Traffic Engineering using Multipath Routing Approaches

Gaston KUZAMUNU MAZANDU

Thesis presented in partial fulfilment
of the requirements for the degree of
Master of Science
at the University of Stellenbosch

Dr. Antoine B. Bagula
December 2007

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: .... Date: ....

Abstract

It is widely recognized that Traffic engineering (TE) mechanisms have to be added to the IP transport functionalities to provide QoS guarantees while ensuring efficient use of network resources. Traffic engineering is a network management technique which routes traffic to where bandwidth is available in the network to achieve QoS agreements between current and future demands and the available network resources. Multi-path routing has been proven to be a more efficient TE mechanism than Shortest Path First (SPF) routing in terms of profit maximization and resource usage optimization. However the identification of set of paths over which traffic is forwarded from source to the destination and the distribution of traffic among these paths are two issues that have been widely addressed by the IP community but remain an open issue for the emerging generation IP networks.

Building upon different frameworks, this thesis revisits the issue of multi-path routing to present and evaluate the performance of different traffic splitting mechanisms to achieve QoS routing in Multi-Protocol Label Switching (MPLS) and Wireless Sensor Networks (WSNs). Three main contributions are identified in this thesis. First, we extend an optimization model that used the M/M/1 queueing model on a simple network consisting of a single source-destination pair by using the M/M/s queueing model on a general network consisting of several source-destination pairs. The model solves a multi-path routing problem by defining a Hamiltonian as a function of delay incurred and subjecting this Hamiltonian to Pontryagin's cost minimization to achieve efficient diffusion of traffic over the available parallel paths. Second, we revisit the problem of cost-based optimization in a multi-path setting by using a Game theoretical framework to propose and evaluate the performance of competitive and cooperative multi-path routing schemes and the impact of the routing metric (cost) on the difference between these two schemes. Finally, building upon a previously proposed optimization benchmark, we propose an Energy constrained QoS routing scheme for Wireless Sensor Networks and show through simulation that our scheme outperforms the benchmark scheme.

Opsomming

Dit word algemeen aanvaar dat verkeersontwerp (*traffic engineering*) meganismes by die Internetprotokol (IP) vervoerfunksionaliteite bygevoeg moet word sodat kwaliteitdiens (*Quality of Service*) sowel as die effektiewe gebruik van netwerkbronne gewaarborg kan word. Verkeersontwerp meganismes is netwerkbeheertegnieke wat verkeer roeteer na waar daar genoeg bandwydte beskikbaar is sodat die nodige kwaliteitdiens ooreenkomste tussen die huidige en toekomstige aanvraag en die beskikbare netwerkbronne vervul kan word. Multi-pad roetering is bewys om 'n meer effektiewe verkeersontwerp meganisme as kortste-pad-eerste (*Shortest Path First*) roetering te wees in terme van wins maksimering en bronge-bruik optimering. Nietemin, die identifisering van 'n versameling paaie waaroor verkeer van beginpunt tot indpunt gestuur kan word en die verdeling van verkeer tussen hierdie paaie, is twee kwelpunte wat wyd aandag geniet in die IP gemeenskap, maar dit bly 'n onopgelosde vraag vir die ontluikende generasie IP netwerke.

Hierdie tesis bou voort op verskillende raamwerke en hersien die uitvloeisel van multi-pad roetering om die werkeng van verskillende verkeersverdeling meganismes om kwaliteit-diens roetering in Multiprotokol Etiketwisseling (*Multi-Protocol Label Switching*) (MPLS) en Draadlose Sensor Netwerke (*Wireless Sensor Networks*) (WSNs) te verrig, voor te stel en te evalueer. Drie hoofbydraes word in die tesis geïdentifiseer. Eerstens, 'n optime-ringsmodel wat the M/M/1 toustaan model op 'n eenvoudige netwerk bestaande uit 'n enkele begin-indpunt paar gebruik, word uitgebrei na die M/M/s toustaan model op 'n algemene netwerk bestaande uit verskeie begin-indpunt pare. Die model los 'n multi-pad roeteringsprobleem op deur 'n Hamiltonaan as funksie van vertraging te definieer en hierdie funksie te onderwerp aan Pontryagin se koste minimering om effektiewe verdeling van verkeer oor die beskikbare parallelle paaie te bewerkstellig. Tweedens, hersien ons die probleem van koste-gebaseerde optimering in 'n multi-pad opstelling deur 'n Spel-teoretiese raamwerk te gebruik vir die voorstelling en evalueering van die werking van kompeterende en meewerkende multi-pad roeteringskemas, en die impak van die roetering koste-maatstaf op die verskil tussen hierdie twee skemas te meet. Laastens, deur voort te bou op 'n voorheen voorgestelde optimering maatstaf (*benchmark*), stel ons 'n Energie beperkde kwaliteit-diens roeteringskema vir Draadlose Sensor Netwerke voor en wys deur simulasie dat ons skema die maatstafskema oortref.

Acknowledgements

This thesis is in partial fulfilment of the requirements for the degree of Master of Science in Computer Science at the department of Mathematical Sciences, University of Stellenbosch. Without the help from many people, this work would not have been possible. As such, this thesis would not be possible without the financial support of the University of Stellenbosch, and the Telkom-Siemens Centre of Excellence (COE) in ATM and Broadband Networks and their Applications.

I would like to express my deep recognition to my thesis adviser Dr. Antoine Bagula. He has supported me through the difficult process of defining the area of research and later bringing those ideas to actual contributions. He has had an infinite patience to discuss with me on and on, even when my ideas weren't clear sometimes. For his unlimited help and dedication through the whole process, for helping me beyond his duties at a human level, my sincere thanks.

A special thanks to Prof. Antony Krzesinski the co-ordinator of the Stellenbosch University Unit of the Telkom-Siemens COE for his invaluable support throughout this process, from its beginning to its end. Thanks to Prof. Brink van der Merwe, the head of Computer Science division for admitting me to this division and specially for his willingness to help.

I must also thank Yvette Bagula for her great encouragement. I would like to thank all my colleagues of the COE group at the University of Stellenbosh for their advices.

In all the cases, the discussion about technical matters within and outside have contributed to the improvement of this thesis. Anyone who has passed by the research domain knows there is more to it than business.

May all those who have contributed by far or of close to the realization of this dissertation, find through this line the mark of our gratitude.

To my parents for their support through time and distance. I am proud of bringing them this accomplishment.

But mostly I owe all this to my family. Malungidi Mbambi Marie-Paul my wife, Kuzamunu Jemima Mambote, Kuzamunu Glodi Mazandu and Kuzamunu Keren Kinzuemi our three children, whose life is almost sacrifice at every step of the long way of this fighting. This increased my determination to keep fighting until I won a few. This dissertation is dedicated to them.

List of Publications

K.G. Mazandu, A. B. Bagula and A. Muchanga, *On Using Differentiated Queuing to Support Traffic Engineering*, In Proceedings Southern African Telecommunication Networks and Applications Conference (SATNAC), Sept. 2006.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The recent years have experienced an exponential growth of the Internet popularity resulting from the increasing demand to carry a mixture of applications with different QoS requirements. This requires a good approach of network and traffic management and enhanced traffic and network control mechanisms such as traffic engineering (TE) and network engineering (NE) in order to meet the quality of service (QoS) demanded and users expectations. Such a quality of service (QoS) can not be provided by the traditional IP networks which are based on single path routing algorithms for path computation, *i.e.*, the Shortest Path First (SPF) routing with link cost inversely proportional to its capacity as proposed by Cisco for the Open Shortest Path First (OSPF) protocol [1]. The Shortest Path First (SPF) routing is widely implemented to achieve IP routing by using IGP protocols such as (OSPF) or Intermediate System to Intermediate System (IS-IS) [2]. These two protocols are the most widely used link state protocols for intra-domain routing. However, SPF routing does not often optimally use network resources and does not care about network utilization aspects. Indeed, the traffic is forwarded over the shortest path even if the shortest path is overloaded while there still exist unused alternative paths in the network. This can yield an unbalanced network configuration over-utilizing some links while leaving other under-utilized. Thus, QoS requirements for certain real-time applications such as streaming media (Video or Audio) which consider multiple metrics, such as bandwidth and delay, can not be guaranteed by traditional SPF routing. Such QoS requirements can not be guaranteed even by an enhancement to SPF, Constrained Shortest Path First (CSPF), which uses the link cost that takes into account the available bandwidth.

As an illustration, we consider a network with 7 routers labeled from 1 to 7 interconnected by 14 links described by figure 1.1. Each link has its link state determined by its link bandwidth in unit of bandwidth referred to as ub.

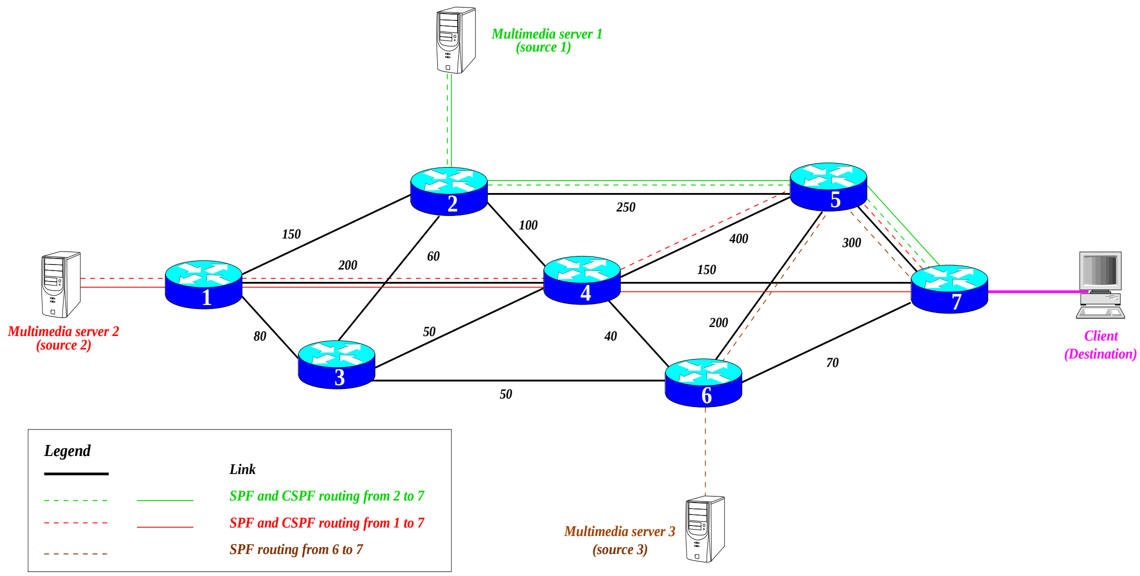


Figure 1.1: *SPF and CSPF Routing Inefficiency*

Suppose that video streams with bandwidth requirements of 150 ub, 50 ub and 160 ub are to be respectively transmitted from the server nodes 2, 1, 6 to the client node 7. SPF routing with link cost set inversely proportional to the link capacity may lead to a congested network overloading the link $5-7$. CSPF routing can not satisfy the end-to-end bandwidth requirement of 160 ub from node 6 to node 7 if routed after the requirements $2-7$ and $1-7$.

In order to overcome certain limitations of traditional IP routing, Multi-Protocol Label Switching (MPLS) using source or flow-based routing was proposed to engineer the routing of traffic in a network. MPLS provides flexibility in the control of traffic [3] and achieves IP routing at switching speed by avoiding the longest prefix-match of traditional IP networks [4]. MPLS is based on a routing model borrowed from ATM virtual connection paradigm where the traffic is routed over bandwidth-guaranteed tunnels referred to as Label Switched Paths (LSPs) [5]. In fact, MPLS extends the IPv4 routing protocols to provide new and scalable routing capabilities. These include Traffic Engineering (TE) and Virtual Private Networks (VPNs) support. From a network administration point of view, MPLS provides more scalable network operation by off-loading the network administrator from the task of monitoring the state of the network and executing routing and compensation when problems arise [6].

Recent advances in wireless sensor networks (WSNs) have led to several new routing protocols that minimize the energy consumption, considered as key to low power wireless sensor networks. When deployed in WSNs, Single Path Routing may be simple and minimize energy consumption and consequently increase the lifetime of the network. However, the wireless topology is highly dynamic, resulting in frequent changes to the route, high path loss and channel fading. Moreover, depending on the importance of the data inside the packet, the packet can be required to reach the base station (sink node) with a certain desired level of reliability. This may not be guaranteed by single path routing.

1.1 Traffic Management and Main Issues

The aim of engineering a network is to use its underlying infrastructure to satisfy the needs of users and applications. Broadly speaking, there are two opposite in their way of operating but complementary traffic management techniques, namely traffic engineering (TE) and network engineering (NE). NE moves bandwidth to where the traffic is offered to the network while TE moves the offered traffic to where bandwidth is available in the network to achieve QoS agreement between the available resources and the application requirements [7]. The main objective of these traffic and network management techniques is to improve the performance of the network from a perspective of reliability, efficient use of the network resources and planning of network capacity. Traffic engineering (TE) and network engineering (NE) are today widely deployed in telecommunication networks with the use of MPLS engineering in the Internet backbone. This deployment is in part owed to the need for bandwidth optimization, fast recovery and strict QoS guarantees to carry sensitive traffic over networks. However, the distribution of traffic over a network remains a challenging traffic engineering problem in MPLS [8].

In packet switched lossy networks, such as the Internet, ad hoc, and diversity wireless networks, packet losses and delays are realities which must be taken into account when seeking to offer higher user satisfaction. For packet losses, the retransmission of the lost packet is conventionally the simplest strategy to deal with. However, the retransmission does not guarantee that the packet lost will be successfully received at the destination, which means that the delay may become arbitrary large. For real-time services such as streaming media (video or audio), where the transport delay variation determine the buffer size required and buffering delay, the retransmission strategy can yield priceless consequences. One of the most interesting and attractive solution proposed, especially for streaming media, is the multiple description coding (MDC) where a multiple description (MD) coder partitions the source data into several sets and then compresses them independently to produce descriptions, and any of these descriptions can be used to approximate the source data [9]. Unfortunately, its efficiency widely depends on the number of MDC successfully forwarded to the destination [10].

This reveals that TE and NE are tasks that must be carried out carefully to avoid the performance degradation that could occur from an inefficient path identification and/or traffic distribution. Thereby, the routing strategy and congestion control mechanisms used by a multi-path routing scheme are two issues that still need to be addressed in efficient network planning and deployment.

Routing strategy

The Internet is currently carrying a mixture of best effort and real-time applications demanding QoS guarantees from the network infrastructure. Most of these real-time applications such as video-on-demand systems require either high end-to-end bandwidth routing or more MDC descriptions to be received at the destination when using such a policy to mitigate transport failure. In the current Internet and in bandwidth limited networks such as wireless sensor networks, such requirement can not be guaranteed with the traditional single shortest-path routing. One may claim that the simplest way to deal with the increasing bandwidth requests and to meet some specific quality of service (QoS) or service level agreement (SLA) is to over-supply the network. Though massively adopted by the providers community in the pats, this approach does not consider the economical aspects such as Operational expanditures (OPEX) and Capital expenditures (CAPEX) and has become more and nore questionable. Several studies have proposed multi-path routing to improve the network performance by splitting the offered traffic among several paths or sending multiple copies of the same data over different paths and thus, offering flow transfer higher than what is possible with any one path when considering bandwidth as a routing constraint. In WSNs, sending multiples copies of each packet along multiples paths from source to destination (sink or base station) increases the level of reliability and the probability of at least one of the copies to arrive to the destination in time and without errors. As observed in the network proposed in the figure 1.1, even an enhancement to OSPF routing, the CSPF routing can not overcome the routing issue of 160 ub from node 6 to node 7 if transmitted as the last request. Using multi-path routing to share the traffic among the paths $6-5-7$ and $6-7$, the bandwidth requirement can be met. However, multi-path routing raises problems relative to (1) the computation of paths which must be used for shipping the traffic from source to a destination, and (2) the optimal distribution of the offered traffic flows among the selected paths.

Congestion control mechanism

Network congestion is defined as a situation in which traffic flows gather at nodes and links while at the same time the network efficiency decreases by affecting the network throughput [11]. This situation is usually due to the global or localized overload. However, congestion can arise even in the presence of a small average load as a consequence of unpredictable fluctuations during which the instantaneous growth of a stochastic load is amplified and propagated throughout the network. Despite the significant increase in bandwidth, management of congestion is still a major problem in communication network as illustrated by the well known Braess paradox [12, 13, 14]. The two wide-spread ways used to deal with congestion is to drop packets, which means that some information must be resent, or to delay traffic. These strategies are not efficient, especially for real-time service since they increase delays. A congestion control procedure was proposed to overcome network congestion in [15], by limiting the intensity of traffic admitted to the network. Unfortunately, this strategy affects somehow network efficiency when considering flow acceptance in the network as system-wide measure of performance. Thus, for an efficient network operation, it is crucial to prevent the occurence of congestion. Without lost of generality, throughout this thesis we are examining strategies that keep a link load in the network far from congestion. Hence congestion control mechanisms are outside of the scope of this thesis.

1.2 Contributions and Outline

The focus of this thesis lies on multi-path to achieve network optimization in MPLS and wireless sensor networks. We examine two key issues related to the splitting of traffic over a set of parallel paths in MPLS and WSNs, namely

1. How to select the set of candidate routes , and
2. How to distribute the offered traffic among the selected paths.

Our main contributions are threefold

- **Pontryagin-based multi-path routing in MPLS networks**

  Pontryagin Maximum/Minimum Principle (PmP) is widely applied to solve different engineering and network optimization problems by using a diffusion model under temporal constraints. We extend the work done in [11] to consider an optimization model where the network links are modeled as M/M/1 queueing systems to a more general model where the links are modelled as M/M/s queueing systems and consider more general topologies where the traffic is shared among multiple source-destination pairs. This differs from the single source-destination pair modelled by [11]. Our model solves a multi-path routing problem by defining a Hamiltonian as a function of delay incurred and subjecting this Hamiltonian to Pontryagin's cost minimization to achieve efficient diffusion of traffic over the available parallel paths.

- **Game-based multi-path routing in MPLS networks**

  Several studies appeared in the literature where a link of a network is described by an M/M/1 queueing system. However the emerging and next generation IP networks are expected to carry several types of applications requiring more complex routing metrics (cost functions) and involving multiple routing constraints. We apply the LIOA cost metric [16] to the Game Theoretical (GT) framework borrowed from [17] to propose a cost-based optimization in a multi-path setting and evaluate the performance of competitive and cooperative multi-path routing schemes and the impact of the routing metric on the difference between these two schemes.

- **Multi-path routing scheme in wireless sensor networks**

  It has been stated that sensor networks require topological control with per-node transmission power adjustment since sensor nodes are limited in power, computational capacities and memory. We propose in this thesis a new QoS multi-path routing for wireless systems where nodes communicate only with a few closely positioned neighbor using low power communication scheme. Our proposed scheme builds upon the work done by X. Huang and Y. Fang [18] to design and evaluate the performance of an energy efficient WSN forwarding model with only local knowledge. Simulation reveals that our model outperforms the model proposed in [18].

These contributions are depicted by figure 1.2 where the different QoS multi-path routing schemes subdivised into two parts: (1) multi-path routing schemes for wired MPLS networks and (2) multi-path routing schemes for wireless sensor networks.

The rest of this thesis is presented as follows. We survey multi-path routing approaches in wired networkin chapter 2. Chapter 3, presents a Pontryagin minimum Principle (PmP) approach of multi-path routing to address the issue of load balancing in the network. In chapter 4, we study Game Theoretical approach of multi-path routing addressing the problem of timely delivery of data. Experimental results on the performance achieved by the two different approaches used in MPLS networks are presented in chapter 5. In chapter 6, we examine the issue of energy efficiency for wireless sensor networks and propose a new energy efficient QoS multi-path routing scheme. We conclude the thesis in chapter 7 and discuss future research directions.

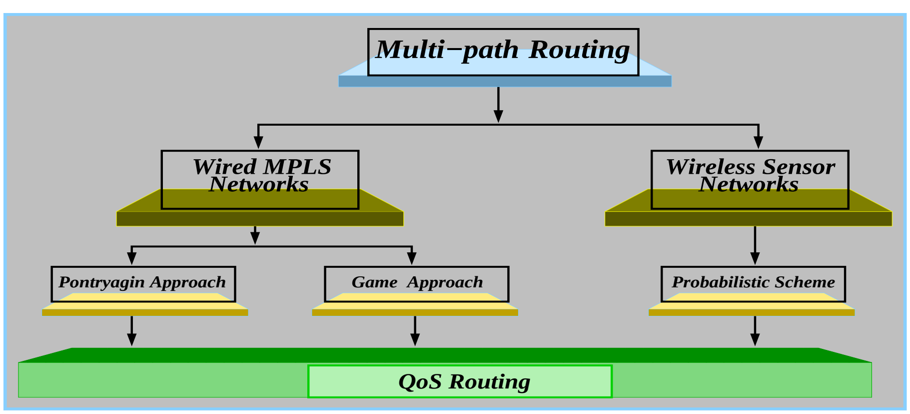


Figure 1.2: *QoS Multi-path Routing Approaches*

Chapter 2

Overview of Multi-path Routing

The emergence of new technologies has enabled the Internet to carry the traffic offered by real-time applications with higher bandwidth and minimum delay requirements such as streaming video and voice-over-IP (VoIP). The issue of routing this traffic in a network with the objective of finding an optimal network configuration, minimizing delay and packet loss, and optimizing bandwidth utilization, may be addressed by applying optimized multi-path routing through efficient path identification and traffic distribution.

1. The path identification selects paths with certain desirable properties such as minimum delay, maximum bandwidth, etc.
2. The traffic distribution optimally splits the traffic among several paths so as to achieve load balancing.

In traditional IP networks, each router implements the SPF routing protocol which selects the path followed by the IP traffic to another router in a network and adapts the path to changes in network topology and router configuration. SPF routing provides a simple and scalable routing approach. However, networks running SPF routing are often unbalanced and heavily loaded. This undesirable situation can lead to additional propagation delay and risk of traffic congestion.

Furthermore, a path may be selected randomly in case of multiple equal cost paths between nodes when no explicit indication is made to take advantage of this. Thus, multi-path routing can be deployed to benefit from the available network resources. In fact, multi-path routing makes efficient use of network resources providing improved throughput, lower delay and/or higher bandwidth compared to SPF routing as illustrated early in the introduction. However, mechanisms are needed that can manage traffic load and distribute traffic demand so as to adapt traffic to the network conditions. This is where traffic distribution comes into play, seeking a flow pattern that defines the proportion of traffic which may be assigned to each path. Figure 2.1 depicts our proposed traffic forwarding process in multi-path routing.

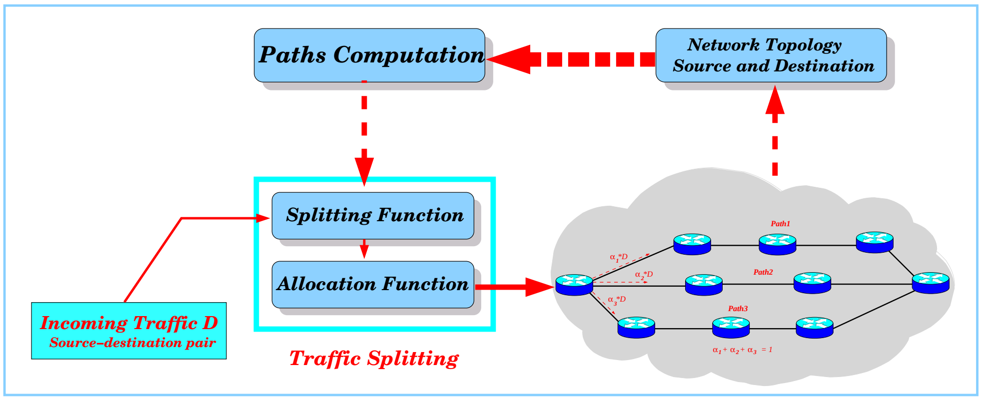


Figure 2.1: *Multiple Paths with Flow Pattern*

Knowing the source and the destination, and the topology of the network, the first step of a multi-path forwarding scheme consists of determining the optimal set of paths which will be used to route the incoming traffic to the destination. Thereafter, two functions are needed to effectively perform traffic forwarding operation through these paths across the network, namely splitting and allocation (also referred to as provisioning) function. The splitting function splits the incoming traffic optimally between the available paths and determines where each traffic proportion have to be forwarded to the destination, and the allocation function determine the time when each traffic proportion should be forwarded to the destination.

Multi-path routing, using load balancing to improve network resource utilization and minimize congestion, has been an active research area. In particular, [19] and [20] considered multi-path routing as an optimization problem with objective of minimizing congestion in the network. Sridharany et al. [21] studied the improvements in the delay achieved by distributing the aggregate traffic across multiple paths. Authors [22] and [23] investigated multi-path routing to achieve better load balancing in MPLS networks, and in [24] the authors proposed a dynamic multi-path traffic engineering mechanism to achieve load distribution over multiple paths. In [25] an heuristic scheme was proposed to proportionally split traffic among several paths that are disjoint. Both [26] and [27] studied QoS routing scheme using multiple paths. However, the problem of finding the optimal set of paths over which incoming traffic should be spread and traffic splitting strategy in multi-path routing is still an open issue.

In this chapter, we present the basic concepts of multi-path routing and graze briefly multi-path routing in traditional IP networks and in MPLS networks. Thereafter, we examine some advantages of multi-path routing compared to SPF routing.

2.1 Basic Concepts of Multi-path Routing

Multi-path routing schemes and illustration

Multi-path routing provides multiple paths between source-destination pairs to make more efficient use of the resources of the underlying physical network. Thereby, in multi-path routing a set of paths between each source-destination pair can be computed based on one of these characteristics.

1. Path multiplicity which refers to the number of available paths between source and destination pairs.
2. Path diversity referring to the number of available node and/or link disjoint paths between source and destination pairs.

To illustrate these two multi-path routing schemes, let consider the network given by figure 2.2 with 7 routers and 11 links. Each link has a capacity of 10 ub and is modeled as an $M/M/1$ queueing system. Using Constraint Shortest Path First (CSPF), the cost of a link ℓ is set to $D_\ell = 1/\left(C_\ell - f_\ell\right)$ referred to as latency function of link ℓ where C_ℓ and f_ℓ are respectively the capacity and the flow carried by the link ℓ.

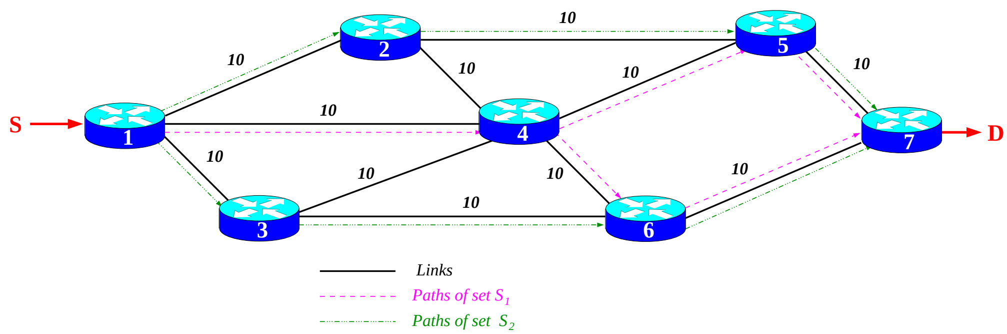


Figure 2.2: *Illustration of Multi-path Routing Schemes*

Considering the single path set $S_0 = \{1-4-6-7\}$ and the two set of paths, each with two paths $S_1 = \{1-4-5-7, 1-4-6-7\}$ and $S_2 = \{1-2-5-7, 1-3-6-7\}$, S_2 is found using path diversity characteristic. Assume that 8 ub are requested from the source S to the destination D and that the traffic is evenly distributed among the two paths using multi-path routing approach with precomputed paths from S_1 and from S_2. With SPF routing using the path $1-4-6-7$ for example, the available bandwidth to carry the offered traffic will be 10 ub. The same 10 ub will be available for the set of multiple paths S_1 while S_2 provides 20 ub. The delays resulting from routing the traffic for these three scenarios are respectively $1\frac{1}{2}$, $\frac{5}{6}$ and $\frac{1}{2}$ where the path delay D_p is expressed by the sum of the link delays $D_p = \sum_{\ell \in p} D_\ell$. The set of paths found using the diversity property leads to better use of network resources and is less likely to be congested. Except otherwise stated, the paths considered in this thesis are node disjoint which means that the set of paths is computed according to the path node diversity feature.

