INTEREST-BASED ROUTING IN INTELLIGENT OPTICAL TRANSPORT NETWORKS

A Thesis

Presented in Partial Fulfillment of the Requirements for

the Degree of Master of Science in the

Graduate School of the Ohio State University

By

Santosh Kumar, B. Tech.

* * * * *

The Ohio State University
2002

Master's Examination Committee:

Professor Anish Arora, Advisor

Professor Ten-Hwang Lai, and

Dr. Arjan Durresi

ABSTRACT

Transport networks are becoming dynamic with the introduction of intelligent optical switches. The physical capacity available in these networks has increased to several Terabits per second due to advances in the Dense Wavelength Division Multiplexing technology and this capacity is increasing at an accelerated rate with further advances in the hardware technology. What fraction of this available physical capacity can be used for carrying user traffic depends on the routing and wavelength assignment algorithm used for provisioning new connections. This thesis addresses the sub-problem of route-computation in optical transport networks.

The transport network is a circuit-switched network. A connection has to be established between two nodes through the network before any user traffic can be exchanged between them. The problem of route-computation in a dynamic optical transport network is to select a path for each arriving connection request such that the usable network capacity in the network is maximized. Establishing a connection on some paths in an optical network affects the likelihood of acceptance of connections being established concurrently and connections arriving later. This interaction between connections is referred to as interference. Existing route-computation algorithms do not minimize interference well. This thesis proposes a new route-computation algorithm called Interest-based routing. The basic idea in the Interest-based algorithm is to approximate the amount of interference between the various connection requests, referred to as *interest* in this thesis, and then use this information to compute a path of least

interference for every connection request. The major contributions of this thesis are a new algorithm to compute the value of interest and a new way to use this interest information during route-computation.

Interest-based algorithm offers a significant increase in the usable network capacity over other route-computation algorithms. Additionally, the connections accepted by the Interest-based algorithm have a potential for generating higher revenues. Interest-based algorithm is better suited for computing routes dynamically, as well. Suitable performance metrics have been used in this thesis to illustrate each of these improvements through simulation.

Several route-computation algorithms use a link's physical length in route-computation. This thesis shows that the use of a link's physical length causes a significant loss in the performance of a route-computation algorithm. The primary reason for this loss in performance is that a link's length is a static property of the link, and has no relation to the interference expected on it, which inherently is a dynamic property of the link.

This thesis also proposes a way to compute the maximum load that should be offered to any route-computation algorithm during simulation on a given optical network.

Dedicated to my parents:

Late B. Choudhary and Smt. Sunita Kumari

ACKNOWLEDGEMENTS

I am grateful to my advisor Prof Anish Arora for showing me a new way to look at the networks in general and the Internet, in particular. Prof Arora taught me that defining the problem is at least as important as finding a solution. This thesis would not have been possible without his continuous encouragement and support. I have always enjoyed and learned from his engaging discussions and his insightful comments. I am thankful to him for giving me the right amount of freedom in my research. I look forward to working with him in future.

Prof Ten-Hwang Lai has been a great teacher and a great friend to me. I am grateful to Prof Lai for being so accessible and so helpful. He taught me how to present my research work to the world.

I am thankful to Dr. Arjan Durresi for his guidance. I admire his breadth of knowledge. He is always aware of the latest progress in the field of networking research. He has taught me the value of urgency in research.

Several fellow researchers have provided me with useful comments and reviews during the progress of this work. I am specially thankful to Vinodkrishnan Kulathamani and Nikhil Chandhok for their repeated reviews. I am also thankful to Varun Gupta for being a source of constant encouragement and providing me with detailed reviews on this work.

I am eternally indebted to my father Late B. Choudhary and my mother Smt. Sunita Kumari for teaching me the value of education in life. They have made great sacrifices in their lives to bring me upto this level of education.

VITA

May 15, 1977	Born – Begusarai, India
1998	B. Tech., Computer Science & Engineering, Institute of Technology, Banaras Hindu University, India
1998-2000	Software Engineer, Siemens Communications Software, Bangalore, India
2000-present	Graduate Teaching and Research Associate, The Ohio State University

FIELDS OF STUDY

Major Field: Computer and Information Science

Specialization: Computer Networks

TABLE OF CONTENTS

Page

Chapters

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Dynamic provisioning is making its way into transport networks with the deployment of optical switches. Optical switches have the ability to reconfigure their switching states dynamically. They can exchange control messages with other switches in the network and work cooperatively to establish a new connection and tear down an existing connection without any manual intervention. Transport networks consisting of these intelligent switches are therefore able to provision new connections online as the requests arrive.

The physical capacity available in transport networks is several Terabits per second and is increasing with new advances in optical components technology. This has been made possible by recent advances in the Dense Wavelength Division Multiplexing (DWDM) multiplexing technology and the commercial viability of optical switches. DWDM enables a single fiber strand to carry several hundred wavelengths. Each wavelength can carry data at a speed of 10-40 Gigabits per second. Optical switches are capable of switching data at aggregate speeds exceeding several Terabits per second. With continuing advances in the optical components technology this capacity is doubling every twelve months.

What fraction of this available physical capacity can be used for carrying user traffic in an optical transport network depends on the routing and wavelength assignment algorithm used. Optical transport network is a circuit-switched network. Any data can be carried through this network only if a connection exists between its source and destination. A dynamic routing and wavelength assignment (RWA) algorithm is used to establish connections in such a network. A dynamic RWA algorithm computes a path and selects a wavelength online as the connection requests arrive. The successful establishment of one connection along a particular path and on a particular wavelength in an optical network affects the likelihood of successful establishment of subsequent connections. This phenomenon is referred to as interference, to capture the adverse effect of establishing one connection along a particular path and on a particular wavelength, on other connections. Number of available paths and wavelengths for an arriving connection request decreases as links in the network start running out of capacity. After some time, the network state (number free wavelengths on each link in the network) may be such that some connection requests do not find any path with a free wavelength and therefore, get rejected. The total number of connections that can be established in a given optical transport network thus depends on the RWA algorithm used.

RWA algorithms are used by the provisioning software in an optical network. Dynamic provisioning in an optical transport network consists of three tasks: computing a path, selecting a wavelength, and reserving the selected wavelength on the computed path. When a connection request arrives, first a path is computed for it that has at least one free wavelength. One wavelength from the set of free wavelengths on this path is then selected for this connection. After a path has been computed and a free wavelength has been selected, an attempt is made to reserve this wavelength on the computed path.

Path-computation has the most dominant effect on the total number of connections that can be established in a network[15]. Algorithms for path-computation are called routing or route-computation algorithms. This thesis proposes a new routing algorithm called "Interest-based routing" algorithm. The basic idea in the Interest-based algorithm is to approximate the amount of interference between the various connection requests and use this information to compute a path of least interference for every connection request. The major contributions of this thesis are a new algorithm to approximate the potential interference and a new way to use this information when computing paths. Interest-based algorithm runs in two phases. In the first phase, interests of all currently pending connection requests is computed and communicated to all ingress nodes. In the second phase, all ingress nodes compute paths for the connection requests pending at them by using the value of interest and the remaining capacity in the network.

The performance of the Interest-based algorithm is compared with the Minimum-hop, the Availability-based, the Future-based, the Cost-based, and the Adaptive Dynamic Routing (ADR) algorithms in this thesis. Through simulation, it is shown that the Interest-based algorithm always offers higher usable network capacities and lower blocking probabilities over the other five algorithms. In a distributed environment, routes may need to be computed with an outdated knowledge of the network state. It is shown that the Interest-based algorithm continues to perform well when the network state updates arrive periodically. This shows that the Interest-based algorithm is well-suited for distributed implementation. Further, it is shown that the Interest-based algorithm offers much lower variation in performance when the order of connection request arrivals is varied. This shows that the Interest-based algorithm is better suited for dynamic route-computation.

