t\to\infty} \mathcal{L}_1(q(t)) = \mathcal{L}_* > 0$, define $\mathcal{L}_2(q(t)) = \frac{\mathcal{L}_1(q(t))}{\mathcal{L}_*} - 1$. Clearly, $\mathcal{L}_2(q(t))$ decreases to zero along any fluid path. Let β be a universal constant (denoted as κ in (60) of [88]). Then, we have the following result.

Lemma 4.13 *Under our scheduling policy, assume that there is a subsequence such that, along this, $\hat{v}^n \to \hat{v}$. Suppose further that along this subsequence, we have*

$$s_m \to s \geq 0, \hat{w}^m(s_m) \to K > 0, \tag{4.136}$$

$$\limsup_{m\to\infty} |\hat{q}^m(s_m)| < K_1 K, \tag{4.137}$$

for some fixed $K_1 > 1$. Let $\delta > 0$ be chosen such that,

$$\epsilon = O_{\hat{u}}([s - 3\delta, s + 3\delta]^+) < 0.5K, \tag{4.138}$$

where $O_{\hat{u}}[a,b] = \sup_{x,y\in[a,b]} |u(x) - u(y)|$. Let $K_2 = \beta^2 K_1 K + 2\epsilon$. Then, for any $\epsilon_2 > 0$ sufficiently small, there exists a time T such that, for m sufficiently large, we have,

$$K - 2\epsilon < \tilde{w}^{m,0}(u) < K_2, \text{ for } \ u \in [0,T], \tag{4.139}$$

$$(K - 2\epsilon)/\beta < |\tilde{q}^{m,0}(u)| < 2\beta K_2. \tag{4.140}$$

For $l \in [1, 2\delta r T^{-1} - 1] \cap \mathbb{Z}$, we have,

$$\mathcal{L}_2(\tilde{q}^{m.l}(0)) < 2\epsilon_2, \tag{4.141}$$

$$\mathcal{L}_2(\tilde{q}^{m.l}(T)) < 2\epsilon_2, \tag{4.142}$$

$$\mathcal{L}_2(\tilde{q}^{m.l}(u)) < 3\epsilon_2, \text{ for } \ u \in [0,T], \tag{4.143}$$

$$\tilde{v}^{m,l}(u) = \tilde{v}^{m,l}(u) - \tilde{v}^{m,l}(0) = 0, \text{ for } \ u \in [0,T], \tag{4.144}$$

$$K - 2\epsilon < \tilde{w}^{m,l}(u) < K_2, \text{ for } \ u \in [0,T], \tag{4.145}$$

$$(K - 2\epsilon)/\beta < |\tilde{q}^{m,l}(u)| < 2\beta K_2, \tag{4.146}$$

The proof of this Lemma is an adaptation of the proof of Lemma 7 [88] to our case. We present the main arguments below.

Proof: [Proof of Lemma 4.13] Observe that, since $\mathcal{L}_2$ is decreasing to zero, there exists a time T, such that,

$$\mathcal{L}_2(t) \leq \epsilon_2, \ \forall t \geq T. \tag{4.147}$$

Consider the case $l = 0$. First, observe that, for m large enough,

$$\limsup_{m\to\infty} \sup_{u\in[0,T]} |\tilde{q}^{m,0}(u)| < \beta \limsup_{m\to\infty} |\tilde{q}^{m,0}(u)| \tag{4.148}$$

This is true because, if it were not, using Lemma 4.12, we could have a sequence of $\tilde{z}^{m,0}$ which converge to a fluid limit z with $|q(u)| \geq \beta|q(0)|$ for some u. However, this is not possible since,

$$\sup_{t\geq 0} |q(t)| < \beta|q(0)|. \tag{4.149}$$

Alongwith our assumptions on m, this implies that,

$$\limsup_{m\to\infty} \sup_{u\in[0,T]} |\tilde{q}^{m,0}(u)| < \beta \limsup_{m\to\infty} |\tilde{q}^{m,0}(u)| < \beta K_1 K, \tag{4.150}$$

$$\limsup_{m\to\infty} \sup_{u\in[0,T]} \tilde{w}^{m,0}(u) < \beta^2 K_1 K. \tag{4.151}$$

Using the non decreasing property of w, we can show,

$$\liminf_{m\to\infty} \inf_{u\in[0,T]} \tilde{w}^{m,0}(u) \geq K. \tag{4.152}$$

Choosing T large enough, we can have,

$$\mathcal{L}_2(\tilde{q}^{m,0}(T)) < 2\epsilon_2. \tag{4.153}$$

Since $\tilde{q}^{m,0}(T) = \tilde{q}^{m,1}(0)$, it also follows that,

$$\mathcal{L}_2(\tilde{q}^{m,1}(0)) < 2\epsilon_2. \tag{4.154}$$

Now, consider the following properties, for $l \in [1, 2\delta m/T - 1]$.

$$\mathcal{L}_2(\tilde{q}^{m.l}(0)) < 2\epsilon_2, \tag{4.155}$$

$$\mathcal{L}_2(\tilde{q}^{m.l}(T)) < 2\epsilon_2, \tag{4.156}$$

$$\mathcal{L}_2(\tilde{q}^{m.l}(u)) < 3\epsilon_2, \text{ for } \ u \in [0,T], \tag{4.157}$$

$$\tilde{v}^{m,l}(u) = \tilde{v}^{m,l}(u) - \tilde{v}^{m,l}(0) = 0, \text{ for } \ u \in [0,T], \tag{4.158}$$

$$K - 2\epsilon < \tilde{w}^{m,l}(u) < K_2, \text{ for } \ u \in [0,T], \tag{4.159}$$

$$(K-2\epsilon)/\beta < |\tilde{q}^{m,l}(u)| < 2\beta K_2. \tag{4.160}$$

We will show these hold, by induction on l. Asssume the properties hold for all $l < l_1$, but at least one of the abover properties is violated for $l = l_1$. Since the properties hold up to $l = l_1 - 1$, we have that,

$$\mathcal{L}_2(\tilde{q}^{m,l_1}(0)) = \mathcal{L}_2(\tilde{q}^{m,l_1-1}(T)) < 2\epsilon_2. \tag{4.161}$$

Since w is non decreasing, we have,

$$\tilde{w}^{m,l_1}(0) > K - 2\epsilon. \tag{4.162}$$

From the relation between $|q|$ and w it follows that,

$$|\tilde{q}^{m,l_1}(0)| \in \left[\frac{K-2\epsilon}{\beta}, 2\beta K_1\right]. \tag{4.163}$$

Thus, for a choice of T appropriately large, we will have,

$$\mathcal{L}_2(\tilde{q}^{m.l_1}(0)) < 2\epsilon_2, \tag{4.164}$$

$$\mathcal{L}_2(\tilde{q}^{m.l_1}(T)) < 2\epsilon_2, \tag{4.165}$$

$$\mathcal{L}_2(\tilde{q}^{m.l_1}(u)) < 3\epsilon_2, \text{ for } \ u \in [0,T]. \tag{4.166}$$

To show the non-increasing property of $\tilde{v}$ as in (4.158), observe that the queue length and workload are strictly positive as shown above. Since we had,

$$v^{m,l}(t) = x^{m,l}(t) - \langle \psi, d^{m,l}(t) - r^{m,l}(t)\rangle, \tag{4.167}$$

and since our optimization is such that we choose the allocation vector μ^* such that,

$$\mu^* = \arg_\mu \max \sum_{i,j,f} \alpha(q_i^f) q_{ij}^f \mu_{ij}^f, \tag{4.168}$$

$$= \arg_\mu \max \sum_{i,j,f} \alpha(q_i^f)(q_i^f - q_j^f)\mu_{ij}^f. \tag{4.169}$$

The second equation holds because the allocation vector $\dot{s}_{ij}^f(t)$ is zero when $q_i^f - q_j^f \leq 0$. This optimization may be rewritten as a function of new variables $\tilde{\mu}$, where $\tilde{\mu}_i^f = \sum_j \mu_{ij}^f - \sum_k \mu_{ki}^f$. We have the optimal $\tilde{\mu}^*$ given by,

$$\tilde{\mu}^* = \arg_{\tilde{\mu}} \max \sum_{i,j,f} \alpha(q_i^f) q_i^f \tilde{\mu}_i^f. \tag{4.170}$$

Since (4.166) holds, it will be that (choosing ϵ_2 small enough), this is exactly the result of the optimization,

$$\tilde{\mu}^* = \arg_{\tilde{\mu}} \max \sum_{i,j,f} \psi_i^f \tilde{\mu}_i^f, \tag{4.171}$$

since the function $\mathcal{L}_2$ indicates how close we are to the collapse vector ψ. From the definition of X, it follows that the scaled $\tilde{x}$ attains the value given above, and hence $\tilde{v}$ does not increase in the interval.

