12-2002

Load Balancing and Congestion Avoidance Routing

Konstantinos N. Kokkinos
Western Michigan University

LOAD BALANCING AND CONGESTION AVOIDANCE ROUTING

by

Konstantinos N. Kokkinos

A Dissertation
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Doctor of Philosophy
Department of Computer Science

Western Michigan University
Kalamazoo, Michigan
December 2002

LOAD BALANCING AND CONGESTION AVOIDANCE ROUTING

Konstantinos N. Kokkinos, Ph.D.

Western Michigan University, 2002

Today's high speed backbone networks are expected to support a wide range of communication-intensive applications. One of the most important issues in Quality of Service (QoS) is efficient routing. Many QoS routing solutions have been published lately for different criteria of QoS requirements and resource constraints.

In this dissertation we focus on the design of regular network topologies and suggest efficient routing schemes to reduce the probability of hot spot creation in the network. Furthermore, we provide a detection of congestion mechanism that reroutes traffic to maintain balancing with small communication cost.

Several theoretical results relatively to network traffic balancing have been derived. Moreover, heuristic algorithms for the case of static rerouting without bandwidth guarantees and the case of QoS link state routing with bandwidth guarantees have been suggested.

A network simulator has been developed for the project experiments. The experimental results verify our theoretical model.

UMI Number: 3077379

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis advisor Dr. Dionysios Kountanis who started and kept me involved in to Computer Science research. Through all these tough years he has been like a father to me. Also to Dr. Ajay Gupta for the financial support from the department and his understanding. Also to Dr. Karlis Kaugars for his suggestions and his nice personality. A special thanks also to Dr. Gary Chartrand for his excellent comments for this dissertation manuscript.

Thanks to Dr. Elise De Donker for some excellent theoretical suggestions and for just being herself.

Great thanks to my wife Despina. Without her love and support I wouldn't being able to finish my work. And to my wonderful daughter Nikoletta and son Nicholas who have changed my life for ever and made me love more and think more.

Finally special thanks to my parents for the love and support throughout the years and to my parents in law for their compantionate love.

Konstantinos N. Kokkinos

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Dissertation motivation and general description

Networks are expected to have the necessary resources to accommodate communication demands at all times. However, the routing schemes used to direct traffic do not always result in delay-free communications. The reason is the existence of a dynamic behavior of user communication patterns.

The idea of redistributing network traffic to achieve link load balancing is very attractive. Our motivation comes from the fact that link load balancing reduces the probability of hot spot creation. Therefore, all network links are equally graded in resource availability. Achieving a load balance equilibrium at all times, while reducing the communication cost of individual transmissions is of great benefit. The reason is that at the same time, we increase the average throughput rate of the system and we reduce the message delays on intermediate links. A load balancing control on the network layer increases the network fault tolerance by reducing the number of slow-down occurrences due to congested links.

In this dissertation, we deal with the designing and the routing of regular network topologies to achieve link load balancing. Our approach has a global

flavor. Rather than minimizing the cost of individual transmissions, we consider all site pairwise communication demands. This is advantageous since, a global view of the system increases the reliability of a routing scheme.

We introduce an optimization function based on global network Quality of Service, (QoS), parameters. We prove that the minimization of this function at all times provides routing schemes which minimize the spread of link loads therefore, obtaining a balance equilibrium. This function is:

$$V \times C_a^2 \times L_m^2$$

where, V denotes the variance of link loads, C_a denotes the average communication cost obtained by a routing scheme R on a specific communication pattern and L_m denotes the maximum buffer load occurred due to R.

We prove that the problems of designing a regular topology and obtaining a routing scheme for that topology to minimize $V \times C_a^2 \times L_m^2$ for a set of network demands are NP-Complete problems. Therefore, we provide heuristic algorithms for the static case without bandwidth guarantees and the Quality of Service case with bandwidth guarantees.

We also prove that the minimization of the above function achieves a reduction of congestion for the network. We base our approach to the congestion metric findings by Monteiro et. al., [62]. This is the only congestion metric to our knowledge. We also provide a congestion avoidance mechanism which projects future network demands using the past and recent history of the system's com-

munication behavior.

In the rest of this chapter we provide the necessary background information relative to the subject, we review related literature and finally we inform about the organization of this manuscript.

Background

In the last few years networking demands have emerged to a point of exponential growth. The Internet use along with the tremendous increase of computer information exchange among various organizations demands for an internetwork infrastructure to accommodate all this growth. The utilization of local area networks and intra-nets needs also to be maximized by merging them. Therefore, efficient interconnection networks are necessary in order to accommodate multicomputing and multiprocessing.

Big advances in the area of hardware have alleviated a lot of the tremendous interconnection needs among Local Area Networks (LAN) and Wide Area Networks (WAN). The new technologies of Network Switches, Bridges and Routers are not enough though. The development of software is a major factor that not only can maximize the network utilization but also can create security barriers and fault tolerance mechanisms that guarantee this maximization of utilization at all times. If we accept the *Seven-Layer Open Systems Interconnection* model (OSI) as our standard, then emphasis must be given to the *Data Link Layer* and

the *Network Layer* in order to develop these mechanisms. The Network Layer is responsible for fast delivery of message transmissions from source sites to destination sites inside a network or across networks. Furthermore, it resolves the issues of incompatibility in terms of different machine architectures and different communication protocols. On the other hand the Data Link Layer is responsible of *framing* (message splitting to manageable data units), *flow control*, *error control* and the *access control* of the networks.

The development of efficient routing algorithms is necessary in order to impose a good flow control of packets across the network links. This also increases the reliability of the network in terms of clotting prevention and error occurrence. Extended research has proven the difficulty of this task. Since networks embed a dynamic behavior relative to traffic needs across time, similar to the one that Figure 1 shows illustrates, the solution to the above task is hidden in mechanisms of dynamic nature. We have to consider the current state of the network and dynamically reconfigure the hardware for rerouting messages such that we always minimize the probability of congestion.

Routing is the technique used to completely specify the data flow of the network as well as the algorithm of how each network site communicates with each other. The routing control of a distributed system may be solely dependent on a master CPU (centralized approach) or it may be distributed among the various sites (distributed approach). Regardless of the two basic types above, we

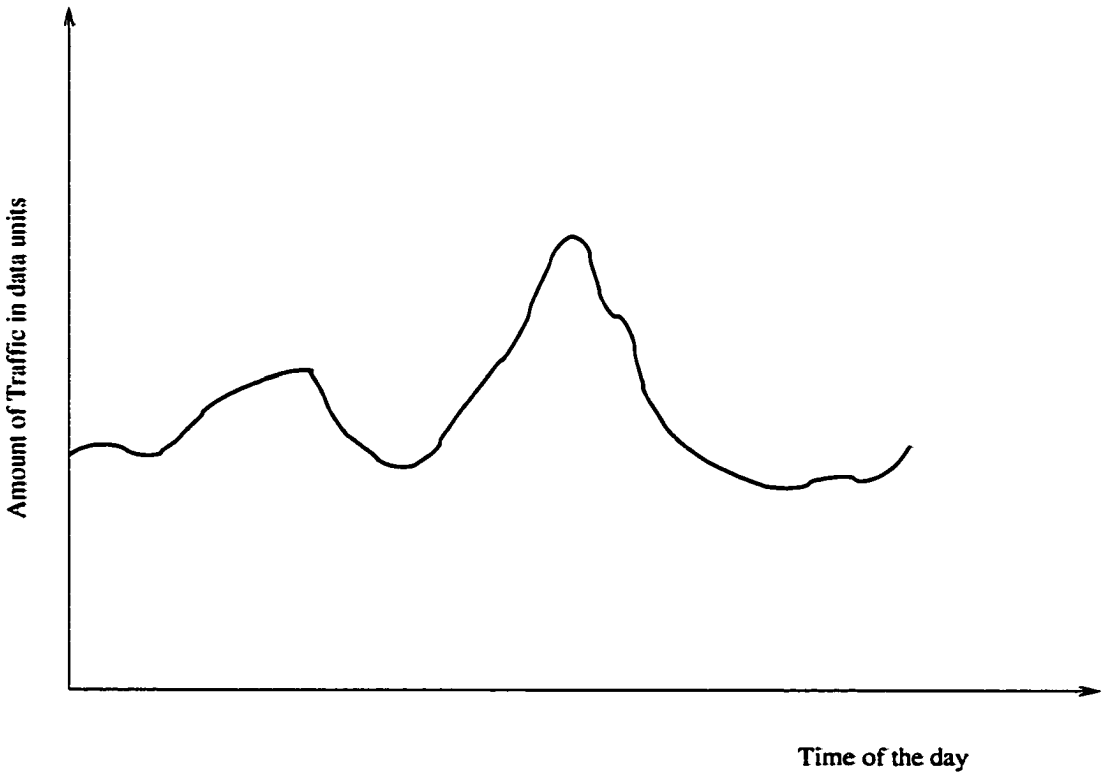

Figure 1. Dynamic traffic network behavior across time.

can characterize routing techniques as *static* or *adaptive*. Static routing does not depend on the workload and the data flow of the communication links over time. It is decided prior to the network usage and never changes according to the network's needs, performance and fault rate. This technique is also called *oblivious*. On the other hand, adaptive routing can dynamically modify the communication pattern among network sites when it detects low performance or high communication overhead. Other approaches for routing have also been attempted over the years.

An example is the case of *chaotic* routing which is shown to be applicable to all finite size networks of bounded degree with bi-directionally connected links [9]. This design allows messages to follow minimum latency paths when the network load is light and provides sufficient buffering to ensure high throughput when the network load is heavy. Even though the technique has not been proven to be the most efficient way to route messages, it provides us with a good average communication cost, including the case when the network load is heavy.

Routing problems have been widely studied over the years. The two primary reasons for this is the existence of a great number of applications in networking that can be utilized with better routing algorithms and the communication overhead decrease potential they offer. However, most of the existing research focuses on specific problems that can arise in networking such as, how to increase fault tolerance when a node fails or a link fails, how to minimize the communication cost for broadcasting across the network, etc. Moreover, a big repertoire of studies deal with routing issues on just one specific type of static or dynamic networks.

In this chapter, we classify the existing research on routing techniques into the following four major categories:

1. Research developed on static and adaptive routing techniques for multicomputer systems.

2. Research developed on the adaptation of static routing protocols into

dynamic ones.

3. Research developed on flow control techniques, deadlock avoidance techniques and also basic message transmission operations such as broadcasting, gossiping, multi-casting, etc.

4. Research developed on creation of dynamic and static routing schemes that preserve a load balancing in the network.

In each of the following four sections, we explore these four research development areas described above.

Static and adaptive routing on multicomputer and multiprocessor systems

For both the multicomputer and multiprocessor machines, packet routing has emerged as the most widely used switching technique. For the packet routing, the message is divided into packets that are independently routed toward their destination. The destination is encoded in the header of each packet. The main advantage of this technique is that the channel resource is occupied only when a packet is transmitted. On the other hand, the drawback is that, since the packet is stored entirely at each intermediate node, the time to transmit the packet from source to destination is directly proportional to the number of hops in the path. Furthermore, each node needs a buffer to hold at least one packet. Reduction of the transmission time has come by the introduction of a technique called *virtual-cut-through* [48]. The idea is simple. We store the packet only if the next link

for the transmission is occupied by another packet. The time is tremendously decreased. The only disadvantage is the increase of the storage requirements since all nodes must have enough buffer space for multiple packets that are blocked.

Wormhole routing is a variant of the virtual-cut-through technique. Here, packets are transmitted in units of *flits*, the smallest units of a message on which flow control can be performed. The research therefore moved to-wards the direction of worm hole routers. A detailed survey for the case of the worm hole routing is presented in [15].

For static or deterministic routing, the path from source to destination is determined by the current node address and the destination node address. Deadlocks are avoided by ordering the paths that a message needs to traverse. The problem though with the static routing is the distribution of the routing tables and the space needed for those in each node. Work about the efficiency of space of the routing tables is presented in [30] and in [28]. Moreover, Buhrman et. al. in [12] show that for almost all nets, $\Theta(n^2)$ bits are necessary and sufficient for shortest paths routing among n-nodes. The authors also prove that for worst case static routing schemes constructed for explicit graphs, $\Omega(n^2 log n)$ is a lower bound on the total space requirements needed, with n being the number of sites participating. The techniques used are incompressibility arguments based on the Kolmogorov complexity [51]. In addition to the optimality of the routing tables a work by Degermark et al. [23] presents the use of an interval binary tree based

data structure to create small forwarding tables for fast routing lookups. The strategy is software oriented and the experimental results show that it is fast enough to support routing at gigabit connections.

The research of adaptive routing is divided into two major areas: The area of creation of *fully adaptive algorithms* and the area of *partially adaptive algorithms*. Major contributions toward this research are summarized into an extensive survey by Mohapatra in [61]. Mohapatra provides a good classification of the routing algorithms and router characteristics for multicomputer systems.

The former case of fully adaptive algorithms lets a message use all possible physical paths between source and destination. Here the deadlocks are usually avoided with the use of *virtual channels*. Therefore the classification of fully adaptive algorithms is on the basis of the number of channels required per physical channel. Representative research work on fully adaptive routing techniques includes Linder and Harder [56], Dally and Aoki in [21] and Schwiebert and Jayasimha in [73].

In the case of partially adaptive algorithms, we allow routing freedom to be traded for router speed while assuring deadlock freedom. Partially adaptive algorithms use only a subset of the physical channels between communicating pairs. The most representative research to our knowledge is the *turn model* for adaptive routing suggested in [34]. This model imposes a method of designing worm hole routing algorithms without adding new physical or virtual links. However, the

optimization in performance is done by analyzing, allowing and controlling cycles in the network paths for packetization. The authors focus on the n-dimensional mesh paradigm but the method can be extended to arbitrary networks.

Adaptation of static routing into a dynamic routing

In the case of dynamic routing there have been two models developed that analyze the network performance in terms of injection of packets of an injection rate λ. These models are the *stochastic* and the *adversarial* model.

In the stochastic model the packets are injected by a set of generators, each of them mapped to one of the nodes in the network. We do not impose any restriction in terms of the relationship between the number of generators and the number of nodes in the net. For each packet the generator associated with one node randomly selects a destination according to an arbitrary, fixed probability distribution.

In the adversarial model, an adversary is allowed to demand network bandwidth up to a prescribed injection rate λ. For any path length w with $\lambda > 0$, an adversary is called *bounded adversary of rate* (w, λ) if for all edges e and all time intervals I of length w, it injects no more than $\lambda.w$ packets during I that require to pass e.

Not much knowledge exists in terms of how to transform static routing protocols into dynamic ones. The first result is shown in [70]. There, the authors

deal with a model called *Growing rank Protocol* where they try to adapt static networking into a dynamic one by packet injection rate $\lambda < (\frac{1}{e})$, where λ is the injection rate and e is the number of edges.

Also lately, a new model called *adversarial Queuing Theory* has emerged. Borodin et. al. in [10], show several stability results for greedy protocols of directed cycles. Also Andrews in [1] presents a transformation of the static protocol presented in [55] into a dynamic protocol that is stable for any admissible injection rate.

The latest work to our knowledge corresponds to the adaptation of Store-and-Forward Protocols by [71]. The authors here investigate how algorithms that have been designed for the case that all packets are injected at the same rate can be adapted to more realistic scenarios in which packets are continuously injected into the network. Their basic result is a dynamic routing algorithm for layered networks that is stable for arbitrary admissible injection rates and that works with packet buffers of size depending solely on the injection rate and the node degree, but not on the size of the network.

Message transmissions and deadlock avoidance techniques

Two of the most important issues of information dissemination are the ones that deal with the basic message transmission operations and the deadlock avoidance techniques.

Basic message transmission operations include the case of *broadcasting*, the case of *gossiping* and the case of *multicasting*. We discuss these operations individually and comment on the available research regarding these issues.

The operation of broadcasting in networks is defined as the process of transmission of a message from a single node to every other node of the network. Gossiping is defined as the situation in a network where all nodes have some piece of information, a gossip, that they want to exchange so that all nodes will be aware of all of the information at the end of the process. To our knowledge, the most recent survey that summarizes the research in the operations of broadcasting and gossiping up to 1986 is presented in Hedetniemi et al. [36]. Since then, other work in broadcasting deals mainly with the minimization on the number of hops needed to broadcast a message. This research though only attempts to find approximation algorithms to solve the problem since this has been proved to be NP-Hard [75]. Even for particular classes of network graphs the problem still remains NP-Hard as it is shown in [60]

The most recent work to our knowledge on the problems of broadcasting and gossiping belongs to Fraigniaud et. al. [31]. The authors focus on two basic measurements: round complexity and step complexity. They present a polynomial approximation algorithm for broadcasting large messages. This algorithm is based on the construction of λ *edge-disjoint* spanning trees in the considered graph (λ denotes the edge connectivity i.e., the minimum number of edges whose

destruction disconnects the graph). Their results include the cases of complete graphs, hypercubes, meshes and CCC's. For the case of gossiping individually also refer to [26], [27] and most recently to [52].

Multicasting on the other hand is a kind of group communication which requires simultaneous transmission of messages from a source to a group of destinations. There are two steps for a multicast establishment: routing and configuration of the individual connections. Routing here follows the methods extensively discussed above. The configuration of the connections includes specification of a multicast tree for transmission, reservation of network resources etc. Finding the minimum cost tree is a well known NP-Complete problem called *The Steiner Problem* [33]. Because of the inability to find optimal solutions to the Steiner problem in polynomial time, many approximation algorithms have been developed. The most recent representative work in multicasting operations appears in [7], [46] and [45].

A similar problem to broadcasting is the derivation of a transmission scheme that a host processor uses in order to send identical messages to all other processors [37]. More specifically, we allow the processors to receive the message in at most t time units and require that the host processor sends the original message at most s times. This scheme is called *(t-s)-transmitting scheme* where both t and s are supposed to be minimized. The authors present some optimal schemes for linear arrays, rings, complete binary trees, stars and de Bruijn graphs.

Another provision of good network designing and routing is the deadlock avoidance technique used. Without deadlock avoidance, we often have multiple node failures due to inability of transmitting the buffered messages. Despite the wide variety of buffer reservation algorithms for deadlock avoidance, they all share a common approach: the removal of the adaptivity and the creation of an oblivious routing that incorporates a total ordering of the buffers such that all packets move to a higher rank buffers at any time. It has been shown in Cypher et al. [20] that every deadlock-free adaptive routing algorithm contains a deadlock-free oblivious routing algorithm within it.

For the issue of deadlocks in networks, a large repertoire of research has been developed. Representative works include the research papers by Cypher in [19], Awerbuch et. al. in [5] and the most recent work in [54]. The latter is the most representative since here the authors prove that the Deadlock-Free Routing problem is NP-Complete by using a reduction from the well known 3-SAT NP-Complete problem. The authors also provide with some good approximation algorithms based on the *Merge Paradigm*, a generalization of the $M3$ algorithm proposed by Foulser et. al. [29].

Routing schemes and preservation of network load balancing

In this section we provide a review of the literature in terms of the inter-related work of balanced graphs, balancing traffic in networks and corresponding

routing schemes to accommodate that. We believe that the paper references in this section, even though some of them are not related, give a good starting point of the research problems we consider in this dissertation.

Balanced graphs are discussed in [67]. A balanced graph G is defined as the one where the maximum ratio of links to nodes, taken over all subgraphs of G occurs at G itself. The author used the max-flow/min-cut theorem to prove a good characterization of balanced graphs. This characterization is then applied to some results on how balanced graphs may by combined to form a larger balanced graph.

In [68] there is a presentation of a random model for message transmission through a network with unreliable nodes and links. The author's assumption is that nodes and links independently operate with probabilities a and b respectively. He also derives the following performance measures: the probability of reaching k-nodes by transmission m, the probability an arbitrary operable node is reached on the m^{th} transmission and the probability that d transmissions are required to reach all operable nodes.

Jaillet in [43] dealt with the problem of finding shortest paths from a source node to a sink node in a complete network. According to the problem instance, at any given time of the routing scheme, the sequence of nodes is preserved but only the permissible nodes are traversed, the others are skipped. This problem is defined as the *Probabilistic Shortest Path Problem*, (PSPP). The author shows

that the problem is NP-Hard and develops some approximation algorithms for some special cases. Similar problem also is the *Probabilistic Minimum Spanning Tree Problem* found in [11].

Benavent et. al. in [50] have dealt with the *Capacitated Arc Routing Problem*, (CARP). The CARP is defined as follows: Let $G = (S, E)$ be a connected graph. For every edge $e \in G$ there exists a load $q_e \geq 0$ and a traversing cost $c_e \geq 0$ associated. Given vertex 1 which represents the depot, the CARP consists of finding a set of routes with minimum total cost, such that each route contains the depot, each edge with positive load is serviced exactly once and, the capacity does not exceed a certain bound. The general case of CARP is NP-Hard even though the authors refer to some polynomial special cases. The authors find some new approximation algorithms for CARP with lower bounds.

In [25] the author presents a *Self Stabilizing System* which is a distributed system that can tolerate any number and any type of faults. After each fault occurs, the system converges into a legitimate behavior. It is assumed though that no additional faults occur during this convergence. Two protocols of stabilization are described: the Multiple BFS Trees protocol and the Counting Protocol. Both protocols include rerouting algorithms, leader election algorithms and topology updates.

The authors in [40] formulate several fixed change network design models, capacitated or uncapacitated, directed or undirected, possibly with staircase costs

and survivability requirements. They propose a common solution approach based on Lagrangean relaxation, sub-gradient optimization incorporated into a branch and bound framework.

The authors in [24] present an extensive theoretical analysis of the Load Balancing Problem, LBP, in a network of processing units. Their objective is to minimize the time spent to finish all jobs. They categorize their analysis into a centralized approach and a distributed approach. For the centralized LBP, all nodes have complete information of the load distribution over the network. They show that the problem is NP-Complete when jobs are of different sizes even with pre-emptive scheduling and routing. In the case that the jobs are of the same size, they give a polynomial algorithm using network-flow mechanisms. This algorithm can be extended to approximate solutions for jobs of different sizes. They apply their theoretical results into three network topologies: Complete graphs, rings and Hierarchical *k-ary* trees.

The work in [59] describes the experimental analysis of one common load measure in regular distributed systems, the UNIX-load average and its relationship to the run time of computation bound parallel programs.

The authors in [35] introduce a method for predicting the network traffic that will be generated by collaborative virtual environment applications with varying numbers of participants. The controlled traffic measurements combined with an analysis of the application architecture and network topology result in a

system behavior model. Their objective is to make a network model that safely predicts mean and peak levels of traffic on a given link. The mean traffic on a link is predicted from the average frequencies of events. The peak traffic is given by the worst case situation, where network users act as much as possible.

In [77] the authors present the Efficient Reservation Virtual Circuit, (ERVC) protocol as a competitive candidate for network protocols in future high speed networks. Their goal is to design a protocol that would use the capacity efficiently in the presence of large propagation delays, and avoid unnecessarily prolonged set-up phases. Furthermore, the authors claim that the protocol does particularly well in terms of utilizing capacity efficiently and that its blocking performance is significantly better than of regular reservation schemes.

Day et al. in [22] present a uniform mathematical characterization of interconnection network classes referred to as *product-closed* networks, PCN. A large variety of regular networks fall under this category. The study shows that an unlimited number of networks can also be defined under this type. A particular feature of PCN's is their closure under the Cartesian Product. The authors evaluate a number of common network properties like the degree, the diameter, the connectivity and the fault diameter for all PCN's and show simple distributed routings, complete sets of disjoint paths, attractive embeddings, distributed broadcasting and fault tolerance mechanisms.

Lately, there is some research which deals with the case of permutation

routing via matchings [41]. Here the routing problem is considered as follows: Initially each node of the network contains exactly one packet. Moreover, each node is the destination of only one packet. Therefore, the initial state can be considered as a permutation of packets. The packets are routed in their destination nodes by a sequence of steps. In one step, each packet can either remain in its current location or it can be swapped with a neighbor, i.e., the step is determined by a matching of the participating nodes. An algorithm in [66] gives the routing of all packets to their destination. The authors prove that this algorithm is bounded by the value of $\frac{13}{5n}$, where n is the number of sites in the network. However, Hoyer and Larsen illustrated a better algorithm in [41] for the same problem and proved a bound of $2n - 3$ with $n \geq 2$.

Organization of the dissertation

As we mentioned in the beginning of this chapter, we consider two basic problems.

The first problem is the design of a regular network topology and a static routing scheme for this topology to minimize the probability of network congestion. The input set for the design of a network of n sites is all the pairwise traffic demands. These traffic demands are represented as average frequencies of communication between any pair of sites and indicate the projected network behavior.

The network design assumes half duplex links. We also assume the existence of a collision detection mechanism as we concentrate on the topology design and the derivation of routing algorithms. We transform this problem to a minimization problem by introducing an optimization function that minimizes the probability of network hot spot creation. This optimization function is the product

$$V \times C_a^2 \times L_m^2$$

where, V is the variance of link loads, C_a is the average communication cost and L_m represents the maximum load among all network links. Moreover, the optimization of $V \times C_a^2 \times L_m^2$ guarantees balancing of link usage across the network while minimizing the average path hop count.

In Chapter II we formulate the network design and routing problem and we prove that this problem is NP-Complete. For this proof we consider the special case of ring networks with restricted communication patterns.

However, note that the dynamic communication pattern behavior of networks in use may still cause congestion when a static routing scheme is applied. That leads to the second problem in consideration which is the detection of congestion mechanism and the triggering of rerouting procedures to accommodate the new traffic demands. The goal remains the same, that is to achieve a routing scheme that minimizes the probability of congestion by minimizing the product $V \times C_a^2 \times L_m^2$. This problem may be seen as a subproblem of the first problem applied on existing network topologies. In Chapter III we prove that the rerouting

problem is also an NP-Complete problem. The methodology in proving the NP-Completeness is based on a construction of regular graphs resulting from instances of the *Three Dimensional Matching Problem*, ($3DM$), a well known NP-Complete problem.

In Chapter IV we adapt a definition of network congestion measurement based on the deviation of symmetric *Quality of Service* parameters. This definition is introduced by Monteiro et al. [62] and measures how individual network users experience Quality of Service degradation in a congested network. Assuming this congestion definition we show that the minimization of $V \times C_a^2 \times L_m^2$ also guarantees network congestion minimization with high probability.

Furthermore in Chapter IV we describe a methodology of congestion detection in the network. This process is based on the global network state which is known by all sites. We project future communication frequencies under the assumption that traffic demand behaviors are repeated over time. We compute these frequencies as an exponential average of their history in order to capture behavior localities. We also comment on the effect of the interval of this exponential average sampling.

We finally present the data structures involved for this detection and calculate the necessary storage needed in each site including the routing tables.

In Chapter V we proceed with the proposition of routing algorithms. We distinguish two cases: *Off-Line Routing without Bandwidth Guarantees* and *QoS*

Routing with Bandwidth Guarantees.

For the first case we introduce three routing heuristics to minimize the optimization function $V \times C_a^2 \times L_m^2$. All the heuristics are based on the greedy computational method. For the three approximation algorithms we compute their computational complexity and calculate their upper bounds in the ring topology.

For the second case we introduce a hybrid model of static and dynamic routing. This model uses the precomputed paths of the static routing heuristics above when the network experiences light volumes of traffic. This results in a reduction of the time needed for site probing when establishing sessions between any two network sites. However when sessions cannot be established, the model switches from static to dynamic with a backtracking routing method that avoids path cycles while balancing data flow.

The verification of our findings is justified experimentally in Chapter VI. We have developed a network simulator that integrates the proposed heuristics on popular regular network topologies and under various distributions of traffic data. The QoS parameters we considered were the average throughput rate, the average end-to-end delay, the average delay jitter and the average number of reroutings in a traffic data set. The simulations show the interrelation of the routing heuristics, network topologies and traffic distributions and their effect on the above QoS criteria.

CHAPTER II

NETWORK DESIGN AND ROUTING WITH COMMUNICATION BALANCING CONSTRAINTS

Introduction

The network design optimization problem is of great importance in the telecommunication and computer network industry. Such networks involve large financial investments and therefore their prototyping and design is essential for their practicality and efficiency.

Usually new networks are created by merging some existing ones. The design of old networks was based on the technology availability, constraints and prototype specifications at the time of design. The new network must integrate the restrictions of the old models into the new design. An example of such a situation is to integrate an old copper cable network by attaching it to a backbone network. The old topology could be a tree (possible Steiner) having as a goal to minimize the transfer delay. On the other hand in today's technologies of fiber optical, wireless or satellite backbone networks transfer delays have decreased tremendously. However other criteria drive network designs nowadays. These include maximization of bandwidth, minimization of congestion, availability of

bandwidth at all times and balancing of linkage utilization.

Modern network designs must not depend on short-term planning criteria. A designer must not leave unaccounted the issues of further network extensibility and traffic increase. Most designers temporarily solve these problems by increasing the link bandwidth. However, in the case of designing a high traffic fiber optical or wireless backbone network, the primary cost component concerns the canalization of fibers and the synchronization of frequencies to achieve maximum linkage utilization with minimum delays, as opposed to the fiber capacity increase.

Clear criteria and goals must be set before the network design starts. These criteria must apply not only in the time of design but also must incorporate hardware and software solutions when network expansion is considered. Furthermore these criteria must not be set as empirical "rules of thumb" but must provide models for efficiently finding good solutions.

There is a lot of research lately in designing networks that satisfy *Quality of Service*, QOS criteria. Typical references include [38, 39]. The authors provide Lagrangean Heuristics and Branch-and-Bound methods in designing capacitated and uncapacitated networks. However, most of the techniques deal with minimizing the traveling distance of information from site to site in a network. This translates into the minimization of the average communication cost. Unfortunately this criterion by itself is not enough to satisfy performance, especially in networks that claim time delay guarantees in information transferring. The reason

is simple: It all depends on the user behavior, a factor inherently dynamic and of intractable nature. The result is the creation of congestion on heavily used links while under utilizing others which causes additional transfer slowdown.

We concentrate in the design of backbone high speed networks. In today's technology all these networks are regular, with the ring topology to be the most popular due to its simplicity and cost. Our approach does not exclude other regular topologies. On the contrary our analysis is expanded to other topologies of the regular class of graphs including Hypercubes and Z-Cubes. As we will show in the following sections of this chapter, the design problem is equal in difficulty to the communication cost minimization problem. This was somewhat expected since *QoS* mechanisms become more computationally intractable as we increase the number of criteria set. We refer to [3] and [18] as well as the citations therein for this claim.

This dissertation is differentiated from other approaches by integrating the balancing of data flow as an additional criterion along with the minimization of communication cost criterion. This is obtained by monitoring the user pattern behavior and the localities in change of that behavior. As we have mentioned, the goal is not simply to minimize communication cost but to do this without compromising time delay guarantees due to high traffic.

Our research provides algorithmic solutions to minimize the variance of network link loads while at the same time reducing the average communication

cost between sites of the network and reducing the maximum load at any link. By doing this we produce durable as well as fast networks to accommodate high traffic. Our goal is the provision of a software methodology to automatically adapt to the current network load by monitoring the user behavior. This is done by rerouting algorithms which are triggered by a detection of imbalance mechanism.

The optimization function used in order to obtain traffic balance is the function $V \times C_a^2 \times L_m^2$ where, V denotes the variance of link loads, C_a denotes the average communication cost for the network relatively to traffic demands and a routing scheme and, L_m denotes the maximum traffic load in any link.

Before we describe in detail the network design and routing problem we justify the choice of the above optimization function to achieve link load balancing. In the following sections, we prove the NP-Completeness of the network design and routing problem with the help of several lemmas stated in the third section.

Justification of the optimization function choice

The purpose of minimizing the variance of all link loads in the network is to obtain a routing scheme that provides minimum possible spreads among traffic loads. Someone may also consider the minimization of the standard deviation of these loads. However, it is convenient to choose the variance since, the avoidance of the square root helps to convert our problem to a integer number theory problem.

The minimization of variance by itself however is not enough since, it does

not guarantee congestion reduction. The reason is that in order to minimize the variance of loads, a routing scheme may increase the path lengths of individual transmissions charging the network links with unnecessary load. We like to achieve this balance equilibrium while we also minimize the maximum buffer load occurring for a specific communication pattern. This justifies the inclusion of L_m to the optimization function.

Moreover, if we assume a routing scheme that results in a specific maximum buffer load, we also like to obtain balancing such that, the average communication cost is minimized while the L_m value restriction holds. Thus, we try to achieve the best possible balance within the limits of a load maximum but at the same time minimizing the individual path lengths of transmissions.

Note that among the parameters taken, V is always the smallest in value, C_a^2 follows and finally, L_m^2 is the maximum of the three. However, the choice of taking their product makes them all equal in importance. Someone may argue that, the choice of taken the summation of the above parameters achieves also the same result. This is not the case though since, any routing algorithm would prioritize the minimization of L_m since it contributes more to the summation. That restriction along with the minimization of C_a can cause greater variance values and lead to routing schemes which tend not to use all the available links.

The network design and routing problem

In this section we formalize the Network Design and Routing Problem, ($NDRP$). The input data to the problem is the number of sites for the network to be designed and the set of pairwise projected frequencies of communication between any two sites. The goal is to create a topology and a routing scheme for that topology which relatively balances the network flow while reduces the average communication cost and the maximum size for each of the buffers that the routing scheme imposes. We first define two terms: *Frequency of Communication* and *Frequency of Use*. The first term is a measure of the communication pattern between any pair of sites in the network. The second term is a measure of the buffer usage (or the corresponding traffic load) that the routing scheme imposes to each output buffer of the network.

Definition 1 : *Frequency of Communication* between any two sites (v_i, v_j) in the network is defined to be a number $f_{i,j} \in Z^+$ indicating the directed communication demand of the two sites in a time interval.

We assume that in general $f_{i,j} \neq f_{j,i}$, i.e., we consider end-to-end communications as opposed to pairwise symmetric communications. Note that, we can represent these specifications as a complete weighted graph with $f_{i,j}$ to be the weight for the directed edge (v_i, v_j).

The term Frequency of Use is used to define half duplex traffic in a network

link:

Definition 2 : *Frequency of Use* of a buffer (v_i, v_j) associated with a link $\{v_i, v_j\}$ in a network is defined to be a number $u_{i,j} \in Z^+$ indicating the traffic passed through it during a time interval and for given routing scheme.

If we assume bidirectional half duplex links, every link $\{i, j\}$ of the network defines two output queues, one in each direction. Therefore, if we refer to frequencies of use we actually refer to the traffic loads for each buffer in the network. For the remainder of this manuscript whenever we refer to the term frequency of use or traffic load we refer to the load on bidirectional network buffers of the network.

Note that, a network of $|E|$ links corresponds to a frequency of use set U_e of cardinality $2|E|$. Note also that, a shortest path routing in a complete network corresponds to buffer frequencies of use $u_{i,j} = f_{i,j}$.

An optimal design and a routing scheme according to our criteria should provide a network of balanced frequencies of use U_e. Moreover, this should be achieved without compromising the performance of the network due to congestion and other factors. Therefore, this traffic balance should be done while minimization of the communication cost and minimization of the maximum possible traffic occurs in each buffer.

Definition 3 : Let $G = (S, E, F_e)$ be a network of S sites and E bidirectional links with traffic demands F_e and R_G be a routing scheme that routes these demands. We define *Maximum Buffer Load*, denoted as L_m, to be the maximum buffer size

imposed by R_G for the communication pattern F_e.

The frequencies of communication between potential network sites may be projected. These projected frequencies reflect specific user behavior patterns. With today's network monitor mechanisms, we can roughly estimate the amount of traffic between network sites by examining similar networks. Moreover, the number of sites in each of the LAN's gives a good indication of traffic demands among the sites of the backbone network. For the instance of high speed backbone networks, these statistics are widely available since, they are published from the big telecommunication and Internet provider companies.

We now define the $NDRP$. The form used in the definition of the problem is similar to the one that Garey and Johnson [32] use, i.e., the form of *instance-question*:

Instance of $NDRP$: A complete graph $G = (S, E, F_e)$ where, S is the set of network sites, E is the set of bidirectional and half duplex links, F_e is the set of frequencies of communication for G and bounds $B \in Z^+$, $d \in Z^+$ with $d \leq |S|-1$.

Question : Is there a regular subgraph topology $G' = (S, E')$ of G with degree d and a routing scheme $R_{G'} = \{p_{s_i,s_j} \mid p_{s_i,s_j} = \text{a path } \forall\ s_i, s_j \in S \text{ with } i \neq j\}$ that results into a set U_e of frequencies of use such that,

$$\sum_{i=1}^{2|E'|} \sum_{j>i}^{2|E'|} (u_i - u_j)^2 \times \left(\sum_{i=1}^{2|E'|} u_i \right)^2 \times L_m^2 \leq B\,(d|S|)^4\ ?$$

Note that for G' and $R_{G'}$ the average communication cost is defined to be the summation of the traffic loads of all network buffers divided by the number of

available buffers, i.e.,

$$C_a = \frac{\sum_{i=1}^{2|E'|} u_i}{2|E'|}$$

Let $n = 2|E'|$ the set of buffers for the regular network G'. Then the variance V for a set U_e of values $\{u_1, u_2, \cdots, u_n\}$ is:

$$\begin{aligned} V &= \frac{1}{n} \times \sum_{i=1}^{n} (u_i - C_a)^2 \\ &= \frac{1}{n}(\sum_{i=1}^{n} u_i^2 - nC_a^2) \\ &= \frac{1}{n}(\sum_{i=1}^{n} u_i^2 - \frac{(\sum_{i=1}^{n} u_i)^2}{n}) \\ &= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (u_i - u_j)^2 \end{aligned}$$

Then $V \times C_a^2 \times L_m^2 = \frac{1}{n^4}\left(\sum_{i=1}^{n}\sum_{j>i}^{n}(u_i - u_j)^2 \times \left(\sum_{i=1}^{n} u_i\right)^2 \times L_m^2\right)$. Thus for $n = 2|E'| = |S|d$ the quantity in the $NDRP$ question above is equal to $\frac{V \times C_a^2 \times L_m^2}{n^4}$. Therefore, the minimization of this quantity is equivalent to minimization of the function $V \times C_a^2 \times L_m^2$ for a regular graph of $|S|$ sites and degree d.

We will show that $NDRP \in NP$_Complete. The method that we will use is, to prove that $NDRP \in NP$ and a subproblem of $NDRP$ is NP_Complete. We call this subproblem "The Two Source Ring Problem" denoted as $TSRP$. This subproblem is a special case of $NDRP$ for the family of ring networks where only two sites produce traffic. The methodology that we use to achieve the $TSRP$ NP_Completeness proof is illustrated in Figure 2 which can be used as a road map for the organization of this chapter.

To define $TSRP$ we impose a restriction on the set of frequencies F_e con-

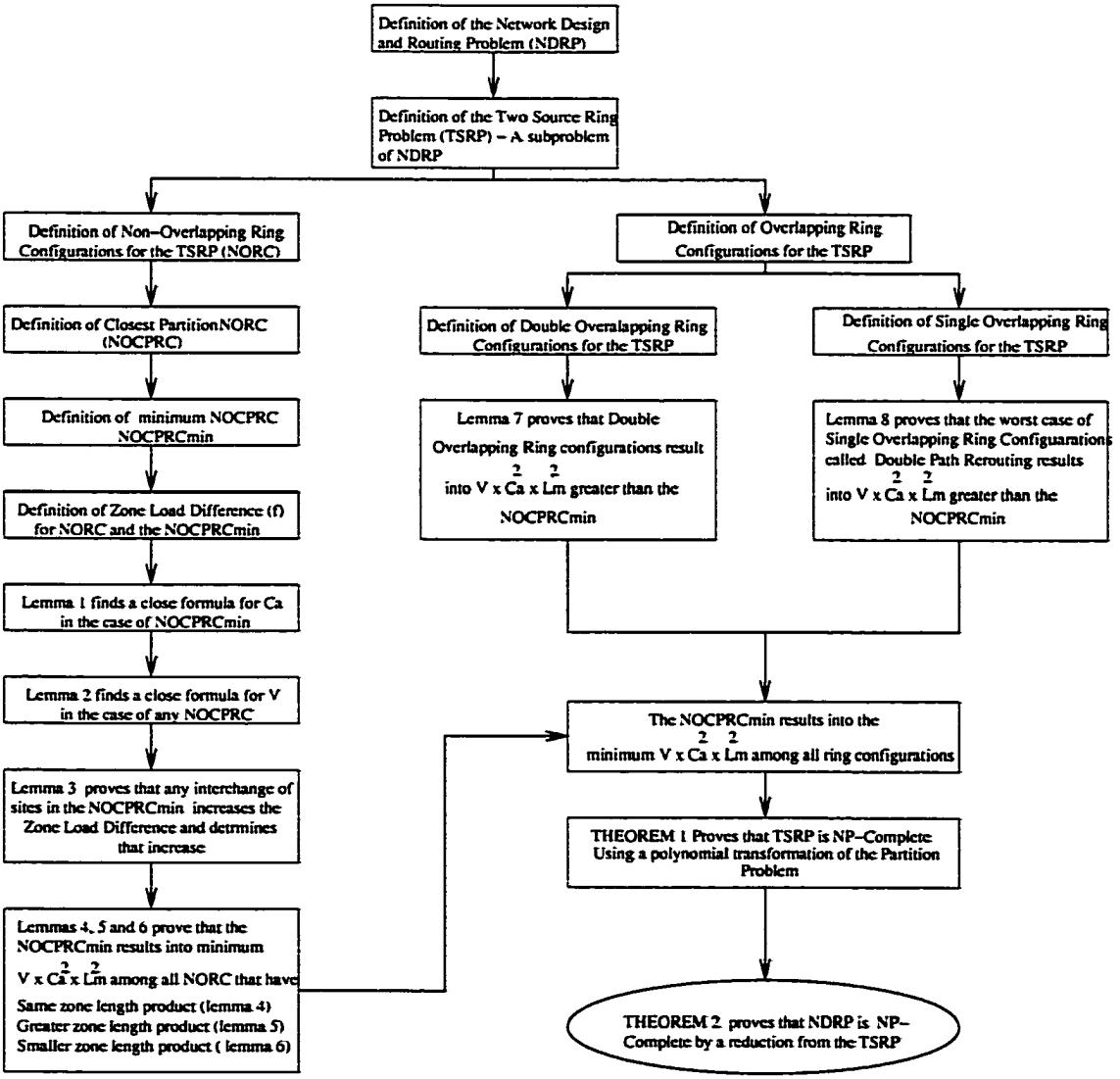

Figure 2. Organizational diagram of the second chapter.

sidering only frequencies which satisfy the *triangulation property*:

Definition 4 : The set F_e of frequencies of communication satisfies the triangulation property if for a network of n-sites:

$$f_{i,j} + f_{k,l} \geq f_{m,r} \quad \forall \ 1 \leq i, j, k, l, m, r \leq n$$

For example $F_e = \{20, 18, 17, 11, 9\}$ satisfies the triangulation property but the set $F_e = \{20, 17, 11, 10, 8\}$ does not since $10 + 8 < 20$.

Instance of $TSRP$: A complete graph $G = (S, E, F_e)$ where, S is the set of network sites, with two dedicated sites s_1 and s_2, called *sources*, E is the set of bidirectional links, $F_e = \{f_{s_i,s_j} |$ where $1 \leq i, j \leq |S|$ and $i \neq j\}$ is the set of frequencies of communication for every pair of sites (s_i, s_j) which satisfies the triangulation property. Also the following restrictions are imposed on F_e:

1. $f_{s_1,s_2} = f_{s_2,s_1} = 0$
2. $f_{s_1,s_j} = f_{s_2,s_j} = f_{s_j,s_1} = f_{s_j,s_2} \neq 0 \quad \forall \quad j \neq s_1, s_2$
3. $f_{s_i,s_j} = 0 \quad \forall \quad i, j \neq s_1, s_2$

the degree of the regular network is $d = 2$ and $B \in Z^+$.

Question : Is there a ring $G' = (S, E')$ of G and a routing scheme $R_{G'}$ of paths connecting the communication pairs of sites that results into a set U_e of frequencies of use such that:

$$\sum_{i=1}^{2|E'|} \sum_{j>i}^{2|E'|} (u_i - u_j)^2 \times \left(\sum_{i=1}^{2|E'|} u_i \right)^2 \times L_m^2 \leq B \left(|S| \right)^4 ?$$

TSRP is a subproblem of NDRP

Let D_{TSRP} be the domain of all instances of $TSRP$ and D_{NDRP} the domain of

all instances of $NDRP$. Let also Y_{TSRP} denote the set of all "yes" instances of $TSRP$ and Y_{NDRP} denote the set of all "yes" instances of $NDRP$. To show that $TSRP$ is a subproblem of $NDRP$ we need to show that $D_{TSRP} \subseteq D_{NDRP}$ and $Y_{TSRP} = Y_{NDRP} \cap D_{TSRP}$. Since $TSRP$ refers only to rings (a specific family of regular graphs) and the $NDRP$ domain includes all regular graph instances, then $D_{TSRP} \subseteq D_{NDRP}$. Also any "yes" answer to $TSRP$ belongs to D_{TSRP} and also is a "yes" answer to $NDRP$ for any arbitrary bound B and $d = 2$. Also any "no" instance of $TSRP$ does not belong to Y_{NDRP}. Therefore $Y_{TSRP} = Y_{NDRP} \cap D_{TSRP}$.

The polynomial transformation used in the proof is from an NP-Complete problem (a general case of the Partition problem), called the *Triangulation Property Closest Partition Problem* (TPCPP) which is defined as follows:

Instance of (TPCPP) : Finite set A and a size $s(a_i) \in Z^+$ for each $a_i \in A$, $s(a_i) + s(a_j) \geq s(a_k) \quad \forall \ 1 \leq i, j, k \leq |A|, \quad B \in Z^+$.

Question : Is there a subset $A' \subseteq A$ such that

$$\Big| \sum_{a_i \in A'} s(a_i) - \sum_{a_i \in A - A'} s(a_i) \Big| \leq B \ ?$$

Note that the $TPCPP$ is a variation of the partition problem with the condition that all elements in the set A must satisfy the triangulation property. However, this restriction does not make $TPCPP$ polynomial. To prove the NP-Completeness of $TPCPP$ we polynomially transform another subproblem of the partition problem to a subproblem of $TPCPP$. The two subproblems restrict into exact partition solutions with equal length partites and they are named *Ex-*

act Partition with Equal Partites, ($EPEP$) and *Triangulation Property Exact Partition with Equal Partites*, ($TPEPEP$) respectively. The definition of $EPEP$ and $TPEPEP$ follows:

Instance of (EPEP) : Finite set A and a size $s(a_i) \in Z^+$ for each $a_i \in A$,

Question : Is there a subset $A' \subseteq A$ with $|A'| = |A - A'|$ such that

$$\sum_{a_i \in A'} s(a_i) - \sum_{a_i \in A - A'} s(a_i) = 0?$$

Instance of (TPEPEP) : Finite set A and a size $s(a_i) \in Z^+$ for each $a_i \in A$, $s(a_i) + s(a_j) \geq s(a_k) \quad \forall \ 1 \leq i, j, k \leq |A|$.

Question : Is there a subset $A' \subseteq A$ with $|A'| = |A - A'|$ such that

$$\sum_{a_i \in A'} s(a_i) - \sum_{a_i \in A - A'} s(a_i) = 0?$$

Clearly $EPEP$ is a subproblem of the partition problem. This problem has be proven to be NP-Complete by Karp in [47]. Also $TPEPEP$ is a subproblem of $TPCPP$ with the restrictions that $B = 0$ and there is an equal partite solution. Theorem 1 proves that $TPEPEP$ is NP-Complete.

Theorem 1: $TPEPEP$ is NP-Complete.

Proof: To prove that $TPEPEP$ is an NP-Complete problem, we need to prove that $TPEPEP \in NP$ and also that $EPEP$ polynomially transforms to $TPEPEP$.

Let $|A| = n$. Since a non deterministic Turing machine need only guess a subset of A, namely A' and check in $O(n)$ time that

$$\sum_{a_i \in A'} s(a_i) - \sum_{a_i \in A - A'} s(a_i) = 0$$

it follows that, $TPEPEP \in NP$.

Let I_E be an instance of $EPEP$ and I_T an instance of $TPEPEP$. We will polynomially construct I_T from I_E and will show that if there is a solution for I_E, then there is a solution for I_T and vice versa.

Case $\Rightarrow$ Let $a_m = max\{a_i | a_i \in A\}$ in I_E. Construct a set T for the I_T instance as follows: $\forall \ a_i \in A, \ t_i = a_i + a_m$. Note that $|A| = |T| = n$ and the maximum element in T has value $2a_m$. Note also that, $\forall \ t_i, t_j, t_k, \ 1 \leq i, j, k \leq n$ the triangulation property is satisfied since, $t_i + t_j = a_i + a_j + 2a_m \geq 2a_m \geq t_k$. Furthermore, if there exists an equal partite exact partition on I_E then, the same holds for I_T since I_T results from I_E by increasing all elements in A by the same constant a_m.

Case $\Leftarrow$ Let T' and $T - T'$ with $|T'| = |T - T'|$ be a solution in I_T. Let also $t_m = max\{t_i | t_i \in T\}$. Since $\frac{t_m}{2}$ is always an integer, there is always a solution in I_E with $A' = \{a_i \mid a_i = t_i - \frac{t_m}{2}, \ ti \in T'\}$ and all elements of T result from A by adding the same constant a_m. □

Corollary 1 : $TPCPP \in NP$-Complete.

Proof : Since $TPEPEP$ is a subproblem of $TRCPP$ for the bound $B = 0$ and $TPEPEP \in NP$-Complete by theorem 1 then, it also follows that $TPCPP \in NP$-Complete. □

We will prove several lemmas for the case of $TSRP$ that will help us verify its NP-Completeness. Relatively to the permutation of the sites in the ring and

the routing scheme used for the $TSRP$, we distinguish the following three types of configurations:

Definition 5 : Let R_G be the routing scheme for the two source ring problem. Let also s_1 and s_2 be the two sources for which the frequencies of communication are positive. The rest of the sites create two zones using a clockwise scanning denoted as (s_1, s_2) and (s_2, s_1) respectively with

$$f_1 = |u_{s_{1_1}} - u_{s_{1_2}}| = |\sum_{i \in (s_1, s_2)} f_{s_1, s_i} - \sum_{i \in (s_2, s_1)} f_{s_1, s_i}|$$

and

$$f_2 = |u_{s_{2_1}} - u_{s_{2_2}}| = |\sum_{i \in (s_1, s_2)} f_{s_2, s_i} - \sum_{i \in (s_2, s_1)} f_{s_2, s_i}|$$

where $u_{s_{1_1}}, u_{s_{1_2}}$ are the loads of the two adjacent output buffers of source s_1 and $u_{s_{2_1}}, u_{s_{2_2}}$ are the loads of the two adjacent output buffers of source s_2. If $f_1 = f_2 = minimum$ over all permutations of sites we call such a ring configuration *Closest Partition Ring Configuration* ($CPRC$) and $f = f_1 = f_2$ the *Zone Load Difference* for the $TSRP$.

An example of such a ring configuration is depicted in Figure 3.