The rest of the thesis is organized as follows: Chapter 2 describes the network model used in this thesis and formally states the problem of route-computation. Chapter 3 describes the previous route-computation algorithms and their qualitative analysis. Chapter 4 presents the Interest-based route-computation algorithm. Chapter 5 describes the metrics used to evaluate various route-computation algorithms. Two new metrics have been proposed in this chapter. This chapter also presents a simple way to compute the maximum load that should be offered to any route-computation algorithm during simulation on a given network. Chapter 6 presents the simulation results comparing the Interest-based routing algorithm with other algorithms. Chapter 7 concludes the thesis and describes future work in this direction.

CHAPTER 2

NETWORK MODEL AND PROBLEM DEFINITION

This chapter describes the network model used in this thesis and defines the problem of route-computation.

2.1 NETWORK MODEL

An optical transport network is modeled as a connected graph $G = <V,E>$, where V is the set of vertices and E is the set of edges. An example network appears in Figure 2.1. The edges or links of G are denoted by L_i, where $i=1,2,\ldots,N$, and $N = |E|$. All the links in the network are bi-directional. Maximum capacity of a link L_i is the total number of wavelengths that can be carried on it and is denoted by $F(L_i)$. Each node is equipped with an optical cross-connect (OXC) and a controller. For the purpose of route-computation, it has been assumed that each node has full-wavelength conversion capability. Controllers at each node maintain the status of all links that emanate from it. A signaling channel exists between every pair of nodes, which is used to exchange link state information and for establishing new connections.

Every connection request specifies a source (s) and a destination (d). A source node is also sometimes referred to as an ingress node. Similarly, a destination node is sometimes

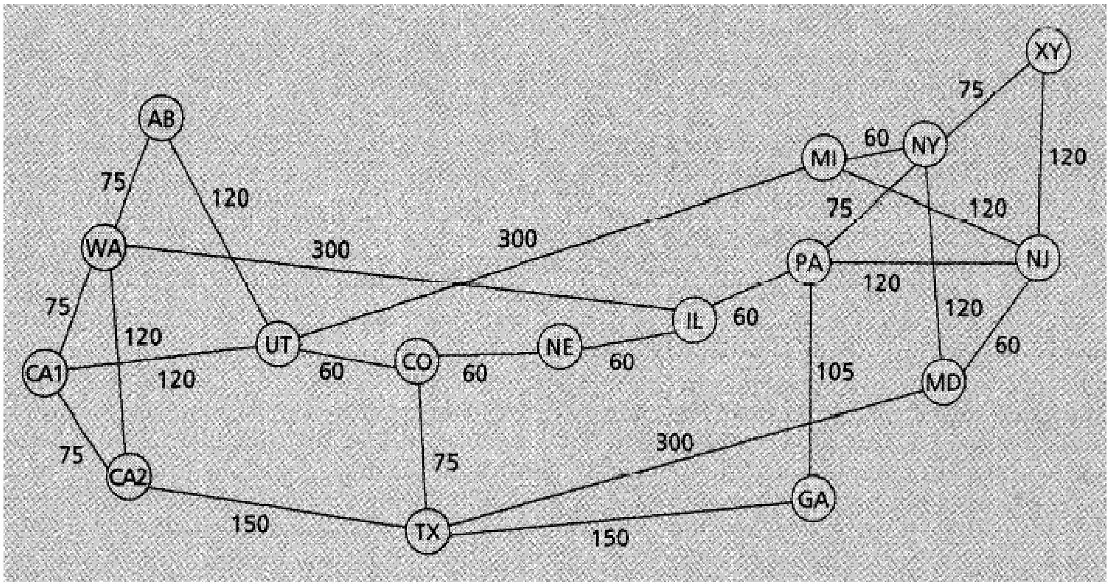

Figure 2.1: NSFNET National Network

referred to as an egress node. A connection request is presented to its source node *s*. Every connection requires one wavelength from *s* to *d*. This thesis assumes that connections once established, stay in the network permanently.

A *k*-hop path *p* in the network is specified as a sequence of links: $p = <L_{f(1)}, L_{f(2)}, \ldots, L_{f(k)}>$, where $L_{f(1)}$ is the id of the first link in the path, $L_{f(2)}$ is the id of the second link, and so on. Every pair of adjacent links $L_{f(i)}$ and $L_{f(i+1)}$ are connected by an intermediate node.

A network state is defined in terms of the available capacities in each of the links of the network. It is a sequence $<A(L_1), A(L_2), \ldots., A(L_N)>$, where $A(L_i)$ is the current available capacity in link L_i.

2.2 PROBLEM STATEMENT

The problem of route-computation in an optical transport network with full wavelength conversion is defined as follows:

“For a given sequence of arriving connection requests and the current network state, find a free path from source to destination for every connection request, such that the total number of connections accepted in the network is maximized.” A free path is a path where each of its links have at least one wavelength free.

The corresponding decision problem is NP-complete. The decision problem can be stated as follows: “Given r connection requests and a network G with W wavelengths on each link, is it possible to route all r connections through G?” This can easily be proved NP–complete by restricting the above problem to an instance where each link in the network has a maximum of one wavelength, i.e. $F(L_i) = 1$ for all i. The r-commodity integral flow problem in an undirected graph with one total requirement R, described in [12] can be trivially reduced to the problem of routing in an optical network with $R = r$.

CHAPTER 3

EXISTING ROUTE-COMPUTATION ALGORITHMS AND THEIR ANALYSIS

This chapter describes the nature of route-computation in optical transport networks. Various existing route-computation algorithms are then classified as static, dynamic, adaptive, and non-adaptive in Section 3.1. Section 3.2 briefly describes the route-computation approaches in public-switched telephone networks and packet-switched Internet. The purpose is to analyze what ideas from these approaches can be used in the route-computation algorithms for optical transport networks and also point out the reasons why these approaches are not directly applicable to optical transport networks. Five existing route-computation algorithms for optical transport networks are described in considerable detail in Section 3.3. An intuitive explanation for some of these algorithms are offered along the way and a qualitative analysis is made to explain why a further improvement is possible.

3.1 CLASSIFICATION OF ROUTE-COMPUTATION ALGORITHMS

Route-computation algorithms for an optical network can be broadly classified into two categories: static route-computation algorithms and dynamic route-computation algorithms. They are also sometimes referred to as static lightpath establishment (SLE) and dynamic lightpath establishment (DLE) algorithms. The major difference between the two approaches is that the operation of a static route-computation algorithm does not change over time whereas the operation of a dynamic route-computation algorithm changes over time. The dynamic route-computation algorithms can be further subdivided into two categories: those that take the current state of the network into consideration while computing a path and those that do not. The former is referred to as an adaptive algorithm and the latter as a non-adaptive algorithm. The adaptive route-computation schemes are the most advanced algorithms because they adapt their operation with the change in traffic demands. This thesis considers only adaptive route-computation algorithms. The terms dynamic route-computation algorithms and adaptive route-computation algorithms have been used interchangeably in this thesis to refer to adaptive algorithms, as is common in the literature.

3.2 LESSONS FROM TELEPHONE NETWORKS AND THE INTERNET

This section compares optical transport networks with telephone networks, a representative of circuit-switched networks; and with the Internet, a representative of packet-switched networks. The comparison of optical transport network with these two networks helps understand what ideas from the work done in the area of route-computation in these two types of networks can be reused in route-computation for an optical transport network and why the algorithms developed for these network are not directly applicable to route-computation in optical transport networks.

