

Simple Cross-Layer Network Utility Maximization Framework for Multiflow Mobile Ad-Hoc Networks

by

Ammar Mottie Alhosainy

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Carleton University
Ottawa, Ontario

©2017

Ammar Mottie Alhosainy

Abstract

Efficiently using the network resources of Mobile Ad-hoc NETWORKS (MANETs) is challenging. The absence of a centralized administration leads to a congestion problem (Transport layer). The flows are usually routed through shortest routes, typically through the same central part of the network (Network layer). Communicating via shared wireless links raises a contention problem (MAC layer). Multi-hop transmissions cause flows not only to interfere with each other, but also with themselves.

We focus on jointly solving the contention and congestion distributed control problem in a bounded queue MANETs. The resulting flow rates satisfy fairness criteria according to a given Network Utility Maximization (NUM) function. In recent years a number of papers have presented solutions to the same problem based on NUM algorithms. However, this work typically necessitates either complex computations, heavy signalling/control overhead, and/or approximated sub-optimal results. In this work, we employ and adapt the IEEE 802.11 protocol in the NUM with a simple and efficient queue management mechanism. Unlike the majority of the published work in this area, we focus on the feasibility of the proposed solution in case of random static and mobile networks considering the overheads and the signalling methods.

We propose a novel algorithm that jointly solves the congestion, multipath routing, and contention distributed control problem for MANETs. The objective is to find the end-to-end optimal source rates at the transport layer, sub-flow rates for each path of the multipath sessions at the network layer, and persistence probability at the MAC layer. The primal problem formulation is a non-convex, non-separable NUM optimization. By introducing new variables, applying certain transformations, and using an analogy based on Ohm's law, we develop a distributed algorithm that can find the optimal solution for general concave utility functions.

The algorithms are implemented in NS-3 and evaluated against non-idealistic scenarios, i.e. link failures, message losses, asynchronous updates, and with the presence of inaccurate topology information. We evaluate the overhead and signalling associated with the algorithms quantitatively and qualitatively and provide absolute gain

values. The results show that the proposed algorithms significantly outperforms layered approaches, using standard protocols such as TFRC.



*To my mother, Sanaa, for
making me who I am,
and to the memory
of my father,
Mottie*



Acknowledgments

Thanks to Almighty Allah for blessing me with the strength to complete this Ph.D. thesis.

I would like to thank my supervisor, Prof. Thomas Kunz, for his support, guidance, and insistence on achieving high quality research. The numerous fruitful discussions and arguments that we have had during my study have made a significant difference on my way of thinking and my work.

I would like also to thank Prof. John Chinneck for sharing his knowledge and his many recommendation letters.

I thank my sister Fatema, my brothers Moaz and Mohamed, and my friend Ahmad Alkabbany for their support and encouragement.

Thanks to everyone who support me during the course of my Ph.D. studies.

Table of Contents

Abstract	ii
Acknowledgments	vi
Table of Contents	vii
List of Tables	x
List of Figures	xi
List of Abbreviations	xiii
List of Symbols	xv
1 Introduction	1
1.1 Research Motivations	2
1.2 Research Direction	4
1.3 Research Contributions	5
1.3.1 IEEE 802.11 MAC Protocol and Active Queue Management	5
1.3.2 Novel Multipath Simple Network Utility Maximization Algorithm	6
1.3.3 Novel Realistic Coordination Methods	6
1.4 Organization of the Thesis Document	7
2 Literature Review	8
2.1 NUM Based Congestion Control	10
2.2 Multipath NUM	12
2.3 Summary	14

3	Algorithm Realization	15
3.1	Algorithm Selection	15
3.2	IEEE 802.11 MAC Integration	18
3.2.1	Optimization Model and Decomposition	20
3.2.2	Distributed Solution	21
3.2.3	Contention Window Adjustment	23
3.3	Active Queue Management	23
3.4	Algorithm Step-Size Adaptation	25
3.5	The SNUM Algorithm	27
3.6	Summary	28
4	Initial Performance Studies	30
4.1	MAC Efficiency	30
4.2	Multipath Effect	33
4.2.1	Number of Paths	33
4.2.2	Path Selection	39
4.3	Summary	40
5	Multipath SNUM	43
5.1	The mNUM Algorithm	43
5.1.1	Problem Formulation and Notation	43
5.1.2	Convex Formulation	45
5.1.3	Dual Decomposition and the Distributed Algorithm	48
5.1.4	Numerical Example	53
5.2	mNUM Realization	55
5.3	Routes Search	55
5.3.1	Search Method	56
5.3.2	Feeding Paths to mSNUM	57
5.3.3	The Active and the Backup Lists	58
5.4	Summary	60
6	The Framework and the Algorithm Implementation	61
6.1	The Framework	61
6.2	Prices	62
6.3	Vertical Coordination	63

6.3.1	Transport Layer	63
6.3.2	Network Layer	63
6.3.3	MAC Layer	65
6.4	Horizontal Coordination	65
6.4.1	Individual Update Messages	66
6.4.2	Piggybacking other Packets	67
6.4.3	Link and Path Failures	70
6.5	Summary	71
7	NS-3 Simulation Results	72
7.1	SNUM Simulations	74
7.1.1	Fixed Topologies with Static Routes	74
7.1.2	Static Networks	78
7.1.3	Mobile Networks	81
7.2	mSNUM Simulations	83
7.2.1	Static Networks	83
7.2.2	Mobile Networks	89
7.3	Summary	90
8	Conclusions and Future Work	92
8.1	Conclusion	92
8.2	Future Work	94
8.2.1	Better Capacity Estimate	94
8.2.2	Contention-Free MAC Protocols	95
8.2.3	Gateway Selection Problem and Network Lifetime	95
8.2.4	Flexible CLD Framework	96
	List of References	98
	Appendix A KKT Optimality Proofs	104

List of Tables

3.1	Distributed Implementation of SNUM	28
4.1	Path Selection Ordered Descendingly Based on the Throughput.	42
5.1	mNUM Algorithm.	52
5.2	mNUM Optimum vs. Global Optimum	55
7.1	Comparison Results SNUM vs. TFRC for Three Topologies	76
7.2	Static Network Topologies	79

List of Figures

3.1	The Aggregate Session Rates vs. Number of Iterations.	17
3.2	QF for different Z values and $Q_{n_{max}} = 400$	25
3.3	SNUM Behaviours with and without Adaptive Step-Size.	26
4.1	Topology of a 3x3 Grid Network.	31
4.2	Active Links for a Multipath Session from Node A to J.	32
4.3	Session with n Hops and m Mutually Interference-Free Paths.	34
4.4	Multihop Session Throughput with Interference Free Paths.	34
4.5	ALOHA Hops Prices for 10 Hops Session.	35
4.6	Multihop Session Throughput with Interfering Paths.	36
4.7	The Multihop Session Throughput with Interference-Free Paths.	37
4.8	Session Throughput per Path.	38
4.9	IEEE 802.11 Hops Prices for 10 Hops Session.	38
4.10	Multipath Session with Different Path Lengths and Mutual Interference.	39
4.11	Different Multipath Types.	41
5.1	The Equivalent Session Prices.	49
5.2	Network Topology and the Associated Multipath Sessions.	53
5.3	The Evolution of Path Rates, Splitting Factor, Path Prices, and Session Rates vs. Number of Iterations.	54
5.4	Searching and Path Feeding Method.	59
6.1	Illustration of the CLD Framework.	62
6.2	CLD Flowchart per Node.	64
6.3	IPv4 Option Filed TLV.	68
6.4	Illustration of Overhead Piggybacking.	68
6.5	Session Path with a Broken Link.	70
7.1	Experimental Wireless Network Topologies.	74
7.2	The Throughput of each Session.	75
7.3	Throughput Convergence of Topology (b).	77

7.4	Utility Loss Percent due to Update Information Loss for Topologies (b) and (c).	78
7.5	Rate Standard Deviation increase due to Update Information Loss for Topologies (b) and (c).	79
7.6	The Effect of Aggressiveness Parameter Z on SNUM Average Delay and Throughput.	80
7.8	Average Delay, Overhead, and Packet Loss for Static Topologies. . . .	80
7.7	Aggregate Log rates/session for Static Topologies.	81
7.9	Aggregate Session Log Rates for Mobile Topologies.	82
7.10	Average Delay, Overhead, and Packet Loss for Mobile Topologies. . . .	82
7.11	Topology 1, 9 Nodes and 3 Sessions.	84
7.12	Topology 2, 15 Nodes and 3 Sessions.	84
7.13	The Log Rates for two Topologies with the 95% Confidence Intervals.	85
7.14	The Aggregate Throughput.	86
7.15	Percentage of Packet Losses.	87
7.16	Overhead Rates.	87
7.17	The Effect of Aggressiveness Parameter Z on mSNUM Network's Average Delay and Throughput.	88
7.18	The Number of Paths used by mSNUM with different Z Values. . . .	88
7.19	The Log Rates for two Topologies with the 95% Confidence Intervals.	89
7.20	Percentage of Packet Losses.	89
7.21	Overhead Rates.	90
8.1	Wireless Network Connected to a Fixed Network through Gateways. . .	96

List of Abbreviations

ACK	ACKnowledgement
bps	bits-per-second
CLD	Cross-Layer Design
CSMA-CA	Carrier Sense Multiple Access-Collision Avoidance
CW	Contention Window
CTS	Clear To Send
DCF	Distributed Coordination Function
DIFS	DCF InterFrame Space
DSR	Dynamic Source Routing
ID	IDentifier
IoT	Internet of Things
IP	Internet Protocol
LHS	Left Hand Side
mNUM	multipath Network Utility Maximization algorithm
mSNUM	multipath Simple Network Utility Maximization algorithm
MAC	Medium Access Control
MANET	Mobile Ad-hoc NETwork
MP-DSR	MultiPath-Dynamic Source Routing
NUM	Network Utility Maximization

OLD	Oblivious Layer Design
OLSR	Optimized Link State Routing
PDU	Protocol Data Unit
QF	Queue-length-based Factor
RERR	Route ERRor
RREP	Route REPlY
RREQ	Route REQuest
RTT	Round Trip Time
RTS	Request To Send
SIFS	Short InterFrame Space
SNUM	Simple Network Utility Maximization algorithm
TDMA	Time Division Multiple Access
TCP	Transmission Control Protocol
TFRC	TCP Friendly Rate Control
TLV	Type-Length-Value
UDP	User Datagram Protocol

List of Symbols

α_s^ℓ	Flow fraction on link ℓ contributed by source s .
$\alpha_{s,m}^\ell$	Flow fraction on link ℓ contributed by the path m of session source s .
$\boldsymbol{\alpha}$	Vector of flow fractions.
β	Step-size.
B_n	Bianchi's throughput.
c_l	Capacity of link ℓ .
c_{lQ}	Capacity of link ℓ adjusted by the queue factor.
\mathbf{c}	Vector of link capacities.
C_ℓ	Physical capacity of link ℓ .
C	A positive constant.
CW_{opt}	Optimum contention window.
CW_n	Contention window of node n .
D	Dual function.
$E(M)$	Average message payload.
F	Instability flag ratio.
G	A network graph.
H	Number of hops.
ℓ	A network link.
\mathcal{L}	Set of network links.

$\mathcal{L}_{out}(n)$	Set of outgoing links from node n .
$\mathcal{L}_{in}(n)$	Set of incoming links to node n .
$\mathcal{L}(s)$	Set of links utilized by the source s .
$\mathcal{L}_I(n)$	Set of links whose reception are adversely affected by the transmission of node n , excluding $\ell \in \mathcal{L}_{out}(n)$.
$\mathcal{L}_s(m)$	Set of sequential links used by session source s for path m .
λ_s	Session price.
λ^ℓ	Link price.
λ_n	Node price.
$\lambda_{s,m}$	Session-path price, path price of session source s .
λ_s^ℓ	Lagrange parameter (price) associated with single path session s on link ℓ , session-link price, single path session hop price.
$\lambda_{s,m}^\ell$	Lagrange parameter (price) associated with session-path (s, m) on link ℓ , multipath session hop price.
$\lambda_{s,m}^{\ell_{fair}}$	Fair price associated to the session-path s, m on link ℓ .
λ	Vector of Lagrange parameters (prices).
λ^*	Vector of the optimum set Lagrange parameters (optimum prices).
Λ	Projection operator for the flow fraction α_s^ℓ .
$\mathcal{N}_I(\ell)$	Set of nodes whose transmission impact the receiver of the link ℓ .
m	A network path route.
M	Number of paths.
\mathcal{M}_s	Set of paths used by the session source s .
\mathcal{M}_ℓ	Set of session-paths (s, m) sharing the link ℓ .
MSG_{Size}	Message size.
n	A network node.
N	Number of nodes.

\mathcal{N}	Set of network nodes.
$\mathcal{N}_{CA}(n)$	Set of nodes in the same contention area of node n .
$\mathcal{N}_I(\ell)$	Set of nodes whose transmission adversely affect the reception of link ℓ , excluding the transmitter of link ℓ .
p_ℓ	Link persistence probability.
Π	Projection operator for p_ℓ .
\mathbf{p}	Vector of links persistence probabilities.
P_{tr}	Probability that there is at least one transmission in the considered time slot.
P_s	Probability of successful transmission.
Pr_{bytes}	Size of the price register in bytes.
Ψ	Projection operator of p_ℓ and q_n .
q_n	Node persistence probability.
\mathbf{q}	Nodes persistence probability vector.
Q_n	Queue length at node n .
$Q_{n_{max}}$	Maximum queue length at node n .
QF_n	The queue factor at node n .
r_ℓ	Rate of link ℓ .
r_s	Rate of source s .
r_s^ℓ	Rate of source s allocated on link ℓ .
$r_{s,m}$	Rate of source s allocated on path m .
\mathbf{r}	Rates vector.
\mathcal{R}	Routing matrix.
s	A network session.
\mathcal{S}	Set of network sources.
$\mathcal{S}(\ell)$	Set of sources utilizing link ℓ .

σ	Duration of backoff timeslot in IEEE 802.11.
t	Iteration number.
T_s	The average time the channel is sensed busy due to a successful transmission.
T_c	The average time the channel is sensed busy during a collision.
θ	Utility fairness parameter.
$U_s()$	Utility function of source s .
U^z	Transformed utility function for log rates.
$v_{s,m}$	Splitting factor for session s over path m .
\mathbf{v}	Vector of splitting factors.
x	An auxiliary variable.
\mathbf{x}	Vector of auxiliary variables.
z_ℓ	Log rate of the link rate r_ℓ .
$z_{s,m}$	Log rate of the session-path rate $r_{s,m}$.
Z	Aggressiveness parameter.
$ A $	Number of elements in A .
$\{a\}^+$	Defined as $\max(a, 0)$.

Chapter 1

Introduction

Mobile Ad-hoc Networks (MANETs) are infrastructure-less networks of potentially mobile nodes communicating via wireless links in a multi-hop fashion. The main advantages of the MANET technology, such as the non-stationary and decentralized administration, make it easier to build low-cost networks that cover potentially very large geographical areas. MANETs could even be used to provide Internet access in the developing world. For example, project Loon [1] aims to connect the rural areas in developing countries, which lack a communication infrastructure, providing them with Internet services using floating Balloons in a multihop wireless fashion.

With the main advantages come the main challenges for MANETs: mobility, the absence of a centralized administration, and a severe resource limitation. These challenges raise congestion, contention, routing and fairness problems that require adaptive solutions to handle the constant change of the network topology. The need for a network design and protocols that increase coordination between network nodes and layers and increases the network efficiency is crucial in MANETs.

More than 40 years ago, in order to reduce design complexity, network designers organized protocols and the network hardware and software that implement the protocols in layers. It helped in optimizing individual protocols to make the network more efficient in one or more aspects (e.g. energy, delay, throughput, or network capacity). However, a good protocol design for a specific layer can negatively affect the performance of other layers, for example, the routing protocol at the network layer that finds the shortest path can severely harm the network fairness and worsen the congestion problem at the transport layer [2]. In general, there is always a tradeoff between modularity and efficiency, simplicity and optimization.

The communication via shared wireless links raises a contention problem (MAC

layer). The flows are usually routed through shortest routes, typically through the same central part of the network (Network layer). The absence of a fixed infrastructure and centralized administration add an efficiency, congestion, and fairness problems where flows or end-to-end data sessions are inefficiently allocated to already congested areas, which wastes underused capacity of other parts in the network (Network and Transport layers). Overloading the available network capacity degrades the overall network throughput which requires congestion control (Transport layer). Multi-hop transmissions cause flows not only to interfere with each other but also with themselves. Finally, node mobility not only limits the energy resources of the nodes but also requires adaptive solutions to handle the constant change of the network topology and the node's local contention neighbourhood.

1.1 Research Motivations

It is inefficient to use a strictly layered architecture in MANET, the lack of coordination between layers in that architecture limits the performance of the design. Sharing knowledge about layer states and conditions is essential for optimizing the performance of any MANET architecture.

A cross-layer network design provides vertical coordination between network layers. A cross-layer framework that has a distributed horizontal coordination system between nodes provides flexibility and efficiency to the network that compensates for the absence of a centralized administration. The key idea here is that, rather than individual protocol layers solving one or a subset of the stated problems, potentially conflicting with the solution of other problems at other layers, all layers jointly address these problems to derive a consistent, optimal operation of the network. For that reason, Network Utility Maximization (NUM) based Cross-Layer Design (CLD) was proposed to optimally manage network resources exploiting the coordination between layers.

We can divide the cross-layer approaches into two general categories. The first category could be called the revolutionary category that focuses on the performance more than the compatibility with traditional layered networks [3–5]; consequently, it does not maintain the boundaries between layers. One of the approaches in this category is to combine two or more layers into one; this will help to design more efficient and application-specific protocols. But as the number of combined layers

increases, the complexity will increase; it will also limit the ability to communicate with other entities (e.g. IP networks) due to the absence of strict layer boundaries.

Another revolutionary approach is to build a middleware (centralized optimizer) to collect information, coordinate between layers, and control parameters and take decisions at each layer. This approach violates the layered network architecture because all layers should send their internal protocol parameters and information to the middleware and then the middleware decides what is best for all in order to optimize the performance. These revolutionary approaches can be seen in wireless sensor network applications [6, 7]. Because they are usually designed for a certain task and rarely need to communicate directly with a standard IP-based network except through a compatible gateway, this approach is feasible in this context.

The second category of cross-layered designs is an evolutionary one that focuses on compatibility more than performance. The idea of this approach is to exchange states and conditions between two or more layers with less change in the layers' implementations [8]. This will preserve the layer boundaries and functionalities to maintain compatibility with other entities. There are a lot of interactions and dependencies that can then be exploited to optimize the network performance through coordination between network layers.

Many proposed distributed CLD frameworks, supported by solid mathematical foundations, do not reach the stage of implementation in real networks due to different reasons; the overly simplistic communication model used is one major reason. The communication problems appear in the infrastructure-less networks that depend on exchanging information to achieve horizontal coordination between nodes. The required level of communication resilience may not be achieved in realistic network environments. Commonly, when testing the stability and robustness of algorithms and designs, only deterministic factors are typically considered. For example, assuming that accurate topology information is available at each node in the network is problematic in the case of dynamic topologies. These type of assumptions increase the gap between theoretical performance and the actual behaviour and performance of network protocols under more realistic conditions. Therefore, modelling, simulating, and evaluating distributed network algorithms requires foreseeing a variety of possible situations and using what-if analysis paradigm.

A well-connected wireless network can provide the capability for multipath routing that can maximize the aggregate session rates and maintain connectivity in multihop

sessions. However, unlike wired networks [9], the wireless interference changes the channel capacity seen by each node that leads to different capacities for each link. Moreover, the interference adversely affects the total network capacity and throughput. For that reason, solving the multipath routing problem is tied to the congestion and contention problems. The optimization problem including three layers, Transport, Network, and MAC is always either complex, approximated, sub-optimal and/or requires centralized administration [9, 10].

Our focus is to provide a basic tool for rate optimization that provides end-to-end congestion and fairness control which minimizes the number of lost packets in the network. The tool is suitable for rate adaptive applications (elastic rates). However, this tool does not provide reliable transmission, for that reason, the applications with high reliability requirements would need a layer on top of our optimization tool to provide reliability.

1.2 Research Direction

The performance seen at the level of the application of any network is a function of the parameters at all the layers below it. Optimizing the performance can be achieved by including all these parameters into one optimization problem, but this may not be possible in real life. To find a solution to this problem, we set some desired features to follow in our research:

- The optimization should be a distributed algorithm, i.e. the decisions should be made separately at each node and at each layer.
- The layered network architecture should be maintained, i.e. the decisions related to each layer should be made at this particular layer. Each layer must collect the required information horizontally (from different nodes) and vertically (from different layers).
- The goal is to provide an online solution that can show resilience in a realistic network environment, i.e. in the presence of noise, topology changes, and packet losses.

For example, in the case of a MANET, there is no fixed or centralized infrastructure for the network, so in order to maintain scalability and mobility, we need

a distributed optimization algorithm at each node that collects required information horizontally from the neighbouring nodes and provides it to its particular layers. At the same time, to maintain the network layer architecture, each layer needs to collect relevant information vertically from the other layers in the node and take its own decisions, while at the same time participating in the overall optimization process. The horizontal coordination needs to withstand certain noise levels, packet losses, and inaccurate topology information.

1.3 Research Contributions

We propose a realistic and modular CLD framework that optimizes the Transport, Network, and MAC layers problems, preserves compatibility and provides substantial performance.

1.3.1 IEEE 802.11 MAC Protocol and Active Queue Management

We propose a novel way to integrate the IEEE 802.11 MAC protocol in the NUM, replacing the simple ALOHA protocol used in earlier work [11]. Our approach simplifies the NUM algorithm by using a channel capacity estimation based on Bianchi's model [12]. One of the main issues that arises when moving from theory to practice in rate control algorithms and NUM are the queues and delays in the network. We propose a technique to incorporate queue management into the optimization process that solves the congestion and contention problems without adding more constraints, increasing complexity or overheads. The technique acts as an Active Queue Management (AQM) that uses the NUM coordination parameters to notify the data source about incipient congestion. We are also adapting the algorithm step-size to speed up the convergence and to avoid instability that comes with dynamic topologies where nodes (and links) continuously appear and disappear. We call our adapted algorithm with AQM and IEEE 802.11 MAC protocol the Simple Network Utility Maximization (SNUM) algorithm.

This contribution is published in the following peer reviewed papers:

- *Ammar Alhosainy and Thomas Kunz, "Cross-layer Design for Multihop MANETs*

Utility Optimization with Active Queue Management,” IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2017), Montreal, QC, Canada, October 8-13, 2017.

- *Ammar Alhosainy and Thomas Kunz, ”Cross-layer design for IEEE 802.11 wireless ad-hoc network utility maximization with active queue management,” Proceedings of the Sixth International Conference on Selected Topics in Mobile & Wireless Networking (MoWNet’17), Avignon, France, May 2017.*

1.3.2 Novel Multipath Simple Network Utility Maximization Algorithm

We developed a multipath Simple Network Utility Maximization (mSNUM) algorithm that considers the end-to-end flow congestion, contention between flows, multipath sub-flows, and multipath routing problems in a distributed fashion. The algorithm can optimally split the flows over a set of paths. Among the available paths, some paths will not be neglected while others will be assigned a certain fraction of the traffic such that we maximize the total network utility. Unlike prior work, our reformulated problem is convex with a single relaxed constraint that can achieve optimal results in a distributed manner. The optimization is achieved by the coordination vertically between network layers and horizontally between nodes using a single coordination parameter. This provides stability in the algorithm, facilitates keeping boundaries between layers, and reduces communication overhead.

This contribution has been published in the IEEE Communications Letters.

- *A. Alhosainy; T. Kunz, ”Joint Optimal Congestion, Multipath Routing, and Contention Control for Wireless Ad-hoc Networks,” to appear in IEEE Communications Letters, accepted Aug. 14th, 2017. 10.1109/LCOMM.2017.2739139*

1.3.3 Novel Realistic Coordination Methods

The stability of the cross-layer optimization algorithm in many cases can be proved mathematically [10, 11, 13] assuming perfect transmissions, synchronization between nodes, and availability of accurate topology information. However, the actual performance stays uncertain due to non-deterministic factors in real network environments. We provide a realistic implementation for SNUM and mSNUM, where boundaries

and modularity of the network stack layers are maintained. The algorithms then are evaluated against link failures, message losses, and with the presence of inaccurate topology information in dynamic networks. We use ns-3 to provide the realistic wireless network implementation in case of static and dynamic topologies. We evaluate the overhead and signaling associated with the algorithms quantitatively and qualitatively and provide absolute gain values.

The work regarding the algorithm realistic implementation was published in one journal and two conference papers.

- *Ammar Alhosainy and Thomas Kunz, "Robustness, stability, and gains of utility maximization algorithms for mobile ad-hoc networks," International Journal of Wireless Information Networks, Vol. 23, No. 4, pages 257-272, December 2016.*
- *Ammar Alhosainy, Thomas Kunz, Li Li, and Philip J. Vigneron, "Cross-layer design gains in MANETs," Proceedings of the 13th IEEE IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2014), pp. 8-14, Piran, Slovenia, June 2014.*
- *Ammar Alhosainy, Thomas Kunz, and Li Li, "Robustness and stability of utility maximization algorithms for MANETs," Proceedings of the 13th IEEE IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2014), pp. 15-22. Piran, Slovenia, June 2014.*

1.4 Organization of the Thesis Document

Chapter 2 discusses the related work and the selected basic algorithm for our work. Chapter 3 discusses the algorithm realization using the IEEE 802.11 MAC, the proposed queue management mechanism, and the step-size adaptation. The study about the used MAC protocol efficiency and the effect of the multipath routing in NUM is provided in Chapter 4. The novel multipath algorithm and multipath routes selection are explained in Chapter 5. Chapter 6 explains the CLD framework and the algorithm implementation with vertical and horizontal coordination. The simulations and results using NS-3 are provided in Chapter 7 followed by our conclusions in Chapter 8.

Chapter 2

Literature Review

The basic NUM formulation is known as Monotropic Programming [14, 15],

$$\begin{aligned} \max \quad & \sum_s U_s(r_s) \\ \text{s.t.} \quad & \mathcal{R}\mathbf{r} \leq \mathbf{c} \end{aligned} \tag{2.1}$$

where the session rate vector $\mathbf{r} \geq \mathbf{0}$ is the only optimization variable set, the routing matrix \mathcal{R} and the link capacity vector \mathbf{c} are constant. The Utility functions U_s are often chosen to be monotonically increasing concave functions of the session rate r_s .

Using the network utility maximization framework presented in [14] and the trade-off formula between fairness and efficiency proposed in [16], by controlling the θ parameter in the following utility function, we can assign utility values to each network user as a function of their throughput r that reflect different fairness criteria:

$$U_\theta(r) = \begin{cases} (1 - \theta)^{-1} r^{1-\theta} & \theta \neq 1, \\ \log(r) & \theta = 1, \end{cases} \tag{2.2}$$

To maximize the network throughput, the utility function will be simply equal to r when $\theta = 0$, for achieving proportional fairness between users we set $\theta = 1$, and min-max fairness is achieved as $\theta \rightarrow \infty$. The tutorials on decomposition and cross-layer optimization in [17] and [18] provide several examples on how to decompose the NUM problem based on the network stack layers functions.

Unlike wired networks, the wireless interference changes the channel capacity seen by each node that leads to a different capacity for each link. Moreover, the interference adversely affects the total network capacity and throughput. For that reason,

the link capacity variations need to be considered in the wireless network model. Several works have been published that combine many network parameters into a single Network Utility Maximization (NUM) problem that is implemented in a Cross-Layer Design (CLD) [9, 15]. Due to the complex nature of wireless networking, the contention and interference between links, the optimization problem becomes generally nonconvex [10, 19]. That requires centralized control [20], approximation, and/or complex calculations [10], in some cases the optimization process does not have a polynomial time solution [21].

Using convex optimizations, researchers proposed different approaches to find the optimum values of different parameters at the MAC and the Transport layers that jointly solve the contention and congestion control problem in a distributed fashion [11, 22–25]. These approaches require no centralized coordinator to optimize the performance of contention-based medium access protocols, such as ALOHA, and can optimize multihop end-to-end session rates. In [26] the authors formulated the problem to maximize the network utility by assigning the rates and finding the optimum node probabilities to access the medium. They found the optimum solution but only in a centralized network. The work in [27, 28] maximized the network utility using scheduling solutions to assign rates under delay constraint considering fixed link capacities.

Considering the mentioned basic assumptions in our research direction, we studied different network utility maximization based congestion and contention control algorithms for wireless ad-hoc networks, using single path and multipath routing. We focus on optimally assigning end-to-end fair session rates at the transport layer and sub-flow rates on multipath routes at the Network layer. The rate assignment is achieved by controlling the medium access opportunity given to each node and link at the MAC layer. In our work, we assume cooperating wireless nodes with a common objective.

Video streaming, an example of an application requiring high data rates, has become the source of more than half of the aggregate Internet traffic [29]. It also requires delivering a certain level of QoS to the end user and optimally utilizing the network resources. One of the technologies currently used to overcome the unpredictability of the underlying network capacity is through playback buffering. The more bursty traffic is generated, the larger a buffer is required. Having a predictable network capacity and consistent rates help to minimize the buffer required for such applications

and enhances the QoS provided.

2.1 NUM Based Congestion Control

In a random access based MAC protocol each node adjusts its own persistence probability q_n and the transmission probability of each of its outgoing links p_ℓ . The successful transmission, in this case, is calculated according to [30] as follows,

$$\text{Probability of Success} = p_\ell \prod_{n \in \mathcal{N}_I(\ell)} (1 - q_n) \quad (2.3)$$

where the $\mathcal{N}_I(\ell)$ vector contains all the nodes whose transmission will impact the receiver of link ℓ , which are the receiver node as well as all its neighbours.