Note that in Figure 3, the depicted loads are not the edge loads but the loads of the two corresponding buffers attached to each link. For example, $u_{s_1,7} = u_{7,s_1} = 26$ means that the load of the buffer $(s_1, 7)$ and the load of the buffer $(7, s_1)$ are equal to 26.

Definition 6 : A $CPRC$ is called *Non-Overlapping* ($NOCPRC$) when the route for each path p_{s_1,s_i} with frequency of communication f_{s_1,s_i} and each path p_{s_i,s_1}

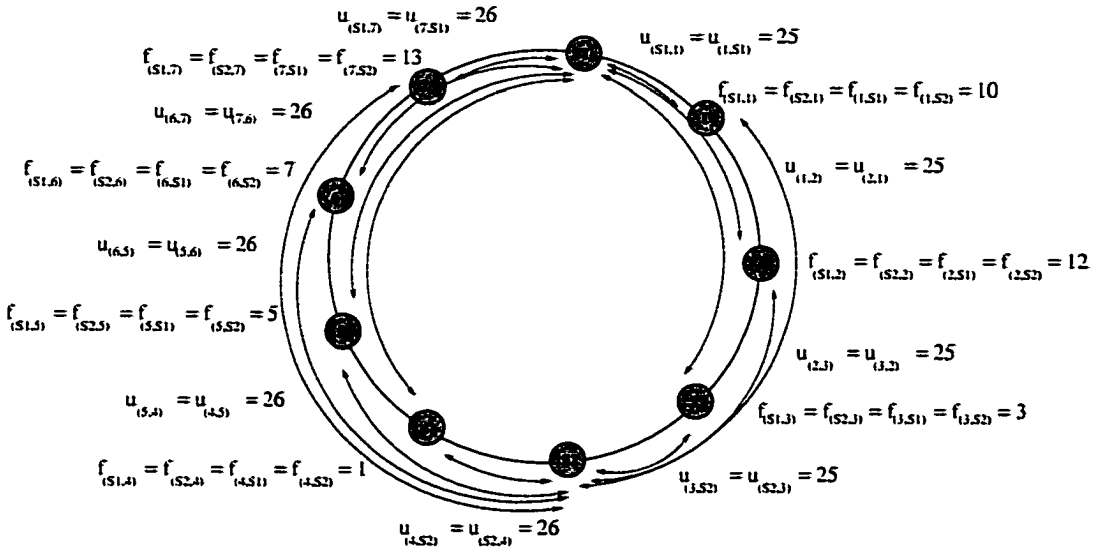

Figure 3. Closest partition ring configuration.

with frequency of communication f_{s_i,s_1} does not pass from the source s_2 and the route for each path p_{s_2,s_i} with frequency of communication f_{s_2,s_i} and each path p_{s_i,s_2} with frequency of communication f_{s_i,s_2} does not pass from the source s_1. Such a routing scheme for the $TSRP$ is illustrated in Figure 3.

Definition 7 : Any $TSRP$ configuration for which there is at least a path emanating from one source which passes through the other source is called *Overlapping* and it is denoted as $OCPRC$.

An example of an $OCPRC$ is depicted in Figure 4 where the path $p_{s_1,4}$ and the path $p_{s_2,2}$ are overlapping.

Lemma 1 : Let R_i be a $NOCPRC$ for the $TSRP$ of n-sites. Let l be the length

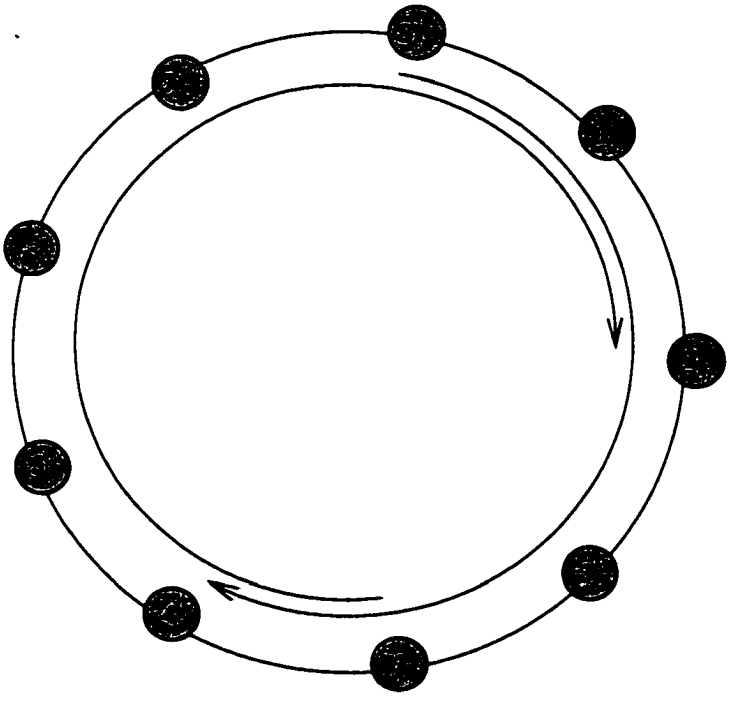

Figure 4. Overlapping closest partition ring configuration.

of zone 1 with $(n-l)$ the length of the zone 2. Let w_1 be the load of all buffers in zone 1 and w_2 the load in zone 2 respectively with $w = min\{w_1, w_2\}$. Then the average communication cost is given as :

$$\begin{aligned} C_a &= \frac{nw + (n-l)f}{n} \quad \text{if} \quad w_2 > w_1, \\ C_a &= \frac{nw + lf}{n} \quad \text{if} \quad w_1 > w_2 \quad \text{and} \\ C_a &= w \quad \text{if} \quad w_1 = w_2 \end{aligned}$$

Proof : For a ring of n-sites $C_a = \frac{T}{2n}$, where $T = 2lw_1 + 2(n-l)w_2$ is the total communication cost.

For the first case, $f = w_2 - w_1$ and $w = w_1$. Then

$$C_a = \frac{2lw_1 + 2(n-l)w_2}{2n} = \frac{lw_1 + (n-l)(w_1 + f)}{n} = \frac{nw + (n-l)f}{n}$$

For the second case, $f = w_1 - w_2$ and $w = w_2$. Then

$$C_a = \frac{2lw_1 + 2(n-l)w_2}{2n} = \frac{l(w_2 + f) + (n-l)w_2}{n} = \frac{nw + lf}{n}$$

Finally for the third case, $f = 0$ and $w = w_1 = w_2$ resulting in $C_a = w$. □

Non-overlapping closest partition ring configurations

In this section we prove several lemmas relative to $NOCPRC$ for the $TSRP$. The first lemma finds a closed formula for the variance of the edge loads in the case of $NOCPRC$. We show that there are ring configurations with the same zone load difference but different variance. We then define the minimum $NOCPRC$ which is a configuration with minimum load zone difference, minimum variance and minimum average communication cost. Furthermore, the closed formula introduces a classification of the non-overlapping ring configurations relative to the relation of the lengths between the two zones into three different types. The second lemma proves that when we disturb the $NOCPRC$ by swapping sites between the zones, we either obtain a configuration that results in greater C_a and/or we increase the absolute difference of buffer loads of the two zones by at least $2f$. Using these two lemmas, we then prove three lemmas relative to each one of the types of ring configurations found by the general variance formula. We show that

each class of ring configurations always produces $V \times C_a^2 \times L_m^2$ greater than the corresponding product of the minimum $NOCPRC$.

Lemma 2 : The variance V for the case of a $NOCPRC$ with n sites is given by the formula:

$$V = \frac{l(n-l)f^2}{n^2}$$

Proof : Let l denote the number of links in zone 1 with $(n-l)$ the number of links of zone 2. This means that there are $2l$ buffers in zone 1 and $2(n-l)$ buffers in zone 2. Note that any non-overlapping configuration results in buffer loads such that all buffers in zone 1 have the same load and also all buffers in zone 2 have the same load. Let w_1 be the buffer loads of zone 1 and w_2 the buffer loads of zone 2 with zone load difference $f = |w_1 - w_2|$. The variance for that configuration is:

$$\begin{aligned}
V &= \frac{1}{4n^2}\sum_{i=1}^{2n}\sum_{j=i+1}^{2n}(u_i - u_j)^2 \\
&= \frac{1}{4n^2}\left(\sum_{i=1}^{2l}\sum_{j=i+1}^{2l}(w_1 - w_1)^2 + \sum_{i=1}^{2l}\sum_{j=2l+1}^{2n}(w_1 - w_2)^2 + \sum_{i=2l+1}^{2n}\sum_{j=i+1}^{2n}(w_2 - w_2)^2\right) \\
&= \frac{1}{4n^2}\left(\sum_{i=1}^{2l}\sum_{j=2l+1}^{2n}(w_1 - w_2)^2\right) \\
&= \frac{1}{4n^2}4l(n-l)(w_1 - w_2)^2 \\
&= \frac{l(n-l)f^2}{n^2} \quad \square
\end{aligned}$$

Definition 8 : Let R be the set of ring $NOCPRC$ configurations for a set of frequencies of communication in the $TSRP$. Let $R' \subseteq R$ the set of configurations that minimize the variance of edge loads, i.e., the ones that minimize the product $l(n-l)$. We call the set R' the *Minimum Variance NOCPRC* configurations for

$TSRP$ denoting it as $NOCPRC_{minV}$.

Note that in definition 8, R' contains all $NOCPRC$ that maximize the absolute difference between the lengths of the two zones. This is always true since the length of the smaller zone is bounded by $\lfloor \frac{n}{2} \rfloor$. An example of a $NOCPRC_{minV}$ is shown in Figure 5 where the right configuration minimizes the variance.

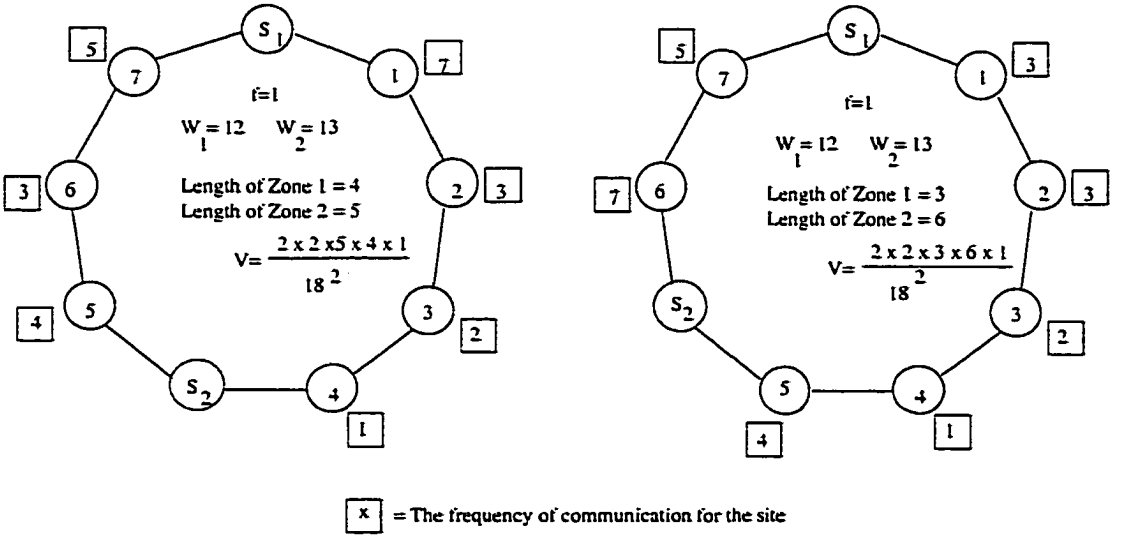

Figure 5. A minimum variance $NOCPRC$ in $TSRP$.

Definition 9 : Let R' be the set of $NOCPRC_{minV}$ in $TSRP$. Let $R'' \subseteq R'$ be the subset of configurations that minimize the average communication cost for the ring. We call R'' the set of *Minimum variance and average NOCPRC*, denoting it as $NOCPRC_{min\ V\ C_a}$.

Note that in definition 9, $NOCPRC_{min\ V\ C_a}$ occurs when the zone of smaller length has the greater edge loads. Figure 6 shows two configurations

for a set of frequencies of communication that both have minimum variance but not the same average communication cost.

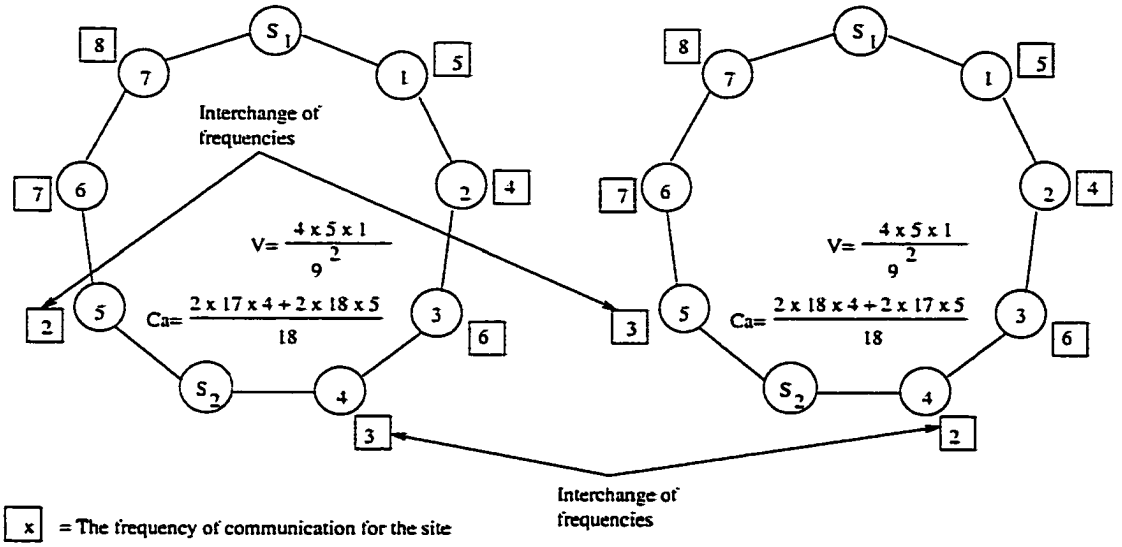

Figure 6. A minimum variance and average cost $NOCPRC$ in $TSRP$.

Definition 10 : A $NOCPRC$ R_i is called *Minimum* $NOCPRC$ if $R \in NOCPRC_{min\ V\ C_a}$.

Lemma 3 : Let R_i be a minimum $NOCPRC$ for $TSRP$ with zone lengths l, $(l \leq \lfloor \frac{n}{2} \rfloor)$ and $(n-l)$ and zone buffer loads w_1 and w_2, respectively. Also let s_a be a site in zone 1 and s_b a site in zone 2 with corresponding frequencies of communication $f_{s_1,s_a} = f_{s_a,s_1} = f_a$ and $f_{s_2,s_b} = f_{s_b,s_2} = f_b$, respectively. Then a swap of sites s_a, s_b creates a configuration R_i' with edge loads (w_1', w_2') such that:

1. If $w_1 > w_2$ and $(f_a - f_b) = f$, then $C_a' \geq C_a$.
2. If $|f_a - f_b| = a > f$, then $|w_1' - w_2'| \geq f + 2a \geq 3f$.

Proof : If $w_1 > w_2$ and $(f_a - f_b) = f$, then the interchange of sites creates a configuration of the same variance as the minimum $NOCPRC$ and $w'_2 > w'_1$. Since $l \leq (n-l)$, it follows that $C'_a \geq C_a$.

Note that if $w_1 \leq w_2$ and $(f_b - f_a) = f$, then the interchange of sites gives a smaller average communication cost. Thus this case should be the minimum $NOCPRC$ assumed by the lemma. That proves the first conjecture.

If $|f_a - f_b| = a > f$, then we investigate two cases relatively to the relation of w_1 and w_2.

1. If $w_1 > w_2$, then $w'_1 = w_2 + f - f_a + f_b$ and $w'_2 = w_2 + f_a - f_b$.

•If $w'_1 > w'_2$, then $w'_1 - w'_2 = f + 2(f_b - f_a)$. Note that $(f_b - f_a) > f$ since, if this was not the case, then R_i would not be minimum. Thus $w'_1 - w'_2 \geq f + 2a \geq 3f$.

•If $w'_1 < w'_2$, then $w'_2 - w'_1 = 2(f_a - f_b) - f$. However in this case $f_a - f_b > 2f$ since f_a must be greater or equal to $f_b + 2f$ in order to cover the difference between w_1 and w_2 and make the second load greater by at least f. Thus $w'_2 - w'_1 \geq -f + 2(2f) \geq f + 2a \geq 3f$.

2. If $(w_1 \leq w_2)$, then $w'_1 = w_1 + f_b - f_a$ and $w'_2 = w_1 + f + f_a - f_b$.

•If $w'_1 > w'_2$, then $w'_1 - w'_2 = 2(f_b - f_a) - f$. However in this case $f_b - f_a > 2f$ since f_b must be greater or equal to $f_a + 2f$ in order to cover the difference between w_2 and w_1 and make the second load greater by at least f. Thus $w'_1 - w'_2 \geq -f + 2(2f) \geq f + 2a \geq 3f$.

•If $w_1' < w_2'$, then $w_2' - w_1' = 2(f_a - f_b) + f$. Note that $(f_a - f_b) > f$ since, if this was not the case, then R_i would not be minimum. Thus $w_1' - w_2' \geq f + 2a \geq 3f$.

Thus for all the cases above $|w_1' - w_2'| \geq f + 2a \geq 3f$, which proves the second conjecture. □

In the following lemmas we will prove that any other non-overlapping configuration that does not satisfy the closest partition property results into a greater $V \times C_a^2 \times L_m^2$. Lemma 4 refers to any non-closed partition configuration which has the same lengths l and $(n - l)$ as the closest partition configuration. The same is proved in Lemmas 5 and 6 but for the case of any non-closed partition configuration with different zone lengths.

Lemma 4 : Any non-overlapping non-closed partition ring configuration with the same zone lengths as the $NOCPRC$ results in $V' \times C_a'^2 \times L_m'^2 \geq V \times C_a^2 \times L_m^2$ of the minimum $NOCPRC$.

Proof: Let R_i be a minimum $NOCPRC$ for $TSRP$ with zone lengths l, $(l \leq \lfloor \frac{n}{2} \rfloor)$ and $(n - l)$ and zone buffer loads w_1 and w_2, respectively.

Then, by Lemmas 1 and 2, R_i results in

$$V \times C_a^2 \times L_m^2 = \frac{l(n-l)f^2}{n^2} \times \left(\frac{2lw_1 + 2(n-l)w_2}{2n} \right)^2 \times (max\{w_1, w_2\})^2,$$

where $f = |w_1 - w_2|$ is the zone buffer difference which is minimum by assumption. Also let R_i' be a non-closed partition configuration with non-overlapping routing with the same lengths as the minimum $NOCPRC$ above. Let w_1' and w_2' be the

buffer loads of the two zones for R'_i. Note that R'_i can be obtained from R_i by interchanging sites between the two zones. Thus,

$$V' \times C_a'^2 \times L_m'^2 = \frac{l(n-l)|w'_1 - w'_2|^2}{n^2} \times \frac{2lw'_1 + 2(n-l)w'_2}{2n} \times (max\{w'_1, w'_2\})^2$$

If $|w'_1 - w'_2| = f$, then by Lemma 2, $V = V'$ and $L_m = L'_m$. However from Lemma 3, in that case the average communication cost of R_i is less than or equal to the average communication cost of R'_i. Thus $V' \times C_a'^2 \times L_m'^2 \geq V \times C_a^2 \times L_m^2$.

The interesting case here is when $|w'_1 - w'_2| = f + 2a > |w_1 - w_2|$ where $2a$ is the additional difference between the buffer loads of the two zones by Lemma 3 with $a \geq f$. We show that $V' \times C_a'^2 \times L_m'^2 \geq V \times C_a^2 \times L_m^2$ for any possible relation between w_1, w_2, w'_1 and w'_2. Consider the cases:

1. $w_1 < w_2$ and $w'_1 < w'_2$
2. $w_1 < w_2$ and $w'_1 > w'_2$
3. $w_1 > w_2$ and $w'_1 > w'_2$
4. $w_1 > w_2$ and $w'_1 < w'_2$

Case 1 : Since $w_1 < w_2$, $C_a = \frac{nw_1 + (n-l)f}{n}$ by Lemma 1. Also since $w'_1 < w'_2$, we have by Lemma 1 again that

$$\begin{aligned} C'_a &= \frac{lw'_1 + (n-l)w'_2}{n} = \frac{l(w_1 - a) + (n-l)(w_1 + f + a)}{n} \\ &= \frac{nw_1 + (n-l)f + (n-2l)a}{n} \end{aligned}$$

Thus $C'_a - C_a = \frac{(n-2l)a}{n}$. Note that $V' > V$ by Lemma 2 because $(f+2a)^2 > f^2$ and $C'_a - C_a = \frac{(n-2l)a}{n}$. But $(n-2l) \geq 0$ since $l \leq \lfloor \frac{n}{2} \rfloor$. Also $L'_m = w_2 + a > w_2 = L_m$.

Thus $V' \times C_a'^2 \times L_m'^2 \geq V \times C_a^2 \times L_m^2$.

Case 2 : Since $w_1 < w_2$ and $w_1' > w_2'$ by Lemma 1, it follows that $C_a = \frac{nw_1+(n-l)f}{n}$ and

$$C_a' = \frac{lw_1' + (n-l)w_2'}{n} = \frac{l(w_1+f+a)+(n-l)(w_1-a)}{n} = \frac{nw_1+(2l-n)a+lf}{n}$$

The difference between C_a and C_a' is :

$$C_a - C_a' = \frac{(n-2l)(f+a)}{n} < (f+a)$$

and C_a' can be written relative to C_a as:

$$C_a' = C_a - \frac{(n-2l)(f+a)}{n}$$

Note that $a \geq f$ by Lemma 3. Then we can show by the following sequence of inequalities that our claim holds:

$$\begin{aligned}
V' \times C_a'^2 \times L_m'^2 &\geq V \times C_a^2 \times L_m^2 \\
\frac{l(n-l)(f+2a)^2}{n^2} \times C_a'^2 \times L_m'^2 &\geq \frac{l(n-l)f^2}{n^2} \times C_a^2 \times L_m^2 \\
9f^2 \times C_a'^2 \times L_m'^2 &\geq f^2 \times C_a^2 \times L_m^2 \\
3C_a' \times L_m' &\geq C_a \times L_m
\end{aligned}$$

Since $L_m' > L_m$, we need only show that $3C_a' \geq C_a$. Note that $\frac{(n-2l)(f+a)}{n} < (f+a)$ since $\frac{n-2l}{n} < 1$. Then :

$$3C_a - 3\left(\frac{(n-2l)(f+a)}{n}\right) > C_a \;\Rightarrow\; 2C_a > 3\left(\frac{(n-2l)(f+a)}{n}\right)$$

which is always true since $C_a > w_1 > 2(f+a)$ and $\frac{n-2l}{n} < \frac{1}{2}$ because of the triangulation property imposed.

Case 3 : Since $w_1 > w_2$, it follows that $C_a = \frac{nw_2+lf}{n}$ by Lemma 1. Also since $w'_1 > w'_2$, we have by Lemma 1 again that

$$\begin{aligned} C'_a &= \frac{lw'_1 + (n-l)w'_2}{n} \\ &= \frac{l(w_2+f+a) + (n-l)(w_2-a)}{n} \\ &= \frac{nw_2 + lf + (n-2l)a}{n} \end{aligned}$$

Thus $C'_a - C_a = \frac{(n-2l)a}{n} > 0$. Note also that $V' > V$ by Lemma 2. Moreover $L'_m > L_m$. Thus the lemma inequality is true.

Case 4: Since $w_1 > w_2$, $C_a = \frac{nw_2+lf}{n}$ by Lemma 1. Also since $w'_1 < w'_2$, we have by Lemma 1 again that

$$\begin{aligned} C'_a &= \frac{lw'_1 + (n-l)w'_2}{n} \\ &= \frac{l(w_2-a) + (n-l)(w_2+f+a)}{n} \\ &= \frac{nw_2 + (n-2l)a + (n-l)f}{n} \end{aligned}$$

The difference between the two averages is:

$$C'_a - C_a = \frac{(n-2l)(f+a)}{n} > 0$$

Moreover $V' > V$ by Lemma 2 and $L'_m > L_m$. Thus the lemma inequality is true.

□

Lemma 5 : Any non-overlapping non-closed partition ring configuration with zone lengths $(l+x) \leq \lfloor \frac{n}{2} \rfloor$ and $(n-l-x)$, $x \in Z^+$, results in $V' \times {C'_a}^2 \times {L'_m}^2 \geq V \times C_a^2 \times L_m^2$, where $V \times C_a^2 \times L_m^2$ is for the minimum $NOCPRC$ with zone lengths

l and $(n-l)$ respectively.

Proof : For the minimum $NOCPRC$ R_i let l and $(n-l)$ be the lengths of zone 1 and 2 respectively with $l \leq \lfloor \frac{n}{2} \rfloor$ and w_1, w_2 be the buffer loads. Let R_i' be a non-closed partition non-overlapping ring configuration. Let $(l+x)$ and $(n-l-x)$ be the lengths of the two zones for R_i' with $(l+x) \leq \lfloor \frac{n}{2} \rfloor$. Let also w_1', w_2' be the corresponding buffer loads for R_i'.

If $|w_1' - w_2'| = f$, then $V' > V$ since $(l+x)(n-l-x) > l(n-l)$ because $(l+x) \leq \lfloor \frac{n}{2} \rfloor$. Also $C_a' > C_a$ otherwise, R_i' would be the minimum. Thus in that case the lemma inequality holds.

The interesting case here is when $|w_2' - w_1'| \geq (f+2a)$ from Lemma 3 with $(a \geq f)$. Then to prove the lemma inequality we have to prove that :

$$\frac{(l+x)(n-l-x)(f+2a)^2}{n^2} \times {C_a'}^2 \times {L_m'}^2 > \frac{l(n-l)f^2}{n^2} \times {C_a}^2 \times L_m^2$$

Note that $(l+x)(n-l-x) > l(n-l)$ and by Lemma 3 that $(f+2a) \geq 3f$. Moreover $L_m' = L_m + a \geq L_m$. Therefore to prove the above inequality it suffices to prove that $9{C_a'}^2 > C_a^2$ or $3C_a' > C_a$. We will show that the last inequality holds for any relation between w_1, w_2 and w_1', w_2'. We distinguish the following cases:

1. $w_1 < w_2$ and $w_1' < w_2'$. Then by Lemma 1, $C_a = \frac{nw_1+(n-l)f}{n}$ and $C_a' = \frac{(l+x)(w_1-a)+(n-l-x)(w_1+f+a)}{n}$. Thus, $C_a' - C_a = \frac{xf+(n-2l-2x)a}{n} > 0$ since $(l+x) \leq \lfloor \frac{n}{2} \rfloor$. Therefore $C_a' > C_a$ and the inequality holds.

2. $w_1 < w_2$ and $w_1' > w_2'$. Then by Lemma 1, $C_a = \frac{nw_1+(n-l)f}{n}$ and $C_a' = \frac{(l+x)(w_1+f+a)+(n-l-x)(w_1-a)}{n}$. In that case $C_a - C_a' = \frac{(n-2l-x)f+(n-2l-2x)a}{n} < f+a$.

However, note that $3(C_a - (f+a)) > C_a$ since $2C_a > 3(f+a)$ because $C_a > w_1$. Therefore the inequality holds.

3. $w_1 > w_2$ and $w_1' > w_2'$. Then by Lemma 1, $C_a = \frac{nw_2+lf}{n}$ and $C_a' = \frac{(l+x)(w_2+f+a)+(n-l-x)(w_2-a)}{n}$. In that case $C_a' - C_a = \frac{xf-(n-2l-2x)a}{n} < -a$. However, $3(C_a - a) > C_a$ since $2C_a > 3a$ because $C_a > w_1$ and the inequality holds.

4. $w_1 > w_2$ and $w_1' < w_2'$. Then by Lemma 1, $C_a = \frac{nw_2+lf}{n}$ and $C_a' = \frac{(l+x)(w-a)+(n-l-x)(w_2+f+a)}{n}$. Thus $C_a' = C_a + \frac{(n-2l-x)f+(n-2l-2x)a}{n}$ and the inequality holds since $C_a' > C_a$. □

Lemma 6 : Any non-overlapping non-closed partition ring configuration with zone lengths $l-x$ and $n-l+x$, respectively, results in $V' \times {C_a'}^2 \times {L_m'}^2 \geq V \times C_a^2 \times L_m^2$ where $V \times C_a^2 \times L_m^2$ is for the minimum $NOCPRC$ with zone lengths l and $n-l$, respectively.

Proof : For the minimum $NOCPRC$ R_i let l and $n-l$ be the lengths of zone 1 and 2, respectively, with $l \leq \lfloor \frac{n}{2} \rfloor$, and let w_1, w_2 be the buffer loads. Furthermore, let R_i' be a non-close partition non-overlapping ring configuration. Let $(l-x)$ and $(n-l+x)$ be the lengths of the two zones for R_i' with $(l-x) \leq \lfloor \frac{n}{2} \rfloor$. Let also w_1', w_2' be the corresponding buffer loads for R_i'.

If $|w_1' - w_2'| = f$, then R_i' results in a greater product $(l-x)(n-l+x)$. That means that $V' > V$ which is not possible otherwise, R_i' should be the minimum configuration.

If this is not the case and $|w_2' - w_1'| = f + 2a$ $a \geq f$ from Lemma 3, then

we have to show that:

$$\frac{(l-x)(n-l+x)(f+2a)^2}{n^2} \times C_a'^2 \times L_m'^2 \geq \frac{l(n-l)f^2}{n^2} \times C_a^2 \times L_m^2$$

Note that $(f+2a)^2 \geq 9f^2$ and $L_m' = L_m + a > L_m$. Then it suffices to show that $9(l-x)(n-l+x)C_a'^2 \geq l(n-l)C_a^2$.

We will show that

$$8(l-x)(n-l+x)C_a'^2 \geq l(n-l)C_a^2 \;\Rightarrow\; 2(l-x)(n-l+x)4C_a'^2 \geq l(n-l)C_a^2$$

which is a stronger inequality. Moreover we will split this inequality into two parts:

1. $2(l-x)(n-l+x) > l(n-l)$ and
2. $4C_a'^2 > C_a^2$

and prove those parts separately.

First Inequality : Note that :

$$\begin{aligned} 2(l-x)(n-l+x) &> l(n-l) \\ 2l(n-l) - 2x(n-2l+x) &> l(n-l) \\ l(n-l) &> 2x(n-2l+x) \end{aligned}$$

Note also that $n-l > n-2l+x$. It also must be that $2x < l$ or $x < \frac{l}{2}$, for if if this were not the case, then $n-2l+x > 2(l-x)$, i.e., zone 2 contains at least double the sites that zone 1 contains. However, because the triangulation property holds for the set F_e, the load zone difference for R_i' is $f+2a > lf$. Therefore,

$$V' = \frac{(l-x)(n-l+x)l^2f^2}{n^2} > l(l-x)V$$

and $C'_a > C_a$. Thus the inequality $2(l-x)(n-l+x) > l(n-l)$ holds.

Second Inequality : To prove the second inequality, we examine any possible relation between w_1, w_2 and w'_1, w'_2.

1. $w_1 < w_2$ and $w'_1 < w'_2$. By Lemma 1, $C_a = \frac{nw_1+(n-l)f}{n}$ and also $C'_a = \frac{(l-x)(w_1-a)+(n-l+x)(w_1+f+a)}{n}$. Thus $C'_a - C_a = \frac{xf+(n-2l+2x)a}{n} > 0$ and the conjecture $4C'^2_a > C^2_a$ is true.

2. $w_1 < w_2$ and $w'_1 > w'_2$. By Lemma 1, $C_a = \frac{nw_1+(n-l)f}{n}$ and also $C'_a = \frac{(l-x)(w_1+f+a)+(n-l+x)(w_1-a)}{n}$. Thus $C_a - C'_a = \frac{(n-2l+x)f+(n-2l+2x)a}{n} < f+a$. Then $4(C_a-(f+a))^2 > C^2_a$ since $C_a > min(w_1, w_2) > f+a$.

3. $w_1 > w_2$ and $w'_1 > w'_2$. By Lemma 1, $C_a = \frac{nw_2+lf}{n}$ and also $C'_a = \frac{(l-x)(w_2+f+a)+(n-l+x)(w_2-a)}{n}$. In that case, $C_a - C'_a = \frac{xf+(n-2l+2x)a}{n} < f+a$. Then $4(C_a-(f+a))^2 > C^2_a$ since $C_a > min(w_1, w_2) > f+a$.

4. $w_1 > w_2$ and $w'_1 < w'_2$. By Lemma 1, $C_a = \frac{nw_2+lf}{n}$ and also $C'_a = \frac{(l-x)(w_2-a)+(n-l+x)(w_2+f+a)}{n}$. In that case, $C'_a - C_a = \frac{(n-2l+x)f+(n-2l+2x)a}{n} > 0$ and $4C'^2_a > C^2_a$ is true. □

Overlapping ring configurations

In the case of overlapping ring configurations we distinguish two types of overlapping which are defined as follows:

Definition 11: Let R_i be an overlapping routing ring configuration for the $TSRP$ and let s_i be a site other than the two sources s_1 and s_2. If the paths p_{s_1,s_i} and

p_{s_i,s_1} pass from the source s_2 and the paths p_{s_2,s_i} and p_{s_i,s_2} pass from from the source s_1, then s_i is called a *Double Overlapping Site*. If at least one of the four paths above does not have s_1 or s_2 as an intermediate site, then s_i is called a *Single Overlapping Site*.

Definition 12: Ring routings that include single or double overlapping sites are called single or double overlapping routings, respectively.

A routing scheme for the $TSRP$ can produce a mix of non-overlapping, single-overlapping and double-overlapping sites. In the previous section we exhausted the case of non-overlapping routings and proved that such routings always produce $V \times C_a^2 \times L_m^2$ greater than in the case of the minimum $NOCPRC$. In this section we will show the same for the case of overlapping routings when we retain the site permutation of the minimum $NOCPRC$. We will prove two lemmas distinguishing single and double overlapping routings.

Lemma 7: Any double overlapping routing configuration for the $TSRP$ produces $V' \times {C'_a}^2 \times {L'_m}^2$ greater than the $V \times C_a^2 \times L_m^2$ of the minimum $NOCPRC$.

Proof: Let R_i be a minimum $NOCPRC$. Also let l , $l \leq \lfloor \frac{n}{2} \rfloor$ and $n-l$ the lengths of zone 1 and 2, respectively. Furthermore, let w_1 and w_2 be the buffer loads of the two zones. Moreover, let R'_i be a double overlapping routing configuration that maintains the zone lengths of R_i. Consider s_i to be a double overlapping site in R'_i and $f_{s_i,s_1} = f_{s_i,s_2} = f_{s_1,s_i} = f_{s_2,s_1} = f_i$ to be its corresponding communication frequencies. We investigate two cases depending on the zone which s_i resides.

Case : $s_i \in$ zone of length l

Such a configuration is shown in Figure 7.

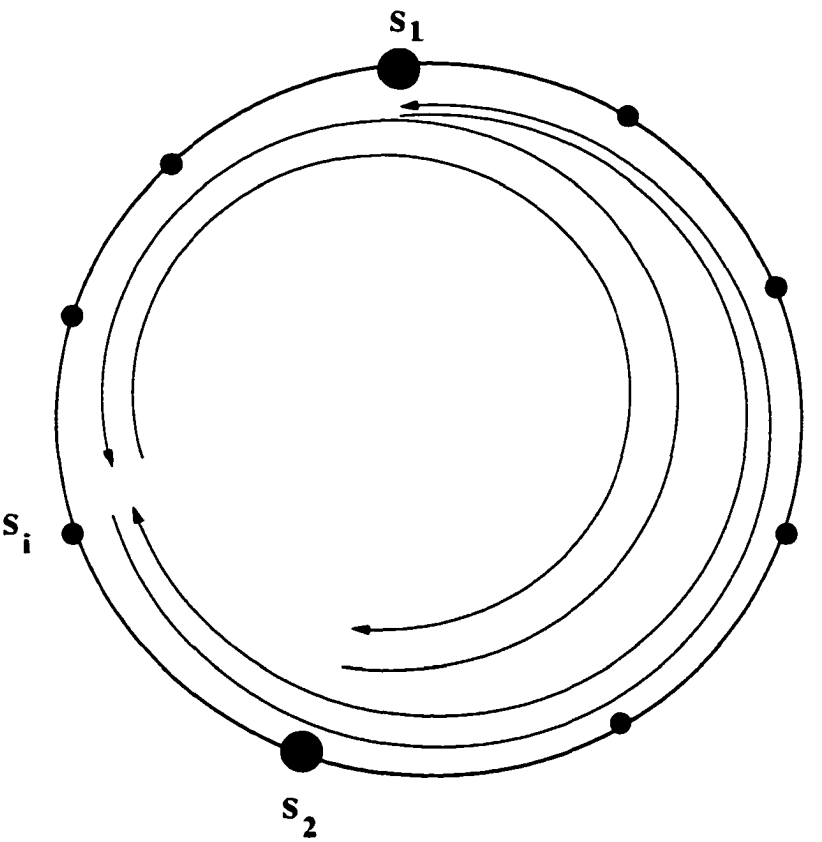

Figure 7. Double overlapping routing for s_i being in zone with length l.

We show that $L'_m > L_m$, $C'_a > C_a$ and $V' > V$ for the cases that $w_1 \geq w_2$ and $w_1 < w_2$. As Figure 7 illustrates, all buffers in the $n - l$ partite are charged with an additional load equal to $2f_i$ relative to R_i because of the change in routing of the site s_i.

1. $w_1 \geq w_2$. Then $f_i > f$ where f is the minimum buffer load difference

of R_i because if this was not the case then s_i would belong to the other partite. For the configuration R'_i, note also that $w'_1 = w_1$ and $w'_2 = w_2 + 2f_i$. Since $w_1 - w_2 = f$, it follows that $L_m = w_1$. Moreover, because $w'_2 = w_2 + 2f_i > w_1$, then $L'_m = w'_2 > w_1 > L_m$.

By Lemma 1, $C_a = \frac{lw_1+(n-l)w_2}{n}$ and $C'_a = \frac{lw'_1+(n-l)w'_2}{n}$. Since $w'_1 = w_1$ and $w'_2 > w_2$, we have that $C'_a > C_a$.

Furthermore, by Lemma 2, $V = \frac{l(n-l)f^2}{n^2}$ and $V' = \frac{l(n-l)(w'_2-w'_1)^2}{n^2} = \frac{l(n-l)(w_2+2f_i-w_2-f)^2}{n^2}$ and since $f_i > f$, it follows that $V' > V$.

2. $w_1 < w_2$. Since s_i belongs in the l partite, $w'_1 = w_1$ and $w'_2 = w_2 + 2f_i$. Then $L_m = w_2$ and $L'_m = w_2 + 2f_i > L_m$.

By Lemma 1, $C_a = \frac{lw_1+(n-l)w_2}{n}$ and $C'_a = \frac{lw'_1+(n-l)w'_2}{n}$. Since $w'_2 > w_2$, it follows that $C'_a > C_a$.

Also by Lemma 2, $V = \frac{l(n-l)f^2}{n^2}$ and $V' = \frac{l(n-l)(w'_2-w'_1)^2}{n^2} = \frac{l(n-l)(w_2+2f_i-w_1+f)^2}{n^2}$. Thus $V' > V$.

Case : $s_i \in$ zone of length $n - l$

Such a configuration is shown in Figure 8.

We show that $L'_m > L_m$, $C'_a > C_a$ and $V' > V$ for the cases that $w_1 \geq w_2$ and $w_1 < w_2$. As Figure 8 illustrates all buffers in the l partite are charged with an additional load equal to $2f_i$ relative to R_i. This is because of the change in routing for the site s_i.

1. $w_1 \geq w_2$. Then $f_i > f$ where f is the minimum buffer load difference of

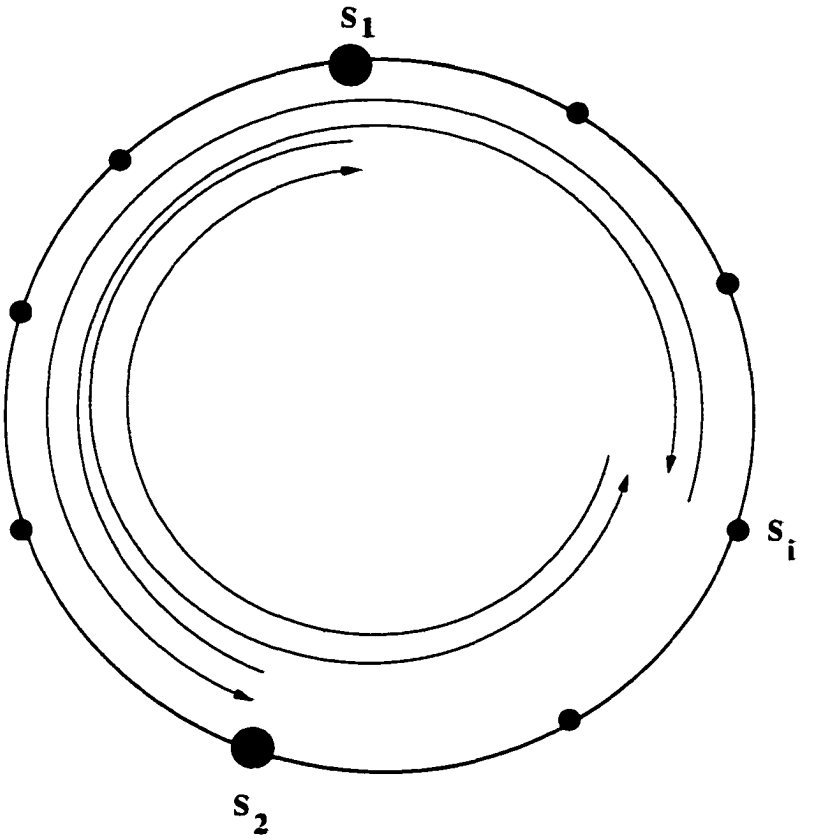

Figure 8. Double overlapping routing for s_i being in zone with length $(n-l)$.

R_i, for if this was not the case, then s_i would belong to the other partite. For the configuration R'_i, note also that $w'_2 = w_2$ and $w'_1 = w_1 + 2f_i$. Since $w_1 - w_2 = f$, it follows that $L_m = w_1$. Also since $w'_1 = w_1 + 2f_i > w_1$, $L'_m = w'_1 > w_1$. Thus $L'_m > L_m$.

By Lemma 1, $C_a = \frac{lw_1+(n-l)w_2}{n}$ and also $C'_a = \frac{lw'_1+(n-l)w'_2}{n}$. Since $w'_1 > w_1$ and $w'_2 = w_2$, it follows that $C'_a > C_a$.

Also by Lemma 2, $V = \frac{l(n-l)f^2}{n^2}$ and $V' = \frac{l(n-l)(w'_1-w'_2)^2}{n^2} = \frac{l(n-l)(w_2+2f_i-w_2+f)^2}{n^2}$ and

since $f_i > f$, it follows that $V' > V$.

2. $w_1 < w_2$. Since s_i belongs in the $(n-l)$ partite, $w_1' = w_1 + 2f_i > w_1$ and $w_2' = w_2$. Also $f_i > f$ because if this was not the case, then s_i would belong to the other partite. However note that $L_m = w_2$. Furthermore, $w_1' > w_2$ since $f_i > f$. Thus $L_m' > L_m$.

By Lemma 1, $C_a = \frac{lw_1+(n-l)w_2}{n}$ and also $C_a' = \frac{lw_1'+(n-l)w_2'}{n}$. Since $w_1' > w_1$, it follows that $C_a' > C_a$.

Also by Lemma 2, $V = \frac{l(n-l)f^2}{n^2}$ and $V' = \frac{l(n-l)(w_1'-w_2')^2}{n^2} = \frac{l(n-l)(w_1+2f_i-w_1-f)^2}{n^2}$. Thus $V' > V$.

We showed that in all cases $L_m' > L_m$, $C_a' > C_a$ and $V' > V$. That proves lemma 7. □

We also investigate the case of single overlapping routings. Out of all single overlapping configurations only those that decrease the average communication cost relative to the cost of the minimum $NOCPRC$ are of special interest. As we show in the following Lemma 8, this happens only when single overlapping sites reside in the larger length-partite of the ring. In Figure 9 the ring of n sites is divided into four different zones. Zones A and B are of length l and zones C and D are of length $\lfloor \frac{n-2l}{2} \rfloor$.

We show in Lemma 8 that only single overlapping sites that reside in zones C and D reduce the average communication cost relative to the average communication cost C_a of the minimum $NOCPRC$.

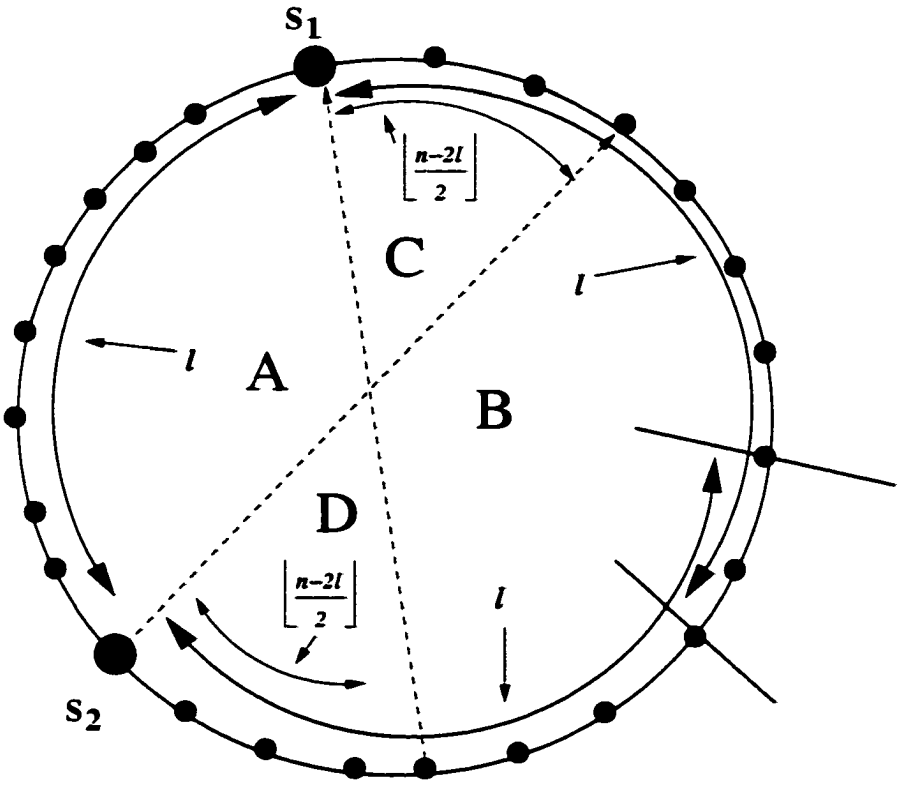

Figure 9. General case of C_a decrease by single overlapping paths.

Figure 10 shows a single overlapping site s_i in zone D that reduces the average communication cost. Note that there are four frequencies of communication associated with s_i and therefore four paths must be routed relative to s_i.

Definition 13: Any single overlapping site in a ring configuration whose two paths cross the other source is called *double path single overlapping site*. Any single overlapping site in a ring configuration whose only a single path crosses the other source is called *single path single overlapping site*.

In Figure 10(a) the single overlapping site s_i is a double-path but in Figure 10(b) it is a single-path. Note that double-path single overlapping sites reduce more the average communication cost than the single-path single overlapping sites. Thus for the rest of the lemmas in this chapter whenever we refer to single overlapping sites we only consider double-path single overlapping sites.

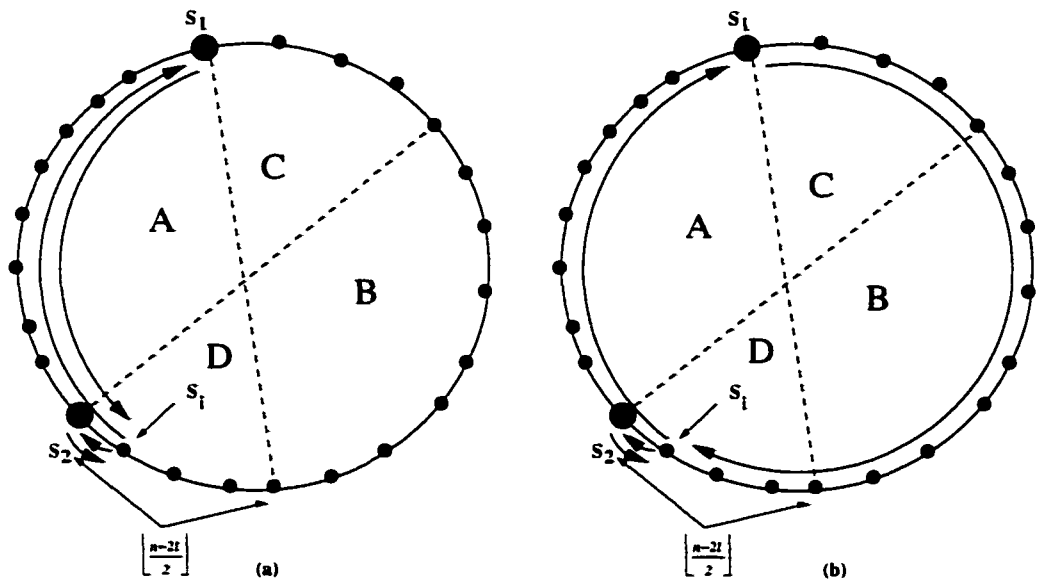

Figure 10. Two ways to create single overlapping paths between site s_i and the longer distance source.

Lemma 8: Let R_i be a minimum *NOCPRC* with partites of length l, with $l \leq \lfloor \frac{n}{2} \rfloor$ and $n - l$, respectively. Let C_a be the average communication cost resulted by R_i. Let also C and D be two sub-zones of the $n - l$ partite, each one adjacent to the sources s_1 and s_2, respectively, having length equal to $\lfloor \frac{n-2l}{2} \rfloor$. Figure 9 illustrates this configuration. Moreover, let R_i' be an overlapping ring

configuration resulted from R_i by one single overlapping site s_i. Furthermore, let C'_a be the average communication cost for R'_i. Then

$$C'_a = \begin{cases} \leq C_a & \text{if } s_i \in C \text{ or } s_i \in D \\ > C_a & \text{otherwise} \end{cases}$$

Proof: Let $s_i \in C$ where C is the sub-zone adjacent to source s_1. Let x be the distance of s_i to s_1 with $x \leq \lfloor \frac{n-2l}{2} \rfloor$. By the restriction of the $TSRP$, we assume that $f_{s_1,s_i} = f_{s_2,s_i} = f_{s_i,s_1} = f_{s_i,s_2} = f_i$. If s_i is not an overlapping site, then the routing of the above frequencies f_i contributes to the total communication cost for R'_i the value of

$$c_i = xf_i + (n-l-x)f_i + xf_i + (n-l-x)f_i = 2(n-l)f_i.$$

Figure 11(a) shows the routing of s_i. In that case, s_i is a single overlapping site. As Figure 11(b) illustrates, the routing of s_i contributes to the total communication cost for R'_i the value of

$$c'_i = xf_i + (l+x)f_i + xf_i + (l+x)f_i = 2(l+2x)f_i.$$

But $x \leq \lfloor \frac{n-2l}{2} \rfloor$. Thus

$$\begin{aligned} c'_i &\leq 2\left(l + 2(\lfloor \frac{n-2l}{2} \rfloor)\right) f_i \\ &\leq 2\left(l + 2(\frac{n-2l}{2})\right) f_i \\ &\leq 2(l+n-2l)f_i \\ &\leq 2(n-l)f_i \end{aligned}$$

$$\leq c_i$$

Note also that if $x < \lfloor \frac{n-2l}{2} \rfloor$, then $c_i - c_i' = 2(l+2x) - 2(n-l) = 2(n-2l-2x)$ which contributes to the average communication cost a decrease of $\frac{2(n-2l-2x)}{2n} = \frac{(n-2l-2x)}{n}$.