The optical transport network is a circuit-switched network, and is similar in nature to telephone networks. Adaptive route-computation algorithms like Real Time Network

Routing (RTNR) and Dynamic Alternative Routing (DAR) were introduced in telephone networks in mid 80s[3,4]. These algorithms assume that a direct link exists between every pair of source-destination. This link can be a logical link. When a call arrives, the direct link is always chosen for this call, if there is any spare capacity. Various adaptive route-computation algorithms in telephone networks differ in the way they choose a two-link alternative path for an arriving call, if the direct link for this call is not free. Certain capacity on each direct link is reserved for calls whose source-destination nodes are directly connected by this link. The unreserved capacity can be used for indirect calls. Expected traffic demand is used to decide the amount of reservation. This idea of using expected traffic in route-computation can be used in optical networks too.

Most of the adaptive route-computation algorithms developed and in use in a telephone network work well on a complete graph or at least on a regular graph topology. Topology of a transport network, on the other hand can be an arbitrary mesh, which need not be even a regular graph. In transport networks, the route-computation can not assume a direct link between a given source-destination pair. So, the idea of choosing a direct link when it is free and a two-link alternate path when the direct link is not available, is not directly applicable to transport networks.

The route-computation schemes in packet-switched networks like the Internet, work on arbitrary topologies. For every destination of interest, a source node calculates a shortest-path to the destination. Dijkstra’s shortest path algorithm or Bellman-Ford’s shortest path algorithm is used to compute the shortest path. Dijkstra’s shortest path algorithm works with global network state information. The idea of using a shortest-path algorithm to compute paths in a network with arbitrary topology can be used in optical networks too.

However, the route-computation algorithms used in packet-switched networks are not directly applicable to optical networks. The objective of a route-computation scheme in a packet-switched network is mostly that of minimizing the delay for every packet. Therefore, the weight of each link typically is proportional to its physical length. Although this choice of link weight does not minimize the delay for every packet in the network, it still works in packet-switched networks by assuming the capacity of each link to be infinite when compared to the bandwidth requirement of an individual packet. Due to this approximation, these networks offer a best-effort service, where congestion can occur and some packets may get lost. On the other hand, the objective of a route-computation algorithm in an optical network is that of maximizing the total number of connections that can be established in the network. This objective is a system-side optimization problem. The problem of route-computation in circuit-switched networks is further complicated by the fact that the capacity of each link is limited and every connection established occupies a finite capacity on its path. Because the holding time for a connection in an optical network is on the order of hours, even days, the interference among connections becomes pronounced. If the paths for connections are not chosen carefully, establishing one connection along a particular path may cause multiple other connections to be blocked. Therefore, link weights have to be chosen very carefully in the route-computation algorithms for a transport network, taking the system-wide goal into consideration.

3.3 EXISTING ROUTE-COMPUTATION ALGORITHMS

As noted in Section 3.2, Dijkstra's shortest-path algorithm can be used to compute paths in networks with arbitrary topologies. Because optical transport networks may have arbitrary topologies most of the dynamic route-computation algorithms proposed recently use Dijkstra's shortest-path algorithm [5,6,7,8,9,13,15]. In these algorithms, each link L_i is assigned a weight $W(L_i)$. All of the algorithms described in this thesis execute Dijkstra's shortest-path algorithm to find a path with the lowest total weight. The total weight of a path *p*, $TW(p)$, is the summation of the weights of all its links :

$$TW(p) = \sum_{j=1}^{N(p)} W(L_{f(j)}$$

where, $N(p)$ is the number of links in path *p*, and $f(j)$ is the link id corresponding to the j^{th} link in path *p*. Various algorithms differ in the way they define $W(L_i)$.

3.3.1 THE MINIMUM-HOP ALGORITHM

This is the simplest algorithm. The weight of each link is defined as:

$$W(L_i) = 1$$

For any arriving connection request, this algorithm selects a free path with the least number of hops. The intuition is to use as little resources from the network as possible. However, it does not take into consideration the effect of routing one connection over connections that arrive later in the network. Establishing one connection over a *k*-hop path has a potential of blocking *k* future connections from the network. This situation can arise if each of the *k* hops used by this connection happens to be critical for *k* other connections that arrive later. Had a different path (possibly longer than the current path) been selected for this connection, at least some of those *k* connections may have been successfully established. Therefore, this algorithm does not perform very well.

Other three algorithm use the available capacity of a link in defining its weight.

3.3.2 THE AVAILABILITY-BASED ALGORITHM [6]

In this algorithm, the weight of a link is defined as:

$$W(L_i) = \frac{1}{A(L_i)}$$

where, $A(L_i)$ is the number of free channels in the link L_i. Here, the weight of a link is inversely proportional to the number of free channels in it. The idea is to make a link expensive if it is high in demand, expecting that this link will continue to be high in demand by all future connection requests. However, if for the same set of connection requests, the order of connection arrivals is changed so that links high in demand earlier on are rarely needed by later connections; this strategy does not work well. In [6], this algorithm has been referred to as the Total-cost based algorithm. A different name has been used here to reflect its definition of link weight.

3.3.3 THE FUTURE-BASED ALGORITHM [6]

This is a variation of the Availability-based algorithm. This algorithm tries to reserve links which have only one free channel for later use. The weight of a link is defined as:

$$W(L_i) = \frac{1}{(A(L_i) - 1)}$$

If no path can be found using the above definition of link weight, the Availability-based algorithm's definition of link weight is used to re-compute a path.

3.3.4 THE COST-BASED ALGORITHM

This algorithm is a variation of the Availability-based algorithm. The only difference from the Availability-based algorithm is the use of link length in defining link weight. This algorithm is currently used by some service providers. Here, the weight of a link is defined as:

$$W(L_i) = \frac{Cost\ (L_i)}{A(L_i)}$$

where, *Cost*(L_i) is the fixed cost of the link L_i. *Cost*(L_i) most frequently depends on the physical length of the link L_i. In some cases, it may depend on its other fixed properties, such as its quality, maximum bandwidth it can support, etc. In this thesis, *Cost*(L_i) reflects link L_i's physical length.

3.3.5 THE ADAPTIVE DYNAMIC ROUTE-COMPUTATION ALGORITHM

This algorithm was proposed in [8]. Its weight was defined in [7] as:

$$W(L_i) = 1 + \frac{1}{A(L_i)}$$

This algorithm gives more importance to the number of links in a path than the Availability-based algorithm. However, it was found through simulation that the Availability-based algorithm always outperforms this algorithm. Hence, the results comparing this algorithm with other algorithms has not been presented in this thesis.

CHAPTER 4

INTEREST-BASED ROUTE-COMPUTATION ALGORITHM

This chapter presents the Interest-based routing algorithm. A value of interest is associated with each link. The weight of a link is defined in terms of the interest associated with this link and it's available capacity. Dijkstra's shortest-path algorithm, described in Section 3.3, is then used to find a path with the lowest total weight for a given connection request. Section 4.3 discusses how this algorithm can be implemented in an intelligent optical network.

Optical networks have a large capacity. Each wavelength can carry traffic at a speed of upto 40 Gigabits per second (Gbps). The connections in an optical network are, therefore established only after a proper agreement between the customer and the service provider. These agreements are negotiated under the Service Level Agreements (SLAs). In order to maximize the network capacity and reduce the cost of network operation, several connections (derived from the SLAs) are established together, also referred to as a batch of connection requests. One batch of connection requests that are established concurrently is referred to as a *concurrent set* of connection requests in this thesis.

When the connection establishment in an optical network becomes truly dynamic, it is conceivable that several connection requests may arrive concurrently in the network. In that situation the connection requests that arrive concurrently can be called a concurrent set of connection requests.