For such a network, the main constraint of the Optimization Problem (2.1) will control the link rates to minimize collisions and ensure high system throughput. The rate, in bits-per-second (bps), at each link in the network will have a rate constraint as follows,

$$r_\ell \leq C_\ell p_\ell \prod_{n \in \mathcal{N}_I(\ell)} (1 - q_n) \quad (2.4)$$

$$r_\ell = \sum_{s \in \mathcal{S}(\ell)} r_s, \quad (\mathbf{q}, \mathbf{p}) \in \Psi,$$

where r_ℓ is the link rate, r_s is the rate of session s , $\mathcal{S}(\ell)$ is the set of sessions using link ℓ , and C_ℓ is the physical capacity of link ℓ , \mathbf{q} and \mathbf{p} are the persistence probability vectors of the nodes and links, respectively, $\Psi = \{\sum_{\ell \in \mathcal{L}_{out}(n)} p_\ell = q_n, 0 \leq p_\ell \leq 1, 0 \leq q_n \leq 1\}$ is the projection operator of p_ℓ and q_n . $\mathcal{L}_{out}(n)$ is the set of outgoing links from node n . Constraint (2.4) is non-convex and requires each transmitter node in the network to be aware of the MAC layer attempt probabilities of the receiver as well as the attempt probabilities of the 1st hop neighbours of the receiver.

Wang *et al.* [22] proposed two approaches to solve the non-convexity of the problem. The first one is the dual-based approach, in which the optimization problem is divided into two sub-problems, one for the transport layer and one for the link layer. The two sub-problems work at different time scales and are connected via Lagrange parameters. At the sub-problem of the transport layer, the algorithm tries to find the optimum session rates assuming fixed link rates. The iterations continue until the Lagrange parameters converge within a small margin, at that point the session rates

are optimum for the current link attempt probabilities and link rates. The resulting Lagrange parameters, link attempt probabilities, and session rates are communicated to the 1st and 2nd hop neighbors. Then, the MAC layer starts by updating the link attempt probabilities based on the nodes attempt probabilities, the Lagrange parameters, and session rates of the two-hop neighbors of each node. The second approach presented in [22] is the penalty-based method. This method merges the optimization problem of the transport layer with that of the link layer in one optimization problem. To overcome the non-convexity of the problem in Constraint (2.4), the authors presented an equivalent convex model that uses the log of the link rates.

Lee *et al.* in [23] started from the basic NUM approach to maximize the aggregate link rates in the network, not the session rates. The authors solved the convexity problem of Constraint (2.4) by the log representation of the rates $z_\ell = \log(r_\ell)$ as follows,

$$z_\ell = \log(r_\ell) = \log(p_\ell) + \sum_{n \in \mathcal{N}_I(\ell)} \log(1 - q_n)$$

To solve this optimization problem the authors used the Dual approach. As they are optimizing the link rates and not the session rates, there is no need for the nested loop of the transport layer as in the Dual method presented in [22]. We can see that the method proposed in [23] maximizes the link rates and not the session rates, so it is a single hop flow maximization algorithm. The authors of [23] extended their work to maximize the end-to-end flow rates in [24] using the penalty based method, but their method is very similar to the penalty based method presented in [22] with minor changes.

Yu *et al.* [11] extended the work of [23] to change the goal of the algorithm to maximize the end-to-end flows instead of the single hop flows using the Dual based algorithm. In [11], the main difference, and nearly the only one, is the introduction of a new variable α_s^ℓ , which is the fraction of the flow on link ℓ that is contributed by source s . Introducing this variable allows us to replace r_ℓ from the LHS of the network Constraint (2.4) with r_s^ℓ , as follows,

$$r_s^\ell \leq \alpha_s^\ell C_\ell p_\ell \prod_{n \in \mathcal{N}_I(\ell)} (1 - q_n)$$

$$(\mathbf{q}, \mathbf{p}) \in \Psi, \quad \boldsymbol{\alpha} \in \Lambda$$

where $\Lambda = \{\sum_{s \in \mathcal{S}(\ell)} \alpha_s^\ell = 1, 0 \leq \alpha_s^\ell \leq 1\}$ is the projection operator for α_s^ℓ , $\boldsymbol{\alpha}$ is the flow

fraction vector, and r_s^ℓ is the link capacity fraction allocated for the rate of session s . The authors of [11] used the same optimization method employed in [23], they worked with the log of the rate to overcome the non-convexity of the problem and applied the Lagrange optimization method. The published optimization approaches in [11, 22–24] solved the optimization problems analytically without providing means for coordination between network layers. The analytical optimization solutions assumed that the parameter values are instantly accessible to each layer, this is not possible without affecting the layered network architecture [31].

Some work [32–35] considered the queue bounded networks in their NUM to optimize the end-to-end fair session rates in a distributed fashion for multihop wireless networks. In [32, 33], the authors used a stochastic queueing model to estimate the delay and added a constraint in the utility optimization problem. The problem is convex because of the interference model and link capacitance as they solve it for the wired network. In order to incorporate the queue management in the optimization process, [34] and [35] looked at the end-to-end average delay of each session. They need to collect data along the session path and provide it to each link in the session so that each link has information about the delay prices. This information should be collected before any decision can be taken about the delay and queue buildups. The resultant algorithm consumes extra overhead, beside the original NUM overhead, for the delay constraint.

2.2 Multipath NUM

A well-connected wireless network can provide the capability for multipath routing that can maximize the aggregate session rates and maintain connectivity in multihop sessions [36]. Recent work by Pham *et al.* [13] and Jin *et al.* [37] showed multipath routing can maximize bandwidth usage efficiency and increase reliability and fault tolerance through load balancing.

However, unlike wired networks, the interference in a well-connected wireless network adversely affects the total network capacity and throughput [38]. What makes the problem more complicated is that the interference changes the channel capacity seen by each node that leads to a different capacity for each link. Moreover, the interference adversely affects the total network capacity and throughput. For that reason,

solving the multipath routing problem is tied to the congestion and contention problems. Similar to the single path problem Formulation (2.1), the optimization problem in case of multipath can be formulated as follows,

$$\begin{aligned} \max \quad & \sum_s U_s \left(\sum_{m \in \mathcal{M}_s} r_{s,m} \right) \\ \text{s.t.} \quad & \mathcal{R}\mathbf{r} \leq \mathbf{c} \end{aligned} \quad (2.5)$$

where $r_{s,m}$ is the path rate assigned over the path $m \in \mathcal{M}_s$ and \mathcal{M}_s is the set of paths used by the session source s . The utility functions U_s in (2.5) are no longer strictly concave functions due to the linearity between paths $\sum_{m \in \mathcal{M}_s} r_{s,m}$, as a result, the dual of the problem may not be differentiable at every point. The linearity problem affects the stability of the algorithm due to the possibility of algorithm oscillation between paths. Lin *et al.* [39] proposed a solution to the problem using Proximal Optimization Algorithms by adding a quadratic term to the optimization objective. Their optimization objective function became as follows,

$$\sum_s U_s \left(\sum_{m \in \mathcal{M}_s} r_{s,m} \right) - \sum_s \sum_{m \in \mathcal{M}_s} \frac{C_s}{2} (r_{s,m} - x_{s,m})^2$$

where C_s is a positive constant, and $x_{s,m}$ is an additional variable. With the quadratic term, for a given vector \mathbf{x} the objective function becomes strictly concave with respect to the vector \mathbf{r} and a unique solution can be found. The added quadratic term is used as a low-pass filter to prevent the algorithm from oscillating between different solutions and force it to converge. In each iteration, for a given \mathbf{x} , they optimally solve the problem with respect to \mathbf{r} then \mathbf{x} is updated with the new \mathbf{r} and so on. They indicated that the two-loops algorithm solution is not suitable as the inner loop may not have any stopping condition as it keeps updating the auxiliary variable [9].

Supittayapornpong *et al.* [40] solved the convexity and the linearity problem by joining the multipath rates into a harmonic mean rate then split the rate approximately equal over the multipath routes. Their solution idea circumvented the convexity and the linearity problems associated with multipath routing but resulted in a suboptimal solution.

In [10], the authors extended the previous work in [11] and included the network layer in the optimization problem. For that, they added a new constraint for flow conservation. The conservation constraint along with the congestion and contention problem formulation solve the multipath routing. Because of the added constraint,

the optimization problem becomes non-convex, and the authors use a successive approximation approach to solve it. They found an approximated sub-optimal solution, they claim that they can find at least a locally optimum solution, but it requires a large amount of coordination and message passing.

Pham *et al.* [13] developed a hop-by-hop rate control algorithm considering the lossy wireless medium. The authors managed to overcome the convexity problem and provided a convex formulation. However, their solution resulted in a two-loop algorithm where they update and communicate different coordination parameters.

2.3 Summary

In this chapter, we examined different network utility maximization algorithms that optimize the medium access probabilities using the slotted ALOHA protocol at the MAC layer jointly with the end-to-end source rates at the transport layer. Different multipath NUM algorithm have been reviewed. We discussed different methods to overcome the non-convexity and linearity problems associated with the multipath formulations. As soon as some of the sources have the capability to send traffic over multipath, the problem becomes unstable and keeps oscillating between paths. The proposed solutions in the literature are either sub-optimal or require nested convergence loops. In our proposed multipath NUM, we alleviated the linearity between sub-flows and convexified the optimization problem without the use of a quadratic term. Our proposed algorithm can jointly solve the congestion, multipath routing, and contention wireless multihop distributed optimization problems using single relaxed constraint, i.e. using a single coordination parameter and a single loop for optimality convergence. The benefits of using single coordination parameter are to reduce the amount of communication overhead between nodes and also help to keep the boundaries between the network layers.

Chapter 3

Algorithm Realization

In this chapter, we simulate similar basic NUM algorithms that solve similar problems and then we select the most promising one to realize and extend further. We are discussing the integration of the IEEE 802.11 MAC protocol in the NUM, replacing the simple ALOHA protocol used in earlier work [11]. Our approach simplifies the NUM algorithm by using a channel capacity estimation based on Bianchi's model [12]. One of the main issues that arises when moving from theory to practice in rate control algorithms and NUM are the queues and delays in the network. We are integrating an AQM mechanism into the pricing scheme at the MAC layer to deal with the queues in the network. We are also adapting the algorithm step-size to speed up the convergence and to avoid instability that comes with dynamic topologies where nodes (and links) continuously appear and disappear. We call our adapted algorithm with AQM and IEEE 802.11 MAC protocol the Simple Network Utility Maximization (SNUM) algorithm.

3.1 Algorithm Selection

As we are mainly concerned with the practicality of the algorithm, whether it is stable, consistent, and fast in real networks, we evaluate the different algorithms from the practicality perspective. Which means, we are mainly concerned with the number of messages that need to be communicated between nodes, the stability of the algorithm against message loss and asynchronous updates, and how fast the algorithm converges.

There are two main reasons for performance degradation, which appear in any

realistic wireless network with distributed algorithms, which could diverge the distributed algorithm away from the optimum values and/or the stable state. The first one is the absence of time synchronization between nodes. The asynchronous network nodes, executing a distributed iterative algorithm, could diverge as a result of iterations or communication/task mismatch (asynchronous updates) between nodes [41]. The chances of such algorithm divergence increase if the absence of synchronization is combined with update packet losses.

The second reason is inaccurate topology information. Due to mobility or incorrect neighbor information, the neighborhood information stored in the wireless network nodes becomes obsolete. That may have a severe effect on the network performance; many researchers discussed this problem and tried to minimize its effect in a variety of ways [42–45]. Yet, most of the researchers considered this issue a routing problem and studied it from the network layer perspective, neglecting its effect on the other layers.

Considering the previously mentioned two reasons for distributed algorithms performance degradation, we examined and tested the optimization algorithms presented in [11, 22], exploring their strengths and weaknesses. Our goal is to select the algorithm that is fast and needs the minimum amount of information to be exchanged between network nodes, which would result in a more consistent and stable behaviour in a realistic network environment.

The dual-based approach presented in [22] works on two different time scales. Two nested convergence loops are used, the inner loop for the session rate and the outer one for the link attempt probabilities. The penalty-based approach presented in [22] updates the link attempt probabilities and the session rates in the same iteration. Similar to the dual-based method, it requires exchanging messages that carry two different types of information, but at the same time scale. In the dual-based approach presented in [11], there is only one update function used for dual prices on each link that depends only on the link congestion. The prices are the only information that needs to be updated, the rest of the parameters are calculated directly from these prices.

In [22], the chances of instability are higher in the penalty-based approach than in the dual-based approach because, in each iteration, the session rates and the link attempt probabilities are updated depending on each other. In contrast, in the dual-based approach, we first find the optimum session rates at the transport layer, and

then we use the prices to find the optimum link attempt probabilities at the MAC layer. Also, in both approaches, the updated probabilities depend on the old ones and the old prices, which cause the final value to be sensitive to the initial values of the probabilities. The authors did not provide a proof to support the parameter update formulas, the only evidence that the provided formulas can find the optimum are the reported experimental results. In [22], we need to communicate two parameter values that are used in sequential updates, which increases the algorithm vulnerability to packet loss and asynchronous updates. The convergence speed in distributed consensus algorithms varies based on the complexity of the updates and number of updated variables, it becomes worse in the presence of a noisy medium and dynamic topology [46].

In the dual-based method [11], the prices are the only parameter that needs to be updated, then the probabilities and the session rates are recalculated using only the updated prices. There is only one parameter that needs to be iteratively updated, which drives the solution of the optimization problem to the optimum values.

We demonstrate the different behaviors of the optimization approaches, dual-based [22], penalty-based [22], and dual-based [11], in terms of convergence speed. We consider a general wireless network to find the optimum session rates. Figure 3.1 shows the convergence of the three distributed methods using Matlab. The dual-based method of [11] is faster and nearly without perturbations, compared to the other two methods.

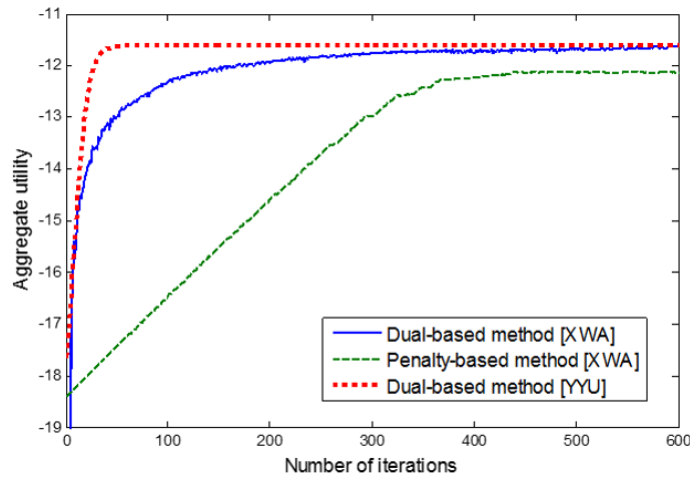


Figure 3.1: The Aggregate Session Rates vs. Number of Iterations.

3.2 IEEE 802.11 MAC Integration

IEEE 802.11 is a wireless network standard protocol [47] at the MAC layer with physical specifications at the physical layer. IEEE 802.11 uses a Carrier Sense Medium Access /Collision Avoidance technique (CSMA/CA) as its basic Distributed Coordination Function (DCF). Using DCF, a station waits until the media is clear and then transmits data. If a collision happens, the stations involved in the collision back off for a certain period of time, otherwise transmission starts. After completing the transmission, the station waits for an acknowledgment message from the recipient to indicate the data was received successfully. If the acknowledgment message is not received, the data is retransmitted and marked to let the recipient know the station is sending the data again.

In order to generalize the algorithm to be applied to more modern networks with CSMA-CA MAC protocols, we looked at the attempt probabilities from a different perspective: the time or the opportunity to transmit. In the ALOHA-based network model, we start with dividing the transmission attempt probability over the number of nodes in the same collision region according to a specific fairness scheme, these attempt probabilities can be interpreted as the time given to each node to transmit. Assuming the maximum physical capacity of the wireless links is 1, we calculate the transmission rates of each link as follows: the link transmission rate equals the attempt probability of the link times the probabilities of no transmission from the receiver and its 1st hop neighbours. This can be formulated similar to Equation (2.4) as follows,

$$r_\ell = C_\ell p_\ell \prod_{n \in \mathcal{N}_I(\ell)} (1 - q_n)$$

This calculation removes the wasted transmissions lost in collisions; so, the resultant rates are only the successful transmission rates. If the receiver neither sends any traffic nor has any other neighbours except the sender, then the transmission rate will be equal to the attempt probability of that link, because there is no chance of any collisions to occur.

In IEEE 802.11 MAC, the CSMA-CA based DCF protocol differs from the slotted ALOHA protocol in the calculation of rates. For simplicity, let us assume that the collision avoidance scheme works perfectly and there are no collisions. In order to achieve that, there will be a contention window that is large enough to avoid collisions. In this case there will be a percentage of time spent in contention and the exchange

of control signals; the rest of the time will be divided over the nodes located in the same collision region according to a specific fairness scheme. We can see that the collisions (and therefore wasted transmissions) in the slotted ALOHA protocol are replaced here by the time spent in the contention stage, and the transmission rate of each node will be directly proportional to the time percentage given to each node.

With the same previous analogy, we estimate the total time that is not exploited in a successful transmission and assign the rest of the time to the nodes according to the required fairness criteria. In [48], the optimum contention window size that maximizes throughput depends on the number of contending nodes and is given by $CW_{opt} = N\sqrt{2T_c}$, where T_c is the time lost due to collisions and N is the number of active nodes. But there is also control signal time (i.e. DIFS, SIFS, RTS, CTS, and ACK) that we should consider. Therefore we employ Bianchi's model [12], with the RTS/CTS and the basic access methods, which provides an estimate of the throughput limit as a function of the number of nodes and the packet length.

The medium access attempt probability used in the slotted ALOHA protocol represents the transmission opportunity given to the node. This transmission opportunity is determined by the algorithm based on the need for each link to offer enough capacity to carry, at a minimum, all sessions passing through it. This transmission opportunity can be controlled in the IEEE 802.11 MAC protocol by tuning the contention window [49] to achieve node fairness, but in the optimization process it will be used to optimize end-to-end session rates subject to a specific fairness criteria.

Bianchi's model [12] provides us with the saturation throughput as a function of different parameters. The following formula provides the saturation throughput of transmitter n ,

$$B_n = \frac{P_s P_{tr} E(M)}{(1 - P_{tr})\sigma + P_s P_{tr} T_s + P_{tr} (1 - P_s) T_c} \quad (3.1)$$

$E(M) \equiv$ Average message payload.

$P_{tr} \equiv$ Probability that there is at least one transmission in the considered time slot.

$P_s \equiv$ Probability of successful transmission.

$T_s \equiv$ The average time the channel is sensed busy due to a successful transmission.

$T_c \equiv$ The average time the channel is sensed busy during a collision.

$\sigma \equiv$ Slot time size.

The main difference between the two methods, RTS/CTS and the basic access, is

the reduced channel average busy time during collisions T_c in the RTS/CTS method compared to the basic one. In case of the basic access method, the collision time includes the length of the packet payload involved in the collision, while in case of the RTS/CTS method, the collision time includes only the control signal RTS. The throughput gap of the two methods increases as the number of contending nodes increases. The saturation throughput as a function of the number of contending nodes represents the percentage of the useful transmission opportunities given to the nodes. Under saturation conditions, each node always has a packet available for transmission, so changing the nodes transmission opportunity via contention window tuning [49] will not affect the resultant aggregate throughput.

3.2.1 Optimization Model and Decomposition

In our work, we started from the NUM optimization algorithm in [11], discussed in Section (2.1). The optimization problem is formulated considering a single channel wireless network, modelled as a directional graph $G = (\mathcal{N}, \mathcal{L})$ with $|\mathcal{N}|$ nodes and $|\mathcal{L}|$ logical links. Each link has a feasible physical capacity of C_ℓ bps, and \mathcal{S} sources transmit at a source rate of r_s bps. $\mathcal{L}_{out}(n)$ is the set of outgoing links from node $n \in \mathcal{N}$. Each source s emits one flow, using a fixed set $\mathcal{L}(s)$ of links on its path, and has a utility function $U_s(r_s)$. $U_s(r_s)$ is a concave function as discussed in Section (2.1) and a number of different fairness functions are possible (see Equation (2.2)). Here we define it as $\log(r_s)$, which aims for proportional fairness among the sessions. Each link $\ell \in \mathcal{L}$ can be shared by a set $\mathcal{S}(\ell)$ of sources. We assume that each node can receive from at most one adjacent node at a time, it cannot receive and transmit simultaneously. Also for a successful transmission, all the nodes in the same contention area should be silent except the transmitter. The nodes in the same contention area are the transmitter, receiver, 1st hop neighbours of the transmitter, and 1st hop neighbours of the receiver. We define the set $\mathcal{N}_{CA}(n)$ as the set of nodes in the same contention area of node n , including node n . The optimization problem is,

$$\begin{aligned} \max \quad & \sum_s U_s(r_s) \\ \text{s.t.} \quad & \sum_{s \in \mathcal{S}(\ell)} r_s \leq c_\ell \end{aligned} \tag{3.2}$$

where c_ℓ is the link capacity, which is an estimate calculated based on Bianchi's model [12] for IEEE 802.11 according to the following formula,

$$c_\ell = C_\ell p_\ell B_n \quad (3.3)$$

where p_ℓ is the link transmission opportunity. B_n is the percentage of useful transmission opportunity given to the transmitter n .

The utility maximization problem is solved using the Primal-dual approach and decomposed into three sub-problems connected by Lagrange parameters (the prices). Two subproblems, addressing the transport layer to optimize the total session rates and the link utility fraction of each session, respectively, follow the same steps as in [11],

$$\max_{0 \leq r_s \leq 1} (U_s(r_s) - r_s \lambda_s) \quad (3.4)$$

$$\max_{\alpha_s^\ell \in \Lambda} \left(\sum_{s \in \mathcal{S}(\ell)} \lambda_s^\ell \log(\alpha_s^\ell) \right) \quad (3.5)$$

where α_s^ℓ is the fraction of the rate on link ℓ that is contributed by source session s , λ_s^ℓ is the session-link (hop) price that is the Lagrange parameter for session s on link ℓ , and $\lambda_s = \sum_{\ell \in \mathcal{L}(s)} \lambda_s^\ell$ is the session price.

The third maximization sub-problem addresses the MAC layer. It is a generalization for weighted medium access control. The goal is to optimize the transmission opportunity given to each link in the same contention area so that the total network reward, user utilities, is maximized as follow,

$$\max_{p_\ell \in \Pi} \left(\sum_{k \in \mathcal{N}_{CA}(n)} \sum_{\ell \in \mathcal{L}_{out}(k)} \lambda^\ell \log(p_\ell) \right) \quad (3.6)$$

where $\lambda^\ell = \sum_{s \in \mathcal{S}(\ell)} \lambda_s^\ell$ is the link price, p_ℓ is the transmission opportunity of link ℓ outgoing from node k that is in the contention area of node n , and $\Pi = \{\sum_{k \in \mathcal{N}_{CA}(n)} \sum_{\ell \in \mathcal{L}_{out}(k)} p_\ell = 1, 0 \leq p_\ell \leq 1\}$ is the projection operator for p_ℓ .

3.2.2 Distributed Solution

The first two sub-problems (3.4) and (3.5) are convex, their closed form solution proofs are similar to the ones found in Appendix A. Given the session-link prices λ_s^ℓ of all sessions passing through link ℓ , the optimum link capacity fraction that should

be assigned to each session is given by,

$$\alpha_s^\ell(\boldsymbol{\lambda}) = \begin{cases} \frac{\lambda_s^\ell}{\sum_{\dot{s} \in \mathcal{S}(\ell)} \lambda_{\dot{s}}^\ell} & \text{if } \sum_{\dot{s} \in \mathcal{S}(\ell)} \lambda_{\dot{s}}^\ell \neq 0, \\ \frac{1}{|\mathcal{S}(\ell)|} & \text{otherwise,} \end{cases} \quad (3.7)$$

where $|\mathcal{S}(\ell)|$ is the number of elements in $\mathcal{S}(\ell)$.

The third sub-problem (3.6) is similar to (3.5) for which we can use a similar solution. Given the two hop neighbour's prices, i.e. prices belonging to the nodes in the same contention area with node n , we can derive p_ℓ as follows,

$$p_\ell(\boldsymbol{\lambda}) = \begin{cases} \frac{\lambda^\ell}{\sum_{k \in \mathcal{N}_{CA}(n)} \sum_{j \in \mathcal{L}_{out}(k)} \lambda_j}, & \text{if } \sum_{k \in \mathcal{N}_{CA}(n)} \sum_{j \in \mathcal{L}_{out}(k)} \lambda_j \neq 0, \\ \frac{1}{\sum_{k \in \mathcal{N}_{CA}(n)} |\mathcal{L}_{out}(k)|}, & \text{otherwise,} \end{cases} \quad (3.8)$$

The node transmission opportunity will be given by,

$$q_n(\boldsymbol{\lambda}) = \sum_{\ell \in \mathcal{L}_{out}(n)} p_\ell(\boldsymbol{\lambda}) = \begin{cases} \frac{\lambda_n}{\sum_{k \in \mathcal{N}_{CA}(n)} \lambda_k}, & \text{if } \sum_{k \in \mathcal{N}_{CA}(n)} \lambda_k \neq 0, \\ \frac{|\mathcal{L}_{out}(n)|}{\sum_{k \in \mathcal{N}_{CA}(n)} |\mathcal{L}_{out}(k)|}, & \text{otherwise,} \end{cases} \quad (3.9)$$

where $\lambda_n = \sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell$ is the node price.

Equation (3.9) has an economic interpretation. q_n represents the transmission opportunity given to the node n , this opportunity should be increased or decreased based on the network rewards from the transmission of the nodes. The node price λ_n is the sum of the outgoing link prices λ^ℓ . If a packet is transmitted successfully through link ℓ , this link contributes a reward λ^ℓ to the network. So q_n is the transmission reward ratio between the node n and the all the nodes in the same contention area, this ratio represent the optimum transmission opportunity for each node. This interpretation applies to Equations (3.7) and (3.8) as well.

Using the projected subgradient method [11, 50], and the solution of the three sub-problems, the dual problem is solved at each node n for each link $\ell \in \mathcal{L}_{out}(n)$, and for all $s \in \mathcal{S}(\ell)$. There is only one price that needs to be updated, the session-link price. They are adjusted at each iteration t according to the following formula,

$$\lambda_s^\ell(t+1) = [\lambda_s^\ell(t) + \beta (\log(r_s(t)) - \log(\alpha_s^\ell(t) c_\ell(t)))]^+ \quad (3.10)$$

where β is the step size, and $\{a\}^+ = \max(a, 0)$.

3.2.3 Contention Window Adjustment

SNUM assigns the available resources fairly among the sessions. The capacity is limited by the throughput that can be achieved by the MAC layer. SNUM assigns the resources by determining the optimum access opportunity that should be given to each node.

Bianchi's model [12] accurately estimates the throughput of IEEE 802.11 with basic and RTS/CTS access. The estimated value will be calculated based on the optimum saturation throughput achievable by the MAC layer. [51] and [43] optimized the IEEE 802.11 MAC throughput by tuning the contention window (CW) which affects the transmission probability for each node to achieve fairness with higher throughput among the contending nodes. But these values consider only fairness and throughput at the MAC layer, while the CLD optimizes for end-to-end session fairness and throughput.

In [12], the optimum throughput achievable is shown to be independent of the number of nodes contending for access to the medium. We used these optimum throughput values as an upper limit for the expected link capacities of the contending nodes. Then, using (3.3), the capacity will be estimated using SNUM's calculated transmission opportunity. The CW of each node will be tuned according to the following widely-used formula [12, 43, 51, 52],

$$CW_n = \frac{2}{q_n} - 1, \quad (3.11)$$

that leads to high transmission rate as we assume optimum throughput all time. This will keep the network saturated and maximizes the aggregate utilities. That leads to the question of what will happen if there are any queue buildups due to overestimated capacity? For that we introduce the active queue management mechanism.

3.3 Active Queue Management

Our queue management mechanism has the same fundamentals as the work of Qiu et al. [53] but with a more localized, simpler implementation, and less overhead. We start with the problem formulation in (3.2). After decomposition, the resulting sub-problems can be implemented in two different layers, the first one maximizes the network utility by maximizing the session rates with a fairness criteria at the transport

layer. The second one optimizes the sessions' share on each link, implemented at the link layer. The third sub-problem, which is concerned with the queue management, works on the contention between nodes to access the medium at the MAC layer. The goal of the second sub-problem is to find the optimum transmission opportunity p_ℓ for each outgoing link ℓ in each node n and set the suitable link capacity according to (3.3).

In order to incorporate the queue management in the optimization process, we look at the individual queue of each node in the session path and deal with each one locally. If the queue builds up in any of the nodes in the path of the multi-hop session, it means that the capacity estimates of these active outgoing links are not correct. That could happen due to various reasons such as unaccounted external traffic, interference, and/or noise. The external traffic that is not accounted for in the traffic calculations is considered, in this case, an interference and noise affecting the network capacity. For example, if parts of the network, parts of the session paths, are exposed to an unexpected high level of noise or interference, the received signal will have a low signal-to-noise-ratio and could be neglected by the receiver. In case of the IEEE 802.11 MAC protocol, the transmitted packet will be retransmitted several times until an acknowledgement is received or the packet dropped. That will lead to queue buildups in the noisy areas. We interpret that as a degradation of the link capacity. In our proposed solution, we adjust the SNUM-estimated link capacity by a Queue-length-based Factor (QF). The new adjusted capacity c_{ℓ_Q} is calculated as follows,

$$c_{\ell_Q} = C_\ell \times p_\ell \times B_n \times QF_n, \quad \forall \ell \in \mathcal{L}_{out}(n) \quad (3.12)$$

where

$$QF_n = e^{-\frac{1}{Z} \left(\frac{Q_n}{Q_{n_{max}}} \right)^2} \quad (3.13)$$

and Q_n is the queue length at node n , $Q_{n_{max}}$ is the maximum queue length at node n , and Z is the aggressiveness parameter.

The Queue Factor throttles the estimated capacity seen by SNUM. Reducing the link capacity will directly affect the link price of that specific link according to (3.8). The throttled capacity will increase the session-link price, hence it increases the link price, session prices, and node price. The session price's increase will lead to lowering the rates of all the sessions that are using this link, hence fairly sharing the reduction in utilities. Furthermore, the increase in the node price will lead, according to (3.8)

and (3.9), to redistributing the transmission opportunities among the nodes in the same contention area which, in turn, leads to providing more access opportunity to the node with longer queue length, also in a fair manner. The aggressiveness parameter Z is used to provide different capacity throttling aggressiveness for different delay requirements in the network. Figure 3.2 shows the different steepness of the QF with different Z values for a maximum queue length $Q_{n_{max}} = 400$ packets.