Startup delay and aggregate bandwidth

Let consider a network $G = (\mathcal{N}, \mathcal{L})$ where $\mathcal{N}$ is a set of nodes (vertices) and $\mathcal{L}$ a set of links (edges). Let C_ℓ denote the capacity or available bandwidth of link $\ell \in \mathcal{L}$ and $\mathcal{P}_{s-d} = \{p_1, \ldots, p_M\}$ denote the set of M multiple paths connecting the source s and the destination d. The capacity (available bandwidth) and the delay of the path $p \in \mathcal{P}_{s-d}$ are respectively given by

$$c_p = \min_{\ell \in p} C_\ell, \tag{2.1}$$

and

$$\frac{1}{\mu_p} = \sum_{\ell \in \mathcal{L}} \frac{\delta_{\ell p}}{\mu_\ell} \tag{2.2}$$

where $1/\mu_\ell$ denotes the delay over the link ℓ and $\delta_{\ell p}$ the indicator defined by

$$\delta_{\ell p} = \begin{cases} 1, \text{ if } \ell \in p \\ 0 \ otherwise \end{cases} \tag{2.3}$$

Equations (2.1) and (2.2) express respectively the available bandwidth of a path which is the minimum of bandwidth of all links on the path, and the delay of the path expressed by the sum of delays of all links on the path.

In case where a video stream is routed over the M paths for example, since the packets are routed simultaneously over the M paths the video will stall until the last packet is received at the destination. The startup delay is therefore given by

$$\mathcal{D}\left(\mathcal{P}_{s-d}\right) = \max\{\frac{1}{\mu_p}, p \in \mathcal{P}_{s-d}\}, \tag{2.4}$$

where $1/\mu_p$ is the delay over the path $p \in \mathcal{P}_{s-d}$ given in (2.2). Unless specified explicitly, in the rest of this thesis the term delay will be used to express the startup delay experienced by a packet through the network.

The aggregate bandwidth of $\mathcal{P}_{s-d}$ denoted C is the sum of capacities of all paths $p \in \mathcal{P}_{s-d}$, i.e.,

$$C = \sum_{p \in \mathcal{P}_{s-d}} c_p, \tag{2.5}$$

where c_p is the capacity or available bandwidth of the path $p \in \mathcal{P}_{s-d}$ given in (2.1).

Traffic distribution

Suppose that at a given time t, traffic demand $\lambda\left(t\right)$ is to be forwarded from source s to destination d using the set $\mathcal{P}_{s-d}$ of paths. The decision about how to distribute the traffic demand $\lambda\left(t\right)$ among the M multiple paths so as to minimize the waiting time of other traffic demand in the network arises. The traffic distribution problem consists of finding the optimal proportions α_m of traffic demand $\lambda\left(t\right)$ to assign to each path m, $m = 1, \ldots, M$ such that

$$\lambda^m\left(t\right) = \alpha_m \lambda\left(t\right), \quad m = 1, \ldots, M \tag{2.6}$$

where $\lambda^m(t)$ is the portion of traffic assigned to the path m at time t and

$$\sum_{m=1}^{M} \lambda^m(t) = \lambda(t) \quad \textit{and} \quad 0 \leq \lambda^m(t) < c_m, \quad m = 1, \ldots, M \tag{2.7}$$

We naturally have $\sum_{m=1}^{M} \alpha_m = 1$, and in particular case of equal cost paths $\alpha_m = 1/M$.

The flow pattern $\left(\lambda^1(t), \ldots, \lambda^m(t)\right)$ is said to belong to the set $\mathcal{F}$ of feasible flow patterns if it satisfies the feasibility constraints (2.7). This is of course under the assumption that the capacity configuration $c = (c_1, \ldots, c_M)$ is able to absorb the traffic demand $\lambda(t)$ at time t. This is referred to as the stability constraint, which is expressed by

$$\sum_{m=1}^{M} \lambda^m(t) < C \quad \textit{or} \quad \lambda(t) < C, \tag{2.8}$$

where C is the aggregate bandwidth given in (2.5).

2.2 Multi-path Routing in Traditional IP Networks

In IGP protocols such as OSPF, the computation of paths is based on the "shortest path first" algorithm. Considering the topology of the network obtained using link state signaling, each router constructs a tree consisting of the shortest paths to other routers with itself as root. These shortest paths are computed using Dijkstra's algorithm and updated when changes in the network topology are detected. Indeed, each router is informed about the changes in a distributed manner through a mechanism where the originator generates the link metric and other information about the link, called link-state in the OSPF context, and floods it throughout the network. This process is commonly referred to as the link-state advertisement (LSA) in the link-state routing protocol [28] and guarantees somehow the robustness of IP networks for link or node outage.

The routing decision is based on the topology of the network represented by a data structure referred to as link-state database, which includes information about the state of the link in the network. However, there may exist several paths with an equal minimum cost between two routers in the network. Without explicit indication, only one path is arbitrary chosen to forward traffic flow to the destination. As illustrated by figure 2.2, using all shortest paths one can improve network performance by minimizing the delay and/or maximizing bandwidth, and thus avoiding certain unattractive network behavior such as network congestion, unbalanced network, etc.

Equal Cost Multi-Path routing (ECMP)

Equal Cost Multi-Path routing is an extension to the OSPF routing where the traffic can be distributed over several paths of equal cost. The traffic load is evenly distributed over multiple equal-cost paths using the following three approaches [29].

- Per packet round robin forwarding.
- Dividing destination prefixes among available next hops in the forwarding entries.
- Dividing traffic according to the hash function applied to the source and destination pair.

The problem of using the round robin approach is that it can lead to the collapse of TCP (Transmission Control Protocol) if the delays of different paths are not close to each other. This is because sending TCP packets from a single flow on multiple paths with different round-trip-time (RTT) may lead to out-of-order packet arrivals. Dividing packets based on the destination prefixes among available next hops may lead to the coarse and unpredictable load split. The third approach is the most used. In this approach, a hash function, such as CRC-16 (Cyclic Redundancy Check 16) is applied to the source and destination addresses and the hash space is equally split among the available paths [29].

Without lost of generality, in this thesis we assume that ECMP refers to splitting evenly traffic volume since we are interested in network level impact. As illustration, let consider a network given in figure 2.3 in which each link has a capacity of 100 ub and the link cost set to 1 (Minimum Hop-Routing).

Let suppose that traffic of 80 ub is requested from the source S to the destination D. The shortest-hop path routing finds only one path $1-4-7$, with delay (latency) $\frac{1}{10} = 0.1$. An important issue is to find the link cost configuration which can reduce this delay. Indeed, if we set the cost of link $4-7$ to 2 without changing the cost of other links, three equal

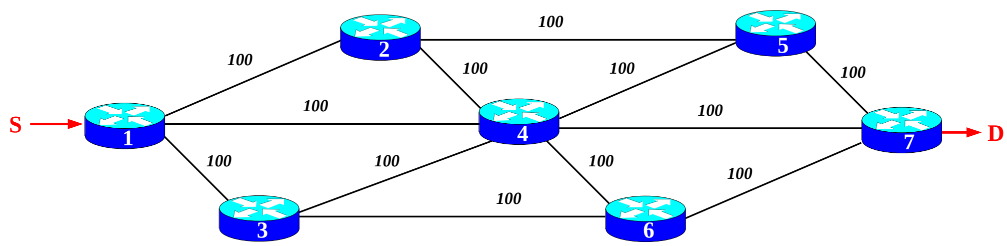


Figure 2.3: *Illustration of ECMP Routing*

cost paths $1-2-5-7$, $1-4-7$, and $1-3-6-7$ may be found. By splitting the traffic demand over these three paths, the delay is now reduced to $\frac{9}{220} \approx 0.0409$.

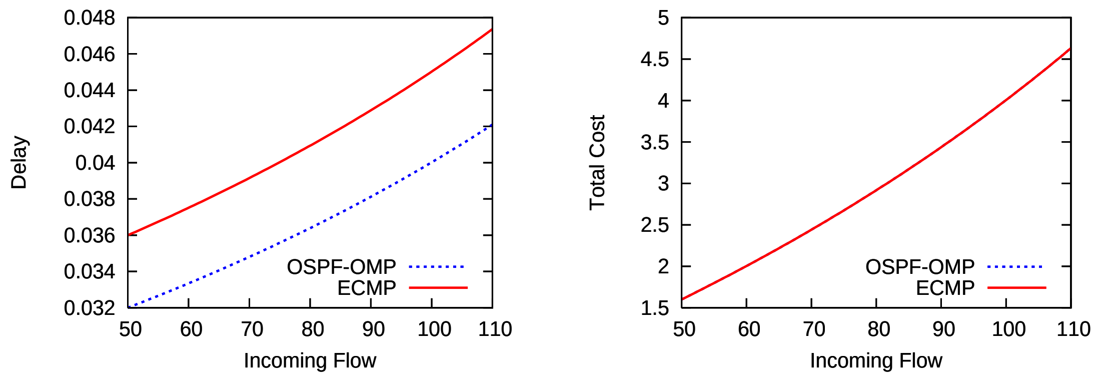
However considering latency of links Equal Cost Multi-Path routing does not always lead to the optimal splitting. Optimal splitting with equal cost paths has been researched in OSPF Optimized Multi-Path (OSPF-OMP). OSPF-OMP uses equal cost paths to support traffic-aware routing by splitting traffic load along the shortest path unequally with a fine granularity dividing hash space in different proportions [29].

For the network described by figure 2.3, let assume that the OSPF-OMP requires the splitting strategy which yield the identical latency. This requirement is satisfied by solving the equation (2.9) in λ, where λ is the traffic carried by the path $1-4-7$ and the remaining traffic $(80-\lambda)$ is equally split over paths $1-2-5-7$ and $1-3-6-4$.

$$\frac{2}{100-\lambda} = \frac{3}{100-\frac{80-\lambda}{2}} \tag{2.9}$$

Solving equation (2.9) leads to $\lambda = 45$ ub. This means that we have to forward 45 ub over path $1-4-7$ and the two remaining paths must each carry 17.5 ub. With this OSPF-OMP approach the delay is reduced to $\frac{2}{55} \approx 0.0364$.

The results computed for different incoming flows are presented in figures 2.4(a)- 2.4(b). By comparing these three approaches in term of delay, we find that ECMP routing is outperformed by OSPF-OMP. It follows that the performance of the OSPF network can be improved significantly by optimal splitting ratio of incoming traffic flow compared to the equal splitting in ECMP.



(a) *ECMP and OSPF-OMP Delay* (b) *ECMP and OMP Total Cost*

Figure 2.4: *ECMP and OSPF-OMP Delay*

Computation of paths

The paths found by the Open Shortest Path First (OSPF) protocol directly result from the SPF using Dijkstra's algorithm with a link weight system setup by the network operator so as to optimize measures of the system performance such as link utilization, delay, throughput, etc. Finding the optimal link weight system in a large network requires the application of optimization techniques. The optimization problem then consists of finding the optimal link weight system driving the traffic forwarding process on optimal paths. However, the problem of link weight optimization in SPF routing is an NP-hard and a number of heuristic methods for solving this problem have been proposed [28].

Thus OSPF protocol presents limitations for the computation of the candidate paths on which traffic can be split for a given source and destination pair. As seen before, it is possible to overcome these limitations by carefully selecting the link weights. However, the most serious difficulty arises when link fails and the OSPF-weights are suboptimal for the remaining topology [30].

2.3 Multi-path Routing in MPLS Networks

Multi-Protocol Label Switching (MPLS) [30] provides capabilities to set up explicitly multiple paths between a given source and destination referred to as explicit routing. This explicit path setting process is established and supported by the two main signaling protocols required for establishing intra-domain MPLS Label Switched Paths (LSPs) with bandwidth guarantees.

1. Resource Reservation Protocol with Traffic Engineering extension (RSVP-TE).
2. Constraint-Routed Label Distribution Protocol (CR-LDP).

MPLS allows traffic to be split to several paths, thus enabling to balance the load across paths to achieve optimal link loads and to improve the performance of the network. Using these capabilities of MPLS networks, two traffic splitting strategies are presented respectively in chapters 3 and 4, namely Pontryagin Minimum Principle (PmP) and Game Theoretic (GT) approaches.

Computation of paths

The computation of paths in MPLS based networks is more easier than in traditional IP networks. This is due to capabilities of MPLS networks to set up bandwidth guaranteed tunnels referred to as Label Switched Paths (LSPs). The network operator is thus released from the duty of selecting link weights that satisfy the network's performance goal, which is computationally challenging (NP-complete time complexity). The terms LSP, path, route and tunnel are used equivalently in the rest of this thesis.

Taking advantage of the efficient routing capability provided by MPLS networks, we make use of the Shortest Pair Vertex-Disjoint Paths algorithm [28, 31] to design an algorithm for finding the set of shortest vertex-disjoint paths used for MPLS networks. The algorithm uses the following methods.

- setDirectedEdge($\mathcal{G}$,P): replaces each edge on the shortest path P by a single arc directed towards the destination vertex.

- splitVertices($\mathcal{G}, P$): returns the graph $\mathcal{G}'$ obtained from $\mathcal{G}$ by splitting each intermediate vertex v on the shortest path P into two co-located sub-vertices v' and v" joined by an arc of zero weight directed towards the destination vertex. For each intermediate vertex v on the path P replace each external edge connected to v, by two oppositely directed arcs (of weight equal to the weight of the original edge); let one arc terminate in v' and the other originate in v", so that the three arcs (the arc from v' to v", the arc to v' and the arc from v") form a cycle.

- setReverseEdge($\mathcal{G}'$,P): reverses the direction of the arc on the shortest path P, and negates the weight of each such arc.

- disableVerticesAndEdges($\mathcal{G}'$,P,Q): transforms the modified graph $\mathcal{G}'$ back to the original $\mathcal{G}$ by removing all the interlacing edges of paths P and Q where Q is the shortest path found in the graph $\mathcal{G}'$. Removes the zero weight arc, coalesces the sub-vertices into their parent vertices and replaces the single arc of the shortest path P with their original edges (of positive weight).

- disableVertices($\mathcal{G}$,P): removes all the intermediate vertices of the path P from $\mathcal{G}$, together with all edges incident to the deleted vertices.

```
function SVDP-S(G, s, d)
00: S := φ;
01: P := BFS(G, s, d);
02: while P ≠ NULL do
03:     setDirectedEdge(G,P);
04:     G' := splitVertices(G,P);
05:     setReverseEdge(G',P);
06:     Q := BFS(G',s,d);
07:     if Q ≠ NULL then
08:         G := disableVerticesAndEdges(G',P,Q);
09:         k := 1;
10:         while k ≤ 2 do
11:             P := BFS(G,s,d);
12:             S := S ∪ {P};
13:             disableVertices(G,P);
14:             k := k + 1;
15:         P := BFS(G,s,d);
16:     else
17:         S := S ∪ {P};
18:         P := NULL
19: return S
```

Algorithm 1. *Shortest Vertex-Disjoint Paths Set Algorithm*

The procedure for finding the candidate paths is described by the algorithm 1 depicted above. The procedure returns the set of node-disjoint paths used in MPLS based networks to split the traffic across the network from a source s to the destination d. Note that we are using Bread First Search (BFS) Shortest-Path Algorithm where the computation of the shortest path is required: steps 06, 11 and 15. This is due to the fact that (1) the original Dijkstra algorithm can fail to find the shortest path in case where some links of the graph considered have negative weights, and (2) the complexity of the modified version of the Dijkstra's algorithm proposed to overcome the weakness of the original Dijkstra algorithm is still $O(n^2)$ in the worst case where n is the number of nodes of the graph, whereas the complexity of BFS algorithm is $O(n+m)$ with m the number of links of the graph [28, 31, 32].

The shortest vertex-disjoint paths set algorithm is based on BFS shortest path algorithm. Thus, by using a pertinent data structure the shortest vertex-disjoint paths set algorithm has $O\big(p(m+n)\big)$ complexity where p is the number of paths, for a graph of m links and n nodes.

Traffic engineering scheme

Our proposed traffic engineering model is presented in figure 2.5. In this model, we consider as input to the design of our routing algorithm the information about the network state and the traffic offered by users to the operational network.

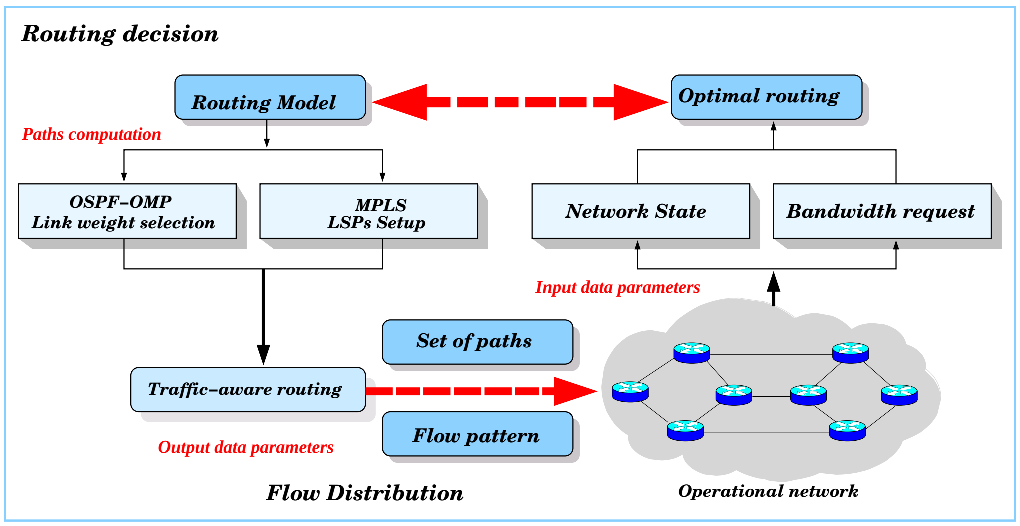


Figure 2.5: *Traffic Engineering Architecture*

The routing model relies upon underlying MPLS protocols to either use explicit path mechanisms provided by MPLS or find the set of link-cost metrics for all links referred to as link metric system that provides the best paths in case of traditional IP networks. The optimization process then consists of finding, for each traffic demand, a set of paths and a corresponding flow pattern that specifies how the traffic is distributed into the network given its topology. As result, we have a traffic-aware routing which should provide better performance compared to the standard SPF routing.

2.4 Advantages of Multi-path Routing

Multi-path routing offers several advantages as pointed out in some illustrations along this chapter. These include (1) load balancing, (2) quality of service (QoS) and (3) reliability.

Load balancing and quality of service

The primary idea behind the utilization of multiple paths is to spread the load more evenly over the network. This is where load balancing is used to enhance the utilization of the network resources by minimizing the maximum link utilization, reducing delay and risk of traffic congestion. In SPF routing, network operators rely on the adjustment of link weights over a network so as to improve the network performance. This adjustment may unfortunately be sometimes impossible and may affect the overall traffic load. With multi-path routing, load balancing can be achieved by splitting traffic load among multiple paths based on both quality and load of these paths. Hence traffic may be controlled to flow optimally through certain paths, thus offering the best service such as minimum packet delay and packet losses, etc. Moreover, multi-path routing has the potential to aggregate bandwidth, thus allowing a network to support higher traffic transfer than with any single path.

Reliability

Beside load balancing and quality of service that may be provided, multi-path routing is actively used in path protection schemes such as "1:1" and "1+1" protection where a full protection is provided to the active (working) path. The active path is protected by an alternative (backup) path so that when the active path fails, the alternative path is immediately deployed. This leads to a more reliable network ensuring fast recovery and re-routing.

Chapter 3

Pontryagin Routing Approach

One of the challenges in the design of communication networks is to find efficient traffic rules that optimize system-wide measures of performance such as throughput, delay, etc. When using a set of parallel paths connecting source and destination pair, it may seem better to use the path with larger capacity or lowest waiting time. However, it appear that this strategy does not guarantee that the total waiting time of traffic flows in the network is minimized. This happens because the assignment of traffic to a given path influences the waiting time of the future traffic that will be forwarded to that path.

In this chapter we address the issue of distributing the traffic offered to a network among a set of paths to improve network performance. This is achieved in a multi-path setting by using a pre-planned model where paths identification and traffic distribution are performed separately [6]. We extend the model proposed by Filipiak [11] where traffic is distributed to a single source and a single destination connected by a set of parallel links to a general topology where sets of paths between multiple source and destination pairs are used to find an optimal network configuration minimizing the waiting time of the traffic flows in the network. Whereas Filipiak routing model uses the M/M/1 queueing system to model the links of a network, we model the links of a network as M/M/s queueing systems and propose an optimal routing strategy where an Hamiltonian is used to define the waiting time in the network. This Hamiltonian is subjected to Pontryagin's cost minimization to diffuse the traffic offered to a network over the available parallel paths.

We apply Pontryagin Minimum Principle (PmP) to examine one of the key issues that needs to be addressed in forwarding flows over multiple paths, *i.e.*, when each path should be used and what proportion of total inflow should be directed to it. We develop a time dependent model using a systematic approach optimizing dynamic flows between source and destination pair to achieve better overall performance. Our study provides an adaptive routing solution where the decision about flow allocation is taken according to the actual state of the network. This extends the functionality of MPLS [30] where deterministic mechanisms have been proposed to facilitate the dispersion of traffic over multiple paths for a given source and destination pair.

In the following section, we present the parallel paths modeling scheme and formulate the traffic distribution problem as a dynamic flow optimization problem which is solved using Pontryagin Minimum Principle (PmP). We then propose an extension to the PmP solution that achieves efficient bandwidth sharing on shared links and evaluate its efficiency in a 10−node test network.

3.1 Modeling a Set of Parallel Paths

Consider a network represented by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ set of links, each link having the maximum reservable bandwidth (capacity) C_ℓ, for $\ell \in \mathcal{L}$. Let $\mathcal{P}_{s-d}$ be a set of M parallel or disjoint routes between a source-destination (s,d) pair. Let $\lambda(t)$ denote the flow arrival rate at time t from node s to node d and assume that each flow transmits packets at a fixed rate for random duration with rate μ_m^{-1} along the route $p_m \in \mathcal{P}_{s-d}$ it is assigned to. The traffic load or the amount of traffic on route p_m at time t, denoted by $x_m(t)$, is the sum of the flows currently routed across it. The flow arrival rate to route p_m is denoted by $\lambda^m(t)$, *i.e.*, the portion of total arrival between node s and node d allocated to route p_m. To consider the fact that in our model the routing decision is made based on the current state on the network, we assume that traffic arrivals during $[0, T]$ are routed based on the state of path p_m available at initial time t = 0. It is known from the optimal control theory that the time evolution of route state $x_m(t)$ within a dynamic flow system can be expressed by the first order differential equations

$$\dot{x}_m(t) = -\mu_m G_m[x_m(t)] + \lambda^m(t) \quad m = 1, \ldots, M \tag{3.1}$$

where the function $G_m[x_m(t)]$ approximates the system utilization while $\lambda^m(t)$ represents the assignment of incoming traffic flows to route p_m.

In a multi-path setting, the traffic offered to the source-destination pair (s,d) is split among the M paths to achieve load balancing. This load balancing process is conditioned by the feasibility constraints expressed by

$$\sum_{m=1}^{M} \lambda^m(t) = \lambda(t) \quad and \quad 0 \leq \lambda^m(t) \leq \lambda(t) \tag{3.2}$$

For each $m = 1, \ldots, M$, we thus seek to find the proportion of traffic $\alpha_m(t) = \lambda^m(t)/\lambda$ routed to each path m. The flow pattern achieved is given by $\hat{\lambda}(t) = (\lambda^1(t), \lambda^2(t), \ldots, \lambda^M(t))$ and belongs to the set of feasible flow patterns $\mathcal{F}$ as defined in section 2.1 of chapter 2.

The functions G_m are determined by the stochastic properties of the arrival and service processes. However, all flows routed to a given path are formed from one input process by random selection. Thus, for the Poisson process they have the same stochastic properties as the input process itself [33]. Therefore, $G_m = G$ and the model of the parallel paths set system is described by the equations

$$\dot{x}_m(t) = -\mu_m G[x_m(t)] + \lambda^m(t), \quad m = 1, \ldots, M. \tag{3.3}$$

The cost functional which is the total waiting time of the total traffic load in the system during time $[0, T]$ is given by

$$\tau = \int_0^T \sum_{m=1}^{M} x_m(t) dt \tag{3.4}$$

Equation (3.4) expresses the fact that the waiting cost in the system is proportional to the system's total traffic load: a heavily loaded network will keep traffic flows longer in the network while a lightly loaded network will release the traffic flows quicker.

3.1.1 Problem formulation

Having the model and the criterion τ, our objective is to assign flow $\lambda(t)$ within the system described by equations (3.3) so as to satisfy feasibility constraints (3.2) and minimize criterion τ (3.4). Equation (3.3) expresses path load dynamics that may impact the routing decision, *i.e.*, the more a path is loaded, the faster traffic flows depart from it.

Thus, given the current path states $x_m(0)$, $m = 1, \ldots, M$ and the flow arrival $\lambda(t)$, we are able to determine the optimal traffic distribution that minimizes the cost functional. In the following section, we derive such a solution by finding L paths among M paths selected to disperse the traffic and the portion of traffic corresponding to each path. The following intuitive behavior is achieved: if the network is lightly loaded, it is beneficial to consistently use only the paths whose unused capacity is high or traffic intensity is small; if the network becomes more loaded, it is advantageous to distribute the load over a larger set of paths so as to accommodate the incoming flows.

3.1.2 Pontryagin solution

From Pontryagin Minimum Principle [34], an optimal flow pattern $\lambda^{m^*}(t), m = 1, \ldots, M$ that solves the problem considered must minimize, at each time t, the Hamiltonian function over all flow patterns satisfying the feasibility constraints (3.2), *i.e.*,

$$H\left[x, q, \lambda^*\right] = \min_{\lambda \in \mathcal{F}} H\left[x, q, \lambda\right] \tag{3.5}$$

where the Hamiltonian function $H\left[x, q, \lambda\right]$ is given by

$$H\left[x, q, \lambda\right] = \sum_{m=1}^{M} x_m(t) + \sum_{m=1}^{M} q_m \dot{x}_m(t) = \sum_{m=1}^{M} q_m \lambda^m(t) + \sum_{m=1}^{M} x_m(t) - \sum_{m=1}^{M} q_m \mu_m G\left[x_m(t)\right] \tag{3.6}$$

with

$$\dot{q}_m(t) = -\frac{\partial H}{\partial x_m}, \quad m = 1, \ldots, M \tag{3.7}$$

called the co-state variable corresponding to paths state equation (3.3).

Differentiating $H\left[x, q, \lambda\right]$ in (3.6) with respect to x_m and combining the expression obtained with relation (3.7) yield the following expression

$$\dot{q}_m(t) = -1 + \mu_m q_m \frac{dG(x_m)}{dx_m}. \tag{3.8}$$

Furthermore, since in the expression of $H\left[x, q, \lambda\right]$ only the first term depends on $\lambda^m(t)$, the optimization problem (3.5) is equivalent to the following problem.

$$\begin{aligned} &\min_{\lambda^1(t),\ldots,\lambda^M(t)} \sum_{m=1}^{M} q_m \lambda^m(t) \\ &\text{subject to} \quad \sum_{m=1}^{M} \lambda^m(t) = \lambda(t) \\ &\lambda^m(t) \geq 0,\, m = 1,\ldots,M \end{aligned} \tag{3.9}$$

Let $\hat{q}(t) = min\{q_m(t), 1 \leq m \leq M\}$. From minimization of (3.9) it follows that

$$\lambda^m(t) > 0 \quad \text{only if} \quad q_m(t) = \hat{q}(t), \quad m = 1,\ldots,M \tag{3.10}$$

If $\hat{q}(t)$ is attained by only one co-state variable then $q_m(t) = \hat{q}(t)$, which implies that $\lambda^m(t) = \lambda(t)$.

However, if $q_m(t) = \hat{q}(t)$ for more than one path then intensities $\lambda^m(t) > 0$ cannot be directly specified. We shall now try to find a flow pattern which corresponds to the more interesting situation with several paths engaged. Thus, let us assume that during some interval $\langle t_a, t_b \rangle$ traffic flows are forwarded over L paths, *i.e.*, for $t \in \langle t_a, t_b \rangle$ the following conditions hold.

$$\begin{aligned} &\lambda^m(t) > 0 \text{ for } m = 1,\ldots,L \\ &\lambda^m(t) = 0 \text{ for } m = L+1,\ldots,M \end{aligned}$$

where the numbering of L paths is rearranged for the sake of convenience, then according to (3.10) for $t \in \langle t_a, t_b \rangle$ we have

$$q_1(t) = q_2(t) = \cdots = q_L(t) = \hat{q}(t), \quad \textit{and} \tag{3.11}$$

$$q_m(t) > \hat{q}(t), \quad m = L+1,\ldots,M. \tag{3.12}$$

From relation (3.11) it follows that inside the particular time interval we also have

$$\dot{q}_1(t) = \dot{q}_2(t) = \cdots = \dot{q}_M(t) \tag{3.13}$$

Finally, if conditions (3.11) and (3.13) are satisfied, then for $t \in \langle t_a, t_b \rangle$ the relation (3.8) leads to the following relation.

$$\mu_m \frac{dG(x_m)}{dx_m} = \mu_{m+1} \frac{dG(x_{m+1})}{dx_{m+1}}, \quad m = 1,\ldots,L-1 \tag{3.14}$$

where L is the number of paths used by $\mathcal{P}_{s-d}$ to forward traffic to the destination. This relation, which is necessary for the optimality, is called the basic property of the optimal flow pattern.