Since $\tilde{v}$ remains at zero, we can see that any increase in $\tilde{w}$ is an increase in $\tilde{u}$, and hence,

$$\tilde{w}^{m,l_1}(u) = \tilde{w}^{m,0}(T) + \tilde{u}^m(t_m + l_1 T/m + u/m) - \tilde{u}^m(t_m + T/m). \tag{4.172}$$

Since the oscillation of $\hat{u}$ is bounded and since $\tilde{u}^m \to \hat{u}$, the bounds (4.159) and (4.160) also follow for l_1. Hence, we have inductively shown that the properties (4.141)-(4.146) hold. □

With the above result, we also obtain the properties of $\hat{v}$.

4.9.1 Proof of the properties of $\hat{v}$

The proof of this result follows as an application of Lemma 4.13, as in the proof of Theorem 1 in [88]. We give a brief outline below.

First we show that $\hat{v}(t)$ is finite for all $t \in [0, \infty)$. Suppose this is not true. Then we will have some $t^0 = \inf\{t \geq 0 : \hat{v}(t) = \infty\}$. Fix $\delta > 0$, and $\epsilon = O_{\hat{u}}[t - 4\delta, t + 4\delta]_+$. Choose $\Delta \in (0, \min(t, \delta))$ and $C > \hat{w}(t - \Delta) + 2\epsilon$. Define the sequence, $t_n = \min\{s \geq t - \Delta : \hat{w}(s) \geq C\}$. Since $\hat{v}$ is RCLL, and since $\hat{v}(t) = \infty$, it follows that,

$$\limsup_n t_n \leq t. \tag{4.173}$$

Also, $\limsup_n \hat{w}^n(t - \Delta) < C$. Now, in a small interval, the process $\hat{w}$ will not have jumps, since,

$$\hat{w}^n(t) - \hat{w}^n(t-) \leq \langle \psi, \hat{a}^n(t) - \hat{a}^n(t-) \rangle + \langle \psi, \hat{r}^n(t) - \hat{r}^n(t-) \rangle, \tag{4.174}$$

and since the process R is bounded by the i.i.d channel process H. Using Lemma 4.6, the above quantity goes to zero. Hence it will follow that, as $n \to \infty$,

$$\hat{w}^n(t_n) \to C. \tag{4.175}$$

Choose a further subsequence along which,

$$t_n \to t^{'} \in [t - \Delta, t]. \tag{4.176}$$

Along this, applying Lemma 4.13, we see that $\hat{v}$ is finite on the interval $[0, t^{'} + \delta]$. Thus we have a contradiction, and hence $\hat{v}$ is finite. Note that a similar construction can be done for $t = 0$ as well.

The proof for continuity can also be done similarly, by finding point t which is a point of discontinuity. Choosing a suitable time before t, one can construct a sequence as before, which converges to a value C. Once again, we will use Lemma 4.13 to claim a contradiction. A similar proof holds for the other properties of $\hat{v}$ as well.

4.D Proof of (4.74)

Due to the strong Markov property, it suffices to show that,

$$\mathbb{E}\int_0^{\mathcal{T}_1}(1+|\hat{q}_x^n(s)|ds) \leq c_0(1+|x|^2). \tag{4.177}$$

Observe that,

$$Q^n(n^2t) = x + q^n(n^2t) + A^n(n^2t) - a^n(n^2t) + R^n(n^2t) - r^n(n^2t) - D^n(n^2t) + d^n(n^2t), \tag{4.178}$$

where,

$$q^n(t) = a^n(t) + r^n(t) - d^n(t), \tag{4.179}$$

is the fluid limit corresponding to the n-th system. Thus, one obtains the inequality,

$$\mathbb{E}[\sup_{0\leq t\leq \mathcal{T}} Q^n(n^2t)] \leq x + \mathbb{E}[\sup_{0\leq t\leq \mathcal{T}} q^n(n^2t)] + \mathbb{E}[\sup_{0\leq t\leq \mathcal{T}} |A^n(n^2t) - a^n(n^2t)|] \tag{4.180}$$

$$+\mathbb{E}[\sup_{0\leq t\leq \mathcal{T}} |R^n(n^2t) - r^n(n^2t)|] + \mathbb{E}[\sup_{0\leq t\leq \mathcal{T}} |D^n(n^2t) - d^n(n^2t)|]. \tag{4.181}$$

First we observe that,

$$\sup_{0\leq t\leq \mathcal{T}} q^n(n^2t) \leq \sup_t q^n(t). \tag{4.182}$$

Since the queue is non zero only till the draining time (given by (4.62)), and since the total input rate to a queue is bounded by the sum of all mean arrival rates and mean channel gains, it follows that there exists a constant c independent of t and n, such that,

$$\sup_t q^n(t) \leq x + nT_1, \tag{4.183}$$

where T_1 is a constant (see (4.62)). For the process A (and similarly for R and D), we can see that,

$$\mathbb{E}[\sup_{0\leq t\leq \mathcal{T}} |A^n(n^2t) - a^n(n^2t)|] = \mathbb{E}\left[\sqrt{\sup_{0\leq t\leq \mathcal{T}} |A^n(n^2t) - a^n(n^2t)|^2}\right] \tag{4.184}$$

$$\leq \sqrt{\mathbb{E}[\sup_{0\leq t\leq \mathcal{T}} |A^n(n^2t) - a^n(n^2t)|^2]}, \tag{4.185}$$

where the second inequality followed from Jensen's inequality. Using the bounds (4.65),(4.66) and (4.67), in the above equation, and plugging this as well as (4.183) in (4.180), we see that, for some constant c_1

$$\mathbb{E}[\sup_{0\leq t\leq \mathcal{T}} \hat{q}_x^n(t)] \leq c_1(1+x+\mathcal{T}). \tag{4.186}$$

By definition, $\mathcal{T}_1 \leq c_2(1+|x|)$. Using this fact and the above bound in the LHS of (4.177), we see that (4.177) is indeed true.

4.E Proof of (4.79)

This is adapted from the proof of Theorem 3.5 of [20] and Proposition 5.4 of [28].

We are given that, for all x,

$$\sup_n \int_0^{\tau^n(\delta)} (1 + |\hat{q}_x^n(t)|)dt \leq c(1 + |x|^2). \tag{4.187}$$

First we show that there exists a finite B such that,

$$\mathbb{E} \int_0^{\tau^n(t)} (1 + |\hat{q}_x^n(s)|)ds \leq \hat{V}_n(x) + Bt. \tag{4.188}$$

Since the LHS is increasing in t, we only need to show it holds for times of the form $m\delta$, $m = 1, 2, \dots,$ which will imply the result for t. Thus, we show that, for all m,

$$\mathbb{E} \int_0^{\tau^n(m\delta)} (1 + |\hat{q}_x^n(s)|)ds \leq \hat{V}_n(x) + mb, \tag{4.189}$$

where $b = \sup_n \sup_x \in \mathbb{D}\hat{V}_n(x)$. This is done by induction. The statement is true for $m = 1$. Assume it is true up to some m. Then, we have,

$$\mathbb{E} \int_0^{\tau^n((m+1)\delta)} (1 + |\hat{q}_x^n(s)|)ds = \mathbb{E} \int_0^{\tau^n(\delta)} (1 + |\hat{q}_x^n(s)|)ds + \mathbb{E}\mathbb{E}_{\hat{q}^n_{\tau^n(\delta)}} \int_0^{\tau^n((m)\delta)} (1 + |\hat{q}_x^n(s)|)ds \tag{4.190}$$

$$\leq \hat{V}_n(x) + \sup_n \sup_{x\in\mathbb{D}} \hat{V}^n(x) + mb, \tag{4.191}$$

$$\leq \hat{V}^n(x) + (m+1)b. \tag{4.192}$$

It follows that (4.188) holds for all t, with some $B \leq 2b$. Then, suing the strong Markov property, we obtain,

$$\mathbb{E}[\hat{V}_n(\hat{q}_x^n(t))] = \hat{V}_n(x) + \mathbb{E}\mathbb{E}_{\hat{q}^n_{\tau^n(\delta)}} \int_0^{\tau^n(t)} (1 + |\hat{q}_x^n(s)|)ds - \int_0^t \mathbb{E}_x[1 + |\hat{q}_x^n(s)|ds], \tag{4.193}$$

$$\leq \hat{V}^n(x) + Bt + b - \int_0^t \mathbb{E}_x[1 + |\hat{q}_x^n(s)|ds]. \tag{4.194}$$

This, along with (4.188), yields the result.