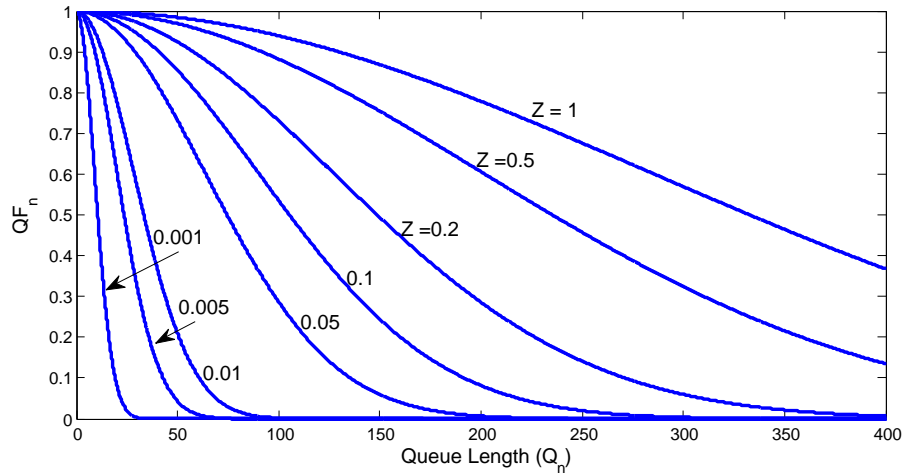


Figure 3.2: QF for different Z values and $Q_{n_{max}} = 400$.

This mechanism uses the prices that are already available in the CLD. Therefore, we avoid the need for a different type of prices or special control signalling. Also, in this CLD there is only one price that needs to be updated and communicated between layers and nodes. Our mechanism acts based on the local queue information first, then the prices propagate to the source nodes and the neighbours to re-assign utilities in a fair manner.

3.4 Algorithm Step-Size Adaptation

The algorithm presented in [11] requires setting the number of iterations required for convergence. For that, we can stop iterating if there is no change in the link prices greater than a certain tolerance for two or three consecutive iterations. Obviously though, each node will continue to optimize the link rates in case it receives new prices or in case of topology changes. This step-size should be suitable for the network, depending on the topology, network size, and number of links. If the step-size is

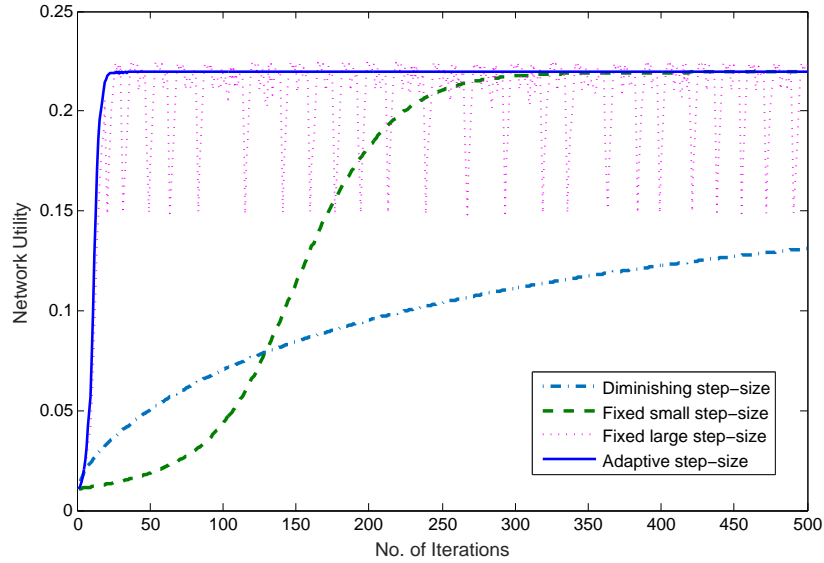


Figure 3.3: SNUM Behaviours with and without Adaptive Step-Size.

inappropriately large, it will cause the prices to bounce in each iteration between zero and a relatively high number, and the algorithm will enter an unstable state. A sudden change in the network topology may suddenly change the link prices and the node prices which also leads to an unstable state. There is a trivial solution that works with any network, that is, setting the step-size to a very small value, but that will increase the number of iterations required for convergence. For that reason, we adapt the update step-size parameter β in Equation (3.10).

The subgradient projection method used to solve the dual problem can work using a diminishing, fixed, or dynamically chosen update step-size based on the current convergence state ([50], Proposition 6.3.1). The diminishing step-size will drive the algorithm to converge with high accuracy but that will require a large number of iterations. A fixed step-size is used for faster convergence if a larger error margin is acceptable based on the application. We combine the advantage of both fixed and diminishing step-sizes by adapting the step-size based on the convergence state.

The initial step-size is chosen to be fixed and large enough to speed up convergence. We use a flag that indicates if the current step-size is still suitable for the next iteration or needs to be smaller to avoid instability. The nodes can indicate this instability by

continuously monitoring the following ratio,

$$F = \lambda_s^\ell / [\log(r_s) - \log(\alpha_s^\ell c_\ell)] \quad (3.14)$$

which is the ratio between the rate-capacity difference (link utilization) and the prices for each of the node's outgoing links. Whenever the ratio is greater than one, the transmitter node of the link raises an instability flag. Based on that flag, we set β to a smaller value only for the specific session on the indicated link where the flag was raised. The value of β is set to be half the ratio F as follows,

$$\beta = \frac{\lambda_s^\ell}{2 [\log(r_s) - \log(\alpha_s^\ell c_\ell)]} \quad (3.15)$$

β will be determined locally at each node, and it will be updated only if it is required. Figure 3.3 shows the performance of SNUM in case of a fixed topology network diminishing, fixed, and adaptive step-sizes.

3.5 The SNUM Algorithm

SNUM can be implemented while preserving the boundaries between network layers and with minimal effect on other network protocols. SNUM is divided into sub-modules that can be separately implemented in each layer. The modules communicate with each other to exchange prices. Each layer takes its own decisions based on the prices.

The distributed implementation of SNUM is given in Table 3.1 with a separate function for each layer. The algorithm convergence condition can be set based on the application, expected rate of topology change, and/or network size. In general, we consider that the algorithm converged if the difference between the prices generated after two consecutive iterations is less than a certain convergence error value.

Table 3.1: Distributed Implementation of SNUM

	MAC layer	Transport layer
0	<i>Initialize the algorithm, set the initial link-session prices = 10;</i>	
1	<i>Calculate the algorithm prices</i>	
	<i>- Calculate outgoing link prices, - Calculate session fractions (3.7), - Calculate node price</i>	<i>- Calculate session prices</i>
2	<i>Calculate the rates</i>	
	<i>- Calculate link rates</i>	<i>- Calculate and set session rates</i>
3	<i>Calculate the probabilities for links (3.8) and nodes (3.9)</i>	
	<i>- Tune CW (3.11)</i>	
4	<i>Estimate link capacities</i>	
	<i>- Calculate The QF (3.13),</i>	
	<i>- Calculate Bianchi's throughput,</i>	
	<i>- Estimate the overhead rate, - Calculate adjusted capacity (3.12)</i>	
5	<i>Update the link-session prices (3.10)</i>	
6	<i>If the algorithm converges,</i>	
	<i>- end,</i>	
	<i>else</i>	
	<i>- Communicate updated prices, - Go to 1</i>	

3.6 Summary

Among the studied optimization approaches, we found that the dual based approach presented in [11] is the one with the fastest convergence speed. It also has the potential to withstand realistic network environment conditions. The potential lies in that fact that the algorithm requires a minimum amount of information to be exchanged between the nodes with only one update function. The queue management has been considered in previous NUM optimization algorithms, however, that usually incurs extra overhead in the network.

We proposed our CLD algorithm, called SNUM, that jointly optimizes the congestion and the contention problem employing the CSMA-CA MAC protocol with the RTS/CTS and the basic access methods. The model employs Bianchi's IEEE 802.11 network formulations to estimate capacities. The CLD is equipped with an

active queue management mechanism for queue bounded networks. Finally, an adaptive step-size mechanism is proposed to speed up SNUM's convergence and to avoid instability.

Chapter 4

Initial Performance Studies

4.1 MAC Efficiency

The interference in wireless networks can severely degrade the network performance, especially in multihop transmissions. For successful transmission, the shared medium and omnidirectional propagation of the transmission signal dictates a mutually exclusive access to the medium between interfering links. For example, the throughput of a 2 hop transmission is halved compared to that of a single hop transmission because of the self-interference of the intermediate node. There have been several studies on the wireless multihop transmission trying to find the upper bound network throughput for a given number of nodes. A widely cited paper by Kamal Jain *et. al.* [54] provides a method to find the optimum throughput bounds in a wireless multihop network assuming omniscience and omnipotence. The conditions assumed are unrealistic but the results act as a reference to compare different MAC protocols and performances.

The choice of the MAC protocol is based on the application and the network capability. There exists a variety of mechanisms ranging from a very simple one that does not require any coordination such as ALOHA, semi-coordination as IEEE 802.11, and fully coordinated as scheduling mechanisms. In the case of wireless networks, especially in dynamic topologies, contention-based MAC protocols are preferred over contention-free (scheduling) solutions. The former does not require any sort of centralized controller and can cope with the dynamics of the network. But that comes at a price of wasting resources because of collisions and/or the overheads inherent in collision avoidance.

In order to study the MAC efficiency using SNUM, we compare the resultant

throughput of the distributed algorithm to the optimum throughput generated using the scheduling mechanism in [54], assuming omniscience and omnipotence. Which means that the packets at the individual nodes are being finely controlled and carefully scheduled by an omniscient and omnipotent central entity. For comparable results, we consider a MAC protocol based on CSMA-CA with perfect carrier sensing mechanism where nodes can know exactly when the medium is busy or free to transmit. In other words, there is no wasted time in backing-off or in gaps between transmissions.

Consider the network topology shown in Fig. 4.1, with 9 nodes placed in a 3x3 grid. We assume an IEEE 802.11 interference model which requires both the sender and the receiver to be free of interference for successful transmission. For example, for a successful transmission on the active link, in Fig. 4.1, from node A and D, all outgoing and incoming links of nodes D, E, G, B should be silent.

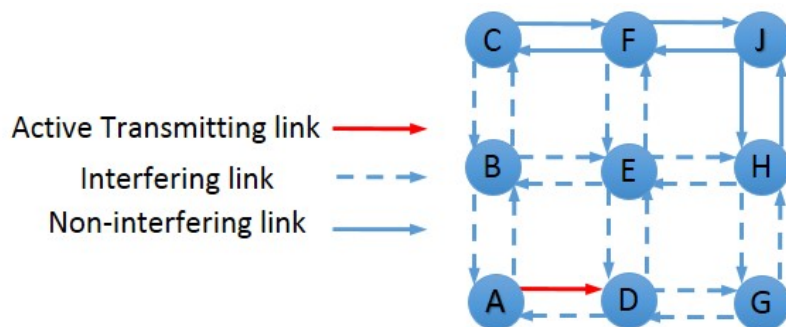


Figure 4.1: Topology of a 3x3 Grid Network.

We found that the distributed algorithm throughput is 0.4 and the optimally scheduled model throughput is 0.5. The reason for that is the absence of a centralized scheduler that can give mutually exclusive medium access to the non-interfering links simultaneously. The scheduling mechanism guarantees a time slot for each of the active links in each scheduling cycle and denies access to any other link in the same time slot. That is not the case in the random access mechanism where the transmission is based on probabilities each active link may/may not transmit at any time. In our model, the access opportunity of each link is divided over all the active links in the same interference area. As there is no mutually exclusive link access in random access mechanisms, each of the active links in the same interference area will account for the access probabilities of the rest of the links.

For a more in-depth explanation, consider a multihop session with multiple paths

from the source node A to the destination node J as shown in Fig. 4.2. The first path uses links 1 to 4, the second path uses links 5 to 8. The perfect scheduling, as discussed in [54], sets 4 time slots in a frame. The 4 time slots allow medium access to the active links in the following order, 1st (1 and 7), 2nd (2 and 8), 3rd (3 and 5), 4th (4 and 6). The frame is repeated as long as the session is active. That gives access opportunity of 0.25 for each of the active links, which means the first and second path throughputs are 0.25 each and the total throughput is 0.5.

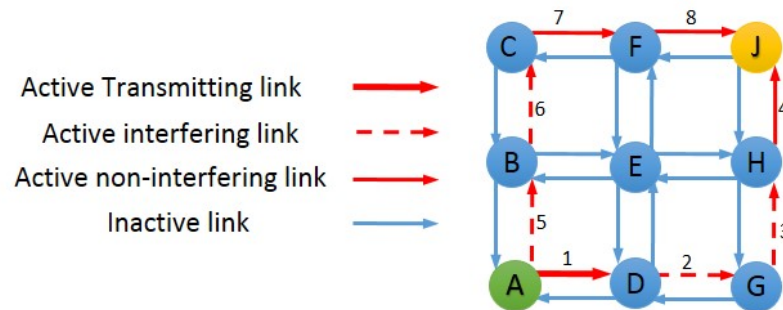


Figure 4.2: Active Links for a Multipath Session from Node A to J.

The random access mechanism assigns access opportunities for each link based on the number of contending links in the same interference area. As there is no guarantee for any of the links to be silent or active, the access opportunity is divided equally over all active links in the same contention area. For example, all five active links 1, 2, 3, 5 and 6 in the interference area with link 1 will have the same access opportunity of 0.2. As a result, each of the two paths has a throughput of 0.2 and the total session throughput is 0.4.

The 20% extra throughput provided by the scheduling mechanism is a result of global control over medium access. It can permit/deny medium access for two or more links far apart from each other simultaneously. In random access mechanisms, as we are giving individual nodes the ability to decide whether to transmit or not, any node can access the medium and establish a link at any time as long as the medium is free. Which means all contending links will have access opportunity, resulting in less access opportunity per link, and lower total throughput.

4.2 Multipath Effect

A well-connected wireless network can provide the capability for multipath routing to maximize the aggregate session throughput and maintain connectivity in multihop sessions. However, unlike wired networks, the wireless interference changes the channel capacity seen by each node, leading to different capacities for each link. Moreover, the interference adversely affects the total network capacity and throughput.

Our discussion is based on the calculation of the network throughput using the SNUM algorithm. For simplicity, we use an example of an isolated multihop session, without any other source of interference around it. That helps us to understand the effect of multipath routing on the individual sessions without the complications of multi-session interference and fairness. For multipath, we generate distinct flows from the same source to the same destination. The optimized network parameters are calculated by jointly solving the congestion and contention problems. We are using the prices reflected from the congestion and contention problems of the network to explain the results.

4.2.1 Number of Paths

In this section, we are discussing the question of how many paths are needed to maximize the multihop session throughput and the overall network utility considering the network congestion and contention. The discussion includes two popular contention-based MAC protocols: ALOHA and IEEE 802.11.

For a successful transmission using the ALOHA protocol, the receiver node and its first hop neighbors should be silent during transmission. As for the IEEE 802.11 protocol, the interference model requires that both the transmitter and the receiver should be free of interference [54], that is, all the incoming links to the transmitter, receiver, and their first hop neighbours should be inactive. Because of these differences in the interference models, the impact of multipath transmission over these MAC protocols is different.

Let us consider a number of mutually interference-free paths, the paths are similar (same number of hops). In this case, interference occurs only at the source and destination, which is the ideal case for multipath routing. As shown in Fig. 4.3, we have a session with n hops and m paths.

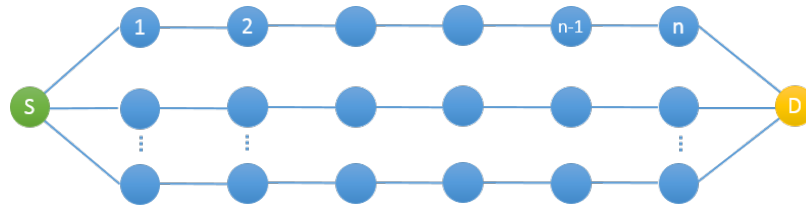


Figure 4.3: Session with n Hops and m Mutually Interference-Free Paths.

ALOHA Protocol

Using the SNUM algorithm, we calculated the optimum throughput of the multi-hop session for different numbers of paths. The results show that multipath routing, up to 3 paths, has a positive effect and increases the total session throughput. As the number of paths becomes 4 and up, the total throughput degrades, as shown in Fig. 4.4.

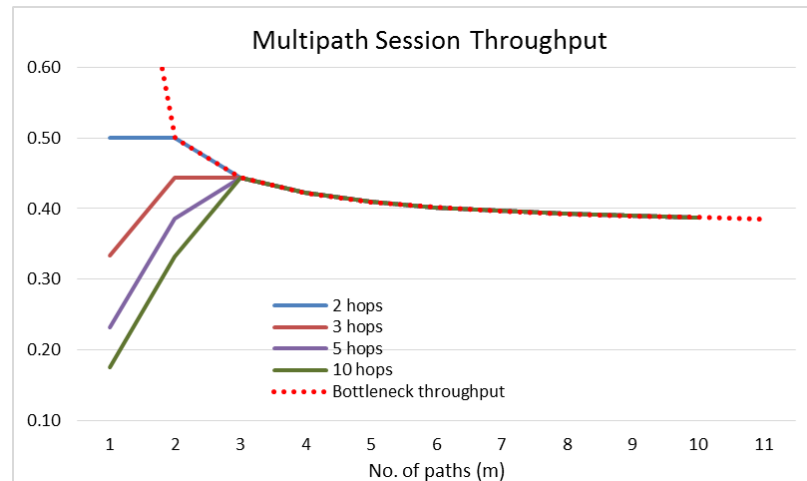


Figure 4.4: Multihop Session Throughput with Interference Free Paths.

To understand why the throughput starts to decay in the case of more than 3 paths, we need to take a closer look at the prices of each link used by the flows (hop prices) in each case. The prices are a reflection of the contention and congestion in the network, the ratio between prices in the same session is what determines where and how strong the bottleneck is. As the differences between the hops prices increase, the higher priced hops more and more determine the overall session throughput.

Figure 4.5 shows the normalized average hop prices in case of 1, 2, 3, and 4 paths over 10 hops. We can see that the single path session has a bottleneck in the middle of

the path, at hops 5 and 6. That is a result of the self-interference of the intermediate nodes, which leads to congestion and high prices. As the number of paths increases, we start to see that the bottleneck moves from the middle to the edges of the session. In the case of 3 paths, the bottleneck at the destination determines the throughput. The high difference between the hops prices indicates how strong the bottleneck is. The low hop prices in case of 3 and 4 paths are around 10^{-20} compared to the last hop price.

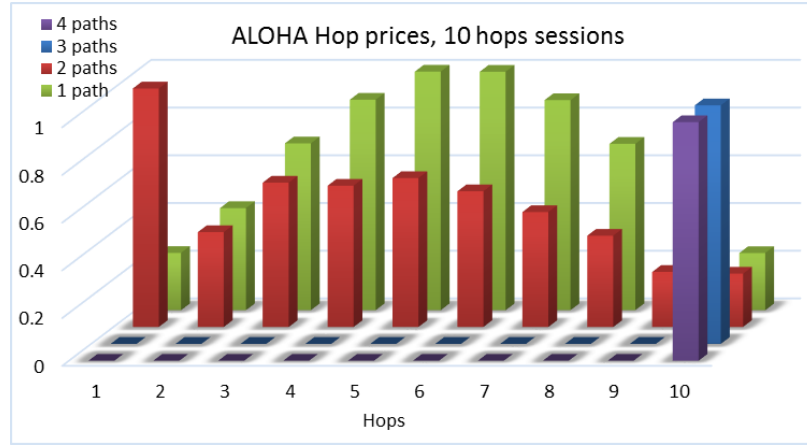


Figure 4.5: ALOHA Hops Prices for 10 Hops Session.

At this point, the session throughput is determined by the bottleneck at the last session hop where most of the collisions happen. The reason is that the last hop contains the highest number of nodes contending to access the medium without any coordination between them. This tells also that the degradation starts with more than 3 paths because we chose mutually interference free paths. If there are additional interfering hops in the session, they would be the one controlling the session throughput. The formula to calculate the throughput at the bottleneck based on the last hop nodes is as follows:

Let p_i be the transmission probability of the last hop node of path i , and $i = 1, 2, \dots, M$. The total number of paths used is M (which is also the number of last hop nodes contending to transmit). For a successful transmission, all the other nodes should be silent. The successful transmission probability of node i is,

$$P_{success} = p_i \prod_{j=1}^M (1 - p_j)$$

By giving equal opportunity to each of the identical paths $p_i = 1/M$, we get,

$$P_{success} = \frac{1}{M} \left(1 - \frac{1}{M}\right)^{(1-M)}$$

The total session throughput for physical capacity of 1 will be,

$$Bottleneck\ throughput = \left(1 - \frac{1}{M}\right)^{(1-M)} \quad (4.1)$$

Figure 4.4 shows the behaviour of the degraded throughput matches the bottleneck derived in (4.1) which demonstrates that the last hop bottleneck is the one in control of the throughput. The example we discussed shows that even with the addition of mutually interference-free paths to the session, the throughput cannot go higher than a certain point. In this case, the peak rate is reached with 3 paths. It may require more than 3 paths to reach the peak rate for the session throughput if there is interference in the middle of the path. Figure 4.6 shows that there is a little enhancement by the fourth path in the case of mutual interference between paths at the middle. We can see that the throughput in this case also follows the boundary of the Formula (4.1). We can conclude that there is a limited number of paths required to maximize the multipath sessions. Increasing the used number of paths may increase the total session throughput but with diminishing returns.

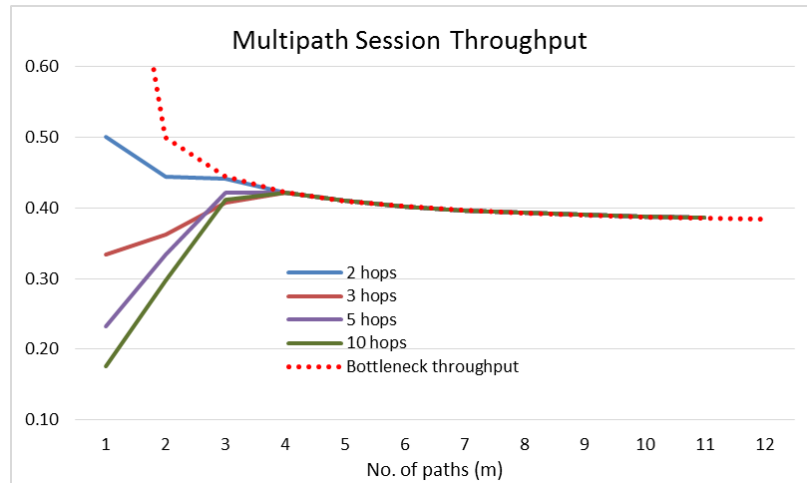


Figure 4.6: Multihop Session Throughput with Interfering Paths.

IEEE 802.11 Protocol

The different interference model and the semi-coordination of the IEEE 802.11 protocol leads to a different impact of multipath routing compared to the ALOHA protocol. Semi-coordination here means that the nodes can sense the medium and defer from transmission if it is busy.

The calculated optimum throughput of the multihop session with a different number of paths shows that multipath increases the total session throughput as shown in Fig. 4.7. As the number of paths increases, the returns become smaller and smaller as shown by the throughput/path values in Fig. 4.8.

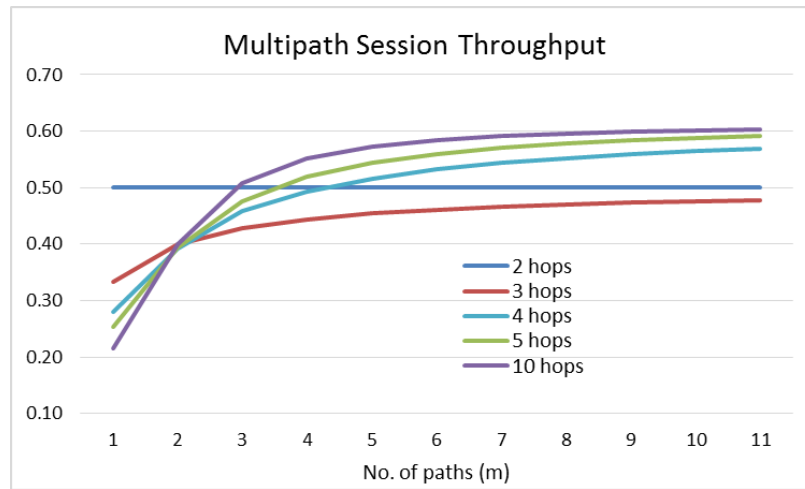


Figure 4.7: The Multihop Session Throughput with Interference-Free Paths.

The reason for the diminishing returns in case of the IEEE 802.11 protocol differs from the ALOHA case. Figure 4.9 shows the hop prices determined by the algorithm. The single path session bottleneck moves to the edges, both source and destination, as the number of paths increases. The bottleneck is much stronger at the source because the source medium access opportunity gets smaller as the number of paths increase. The source has to transmit and leave an opportunity to the first hop neighbors to access the medium as well. Bianchi's model [12] for IEEE 802.11 shows that, using the optimum access opportunity for each node, the optimum throughput achievable is independent of the number of nodes contending for medium access. So, increasing the number of contending nodes, using the semi-coordination and network utility maximization that assign the node optimum access opportunities, does not degrade

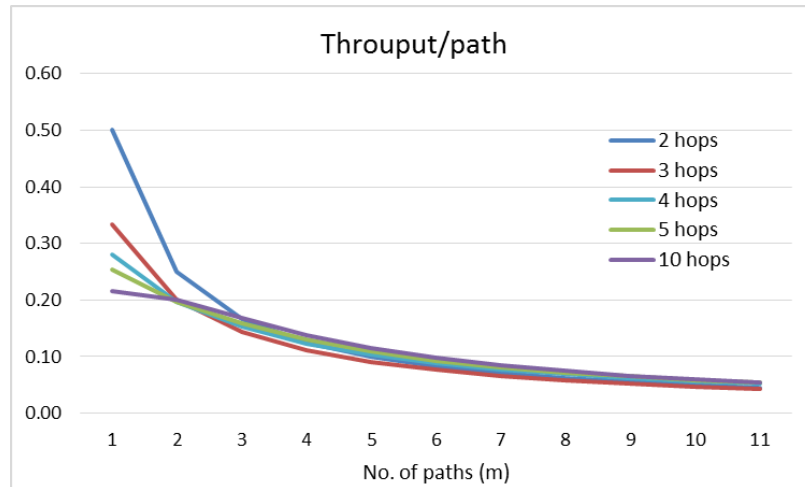


Figure 4.8: Session Throughput per Path.

the network performance. Unlike ALOHA, IEEE 802.11 can handle the last hop transmissions to the destination using the protocol semi-coordination to access the medium. That is why the throughput does not degrade with increasing the number of paths as was the case for ALOHA.

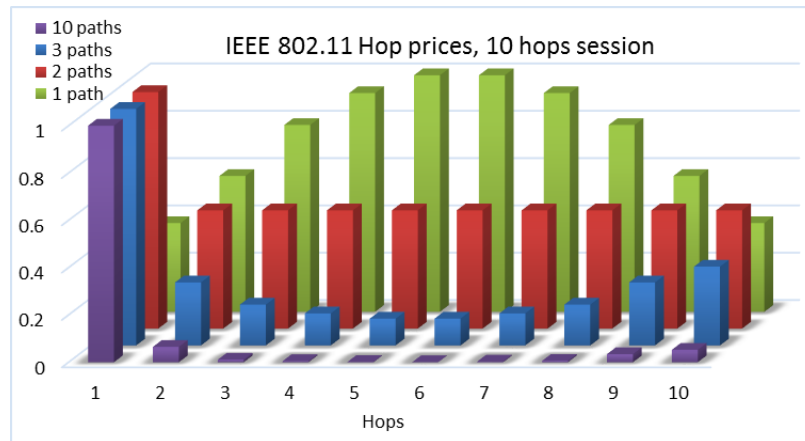


Figure 4.9: IEEE 802.11 Hops Prices for 10 Hops Session.

We can see that if the bottleneck moves from the middle of the session path to the edges, the effect of multipath becomes smaller. As a result, increasing the number of paths has a diminishing returns. Which means that the multipath utilization requires a smart metric that can select the right number of paths to use in order to provide a fair rate for each session.

4.2.2 Path Selection

We study the SNUM optimization results in case there are different paths from which we can chose. We need to know how to select the paths that achieve higher aggregate session throughput. What are the criteria to choose the paths? And is it beneficial for the network to have multipath with joint nodes or links? The test is done using the IEEE 802.11 MAC protocol.

Shortest Path vs. Interference Avoidance

In order to find the selection priorities to maximize session throughput, we test a network session with paths that differ in length and the amount of mutually interference. Figure 4.10 shows a session with 4 paths to connect the source to the destination. The paths A, B, C, and D are of lengths 5, 7, 8, and 12 hops respectively. Paths A and B have strong mutual interfere, route C is partially interfering with A. Route D does not interfere with any other route.

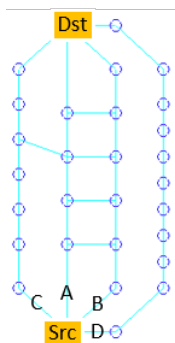


Figure 4.10: Multipath Session with Different Path Lengths and Mutual Interference.

The session is tested with 2 active paths. There are 10 route combinations, their throughputs are ordered in descending order in Table 4.1 according to the resultant throughput.

We can see that the interference free paths achieve the highest throughput, the number of hops matters less as long as two paths are used. Comparing scenario 2 to scenario 3 in Table 4.1, the path with more hops and less interference (route C) is better than the shorter route with high interference (route B). This seems to indicate that longer disjoint interference-free paths are better than short paths for throughput

maximization. As a result, the routing metric used to find the multipath route need to be based on the congestion and interference in the path instead of the number of hops.