The concurrent set of connection requests can also be derived from the traffic pattern observed, if one exists. It has been observed by service providers that there is a regular traffic pattern in transport networks. During the daytime, there are heavy traffic demands between central offices that serve city areas, densely populated with offices. During the evening/night time there are higher traffic demands between central offices serving city areas, densely populated with residences. This pattern repeats over a period of time (every day). This period is referred to as a *traffic cycle*. It is foreseen that in future, when transport networks become truly dynamic in nature, a set of connection requests will arrive during the morning hours to serve daytime traffic among offices, which will be terminated by the evening. Another set of connection requests will arrive in the evening to serve residential areas, which will be terminated sometime during the night. The same set of connection requests will arrive every morning and every evening with minor variations. If such a pattern is available in a network then the set of concurrent connection requests can include all the connections that overlap with each other temporally. In this case, the Interest-based routing algorithm can perform even better.

4.1 INTEREST CALCULATION ALGORITHM

This section presents the Interest calculation procedure.

Every source node computes its interest in all the links of the network in the following manner. For every connection request, that is in the concurrent set of a source node and is yet to be established, a minimum-hop path *p1* is computed in the current state

```
Calculate_Interest (j, concurrent set C(j), Interest(j,L_i))
1. Interest(j, L_i) ← 0, for all links L_i in the network
2. For each connection request in C(j), perform steps 3, 4, 5, and 6.
3. Compute a minimum-hop path, p1. Call the number of hops in this minimum-hop
   path hop_count(p1).
4. If hop_count(p1) > 1, hop_count(p1) ← 2, and compute another minimum-hop
   path p2, which differs from p1 in at least one link. If a different path can not be
   computed by excluding a particular link, mark this link as "crucial."
5. For all the links L_i in the path p1,
      a. Interest(j, L_i) ←Interest(j, L_i) + 1, if L_i is marked "crucial,"
      b. Interest(j, L_i) ← Interest(j ,L_i) + 1/hop_count(p1), otherwise.
6. If a path p2 exists, then for all the links L_i in the path p2,
      a. Interest(j, L_i) ← Interest(j ,L_i) + 1/hop_count(p1).
```

Figure 4.1: Calculate_Interest Algorithm

of the network. If *p1* contains only one link, then the interest of this link is incremented by one. If *p1* has more than one link, then one more minimum-hop path, *p2* is calculated which differs from *p1* in at least one link. If no such path *p2* exists, by excluding some link in *p1*, this link is marked "crucial" and its interest is incremented by one. If both paths *p1* and *p2* can be found, then the interest of all links in *p1* and *p2*, that are not marked "crucial," is incremented by ½. The interest of any link that is common to both *p1* and *p2* is incremented by ½, twice. It is also possible to apply *k*-disjoint shortest path algorithms to find k=2 disjoint shortest paths simultaneously.

Intuitively, every connection request expresses its interest in the network. If a connection has two or more hops in its shortest possible path, then it distributes its interest on two minimum-hop paths. Distributing interest on more than two paths further dilutes the interest and is therefore not useful.

Every source node *j* executes the Calculate_Interest algorithm shown in Figure 4.1 to calculate its interest in all links of the network. Concurrent set *C(j)* is the set of connection requests pending at source node *j*.

After the source nodes have individually calculated their corresponding interests, they broadcast their interest values to all the other nodes in the network. Every source node, when it receives interest values from other nodes, it adds that to the interest value calculated by itself. When it has received the interest values from all the other nodes in the network, interest of a particular link, $Interest(L_i)$ is calculated as follows:

$$Interest(L_i) = \sum_{j=1}^{m} Interest(j, L_i)$$

where, m is the total number of source nodes in the network.

Interest calculation is performed on the current state of the network. When a connection has been established, the interest corresponding to this connection is deleted from the network. Operationally, the source node *j* of this connection updates $Interest(L_i, j)$ for all links L_i in the network, and broadcasts the new values of its $Interest(L_i, j)$ to all the other nodes. Other nodes, upon receipt of this message from node *j*, replace the old value of interest corresponding to node *j*, with the recently received fresh value and recalculate $Interest(L_i)$ for all links L_i in the network.

If some link becomes full, so that it can no longer be used to carry any additional connections, then for all the connections corresponding to which an interest was expressed on this link, and that are yet to be established, interest is recomputed. The updated value of interest is broadcast by the corresponding source nodes.

4.2 ROUTE-COMPUTATION

This section describes the Interest-based route-computation algorithm. It first describes the algorithm and then explains the motivation behind it.

According to the current definition of interest, interest indicates the demand for a particular link in the network. If the Interest for a link is high, this link is preferred by many connection requests. However, the interest of a link alone does not truly reflect the interference expected on it. If a link has a high interest value and a large number of free wavelengths, it should have a lower weight than a link which has a low interest value but an even lower number of available wavelengths. To measure the deficit of demand and supply for a link L_i, we compute its *Base Weight*, $BW(L_i)$, defined as:

$$BW(L_i) = Interest(L_i) - A(L_i)$$

The weight of a link, $W(L_i)$ is defined as follows:

If $BW(L_i) < 0$, then

$$W(L_i) = \frac{1}{A(L_i)}$$

If $BW(L_i) \geq 0$, then

$$W(L_i) = \frac{BW(L_i) + 2}{A(L_i)} \qquad \text{...(1)}$$

If two links have different values of BW, which are both negative, it means that the demand for them is lower than their available capacities. Therefore, their weights should be inversely proportional to their available capacities. If a link has a positive value of

BW, then its weight should reflect that the demand for this link is higher than it can support, and therefore should have a higher weight. Moreover, the weight of such a link should be proportional to its deficit, which is captured by its value of BW. If a link has demand equal to its available capacity, then its weight should be proportional to its available capacity, and at the same time, its weight should be higher than a link which has the same available capacity but a negative value of BW. Therefore, a numerical value of 2 is added to the numerator of $W(L_i)$ in Equation (1).

When computing a route for a given connection request, all paths having less than two links with $BW(L_i) \geq 1$ in it, are considered, and one with the lowest total weight is selected. Any link with $BW(L_i) \geq 1$ has either $k+1$ one-hop connections expected to use it or $2k+2$ multi-hop connections expected to use it; where k is the available capacity on this link. If there is only one link with $BW(L_i) \geq 1$ in a path then this path can be taken if this is the minimum-weight path, because at most one other connection will be rejected as a result. If there are two links with $BW(L_i) \geq 1$ in a path, then it is reasonably certain that establishing a connection on that path has a high likelihood of rejecting at least one connection, and a maximum of two connections. Therefore, if a connection can not find any path with less than two links with $BW(L_i) \geq 1$, it is rejected.

Observe that one-hop connections are always accepted by the Interest-based algorithm if there is capacity available on its direct link. This is desirable since one-hop connections, when established on its direct link offer least interference to other connections. If there is an alternative path available for this one-hop connection request then the same alternative path can be used by any other multi-hop connection request that was expected to use this direct link. Thus, there is no loss in terms of the total number of connections established in the network. This argument does not hold for multi-hop connection requests.

4.3 A NOTE ON IMPLEMENTATION

This section describes how the current implementations may be augmented to implement the Interest-based route-computation algorithm. It has been assumed that current implementations of routing algorithms use the available capacity information in their route-computation.

As soon as a connection is established, two changes take place in the network – the available capacities on some links decrease and the interest values of some links decrease. The same signaling protocol that is used to broadcast the change in available capacities can be used to broadcast the change in interest values.

The Interest-based algorithm can be used in optical networks with full wavelength conversion, partial wavelength conversion, or no wavelength conversion. Wavelength converters are now becoming cheaper and therefore networks of tomorrow can be expected to have full-wavelength conversion. The regenerators placed in optical transport networks can be used to provide wavelength conversion. Most of the optical switches used today are hybrid switches, which have both electronic and optical switching fabrics. These switches can perform wavelength conversion. In summary, it is reasonable to expect the transport networks to have full wavelength conversion. In networks with full wavelength conversion, the Interest-based algorithm can be used to compute a path and any free wavelength can be used to carry the connection. However, if there is no wavelength conversion or if there is only partial wavelength conversion in a network, the

Interest-based algorithm can be used to compute a path, and a wavelength selection algorithms can be used to select and reserve a wavelength.