Chapter 5

Minimizing Age in a Multihop Wireless Network

In the thesis thus far we have been considering end-to-end mean delay or hard delay deadline for different flows passing through a multihop wireless network. These are traditional QoS requirements useful in transmission of data files, or real time traffic. However, in IoT sometimes the receiver is concerned about the latest data from the source. For example, in an industrial sensor network, the fusion center may be interested in the latest state of the system and the past states may not be important. Now it is not needed to have all the packets to be received at the destination. In this chapter we consider this new QoS for scheduling of wireless channels. We will see that the solution obtained so far in the thesis will not be appropriate at all for this scenario. But, the insights obtained and the algorithms developed will help us obtain a good solution even for this problem.

There are multiple source-destination pairs, transmitting data through multiple wireless channels, over multiple hops. We propose a network control policy which consists of a distributed scheduling algorithm, utilizing channel state information and queue lengths at each link, in combination with a packet dropping rule. Dropping of older packets locally at queues is seen to reduce the average age of flows, even below what can be achieved by Last Come First Served (LCFS) scheduling. Dropping of older packets also allows us to use the network without congestion, irrespective of the rate at which updates are generated. Furthermore, exploiting system state information substantially improves performance. The proposed scheduling policy obtains average age values close to a theoretical lower bound as well.

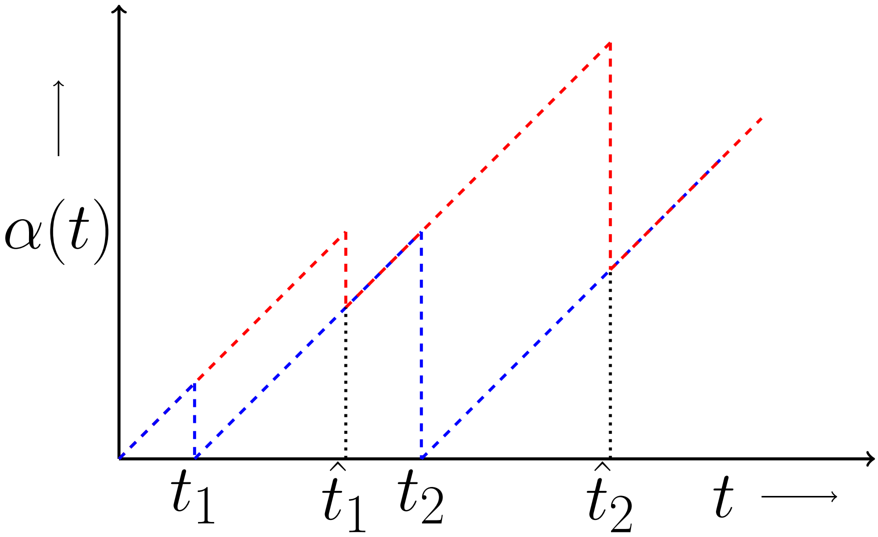

Figure 5.1: Evolution of Age. The red and blue lines show the evolution of the age of information at the destination and source respectively, as a function of time. At times t_1 and t_2, the first and second packets are generated at the source. These are received at the destination at times $\hat{t}_1$ and $\hat{t}_2$.

5.1 Age of Information

Consider a source generating packets to be sent to a destination, across a network. Let the packets be generated at the source at times $t_1, t_2, t_3, \ldots$. Let the same packets be received at the destination at times $\hat{t}_1, \hat{t}_2, \hat{t}_3 \ldots$. Note that the packets need not be received in the same order in which they were generated. Define,

$$n^*(t) = \arg_n \max\{t_n : \hat{t}_n \leq t\}. \tag{5.1}$$

This is the index of that packet among all packets received at the destination, till time t, which has been generated most recently, i.e., the freshest packet present at the destination. The age of information (at the destination) is defined to be the age of this packet, i.e.,

$$\alpha(t) = t - t_{n^*(t)}. \tag{5.2}$$

The evolution of the age function $\alpha(t)$ is given in Figure 5.1. Note that AoI can be defined for the source as well, seeing it as a point that receives the packets with zero delay. We define the

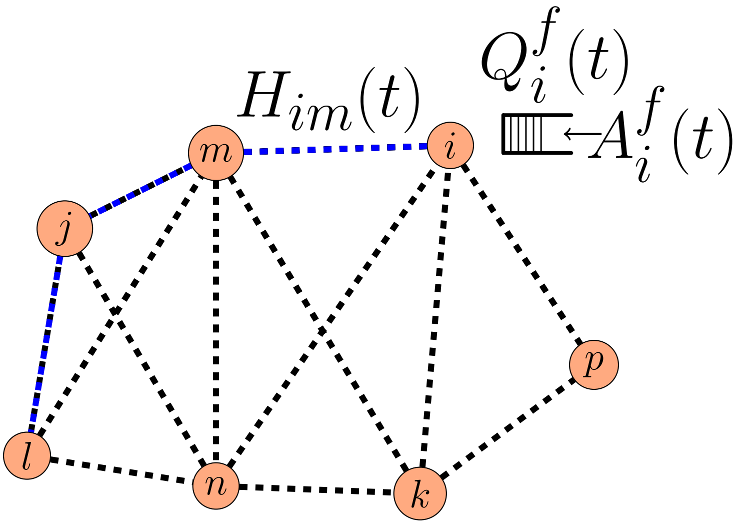

Figure 5.2: A simplified depiction of a Wireless Multihop Network. The flow f follows the path $i \to m \to j \to l$.

Average AoI $\bar{\alpha}(t)$ as,

$$\bar{\alpha}(t) = \frac{1}{t}\int_0^t \alpha(\tau)d\tau. \tag{5.3}$$

We will refer to the (average) age at the destination node to be the (average) age of the flow. Between the source and destination, packets experience queueing delays and transmission delays. This system can be modelled as a system of queues. While the queueing delay contributes to the age process, delay by itself is not identical to age. The age process depends on both the queueing delay and the rate at which packetized updates are being generated at the source. One can reduce the packet generation rate, which may lead to lower buffer levels, and hence, lower delays. However, owing to fewer updates, the age process may not reduce. On the other hand, sending too many updates may lead to congestion in the network.

5.2 System Model and Problem Formulation

We consider a multihop wireless network (see Fig (5.2)), modelled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges (links) on $\mathcal{V}$. In the network, packets are generated at *source nodes*, to be sent to various *destination nodes*. Each such stream of packets, corresponding to a source-destination pair, is called a *flow*. The set of all flows in the network will be denoted by $\mathcal{F}$. For any flow $f \in \mathcal{F}$, we will use $src(f)$ and $des(f)$ to denote its source and destination nodes, and $path(f) \subseteq \mathcal{V}$ to be the path of nodes connecting the source of flow f to its destination. We assume that paths are fixed and known a priori. This would imply that a routing algorithm was employed beforehand to create these routes (see [1] for a survey of common routing algorithms in wireless sensor networks).

We have a slotted system, with time index $t \in \{0, 1, 2, \dots\}$. Each slot is of unit length and time duration $[t, t + 1)$ denotes slot t. The arrival process for a flow f with source node $src(f) = i$ is denoted by $A_i^f(t)$. We will assume that $A_i^f(t)$ evolves as an independent and identically distributed (i.i.d.) sequence across time slots and independent of other flows. The wireless channel gain of a link $(i, j) \in \mathcal{E}$ at time t will be denoted by $H_{ij}(t)$. This is also i.i.d. across time for a link, and is independent across links. The overall channel state is denoted by $H(t) = \{H_{ij}(t)\}_{(i,j)\in\mathcal{E}}$. We transmit at a constant power and a fixed rate. If a channel gain is above a threshold and interference from other channels is limited then we assume that there is a successful transmission. At each node i, there is a queue $Q_i^f(t)$ which consists of packets of flow f present at node i. Let $S_{ij}^f(t)$ denote the number of packets of flow f sent over link (i, j) in time slot t. Then, we can write the queue evolution equation as,

$$Q_i^f(t+1) = Q_i^f(t) + \sum_j S_{ji}^f(t) - \sum_k S_{ik}^f(t), \tag{5.4}$$

where $i \neq des(f)$. By $\alpha^f(t)$ and $\bar{\alpha}^f(t)$ we denote the AoI and average AoI of flow f at its destination node, as defined by (5.2) and (5.3).

We will assume that the links fall into interference sets. An interference set is a subset of $\mathcal{E}$ such that no two members of that set can transmit simultaneously. These sets define the *interference constraints* of the system. Subject to these constraints, only certain configurations of links can be activated at a time.

We define a *schedule* as a mapping $s : \mathcal{E} \times \mathcal{F} \to \{0, 1\}$. If $s(e, f) = 1$, then flow f is scheduled to be transmitted on link e in that slot. Not all mappings from $\mathcal{E} \times \mathcal{F}$ to $\{0, 1\}$ are feasible schedules. The links that are active must obey the interference constraints. Further, two flows cannot be simultaneously scheduled on a link. The schedules that obey these constraints are called *feasible* schedules. Denote the set of all feasible schedules by $\mathcal{S}$. Corresponding to each feasible schedule s and channel state H, there is a rate vector $R = \{R_{ij}^f\}_{(i,j)\in\mathcal{E}, f\in\mathcal{F}}$. Let $\alpha_i^f(t)$ denote the age of flow f at node i. We are interested in obtaining control policies that can reduce the AoI at the destinations. To this end, we propose the following policy.