Relating to our earlier discussion, the reason for this behaviour is the bottleneck created in the middle of the single path multihop session. As long as the source and destination can handle more traffic, the middle bottleneck stays in control of the session throughput. As a result, additional interference between the middle nodes can severely degrade the total session throughput compared to the number of hops in the path.

Multipath with Joint Node and Links

The previous discussions were based on the best path cases in terms of reliability, they are independent and do not share links or nodes. A communication failure on one path has no influence on the other. In this section, we are testing paths with joint nodes and links, and how the joint resources can affect the total throughput of the session.

We use an 8 hops multipath session with different joint links and nodes as shown in Fig. 4.11. Consistent with the interference results, joining even a single node (b), the path will degrade the session throughput more than 30% compared to the disjoint paths. The same reason explaining the degradation of the throughput due to the middle node bottleneck can be used to explain the degradation due to the joint middle node. As for the joint links paths (c), the throughput value is near to the single path throughput. The benefits of such paths may be more obvious in the presence of other interfering sessions affecting one side of the split portion.

We conclude that multipath with joint portions of the paths may not bring an effective throughput enhancement over the single path. The disjoint paths are preferred over the short paths with joint nodes and/or links.

4.3 Summary

The MAC layer throughput depends on the medium access mechanism used, the random access mechanism can support a fully distributed communication system but at a price of lower throughput compared to scheduling mechanisms.

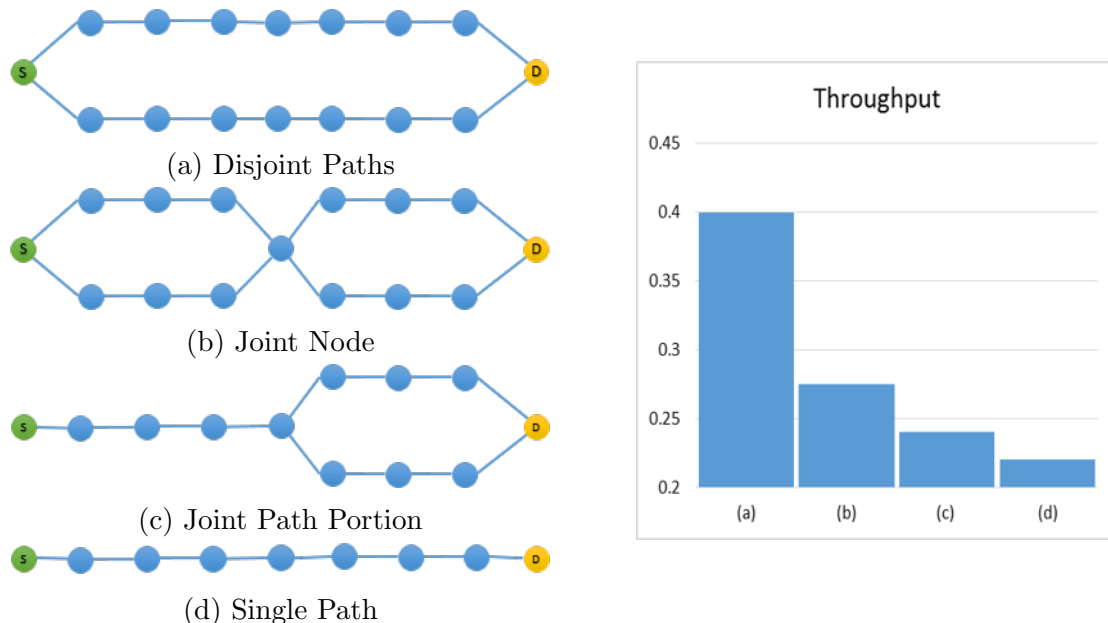
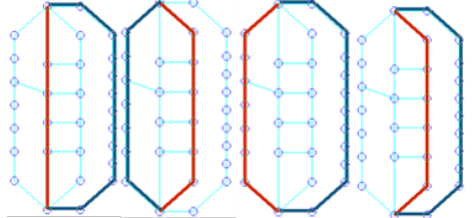
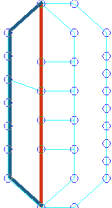
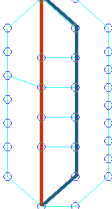
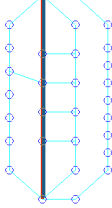
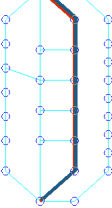
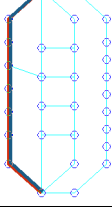
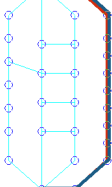


Figure 4.11: Different Multipath Types.

We can conclude that multipath routing is not always beneficial for the throughput unless the paths are carefully chosen. We can see that multipath routing enhances the throughput if the bottleneck of the session is at the relaying nodes of the paths, other than that, multipath is not useful. In the case of a multi-session network, where sessions may affect each other's throughput, multipath routing may not always be the solution. If such external (to the session) interference affects the relaying nodes, multipath routing could alleviate the bottleneck and enhance the throughput. Otherwise, if the external interference is at the source and/or the destination, multipath routing will not be able to enhance the session throughput. We can see here the advantage of having a network utility maximization algorithm that can assign throughputs according to the benefit and fairness of all the other sessions in the network. We need an algorithm that is able to choose the multipath routes considering the interference and the congestion in the entire network.

Table 4.1: Path Selection Ordered Descendingly Based on the Throughput.

1	No interference (optimum)		Throughput = 0.4
2	Long path & Low interference		0.3880
3	Short path and high interference		0.3208
4	(5 hops) same path		0.2541
5	(7 hops) same path		0.2297
6	(8 hops) same path		0.2234
7	(12 hops) same path		0.2104

Chapter 5

Multipath SNUM

In this chapter, we propose a novel multipath Network Utility Maximization algorithm (mNUM) that considers the end-to-end flow congestion, contention between flows, multipath sub-flows, and multipath routing problems in a distributed fashion. We show how to realize mNUM by using the 802.11 MAC protocol, adding the queue management and the speed up mechanisms discussed in Chapter 3 to convert mNUM to multipath SNUM (mSNUM). Then we provide a multipath searching mechanism to be used with mSNUM.

5.1 The mNUM Algorithm

In this section, we convexify and prove the mNUM optimality by satisfying the KKT conditions. The primal problem formulation is a non-convex, non-separable NUM optimization. By introducing new variables, applying certain transformations, and using an analogy based on Ohm's law, we develop a distributed algorithm that can find the optimal solution for general concave utility functions.

5.1.1 Problem Formulation and Notation

The optimization problem is formulated considering a single channel multipath wireless network, modelled as a directional graph $G(\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of nodes and \mathcal{L} is the set of logical links. Each link $\ell \in \mathcal{L}$ has a feasible physical capacity of C_ℓ bit-per-second (bps). $\mathcal{L}_{out}(n)$ and $\mathcal{L}_{in}(n)$ represent the set of outgoing and incoming links of node $n \in \mathcal{N}$ respectively. The network is utilized by the set \mathcal{S} of sessions, each session $s \in \mathcal{S}$ identifies a source-destination pair.

Similar to [11, 55], we assume that each node can receive from at most one adjacent node at a time, it cannot receive and transmit simultaneously. Furthermore, for a successful transmission, all the nodes in the same contention area should be silent except for the transmitter. We consider random access based MAC protocols where each node can access the medium with persistence probability q_n . Each link $\ell \in \mathcal{L}_{out}(n)$ can gain access to the medium with persistence probability p_ℓ so that $\sum_{\ell \in \mathcal{L}_{out}(n)} p_\ell = q_n, \forall n$. We define node and link sets to show the interference relations between nodes and links in the contention area. $\mathcal{L}_I(n)$ is the set of links whose reception are adversely affected by the transmission of node $n \in \mathcal{N}$, excluding $\ell \in \mathcal{L}_{out}(n)$. $\mathcal{N}_I(\ell)$ is the set of nodes whose transmission adversely affect the reception of link ℓ , excluding the transmitter of link ℓ .

Each session source can send packets to its destination over a set \mathcal{M}_s of candidate paths, each path $m \in \mathcal{M}_s$ uses a set $\mathcal{L}_s(m)$ of sequential links. The traffic of session s can be split over two or more paths at the source. The sub-flows can share the same link as individual sub-flows. Each link $\ell \in \mathcal{L}$ can be shared by different paths of different session-paths (s, m) among the set \mathcal{M}_ℓ . For each session s , assume the flow rate on path $m \in \mathcal{M}_s$ (sub-flow) is $r_{s,m}$, then the total flow rate of the session s is $\sum_{m \in \mathcal{M}_s} r_{s,m} = r_s$ bps and the utility attained by session s is given by $U_s(r_s)$. The utility function is a general concave function that can be selected based on the required fairness as in (2.2).

Our objective is to optimize the aggregate utility of all sessions in the network by allocating the sub-flow rates $\mathbf{r} := \{r_{s,m} | s \in \mathcal{S}, m \in \mathcal{M}_s\}$ and choose the node and link persistence probabilities $(\mathbf{q}, \mathbf{p}) := \{(q_n, p_\ell) | n \in \mathcal{N}, \ell \in \mathcal{L}_{out}(n)\}$. The utility maximization problem can be formulated subject to constraints at the Transport, Network, and MAC layers. We optimize the end-to-end session rate variable r_s at the transport layer, sub-flow rate allocation for all paths of a session at the Network layer, and persistence probabilities at the MAC layer. The formulation is as follows,

$$\begin{aligned}
\mathbf{P1} \quad & \max_{r_s} && \sum_{s \in \mathcal{S}} U_s(r_s) \\
& s.t. && \sum_{m \in \mathcal{M}_s} r_{s,m} = r_s, \forall s, && (5.1) \\
& && \sum_{s,m \in \mathcal{M}_\ell} r_{s,m} \leq C_\ell p_\ell \prod_{k \in \mathcal{N}_I(\ell)} (1 - q_k), \forall \ell, && (5.2) \\
& && \sum_{\ell \in \mathcal{L}_{out}(n)} p_\ell = q_n \leq 1, \forall n, && (5.3) \\
& && 0 \leq p_\ell \leq 1, \forall \ell, \\
& && r_s \geq 0, \forall s,
\end{aligned}$$

The equality Constraint (5.1) gives the session multiple alternative paths between the source and destination. The linear relation between the session total flow rate and its sub-flows makes the utility function not strictly concave, even if the utility function $U(\cdot)$ is strictly concave, introducing instability in the NUM algorithms [38, 56]. Constraint (5.2) states that the aggregate sub-flows allocated to the link ℓ must be less than or equal to the link throughput. Due to Constraint (5.2), the formulation becomes in general non-convex and non-separable. Constraints (5.1) and (5.2) make the problem difficult to be solved in a distributed fashion and converge toward the optimum. However, with the proper transformation under certain conditions, introducing new variables, and Ohm's law analogy, the problem can become convex, be solved in a distributed fashion, and converge toward the global optimum solution.

5.1.2 Convex Formulation

We formulate the problem **P1** into a convex separable optimization problem by merging the two Constraints (5.1) and (5.2) into a single inequality. Then, we transform it to a convex one and we develop a dual-based distributed algorithm to find the optimum solution.

First we introduce the splitting variable set $\mathbf{v} := \{v_{s,m} | 1 \leq v_{s,m}, \sum_{m \in \mathcal{M}_s} 1/v_{s,m} = 1, \forall s, m \in \mathcal{M}_s\}$, where $1/v_{s,m}$ is the flow rate fraction of session s assigned to path m , which we call the splitting factor. The relation between the total flow rate and their sub-flows become as follows,

$$r_{s,m} = r_s / v_{s,m}, \forall m \in \mathcal{M}_s \quad (5.4)$$

$v_{s,m}$ will have a minimum value of 1 if s is a single-path session, i.e. $|\mathcal{M}_s| = 1$. Replacing the sub-flow rate $r_{s,m}$ of Constraint (5.2) with its equivalent value from (5.4) yields a single constraint as follows,

$$\sum_{s,m \in \mathcal{M}_\ell} r_{s,m}/v_{s,m} \leq C_\ell p_\ell \prod_{k \in \mathcal{N}_I(\ell)} (1 - q_k), \forall \ell \quad (5.5)$$

Merging the two Constraints (5.1) and (5.2) will alleviate the difficulties inherited from the multipath capability of the network, which is the linearity between paths that causes oscillation. The problem is still non-convex and non-separable due to the merged constraint (5.5). The problem can be transformed into a convex one by taking the logarithm on both sides of constraint (5.5) and replacing the rate variables with their logarithms [55], that is $z_{s,m} = \log(r_{s,m})$ and $z_s = \log(r_s)$. However, the summation on the left hand side of (5.5) will prevent us from achieving our goal.

The problem lies in the summation that aggregates all session-path rates sharing a link. Transforming the rates to its log values on both sides of Constraint (5.5) leads to the term $\log(\sum_{s,m \in \mathcal{M}_\ell} (z_s - \log v_{s,m}))$, which is not convex and not separable. For that, we need to separate the rates of each session on each link, i.e. dividing the available link capacity into sub-capacities for each session-path $s, m \in \mathcal{M}_\ell$. This can be achieved by introducing a new variable $\alpha := \{\alpha_{s,m}^\ell | 0 \leq \alpha_{s,m}^\ell \leq 1, \sum_{s,m \in \mathcal{M}_\ell} \alpha_{s,m}^\ell = 1, \ell \in \mathcal{L}, (s, m) \in \mathcal{M}_\ell\}$, where $\alpha_{s,m}^\ell$ can be interpreted as the fraction of the traffic on link ℓ contributed by the session-path $s, m \in \mathcal{M}_\ell$. As a result, each session-path rate is constraint by the link capacity fraction $\alpha_{s,m}^\ell C_\ell p_\ell \prod_{k \in \mathcal{N}_I(\ell)} (1 - q_k)$, and Constraint (5.5) becomes,

$$r_s \leq v_{s,m} \alpha_{s,m}^\ell C_\ell p_\ell \prod_{k \in \mathcal{N}_I(\ell)} (1 - q_k), \forall \ell, \forall s, m \in \mathcal{M}_\ell \quad (5.6)$$

Taking the logarithm for both sides of Constraint (5.6) leads to the following convex formulation,

$$\mathbf{P2} \quad \max_{z_s} \quad \sum_{s \in \mathcal{S}} U^z(z_s) \quad (5.7)$$

$$\begin{aligned} \text{s.t.} \quad & z_s - \log v_{s,m} - \log \alpha_{s,m}^\ell - \log C_\ell - \log p_\ell \\ & - \sum_{k \in \mathcal{N}_I(\ell)} \log(1 - q_k) \leq 0, \quad \forall \ell, \forall s, m \in \mathcal{M}_\ell, \quad (5.8) \\ & \sum_{m \in \mathcal{M}_s} 1/v_{s,m} = 1, \quad 1 \leq v_{s,m}, \quad \forall s, m \in \mathcal{M}_s, \\ & \sum_{s,m \in \mathcal{M}_\ell} \alpha_{s,m}^\ell = 1, \quad 0 \leq \alpha_{s,m}^\ell \leq 1, \quad \ell \in \mathcal{L}, (s, m) \in \mathcal{M}_\ell, \\ & \sum_{\ell \in \mathcal{L}_{out}(n)} p_\ell = q_n \leq 1, \quad \forall n, \\ & 0 \leq p_\ell \leq 1, \quad \forall \ell, \end{aligned}$$

where $U^z(z_s) := U(e^{z_s})$ is the transformed utilities convenient for the logarithm of the rates. Even though the utility function $U(r_s)$ is strictly concave, the log rate transformed utility function $U^z(z_s)$ may not be strictly concave. The following Lemma [55] provides a sufficient condition to maintain the strict concavity of the transformed utility function by choosing an appropriate fairness parameter.

Considering that,

$$g(r_s) = \frac{d^2 U(r_s)}{dr_s^2} r_s + \frac{dU(r_s)}{dr_s} \quad (5.9)$$

we can follow [55] to prove the following Lemma.

Lemma 5.1.1. *If $g(r_s) < 0$, then $U^z(z_s)$ is strictly concave in z_s .*

The condition to guarantee strict concavity of the transformed utility function dictates that the parameter θ in (2.2) must be $\theta > 1$ according to [55]. In order to achieve the proportional fairness using the transformed utility function we need to set $\theta = 2$ and harmonic mean fairness using $\theta = 3$. Throughout the document, we assume that the condition in Lemma 5.1.1 is satisfied.

Given that the condition in Lemma 5.1.1 is met and that the summation of the concave functions, $-\log(\cdot)$ in Constraint (5.8), is concave [57], then Problem **P2** is a convex and separable problem.

5.1.3 Dual Decomposition and the Distributed Algorithm

We solve Problem **P2** by using the dual decomposition approach as all the log rates are decoupled and the variables are separable. We start with the Lagrangian function, consider $\lambda_{s,m}^\ell$ is the Lagrange parameter, in other words, the price associated to the session-path s, m on link ℓ . The Lagrangian formulation is as follows,

$$\begin{aligned}
L(\boldsymbol{\lambda}, \mathbf{z}, \mathbf{p}, \mathbf{q}, \boldsymbol{\alpha}, \mathbf{v}) &= \sum_{s \in \mathcal{S}} U^z(z_s) \\
&+ \sum_{\ell \in \mathcal{L}} \sum_{s, m \in \mathcal{M}_\ell} \lambda_{s,m}^\ell \left(\log \left(\alpha_{s,m}^\ell v_{s,m} C_\ell p_\ell \prod_{k \in \mathcal{N}_I(\ell)} (1 - q_k) \right) - z_s \right) \\
&= \sum_{s \in \mathcal{S}} (U^z(z_s) - z_s \lambda_s) + \sum_{s \in \mathcal{S}} \sum_{m \in \mathcal{M}_s} \lambda_{s,m} \log v_{s,m} \\
&+ \sum_{\ell \in \mathcal{L}} \lambda^\ell \log C_\ell + \sum_n \sum_{\ell \in \mathcal{L}_{out}(n)} \sum_{s, m \in \mathcal{M}_\ell} \lambda_{s,m}^\ell \log \alpha_{s,m}^\ell \\
&+ \sum_n \left(\sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell \log p_\ell + \sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell \log(1 - q_n) \right)
\end{aligned} \tag{5.10}$$

where $\lambda^\ell = \sum_{s, m \in \mathcal{M}_\ell} \lambda_{s,m}^\ell$ is the link price, $\lambda_{s,m} = \sum_{\ell \in \mathcal{L}_s(m)} \lambda_{s,m}^\ell$ is the path price, and λ_s is the session price which is the equivalent price for all paths $m \in \mathcal{M}_s$ associated with the session s .

The prices $\lambda_{s,m}^\ell$ are a representation of the congestion facing each session-path in each link, in other words, the resistances of the flow rate in its path. We use the same analogy in dealing with parallel paths. Due to the parallelism in the sessions' paths, all connecting one source to one destination, the equivalent session price λ_s cannot be a simple summation over the path prices $\lambda_{s,m}$. The equivalent price should follow Ohm's law analogy in finding the parallel resistance equivalent as shown in Fig. 5.1.

The equivalent session prices will be calculated as follows,

$$\lambda_s = \frac{1}{\sum_{m \in \mathcal{M}_s} \frac{1}{\lambda_{s,m}}} \tag{5.11}$$

The price value resulting from (5.11) will be less than the minimum path price among all $m \in \mathcal{M}_s$, which requires adjustments as will be discussed later.

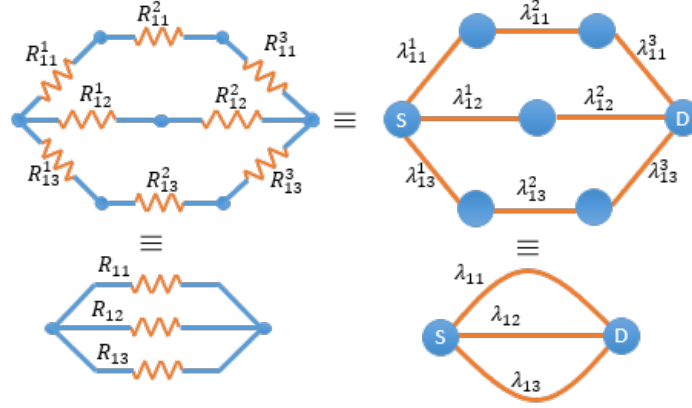


Figure 5.1: The Equivalent Session Prices.

The Lagrangian dual function is,

$$D(\boldsymbol{\lambda}) = \max_{\substack{\sum_{m \in \mathcal{M}_s} 1/v_{s,m} = 1, 1 \leq v_{s,m}, \forall s, \forall (s,m), \\ \sum_{s,m \in \mathcal{M}_\ell} \alpha_{s,m}^\ell = 1, 0 \leq \alpha_{s,m}^\ell \leq 1, \forall \ell, s, m \in \mathcal{M}_\ell, \\ \sum_{\ell \in \mathcal{L}_{out}(n)} p_\ell = q_n \leq 1, \forall n, \\ 0 \leq p_\ell \leq 1, \forall \ell}} L(\boldsymbol{\lambda}, \mathbf{z}, \mathbf{p}, \mathbf{q}, \boldsymbol{\alpha}, \mathbf{v}) \quad (5.12)$$

The dual problem, given $\boldsymbol{\lambda}$, then corresponds to minimizing D over the dual variable $\boldsymbol{\lambda}$

$$D : \min_{\boldsymbol{\lambda} \geq 0} D(\boldsymbol{\lambda}) \quad (5.13)$$

Using standard convex optimization tools [55,57] we can prove the following proposition.

Proposition 5.1.2. *The duality gap is zero, the maximum value of $\mathbf{P2}$ is equal to the minimum value of (5.13).*

First, we solve the maximization problem (5.12) over the network parameters $(\mathbf{z}, \mathbf{p}, \mathbf{q}, \boldsymbol{\alpha}, \mathbf{v})$ given $\boldsymbol{\lambda}$. The problem can be decomposed into 4 subproblems, two at each session source and the other two at each node. The session source subproblems are,

$$\max_{z_s} (U^z(z_s) - z_s \lambda_s), \quad (5.14)$$

and

$$\max_{\substack{\sum_{m \in \mathcal{M}_s} 1/v_{s,m} = 1, \\ 1 \leq v_{s,m}}} \sum_{m \in \mathcal{M}_s} \lambda_{s,m} \log v_{s,m} \quad (5.15)$$

The maximum of (5.14) can be found using $\frac{d}{dz_s} U^z(z_s) = \lambda_s$, since $U^z(z_s)$ is a

strictly concave function over z_s , then a unique maximizer exists. The other two subproblems are solved for every link $\ell \in \mathcal{L}_{out}(n)$, for each node n ,

$$\max_{\substack{\sum_{s,m \in \mathcal{M}_\ell} \alpha_{s,m}^\ell = 1, \\ 0 \leq \alpha_{s,m}^\ell \leq 1}} \sum_{s,m \in \mathcal{M}_\ell} \lambda_{s,m}^\ell \log \alpha_{s,m}^\ell, \quad (5.16)$$

and

$$\max_{\substack{\sum_{\ell \in \mathcal{L}_{out}(n)} p_\ell = q_n \leq 1, \\ 0 \leq p_\ell \leq 1}} \sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell \log p_\ell + \sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell \log(1 - q_n) \quad (5.17)$$

Subproblems (5.15)-(5.17) are convex and can be solved in a closed form by solving the KKT optimality conditions for a given λ .

Proposition 5.1.3. *Given the Lagrange parameters λ , the closed form solution for (5.15) for session s and all $m \in \mathcal{M}_s$ is,*

$$v_{s,m}(\lambda) = \begin{cases} \frac{\sum_{\hat{m} \in \mathcal{M}_s} 1/\lambda_{s,\hat{m}}}{1/\lambda_{s,m}} & \text{if } \lambda_{s,\hat{m}} > 0, \forall \hat{m} \in \mathcal{M}_s, \\ |\mathcal{M}_s| & \text{otherwise,} \end{cases} \quad (5.18)$$

the solutions for (5.16) for node n and link $\ell \in \mathcal{L}_{out}(n)$ is,

$$\alpha_{s,m}^\ell(\lambda) = \begin{cases} \frac{\lambda_{s,m}^\ell}{\sum_{\hat{m} \in \mathcal{M}_\ell} \lambda_{s,\hat{m}}^\ell} & \text{if } \sum_{\hat{m} \in \mathcal{M}_\ell} \lambda_{s,\hat{m}}^\ell \neq 0, \\ \frac{1}{|\mathcal{M}_\ell|} & \text{if } \sum_{\hat{m} \in \mathcal{M}_\ell} \lambda_{s,\hat{m}}^\ell = 0, \end{cases} \quad (5.19)$$

and $p(\lambda)$, $q(\lambda)$ solving (5.17) for node n and link $\ell \in \mathcal{L}_{out}(n)$ are,

$$p_\ell(\lambda) = \begin{cases} \frac{\lambda^\ell}{\sum_{\hat{\ell} \in \mathcal{L}_{out}(n)} \lambda^{\hat{\ell}} + \sum_{\hat{\ell} \in \mathcal{L}_I(n)} \lambda^{\hat{\ell}}} & \text{if } k_n \neq 0, \\ \frac{1}{|\mathcal{L}_{out}(n)| + |\mathcal{L}_I(n)|} & \text{if } k_n = 0, \end{cases} \quad (5.20)$$

$$q_n(\lambda) = \begin{cases} \frac{\sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell}{\sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell + \sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell} & \text{if } k_n \neq 0, \\ \frac{|\mathcal{L}_{out}(n)|}{|\mathcal{L}_{out}(n)| + |\mathcal{L}_I(n)|} & \text{if } k_n = 0, \end{cases} \quad (5.21)$$

where $k_n = \sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell + \sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell$.

Proof. The proof of Proposition 5.1.3 can be found in Appendix A. \square

The solutions of Proposition 5.1.3 are all depending on the prices, the price $\lambda_{s,m}^\ell$ that represent the congestion facing each session sub-flow in each link. For (5.18),

the splitting variable $v_{s,m}$ is proportional to the path price $\lambda_{s,m}$ and also inversely proportional to the flow rate $r_{s,m}$ assigned to the path according to (5.4). That leads to allocating a higher fraction of the session rate to the session-path that has a lower price, i.e. is less congested. As for (5.19)-(5.21), higher prices imply that more resources are required. Sessions and links that have high prices get more resources compared to other sessions and links. That is why the Solutions (5.19)-(5.21) are normalized by the prices of all competing sessions and links. There is an adjustment required for fair resources assignment. Recall that Equation (5.11) gives session prices that are less than individual session-path prices seen at each link, which results in an unfair resource assignment to the session-paths. In order to correct that, the session-path prices at each link $\lambda_{s,m}^\ell$ fed to the Solutions (5.19)-(5.21) will be replaced by the equivalent session prices at each link. That can be done using the path price and the session price based on the parameter θ in (2.2) as follows,

$$\lambda_{s,m_{fair}}^\ell = \frac{\lambda_{s,m}^\ell}{\lambda_{s,m}} \times \frac{\lambda_s^{\theta-1}}{\lambda_{s,m}}, \forall (s, m), \forall \ell \in \mathcal{L}_s(m). \quad (5.22)$$

The adjustment in (5.22) is optional in case of multipath routing to give the single path sessions a fair share of resources compared to multipath sessions. Neglecting this adjustment will result in individually optimizing each sub-flow for each session, which will assign resources proportional to the number of paths available for each session.

Now, after we solved the subproblems of (5.12), we can solve the dual problem (5.13) using a subgradient projection method [57]. For each node n and $\forall \ell \in \mathcal{L}_{out}(n)$ and $\forall s, m \in \mathcal{M}_\ell$, the session-path prices at each link are updated at iteration $t + 1$ as follows,

$$\lambda_{s,m}^\ell(t+1) = \left[\lambda_{s,m}^\ell(t) - \beta(t) \frac{\partial D(\boldsymbol{\lambda}(t))}{\partial \lambda_{s,m}^\ell} \right]^+, \quad (5.23)$$

where $\beta(t) > 0$ is a step size, and,

$$\begin{aligned} \frac{\partial D(\boldsymbol{\lambda}(t))}{\partial \lambda_{s,m}^\ell} &= \log v_{s,m}(\boldsymbol{\lambda}(t)) + \log \alpha_{s,m}^\ell(\boldsymbol{\lambda}(t)) + \log C_\ell + \log p_\ell(\boldsymbol{\lambda}(t)) \\ &\quad + \sum_{k \in \mathcal{N}_I(\ell)} \log(1 - q_k(\boldsymbol{\lambda}(t))) - z_s(\boldsymbol{\lambda}(t)) \end{aligned} \quad (5.24)$$

The convergence of the algorithm using (5.23) is asserted as follows [10, 11],

Proposition 5.1.4. *Denote $\boldsymbol{\lambda}^*$ the minimizer of the dual problem (5.13), there exists*

a sequence of step sizes $\{\beta(t)\}_{(t=1)}^{\infty}$ (i.e. $\beta(t) = 1/t$) to guarantee $\lim_{t \rightarrow \infty} \boldsymbol{\lambda}(t) = \boldsymbol{\lambda}^*$, if the following conditions hold,

$$C1) \lambda_{s,m}^* \neq 0, m \in \mathcal{M}_s, |\mathcal{M}_s| \geq 1, \forall s$$

$$C2) k_n = \sum_{\ell \in \mathcal{L}_{out}(n)} \sum_{s,m \in \mathcal{M}_\ell} \lambda_{s,m}^{\ell*} + \sum_{\ell \in \mathcal{L}_I(n)} \sum_{s,m \in \mathcal{M}_\ell} \lambda_{s,m}^{\ell*} \neq 0, \quad \forall n \in \mathcal{N}$$

and solutions to (5.14)-(5.17) are the optimal solution for optimization problem **P2**.

Proof. The optimization Problem **P2** is convex and Conditions C1 and C2 ensure that Problem (5.10) has a unique solution at $\boldsymbol{\lambda}^*$. Thus, the subgradient method with appropriate sequence of step size converges toward the optimum $\boldsymbol{\lambda}^*$ according to Theorem 2.2 of [58]. \square

Note that this proposition is true whether we use normal prices or the adjusted ones based on (5.22). Table 5.1 shows the overall optimization algorithm that obtains the KKT point of the optimization Problem **P2**.

Table 5.1: mNUM Algorithm.