CHAPTER 5

METRICS FOR PERFORMANCE EVALUATION

This chapter defines the metrics used for evaluating the performance of various route-computation algorithms in an optical network. In addition to defining blocking probability and usable network capacity, which are used widely in the literature, two new metrics have been proposed here: revenue deficit and performance stability. Revenue deficit, which is similar to the blocking probability, takes into account not only the total number of connections accepted by a route-computation algorithm but also the lengths of those connections. The metric of performance stability indicates the suitability of a route-computation algorithm for dynamic route-computation. This chapter also presents the derivation of maximum load that should be offered to any route-computation algorithm during simulation on a given network.

5.1 METRICS

This section defines four metrics for evaluating the performance of various route-computation algorithms in an optical network. These four metrics are – blocking probability, usable network capacity, revenue deficit, and performance stability.

5.1.1 BLOCKING PROBABILITY

Definition 1: (Blocking probability) - Let *TC* be the set of connections offered to the network and *AC* be the set of connections accepted in the network using a particular route-computation algorithm. Blocking probability of this route-computation algorithm can then be defined as:

$$\textit{blocking probability} = \frac{(|TC| - |AC|)}{|TC|}$$

where, $|TC|$ is the number of connections in the set TC and $|AC|$ is the number of connections in the set AC.

The metric of blocking probability is widely used in the literature.

5.1.2 USABLE NETWORK CAPACITY

Usable network capacity is defined here to be the number of connections that can be accepted by a network with a 1% blocking probability. This metric is of interest to service providers because it indicates how much revenue can be generated from a network by using a specific route-computation algorithm.

5.1.3 REVENUE DEFICIT

The metrics of blocking probability and usable network capacity assume that all connections are equally important. However, in practice there might well be a case where longer connections, which span across multiple hops (on their ideal shortest paths through the network) generate higher revenues.

Definition 2: (Average Hop Count) - Let *HC(C)* be the hop count of a connection *C* in an empty network. *AHC(S)* of a set of connections *S* is defined to be the average over the hop counts of all connections in the set *S*. More specifically,

$$AHC(S) = \frac{\sum_{C \in S} HC(C)}{|S|}$$

where, $|S|$ is the number of connection in the set *S*. *AHC(S)* is referred to as the *average hop count* of the set of connections *S*.

Notice that *HC(C*) is the property of connection *C* and the network topology. It does not depend on the particular path on which connection *C* is established by a route-computation algorithm Similarly, *AHC(S)* does not depend on what paths are actually used to establish the connections in *S*.

To take the hop counts of connections into account, the revenue generated by various route-computation algorithms can be computed.

Definition 3: (Revenue generated by a connection) - Revenue generated by establishing a connection *C* in a given network is defined as:

$$REV(C) = k * HC_C$$

where, k is some constant. In this thesis, we assume $k = 1$. Therefore, *REV(C)* is equal to *HC(C*).

Definition 4: (Revenue generated by an algorithm) - Revenue generated by an algorithm is defined as the sum of the revenues generated by each of the connections established by using this algorithm. Let the set of connections established by this algorithm be *AC*. Then, revenue generated by this algorithm is defined as:

$$REV(AC) = \sum_{C \in AC} REV(C) = \sum_{C \in AC} HC(C) = |AC| \times \frac{\sum_{C \in AC} HC(C)}{|AC|} = |AC| \times AHC(AC).$$

The revenue generated in a network by using a route-computation algorithm is dependent on the maximum capacity of the links. As the maximum capacities of links are increased, the revenue generated also increases. In order to offset the effect of variation in the maximum capacities of links in a network, *revenue deficit* can be computed.

Definition 5: (Revenue Deficit) – Revenue Deficit, *RD(TC,AC)* of a route-computation algorithm that accepts a set of connections *AC* from the offered set *TC*, is defined as:

$$RD(TC, AC) = \frac{REV(TC) - REV(AC)}{REV(TC)} = \frac{|TC| \times AHC(TC) - |AC| \times AHC(AC)}{|TC| \times AHC(TC)}$$

5.1.4 PERFORMANCE STABILITY

In dynamic route-computation algorithms, paths for arriving connection requests are calculated instantly. The number of connections that can be established by an algorithm depends not only on the particular set of connection requests that it gets over a period of time but also on the order in which these connection requests arrive. The standard deviation in the number of connections established by an algorithm when the order in which connections arrive is varied, is computed to measure the stability in the performance of a routing algorithm. This metric is called *performance stability* in this thesis. Performance stability indicates how well suited an algorithm is for dynamic route-computation. Lower the value of the standard deviation, more stable is the performance of a route-computation algorithm.

5.2 DERIVATION OF MAXIMUM LOAD

This section presents the derivation of maximum load that should be offered to any route-computation algorithm during simulation. This result is used in Section 6.2.

Definition 6: Full Capacity - Let *FC* denote the full capacity of a network. Full capacity of a network is the summation of the maximum capacities of all links in the network. It can be defined as:

$$FC = \sum_{i=1}^{N} F(L_i)$$

where, $F(L_i)$ is the maximum capacity of link L_i.

Notice that $F(L_i)$ is equal to $A(L_i)$ in an unoccupied network.

Definition 7: (Maximum Load) - For a given network, Maximum Load M is the smallest set of connections K such that

$$AHC(K) \times |K| \geq FC$$

Definition 8: (Total Utilization) - Let AC be the set of connections accepted by a route computation algorithm. Let $PHC(C)$ be the number of hops in the path on which connection C is actually established by this route-computation algorithm. Let $APHC(AC)$ be the average over the $PHC(C)$ of all connections C in the set AC. $APHC(AC)$ is defined as:

$$APHC(AC) = \frac{\sum_{C \in AC} PHC(C)}{|AC|}$$

Total Utilization of an algorithm that accepts a set of connections AC, $TU(AC)$ is defined as:

$$TU(AC) = \sum_{C \in AC} PHC(C) = |AC| \times \frac{\sum_{C \in AC} PHC(C)}{|AC|} = |AC| \times APHC(AC)$$

Lemma 1: $AHC(AC) \leq APHC(AC)$.

Proof: Notice that $HC(C)$ is the number of hops in the ideal shortest path that any connection C can be established on. Since $PHC(C)$ is the actual path on which connection C is established it can not be shorter than the ideal shortest path for C, i.e. $HC(C) \leq PHC(C)$. It follows then that $AHC(AC) \leq APHC(AC)$ by the following equation:

$$\forall C : HC(C) \le PHC(C) \Rightarrow \sum_{C \in AC} HC(C) \le \sum_{C \in AC} PHC(C) \Rightarrow \frac{\sum_{C \in AC} HC(C)}{|AC|} \le \frac{\sum_{C \in AC} PHC(C)}{|AC|}$$

$\Rightarrow AHC(AC) \le APHC(AC)$.

Theorem 1: In a given network, for any route-computation algorithm that establishes a set of connections AC,

$$|AC| \le |M|$$

where M is the maximum load of the network and $AHC(AC) \approx AHC(M)$.

Proof : Let AC be the set of connections accepted in the network by any route-computation algorithm. Clearly, $TU(AC) \le FC$, since the number of links utilized in a network can not exceed the total number of links available in the network. Hence, we have from definitions 7, 8 and *Lemma* 1,

$$TU \le FC \Rightarrow AHC(AC) \times |AC| \le APHC(AC) \times |AC| \le FC \le AHC(M) \times |M| \Rightarrow |AC| \le |M|.$$

M is a very practical limit on the maximum load that should be offered to any route-computation algorithm during simulation in a given network because even the perfect route-computation algorithm will reject some connections if it is offered more than $|M|$ number of connections, where the average hop counts of both the sets are approximately equal. During most simulations the connection requests are generated randomly and therefore the average hop count of the connections across various sets are approximately equal if the set of connection requests generated is large. For example, it was observed that the maximum difference between the average hop counts of various sets of connections that were generated in the simulation experiments for this thesis, was 0.02.