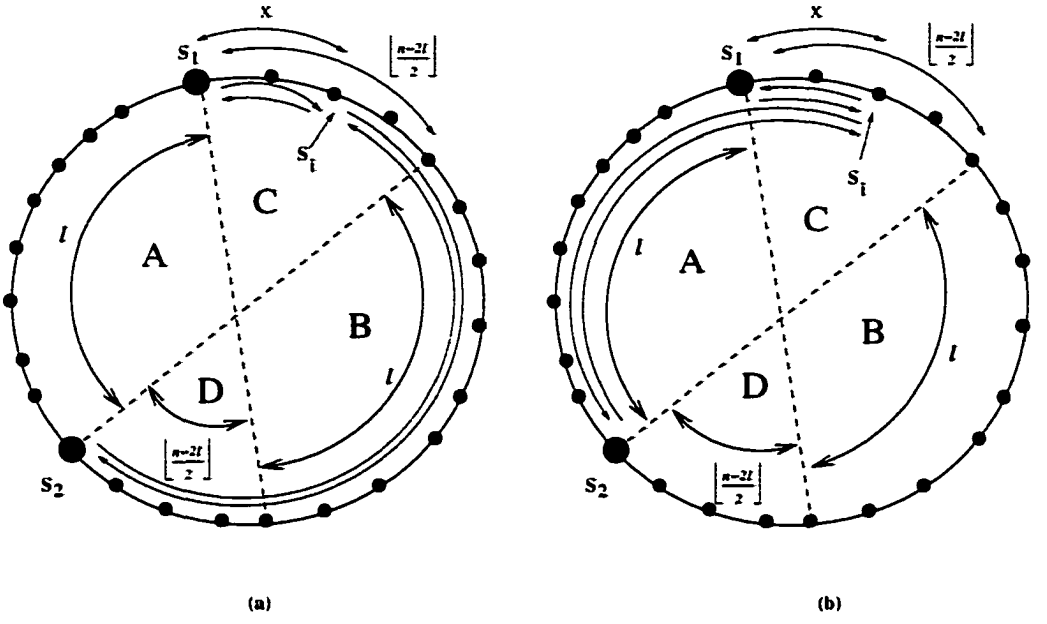

Figure 11. Non overlapping site s_i (a) and single overlapping s_i (b) in zone C that reduces C_a.

The exact logic is applied for the case that $s_i \in D$ with $x \leq \lfloor \frac{n-2l}{2} \rfloor$ being the distance of s_1 to source s_2. Thus, if $s_i \in C$ or $s_i \in D$, then $C_a' \leq C_a$.

Let $s_i \in A$ or $s_i \in B$ as Figure 12 illustrates. We show that if s_i is a single overlapping site, then R_i' results in an average cost C_a' such that, $C_a' > C_a$. We prove the case of $s_i \in B$. The case that $s_i \in A$ is proved in a similar way.

Let $s_i \in B$. The graphical illustration of this proof is shown in Figure 12(a).

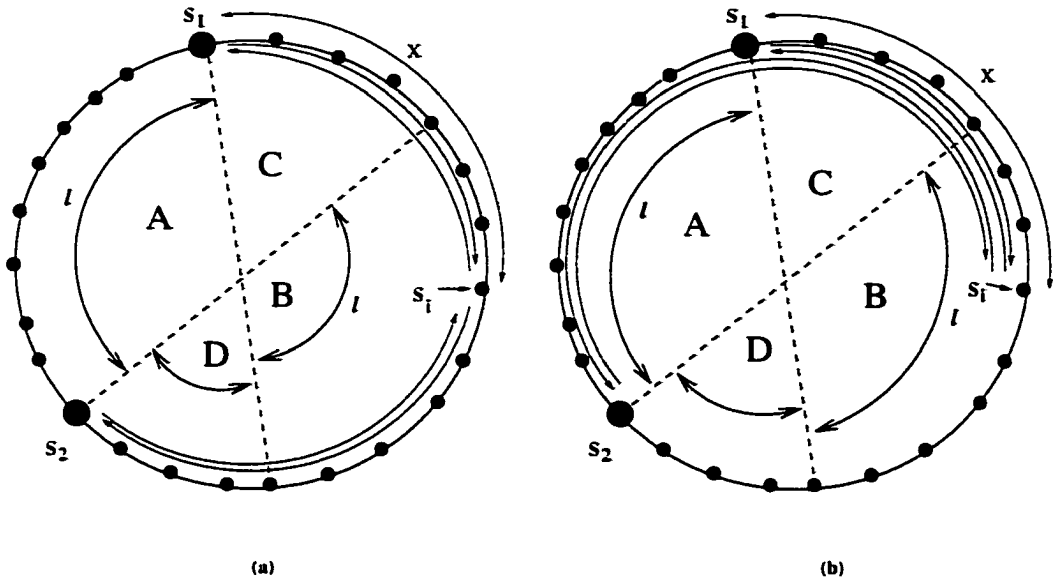

Figure 12. Non overlapping site s_i (a) and single overlapping s_i (b) in zone B that reduces C_a.

By the restriction of the $TSRP$, we assume that $f_{s_1,s_i} = f_{s_2,s_i} = f_{s_i,s_1} = f_{s_i,s_2} = f_i$. Let x be the distance of s_i to s_1 with $x > \lfloor \frac{n-2l}{2} \rfloor$. Let also $n-l-x$ be the distance of s_i to s_2. If s_i is not an overlapping site, then the routing of the frequency f_i contributes to the total communication cost the value of

$$c_i = xf_i + (n-l-x)f_i + xf_i + (n-l-x)f_i = 2(n-l)f_i.$$

If s_i is a single overlapping site (as Figure 12(b) shows), then the routing of its f_i frequencies contributes to the total communication cost for R_i'

$$c_i' = (l+x)f_i + (l+x)f_i + xf_i + xi = 2(l+2x)f_i$$

Since by assumption $x > \lfloor \frac{n-2l}{2} \rfloor$, it follows that

$$\begin{aligned} c_i' &> 2\left(l + 2(\lfloor \frac{n-2l}{2} \rfloor)\right) f_i \\ &> 2\left(l + 2(\frac{n-2l}{2})\right) f_i \\ &> 2(l+n-2l) f_i \\ &> 2(n-l) f_i \\ &> c_i \end{aligned}$$

Thus $C_a' > C_a$ which proves lemma 8. □

In the following lemma we prove that an overlapping ring configuration R_i' resulted from a minimum $NOCPRC$ R_i, which decreases the average communication cost relative to R_i results in variance V' such that $V' \geq 4V$ where V is the variance of R_i.

Lemma 9: Let R_i be a minimum $NOCPRC$ of n sites with partite lengths l, with $l \leq \lfloor \frac{n}{2} \rfloor$ and $n-l$. Let w_1 and w_2 be the buffer loads resulted by R_i and V the variance. Let also R_i' be a single overlapping ring configuration resulted from R_i by overlapping sites in the region $\lfloor \frac{n-2l}{2} \rfloor$ and V' its variance. Then $V' \geq 4V$.

Proof: For the minimum $NOCPRC$ by Lemma 2, the variance is $V = \frac{l(n-l)f^2}{n^2}$. Without loss of generality let's consider one site, namely s_i, that is adjacent to source s_2 as Figure 13 illustrates.

Let f_i be the values of the communication frequencies associated with s_i. The overlapping of routing for s_i creates 3 zones of different buffer loads in the

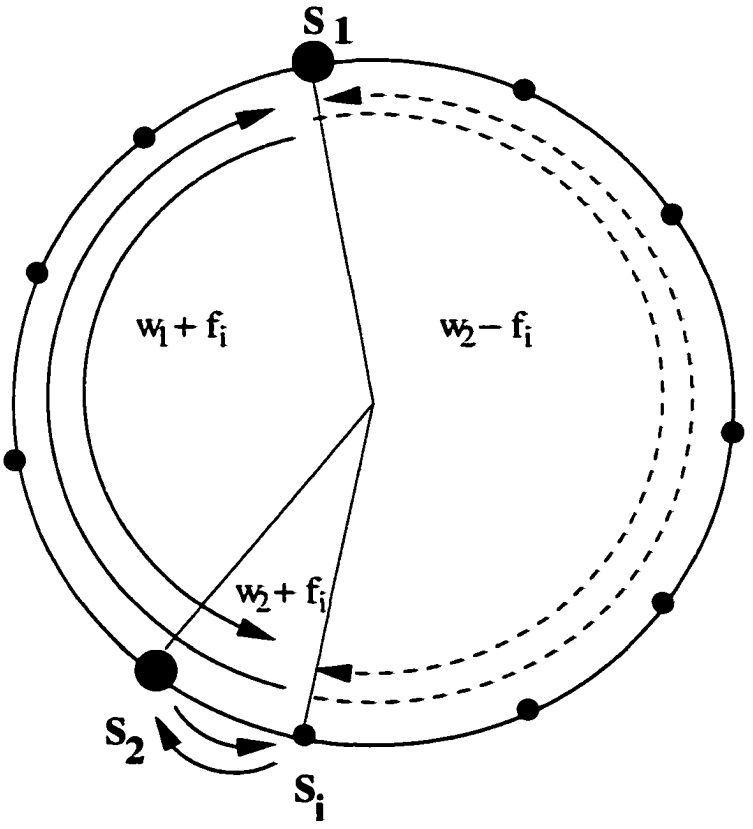

Figure 13. Variance for single overlapping ring configurations.

ring. The l partite buffers are charged with an additional load equal to f_i. The same happens for the two buffers associated with the link $\{s_i, s_2\}$. However, the remaining $2(n-l-1)$ buffers of the $n-l$ partite reduce their load by f_i. Furthermore, note that $f_i > f$ where, f is the minimum buffer load difference for R_i for if this was not the case, then s_i would belong to the other partite. Then

the resulting variance for this configuration of $2n$ buffer loads is

$$\begin{aligned}
V' &= \frac{1}{4n^2}\sum_{i=1}^{2l}\sum_{j>i}^{2l}(w_1+f_1-w_1-f_1)^2 \\
&+ \frac{1}{4n^2}\sum_{i=1}^{2l}\sum_{j=1}^{2(n-l-1)}(w_1+f_1-w_2+f_1)^2 \\
&+ \frac{1}{4n^2}\sum_{i=1}^{2l}\sum_{j=1}^{2}(w_1+f_1-w_2-f_1)^2 \\
&+ \frac{1}{4n^2}\sum_{i=1}^{2(n-l-1)}\sum_{j>i}^{2(n-l-1)}(w_2-f_1-w_2+f_1)^2 \\
&+ \frac{1}{4n^2}\sum_{i=1}^{2(n-l-1)}\sum_{j=1}^{2}(w_2-f_1-w_2-f_1)^2 \\
&+ \frac{1}{4n^2}\sum_{i=1}^{2}\sum_{j>i}^{2}(w_2+f_1-w_2-f_1)^2
\end{aligned}$$

Note that the first, fourth and sixth term of the above equation are zero. Note also that the load difference in the second term is $(w_1 - w_2 + 2f_i)$. However, $f_i > 2f$, because, if this was not the case, then this site would belong to the other partite. Therefore, the least value of $(w_1 - w_2 + 2f_i)$ is at least $3f$ as we have proven in Lemma 3. For the third term, the buffer load difference is f but for the fifth term the buffer load difference is $2f_i$.

Simplifying the above equation we get

$$\begin{aligned}
V' &= \frac{1}{4n^2}\left(4l(n-l-1)(w_1-w_2+2f_i)^2+4lf^2+4(n-l-1)(2f_i)^2\right) \\
&> \frac{1}{n^2}\left(l(n-l-1)(3f)^2+lf^2+4(n-l-1)f_i^2\right) \\
&= \frac{1}{n^2}\left(9l(n-l-1)f^2+lf^2+4(n-l-1)f_i^2\right) \\
&> \frac{1}{n^2}\left(4l(n-l-1)f^2+4(n-l-1)f_i^2\right)
\end{aligned}$$

$$> \frac{4(l+1)(n-l-1)f^2}{n^2}$$
$$> \frac{4l(n-l)f^2}{n^2}$$
$$> 4V$$

since $(l+1)(n-l-1) > l(n-l)$. Furthermore any additional single overlapping sites that decrease the average communication cost for R'_i will charge the buffers of the l partite with additional load and decrease the load of the buffers in the $n-l$ partite resulting in even greater variance V' $\square$.

Finally in the following lemma we prove that regardless if single overlapping routing decreases the average communication cost relatively to the minimum $NOCPRC$, the resulted variance and the maximum buffer load increase . This causes the optimization function $V \times {C_a}^2 \times {L_m}^2$ to have greater value than the one of the minimum $NOCPRC$ case.

Lemma 10: For any number of sites in the region $\lfloor \frac{n-2l}{2} \rfloor$, C'_a decreases but, V' and L'_m increase such that, $V' \times {C'_a}^2 \times {L'_m}^2 \geq V \times C_a^2 \times L_m^2$ of the minimum $NOCPRC$.

Proof : To prove this lemma we show first that $L'_m > L_m$. Let s_i be a single overlapping site in the region $\lfloor \frac{n-2l}{2} \rfloor$ of the $n-l$ partite with associated communication frequency f_i. Note that the overlapping routing caused by s_i charges the buffers of the l partite with an additional load f_i. We show that $L'_m > L_m$. If $w_1 > w_2$ then $L_m = w_1$ and $L'_m = w'_1 = w_1 + f_i$ therefore $L'_m > L_m$. If $w_1 \leq w_2$ then $L_m = w_2$. But $w'_1 = w_1 + f_i > w_2$ because if this was not the case, then s_i

would belong to the other partite. Thus $L'_m = w'_1 > w_2$ and $L'_m > L_m$. Therefore in all cases $L'_m > L_m$.

Since the maximum buffer load always increases, it suffices to prove that $V' \times C_a'^2 \geq V \times C_a^2$.

Let's index the sites in the region $\lfloor \frac{n-2l}{2} \rfloor$ as $\{s_{i_1}, s_{i_2}, \cdots, s_{i_{\lfloor \frac{n-2l}{2} \rfloor - 1}}\}$ where s_{i_1} is the farthest site from the source s_2 and $s_{i_{\lfloor \frac{n-2l}{2} \rfloor - 1}}$ is the adjacent site to s_2. We refer to Figure 14 for this illustration. Let's also index the communication frequencies that correspond to these sites as $\{f_1, f_2, \cdots, f_{\lfloor \frac{n-2l}{2} \rfloor - 1}\}$ in that order.

We also compute the biggest possible decrease relative to C_a. This happens when all sites in the region $\lfloor \frac{n-2l}{2} \rfloor$ are single overlapping sites with double path rerouting. Note in Figure 14 that, the path $(s_{i_1}, s1)$ when routed clockwise is smaller than the path $(s_{i_1}, s1)$ when routed counter-clockwise by 2 hops. Also the path $(s_{i_2}, s1)$ decreases in length by 4 hops when routed clockwise. Finally, the last in order site $s_{i_{\lfloor \frac{n-2l}{2} \rfloor - 1}}$ decreases its path length by $2(\lfloor \frac{n-2l}{2} \rfloor - 1)$ when it is routed to s_1 clockwise. Thus, the maximum possible decrease in C_a happens when all sites in the region $\lfloor \frac{n-2l}{2} \rfloor$ route their paths to s_1 with the clockwise direction and therefore

$$C'_a = C_a - \frac{1}{n}\left(2f_1 + 4f_2 + 6f_3 + \cdots + 2(\lfloor \frac{n-2l}{2} \rfloor - 1) f_{\lfloor \frac{n-2l}{2} \rfloor - 1}\right) = C_a - A$$

where $A = \frac{1}{n}\left(2f_1 + 4f_2 + 6f_3 + \cdots + 2(\lfloor \frac{n-2l}{2} \rfloor - 1) f_{\lfloor \frac{n-2l}{2} \rfloor - 1}\right)$.

In Lemma 9 we proved that only one single overlapping site in the region $\lfloor \frac{n-2l}{2} \rfloor$ results in variance V' such that $V' > 4V$ where V is the variance of the

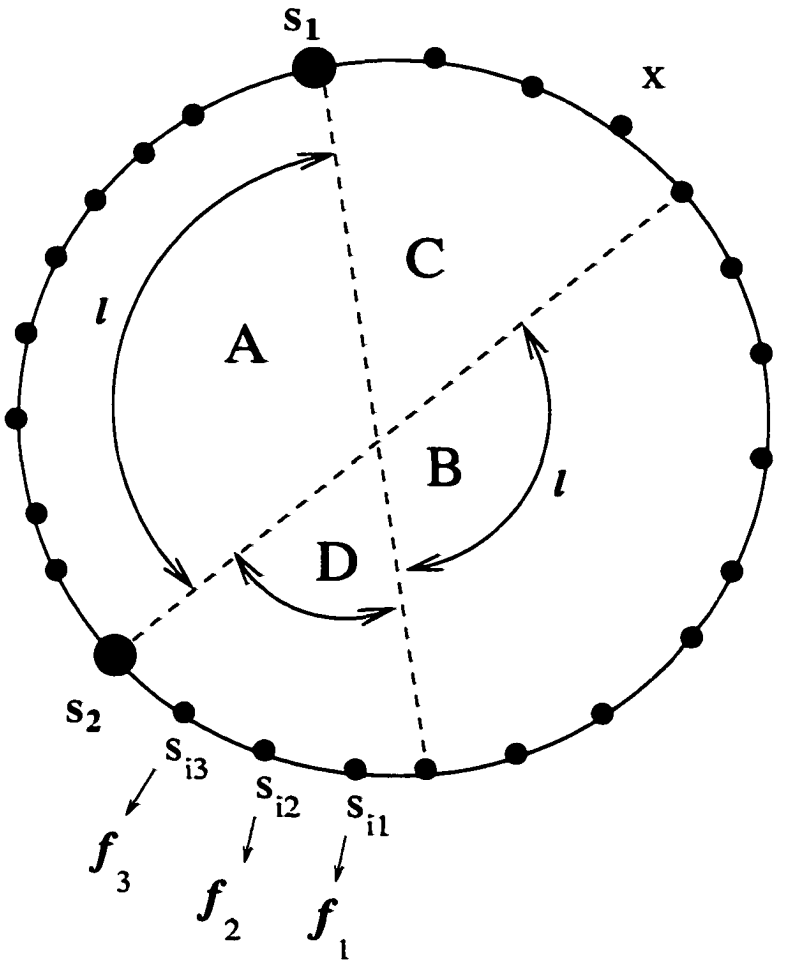

Figure 14. Single overlapping sites s_i that reduce the C_a.

minimum $NOCPRC$. However, the increase of variance due to only one site is enough to prove that $V' \times C_a'^2 > V \times C_a^2$:

$$\begin{aligned} V'C_a'^2 &> VC_a^2 \\ 4V\left((C_a - A)^2\right) &> VC_a^2 \\ 4(C_a - A)^2 &> C_a^2 \end{aligned}$$

$$\begin{aligned} 4(C_a^2 + A^2 - 2C_aA) &> C_a^2 \\ 3C_a^2 + A^2 &> 8C_aA \\ 3C_a^2 &> 8C_aA \\ 3C_a &> 8A \end{aligned}$$

To show the intermediate result $3C_a > 8A$ we make the following three observations:

1. Due to the triangulation property assumption for the $TSRP$, $l \geq \frac{n}{3}$ because if this was not the case, then R_i would not be the minimum $NOCPRC$.

2. For the quantity A defined above:

$$\begin{aligned} A &= \frac{1}{n}\left(2f_1 + 4f_2 + 6f_3 + \cdots + 2(\lfloor\frac{n-2l}{2}\rfloor - 1)f_{\lfloor\frac{n-2l}{2}\rfloor - 1}\right) \\ &< \frac{2}{n}\left(\lfloor\frac{n-2l}{2}\rfloor - 1\right)\sum_{i=1}^{\lfloor\frac{n-2l}{2}\rfloor-1} f_i \text{ and} \end{aligned}$$

3. $\sum_{i=1}^{\lfloor\frac{n-2l}{2}\rfloor-1} f_i < \frac{C_a}{2}$. This is because $C_a > min\{w_1, w_2\}$, the difference between w_1 and w_2 is less than any frequency f_i and the quantity w_2 is the summation of $\sum_{i=1}^{\lfloor\frac{n-2l}{2}\rfloor-1} f_i$ plus $(n - l - (\lfloor\frac{n-2l}{2}\rfloor - 1)$ more frequencies. However, $(n - l - (\lfloor\frac{n-2l}{2}\rfloor - 1)) > 2(\lfloor\frac{n-2l}{2}\rfloor - 1)$ and due to the triangulation property, the rest of the frequencies in the $n - l$ partite must have summation greater or equal than twice the summation of the frequencies of the sites in the region $\lfloor\frac{n-2l}{2}\rfloor$. Thus $\sum_{i=1}^{\lfloor\frac{n-2l}{2}\rfloor-1} f_i < \frac{C_a}{2}$.

Therefore,

$$3C_a > 8A$$

$$
\begin{aligned}
3C_a &> 8 \times \frac{2}{n}\left(\lfloor\frac{n-2l}{2}\rfloor - 1\right) \sum_{i=1}^{\lfloor\frac{n-2l}{2}\rfloor-1} f_i \\
3nC_a &> 16(n-2l) \sum_{i=1}^{\lfloor\frac{n-2l}{2}\rfloor-1} f_i \qquad \text{(Substitution } \sum_{i=1}^{\lfloor\frac{n-2l}{2}\rfloor-1} f_i < \frac{C_a}{2}\text{)} \\
6nC_a &> 16(n-2l)C_a \qquad \text{(Substitution } l \geq \frac{n}{3}\text{)} \\
6n &> \frac{16n}{3} \\
18n &> 16n
\end{aligned}
$$

which proves Lemma 10 □

NP-Completeness of the TSRP

In this section we prove that $TSRP \in$ NP_Complete. The polynomial transformation is done from the Closest Partition problem.

Theorem 2: $TSRP \in NP$_Complete.

Proof: Since a non-deterministic Turing machine needs only guess a permutation of n sites residing in zone 1 and zone 2 of the $TSRP$ design, compute in $O(n^2)$ time the variance and the average communication cost of the configuration and check for a given bound B if

$$\frac{\sum_{i=1}^{2|E'|}\sum_{j>i}^{2|E'|}(u_i-u_j)^2 \times \left(\sum_{i=1}^{2|E'|} u_i\right)^2 \times L_m^2}{(2|S|)^4} \leq B,$$

it follows that $TSRP \in$ NP.

Polynomial Transformation

Let $I_{CP} = (F)$ be an instance of the closest partition problem CP and

$I_{TSRP} = (F, s_1, s_2, B)$ an instance of the $TSRP$ for the set F with $|F| = n$ $n \in Z^+$ and $B \in Z^+$. Let also A and A' be the partites of the I_{CP} solution. Without loss of generality let $|A| = l$, with $l \leq \lfloor \frac{n}{2} \rfloor$ and $|A'| = n - l$.

Case $\Rightarrow$ For the solution $\{A, A'\}$ of I_{CP} let $w_1 = \sum_{i=1}^{|A|} f_i$ and $w_2 = \sum_{i=1}^{|A'|} f_i$ with $|w_1 - w_2| = f$. If $w_1 \geq w_2$, then the two zones of the ring in $TSRP$ can be defined. Moreover, a non-overlapping ring configuration R_i can be created. Note that R_i is the minimum $NOCPRC$ for the I_{TSRP} since it has the minimum possible variance and the minimum possible average communication cost (Lemmas 1 to 10).

If $w_1 < w_2$, then we have to check if there is an $f_i \in A$ and an $f_j \in A'$ such that, $(f_j - f_i) = f$ and their swap creates partites such that, $w_1 > w_2$ with $(w_1 - w_2) = f$. This checking can be done in order $O(n(n-l))$ and the result defines the minimum $NOCPRC$.

By Lemmas 1 and 2, C_a and V are defined for R_i. Furthermore, $L_m = max\{w_1, w_2\}$. If $V \times C_a^2 \times L_m^2 \leq B$, then a solution to $TSRP$ exists, else there is no solution.

Case $\Leftarrow$ Let R_i be the minimum $NOCPRC$ for the I_{TSRP} instance. Also let A be the set of communication frequencies associated with the sites in zone 1 of R_i and $A' = F_e - A$. Then A and A' define the two partites for I_{CP}. By assumption R_i corresponds to the minimum possible difference of buffer loads between its two zones. Thus, the A and A' partites correspond to the I_C solution.

Since $CP \in$ NP_Complete, it follows that $TSRP \in$ NP_Complete. □

An illustration of this polynomial transformation is depicted in Figure 15.

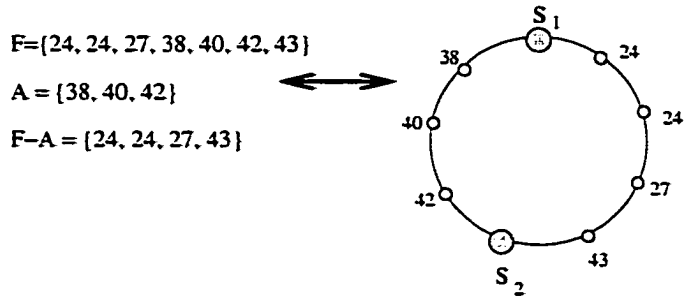

Figure 15. Polynomial transformation from CP to TSRP.

Theorem 3: $NDRP \in NP\_\text{Complete}$.

Proof: $NDRP \in NP$ since a nondeterministic Turing machine needs only guess a regular network topology of n sites and a routing scheme R, apply R for the set F_e, compute in $O(n^2)$ V, C_a and L_m and check for a given bound B if $V \times C_a^2 \times L_m^2 \leq B$

In Theorem 2 we proved that $TSRP \in$ NP_Complete. Since $TSRP$ is a subproblem of $NDRP$, it follows that $NDRP \in NP\_\text{Complete}$. □

CHAPTER III

REROUTING TO BALANCE LINK LOAD

Introduction

In the previous chapter we considered the design of a regular topology network and a routing scheme for this network. The design criterion used was the minimization of the quantity $V \times C_a^2 \times L_m^2$, i.e, the product of the variance of the edge loads with the square of the average buffer load and the square of the maximum buffer load across the network. The input data was the predicted frequencies with which, each pair of sites in the network communicates.

In this chapter we assume a regular network designed according to the set criterion. Once the network is in use for some time, the minimization of $V \times C_a^2 \times L_m^2$ might not hold. This will happen when the user communication patterns change. To restore the balance of the network, we have to change the routing scheme that accommodates the new patterns of user communication.

In this chapter we show that the rerouting is an NP-Complete problem. The proof of the NP-Completeness of this problem is based in the NP-Completeness of one of its subproblems. The subproblem we consider is a regular graph with specific structure and communication patterns.

We call the problem *Network Rerouting Problem*, (NRP) which is defined as follows:

Instance of (NRP) : Let $G = (S, E, F_e)$ be a connected regular network where, S is the set of sites, E is the set of bidirectional links connecting the sites, with $2|E|$ corresponding output buffers, $F_e = \{f_{i,j} = f_{j,i} | i, j \in S \text{ and } i \neq j\}$ is a set of frequencies of communication between any pair of sites and $B \in Z^+$.

Question : Is there a routing scheme $R_G = \{p_{i,j} \mid \text{a path } \forall\, i, j \in S \text{ and } i \neq j\}$ that will produce a set of buffer frequencies of use U_e such that

$$\sum_{i=1}^{|2E|} \sum_{j>i}^{|2E|} (u_i - u_j)^2 \left(\sum_{i=1}^{|2E|} u_i \right)^2 L_m^2 \leq B \, ?$$

Note that minimizing $V \times C_a^2 \times L_m^2$ is equivalent to minimizing the quantity $\sum_{i=1}^{|2E|} \sum_{j>i}^{|2E|} (u_i - u_j)^2 \left(\sum_{i=1}^{|2E|} u_i \right)^2 L_m^2$ since

$$V \times C_a^2 \times L_m^2 = \frac{\sum_{i=1}^{|2E|} \sum_{j>i}^{|2E|} (u_i - u_j)^2 \left(\sum_{i=1}^{|2E|} u_i \right)^2 L_m^2}{|2E|^4}$$

and $|E|$ is constant.

This chapter is divided in four sections. In the first section we prove two lemmas relatively to the graphicability of multi-sets. In the second section we prove two lemmas relatively to the bounds on the variance of specific sets of integers. In the third section we define a family of regular graphs which can be polynomially constructed from instances of the *Three Dimensional Matching Problem*, $(3DM)$, a well known NP-Complete problem. We show that any instance of the $3DM$ problem maps to a regular graph. We call these graphs $3DM -$

Regular. In the fourth section we define a subproblem of NRP and show that it is NP-Complete. This subproblem is a restriction of NRP on $3DM$-regular graphs. Finally we prove that the general problem of finding a routing scheme to minimize the product $V \times C_a^2 \times L_m^2$ in regular graphs, (denoted as $MNRP$), is also an NP-Complete problem using a polynomial reduction from the NRP.

Graphicability lemmas

In this section we prove two lemmas relatively to the graphicability of certain multi-sets. A *multi-set* is a set that allows multiple elements having the same value. Let A be a multi-set and $n = |A|$. A is called *graphical* if there is a graph of n-nodes with degrees equal to the elements $a_i \in A$. These graphs are used as subgraph components of a larger regular graph.

Lemma 11 : Let $a, b \in Z^+$ be odd integers with $a \leq b$. Then, the multi-set $A = \{a, b, b, \cdots, b\}$ with $|A| = b + 3 = n$ is always graphical.

Proof : Let $a_1 = a, a_2 = b, a_3 = b, \cdots, a_n = b$. For A to be graphical suffices to satisfy the *Erdõs-Gallai* inequality, i.e.,

$$\sum_{i=1}^{k} a_i \leq k(k-1) + \sum_{i=k+1}^{n} min(k, a_i) \quad \forall\, k \in [1, n]$$

For $k \leq b$ the above inequality becomes

$$a + (k-1)b \leq k(k-1) + k(b+3-k) \quad \Rightarrow a \leq 2k + b$$

which is true by the assumption $a \leq b$.

For $k > b$ the inequality becomes:

$$a + (k-1)b \leq k(k-1) + (b+3-k)b$$

which is true because $(k-1)b \leq k(k-1)$, since $b \leq k$, $a \leq (b+3-k)b$ and $b+3 \geq k$ □

Lemma 12 : Let $a, b \in Z^+$ and $a < b$ with a even and b odd. Then the multi-set $A = \{a, b, b, \cdots, b\}$ with $|A| = b+2 = n$ is always graphical.

Proof : Let $a_1 = a, a_2 = b, a_3 = b, \cdots, a_n = b$. For A to be graphical suffices to satisfy the *Erdös-Gallai* inequality, i.e.,

$$\sum_{i=1}^{k} a_i \leq k(k-1) + \sum_{i=k+1}^{n} min(k, a_i) \ \ \forall\, k \in [1, n]$$

For $k \leq b$ the above inequality becomes

$$a + (k-1)b \leq k(k-1) + k(b+2-k)$$

i.e. $a \leq k + b$ which is true because of the assumption $a < b$.

For $k > b$ the inequality becomes:

$$a + (k-1)b \leq k(k-1) + (b+2-k)b$$

which is true because $(k-1)b \leq k(k-1)$, since $b < k$, $a < (b+2-k)b$ and $(b+2) > k$ □

Variance calculations for sets with special values

In this section we prove two lemmas relatively to the bounds on the variance on sets with special values. In the first lemma we consider a set of values which can be either 0 or 1. In the second lemma we consider sets having some positive

integers and the rest of the elements have value equal with zero. In this case we show that the minimum possible variance occurs if and only if, all the positive integers have value equal with 1 regardless of the number of zero elements. We use these lemmas to prove that NRP is NP-Complete.

Lemma 13 : Let $A = \{0, 0, \cdots, 0, 1, 1, \cdots, 1\}$ be a set of zero or one integers with $|A| = n$. Let m be the number of zeros and r be the number of ones in the set, with $m + r = n$. Then the following are true:

1. If $n = 2k$ with $k \in Z^+$, then the maximum variance V in the set A occurs when $m = r$.

2. If $n = 2k+1$ with $k \in Z^+$, then there are two maxima for the variance V of A and these occur when the values of the pair (m, r) are $(\lfloor \frac{n}{2} \rfloor, \lfloor \frac{n}{2} \rfloor + 1)$ or $(\lfloor \frac{n}{2} \rfloor + 1, \lfloor \frac{n}{2} \rfloor)$.

3. The variance of A decreases when the quantity $|m - r|$ increases.

Proof : The set A has m 0-elements and r 1-elements, with $m + r = n$. The average of A is $\alpha = \frac{r}{m+r} = \frac{r}{n}$. Then the variance of A is:

$$V = \frac{1}{n}\sum_{i=1}^{n}(x_i - \alpha)^2 = \frac{1}{n}\left(\sum_{i=1}^{m}(\frac{r}{n})^2 + \sum_{i=1}^{r}(\frac{m}{n})^2\right) = \frac{mr(r+m)}{n^3} = \frac{mr}{n^2}$$

Thus, V is maximized when the product mr is maximized. For the first case, this happens when $m = r = \frac{n}{2}$. For the second case, this happens when $m = \lfloor \frac{n}{2} \rfloor$ and $r = \lceil \frac{n}{2} \rceil$ or $r = \lfloor \frac{n}{2} \rfloor$ and $m = \lceil \frac{n}{2} \rceil$.

To prove the third case, we distinguish all possible relations between m and r. If $m > r$, then $|m - r|$ increases when m increases. Let $x > 0$ be the increase

on m. Then $(m+x)(r-x) < mr$ thus, V decreases in the equation above. If $r > m$, then $|m-r|$ increases when r increases. Let $x > 0$ be the increase on r. Then $(r+x)(m-x) < mr$ thus, V decreases also. Finally, if $m = r$, then any increase on m or r decreases V. □

Lemma 14 : Let $A = \{0, 0, 0, \cdots, 0, a_1, a_2, \cdots, a_r\}$, $r > 0$ be a set of non negative integers with $|A| = n$ and $m = n - r$. Let also $a_i \geq 0 \; \forall \; 1 \leq i \leq r$ and $\sum_{i=1}^{r} a_i = r$. Then the variance of the set A is minimized when $a_1 = a_2 = \cdots = a_r = 1$.

Proof : For the set A, the average is $\alpha = \frac{r}{n}$ by the assumption of the lemma. The variance of A on the other hand is:

$$\begin{aligned} V &= \frac{1}{n}\sum_{i=1}^{n}(x_i - \alpha)^2 \\ &= \frac{1}{n}\left(\sum_{i=1}^{m}(0 - \frac{r}{n})^2 + \sum_{i=1}^{r}(a_i - \frac{r}{n})^2\right) \\ &= \frac{1}{n}\left(\frac{mr^2}{n^2} + \sum_{i=1}^{r}\left(a_i^2 - 2a_i\frac{r}{n} + \frac{r^2}{n^2}\right)\right) \\ &= \frac{1}{n}\left(\frac{mr^2}{n^2} + \sum_{i=1}^{r}a_i^2 - \frac{2r^2}{n} + \frac{r^3}{n^2}\right) \end{aligned}$$

Then V is minimized when the quantity $\sum_{i=1}^{r} a_i^2$ is minimized. Let $a_k = t > 1$, $1 \leq k \leq r$ be any of the a_i elements in A. Since $\sum_{i=1}^{r} a_i = r$, it follows that there must be $(t-1)$ a_i elements with value equal to zero. But $t^2 > t = \underbrace{1^2 + 1^2 + \cdots + 1^2}_{t}$.

Also in reverse, let $a_k = 0$, $1 \leq k \leq r$ be any of the a_i elements. Since $\sum_{i=1}^{r} a_i = r$, it follows that there must be another a_i element, namely a_q, $1 \leq q \leq r$ such that $a_q \geq 2$. But $a_q^2 > 2 = 1^2 + 1^2$. Thus, $\sum_{i=1}^{r} a_i^2$ is minimized when $a_1 = a_2 = \cdots = a_r = 1$ □

Construction of $3DM$-regular graphs

As we mentioned in the introduction, we use the $3DM$ problem as the basis for the NP-Completeness proof of a restrictive version of NRP. The $3DM$ problem is stated as follows, [32]:

Generic Instance of $3DM$: A set $M \subseteq A \times B \times C$ where A, B, C are disjoint sets having the same cardinality n.

Question : Does M contain a matching, i.e., a subset $M' \subseteq M$ such that $|M'| = n$ and no two elements of M' agree on any coordinate?

We show that for any instance of $3DM$, we can polynomially construct a $3DM$ regular graph. Let $m = |M|$ be the cardinality of the $3DM$ collection of triplets and n the cardinality of the sets A, B, C. We prove that for the $3DM$ regular graphs $G = (S, E)$, $|S|$ and $|E|$ are bounded by a polynomial on m, n.

Lemma 15 : Let $M \subseteq A \times B \times C$ be an instance of the $3DM$ problem, where A, B, C are disjoint sets having the same cardinality n. Then there is a regular graph $G = (S, E)$ polynomially constructed from M.

Proof : Let $m = |M|$ and $m_i = (a_i, b_i, c_i)$ be a triplet in M. The construction of the regular graph $G = (S, E)$ is as follows:

Description of S

$S = \{s_1, s_2\} \cup S_A \cup S_B \cup S_C \cup S_G \cup S_P$ where:

1. s_1 and s_2 are two dedicated vertices called *sources*.

2. Let t_{a_i} be the cardinality of the multi-set $\{a_i | a_i = \pi_1(m_i), m_i \in M\}$ where, π_1 denotes the first projection on M. Then $S_A = \cup_{i=1}^{n} S_{a_i}$ where, $S_{a_i} = \{a_{i_1}, a_{i_2}, \cdots a_{i_{t_{a_i}-1}}\}$ and $|S_A| = \sum_{i=1}^{n} |S_{a_i}| = m - n$. If $m \leq n$ then S_A is empty.

3. $S_B = \{b_i | b_i = \pi_2(m_i), m_i \in M\}$ where, π_2 denotes the second projection on M, i.e., the set of the b_i variables in M. Then $|S_B| \leq n$.

4. $S_C = \{c_i | c_i = \pi_3(m_i), m_i \in M\}$ where, π_3 denotes the third projection on M, i.e., the set of the c_i variables in M. Then $|S_C| \leq n$.

5. $S_G = \cup_{i=1}^{m} \{x_i, y_i\}$, i.e., two vertices for each $m_i \in M$. Then $|S_G| = 2m$.

6. Finally, S_P contains the additional vertices required by the graphicability lemmas to make the components regular graphs for each one of the sets of vertices $\{s_1, s_2\}, S_A, S_B, S_C, S_G$. Further details of how S_P is constructed will be given after the description of the set E.

Description of E

1. $\forall x_i \in S_G$ introduce the edge $\{s_1, x_i\}$.

2. $\forall b_i \in S_B$ introduce the edge $\{s_2, b_i\}$.

3. $\forall \{x_i, y_i\} \in S_G$ introduce the edge $\{x_i, y_i\}$.

4. $\forall b_i \in S_B$ introduce the edge $\{b_i, x_i\}$ if b_i appears in m_i.

5. $\forall y_i \in S_G$ introduce the edge $\{y_i, c_i\}$ if c_i appears in m_i.

6. $\forall y_i \in S_G$ introduce an edge $\{y_i, a_{i_j}\}$ for every vertex $a_{i_j} \in S_{a_i}$, i.e. if $a_i \in m_i$, then introduce edges emanating from y_i for every vertex introduced for a_i.

Figure 16 illustrates the construction of G thus far.

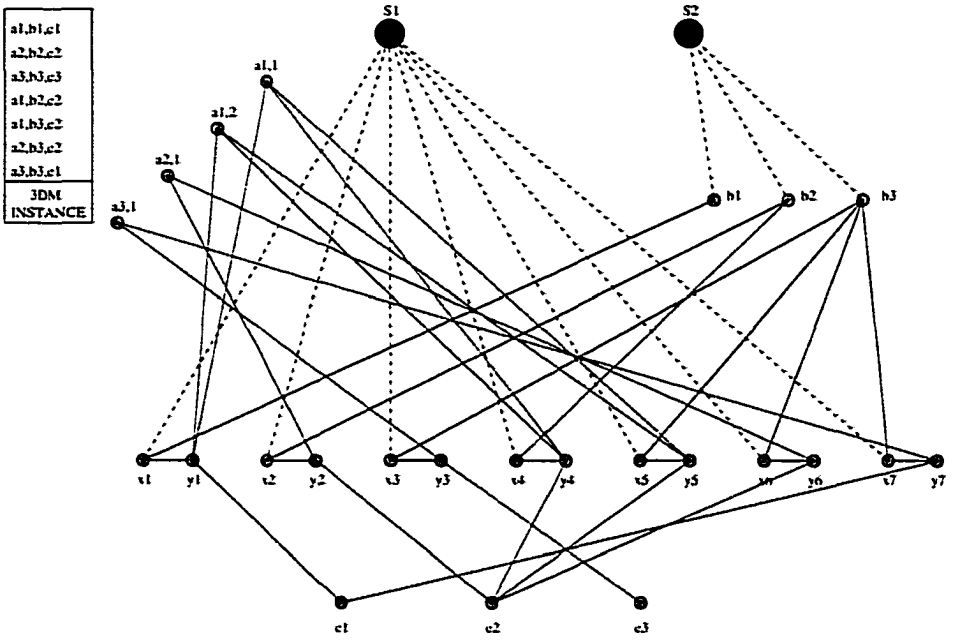

Figure 16. Partial construction of regular graph from a $3DM$ instance.

Note that the graph constructed thus far is not regular. We can make it regular by using the graphicability lemmas proved in the second section of this chapter. We introduce subgraph components, one for each of the vertices in $S - S_P$. Note that the vertex s_1 has the highest degree m. To make the graph regular, the degree of G must be higher than m. Let I_m be the minimum odd integer such that, $I_m > m$. Consider any vertex $s_i \in (S - S_P)$ with d_i its degree. If $I_m - d_i$ is odd, introduce a subgraph component attached to s_i using the multi-set $\{(I_m - d_i), I_m, I_m, \cdots, I_m\}$ of cardinality $I_m + 3$ according to Lemma 11. If $I_m - d_i$ is even, introduce a subgraph component attached to s_i using the multi-set

$\{(I_m - d_i), I_m, I_m, \cdots, I_m\}$ of cardinality $I_m + 2$ according to Lemma 12. Let V_{s_i} with $|V_{s_i}| = I_m + 2$ or, $|V_{s_i}| = I_m + 1$ be the set of the additional required vertices to make the i^{th} component regular. Then $S_P = \cup V_{s_i}$ and $E_P = \cup E_{s_i}$ is the set of edges introduced in the construction of these components. That completes the construction of G which is illustrated in Figure 17.

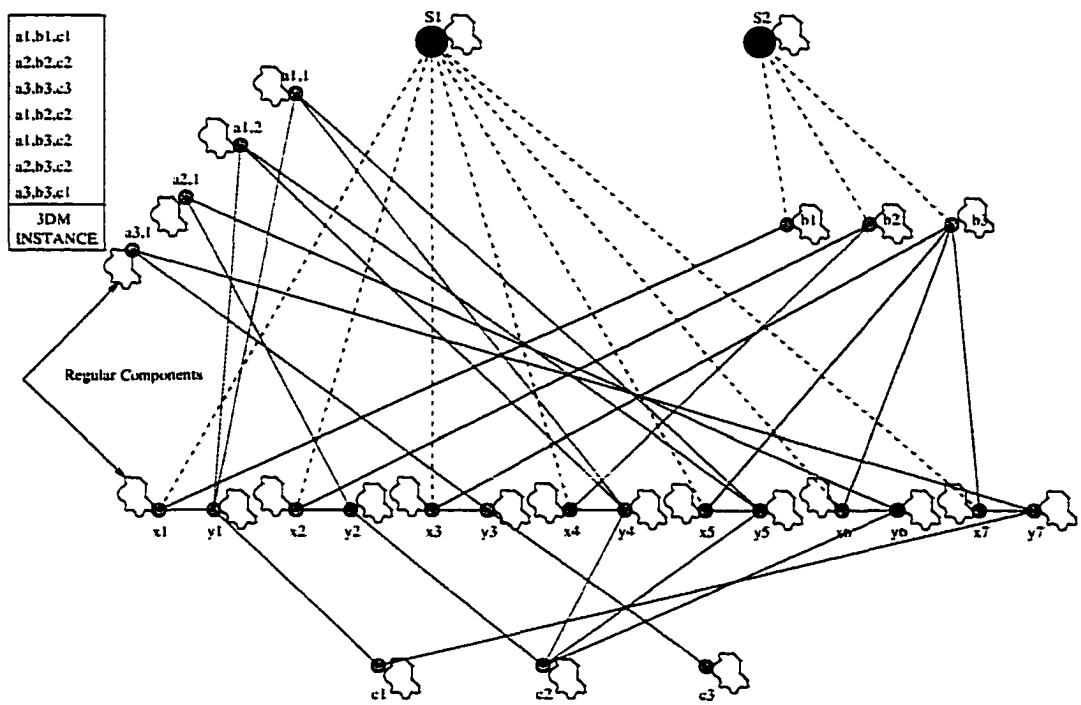

Figure 17. Full regular graph illustration from a $3DM$ instance.

Polynomial construction of G

Because $m \leq |A \times B \times C| \leq n^3$ and n polynomially relates to n^3, then n is a legal length function for $3DM$. We show that this construction of G from a $3DM$ instance can be done in polynomial time on the length function n. Note that

initially we introduced two vertices $\{s_1, s_2\}$, m pairs $\{x_i, y_i\}$, n b_i vertices, n c_i vertices and $(m-n)$ a_{i_j} vertices for a total of $(3m+n+2)$ vertices. For each of these vertices, we introduce at most (I_m+3) vertices to make the graph regular where $I_m < m+3$. Thus,

$$\begin{aligned}
|S| &\leq (|\{s_1,s_2\}| + |S_A| + |S_B| + |S_C| + |S_G| + |S_P|) \\
&\leq (|\{s_1,s_2\}| + |S_A| + |S_B| + |S_C| + |S_G|) + \\
&\quad (|\{s_1,s_2\}| + |S_A| + |S_B| + |S_C| + |S_G|)\,(I_m+2) \\
&\leq (|\{s_1,s_2\}| + |S_A| + |S_B| + |S_C| + |S_G|)\,(I_m+3) \\
&\leq (3m+n+2)(I_m+3) \leq (3m+n+2)(m+6) \\
&\leq (3n^3+n+2)(n^3+6).
\end{aligned}$$

Thus, $|S|$ is bounded by a polynomial on n.

The degree of G is $I_m \leq (m+3) \leq (n^3+3)$. Therefore,

$$|E| \leq \frac{|S|(m+3)}{2} \leq \frac{(3n^3+n+2)(n^3+6)(n^3+3)}{2}, \text{ i.e.,}$$

$|E|$ is bounded by a polynomial on n. Therefore, the transformation of $3DM$ to $3DM$-Regular graphs is polynomial. □

NP-Completeness of the NRP

In the introduction of this chapter we claimed that NRP is NP-Complete. To prove the claim, it suffices to prove that $NRP \in NP$ and a subproblem of NRP is NP-Complete. This subproblem belongs to the class of $3DM$-Regular networks

with a restricted communication frequency set F_e and a specific bound B. We refer to a similar proof by [49] who proved the NP-Completeness of unsplittable paths in networks. The subproblem $SNRP$ of NRP is as follows:

Generic Instance of $(SNRP)$: A $3DM$-Regular graph $G = (S, E, F_e)$ with $\{s_1, s_2\}$ the sources, $|S_B| = |S_C| = n$ and $|S_G| = 2m$, $F_e = F_1 \cup F_2 \cup F_z$ where $F_1 = \{f_{s_1,j} | f_{s_1,s_j} = 1 \ \forall s_j \in S_A\}$, $F_2 = \{f_{s_2,s_j} | f_{s_2,s_j} = 1 \ \forall s_j \in S_C\}$ and $F_z = \{f_{s_i,s_j} | f_{s_i,s_j} = 0 \ \text{ and } f_{s_i,s_j} \notin F_1 \cup F_2\}$, a bound $B = r(3m+n)^3 \in Z^+$ where $r = |2E| - (3m+n)$.

Question : Is there a routing scheme R_G for G that results into a set of frequencies of use U_e such that:

$$\sum_{i=1}^{|2E|} \sum_{j>i}^{|2E|} (u_i - u_j)^2 \left(\sum_{i=1}^{|2E|} u_i \right)^2 L_m^2 \leq B \ ?$$

Note that we use the same method of setting the minimization function as with the NRP, since minimization of $\sum_{i=1}^{|2E|} \sum_{j>i}^{|2E|} (u_i - u_j)^2 \left(\sum_{i=1}^{|2E|} u_i \right)^2 L_m^2$ is equivalent with the minimization of $V \times C_a^2 \times L_m^2$.

SNRP is a subproblem of NRP

Let D_{SNRP} be the domain of all instances of $SNRP$ and D_{NRP} the domain of all instances of NRP. Let also Y_{SNRP} denote the set of all "yes" instances of $SNRP$ and Y_{NRP} denote the set of all "yes" instances of NRP. To show that $SNRP$ is a subproblem of NRP, we need to show that $D_{SNRP} \subseteq D_{NRP}$ and $Y_{SNRP} = Y_{NRP} \cap D_{SNRP}$. Since $SNRP$ refers to a specific family of regular graphs and the NRP domain includes all regular graph instances, $D_{SNRP} \subseteq D_{NRP}$. Also

any "yes" answer to $SNRP$ belongs to D_{SNRP} and moreover, it is a "yes" answer to NRP for a given bound B. Furthermore, any "no" instance of $SNRP$ does not belong to Y_{NRP}. Therefore, $Y_{SNRP} = Y_{NRP} \cap D_{SNRP}$.

Theorem 4 : $SNRP \in$ NP-Complete.

Proof : To prove that $SNRP \in$ NP-Complete we show that $SNRP \in$ NP and also that $3DM$ polynomially transforms to $SNRP$.

Since a nondeterministic Turing machine need only guess a routing scheme R_G for G, of n-sites, apply the routing, compute in $O(n^3)$ the set U_e for G and check if $\sum_{i=1}^{|2E|} \sum_{j>i}^{|2E|} (u_i - u_j)^2 \left(\sum_{i=1}^{|2E|} u_i \right)^2 L_m^2 \leq B$, it follows that $SNRP \in NP$.