<i>MAC layer</i>	<i>Network layer</i>	<i>Transport, layer</i>
Initialize: Set the initial link-session-path prices $\lambda_{s,m}^\ell = 10^{(\theta-1)}$;		
Repeat: - Each source receives the path prices $\lambda_{s,m}$ from session paths; - Each node receives session prices λ_s from sources and link prices λ^ℓ from neighbours in $\mathcal{N}_I(\ell)$;		
Each node calculates: - Fair prices (5.22); - Link rate fractions (5.19); - Links pers. prob. p_ℓ , (5.20); - Node pers. prob. q_n , (5.21);	Each source calculates: - Splitting factors $v_{s,m}$ (5.18); - Session-path rates;	Each source calculates: - Session price (5.11); - Session rates (5.14);
- Update prices $\lambda_{s,m}^\ell$ for each link (5.23), accumulate $\lambda_{s,m}$ and λ^ℓ ; - Each source communicates session prices λ_s ; - Each node communicates path prices $\lambda_{s,m}$ to sources and link prices λ^ℓ to neighbours in $\mathcal{N}_I(\ell)$;		

We can see that the algorithm needs to communicate only price accumulations along the session-paths and to neighbours in the interference area. That can be achieved by one of two methods. The first one is to use dedicated update messages to

deliver the prices information. The second one is by piggybacking the information on any control broadcast messages and/or the data packets to deliver prices along its session-path. Both methods will be discussed in details in Chapter 6.

5.1.4 Numerical Example

This example is based on the network topology shown in Fig. 5.2. We have three sessions, two of which have two candidate paths. We used two strictly concave transformed utilities based on $U^z(z_s) = e^{z_s(1-\theta)}/(1-\theta)$, with fairness parameters $\theta = 2$ and 10. We set the physical capacity C_ℓ to 1 Mbps. mNUM has been tested with random initial prices and converged toward the optimum in all cases. We used a fixed step size $\beta = 0.05$. For the purpose of simulation, the convergence condition is considered met and the program terminates if the difference between the aggregate network rates in two successive iterations is less than 10^{-6} . The global optimum session rates are found with marginal error, without using mNUM, by searching among all possible weights for the path rates with diminishing step size as we approach the optimum. Near the global optimum, the step size reached 10^{-6} .

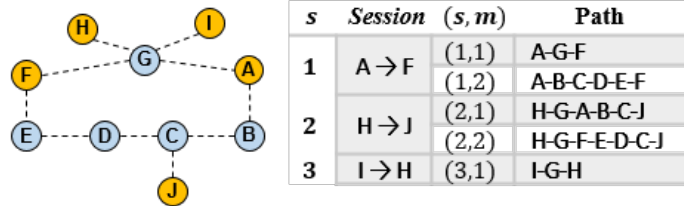


Figure 5.2: Network Topology and the Associated Multipath Sessions.

Figure 5.3 shows the convergence of the 3 sessions. In case $\theta = 2$, mNUM reached 90% of the optimum in 86 iterations. It split the second session on its first and second paths with weights 0.4518 and 0.5482 respectively. For the first session, mNUM converged to 0.00001 and 0.99999 weights for the first and second paths as shown in Fig. 5.3b. Note that mNUM selected the longest and less congested path instead of the shortest and congested one. The mNUM prices for each session-path are shown in Fig. 5.3c, the price of the first, not used, path of session 1 converged to a very high value (outside the range of Fig. 5.3c) which led to neglecting the path. mNUM behaved similarly for $\theta = 10$, splitted and selected the less congested paths, the resultant total session rates are nearly equal (Fig. 5.3d). The higher fairness

achieved with $\theta = 10$ came at the cost of lower aggregate utilities and lower total rate. Compared to the centralized global optimum solutions, Table 5.2 shows that mNUM did converge to the optimum rates that maximize the proportionally fair log rates with marginal error. The optimized aggregate session rates of the shortest single path routes using the NUM algorithm in [11] for $\theta = 2$ and 10 are 0.2585 and 0.2564 Mbps (log rates -3.2095 and -3.2048) respectively. The multipath gain using mNUM is 35%.

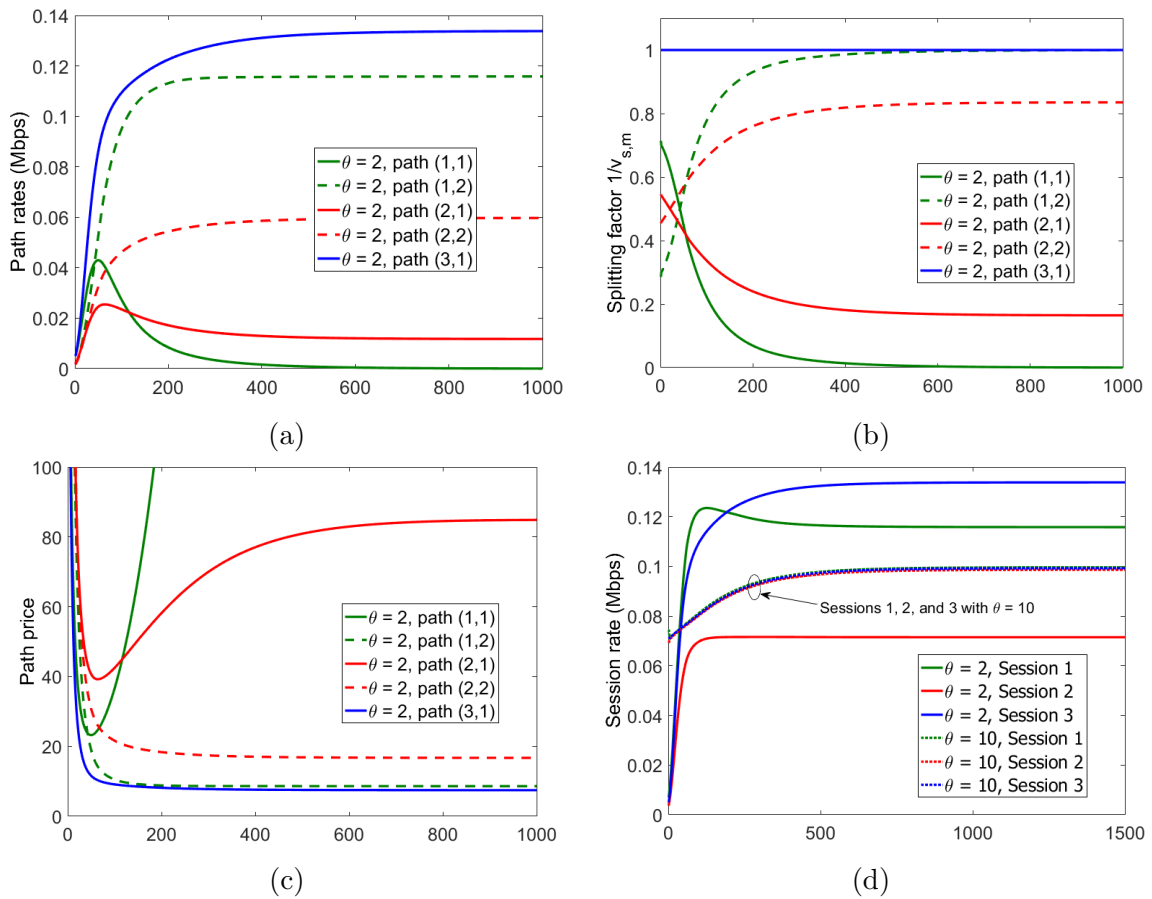


Figure 5.3: The Evolution of Path Rates, Splitting Factor, Path Prices, and Session Rates vs. Number of Iterations.

Table 5.2: mNUM Optimum vs. Global Optimum

	Fairness $\theta = 2$				Fairness $\theta = 10$			
	mNUM Optimum		Centralized Optimum		mNUM Optimum		Centralized Optimum	
(1, 1)	0.0000	0.1205	0.0000	0.1205	0.0000	0.1013	0.0000	0.1056
(1, 2)	0.1205		0.1205		0.1013			
(2, 1)	0.0283	0.0621	0.0283	0.0621	0.0540	0.1008	0.0503	0.0974
(2, 2)	0.0338		0.0338		0.0468			
(3, 1)	0.1678		0.1678		0.1019		0.1014	
Total	0.3503		0.3503		0.3040		0.3044	
Log Rate	-2.9014		-2.9014		-2.9826		-2.9813	

5.2 mNUM Realization

The integration of IEEE 802.11 with mNUM will be achieved as discussed in Chapter 3. The presence of different paths for each session does not affect the formulation regarding the node and link probabilities, as they are implemented at the MAC layer, away from the session and multipath problems. The link capacity formula in Constraint (5.2) will be replaced by Bianchi's estimation Formula (3.3) and the new node and link access opportunities will be given by Equations (3.8) and (3.9). The contention window adjustment will follow Equation (3.11).

The AQM and adaptive step-size mechanisms are employed in the exact way discussed in Chapter 3. We call our adapted algorithm with AQM and the IEEE 802.11 MAC protocol the multipath Simple Network Utility Maximization (mSNUM) algorithm.

5.3 Routes Search

In Section 5.1 we convexified the network utility optimization problem, which includes congestion, multipath routing, and contention control problems. That is, given a set of paths for each session in the network, the mSNUM algorithm can assign rates for each session over its paths so that the overall network utility is maximized according to a certain fairness criteria. The question now is how to find the set of paths for each session.

Other algorithms, mentioned in literature, that solved similar multipath routing problems, included a flow conservation constraint in the problem formulation. With the help of the flow conservation constraint and after enough iterations, the algorithm can find the routes that contribute to the optimization objective by eliminating the

links one by one until the best routes are found and the algorithm converges. Such algorithms would not be suitable for online solutions, as they have to converge to the optimum first before data transmission starts. Otherwise there is no guarantee that the data is moving forward toward its destination.

In our solution, we are mainly concerned with practicality and feasibility of the algorithm for wireless ad-hoc networks. It will not be feasible to wait for the algorithm to converge first, especially in a dynamic topology. Rather, we use the general (standard) routing methods to find multipath routes. That is, mSNUM can work with the aid of any route searching method. The difference is that the algorithm will select that paths that will increase the overall network utility considering contention, congestion and fairness between sessions. The selection and rate allocation over the different paths will be achieved by mSNUM based on prices metric, which reflects the congestion and contention in the network. The route searching method used will keep looking for different routes and provide mSNUM with every new route.

The fluctuations accompanied with the multipath network capabilities, discussed in Section 2.2, have been alleviated by removing the linearity between the network paths in Constraint (5.5). The convexification of the multipath problem makes the objective function sensitive to the paths that can maximize or negatively affect the network utility. As a result, the routes capable of maximizing the network utilities will keep minimal path prices during mSNUM convergence. The other paths will have higher and higher prices, as mSNUM converges, then will be automatically assigned lower and lower rates until they are completely neglected.

For simplicity of the algorithm implementation, only the source of each session is responsible for splitting and assigning rates for each session-path. The intermediate nodes are oblivious about the different paths. For that reason, in order to keep the source node in control of the packet path to destination, we are employing a routing mechanism similar to Dynamic Source Routing (DSR) protocol [59].

5.3.1 Search Method

DSR starts its route search when it has a packet to send to a certain destination by flooding RREQs through the network. A RREQ contains the source ID (initiator), destination ID (target) and the RREQ Sequence Number. Each node other than the target, after receiving the RREQ, appends its ID and rebroadcasts the RREQ if it has not forwarded the same RREQ before, otherwise, the RREQ is discarded. If

the relay node already has information about the route to the target in its cache or if, eventually, the target node receives one or multiple RREQ messages, they will select the shortest path and initiate a Route REPLY (RREP) message to the source containing the selected routes.

In order for the algorithm to find the optimum solution, we need to find all possible paths from each session's source to its destination. This means that the RREQ will be forwarded from each network node as long as the node is not the target or the path traversed by the RREQ does not contain the node's address (loop). We also avoid all the paths containing portions that can be shortened by one hop transmission. Which means that if a path takes more than one hop to go from any node X to X's first hop neighbour, the path will be neglected.

The relays and/or the target node will receive RREQs from different paths. The relay nodes with route cache to the target and/or the target node will send a RREP to the source. The source will provide mSNUM with all the different paths learned from the RREPs received using a according to a feeding mechanism.

5.3.2 Feeding Paths to mSNUM

mSNUM will keep two different lists to track the different paths from the source to destination. The first list is the Active paths list (A list), it contains all the paths that are currently active and used by mSNUM to deliver packets to the destination. The second list is Backup paths list (B list), it contains the paths that are not active either because they have been tested/rejected or haven't been tested yet by mSNUM.

mSNUM starts optimizing rates as soon as there is a path available for at least one session. Initially, mSNUM assigns equal hop prices for each session-path. The initial path price will depend on the number of hops in the path. The first path learned by the source node will be added to A, mSNUM will assign the whole session rate to the single path in A according to the initial price. If a second or third path found, it/they will be added to A. The session rate and the splitting factor will be re-calculated according to the available path prices located in A. Any additional path learned after the third will be added to B unless the new path is shorter than the shortest path in A, in this case, the new path will be added to A directly.

The reason for not adding all the paths found directly to A is to give mSNUM a chance to partially converge, select the appropriate paths, and reject the rest. If, at any time, the splitting factor of any path in A becomes less than 0.05 (5% of the total

session rate), we assume that this implies that mSNUM rejected the path, it will be marked as tested and moved to B. At the same time the shortest untested path from B will be moved to A to be tested. The priority is always given to the shortest paths to be tested first as most probably they will be selected or rejected by mSNUM faster than the longer paths, which helps speeding up convergence.

Beside the exchange happens to paths between A and B whenever one path gets rejected by mSNUM, there is a periodic path move from B to A as long as there still untested paths in B. Every pre-specified period, the shortest untested path from B is moved to A to be tested. The pre-specified period is chosen to give a chance to mSNUM to partially converges, it depends on the iteration period of the algorithm. The flowchart in Fig. 5.4 shows the mechanism to provide mSNUM with the paths found.

If the path being tested happens to be in a congested area where it will affect the overall network utility, its price will increase. As the price increases, the rate assigned to the path decreases. That will continue until one of two cases happens. The first case is that the path reaches a balanced state where the assigned traffic's positive contribution to the aggregate network utility is higher than the congestion and contention it induces. The second case is that the price keeps increasing and the splitting factor reaches a low value, lower than 0.05, at which point the path is neglected and moves from A to B.

5.3.3 The Active and the Backup Lists

The Active list contains the paths that are being used by the routing protocol to deliver packets. Anytime the routing protocol needs to send a packet, mSNUM selects the appropriate path from A according to the splitting factor and the rate allocated for each path. With every new path price received, the splitting factors of all the paths are re-calculated.

The DSR mechanism to detect the broken links and the broken paths is used to keep A and B updated. In the case of a path breakage, mSNUM has its own mechanism to update the source and all the nodes upstream about the broken link. The broken link hop price for each session will be given a very high price (relatively infinity). The mechanism of updating the source about the path failure is discussed in Section 6.4.3. However, the DSR Route ERRor (RERR) message is also used for

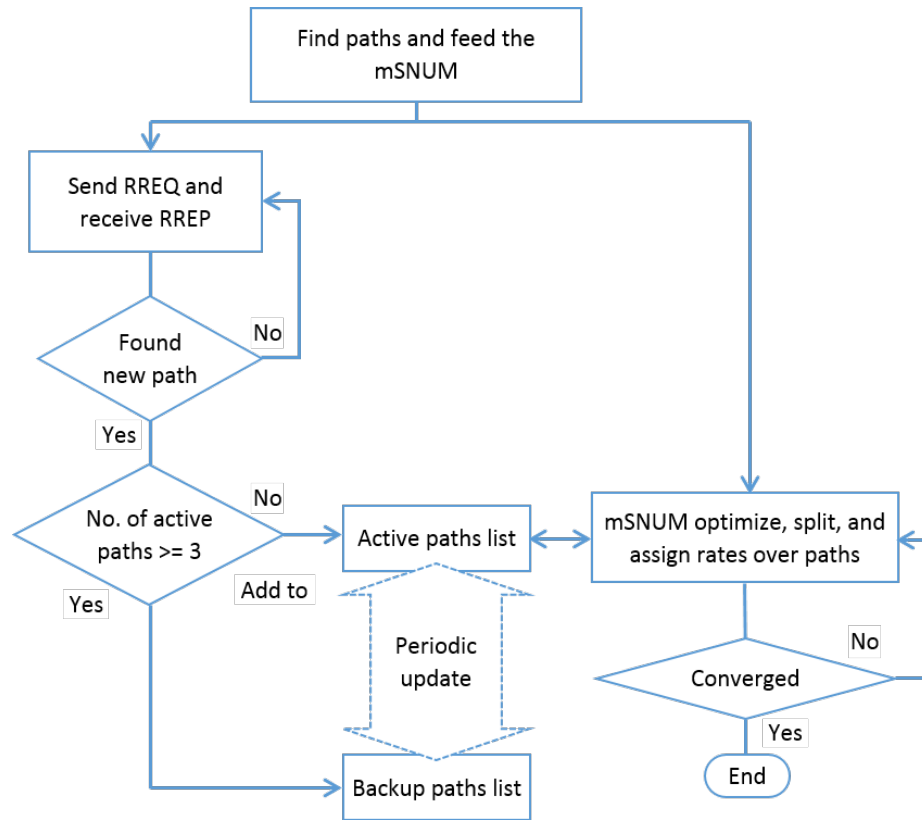


Figure 5.4: Searching and Path Feeding Method.

the same function. RERR is a dedicated message, initiated at the relay node that sensed the broken link, to inform the source node about the path failure. mSNUM takes advantage of both mechanisms, its own or the dedicated DSR RERR message, whichever is faster, to inform the source node and update stale paths in both A and B lists.

mSNUM will automatically reallocate the rate of the broken path to the other active paths available for the session and bring a replacement path from B if A has less than 3 paths after removing the broken path. The replacement path will be selected from the untested paths, if none available, then from the tested paths. If the broken path was the only available path for the session, in both A and B, the routing protocol will issue a new round of RREQs.

5.4 Summary

We formulate a joint congestion, multipath routing, and contention control problem using the NUM framework. By introducing a splitting factor to replace the linearity in the relation between each session and its multipath routes, we extend earlier models to include routing over potentially multiple paths in the optimization framework. The new variable, proper transformation, and using Ohm's law analogy lead to a convex and decoupled optimization framework that can find the optimum solution in a distributed fashion. The distributed algorithm, called mSNUM, requires a single coordination parameter (prices), which reflects the lack/surplus of resources in each link and each session-path. The prices are used as a metric for routing decisions by the session sources.

For the route searching part, we use a searching method similar to the DSR protocol. The mechanism to provide mSNUM with the paths found is discussed in details.

Chapter 6

The Framework and the Algorithm Implementation

In this chapter we discuss the framework and the coordination methods. We show the vertical coordination between the three layers (Transport, Network and MAC). We discuss the horizontal coordination signalling and overhead associated with the CLD.

6.1 The Framework

Our proposed framework preserves the layered architecture of the network by allowing each layer to make its own decisions based on the local experienced dynamics and the messages exchanged among the layers, as shown in Fig. 6.1

The information exchanged between the network layers (represented by the up and down vertical arrows) will carry vertical coordination prices required for the cross-layer optimization process. The information exchanged between the nodes (represented by the horizontal arrows) is required for the horizontal coordination between the nodes. This information will carry the coordination information required by the individual protocols as well as the prices used/generated by the optimization algorithm.

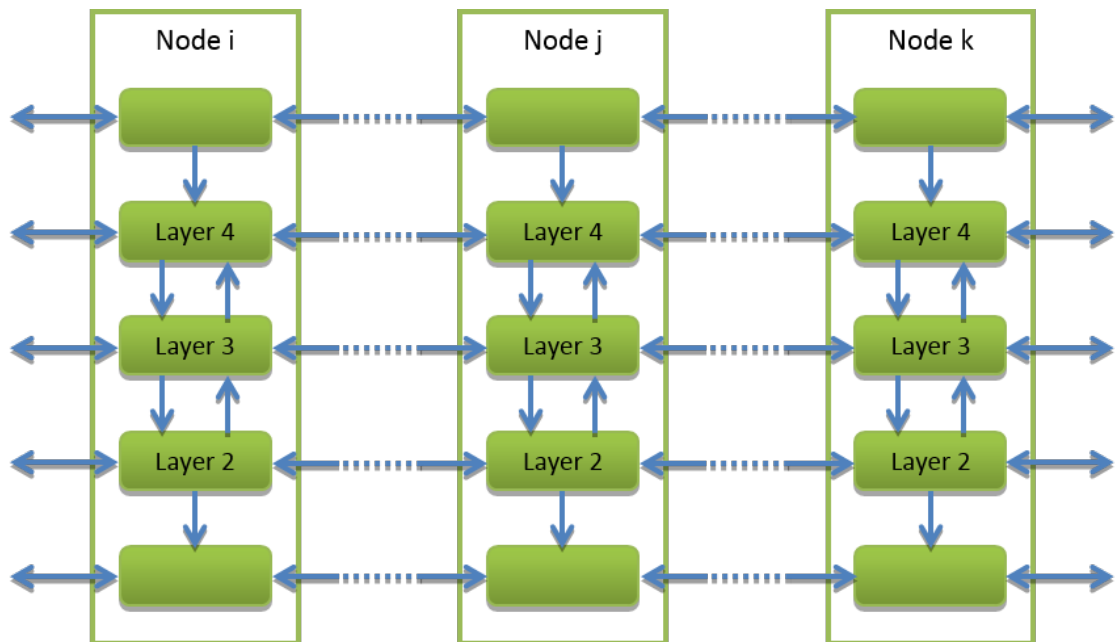


Figure 6.1: Illustration of the CLD Framework.

6.2 Prices

mSNUM individually updates the prices of each path hop for each session, we call them session-path-link prices or hops prices for short. The path price is the sum of all the individual hop prices on each link used by the path, sometimes we call it session-path price to indicate that the path belongs to a certain session. The session price is the average price of all the parallel paths used in the session according to Ohm's law according to (5.11). The link price is the sum of the hop prices of different paths sharing the same link. The node price is the sum of all outgoing links price from the node. Each of the three layers needs to calculate its related prices based on the other prices. The Transport layer calculates the session prices based on the path prices to set the session rates. The Network layer needs the path prices to calculate the splitting factor and the path rates. The MAC layer calculates the link prices based on the hop prices to find the node and link access opportunity, as well as to adapt the contention window size in the case of IEEE 802.11 protocol.

6.3 Vertical Coordination

mSNUM starts with initial session-path-link (hop) prices and follows the flowchart shown in Fig. 6.2.

6.3.1 Transport Layer

The transport layer is responsible for adapting the session rates based on the session prices. It collects the session-path prices from the Network layer, calculates the session prices, and assigns the total session rates based on the defined utility function. We use the User Datagram Protocol (UDP) and control the session rates at the transport layer. The Protocol Data Units (PDU) are forwarded down to the network layer according to the session rates assigned.

If one of the path prices received at the transport layer is infinity, this inherently means that this path is broken and should be neglected. If all the paths that belong to the same session have price infinity, this inherently means that all paths are broken and the session is no longer active. This can only happen if the B list, which holds the backup paths, does not have any suitable connected path for the session. In this case, the transport layer indicates the session as inactive and motivates the network layer to search for a new route.

6.3.2 Network Layer

The network layer collects the path prices from the contributing nodes for every session, which is an accumulation of hop prices over the path. Then, it calculates the splitting factor accordingly. The PDU received from the transport layer are then forwarded to each path according to the splitting factor.

In the case of SNUM, the CLD is independent of the routing protocol used. In the case of mSNUM, the source node is the one determining the paths used and set the paths rate according to the splitting factor. The intermediate nodes, in this case, are not responsible for any extra calculations or processing. For that reason, we use a source routing protocol to give the source node the power it needs to forward packets according to the rates set for each path.

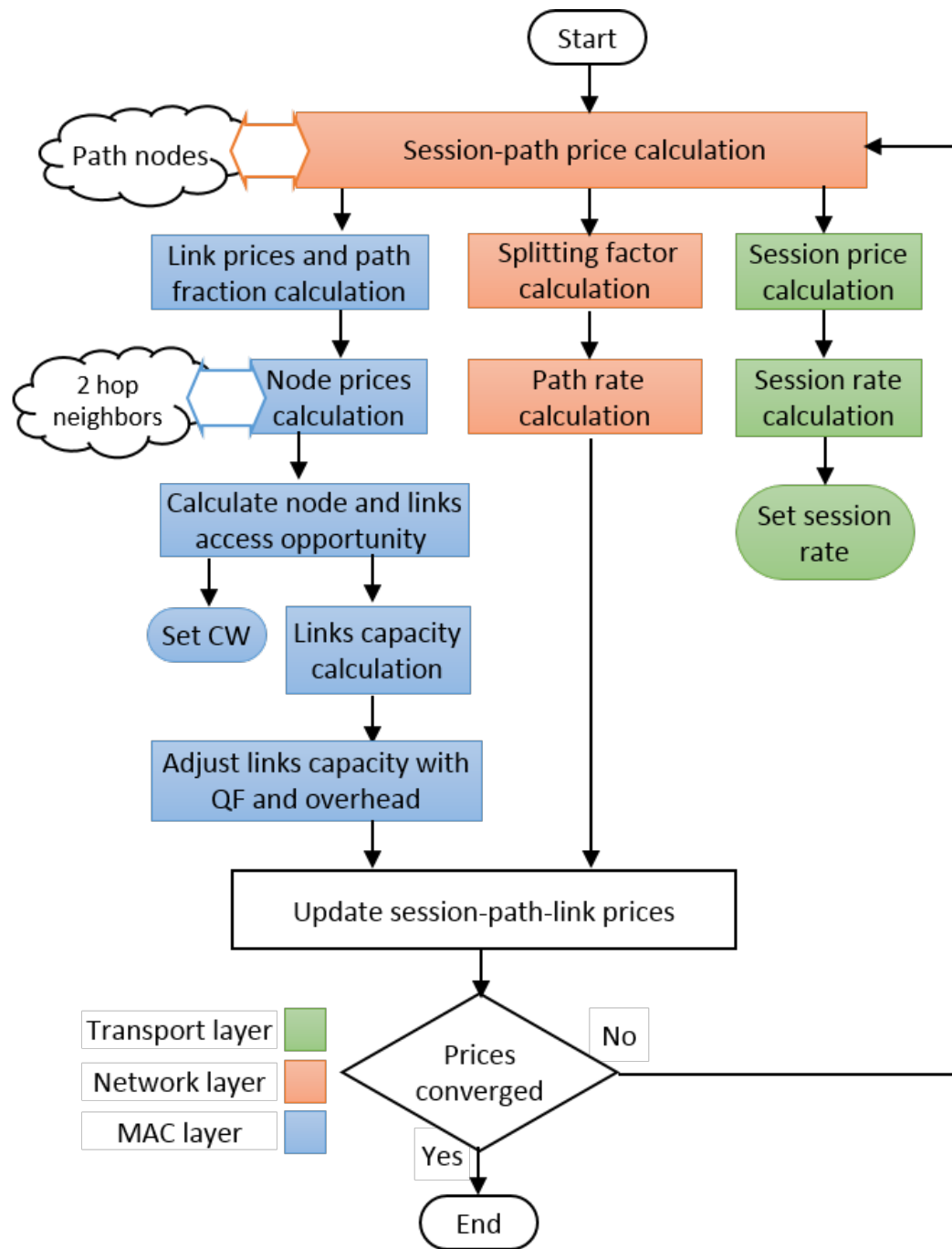


Figure 6.2: CLD Flowchart per Node.

6.3.3 MAC Layer

The MAC layer calculates the node and link prices, collects the 1st and 2nd hop neighbour node prices, and calculates the node and link access opportunities accordingly. The access opportunity is used as the persistent probability in case of the ALOHA protocol or used to calculate the contention window size in case of the IEEE 802.11 protocol. The link capacities are then calculated based on the access opportunities given to the node and to each outgoing link.

The queue length is calculated every iteration to generate the Queue Factor (QF) and the link capacities are adjusted accordingly. The overhead is also considered in the optimization process. The overhead is monitored at the MAC layer and estimated using a sliding window average method. At each iteration, after the link capacities are calculated, the overhead rates are subtracted from the link capacities, and then mSNUM continues its iterations until it converges. With each iteration, the transmission opportunities are adjusted for the node and the links considering the subtracted overhead rates. So, the overhead rate is included in the optimization and the fairness calculations, which help increasing the network utility as the objective function is concerned about the session fairness and not the node fairness.

6.4 Horizontal Coordination

Every node contributing to the optimization process, as session source, destination, or relaying node, has to send update information (overhead) periodically. Sending this information consumes part of the available capacity, but it is necessary in order to deliver the prices to the rest of the contributing nodes. Normally, two types of information transmissions required. The first one is broadcasting, to propagate prices to the 1st and 2nd hop neighbours. The second one uses point-to-point transmissions that send the path prices forward and backward along the paths.

We propose two methods to deliver the periodic updates while keeping the consumed capacity at a minimum level. In the first method, we send the update information in individual messages. In the second method, we piggyback the update information onto other messages.

6.4.1 Individual Update Messages

The goal of the individual update message method is to keep the CLD modular and compatible with the global standards layered network architecture, but that comes at a price of increasing the consumed resources accordingly. We define a separate control message to carry the information required. We combine all the coordination and update information required by the algorithm in only one control message so that we avoid the need for point-to-point message transmission. The message is broadcasted from every active node to its 1st hop neighbours.

The information in the message broadcast by node n are the node price, the 1st hop neighbours' prices, the session price in case of multipath, and the hops prices of each session $s \in \mathcal{S}(n)$ passing through node n . The size of the message broadcast from node n is equal the number of prices carried multiplied by the size of the price in bytes Pr_{bytes} as follows,

$$MSG_{Size} = \left(1 + N + \sum_{s \in \mathcal{S}(n)} (1 + H_s) \right) \times Pr_{bytes} \quad (6.1)$$

where N is the number of 1st hop neighbours of node n , and H_s is the number of hops in the session $s \in \mathcal{S}(n)$.