The quantity $\delta_m(x_m) = \mu_m \frac{dG(x_m)}{dx_m}$ represents the incremental increase in the output of the m^{th} path due to a small change in the traffic to be forwarded and awaiting service. Equations (3.14) state that these quantities should be equal for active paths. To specify the flow pattern, it is necessary to know when each path should be used and which quantities of total inflow should be directed to it.

Computing the switching times

When the traffic become less heavy it may be worthwhile to do not use one of the paths with a small capacity because the delay in the path with large capacity or smallest traffic intensity is shorter. For two paths with same capacity, their sequence make no difference. By assuming that the traffic load is always increasing, we expect it to not exceed some maximum, this means that function $G(x)$, which approximates the total traffic load must be concave. Now assume that the system starts an operation from the empty state, $x_m(0) = 0$, $m = 1, \ldots, M$, and the paths are numbered in decreasing capacity order, so that at the beginning all the inflow is directed to path 1 with the highest capacity. By doing so we increase x_1 and consequently, owing to the concavity of G, decrease $\frac{dG(x_1)}{dx_1}$. If the load is sufficiently large, after some time t_1, $\delta_1(x_1)$ may reach $\delta_2(0)$. When $\delta_1(x_1)$ first becomes equal to $\delta_2(0)$, according to the necessary conditions for the optimality (3.14), path 2 is going to be used. If the load still continues to increase, inflows in the paths 1 and 2 grow and $\delta_1(x_1) = \delta_2(x_2)$ may become so small that they reach $\delta_3(0)$. Then the path 3 is switched on. That reasoning, and the argument that a path with larger capacity is switched on before the path with smaller capacity, suggest that we define the supremum $\hat{\delta}(t)$ as follows

$$\hat{\delta}(t) = \max_{1 \leq m \leq M} \{\delta_m(x_m)\} \tag{3.15}$$

and the path m is switched on when $\delta_m(x_m)$ first becomes equal to $\hat{\delta}(t)$.

Computing the flow intensities

We shall consider the assignment of traffic to L paths which we know are to be used. Differentiating (3.14) with respect to time and substituting from path state equations (3.3) for $\dot{x}_m$ and $\dot{x}_{m+1}$, we find that flow intensities satisfy the equations:

$$-\mu_m^2 \frac{d^2G(x_m)}{dx_m^2} G(x_m) + \mu_m \frac{d^2G(x_m)}{dx_m^2} \lambda^m = -\mu_{m+1}^2 \frac{d^2G(x_{m+1})}{dx_{m+1}^2} G(x_{m+1}) + \mu_{m+1} \frac{d^2G(x_{m+1})}{dx_{m+1}^2} \lambda^{m+1}. \tag{3.16}$$

Thus, we have $L-1$ independent equations which are linear with respect to flow intensities. Using the flow conservation condition given by

$$\lambda^1(t) + \lambda^2(t) + \cdots + \lambda^L(t) = \lambda(t) \tag{3.17}$$

we now have a system of L linear equations and L unknown variables. The flow intensities $\lambda^m(t), m = 1, \ldots, L$, are found by solving this system of linear equations.

3.1.3 Traffic distribution algorithm

Based on the procedure followed in solving the problem of finding an optimal traffic splitting over a set of parallel paths we derive two algorithms, namely the Pontryagin minimum Principle (PmP) Flow Allocation algorithm and Pontryagin Routing algorithm.

PmP Flow Allocation algorithm

Given a set of parallel paths $\mathcal{P}$, the traffic λ offered to this set, and time horizon T. The traffic offered can be dispersed across the set of parallel paths using the PmP algorithm which consists of finding (1) optimal quantities of total incoming flow which must be forwarded over each path $p \in \mathcal{P}$, and (2) their corresponding switching times, i.e., time when each path should be switched on.

function PmP$(\mathcal{P}, \lambda, \mathbf{T})$

$t_1 := 0$; $\lambda^1 := \lambda$;

for $2 \leq m \leq M$ **do**

  $\lambda^m := 0$; $t_m := T$;

for $1 \leq m \leq M$ **do**

  Integrate state paths equation (3.3) to find the traffic load $x_m(t)$;

$m := 2$;

while $m \leq M$ **do**

  Compute $\delta_m\left(x_m\right) = \mu_m \frac{dG(x_m)}{dx_m}$;

  $t_s := t_{m-1}$;

  while $t_s < T$ **do**

    Compute $\delta_{m-1}\left(x_{m-1}\right) = \mu_{m-1} \frac{dG(x_{m-1}(t_s))}{dx_{m-1}}$;

    if $\delta_{m-1}\left(x_{m-1}\right) = \delta_m\left(x_m\right)$ **then**

      Switch on the path m *at time* t_s;

      $t_m := t_s$;

      Find $\hat{\lambda} := \left(\lambda^1, \ldots, \lambda^M\right)$ *by solving system of equations (3.16) and (3.17);*

      for $1 \leq n \leq M$ **do**

        Integrate paths state equation (3.3) to find the traffic load $x_n(t)$,

      break;

    $t_s := t_s + h$; *//h is the step size when numerical solving equation (3.3)*

  $m := m + 1$;

return $\hat{\lambda}$ and $\hat{t} = (t_1, \ldots, t_M)$;

Algorithm 2. *PmP Flow Allocation Algorithm*

Indeed, at the beginning of the operation the flows are forwarded over the path with large capacity or equivalently with small traffic intensity (least-loaded path) numbered 1. The smaller capacity is used only if the flows become bigger and consequently the queue becomes long at the path 1.

Thus, in spite of slow service the delay at the other paths is smaller than the delay caused by queueing at path 1. This means that the initial flow pattern must be set as follows.

$$\lambda^1(t) = \lambda \quad and \quad \lambda^m(t) = 0, \quad for \quad m = 2, \ldots, M \tag{3.18}$$

From relation (3.18), the inequality $\delta_1(x_1) > \delta_2(x_2)$ holds during the time interval $\big[0, t_s\big\rangle$. At t_s, condition $\delta_1(x_1) = \delta_2(x_2)$ starts to be valid and then the following path should be used at time t_s found by solving numerical integration of the path state equation (3.3). The underlying algorithm is built around following features.

1. *Numbering of paths in decreasing order of capacities, and Computation of quantities $\delta_m(x_m)$ for each path.*
2. *Determination of path p_m which must be used, i.e., path for which $\delta_m(x_m) = \hat{\delta}(t)$.*
3. *Computation of the feasible flow over each path by solving the system of equations (3.16) and (3.17).*
4. *Integration of state paths equation (3.3) to find the traffic load over paths.*

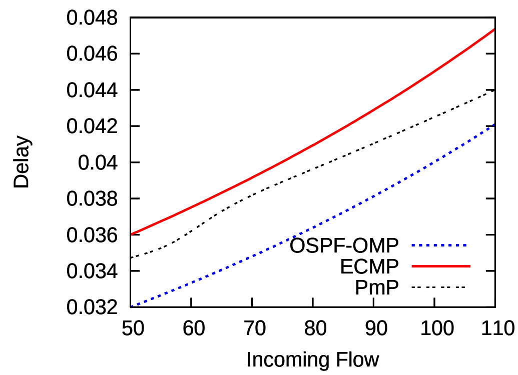
To assess the efficiency of the PmP approach, we consider the routing illustration given in section 2.2 of chapter 2 in context of OSPF networks. We have observed the following facts for the network considered in figure 2.3 when 80 ub is requested from the source S to the destination D.

- Using Equal Cost Multi-Path Routing (ECMP), each path will carry 80/3 ub and the delay achieved is 0.0409.
- Using OSPF-OMP (e.g., identical latency path), 45 ub will be directed to the path $1-4-7$, and 17.5 ub to each remaining paths. The delay is now 0.0364

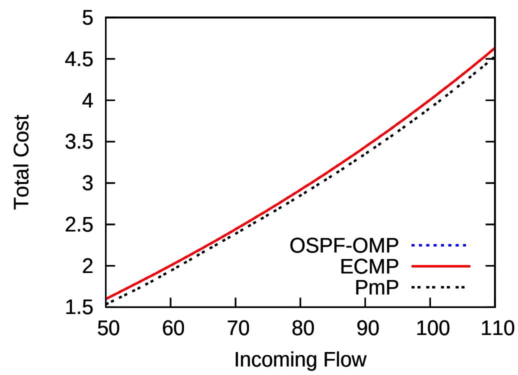
With PmP approach, 31.441044 ub will follow the path $1-4-7$, and 24.279475 ub will be sent over each remaining path and the delay is thus 0.039621. We made following observations based on global results presented in figure 3.1(a)- 3.1(b) comparing these approaches considering achieved delay and total cost incurred by the network.

1. PmP approach outperforms Equal Cost Multi-Path (ECMP) routing in term of delay.
2. OSPF-OMP considered offers minimum delay compared to PmP approach.

However, the PmP approach provides better trade-off between delay and total cost. Looking at the results depicted by figure 3.1 for different incoming flows, the PmP approach provides a “good” traffic distribution, keeping the network far from congestion while ensuring the minimum cost. These facts justify the efficiency of the PmP approach proposed when applied to MPLS networks.



(a) *ECMP, OMP and PmP Delay*



(b) *ECMP, OMP and PmP Cost*

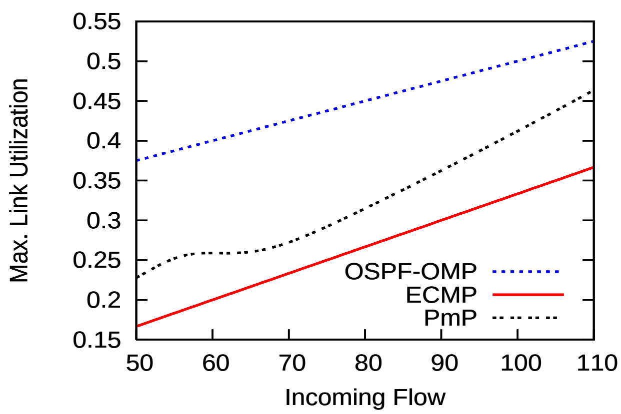


Figure 3.1: *ECMP, OSPF-OMP and PmP Link Utilization*

Pontryagin Routing algorithm

We consider the routing of traffic offered to a given source-destination pair in this network using a two-step scheme consisting of (1) finding a set of parallel paths connecting the source and destination, and (2) distributing traffic to these parallel paths.

Indeed, the computation of the “best” set of disjoint paths between source and destination is accomplished using the Shortest Vertex-Disjoint Paths Set (SVDP-S) algorithm proposed in section 2.3 of chapter 2. And the traffic distribution is done by the PmP Flow Allocation algorithm above.

Thus, given a network $\mathcal{G}$ and a request to route λ_{s-d} bandwidth units between two nodes s and d, the Pontryagin Routing algorithm executes these three following steps in routing this request.

Step 1. Find a set $\mathcal{P}_{s-d}$ of paths using the Shortest Vertex-Disjoint Paths Set algorithm:
(a): $\mathcal{P}_{s-d} := SVDP - S(\mathcal{G}, s, d)$;
(b): *Compute the aggregate bandwidth* $C = \sum_{m=1}^{M} c_m$;
Step 2. Route the request λ_{s-d} :
(a): **if** $\lambda_{s-d} < C$ **then**
(b): - *Traffic splitting: Compute flow pattern* $\left(\lambda^1, \ldots, \lambda^M\right)$ *using PmP algorithm;*
(c): - *Traffic allocation: Assign each portion* λ^m *to the corresponding path* $p_m \in \mathcal{G}$;
Step 3. Update the link f_ℓ flows.
(a): **for** $\ell \in p_m$ **do**
(b): $f_\ell := f_\ell + \lambda^m$;

Algorithm 3. *Pontryagin Routing Algorithm*

3.2 Using Different Queueing Models

In this section, we propose an optimal bandwidth allocation strategy using differentiated queueing models for different ingress-egress pairs to split traffic over paths using PmP Flow allocation algorithm. We start by using different queueing models to compute the values of the function $G(x)$ approximating the system utilization. We then apply the proposed strategy to the 10-node network in order to evaluate its efficiency.

3.2.1 Modeling system utilization

In order to investigate the sensitivity of the set of paths in term of switching times when links are modeled using different queueing system, namely M/M/1, and generally M/M/s in which arrival process is described by a Poisson distribution and service time follows an exponential distribution, we first present the basic expressions of $G(x)$ for M/M/1 and M/M/s queueing models. These formulas will be used in the next subsections.

Using the M/M/1 queueing model

From queueing theory [35, 36], it is known that, on average, traffic arrivals at the steady state in the system modeled as M/M/1 is given by

$$x = \frac{\rho}{1-\rho} \tag{3.19}$$

provided that $\rho < 1$, which is the traffic intensity or offered load in the system, given by $\rho = \lambda/\mu$.

Thus, the function $G(x)$ in (3.3), which approximates the system utilization is given by

$$G(x) \equiv \rho(x) = \frac{x}{1+x}. \tag{3.20}$$

Using the M/M/s queueing model

The M/M/s queueing model is a generalization of the M/M/1 queueing model [35, 36], extending it to the case of multiple parallel service facilities. For this case, on average , traffic arrivals at the steady state in the system is given by

$$x = x_q + \frac{\lambda}{\mu} \quad or \quad x = x_q + s\rho \tag{3.21}$$

where $\rho = \lambda/s\mu < 1$ and x_q representing on average traffic arrivals waiting in queue is given by

$$x_q = \left[\frac{(\lambda/\mu)^{s+1}/s}{s!\big(1-\lambda/(s\mu)\big)^2}\right]p_0 = \left[\frac{(s\rho)^{s+1}}{ss!\,(1-\rho)^2}\right]p_0$$

where p_0 is the probability that no request (customer) is found in the system at an arbitrary point of time after the process has reached its statical equilibrium given by

$$p_0 = \left[\sum_{n=0}^{s-1}\frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n + \frac{1}{s!\,(1-\rho)}\left(\frac{\lambda}{\mu}\right)^s\right]^{-1} = \left[\sum_{n=0}^{s-1}\frac{1}{n!}\,(s\rho)^n + \frac{(s\rho)^s}{s!\,(1-\rho)}\right]^{-1}$$

It is obtained from the steady-state probability p_n of finding n customers in the system given by

$$p_n = \begin{cases} \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n p_0 & for \quad 1 \leq n \leq s \\ \frac{1}{s^{n-s}s!}\left(\frac{\lambda}{\mu}\right)^n p_0 & for \quad s \geq n \end{cases}$$

System utilization $G(x)$ generalization from M/M/1 to M/M/s queueing

The determination of the function $G(x)$, which approximates the system utilization for M/M/s queueing model, requires to solve equation (3.21) in order to obtain ρ as a function of x. However, the explicit expression of ρ is hard to obtain. We thus make use of a Gunther approximative formula [37] given by

$$x \approx \frac{s\rho}{1-\rho^s} \tag{3.22}$$

Thus, to determine the value of $\rho \equiv G(x)$ we have to solve the following equation in ρ, where x is taken as parameter.

$$x\rho^s + s\rho - x = 0 \tag{3.23}$$

which can be solved by any root-finder program (e.g., with *xmaxima* package [38]).

Solving equation (3.23) for $s = 2$, and $s = 3$, the functions $G(x)$ which approximates the system utilization for M/M/2 and M/M/3 queueing models are given below.

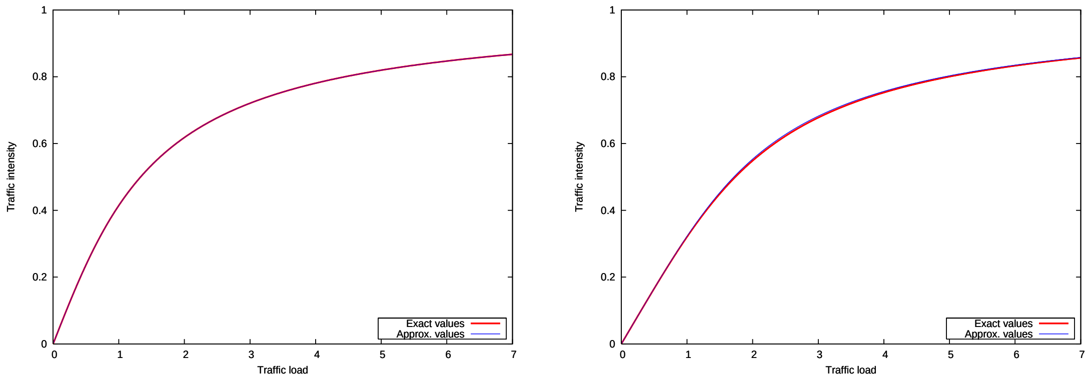
- For M/M/2 queueing ($s = 2$), we have

$$G(x) = \begin{cases} \frac{-1+\sqrt{1+x^2}}{x} & for \quad x \neq 0 \\ 0 & for \quad x = 0 \end{cases} \tag{3.24}$$

- For M/M/3 queueing ($s = 3$), we have

$$G(x) = \begin{cases} \left(\frac{\sqrt{\frac{4+x^3}{x}}+x}{2x}\right)^{1/3} - \frac{1}{x\left(\frac{\sqrt{\frac{4+x^3}{x}}+x}{2x}\right)^{1/3}} & for \quad x \neq 0 \\ 0 & for \quad x = 0 \end{cases} \tag{3.25}$$

In order to estimate the error made by approximating the function $G(x)$ for M/M/2 and M/M/3 using equation (3.23), for each case we plot $G(x)$ and its Gunther approximative function, and these plots are given in figure 3.2.



(a) *M/M/2 Queueing* (b) *M/M/3 Queueing*

Figure 3.2: *Comparison of Exact and Approximative Values of G(x)*

These figures show that the error made by replacing the function $G(x)$, which approximates the system utilization, by its approximative function constructed from equation (3.23) is negligible. This reveals the relevance of replacing the function $G(x)$ by its approximative function obtained from relation (3.23).

M/M/s queueing behavior

Fact 3.2.1. *For the system modeled using M/M/s with $s \geq 1$ queueing model, the function $G(x)$ is concave and satisfies the fact that the traffic carried on a given path is smaller than the capacity of the path. The later fact is mathematically expressed by*

$$\lim_{x\to\infty} \frac{dG(x)}{dx} = 0 \tag{3.26}$$

★

Indeed, the above result is obvious for M/M/1 queueing model. For the case where $s > 1$, when differentiating twice the relation (3.23) we have that

$$\frac{dG(x)}{dx} = \frac{1 - \big(G(x)\big)^s}{s\Big[1 + x\big(G(x)\big)^{s-1}\Big]} \tag{3.27}$$

and

$$\frac{d^2G(x)}{dx^2} = -\frac{\big(G(x)\big)^{s-2}\Big[1-\big(G(x)\big)^s\Big]\Big[(s+1)x\big(G(x)\big)^s + 2sG(x) + (s-1)x\Big]}{s^2\Big[1+x\big(G(x)\big)^{s-1}\Big]^3} \tag{3.28}$$

As G is a bounded function (above by 1 and below by 0) and $x \geq 0$, it follows that the second derivative of G given in relation (3.28) is negative, which means that the function G is concave, and from relation (3.27) we have

$$\lim_{x\to\infty} \frac{dG(x)}{dx} = 0 \tag{3.29}$$

meaning that for the large loads the traffic carried is smaller than the path capacity. This proves the result given in fact 3.2.1.

Switching times behavior

During network operation, if the traffic arrival rate increases, the traffic intensity increase and traffic flows might require to be dispersed over a larger set of paths. On the other hand, switching times are function of incoming traffic $\lambda(t)$ and paths' capacities (in term of holding time) or traffic intensities on paths. This means that the higher the traffic, the earlier the following paths are used and the higher the paths' capacities or equivalently the smaller the traffic intensity on paths, the later the following paths are used.

The intuition for the above observation, lies in that the system utilization function $G(x)$ is a concave function and ensures that the traffic carried is smaller than the path capacity.

For a given traffic load, we would like to compare the duration of switching times between different queueing models, *i.e.*, how switching times vary in term of the s value in M/M/s queueing model under consideration.

In figure 3.3, we plot the system utilization function G for three different values of s(=1, 2, 3), *i.e.*, M/M/1, M/M/2 and M/M/3 queueing models.

From this figure, we find that for a given traffic load the traffic intensity or equivalently the system utilization decreases with the value of s. In fact, this behavior comes from the relation (3.21) or (3.22) in which, it can be seen that for a given load the traffic intensity

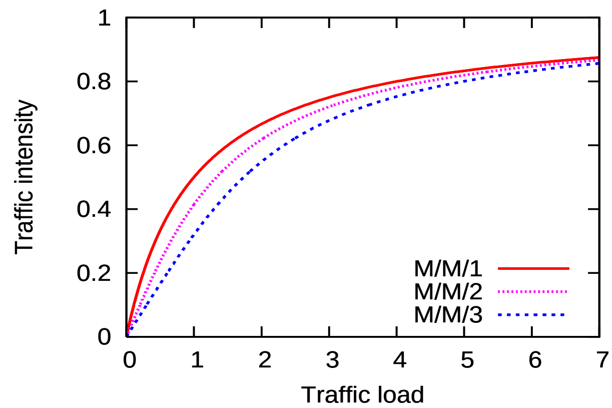


Figure 3.3: *Queueing Behavior*

is inversely proportional to the s value of the considered queueing model. Thus, since the switching times decrease with the traffic intensity, it follows that the switching times will increase with the s values in the queueing models under consideration. This observation lead us to state the following import result.

Fact 3.2.2. *The switching times increase with the value of s. Therefore using $M/M/s$ queueing model leads to longer switching times compared to $M/M/1$.*

★

To illustrate our statement, we plot the switching times for a system formed by two parallel links with holding times respective 1/150 and 1/60 when the system utilization is modeled using M/M/1, M/M/2 and M/M/3 queueing models. The results are given in figure 3.4 which shows that M/M/3 switching times are longer than M/M/2 switching times and of course these of M/M/2 are longer than M/M/1 switching times, thus confirming what is stated in fact 3.2.2. The impact of these findings is evaluated in the section 3.2.3 where these results are used to achieve different routing configurations.

The above fact ensures that the use of PmP Flow Allocation algorithm may lead to different network configurations depending on how the ingress-egress pairs are modeled. It also guarantees that using differentiated queueing to model different set of parallel paths can lead to more efficient use of the network resources by reducing the impact of the competition on bottleneck links through switching time differentiation. Using this approach to manage a network can increase its robustness by minimizing the packet loss due to the interference among competing flows on links and increasing the network throughput to each destination.

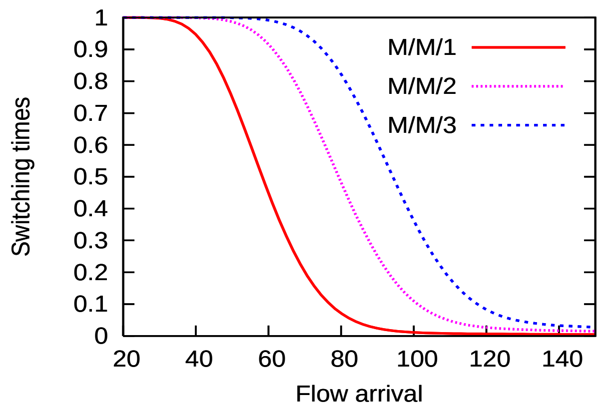


Figure 3.4: *Switching Times Variations*

This leads to improved overall network performance as illustrated in section 3.2.3.

3.2.2 Algorithmic solution

Given a network and two different source-destination pairs, we consider the routing of traffic offered to this network using a two- or four-step algorithm consisting of (1) finding sets of parallel paths for each source-destination pair and defining bandwidth usage on interference links, and (2) dimensioning the path and using the *PmP* Flow Allocation algorithm above to distribute traffic to the network. More precisely, the algorithm performs the following steps.

Step 1. Computation of paths sets and the interference on links:

Initialization of link cost L_ℓ *and interference* π_ℓ
  for *all* $\ell \in \mathcal{L}$ **do**
    $L_\ell := 1;$
    $\pi_\ell := 0;$
for *each ingress-egress* $i - e$ *pair* **do**
  Find the set $\mathcal{P}_{i-e}$ *by running SVDP-S algorithm;*
  for *all* $\ell \in p$ *with* $p \in \mathcal{P}_{i-e}$ **do**
    $\pi_\ell := \pi_\ell + 1;$ *//update the path interference on link* π_ℓ

Step 2. Set up link sharing bandwidth:

for *all* $\ell \in \bigcup_{i-e} \mathcal{P}_{i-e}$ **do**
  $B_\ell := C_\ell / \pi_\ell;$

Step 3. Path dimensioning:

for *each source-destination pair* $\mathcal{P}_{i-e}$ *and* $p \in \mathcal{P}_{i-e}$ **do**
  $\mu_p := s \cdot \min_{\ell \in p} B_\ell;$

Step 4. Traffic distribution:

Run PmP algorithm to find the optimal traffic distribution;

Algorithm 4. *Bandwidth Sharing Algorithm*

The underlying algorithm is built around a design-based approach where the path set configurations correspond to different models of a parallel paths sets and the computation of the interference on links is done when finding the set of parallel paths for each source-destination pair using SVDP-S algorithm. In this context, the interference on link $\ell \in \mathcal{L}$ is defined by the number of paths traversing ℓ expressed by

$$\pi_\ell = \sum_{p \in \cup \mathcal{P}_{s-d}} \delta_{\ell,p} \tag{3.30}$$

where

$$\delta_{\ell,p} = \begin{cases} 1 & \text{if the path } p \text{ traverses the link } \ell \\ 0 & \textit{otherwise.} \end{cases} \tag{3.31}$$

The proposed design strategy achieves an optimal bandwidth allocation to the links of the network that leads to an optimal network flow configuration. This may be applied to support Virtual Private networks (VPNs) in emerging MPLS networks using a preplanned model where a set of precomputed LSPs is dimensioned based on a-priori estimation of a traffic matrix.

Furthermore, we note that this design may be executed based on a-priori knowledge of the performance of a given queueing model in term of switching times under different traffic profile.

3.2.3 Numerical illustration

This section presents numerical results obtained from applying the Bandwidth Sharing algorithm to engineer a 10-node test network where the incoming traffic to these two source-destination pairs $S1 - D1$ and $S2 - D2$ can be split over two routes as shown in the figure 3.5.

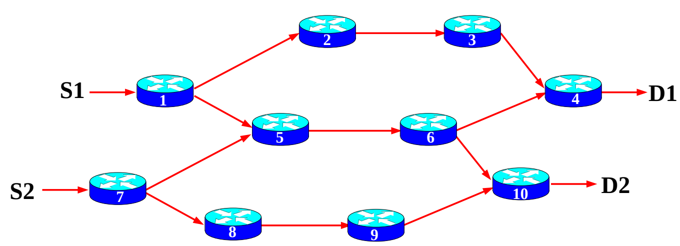


Figure 3.5: *Network Test*

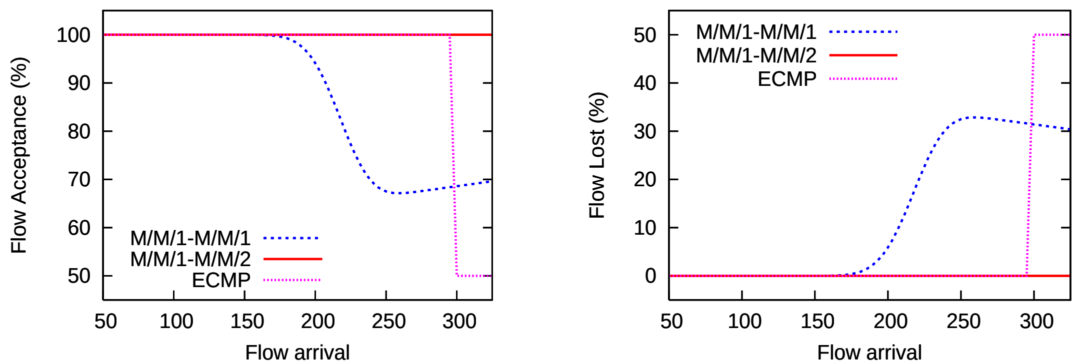
The parameter setting for the test network illustrated by Figure 3.5 are as follows.

Traffic is offered to 2 source-destination pairs modeled either as $M/M/1$ or $M/M/2$ queueing system. The maximum link capacity is set to $C = 300$ ub. The flow bandwidth requests vary in the interval $[50, 325]$. To meet the feasibility constraint, the traffic flow which are not within the link capacity limits are lost.