The polynomial transformation from $3DM$ to $SNRP$ is the one described in Lemma 15.

We now show that if there is a solution to $3DM$ instance, then there is a routing R_G in G that produces a set U_e to satisfy

$$\sum_{i=1}^{|2E|} \sum_{j>i}^{|2E|} (u_i - u_j)^2 \left(\sum_{i=1}^{|2E|} u_i \right)^2 L_m^2 \leq r(3m+n)^2$$

where $r = |2E| - (3m+n)$ and vice versa if there is a routing R_G that satisfies the above bound, then there is a solution to the $3DM$.

Case $\Rightarrow$: If there is a $3D$ matching, then there exists a set of paths that connect s_2 with all c_i's and s_1 with all $a_{i_j} \in S_A$ which are as follows:

For a triplet $m_i = (a_i, b_i, c_i)$ in the $3DM$ solution, use the corresponding pair of (x_i, y_i) sites and connect s_2 to c_i's through the path $(s_2, b_i, x_i, y_i, c_i)$. Since the cardinality of a $3DM$ solution is n and also the cardinality of the set S_C is n,

it follows that there is a one to one mapping between b_i's, (x_i, y_i)'s and c_i's. Note also that there are $m - n = |S_A|$ unused (x_i, y_i) pairs.

For each one of the sites in S_A, use the path (s_1, x_i, y_i, a_{i_j}) to connect a_{i_j} with s_1 where, (x_i, y_i) is the corresponding pair of vertices for a_{i_j}. That completes R_G.

Note that all paths are non-overlapping and have the shortest possible length. Therefore, the maximum buffer load produced by the set of frequencies of communication and the routing scheme is $L_m = 1$. Every path emanating from s_2 is of length 4, whereas, every path emanating from s_1 is of length 3. Thus, the total resulting communication cost is $T = 4n + 3(m - n) = 3m + n$. For the $|2E|$ buffers in G, $(3m - n)$ buffers have frequency of use 1 and $r = (|2E| - 3m + n)$ have frequency of use zero. Thus, $V \times C_a^2 \times L_m^2 \times |2E|^4$ is equal to

$$
\begin{aligned}
& \sum_{i=1}^{|2E|} \sum_{j>i}^{|2E|} (u_i - u_j)^2 \left(\sum_{i=1}^{|2E|} u_i \right)^2 \\
= & \left(\sum_{i=1}^{r} \sum_{j>i}^{r} (0-0)^2 + \sum_{i=1}^{r} \sum_{j>i}^{|2E|-r} (0-1)^2 + \sum_{i=1}^{|2E|-r} \sum_{j>i}^{|2E|-r} (1-1)^2 \right) \left(\sum_{i=1}^{|2E|} u_i \right)^2 \\
= & \left(\sum_{i=1}^{r} \sum_{j=1}^{3m+n} (0-1)^2 \right) (3m+n)^2 \\
= & \; r(3m+n)^3.
\end{aligned}
$$

For the example of the $3DM$ instance shown in Lemma 15 and Figure 17, the solution is illustrated in Figure 18 where, the selected paths are depicted with bold lines.

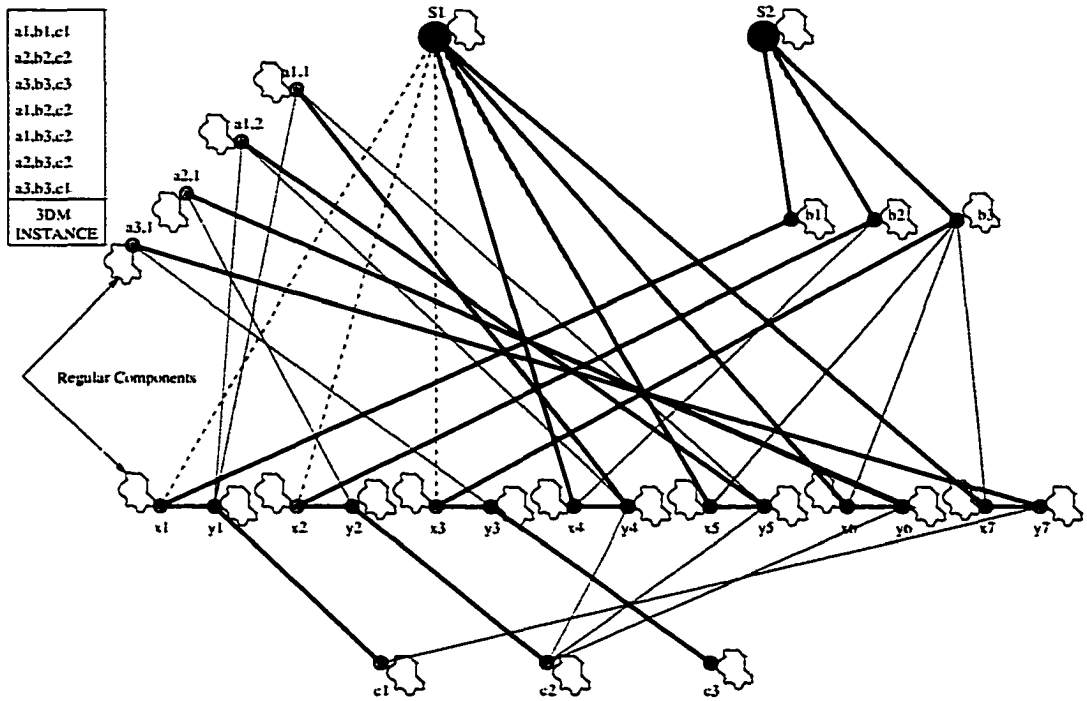

Figure 18. R_G solution produced by the solution of the $3DM$ instance.

Case $\Leftarrow$: Given a routing R_G that satisfies the bound, i.e.,

$$\sum_{i=1}^{|2E|}\sum_{j>i}^{|2E|}(u_i - u_j)^2 \left(\sum_{i=1}^{|2E|} u_i\right)^2 L_m^2 \leq r(3m+n)^3$$

we show that there is a $3D$ matching. Notice that the shortest paths from s_2 to c_i's are of length 4 and the shortest paths from s_1 to a_{i_j}'s are of length 3. Therefore, the minimum total communication cost is $T = 4n + 3(m-n) = 3m + n$. There are m (x_i, y_i) pairs of sites. The communication paths are disjoint because, there is a one to one correspondence between the set $S_A \cup S_B$ with the set S_G. Thus,

the maximum buffer load across the network is $L_m = 1$. For this routing

$$\sum_{i=1}^{|2E|}\sum_{j>i}^{|2E|}(u_i - u_j)^2\left(\sum_{i=1}^{|2E|}u_i\right)^2 L_m^2 = r(3m+n)^3$$

For any other routing R'_G which does not consist of mutually disjoint paths,

$$\sum_{i=1}^{|2E|}\sum_{j>i}^{|2E|}(u_i - u_j)^2\left(\sum_{i=1}^{|2E|}u_i\right)^2 L_m^2 > r(3m+n)^3$$

by Lemmas 13 and 14. Note that the correspondence of b_i's and c_i's in R_G defines a $2D$ matching. The paths which define this matching, i.e., $(s_2, b_i, x_i, y_i, c_i)$ pass through distinct (x_i, y_i) pairs. These pairs imply the existence of the corresponding vertices $\{a_1, a_2, \cdots, a_n\}$. Therefore, the a_i's with the $2DM$ for b_i's and c_i's define a $3DM$ for M. Since the $3DM$ problem is NP-Complete, it follows that $SNRP$ is also NP-Complete □.

Theorem 5 : $NRP \in$ NP-Complete.

Proof : $NRP \in$ NP-Complete since its subproblem $SNRP \in$ NP-Complete. □

We now define the minimization problem that corresponds to NRP which is denoted as $MNRP$. Furthermore, we prove that $MNRP$ and NRP are computationally equivalent which means that $MNRP \in NP$-Complete also.

Instance of $(MNRP)$: Let $G = (S, E, F_e)$ be a connected regular network where, S is the set of sites, E is the set of bidirectional links connecting the sites, $2|E|$ are the corresponding output buffers, $F_e = \{f_{i,j} = f_{j,i} \mid i, j \in S \text{ and } i \neq j\}$ is a set of frequencies of communication between any pair of sites and $B \in Z^+$.

Question : Is there a routing scheme $R_G = \{p_{i,j} \mid$ a path $\forall\, i, j \in S$ and $i \neq j\}$

that will produce a set of buffer frequencies of use U_e which minimize $V \times C_a^2 \times L_m^2$?

Theorem 6 : NRP and $MNRP$ are computationally equivalent thus, $MNRP \in NP\_Complete$.

Proof : To prove that NRP and $MNRP$ are computationally equivalent we show that, there is a polynomial reduction from one problem to the other. Let $S_{NRP}(S, E, F_e, B)$ be a subroutine which solves NRP. Then the $MNRP$ can be solved by a polynomial number of calls to $S_{NRP}(S, E, F_e, B)$ as the following algorithm in Figure 19 shows:

```
Step 1 : Set L =0 and H = B
Step 2 : While (H >L) do {
            Set mid = (L+H) / 2
            Call S_NRP(S, E, F_e, mid)
            if answer of S_NRP(S, E, F_e, mid) is "YES" then
                H = mid -1
            else
                L = mid +1
            }

OUTPUTS B* = The minimum bound solution
```

Figure 19. Algorithm for the polynomial equivalence of NRP and $MNRP$.

Let also $S_{MNRP}(S, E, F_e)$ be a subroutine which solves $MNRP$ and B^* the minimum bound solution found. If $B \leq B^*$, then there is no solution to NRP, else, there is a solution to NRP. Thus, the two problems are computationally

equivalent and therefore $MNRP \in NP\_\text{Complete}$. □.

CHAPTER IV

QOS NETWORK CONGESTION AVOIDANCE MECHANISMS AND DATA STRUCTURES

Introduction

In Chapters II and III we proved that the theoretical problems of designing and rerouting a regular network to minimize $V \times C_a^2 \times L_m^2$ belong to the class of $NP-$Complete problems.

In this chapter we establish that the minimization of $V \times C_a^2 \times L_m^2$ of frequencies of use reduces the probability for the network to experience congestion. This justifies that, the choice of parameters used in the above minimization function is essential in providing Quality of Service for the network.

QoS support is a growing need for network durability and also for all Internet applications. The two most popular efforts to provide QoS are the *Best Effort Mechanism* and the *Resource Reservation Mechanism*. For the case of Best Effort Service, paths for end-to-end services are established according to the current state of the network. If the communication required must be reliable, then it must be accomplished by making sure that the message sender retransmits lost or rejected packets/cells. Unfortunately, this method does not provide a mechanism

for congestion avoidance but rather attempts to cure the problem when it occurs. Therefore, communication networks based on the Best Effort paradigm are no capable of providing Quality of Service necessary for high traffic applications such as voice, video and multimedia services.

On the other hand, the Resource Reservation method for unicast and multicast connections employs mechanisms for bandwidth reservation for reliable communications. However, when the traffic becomes heavier, only higher priority services keep transmitting information, whereas, lower priority applications experience *bandwidth starvation*. It can happen that, the network is not congested (from a global point of view) but from the user's point of view, it looks congested since some processes do not progress. Therefore, QoS guarantees can only be achieved in the resource reservation protocol, but only when we make sure that lower traffic applications have a fair treatment by the system.

We show in this chapter that the minimization of $V \times C_a^2 \times L_m^2$ reduces the probability of congestion in the network. However, to measure the congestion, we need to define congestion first. The most common definitions of congestion are:

Definition 14 : Congestion in a network system occurs when the average throughput rate of the network decreases below a threshold.

Definition 15 : Congestion in a network system occurs when the average transit delay of cells/packets at intermediate sites/links along their path increases due to high traffic.

However, the above definitions characterize congestion with the indications of side effects caused by it and not with the indication of reasons that cause the phenomenon. Thus, the measurement of the average throughput rate or the average transit delay is not an appropriate substitution for the congestion measurement.

An attempt for the quantification of congestion has been done lately by Monteiro et. al. in [62]. The authors give a new definition for the congestion of a communication system:

Definition 16 : A communication system is congested whenever the functioning of communication services is affected in a way adversely percepted by their users.

The authors provide also a metric for congestion which depends on each of the parameters defined for the QoS model imposed. Since this is the only congestion metric to our knowledge, we use it to justify that the global criterion of minimizing $V \times C_a^2 \times L_m^2$ reduces the probability of congestion of every network link.

The most common network model, (and the most realistic one) assumes *bursty* communication demands in time. We use this assumption to discover unsplittable flow routings that provide QoS guarantees under the criterion of minimizing the probability of hot spot creation. More specifically, there are two goals: Primarily to divide the available resources (mostly bandwidth) as fairly as possible and secondly to maximize the total network communication throughput.

Even though the most acceptable logic in Internet applications is the *Best Effort Routing* (BER), usually expressed in the form of *max-min fairness*, [53], there are variations in the QoS criteria used. We select to minimize at any time the product $V \times C_a^2 \times L_m^2$ since this is the optimization function of interest.

A selection criterion is the minimization of the maximum link load at any time in the network. This problem has be proven to be NP-Complete by proving NP-Completeness of a special case, the *Ring Loading Problem*, [72]. But this criterion does not guarantee congestion avoidance. Note that the criterion does not restrict the case of having all network links fairly unbalanced, which is a major factor in increasing the probability of hot spots. Also the criterion to minimize the average link load C_a when taken with no other variables is not sufficient for the same reason.

On the other hand, the variance minimization of link loads criterion does not guarantee reduction of the probability of congestion. The reason is that, when we minimize V, we may obtain routing schemes that actually increase the probability of congestion since, the minimization of variance may force the routing algorithm to increase path lengths and link loads and therefore charge the buffers with unnecessary extra load.

However, the selected criterion $V \times C_a^2 \times L_m^2$ guarantees reduction of the variance but only in low levels of C_a and L_m. Minimizing this quantity in the network, it provides a method of congestion prevention. For bursty communica-

tion demands, we cannot have an accurate projection of how link loads variate in time. But using the history of network demands between pairs of sites, we can have a good prediction of the buffer loads. We use these projections of buffer loads to predict the probability of the network system to experience congestion in the next period. This prediction is done by calculating the projected frequencies of use for each network buffer. When the prediction of congestion turns positive for a certain percentage of buffers, then we try to avoid congestion by slowing down the injection rate of cells/packets from certain sources or more drastically by rerouting the system. The rerouting process attempts to reallocate fairly buffer loads according to our criteria such that, the new bandwidth allocation will reduce the probability of congestion. This idea is used extensively in fair allocation of resources in operating systems and it can be applied in communication networks.

We consider only static routings. In today's Internet applications where, someone can transmit high volume data (movies, sound, teleconferencing etc.), static routing is used since it allows an algorithmic help to traffic predictions and macroscopic congestion avoidance.

The following five sections in this chapter describe our model and tie it with today's technology. The first section describes all the assumptions used for the communication mechanisms in backbone networks and intra-nets. This includes topologies, connections through routers, (single or multi-protocol), link configuration and data control. The next section justifies that, the QoS minimization

criterion $V \times C_a^2 \times L_m^2$ reduces the probability of congestion in the network using the metric described in [62]. In the third section we describe the basis of the routing schemes which we use. We delay their algorithmic description for later and we only describe the idea of predictability for communication frequencies through an exponential average formula. The next section describes the data structures involved in our model and summarizes the storage requirements for each server to accommodate this operation. Finally, the last section describes algorithmically the method of congestion detection and suggests an improved technique of packet rejection that is tied to the traffic demands of the system.

Model assumptions and technology relevance

The backbone and metropolitan area networks are formed by connecting LANs at gateway points leaving them unchanged. We are able to connect rings, stars, buses on a ring or a bus on a hypercube. This is accomplished by installing a router at their connection point. Figure 20 illustrates such a typical backbone topology. In this section we describe the existed technology on routers, the physical layer implementation protocols and the methods used to implement the data control layer.

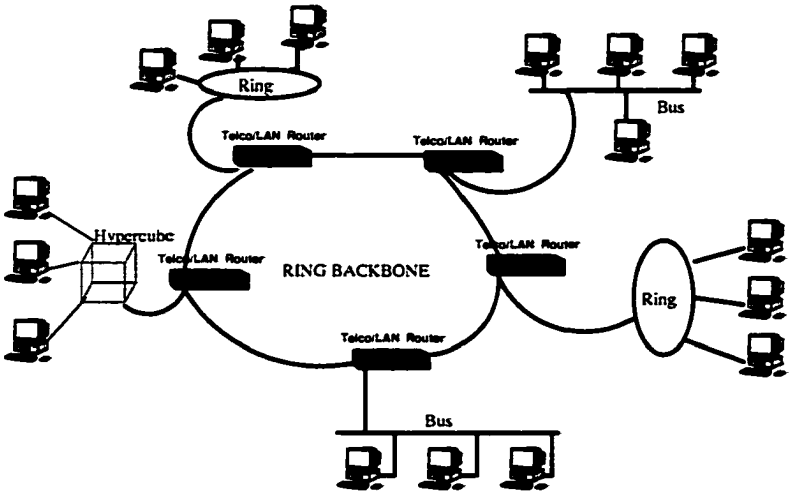

Figure 20. A typical backbone network.

Routers and backbone nets

The routers came to replace repeaters and bridges few years ago since they are more sophisticated hardware devices. Routers have access to the different network layer addresses and also contain necessary software that enables them to determine routing/rerouting schemes. They operate on the three lowest levels of the OSI model, i.e., the physical, the data link and the network layer. We concentrate our interest on the communication of LANs leaving the existed local routing schemes to operate independently. The routers can operate in both static or adaptive routing schemes.

Physical layer implementation and topologies

The most common protocol used in the physical layer for backbone and metropolitan area nets is called *Distributed Queue Dual Bus* (DQDB). This protocol has been established by international standards organizations (IEEE project 802.6). A popular service based on this protocol is the *Switched Multi-megabit Data Service* (SMDS) for handling high speed communications. SMDS is a *packet-switched datagram service* provided by common carriers. Subscriber LANS link to SMDS through routers. The links used are bidirectional either half or full duplex. For half duplex linkage there is provision of collision control. Making bandwidth reservation requests and transmitting messages requires buffering which is implemented by using two queues for each link, one for each traffic direction. Our model complies with the half duplex DQDB. Since the traffic is bidirectional for every link, there are two output queues associated, one in each of the adjacent sites as Figure 21 illustrates.

However, packetization of messages needs additional handling due to the fact that messages can be split into packets of different sizes. This complicates any prediction statistics relatively to the distribution of arrival times of packets. Few years ago, a new protocol was created to sit on top of the DQDS in order to handle this problem. It is called the *Asynchronous Transfer Mode Protocol* (ATM) and it rapidly becomes today's standard. The basic innovation of ATM

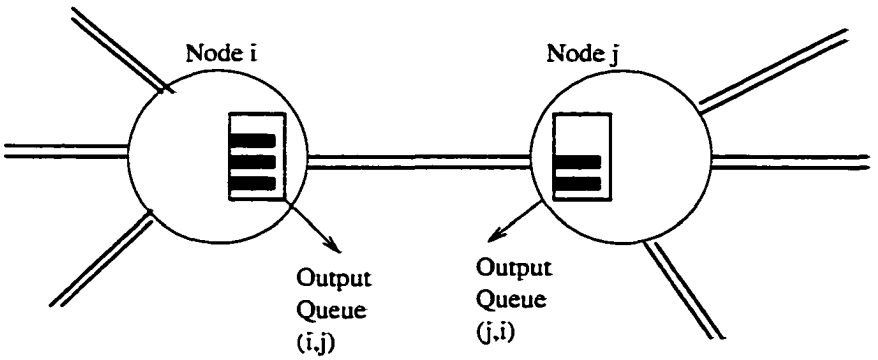

Figure 21. Output queues for a link.

is the splitting of messages into *cells*, i.e., fixed size message units. In a cell network, all data are loaded into identical cells that can be transmitted with complete predictability and uniformity. The advantage of ATM protocol is that, the network never experiences congestion since all sites inject cells into the network with a constant rate (Constant Bit Rate). However, from the users point of view, the existence of congestion lies on the fact that, the transmission process is slowing down. Our model can been seen as a hybrid approach of DQDS and ATM. The goal is to keep the potential of the network to handle high traffic and at the same time to prevent congestion. This is achieved by a mechanism that detects the probability of future congestion and a procedure to avoid its occurrence.

Data link control

Our model assumes that, no station is allowed to transmit or forward messages to a prospective receiver unless, there is evidence that the receiver is prepared to accept and process the transmission. This function is a basic function of the *data link control* layer of OSI. We assume the existence of the *Line Discipline* functionality of our model based on one of the two most popular methods : *enquiry/acknowledgment* (ENQ/ACK) or *poll/select*.

Congestion reduction with $V \times C_a^2 \times L_m^2$ minimization

In this section we show that, whenever we minimize $V \times C_a^2 \times L_m^2$, we also reduce the probability for the network to experience congestion. As we mentioned in the introduction we use the congestion metric introduced in [62]. The method of measuring congestion for a network that guarantees several QoS parameters is simple. For every parameter q_i relative to a service s_i, the QoS designer implies an *optimal zone* in which, the value of q_i must exist at all times. In our case, we consider the buffer load as the QoS parameter at hand. Note that, the smaller the quantity of traffic that passes through a link relatively to the routing scheme, the smaller is the probability for that link to become a hot spot. Because our model assumes half duplex links, the link load is distributed to the two buffers associated with each link.

Consider a network with traffic demands given by the set of frequencies of communication F_e. Note that, F_e is measured across a time interval (Δt). Let R be the routing scheme chosen for the communications at hand. Since the goal is to minimally balance the traffic in every buffer of the network, somebody could suggest that this can happen when we minimize the average communication cost C_a. However, this is not true since the minimization of C_a can result into individual link loads fairly unbalanced.

If we try to measure the congestion of the network in every interval Δt, we have to measure how the user who requests a service experiences this congestion. If the routing scheme results into unbalanced link loads, this indicates that, the system does not utilize the available resources in low traffic links where at the same time, the higher transit delay in high traffic links deteriorates the network's performance. This means that not only the over-utilization of certain links but also the under-utilization of others contributes to the congestion increase. If we assume an optimal link or buffer load zone for the network, then the congestion depends on the deviation of the buffer loads which also indicates that the buffer load behaves as a symmetric QoS parameter.

The authors in [62] measure the deviation of symmetric QoS parameters using the following definition:

Definition 17 : Let q_i be a QoS parameter, $v_i(t)$ its value at instance t, m_i and M_i its normal variation limits, with θ_{m_i} and θ_{M_i} its QoS degradation thresholds.

Then the QoS parameter deviation for a link $\{j,k\}$ is given by the formula

$$Id_{q_i,j,k,t} = \begin{cases} 0 & \text{for } m_i \leq v_i(t) \leq M_i \\ 1 - 10^{-(\frac{m_i - v_i(t)}{\theta_{m_i}})} & \text{for } v_i(t) < m_i \\ 1 - 10^{-(\frac{v_i(t) - M_i}{\theta_{M_i}})} & \text{for } v_i(t) > M_i \end{cases}$$

The above $Id_{q_i,j,k,t}$ function can is graphically illustrated in Figure 22.

Note that, the above definition corresponds to instantaneous values of the QoS parameters but, it can be used in exactly the same way when the buffer or link load measurements are taken across a time interval Δt. In our case, $v_i(t)$ becomes $v_i(\Delta t)$ and $v_i(\Delta t) = u_{i,j}(\Delta t)$ which is the frequency of use for a buffer (i,j) with m_i and M_i to be the lower and upper limit, respectively, for the optimal load zone imposed.

The value of the deviation of each buffer load is the contribution of that buffer to the global congestion of the network. Then for a network of E half duplex links, the total congestion is the summation of the deviations of all buffer loads and it is given by the following formula:

$$Con(\Delta t) = \sum_{j=1}^{2|E|} Id_{q_i}$$

We now show that the minimization of $V \times C_a^2 \times L_m^2$ also guarantees reduction of the global network congestion with high probability. To illustrate our claim we choose the Two Source Routing Problem, $TSRP$ from Chapter II. We then generalize to any regular topology.

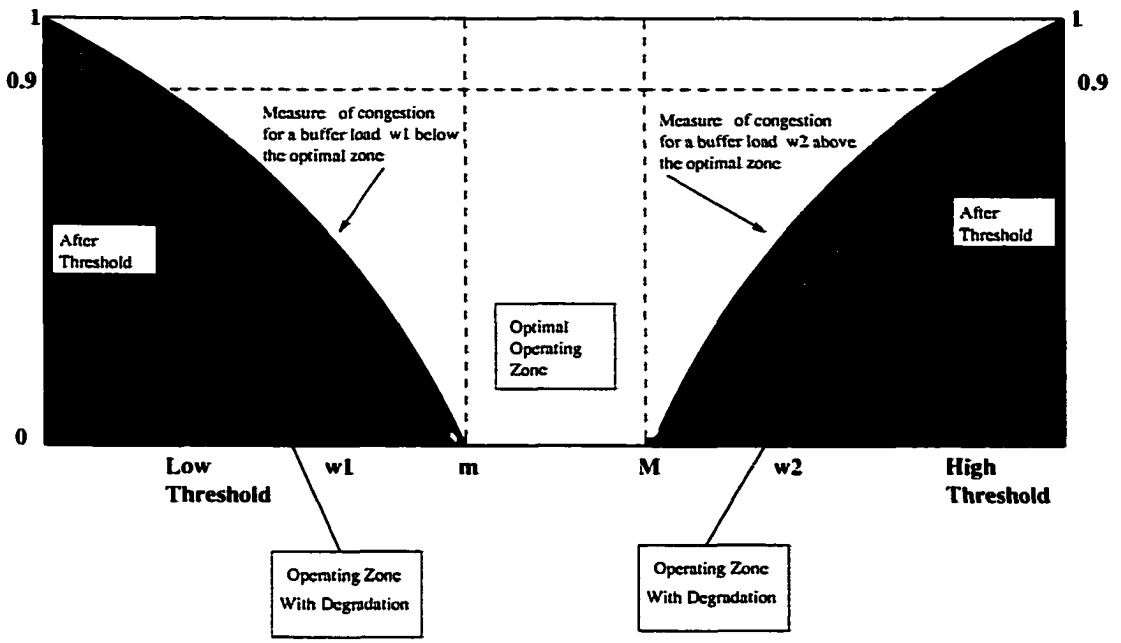

Figure 22. Deviation of QoS parameters.

Theorem 7 : Let $G = (S, E, F_e)$ be a ring of $|S| = n$ sites with $|E| = n$ half duplex links, $2n$ buffers associated with the n links and F_e be the set of frequencies of communication measured at some time interval Δt. Let also the set F_e satisfy the following conditions for the $TSRP$:

1. $f_{s_1,s_2} = f_{s_2,s_1} = 0$ for the two dedicated sources s_1, s_2.
2. $f_{s_1,s_i} = f_{s_2,s_i} = f_{s_i,s_1} = f_{s_i,s_2} > 0 \quad \forall \quad 1 \leq i \leq n$ and $i \neq 1, 2$.
3. $f_{s_i,s_j} = 0 \quad \forall \quad i, j \neq 1, 2$.
4. The set F_e satisfies the triangulation property.

Then the minimization of $V \times C_a^2 \times L_m^2$ reduces the probability for congestion of

the ring G.

Proof: For the ring G, let the congestion be defined as the summation of the deviations of the buffer loads resulted by a routing scheme R.

Let $CPRC$ be the ring configuration that results in the closest partition of the communication frequencies $f_{i,j}$. Moreover, let R be the routing scheme for that configuration such that, R minimizes $V \times C_a^2 \times L_m^2$. In Chapter II we have proven that R must be non-overlapping configuration. Let w_1, w_2 be the buffer loads for the ring when R is applied. Without loss of generality, let $w_1 \leq w_2$, where $f = w_2 - w_1$ is the minimum possible buffer load difference. We prove that R results into reduction of the probability for G to get congested for the set F_e.

Let m and M be the values of the lower and upper buffer load. These values define the optimal deviation zone in G according to Definition 17. Let also θ_m and θ_M be the lower and upper degradation thresholds for these deviations.

Let R_i be any other routing scheme for G. The configuration R_i results in buffer loads $\{w_{i_1}, w_{i_2}, \cdots, w_{i_{2n}}\}$ with possibly some of the w_{i_j}'s to be equal. Note that, there is no value w_{i_j} such that $w_1 < w_{i_j} < w_2$. If there was such a buffer load, then the $f_{i,j}$'s that constitute w_{i_j} could also define a closer non overlapping partition in G and therefore could result into a smaller product $V \times C_a^2 \times L_m^2$. Figure 23 illustrates five cases referring to the relative positions of w_1, w_2, m and M.

For the cases (A), (B) and (C), R results in no congestion for G. Note

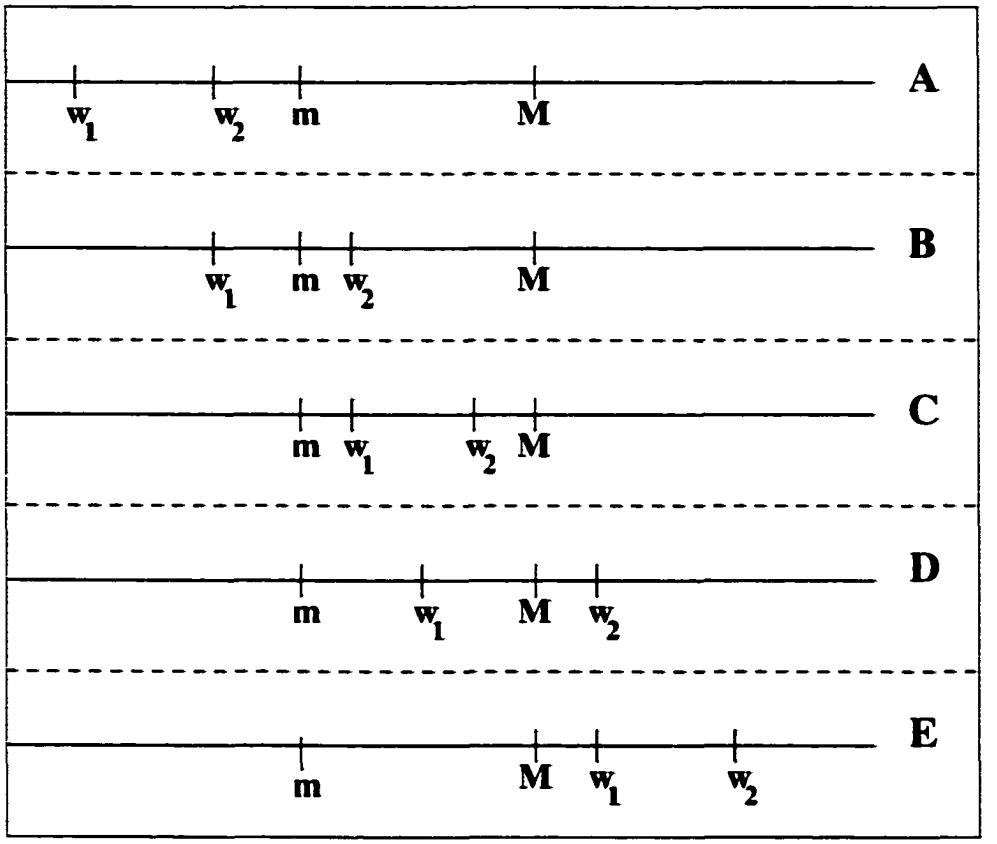

Figure 23. Relative position of buffer loads and optimal operating zone to avoid congestion.

that since w_2 is the maximum load in G and $w_2 < M$, all buffers in G operate with loads within or under the optimal zone. In that case, the buffer load is not a symmetric QoS parameter. However, R_i results in buffer loads greater than w_2. The reason is that, R gives the closest partition of the frequencies thus, any other routing scheme results in a maximum load increase among all buffers. For all loads w_{i_j} in R_i such that $w_{i_j} > w_2$, we increase the probability for $w_{i_j} > M$. In

that case, R_i results in congestion for G.

In case (D), all buffers with load w_2 cause congestion in G. However, any other routing scheme R_i results in loads $\{w_{i_1}, w_{i_2}, \cdots, w_{i_{2n}}\}$ such that, there is no value w_{i_j} with $w_1 < w_{i_j} < w_2$. Note that, R_i results in $w_{i_{max}} - w_{i_{min}} > f$ of R, increasing the deviations for all $w_{i_j} > w_2$ and all $w_{i_j} < m$. Also any routing scheme R_i that results in smaller variance than R, it succeeds to do so at higher values of link load. In both cases, R_i increases the probability of hot spot creation in G.

Finally, for the case (E) note that, R results in buffer loads which are all above the upper limit of the optimal operating zone. The reason is that, all buffers have load greater than M. However, R results in a the minimum possible L_m for G. Let R_i be any other routing scheme for G. As we have shown above, there is no load w_{i_j} in R_i with value $w_1 < w_{i_j} < w_2$. For all the loads greater than w_2, greater congestion is produced resulting into higher probability for a hot spot creation in G. Also if R_i results into smaller variance than R, this must be at higher values of loads producing greater congestion. □

For the case of any regular network topology the minimization of the optimization function $V \times C_a^2 \times L_m^2$ tends to reduce the variance of link loads at low levels of average communication cost and maximum buffer load. Routing schemes that minimize $V \times C_a^2 \times L_m^2$ increase the probability of congestion avoidance. The reason is that any other routing scheme will either increase the variance of

link loads and with high probability increase the congestion or will decrease the variance of link loads by increasing the average communication cost, therefore increasing the probability of network congestion.

Theorem 7 does not assume any relation between the size of the optimal operating zone and the difference between the maximum and minimum buffer load. However, this is important to the network design since, it directly depends on the projection of pairwise demands. Using the projected pairwise frequencies of communication, we would like to create an optimal operating zone which would include the minimum and maximum buffer load value of the routing scheme that minimizes $V \times C_a^2 \times L_m^2$. That gives also an indication of the size of bandwidth which is needed for the network to operate. On the other hand, the degradation thresholds θ_m and θ_M directly depend on the buffer size and the bandwidth availability of the network. The bigger the size of the Distributed Queue Dual Bus, the bigger is the congestion threshold we can afford. Also the bigger the bandwidth of the network the smaller the thresholds can be for fine tuned network systems since, there is a decrease of the transit delay and therefore packets stay less time in the buffers. Typical networks operate with degradation thresholds to be a fraction of the buffer size (20% to 30%).

Exponential average traffic prediction

The proposed network model uses static and unsplittable flow routing techniques. Some adaptivity is supported indirectly with the use of predictions of future traffic volumes between pairs of sites. At fixed time intervals, the system checks the value of $f_{i,j}$ for each pair (v_i, v_j). This is the amount of communication in the recent past. The system predicts the near future value of $f_{i,j}$ and this prediction is based on the history of this communication pattern.

The prediction approach assumes that, network demands have a bursty nature. We may not know the next traffic burst between two sites but, we can predict its value by expecting this value to be similar to the previous ones. The next traffic demand is generally predicted as an exponential average of the measured demands of previous intervals.

Let $F_{i,j}^n$ be the measured traffic in the n^{th} interval. Let also $f_{i,j}^{n+1}$ be the predicted value for the frequency of communication for the next interval. Then

$$f_{i,j}^{n+1} = \alpha F_{i,j}^n + (1-\alpha) f_{i,j}^n$$

The value of α defines the ratio relevance of the past and recent history. If for example $\alpha = \frac{1}{2}$, this means that they are equally weighted. For the case of $\alpha = 0$, the recent history has no effect. On the other hand, if $\alpha = 1$, then only the most recent history affects the predicted value without any effect from past predictions.

The same technique is used extensively in the multitasking operating sys-

tems for calculating the next CPU burst in the case of *short term CPU scheduling* [74]. Note that if we expand the above exponential average formula, then we get

$$f_{i,j}^{n+1} = \alpha F_{i,j}^{n} + (1-\alpha)\alpha F_{i,j}^{n-1} + \cdots + (1-\alpha)^{k}\alpha F_{i,j}^{n-k} + \cdots + (1-\alpha)^{n+1} f_{i,j}^{0}$$

which shows that, the initial network designer projections relatively to traffic demands are kept in the very last term and also, all the behavior of the system from the time it has been in use affects the next predicted value. A typical value of α can be based 75% in the recent history and 25% in the past history. This ratio of percentages captures the drastic changes in $f_{i,j}$'s which occur in different times of the day.

Keeping track of the current value of $F_{i,j}$ as well the previous prediction $f_{i,j}$ is important to compute its next predicted value. However, the discussion of how these values are stored in every server is delayed for a later section.

The effect of the exponential average time interval

The interval Δt for the measurement of the frequencies of communication as well as the buffer frequencies of use, represents the time period over which, the link and buffer utilization variables are updated in the system.

A sufficiently small value for Δt obtains finer tuning of the network system since, the predictability of traffic changes is more reliable. This also means that, the routing changes adapt faster to these traffic behavior alternations. However, the smaller the value of Δt, the higher the communication overhead needed among

servers for the decision making process of measuring and treating network congestion. Note that, this server communication is done using the *Flooding Technique* (each server informs its neighbors about local statistics and eventually all servers share the state of every network link).

The larger the value of Δt, the coarser the tuning of the network system is. This has the advantage of reducing the communication overhead. On the other hand, bigger intervals Δt reduce the ability of the system to capture drastic traffic changes and therefore delay its response.

The value of Δt must have the ability to be tunable by the software to adapt future bandwidth increases and new technologies. Some networks, (mostly fiber optical) allow the use of dedicated fibers for information sharing among servers. This can allow the value of Δt to further decrease resulting in more responsive systems.

Congestion detection and traffic control actions

In the previous sections we described how the imbalance of link loads affects the network congestion. We also provided a metric for the congestion that depends on the deviation of loads. Since the network traffic behavior has a dynamic bursty nature, we also provided an exponential average formula for traffic predictions. These predictions may be used for rerouting the network in order to reduce the probability of hot spot creation.

However, the rerouting of a network is costly. Therefore, we need a software mechanism to detect the congestion and decide whether or not rerouting is needed. This software module is an essential part of the data link and network layer of every network.

Our model provides such a mechanism to monitor user behavior and determine link congestion. This mechanism is fully distributed to each site and it is based on statistics gathered at the local output queues. Each site predicts the future load that passes from its adjacent links at every time interval Δt. At the same time, we also measure the frequencies of use for each link and compute the average frequencies of use for the interval Δt.

Let m and M be the lower and upper limit for the traffic load operating zone of the queues in a network. Let also θ_m and θ_M be the lower and upper degradation thresholds respectively. Then the deviation for each average frequency of use can be computed. Overloaded and underutilized links contribute more to this deviation. If the average frequency of use is greater than $M+\theta_M$, then the link is overloaded. If the average frequency of use has value less than $m-\theta_m$, then the link is underutilized.

Every site maintains an array of all adjacent output queues indicating the congestion status for each one at every interval Δt. We call this array the *Site Congestion Vector* (SCV). Each element of the array may have one of the following three values: 1 if the associated link is over-loaded, 0 if the associated

link operates without exceeding the degradation limits and -1 if the associated link is under-utilized.

Each site communicates the states of its adjacent links (or local output queues) with its neighbor sites. With the use of flooding technique or by transmissions via dedicated fibers in an fiber optical medium finally, all sites know the congestion status of every other site in the network. However, for a network of n sites and degree d, we need an $n \times d$ table in each site to store the congestion status of the whole network.

Every site computes the *Percentage of Over-Loaded Links*, (POL) and the *Percentage of Under-Utilized Links*, (PUL) for the whole network. The computation of POL is done by counting all the values of 1 in the congestion vectors. The computation of PUL is done by counting all the values of -1 in the congestion vectors. We set two thresholds relative to POL and PUL, namely θ_{POL} and θ_{PUL}. The values of θ_{POL} and θ_{PUL} are known at each server site.

If $POL > \theta_{POL}$ but $PUL < \theta_{PUL}$, this indicates that the system experiences congestion but the number of under-utilized links is not sufficient to accommodate the extra traffic in the over-loaded links. In that case, rerouting of the network may result in a better link load balancing but it cannot avoid the congestion caused by the extra traffic. Therefore, rerouting is not triggered in that case. On the other hand, the data link control initiates a slow-down on the traffic passing through the over-loaded links by rejecting packets/cells. Details on

the packet/cell rejection policy used are given in the end of this section.

If $POL < \theta_{POL}$, the system continues to deny an initiation of network rerouting regardless of the value of PUL. This case indicates that, the network has very few over-loaded links but possibly several under-utilized links. Even though the rerouting of the network seems logical, the introduced communication overhead cannot be over-performed by the load alleviation of very few links. Packet rejections will still happen in the over-loaded links but, the side effects in network degradation performance are less significant than the previous case due to the small number of the links participating.

Finally in the case that $POL > \theta_{POL}$ and $PUL > \theta_{PUL}$, this indicates that there are a lot of over-loaded links but at the same time there are also enough under-utilized links to accommodate the extra traffic. In this case, rerouting is triggered. Furthermore, at the same time, all sites stop initiating new traffic and all packets at intermediate nodes are quickly transmitted to their destinations.

The thresholds θ_{POL} and θ_{PUL} are software variables and they are subject to the fine tuning of the network model. A detailed analysis of the pairwise traffic distributions may produce θ_{POL} and θ_{PUL} such that, a decrease on the average number of reroutings does not compromise the overall system performance. In Figure 24 we illustrate the congestion detection algorithm and we proceed with the description of existed packet rejection policies. We also introduce a new variation of a packet drop policy which is based on the distributions of pairwise

traffic in the network.

```
congestion_detection()
{
    if (mod(time, Δt) ==0)
    {
        compute the projected frequencies of use for each buffer
        set the site congestion vector, (SCV) for every site
        flood the SCV to every site
        compute POL and PUL
        if POL > θ_POL and PUL > θ_PUL then
            Trigger Rerouting
        else
            Initiate Packet Drop Policy on overloaded links
            reset frequency of communication counters
            continue
        endif
    }
}
```

Figure 24. Detection congestion algorithm.

Packet rejection when congestion occurs

When the system cannot avoid congestion, it must provide mechanisms to recover from it. The only way to do it is by slowing down incoming traffic. Also in networks that guarantee sessions at a certain bandwidth rate, the system may also refuse to establish such connections. However, today's technologies cure the problem by introducing selective or random packet/cell dropping policies. We briefly refer to these policies and we also introduce a new policy called *Probabilistic*

Packet/Cell Dropping Policy. This policy complies with our exponential average model of projecting future traffic volumes based on the history of the system.

Existed packet/cell dropping policies

Selective packet dropping policies have been used to reduce congestion and transmission of traffic that would inevitably be retransmitted. For data applications using best-effort services, packet dropping policies (PDPs) are congestion management mechanisms implemented at each intermediate node that decide, reactively or pro-actively, to drop packets/cells to reduce congestion and free up precious buffer space. While the primary goal of PDPs is to combat congestion, the individual PDP designs can significantly affect application throughput, network utilization and performance fairness. The following policies are the most popular:

1. **Drop Tail** : The drop tail scheme is the simplest of all policies. The Drop Tail scheme does not selectively drop packets/cells, it just drops them whenever there is no buffer space available. Moreover, there is no consideration about the state of message generation between sources, destinations and intermediate sites along their paths.

2. **Drop from Front** : This strategy tries to improve the performance of the system by inserting a little more education in the decision to which packets are dropped. Whenever the buffer is full and a new cell/packet comes in, we

make room for the incoming cell by dropping the cell closest to the head-of-line position. The idea behind this decision is that, with high probability most of the buffer contains packets/cells from the source currently forwarded at that instance. However, this policy seems more randomly oriented rather than probabilistic since, it solely depends on the instance of the new incoming traffic and the head of the buffer at that time.

3. **Early Random Drop** : According to this policy, whenever the output queue is filled up to a certain level, then the server drops every new incoming traffic packet with a fixed drop probability relative to the total number of (source, destination) pair paths that are routed through that buffer. This policy is the most fair among all but, it normalizes the probability of drop-rates for every source without considering the individual projected volumes of traffic between any pair of sites.

Probabilistic packet/cell dropping policy

This technique is similar to the *Early Random Drop* policy. However the decision making process to which packets/cells are to be dropped when buffer sizes reach a certain level is not based on uniform probabilities of traffic. Since the model is able to project the frequencies of frequency for every pair of (source, destination) that contributes to the specific buffer congestion, we use these frequencies to compute the *weighted probabilities of dropping rates* for every pair

of communicating sites. The fraction $fr_{s,d}$ for the source-destination pair (s, d), defined as

$$fr_{s,d} = \frac{f_{s,d}}{\sum_{k=1}^{r} f_{i,j}}$$

indicates the probability for the buffer to receive a packet/cell from the (s, d) communicating pair in the interval Δt. The value of $fr_{s,d}$ is recomputed in each time interval. Note that, since $f_{s,d}$ is kept locally in each site, this computation needs no flooding. For each pair of communicating sites that pass through a site i, the percentage $fr_{s,d}$ is kept in the local table.

When decision for dropping packets/cells in a full buffer is needed, the first candidate packet to be dropped is the one that is associated with the highest $fr_{s,d}$. Then this table entry is tagged so that next time, we drop a packet of the communicating pair with the second higher value $fr_{s,d}$ and so forth.

This policy increases fairness over the Early Random Drop policy because, it imposes an ordering of dropping packets/cells relative to the probability of their occurrence.

Data structures used for statistics storage

In this section we summarize the data structures involved in each site for the implementation of network load balancing. We classify these data structures relatively to routing, traffic prediction and congestion detection. We also comment on the space requirements needed for our model. In the end of the section, we

illustrate these data structures in a table.

1. Routing : Note that the next hop for a packet at a site i depends not only on its destination but also on its source. This means that, the traffic (i_1, k) and (i_2, k) passing from the site i is not necessarily forwarded to the same output queue. Therefore, an $n \times n$ routing table is needed at each site to accommodate the current routing scheme and future reroutings. The entries to the routing table at the site i are the *ids* of the next site in the path of each communication pair. For the communication pairs which do not include the site i in their paths, the corresponding routing table entry is set to empty or -1.

2. Traffic Prediction : At every time interval Δt, the predicted traffic for each communication pair passing through the site i is computed locally. This prediction is based on the previous interval prediction, which records the history but also on the measure of the actual traffic that passed through. Therefore, for each communication pair, two variables are needed, namely, $f_{i,j}$ and $F_{i,j}^{PR}$. The variable $F_{i,j}^{PR}$ is used as a counter that counts the incoming traffic in each time interval. These variables may be included in the routing table for each communication making the routing table entries records of multiple entities. The values of α and Δt are also kept in each site.

3. Congestion Detection : The site congestion vector is a local data structure in each site. For regular networks of degree d, the size of SCV is also set. For a site i with d associated output queues, the SCV is represented as an array of

pairs of integers:

$$[\, Site\ i \mid (i_1, st_1),\ (i_2, st_2),\ \cdots\ (i_d, st_d)\,]$$

where the ordered pair (i_j, st_j) corresponds to the output queue $(i, i_j$ with congestion status $st_j = 1$ or 0 or -1. The values of m, M, θ_m and θ_M are also kept in each site to compute the SCV.

However, in order to store the congestion status of every output queue in the network, we need an $n \times n$ table. For convenience, we include the congestion status as an extra field in the routing table. More specifically, in the (j, k) position of the routing table at site i, we add a field to denote the congestion status of the output queue (j, k) of the network. The values of θ_{POL} and θ_{PUL} are also kept in each site to determine if routing is needed.

Finally, at the same record entry (j, k), an extra field is added to store the weighted probability of dropping rate for the traffic (j, k) that passes through the site i, named $f_{r_{j,k}}$. The value of $f_{r_{j,k}}$ is calculated only for the local output queues which are over-loaded. Figure 25 illustrates the structure of the routing table and Table 1 summarizes all the data structures.

For the routing table each site needs an $n \times n$ array of structures of 5 integers each. For the site congestion vector, each site needs an array of $(d + 1)$ integers. Finally, for the rest of the variables, each site needs to reserve space for 10 more integers as Table 1 shows. Therefore, the total space requirement for

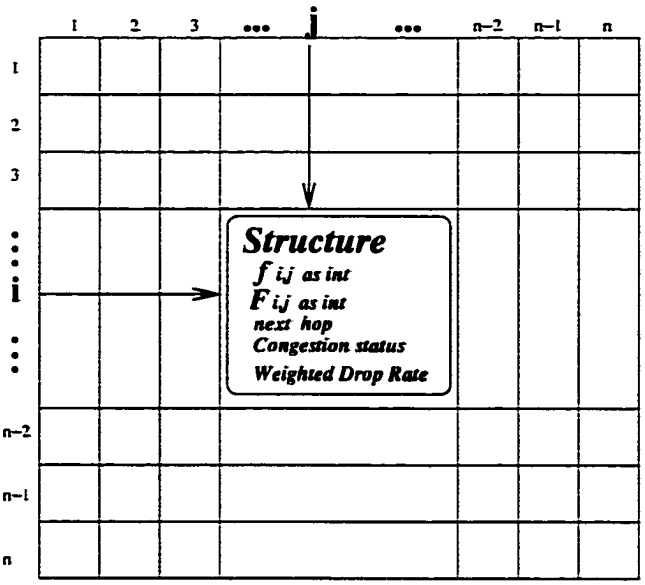

Figure 25. Structure of the routing table and statistics storage.

each site of our model is $5n^2 + (d+1) + 10$. That results into space complexity of $O(n^2)$ for each network site and $O(n^3)$ for the whole network .

Table 1

MODEL DATA STRUCTURES

Routing Table	$n \times n$ array of 5 field records
Exponential Average Variables	$\Delta t,\ \alpha$
Link Congestion Calculation	$m,\ M,\ \theta_m,\ \theta_M$
Site Congestion Vector	Array of $d+1$ entries
Rerouting Decision	$POL,\ PUL,\ \theta_{POL},\ \theta_{PUL}$

CHAPTER V

ROUTING ALGORITHMS

Introduction

Backbone and high speed metropolitan area networks are expected to support high speed interconnections for multiple and often diverse applications (ranging from simple e-mail transmissions to real-time video conferencing). Thus, the main goal in designing routing algorithms is the provision of available resources in order to meet the requirements for each individual transmission. For real-time applications, the reservation of resources must guarantee a constant rate of information delivery at destination sites. Therefore, a set of Quality of Service requirements or parameters must be satisfied when designing routing algorithms. There are several approaches to the subject depending on the optimization parameters considered each time. Examples of these parameters include the *packet loss probability*, the *available bandwidth*, the *delay jitter*, the *end-to-end delay* etc. Regardless though of the optimization parameters chosen, the main goal is to avoid congestion and perform smooth information transmission along the path for each connection.

Different types of routing have been the study subject of various research

proposals [3, 57, 14, 64, 44, 49] and citations therein. However, it has been shown by Wang et. al., [79] that, routing algorithms which attempt to optimize a specific QoS metric such as, message delay or routing distance do not necessarily enhance the overall performance of the network. The reason is simple : The routing algorithms care only to optimize a specific connection/session between source-destination pairs and only consider the performance of the network in the specific path chosen for that session. Therefore, routing strategies must be designed for optimization of *global QoS metrics* such as *average throughput rate*, *average acceptance message rate* in a site, *average routing distance*, *average buffer load* etc. Furthermore, these strategies should not compromise the overall availability of network resources but rather attempt to minimize the *routing cost*. Note that the cost of a route is a composite metric and it is affected not only by the routing distance but also by the state of all the links in the path. For example choosing minimum distance paths consisted of links with heavily loaded output buffers only minimizes the routing distance but, at the same time throughput, end-to-end delay rate and delay jitter are compromised.