The rate of price updates depends on the network dynamics and sizes. A simple way is to broadcast the messages periodically. Another way is to adapt the update rate with the change in the node prices. High variation in the node prices indicates that mSNUM still in a transition stage where most of the rates values are not close to the optimum yet. If the variations in the node prices are low, this means that mSNUM is close to convergence. In order to speed up the convergence and fast-forward the transition stage, we adapt the price update rate according to the variation in the node prices. High price variations lead to higher update message rates. That will temporarily increase the overhead of the network but will speed up mSNUM convergence at which the network resources are optimally utilized. After finishing the transmission stage the prices become more stabilized and the overhead rates return to the minimum levels based on the chosen normal update rate.

6.4.2 Piggybacking other Packets

The goal of piggybacking the overhead is to reduce the consumed resources by the overhead individual messages and increase the CLD efficiency. Compared to the node prices update frequency, the path prices update frequency should be much higher for the following reasons:

- The time required to inform the nodes in each path about the new price depends on the number of hops in the path.
- The queue management technique uses the path price to inform the source about the congestion in the path at each link (every hop).
- If there is a broken link due to the dynamic nature of the network, besides the queue management, the path price is the one responsible for informing the source node not to use this path anymore.

For these reasons, we make the path price update rate as fast as the data packets in the sessions-path. We add the accumulated hop prices to the header of the data packets, in order to deliver it to the downstream nodes. The accumulated hop prices are also added to the MAC acknowledgements to deliver it to the upstream nodes. Note, the accumulated hops price is a single price size. If the data packet transmission rate is high, we need the update frequency to be very fast for fast rate adaptation according to the queue length and broken links. Otherwise, the prices update rate will be as low as the the data packet transmission rate that could save the network resources by avoiding unnecessary frequent price updates.

For example, in the path shown in Fig. 6.4, we need to inform the session nodes downstream and upstream about the prices. For the downstream part, the data packet will carry the accumulated hops price from source to the destination in its IP header options field. The price size = 4 bytes, in case of multipath sessions, we need to distribute the session price as well, the two numbers of size = 8 bytes. The Type-Length-Value (TVL) of the IPv4 option field is shown in Fig. 6.3. At each hop from the source, the hop prices will be added to the one carried in the data IP header and the result will replace the one in the header, then the data packet will be forwarded to the next hop until it reaches the destination. For the upstream part, the MAC layer ACK packet generated by the MAC layer after receiving a data packet will carry the accumulated prices from the destination to the source. At node x for

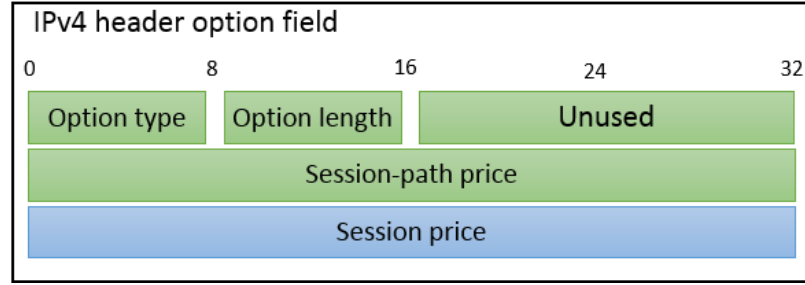


Figure 6.3: IPv4 Option Filed TLV.

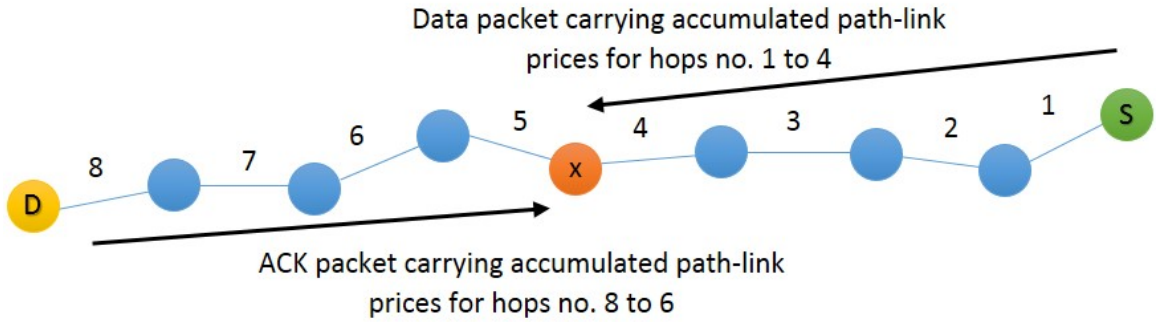


Figure 6.4: Illustration of Overhead Piggybacking.

example, the session-path price will be sum of prices received from the data packet + the ACK packet + the current hop price (no. 5). The session-path price at x will be calculated every $RTT/2$ as the prices are accumulated from both directions.

As for the node prices, we define a separate control message to broadcast them to the 1st and 2nd hop neighbours. The message is broadcast from every active node to its 1st hop neighbours. The information in the message broadcast by node n are the node price λ_n and the 1st hop neighbours' prices. The size of the message broadcast from node n is equal the number of prices carried multiplied by the size of the price in bytes Pr_{bytes} as follows,

$$MSG_{Size} = (1 + N) \times Pr_{bytes} \quad (6.2)$$

Considering the prices as the horizontal coordination language between the nodes, each transmitted packet carrying the transmitter identification in its headers can also carry the transmitter's price (node price). We attach the node price to the data packet MAC header to broadcast the price. The 1st hop neighbour nodes can use the

price in the MAC header of the overheard data message even if the message is not intended to that receiver. In other words, even if the destination MAC address of the data message is not the same as the receiver MAC address, the prices located in the MAC header is used to update the receiver's information about the neighbours' prices. That also keeps the neighbourhood topology information updated regularly without exchanging specialized overhead messages.

Piggybacking the update information will save network resources and increase the convergence speed as compared to the individual update message method. In this case, the MAC layer will need to keep track of the different session paths transmitted by the node. The MAC layer should be able to identify which session-path a given data packet belongs to and piggyback or retrieve prices accordingly. That can be achieved by reading the Network and Transport layer header information in the packet to identify the packet at the session level. Another way is to explicitly provide session-path information for the MAC layer for price identification purposes.

In order to speed up the algorithm convergence, the algorithm parameter updates can be done more frequently with every new packet arrival, i.e. every new price received with every new packet. Normally, the algorithm would update its parameters (link prices, node price, link fractions, splitting factors) every iteration. With piggybacking the algorithm prices on the data packets and 802.11 MAC ACKs, there is a new accumulated session-path price arriving with each packet and ACK received. The algorithm can update its parameters with every new price received instead of waiting for the iteration period. The only drawback associated with this mechanism is that it may remove any time scaling between the parameter updates at the transport, network, and MAC layers. Which means, the changes happen at the MAC layer, such as the queue size, link capacities and access opportunities for each node, will affect the transmission rates and splitting factors with nearly the same rate. In this case, high fluctuations to the link capacities due to fast changes in the queue size will affect the session price and rates as well.

The solution to this problem is simple, we emulate the different time scales between the layers by taking the exponential moving average of the link capacities at the MAC layer after adjusting it with the QF. The smoothing factor of the exponential moving average depends on the average number of new prices we receive during one iteration. For example, if the node receives 20 new prices every iteration period, then the something factor will give a weight of $1/20$ to the new value and $19/20$ to the old one.

This way we keep high convergence rates and low fluctuations in the session rates.

6.4.3 Link and Path Failures

Consider the session path, shown in Fig. 6.5, from a source S to a destination D. The path has a broken link outgoing from node X and incoming to node Z. There will be several steps to detect this breakage and inform the session source to stop using this path.

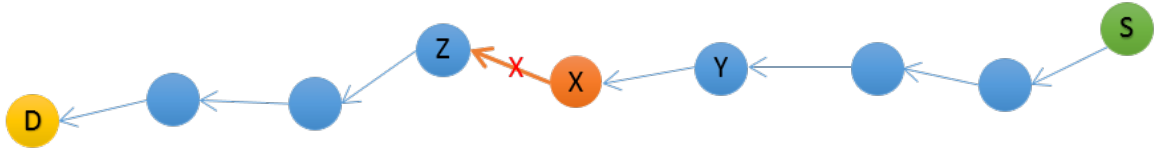


Figure 6.5: Session Path with a Broken Link.

How do we know that the link is broken?

In case of a general MAC protocol that does not send ACK messages after a successful transmission, the sender will declare the node Z as obsolete neighbour if it does not receive its node price update message for a certain period. Otherwise, we use link-level feedback, i.e. if node X does not receive a MAC ACK after a data packet transmission and several retransmission attempts, node X will mark neighbour Z as obsolete, which also implies that link X-Z is broken.

What happen to the session-paths that use the broken link?

The broken link is outgoing from node X, so all paths using this broken link are leaving node X, the hop price for these paths are set by node X. So, after marking neighbour Z as obsolete, the paths using the X-Z link will be marked as inactive. Node X will also assign a hop price of infinity (relatively a very big number) to all inactive paths. That will prevent these session-paths from getting any more resources from node X. This is done by not considering the prices of these session-paths in the total link price or the node price. In other words, no access opportunity is assigned to the broken link and the paths using it.

How to inform the source and the rest of the relaying nodes about the broken session path?

For the upstream relaying nodes, the broken hop price will be added up to the rest of the path hop prices to accumulate the path price, which will be also infinity. When

node X broadcasts the prices update message to its neighbours, or the accumulated prices upstream, this infinity price will spread to all the upstream nodes. Every intermediate node receives the infinity price, will mark the related session as inactive, and stop assigning resources to it. The infinity price will spread upstream with each update message until it reaches the source node. At the session source, the broken session-path will be removed from the path set. In case of a single session path, the session will be declared inactive and the procedure to use a different path will start as discussed in Section 5.3.

For the downstream relaying nodes, they cannot hear the broadcast infinity price of the broken hop. Every node will have an active session time-out to keep track of the session activity. If the session is not active for pre-defined time period, the session will be declared as inactive and the node will stop assign resources to it.

6.5 Summary

We discussed the proposed framework and the use of the prices to connect the different network stack layers. The vertical coordination between the different layers is explained along with the functions executed at each layer. The horizontal coordination implementation is described using two models, individual update messages and piggybacking other packets. Finally, we discussed the mechanism used to identify and inform the session sources about broken links and paths.

Chapter 7

NS-3 Simulation Results

Our approach is based on an unreliable transmission control protocol (ex. UDP [60]). Unlike TCP, UDP does not retransmit the lost packets nor acknowledges reception of messages. For that reason, for comparison fairness, we need a rate control protocol that does not consume resources for packet retransmissions and maintaining packet sequence.

TCP-Friendly Rate Control (TFRC) [61] is a standard protocol based on UDP. Its goal is to keep the UDP traffic rates less than twice the other individual traffic rates over shared resources. Unlike non-standard rate control algorithms mentioned in the literature, TFRC went through many review, test, and enhancement cycles by different scholars and specialists during the standardization process. Another advantage of using TFRC is that its standard document [61] provides all the implementation information of the protocol with enough detail for a complete and thorough simulation in ns-3 . On the other hand, most of the cross-layer designs available are not standardized, did not go through rigorous testing and enhancing processes, and do not provide implementation details suitable for ns-3 implementation and simulation.

The utility function used provides different fairness criteria to each individual session. The network utility metric used, which is the log of the rates, reflects both, the session rates and the fairness between them. We used Jain's fairness index to reflect the fairness between the sessions without reflecting the rate values in the results (pure fairness metric). The speed of convergence of iterative algorithms can be measured by the number of iterations required to converge within a certain error margin. However, the iteration period can be changed according to the design and implementation. So, the absolute time required cannot be a good measure for the speed of convergence. In case of our algorithms, especially in case of mobility, there

are a lot of factors affecting the speed of convergence, other than the algorithm itself. For example, the nodes' speed and the accuracy of the topology information, the state of the medium and the number of packet losses, the number of hops per session and the density of the network. For that reason we choose a general metric, the network utility, to measure how well the algorithm performs compared to other methods. The network utility metric measures how fast the algorithm can adapt to topology changes, inaccuracy and information losses.

Using NS-3, the network simulations are carried out for our Cross-layer SNUM and mSNUM compared to a layered standard adaptive rate control protocol TFRC and the constant rate control protocol UDP under the same conditions. We decided to evaluate the algorithms for worst-case scenarios in IEEE 802.11 networks. The physical link capacity is the lowest available rate, the physical layer uses Direct-Sequence Spread Spectrum (DSSS) with 1 Mbps physical link capacity. The transmission range is set to be half the sensing range, resulting in a transmission range of 100m and a sensing range of 200m. The data message size generated at the application layer is 600 bytes to keep the packet size below the RTS threshold.

For consistency, unless otherwise specified, each value of the following results is an average of 10 runs, using different random number seeds that will change the back-off times, the transmission jitter in each node, and the mobility scenarios. The TFRC parameters are chosen to be the default values provided in TFRC standard document [61]. The fixed rate of the UDP transmitters is set by dividing the physical capacity over the number of sessions available. Each session gets a share of 1 Mbps/Number of sessions in the scenario, which means that each link has enough physical capacity to carry the sum of all the sessions in the network. The default value of the aggressiveness parameter Z related to the QF is set to 0.1 for SNUM and 0.01 for mSNUM, different values of Z are discussed later. MAC layer queue size is 400 packets.

Simulating the static network is an essential part to study the algorithm performance before mobilizing the network nodes. The mobility adds more challenges such as inaccurate topology information, routing, randomness, more packet and coordination information losses. For that reason, the algorithms are tested and evaluated with static networks first to separate the challenges facing the algorithm and to accurately analyze strong/weak performances.

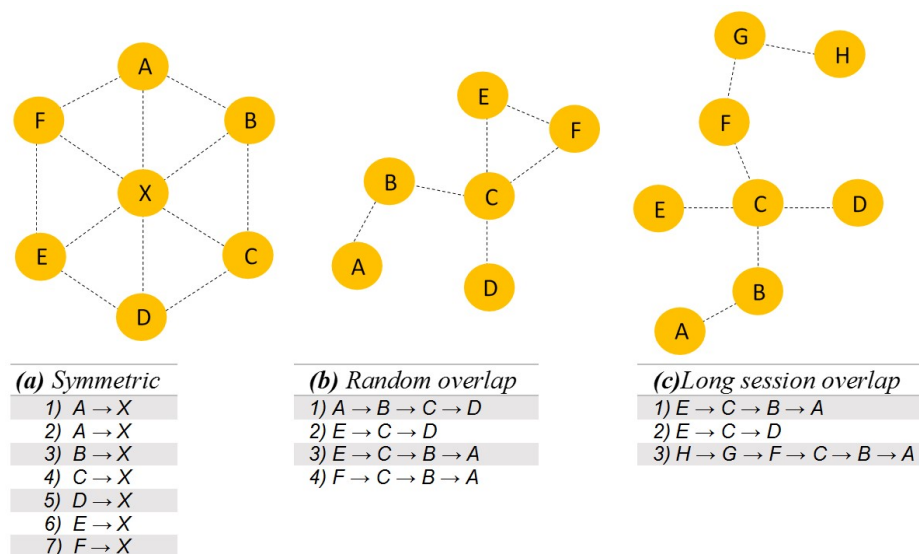


Figure 7.1: Experimental Wireless Network Topologies.

7.1 SNUM Simulations

We test SNUM using different settings and update message methods. First we show its performance with fixed topologies and static routes using the individual update message method. Then, using the piggybacking update method we study the performance on static and mobile networks employing OLSR routing protocol [62].

7.1.1 Fixed Topologies with Static Routes

In order to show the absolute gain (considering overheads) in SNUM performance, compared to the layered one, we created and tested fixed small network topologies. The simulations are carried out using a static route to minimize the disturbing factors in the network and focus on the behavior of SNUM and its components. We use the individual update message method discussed in Section 6.4.1, the update information is send periodically with every new algorithm iteration.

The experiments are conducted under three network configurations, (a) symmetric, (b) randomly overlapped flows, and (c) long/short flows with bottleneck node as depicted with their flows in Fig. 7.1. In (a), each node sends one session to the center node X, except for node A, which hosts two simultaneous sessions transmitting data to X. (b) and (c) have bottlenecks at node C and a range of session with different

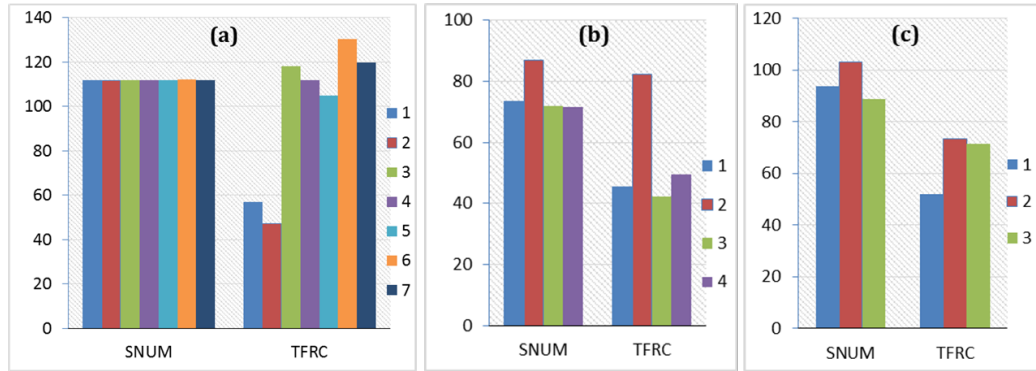


Figure 7.2: The Throughput of each Session.

lengths.

Figure 7.2 shows the throughput of the different sessions for the three topologies. For topology (a), with symmetric interference between nodes, all sessions achieved nearly equal rates, even the first and second sessions that are generated from the same source. On the other hand, TFRC cannot provide fair rates for the first two sessions as it is bounded by the MAC layer fairness. For the general topologies (b) and (c), we can see that SNUM provides higher throughput and better fairness among different session lengths. The shortest session (no. 2) in (b) and the longest (no. 3) in (c) have proportionally fair rates not much higher or lower than the rest of the sessions in the same topology.

Table 7.1 shows the numerical average of different network measurements. The fairness among the sessions at their sources and destinations are calculated using Jain's fairness index. SNUM fairness at the transmitters' side shows that the proportional fair rates set by SNUM are higher than 0.99. The fairness among the sessions at the receivers' side can be lower due to uneven packet losses, congestion, and delays. But SNUM's fairness at the receivers' side still reaches a fairness index of 0.99 even in a random asymmetric topology with different session lengths.

The average delay using SNUM can be reduced around 150 times in single-hop transmissions in topology (a) and at least 3.3 times in multihop sessions with different lengths. SNUM's average packet loss percentage is also 40 times lower than TFRC. The average throughput values show that SNUM can provide 38% and 45% extra throughput, to the networks (b) and (c) respectively, by coordinating the work of the MAC layer with the transport layer. The overhead required for SNUM coordination

Table 7.1: Comparison Results SNUM vs. TFRC for Three Topologies

	<i>(a)</i>		<i>(b)</i>		<i>(c)</i>	
	SNUM	TFRC	SNUM	TFRC	SNUM	TFRC
<i>Jain's fairness (transmitter side)</i>	0.99999	0.84959	0.99335	0.84513	0.99557	0.87547
<i>Jain's fairness (receiver side)</i>	0.99999	0.83278	0.99293	0.83934	0.99628	0.84651
<i>Delay (mSec)</i>	14	2115	439	3664	1210	3982
<i>Packet loss (%)</i>	0.13	8.39	0.35	14.19	0.08	16.67
<i>Total throughput (Kbps)</i>	782.8	688.7	303.6	219.35	285.7	197.0
<i>Overhead (Kbps)</i>	3.55	8.88	3.94	12.49	4.83	11.20
<i>Total Utilities</i>	14.34	13.76	7.51	6.89	5.93	5.44

is less than half of TFRC's overhead in all cases. With the lower overhead, SNUM can provide an additional 45% throughput in the long session topology (c). SNUM overhead rates in (b) and (c) are a function of the number of neighbours and the number of session hops based on (6.1). The TFRC overhead rate depends on the number of session hops and the feedback report update rate. The overall average SNUM utility, which is the aggregate log rate of the sessions, is higher than TFRC in all cases. This is achieved at the same time SNUM results in much lower average delay and packet losses.

Figure 7.3 shows the throughput rates of each session in case of random topology (b). These values are results of a single seed simulation, not an average. SNUM rates converge directly to the optimum rates from any initial point (i.e. any initial prices). The very small turbulences in SNUM (solid lines) between times 100 and 120 seconds are due to QF adjustments happening during convergence. TFRC adjusts its rates after each packet loss and feedback report received, resulting in many peaks and valleys during operation. A more stable/predictable rate is suitable for adaptive applications such as video streaming. Unstable rates would require long queues to enhance the performance, however, the queues would not help if the application is not delay-tolerant.

Testing SNUM against update packet losses, asynchronous updates, and inaccurate topology information are inherently included in the mobility network simulations discussed in the next sections. However, the performance degradation that accompanies mobility could happen due to other reasons, such as imperfect routing. Here we test SNUM with static network and static routes in the presence of update packet

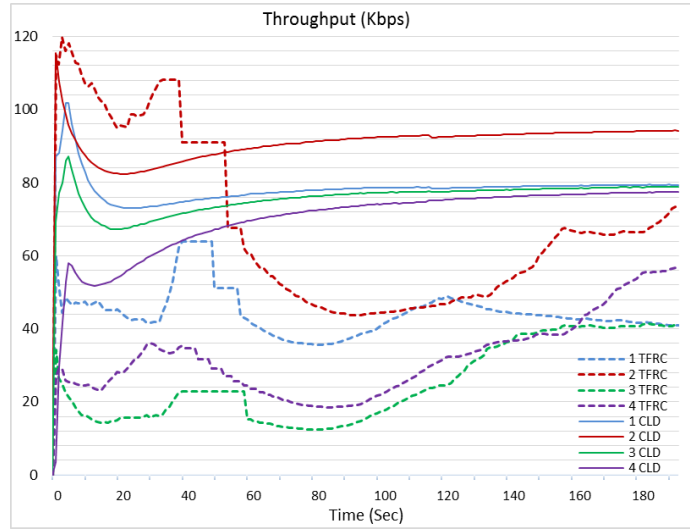


Figure 7.3: Throughput Convergence of Topology (b).

losses, that would suppress any other factor and reflect the vulnerability of SNUM more accurately. We randomly drop/neglect a certain percentage of the received update information at each node. Moreover, in order to induce a mismatch in the update information received, the neglected information can be the node prices only, the session prices only, or both. The results shown in this part (Figs. 7.4 and 7.5) are an average of 5 simulations with different seed numbers.

The simulations show that SNUM manages to converge even with very high update packet loss (up to 95%), but it requires longer time to reach the optimum values. The higher the update information loss rate, the lower is the network utility achieved. Figure 7.4 shows the utility losses for topologies (b) and (c) with different percentages of update information random losses. The utility loss happens due to the slow convergence as the update information gets lost more often. For example, with 10% update loss, the session rates start from zero and approach the optimal values with margin of error in the first 8 seconds. For 60% update loss, the session rates take around 25 seconds to reach the optimum.

The update information loss can only delay the convergence but does not diverge the algorithm away from the optimum. The reason for the algorithm stability even with mismatched update information (node price and session price) is the use of a single coordination parameter and a single update function (3.10). Even if many update prices got lost in the communication between nodes, the fact that all the nodes

use the same price building block (session-link price) helps them converge as long as they eventually receive some updates from each other. Figure 7.5 shows the average standard deviation of the rates vs. different percentages of the update losses. The fluctuation of the rates increases as the losses increase, which means that less smooth convergence happens as the update information losses increase. The continuous update gradually drives the algorithm to the optimum rate with small increments over time, on the other hand, high update loss rates lead to less smooth changes and higher standard deviation. We can see that the noticeable degradation in the performance starts when more than 80% of the update information is lost, which shows a very stable performance.

The effect of the aggressiveness parameter Z on the network average delay is shown in Fig. 7.6 along with the resultant average throughput for network (c). Lowering the Z value tends to sharpen the decaying QF in (3.13) that throttles the link capacity as soon as the queue starts to building up. On the other hand, lowering Z will also lower the total throughput in the network due to lower capacities and rates.

7.1.2 Static Networks

The static network topologies are randomly generated within a square shaped area of node density ≈ 0.00024 node/ m^2 , and session density of $\approx 1/2$ session/node. The routing protocol used is OLSR. From this point forward to the end of the results, we use the piggybacking update message method discussed in Section 6.4.2.

We simulated four different network sizes as shown in Table 7.2. Among the

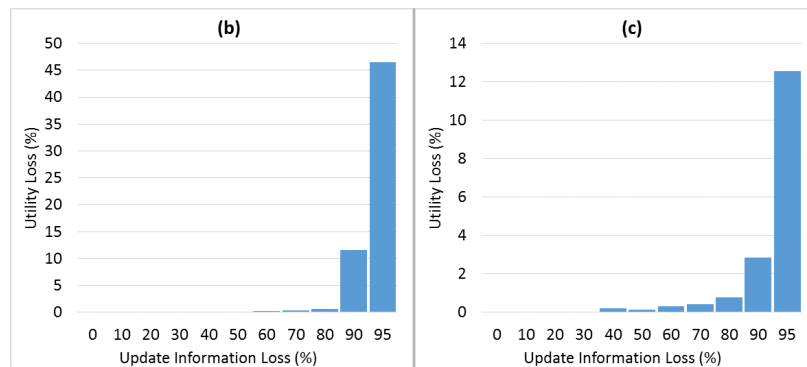


Figure 7.4: Utility Loss Percent due to Update Information Loss for Topologies (b) and (c).

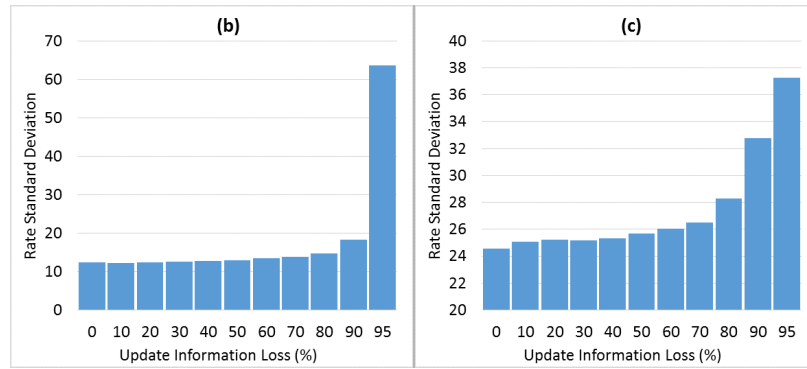


Figure 7.5: Rate Standard Deviation increase due to Update Information Loss for Topologies (b) and (c).

generated static scenarios, only the connected network topologies are selected to be among the 10 different scenarios tested. The average number of hops of all the session in the 10 different scenarios is increasing as the network size increase, from topologies 1 to 4.

Table 7.2: Static Network Topologies

<i>Topology Number</i>	1	2	3	4
<i>no. of Nodes</i>	9	23	40	60
<i>no. of Sessions</i>	5	12	20	30
<i>Square Side length</i>	200	300	400	500
<i>Average no. of hops</i>	1.28	2.49	3.33	4.12

Figure 7.7 shows the aggregate log rates and 95% confidence intervals of SNUM, TFRC, and UDP for the four static topologies. Due to the fixed network resources, the log rates (utilities) for each session decrease as the networks size increases. SNUM's gain is significant, compared to TFRC and UDP in larger networks, where sessions have a higher average number of hops. Longer sessions require more coordination between network nodes and layers to alleviate congestion and optimize transmission opportunities given to each node.

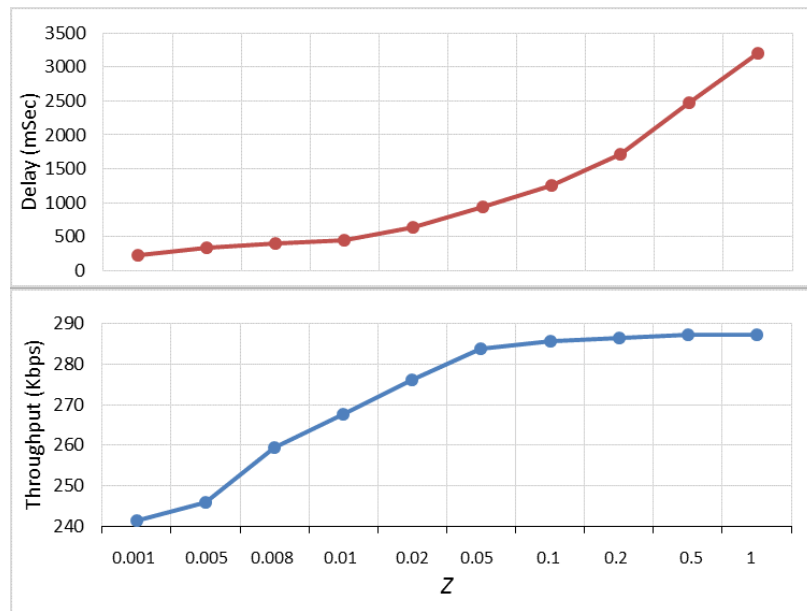


Figure 7.6: The Effect of Aggressiveness Parameter Z on SNUM Average Delay and Throughput.