The objective is to protect the network from congestion by controlling the traffic at the bottleneck link $5 - 6$, where congestion may occur. We evaluated the efficiency of the network when using “differentiated queueing model” and “same queueing model” for all parallel paths sets and when using the Equal Cost Multi-Path routing (ECMP).

The performance parameters used are (1) the percentage flow acceptance referred to as

ACC, which is the percentage of flows which have been successfully forwarded to the destination (2) the shared link utilization referred to as $UTIL$ which defines the average link load and determines the potential for the network to support traffic growth and (3) the percentage of flow lost referred to as PFL, which is the percentage of flows which has been rejected by the shared link.



(a) *ECMP and OSPF-OMP Delay* (b) *ECMP and OMP Total Cost*

Figure 3.6: *ECMP and OSPF-OMP Delay*

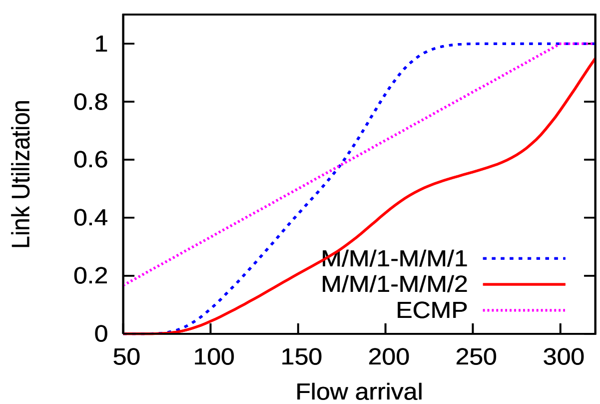


Figure 3.7: *Shared Link Utilization*

We have evaluated the performance achieved by the test network when modeling the sets of parallel paths using the *"same queueing model (M/M/1-M/M/1)"* and *"differentiated queueing model"*. Results depicted by figures 3.6(a), 3.6(b) and figure 3.7 show that using *"differentiated queueing (M/M/1-M/M/2)"* leads to better performance compared to using the *"same queueing model"* or ECMP.

3.3 Summary

This chapter presents our first dynamic multi-path routing scheme. We formulate the routing optimization as a multi-path routing problem over a set of parallel paths which is solved using Pontryagin Minimum Principle. We analyze the critical issue of how to select paths over which to disperse traffic flows, what proportion of traffic must be assigned to each selected path and when this traffic should be provisioned. Using Gunther approximation, we extend the initial PmP solution proposed by Filipiak [11] using an M/M/1 queueing model to the case where the links are modeled as M/M/s queueing systems. Using a "differentiated queueing" paradigm where the links of different source-destination pairs of a network are modeled using different queueing systems, we show through a small example performance improvements compared to the case where the links of each source-destination pair are modeled using the same queueing model.

Chapter 4

Game Routing Scheme

The Internet has evolved from an old closed network into an open communication network allowing competition between Internet Service Providers (ISPs), each seeking how to best route the flow of traffic offered by its subscribers. Thus, instead of a cooperative network where users follow rules defined by the agreed protocols, the current Internet can be modeled as a competitive network consisting of selfish users trying to optimize their own objectives expressed in terms of Quality of Service (QoS) requirements. As such, the routing of traffic in the Internet can be modeled as a $\Sigma-$players game $G = \left(\Sigma, (S_\imath)_{\imath\in\Sigma}, (\pi_\imath)_{\imath\in\Sigma}\right)$ where Σ is the set of players and $S_\imath$ is the strategies (actions) set of player $\imath \in \Sigma$. In such a game the rationality requires strict adherence of a strategy based on received and measured results given by $\pi_\imath(s)$ which is the payoff for player $\imath$ when the multistrategy $s \in S = \times_{\imath\in\Sigma} S_\imath$ is used with $s = (s_1, \ldots, s_\eta)$ a combination of actions chosen by all players. $\eta = |\Sigma|$ and $s_\imath$ is the the action arbitrarily chosen by player $\imath$ from its set of actions $S_\imath$. The outcome of these games, and consequently the overall network performance can be predicted using a game theoretic formulation. The choice of the optimal strategies by each player leads to the optimization of its payoff $\pi_\imath$ which is guaranteed by the fact that the players are selfish and rational. We consider in this chapter a game routing approach where the optimality concept is that of Nash equilibrium applied to the case of finite players. It is defined as follows

Definition 1. A combination of strategies $s^* = \left(s_1^*, \ldots, s_\eta^*\right) \in \times_{\imath\in\Sigma} S_\imath$ is a Nash equilibrium, if it satisfies [39]

$$\pi_\imath(s^*) \geq \pi_\imath\left(s_\imath, s^*_{-\imath}\right) \qquad \forall s_\imath \in S_\imath$$

where $\left(s_i, s^*_{-i}\right) \equiv \left(s^*_1, \ldots, s^*_{i-1}, s_i, s^*_{i+1}, \ldots, s^*_\eta\right)$ and $s^*_{-i} \equiv \left(s^*_1, \ldots, s^*_{i-1}, s^*_{i+1}, \ldots, s^*_\eta\right)$. This means that the player $i \in \Sigma$ has nothing to gain by deviating unilaterally from its policy s^*_i to another feasible policy. The Nash equilibrium is a profile of strategies such that each player's strategy is an optimal response to the other players' strategies.

There has been extensive studies on dynamic routing in distributed computer networks applying game theoretic approach. Akella et al. [40] apply game theory to the problems of congestion control. T. Roughgarden and E. Tardos [41], K. Yamaoka and Y. Sakai [42], E. Altaman et al. [43, 44], R. Azouzi et al. [45], R. J. La and V. Anantharam [46], have studied routing and flow control problem using game theoretic approach. The impact of selfish routing on the global performance of two nodes network with multiple parallel links was evaluated by Orda et al. in [47]. A. Korilis et al. [48] used a game theoretic framework to solve the bandwidth allocation problem while A. Lazar et al. [49] applied it in the context of virtual path bandwidth allocation for ATM networks.

Architecting Non-cooperative Networks [17] is the work most related to ours. It was proposed by A. Korolis et al. in a two nodes network with multiple parallel links where the links of the network are modeled as M/M/1 queues. However, the increase of data traffic in the Internet requires that optimization models of the Internet take into account routing metrics (link costs) which not only reflect the current resource availability such as the M/M/1 link cost but link costs which consider different other routing/reliability objectives, such as the interference among competing flows on a link, the length of the paths found in terms of hop count, etc. The main contribution of this chapter is to extend the work done in [17] from a two node network to a general topology using the LIOA link cost [16, 50].

In the following sections we examine the parallel paths modeling scheme for selfish users sharing resources of identical source-destination (ingress-egress) pair in section 4.1. We then formulate traffic distribution in multi-path settings as a flow optimization problem with the objective of minimizing the total (social) cost incurred by the network in section 4.2. This optimization problem is solved using classical nonlinear optimization methods. Finally we compare competitive and cooperative routing in section 4.3.

4.1 Competitive Game Routing

Consider a network represented by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ the set of links, each link having as maximum capacity C_ℓ, for $\ell \in \mathcal{L}$. Let $\mathcal{P}_{ie} = \{p_1, \ldots, p_M\}$ be a set of M parallel or disjoint paths connecting a given ingress-egress (i, e) pair. Consider N users sharing common network resources between an ingress-egress (i, e) pair. Let $\Lambda_\imath > 0$ be the total flow offered by user $\imath$, $\imath = 1, \ldots, N$. We assume that the user $\imath$ sends his flow by splitting traffic among M available paths to achieve individual performance, *i.e.*, minimum cost in our context. Let $\lambda_\imath^m$ denotes the portion of traffic that the user $\imath$ sends over path p_m, $1 \leq m \leq M$; and $\lambda_{-\imath}^m$ the traffic rate of other users over the same path p_m, *i.e.*, $\lambda_{-\imath}^m = \lambda^m - \lambda_\imath^m$ where λ^m is the total rate of traffic sent over the path p_m. The individual performance of user $\imath$ is conditioned by the feasibility constraints expressed by

$$\Lambda_\imath = \sum_{m=1}^{M} \lambda_\imath^m \quad \textit{and} \quad 0 \leq \lambda_\imath^m < c_m, \tag{4.1}$$

where c_m is the capacity of the path p_m as defined in section 2.1 of section 2. The total flow of all users denoted Λ is given by

$$\Lambda = \sum_{\imath=1}^{N} \Lambda_\imath \quad \textit{or} \quad \Lambda = \sum_{m=1}^{M} \lambda^m. \tag{4.2}$$

Furthermore, the capacity configuration $c = (c_1, \ldots, c_M)$ must be able to absorb the total users demand. This is referred to as the stability constraint expressed by

$$\sum_{\imath=1}^{N} \Lambda_\imath < \sum_{m=1}^{M} c_m \quad \textit{or} \quad \Lambda < C, \tag{4.3}$$

where C is the aggregate bandwidth.

The configuration $\lambda_\imath = \left(\lambda_\imath^1, \ldots, \lambda_\imath^M\right)$ of user $\imath$ that satisfies the feasibility constraints (4.1) is called a routing strategy of user $\imath$, and the set of routing strategies for user $\imath$ denoted $\mathcal{F}_\imath$ given by

$$\mathcal{F}_\imath = \{\lambda_\imath \in \mathbb{R}^M : 0 \leq \lambda_\imath^m < c_m, m = 1, \ldots, M; \ \textit{and} \ \sum_{m=1}^{M} \lambda_\imath^m = \Lambda_\imath\}, \tag{4.4}$$

is called the strategy space of user $\imath$. Without lost of generality we assume that each user $\imath$ needs to find the "best" strategy that minimizes its individual cost denoted by $J_\imath(\lambda)$ with $\lambda = \left(\lambda^1, \ldots, \lambda^M\right)$. The network studied is a non-cooperative network, *i.e.*, network which consists of selfish users, who determine the rate at which traffic is sent, each user having his own cost.

The cost per unit of traffic incurred by the user $\imath$ while using the path p_m is a function of the total traffic rate λ^m sent over p_m denoted $\psi_m\left(\lambda^m\right)$ and represents typically the expected delay over the path p_m. Thus, the cost for user $\imath$ (per unit of time) of sending at rate $\lambda_\imath^m$ over path p_m is $\lambda_\imath^m \psi_m\left(\lambda^m\right)$, and the total cost incurred by the user $\imath$ while sending his traffic through the network is given by

$$J_\imath\left(\lambda\right) = \sum_{m=1}^{M} \lambda_\imath^m \psi_m\left(\lambda^m\right) \tag{4.5}$$

4.1.1 Problem formulation

The game routing problem consists of finding for each user $\imath$, $1 \leq \imath \leq N$ an optimal strategy (flow allocation) that solves the following optimization problem.

$$\begin{aligned} &\min\ J_\imath(\lambda) \\ \text{subject to } &\sum_{m=1}^{M} \lambda_\imath^m = \Lambda_\imath \\ &\text{over } \lambda_\imath^m \geq 0,\ m = 1, \ldots, M, \end{aligned} \tag{4.6}$$

where $\psi_m : \mathbb{R} \longrightarrow [0, +\infty]$, $\lambda^m \mapsto \psi_m(\lambda^m)$, $m = 1, \ldots, M$ are increasing, convex and continuous.

Broadly speaking, the cost functions used in optimization problems are related to some performance measures. In our study we use a cost function which combines two objectives, namely QoS routing under constraints in term of bandwidth utilization maximization and interference among competing flows minimization [50], referred to as LIOA penalty function [16]. It is expressed by

$$L\left(\pi_\ell, f_\ell\right) = \begin{cases} \frac{\pi_\ell^\alpha}{(C_\ell - f_\ell)^{1-\alpha}} & \text{if } f_\ell < C_\ell \\ \infty & otherwise, \end{cases} \tag{4.7}$$

where f_ℓ is the flow carried by the link ℓ and π_ℓ is the number of flows carried by the link ℓ referred to as interference on link ℓ. The calibration parameter α, with $0 \leq \alpha \leq 1$, balances the impact of the two constraints (bandwidth and interference) on the link cost.

Note that according to the value of α, four following important latency functions L can be derived from the LIOA penalty function.

$$L\left(\pi_\ell, f_\ell\right) = \begin{cases} \frac{1}{C_\ell - f_\ell} & \text{for } \alpha = 0 \quad (1) \\ \frac{1}{C_\ell} & \text{for } \alpha = f_\ell = 0 \quad (2) \\ \pi_\ell & \text{for } \alpha = 1 \quad (3) \\ 1 & \text{for } \alpha = \pi_\ell = 1 \quad (4) \end{cases} \tag{4.8}$$

In the LIOA penalty function, equation (1) describes the M/M/1 queue delay function and corresponds to Constraints Shortest Path First (CSPF) routing. Equations (2), (3) and (4) express respectively the link cost used in Open Shortest Path First (OSPF) routing as proposed by Cisco [1], in Interference Optimization (IOPT) routing, and in minimum hop-routing. When $0 < \alpha < 1$ the link cost function yields a mix of (IOPT) and (CSPF) routing, and the stability constraint of link ℓ is manifested through LIOA cost function given in (4.7).

The cost per unit of traffic incurred by the user i while sending traffic over path p_m is given by

$$\psi_m\left(\lambda^m\right) = \sum_{\ell \in p_m} \frac{\pi_\ell^\alpha}{\left(C_\ell - f_\ell\right)^{1-\alpha}} \tag{4.9}$$

Let λ be the incoming flow at a source s to be forwarded to the destination d, through a given set of paths $\mathcal{P}_{s-d}$ in the underlying network, and $\delta_{\ell,p}$ the link-path indicator function, *i.e.*,

$$\delta_{\ell,p} = \begin{cases} 1 & \text{if the path } p \text{ traverses the link } \ell \\ 0 & \textit{otherwise}, \end{cases} \tag{4.10}$$

the flow decomposition theorem of network flows from [51, 52] indicates that some optimal link flow f_ℓ can be always decomposed into path flow $f(p)$ by

$$f_\ell = \sum_{p \in \mathcal{P}_{s-d}} \delta_{\ell,p} f(p) \tag{4.11}$$

As we are dealing with the set of parallel paths, the flow f_ℓ carried by the link $\ell \in p_m$ is equal to the total flow traversing the path p_m, and given by

$$\lambda^m = \sum_{\imath=1}^{N} \lambda_\imath^m \quad for \quad m = 1, \ldots, M \tag{4.12}$$

and thus, the cost over path p_m is

$$\psi_m\left(\lambda^m\right) = \sum_{\ell \in p_m} \frac{\pi_\ell^\alpha}{\left(C_\ell - \lambda^m\right)^{1-\alpha}} \tag{4.13}$$

which is continuous, increasing and convex. Finally, the total cost $J_\imath\left(\lambda\right)$ incurred by the user $\imath$ is

$$J_\imath\left(\lambda\right) = \sum_{m=1}^{M} \left[\sum_{\ell \in p_m} \frac{\pi_\ell^\alpha}{\left(C_\ell - \lambda^m\right)^{1-\alpha}}\right] \lambda_\imath^m, \tag{4.14}$$

which is, in fact, the overall cost of routing user $\imath$ traffic flows in the network. Note that $J_\imath(\lambda)$ is differentiable in its domain $\mathcal{F}_\imath$.

Problem 4.1. *The selfish or competitive flow allocation problem consists of finding for each user $\imath$, $\imath = 1, \ldots, N$ the strategy $\lambda_\imath^* = \left(\lambda_\imath^1, \ldots, \lambda_\imath^M\right)$ solution of the following optimization problem.*

$$\min \sum_{m=1}^{M} \left[\sum_{\ell \in p_m} \frac{\pi_\ell^\alpha}{\left(C_\ell - \lambda^m\right)^{1-\alpha}}\right] \lambda_\imath^m \tag{4.15}$$

$$\textit{subject to} \quad \sum_{m=1}^{M} \lambda_\imath^m = \Lambda_i \tag{4.16}$$

$$\lambda_\imath^m \geq 0, \; m = 1, \ldots, M. \tag{4.17}$$

★

4.1.2 Solving the optimization problem

From convex optimization problem [53], it follows that there exists an unique allocation rate (Nash equilibrium) denoted $\lambda_\imath^* = \left(\lambda_\imath^1, \ldots, \lambda_\imath^M\right)$ which is the optimal solution to the problem 4.1. According to Karush-Kuhn-Tucker (KKT) optimality conditions, this unique allocation is in fact the critical point of the Lagrangian function $\phi : \mathbb{R}^M \times \mathbb{R} \longrightarrow \mathbb{R}$

associated with the problem (4.15)-(4.17) given by

$$\phi\left(\lambda_{\imath}, \mu\right) = \sum_{m=1}^{M}\left[\sum_{\ell\in p_m}\frac{\pi_{\ell}^{\alpha}}{\left(C_{\ell}-\lambda^{m}\right)^{1-\alpha}}\right]\lambda_{\imath}^{m} - \mu\Big(\sum_{m=1}^{M}\lambda_{\imath}^{m}-\Lambda_{\imath}\Big), \tag{4.18}$$

with domain $\mathcal{D} = \mathcal{F}_{\imath}\times\mathbb{R}$, where $\mathcal{F}_{\imath}$ is given in (4.4) and we refer to μ as the Lagrangian multiplier associated with the constraint (4.16).

We thus solve our constrained problem (4.15)-(4.17) by finding the stationary or critical point of ϕ regarded as a function of $M+1$ independent variables $\left(\lambda_{\imath}^{1},\ldots,\lambda_{\imath}^{M},\mu\right)$. This means that we have to solve the following non-linear system of equations.

$$\begin{aligned} \nabla J_{\imath}(\lambda) &= \mu\nabla g(\lambda) \\ \sum_{m=1}^{M}\lambda_{\imath}^{m} &= \Lambda_{\imath} \end{aligned} \tag{4.19}$$

or equivalently

$$\begin{aligned} \left[\frac{\partial J_{\imath}(\lambda)}{\partial\lambda_{\imath}^{1}},\ldots,\frac{\partial J_{\imath}(\lambda)}{\partial\lambda_{\imath}^{M}}\right] &= \mu\left[\frac{\partial g(\lambda)}{\partial\lambda_{\imath}^{1}},\ldots,\frac{\partial g(\lambda)}{\partial\lambda_{\imath}^{M}}\right] \\ \sum_{m=1}^{M}\lambda_{\imath}^{m} &= \Lambda_{\imath} \end{aligned} \tag{4.20}$$

where $J_{\imath}(\lambda)$ is given in (4.14) and $g(\lambda) = \sum_{m=1}^{M}\lambda_{\imath}^{m} - \Lambda_{\imath}$. Differentiating $J_{\imath}(\lambda)$ with respect to $\lambda_{\imath}^{m}$ leads to the following expression.

$$\frac{\partial J_{\imath}(\lambda)}{\partial\lambda_{\imath}^{m}} = \sum_{\ell\in p_m}\frac{\pi_{\ell}^{\alpha}\left(C_{\ell}-\lambda_{-\imath}^{m}-\alpha\lambda_{\imath}^{m}\right)}{\left(C_{\ell}-\lambda^{m}\right)^{2-\alpha}} \quad for \quad m=1,\ldots,M. \tag{4.21}$$

Therefore, the nonlinear system of equations (4.19) is equivalent to the following nonlinear system.

$$\begin{aligned} \sum_{\ell\in p_m}\frac{\pi_{\ell}^{\alpha}\left(C_{\ell}-\lambda_{-\imath}^{m}-\alpha\lambda_{\imath}^{m}\right)}{\left(C_{\ell}-\lambda^{m}\right)^{2-\alpha}} &= \mu \quad for \quad m=1,\ldots,M \\ \sum_{m=1}^{M}\lambda_{\imath}^{m} &= \Lambda_{\imath}. \end{aligned} \tag{4.22}$$

Indeed, equations (4.22) is the system of nonlinear equations with $M+1$ equations and $M+1$ unknowns variables. Thereby we are now in a position to compute the optimal response $\lambda_\imath^* = \left(\lambda_\imath^1, \ldots, \lambda_\imath^M\right)$ of user $\imath$. Furthermore, the user $\imath$ has to determine only $M-1$ decisions variables $\lambda_\imath^1, \ldots, \lambda_\imath^{M-1}$, the value of $\lambda_\imath^M$ will be then deduce from the feasibility constraint. Thus, the strategy $\lambda_\imath^*$ of user $\imath$, $1 \leq \imath \leq N$, is the optimal response if it solves the following non-linear system of M equations with M unknowns.

$$\begin{aligned} \sum_{\ell \in p_m} \frac{\pi_\ell^\alpha \left(C_\ell - \lambda_{-\imath}^m - \alpha\lambda_\imath^m\right)}{\left(C_\ell - \lambda^m\right)^{2-\alpha}} - \sum_{\ell \in p_M} \frac{\pi_\ell^\alpha \left(C_\ell - \lambda_{-\imath}^M - \alpha\lambda_\imath^M\right)}{\left(C_\ell - \lambda^M\right)^{2-\alpha}} = 0 \quad for \quad m = 1, \ldots, M-1 \\ \sum_{m=1}^{M} \lambda_\imath^m - \Lambda_\imath = 0. \end{aligned} \tag{4.23}$$

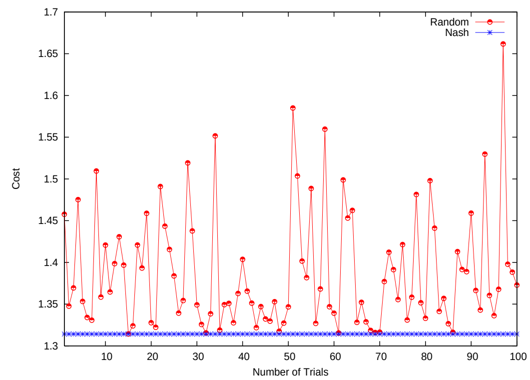
4.1.3 Numerical solution

The unique solution to the nonlinear system of $N \times M$ equations with $N \times M$ unknowns can be found using the Newton-Raphson Method for Nonlinear Systems of Equations [54]. However, it is well-known that the Newton-Raphson method raises the issue of the identification of the neighborhood of a root, *i.e.*, the problem of finding the starting point [54, 55].

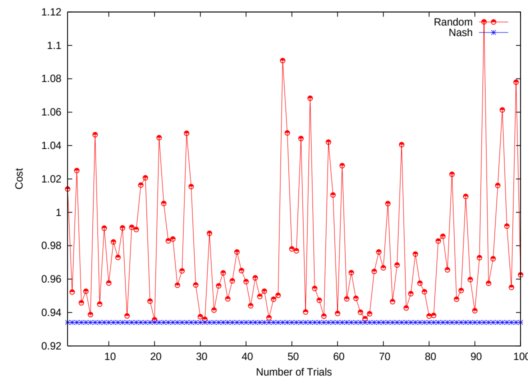
We deal with the problem of starting point intuitively. Indeed, in competitive or selfish routing environment, each user tries to optimize his own performance. In our case, each user aims at minimizing the delay of his flow's sojourn in the network by using the minimum delay path from ingress to egress node. Thus, we number paths in increasing order of their delays, *i.e.*, the path p_1 is the "best" path (path with minimum delay); and at the beginning we suppose that all the users attempt to ship their traffic flow over the paths in an inversely proportional way to the cost incurred by the traffic along the paths. This means that for each user $\imath$, $1 \leq \imath \leq N$ the starting point must be set as follows.

$$\lambda_\imath^m = \frac{\Lambda_\imath \cdot \mu_m}{\mu} \quad for \quad m = 1, \ldots, M \tag{4.24}$$

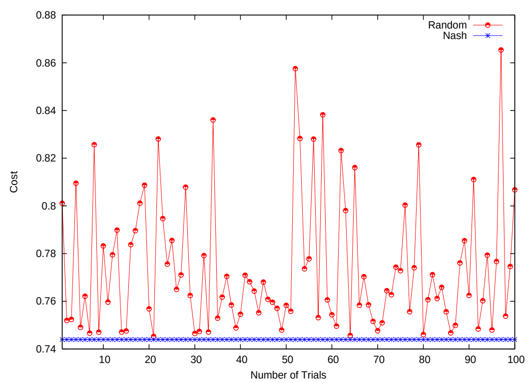
where $1/\mu_m$ is the cost incurred over the path p_m and μ is the sum of μ_m. Looking at the nonlinear system of equations (4.23), the optimal response requires from each user to know only the sum of flows on each link, and not their individual value, which makes implementation easier. Knowing his delay on each link and his own flow, a user is able to determine the sum of flows of other users on each link. Of course with the initialization (4.24), the system is not in equilibrium and applying Newton-Raphson algorithm the system converges



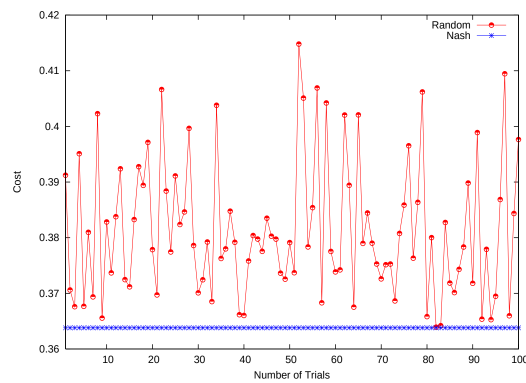
(a) *Cost achieved by user* 1



(b) *Cost achieved by user* 2



(c) *Cost achieved by user* 3



(d) *Cost achieved by user* 4

Figure 4.1: *Illustration of Nash flow configuration for* 4 *users*

to the equilibrium and once the traffic equilibrium, known as Nash equilibrium, is reached the system will remain in the equilibrium.

The uniqueness of an equilibrium is quite a desirable property, if we wish to predict what will be the network behavior. This is particularly important in the context of network administration and management, where one is interested in optimally setting the network design parameters, taking into account their impact on the performance at the equilibrium.

As illustration we consider the network described by figure 2.3 of section 2.2, with 4 selfish users requesting respectively 35, 25, 20, and 10 ub from source S to the destination D. We assume that each link is modeled as an M/M/1 queueing system, *i.e.*, we set $\alpha = 0$ in LIOA penalty function.

Figure 4.1 shows the cost achieved by each user when using random and Nash flow configuration. The random flow configuration is a network configuration where traffic is distributed randomly to the network. As expected, the Nash flow configuration always achieves the "best" cost compared to the random routing strategy.

4.2 Cooperative Game Routing

While in competitive routing each user tries to find the best routing strategy that minimizes its individual cost, it is common practice for network operators to minimize the total (social) cost. This is achieved by controlling the total flow of traffic in the network and routing the offered traffic to achieve a network configuration that minimize the total (social) cost expressed by

$$J(\lambda) = \sum_{\imath=1}^{N} J_{\imath}(\lambda). \tag{4.25}$$

We want to set up the optimal routing policy that is obtained when there is one actor, namely network operator, with the offered traffic Λ. By replacing the cost $J_{\imath}(\lambda)$ in (4.25) by its expression given in (4.14), we have

$$J(\lambda) = \sum_{m=1}^{M} \left[\sum_{\ell \in p_m} \frac{\pi_\ell^\alpha}{(C_\ell - \lambda^m)^{1-\alpha}} \right] \lambda^m. \tag{4.26}$$

Similarly to the previous problem, we can formulate the problem of optimally routing the total requested bandwidth Λ over the set of parallel routes $\mathcal{P}_{ie}$.

4.2.1 Problem formulation

Problem 4.2. *The cooperative flow allocation problem consists of finding a paths flow configuration* $\lambda^{opt} = \left(\lambda^1, \ldots, \lambda^M\right)$ *solution of the following optimization problem.*

$$\min \sum_{m=1}^{M} \left[\sum_{\ell \in p_m} \frac{\pi_\ell^\alpha}{\left(C_\ell - \lambda^m\right)^{1-\alpha}} \right] \lambda^m \tag{4.27}$$

$$\textit{subject to} \quad \sum_{m=1}^{M} \lambda^m = \Lambda \tag{4.28}$$

$$\lambda^m \geq 0, \;\; m = 1, \cdots, M. \tag{4.29}$$

★

As before, according to KKT optimality conditions, the unique paths flow configuration solution to the optimization problem (4.2) is the critical or stationary point of the Lagrangian function $\phi : \mathbb{R}^M \times \mathbb{R} \longrightarrow \mathbb{R}$ associated with the problem (4.27)-(4.29) given by

$$\phi\left(\lambda, \mu\right) = \sum_{m=1}^{M} \left[\sum_{\ell \in p_m} \frac{\pi_\ell^\alpha}{\left(C_\ell - \lambda^m\right)^{1-\alpha}} \right] \lambda^m - \mu \Big(\sum_{m=1}^{M} \lambda^m - \Lambda \Big), \tag{4.30}$$

with domain $\mathcal{D} = \mathcal{F} \times \mathbb{R}$, where

$$\mathcal{F} = \{\lambda \in \mathbb{R}^M : 0 \leq \lambda^m < c_m, m = 1, \ldots, M; \; \textit{and} \; \sum_{m=1}^{M} \lambda^m = \Lambda\},$$

and we refer to μ as the Lagrangian multiplier associated with the constraint (4.28).