The "rule of thumb" for an efficient routing algorithm is simple: "The more information is available about the state of the network the higher the probability is for the right choice of paths". Unfortunately, the amount of information that needs to be shared among network sites makes the routing algorithms unsuitable for their practical implementation. Therefore, a balance must be found between

the amount of global network state information needed and the computational complexity of these algorithms. Note also that, this balance must be independent of the size of the network, i.e., the algorithms must address the network scalability issue.

In this chapter we first describe the basic categories of routing algorithms and we comment on their advantages and disadvantages. Furthermore, we illustrate the basic concepts of QoS routing and we establish the objectives of a QoS routing algorithm. The rest of the chapter is divided into two major parts:

The first part corresponds to routing in the case of off-line applications and message transmissions. This routing is global static and without bandwidth guarantees for individual sessions. Relatively to this type of routing, we introduce three heuristics to achieve minimization of $V \times C_a^2 \times L_m^2$ for the whole network. As it has been shown in Chapter IV, this also achieves a reduction of the congestion probability and a satisfactory flow balancing for each network buffer. We analyze the performance of the three heuristics and we compute their order of complexity. We also find upper bounds for these heuristics for the ring topology. The analysis on the upper bounds of the heuristics in other topologies will be a future work.

In the second part of this chapter, we concentrate to the case of routing with bandwidth guarantees, (QoS routing). For this type of routing, we assume the existence of the *resource reservation protocol*, i.e., we only deal with the category of *Link State Routings* since, we assume the existence of a distributed dissemination

of link parameters in the network through flooding. For this type of protocol, we introduce a hybrid model of static and dynamically-distributed routing to achieve Quality of Service. This model uses the pre-computed global static paths for each source-destination pair as in the first part. The routing pre-computation is based on the three heuristics to minimize $V \times C_a^2 \times L_m^2$ as mentioned above. Furthermore, this pre-computation is repeated based on the congestion-detection algorithm introduced in Chapter IV. In the case that bandwidth guarantees are not satisfied for specific sessions, a distributed link state algorithm is triggered to discover the "least cost path" based on a "best effort mechanism". This hybrid model achieves a considerable amount of computation avoidance for the case of QoS routings as opposed to a purely dynamic and distributed protocol such as, the *Open Shortest Path First* (OSPF) protocol [80, 63].

Background

In general, there are two types of routing, *static* and *dynamic*. In static routings, the paths are precomputed and don't change during network transmissions unless the rerouting algorithm is triggered. In the case of dynamic routings, we have to compute the path for each flow initiated in the network. The dynamic routing is depended on updated network state information and therefore, it is more accurate. On the other hand, its implementation is slow and more difficult because of the "path loop problem".

Relatively to dynamic routing, there are two basic categories: *source routing* and *distributed hop-by-hop routing*. The difference between the two categories resides on where the path is computed. In the case of source routing, only the source computes the path of the specific transmission. In the case of the hop-by-hop routing, this computation is distributed among the network sites which participate in the path, i.e., the flows are forwarded to each side which in turn decides the next hop based on local information.

QoS routing

To achieve QoS routing, we first have to establish concrete definitions of QoS metrics. A general categorization of QoS metrics includes three classes, [44]:

Definition 18 : Let $P_{i,j} = \{s_i, s_{i_1}, s_{i_2}, \cdots, s_{i_m}, s_j\}$ be a path connecting the sites s_i and s_j using a routing scheme R. Let also $q(s_k, s_l)$ be a QoS metric measured at the link (s_k, s_l). Then q is called:

1. *Additive* if $q(P_{i,j}) = q(s_i, s_{i_1}) + q(s_{i_1}, s_{i_2}) + \cdots + q(s_m, s_j)$.
2. *Multiplicative* if $q(P_{i,j}) = q(s_i, s_{i_1}) * q(s_{i_1}, s_{i_2}) * \cdots * q(s_m, s_j)$.
3. *Concave* if $q(P_{i,j}) = min\{q(s_i, s_{i_1}), q(s_{i_1}, s_{i_2}), \cdots, q(s_m, s_j)\}$.

Examples of additive QoS metrics are the delay jitter, the routing distance, the hop-count etc. Examples of multiplicative metrics include the network reliability in which $0 \leq q(P_{i,j}) \leq 1$. Finally, an example of a concave type is the bandwidth which means that, the available bandwidth of a path equals with the minimum of

all available bandwidths among all links in the path.

A routing algorithm however is not enough to determine the network routing. Another essential factor is the *Routing Protocol* used for the routing algorithm implementation. A routing protocol must be used in every router to achieve the complete awareness of the network topology and the network state at any time. In general, every protocol must be *dynamic* and *distributed*. The term "dynamic" corresponds to the adaptive awareness of every router of the network topology in time. On the other hand, the term "distributed" corresponds to the techniques used, (mostly flooding) to implement this router awareness, i.e., all routers contribute to the network state information sharing.

Objectives of QoS-based routing and popular QoS routing algorithms

Most of the protocols in today's technology such as, *Open Shortest Path First* (*OSPF*), *Routing Information Protocol*, (*RIP*), and *Border Gateway Protocol*, (*BGP*) fall into the category of *Best Effort Protocols*. Unfortunately, the Best Effort Service Algorithms always choose the minimum cost path (where the minimum cost is defined as the composite cost of all QoS metrics). This service tends to always make routing changes whenever a "better" path is found even in the case that, current paths meet all the traffic requirements. The result is that, Best Effort Algorithms can cause an increase of the variance, of the delay jitter and of the throughput rate. Furthermore, these algorithms also charge the

network with additional communication overhead.

For a QoS algorithm to be efficient, it must meet certain objectives that increase the overall network performance. Jain in his survey [44] provides a set of three objectives that a QoS routing algorithm must achieve. We add to that list two more objectives which we believe are essential to this issue. These objectives are as follows:

The QoS routing algorithm must:

1. Meet the QoS requirements for end users.
2. Optimize the network resource usage.
3. Gracefully degrade network performance when congestion occurs, i.e. it is expected to perform better than other algorithms which dramatically degrade the performance of the network.
4. Be designed to use as little as possible global state information.
5. Be able to optimize multiple QoS metrics.

The objective of the heuristics which we are going to introduce in this chapter is the minimization of $V \times C_a^2 \times L_m^2$. However, this minimization affects indirectly all the above QoS criteria. As we have proved in Chapter IV, the minimization of $V \times C_a^2 \times L_m^2$ reduces the probability of congestion in the network. At the same time, the load balancing among all links allows a fair distribution of network resources among all the pairwise communications. Furthermore, in the case of a congested link, the packet drop policy used distributes fairly the data trans-

mission slow-down among all communications passing through that link. Therefore, the degradation of the system is graceful and appears fair among all the users affected. Moreover, the network state information that needs to be communicated is only the site congestion vector for each site. This minimum communication is an advantage of our model since, it also reduces the communication overhead.

The following table shows the most popular QoS routings existed in the literature. Note that all these algorithms are unicast and they try to optimize one or multiple QoS criteria. This table is part of the table from [44] who published a thorough survey paper in QoS routings:

Table 2

Unicast QoS Routing Algorithms

Algorithm	*QoS-Constraints*	*Routing Strategy*	*Complexity*
Wang-Crowcroft [78]	Bandwidth-Delay	Source Routing	$O(vlogv + e)$
Ma-Steenkiste [57]	Multi-constraint	Source Routing	$O(kve)$
Guerin-Orda [3]	Bandwidth	Source	$O(vlogv + e)$
Salama et. al.[69]	Delay-Cost	Distributed	$O(v^3)$
Chen [14]	Generic	Distributed	$O(e)$

Off-line routing without bandwidth guarantees

In this section we concentrate on routings that do not guarantee bandwidth reservation. There is a large repertoire of network applications that fall into his category. Typical examples include Remote Procedure Call (RPC) network applications, Web browsing, I/O intensive scientific computations, etc. As

opposed to on-line applications where, a continuous rate traffic is needed, the key performance factor here is the fair share of resources among all connections in the network. Our approach does not specialize in individual connections. Instead, the best effort traffic balancing is achieved at a coarser level scale. More specifically, the routing mechanism directs traffic along less congested paths to balance the network load. Referring however to individual connections, it is the *congestion control* that dynamically adjusts the source transmission rate so that, the connection only receives the available bandwidth at specific paths. The rerouting mechanisms along with the congestion control makes it possible to provide the network with a load-sensitive approach for distributing information across.

We will use a global routing scheme where every site runs the same algorithm when congestion is detected. The information needed as input to the algorithm is the amount of traffic generated for each source-destination pair. This information is distributed across the network sites with flooding. Every site after running the algorithm updates only its local routing table relatively to source-destination connections that pass through that site. Traffic to the same destination sites and from different sources that pass through an intermediate site do not necessarily follow the same path. Therefore, the routing table that corresponds to each site is a sparse two-dimensional array as it has been illustrated in Chapter IV.

The heuristics introduced in this chapter are based on the minimization of

$V \times C_a^2 \times L_m^2$. Using the $V \times C_a^2 \times L_m^2$ as a cost function for a routing scheme is unique. Most of the routing algorithms used in today's technology try to minimize the number of hops for paths or the end-to-end delay. A more global approach also has been studied by [58] where the cost function was the *max-min fair rate*, which is a more global approach and similar to ours. However, the min-max fair rate approach unfortunately prefers longer paths to achieve flow balance. Our approach on the other hand restricts the routing to as small as possible paths since, the average communication cost needs also to be minimized. Our heuristics are based on the greedy computational method and they are:

1. The *Average Dijkstra Heuristic*
2. The *Average Floyd-Warshall Heuristic* and
3. The *Hierarchical Heuristic.*

In the following sub-sections we illustrate each heuristic algorithmically and we compute their order of computational complexity. Furthermore, we find upper bounds for these algorithms for the case of ring topologies.

The average Dijkstra heuristic

Given the set of frequencies of communication F_e in the network, the goal is a routing scheme that balances the buffer loads while minimizing at the same time the average communication cost C_a and the maximum buffer load L_m. Assuming that every source-destination path is a minimum-hop path, then the total

communication cost divided by the number of available buffers gives the minimum average communication cost (C_a^{MIN}) for the network topology and the set F_e.

The ideal scenario is the application of the min-hop paths to also minimize the variance on the frequencies of use for the buffers. Even though this is almost impossible and solely depends on the distribution of the numbers in F_e, it gives a lower bound on the buffer frequencies of use since, $\lceil C_a^{MIN} \rceil$ is the minimum possible value that L_m may get. In the case that there is a frequency with value greater than $\lceil C_a^{MIN} \rceil$, then L_m is set to that value. This idea is the key point for the average Dijkstra heuristic.

The algorithm first computes the minimum possible average communication cost with a call to Floyd-Warshall algorithm assuming that all loads in the network are equal to 0. The Floyd-Warshall algorithm, [16], finds the minimum cost paths between all source-destination pairs in the network. Moreover, this algorithm uses two 3-dimensional arrays to store the distances between nodes and the paths found among all pairs. Let $w_{i,j}$ be the current link load for each directed link (i, j) in the network. We first show the initialization of the distance array D

$$D_{i,j}^0 = \begin{cases} 0 & i = j \\ w_{i,j} & i \neq j \text{ and connected} \\ \infty & i, j \text{ not connected} \end{cases}$$

and the path array Π in that algorithm

$$\Pi_{i,j}^{0} = \begin{cases} NIL & \text{if } i = j \text{ or } w_{i,j} = \infty \\ i & i \neq j \text{ and } w_{i,j} < \infty \end{cases}$$

and then we illustrate the algorithm in Figure 26.

```
Floyd_Warshall_Algorithm(Input:G = (S, E, W_e))
{
    /* Uses an 3-dimensional array Π^{|S|}_{i,j} of size |S| to store
    the paths and a 3-dimensional array of distances D
    to store distances between sites */

    n ← |S|
    Initialize D^0_{i,j} and Π^0_{i,j}
    FOR k ← 1 to n
        FOR i ← 1 to n
            FOR j ← 1 to n
                IF D^{k-1}_{i,j} ≤ D^{k-1}_{i,k} + D^{k-1}_{k,j}
                    D^k_{i,j} ← D^{k-1}_{i,j}
                    Π^k_{i,j} ← Π^{k-1}_{i,j}
                ELSE
                    D^k_{i,j} ← D^{k-1}_{i,k} + D^{k-1}_{k,j}
                    Π^k_{i,j} ← Π^{k-1}_{k,j}
                END-IF
            END-FOR
        END-FOR
    END-FOR
}
```

Figure 26. Floyd-Warshal algorithm.

The routing algorithm then sorts the set F_e is descending order for the

reason that the insertion of frequencies in the routing scheme follows that order.

The next step in the algorithm is the insertion of communication pairs in the routing scheme starting from the largest possible $f_{i,j}$'s and using the paths found by the call to Warshall. The algorithm keeps inserting frequencies as long as the present maximum buffer load is less than or equal to the present L_m bound which is equal to $\lceil C_a^{MIN} \rceil$. If the insertion of a frequency makes the present maximum buffer load greater, then this insertion is delayed.

When no more frequencies can be inserted without exceeding the maximum buffer load bound, a new bound for L_m is chosen. This is succeeded by inserting the maximum in value frequency of the remaining frequencies. In that case, L_m will be increased. The path that is chosen for that frequency is the one that minimizes this increase.

The process of insertion of the remaining frequencies is repeated for the new bound. If there are still uninserted frequencies, a new bound for the maximum buffer load is set using the same process as above.

The insertion of each frequency in the routing scheme uses the path returned by a call to the procedure *Find_Minimum_Loaded_Path()* which is illustrated in Figure 27. This procedure first increases all the buffer loads in the network by the $f_{i,j}$ value at hand. Then, all the buffers with load greater than the current maximum load are excluded by setting their load to ∞. What follows is a call to the *Dijkstra* procedure, [16], which finds the minimum cost path for the

```
Procedure :  Find_Minimum_Loaded_Path(G = (S, E), s, d, f_{s,d})
{
    FOR i ← 1 to number_of_sites
        FOR j ← 1 to number_of_sites
            IF e = (s_i, s_j) ∈ E
                u_{i,j} ← u_{i,j} + f_{s,d}
                IF u_{i,j} > L_m bound
                    u_{i,j} ← ∞
                END-IF
            END-IF
        END-FOR
    END-FOR
    Call Dikjstra(G=(S,E), source, dest)
}
```

Figure 27. Procedure to compute the minimum loaded path.

current frequency and the current network state. Finally, all buffers which have not participated in the path for $f_{i,j}$ are set back to their original load values. If a path cannot be established, then we delay the insertion of $f_{i,j}$.

In Figure 28 we illustrate the average Dijkstra heuristic and in Theorem 8 we prove that its computational complexity for a regular network of n sites and degree d is $O(\frac{n^6}{d})$.

Theorem 8 : The average Dijkstra heuristic has order of complexity $O(\frac{n^6}{d})$ for regular networks of n sites and degree d.

Proof : Let $G = (S, E, F_e)$ be a regular network of $|S| = n$ sites, degree d and communication frequency set F_e, with $|F_e| = n(n-1)$. Note that, the first three

```
Average_Dijkstra_Algorithm(Input:F_e, n)
{
    Step 1 :  Call Floyd_Warshall_Algorithm
              Compute C_a^MIN
              L_m ← max{max f_{i,j}, ⌈C_a^MIN⌉}
    Step 2 :  Sort the set F_e in descending order
    Step 3 :  Set all buffer loads to 0
    Step 4 :  WHILE ∃ f_{i,j}'s and Min_Possible_Lm holds
                  FOR i ← 1 to |F_e| of remaining f_{i,j}'s
                       call Find_Minimum_Loaded_Path for f_{i,j}
                       IF ∃ a path for the f_{i,j}
                           Update routing table and buffer loads
                       END-IF
                  END-FOR
              END-WHILE
    Step 5 :  IF ∃ f_{i,j} not inserted yet
                  find the max f_{i,j} (not inserted)
                  call Find_Minimum_Loaded_Path for f_{i,j}
                  Update routing table and buffer loads
                  Update L_m bound
                  GOTO Step 4
              ELSE
                  Exit
              END-IF
}
```

Figure 28. The average Dijkstra routing algorithm.

steps of the algorithm are sequential.

Step 1 is a call to Floyd-Warshall procedure which takes $O(n^3)$ time as it has been proven in [16].

The second step of the algorithm sorts the set F_e in descending order.

Therefore, this step has complexity $O(n^2 logn)$ since, there are $n(n-1)$ $f_{i,j}$'s in the input set.

Step 3 takes in the worst case of complete networks time $O(n^2)$.

For the steps 4 and 5 we will compute the compound cost since, they run in conjunction. The WHILE-loop in step 4 will be executed at least one time. However, the total number of executions of the loop is equal to the number of executions of step 5 plus one. This number refers to the number of times we force the insertion of an $f_{i,j}$ in the routing scheme and increase the present L_m bound. Therefore, we will concentrate on the $f_{i,j}$ distributions that maximize the number of these forced insertions.

Note that even though paths are assigned to frequencies through a call to Dijkstra's algorithm, the exclusion of certain links may force these paths to be of length $(n-1)$, i.e., the maximum length possible cycle-free paths. Note also that, the number of buffers in G is nd. If we assume in the worst case that, every frequency is assigned a path of length $(n-1)$, then the minimum number of frequencies that can be inserted each time step 4 is executed is $\frac{nd}{(n-1)}$. This is the minimum possible number of frequencies that can be inserted each time without increasing the present L_m bound. Therefore, for a total of $n(n-1)$ frequencies, step 4 will be executed $\lceil \frac{n(n-1)}{\frac{nd}{n-1}} \rceil = \lceil \frac{(n-1)^2}{d} \rceil$ times and step 5 $(\lceil \frac{(n-1)^2}{d} \rceil - 1)$ times.

To calculate the complexity of step 5, we observe that it contains a call to *Find_Minimum_Loaded_Path()*. This procedure takes $O(nd)$ time for updating

the buffers of G and $O(n^2)$ time for the call to Dijkstra thus, overall $O(n^2)$ time is spent. Moreover, the update of the routing table for each frequency takes $O(n)$ time and the update of the buffers after the insertion $O(n)$ time. Thus, each execution of step 5 takes $O(n^2)$ and there are $(\lceil \frac{(n-1)^2}{d} \rceil - 1)$ such executions in the worst case. Therefore, the total complexity of step 5 is $(\lceil \frac{(n-1)^2}{d} \rceil - 1) \times O(n^2) = O(\frac{n^4}{d})$.

Finally, step 4 contains a FOR-Loop on the uninserted frequencies after each execution of step 5. The body of the FOR-Loop contains a call to *Find_Minimum_Loaded_Path()* which takes $O(n^2)$ time and the update of the routing table and buffers which takes $O(n)$ time. Thus, we spend $O(n^2)$ time to execute the body of the FOR-Loop. However in the worst case, the FOR-Loop will be executed:

$$\begin{aligned} T(n) &= n(n-1) + \left(n(n-1) - \frac{nd}{(n-1)}\right) + \cdots + \left(n(n-1) \ \mathit{Mod} \ \frac{nd}{(n-1)}\right) \\ &\leq \left(\frac{n(n-1)}{\frac{nd}{(n-1)}} + 1\right) \times \left(n(n-1) - \frac{nd}{(n-1)}\right) \\ &= \left(\frac{(n-1)^2}{d} + 1\right) \times \left(n(n-1) - \frac{nd}{(n-1)}\right) \\ &= O(\frac{n^4}{d}) \text{ times} \end{aligned}$$

Thus, the complexity of step 4 is $O(n^2) \times O(\frac{n^4}{d}) = O(\frac{n^6}{d})$.

Therefore, the total complexity of the algorithm is the highest complexity among all steps and this is the complexity of step 4 which is $O(\frac{n^6}{d})$. Note that, the complexity of the above algorithm was computed based on the number of sites

in the network and not on the input data set count. However, the input count for the algorithm is $N = n(n-1)$ which makes our algorithm of complexity $O(\frac{N^3}{d})$.

□

The average Floyd-Warshall heuristic

The average Floyd-Warshall heuristic is similar to the average Dijkstra heuristic in the sense that, they both try to keep the maximum buffer load as small as possible while trying to balance the load in all buffers. The difference is in the process of finding paths for individual frequencies. The Dijksta heuristic finds one by one the paths for the uniserted frequencies without exceeding the present maximum buffer load. On the other hand, the Floyd-Warshall heuristic finds all the paths for the uniserted frequencies at once by a call to Floyd-Warshall procedure, [16]. However, only the frequencies for which the present L_m remains the same are inserted. The insertion of the remaining frequencies is delayed until a new bound for L_m is found.

More specifically, this heuristic executes the same first three steps as the average Dijkstra heuristic. After finding the value of C_a^{MIN}, the first bound for L_m is set to be the maximum between the maximum $f_{i,j}$ in F_e and $\lceil C_a^{MIN} \rceil$. The paths found by the first call to Floyd-Warshall are used for the insertion of frequencies without exceeding this bound.

The frequencies are inserted in decreasing order of their value. In the case

that, a frequency insertion causes the maximum buffer load to exceed the present bound, this insertion is delayed. If no more frequencies can be inserted in the routing scheme keeping the current maximum buffer load, then the maximum in value of the remaining frequencies is forced an insertion. This frequency increases the current maximum buffer load. However, the path chosen for that frequency is the one that minimizes this increase.

The process for the uniserted frequencies is then repeated until all of them are inserted in the routing scheme or a new bound for L_m is set. In Figure 29 we illustrate the average Floyd-Warshall heuristic and in Theorem 9 we prove its computational complexity.

Theorem 9 : The average Floyd Warshall heuristic for a regular network $G = (S, E, F_e)$ of $|S| = n$ sites and degree d has order of complexity $O(\frac{n^5}{d})$.

Proof : The first three steps of the algorithm are sequential and the same as in the average Dijkstra heuristic. In Theorem 8, we proved that step 1 takes $O(n^3)$ time, step 2 $O(n^2 longn)$ time and step 3 $O(n^2)$ time.

We will compute the complexity of the steps 4 and 5. The worst case for the algorithm occurs when, the number of executions of step 5 is maximized. This is the case when, all frequencies are assigned paths of length $(n-1)$. Hence, for a network of nd buffers, we can insert at least $\frac{nd}{(n-1)}$ frequencies without exceeding the present maximum buffer load. Thus as we proved in Theorem 8, the maximum number of executions of step 5 is $(\lceil \frac{(n-1)^2}{d} \rceil + 1)$ and in that case, step 4 executes

```
Average_Floyd_Warshall_Algorithm(Input:F_e, n)
{
    Step 1 :  Call Floyd_Warshal_Algorithm
              Compute C_a^MIN
              L_m ← max{max f_{i,j}, ⌈C_a^MIN⌉}
    Step 2 :  Sort the set F_e in descending order
    Step 3 :  Set all buffer loads to 0
    Step 4 :  WHILE ∃ f_{i,j}'s and L_m holds
                 FOR i ← 1 to |F_e| of remaining f_{i,j}'s
                      consider the path from Floyd Warshall
                      compute L_m considering f_{i,j}
                      IF L_m still holds
                          Update the routing table and buffer loads
                      END-IF
                 END-FOR
              END-WHILE
    Step 5 :  IF ∃ f_{i,j} not inserted yet
                 Find the max f_{i,j} (not inserted)
                 Find_Minimum_Loaded_Path() for f_{i,j}
                 Update the routing table and buffer loads
                 Update L_m
                 Call Floyd_Warshal_Algorithm
                 GOTO Step 4
              ELSE
                 Exit
              END-IF
}
```

Figure 29. Average Floyd-Warshall routing algorithm.

$\lceil \frac{(n-1)^2}{d} \rceil$ times.

Step 5 contains a call to *Find_Minimum_Loaded_Path()* which is proved in Theorem 8 to be of complexity $O(n^2)$. The update of the routing table and the associated buffers takes $O(n)$ time. Step 5 contains also a call to Floyd-Warshall procedure which takes $O(n^3)$ time. Therefore, the largest in complexity operation of step 5 takes $O(n^3)$ time. Thus, the total complexity of step 5 is $(\lceil \frac{(n-1)^2}{d} \rceil - 1) \times O(n^3) = O(\frac{n^5}{d})$.

Step 4 is executed $\lceil \frac{(n-1)^2}{d} \rceil$ times. However, this step executes a FOR-Loop on the remaining frequencies each time. In Theorem 8 we computed the total executions of the body of this FOR-Loop to be $O(\frac{n^4}{d})$ times. Furthermore, for each frequency under consideration in the FOR-Loop, we check if the maximum buffer load exceeds the present bound for L_m. This operation takes $O(n)$ time since, we only consider at most $(n-1)$ buffers for each frequency path. The update of the routing table and the associated buffers takes $O(n)$ time also. Therefore, the complexity of step 4 is $\lceil \frac{(n-1)^2}{d} \rceil \times O(n) = O(\frac{n^5}{d})$.

Note that the largest complexity among all five steps is $O(\frac{n^5}{d})$ which is also the complexity of the algorithm. As we also mentioned in Theorem 8, this complexity computation is based on the number of network sites n and not on the input data count of the set F_e which is $N = n(n-1)$. For this input count, the complexity of the algorithm becomes $(\frac{N^2\sqrt{N}}{d})$. □

The Hierarchical routing heuristic

In the two previous heuristics, the priority in the insertion of frequencies into the routing scheme was given to large frequencies. This leaves small frequencies to the end of the insertion process therefore, it reduces the probability to have large variance.

In this heuristic, we use the path length as a criterion for the insertion of frequencies regardless of their value. More specifically, we impose a hierarchy on the frequencies relatively to their minimum cost paths associated with them. We try to insert frequencies in the routing scheme starting from those with the smallest paths. Larger paths are also considered but after we examine the insertion of small paths. However, the balancing of buffer loads is also important. This is done by setting bounds on the maximum buffer load as in the previous two heuristics. This heuristic reduces the probability of assigning large paths to frequencies therefore, it also affects the reduction of the average communication cost.

The algorithm first initializes the buffers of the network and then, it inserts all $f_{i,j}$'s that can be inserted with paths of length 1. These frequencies are known to each site since, all sites are aware of the network's topology. We also sort the rest of the remaining frequencies in a decreasing order of their values since, it helps us to find the current maximum frequency later on.

In the next step we have a call to the Floyd-Warshall procedure for the

remaining frequencies on the existed weighted network resulted from the previous step. Note that, the call to Floyd-Warshall finds the minimum cost paths for the remaining frequencies. Therefore, the minimum possible average communication cost C_a^{MIN} can be computed also. We set the first bound for the present maximum buffer load L_m to be $\lceil C_a^{MIN} \rceil$ since this is the least possible L_m we can have. In the case that, there is a frequency with value greater than $\lceil C_a^{MIN} \rceil$, then L_m is set to that value. We also update the path array used and we sort it in ascending order of path lengths.

The next step of the algorithm inserts frequencies to the routing scheme using the paths found from Floyd-Warshall. The frequency is inserted only if the loads of the participating buffers remains less than or equal to the present maximum buffer load bound. Furthermore, the ordering used for the insertion of frequencies is based on the path lengths. First, frequencies with path length 2 are considered, then with path length 3 etc. If a frequency insertion does not preserve the current L_m bound it is skipped for later.

When no more frequencies can be inserted without exceeding the current L_m bound, we find the largest in value of the remaining frequencies and we force its insertion by finding the minimum loaded path for it. This process is exactly the same as in the Floyd-Warshall heuristic and it sets a new bound for L_m. We then repeat a call to Floyd-Warshall on the current weighted network for the remaining frequencies.

We repeat the process of frequency insertion based on the new path lengths and the new L_m bound until, all frequencies are inserted or a new bound for L_m is found with a forced insertion. Figure 30 illustrates the Hierarchical Routing Algorithm and Theorem 10 proves its computational complexity.

Theorem 10 : Let $G(S, E, F_e)$ be a regular network of $|S| = n$ sites, degree d and communication frequency set F_e with $|F_e| = n(n-1)$. Then, the computational complexity of the hierarchical routing heuristic is $O(n^5)$.

Proof : Note that, the first 7 steps of the algorithm are sequential. However steps 8 and 9 run in conjunction. We first find the complexities of the first 7 steps and then analyze the worst case execution of steps 8 and 9.

Step 1 takes $O(nd)$ time which is equal with the number of buffers in the topology. Step 2 is also of complexity $O(nd)$ since, this is the largest number of frequencies that can be inserted in the routing scheme with path length one. Step 3 is of complexity $O(nd)$ for the same reason as in step 1. Step 4 sorts the remaining uninserted frequencies in descending order. There are $(n(n-1) - nd)$ frequencies remaining. Thus, the complexity of this step is $O(n^2 logn)$. In step 5, we have a call to Floyd-Warshall which takes time $O(n^3)$. Also, the computation of the minimum average communication cost takes $O(nd)$ time since, it involves the loads of all buffers. Step 6 uses an auxiliary array holding the lengths of the paths found in step 5. Therefore, its sorting will take less than $O(n^2 logn)$ time. On the other hand, step 7 takes constant time. By comparing the time

```
Hierarchical Routing Algorithm(Input:F_e, n)
{
    Step 1 :  Initialize buffer loads to 0
    Step 2 :  INSERT all f_{i,j}'s with path_length = 1 in the R.T
    Step 3 :  Update current buffer loads
    Step 4 :  Sort the remaining frequencies in ascending order
    Step 5 :  Call Floyd_Warshall and compute C_a^MIN
    Step 6 :  Update Path array and sort the paths
    Step 7 :  L_m ← max{max f_{i,j}, ⌈C_a^MIN⌉}
    Step 8 :  WHILE max_buffer_load ≤ L_max
                  current_path_length ← 2
                  FOR i = 1 to |F_e| in the sorted list
                      IF Length(p_{i,j}) = current_path_length
                          compute max_buffer_load for f_{i,j}
                          IF max_buffer_load ≤ L_max
                              INSERT p_{i,j} in the R.T.
                              Update buffer loads
                          END-IF
                      END-IF
                  END-FOR
                  current_path_length = current_path_length + 1
              END-WHILE
    Step 9 :  IF ∃ f_{i,j} not inserted yet
                  Find the maximum of the remaining frequencies
                  Call Find_Minimum_Loaded_Path()
                  Update the Routing Table and the buffers
                  L_max ← max_buffer_load
                  Call Floyd_Warshall
                  Update Path array and sort the paths
                  GOTO Step 8
              ELSE
                  Exit
              END-IF
}
```

Figure 30. Hierarchical routing algorithm.

complexities of the first seven steps of the heuristic, we see that, the most costly is step 5.

As we mentioned above, steps 8 and 9 run in conjunction. The worst case for the algorithm occurs when, all the insertions of the remaining frequencies are forced to happen in step 9. In that case, the number of executions of step 9 will be $(n(n-1)-nd)$. However, step 9 contains:

1. A call to find the maximum in value frequency which is found in $O(n^2)$ time in the worst case,

2. A call to Find_Minimum_Loaded_Path() which is proved to be of complexity $O(n^2)$,

3. The update of the routing table and the buffers of the topology which takes $O(n)$ time for a single frequency,

4. A call to Floyd-Warshall which is of complexity $O(n^3)$ and

5. A call to sort the auxiliary array of the new paths found which takes $O(n^2 logn)$ time is the worst case.

Note that, the most expensive operation is of complexity $O(n^3)$. Thus, the total complexity of step 9 is $(n(n-1)-nd) \times O(n^3) = O(n^5)$.

Step 8 on the other hand has to be executed regardless of the fact that, not a single frequency is inserted throughout its execution. Note that, this step will also be executed $(n(n-1)-nd)$ times. However, this step will examine the first time $(n(n-1)-nd)$ frequencies the second time $(n(n-1)-nd-1)$ frequencies,

etc. Thus, the total number of frequency examinations will be :

$$\begin{aligned} T(n) &= (n(n-1)-nd)+(n(n-1)-nd-1)+\cdots+2+1 \\ &= \frac{(n(n-1)-nd)\times(n(n-1)-nd+1)}{2} \\ &= O(n^4) \text{ times} \end{aligned}$$

Furthermore, for each frequency examination, we apply the path suggested by Floyd-Warshall's algorithm which takes $O(n)$ time and check if the maximum buffer load bound still holds which also takes the same time. Thus, the total complexity of step 8 is $O(n^5)$.

The total time spent for the execution of the algorithm is the summation of times spent for each one of its steps. However, steps 8 and 9 are the most expensive and they are both of complexity $O(n^5)$. Therefore, the algorithm has complexity $O(n^5)$.

As we also mentioned in Theorems 8 and 9, this complexity computation is based on the number of network sites n and not on the input data count of the set F_e which is $N = n(n-1)$. For this input count the complexity of the algorithm becomes $(N^2\sqrt{N})$. □

Upper bounds for the heuristics in the ring topology

Since our optimization problem at hand has been proven to be NP Complete, we know that a polynomial optimization algorithm cannot not be found.

Therefore, we suggested three approximation algorithms that run in polynomial time. The question "How close the solutions of the heuristics are to the optimal solution ?" though still remains.

In this section, we focus on the performance of the three heuristics for the ring topologies. We show that, there are communication frequency distributions for which, all three heuristics are unbounded when compared to the optimal solution for minimizing $V \times C_a^2 \times L_m^2$.

For the proof of our claims, we will assume instances I of rings $G = (S, E, F_e)$ and we compare the solutions of the three heuristics relatively to the optimal solution. We leave the upper bound findings of the heuristics for other regular topologies as a future research. Throughout this section we call $OPT(I)$, $D(I)$, $F(I)$ and $H(I)$ to be the optimal, average Dijkstra, average Floyd-Warshall and Hierarchical solutions, respectively.

Theorem 11 proves that there are F_e distributions for which there is no upper bound for $D(I), F(I)$ and $H(I)$.

Theorem 11 : Let $G = (S, E, F_e)$ be a ring and $OPT(I), D(I), F(I)$, $H(I)$ the solutions for the routing of the set F_e in G. Let also V_{OPT}, V_D, V_F and V_H be the corresponding variances of the ring buffer loads resulted by the above algorithms. Then, $\exists$ an F_e distribution for which, $V_{OPT} = 0$ but $V_H, V_D, V_F \neq 0$ therefore indicating that, there is no upper bound for $D(I), F(I)$ and $H(I)$.

Proof : An example of such an F_e distribution suffices for the proof of our claim.

Let for the ring G, $|S| = n = 2m,\ \ m \geq 3$, and F_e be defined as follows:

$$F_e = \begin{cases} f_{n,2} = c \\ f_{2,n} = c \\ f_{\frac{n}{2}-1,\frac{n}{2}+1} = c \\ f_{\frac{n}{2}+1,\frac{n}{2}-1} = c \\ f_{i,j} = 0 \ \forall \ (i,j) \neq \{(n,2),(2,n),(\frac{n}{2}-1,\frac{n}{2}+1),(\frac{n}{2}+1,\frac{n}{2}-1)\} \end{cases}$$

Figure 31(a) illustrates the solution of $OPT(I)$. $D(I)$, $F(I)$ and $H(I)$ result in the same routing scheme which is illustrated in Figure 31(b). Note that, $OPT(I)$ results into $V \times C_a^2 \times L_m^2 = 0$ due to the fact that $V_{OPT} = 0$. This is because $OPT(I)$ prefers paths of length $(n-2)$ for the two out of the four frequencies and charges all the output buffers with load c.

On the other hand, $D(I)$, $F(I)$, and $H(I)$ first set the L_m bound to be equal with c. All four frequencies are inserted with paths of length 2 since, these are the smallest in length paths and they are non-overlapping. Thus, $V_{OPT}, V_D, V_F > 0$, $L_m = c$ and $C_a > 0$ for all three heuristics. Therefore, $D(I)$, $F(I)$, and $H(I)$ have no upper bounds. □

The counter example used in Theorem 11 is sufficient to prove the theoretical result that, all three heuristics are unbounded. However, these distributions correspond to a small percentage of the F_e distribution set. We are interested to compare the solutions of the three heuristics with the optimal solution in the case

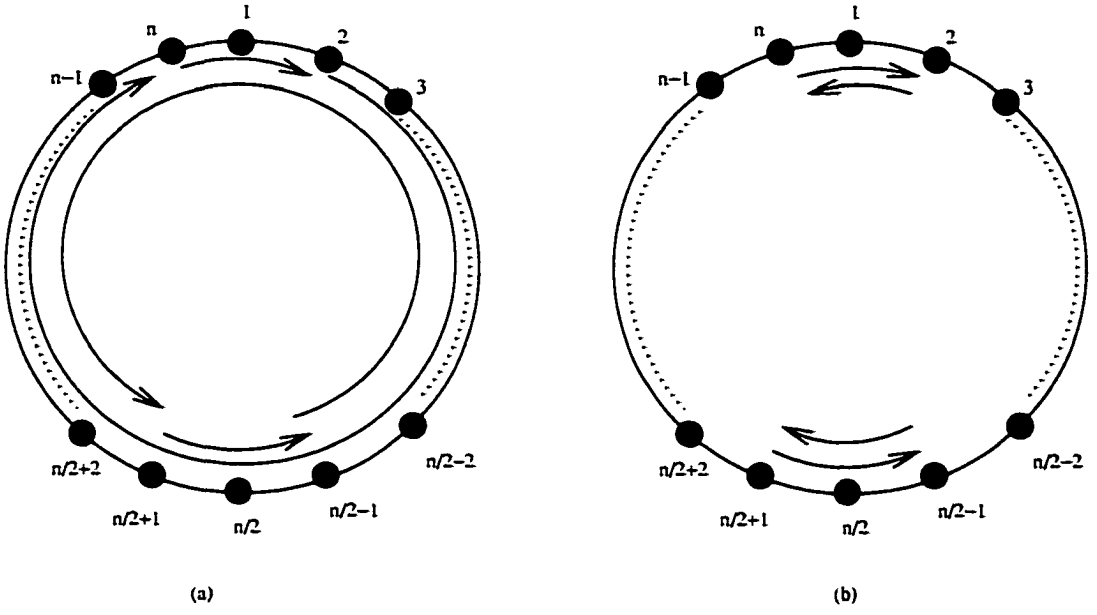

Figure 31. Verification example for the unboundedness of the three heuristics.

that, the variance of the buffer loads is not equal to zero. This scenario is the most realistic one and it covers the vast majority of frequency distributions.

We classify our analysis in two subsections. The first subsection compares the Dijkstra and the Floyd-Warshall heuristic with the optimal solution. The second subsection does the same comparison but for the case of the hierarchical heuristic. The reason that the Dijkstra and the Floyd-Warshall heuristics are discussed simultaneously is that, they have the same performance when the worst case distribution is encountered. This is due to the arethmitization of sites imposed by both the Dijkstra and the Floyd-Warshall algorithm in computing the distance array of path costs and also due to the fact that, frequencies are examined

in a decreasing order of their value. .

Upper bounds for the average Dijkstra and Floyd-Warshall heuristics

For an input instance I of rings $G = (S, E, F_e)$, let $OPT(I)$, $D(I)$ and $F(I)$ be the solutions of the optimal, Dijkstra heuristic and Floyd-Warshall heuristic respectively. We will concentrate our discussion into F_e distributions which, regardless of the routing paths assigned to $f_{i,j}$'s, they always result into a variance $V \neq 0$. For this case, we want to compare the performance of the $D(I)$ and $F(I)$ relatively to $OPT(I)$.

The worst case F_e distribution for $D(I)$ and $F(I)$ is the one that maximizes the variance of buffer loads and the maximum buffer load. Note that, the average communication cost C_a is indirectly affected since, it grows at a slower rate than V and L_m. Assume a set of frequencies that concentrates high traffic in all links adjacent to a single site where at the same time, it forces the overlapping of routing paths that increase the maximum buffer load. We construct the worst case F_e distribution to defeat the advantage of $D(I)$ and $F(I)$ of inserting larger frequencies first.

More specifically, let $G = (S, E, F_e)$ be a ring of $|S| = n$ sites and $|E| = n$ half duplex links with $2n$ output associated buffers. Without loss of generality,

let $n = 2m,\ m \in Z^+$. Let also the set F_e be defined as follows:

$$F_e = \begin{cases} f_{1,\frac{n}{2}} = f \\ f_{\frac{n}{2},1} = f \\ f_{2,\frac{n}{2}+1} = f - 1 \\ f_{i,j} = 0 \ \forall\ (i,j) \neq \{(1,\frac{n}{2}),(\frac{n}{2},1),(2,\frac{n}{2}+1)\} \end{cases}$$

We call this instance I_W since, this is the worst case distribution for the two heuristics at hand. The reason is that, the frequencies $f_{1,\frac{n}{2}}$ and $f_{\frac{n}{2},1}$ are the maximum frequencies in F_e and therefore, $D(I)$ and $F(I)$ routes them first. Both heuristics will route $f_{1,\frac{n}{2}}$ and $f_{\frac{n}{2},1}$ with path length $(\frac{n}{2}-1)$ since, these paths do not overlap. However, the insertion of $f_{2,\frac{n}{2}+1}$ forces an overlapping on the buffers $\{(2,3),(3,4),\cdots,(\frac{n}{2}-2,\frac{n}{2}-1)\}$. Figure 32(a) shows the routing scheme resulted by $D(I)$ and $F(I)$.

On the other hand, $OPT(I)$ routes $f_{\frac{n}{2},1}$ and $f_{2,\frac{n}{2}+1}$ using opposite direction paths. That causes L_m to be reduced to f which is the minimum possible. It also achieves the best possible balance on the buffers and this happens with a minimum increase on the average communication cost. The routing resulted by $OPT(I)$ is depicted in Figure 32(b).

In the following lemma, we prove that, $\frac{D(I_W)}{OPT(I_W)}$ is a monotonically increasing function for any ring topology of n sites and any value f of the maximum frequency in I_w upper bounded by the value of $\frac{44}{3}$. The same is true for the $\frac{F(I_W)}{OPT(I_W)}$ ratio.

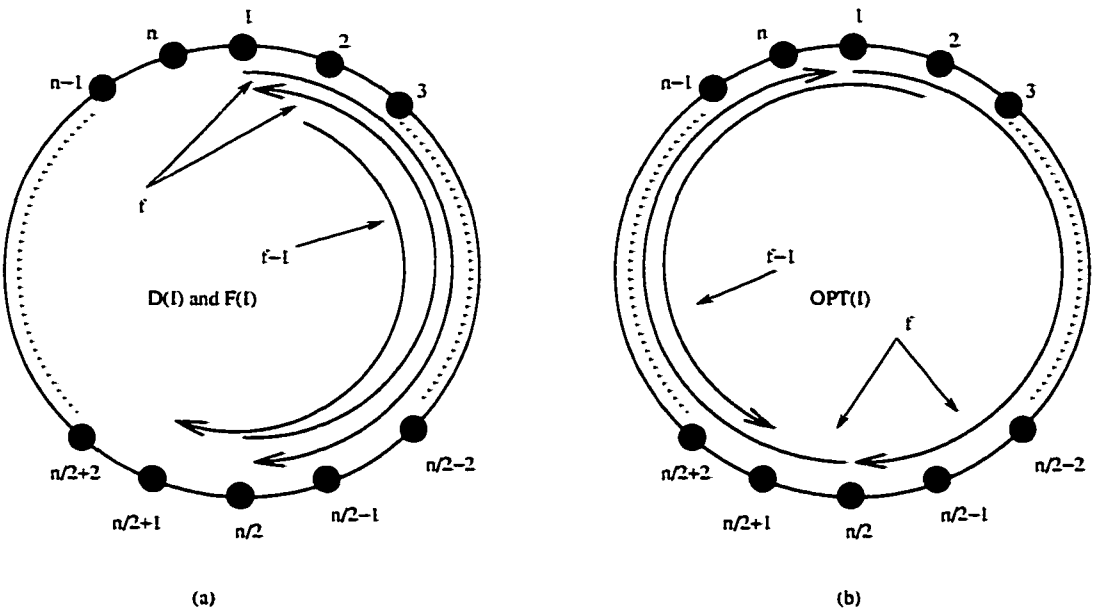

Figure 32. Worst case F_e distribution routing for $D(I)$, $F(I)$ and $OPT(I)$.

Lemma 16 : For the worst case F_e distribution instance I_W, the ratios $\frac{D(I_W)}{OPT(I_W)}$ and $\frac{F(I_W)}{OPT(I_W)}$ are monotonically increasing functions on n and f, bounded by an upper bound of value $\frac{44}{3}$.

Proof : Since $D(I)$ and $F(I)$ result into the same product $V \times C_a^2 \times L_m^2$, it suffices to prove our claim for the ratio $\frac{D(I_W)}{OPT(I_W)}$. We will show that

$$\lim_{n \to \infty,\, f \to \infty} \frac{D(I_W)}{OPT(I_W)} = \frac{44}{3}$$

Without loss of generality, let $n = 2m$ with, $m \in Z^+$ be the number of sites in the ring. Let L_m^D, C_a^D and V^D be the maximum buffer load, average communication cost and buffer variance resulted by the solution $D(I_W)$. Moreover,

let L_m^O, C_a^O and V^O be the corresponding quantities for the optimal solution.

The routing of $D(I)$ results into three paths of length $(\frac{n}{2}-1)$. Out of the $2n$ network buffers, $(\frac{n}{2}-2)$ are charged with load $(2f-1)$, $(\frac{n}{2}+1)$ are charged with load f and the remaining $(n+1)$ buffers have zero load. Then,

$$
\begin{aligned}
L_m^D &= 2f-1 \\
C_a^D &= \frac{(\frac{n}{2}-2)(2f-1)+(\frac{n}{2}+1)f}{2n} = \frac{3nf-6f-n+4}{4n} \quad \text{and} \\
V^D &= \frac{1}{2n}\Big[(\frac{n}{2}+1)(f-C_a^D)^2+(\frac{n}{2}-2)(2f-1-C_a^D)^2+(n+1)C_a^{D^2}\Big] \\
&= \frac{1}{2n}\Big[(\frac{n}{2}+1)(f-\frac{3nf-6f-n+4}{4n})^2 \\
&\quad +(\frac{n}{2}-2)(2f-1-\frac{3nf-6f-n+4}{4n})^2+(n+1)(\frac{3nf-6f-n+4}{4n})^2\Big] \\
&= \frac{1}{64n^3}\Big[(n+2)(nf+6f+n-4)^2+(n-4)(5nf-3n+6f-4)^2 \\
&\quad +(2n+2)(3nf-6f-n+4)^2\Big] \\
&= \frac{1}{64n^3}\Big[44n^3f^2-144nf^2+12n^3-64n-80n^2f^2-40n^3f \\
&\quad +112n^2f+288nf-32n^2\Big] \\
&= \frac{1}{16n^2}\Big[11n^2f^2-36f^2+3n^2-16-20nf^2-10n^2f-28nf+72f-8n\Big]
\end{aligned}
$$

Out of the $2n$ buffers in the optimal solution n are of load f, $(\frac{n}{2}+1)$ of load $(f-1)$ and the remaining $(\frac{n}{2}-1)$ buffers have zero load. Then,

$$
\begin{aligned}
L_m^O &= f \\
C_a^O &= \frac{nf+(\frac{n}{2}+1)(f-1)}{2n} = \frac{3nf-n+2f-2}{4n} \quad \text{and} \\
V^O &= \frac{1}{2n}\Big[n(f-C_a^O)^2+(\frac{n}{2}+1)(f-1-C_a^O)^2+(\frac{n}{2}-1)C_a^{O^2}\Big]
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{2n}\Big[n(f-\frac{3nf-n+2f-2}{4n})^2+(\frac{n}{2}+1)(f-1-\frac{3nf-n+2f-2}{4n})^2 \\
&\quad +(\frac{n}{2}-1)(\frac{3nf-n+2f-2}{4n})^2\Big] \\
&= \frac{1}{64n^3}\Big[2n(nf+n-2f+2)^2+(n+2)(nf-3n-2f+2)^2 \\
&\quad +(n-2)(3nf-n+2f-2)^2\Big] \\
&= \frac{1}{64n^3}\Big[12n^3f^2+12n^3-16nf^2-16n-8n^3f-16n^2f^2 \\
&\quad +16n^2+32nf\Big] \\
&= \frac{1}{16n^2}\Big[3n^2f^2+3n^2-4f^2-4-2n^2f-4nf^2+4n+8f\Big]
\end{aligned}
$$

Simple substitution of the values above for the product $V \times C_a^2 \times L_m^2$ gives the value of $D(I_W)$ and $OPT(I_W)$. To find the limit of the ratio we observe the following:

1. $\lim_{n\to\infty,f\to\infty} \frac{C_a^D}{C_a^O} = 1$,
2. $\lim_{n\to\infty,f\to\infty} \frac{L_m^D}{L_m^O} = 2$ and
3. $\lim_{n\to\infty,f\to\infty} \frac{V^D}{V^O} = \frac{11}{3}$

Because L_m and C_a are both raised into the second power then

$$\lim_{n\to\infty,f\to\infty} \frac{D(I_W)}{OPT(I_W)} = \frac{4\times 11}{3} = \frac{44}{3}$$

which proves the lemma. □

The ratio $\frac{D(I_W)}{OPT(I_W)}$ is also plotted on Figure 33 to verify graphically our claim.

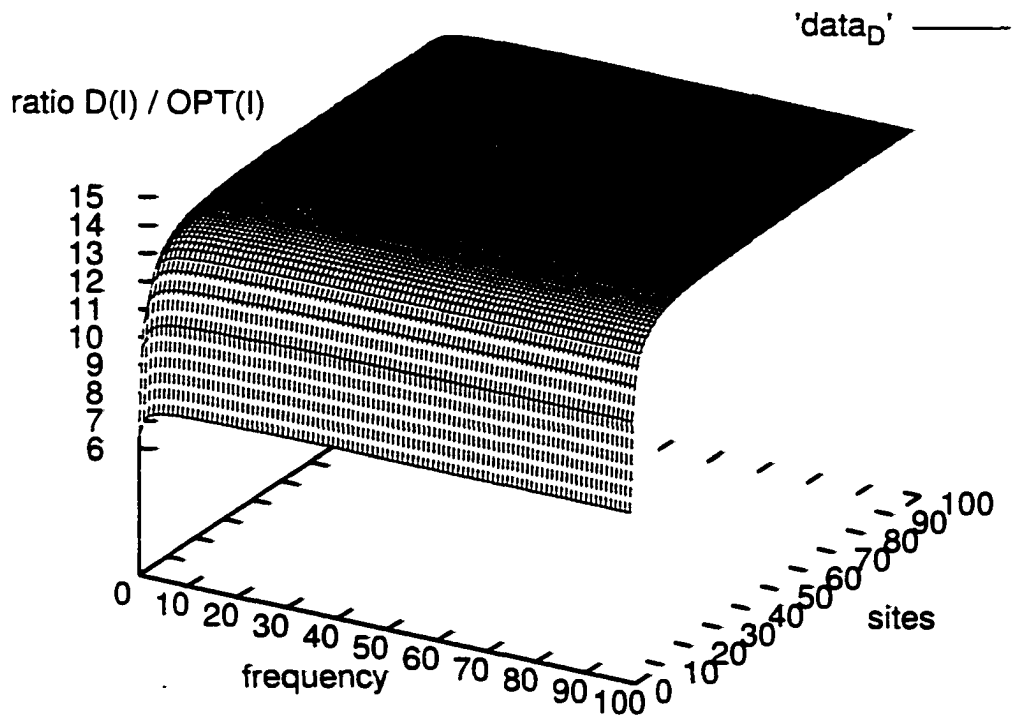

Figure 33. Graphical representation of the ratio $\frac{D(I_W)}{OPT(I_W)}$ in various ring topologies.