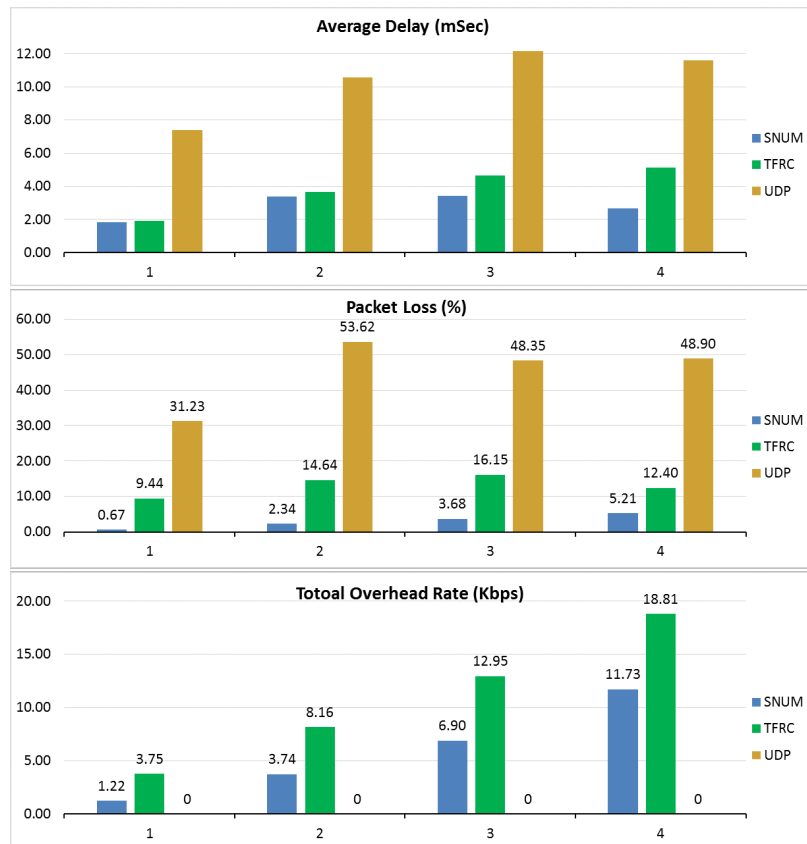


Figure 7.8: Average Delay, Overhead, and Packet Loss for Static Topologies.

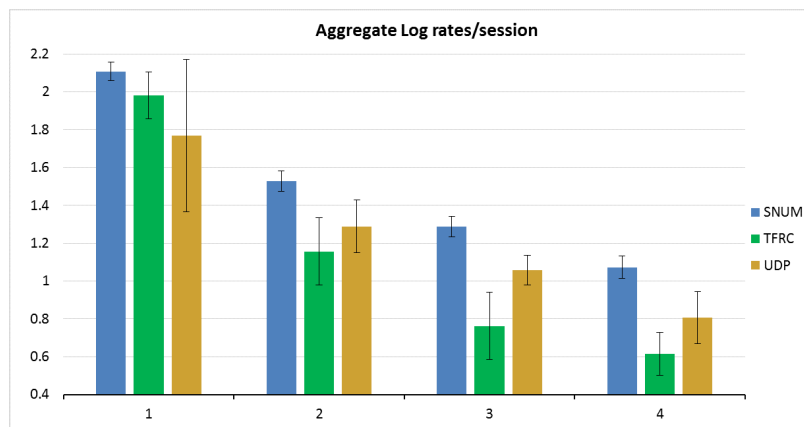


Figure 7.7: Aggregate Log rates/session for Static Topologies.

From the UDP results in Fig. 7.7, it may seem that the fixed rate UDP performs better than the adaptive rate TFRC. In order to see the full picture, we show the drawbacks of the fixed rate UDP sessions in Fig. 7.8 in terms of the average delay and packet losses. For the fixed rate protocol, there is no adaptation based on the network state that assigns rates (resources) based on the packet delay and losses. UDP loses more than 50% of the packets for Topologies 2 to 4. It also has almost three times average packet delay, compared to SNUM and TFRC. Reducing the UDP rates will reduce the packet loss but will also reduce the network utilization. For example, setting the UDP fixed rates based on the hardest expected bottleneck capacity in the network may help utilize the capacity in the area around the bottleneck but it will also underutilize other areas. SNUM not only shows higher average log rates, it also generates approximately half the overhead generated by TFRC and results in much reduced average packet loss.

7.1.3 Mobile Networks

The dynamic scenarios, including node mobility, are generated using the Random Direction 2D mobility model, for 30 nodes in a square shaped area of 300m side length. The session density is 1/2 session/node. We ran the simulation with three different speeds: 0.5, 2, and 5 m/sec with periodic pause intervals of 2 seconds every 4 seconds. Figure 7.9 shows that the aggregate log rates for SNUM and UDP exceed the results for TFRC for speeds of 0.5 and 2 m/sec. Simulations with a speed of 5 m/sec tend to give similar results for all protocols. Figure 7.10 shows that SNUM results outperform TFRC and fixed rate UDP in terms of average delay and packet

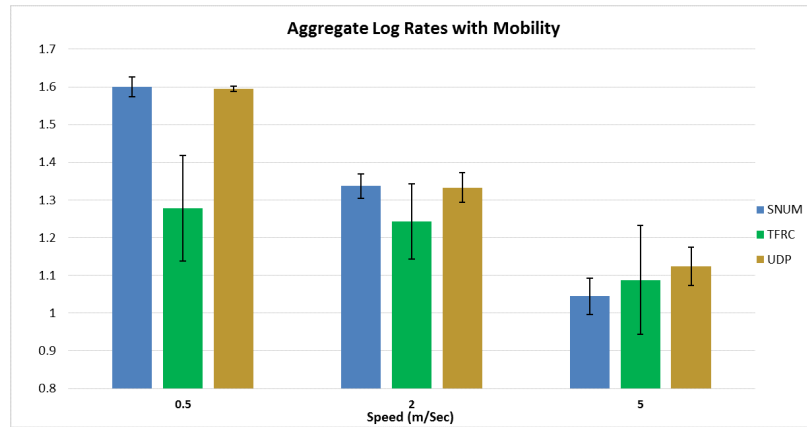


Figure 7.9: Aggregate Session Log Rates for Mobile Topologies.

losses by at least 50%.

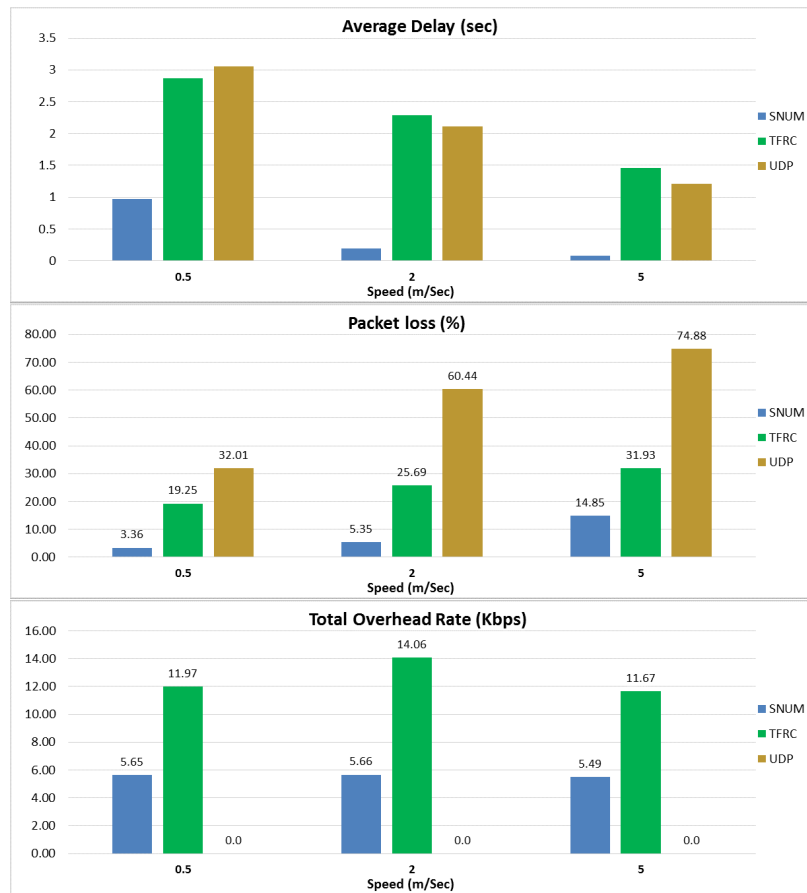


Figure 7.10: Average Delay, Overhead, and Packet Loss for Mobile Topologies.

7.2 mSNUM Simulations

After searching and consulting several researchers working in the same area, we found that there is no publicly available multipath routing module code for NS-3, we had to build one. We extended the DSR routing module that already exists in NS-3 to add multipath capability to it. The extended module can find link disjoint paths based on the MultiPath-DSR (MP-DSR) draft [63] and the path searching method described in [64]. With every packet transmission, the model randomly (obviously) selects and use one path among the available paths to forward the data packet.

Our comparisons include the performance of rate control algorithms based on multipath and single path routing protocols for CLD and OLDs. The standard DSR routing protocol is used for all single path rate control scenarios. MP-DSR is used for TFRC and UDP multipath rate control scenarios. We denote the use of SNUM over single path routing protocol as SNUM/DSR, the same notation is also true for TFRC and UDP. TFRC/MP-DSR and UDP/MP-DSR notations are for the use of TFRC and UDP respectively over the multipath DSR routing protocol.

The metric used in our comparisons is the aggregate log rates for all the network sessions which reflects the throughput and its proportional fairness of the method used. Each point in the graph is an average of 10 numbers resulted from running the simulation 10 times with different seed numbers. In the previous section, we showed SNUM's performance against different random network topologies and sizes. In this section we focus more on the multipath vs. single path with certain topologies that emphasis the performance differences.

7.2.1 Static Networks

As shown in Section 4.2, the multipath use is not always beneficial for the overall network utilities. We show the performance of mSNUM compared to SNUM and other OLDs using two fixed topologies that have different responses toward multipath. The first topology, shown in Fig. 7.11, is of a square area, 9 nodes and 3 sessions. Each session can use 3 to 4 paths, the paths vary in the number of hops from 2 to 5 hops. Topology 1 does not benefit much from the multipath routing, i.e. mSNUM always converges to a single path route for each session.

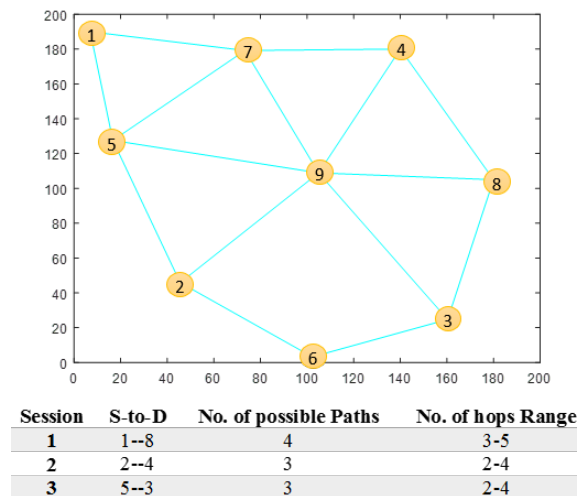


Figure 7.11: Topology 1, 9 Nodes and 3 Sessions.

The second topology, shown in Fig. 7.12, contains 15 nodes and 3 sessions, 2 of them can use up to 3 paths to the destination while the third one can use up to 25 different paths. There is also a variety of session path lengths for each session.

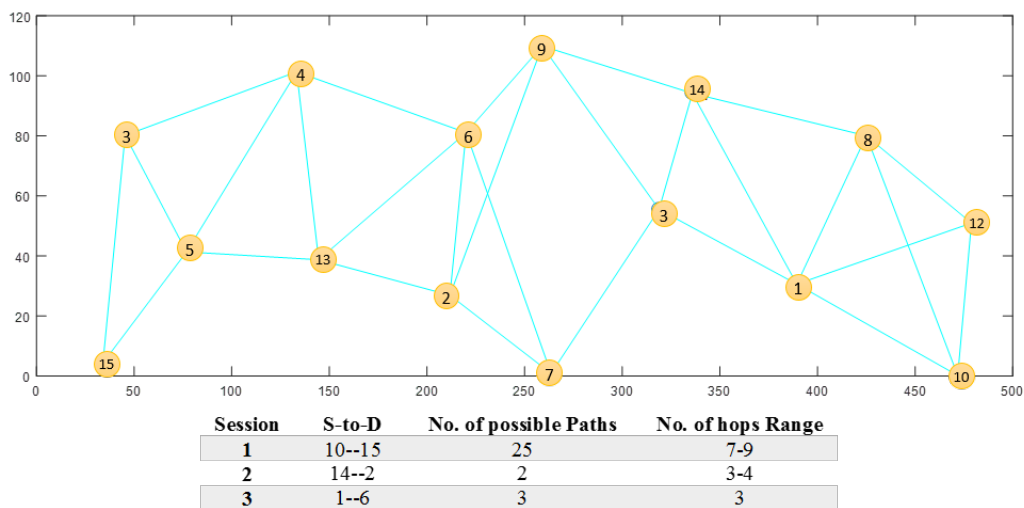


Figure 7.12: Topology 2, 15 Nodes and 3 Sessions.

Figure 7.13 shows the aggregate log rates of the multipath approach compared to single path using the CLDs, TFRC, and UDP for the two topologies. As mentioned before, Topology 1 does not benefit from multipath a lot as mSNUM always allocates the rates to single path just like SNUM/DSR. The reason for that is the high interference which takes away the multipath benefits. For example, if any of the sessions

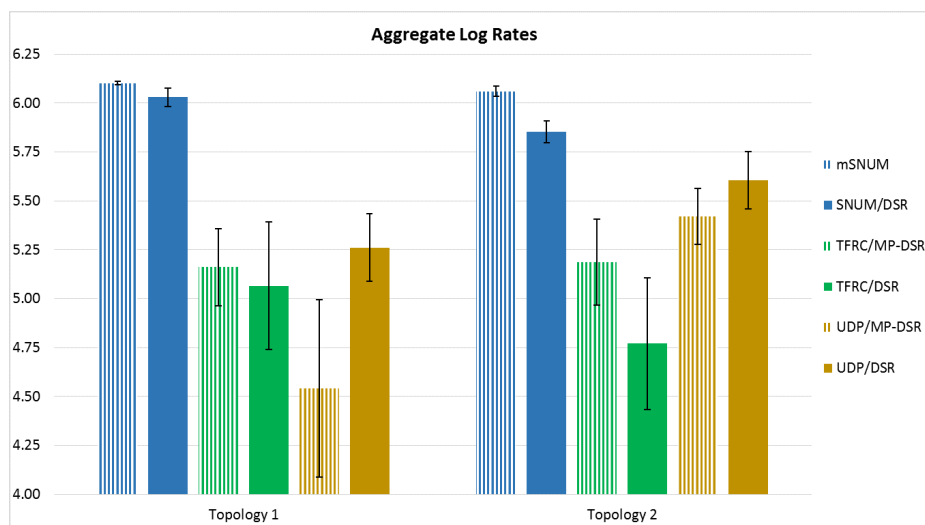


Figure 7.13: The Log Rates for two Topologies with the 95% Confidence Intervals.

use node 9 in one of its paths, due to IEEE 802.11 interference model, all the other paths should be silent during node 9's transmission, this is applied on the rest of the sessions as well. So, there is no point of activating several paths to a destination as long as only one path can be active at a time.

We can still see a slight gain in the utility of mSNUM compared to SNUM/DSR, the reason is that mSNUM tests all the paths available and selects the one that has the minimum contention and congestion effect and provides higher throughput (i.e. results in higher network utility). On the other hand, SNUM uses the path provided by the single path DSR, which could be the shortest path but not the best one to minimize congestion and contention. Another reason for the performance gain is the presence of different backup paths instantaneously available to be used by mSNUM in case the primary path fails, while SNUM/DSR may need to search for a new path in case of path failure. That is why we can see a small mSNUM network utility gain even without activating more than a single path to the destination.

For Topology 2, the utility gain of mSNUM compared to SNUM is clear, mSNUM used at least 8 different paths for the long session (10 to 15), which helped dividing the load over the congested nodes in the middle of the path to the destination. The non-overlapping 95% confidence intervals show a consistent gain and also a better connection stability using mSNUM compared to SNUM.

Multipath DSR provides more resources to the rate control methods used, whether

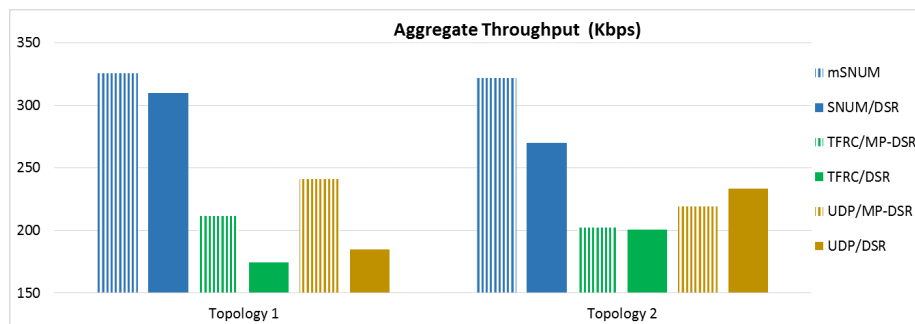


Figure 7.14: The Aggregate Throughput.

it is adaptive rate control, i.e. TFRC, or fixed rate control, i.e. UDP. UDP/MP-DSR shows degradation in the overall network utility compared to UDP/DSR. The reason for that can be better explained after we look at the UDP throughput for Topology 1, shown in Fig. 7.14. Even though UDP/MP-DSR achieves higher throughput compared to UDP/DSR, its log rates in Fig. 7.13, which reflects the throughput proportional fairness, is very low. Which means that the fairness is severely impacted when we use fixed rate control with multipath routing. For better explanation, consider that Session 2 uses three paths to the destination while Session 3 uses only one path. Session 2 will be able to flood the entire topology with its data packets and achieve much more throughput compared to Session 3. Session 3 will not only suffer from unfair resource assignments, but also from the interference of the many paths used by Session 2. The presence of the multipath routing capability in the absence of any fairness criteria in UDP worsens the performance. On the other hand, the fairness provided in TFRC, which keeps the maximum-to-minimum session rate ratios lower than 2, controls the rate of the sessions and keep it fair even if one session uses many more paths than other sessions.

For Topology 2, the gain of using multipath compared to single path for TFRC and UDP is much better than for Topology 1. This time the multipath acts in favour of the fairness between sessions. The long session (no. 1), which would normally achieve the lowest throughput among all sessions, has many paths to exploit. That gives the long session chances comparable to the short sessions to achieve high proportional fairness.

Figure 7.15 shows packet loss as a percentage of the transmitted packets. mSNUM losses are much smaller than SNUM because it exploits different paths for backup or concurrent transmissions. As expected, UDP floods the network with data packets

and less than 25% of the packets get delivered.

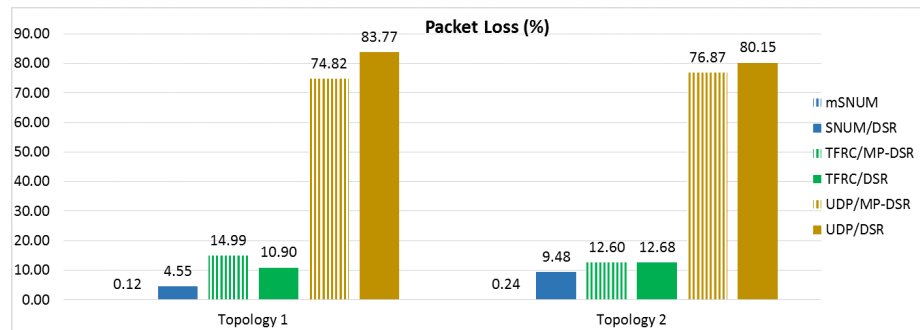


Figure 7.15: Percentage of Packet Losses.

The overhead of the different methods is shown in Fig. 7.16, mSNUM's overhead is higher than SNUM because of the coordination of different paths. Even if mSNUM uses only one path, the source node still needs to be updated about the status of the other potential (or backup) paths available, which means that mSNUM is activated in much more network nodes compared to SNUM. The overhead of TFRC/MP-DSR is higher than TFRC/DSR, the reason for that is the TFRC feedback report update rate that depends on the RTT and the packet losses. The multipath available for each TFRC session shortens its average RTT, which tends to increase the feedback message rates and the overall overhead.

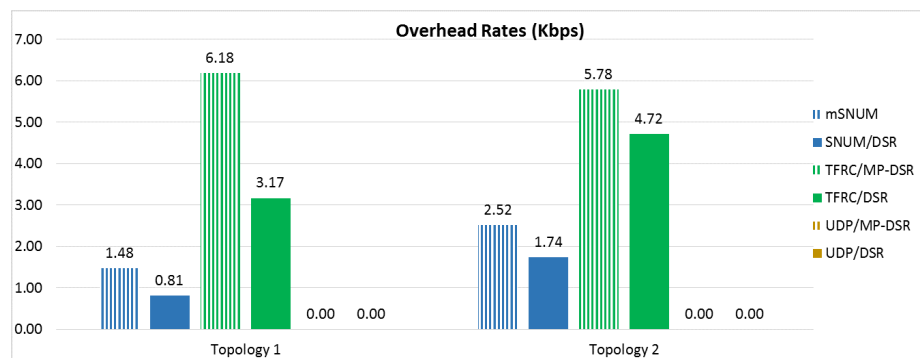


Figure 7.16: Overhead Rates.

The aggressiveness parameter Z has a different effect on mSNUM as compared to SNUM. The values shown in Figs. 7.17 and 7.18 are an average of 5 simulation results with different seed numbers for Topology 2. As shown in Fig. 7.17, the delay increases with increasing Z but the throughput does not monotonically decrease as the case of

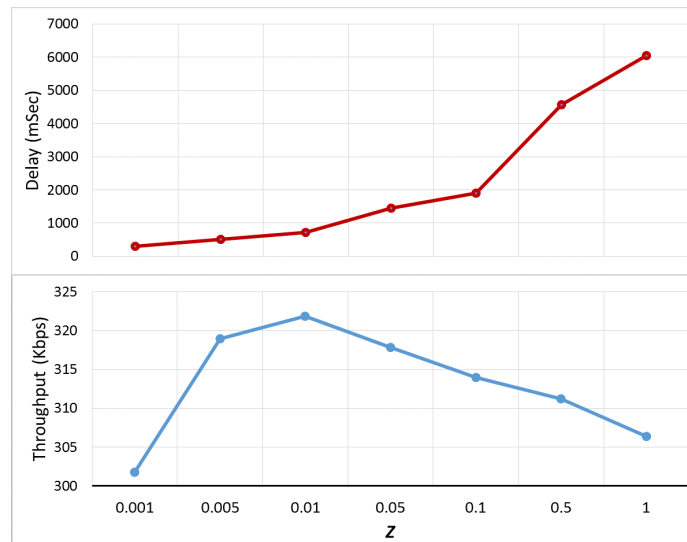


Figure 7.17: The Effect of Aggressiveness Parameter Z on mSNUM Network's Average Delay and Throughput.

SNUM shown in Fig. 7.6. mSNUM uses the QF to provide a feedback to the source about potential congestion and the inaccurate estimate of the available capacity on a certain path. Increasing the value Z after a certain point reduces the impact of the queue length on the path price, that reduces the ability of mSNUM to decide whether to keep the path among the active paths or not. The lack of decisiveness of mSNUM leads to keeping unnecessary paths active for a longer time even if the path would harm the overall network utility. Figure 7.18 shows the number of active paths used by mSNUM as the value Z increases. It shows that an unnecessarily high number of paths leads to spreading the session's data packets all over the network and creates more interference/congestion that is negatively affect the overall network utility.

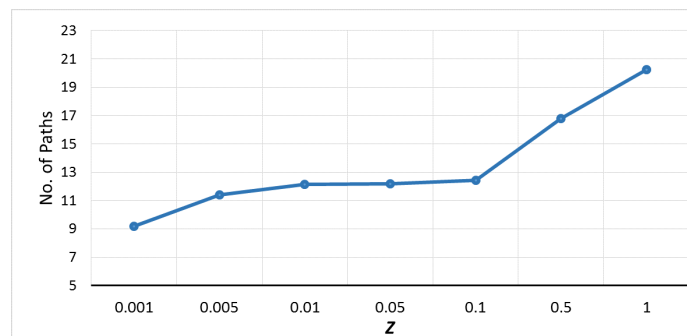


Figure 7.18: The Number of Paths used by mSNUM with different Z Values.

7.2.2 Mobile Networks

The node mobility is based on the Random Direction 2D mobility model, with 25 nodes in a rectangle shaped area of 150mX450m side lengths. We use ten sessions. We ran the simulations with 2 different node speeds, 2 and 5 m/sec with periodic pause intervals of 2 seconds every 4 seconds. Figure 7.19 shows that, even with mobility, which induces inaccurate topology information and update packet losses, the aggregate log rates for mSNUM outperform the rest of the methods but with wider 95% confidence. The packet losses of mSNUM and SNUM, shown in Fig. 7.20, are higher than static networks but less than 10% and much smaller than TFRC's packet loss. The overhead rate, shown in Fig. 7.21, increases in general with mobility, changing topologies and high packet losses increases the need for coordination, for mSNUM, SNUM and TFRC.

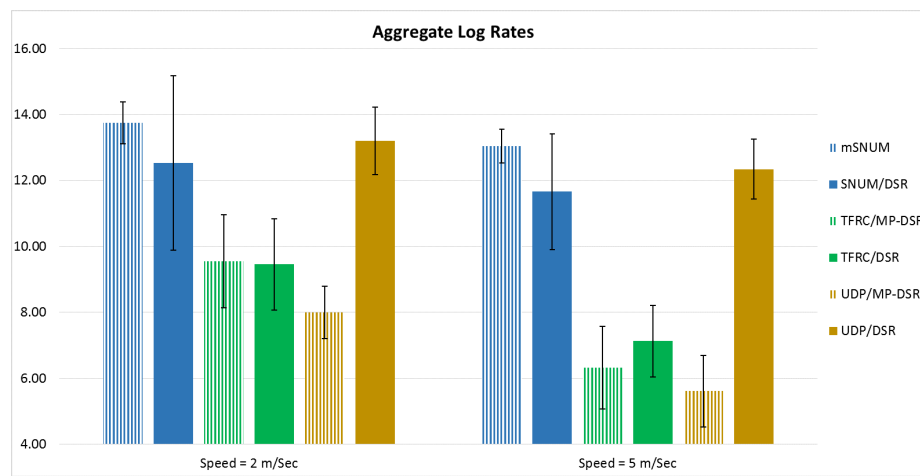


Figure 7.19: The Log Rates for two Topologies with the 95% Confidence Intervals.

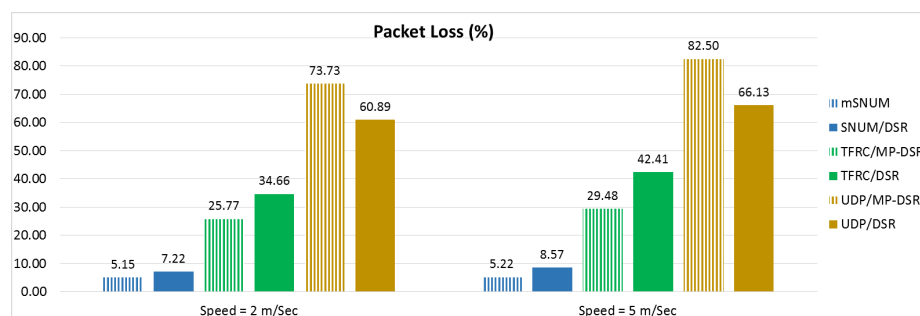


Figure 7.20: Percentage of Packet Losses.

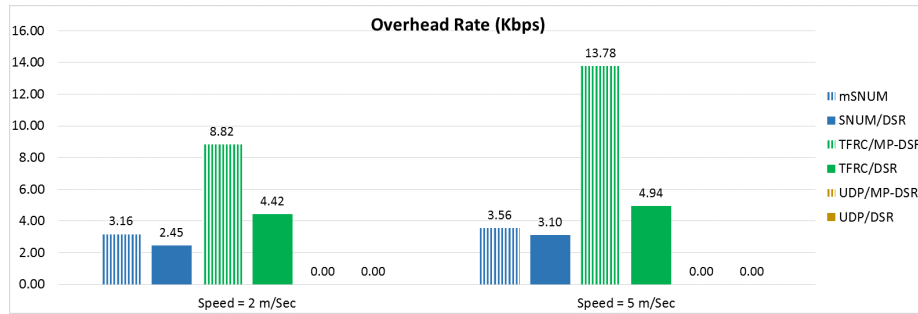


Figure 7.21: Overhead Rates.

7.3 Summary

The proposed localized queue management mechanism used in SNUM can guarantee a stable network with bounded queue length and delay without increased overhead or additional control signaling. The NS-3 simulations show that SNUM can coordinate between network layers and network nodes to provide very high fairness between end-to-end flows, surpassing 0.99. It also increases the total throughput up to 45% while reducing the network average delay at least 3.3 times with less than 0.4% packet loss, compared to TFRC. SNUM provides a stable performance even with very high rate of update message losses (more than 80% losses). Our algorithms can be used in networks with reliability requirements given that a reliable transmission protocol is provided on top of either SNUM or mSNUM. Such a protocol will have to provide sequencing and retransmission techniques for the application layer.

Load balancing is desirable in multipath communication for better exploiting network resources. Leaving the load balancing function to be executed obliviously, without reflecting the congestion and contention facing each path, may not be the optimum solution. The two or more disjoint paths used may have different lengths, interference, existing traffic, and congestion. It is therefore better to assign each path of the multipath routes with the amount of traffic considering the different condition of each path. That requires an algorithm capable of including the congestion and contention of the network in its rate assignment decisions.

mSNUM tests all the paths available and selects the one that has the minimum contention and congestion effect and provides higher throughput (i.e. results in higher network utility). On the other hand, SNUM uses the paths provided by the oblivious single path routing protocol, which could be the shortest path but not the best one

to minimize congestion and contention. mSNUM can provide network utility gain even without simultaneously activating more than a single path to the destination. The mobility tests of both SNUM and mSNUM show the ability of the CLDs to withstand inaccurate topology information and update packets loss. The mobility test also shows that mSNUM can provide multipath connections to the destination that gives more connection stability compared to SNUM.

The capacity estimate plays an important role in the network resources assignment. SNUM uses an optimistic capacity estimate (high value) to keep the network saturated in order to reach the maximum achievable utilization with a single path. mSNUM provides multipath capabilities and assigns more resources for further utilization of the network. mSNUM is sensitive towards the path prices. The queue buildups, which result from inaccurate capacity estimates, and the QF parameters disturb the calculations of mSNUM, which could lead to less than optimal results.

Chapter 8

Conclusions and Future Work

8.1 Conclusion

We focus on the problem of cross-layer network utility maximization considering the MAC, Network, and Transport layers. We started by examining and simulating different network utility maximization algorithms that optimize the medium access probabilities using the slotted ALOHA protocol at the MAC layer jointly with the end-to-end source rates at the transport layer. Among the studied optimization approaches, we found that the dual-based approach presented in [11] is the one with the fastest convergence speed. It also has the potential to withstand realistic network environment conditions. The potential lies in that fact that the algorithm requires a minimum amount of information to be exchanged between the nodes with only one update function.