Thus, the strategy that leads to the optimal social cost is obtained by solving the following nonlinear system of equations

$$\begin{aligned} \sum_{\ell \in p_m} \frac{\pi_\ell^\alpha \left(C_\ell - \alpha \lambda^m\right)}{\left(C_\ell - \lambda^m\right)^{2-\alpha}} - \sum_{\ell \in p_M} \frac{\pi_\ell^\alpha \left(C_\ell - \alpha \lambda^M\right)}{\left(C_\ell - \lambda^M\right)^{2-\alpha}} = 0 \quad for \quad m = 1, \ldots, M-1 \\ \sum_{m=1}^{M} \lambda^m - \Lambda = 0. \end{aligned} \tag{4.31}$$

We thus apply the Newton-Raphson method for nonlinear system of equations to solve the system of equations (4.31) to find the paths flow configuration $\lambda = \left(\lambda^1, \ldots, \lambda^M\right)$ which leads to the optimal social cost. This is done by applying the same reasoning as previously to choose the starting point.

Considering the same illustration as in case of 4 selfish users but supposing that the network operator is the unique user of the system who is sending the traffic generated by the users of the network. This means that network operator have to send 90 ub requested from the source S to the destination D.

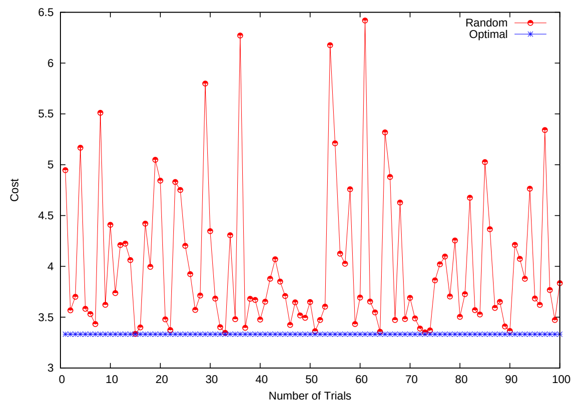


Figure 4.2: *Illustration of Optimal flow configuration*

Figure 4.2 shows the cost incurred by the system when using random and the "optimal" flow configuration. The "optimal" flow configuration leads effectively to the "best" total or social cost compared to the random routing strategy.

4.2.2 Cooperative Routing algorithm

In this section, we propose a routing algorithm to route flow of λ_{s-d} bandwidth units between 2 nodes s and d in MPLS networks. A high level description of this algorithm consists of 2 steps, namely

1. Finding a set of parallel paths connecting the the source s and destination d by deploying the Shortest Vertex-Disjoint Paths Set (SVDP-S) algorithm proposed in section 2.3 of chapter 2.

2. Distributing traffic to these parallel paths using cooperative flow allocation procedure developed in the above section.

The complete description of the algorithm is given below.

Step 1. Find a set $\mathcal{P}_{s-d}$ of paths by using Shortest Vertex-Disjoint Paths Set algorithm:
(a): $\mathcal{P}_{s-d} := SVDP-S(\mathcal{G}, s, d)$;
(b): *Compute the aggregate bandwidth* $C = \sum_{m=1}^{M} c_m$;
Step 2. Route the request λ_{s-d} :
(a): **if** $\lambda_{s-d} < C$ **then**
(b): - *Update the link interference* π_ℓ;
(c): **for** *each* $\ell \in p$ *with* $p \in \mathcal{P}_{s-d}$ **do**
(d): $\pi_\ell := \pi_\ell + 1$;
(e): - *Traffic splitting: Computation of flow pattern* $\left(\lambda^1, \ldots, \lambda^M\right)$;
(f): *Initialization of flow pattern*
(g): **for** *each* $1 \le m \le M$ **do**
(h): $\lambda^m := \frac{\lambda_{s-d} \cdot \mu_m}{\mu}$;
(i): *Run Newton-Raphson to solve the nonlinear system of equations (4.31);*
(j): - *Traffic allocation: Assign each portion* λ^m *to the corresponding path* $p_m \in \mathcal{G}$;
(k): - *Decrease the link interference* π_ℓ *on no used paths;*
(l): **for** *each* $\ell \in p_m$ *with* $\lambda^m = 0$ **do**
(m): $\pi_\ell := \pi_\ell - 1$;
Step 3. Update the link flows f_ℓ.
(a): **for** *each* $\ell \in p_m$ **do**
(b): $f_\ell := f_\ell + \lambda^m$;

Algorithm 5. *Cooperative (Game) Routing Algorithm*

The efficiency of the proposed routing algorithm is evaluated in chapter 5 where the routing algorithm is used to engineer an European and USA networks.

Table 4.1: *Nash and Optimal Flow Configurations Costs Comparison*

User	Routing Strategy and Cost Incurred				
	p_1	p_2	p_3	Tot (Λ^i)	Cost
user 1	14.4729	10.2626	10.2626	35.0	1.314230
user 2	11.5706	6.71470	6.71470	25.0	0.934057
user 3	10.1198	4.94009	4.94009	20.0	0.743968
user 4	7.21822	1.39089	1.39089	10.0	0.363795
Competitive	43.38152	23.30828	23.30828	90.0	3.356050
Cooperative	39.1217	25.4392	25.4392	90.0	3.332360

4.3 Comparing Competitive and Cooperative Routing

In Table 4.1, we presents the results found when the system is used by selfish users (competitive routing) and by the network operator (cooperative routing). p_1 represents the path $1-4-7$, p_2 the path $1-2-5-7$ and p_3 represents the path $1-3-6-7$.

We observe that the competitive flow, *i.e.*, flow at Nash equilibrium and the cooperative flow configurations are not equal. This shows that traffic flow at Nash equilibrium do not in general minimize the total latency. An immediate question that arises is how far the flow at Nash equilibrium, is from the optimal paths flow configuration in terms of network objective or what is the worst ratio between the total latency of a flow at Nash equilibrium and that of the best coordinated outcome of a flow minimizing the total latency. The proposition 4.1 [28] provides important upper bound of the ratio ρ between the cost of a routing at Nash equilibrium and of routing using a minimum-latency model by showing the relation between the competitive cost $J(\lambda^*)$ and the cooperative cost $J(\lambda^{opt})$.

Proposition 4.1. *Consider the latency function $L_\ell(\cdot)$ and the constant $\gamma \geq 1$ that satisfy*

$$x \cdot L_\ell(x) \leq \gamma \cdot \int_0^x L_\ell(t)dt \tag{4.32}$$

for all links ℓ and all positive real number x. Then the ratio of the total cost of flow at Nash equilibrium λ^* to that of system optimality λ^{opt} is bounded from above by γ, i.e.,

$$\frac{J\left(\lambda^*\right)}{J\left(\lambda^{opt}\right)} \leq \gamma \quad or \quad J\left(\lambda^*\right) \leq \gamma \cdot J\left(\lambda^{opt}\right) \tag{4.33}$$

Considering LIOA penalty function L defined by (4.7), applying proposition 4.1 as expressed by the condition (4.32) leads to

$$\frac{\pi_\ell^\alpha f_\ell}{(C_\ell - f_\ell)^{1-\alpha}} \leq \gamma \cdot \int_0^{f_\ell} \frac{\pi_\ell^\alpha}{(C_\ell - t)^{1-\alpha}} dt \tag{4.34}$$

- Case 1: For $\alpha = 1$, the following equality is satisfied

$$f_\ell \cdot L(\pi_\ell, f_\ell) = \pi_\ell \cdot f_\ell = 1 \cdot \int_0^{f_\ell} \pi_\ell dt = 1 \cdot \int_0^{f_\ell} L(\pi_\ell, t)\, dt.$$

This means that $\gamma = 1$, and from the proposition 4.1 it follows that $J(\lambda^*) = J(\lambda^{opt})$. Therefore, flow at Nash equilibrium is equal to the optimal flow when $\alpha = 1$.

- Case 2: For $0 < \alpha < 1$, the relation (4.34) becomes

$$\frac{\beta_\ell}{(1-\beta_\ell)^{1-\alpha}} \leq \gamma \cdot \left(\frac{1-(1-\beta_\ell)^\alpha}{\alpha} \right) \tag{4.35}$$

after simplification where $\beta_\ell = f_\ell / C_\ell$ is the link utilization for link ℓ.

- Case 3: For $\alpha = 0$, the requirement (4.32) is expressed by

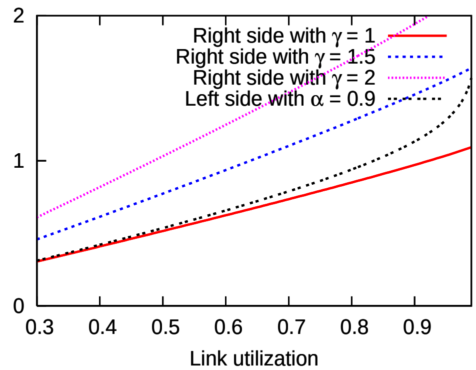
$$\frac{\beta_\ell}{(1-\beta_\ell)} \leq -\gamma \cdot \ln(1-\beta_\ell). \tag{4.36}$$

Case 2 is analyzed using the figures 4.3(a)-4.3(e). These figures plot the left side of the relation (4.35) and its right side for four different values of α ($\alpha = 0.9, 0.6, 0.5, 0.4, 0.1$) and three different values of γ ($\gamma = 1, 1.5, 2$). Similarly, figure 4.3(f) plots the left side of relation (4.36) and its right side for the case where $\alpha = 0$.

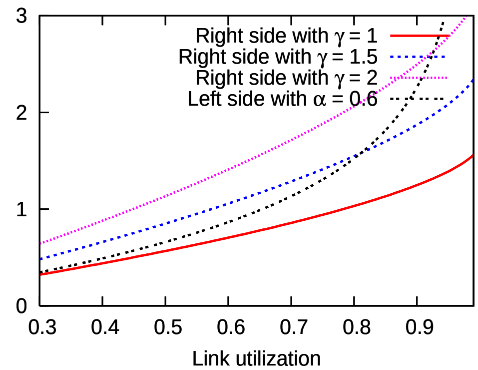
From these figures, we made the following observations:

- For $\alpha = 0.9$, the requirement (4.32) always holds when $\gamma = 1.5$ provided that $\beta_\ell < 1$.
- For $\alpha = 0.6$, the requirement is satisfied if the link utilization is maintained below 80% when $\gamma = 1.5$, and if γ increased to 2, the requirement is satisfied at a higher utilization of 90%. From the proposition 4.1, this means that

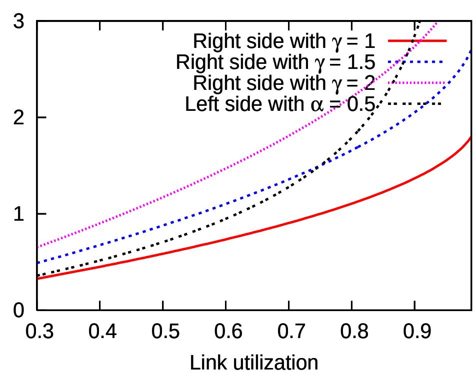
  - If $\beta_\ell \leq 0.8$ for all link ℓ then $J(\lambda^*) \leq 1.5 \cdot J(\lambda^{opt})$. And if $\beta_\ell \leq 0.9$ for all link ℓ then $J(\lambda^*) \leq 2 \cdot J(\lambda^{opt})$.



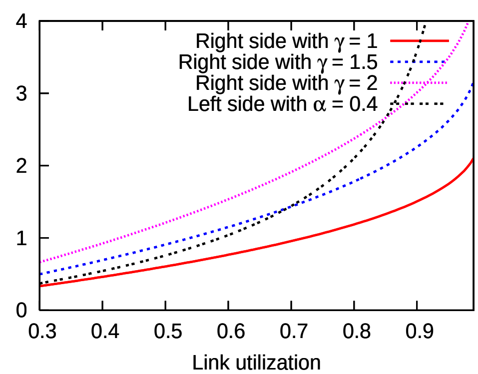
(a) *Bound comparison for* $\alpha = 0.9$



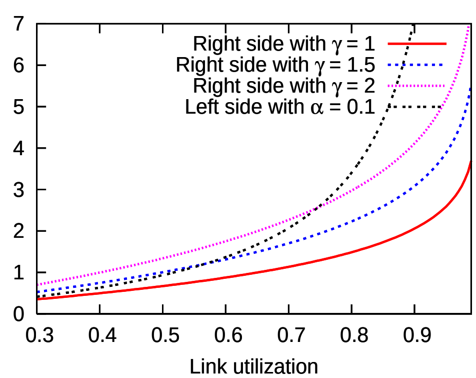
(b) *Bound comparison for* $\alpha = 0.6$



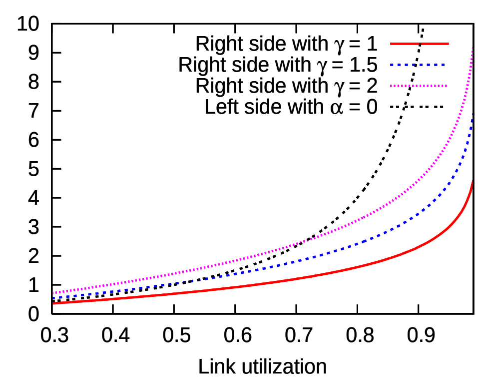
(c) *Bound comparison for* $\alpha = 0.5$



(d) *Bound comparison for* $\alpha = 0.4$



(e) *Bound comparison for* $\alpha = 0.1$



(f) *Bound comparison for* $\alpha = 0$

Figure 4.3: *Bound of the ratio* ρ *between Nash and Optimal cost*

- For $\alpha = 0.5$, the requirement is satisfied if the link utilization is maintained below 70% when $\gamma = 1.5$, and below 85% when $\gamma = 2$ meaning that if $\beta_\ell \leq 0.7$ for all link ℓ then $J(\lambda^*) \leq 1.5 \cdot J(\lambda^{opt})$. And if $\beta_\ell \leq 0.85$ for all link ℓ then $J(\lambda^*) \leq 2 \cdot J(\lambda^{opt})$.
- For $\alpha = 0.4$, the requirement is satisfied if the link utilization is maintained below 70% when $\gamma = 1.5$, and below 80% when $\gamma = 2$ or equivalently if $\beta_\ell \leq 0.7$ for all link ℓ then $J(\lambda^*) \leq 1.5 \cdot J(\lambda^{opt})$. And if $\beta_\ell \leq 0.8$ for all link ℓ then $J(\lambda^*) \leq 2 \cdot J(\lambda^{opt})$.
- For $\alpha = 0.1$, the requirement is satisfied if the link utilization is maintained below 55% when $\gamma = 1.5$, and below 70% when $\gamma = 2$, *i.e.*, if $\beta_\ell \leq 0.55$ for all link ℓ then $J(\lambda^*) \leq 1.5 \cdot J(\lambda^{opt})$. And if $\beta_\ell \leq 0.7$ for all link ℓ then $J(\lambda^*) \leq 2 \cdot J(\lambda^{opt})$.
- For $\alpha = 0$, the requirement is satisfied if the link utilization is maintained below 50% when $\gamma = 1.5$, and below 70% when $\gamma = 2$. This means that If $\beta_\ell \leq 0.5$ for all link ℓ then $J(\lambda^*) \leq 1.5 \cdot J(\lambda^{opt})$. And if $\beta_\ell \leq 0.7$ for all link ℓ then $J(\lambda^*) \leq 2 \cdot J(\lambda^{opt})$.

The results above reveal that a network which is engineered using the LIOA cost function can achieve competitive routing at the cost of cooperative routing for $\alpha = 1$. Competitive routing performs worse than cooperative routing for values of $\alpha \neq 1$. The difference between the two routing models depends on the network load. Further details on this issue are beyond the scope of this thesis.

4.4 Summary

We propose in this chapter a new dynamic routing scheme which improves the overall performance achieved by traffic flows during their sojourn in the network. We formulate the routing of traffic flows as both competitive and cooperative routing problems. Using exact methods borrowed from the nonlinear optimization framework, we solve these two problems using the Newton Raphson method. We compared the cost achieved by competitive multi-path routing and cooperative multi-path routing schemes. The results obtained for a small network consisting of 7 nodes and 12 links shows that in multi-path settings, the cost achieved by competitive routing can be far from the cost achieved by cooperative routing. Using different link utilizations and values of the calibration parameter α, we show how far the cost of competitive routing can be from the cooperative routing cost.

Chapter 5

Quantitative Analysis

We have developed in the two previous chapters two schemes to solve the flow allocation problem in multi-path routing: A Pontryagin and a Game routing schemes both used to achieve optimal traffic flow allocation over a set of parallel paths. These routing schemes may be integrated as traffic distribution engines in a two-step routing approach using path selection and traffic distribution to achieve load balancing in emerging generation IP networks such as MPLS networks. In the present chapter, we evaluate the performance of the two schemes when used to route the traffic offered to larger network topologies such as fictitious Europe and United States (USA) networks depicted by Figures 5.1(a) and 5.1(b). Experimental results under different traffic load conditions are compared with those resulting from the well-known Flow Deviation (FD) Algorithm when the network links are modeled as M/M/1 queues.

5.1 Using PmP and Game algorithms in general networks

Given a network we consider the routing of traffic offered to this network using four steps described by the algorithm 6 below.

Step 1. Initialization of link interferences:
(a): **for** *all* $\ell \in \mathcal{L}$ **do**
(b): $\pi_\ell := 0;$
Step 2. Update the network topology:
(a): **for** *all* $\ell \in \mathcal{L}$ **do**
(b): $B_\ell := C_\ell - f_\ell$; *//find the residual (or available) bandwidth* B_ℓ
Step 3. Traffic distribution:
(a): *Run Pontryagin or Game Routing algorithm to distribute traffic;*
Step 4. For each source-destination pair under consideration:
(a): *Repeat the procedure outlined in the step 2, and 3;*

Algorithm 6. *Flow Allocation Algorithm*

5.2 Overview of Flow Deviation Algorithm

In the MPLS context, the Flow Deviation Algorithm incrementally improves the set $\mathcal{P}$ of paths and improves the distribution of traffic over multiple paths in $\mathcal{P}$ from the same source source and the same destination [56]. In our case, as the set of paths is given, the Flow Deviation Algorithm is used as a proactive algorithm that optimally balances the traffic load over a precomputed set of parallel paths between source-destination pairs of a network.

As the link cost functions considered in this thesis are convex, the well-known result from [57] states that the optimal routing results from the fact that flow travels along Minimum First Derivative Length (MFDL) paths for each ingress-egress pair. This means that the "best" strategy to improve the distribution of the traffic flow over a given set of paths would be to iteratively deviate flow from non MFDL paths to MFDL paths.

Indeed, let consider the set $\mathcal{P}_{s-d}$ of disjoint paths between a given source-destination (s-d) pair. Let $J_m(\lambda^m)$ the cost on path $p_m \in \mathcal{P}_{s-d}$ when λ^m amount of traffic flow is routed on the path p_m and $J_m(\lambda^m + \varepsilon)$ be the new cost when the amount ε of traffic is added to the path. Then $J_m(\lambda^m + \varepsilon) - J_m(\lambda^m)$ can be estimated as $\varepsilon J'_m(\lambda^m) + o(\varepsilon)$ to first order and the total cost $J(\lambda)$ would increase with the amount of increase given by

$$\Delta J = J_m(\lambda^m + \varepsilon) - J_m(\lambda^m) \quad (5.1)$$

$$= \varepsilon J'_m(\lambda^m) + o(\varepsilon), \quad (5.2)$$

with $\lim_{\varepsilon \to 0} o(\varepsilon) = 0$.

Thus, by increasing flow over path $p_m \in \mathcal{P}_{s-d}$ by ε, the cost increases by $\varepsilon J'_m(\lambda^m) + o(\varepsilon)$. Similarly, if we decrease λ^m by ε, the cost will decrease by $\varepsilon J'_m(\lambda^m) + o(\varepsilon)$. If we assume that two small changes were made on path p_m, the effect on $J(\lambda)$ would obviously be the sum of effects.

By considering two different paths p_m, $p_n \in \mathcal{P}_{s-d}$, if p_m was increased by ε and p_n was simultaneously decreased by the same amount of traffic flow, *i.e.*, if ε units of traffic flow were shifted to path p_m from path p_n, then the change in cost would be

$$\Delta J = \varepsilon\big(J'_m(\lambda^m) - J'_n(\lambda^n)\big) + o(\varepsilon). \quad (5.3)$$

It follows that, for small enough ε, the Flow Deviation will decrease the cost $J(\lambda)$ if $\Delta J < 0$, *i.e.*, if traffic flow is shifted from a longer path to a shorter route, loading the length of path p_m to $J'_m(\lambda^m)$.

Based on observation above, the following version of the Flow Deviation Algorithm executes the three following steps [58] at each iteration, starting from a given feasible paths flow configuration λ to find the optimal distribution of the traffic among a set $\mathcal{P}_{s-d}$ of precomputed paths.

Step 1. Compute the link flow f_j and first derivative link lengths for $j \in p$ with $p \in \mathcal{P}_{s-d}$

Step 2. Find a route $p^* \in \mathcal{P}_{s-d}$ with the minimum first derivative length.

Step 3. Let $\alpha \in [0, 1]$. Deviate a fraction α of flow from each of the other routes in $\mathcal{P}_{s-d}$.

That is, for each $p_m \in \mathcal{P}_{s-d}$, let

$$\overline{\lambda^m} = \begin{cases} (1-\alpha)\cdot\lambda^m & if \quad p_m \neq p^* \\ \lambda^m + \alpha \cdot \sum_{p_k \in \mathcal{P}_{s-d}, p_k \neq p_m} \lambda^k & if \quad p_m = p^* \end{cases}$$

Adjust α to minimize $J(\lambda)$.

Algorithm 7. *Flow Deviation (FD) Algorithm*

The resulting flow paths configuration $\overline{\lambda}$ is the result of the iteration and the process is executed until some stopping rule, such as number of iterations without improving the cost function, is met. The Flow Deviation Method is shown to reduce the value of the cost function to its minimum in the limit [57].

5.3 The Test Network Topologies

We run PmP, Game and FD algorithms on two network topologies: the Europe and USA networks depicted respectively by figure 5.1(a) and figure 5.1(b).

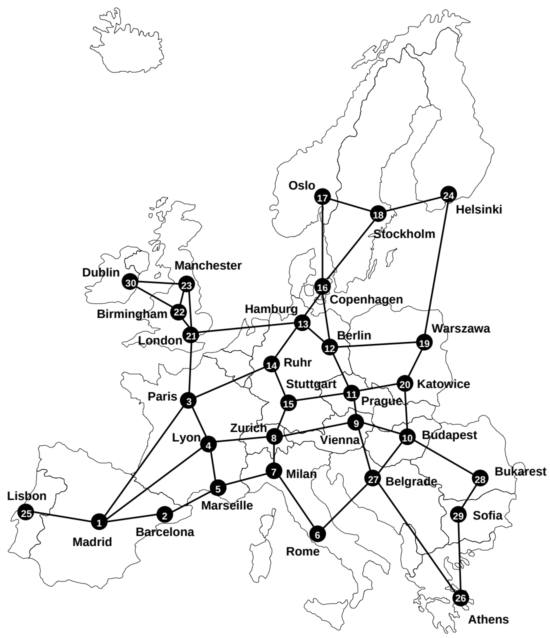
These network topologies have been chosen because they represents the most large connected network topologies used in research. The Europe topology is a 30 nodes and 46 links network, with a total of 2750400 paths connecting all node pairs and the USA topology is a 23 nodes and 38 links network, with a total of 969338 paths.

5.4 Experimental Results

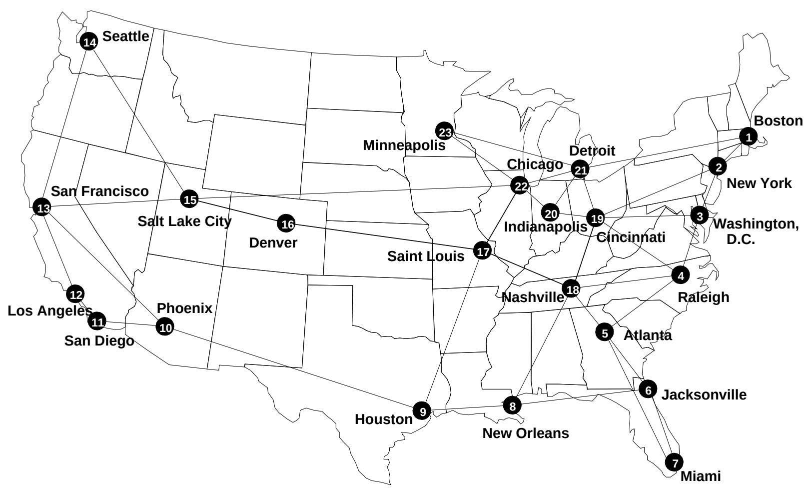
We conducted a set of experiments to compare (1) the quality of paths used by the three algorithms PmP, Game and FD Algorithms, and (2) the network efficiency in terms of total cost, maximum link utilization, average node and link interference achieved when using the set of parallel paths found by Shortest Vertex-Disjoint Paths Set (SVDP-S) Algorithm to disperse the traffic offered to the USA and Europe networks. The link interference is defined by the number of flows carried by a link while the node interference expresses the number of flows traversing a node. The total cost is the sum of the link costs.

5.4.1 Experimental setup

In these experiments, traffic flow is offered to the network along a given source-destination pair according to a Poisson process with parameter $\lambda = 2$, and the flow holding times are exponentially distributed with value $\mu^{-1} = 0.1$. The ingress and the egress nodes are uniformly generated among $|\mathcal{N}|\,(|\mathcal{N}| - 1)$ source-destination pairs, where $|\mathcal{N}|$ is the



(a) *Europe Network*



(b) *USA Network*

Figure 5.1: *Network Test*

number of nodes of the network under consideration. This means the ingress and the egress nodes are uniformly generated among 506 source-destination pairs in USA network, and 870 source-destination pairs in Europe network.

The parameters of the experiments were set as follows: the link capacity C_ℓ is uniformly distributed between 300 and 600 units of bandwidth, the bandwidth request of each flow is taken in the range $[200, 400]$ units. We use dynamic link state $1/\left(C_\ell - f_\ell\right)$, and the link state is updated at time when the flow is admitted. The number of experimental trials was set to $T = 30$ to ensure a 95% confidence interval and the number of flow requests per trial is set to $R = 50$.

5.4.2 Performance parameters

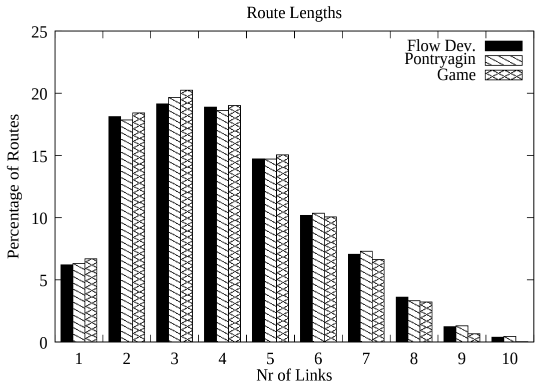
For the quality of paths, the performance parameters used are (1) the path length which gives a picture of the average path length in terms of number of hops, and consequently resource consumption since a longer path will require more resources than a shorter path, (2) the path multiplicity determining the average number of paths used by the source-destination pair, and in our case, this determines the potency of the algorithm to optimize the number of paths used, and (3) the path usage which expresses how the flow is dispersed over the selected paths.

The network efficiency is measured by optimality and reliability parameters. The network optimality is expressed by (1) the percentage flow acceptance which is the percentage of flows which have been admitted in the network, (2) the total cost incurred by the system, and (3) the maximum link utilization expressing the capability of the given strategy to keep the network far from congestion. The network reliability is expressed by (4) the average node and link interference which gives a picture on the number of flows to be rerouted in case of node or link failure.

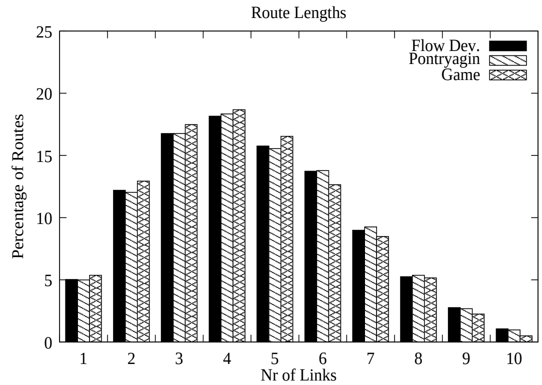
5.4.3 Simulation results

Two experiments were conducted to evaluate the effectiveness of the two proposed schemes compared to the FD Algorithm using (1) the quality of paths used by different algorithms, and (2) the network efficiency under different traffic profiles.