CHAPTER 6

PERFORMANCE EVALUATION OF ROUTE-COMPUTATION ALGORITHMS

This chapter presents simulation results. In Section 6.1, the performance of Interest-based algorithm is compared with four other routing algorithms presented in chapter 4, in terms of the usable network capacity each can support. Their blocking probabilities and revenue deficits with increasing load are compared in Section 6.2. To observe the effect of distribution, the usable network capacities achieved by them was observed when updates in the network state were delayed. The results appear in Section 6.3. In order to evaluate their suitability for use as a dynamic route-computation algorithm, the performance stability values of all the routing algorithms was compared when the same set of connections are presented to them in different orders. The results of this comparison appear in Section 6.4. Finally, a comparison between the performance of the Availability-based and the Cost-based algorithms is summarized in Section 6.5 to demonstrate that the use of a link's physical length is detrimental to the performance of a route-computation algorithm in an optical network.

A16-node NSFNET national network shown in Figure 2.1 was used for simulation. This network has 25 bi-directional links. Each link has 64 wavelengths in each direction. The labels on the links denote its physical length.

The connection requests were generated randomly. The average hop count of the connections generated was observed to be between 2.255 and 2.275. The connections once established, remained in the network for the duration of the experiment. A connection was blocked if there was no free path available for it. In the case of periodic updates when the source nodes did not have a perfect knowledge of the network state, connections also blocked if the path computed for it was occupied by other connections. Only one attempt was made to establish any connection.

6.1 USABLE NETWORK CAPACITY

This section reports the usable network capacity that was achieved by each of the five algorithms. As defined in Section 5.1.2, the usable network capacity is the number of connections that can be accepted by the network with a 1% blocking probability.

Figure 6.1 shows that a 1% blocking occurs for the Minimum-hop, the Cost-based, the Availability-based and the Future-based algorithms at approximately 867, 892, 916, and 917 connections respectively. For the Interest-based algorithm, it occurs at approximately 950 connections. This is a 3.6% improvement over the Future-based algorithm, which performs better than the other three algorithms.

In addition, it is observed that the average hop count (*AHC*) of the connections established by the Interest-based algorithm is always higher than that of the connections established by the Availability-based and the Future-based algorithms. This implies that the revenue generated from the network by using the Interest-based algorithm is even higher as compared to the other four algorithms.[1] If we consider the usable network

---

[1] It may be emphasized here that *AHC*, as defined in Section 5.1.3, does not depend on the particular algorithm used but is the property of the network and the set of connections accepted in the network. For example, if the set of connections established by the Future-based algorithm is *SF* and that established by

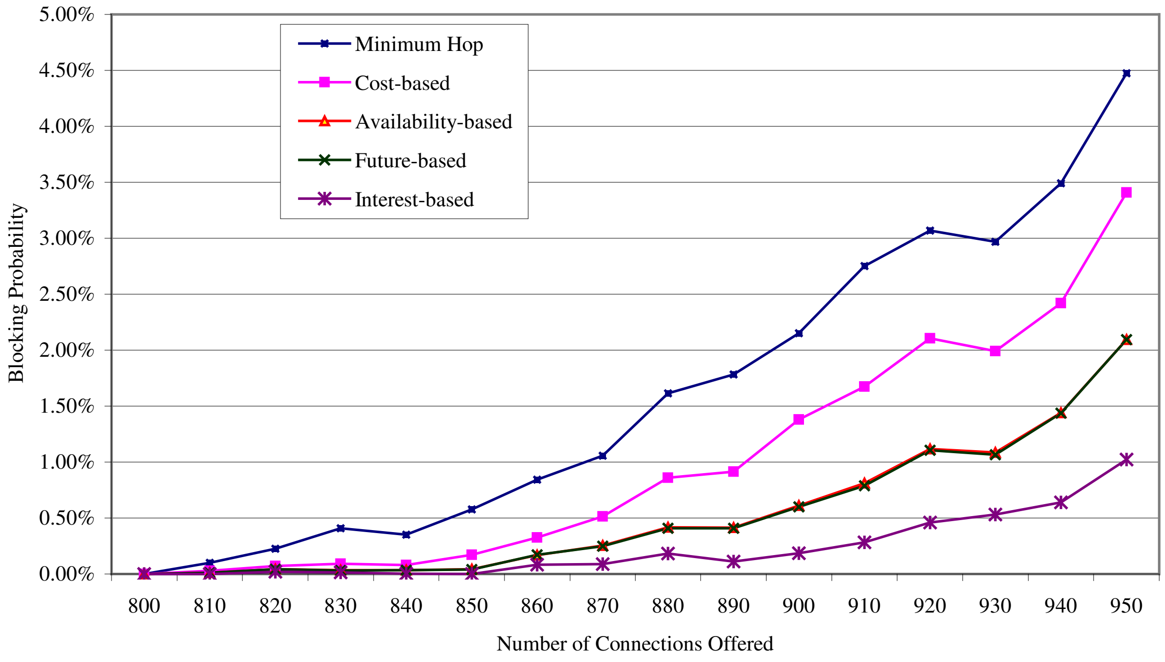

Figure 6.1: Usable Network Capacity (Blocking Probability)

capacity to be the number of connections accepted by the network before it starts losing 1% of its potential revenue, then the usable network capacity achieved by using the Interest-based algorithm is 3.73% higher than that of the Future-based algorithm. Figure 6.2 shows that a 1% revenue deficit occurs for the Minimum-hop, the Cost-based, the Availability-based, and the Future-based algorithms at approximately 860, 880, 911, and 911 connections respectively. For the Interest-based algorithm, it occurs at approximately 945 connections.

Interest-based algorithm is *SI*, and if *AHC(SI)* ≥ *AHC(SF)*, then the connections in the set *SI*, if established in an empty network on their corresponding ideal shortest paths will occupy more links than that occupied by the connections in the set *SF*. *AHC* of a set of connections does not depend on what actual path was used to establish the connections in this set.

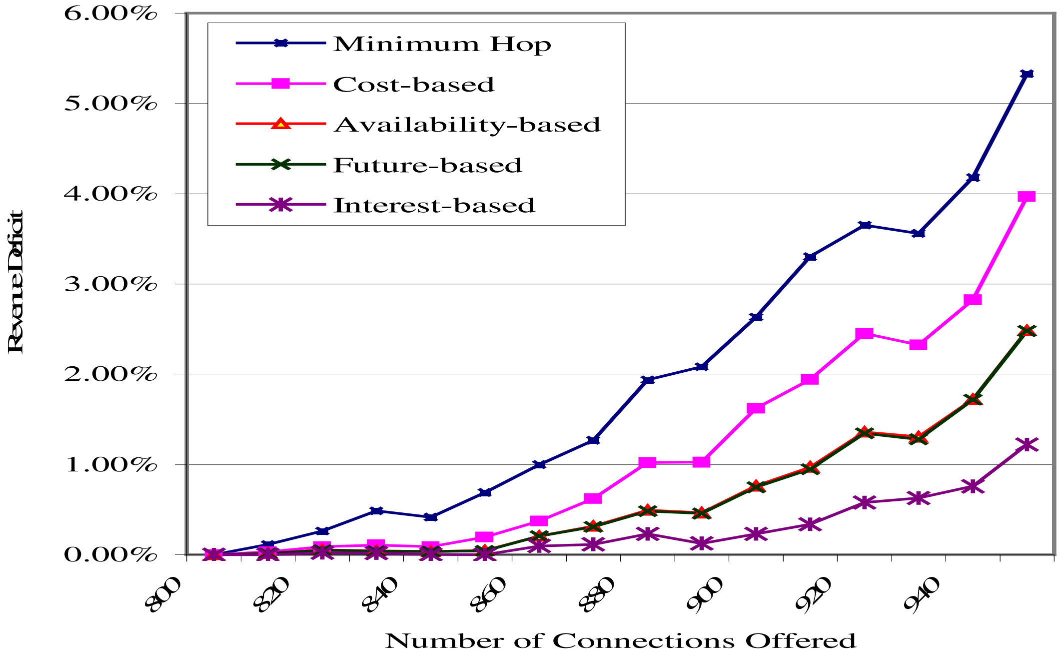

Figure 6.2: Usable Network Capacity (Revenue Deficit)