We proposed our CLD algorithm, called SNUM, that jointly optimizes the congestion and the contention problem employing the CSMA-CA MAC protocol. The model is employ Bianchi's IEEE 802.11 network formulations to estimate capacities. The CLD is equipped with an active queue management mechanism for queue-bounded networks. An adaptive step-size mechanism is proposed to speed up SNUM's convergence and to avoid instability.

The shared wireless links raise a contention problem. Multi-hop transmissions cause flows not only to interfere with each other but also with themselves. This led us to question the potential of multipath routing in general wireless networks. We presented a study on the multipath achievable throughput using SNUM under idealistic assumptions where we consider an isolated session with mutual interference-free multipath routes. We found that multipath routing is not always beneficial for the

throughput unless the paths are carefully chosen. Multipath routing enhances the throughput if the bottleneck of the session is at the relaying nodes of the paths. In the case of a multi-session network, where sessions may affect each other's throughput, multipath routing may not always be the solution. If such external (to the session) interference affects the relaying nodes, multipath routing could alleviate the bottleneck and enhance the throughput. Otherwise, if the external interference is at the source and/or the destination, multipath routing will not be able to enhance the network performance.

The multipath study showed the difficulty of achieving maximum network utilization with oblivious multipath control, i.e. quantity of paths to use, impact of each path on network congestion and contention, etc. A cross-layer network utility maximization algorithm can assign path rates according to the benefit and fairness of all other sessions in the network. We needed an algorithm that is able to choose the multipath routes considering the interference and the congestion in the entire network. Different multipath NUM algorithms have been reviewed and different methods to overcome the non-convexity and linearity problems associated with the multipath formulations have been discussed. We found that the proposed solutions in the literature are either complex, approximated, sub-optimal and/or require centralized administration.

We formulate a joint congestion, multipath routing, and contention control problem based on NUM. By introducing a splitting factor to replace the linearity in the relation between each session and its multipath routes, we extend earlier models to include routing over potentially multiple paths in the optimization framework. The new variable, proper transformation, and using Ohm's law analogy lead to a convex and decoupled optimization algorithm that can find the optimum solution in a distributed fashion. The distributed algorithm, called mSNUM, requires a single coordination parameter (prices), which reflects the lack/surplus of resources in each link and each session-path. The prices are used as a metric for routing decisions by the session sources. Our proposed algorithm can jointly solve the congestion, multipath routing, and contention wireless multihop distributed optimization problems using a single relaxed constraint, i.e. using a single coordination parameter and a single loop for optimality convergence. The benefits of using a single coordination parameter are to reduce the amount of communication overhead between nodes. It also helps to maintain the boundaries between network protocol stack layers.

We discussed the proposed framework and the use of the prices to connect the different network protocol stack layers, specifically the MAC, Network, and Transport layers. The vertical coordination between the different layers is explained along with the functions executed at each layer. The horizontal coordination implementation is described using two methods, individual update messages and piggybacking onto other packets. Finally, we discussed the mechanism used to identify and inform the session sources about broken links and paths.

We implemented SNUM and mSNUM in NS-3. The simulations show that SNUM and the localized queue management mechanism can coordinate between network layers and network nodes to provide very high fairness between end-to-end flows, surpassing 0.99 based on Jain's fairness index. It also increases the total throughput up to 45% while reducing the network average delay at least 3.3 times with fewer than 0.4% packet lost, compared to TFRC. SNUM also shows the ability to reach optimized stable sessions rates much faster than TFRC. A more stable/predictable rate is suitable for adaptive applications such as video streaming. Unstable rates would require long queues to enhance the performance, however, the queues would not help delay intolerant applications. SNUM provides a stable performance even with a very high rate of update message losses (more than 80%).

mSNUM tests all available paths and selects the ones that have minimum contention and congestion effect and provide higher throughput (i.e. results in higher network utility). It can provide network utility gain over SNUM even without simultaneously activating more than a single path to the destination. The mobility tests of both SNUM and mSNUM show the ability of the algorithms to withstand inaccurate topology information, and update packets loss.

8.2 Future Work

8.2.1 Better Capacity Estimate

The capacity estimate plays a key role in the network resources allocation and utility optimization. SNUM uses an optimistic capacity estimate (high value) to keep the network saturated in order to reach the maximum achievable utilization with a single path. mSNUM provides multipath capabilities for sessions and assigns more resources for further increased utilization of the network. mSNUM is sensitive towards the paths prices. The queue buildups, which are a result of inaccurate capacity

estimates, and the QF aggressiveness parameters disturb the calculations of mSNUM, which could lead to less than optimal results. Managing multipath rates and fairness between sessions requires an accurate capacity estimate for the network. For that reason, a better capacity estimate, other than Bianchi's model, is required.

Bianchi's model is simple and suitable for nodes with low processing powers. It can provide a single number to estimate the channel capacity in the vicinity of each node. The simplicity of Bianchi's model is a result of some simplifying assumptions such as considering the network is saturated all the time, i.e. every node has a packet to transmit at any time. Bianchi also assumed constant packet collision probability for each node regardless of its state. Duffy *et al.* [65] extended Bianchi's model and provided a more accurate model that relaxes the restriction of the saturation state. The extended model is more realistic but more complicated. It needs interchanging more information between the nodes and requires complicated calculations.

8.2.2 Contention-Free MAC Protocols

The MAC protocols used so far are contention-based, it can be generalized to include contention-free protocols, i.e. Time Division Multiple Access (TDMA). In this case our goal will be to increase the active time of each link and decrease the scheduling period in order to increase throughput. The scheduling period contains all the active time slots allocated to each link/node in the same contention area and is repeated periodically. The throughput increase goal should be achieved considering the adaptability to topology changes and fairness [66]. That is, the scheduling scheme should allow a new set of communicating nodes to be assigned a fair share in the scheduling period along with the existing nodes. So the link prices will be updated to reflect this goal to achieve fairness while optimizing network utility. Although this is an NP-hard problem, various approximation schemes exist which can be employed [67].

8.2.3 Gateway Selection Problem and Network Lifetime

In the multiple gateways selection problem, if the number of wireless nodes is much greater than the number of gateways, there would be bottlenecks around the gateways. As shown in Fig. 8.1, the number of wireless nodes in the proximity of Gateway 1 is greater than those of Gateway 2. Selecting the gateway according to

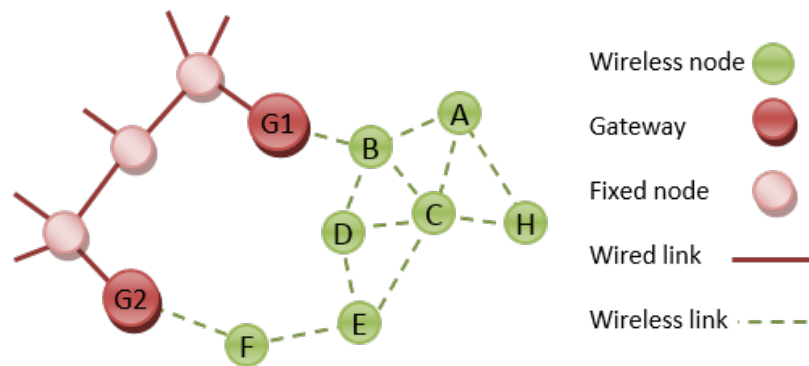


Figure 8.1: Wireless Network Connected to a Fixed Network through Gateways.

the minimum-hop policy will result in traffic congestion at Gateway 1; furthermore, the power of node B will be consumed at a rate faster than the other nodes, which shortens the network lifetime. mSNUM supports multipath for each node in the network to reach both gateways, the rates are split over the paths based on the congestion and contention facing each path. As a result, we could find for example nodes C and D forward most or all of their packets to Gateway 2 even if they can reach Gateway 1 in fewer hops.

The way to select the appropriate path can also be adapted to avoid certain nodes that are running out of energy. The same idea applied to incorporating the queue length in the path prices can be applied on the remaining energy left in the node. This way, the nodes running out of energy will have higher prices that will increase the price of any path relayed through them, which helps the source nodes in making routing decisions to avoid energy depleted nodes. For example, if the energy of node B in Fig. 8.1 has been drained due to a heavy relaying load, its price will be adjusted to reflect its low energy level. As a result, nodes A, C, and D may avoid the paths that use node B and forwarded their data packets toward Gateway 2.

8.2.4 Flexible CLD Framework

Prices are the key feature to connect several parameters from different network layers in the optimization process. In order to manage the performance at different layers without alleviating the boundaries between them, we use prices to coordinate and let each layer decide what is best. Every drawback in the network has a cost and every advantage has a price, based on this principle we can translate the different

objective functions of the optimization problem in different layers into a single reward function that helps each layer to take its own decisions.

With the new telecommunication technologies and Internet of Things (IoT), which are expected to bring a wide range of applications with reliability and high data rates, a flexible control is required. The proposed price-dependant CLD can provide a flexible framework to change the optimization objectives when and wherever it is required in order to adapt to various optimization goals. Based on the application, the CLD optimization framework can prioritize its objectives. For example, maximizing network life time can have a higher priority than fairness, some sessions could be given more advantages over others (Quality of Service), and/or minimizing delay may be more important than energy consumption. The optimization goals could be changed according to applications, users, available resources, time and/or location of the network.

List of References

- [1] “Project loon.” <https://x.company/loon/>. Accessed: 2017-06-30.
- [2] Z. Wang and J. Crowcroft, “Analysis of shortest-path routing algorithms in a dynamic network environment,” *SIGCOMM Comput. Commun. Rev.*, vol. 22, pp. 63–71, Apr. 1992.
- [3] R. Braden, T. Faber, and M. Handley, “From protocol stack to protocol heap: Role-based architecture,” *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 17–22, Jan. 2003.
- [4] M. Conti, G. Maselli, G. Turi, and S. Giordano, “Cross-layering in mobile ad hoc network design,” *Computer*, vol. 37, pp. 48–51, Feb 2004.
- [5] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer, “Application-driven cross-layer optimization for video streaming over wireless networks,” *IEEE Communications Magazine*, vol. 44, pp. 122–130, Jan 2006.
- [6] G. A. Shah, W. Liang, and O. B. Akan, “Cross-layer framework for qos support in wireless multimedia sensor networks,” *IEEE Transactions on Multimedia*, vol. 14, pp. 1442–1455, Oct 2012.
- [7] T. Melodia and I. F. Akyildiz, “Cross-layer qos-aware communication for ultra wide band wireless multimedia sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 653–663, June 2010.
- [8] Q. Liu, S. Zhou, and G. B. Giannakis, “Cross-layer combining of adaptive modulation and coding with truncated arq over wireless links,” *IEEE Transactions on Wireless Communications*, vol. 3, pp. 1746–1755, Sept 2004.
- [9] Q. V. Pham and W. J. Hwang, “Network utility maximization-based congestion control over wireless networks: A survey and potential directives,” *IEEE Communications Surveys Tutorials*, vol. 19, pp. 1173–1200, Secondquarter 2017.
- [10] M. Mardani, S. J. Kim, and G. B. Giannakis, “Cross-layer design of wireless multihop random access networks,” *IEEE Transactions on Signal Processing*, vol. 60, pp. 2562–2574, May 2012.
- [11] Y. Yu and G. B. Giannakis, “Cross-layer congestion and contention control for wireless ad hoc networks,” *IEEE Transactions on Wireless Communications*, vol. 7, pp. 37–42, Jan 2008.

-
- [12] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 535–547, March 2000.
- [13] Q.-V. Pham and W.-J. Hwang, "Network utility maximization in multipath lossy wireless networks," *International Journal of Communication Systems*, vol. 30, no. 5, pp. e3094–n/a, 2017. e3094 dac.3094.
- [14] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.
- [15] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proceedings of the IEEE*, vol. 95, pp. 255–312, Jan 2007.
- [16] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, pp. 556–567, Oct. 2000.
- [17] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 1439–1451, 2006.
- [18] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. A. Commun.*, vol. 24, pp. 1452–1463, Aug. 2006.
- [19] A. Ribeiro and G. B. Giannakis, "Separation principles in wireless networking," *IEEE Transactions on Information Theory*, vol. 56, pp. 4488–4505, Sept 2010.
- [20] F. Wang, X. Liao, S. Guo, and H. Huang, "Optimal rate and power allocation under quality-of-service requirements for wireless multihop networks," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 2343–2365, 2014.
- [21] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3, pp. 1804–1814 vol. 3, March 2005.
- [22] X. Wang and K. Kar, "Cross-layer rate optimization for proportional fairness in multihop wireless networks with random access," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 1548–1559, Aug 2006.
- [23] J. W. Lee, M. Chiang, and A. R. Calderbank, "Utility-optimal random-access control," *IEEE Transactions on Wireless Communications*, vol. 6, pp. 2741–2751, July 2007.
- [24] J.-W. Lee, M. Chiang, and R. A. Calderbank, "Jointly optimal congestion and contention control based on network utility maximization," *IEEE Communications Letters*, vol. 10, pp. 216–218, Mar 2006.
- [25] X. Wang, *Distributed Rate Optimization in Ad-hoc Networks with Random Access*. PhD dissertation, Faculty of Rensselaer Polytechnic Institute, 2007.

-
- [26] M. H. Cheung, A. h. Mohsenian-Rad, V. W. S. Wong, and R. Schober, "Random access for elastic and inelastic traffic in w lans," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 1861–1866, June 2010.
- [27] J. J. Jaramillo and R. Srikant, "Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic," *IEEE/ACM Transactions on Networking*, vol. 19, pp. 1125–1136, Aug 2011.
- [28] F. Qiu and Y. Xue, "Robust joint congestion control and scheduling for time-varying multi-hop wireless networks with feedback delay," *IEEE Transactions on Wireless Communications*, vol. 13, pp. 5211–5222, Sept 2014.
- [29] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http," in *Proceedings of the second annual ACM conference on Multimedia systems*, pp. 157–168, ACM, 2011.
- [30] K. Kar, S. Sarkar, and L. Tassiulas, "Achieving proportional fairness using local information in aloha networks," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1858–1863, Oct 2004.
- [31] F. Fu and M. van der Schaar, "A new systematic framework for autonomous cross-layer optimization," *IEEE Transactions on Vehicular Technology*, vol. 58, pp. 1887–1903, May 2009.
- [32] M. Saad, A. Leon-Garcia, and W. Yu, "Optimal network rate allocation under end-to-end quality-of-service requirements," *IEEE Transactions on Network and Service Management*, vol. 4, pp. 40–49, Dec 2007.
- [33] H. Susanto and B. G. Kim, "Congestion control with qos and delays utility function," in *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–5, July 2013.
- [34] F. Qiu, J. Bai, and Y. Xue, "Optimal rate allocation in wireless networks with delay constraints," *Ad Hoc Networks*, vol. 13, Part B, pp. 282 – 295, 2014.
- [35] S. Jahromizadeh and V. Rakocevic, "Joint rate control and scheduling for providing bounded delay with high efficiency in multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 22, pp. 1686–1698, Oct 2014.
- [36] J. Doma, Z. Duliski, M. Kantor, J. Rzsza, R. Stankiewicz, K. Wajda, and R. Wj-cik, "A survey on methods to provide multipath transmission in wired packet networks," *Computer Networks*, vol. 77, pp. 18 – 41, 2015.
- [37] J. Jin, W.-H. Wang, and M. Palaniswami, "Utility max-min fair resource allocation for communication networks with multipath routing," *Comput. Commun.*, vol. 32, pp. 1802–1809, Nov. 2009.
- [38] J. Zhang and H.-N. Lee, "Energy-efficient utility maximization for wireless networks with/without multipath routing," *AEU-International Journal of Electronics and Communications*, vol. 64, no. 2, pp. 99–111, 2010.

-
- [39] X. Lin and N. B. Shroff, "Utility maximization for communication networks with multipath routing," *IEEE Transactions on Automatic Control*, vol. 51, pp. 766–781, May 2006.
- [40] S. Supittayapornpong and P. Saengudomlert, "Joint flow control, routing and medium access control in random access multi-hop wireless networks," in *2009 IEEE International Conference on Communications*, pp. 1–6, June 2009.
- [41] L. F. Wilson and M. J. Gonzalez, "Synchronization and communication in algorithmic structures," in *Proceedings of 1994 6th IEEE Symposium on Parallel and Distributed Processing*, pp. 196–203, Oct 1994.
- [42] E. Natsheh, A. Jantan, S. Khatun, and S. Subramaniam, "Adaptive optimizing of hello messages in wireless ad-hoc networks," *The International Arab Journal of Information Technology*, vol. 4, July 2007.
- [43] M. Heissenbttel and T. Braun, "Optimizing neighbor table accuracy of position-based routing algorithms," in *Proceedings of IEEE INFOCOM 2005*, March 2005.
- [44] Y. Jia, I. Nikolaidis, and P. Gburzynski, "Alternative paths vs. inaccurate link state information in realistic network topologies," in *Proceedings of SPECTS 2002*, July 2002.
- [45] R. A. Alsaqour, M. S. Abdelhaq, and O. A. Alsukour, "Effect of network parameters on neighbor wireless link breaks in gpsr protocol and enhancement using mobility prediction model," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 171, 2012.
- [46] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Review*, vol. 53, no. 4, pp. 747–772, 2011.
- [47] C. Shao, H. Deng, R. Pazhyannur, F. Bari, R. Zhang, and S. Matsushima, "Ieee 802.11 medium access control (mac) profile for control and provisioning of wireless access points (capwap)," RFC 7494, RFC Editor, April 2015.
- [48] A. Ksentini, A. Nafaa, A. Gueroui, and M. Naimi, "Determinist contention window algorithm for ieee 802.11," in *2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 4, pp. 2712–2716 Vol. 4, Sept 2005.
- [49] Z. Li, S. Nandi, and A. K. Gupta, "Achieving mac fairness in wireless ad-hoc networks using adaptive transmission control," in *Proceedings. ISCC 2004. Ninth International Symposium on Computers And Communications (IEEE Cat. No.04TH8769)*, vol. 1, pp. 176–181 Vol.1, June 2004.
- [50] D. P. Bertsekas, *Convex optimization Theory*. Athena Scientific Belmont, 2009.
- [51] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 785–799, Dec 2000.

-
- [52] J. Lee, H. Lee, Y. Yi, S. Chong, E. W. Knightly, and M. Chiang, "Making 802.11 dcf near-optimal: Design, implementation, and evaluation," *IEEE/ACM Transactions on Networking*, vol. 24, pp. 1745–1758, June 2016.
- [53] F. Qiu, J. Bai, and Y. Xue, "Optimal rate allocation in wireless networks with delay constraints," *Ad Hoc Networks*, vol. 13, Part B, pp. 282 – 295, 2014.
- [54] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pp. 66–80, 2003.
- [55] J.-W. Lee, M. Chiang, and R. Calderbank, "Optimal mac design based on utility maximization: Reverse and forward engineering," in *2006 Proceedings IEEE INFOCOM*, pp. 1–13, April 2006.
- [56] S. Guo, C. Dang, and X. Liao, "Distributed algorithms for resource allocation of physical and transport layers in wireless cognitive ad hoc networks," *Wireless Networks*, vol. 17, no. 2, pp. 337–356, 2011.
- [57] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.
- [58] N. Z. Shor, K. C. Kiwiel, and A. Ruszcayński, *Minimization Methods for Non-differentiable Functions*. New York, NY, USA: Springer-Verlag New York, Inc., 1985.
- [59] D. Johnson, Y. Hu, and D. Maltz, "The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4," RFC 4728, RFC Editor, February 2007. <http://www.rfc-editor.org/rfc/rfc4728.txt>.
- [60] J. Postel, "User datagram protocol," STD 6, RFC Editor, August 1980. <http://www.rfc-editor.org/rfc/rfc768.txt>.
- [61] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Tcp friendly rate control (tfr): Protocol specification," RFC 5348, RFC Editor, September 2008. <http://www.rfc-editor.org/rfc/rfc5348.txt>.
- [62] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," RFC 3626, RFC Editor, October 2003. <http://www.rfc-editor.org/rfc/rfc3626.txt>.
- [63] S. Ahn, "Dsr extensions for multipath routing," Internet-Draft draft-ahn-manet-multipath-dsr-00, IETF Secretariat, December 2015. <http://www.ietf.org/internet-drafts/draft-ahn-manet-multipath-dsr-00.txt>.
- [64] S. J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, vol. 10, pp. 3201–3205 vol.10, 2001.
- [65] G. A. Shah, W. Liang, and O. B. Akan, "Cross-layer framework for qos support in wireless multimedia sensor networks," *IEEE Transactions on Multimedia*, vol. 14, pp. 1442–1455, Oct 2012.

- [66] B. Bruhadeshwar, K. Kothapalli, and I. R. Pulla, “A fully dynamic and self-stabilizing tdma scheme for wireless ad-hoc networks,” in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 511–518, April 2010.
- [67] M. Yao, C. Lin, P. Zhang, Y. Tian, and S. Xu, “Tdma scheduling with maximum throughput and fair rate allocation in wireless sensor networks,” in *2013 IEEE International Conference on Communications (ICC)*, pp. 1576–1581, June 2013.

Appendix A

KKT Optimality Proofs

In this Appendix we provide the method to find the closed form solutions for the Subproblems (5.15)-(5.17).

Solution of Subproblem (5.15)

Subproblem (5.15) is optimizing the allocated rate of session s over the path $m \in \mathcal{M}_s$ given the session-path prices $\lambda_{s,m}$.

The problem:

$$\max_{\substack{\sum_{m \in \mathcal{M}_s} 1/v_{s,m} = 1, \\ 1 \leq v_{s,m}}} \sum_{m \in \mathcal{M}_s} \lambda_{s,m} \log v_{s,m}$$

Proof. Using Lagrange maximization method with equality constraint we get,

$$L(v_{s,m}, \mu) = \sum_{m \in \mathcal{M}_s} \lambda_{s,m} \log v_{s,m} + \mu \left(\sum_{m \in \mathcal{M}_s} 1/v_{s,m} - 1 \right) \quad (\text{A.1})$$

Applying the KKT conditions on (A.1) we get the following,

$$\frac{\partial L(v_{s,m})}{\partial v_{s,m}} = \frac{\lambda_{s,m}}{v_{s,m}} + \mu = 0 \implies v_{s,m} = -\lambda_{s,m}/\mu \quad (\text{A.2})$$

$$\frac{\partial L(\mu)}{\partial \mu} = \sum_{m \in \mathcal{M}_s} 1/v_{s,m} - 1 = 0 \quad (\text{A.3})$$

From (A.2) and (A.3) we get,

$$\begin{aligned}\sum_{m \in \mathcal{M}_s} \frac{-\mu}{\lambda_{s,m}} &= 1 \\ \sum_{m \in \mathcal{M}_s} \frac{1}{\lambda_{s,m}} &= 1 - /\mu\end{aligned}$$

Substituting the result in (A.2), we get,

$$v_{s,m}(\boldsymbol{\lambda}) = \frac{\sum_{\hat{m} \in \mathcal{M}_s} 1/\lambda_{s,\hat{m}}}{1/\lambda_{s,m}} \quad (\text{A.4})$$

If $1/\lambda_{s,m} = 0$, then $v_{s,m}$ can be any feasible solution that satisfy the constraints of the maximization subproblem, and the closed form solution becomes,

$$v_{s,m}(\boldsymbol{\lambda}) = \begin{cases} \frac{\sum_{\hat{m} \in \mathcal{M}_s} 1/\lambda_{s,\hat{m}}}{1/\lambda_{s,m}} & \text{if } \lambda_{s,\hat{m}} > 0, \forall \hat{m} \in \mathcal{M}_s, \\ |M_s| & \text{otherwise,} \end{cases}$$

□

Solution of Subproblem (5.16)

Subproblem (5.16) is optimizing the share of the link given to each session-path $s, m \in \mathcal{M}_\ell$ given the hops prices $\lambda_{s,m}^\ell$.

The problem:

$$\max_{\substack{\sum_{s,m \in \mathcal{M}_\ell} \alpha_{s,m}^\ell = 1, \\ 0 \leq \alpha_{s,m}^\ell \leq 1}} \sum_{s,m \in \mathcal{M}_\ell} \lambda_{s,m}^\ell \log \alpha_{s,m}^\ell,$$

Proof. Using Lagrange maximization method with equality constraint we get,

$$L(\alpha_{s,m}^\ell, \mu) = \sum_{s,m \in \mathcal{M}_\ell} \lambda_{s,m}^\ell \log \alpha_{s,m}^\ell + \mu \left(\sum_{s,m \in \mathcal{M}_\ell} \alpha_{s,m}^\ell - 1 \right) \quad (\text{A.5})$$

Applying the KKT conditions on (A.5) we get the following,

$$\frac{\partial L(\alpha_{s,m}^\ell)}{\partial \alpha_{s,m}^\ell} = \frac{\lambda_{s,m}^\ell}{\alpha_{s,m}^\ell} + \mu = 0 \implies \alpha_{s,m}^\ell = -\lambda_{s,m}^\ell / \mu \quad (\text{A.6})$$

$$\frac{\partial L(\mu)}{\partial \mu} = \sum_{s,m \in \mathcal{M}_\ell} \alpha_{s,m}^\ell - 1 = 0 \quad (\text{A.7})$$

From (A.2) and (A.7) we get,

$$\begin{aligned}\sum_{s,m \in \mathcal{M}_\ell} \frac{\lambda_{s,m}^\ell}{-\mu} &= 1 \\ \sum_{s,m \in \mathcal{M}_\ell} \lambda_{s,m}^\ell &= -\mu\end{aligned}$$

Substituting the result in (A.6), we get,

$$\alpha_{s,m}^\ell(\boldsymbol{\lambda}) = \frac{\lambda_{s,m}^\ell}{\sum_{\acute{s},\acute{m} \in \mathcal{M}_\ell} \lambda_{\acute{s},\acute{m}}^\ell} \quad (\text{A.8})$$

If $1/\lambda_{s,m}^\ell = 0$, then $\alpha_{s,m}^\ell$ can be any feasible solution that satisfy the constraints of the maximization subproblem, and the closed form solution becomes,

$$\alpha_{s,m}^\ell(\boldsymbol{\lambda}) = \begin{cases} \frac{\lambda_{s,m}^\ell}{\sum_{\acute{s},\acute{m} \in \mathcal{M}_\ell} \lambda_{\acute{s},\acute{m}}^\ell} & \text{if } \sum_{\acute{s},\acute{m} \in \mathcal{M}_\ell} \lambda_{\acute{s},\acute{m}}^\ell \neq 0, \\ \frac{1}{|\mathcal{M}_\ell|} & \text{if } \sum_{(s,u) \in \mathcal{M}_\ell} \lambda_{(s,u)} = 0, \end{cases} \quad (\text{A.9})$$

□

Solution of Subproblem (5.17)

Subproblem (5.17) is optimizing the persistence probabilities of each node n and each outgoing link $\ell \in \mathcal{L}_{out}(n)$.

The problem:

$$\max_{\substack{\sum_{\ell \in \mathcal{L}_{out}(n)} p_\ell = q_n \leq 1, \\ 0 \leq p_\ell \leq 1}} \sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell \log p_\ell + \sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell \log(1 - q_n) \quad (\text{A.10})$$

Proof. Using Lagrange maximization method with equality constraint we get,

$$L(p_\ell, q_n, \mu) = \sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell \log p_\ell + \sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell \log(1 - q_n) + \mu \left(q_n - \sum_{\ell \in \mathcal{L}_{out}(n)} p_\ell \right) \quad (\text{A.11})$$

Applying the KKT conditions on (A.11) we get the following,

$$\frac{\partial L(p_\ell)}{\partial p_\ell} = \frac{\lambda^\ell}{p_\ell} - \mu = 0 \quad \implies p_\ell = \frac{\lambda^\ell}{\mu} \quad (\text{A.12})$$

$$\frac{\partial L(q_n)}{\partial q_n} = \frac{-\sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell}{1 - q_n} + \mu = 0 \quad \implies \mu = \frac{-\sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell}{1 - q_n} \quad (\text{A.13})$$

$$\frac{\partial L(\mu)}{\partial \mu} = q_n - \sum_{\ell \in \mathcal{L}_{out}(n)} p_\ell = 0 \quad (\text{A.14})$$

Substituting (A.12) in (A.14) and using (A.13) we get,

$$(1 - q_n) \sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell = q_n \sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell \quad (\text{A.15})$$

From (A.15) we get q_n as follows,

$$q_n = \frac{\sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell}{\sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell + \sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell} \quad (\text{A.16})$$

Using (A.14) and (A.16), p_ℓ will be,

$$p_\ell = \frac{\lambda^\ell}{\sum_{\ell' \in \mathcal{L}_{out}(n)} \lambda^{\ell'} + \sum_{\ell' \in \mathcal{L}_I(n)} \lambda^{\ell'}} \quad (\text{A.17})$$

If $\sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell + \sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell = 0$, then p_ℓ and q_n can be any feasible solutions satisfy the constraints of the maximization subproblem, and the closed form solutions become,

$$p_\ell(\boldsymbol{\lambda}) = \begin{cases} \frac{\lambda^\ell}{\sum_{\ell' \in \mathcal{L}_{out}(n)} \lambda^{\ell'} + \sum_{\ell' \in \mathcal{L}_I(n)} \lambda^{\ell'}} & \text{if } k_n \neq 0, \\ \frac{1}{|\mathcal{L}_{out}(n)| + |\mathcal{L}_I(n)|} & \text{if } k_n = 0, \end{cases}$$

$$q_n(\boldsymbol{\lambda}) = \begin{cases} \frac{\sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell}{\sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell + \sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell} & \text{if } k_n \neq 0, \\ \frac{|\mathcal{L}_{out}(n)|}{|\mathcal{L}_{out}(n)| + |\mathcal{L}_I(n)|} & \text{if } k_n = 0, \end{cases}$$

where $k_n = \sum_{\ell \in \mathcal{L}_{out}(n)} \lambda^\ell + \sum_{\ell \in \mathcal{L}_I(n)} \lambda^\ell$. □