The results are depicted by figures 5.2 for path length, figures 5.3 for path multiplicity, and figures 5.4 for path usage. These figures reveal that Game routing outperforms PmP



(a) *USA Network*



(b) *Europe Network*

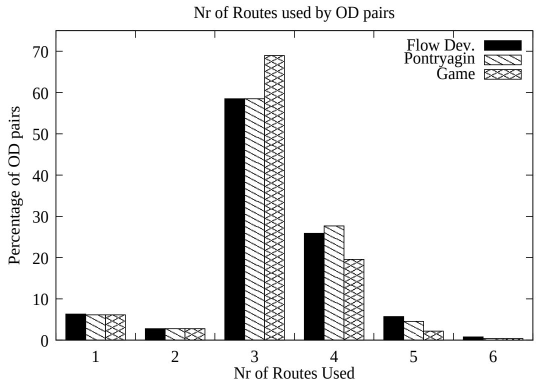
Figure 5.2: *Route Lengths*

and FD algorithms in terms of path length since it uses less paths with higher number of links and more paths with less number of links compared to PmP and FD algorithms. The FD algorithm performs worse in terms of resource consumption. This fact is confirmed by examining the results on path multiplicity given in figure 5.3 which show that the highest percentage of source-destination pairs using large number of routes is achieved by the FD algorithm.

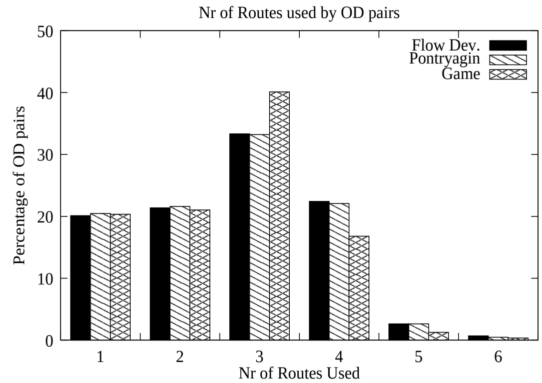
The three algorithms perform equally in term of path usage as shown by in figures 5.4. This may be justified by the fact that the three algorithms used the same set of parallel paths, which is precomputed using SVDP-S Algorithm, to disperse the incoming traffic flow over the network.

To further evaluate the performance of the two proposed schemes, we use other performance parameters, namely optimality and reliability parameters and results are depicted in figure 5.5 for USA network and 5.6 for Europe network.

The figures 5.5(a) and 5.6(a) show the percentage flow acceptance respectively for the the USA and Europe networks, figures 5.5(b) and 5.6(b) the cost incurred , figures 5.5(c) and 5.6(c) the average maximum link utilization, figures 5.5(d) and 5.6(d) maximum node interference, figures 5.5(e) and 5.6(e) average link interference, and figures 5.5(f) and 5.6(f) reveal the average node interference. It can be seen that Game routing outperforms the PmP algorithm and the FD algorithm performs worst.

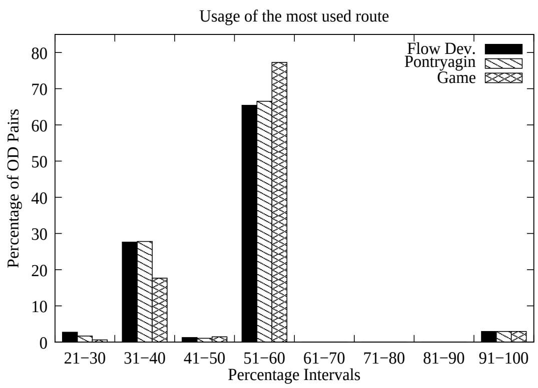


(a) *USA Network*

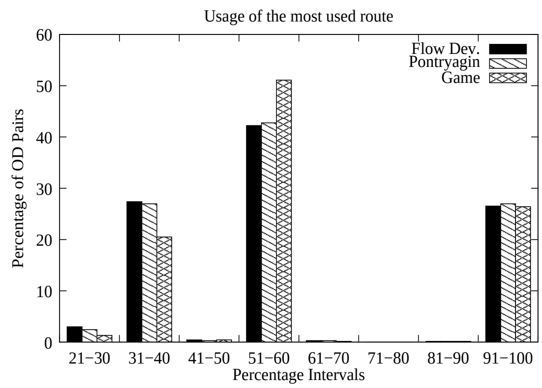


(b) *Europe Network*

Figure 5.3: *Route Multiplicity*



(a) *USA Network*



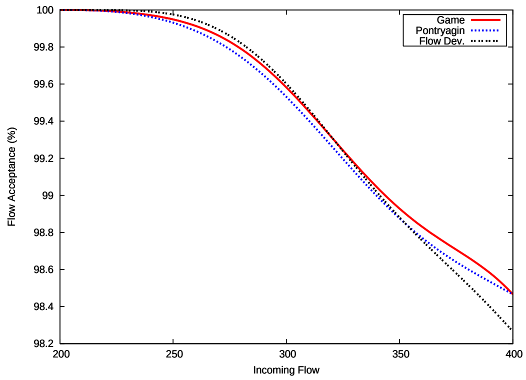
(b) *Europe Network*

Figure 5.4: *Route Usage*

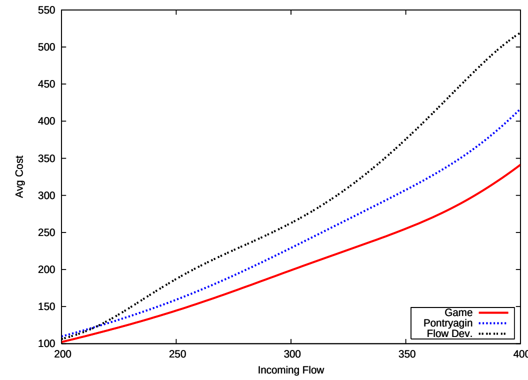
5.4.4 Results on game scheme for different calibration

Another set of experiments was conducted to analyze the impact of the calibration parameter α on the network paths quality and efficiency when using the Game routing scheme. In all these experiments, the link interference π_ℓ was computed on-line.

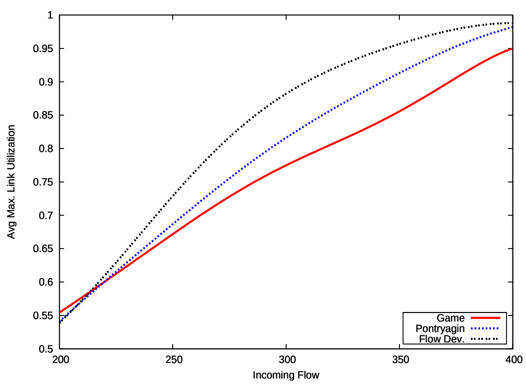
The quality of paths carrying flows is shown in the figures 5.7, 5.8, and 5.9 for different values of α. Four values of α have been considered, $\alpha = 0.0, 0.3, 0.5, 0.8$. These figures reveal that $\alpha = 0.0$ achieved the worst performance in term of path length in figure 5.7 since it uses more paths with higher number of hops and less paths with less number of



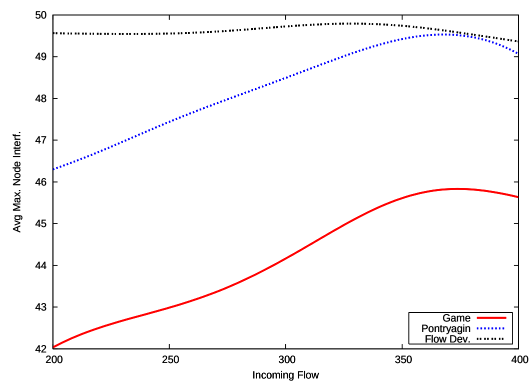
(a) *Flow Acceptance Comparison*



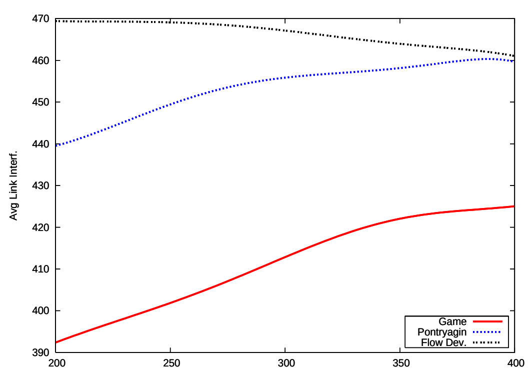
(b) *Avg. Cost Comparison*



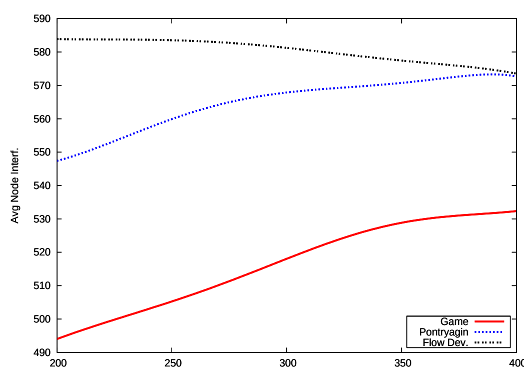
(c) *Max. Link Utilization Comparison*



(d) *Max. Node Interference Comparison*

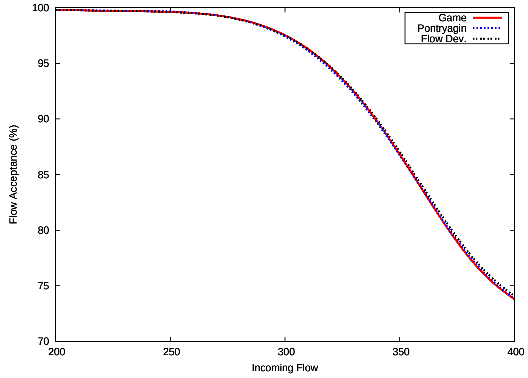


(e) *Link Interference Comparison*

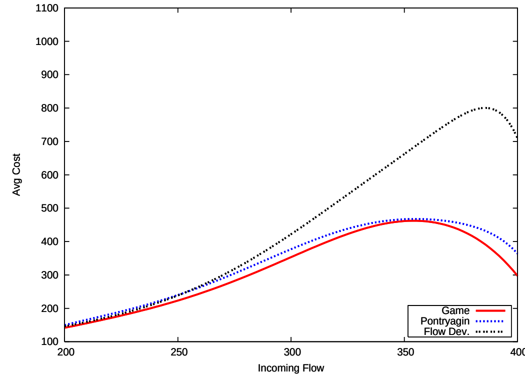


(f) *Node Interference Comparison*

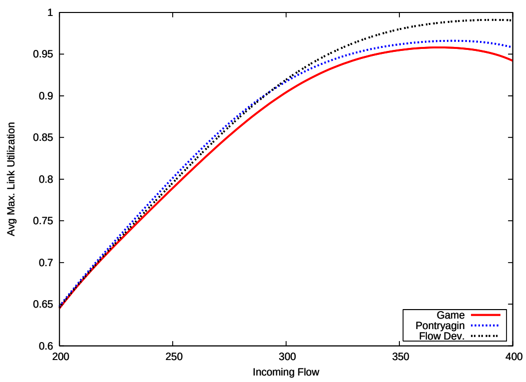
Figure 5.5: *USA Network*



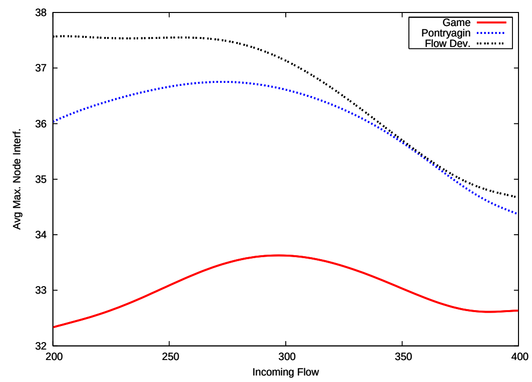
(a) *Flow Acceptance Comparison*



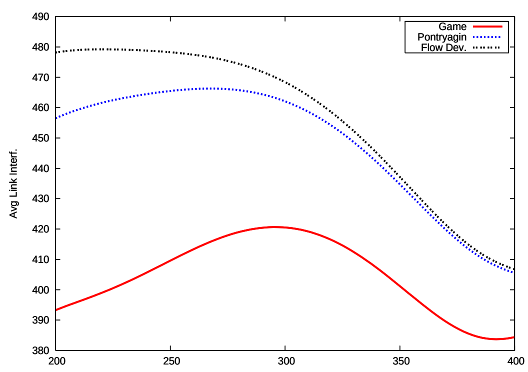
(b) *Avg. Cost Comparison*



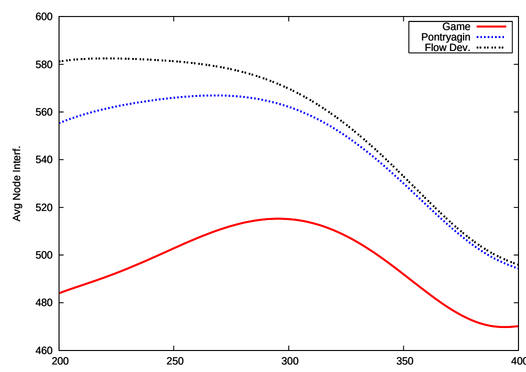
(c) *Max. Link Utilization Comparison*



(d) *Max. Node Interference Comparison*

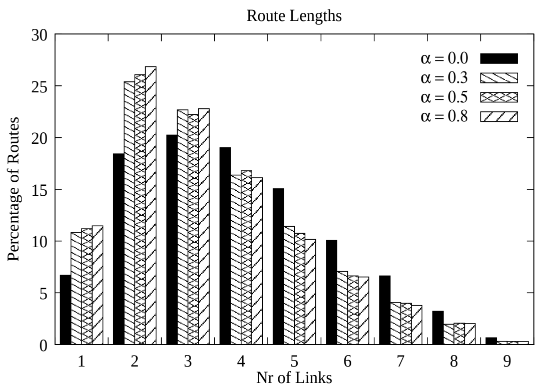


(e) *Link Interference Comparison*

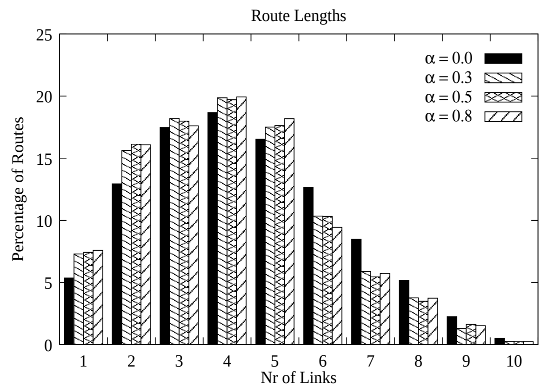


(f) *Node Interference Comparison*

Figure 5.6: *Europe Network*

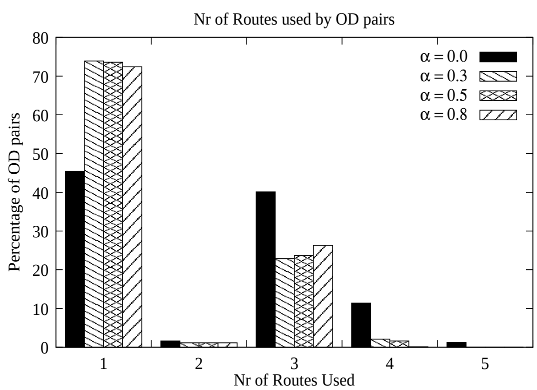


(a) *USA Network*

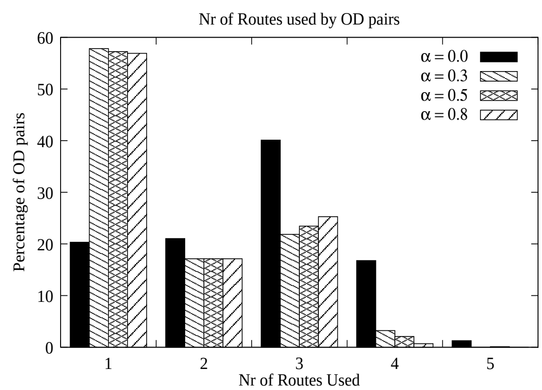


(b) *Europe Network*

Figure 5.7: *Route Lengths*



(a) *USA Network*

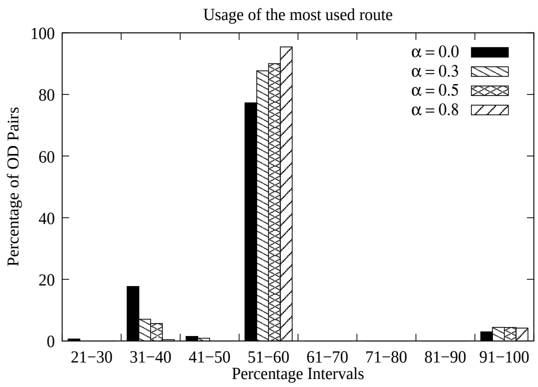


(b) *Europe Network*

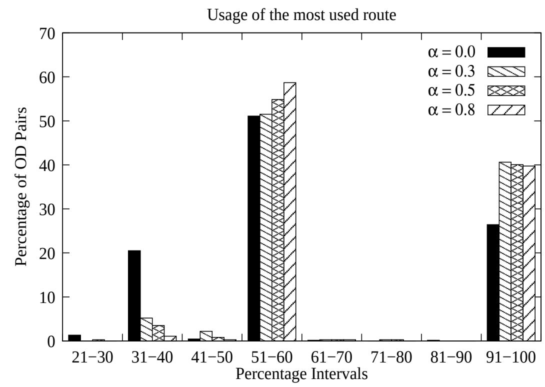
Figure 5.8: *Route Multiplicity*

hops compared to other values of α, and $\alpha = 0.8$, achieved the best performance. The results depicted by figure 5.9 reveal that $\alpha = 0.0$ leads to poorer path usage where the most used paths are used frequently. However, figure 5.8 reveals better path multiplicity for $\alpha = 0.0$.

In term of network efficiency, figures 5.10(a) and 5.11(a) show the percentage flow acceptance respectively by the USA and Europe network while figures 5.10(b) and 5.11(b) present the average maximum link utilization. Figures 5.10(c) and 5.11(c) depict the maximum link interference while figures 5.10(d) and 5.11(d) contain average link interference.



(a) *USA Network*



(b) *Europe Network*

Figure 5.9: *Route Usage*

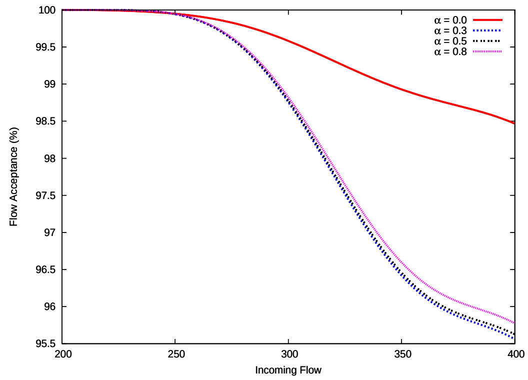
Figures 5.10(e) and 5.11(e) refer to the maximum node interference, and figures 5.10(f) and 5.11(f) the average node interference. These results reveal that in multi-path scheme, routing with $\alpha = 0.0$ provides the "best" percentage flow acceptance but only at the price of the worst performance for other performance parameters.

These experiments show that in multi-path scheme, MPLS routing using LIOA with $\alpha \neq 0.0$ leads to the best performance compared to CSPF or M/M/1 link cost. This is in agreement with previous works [16, 50] that suggested the use of a calibration parameter α which balances the impact of the bandwidth and interference on the link cost.

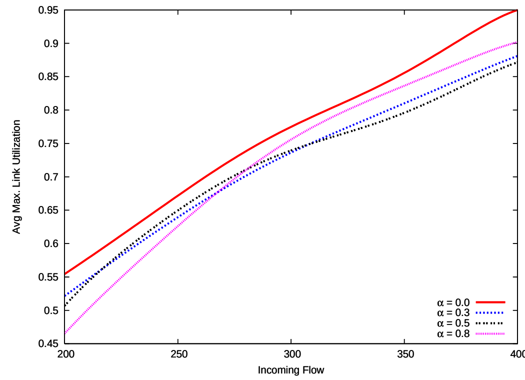
5.4.5 Cooperative and competitive routing cost comparison

Experiments were conducted to compare the cost achieved by cooperative routing and competitive routing scheme. The results are illustrated by the figures 5.12 and 5.13, for different type of the link interference.

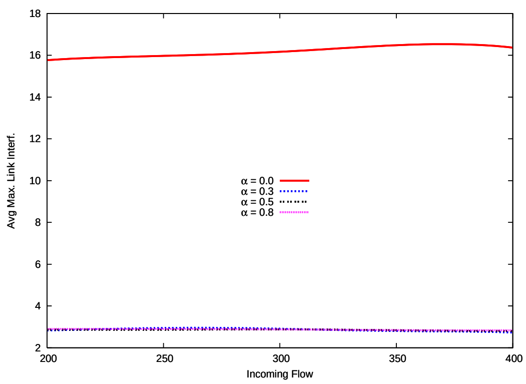
We considered two value of α, $\alpha = 0.0$ and $\alpha = 0.5$. And we assume that each flow at a given source and destination (s-d) pair comes from four selfish users in proportionally way, 1, 2, 3, 4. This means that if λ is the total flow at a given s-d pair, the first user generates $\lambda/10$ unit of bandwidth, the second generates $\lambda/5$ unit of bandwidth, the third $3 \times \lambda/10$, and finally the fourth user generates $2 \times \lambda/5$.



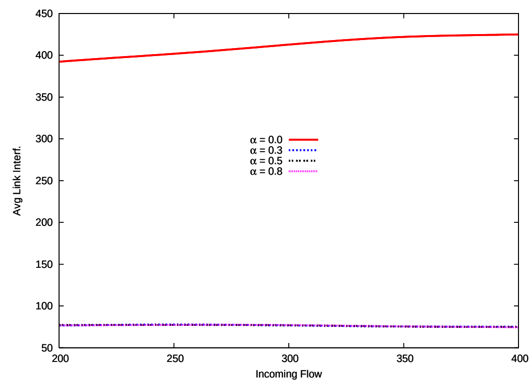
(a) *Flow Acceptance*



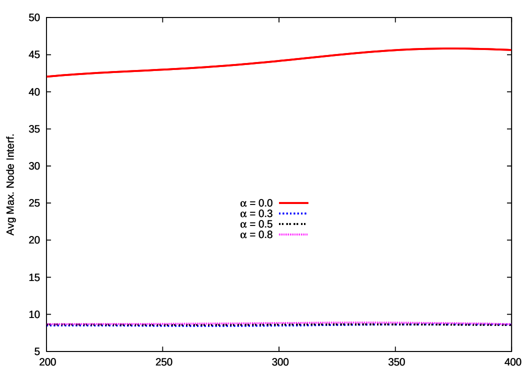
(b) *Max. Link Utilization*



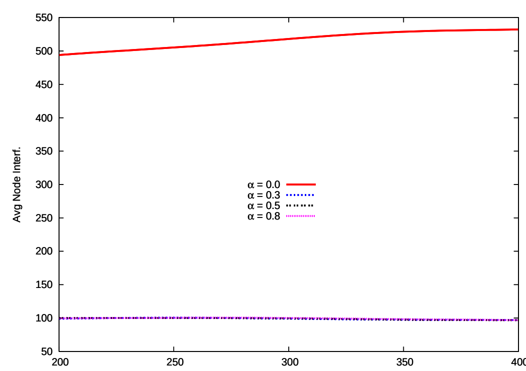
(c) *Max. Link Interference*



(d) *Link Interference*

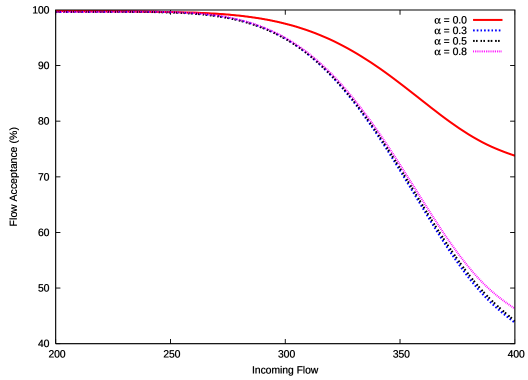


(e) *Max. Node Interference*

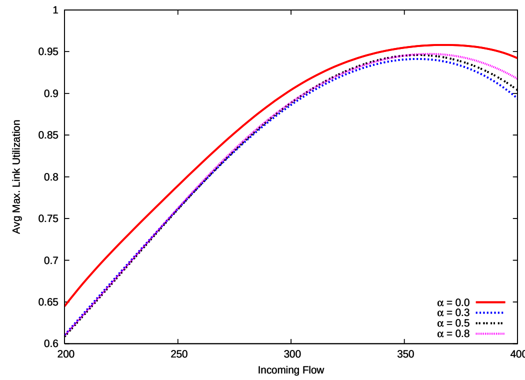


(f) *Node Interference Comparison*

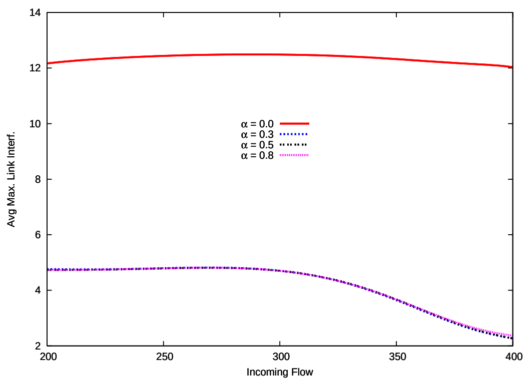
Figure 5.10: *USA Network*



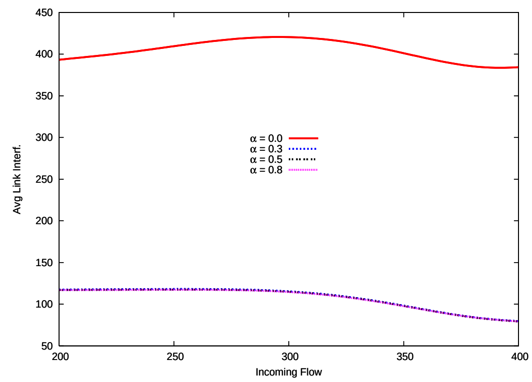
(a) *Flow Acceptance*



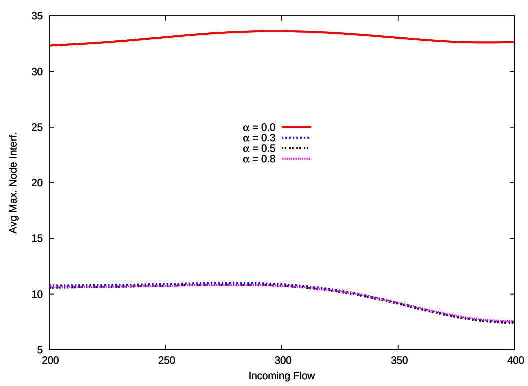
(b) *Max. Link Utilization*



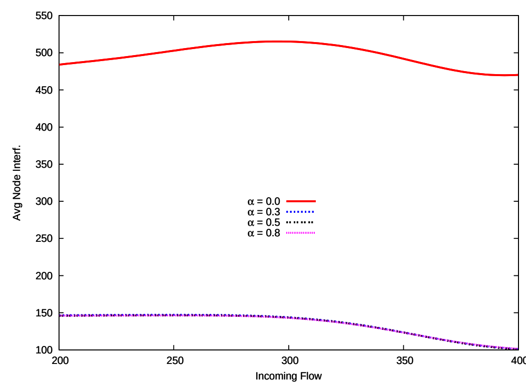
(c) *Max. Link Interference*



(d) *Link Interference*



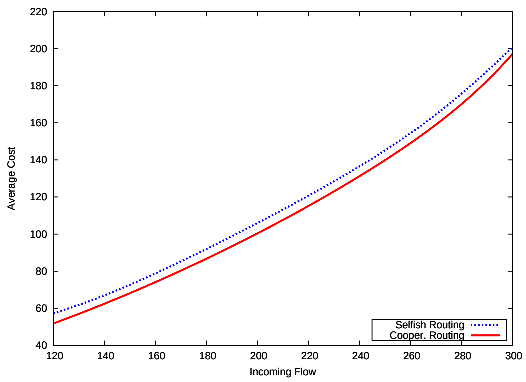
(e) *Max. Node Interference*



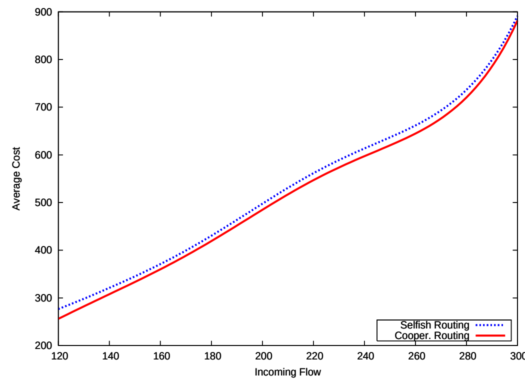
(f) *Node Interference Comparison*

Figure 5.11: *Europe Network*

The results obtained are depicted in the figures 5.12 for USA network and in the figure 5.13 for Europe network. These results reveal that competitive routing achieved the highest cost compared to the cooperative routing. This is in agreement with the fact that traffic flow at Nash equilibrium do not in general minimize the total cost in the system due to the fact that the selfish users do not realize the impact of their action in the performance of the general.