Upper bounds for the Hierarchical heuristic

For an input instance I of rings $G = (S, E, F_e)$, let $OPT(I)$ and $H(I)$ be the solutions of the optimal and the hierarchical heuristic respectively. We concentrate our discussion to F_e distributions which, regardless of the routing paths assigned to $f_{i,j}$'s, they always result into a variance $V \neq 0$. For this case

we want to compare the performance of the $H(I)$ to $OPT(I)$.

The worst case F_e distribution for $H(I)$ is the one that forces unnecessary overlap of routes and therefore, it increases the variance and the maximum buffer load. We construct this instance, namely I_W, with the intention to defeat the advantage of $H(I)$ inserting frequencies in the routing scheme using a path length ordering.

More specifically, let $G = (S, E, F_e)$ be a ring of $|S| = n$ sites and $|E| = n$ half duplex links with $2n$ output buffers associated. Without loss of generality, let $n = 2m$, with $m \in Z^+$. Let also the set F_e be defined as follows:

$$F_e = \begin{cases} f_{1,\frac{n}{2}-1} = f \\ f_{\frac{n}{2}-1,1} = f \\ f_{2,\frac{n}{2}+1} = f \\ f_{i,j} = 0 \ \ \forall \ (i,j) \neq \{(1,\frac{n}{2}-1),(\frac{n}{2}-1,1),(2,\frac{n}{2}+1)\} \end{cases}$$

Note that, $H(I)$ is forced to insert first the frequencies $f_{1,\frac{n}{2}-1}$ and $f_{\frac{n}{2}-1,1}$. The reason is that, these frequencies can be routed with path lengths $(\frac{n}{2}-2)$ and these lengths are the smallest possible for the set F_e at hand. The frequencies $f_{1,\frac{n}{2}-1}$ and $f_{\frac{n}{2}-1,1}$, when routed that way, they do not overlap. Furthermore, the smallest possible length for the frequency $f_{2,\frac{n}{2}+1}$ is $(\frac{n}{2}-1)$ therefore according to $H(I)$ it is routed last. However, the routing of $f_{2,\frac{n}{2}+1}$ causes an overlap with the route of the frequency $f_{1,\frac{n}{2}-1}$ charging the buffers $\{(2,3),(3,4),\cdots,(\frac{n}{2}-2,\frac{n}{2}-1)\}$

with load $2f$. Figure 34(a) illustrates the routing resulted by $H(I)$.

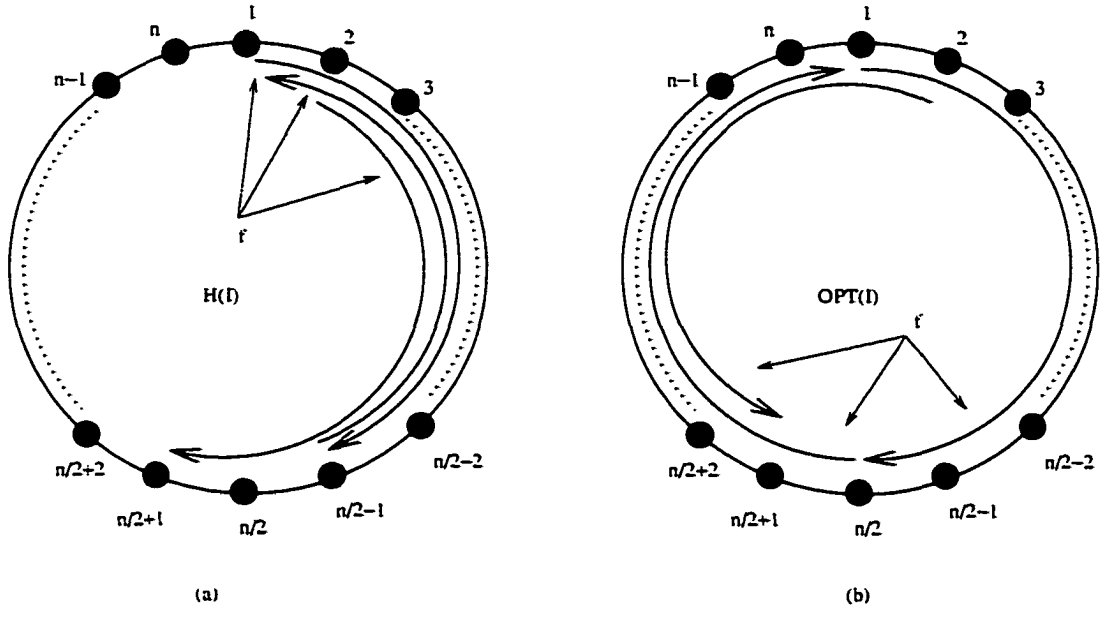

Figure 34. Worst case F_e distribution routing for $H(I)$, and $OPT(I)$.

On the other hand, $OPT(I)$ routes $f_{\frac{n}{2}-1,1}$ and $f_{2,\frac{n}{2}+1}$ using opposite direction paths. This causes L_m to be reduced to f which is the minimum possible. It also achieves the best possible balance on the buffers and this happens with a minimum increase on the average communication cost. The routing resulted by $OPT(I)$ is depicted in Figure 34(b).

In the following lemma, we prove that, $\frac{H(I_W)}{OPT(I_W)}$ is a monotonically increasing function but it is upper bounded by the value of $\frac{44}{3}$ for any ring topology and any value f of the maximum frequency in I_w.

Lemma 17 : For the worst case F_e distribution instance I_W, the ratio $\frac{H(I_W)}{OPT(I_W)}$ is

a monotonically increasing function on n and f, bounded by an upper bound of value $\frac{44}{3}$.

Proof: We show that,

$$\lim_{n\to\infty, f\to\infty} \frac{H(I_W)}{OPT(I_W)} = \frac{44}{3}$$

Without loss of generality, let $n = 2m$, with $m \in Z^+$ be the number of sites in the ring. Let L_m^H, C_a^H and V^H be the maximum buffer load, average communication cost and buffer variance resulted by the solution $H(I_W)$. Let also L_m^O, C_a^O and V^O be the corresponding quantities for the optimal solution.

The routing of $H(I)$ results into two paths of length $(\frac{n}{2}-2)$ and a path of length $(\frac{n}{2}-1)$. Out of the $2n$ network buffers, $(\frac{n}{2}-3)$ are charged with load $(2f)$, $(\frac{n}{2}+1)$ are charged with load f and the remaining $(n+2)$ buffers have zero load. Then,

$$\begin{aligned}
L_m^H &= 2f \\
C_a^H &= \frac{2(\frac{n}{2}-3f)+(\frac{n}{2}+1)f}{2n} = \frac{(3n-10)f}{4n} \quad \text{and} \\
V^H &= \frac{1}{2n}\Big[(\frac{n}{2}+1)(f-C_a^H)^2+(\frac{n}{2}-3)(2f-C_a^H)^2+(n+2)C_a^{H^2}\Big] \\
&= \frac{1}{2n}\Big[(\frac{n}{2}+1)(f-\frac{(3n-10)f}{4n})^2 \\
&\quad +(\frac{n}{2}-3)(2f-\frac{(3n-10)f}{4n})^2+(n+2)(\frac{(3n-10)f}{4n})^2\Big] \\
&= \frac{f^2}{64n^3}\Big[(n+2)(n+10)^2+(n-6)(5n+10)^2+(2n+4)(3n-10)^2\Big] \\
&= \frac{f^2}{64n^3}\Big[44n^3-800n-112n^2\Big] \\
&= \frac{f^2}{16n^2}\Big[11n^2-200-28n\Big]
\end{aligned}$$

Out of the $2n$ buffers in the optimal solution $(\frac{3n}{2}+1)$ are of load f and the remaining $(\frac{n}{2}-1)$ buffers have zero load. Then,

$$\begin{aligned}
L_m^O &= f \\
C_a^O &= \frac{nf+(\frac{n}{2}+1)f}{2n} = \frac{(3n+2)f}{4n} \quad \text{and} \\
V^O &= \frac{1}{2n}\Big[(\frac{3n}{2}+1)(f-C_a^O)^2+(\frac{n}{2}-1)C_a^{O2}\Big] \\
&= \frac{f^2}{64n^3}\Big[(3n+2)(n-2)^2+(n-2)(3n+2)^2\Big] \\
&= \frac{f^2}{64n^3}\Big[4n(3n+2)(n-2)\Big] \\
&= \frac{f^2}{16n^2}\Big[(3n+2)(n-2)\Big]
\end{aligned}$$

Simple substitution of the values above for the product $V \times C_a^2 \times L_m^2$ gives the value of $H(I_W)$ and $OPT(I_W)$. To find the limit of the ratio we observe the following:

1. $\lim_{n\to\infty,f\to\infty} \frac{C_a^H}{C_a^O} = 1$,
2. $\lim_{n\to\infty,f\to\infty} \frac{L_m^H}{L_m^O} = 2$ and
3. $\lim_{n\to\infty,f\to\infty} \frac{V^D}{V^O} = \frac{11}{3}$

Because L_m and C_a are both raised into the second power,

$$\lim_{n\to\infty,f\to\infty} \frac{H(I_W)}{OPT(I_W)} = \frac{4\times 11}{3} = \frac{44}{3}$$

which proves the lemma. □

The ratio $\frac{H(I_W)}{OPT(I_W)}$ is also plotted in Figure 35 to verify graphically our claim.

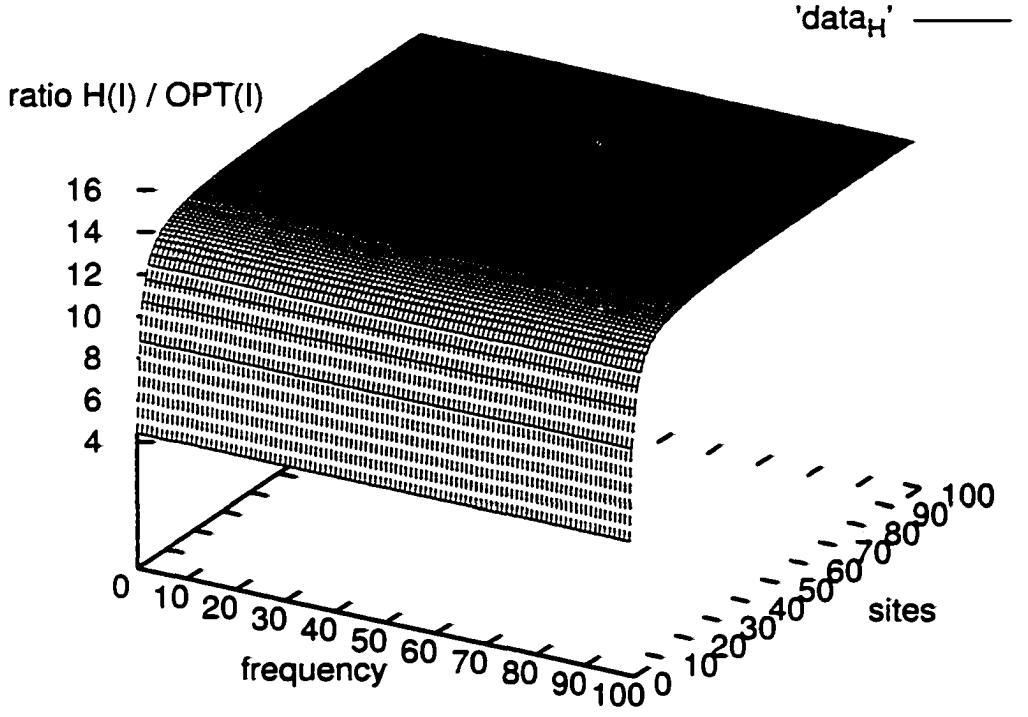

Figure 35. Graphical representation of the ratio $\frac{H(I_W)}{OPT(I_W)}$ in various ring topologies.

Hybrid global static and dynamic distributed QoS routing with bandwidth guarantees

In this section, we propose a new QoS routing model which is hybrid in terms of how routing paths are established. For this model, we assume that, individual source-destination pairs of sites demand bandwidth for information

transmission which must be guaranteed for the session duration. There are a lot of applications in wide-area-networks and the internet which work under the above assumption. A typical example is live-video-conferencing where, the constant bit rate and the end-to-end delay are very critical for the quality of the application.

Even though many algorithms have been established for Quality of Service with bandwidth guarantees, almost all of them concentrate on the optimization aspects of the routing of individual sessions without taking into account the overall performance of the network. Note that, this may sound correct when networks are not heavily loaded by bandwidth-guaranteed live sessions. In the case though that, the opposite happens, the optimization of global resource reservation parameters is an important factor to accommodate the traffic demands. Therefore, QoS routings must be developed which maximize the overall network performance without sacrificing the requirements of specific sessions. These algorithms have to be link-state algorithms since the link-state parameters frequently change and any attempt of source routing is based on inaccurate link-state information of the network.

In distributed routing algorithms, the path-selection computation is distributed among the intermediate nodes therefore, the routing algorithms are more scalable to network sizes. For this case of algorithms though, the routing performance depends substantially on the accuracy of the global link state of the network. Sessions are established by an initiation of an establishing packet injected

from the source node. This packet is forwarded to a node through a specific link that satisfies the session requirements (bandwidth, cost restriction etc.). When this packet reaches the destination node, then the session is established. In the case that, the packet is rejected from a node it returns back to the previous node for a different link choice. There may be a limit on the number of link trials for each node. After the session is established, intermediate nodes know already how to forward the packets/cells for that session.

We present a new model which uses the global bandwidth demands of the network along with the heuristics discussed in the previous section to give the first choice of path selection in distributed routing. In the case that, the heuristic answer does not guarantee the bandwidth requirement and the cost restriction of the session then, a distributed link probing algorithm is triggered. This algorithm is a distributed backtracking algorithm on the intermediate path nodes which makes a choice of a link for forwarding establishing session packets. If the distributed algorithm cannot establish a session, then this session is rejected and delayed for some other time.

In the following subsections, we describe:

1. The network assumptions and the representation of establishing session packets.

2. The data structures and storage requirements in that process.

3. A heuristic to minimize the variance of link loads and the cost of each

session while satisfying the bandwidth requirement.

Model representation and assumptions

The network is represented as a graph $G = (S, E, F_e, C_e, B_e)$ where :

1. S is the set of network sites.

2. E is the set of half doublex links. For every pair of adjacent sites $\{i, j\}$ there are two output buffers (i, j), (j, i) associated with the corresponding flow direction.

3. $C_e = \{c_{i,j} | c_{i,j}$ cost for buffer $(i, j)\}$ is the cost associated with each output buffer. The cost $c_{i,j}$ is a compound cost which includes the packet transit delay, the traffic collision delay etc.

4. F_e is the set of network demands between any source-destination pair. Note that, F_e represents traffic demands with bandwidth guarantees.

5. B_e is the set of bandwidth availabilities for each link direction ($b_{i,j}$ and $b_{j,i}$ are associated with each network link).

The total cost for a path $P = \{s_0, s_1, s_2, \cdots, s_{m-1}, s_m\}$ is equal to:

$$C_P = \sum_{i=0}^{m-1} c_{i,i+1}$$

The cost restriction on the network demands represents an upper bound on the end-to-end delay for paths associated with each session. A network request for a live session with bandwidth guarantees and cost restrictions is represented by

what we call an *Establishing Session Packet* which is an 8-tuple:

$$q_{esp} = (session - id, source, dest, path, b_{s,d}, c_{s,d}, Path.cost, Static)$$

where

1. The *session-id* is a unique *id* for the session to be established.
2. *source* is the source of the connection.
3. *dest* is the destination of the connection.
4. *path* is an array of integers which contains the set of intermediate nodes for the path of the session. Initially all elements of the array are empty.
5. $b_{s,d}$ is the bandwidth requirement for the session and
6. $c_{s,d}$ is the upper bound on the cost for the connection. Every acceptable path $p_{s,d}$ must satisfy the condition that $C_{P_{s,d}} \leq c_{s,d}$.
7. *Path.cost* is the present cost of the path up to the site the q_{esp} has reached. Initially $Path.cost = 0$.
8. *Static* is a flag that indicates whether or not to use the static routing table.

Data structures added for each site

For the implementation of this distributed QoS routing we use 2 routing tables. The first is the static routing table discussed in chapter IV. For a network of n sites, the second table is an $(n \times n)$ array of linked lists of records and it is associated with the *QoS* routing. Every entry of the table corresponds to a

(*source, destination*) pair. For every session relatively to each pair of sites (i, j), (established session or in the process of establishment), there is a record entry in the table of a site k if the path of (i, j) currently includes k. Different sessions for the same pair (i, j) passing from the same intermediate site k are associated with a different record which is linked at the (i, j) table entry of site k. More specifically, each (i, j) linked list contains records of the following form:

```
Record
{
    Session_id      As Integer
    Number_of_Trials As Integer
    Limit_of_Trials As Integer
    Establish_Flag   As Boolean
    Bandwidth        As Integer
    Next_Hop         As Integer
    List_of_links_tried As an array of integers
}
```

Figure 36. QoS Routing Table Linked List entry.

In the case that a session is established, the Session_Id is set and the Establish_Flag is set to True. This is done by a second packet that travels the path informing all intermediate nodes that a session has been established. Furthermore, the Next_Hop entry shows how to forward the session packets and the required bandwidth is indicated in the corresponding variable.

When a session is established or the *Establishing session packet* reaches a

site for the first time, the Number_of_Trials variable is set to zero. This variable records the number of times a specific site has tried various adjacent links in order to establish a path for a specific q_{esp}. When a link is tried to forward traffic the Number_of_Trials is increased by one. On the other hand, the Limit_of_Trials variable is an upper bound on the number of times the site has to try at most to find a forwarding link for the traffic. If Number_of_Trials becomes greater than Limit_of_Trials, then the *establish session packet* is rejected, the whole record is deleted from the linked list and the packet is moved back to the previous node in its chosen path.

QoS distributed heuristic

We describe a distributed QoS heuristic to minimize the variance of link loads, the cost of sessions and the maximum buffer load locally while providing with Quality of Service for sessions with bandwidth guarantees. In this section, we assume that all network sites have the ability to make a distinction between *establishing session packets*, (q_{esp}), simple session packets that transmit data and other packets/cells that don't require bandwidth guarantees.

The heuristic tries to establish live-sessions between network sites by giving the first choice to the path indicated by the static routing table. When this path is rejected due to the inability of the network to guarantee bandwidth, the distributed algorithm dynamically switches to the functionality needed for finding

another path. This functionality is a "backtracking like" method that tries to optimize $V \times C_a^2 \times L_m^2$ locally for each site.

When a (q_{esp}) reaches a site-i, the local QoS routing table entry is checked to find whether this packet is new or not. In the case of a new packet, the static flag of the packet is checked. If the indication is to use the static path, then the static routing table is consulted to find the next hop. We then check whether or not the link found satisfies the bandwidth and cost criteria. If the criteria are satisfied, then the QoS routing table is updated and the q_{esp} packet is forwarded to the next hop.

In the case that, the criteria are not satisfied we set the static flag of the packet to indicate distributed routing and we probe all adjacent links to find the most suitable link. This link is the one that:

1. Avoids a path cycle,
2. Satisfies the bandwidth and cost restrictions and
3. Minimizes the load variance and the maximum buffer load among all adjacent output buffers.

The choice of the next hop must satisfy the first two conditions. Among all candidate links that satisfy the two conditions, the one that minimizes V and L_m locally is chosen.

If there is not a link that satisfies the first two conditions, the packet is rejected, the corresponding QoS routing table entry is deleted and the packet is

forwarded back to the previous site of its path to choose another route.

In the case the packet had been forwarded before but reaches the node again, then another trial is attempted if the number of trials is less than the limit. The link set is limited only to those which have not been tried previously. The function for finding the next link is shown in Figure 37.

```
Function Find_Link(Input: q_esp)
{
    Next_Hop = -1
    FOR i ← 1 to all adjacent links
        IF Perform_Establishement_Test(q_esp, node, i) = True
            Include link (node, i) as a candidate link
        END IF
    END FOR
    IF there are no candidates
        Next_Hop = -1
    ELSE
        FOR i ← 1 to all candidate links
            Choose the one with the minimum current load
        END FOR
        Update q_esp.Path.Cost
        Update q_esp.Path
    END IF
    return Next_Hop
}
```

Figure 37. Function that searches for the appropriate link to minimize locally L_m and V.

Note that, the function returns -1 if there is no candidate link that satisfies the first two criteria mentioned above. These conditions are checked with a call

to the function Perform_Establishment_Test() which is depicted in Figure 38.

```
Function Perform_Establishment_Test(Input:  q_esp, node, next_node)
{
    IF Avail.Bandwidth(node, next_node) ≤ q_esp.bandwidth AND
        q_esp.Path.Cost + Cost(node, next_node) ≤ q_esp.Cost AND
        next_node not in q_esp.Path
            return True
    ELSE
            return False
}
```

Figure 38. Function that performs the path cycle, the bandwidth reservation and the cost reservation test.

Among all candidate links that return True-value to the above function call, the one with the present minimum buffer load is chosen. This choice minimizes the variance of local buffer loads. The reason is that, choosing the one with the minimum present load, we decrease the spread of load values while reducing the probability of increasing the maximum buffer load locally. On the other hand, the cost restriction must be also satisfied.

Finally, the integrated QoS algorithm that runs at each site is illustrated in Figure 39.

```
Distributed QoS Heuristic(Input:  q_esp)
{
    IF ∃ an entry in the QoS R.T. for q_esp
        IF q_esp.static_flag = 1 then q_esp.static_flag = 0
L1:     q_esp.Number_of_Trials ++
        IF q_esp.Number_of_Trials ≤ q_esp.Limit_of_Trials
            Next_Hop = find_link(q_esp)
            IF Next_Hop = -1
                Reject Packet and remove QoS table entry
                Forward packet to q_esp.Path.previous_node
            ELSE
                Update q_esp.Path and q_esp.Path.Cost
                Forward q_esp to Next_Hop
        ELSE
            IF node_id = q_esp.source THEN abort session
            ELSE
                Reject Packet and remove QoS table entry
                Forward packet to q_esp.Path.previous_node
    ELSE
        IF q_esp.static_flag =1
            Next_Hop = Consult Static R.T
            Success = Perform_Establishment_Test(q_esp)
            IF Success = 0
                q_esp.static_flag = 0
                q_esp.Number_of_Trials ++
                Next_Hop = find_link(q_esp)
                IF Next_Hop = -1
                    Reject Packet
                    Forward packet to q_esp.Path.previous_node
                ELSE
                    Accept Packet, Insert QoS R.T. Entry
            ELSE
                Accept Packet, Insert QoS R.T. Entry
        ELSE
            Insert the entry q_esp in QoS R.T.
            GOTO L1
}
```

Figure 39. QoS distributed routing heuristic.

CHAPTER VI

EXPERIMENTAL RESULTS AND HEURISTIC PERFORMANCE ANALYSIS

Introduction

In this chapter we describe and analyze various experimental results relative to the three heuristics introduced in Chapter V. The heuristics have been implemented and run for different topologies such as rings (up to 20 sites), hypercubes of 8 and 16 sites and a Z-Cube of 16 sites. We choose these topologies because they are the most popular regular topologies of degree 2, 3 and 4.

For the need of these experiments we have developed an integrated network simulator application for the Windows and the Windows NT operating system. The simulator is programmed in ©Microsoft Visual Basic version 6.0 and it integrates the three routing heuristics under the above topologies assuming half duplex links, bandwidths ranging from 10 to 100 packets/cells per time unit and output buffers of capacities ranging from 10 to 500 packets/cells. A screen shot of the simulator is shown in Figure 40.

Moreover, we have added capabilities of choosing various distributions for the set of communication frequencies as Figure 41 illustrates.

The duration of the simulation can vary from 2 minutes to 20 minutes. The

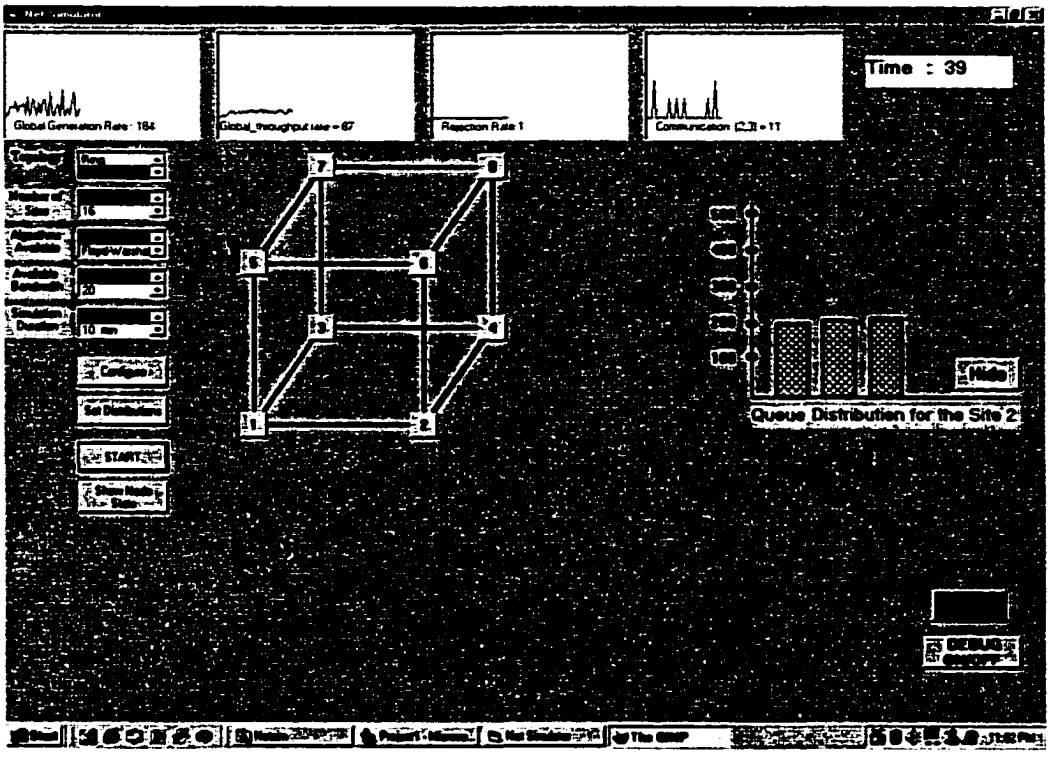

Figure 40. A screen shot of the network simulator.

data set of the communication frequencies can be created before the simulation or during run time.

During the simulation we can watch the capacities of the output buffers for a specific site as well as four curve indicators:

1. The global message generation rate,
2. The average throughput rate of the system,
3. The average rejection rate of packets/cells and
4. The pairwise message generation rate between any pair of network sites.

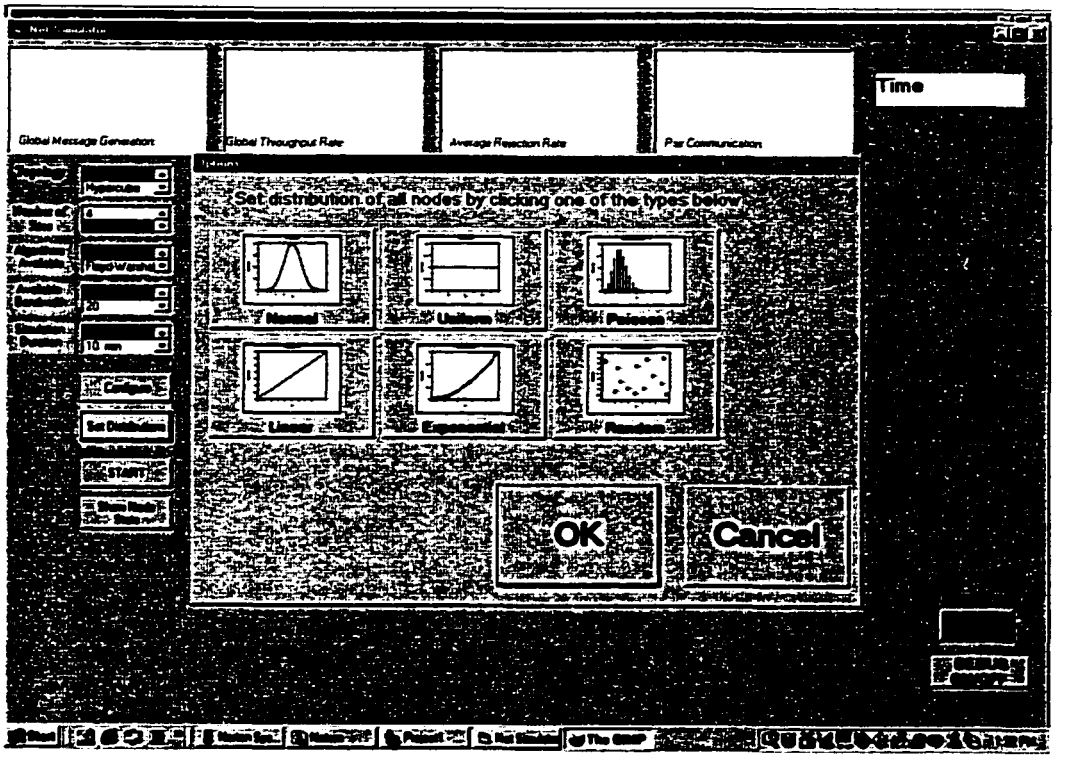

Figure 41. A simulator screen shot of the distribution form.

At the end of each simulation, we provide a three-dimensional bar graph of the statistics kept. These statistics include the average throughput rate, the average end-to-end delay, the average delay jitter, the number of reroutings needed and the average congestion for the network. In Figure 42 we illustrate a captured screen shot of the statistics bar chart. The congestion metric used is the one introduced in Chapter IV.

The organization of this chapter is based on the topology classification. We include three sections one for the rings, one for the hyper-cubes and one for the

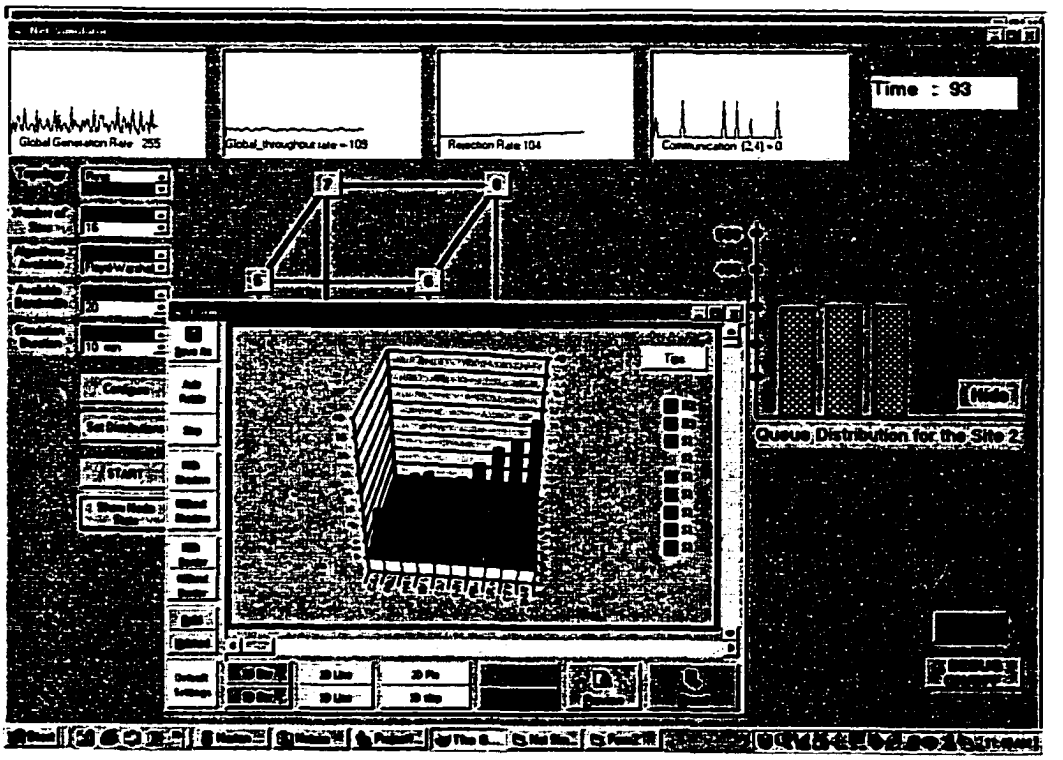

Figure 42. A simulator screen shot of the final statistics graph.

Z-Cubes. For each of the sections, we provide plotted data that indicate the effect of the bandwidth, buffer size and F_e distribution to the throughput, end-to-end delay, delay jitter, number of reroutings and congestion. In each figure there are three curves which compare the behavior of the three heuristics relatively to each measurement. Moreover, we comment on the results and analyze our findings.

Our experiments are data dependent. Even though the data used do not correspond to real network traffic data, we assumed in most of our experiments random traffic demands of bursty nature. This data model is the most acceptable

in today's technology and its behavior is close to a real network environment.

Performance analysis of the heuristics in rings

As we mentioned in the introduction, the simulator can handle rings for up to 20 sites. We run 100 simulations of duration 2 minutes each, using rings of different site numbers. This mix contained 30 rings of 8 sites 30 rings of 16 sites and 40 rings of randomly selected sites. The reason for the chosen percentages was the behavior comparison of rings with the hyper-cubes and Z-Cubes of the same number of sites. Half of the simulations were run with bandwidths increasing from 10 to 100 with step 10 and constant buffer size of 250 cells. The other half was run with constant bandwidth 50 and variable buffer size ranging from 50 to 500 with increments of 50. The distributions for the set F_e used were random bursts, normal distribution bursts with random offsets and constant distribution bursts with random offsets and durations.

The effect of bandwidth for the heuristics in rings

In Figure 43 we show the effect of the bandwidth increase to the average throughput rate for each of the heuristics. The small values of the average throughput rate are because of the half duplex links. Note that in every simulation interrupt, the buffer data can be transmitted only in one link direction.

As Figure 43 illustrates, the Dijkstra heuristic performs asymptotically

better than the other two. However, the Floyd-Warshall and the Hierarchical heuristics are competing very close. This is an indication that, the distribution combinatorics chosen by the Dijkstra heuristic to insert frequencies in the routing scheme create a better balance on the average for the network output buffers. This contributes in better averages of throughput rate of all sites.

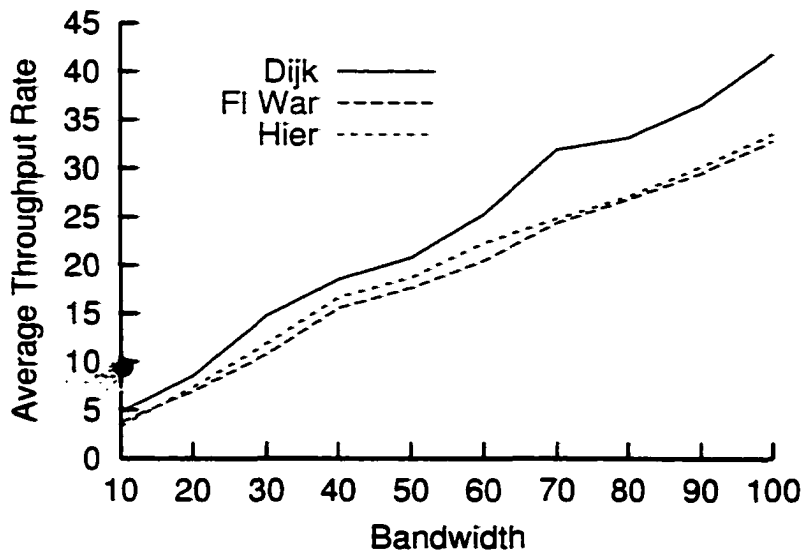

Figure 43. The average throughput rate in relation to the bandwidth in rings.

In Figure 44 we illustrate how the increase of bandwidth affects the end-to-end delay for the three heuristics in rings. Note that, the Dijkstra heuristic again performs better than the other two heuristics in that metric. Also the hierarchical heuristic performs better than the Floyd-Warshall heuristic on the average except for bigger bandwidths where the two compete. The increase of the bandwidth

results in a decrease of the average end-to-end delay for all heuristics as it was expected.

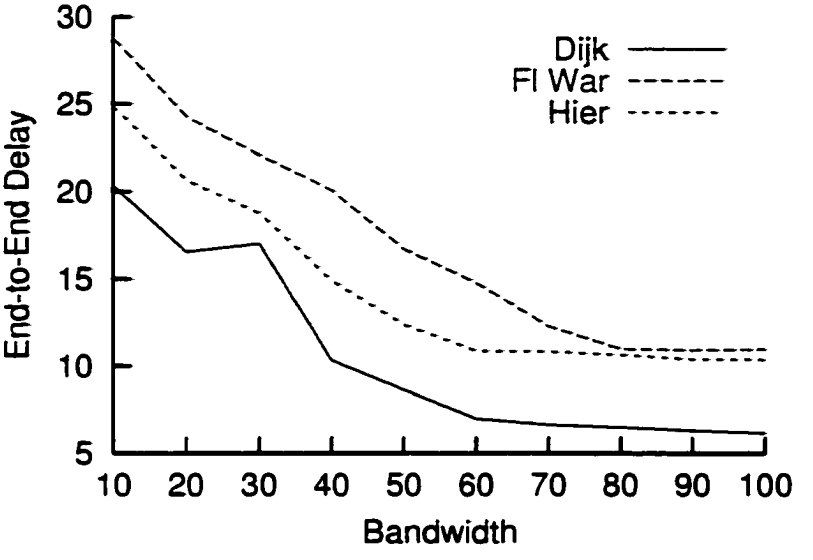

Figure 44. The average end-to-end delay in relation to the bandwidth rate in rings.

The relation of bandwidth and average delay jitter is shown in Figure 45. We have expected that the Dijkstra heuristic would perform better than the other two. This is true except when, the bandwidth ranges between 40 and 50. In that range, the Hierarchical heuristic performs better. This can be explained by the nature of the topology and the actual frequency data sets. For rings of small size, the probability of the three heuristics to result into the same average delay jitter increases. As the ring size increases, the Dijkstra heuristic performs better than the hierarchical heuristic in that metric also.

To justify our claim, we ran 5 additional simulations with rings of 20 sites which are not included in the previous data set. In all the additional simulations, the Dijkstra heuristic performed asymptotically better.

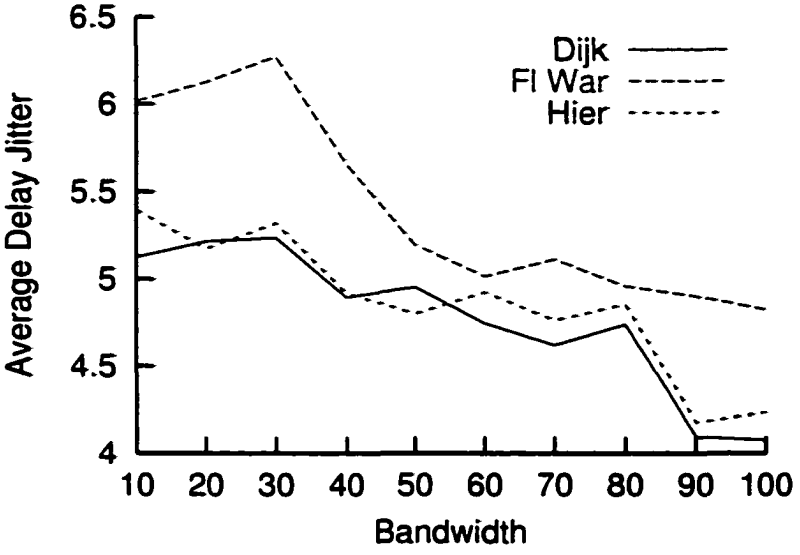

Figure 45. The average delay jitter in relation to the bandwidth in rings.

The measuring of the average number of reroutings for the three heuristics is illustrated in Figure 46. As the figure shows, the Dijkstra heuristic performs better on the average than the other two heuristics. However, in that case, the Floyd-Warshall heuristic performs better than the hierarchical heuristic. We believe that, this interchange in performance between the hierarchical and the Floyd-Warshall heuristic happens due to the randomness of input data set.

Finally, the last measurement taken is the average link congestion experi-

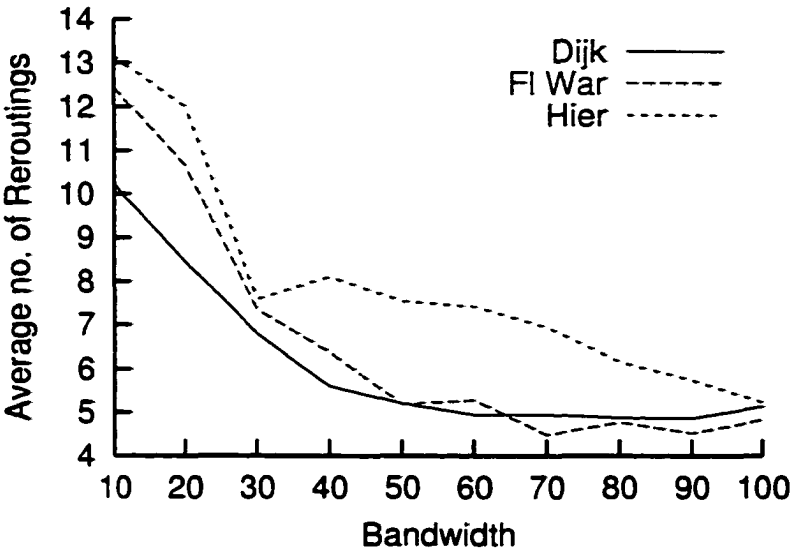

Figure 46. The average number of reroutings in relation to the bandwidth in rings.

enced for the data set using the three heuristics. Our results are shown in Figure 47. For the congestion metric we use the Definition 17 from Chapter IV. Therefore, the average congestion metric at hand corresponds to the average deviation of the buffer loads. The optimal operation zone used is based on the buffer sizes. More specifically, the lower bound is set to be 60% of the buffer size and the upper bound 80% of the buffer size. The lower and upper degradation thresholds are set to be 15% of the value of the corresponding lower and upper optimal zone bounds.

As Figure 47 illustrates, all three heuristics experience high congestion when small bandwidth is used in the network. This is due to the fact that all buffers are of size 250 but the bandwidth is insufficient to accommodate all traffic

demands. The result is an accumulation of waiting packets in all buffers. As the bandwidth increases, we experience a decrease of congestion since, all sites are able to forward more data in the same time unit.

The performance of the Dijkstra heuristic remains superior in that metric also. This indicates that, the Dijkstra heuristic results in routings that asymptotically balance better the traffic at hand.

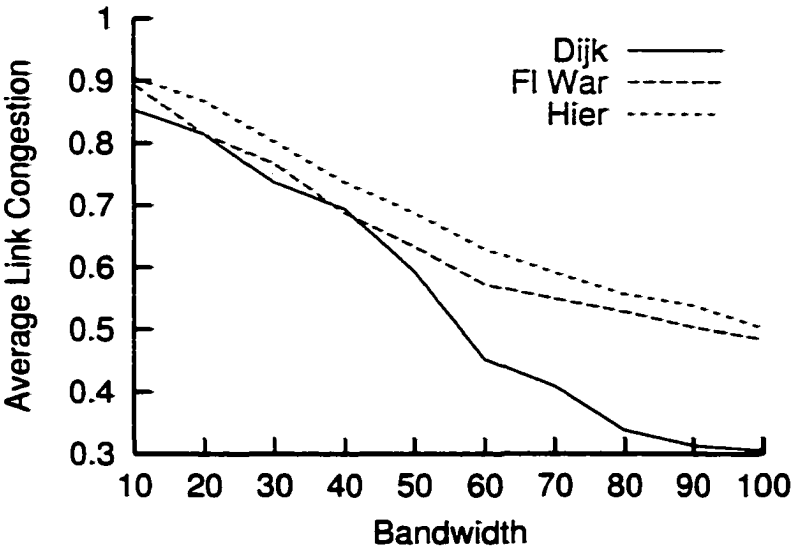

Figure 47. The average link congestion in relation to the bandwidth in rings.

The effect of buffer size for the heuristics in rings

To experiment the effect of the buffer size to the five metrics, we fixed the bandwidth to 50 cells per time unit and we varied the buffer size from 50 to 500

packet capacities. The increment step was set to 50 packets. For this experiment, the global data generation rate is related to the bandwidth. This means that, we don't increase the message generation rate as we increase the buffer size. The reason is that, we want to observe the effect of the buffer size increase under similar network loads and under fixed bandwidth.

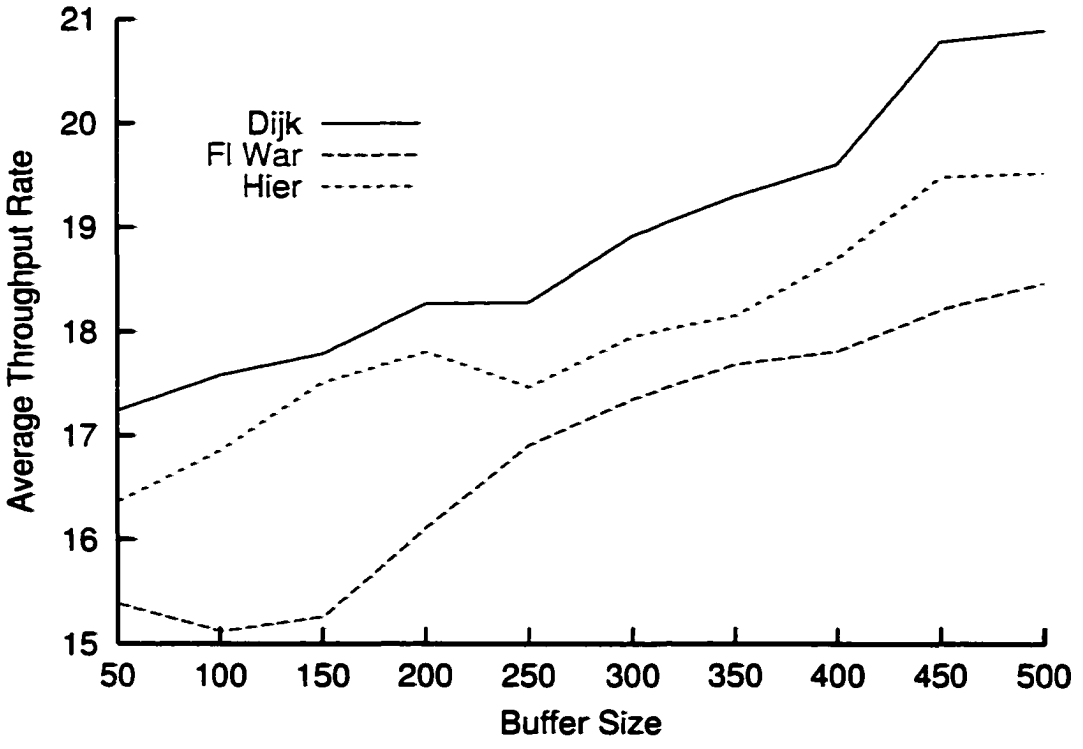

Figure 48. The average throughput rate in relation to the buffer size in rings.

Figure 48 illustrates how the buffer size affects the average throughput. The experiment shows that, there is no significant relevance between the two. Even though there is some increase in the average throughput as the buffer size

increases, this increase is minuscule.

However, the increase in throughput can be explained by the data randomness and by the fact that, the increase of buffer size minimizes the slow-down of message generation. This is not a significant factor in the case of moderate traffic data though.

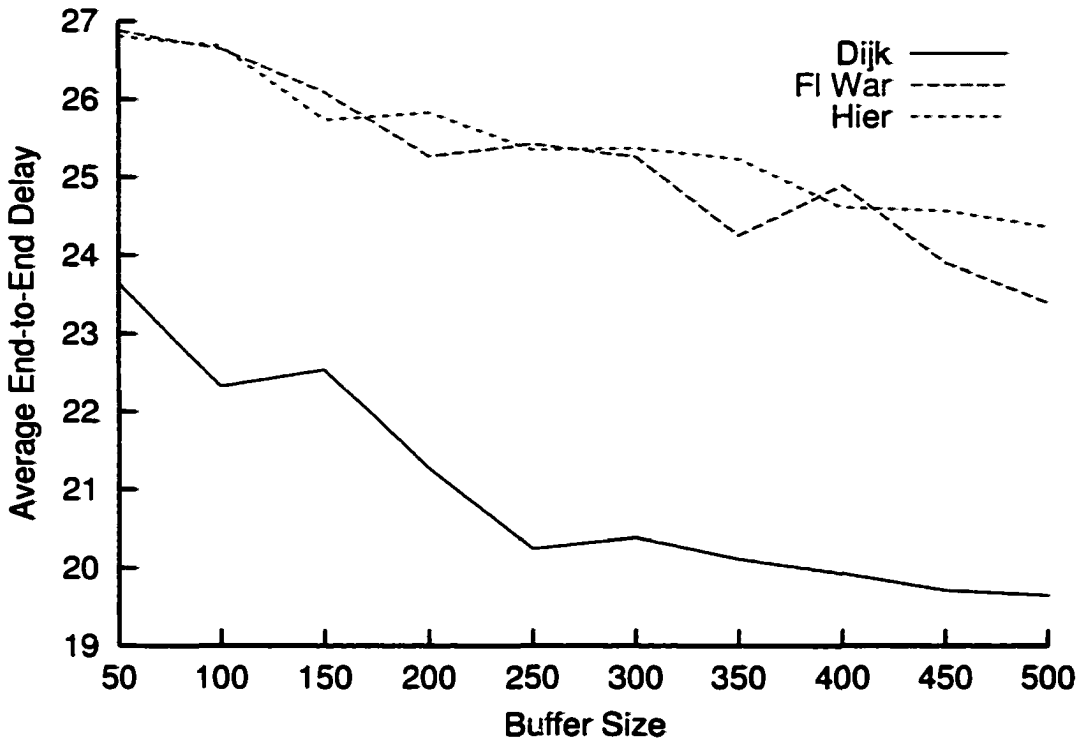

Figure 49. The average end-to-end delay rate in relation to the buffer size in rings.

Figure 49 shows the effect of the buffer size to the average end-to-end delay of the network. As the figure illustrates, there is a significant gap between the behavior of the Dijkstra heuristic and the behavior of the other two heuristics.

This is a representative result that justifies the Dijkstra's superiority in performance. We expected that, the increase of the buffer size would delay further the arrival of messages to their destinations. However in our case, when the buffers are of small size, a large number of messages is rejected and is not forwarded to the destination sites fast enough. As the buffer size increases, fewer messages are rejected and therefore, a decrease in end-to-end delay occurs.