5.2.1 Control Policy

The control policy we propose will be called State Dependent Scheduling with Packet Dropping (SDSPD). This policy consists of two aspects: a service discipline and an optimization rule.

5.2.1.1 Service Discipline

Under the SDSPD policy, at each queue, we keep only the latest packet of a flow, and all others are discarded. Thus, if a more recently generated packet of a flow is received at a queue, all packets generated prior to that packet of that flow at the queue are dropped. This is a local decision that can be implemented at the node level. There is no need for exchange of information between the nodes for this purpose. Consequently, at all nodes i and for all flows f, $Q_i^f \in \{0, 1\}$. Such a service discipline will result in a performance similar to (or better than) an LCFS discipline.

5.2.1.2 Optimization Rule

The schedule at time t is chosen to be $s^*(t)$, where,

$$s^*(t) = \arg_{s \in \mathcal{S}} \max \sum_{i,j,f} w^f(\alpha^f(t)) Q_i^f(t) R_{ij}^f(s, H(t)), \tag{5.5}$$

where w^f is the weight for flow f, which is a function of the age $\alpha^f(t)$ of flow f at time t at its destination node. Also,

$$w^f(x) = \begin{cases} 1 \text{ if } x < \bar{\alpha}^f, \\ 1 + \beta \text{ if } x \geq \bar{\alpha}^f, \end{cases} \tag{5.6}$$

where $\bar{\alpha}^f$ is a desired average age for flow f, and β is a fixed positive quantity. This represents a weighted queue policy with dynamic weights. The weight function w^f enables us to differentiate between the flows, and gives higher priority to some flows, if desired. A flow with a higher weight will be scheduled more often, and consequently its age should decrease. A lower $\bar{\alpha}^f$ gives higher priority to flow f.

Note that the quantity being optimized is different from the traditional maxweight metric, which involves a *backpressure* term. Owing to the packet dropping in our system, the vector $Q(t)$ remains in a bounded set for all time t, and consequently, the system is always stable. Hence, we do not use a maxweight formulation, which is used generally to guarantee stability (within the capacity region of the system).

We will see in Section III that this policy is seen to yield a good performance in terms of the average AoI metric. We compare it with multiple policies, and see the benefit of dropping packets, even compared to policies which do LCFS. In the following section, we describe how we may solve the optimization problem in a distributed manner.

5.2.2 Distributed Implementation

While the optimization (5.5) may be non-convex in general, in case of smaller state spaces, it can be computed by a brute force search. For larger state spaces, it can be approximated by a linear relaxation (relaxing the scheduling variables s to belong to the interval $[0, 1]$ rather than the set $\{0, 1\}$). The relaxed set of feasible vectors s will be denoted by $\mathcal{S}^*$. The relaxed linear program can be written in the form,

$$\arg_s \max \sum_{i,j,f} \theta(i, j, f) s_{ij}^f, \tag{5.7}$$

$$s.t\ s_{ij}^f \in [0, 1],\ \forall\ i, j, f, \tag{5.8}$$

where $\theta(i, j, f) = w^f Q_i^f(t) R_{ij}^f$, and $R_{ij}^f = R_{ij}^f(H(t))$ is the rate that is achievable for the link (i, j) if it is transmitting at fixed power, and none of the links it interferes with is on. This is now a separable linear program, and can then be solved in a distributed fashion.

One algorithm that can be used to solve it in a distributed fashion is the Incremental Gradient Descent algorithm (IGD) [9]. Let $\mathcal{K}$ denote the set of all link-flow pairs, i.e., all elements of the form $((i, j), f)$ where $(i, j) \in \mathcal{E}$ and $f \in \mathcal{F}$. Then, IGD provides,

$$s_{n+1} = \Pi_{\mathcal{S}^*}[(s_n + \alpha v_{k_n} \theta(k_n) s_n], \tag{5.9}$$

with $k_n = n$ modulo $|\mathcal{K}| + 1$, α is a small positive number, v_{k_n} is a vector which is one at its K_n-th position and is zero elsewhere, and $\Pi_{\mathcal{S}^*}$ denotes projection onto the set $\mathcal{S}^*$. Due to the vector v_{k_n}, the update of the vector can be performed in a component wise manner. Thus, one can perform the update in (5.9) in a cyclic manner, going from one element of $\mathcal{K}$ to the next. What this would mean is that at each node, we can do the increment step in (5.9) for all the links that originate at that node, and then move to a neighbour. This process then continues cyclically. Thus, we can peform the optimization (5.7)-(5.8) in a distributed manner, with messages passed between neighbouring nodes.

Since the power of transmission is fixed, and we assume that the channel gains take values from a bounded set, it follows that the rates are bounded by some $\bar{R}$. Further assume that the weights w^f are bounded by some $\bar{w} \in \mathbb{R}$. Let us define,

$$F(s) = \sum_{k \in \mathcal{K}} \theta(k) s(k),\ \ s \in \mathcal{S}^*. \tag{5.10}$$

Then, the following result from [9] holds.

Lemma 5.1 *The iterates given by (5.9) result in a sequence of points $\{s_n\} \in \mathcal{S}^*$ which satisfy,*

$$\lim_{n\to\infty} \sup F(s_n) \geq \max_{s\in\mathcal{S}^*} F(s) - C,$$

where $C = \frac{\alpha \bar{w}^2 \bar{R}^2 |\mathcal{K}|(4|\mathcal{K}|+1)}{2}$.

Thus, we can choose α small enough to come close to the optimal value. Note that the algorithm does not require that the age at the destination be available at every node having that flow for computing the optimization. It is only necessary that it be known whether the age exceeds a threshold or not. We can have mini slots at the beginning of each slot, during which the destination node can broadcast a signal at a fixed power, to indicate whether the age has exceeded a threshold. Absence of the signal would indicate that the age is below the threshold. Using this simple signalling scheme, the one bit information corresponding to each flow can be broadcast.

5.3 Simulation Results and Discussion

We compare the proposed policy, SDSPD, with five other policies. First, we have Backpressure with Dropping (BP-D), which is the same as SDSPD, except that the optimization (5.5) is replaced by,

$$s^*(t) = \arg_{s\in\mathcal{S}} \max \sum_{i,j,f} Q_{ij}(t) R_{ij}^f(s, H(t)), \tag{5.11}$$

where $Q_{ij} = \max_f (Q_i - Q_j)^+$. This can be considered as a maxweight (backpressure) policy with dropping. There are two other variants of the SDSPD policy, which use the same scheduling rule as SDSPD, but they do not drop packets. The first of these is SDSPnD-FCFS, which has the FCFS service discipline, and the second, which has LCFS service, will be called SDSPnD-LCFS. We also compare with BP-LCFS and BP-FCFS. which are backpresssure policies without dropping packets, with LCFS and FCFS service respectively. Finally we have the randomized scheduling policy of [91], which is a randomized stationary policy, which solves an optimization to obtain activation probabilities for links. Thus, it does not use instantaneous state information. Comparing with all these schemes allows us to evaluate the performance of the SDSPD algorithm against common scheduling schemes, some of which have been shown to perform well in terms of age.

We consider two example networks. All simulations are run for 10^4 time slots, and averaged over 100 such trials. For a theoretical comparison, we use the following lower bound on the age.

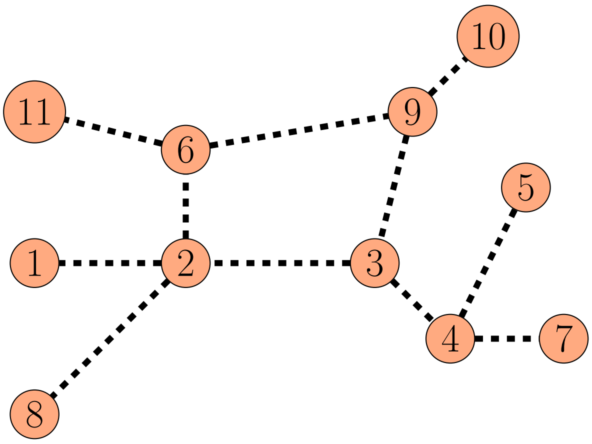

Figure 5.3: Example network 1.