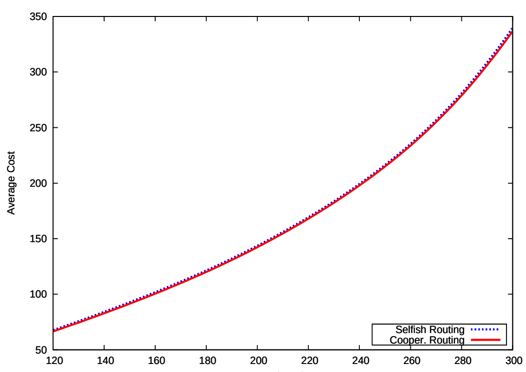


(a) *On-line Interference for* $\alpha = 0.0$

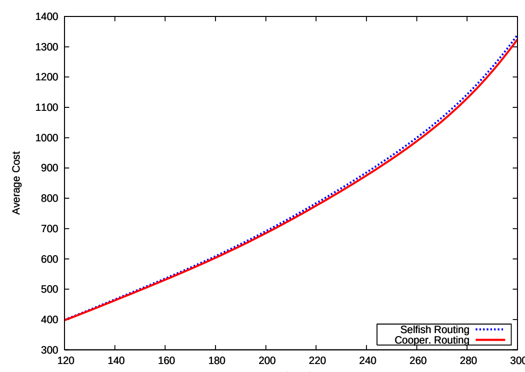


(b) *On-line Interference for* $\alpha = 0.5$

Figure 5.12: *USA Network*



(a) *On-line Interference for* $\alpha = 0.0$



(b) *On-line Interference for* $\alpha = 0.5$

Figure 5.13: *Europe Network*

5.5 Summary

In this chapter we compared the performance of the PmP and Game routing schemes developed in the two previous chapters compared to the traditional Flow Deviation Algorithm. We performed extensive experiments for different network topologies. Based on results obtained, we believe we have identified robust dynamic multi-path routing schemes that can be used to effectively route traffic in MPLS networks.

In addition, we conducted numerical experiments using Game routing scheme considering different values of calibration parameter α. The results show that in multi-path routing schemes MPLS routing with LIOA link cost with $\alpha \neq 0$ achieve better performance compared to CSPF or M/M/1 link cost. By examining the competitive routing cost and the cooperative routing cost, as expected, we found that the competitive routing do not in general minimize the total cost of the network. This means that the network can reach the equilibrium point without reaching the optimum operational point.

Chapter 6

QoS Multi-path Routing for Wireless Sensor Networks

Wireless communication networks these days are one of the fastest growing segments of the communication industry [59]. Wireless sensor networks are being deployed in a wide variety of civil and military applications such as security management, surveillance, automation, and environmental monitoring. A sensor network is composed of large number of sensor nodes, which are densely deployed inside the phenomenon or very closed to it [60]. As pointed out by Akyildiz et al. [60], wireless sensor networks present several limitations. These include

1. Sensor nodes are densely deployed and are range-limited systems, therefore efficient multi-hop routing algorithms are required [61].

2. Sensor nodes are unreliable and prone to failure, and the topology of sensor networks changes very frequently, hence it is desirable to set up energy constrained multi-path routing.

3. Sensor nodes are limited in power, computational capacities and memory, thus the topology control with per-node transmission power adjustment is needed [62].

These limitations have oriented research on sensor networks and several routing algorithms minimizing energy consumption have been proposed in order to (1) enhance the performance in the energy constrained network and (2) increase the network lifetime usually defined as the time period between the outset of the functioning of network and the fading of the energy of the first sensor node. Stojmenovic et al. [63] discussed routing algorithms for wireless networks with the goal of increasing the network lifetime by defining a new power-cost metric based on the combination of both node's lifetime and distance based power metric, thus proposing power aware routing algorithm that attempts to minimize the total power needed to route a message between a source and a destination. Li et al. [64] proposed a protocol that, given a communication network, computes a sub-network such that, for every pair (u, v) of nodes connected in the original network, there is a minimum-energy path u and v in the sub-network where a minimum-energy path is the one that allows messages to be transmitted with a minimum use of energy. J. Liu et al. [62] have considered the problem of topology control in a network of heterogeneous wireless devices with different maximum transmission ranges, where asymmetric wireless links are not uncommon. P. Liu et al. [65] have developed a novel energy-efficient routing called the THEEM (Two Hop-Energy-Efficient Mesh) protocol for wireless sensor network.

As the case in wired networks, in wireless sensor networks single path routing is apparently simple and consumes less energy than multi-path routing. However, reliability and delay are critical in some applications whereas sensor network nodes are unreliable, thus outperforming the single routing path strategies. On the other hand, the reliability of a system can be increased by using multi-path routing, which allows the establishment of more than one paths between source and destination and provides an easy mechanism to increase the likelihood of reliable data delivery by sending multiple copies of data along different paths [66]. Low power consumption is the most important requirement in wireless sensor networks, hence multi-path routing algorithms minimizing the energy consumption are needed in order to increase the network lifetime and satisfy the QoS traffic requirements.

Pointed out by Ganesan et al. [67], the traditional disjoint paths (node disjoint paths) have same attractive resilience properties, but they can be energy inefficient. Alternate node-disjoint path can be longer, and therefore expend significantly more energy than that expended on the primary path. Since this energy can adversely impact the lifetime and the performance of a sensor network, they have considered a slight different kind of multi-path, namely a braided multi-path, which relaxes the requirement for node disjointedness. Alternate paths in a braid are partially disjoint from the primary path, not completely node-disjoint. They have proposed a multi-path scheme for energy-efficient recovery from node failure in wireless sensor networks. Fernandes [61] explored the possibility of extending the braided multi-path routing method proposed by Barrenechea et al. [68] to the case of more general random geometric graphs. The Barrenechea et al. scheme is based on constrained random walks and achieves almost stateless multi-path routing on a grid network. Recently, X. Huang and Y. Fang [18] have proposed a braided multi-path routing scheme delivering packets to the sink on time and at desired reliability based on the information sensed, taking into account unpredictability of network topology and trying to minimize energy consumption. This scheme referred to as MCMP (Multi-Constrained Multi-Path routing) addresses the issue of multi-constrained QoS in wireless sensor networks.

In this chapter, we compare in term of the energy consumption the MCMP scheme with traditional link-disjoint paths referred to as LDPR (Link-Disjoint Paths Routing), in which the number of paths used between the source and the sink is limited by the criticality of the information to be delivered. In the traditional link-disjoint multi-path routing, the number of paths used is function of reliability, *i.e.*, the higher the reliability required, the higher the number of paths used. And we suggest a solution to improve MCMP routing.

In the following sections, we present a sensor network communication architecture and examine the path delay, energy and reliability behavior. Thereafter, we present a brief formulation of the LDPR and MCMP problems. Finally, we propose the modified MCMP referred to as MMCMP which improves the energy consumption of the MCMP scheme and then present simulation results comparing the these approaches in term of energy consumption.

6.1 Sensor Network Communication Architecture

In sensor networks, the sensor nodes are scattered in a target observation area with objective of collecting data, and route it to the end users via the sink or base station. Sensor nodes co-operate for ensuring that every information sensed and data collected are successfully relayed to the sink. We depict an architecture, based on the model presented in [60], for sensor network communication in figure 6.1. In this model, data is forwarded to the end user by a multi-hop infra-structureless network to the base station and the base station may communicate with the task manager node via Internet or satellite.

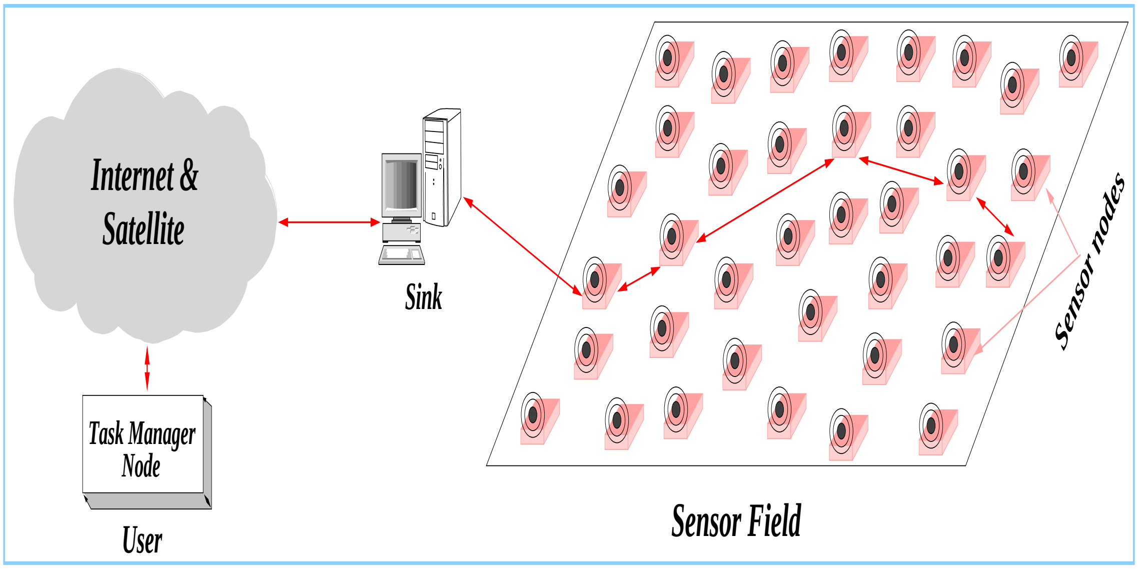


Figure 6.1: *Sensor Nodes Scattered in a Sensor Field*

At the outset of the wireless sensor network operation, sensor node may fall into one of the following states [69]:

1. Sensing: a sensing node monitors the source using an integrated sensor, digitizes the information, processes it, and stores the data in its on-board buffer. These data will be eventually sent to the base station.

2. Relaying: a relaying node receives data from other nodes and forwards it towards their destination.

3. Sleeping: for a sleeping node, most of the device is either shut down or works in low-power mode. A sleeping node does not participate in either sensing or relaying. However, it "wakes up" from time to time and listens to the communication channel in order to answer requests from other nodes. Upon receiving a request, a state transition to "sensing" or "relaying" may occur.

4. Dead: a dead node is no longer available to the sensor network. It has either used up its energy or has suffered vital damage. Once a node is dead, it cannot re-enter any other state.

6.2 Path Delay, Energy and Reliability Behavior

Let consider a sensor network represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ where $\mathcal{N}$ is the set of sensor nodes (location) and $\mathcal{L}$ the set of links. As a data source is usually far from the sink with the distance exceeding the range of communication, there is a need to deploy a certain number of sensor nodes that may act as relays used to route data over a multi-hop path. The multi-hop path between node s_1 and node s_ϵ is represented by $p = (s_1, \ldots, s_\epsilon)$ ordered list of nodes $s_i \in \mathcal{N}$ such that the pair $(s_i, s_{i+1}) \in \mathcal{L}$, for $i = 1, \ldots, \epsilon - 1$.

The path p is a series system of links, the path delay, *i.e.*, the delay between the node s_1 and s_ϵ is given by the sum of link delays

$$\mathcal{D}(p) = \sum_{i=1}^{\epsilon-1} d(s_i, s_{i+1}) \tag{6.1}$$

where $d(s_i, s_{i+1})$ is the delay of data over the link $(s_i, s_{i+1}) \in \mathcal{L}$.

Similarly, the energy consumption between node s_1 and node s_ϵ is given by [60]

$$\mathcal{W}(p) = \sum_{i=1}^{\epsilon-1} \omega(s_i, s_{i+1}) \tag{6.2}$$

where $\omega(s_i, s_{i+1})$ is the energy required to receive and transmit data between the node s_i and s_{i+1}. The necessary energy per bit for a node s_i to receive a bit and then transmits it to the node s_{i+1} is given by [69]

$$\omega_i(s_i, s_{i+1}) = \alpha_1 + \alpha_2 \|x_{s_i} - x_{s_{i+1}}\|^n \tag{6.3}$$

where $\alpha_1 = \alpha_{11} + \alpha_{12}$ with α_{11} the energy per bit consumed by s_i as transmitter and α_{12} the energy per bit consumed as receiver, and α_2 accounts for the energy dissipated in the transmitting operation. Typical values for α_1 and α_2 are respectively $\alpha_1 = 180nJ/bit$ and $\alpha_2 = 10pJ/bit/m^2$ for the path loss exponent experienced by a radio transmission $n = 2$ or $\alpha_2 = 0.001pJ/bit/m^4$ for the path loss exponent experienced by a radio transmission $n = 4$. x_{s_i} is the location of the sensor node s_i, and $\|x_{s_i} - x_{s_{i+1}}\|$ is the euclidean distance between the two sensor nodes s_i and s_{i+1}, $i = 1, \ldots, \epsilon - 1$. Thus, in (6.2), we have

$$\omega(s_i, s_{i+1}) = f_{s_i \to s_{i+1}} \cdot \omega_i(s_i, s_{i+1}) \tag{6.4}$$

where $f_{s_i \to s_{i+1}}$ denotes the data rate on the link $(s_i, s_{i+1}) \in \mathcal{L}$.

Assuming that these links are independent, from [36], the path reliability $\mathcal{R}(p)$ is given by

$$\mathcal{R}(p) = \prod_{i=1}^{n-1} R(s_i, s_{i+1}) \tag{6.5}$$

where $R(s_i, s_{i+1})$ is the reliability of the link $(s_i, s_{i+1}) \in \mathcal{L}$.

Considering the set of parallel paths $\mathcal{P} = \{p_1, \ldots, p_M\}$, the delay experienced and the energy consumed by the data source over $\mathcal{P}$ are respectively given by

$$\mathfrak{D}(\mathcal{P}) = \max\{\mathcal{D}(p) : p \in \mathcal{P}\} \tag{6.6}$$

and

$$\mathfrak{W}(\mathcal{P}) = \sum_{p \in \mathcal{P}} \mathcal{W}(p) \tag{6.7}$$

where $\mathcal{D}(p)$ and $\mathcal{W}(p)$ are computed respectively in (6.1) and (6.2). And finally, from [36], the reliability of the data source over $\mathcal{P}$ is given by

$$\mathfrak{R}(\mathcal{P}) = 1 - \prod_{p \in \mathcal{P}} \big(1 - \mathcal{R}(p)\big) \tag{6.8}$$

where $\mathcal{R}(p)$ is computed using the formula given in (6.5).

6.3 LDPR and MCMP Problem Formulation

Let us consider a wireless sensor network represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of sensor nodes and $\mathcal{L}$ is the set of links. Suppose there exists a data source f at a given location x_s sensed by the node s. This data must be routed to the base station. The data possesses a QoS requirement expressed in term of delay D and reliability R.

Let $\mathcal{P} = \{p_1, \ldots, p_M\}$ denote the set of possible paths from s to the base station b assumed to be stationary. Each path $p_\jmath \in \mathcal{P}$, $\jmath = 1, \ldots, M$, is associated with the delay $d_\jmath$ and reliability r_j. If every path $p \in \mathcal{P}$ has delay larger than the delay D required by the data source, then the data source is dropped for no path can fulfill the delivery of the packet with that constraint. This is not the case for the reliability, since multi-path routing can improve the reliability. However, the use of several path increases energy consumption, which therefore affects the lifetime of the network. Thus, in order to save the energy, the set with minimum number of paths is chosen as forwarding set.

The routing objective is then to find a minimum number of path in $\mathcal{P}$ that satisfy the QoS requirement of a given data source f. This can be formulated as an optimization problem given below.

Problem 6.1. *Find the subset $\mathcal{P}_{s-b} \subseteq \mathcal{P}$ whose paths solve the following zero-one optimization problem.*

$$\min \sum_{j=1}^{M} x_j \tag{6.9}$$

$$\textit{subject to} \quad x_j d_j \leq D \tag{6.10}$$

$$1 - \prod_{j=1}^{M}(1 - x_j r_j) \geq R, \tag{6.11}$$

$$x_j = 0 \ \textit{or} \ 1, \ \textit{for all} \ j = 1, 2, \ldots, M \tag{6.12}$$

★

Beyond the fact that the global knowledge of topology and network characteristic are assumed, the above problem definition requires exact information about path quality, which is almost impossible to get in wireless sensor network. However, QoS requirement can be provided only with certain probability referred to as soft-QoS. This leads to consider the constraints of the above problem must be considered under inaccurate path information defined in probabilistic way. Thus the problem can be reformulated as follows.

Problem 6.2. *Find the subset $\mathcal{P}_{s-b} \subseteq \mathcal{P}$ whose paths solve the following zero-one optimization problem.*

$$\min \sum_{j=1}^{M} x_j \tag{6.13}$$

$$\textit{subject to} \quad \mathbf{P}\big[x_j d_j \leq D\big] \geq \alpha \tag{6.14}$$

$$\mathbf{P}\left[1 - \prod_{j=1}^{M}(1 - x_j r_j) \geq R\right] \geq \beta, \tag{6.15}$$

$$x_j = 0 \textit{ or } 1, \textit{ for all } j = 1, 2, \ldots, M \tag{6.16}$$

★

where α and β are respectively soft-QoS probability for delay and for reliability. The problem 6.2 is a stochastic or probabilistic programming.

Let us remark that the expression $1 - \prod_{j=1}^{M}(1 - x_j r_j) \geq R$ in constraint (6.11) or (6.15) is equivalent to

$$\prod_{j=1}^{M}(1 - x_j r_j) \leq 1 - R \tag{6.17}$$

Taking logarithm in both side of equation (6.17), we obtain

$$\sum_{j=1}^{M} \log(1 - x_j r_j) \leq \log(1 - R). \tag{6.18}$$

Furthermore, since $x_j = 0$ or 1, $\log(1 - x_j r_j) = x_j \log(1 - r_j)$. Thus the constraint (6.15) can be simplified as

$$\mathbf{P}\left[\sum_{j=1}^{M} x_j \log(1 - r_j) \leq \log(1 - R)\right] \geq \beta \tag{6.19}$$

The problem (6.2) is thus the stochastic linear zero-one program and its formulation fits well into a centralized management environment that can optimally route data source, using complete knowledge of the data QoS requirement. However, such routing solution is subject to many no way out challenges in the case of wireless sensor networks since sensor nodes are prone to failure, limited in computational capacities and memory, and the topology of sensor networks changes very frequently. Consequently

- It is impossible to get the exact instantaneous link state information.
- It is really challenging to keep path metrics consistent at all nodes.
- It is almost impossible to store the end-to-end information at a node.

Considering these limitations, X. Huang and Y. Fang [18] proposed a distributed link-based QoS routing scheme that is addressed based on local information and formulated as follows.

Problem 6.3. *At each node $\imath$, find the subset $\mathbf{N}_0 \subseteq \mathbf{N}[\imath]$ the set of neighbors of node $\imath$ that solves the following zero-one stochastic programming problem*

$$\min \sum_{\jmath \in \mathbf{N}[\imath]} x_\jmath \tag{6.20}$$

$$\textit{subject to} \quad \mathbf{P}\left[x_\jmath D_{\imath\jmath} \leq L_\imath^d\right] \geq \alpha \textit{ for } L_\imath^d > 0 \tag{6.21}$$

$$\mathbf{P}\left[\left(1 - \prod_{\jmath \in \mathbf{N}[\imath]} (1 - x_\jmath R_{\imath\jmath})\right) \geq L_\imath^r\right] \geq \beta, \tag{6.22}$$

$$x_\jmath = 0 \textit{ or } 1, \textit{ for all } \jmath \in \mathbf{N}[\imath] \tag{6.23}$$

★

Where $R_{\imath\jmath}$ and $D_{\imath\jmath}$ are respectively the delay and reliability of the link $\ell_{\imath\jmath}$. Reliability and delay are assumed to be random depending on time t omitted for simplicity sake and links are assumed to be independent in term of delay and reliability. $L_\imath^d = (D - D_\imath)/h_\imath$ is the hop requirement at node $\imath$ with $D_\imath$ the actual delay experienced by a packet at node $\imath$, and $h_\imath$ the hop count from node $\imath$ to the sink, and $L_\imath^r = \sqrt[h_\imath]{R_\imath}$ hop requirement for reliability at node $\imath$ and $R_\imath$ is the portion of reliability requirement assigned to the path through node $\imath$ decided by the upstream node of $\imath$.

Along their analysis, they have deduced the deterministic linear zero-one programming problem formulated as follows.

Problem 6.4. *At each node $\imath$, find the subset $\mathbf{N}_0 \subseteq \mathbf{N}[\imath]$ the set of neighbors of node $\imath$ that solves the following zero-one linear program*

$$\min \sum_{\jmath \in \mathbf{N}[\imath]} x_\jmath \tag{6.24}$$

$$\textit{subject to} \quad x_\jmath \left(\frac{\alpha}{1-\alpha} \left(\Delta^d_{\imath\jmath}\right)^2 + 2L^d_\imath d_{\imath\jmath} - d^2_{\imath\jmath} \right) \leq \left(L^d_\imath\right)^2, \ \textit{when } L^d_\imath - d_{\imath\jmath} > 0 \tag{6.25}$$

$$\sum_{\jmath \in \mathbf{N}[\imath]} x_\jmath \log \left(\mathcal{Q}\left(\frac{R_{\imath\jmath} - r_{\imath\jmath}}{\Delta^r_{\imath\jmath}} \right) \right) \geq \log \beta, \tag{6.26}$$

$$\sum_{\jmath \in \mathbf{N}[\imath]} x_\jmath \log\left(1 - R_{\imath\jmath}\right) \leq \log\left(1 - L^r_\imath\right) \tag{6.27}$$

$$0 \leq R_{\imath\jmath} \leq r_{\imath\jmath}, \ \textit{for all } \jmath \in \mathbf{N}[\imath] \tag{6.28}$$

$$x_\jmath = 0 \ \textit{or } 1, \ \textit{for all } \jmath \in \mathbf{N}[\imath] \tag{6.29}$$

★

where the $\mathcal{Q}-$function in (6.26) is defined as

$$\mathcal{Q}(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{1}{2}t^2\right) dt, \tag{6.30}$$

and $\Delta^d_{\imath\jmath}$ and $\Delta^r_{\imath\jmath}$ are respectively standard deviation of $D_{\imath\jmath}$ and $R_{\imath\jmath}$ computed adaptively using RTT estimation for timer management in TCP, *i.e.*, the current $\Delta^d_{\imath\jmath}(t)$ and $\Delta^r_{\imath\jmath}(t)$ are found based on previous values of $d_{\imath\jmath}(t-1)$, $r_{\imath\jmath}(t-1)$, $\Delta^d_{\imath\jmath}(t-1)$, and $\Delta^r_{\imath\jmath}(t-1)$, and the current mean $d_{\imath\jmath}$ of $D_{\imath\jmath}$ and $r_{\imath\jmath}$ of $R_{\imath\jmath}$ as follows [70].

$$\Delta^d_{\imath\jmath}(t) = (1-\rho)\Delta^d_{\imath\jmath}(t-1) + \rho|d_{\imath\jmath}(t) - d_{\imath\jmath}(t-1)| \tag{6.31}$$

$$\Delta^r_{\imath\jmath}(t) = (1-\gamma)\Delta^r_{\imath\jmath}(t-1) + \gamma|r_{\imath\jmath}(t) - r_{\imath\jmath}(t-1)| \tag{6.32}$$

with tunable forgetting parameters ρ and γ for smoothing the variations of $d_{\imath\jmath}$ and $r_{\imath\jmath}$ in time.

6.4 Modified MCMP (MMCMP) Scheme

The MCMP scheme aims only at minimizing the number of paths used in forwarding data source to the sink so as to minimize the total energy transmission. However, the proposed scheme does not really take into consideration the real energy consumption in the network since it does not give any indication for the case where the best choice in term of energy between two links must be made to satisfy the QoS requirement of data source to be forwarded.

To illustrate our proposal, let us consider the figure 6.2 in which the choice must be made between the link $(\imath, \jmath)$ and the link $(\imath, k)$ or equivalently the node $\jmath$ and node k to be added to the subset $\mathbf{N}_0$ of $\mathbf{N}[\imath]$ the set of the neighbors of $\imath$, assuming that the two candidates $\jmath$ and k may satisfy the QoS requirement for data source.

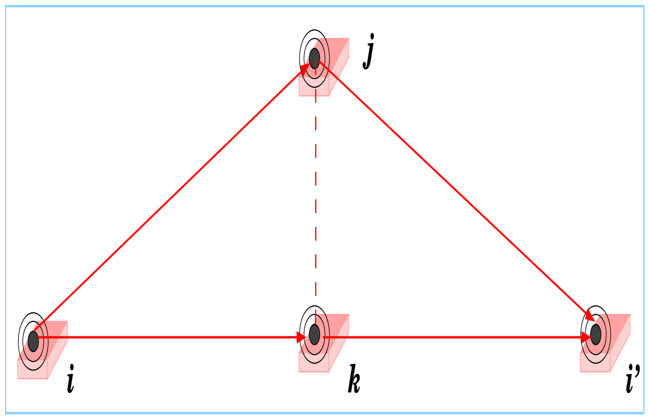


Figure 6.2: *MCMP Scheme Inefficiency*

From Pythagoras' theorem, the distance between node $\imath$ and node $\jmath$ is larger than that between $\imath$ and k. The flip side of the coin is that using the formula in (6.3) for energy transmission computation, the energy transmission between $\imath$ and $\jmath$ is higher than energy transmission between $\imath$ and k. This means that the choice of $\jmath$ leads to the higher energy consumption. However, according to MCMP approach, the choice between the node $\jmath$ and k is arbitrary and this arbitrary choice is not likely to select the best node in term of minimum energy consumption.

As the objective is to send data from source to the sink with the total energy transmission as minimum as possible, the choice between node $\jmath$ and k must be made based on the energy transmission consumed to reach the node. Thus, the solution that overcomes this drawback of the MCMP scheme is to reformulate the MCMP problem. This new model is referred to as MMCMP (modified MCMP), guaranteeing that the data is transmitted with minimum energy. MMCMP scheme finds the subset $\mathbf{N}_0$ of the set $\mathbf{N}[\imath]$ with the fewest expected energy transmission while satisfying the required QoS to deliver the data to the sink. The goal of MMCMP scheme is then to find the subset $\mathbf{N}_0 \subseteq \mathbf{N}[\imath]$ satisfying QoS requirement of data source and minimizing the total energy transmission. Indeed, denoting $\omega(\imath, \jmath)$ the energy required from a node $\imath$ to receive data and then transmits it to the node $\jmath$ given by the formula (6.4), the modified MCMP problem is formulated as follows.

Problem 6.5. *At each node $\imath$, find the subset $\mathbf{N}_0 \subseteq \mathbf{N}[\imath]$ the set of neighbors of node $\imath$ that solves the following linear zero-one program*

$$\min \sum_{\jmath \in \mathbf{N}[\imath]} \omega(\imath, \jmath) x_\jmath \tag{6.33}$$

$$\textit{subject to} \quad x_\jmath \left(\frac{\alpha}{1-\alpha} \left(\Delta^d_{\imath\jmath}\right)^2 + 2L^d_\imath d_{\imath\jmath} - d^2_{\imath\jmath} \right) \leq \left(L^d_\imath\right)^2, \ \textit{when } L^d_\imath - d_{\imath\jmath} > 0 \tag{6.34}$$

$$\sum_{\jmath \in \mathbf{N}[\imath]} x_\jmath \log \left(\mathcal{Q} \left(\frac{R_{\imath\jmath} - r_{\imath\jmath}}{\Delta^r_{\imath\jmath}} \right) \right) \geq \log \beta, \tag{6.35}$$

$$\sum_{\jmath \in \mathbf{N}[\imath]} x_\jmath \log \left(1 - R_{\imath\jmath}\right) \leq \log \left(1 - L^r_\imath\right) \tag{6.36}$$

$$0 \leq R_{\imath\jmath} \leq r_{\imath\jmath}, \ \textit{for all } \jmath \in \mathbf{N}[\imath] \tag{6.37}$$

$$x_\jmath = 0 \ \textit{or } 1, \ \textit{for all } \jmath \in \mathbf{N}[\imath] \tag{6.38}$$

★

The MMCMP problem as well as the MCMP problem are deterministic linear zero-one program and several methods have been proposed in literature to address such kind of problems [71, 72]. In both problems, the number of constraints is $2\,|\mathbf{N}[\imath]| + 2$, and the number of the decision variables is $|\mathbf{N}[\imath]|$ which is the size of $\mathbf{N}[\imath]$. Thus, the problem size is relatively small and might be proportional to the node density.