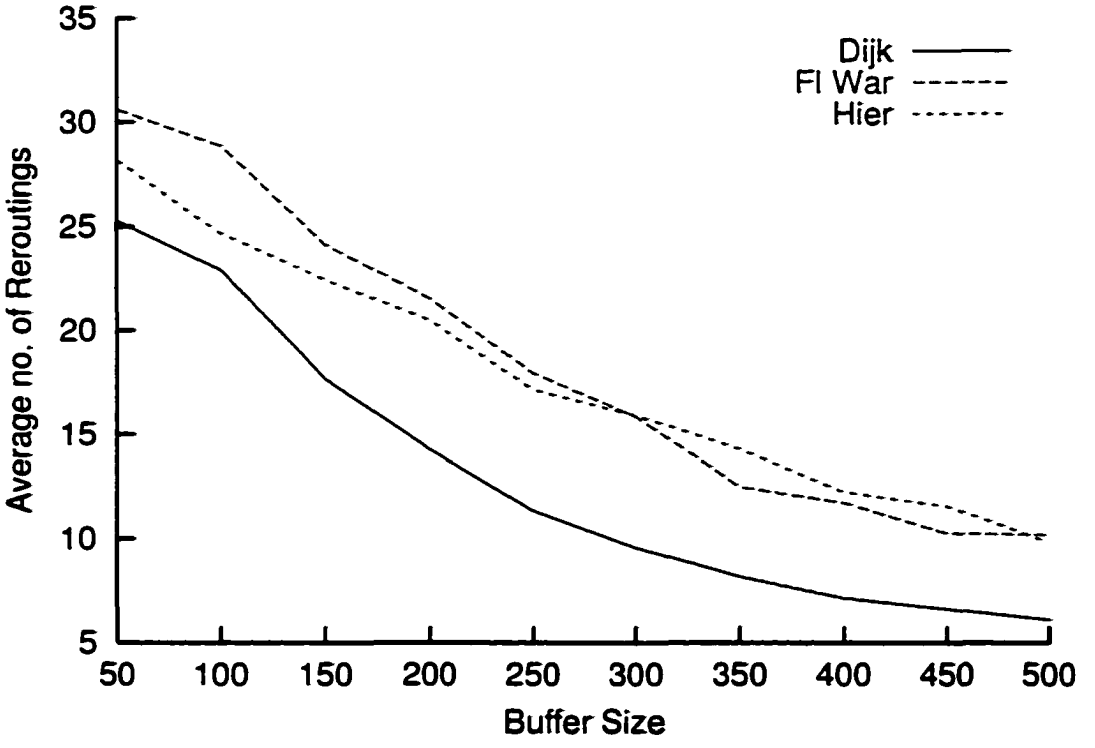

Figure 50. The average number of reroutings in relation to the buffer size in rings.

The gap in performance between the Dijkstra and the other two heuristics is explained in conjunction with the Figure 50. This figure shows the average

number of reroutings occurred for the three heuristics in all the simulations. Note that, the Dijkstra heuristic caused fewer reroutings on the average. This also has a significant effect on the average end-to-end delay since when a rerouting decision is applied then, all generated messages which have not left their source remain there until the new routing scheme is found. Therefore, this delay causes a significant increase on the average end-to-end delay.

The average number of reroutings is also responsible for the performance gap of the Dijkstra heuristic in comparison with the other two heuristics for the case of the delay jitter performance, (Figure 51).

A general observation is that, we have experienced data sets of communication frequencies that result into a significantly large number of reroutings in all heuristic cases. When this happens, the other metrics, (throughput, delay jitter and end-to-end delay) also degrade significantly. This is because, the delay to recompute the new routing scheme affects negatively the rest of the QoS parameters. In that case, the model fails to capture the system localities in order to predict the future frequencies. This is expected because of the randomness of the data.

Figure 52 shows the relation between the network buffer size and the average congestion experienced. Since we have created the optimal operation zone as a function of the buffer size, we observe a significant correlation between the two metrics. As the figure illustrates, the Dijkstra heuristic keeps the performance

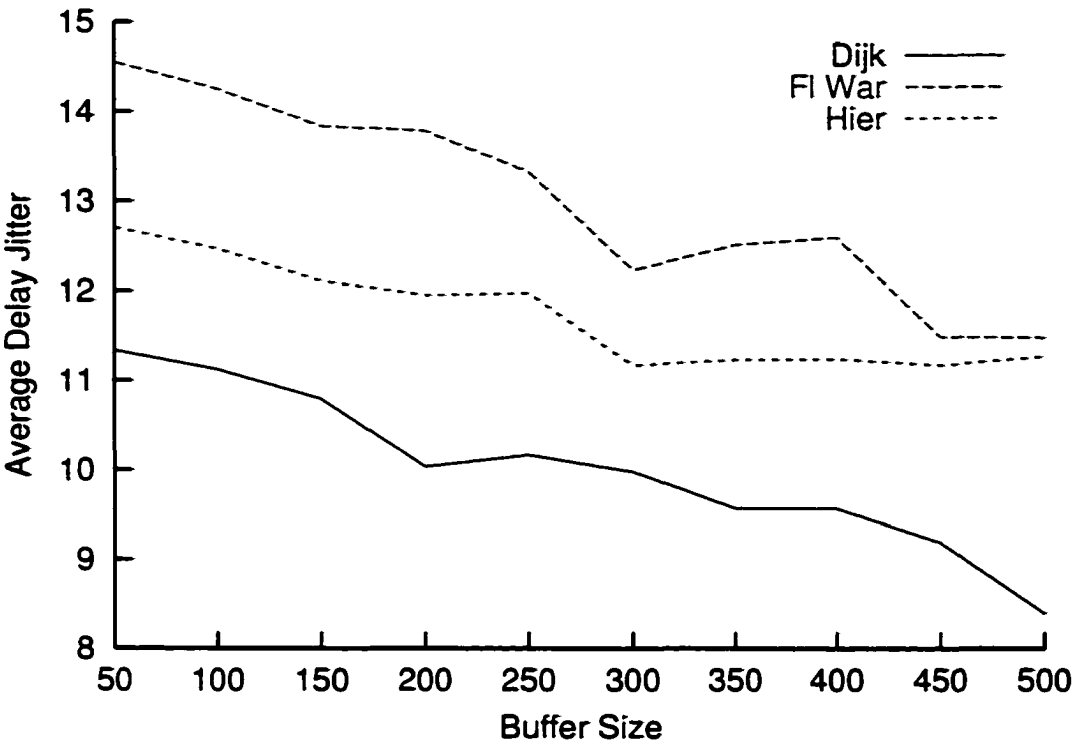

Figure 51. The average delay jitter in relation to the buffer size in rings.

superiority among the three algorithms. The average congestion degrades as the buffer size increases. This is because the degradation zones for the congestion measurement are also increased.

The effect of normal data distributions for the heuristics in rings

We have chosen to experiment with three different types of distributions and to examine their relation to the five metrics above. The distributions chosen

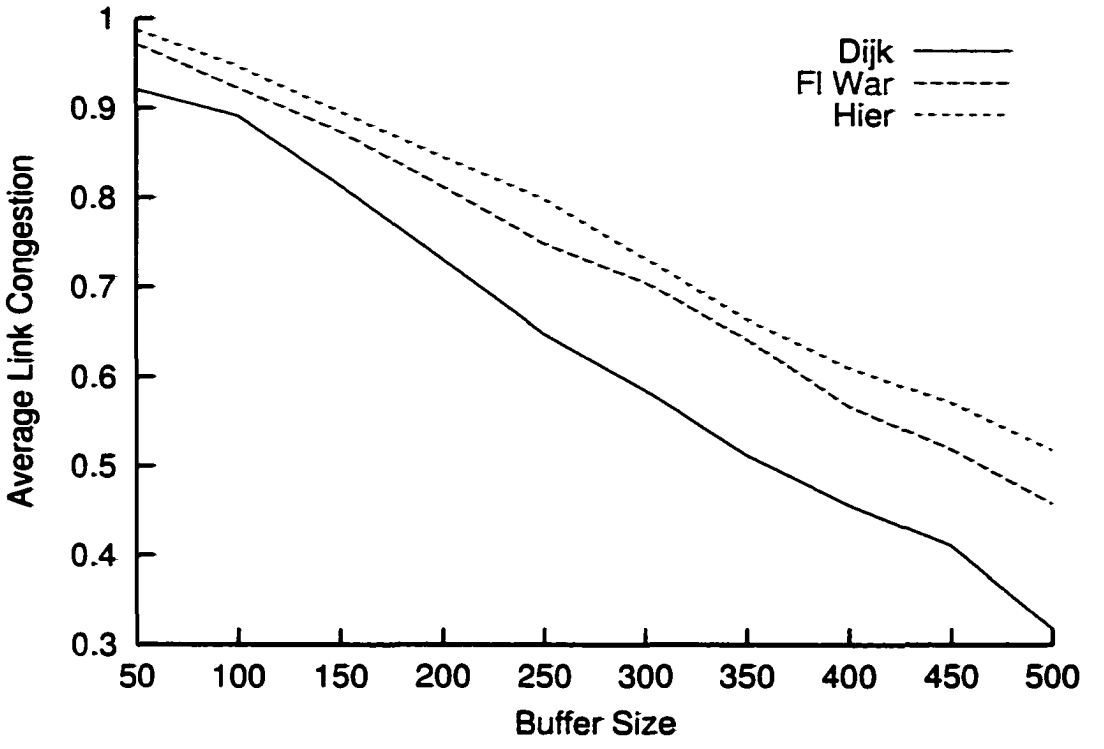

Figure 52. The average link congestion in relation to the buffer size in rings.

are the normal distribution, the constant distribution and the random distribution. More specifically, in the normal distribution, the frequencies of communication between any pair of sites follow a bell curve. However, the generation time offsets of these frequencies are selected randomly. On the other hand, the peaks of these bell curves have fixed value. Since we use random offsets, the global message generation curve over time appears to be random and this is due to the different phases used for the pairwise data generations.

Regarding the constant data distribution, every pair of sites generates a

constant amount of traffic for a certain period of time. However, the time offsets of these pairwise communications are still randomly selected. We believe that, this model is the closest to a realistic one since, traffic demands between network sites follow random bursts requiring a certain amount of bandwidth.

Finally, the random distribution generates random instant bursts of pairwise traffic.

For this experiment, we fixed the bandwidth to 50 cells per time unit and the buffer size to a capacity of 250 cells. The variable was the peak of the bell curves which ranged from 20 to 70.

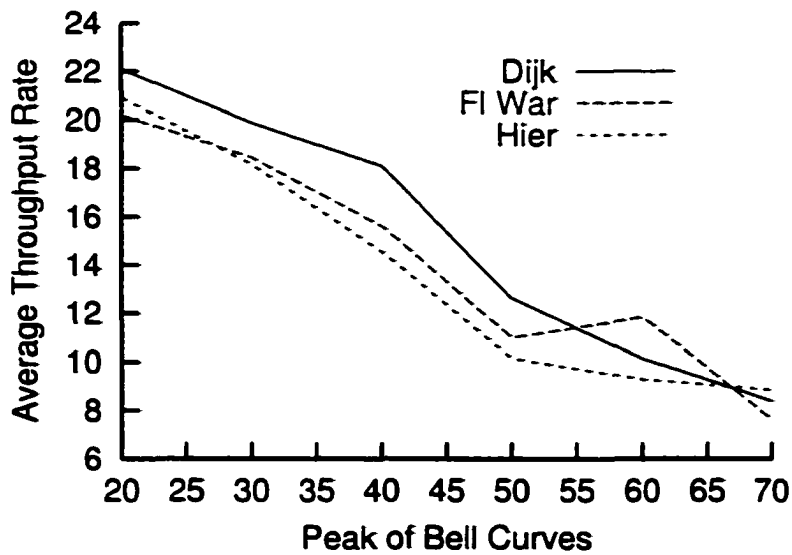

Figure 53. The average throughput rate in relation to the bell curves peak of a normal distribution.

In Figure 53 we illustrate the effect of the bell curve peak value to the av-

erage throughput. As the figure shows, we have a drastic decrease of throughput as the height of the peaks of the bell curves increases. When the peak height reaches the bandwidth value, the average throughput drops dramatically. This is due to the amount of traffic generated and the inability of the system to accommodate it. All buffers appear almost full and the network is consumed to keep track of the statistics and check if a rerouting is in order. However in that case, no rerouting will occur since, there is not a significant amount of under-utilized buffers. Regardless the fact, the rerouting decision process idles the system and therefore contributes negatively to the throughput rate.

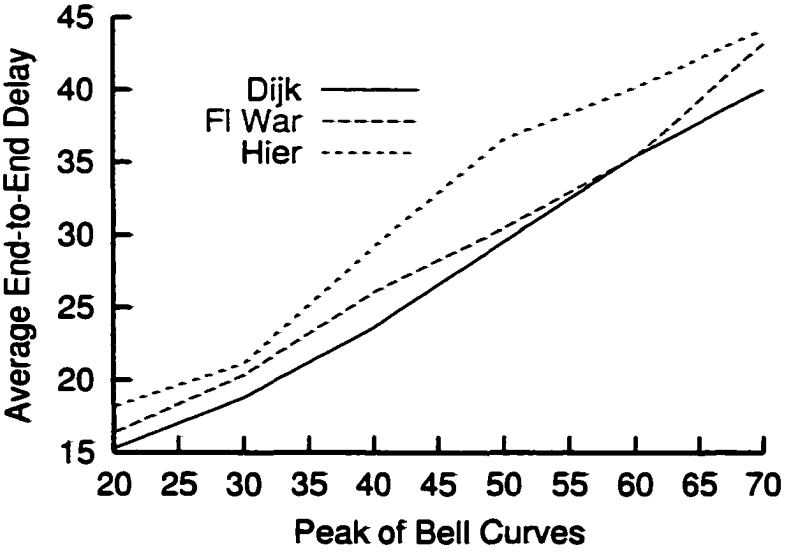

Figure 54. The average end-to-end delay in relation to the bell curves peak of a normal distribution.

Moreover, the end-to-end delay increases as the height of the peaks of

the bell curves increases. This is illustrated in Figure 54. For the same reason explained in the throughput case, the average end-to-end delay increases due to the mass cell generation and consumption of the system to check for reroutings.

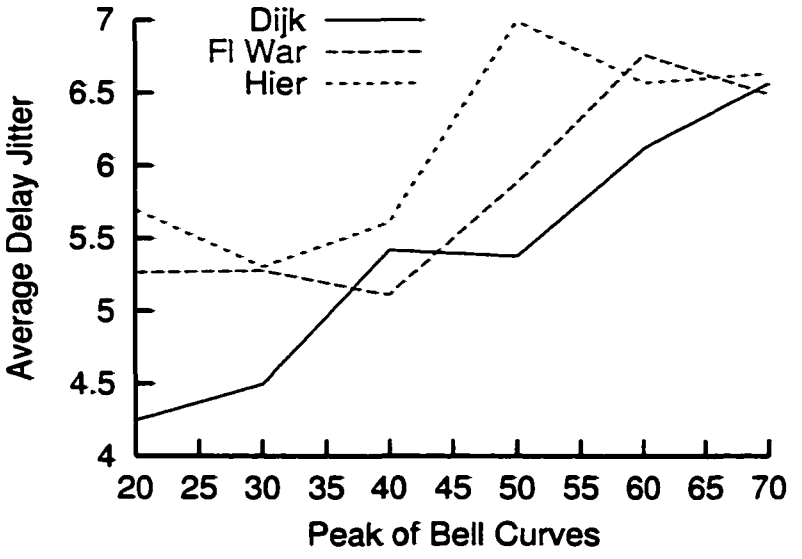

Figure 55. The average delay jitter in relation to the bell curves peak of a normal distribution.

The average delay jitter is also increased. This increase is shown in Figure 55. For buffers of size 250 and bandwidth of 50, someone would expect the average delay jitter to be in the range of 5 for full buffers. However, the additional time spent for rerouting checking increases that average.

The average number of reroutings is almost double than the averages taken during simulations of random data distributions with variable bandwidth and variable buffer size. The refer to Figure 56 for our claim. The mass amount

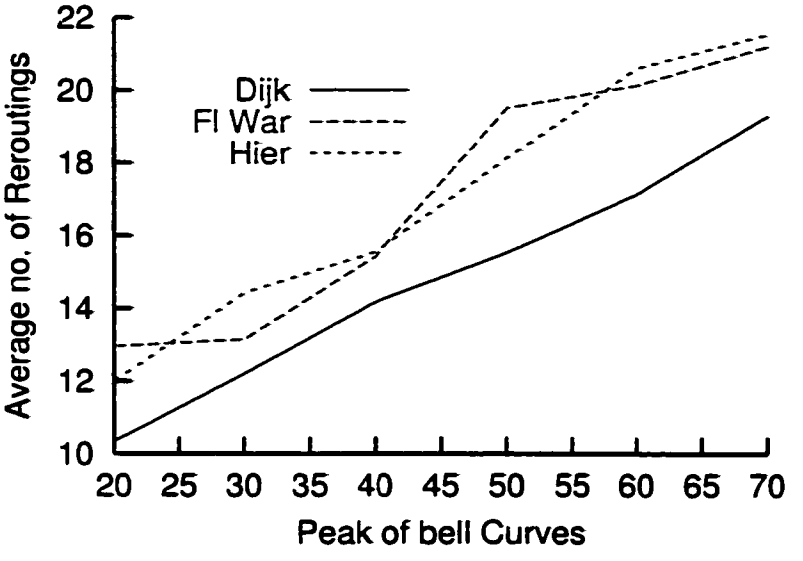

Figure 56. The average number of reroutings in relation to the bell curves peak of a normal distribution.

of cells produced by this data distribution causes more reroutings, especially in the beginning of this global congestion process. As all buffers start to fill up and eventually get congested, no more reroutings appear but, the amount of the initial ones is big enough to contribute to the average.

Figure 57 illustrates the congestion experienced for the three heuristics. The figure shows an increase of congestion as the height of the bell curve peaks increases. However, even for small height peaks, the congestion of the network remains at a considerable level. This is because, the buffer size is fixed and the bandwidth availability is not enough to accommodate the traffic produced.

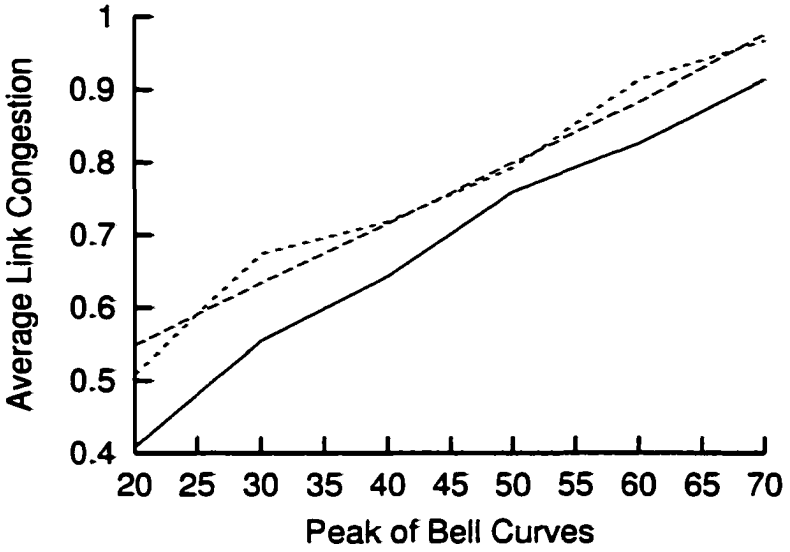

Figure 57. The average link congestion in relation to the bell curves peak of a normal distribution.

The effect of constant data distributions for the heuristics in rings

As we mentioned earlier, constant distributions are made by fixing the height (the demand of bandwidth), of every communication between any pair of nodes. However, the durations of these requests are randomly selected as well as the offsets of their starting point. This creates a global message generation rate which is close to a random frequency distribution. Moreover, this model is the closest to the real network traffic behaviors. We fixed the bandwidth and the buffer size in each simulation to see the effect of the height of peaks to the throughput, end-to-end-delay, delay-jitter, number of reroutings and average link

congestion. The range of these height peaks is again taken between 20 and 70. Also in this case, the bandwidth of the system is set to 50 and the buffer size to 250.

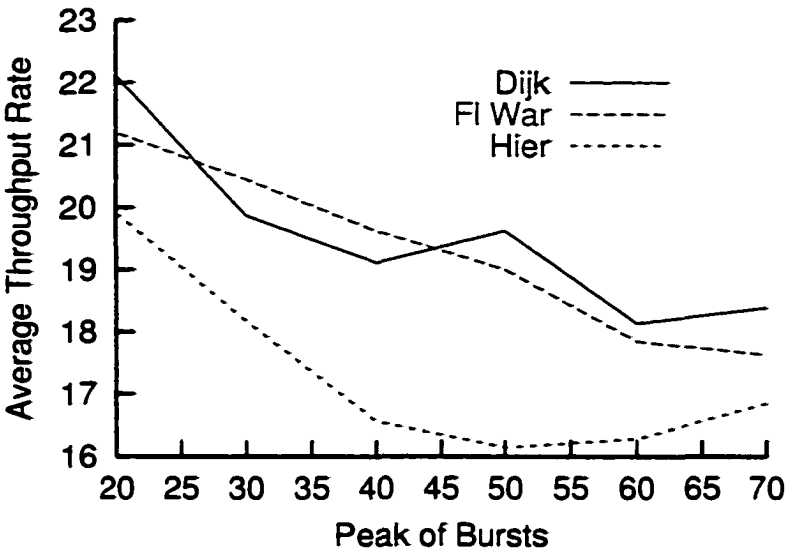

Figure 58. The average throughput rate in relation to the peak heights of a constant distribution.

As Figure 58 depicts, we see a gradual drop of the throughput curves at about 20% in total as the peak of the constant distributions increases. This is justified due to the increase of the data generated and the inability of the buffers to handle all messages. The message rejection rate increases and the throughput drops.

On the other hand, we see a gradual increase of the average end-to-end delay of messages as the peak of the constant distributions increases, (Figure 59).

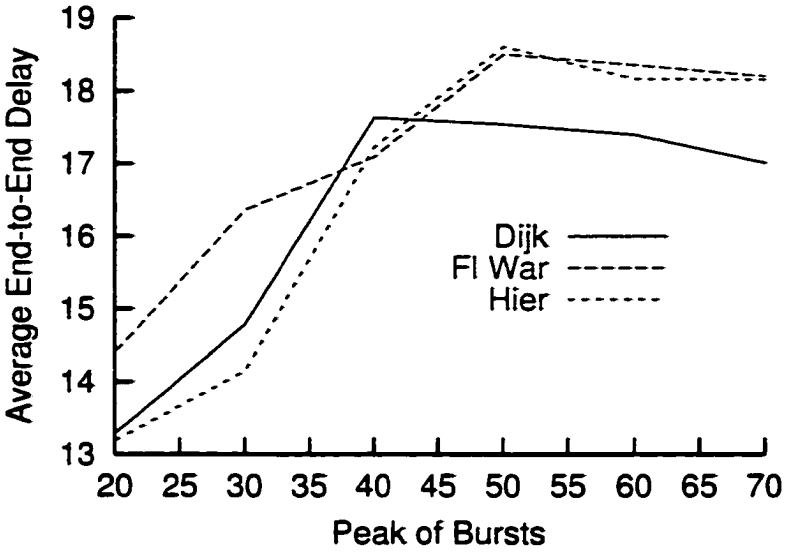

Figure 59. The average end-to-end delay in relation to the peak heights of a constant distribution.

For constant bandwidth, this is justified since, the message generation increases therefore, messages remain more time in the buffers or they are rejected.

In Figure 60 we see the relation of the average delay jitter to the height of constant distribution peaks. As the height of the peaks reaches the bandwidth value, the delay jitter is almost the same for all three heuristics. However, as the peak of the constant distribution gets a value higher than the bandwidth, the hierarchical and the Floyd-Warshall heuristic perform poorly relatively to the Dijkstra heuristic which does not increase the average delay jitter as much.

Also, in Figure 61, we see the effect of the constant distributions to the average number of reroutings for the three heuristics. As long as the peak of the

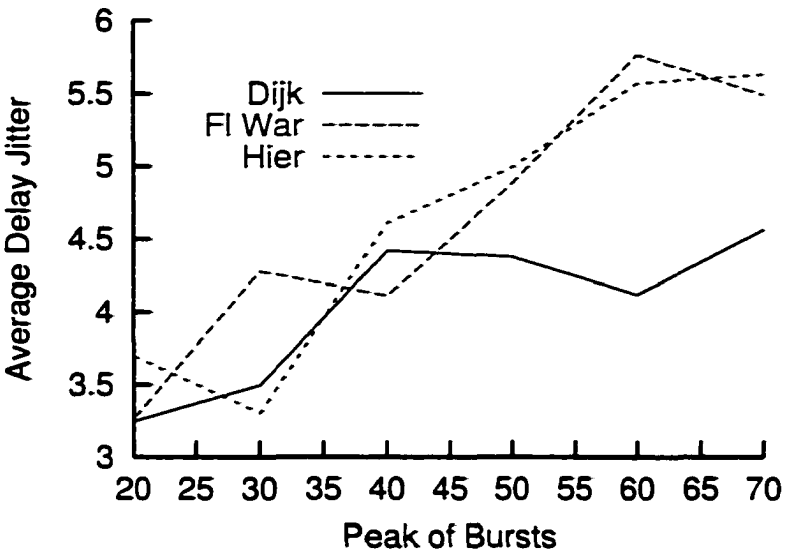

Figure 60. The average delay jitter in relation to the peak heights of a constant distribution.

distribution is small, all heuristics trigger more reroutings. This happens because, not all buffers at this point are equally congested. However, as the peak reaches the value of the system bandwidth, more messages are generated. In that case, almost all buffers are full therefore, more rerouting decisions are negative. That is why we see a gradual drop on the average number of reroutings. We observe in this figure that, the Dijkstra heuristic is more sensitive in triggering reroutings, especially at small peak levels. This means that, the Dijkstra heuristic at small peak levels does not result in a good buffer load balance.

Figure 62 shows the effect of the peak of constant distribution bursts to the average link congestion of the network. As the figure illustrates, we observe

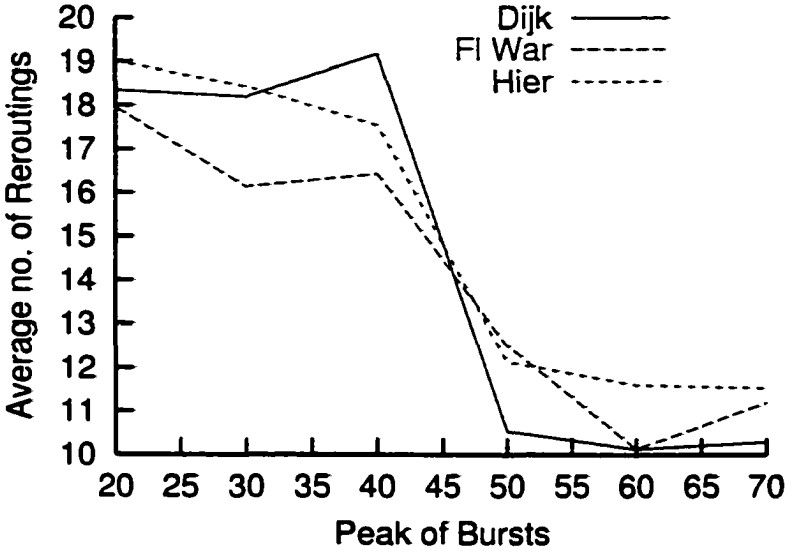

Figure 61. The average number of reroutings in relation to the peak heights of a constant distribution.

an almost sub-linear behavior of the three heuristics regarding the link congestion increase. This behavior is similar to the one observed in the case of normal distributions. In small peak values, the system has the buffer resources to accommodate the generated load. However, the amount of data generated is large compared to the buffer size and the available bandwidth. This is the reason that, we see congestion even in small peak values of the constant bursts.

As the value of the peaks increases, (while the bandwidth and the buffer size remaining constant), we also see and increase in the average link congestion. More messages are accumulated in the buffers and eventually, this amount reaches the value of the upper bound threshold thus resulting into congestion near 1.

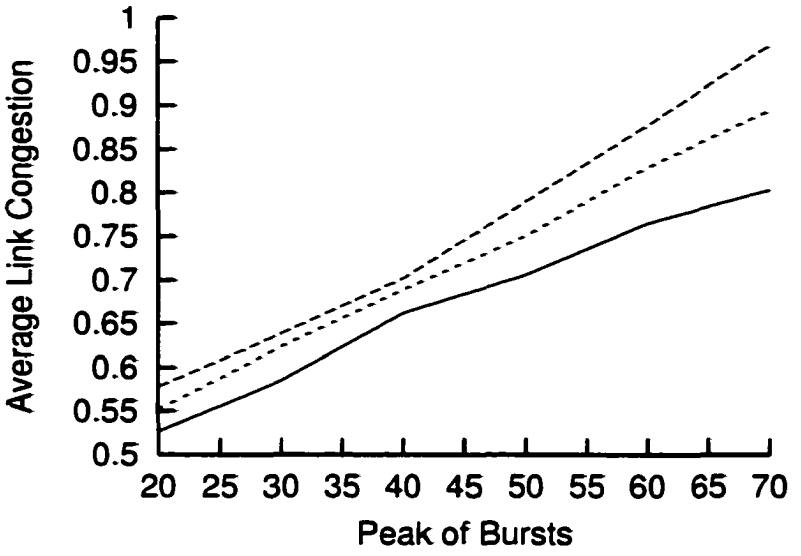

Figure 62. The average link congestion in relation to the peak heights of a constant distribution.

Performance analysis of the heuristics in hyper-cubes

Maintaining the same organization as in the previous section, this section describes how the three rerouting heuristics performed in the case of hyper-cubic topologies.

We include two subsections. The first describes the effect of the bandwidth to the five QoS metrics chosen. This is done while the size of the buffers is fixed to 250 cells. In the second section, we analyze the performance of the heuristics in the case of fixed bandwidth but variable buffer size. We run 100 simulations for each case. Each simulation had duration two minutes with intervals of one

second. We also chose a mix of normal, constant and random distributions. For the two cases half of the simulations are performed on a hyper-cube of 8 sites and the other half on a hyper-cube of 16 sites. However, the statistics are combined to the same curves. As the following figures illustrate, the increase of the network degree from 2 to 3 and 4 increases the average throughput rate also. On the other hand, this increase of degree and number of output buffers decreases the delay, the delay jitter, the number of triggered reroutings and the average link congestion. This general observation holds for both the cases of fixed bandwidth and fixed buffer size.

The effect of bandwidth for the heuristics in hyper-cubes

To see the effect of bandwidth on the throughput rate, we ran simulations changing the bandwidth from 10 to 100 cells per second. Since the links are half duplex this bandwidth is shared among the two buffers associated with each link. In the ideal case of no collision detection delays and no message rejections, the system must result in an average throughput rate of 50 messages per second. As Figure 63 shows, we almost reach this ideal situation for the case of the Dijkstra heuristic when bandwidth reaches the maximum value. The hierarchical heuristic follows closely in performance and the Floyd-Warshall heuristic is the third.

Furthermore, the three heuristics result in an increase of average throughput rate in comparison to the ring topology case. The reason for this increase

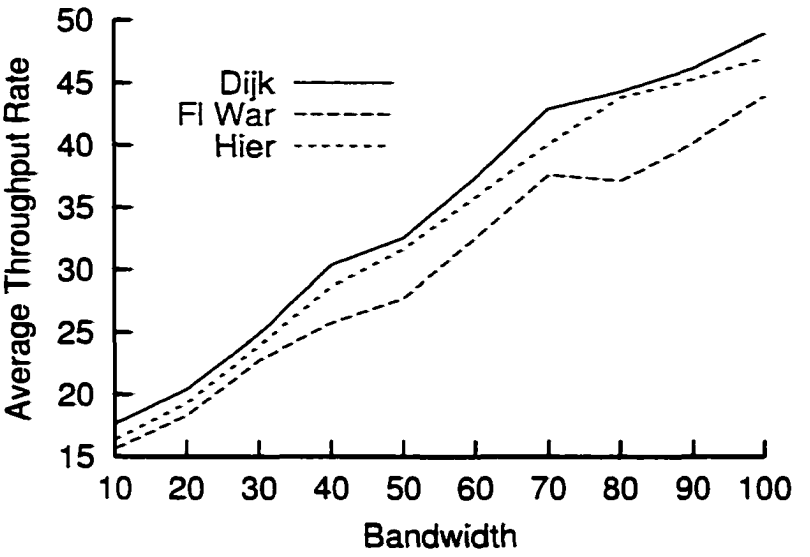

Figure 63. The relation of bandwidth and throughput rate for the heuristics in hyper-cubes.

is the availability of additional buffers therefore, less traffic is forwarded through them. In most of the simulations, the output buffers run half full (about 100 to 120 cells). For some cases, we had delays but the message rejections were relatively few.

In Figure 64, we see the effect of the bandwidth on the average end-to-end delay. For small values of bandwidth, the average end-to-end delay of messages is relatively big. Note that, the diameter of the network is 3 and 4 for the H^8 and the H^{16}, respectively. Therefore in the ideal case, we expect average end-to-end delays in the range of 3.5. However, because of the existence of half duplex links, this ideal average is doubled. As the bandwidth increases, we see a significant

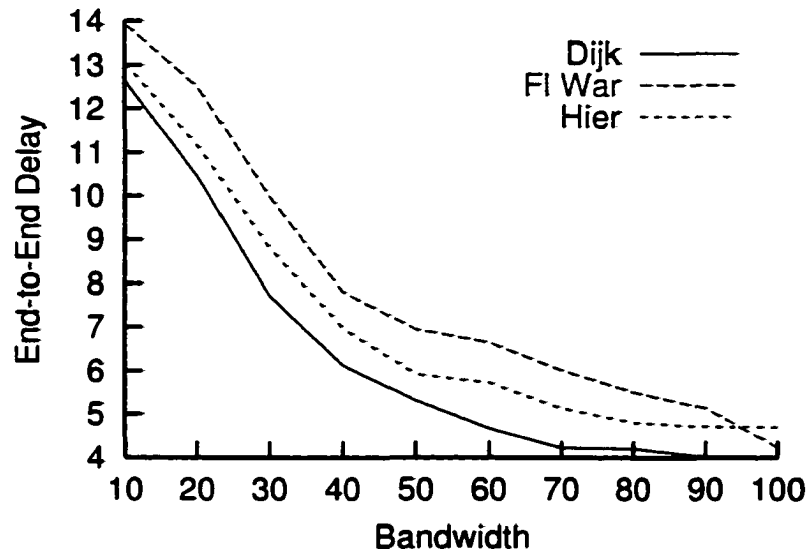

Figure 64. The relation of bandwidth and end-to-end delay for the heuristics in hyper-cubes.

drop on the average end-to-end delay for all heuristics reaching asymptotically the value of 4 seconds. This is a strong indication of a good performance for the heuristics and also an indication that the system utilizes all the available resources to their maximum.

We also ran few simulations with extremely large message generation rates. These simulations were not included in the above set. For these cases, we saw instances of global network congestion.

In Figure 65, we see the effect of bandwidth increase to the average delay jitter for the hyper-cubic topologies. Note again that, the Dijkstra heuristic clearly performs better than the other two. In the ideal case, the average delay jitter must

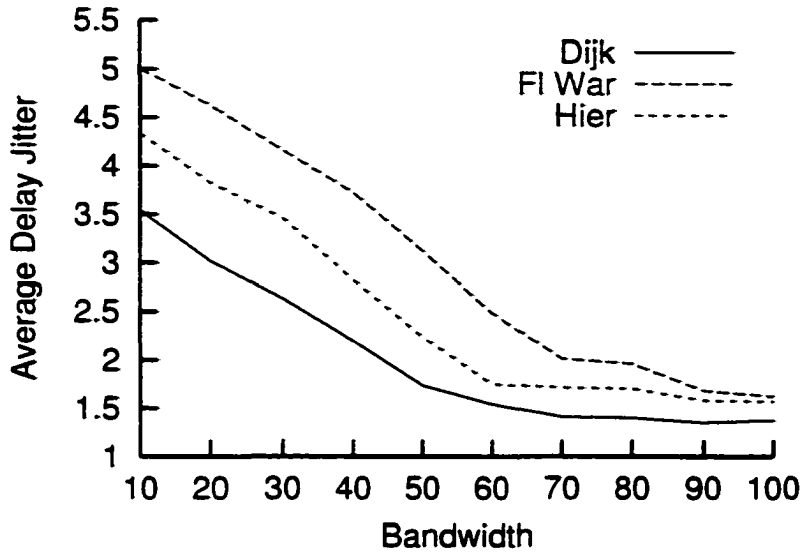

Figure 65. The relation of bandwidth and delay jitter for the heuristics in hypercubes.

be at least 1 for the case of half duplex links. In small bandwidths however, we see delay jitter values quadruple in comparison to the ideal. This is due to the inability of the system to forward traffic because of the small bandwidth available. Messages are delayed staying in the buffers longer. As the bandwidth increases, the delay jitter decreases reaching asymptotically the value of 1.7 for high bandwidths.

The average number of reroutings decreases with the increase of bandwidth available also, (Figure 66). However for small bandwidths, we see a significant value of about 8 reroutings triggered per simulation. This indicates a strong under-utilization of buffers where at the same time, other buffers are congested. The reason is that, all three heuristics are not based on the bandwidth availability

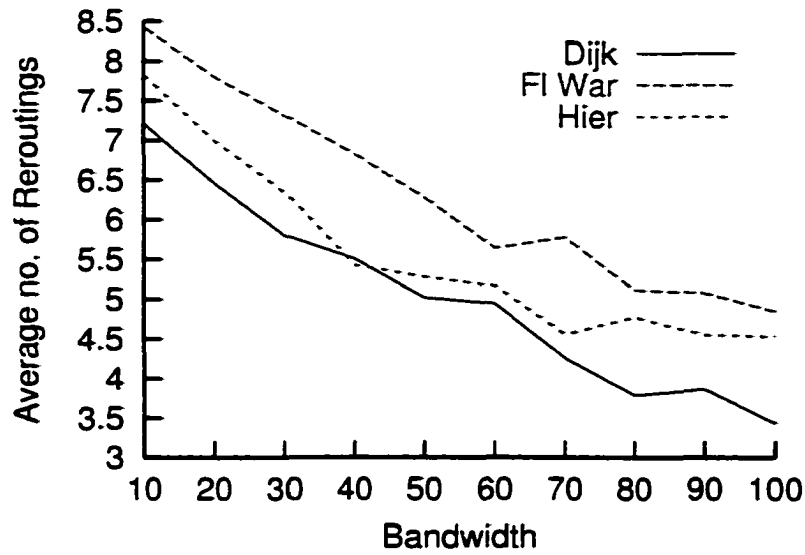

Figure 66. The relation of bandwidth and average number of reroutings for the heuristics in hyper-cubes.

but only on the average traffic demands. As the bandwidth increases, the number of overloaded buffers decreases and therefore, less buffers exceed the the congestion threshold. Thus, the number of rerouting decreases.

In Figure 67 we illustrate the effect of bandwidth increase to the average link congestion in the case of hyper-cubic topologies. We see a significant decrease on the link congestion average when this is compared to the ring topologies. The reason for that is the availability of additional output buffers which share the traffic and therefore reduce the probability of congestion. As this figure shows, no heuristic results in an average buffer load greater than one half of the threshold value set. However as the bandwidth increases, this buffer load also

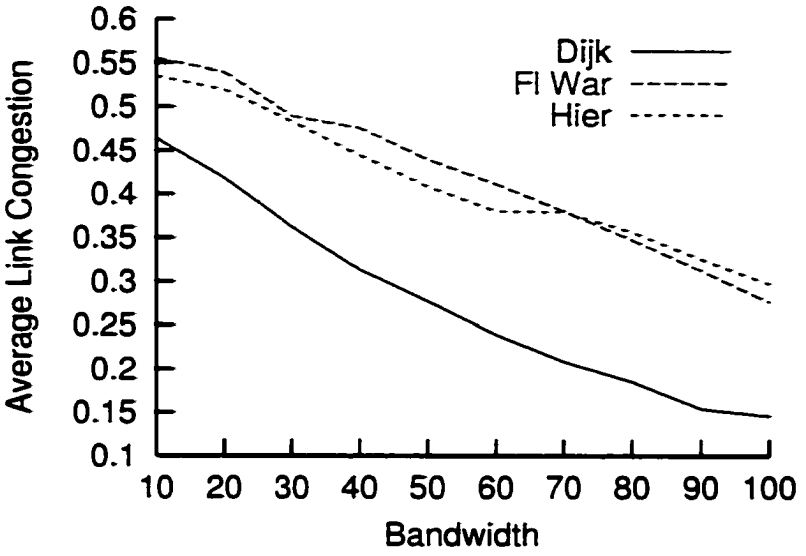

Figure 67. The relation of bandwidth and average link congestion for the heuristics in hyper-cubes.

decreases since, more messages can be forwarded in the same time unit. Thus, the probability of congestion also decreases drastically.

The effect of buffer size for the heuristics in hyper-cubes

As we mentioned in the beginning of this section, we also ran simulations to see the effect of the buffer size to the five metrics. For this experiment, we ran 100 simulations of variable buffer size in the range between 50 and 500 with increments of 50. In all cases the bandwidth was fixed to the value of 50.

As in the case of rings, the global message generation rate here is again

related only to the bandwidth and not to the buffer size. The reason is that, we want to observe the effect of buffer size increase into networks that experience types of traffic with almost equal amount of load but under fixed bandwidth conditions.

As Figure 68 shows, we almost reach ideal throughput rate values as the buffer size reaches 500 for the Dijkstra heuristic. Note that, the ideal case of throughput according to our assumptions is in the range of 25. For the other two heuristics, we see a clear difference on the performance. Note that, this was not visible in the case of rings. The reason iss the peculiarity of the ring topology where, paths can only follow two different directions.

The measurement of the end-to-end delay drop in relation to the buffer size increase is shown in Figure 69. Note that, this is true for all three heuristics. When we increase the buffer size, we really minimize the rejection rate of messages. Furthermore, in the case of fixed bandwidth we also minimize the end-to-end delay. The performance of the Dijkstra heuristic in this case remains superior. However, the other two heuristics perform closely enough so that, there is no clear indication for their comparison.

Figure 70 shows the average delay jitter drop in relation to the buffer size increase. Again the Dijkstra heuristic is the superior in performance among the three.

The drop of the average number of reroutings relatively to the buffer size

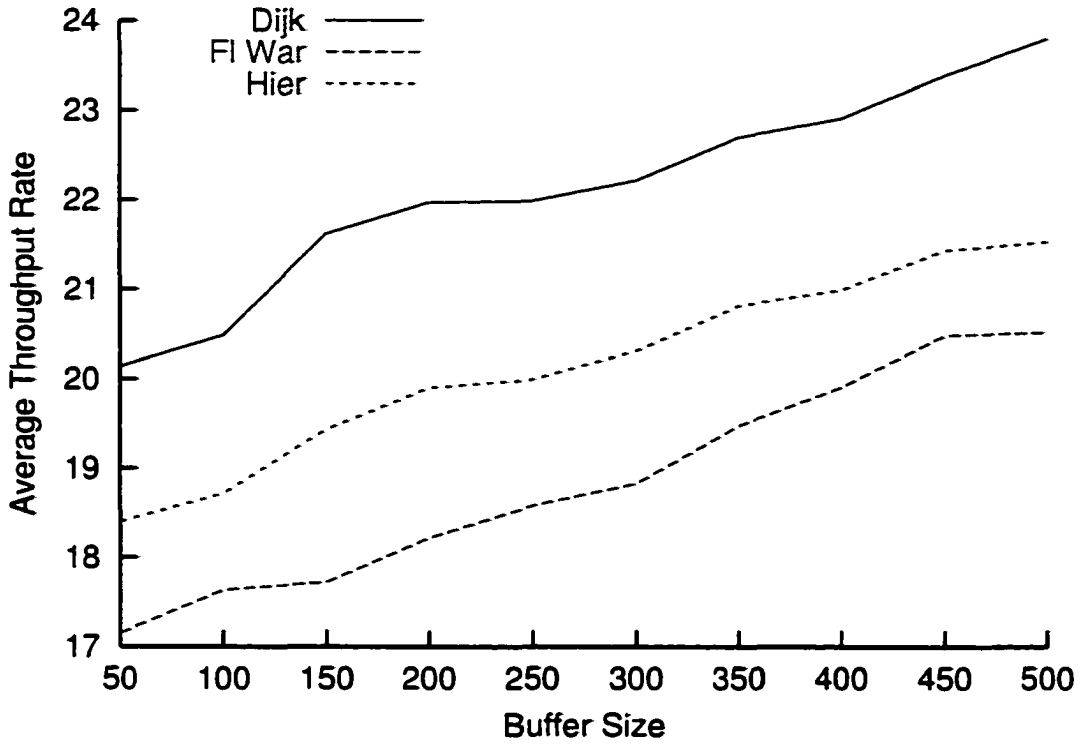

Figure 68. The relation of buffer size and average throughput rate for the heuristics in hyper-cubes.

increase is shown in Figure 71. As the buffer size increases, less messages are rejected. For distributions which do not include extreme message generation rates, this buffer size suffices for the system to run without reroutings. However, small buffers get filled up easier. This triggers the congestion detection algorithm and results into more reroutings.

Finally in Figure 72, we illustrate the drop of congestion as the result of the buffer size increase. We observe a better response of the network system in

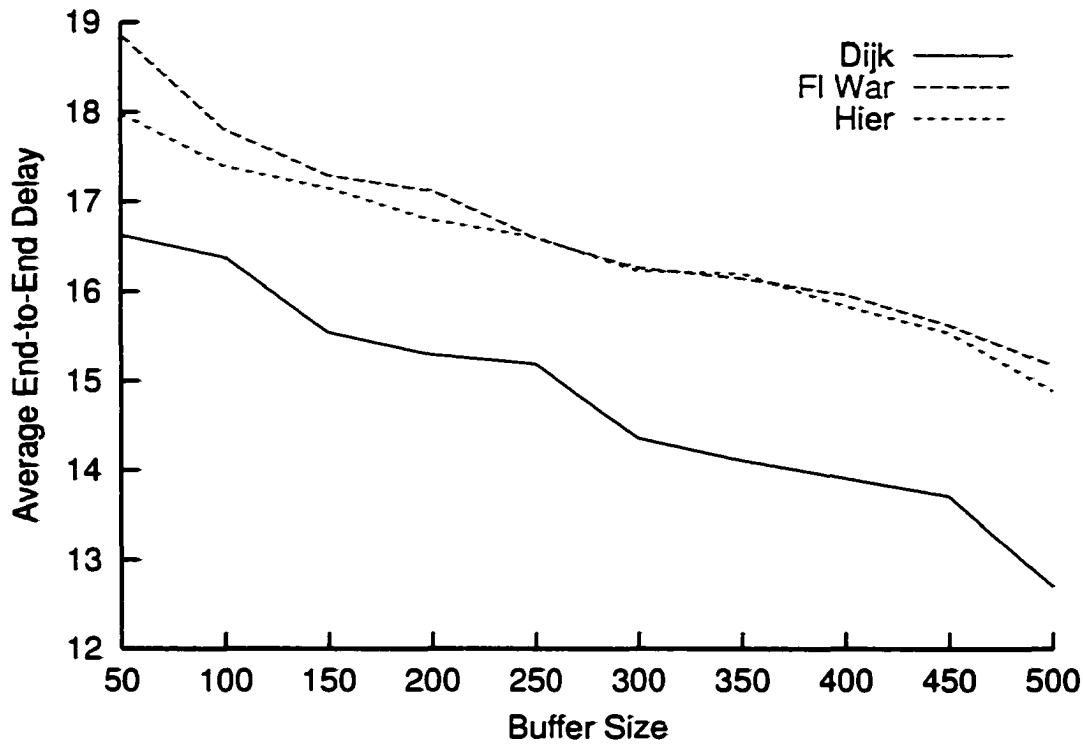

Figure 69. The relation of buffer size and average end-to-end delay for the heuristics in hyper-cubes.

dropping congestion as opposed to the case of ring topologies. This is expected since, the node degree of the regular network also increases therefore, increasing the number of available buffers. For the same load simulations, we expected to see a decrease of the average congestion occurred almost by half. For small values of buffer sizes however, this did not occur. As the buffer size increases, we reach average congestion of 0.17 which is close to the expected drop.

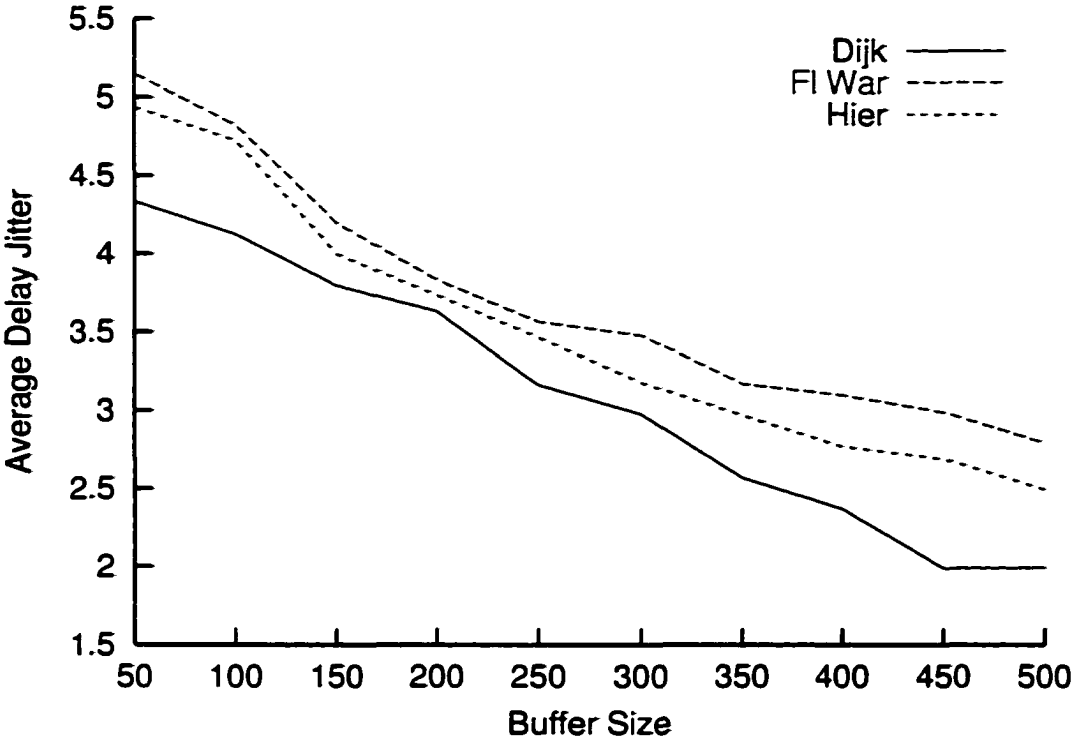

Figure 70. Relation of buffer size and delay jitter for the heuristics in hyper-cubes

Performance analysis of the heuristics in Z-Cubes

The Z-Cube topology was the last topology for which the three heuristics were run. The Z-Cubes are a similar topology to the hyper-cubes. However, the Z^{16} is a network that has degree 3 where the corresponding hyper-cube H^{16} has degree 4. We picked the Z-Cube topology for its close relation to the hyper-cubic topology and to justify the effect of degree and buffer size to the five QoS metrics.