6.2 BLOCKING PROBABILITY

This section reports the blocking probability observed in the network by using each of the five route-computation algorithms. The load on the network was varied from 800 connections to 1400 connections in steps of 50. It was observed that the average hop count (*AHC*) of the connections offered to the network was between 2.255 and 2.275. The total number of links in the network was 25 (Figure 2.1). Each link had a maximum capacity of 128 wavelengths. Hence, as derived in Section 5.2, the maximum number of connections that any optimal algorithm can establish in this network is less than or equal to 1420.

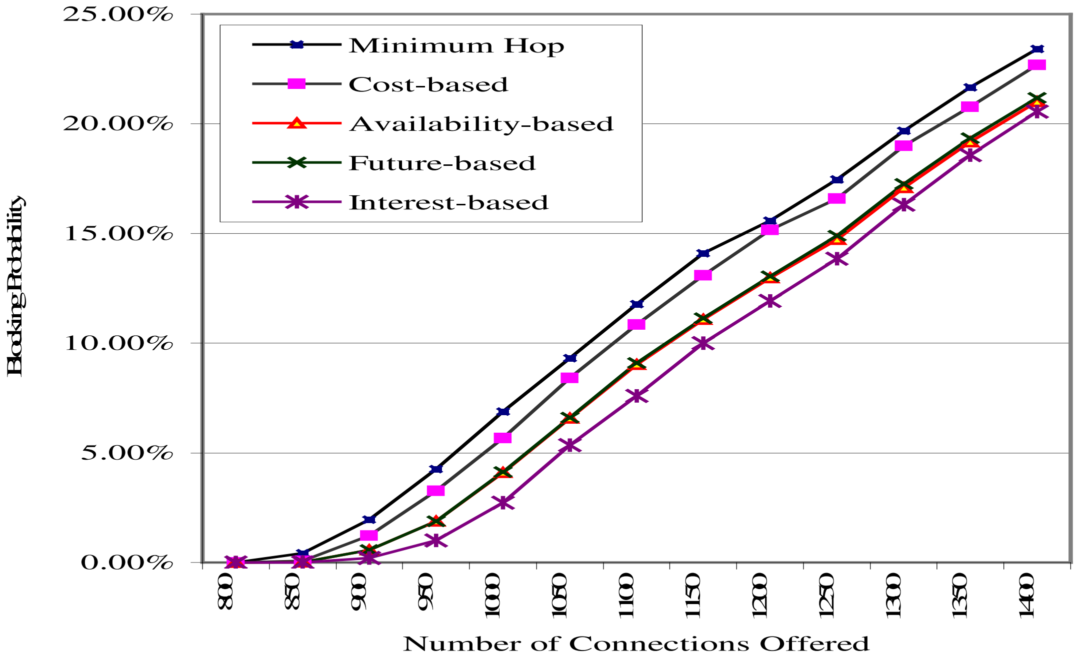

Figure 6.3: Blocking Probability vs. Load

As can be seen from Figure 6.3, the blocking probability of the network is always lower when using the Interest-based algorithm, compared to the other four algorithms considered in this thesis. Figure 6.4 shows that the revenue deficit as defined in Section V(A) is also always lower when using the Interest-based algorithm. Because the average hop count of the connections established by the Interest-based algorithm is always higher than the other algorithms, the improvement achieved by using the Interest-based algorithm, in terms of lowering the revenue deficit, is even better.

As for comparing the performance of other algorithms, the Availability-based and the Future-based algorithms, both offer similar blocking probabilities. For loads less than 950 connections, the Future-based algorithm performs marginally better than the Availability-

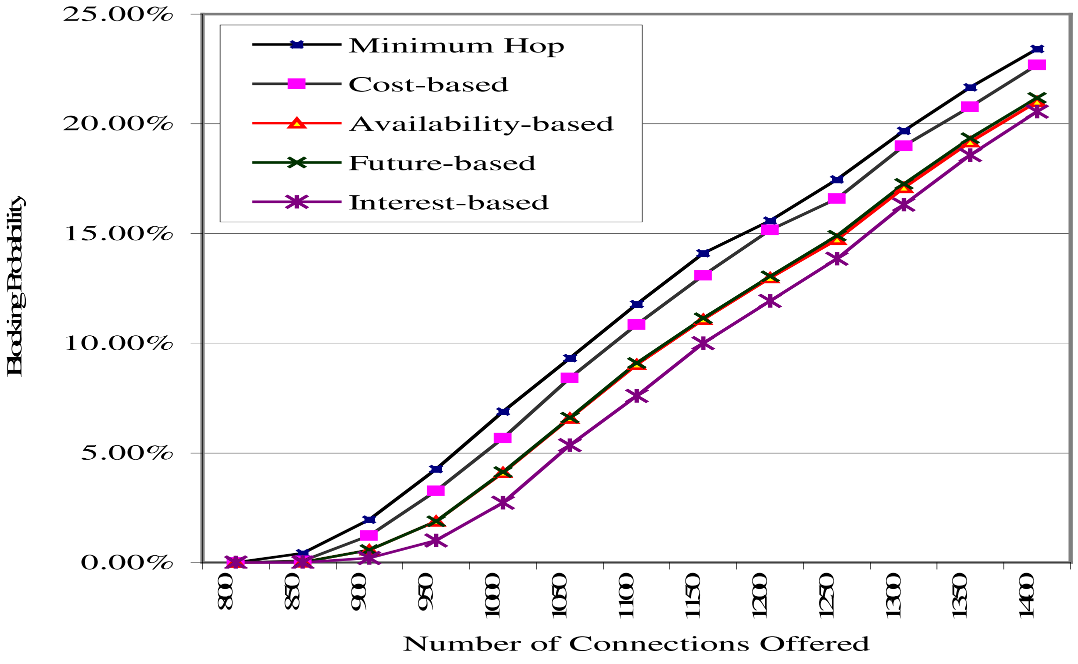

Figure 6.4: Revenue Deficit vs. Load

based algorithm. At a load of 950 connections, they both offer the same performance, and for loads beyond 950 connections, the Availability-based algorithm performs marginally better than the Future-based algorithm. This shows that the Future-based algorithm does not always perform better than the Availability-based algorithm.

6.3 EFFECT OF DISTRIBUTION

This section compares the usable network capacities achieved by various algorithms in a distributed environment. A distributed environment has been modeled by delaying the receipt of updates in the network state at the source nodes. Specifically, the updates

Update Frequency	Minimum-Hop	Cost-based	Availability-based	Future-based	Interest-based
10	837.2	885.6	919.6	919.6	952
20	817.7	880	911.1	910.1	938.1
30	776	883.7	899.8	899.3	925.7

Table 6.1: Usable Network Capacity with Delayed Updates

Algorithm	Accepted	Hop Count	Revenue	Performance Stability
Minimum-Hop	882.2667	2.2663425	1996.933	2.234
Cost	885	2.269972	2008.767	6.316
Availability	895.9333	2.266092	2030.267	2.424
Future	896.0667	2.265938	2030.433	2.516
Interest	899.5667	2.269146	2041.233	0.172

Table 6.2: Performance Stability at n=900

Algorithm	Accepted	Hop Count	Revenue	Performance Stability
Minimum-Hop	906.9667	2.252037	2042.333	4.129
Cost	911.2667	2.264284	2062.833	11.287
Availability	932.9667	2.257473	2106.2	4.237
Future	933.0333	2.257236	2106.133	4.396
Interest	941.9333	2.263742	2132.333	1.022

Table 6.3: Performance Stability at n=950

are delayed until *k* connections are established in the network. Various algorithms were simulated for *k* = 10, 20 and 30. The results appear in Table 6.1.