5.3.1 An Approximate Lower Bound for Age

Consider a discrete time queue, with a Bernoulli arrival process, so that in each slot, a packet arrives with probability p, and with probability $1-p$, no packet arrives. Let X denote the time between two packet arrivals. Clearly,

$$\mathbb{E}X = \frac{1}{p},\ \mathbb{E}X^2 = \frac{2-p}{p^2}. \tag{5.12}$$

The average age of the arrival process will be,

$$\bar{\alpha} = \frac{\mathbb{E}X^2}{2\mathbb{E}X} = \frac{2-p}{2p}. \tag{5.13}$$

If we assume that the channel takes values 0 or 1 with probability $1-q$ and q respectively, the mean time between two time slots in which the channel state is 1, is $\frac{1}{q}$, and this adds to the average age. Across a system of n such links, we can obtain a lower bound on average age as,

$$\frac{2-p}{2p} + \frac{n}{q}. \tag{5.14}$$

Observe that this is a loose bound, because it assumes that there is only one flow in the system. In a system with multiple flows, we may be far away from this lower bound.

5.3.2 Example Network 1

The network considered in this example is given by Figure 5.3. The channel gains take value 0 or 1 with probability 0.5, in each slot. We will assume that if channel gain equals 1, exactly one packet can be successfully transmitted. This models a situation where the channel is above a threshold with probability 0.5, and hence ensures succesful transmission. The flows are from node 1 to 5 (path $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$), from node 6 to node 7 (path $6 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 7$), from node 8 to 10 (path $8 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10$), from node 11 to 9 (path $11 \rightarrow 6 \rightarrow 9$) and from node 11 to node 2 (path $11 \rightarrow 6 \rightarrow 2$). The interference model assumes that any two links that have a common node interfere, and therefore cannot be active simultaneously. All weights w^f in the optimization (5.5) are identically set to one (by choosing $\bar{\alpha}^f = \infty$ for all f). The arrival process is i.i.d Bernoulli across slots, with packet arrival rate 0.1 for all the flows.

Table 5.1 gives the value of average AoI obtained at the destination for each flow, for SDSPD, SDSPnD-FCFS, SDSPnD-LCFS, BP-D, BP-FCFS, BP-LCFS and the stationary policy of [91], as well as the loose lower bound (5.14). From the values in Table 5.1 it is easy to see that SDSPD is the best performing, and improves over the LCFS policy as well. The FCFS policy performs decently, but the age performance of the FCFS policy will deteriorate as we increase the arrival rates. The stationary policy of [91] performs an order worse than the other three, because it does not take into account channel or buffer state information. For SDSPD, the flows also have ages close to the lower bound. Recall that the lower bound was assuming a single flow using up all the resources. Even with five flows in the network, SDSPD performs quite close to the lower bound. The BP-D policy performs close to SDSPD. However, SDSPD offers a slight improvement over BP-D, especially for the first flow.

Table 5.1: Average AoI for different flows under different policies, for the network in figure 5.3, with arrival rates of all flows fixed at 0.1.

	Flow 1→5	Flow 6→7	Flow 8→10	Flow 11→9	Flow 11→2
Lower Bound	17.5	17.5	17.5	13.5	13.5
SDSPD	22.2	20.1	19.2	14.6	17.4
BP-D	24.6	20.5	19.6	14.8	17.9
SDSPnD-LCFS	25.5	24.6	22.8	15.6	18.9
BP-LCFS	37.4	31.9	27.6	16.3	23.5
SDSPnD-FCFS	33.9	30.5	26.2	15.9	21.9
BP-FCFS	47.2	37.3	30.1	16.3	25.4
Policy of [91]	190.2	242.8	149.5	61.6	112.7

We repeated the simulation for arrival rate 0.13 for all the flows. The values obtained are given in Table 5.2. Here we see that the age performances of the non-dropping policies begin to deteriorate, owing to congestion. The SDSPD and BP-D policies perform well. The age of all the flows of the SDSPD system have reduced, when compared to Table 5.1. The policy is able to utilize the higher rate of updates to reduce the overall age.

From the above two tables, it may seem that the policy of [91] has the worst performance.

Table 5.2: Average AoI for different flows under different policies, for the network in figure 5.3, with arrival rates of all flows fixed at 0.13.

	Flow 1→5	Flow 6→7	Flow 8→10	Flow 11→9	Flow 11→2
Lower Bound	15.2	15.2	15.2	11.2	11.2
SDSPD	21.2	18.4	17.3	12.5	16.2
BP-D	24.9	19.2	17.9	12.7	16.9
SDSPnD-LCFS	43.1	51.6	40.4	16.2	19.9
BP-LCFS	95.5	98.3	79.6	19.2	51.1
SDSPnD-FCFS	97.8	100.3	81.9	17.6	50.6
BP-FCFS	160.3	154.0	121.9	20.1	78.1
Policy of [91]	186.5	250.5	163.2	62.6	111.2

However, this is not true in general. As we increase the arrival rates further, we see that the average AoI for the non dropping policies begin to blow up as expected, owing to congestion. This can be seen in Table 5.3, which summarizes the average AoI values for the different algorithms when arrival rate is 0.14. The BP-FCFS algorithm performs the worst.

The above results demonstrate that the SDSPD policy can give low average AoI, close to the lower bound. Next, we demonstrate how we can use the weights w^f to reduce the average AoI even further. This is done by fixing the $\bar{\alpha}^f$ values in (5.6). The results are given in Table 5.4, for the network in figure 5.3, with arrival rates of all flows fixed at 0.14. The first row gives the values of average AoI without targets. In the second row, we fix a target of 18 for the first flow, and obtain an average AoI of 17.3. In the next row, we set the target to be 15, and obtain an average AoI of 16.7. Recall from Table 5.3 that the loose lower bound for AoI assuming that

Table 5.3: Average AoI for different flows under different policies, for the network in figure 5.3, with arrival rates of all flows fixed at 0.14.

	Flow 1→5	Flow 6→7	Flow 8→10	Flow 11→9	Flow 11→2
Lower Bound	14.6	14.6	14.6	10.6	10.6
SDSPD	20.9	18.1	16.8	11.9	16.1
BP-D	25.1	18.9	17.5	12.2	16.9
SDSPnD-LCFS	184.7	195.8	181.2	17.5	21.3
BP-LCFS	251.1	259.6	234.3	21.6	132.6
SDSPnD-FCFS	388.7	396.1	371.9	19.7	200.8
BP-FCFS	408.9	409.2	368.3	23.5	247.8
Policy of [91]	199.1	264.4	163.8	65.8	102.2

Table 5.4: Average AoI for different flows under the SDSPD policy, for the network in figure 5.3, with arrival rates of all flows fixed at 0.14. First column gives the target age for each flow. A $*$ indicates that the target is set to ∞ (i.e., no target).

Target average age $\bar{\alpha}^f$	Flow	Flow	Flow	Flow	Flow
for each flow	1→5	6→7	8→10	11→9	11→2
--*-*-*	20.9	18.1	16.8	11.9	16.9
18-*-*-*-*	17.3	19.6	17.7	12.0	16.4
15-*-*-*-*	16.7	20.3	18.0	12.0	16.6
15-*-*-*-11	16.7	21.6	18.3	12.7	12.3
-16--*-12	22.2	16.6	17.8	12.9	12.8

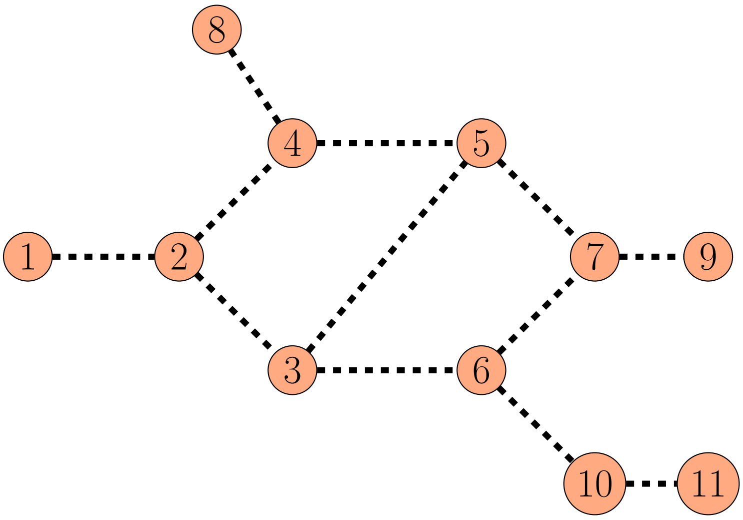

Figure 5.4: Example network 2.

only one flow is present was 14.6, and therefore 16.7 is a good value for average AoI. The AoI of other flows is only marginally increased. In the next row, we set targets of 15 and 11 for the first and last flows (with lower bounds 14.6 and 10.6 respectively), and obtain average AoI values of 16.7 and 12.3. In the last row we set targets of 16 and 12 for the second and last flows, respectively, and obtain 16.6 and 12.8 respectively. Thus, the algorithm can provide close to optimal performance, and can prioritize some flows over others if necessary.