In this case, we follow the same experimentation methodology as in the

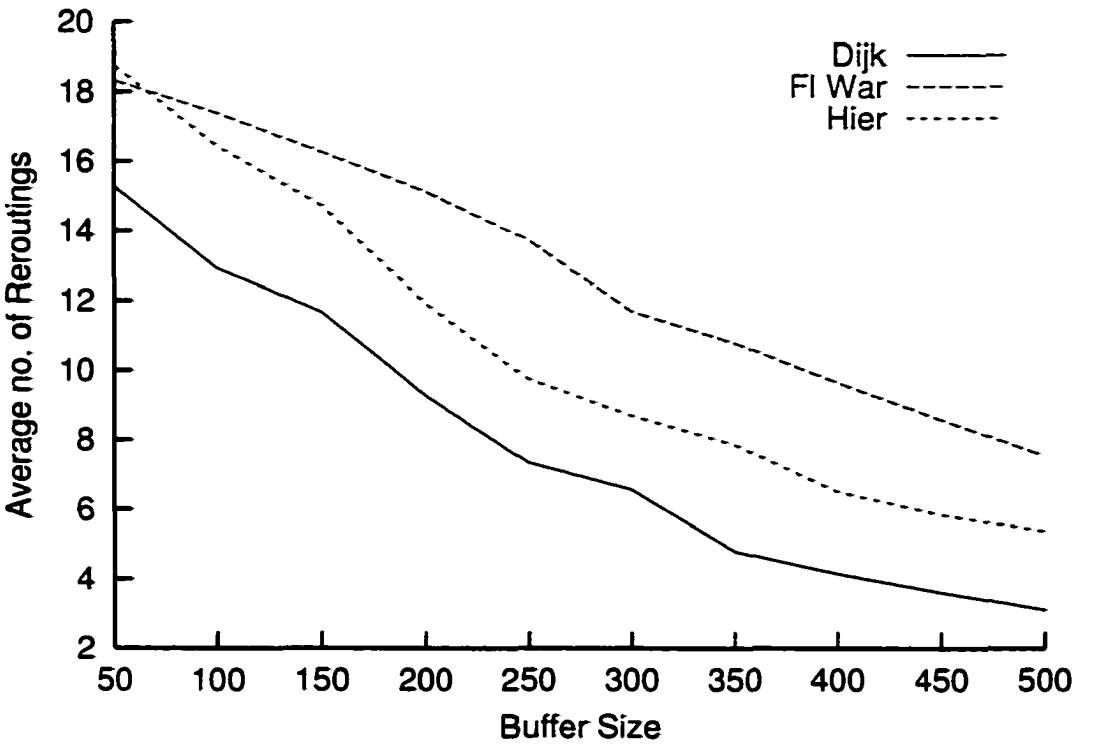

Figure 71. The relation buffer size and average number of reroutings for the heuristics in hyper-cubes.

hyper-cubic topologies. Therefore, we organize this section into two subsections. Similarly, the first subsection investigates the effect of bandwidth increase and the second subsection the effect of buffer size increase to the metrics.

The effect of bandwidth for the heuristics in Z-Cubes

For the case of bandwidth we ran 100 simulations in a Z^{16} topology. The duration of each simulation was two minutes. We fixed the buffer size at 250

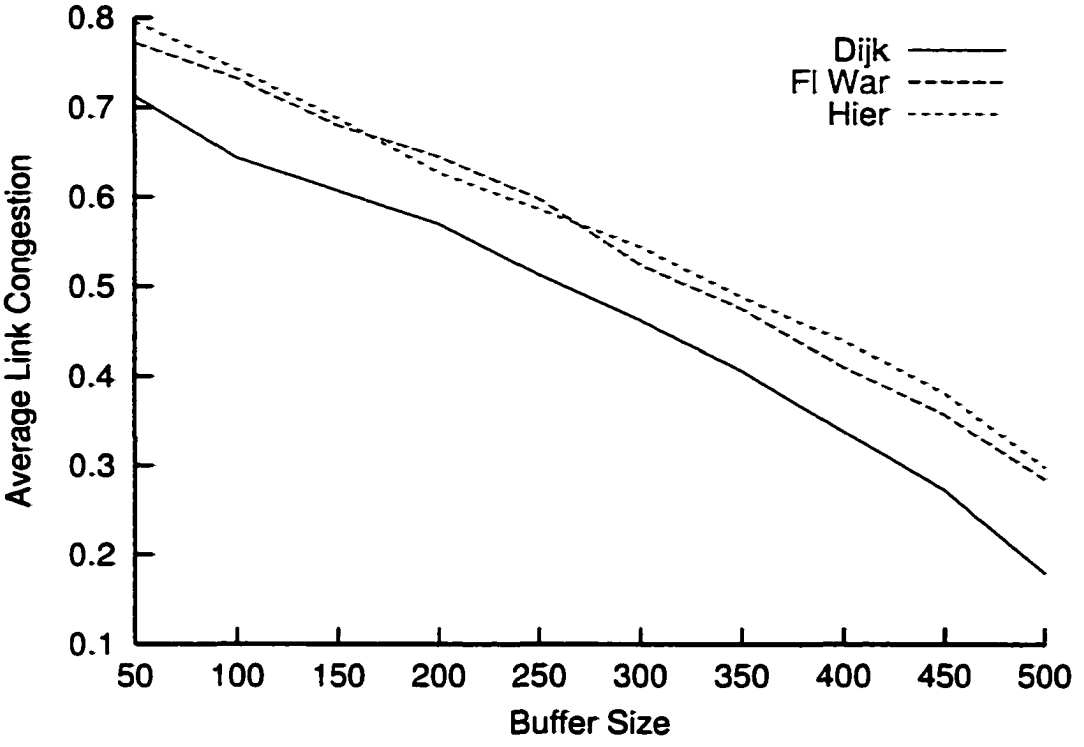

Figure 72. The relation buffer size and average link congestion for the heuristics in hyper-cubes.

messages and we varied the bandwidth from 10 to 100 messages per time unit with increments of 10. The distribution of the communication frequencies used for this experiment was the same mix as in the hyper-cube case.

Figure 73 shows how the bandwidth increase affects the increase of the average throughput rate in a Z^{16}.

If we compare this figure with the corresponding figure for the case of hyper-cubes we see a small difference in the throughput increase. This is expected since,

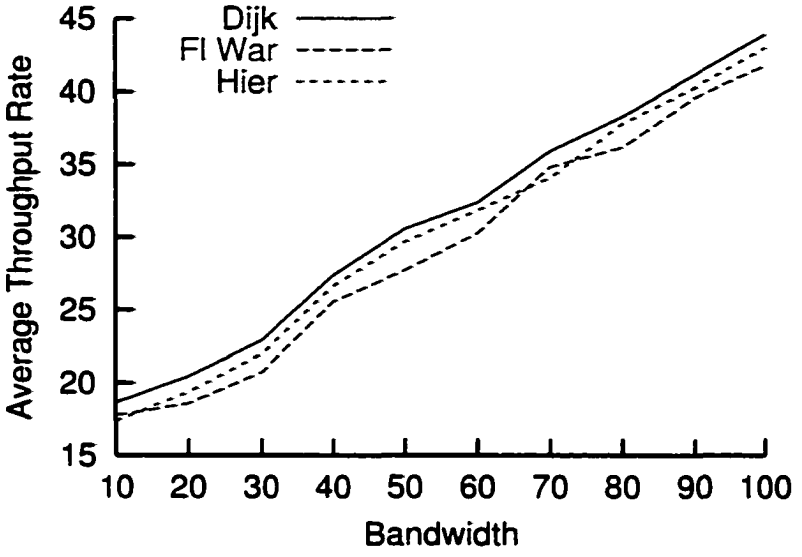

Figure 73. The relation of the bandwidth and the average throughput rate for the heuristics in Z-Cubes.

the degree of the Z^{16} is 3 whereas the mix of hyper-cubic topologies degrees were 3 and 4. Moreover, the Z-Cube topology results in less buffers. This justifies the performance degradation of the three heuristics.

Figure 74 shows how the bandwidth increase affects the drop of the average end-to-end delay in the case of Z-Cubes.

A similar comparison as in the case of the throughput justifies the performance of the three heuristics. As Figure 74 illustrates, the decrease in the end-to-end delay is not as drastic as in the case of hyper-cubes. This is more obvious for small values of bandwidth. In the case of hyper-cubes, we see a drastic drop in the average end-to-end delay when bandwidth ranges between 30 and 40

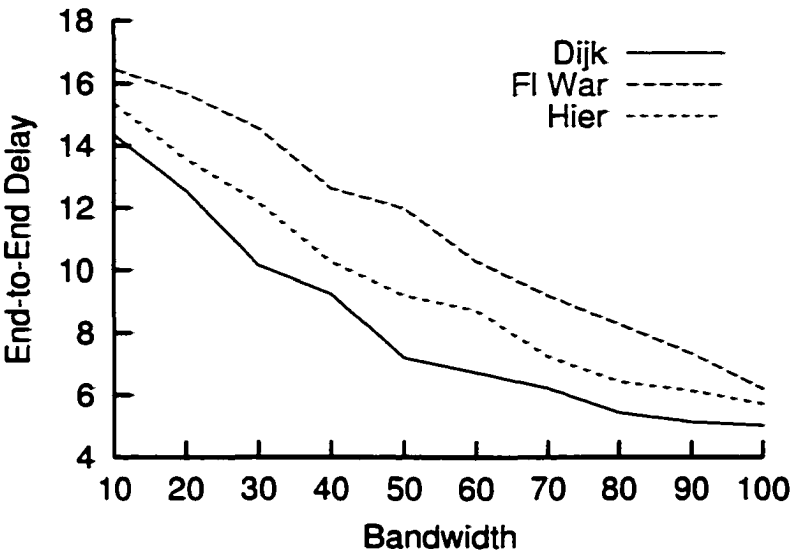

Figure 74. The relation of the bandwidth and the end-to-end delay for the heuristics in Z-Cubes.

cells. Even though both cases reach delays of 4 when bandwidth is 100 cells, the hyper-cube curves behave as inverse exponential. However, this is not the case of the Z-Cube topology where, the curves behave as negative slope linear.

Figure 75 shows the effect of bandwidth increase to the average delay jitter drop.

The same observation as in the end-to-end delay holds also here. A comparison of the heuristics to the hyper-cubic case shows that, the effect of bandwidth is less visible in the Z-Cubes. This is due to the number of available buffers. The Dijkstra heuristic curve in the Z-Cube case asymptotically reaches the jitter value of 2 as opposed to 1.5 in the hyper-cube case. Moreover, the hyper-cube

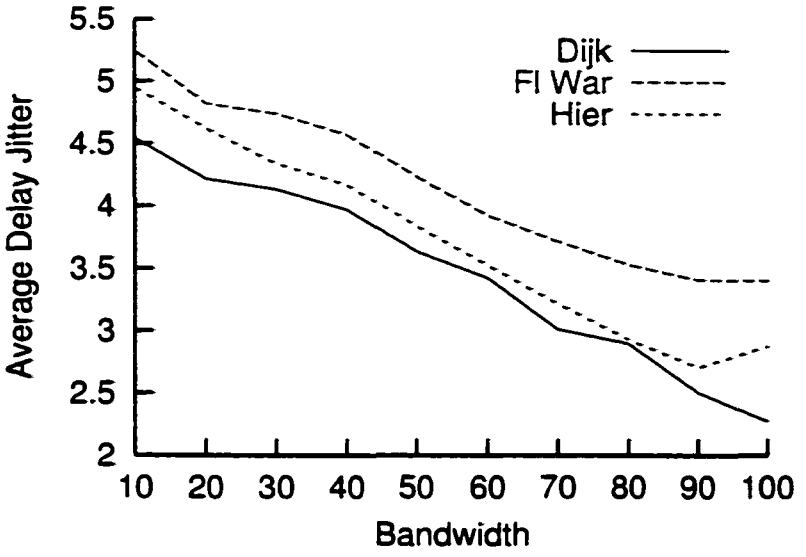

Figure 75. The relation of the bandwidth and the delay jitter for the heuristics in Z-Cubes.

curves behave as inverse exponential with drastic delay jitter decrease when the bandwidth ranges between 35 and 60. However in the Z-Cube case, all heuristic curves show a negative slope linear behavior.

The effect of bandwidth increase to the average number of reroutings is illustrated in Figure 76.

The three heuristics behave in a similar manner to the hyper-cubic case. However, a small degradation in performance exists. This is justified by the existence of fewer available buffers to forward the traffic to destination sites. We experienced a greater number of reroutings for the case of small bandwidths also. In this case, the probability to have unbalanced buffers increases. As the band-

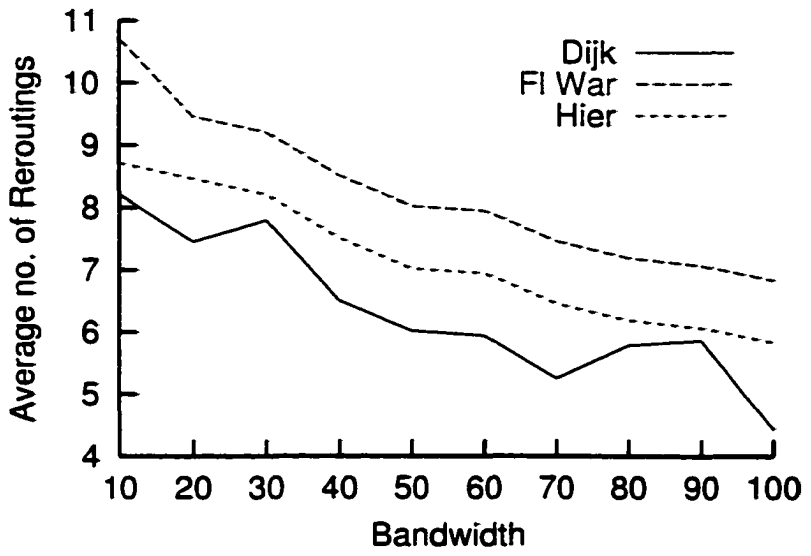

Figure 76. The relation of the bandwidth and the average number of reroutings for the heuristics in Z-Cubes.

width increases, fewer buffers are congested such that, the threshold number of congested buffers is reached with greater difficulty.

Finally in Figure 77, we illustrate the effect of bandwidth increase to the average link congestion. We observe that, the drop of average link congestion is less than the one occurring in the hyper-cube case. This was expected since, fewer buffers are available in the topology. In that case, we see an interchange in performance between the hierarchical and the Floyd-Warshall heuristic since, the two perform very close. This interchange may also be explained by the idiosyncrasies of the traffic distributions run.

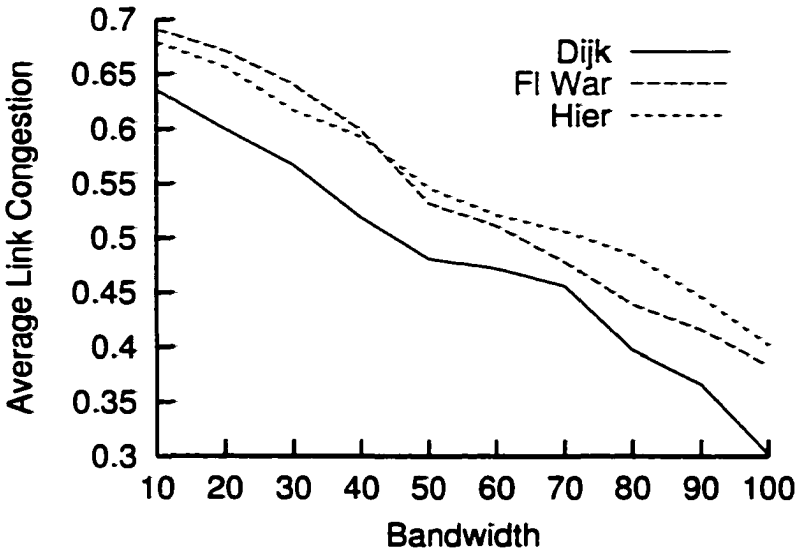

Figure 77. The relation of the bandwidth and the average link congestion for the heuristics in Z-Cubes.

The effect of buffer size for the heuristics in Z-Cubes

As in the hyper-cubic topology case, we ran 100 simulations with fixed bandwidth of 50 cells per time unit. The range of buffer size varied from 50 to 500 with increments of 50. The mix of communication frequency distributions was the same as in the hyper-cubes.

As in the case of rings and hyper-cubes, for this experiment we related the global message generation rate to the fixed bandwidth. Therefore, for all the simulations, we generate somewhat similar traffic. The reason is that, we wanted to isolate the effect of the buffer increase to the system behavior and observe its

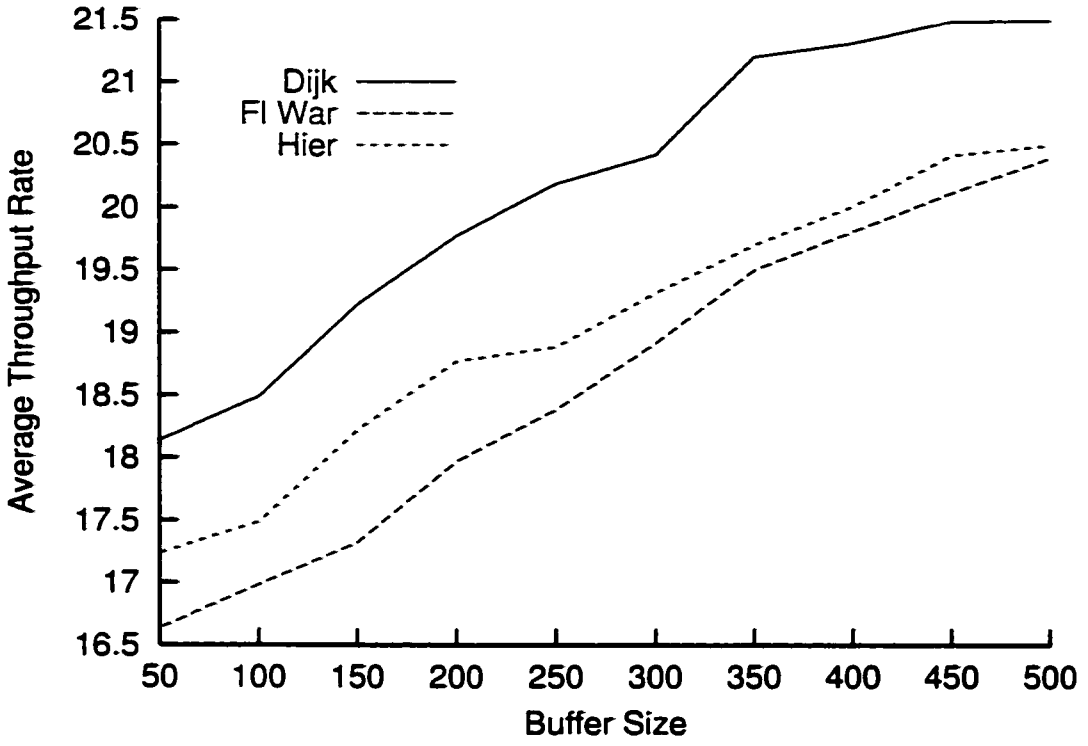

Figure 78. The relation of the buffer size and the average throughput rate for the heuristics in Z-Cubes.

response under fixed bandwidth conditions.

Figure 78 illustrates the relation of buffer size increase to the average throughput rate increase. For bandwidths of 50 cells and half duplex links we expected an ideal throughput rate of 25 cells per time unit.

For the case of the Dijkstra heuristic, we reached throughput rates of 21.5 cells per time unit on the average as the figure illustrates. The hierarchical heuristic performed with a 10% degradation compared to the Dijkstra heuristic. On

the other hand, the Floyd-Warshall heuristic performed with a 20% degradation compared to the Dijkstra heuristic. Their overall performance is somewhat decreased in comparison to the hyper-cubes and it is justified with the existence of fewer available buffers. However, we can never be able to reach the ideal average throughput rate since, the initial transient effect of the first in order messages contributes negatively to the throughput calculation.

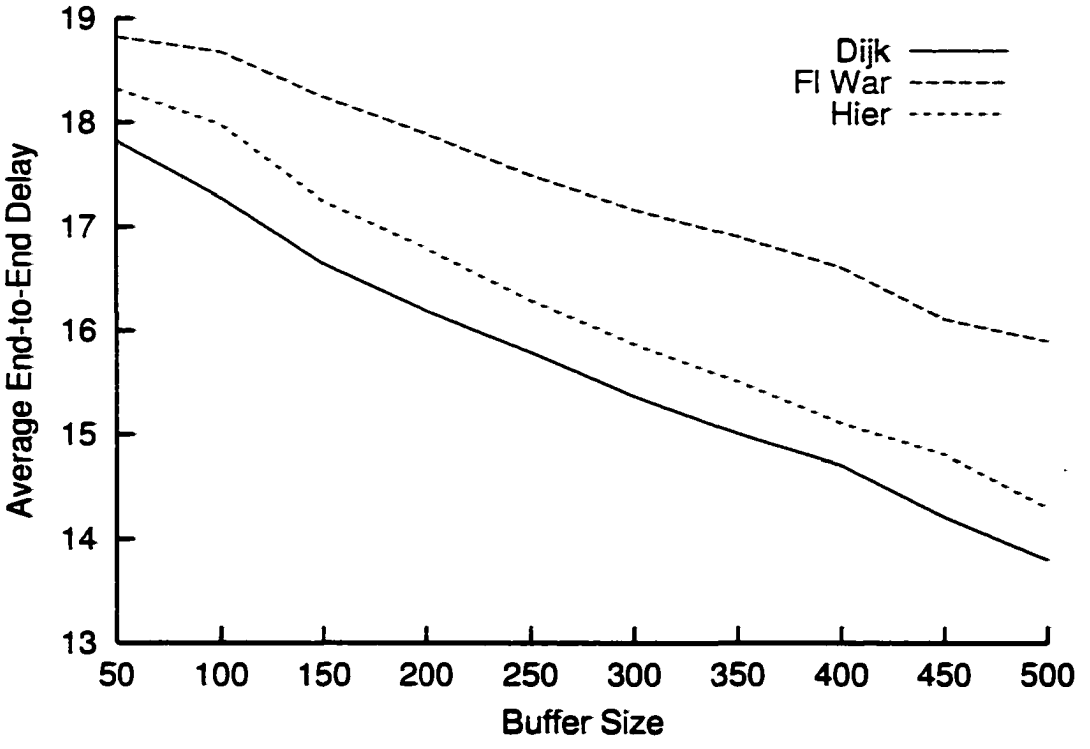

Figure 79. The relation of the buffer size and the average end-to-end delay for the heuristics in Z-Cubes.

The exact performance comparison for the case of the average end-to-end delay is illustrated in Figure 79. The Dijkstra heuristic achieves an almost negative

slope linear performance even though it is somewhat less than the hyper-cubic case. The hierarchical heuristic resulted in a 6% on the average more delays than the Dijkstra heuristic and finally, the Floyd-Warshall heuristic followed with an average of 15% more delays than the Dijkstra. However in this case, the three heuristics perform very close when compared with the hyper-cubic case.

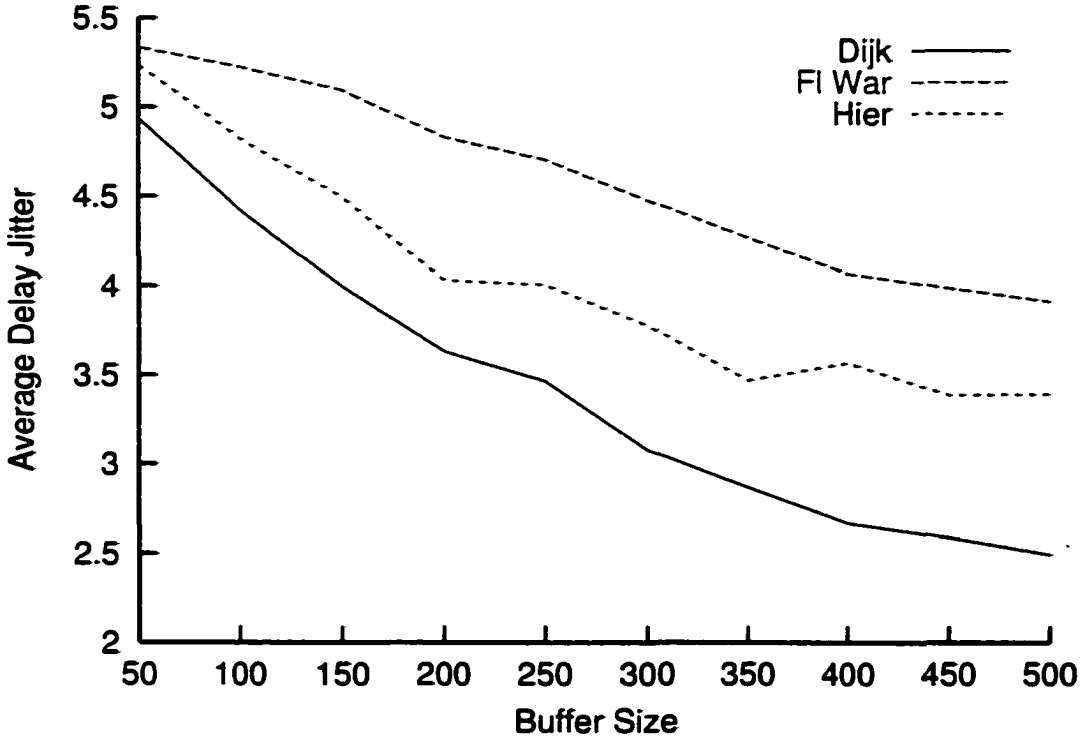

Figure 80. The relation of the buffer size and the delay jitter for the heuristics in Z-Cubes.

Figures 80 and 81 illustrate the relation of the buffer size increase to the decrease of the average delay jitter and the average number of reroutings, respectively. These figures reassured the superiority of the Dijkstra heuristic among the

three. If we compare the corresponding curves from the hyper-cubic case we see a degradation of about 30% in performance for the Z-Cube as we reach buffer sizes of 500 cells. Among the two criteria, the average number of reroutings is affected more.

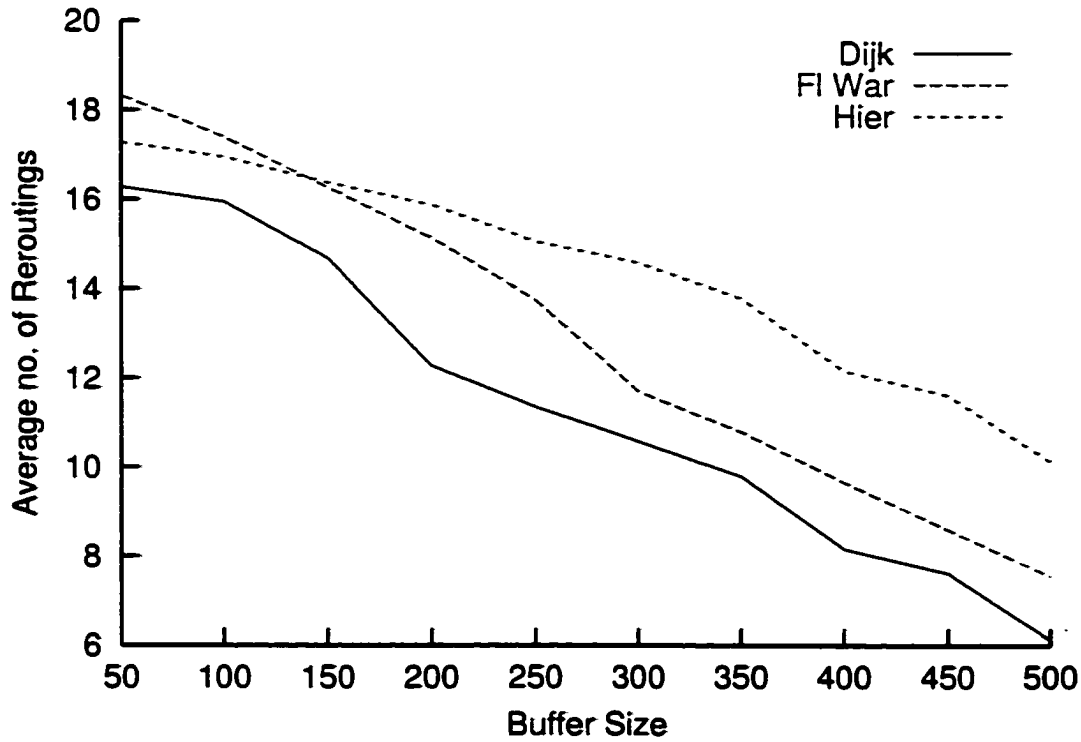

Figure 81. The relation of the buffer size and the average number of reroutings for the heuristics in Z-Cubes.

These observations are justified by the inability of the system to accommodate similar traffic volumes while we decrease the number of available path options and the number of available buffers.

The decrease in the number of buffers concentrates more traffic to the

available ones therefore, increasing the probability of message rejection and network congestion. Moreover, this message rejection rate increase results in a higher average delay jitter and higher average number of reroutings triggered.

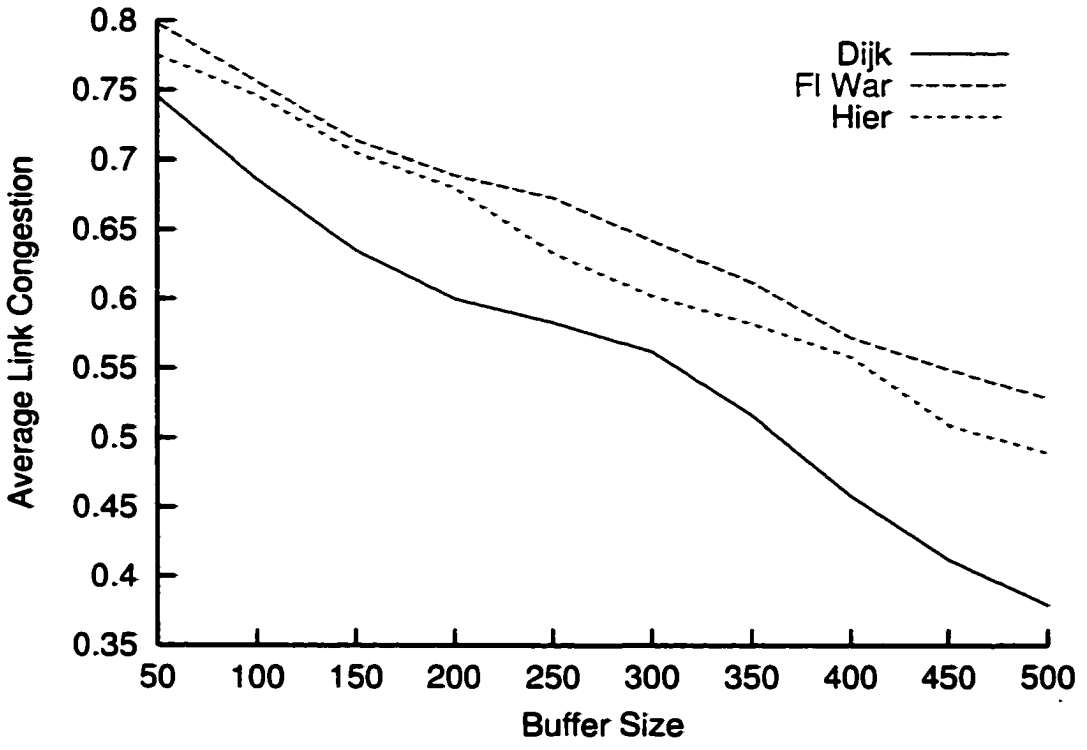

Figure 82. The relation of the buffer size and the average link congestion for the heuristics in Z-Cubes.

Finally, in Figure 82 we illustrate the decrease in the average link congestion in relation to the increase of buffer sizes. Comparing the corresponding curve from the hyper-cubic case we observe a similar behavior here also. Under similar traffic volume distributions and under fixed bandwidth conditions, the buffer increase reduces the message rejection rate providing the ability for the buffers to

contain higher volumes of traffic. This also increases the congestion degradation zone under which the system tolerates congestion.

CHAPTER VII

CONCLUSIONS AND FUTURE RESEARCH

Conclusions

In this dissertation, we formulated the problem of designing and routing a regular network topology to balance network traffic. The goal was to reduce the probability of congestion. Furthermore, we introduced several heuristics to approximate link load balancing for the case of networks that may or may not guarantee Quality of Service.

We have shown that the optimization function $V \times C_a^2 \times L_m^2$ characterizes link load balancing while reducing the average communication cost and the maximum buffer load among all output buffers. More specifically, we introduced two theoretical problems:

The first refers to the designing of a regular network topology and the finding of an appropriate routing scheme for that topology in order to minimize the optimization function $V \times C_a^2 \times L_m^2$. The input to the problem is the number of sites and the traffic requirements for the pairwise communications between these sites. These pairwise communications are given as expected communication frequencies over a time period.

The second problem assumes that, the network topology is given but, the traffic behaviors change so that, a rerouting is needed to balance the load of the links.

Both of the above problems have been proven to be NP_Complete.

To prove the NP_Completeness of the first problem we took a special case of traffic requirements that satisfy the triangulation property and proved that it is NP_Complete to design a ring that minimizes $V \times C_a^2 \times L_m^2$ for these requirements. The proof involved several lemmas for the investigation of all permutations of sites in the ring and all types of routing schemes that could be applied on these permutations. We finally showed that, the permutation and the routing that minimizes $V \times C_a^2 \times L_m^2$ under the imposed restrictions is the one that corresponds to the closest partition of the traffic requirement set.

To prove the NP_Completeness of the second problem we mapped the well known 3DM problem to a regular topology for which all traffic demands emanate from only two sites.

We also provided an integrated network module that reduces the probability of congestion with a mechanism that measures the traffic in each link, keeping track of the system localities through an exponential average formula. This creates a congestion avoidance control which triggers network reroutings so that, under-utilized links share the additional load from over-loaded links. For that reason, we measured the network congestion as the summation of deviations of

link loads from an optimal operating load zone for the system. This procedure is simple and the involved data structures are minimal therefore, easily integrated into the routing table of each network site.

Our theoretical findings showed that, no polynomial routing solution exists to minimize $V \times C_a^2 \times L_m^2$. For that reason, we proposed three approximation algorithms based on the greedy computational method for the case of networks handling traffic with no bandwidth guarantees: The Average Dijkstra, the Average Floyd-Warshall and the Hierarchical algorithm. The three heuristics try to pack routes through all the available output buffers prioritizing the insertion of small length paths. At the same time, the heuristics attempt to reduce the maximum buffer load therefore, approximating a load balance equilibrium.

We have also proven that, the worst case traffic distributions for the above heuristics result in a load variance $V \neq 0$ where at the same time, the corresponding optimal solution results into $V = 0$. For these traffic distributions, there are no performance guarantees for the heuristics since they are all unbounded.

For the worst case traffic distributions that never result into zero load variance, we have found upper computational bounds for the three heuristics. However, this investigation was limited into the ring topologies.

For networks that support applications with bandwidth guarantees, we proposed a hybrid routing model that uses the above static routing heuristics along with a "backtracking like" distributed routing algorithm. The path selec-

tion process first checks if the static routing paths can be used with the bandwidth restriction. This eliminates the site probing by a considerable amount. However, when the static routes do not satisfy the bandwidth condition, a distributed algorithm attempts to establish a session, while at the same time, it tries to minimize $V \times C_a^2 \times L_m^2$ locally in each site.

Finally, we compared the routing algorithms by simulating popular regular network topologies accommodating several traffic distributions. We investigated the effect of bandwidth and buffer size increase to various QoS metrics including throughput, end-to-end delay, delay jitter and number of reroutings occured. We also observed the congestion produced in the network during the application of the different types of routing.

All the experimental results showed that the average Dijkstra heuristic was superior in performance relatively to all QoS metrics.

Future research

In this dissertation we succeeded to set the theoretical fundamentals of the link load balancing problem in regular network topologies. However, several sides of the problem have been left without investigation. In the remaining of this section, we describe some interesting ideas which we believe are worth investigating.

We have observed that, the performance of the proposed algorithms greatly depends on the network bandwidth availability. However, the bandwidth was not

part of the original problem input. We believe that, the bandwidth information for a network is an important factor which may fine tune any load balancing algorithm. Furthermore, we believe that the link bandwidth information may help congestion avoidance models to predict hot spot creations with greater accuracy. We plan to investigate the bandwidth effect in conjunction with the queueing theory findings in order to create an integrated congestion avoidance module that uses traffic localities.

In Chapter IV we proposed an algorithm for triggering network reroutings when congestion occurs. However, the global network rerouting is a time consuming process from which only the congested links may benefit. We plan to investigate procedures that can do partial rerouting by discovering common behaviors of over-loaded and under-utilized links and indentify the paths which cause these similarities.

In keeping statistics and routing information at every network site, we used a two-dimensional array of structures. This storage requirement is very important in the case of routers with storage restrictions. Furthermore, only the routing table entries that refer to (*source, dest*) pairs which pass from a specific site are used, wasting the remaining of the array space. We plan to investigate the theory of sparse matrices with the intention to minimize the routing table storage but without compromising the advantage of table lookups versus table searching.

Finally, we will attempt to integrate our theoretical results and the pro-

posed algorithmic solutions to wireless networks. However in that case, the link load balancing problem is reformed to a modulated frequency assignment problem since, all network links are virtual and the information transmission is done by a frequency synchronization between sites.

BIBLIOGRAPHY

[1] M. Andrews, B. Awerbuch, A. Fernandez, J. Kleinberg, T. Leighton, and Z. Liu. Universal stability results for greedy contention-resolution protocols. In *Proceedings of the 33d IEEE Symposium on Foundations of Computer Science*, pages 380–389, 1996.

[2] G. Apostolopoulos, R. Guerin, S. Kamat, A. Orda, and S. K. Tripathi. Intra-domain qos routing in ip networks: A feasibility and cost/benefit analysis. *IEEE Networks*, 13:42–54, 1999.

[3] G. Apostolopoulos, R. Guerin, S. Kamat, A. Orda, and D. Williams. Qos routing mechanisms and ospf extensions. *Internet RFC*, 1999.

[4] M. A. Arbib and J. A. Robinson. *Natural and Artificial Parallel Computation*. MIT Press, Cambridge, MA, 1990.

[5] B. Awerbuch, S. Kutten, and D. Peleg. Efficient deadlock-free routing. *Proceedings of the ACM Symposioum on Principles of Distributed Computing*, pages 177–188, 1991.

[6] S. Baker, H. J. Beier, T. Bemmerl, A. Bode, H. Ertl, U. Graf, O. Hansen, J. Haunerdinger, P. Hofstetter, R. Knödlseder, J. Kremenek, S. Langenbuch, R. Lindhof, T.Ludwidg, P. Luksh, R. Milner, B. Ries, and T. Tremi. TOPSYS - tools for parallel systems. Technical Report SFB Report 342/12/91, Technical University of Munich, 1991.

[7] F. Bauer and A. Varma. Distributed algorithms for multicast path setup in data networks. *IEEE/ACM Transactions in Networking*, 4:181–191, 1996.

[8] T. Bemmerl. Programming tools for massively parallel supercomputers. In J. J. Dongarra and B. Tourancheau, editors, *Environments and Tools for Parallel Scientific Computing*. Elsevier Science Publishers, 1993.

[9] K. Bolding. Chaotic routing - design and implementation of an adaptive multicomputer network router. Technical report, University of Washington, 1993.

[10] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson. Adversarial queuing theory. In *Proceedings of the TwentyEighth ACM Symposium on Theory of Computing (STOC)*, pages 376–385, 1996.

[11] D. Bretsimas. The probabilistic minimum spanning tree problem. *Networks*, 20:245–275, 1990.

[12] H. Buhrman, J. Hoepman, and P. Vitanyi. Optimal routing tables. In *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 134–142, 1996.

[13] W. Cai. *Parallel Program Monitoring - The Logical Clock Approach and its Deadlock Avoidance.* PhD thesis, Department of Computer Science, University of Exeter, 1991.

[14] S. Chen and K. Nahrstedt. Distributed quality-of-service routing in high-speed networks based on selective probing. In *International Telecommunications Conference*, volume 23, pages 80–89, 1998.

[15] A. Chien. A cost and speed model for k-ary n-cube wormhole routers. In *Proceedings of Hot Interconnects*, pages 529–539, 1993.

[16] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms.* Mc Graw Hill, 1985.

[17] M. Cosnard and D. Trystram. *Parallel Algorithms and Architectures.* International Thomson Computer Press, 1995.

[18] E. Crawley, R. Nair, B. Ragagopalan, and H. Sandick. A framework for qos based routing in the internet. *Internet RFC*, no.2386, August, 1998.

[19] R. Cypher. Minimal deadlock free routing in hypercubes and arbitrary networks. In *Proceedings of the 7th IEEE SYmposioum on Parallel and Distributed Processing*, 1995.

[20] R. Cypher and L. Gravano. Requirements for deadlock-free adaptive packet routing. *SIAM, Journal on Computing*, 23(4):1266–1274, 1994.

[21] W. Dally and H. Aoki. Deadlock free adaptive routing in multiprocessor interconnection networks. *IEEE Transcations in Parallel Distributed Systems*, 36.5:547–553, 1993.

[22] K. Day and A. E. Al-Ayyoub. Product-closed networks. *Journal of Systems Architecture*, 45:323–338, 1998.

[23] M. Degermark, A. Brodnik, S. Carlsson, and S. Pink. Small forwarding tables for fast routing lookups. In *Proceedings of the SIGCOMM 97 Conference on Applications Technologies*, pages 3–14, 1997.

[24] X. Deng, H. Liu, J. Long, and B. Xiao. Competitive analysis of network load balancing. *Journal of Parallel and Distributed Computing*, 40:162–172, 1997.

[25] S. Dolev. Self stabilizing routing and related protocols. *Journal of Parallel and Distributed Computing*, 42:122–127, 1997.

[26] S. Even and B. Monien. On the number of rounds necessary to disseminate information. In *Proceedings of 1989 ACM Symposium on Parallel Algorithms and Architectures, Santa Fe, New Mexico*, pages 318–327, 1989.

[27] A. Farley and A. Proskurowski. Gossiping in grid graphs. *Journal of Combinatorial Information System Science*, pages 161–172, 1980.

[28] M. Flammini, J. vanLeaeuwen, and A. Marchetti-Spaccamela. The complexity of interval routing in random graphs. *Twentieth International Symposium on Mathematical Foundations of Computer Science, Lect. Notes Comp. Sci. (Springer)*, 1995.

[29] D. Foulser, M. Li, and Q. Yang. Theory and algorithms for plan merging. *Artificial Intelligence*, 57:143–181, 1992.

[30] P. Fraigniaud and C. Garoille. Memory requirement for the universal routing schemes. *Fourteenth Annual Sumposium on Principles of Distributed Computing*, pages 223–230, 1995.

[31] P. Fraigniaud and S. Vial. Approximation algorithms for broadcasting and gossiping. *Journal of Parallel and Distributed Computing*, 43:47–55, 1997.

[32] M. Garey and D. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W. H. Freedman and Company, 41 Madison Avenue, New York, N.Y. 10010, 1979.

[33] A. Gilbert and H. Pollac. Steiner minimal tree. *SIAM J. Appl. Math*, 16, 1968.

[34] C. Glass and L. Ni. The turn model for adaptive routing. *International Symposium on Computer Architecture (Selected Papers)*, pages 441–450, 1998.

[35] C. Greenhalgh, S. Benford, A. Bullock, N. Kuijpers, and K. Donkers. Predicting network traffic for collaborative virtual environments. *Computer Networks and ISDN Systems*, 30:1677–1685, 1998.

[36] S. M. Hedetniemi, S. T. Hedetniemi, and A. Liestman. A survey on gossiping and broadcasting in communication networks. *Networks*, 18:319–349, 1986.

[37] T. Y. Ho and L. H. Hsu. Transmitting on various network topologies. *Networks*, 27:145–157, 1996.

[38] K. Holmberg and K. Helstrand. Solving the uncapacitated network design problem by a lagrangean heuristic and branch and bound. *Operations Research*, 46:247–259, 1998.

[39] K. Holmberg and K. Ling. A lagrangean heuristic for the facility location problem with staircase costs. *European Journal of Operational Research*, pages 63–74, 1997.

[40] K. Holmberg and D. Yuan. A lagrangean approach to network design problems. *Proceedings of the Seveth International Conference of IFORS : Information Systems in Logistics and Transportation Gothenburg Sweden*, pages 529–539, 1997.

[41] P. Hoyer and K. Larsen. Permutation routing via matchings. Technical report, Odense University, 1996.

[42] K. Hwang. *Advanced Computer Architecture- Parallelism, Scalability, Programmability.* McGraw Hill Series in Computer Sciencey, 1993.

[43] P. Jaillet. Shortest path problems with node failures. *Networks*, 22:589–605, 1992.

[44] R. Jain and W. Sun. Qos/policy/constraint-based routing. *Carrier IP Telephony 2000 Comprehensive Report by International Engineering Consortium*, 2000.

[45] X. Jia. A distributed algorithm for delay-bounded multicast routing for multimedia applications in wide area networks. *IEEE/ACM Transcations in Networking*, 6(6):828–837, 1998.

[46] X. Jia, N. Pissinou, and K. Makki. A real-time multicast routing algorithm for multimedia applications. *Journal of Computer Communication*, 20(12):1098–1106, 1997.

[47] R. M. Karp. *Reducability among Combinatorial Problems.* Plenum Press, New York, 1972.

[48] A. Kermani and B. Kleinrock. Virtual cut through, a new computer communication switch technique. *Computer Networks*, 3:267–286, 1979.

[49] S. G. Kolliopoulos and C. Stein. Improved approximation algorithms for unsplittable flow problems. In *IEEE Symposium on Foundations of Computer Science*, pages 426–435, 1997.

[50] S. Kolliopulos and C. Stein. The capacitated arc routing problem, lower bounds. *Networks*, 22:669–690, 1992.

[51] A. N. Kolmogorov. Three approaches to the quantitive definition of information. *Problems Information Transmission*, 1:1–7, 1965.

[52] D. Krumme, G. Cybenko, and K. Venkataraman. Gossiping in minimal time. *SIAM Journal on Computing*, 21(1):111–139, 1992.

[53] S. Kunniyur and R. Srikant. End-to-end congestion control schemes: Utility functions, random losses and ECN marks. In *INFOCOM (3)*, pages 1323–1332, 2000.

[54] A. Laing and R. Cypher. Deadlock-free routing in arbitrary networks via the flattest common supersequence method. *ACM SPAA Symposioum 98, Puerto Vallarta Mexico*, pages 55–66, 1998.

[55] T. Leighton, B. Maggs, and S. Rao. Packet routing and job scheduling in o(congestion + dilation) steps. *Combinatorica*, 14(2):167–186, 1994.

[56] D. Linder and J. harden. An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes. *IEEE transactions on Computing*, 40:2–12, 1991.

[57] Q. Ma and P. Steenkiste. On path selection for traffic with bandwidth guarantees. *In Proceedings of IEEE International Conference on Network Protocols, Atlanta, GA, October 1997.*, 1997.

[58] Q. Ma, P. Steenkiste, and H. Zhang. Routing high-bandwidth traffic in max-min fair share networks. In *SIGCOMM*, pages 206–217, 1996.

[59] T. E. Meyer, J. A. Davis, and J. L. Davidson. Analysis of load average and its relationship to program run time on networks of workstations. *Journal of Parallel and Distributed Computing*, 44:141–146, 1997.

[60] M. Middendorf. Minimum broadcast time is np-complete for 3-regular planar graphs and deadline 2. *Information Processing Letters*, 46:281–287, 1993.

[61] P. Mohapatra. Wormhole routing techniques for directly connected multicomputer systems. *ACM Computing Surveys*, 30.3:374–410, 1998.

[62] E. Monteiro, F. Boavida, G. Quadros, and V. Freitas. Specification, quantification and provision of quality of service and congestion control for new communication services. *Proceedings of the Sixthteen AFCEA Europe Symposium and IEEE COMSOC*, 1:58–68, 1997.

[63] J. Moy. Ospf version 2. *RFC no. 1583, Internet RFC, Internet Engineering Task Force*, 1, 1994.

[64] S. Nelakuditi, S. Varadarajan, and Z. Zhang. On localized control in quality-of-service routing, 2000.

[65] S. Nelakuditi, Z. L. Zhang, and R. P. Tsang. Adaptive proportional routing: A localized qos routing approach. In *INFOCOM (3)*, pages 1566–1575, 2000.

[66] A. Noga, F. Chung, and R.L. Graham. Routing permutations on graphs via matchings. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 583–591, 1993.

[67] S. G. Penrice. Balanced graphs and network flows. *Networks*, 29:77–80, 1997.

[68] K. W. Pullen. A random network model of network transmission. *Networks*, 16:397–409, 1986.

[69] H.F. Salama, D. S. Reeves, and Y. Viniotis. A distributed algorithm for delay-constrained unicast routing. In *INFOCOM (1)*, pages 84–91, 1997.

[70] C. Scheideler and B. Vocking. Universal continuous routing strategies. *Theory of Computing Systems*, 31:425–449, 1998.

[71] C. Scheideler and B. Vocking. From static to dynamic routing : Efficient transformations of store-and-forward protocols. In *In Proc. of the Thirty First Annual ACM Symposium on Theory of Computing (STOC)*, pages 215–224, 1999.

[72] A. Schrijver, P. Seymour, and P. Winkler. The ring loading problem. *SIAM Journal on Discrete Mathematics*, 11(1):1–14, 1998.

[73] L. Schwiebert and D. Jayasimha. A necessary and sufficient condition for deadlock-free wormhole routing. *Journal in Parallel and Distributed Computing*, 32:103–117, 1996.

[74] A. Silberschatz and P. Galvin. *Operating Systems Concepts*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1998.

[75] P. Slater, E. Cockayne, and S. Hedetniemi. Information dissemination in trees. *SIAM J. Computing*, 10(4):692–701, 1981.

[76] A. Tanenbaum. *Distributed Operating Systems*. Prentice Hall Englewood Cliffs NJ 07632, 1995.

[77] E. Varvarigos and V. Sharma. An efficient reservation connection control protocol for gigabit networks. *Computer Networks and ISDN Systems*, 30:1135–1156, 1998.

[78] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal of Selected Areas in Communications*, 14(7):1228–1234, 1996.

[79] Z. Wang and J. Crowcroft. Quality of service routing for supporting multimedia applications. *IEEE JSAC*, 14(7):1228–1234, 1997.

[80] Z. Zhang, C. Sanchez, B. Salkewich, and E. Crawely. Quality of service extensions to ospf routing. *Proceedings of IFIP Fifth International Workshop on Quality of Service, Columbia University, New-York*, 4:115–126, 1997.

[81] Y. Zhong and X. Yuan. Impact of resource reservation on the distributed multi–path quality of service routing scheme. In *Eighth International Workshop on Quality of Service (IWQoS2000), Pittsburgh, PA, June 2000.*, 2000.