6.5 Experimental Results

In this section, we evaluate the effectiveness of the modified MCMP (MMCMP) scheme in term of total energy transmission through experiments. Our goal is to compare the average energy consumption, delivery ratio and average data delivery delay of the MMCMP scheme with those of baseline single path (SP) routing, MCMP and LDPR schemes.

- Total energy indicates the total energy consumption in transmission and reception of all packets in the network. This metric shows how efficient is the approach used with respect to the energy consumption.

- Delivery ratio is one of the most important metrics in real-time applications, which indicates the number of packets that could meet the specified QoS level. It is the ratio of successful packet receptions referred to as received packets, to attempted packet transmissions referred to as sent packets.

- Average data delivery delay is the end-to-end delay experienced by successfully received packets.

In addition, we compare the quality of paths used by MCMP and MMCMP schemes in terms of path length (number of hops of paths used) and path multiplicity (average number of paths used to send data to the base station).

6.5.1 Environment setup

We assume 50 sensor nodes are randomly deployed in a sensing field of $100m \times 100m$ square area and the transmission range is $25m$. Among these sensor nodes, 10 are chosen to generate data. Figure 6.3 shows the randomly generated network used for experiments, where black circles represent sensor nodes which are generating data. The sink or base station is in unique form at the top left of the field.

Link reliability and delay are random, reliability is uniformly distributed in the range of $[0.8, 1]$ and delay in $[1, 50]$ ms including queueing time, transmission time, retransmission time and propagation time. The delay requirement is taken in the range of $[120, 210]$ ms with an interval of 10 ms, which produces 10 delay requirement levels and the threshold of reliability is set to 0.5, and both the probability of delay and reliability constraint α and β are set to 95%. The size of a data packet is 150 bytes and is assumed to have an energy field that is updated during the packet transmission to calculate the total energy consumption in the network. We have applied different random seeds to generate different network configuration during the 10 runs. Each simulation lasted 900 sec and for the same setting, the four approaches are simulated for comparison.

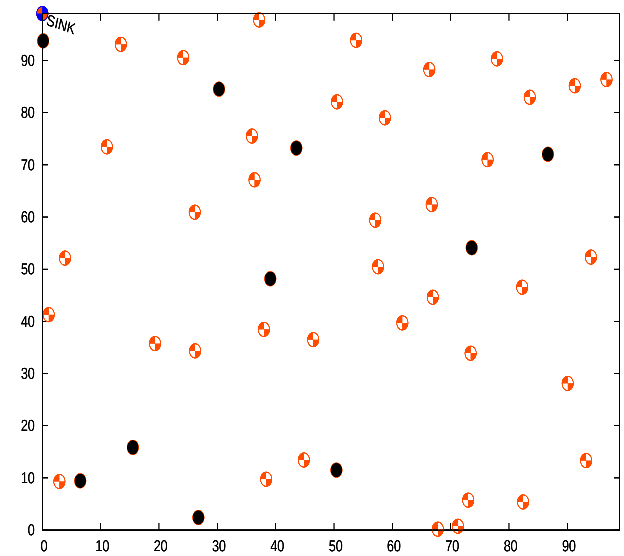
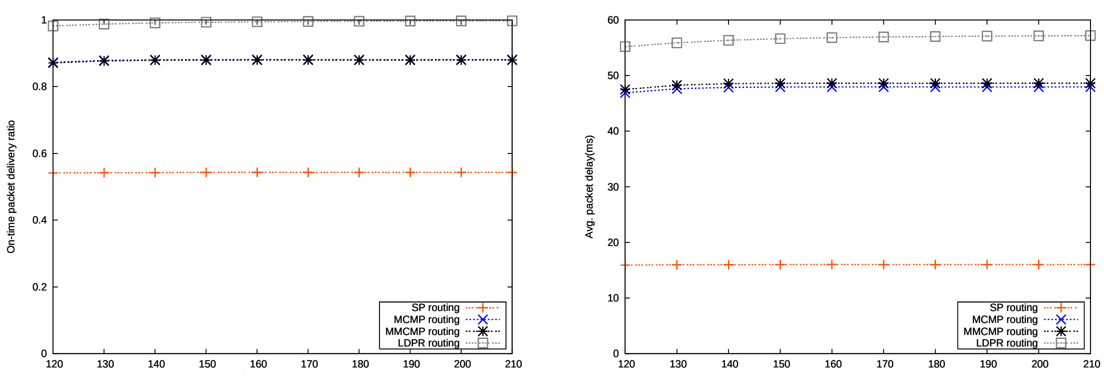


Figure 6.3: *50-Node Random Sensor Network Test*

6.5.2 Results comparison

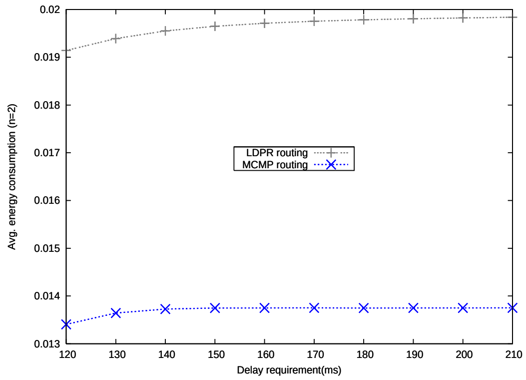
The results are depicted by the figures 6.4(a)-6.4(b) for delivery ratio and data delivery delay while the figures 6.5(a)-6.5(d) are used for the network energy consumption. Figures 6.6(a) and 6.6(b) refer to path length and path multiplicity.



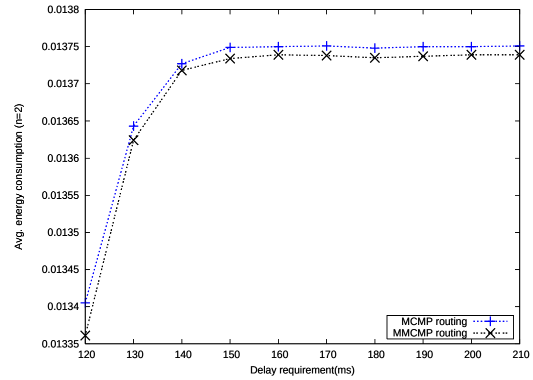
(a) *On-time Packet Delivery Ratio* (b) *Avg. End-to-End Packet Delay*

Figure 6.4: *Delivery Ratio and Data Delay Comparison*

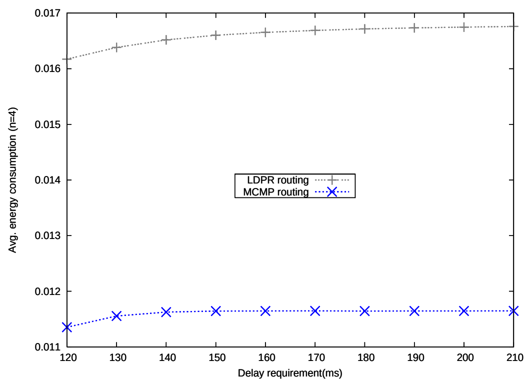
In term of delivery ratio, MMCMP and MCMP schemes perform equally, and outperform single path routing as it can be seen in figure 6.4(a). LDPR scheme of course achieves the best performance since it assumes that each sensor node has complete knowledge of



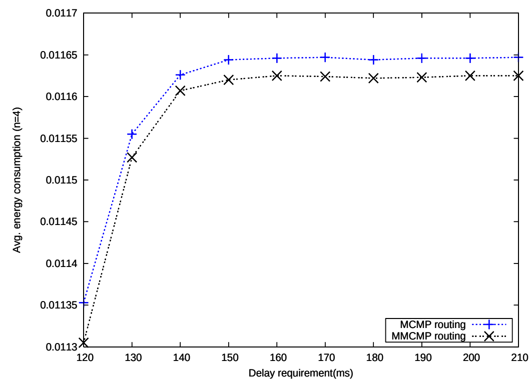
(a) *Energy for LDPR and MCMP* $(n = 2)$



(b) *Energy MCMP and MMCMP* $(n = 2)$

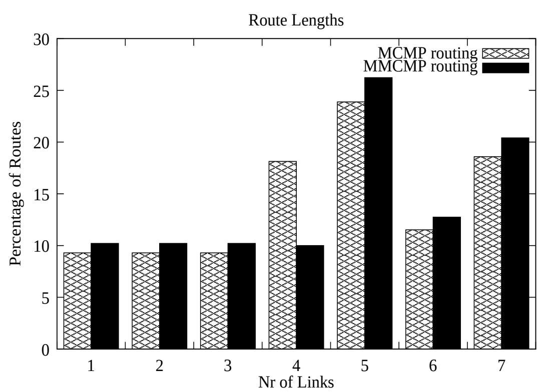


(c) *Energy for LDPR and MCMP* $(n = 4)$

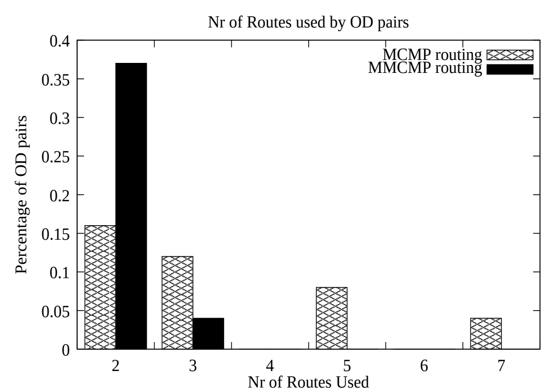


(d) *Energy MCMP and MMCMP* $(n = 4)$

Figure 6.5: *Energy Efficiency Comparison*



(a) *Route Lengths*



(b) *Route Multiplicity*

Figure 6.6: *Comparison of Routes used in terms of Length and Multiplicity*

the network topology. And consequently, LDPR scheme manage delay constraint better than MMCMP and MCMP schemes as shown in figure 6.4(b). The difference of average end-to-end delay between MMCMP and MCMP schemes is due to the fact that the paths used the two schemes may be different in term of number of hops.

Looking at the total energy consumed in the networks, it can be observed that the MMCMP scheme, as expected, performed better compared to MCMP scheme in terms of energy consumption as illustrated by figure 6.5(b) and 6.5(d). Consequently, energy saving is possible. Such a positive impact can be more obvious in dense topology.

Results in figure 6.6(a) reveal that MMCMP scheme uses longer paths (in terms of number of hops) compared to the MCMP scheme. Thus, paths used by MMCMP scheme are more likely to lead to higher end-to-end delays. This justifies the results depicted by the figure 6.4(b) on average end-to-end packet delay. Finally, the two schemes use approximatively 99.6% single paths, and when these algorithms start using more than one path, results are depicted by figure 6.6(b) which shows that MMCMP scheme optimizes better the number of paths used to send data to the base station than MCMP scheme. Thus, the MCMP scheme tends to consume more energy than MMCMP scheme. This is in agreement with the approach used by each scheme, and justifies the results in figures 6.5(b) and 6.5(d) concerning the network energy consumption.

6.6 Summary

In this chapter we analyzed the issue of using multi-path routing in wireless sensor networks and proposed the Modified Multi-Constrained Multi-Path routing (MMCMP) based on the scheme proposed by X. Huang and Y. Fang. The main idea driving MMCMP scheme is that in the context of wireless sensor networks, efficient resource usage not only means efficient bandwidth utilization, but also a minimal usage of energy in its strict term. This means that QoS support in wireless sensor networks should also consider QoS control besides QoS assurance in order to eliminate unnecessary energy consumption in data delivery. The efficiency of the proposed approach is verified through simulation which reveals that MMCMP approach outperforms the MCMP scheme in terms of energy consumption and quality of the paths used.

Chapter 7

Conclusions

This thesis presents multi-path routing schemes to achieve QoS routing in next generation IP wired networks using the MPLS technology and Wireless Sensor networks (WSNs). Building upon a two-steps approach where path finding is followed by traffic distribution, we propose two efficient traffic distributions schemes using Pontryagin Minimum Principle and a Game theoretical framework. We compare the two schemes to the widely known Flow deviation method. Experimental results reveal the robustness of our schemes compared to flow deviation on several performance indexes.

Building upon a previously proposed QoS provisioning benchmark model, we formulate the problem of routing sensed information in WSNs as both a path- and a link-based energy minimization problem subject to QoS routing constraints expressed in terms of reliability, delay and geo-spatial energy consumption. Using methods borrowed from the zero-one optimization framework, we solve the link-based problem and compare the performance achieved by its solution to the benchmark model. Simulation results reveal that our model outperforms the benchmark model in terms of energy consumption and quality of paths used to route the sensed information.

There is room to extend the work proposed in this thesis in different directions. These include

- **Multi-path IP recovery for MPLS networks**

  The failure of paths in the emerging MPLS networks leads to the re-routing of the carried traffic by these paths over recovery paths. Multi-path routing schemes similar to ours may be used in IP recovery to achieve faster rerouting by redefining the flow patterns and forwarding the failed traffic over a reduced set of paths. This may avoid the additional paths computations and resource reservations usually applied in MPLS recovery to achieve faster recovery. Achieving fast rerouting can become more crucial in routing conditions where the failed path is the one with the highest spare capacity since in this case the recovery process may require that the sum of the spare capacities of the remaining paths be strictly greater than the bandwidth used by the flow carried by the failed path. Appropriate multi-path recovery schemes can provide a solution to this problem. The design, evaluation of the performance and implementation of such schemes is a direction for future research.

- **Survivability in wireless sensor networks**

  We propose in this thesis a QoS multi-path routing scheme for wireless sensor networks where the sink or station node is assumed to be stationary. However, in certain circumstances such as battle field environments in military applications, the base station node and/or all the nodes of a WSN may be mobile. This tend to generate more interference, higher packet loss and error during the transmission. These impairments may affect the bandwidth and delay values of the paths. Also, the frequent position update of the station node and the propagation of that information through the network may adversely affect the system energy consumption. There is a need to extend the QoS multi-path routing scheme proposed in this thesis to consider sensor nodes mobility. This issue will be addressed by future research studies.

Bibliography

[1] Cisco. *Configuring OSPF*. Cisco Systems, Inc., http://www.cisco.com/univercd/cc/td /doc/product/software/ios122/122cgcr/fipr_c/ipcprt2/1cfospf.htm♯wp1000881, Nov. 2006.

[2] D. Xu, M. Chiang, and J. RexFord. *DEFT: Distributed Exponentially-weighted Flow Splitting*. IEEE INFOCOM, pp. 71-79, May 2007.

[3] X. Xiao, A. Hamman, B. Bailey, and L. Ni. *Traffic Engineering with MPLS in the Internet*. IEEE Network Magazine, pp. 28-33, March 2000.

[4] A.B. Bagula. *Traffic Engineering Next Generation IP Networks Using Gene Expression Programming*. Proceedings of the 2006 IEEE/IFIP Network Operations & Management Symposium, pp. 230-239, April 2006.

[5] A.B. Bagula. *Hybrid IGP+MPLS Routing in Next Generation IP Networks: An Online Traffic Engineering Model*. Lecture notes in Computer Science, vol. 3375, pp. 325-338, Feb. 2005.

[6] K. G. Mazandu, A.B. Bagula, and A. Muchanga. *On Using Differentiated Queueing to Support Traffic Engineering*. In Proceedings SA Telecommunication Networks and Applications Conference (SATNAC), Sept. 2006.

[7] A.B. Bagula. *Hybrid Routing in Next Generation IP Network*. Computer Communications 29 (2006), pp. 879-892, Aug. 2005.

[8] D. Awduche, J. Malcom, J. Agogbua, M. O'Dell, and J. McManus. *Internet RFC 2702*. Requirements for Traffic Engineering over MPLS, Sept. 1999.

[9] V. K. Goyal. *Multiple Description Coding: Compression Meets the Networks.* IEEE Signal Processing Magazine, vol. 18, pp. 74-93, Sept. 2001.

[10] X. Wu, B. Ma, and N. Sarshar. *Rainbow Network Flow of Multiple Description Codes.* NetCod & ISIT, 2005.

[11] J. Filipiak. *Modelling and Control of Dynamic Flows in Communication Networks.* Springer-verlag, ISBN 3-540-18292-6, 1988.

[12] B. Calvert, W. Salomon, and I. Ziedins. *Braess's Paradox in a Queueing Network with State Dependent Routing.* Journal of Applied Probability, vol. 34, pp. 134-154, 1997.

[13] J.E. Cohen and F.P. Kelly. *A Paradox of Congestion in a Queueing Network.* Journal of Applied Probability, vol. 27, pp. 730-734, 1990.

[14] A. Inoie, H. Kameda, and C.E. Touati. *Braess Paradox in Dynamic Routing for the Cohen-Kelly Network.* From Proceeding (496) Communications, Internet, and Information Technology, 2005.

[15] J.M. Wong and M. Unsoy. *Analysis of Flow Control in Computer Networks.* CCCNG Rep., E-54, pp. 1-37. University of Waterloo, 1976.

[16] A. Bagula, M. Botha, and A. E. Krzesinski. *Online Traffic Enginneering: The Least Interference Optimization Algorithm.* Proceedings of IEEE ICC2004, June 2004.

[17] A. Korolis, A. Lazar, and A. Orda. *Architecting noncooperative Networks.* IEEE Journal on Selected Area in Communications, vol. 13, no. 8, 1995.

[18] X. Huang and Y. Fang. *Multiconstrained QoS Multipath Routing in Wireless Sensor Networks.* ACM Wireless Networks (WINET), 2007.

[19] R. Banner and A. Orda. *Multipath Routing Algorithms for Congestion minimization.* In Proc. IFIP Networking, 2005.

[20] Y. Wang and Z. Wang. *Explicit Routing Algorithms For Internet Traffic Engineering.* In Proc. ICCCN'99, Oct. 1999.

[21] A. Sridharany, S. Bhattacharyya, C. Diot, R. Gueriny, J. Jetchevaz, and N. Taft. *On the impact of Aggregation on the Performance of the Traffic Aware Routing.* In Proc. 17^{th} International Teletraffic Congress, Brazil, Dec. 2001.

[22] J. Kim, C. Kim, S. Seok, and C. Kang. *Traffic Engineering Using Adaptive Multipath-Forwarding Against Dynamic Traffic in MPLS networks.* IEEE ICN, 2004.

[23] J. Kim, C. Kim, S. Seok, and Y. Lee. *A Constrained Multipath Traffic Engineering Scheme for MPLS networks.* IEEE ICC, 2002.

[24] J. Song, S. Kim, and M. Lee. *Dynamic Load Balancing in MPLS Networks.* Lecture notes in Computer Science, Springer Berlin/Heilberg, pp. 989-999, Oct. 2003.

[25] S. Nelakuditi and Z. Zhang. *On Selection of Paths for MultiPath Routing.* In Proc. IWQoS, Germany, 2001.

[26] T. H. Cheng and J. Chen. *Performance of Fast Bandwidth Reservation with Multipath Routing.* IEEE Proc. of Comm., vol. 145, no. 2, April 1998.

[27] Y. Zhong and X. Yuan. *Impact of Resource Reservation on the Distributed Multipath Quality of Service Routing Scheme.* In Proc. IWQoS, 2000.

[28] M. Pioro and M. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks.* Morgan Kaufmann, ISBN 0-12-557189-5, 2004.

[29] C. Villamizar. *OSPF Optimized Multipath (OSPF-OMP).* draft-ietf-ospf-omp-01, http://www3.ietf.org/proceedings/98dec/I-D/draft-ietf-ospf-omp-01.txt, Oct. 1998.

[30] C. Villamizar. *MPLS Optimized Multipath (MPLS-OMP).* Internet Draft, draft-ietf-mpls-omp-01, http://tools.ietf.org/html/draft-villamizar-mpls-omp-01, Feb. 1999.

[31] S. Shu. *Algorithms for Diverse Routing.* Network Technology Research Center, EEE School, Dec. 2001.

[32] M. T. Goodrich and R. Tamissia. *Data Structures & Algorithms in Java, 4^{th} Edition.* John Wiley & Sons, Inc., ISBN 0-471-73884-0, 2005.

[33] E. Parzen. *Stockastic Processes.* Holden-Day, San Francisco, 1962.

[34] M. H. Leslie. *Optimal Control: An Introduction to the Theory with Applications.* Oxford University Press Inc., 1991.

[35] D. Gross and C.M. Harris. *Fundamentals of Queueing Theory.* John Wiley & Sons, ISBN 0-471-32812-X, 1974.

[36] K. S. Trivedi. *Probability and Statistics with Reliability, Queuing and Computer Science Applications.* John Wiley & Sons, ISBN 0-471-33341-7, 2002.

[37] N. J. Gunther. *The Practical Performance Analyst.* McGraw Hill, ISBN 0-595-12674-X, 2000.

[38] Package: xmaxima (5.10.0-6ubuntu1). *http://packages.ubuntu.com/feisty/math/xmaxima.* July 2007.

[39] D. Fudenberg and J. Tirole. *Game Theory.* Massachusetts Institute of Technology Press, ISBN 0-262-06141-4, 1991.

[40] A. Akella, R. Karp, C. Papadimitrou, S. Seshan, and S. Shenker. *Selfish Behaviour and Stability of the Internet: A Game Theoretic Analysis of TCP.* in ACM SIGCOMM, 2002.

[41] T. Roughgarden and E. Tardos. *How bad is selfish routing.* in the 41st Annual Symposium on Foundations of Computer Science, 2000.

[42] K. Yamaoka and Y. Sakai. *A Packet Routing Method based on Game Theory.* Trans. Inst. Electronics, Information and Communication Engineers, B-I, vol. 49, no. 2, pp. 535-538, 1981.

[43] E. Altaman, T. Başar, and N. Shimkin. *Competitive Routing in Network With Polynomial Costs.* IEEE Transaction on Automatic Control, vol. 47, no. 1, pp. 92-96, Jan. 2002.

[44] E. Altaman, T. Başar, and R. Srikant. *Nash Equilibria for Combined Flow Control and Routing in Networks; Asymptotic behavior for a large number of users.* in Proc. 38th IEEE Conference on Decision and Control, vol. 4, pp. 4002-4007, Dec. 1999.

[45] R. El Azouzi, M. El Kamili, M. Abbad, and T. Başar. *Combined Competitive Flow Control and Routing in Multi-User Communication Network with hard side-constraints.* Dec. 2002.

[46] R. J. La and V. Anantharam. *Optimal Routing Control: Game Theoretic Approach.* IEEE Transactions on Automatic Control, vol. 47, no. 3, pp. 437-443, March 2002.

[47] A. Orda, R. Rom, and N. Shimkin. *Competitive Routing in Multiuser Communication Networks.* IEEE/ACM Transactions on Networking, vol. 1, pp. 510-512, Oct. 1993.

[48] A. Korolis, A. Lazar, and A. Orda. *Capacity Allocation under nonCooperative Routing.* IEEE Transactions on Automatic Control, 1994.

[49] A. Lazar, A. Orda, and D. E. Pendarakis. *Virtual Path Bandwidth Allocation in Multi-User Networks.* in Proceedings of the IEEE INFOCOM'95, (Boston, MA), pp. 312-320, April 1995.

[50] A.B. Bagula. *On Achieving Bandwidth-aware LSP/λSP Multiplexing/Separation in Multi-Layer Networks.* Journal of Selected Areas in Communications, Special Issues on Multi-Layer Networks, April 2006.

[51] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms and Applications.* Prentice Hall Inc., ISBN 0-13-617549-X, pp.79-81, 1993.

[52] G. L. Nemhauser, N. H. G. Rinnooy Kan, and M. J. Todd. *Optimization.* Elsevier Science Publishers B. V., ISBN 0-444-87281-1, pp. 236-238, 1989.

[53] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, ISBN 0-521-83378-7, 2004.

[54] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing* 2^{nd} *Edition.* Cambridge University Press, ISBN 0-521-75033-4, 2002.

[55] E. Süli and D. F. Mayers. *An Introduction to Numerical Analysis.* Cambridge University Press, ISBN 0-521-00794-1, 2003.

[56] J.E. Burns, T.J. Ott, Johan M. de Kock, and A.E. Krzesinski. *Path Selection and Bandwidth Allocation in MPLS Networks: a non-Linear Programming Approach*. Proceedings of SPIE, ITCom (4523), 2001.

[57] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall International Editions, UK, 1987.

[58] B. Hajek. *Communication Network Analysis*. notes for ECE 467, pp. 127-131, Dec. 2006.

[59] P. Garcia-Escable and V. Casares-Giner. *Location Management Strategies in Next Generation Communications Services Networks*. Wireless Networks-Communications' New Age, vol. V, no. 1, Feb. 2004.

[60] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. *Wireless Sensor Networks: a Survey*. Computer Networks 38(4), pp. 393-422, 2002.

[61] R. J. Fernandes. *Multipath Routing in Sensor Networks*. CS 552 Project, 2003.

[62] J. Liu and B. Li. *Distributed Topology Control in Wireless Sensor Networks with Asymetric Links*. in the Proceedings of IEEE Globecom, 2003.

[63] I. Stojmenovic and X. Lin. *Power-Aware Localized Routing in Wireless Networks*. IEEE Transaction on Parallel and Distributed Systems, vol. 12, no. 11, pp. 1122-1133, 2001.

[64] L. Li and J. Y. Halpern. *Minimum-Energy Mobile Wireless Networks Revisited*. In IEEE International Conference on Communications (ICC), vol. 1, pp. 278-283, June 2001.

[65] P. X. Liu and Y. Liu. *A Two-Hop Energy-Efficient Mesh Protocol for Wireless Sensor Network*. International Journal of Information Acquisition, vol. 1, no. 3, pp. 237-247, 2004.

[66] S. Dulman, J. Wu, and P. Havinga. *An Energy Efficient Multipath Routing Algorithm for Wireless Sensor Networks*. In Proceedings of the Wireless Communications and Networking Conference, 2003.

[67] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. *Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks.* ACM SIGMOBILE Mobile Computing and Communications Review, Vol 1, no. 2, 2001.

[68] G. Barrenechea and S. Servetto. *Constrained Random Walks on Random Graphs: Routing Algorithms for Large Scale Wireless Sensor Networks.* In Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, 2002.

[69] W. Li and C. G. Cassandras. *A minimum-Power Wireless Sensor Network Self-Deployment Scheme.* WCNC 2005 - IEEE Wireless Communications and Networking Conference, no. 1, pp. 1897-1902, March 2005.

[70] K. K. Leung, T. E. Klein, C. F. Mooney, and M. Haner. *Methods to Improve TCP Throughput in Wireless Networks with High Delay Variability.* Vehicular Technology Conference (VTC) IEEE 60th, vol. 4, pp. 3015-3019, Sept. 2004.

[71] H. A. Taha. *Integer Programming: Theory, Applications, and Computations.* Academic Press, Inc.,ISBN 0-12-682150-X, 1975.

[72] D. L. Kreher W. Kocay. *Graphs, algorithms, and optimization.* Chapman & Hall/CRC Press, ISBN 1-58488-396-0, 2005.

VITA

Gaston KUZAMUNU MAZANDU was born in Kinshasa, Democratic Republic of Congo former Zaïre on August 18, 1969, married to **Malungidi Mbambi Marie Paul**, and father of three children **Kuzamunu Mambote Jemima, Kuzamunu Mazandu Glodi, and Kuzamunu kinzuemi Keren**. He graduated with BSc Honours in Mathematics with a focus on Computer Science from University of Kinshasa in 1996. He was teacher of Mathematics at secondary school (1996-1999), teaching assistant at University of Kinshasa (1999-2004) and at Catholic Faculties of Kinshasa (2002-2004). He was admitted to study at the African Institute for Mathematical Sciences, Cape Town, South Africa on September, 2004 where he obtained a postgraduate diploma in applied mathematics in June 2005. He entered University of Stellenbosch, South Africa on July, 2005 and joined networking research group in Computer Science division, department of Mathematical Sciences. He concluded his research at University of Stellenbosch in September 2007 with a focus on network routing algorithms and obtained MSc degree in Computer Science.

This thesis was typeset with LaTeX by author.

E-mail addresses: gmazandu@cs.sun.ac.za

kuzamunu@aims.ac.za

mazandugaston@yahoo.fr

Home page: http://www.aims.ac.za/~kuzamunu