5.3.3 Example Network 2

The network considered in this example is given in Figure 5.4. The channel, arrival and interference models are the same as in the previous example. The flows are $1 \to 2 \to 4 \to 5 \to 7 \to 9$, $3 \to 2 \to 4 \to 8$, $4 \to 5 \to 3 \to 6 \to 10$ and $4 \to 5 \to 7 \to 6 \to 10 \to 11$. Table 5.5 depicts values of Average AoI for the four flows, under the different policies considered.

Table 5.5: Average AoI for different flows under different policies, for the network in figure 5.4, with arrival rates of all flows fixed at 0.1.

	Flow 1→9	Flow 3→8	Flow 4→10	Flow 4→11
Lower Bound	19.5	15.5	17.5	19.5
SDSPD	25.9	17.5	20.5	20.6
BP-D	29.8	17.7	21.2	21.1
SDSPnD-LCFS	28.0	19.2	26.5	25.9
BP-LCFS	42.7	21.7	27.8	26.2
SDSPnD-FCFS	37.2	20.7	27.9	27.4
BP-FCFS	59.0	22.8	28.9	26.9
Policy of [91]	238.2	104.7	185.7	209.7

In this set of simulations too, we see that the patterns observed in the previous example hold.

Table 5.6: Average AoI for different flows under different policies, for the network in figure 5.4, with arrival rates of all flows fixed at 0.13.

	Flow 1→9	Flow 3→8	Flow 4→10	Flow 4→11
Lower Bound	17.2	13.2	15.2	17.2
SDSPD	25.9	15.6	18.9	18.6
BP-D	32.1	15.9	20.1	19.3
SDSPnD-LCFS	28.8	20.3	48.5	47.2
BP-LCFS	83.1	35.4	55.4	56.1
SDSPnD-FCFS	79.9	32.4	55.9	58.5
BP-FCFS	179.6	49.5	66.0	68.4
Policy of [91]	231.9	101.7	178.7	204.7

5.3.4 Discussion

These experiments seem to suggest that dropping of packets locally at queues can help reduce age. Moreover, we get a policy that is robust to arrival rate variation. Now it may be that in certain applications, it is imperative to get all the packets from the source to the destination, without losing any information. In such cases one may use the SDSPnD-LCFS scheme, which performs the best among all policies without packet dropping. The disadvantage of non-dropping policies, however, is that in case of large arrival rates, the queues will be large, and the time to move all the packets across, from source to destination, will be huge. If the arrival rates are outside the stability region of the policy, this time may very well be not finite. In such a case, it is not even feasible to get all the packets across. Moreover, as the queue lengths build up, the complexity of optimizations used for resource allocation, may also increase. Against all these, SDSPD offers a distinct advantage. Additionally, the dynamically varying weight function allows us to obtain targeted age.

5.4 Conclusion

In this chapter, we have presented a control policy which reduces the average AoI in a multihop wireless network. The control policy involves dropping of older packets at each queue, in favour of the youngest packet, and using the queue lengths and channel gains at each link. This policy is seen to perform better than policies without dropping, including LCFS schemes. Indeed, in many cases the scheme of dropping packets offers a huge improvement over LCFS schemes. It also performs much better than policies which do not use state information. Further, the average age obtained by the proposed policy is quite close to a theoretical lower bound as well. We further show that we can come even closer to the lower bound by using the age information at the destination. For applications for which there is no need to get all packets across to the destination, dropping of packets in the manner presented can help improve the performance in terms of age. Not keeping a backlog of older packets reduces buffering requirements. Moreover, there is no need to spend energy in transmitting packets which are not fresh. The network capacity does not become a bottleneck in the transmission of fresh information. With packet dropping, higher rates of arrivals of packets do not result in an increase in the age due to queueing. We see a monotone decrease in the average age of different flows, as arrival rate increases. What this suggests is that in systems with packet dropping, the network is no longer a constraint on the optimal sampling rate. Thus, we can fix the sampling rate independent of network considerations, and dependent only on the energy or other requirements of the sampler at the source node.

Chapter 6

Conclusions and Future Directions

In this thesis, we have considered the problem of distributed control of a multihop wireless network, with QoS provisions, under the SINR and graphical interference models.

In Chapter 2, we considered the problem of distributed control of a multihop network under the SINR interference model. We proposed a randomized control policy which stabilized a fraction of the capacity region. The policy was implemented in a distributed manner using gossip algorithms. Further, it also gave targeted mean delay and hard deadline QoS to different flows, by dynamically varying the probability with which nodes become transmitters and receivers. This probability is a function of queue length as well as QoS. Flows that did not meet the QoS requirements get a higher weight as compared to flows that did, and non QoS flows. By means of simulations, we compare it with existing distributed policies, and see that it performs much better. From the stability region expressions, as well as from simulations, we can see that there is a tradeoff between stability and QoS. More the QoS that the system tries to provide, less is the traffic that it is able to support with stability.

In Chapter 3, we considered the problem of distributed control of a multihop wireless network under a graphical interference model. We proposed a control policy under the scheme of discrete review. Here, control decisions are taken at only certain time epochs, known as review times, by solving an optimization problem. The optimization problem was formulated using insights from the notion of draining time in fluid networks. The control algorithm was a linear program, which was then solved using Incremental Gradient Descent. This algorithm was implemented in a distributed manner, using a distributed update and projection step. The convergence of the distributed algorithm was also studied. The algorithm gave higher weights to flows with QoS constraints that have not been met. The algorithm was able to provide targeted mean delay and hard deadline QoS to flows. We also proposed a slightly modified optimization, which was shown to be throughput optimal using fluid limit analysis. The modified algorithm was also able to provide mean delay QoS.

In Chapter 4, we considered an algorithm similar to that of Chapter 3, but without the discrete review assumption. This algorithm is also throughput optimal. The algorithm was studied under the heavy traffic regime, under diffusion scaling. A sequence of scaled processes was shown to converge to a reflected Brownian motion with drift. We also show that the stationary distributions of these scaled processes converge to the stationary distribution of the reflected Brownian motion with drift. Consequently, the stationary distribution of the limiting process is a good approximation for the stationary distribution of the system under heavy traffic. By means of simulations, we verify that these theoretical results are accurate.

In Chapter 5, we consider the problem of control of a multihop wireless network with average age of information being the QoS. We present a control policy that involves dropping of older packets at each queue, in favour of younger packets, and involves an optimization that uses system state information. The algorithm is similar to the one proposed initially in Chapter 3. We demonstrate via simulations that the scheme offers improvement in average age over most existing schemes, including those that do LCFS service, and those that do not make use of state information. Comparing with a theoretical lower bound, we demonstrate that we are able to come quite close to this in performance. Further, the scheme may also be used to give dynamic priority to flows so as to give targeted average age to desired flows. Since we drop packets, there is no congestion at the nodes. Consequently, network capacity does not become a bottleneck for the rate at which update packets are generated. This is beneficial, since more updates can now lead to lower age.

6.1 Future Directions

Throughout this work, we have assumed a fixed number of users, non-mobile, and with the flows (number, source, destination) fixed from beginning to end. One could consider a mobile scenario, with users arriving and leaving the system. Similarly there will be flows arriving, getting served, and leaving. There have been very few theoretical studies of such systems. Such a study would be a practically relevant extension of this work.

While the capacity region of the algorithm in Chapter 2 has a lower bound, it will be interesting to see if there are any tight upper bounds to the same. This will be a cap on the maximum achievable throughput of such a distributed randomized policy, and will characterize the tradeoff between distributed and centralized implementation.

We used Little's Law to connect mean delay and mean queue length, to formulate weights for mean delay QoS in Chapter 3. What would be a suitable equivalent to characterize hard deadlines? This remains an open question.

Whether a Brownian limit exists for a control policy under the discrete review setup of Chapter 3 is an open question. It is likely that the scaling to obtain a limit for such a system is different from the traditional diffusion scaling.

Obtaining better theoretical bounds on the age for multihop networks and characterizing age optimal policies is a relevant future direction in the case of age QoS.

Bibliography

[1] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325–349, 2005. 35, 109

[2] Matthew Andrews, Krishnan Kumaran, Kavita Ramanan, Alexander Stolyar, Rajiv Vijayakumar, and Phil Whiting. Scheduling in a queuing system with asynchronously varying service rates. *Probability in the Engineering and Informational Sciences*, 18(2):191–217, 2004. 3, 61

[3] Krishna B Athreya and Soumendra N Lahiri. *Measure theory and probability theory*. Springer Science & Business Media, 2006. 31, 64, 85, 94

[4] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010. 1

[5] Irfan Awan, Muhammad Younas, and Wajia Naveed. Modelling qos in iot applications. In *2014 17th International Conference on Network-Based Information Systems*, pages 99–105. IEEE, 2014. 1

[6] Nicole Bäuerle. Optimal control of queueing networks: An approach via fluid models. *Advances in Applied Probability*, 34(2):313–328, 2002. 4

[7] Ahmed M Bedewy, Yin Sun, and Ness B Shroff. Age-optimal information updates in multihop networks. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 576–580. IEEE, 2017. 7

[8] Dimitri P Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011. 41, 44

[9] Dimitri P Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011. 111

[10] Dimitris Bertsimas, David Gamarnik, and John N Tsitsiklis. Stability conditions for multiclass fluid queueing networks. *IEEE Transactions on Automatic Control*, 41(11): 1618–1631, 1996. 3

[11] Dimitris Bertsimas, Ebrahim Nasrabadi, and Ioannis Ch Paschalidis. Robust fluid processing networks. *IEEE Transactions on Automatic Control*, 60(3):715–728, 2014. 4

[12] Michael Best. The wireless revolution and universal access. *Trends in Telecommunications Reform*, pages 1–24, 2003. 1

[13] Patrick Billingsley. *Convergence of probability measures*. John Wiley & Sons, 1968. 4, 74, 78

[14] Holger Boche and Slawomir Stanczak. Convexity of some feasible qos regions and asymptotic behavior of the minimum total power in cdma systems. *IEEE Transactions on Communications*, 52(12):2190–2197, 2004. 6

[15] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013. 32

[16] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2508–2530, 2006. 3

[17] Maury Bramson. State space collapse with application to heavy traffic limits for multiclass queueing networks. *Queueing Systems*, 30(1-2):89–140, 1998. 5, 79, 92

[18] Maury Bramson. A stable queueing network with unstable fluid model. *Annals of applied probability*, pages 818–853, 1999. 3

[19] Anton Braverman, JG Dai, and Masakiyo Miyazawa. Heavy traffic approximation for the stationary distribution of a generalized jackson network: The bar approach. *Stochastic Systems*, 7(1):143–196, 2017. 5

[20] Amarjit Budhiraja and Chihoon Lee. Stationary distribution convergence for generalized jackson networks in heavy traffic. *Mathematics of Operations Research*, 34(1):45–56, 2009. 5, 82, 105

[21] Loc Bui, Atilla Eryilmaz, R Srikant, and Xinzhou Wu. Joint asynchronous congestion control and distributed scheduling for multi-hop wireless networks. In *Infocom*, 2006. 3

[22] Hong Chen. Fluid approximations and stability of multiclass queueing networks: work-conserving disciplines. *The Annals of Applied Probability*, pages 637–665, 1995. 3

[23] Hong Chen and David D Yao. Dynamic scheduling of a multiclass fluid network. *Operations Research*, 41(6):1104–1115, 1993. 4

[24] Shigang Chen and Klara Nahrstedt. Distributed quality-of-service routing in ad hoc networks. *IEEE Journal on Selected areas in Communications*, 17(8):1488–1505, 1999. 6

[25] Ying Cui, Vincent KN Lau, Rui Wang, Huang Huang, and Shunqing Zhang. A survey on delay-aware resource control for wireless systems—large deviation theory, stochastic lyapunov drift, and distributed stochastic learning. *IEEE Transactions on Information Theory*, 58(3):1677–1701, 2012. 6

[26] Ying Cui, Edmund M Yeh, and Ran Liu. Enhancing the delay performance of dynamic backpressure algorithms. *IEEE/ACM Transactions on Networking (TON)*, 24(2):954–967, 2016. 3

[27] Jim G Dai. On positive harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *The Annals of Applied Probability*, 5(1):49–77, 1995. 3, 55

[28] Jim G Dai and Sean P Meyn. Stability and convergence of moments for multiclass queueing networks via fluid limit models. *IEEE Transactions on Automatic Control*, 40 (11):1889–1904, 1995. 4, 5, 105

[29] Amir Dembo and Ofer Zeitouni. *Large Deviations Techniques and Applications*. Springer, 2010. 6

[30] Alexandros G Dimakis, Soummya Kar, José MF Moura, Michael G Rabbat, and Anna Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010. 3

[31] Atilla Eryilmaz and Rayadurgam Srikant. Asymptotically tight steady-state queue length bounds implied by drift conditions. *Queueing Systems*, 72(3-4):311–359, 2012. 5

[32] Serguei Foss and Artyom Kovalevskii. A stability criterion via fluid limits and its application to a polling system. *Queueing Systems*, 32(1-3):131–168, 1999. 3

[33] David Gamarnik and Assaf Zeevi. Validity of heavy traffic steady-state approximations in generalized jackson networks. *The Annals of Applied Probability*, 16(1):56–90, 2006. 5

[34] Leonidas Georgiadis, Michael J Neely, and Leandros Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends® in Networking*, 1(1): 1–144, 2006. 6

[35] Itai Gurvich. Validity of heavy-traffic steady-state approximations in multiclass queueing networks: The case of queue-ratio disciplines. *Mathematics of Operations Research*, 39 (1):121–162, 2013. 5

[36] Allan Gut. *Stopped random walks.* Springer, 2009. 85

[37] Jean-Paul Haddad and Ravi R Mazumdar. Heavy traffic approximation for the stationary distribution of stochastic fluid networks. *Queueing Systems*, 70(1):3–21, 2012. 5

[38] J Harrison. Brownian motion and stochastic flow systems. 1985. 4

[39] J Michael Harrison. The bigstep approach to flow management in stochastic processing networks. *Stochastic Networks: Theory and Applications*, 4:147–186, 1996. 4

[40] J Michael Harrison. Heavy traffic analysis of a system with parallel servers: asymptotic optimality of discrete-review policies. *Annals of applied probability*, pages 822–848, 1998. 4

[41] J Michael Harrison and Martin I Reiman. Reflected brownian motion on an orthant. *The Annals of Probability*, 9(2):302–308, 1981. 4, 88

[42] Donald L Iglehart and Ward Whitt. Multiple channel queues in heavy traffic. i. *Advances in Applied Probability*, 2(1):150–177, 1970. 4

[43] Bo Ji, Changhee Joo, and Ness B Shroff. Throughput-optimal scheduling in multihop wireless networks without per-flow information. *IEEE/ACM Transactions on Networking (TON)*, 21(2):634–647, 2013. 4

[44] Changhee Joo and Atilla Eryilmaz. Wireless scheduling for information freshness and synchrony: Drift-based design and heavy-traffic analysis. *IEEE/ACM Transactions on Networking (TON)*, 26(6):2556–2568, 2018. 7

[45] Igor Kadota, Abhishek Sinha, and Eytan Modiano. Optimizing age of information in wireless networks with throughput constraints. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1844–1852. IEEE, 2018. 7

[46] Clement Kam, Sastry Kompella, and Anthony Ephremides. Effect of message transmission diversity on status age. In *2014 IEEE International Symposium on Information Theory*, pages 2411–2415. IEEE, 2014. 7

[47] WN Kang, FP Kelly, NH Lee, and RJ Williams. State space collapse and diffusion approximation for a network operating under a fair bandwidth sharing policy. *The Annals of Applied Probability*, 19(5):1719–1780, 2009. 5

[48] Toshiyuki Katsuda. State-space collapse in stationarity and its application to a multiclass single-server queue in heavy traffic. *Queueing Systems*, 65(3):237–273, 2010. 5

[49] Sanjit Kaul, Marco Gruteser, Vinuth Rai, and John Kenney. Minimizing age of information in vehicular networks. In *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 350–358. IEEE, 2011. 7

[50] Sanjit Kaul, Roy Yates, and Marco Gruteser. Real-time status: How often should one update? In *2012 Proceedings IEEE INFOCOM*, pages 2731–2735. IEEE, 2012. 7

[51] Frank Kelly. Charging and rate control for elastic traffic. *European transactions on Telecommunications*, 8(1):33–37, 1997. 6

[52] Frank P Kelly, Aman K Maulloo, and David KH Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society*, 49(3):237–252, 1998. 6

[53] Joohwan Kim, Xiaojun Lin, and Ness B Shroff. Locally optimized scheduling and power control algorithms for multi-hop wireless networks under sinr interference models. In *2007 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops*, pages 1–10. IEEE, 2007. 3

[54] John FC Kingman. On queues in heavy traffic. *Journal of the Royal Statistical Society: Series B (Methodological)*, 24(2):383–392, 1962. 5

[55] Antzela Kosta, Nikolaos Pappas, and Vangelis Angelakis. Age of information: A new concept, metric, and tool. *Foundations and Trends® in Networking*, 12(3):162–259, 2017. 7

[56] PR Kumar and Sean P Meyn. Duality and linear programs for stability and performance analysis of queuing networks and scheduling policies. *IEEE Transactions on Automatic Control*, 41(1):4–17, 1996. 3

[57] Satya Kumar and Vinod Sharma. Joint routing, scheduling and power control providing hard deadline in wireless multihop networks. In *2017 Information Theory and Applications Workshop (ITA)*, pages 1–9. IEEE, 2017. 6

[58] V Satya Kumar, Lava Kumar, and Vinod Sharma. Energy efficient low complexity joint scheduling and routing for wireless networks. In *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 8–15. IEEE, 2015. 6

[59] Hyang-Won Lee, Eytan Modiano, and Long Bao Le. Distributed throughput maximization in wireless networks via random power allocation. *IEEE transactions on mobile computing*, 11(4):577–590, 2011. 3, 14, 19, 21, 22, 23, 24, 26, 33

[60] Bin Li and Rayadurgam Srikant. Queue-proportional rate allocation with per-link information in multihop wireless networks. *Queueing Systems*, 83(3-4):329–359, 2016. 4

[61] Zhi-Quan Luo and Shuzhong Zhang. Dynamic spectrum management: Complexity and duality. *IEEE journal of selected topics in signal processing*, 2(1):57–73, 2008. 6

[62] Constantinos Maglaras. Discrete-review policies for scheduling stochastic networks: Trajectory tracking and fluid-scale asymptotic optimality. *The Annals of Applied Probability*, 10(3):897–929, 2000. 4, 37

[63] Athina P Markopoulou, Fouad A Tobagi, and Mansour J Karam. Assessment of voip quality over internet backbones. In *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 150–159. IEEE, 2002. 1

[64] Sean Meyn. Dynamic safety-stocks for asymptotic optimality in stochastic networks. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 4, pages 3930–3937. IEEE, 2004. 4

[65] Sean Meyn. *Control techniques for complex networks*. Cambridge University Press, 2008. 4, 37

[66] Sean P Meyn. Transience of multiclass queueing networks via fluid limit models. *The Annals of Applied Probability*, 5(4):946–957, 1995. 3

[67] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer-Verlag, London, 1993. 81

[68] Masakiyo Miyazawa. Diffusion approximation for stationary analysis of queues and their networks: a review. *Journal of the Operations Research Society of Japan*, 58(1):104–148, 2015. 5

[69] Elie Najm, Rajai Nasser, and Emre Telatar. Content based status updates. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2266–2270. IEEE, 2018. 7

[70] Isidor Pavlovich Natanson. *Theory of functions of a real variable*. Frederick Ungar Publishing Co, New York, 1964. 56, 95

[71] Michael J Neely, Eytan Modiano, and Charles E Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *IEEE Journal on Selected Areas in Communications*, 23(1):89, 2005. 2, 16, 22, 23, 53

[72] Daniel P Palomar and Mung Chiang. Alternative distributed algorithms for network utility maximization: Framework and applications. *IEEE Transactions on Automatic Control*, 52(12):2254–2269, 2007. 6

[73] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014. 6

[74] Theodore S Rappaport. The wireless revolution. *IEEE Communications Magazine*, 29 (11):52–71, 1991. 1

[75] Walter Rudin. *Principles of mathematical analysis*. McGraw-hill New York, 1964. 81

[76] Aleksandr Nikolaevich Rybko and Alexander L Stolyar. Ergodicity of stochastic processes describing the operation of open queueing networks. *Problemy Peredachi Informatsii*, 28 (3):3–26, 1992. 3

[77] Walid Saad, Mehdi Bennis, and Mingzhe Chen. A vision of 6g wireless systems: Applications, trends, technologies, and open research problems. *arXiv preprint arXiv:1902.10265*, 2019. 1

[78] Devavrat Shah. Gossip algorithms. *Foundations and Trends® in Networking*, 3(1):1–125, 2009. 3, 17, 22

[79] Devavrat Shah and Damon Wischik. Switched networks with maximum weight policies: Fluid approximation and multiplicative state space collapse. *The Annals of Applied Probability*, 22(1):70–127, 2012. 4

[80] Devavrat Shah, NC David, and John N Tsitsiklis. Hardness of low delay network scheduling. *IEEE Transactions on Information Theory*, 57(12):7810–7817, 2011. 6

[81] Tejal Shah, Ali Yavari, Karan Mitra, Saguna Saguna, Prem Prakash Jayaraman, Fethi Rabhi, and Rajiv Ranjan. Remote health care cyber-physical system: quality of service (qos) challenges and opportunities. *IET Cyber-Physical Systems: Theory & Applications*, 1(1):40–48, 2016. 1

[82] Sanjay Shakkottai and Alexander L Stolyar. Scheduling for multiple flows sharing a time-varying channel: The exponential rule. *Translations of the American Mathematical Society-Series 2*, 207:185–202, 2002. 3

[83] Vinod Sharma, D Prasad, and Eitan Altman. Opportunistic scheduling of wireless links. *Managing Traffic Performance in Converged Networks*, pages 1120–1134, 2007. 3

[84] Changyang She and Chenyang Yang. Energy efficiency and delay in wireless systems: Is their relation always a tradeoff? *IEEE Transactions on Wireless Communications*, 15 (11):7215–7228, 2016. 6

[85] Rahul Singh and PR Kumar. Throughput optimal decentralized scheduling of multihop networks with end-to-end deadline constraints: Unreliable links. *IEEE Transactions on Automatic Control*, 64(1):127–142, 2018. 6

[86] Eleni Stai, Symeon Papavassiliou, and John S Baras. Performance-aware cross-layer design in wireless multihop networks via a weighted backpressure approach. *IEEE/ACM Transactions on Networking*, 24(1):245–258, 2014. 6

[87] Alexander L Stolyar. On the stability of multiclass queueing networks: a relaxed sufficient condition via limiting fluid processes. *Markov Processes and Related Fields*, 1(4):491–512, 1995. 3

[88] Alexander L Stolyar. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability*, 14(1): 1–53, 2004. 5, 79, 97, 98, 99, 101

[89] Alexander L Stolyar. Tightness of stationary distributions of a flexible-server system in the halfin-whitt asymptotic regime. *Stochastic Systems*, 5(2):239–267, 2015. 5

[90] Vijay G Subramanian and Douglas J Leith. Draining time based scheduling algorithm. In *2007 46th IEEE Conference on Decision and Control*, pages 1162–1167. IEEE, 2007. 4

[91] Rajat Talak, Sertac Karaman, and Eytan Modiano. Distributed scheduling algorithms for optimizing information freshness in wireless networks. In *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5. IEEE, 2018. 7, 112, 114, 115, 117

[92] Chee Wei Tan, Mung Chiang, and Rayadurgam Srikant. Fast algorithms and performance bounds for sum rate maximization in wireless networks. *IEEE/ACM Transactions on Networking (TON)*, 21(3):706–719, 2013. 6

[93] Leandros Tassiulas. Linear complexity algorithms for maximum throughput in radio networks and input queued switches. In *Proceedings. IEEE INFOCOM'98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No. 98*, volume 2, pages 533–539. Ieee, 1998. 3

[94] Leandros Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 31(12), 1992. 2

[95] Tolga Tezcan and JG Dai. Dynamic control of n-systems with many servers: Asymptotic optimality of a static priority policy in heavy traffic. *Operations Research*, 58(1):94–110, 2010. 5

[96] John Von Neumann. *Functional operators. Volume II, The Geometry of orthogonal spaces.* Princeton University Press, 1950. 42

[97] Pradeep Chathuranga Weeraddana, Marian Codreanu, Matti Latva-aho, Anthony Ephremides, and Carlo Fischione. Weighted sum-rate maximization in wireless networks: A review. *Foundations and Trends® in Networking*, 6(1–2):1–163, 2012. 6, 9, 14, 39

[98] Ward Whitt. *Stochastic-process limits: an introduction to stochastic-process limits and their application to queues*. Springer Science & Business Media, 2002. 3, 4, 78

[99] Ruth J Williams. Diffusion approximations for open multiclass queueing networks: sufficient conditions involving state space collapse. *Queueing systems*, 30(1-2):27–88, 1998. 4

[100] Roy D Yates and Sanjit K Kaul. Status updates over unreliable multiaccess channels. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 331–335. IEEE, 2017. 7

[101] Roy D Yates and Sanjit K Kaul. The age of information: Real-time status updating by multiple sources. *IEEE Transactions on Information Theory*, 65(3):1807–1827, 2019. 7

[102] Heng Qing Ye and Hong Chen. Lyapunov method for the stability of fluid networks. *Operations Research Letters*, 28(3):125–136, 2001. 4

[103] Heng-Qing Ye and David D Yao. Diffusion limit of fair resource control—stationarity and interchange of limits. *Mathematics of Operations Research*, 41(4):1161–1207, 2016. 5

[104] Gil Zussman, Andrew Brzezinski, and Eytan Modiano. Multihop local pooling for distributed throughput maximization in wireless networks. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pages 1139–1147. IEEE, 2008. 3