As can be seen from Table 6.1, the Interest-based algorithm continues to achieve higher usable network capacity even when the updates are delayed for 30 connection establishments. This show that the Interest-based algorithm can be implemented in a distributed environment without much loss in performance. As for the other algorithms, it was observed that the Minimum-hop algorithm is worst affected by distribution.

6.4 SUITABILITY TO DYNAMIC ROUTE-COMPUTATION

This section reports the variation in the performance of each of the five algorithms when the same set of connections was offered to them in different orders. Specifically, *n* connection requests were generated randomly. These were offered to each of the five algorithms, in the order that these requests were generated. Then, the order of connection request arrivals was changed so that they appeared in the increasing order of their hop counts. Once again, the order of arrival of these connections was changed so that now they appeared in the decreasing order of their hop counts. The standard deviation in the total number of connections accepted by each of them was measured. As mentioned in Section 5.1.4, this metric is called performance stability. To observe the performance stability at a given load *n*, *n* connection requests were generated multiple times and the standard deviation in the performance of various algorithms was recorded. The standard deviation reported here for any value of *n* is the average taken over many such sets of *n* connection requests. This experiment was conducted for *n=900* and *n=950*. These load values were chosen because this is the range of load in which a network may be operating and every algorithm blocked some connections for values of load in this range.

As can be seen from Tables 6.2 and 6.3, the Interest-based route-computation algorithm not only established more connections and offered higher revenues, but also the variability in its performance was very less as compared to the other algorithms.

It is also interesting to note here that the variability in the performance was highest for the Cost-based algorithm. On this metric, the Minimum-hop algorithm outperformed the Cost-based algorithm.

6.5 PERORMANCE SUMMARY OF THE INTEREST-BASED ALGORITHM

This section summarizes the performance of the Interest-based routing algorithm.

By approximating the interference among temporally overlapping connection requests, the Interest-based algorithm offered a minimum of 3.6% more network capacity as compared to the other algorithms in use today. The blocking probability of the Interest-based algorithm was always lower than the other algorithms for a wide range of load. Revenue generated by the Interest-based algorithm was even better as compared to the other algorithms. Further, because every source node establishing a connection in the network knows about other connections being established simultaneously from the value of interest, the Interest-based algorithm continues to perform well even if the network state updates are delayed.

A recent work on routing in circuit-switched networks, called Minimum Interference Routing Algorithm (MIRA) [16] was brought to the author's attention by anonymous referees. MIRA is also based on minimizing the interference among various connection requests. However, MIRA is based on a predefined importance assigned to various source-destination pairs and it does not take into account the connections being established simultaneously in the network as is done in the Interest-based algorithm. Following is a qualitative comparison between MIRA and the Interest-based algorithm:

- Selection of critical links – MIRA uses the remaining capacity in links and a predefined importance of various source-destination pairs to determine the set of critical links.

  The Interest-based algorithm computes the set of critical links by using the current set of pending connection requests and the remaining capacity in the network.

- Weight of critical links – MIRA assigns weights to critical links proportional to a predefined importance of the source-destination pairs, whose max-flow decreases by establishing a connection on this link. If this importance assignment is not dynamic as is the case in its current version (the authors of MIRA do not talk about this issue in their paper), then MIRA does not adapt itself well with changing traffic patterns. If this importance assignment is to become dynamic, a mechanism similar to that proposed in thesis needs to be developed for MIRA too.

  The Interest-based algorithm assigns weights to critical links proportional to the excess number of connection requests that are expected to use this link (number of connection requests expected to use this link minus the remaining capacity on this link), referred to as $BW(L_i)$ in this thesis, and inversely proportional to the remaining capacity on this link. The Interest-based algorithm therefore can adapt itself well with changing traffic patterns.

- Weight of non-critical links – In MIRA, the weight of all non-critical links is zero. So, any number of non-critical links may be included in a path.

  In the Interest-based algorithm, a non-zero weight is assigned to a non-critical link that is inversely proportional to its remaining capacity and this weight is less than that for any critical link in the network having the same amount of remaining capacity. Therefore, the number of non-critical links that may be used in a path is limited and it helps prevent overuse of network resources, which may prove useful for connection requests arriving later.

Because MIRA does not take into consideration the concurrent connections being established in the network and uses a predefined importance of various source-destination pairs in deciding the importance of critical links, it may not perform as well as the Interest-based algorithm. Further analysis and simulation are necessary to confirm this conclusion.

6.6 HARMFUL EFFECTS OF USING LINK LENGTH

This section illustrates the harmful effects of using a link's physical length in a route-computation algorithm, based on the performance metrics used in this paper. The performance of the Availability-based algorithm was compared with the Cost-based algorithm. The only difference between these two algorithms, as described in Section 3.3.4, is the use of a link's physical length by the Cost-based algorithm.

It can seen from Figure 6.1 that the Availability-based algorithm provided 2.7% more usable network capacity than the Cost-based algorithm. From Figures 6.3 and 6.4, it can be seen that the blocking probability and the revenue deficit of the Availability-based algorithm is always lower than that of the Cost-based algorithm. Table 6.1 shows that this trend continues in the distributed environment, when the network state updates arrive periodically. The biggest drop in performance was observed while evaluating the suitability of an algorithm for dynamic route-computation. Tables 6.2 and 6.3 show that using a link's physical length in the definition of link weight caused huge variation in its performance. The standard deviation of the Cost-based algorithm was between 6.32 and 11.29, whereas, for the Availability-based algorithm, it was between 2.42 and 4.24. This experiment demonstrates that the use of a link's physical length decreases the performance of a route-computation algorithm and additionally, causes a great variation in its performance.

The link's physical length is not helpful for a route-computation algorithm in an optical network because it does not reflect the interference a connection may expect when using any particular link, with other competing connections. Route-computation algorithms that use links' lengths in finding a shortest path, tend to minimize the total physical length of the path that an individual connection is established on, which does not help optimize the total number of connections established. Instead, it turns out to be harmful. The harmful effects of using a link's physical length can be illustrated with a simple analogy to the road traffic network[10]. If every driver tries to minimize his/her delay, the average delay for the whole network may increase. Even worse, in the case of heavier loads, there might occur a traffic jam! Minimizing total path length for a connection is equivalent to minimizing the propagation delay for every connection. The difference in the propagation delays between various paths through an optical network is so low that it is unnoticeable to applications. Therefore, there is no motivation for choosing a path with minimum physical length for any connection in an optical network.

This thesis also conjectures that the use of any fixed property of a link in the definition of link weight will have a similar adverse effect on the performance of a route-computation algorithm. Again, this is because, fixed properties have no relation to the amount of interference expected on a link, which inherently is a dynamic property of a link.

CHAPTER 7

CONCLUSION AND FUTURE WORK

This chapter concludes the thesis and describes future work in this direction.

7.1 CONCLUSION

This thesis proposed an efficient algorithm to approximate the interference between various connection requests in a network. It also proposed a new way to use this information during route-computation. These two improvements together were referred to as the Interest-based routing. Simulation showed that the Interest-based routing offered a better usable network capacity and a lower blocking probability than other algorithms considered in this thesis. Because route-computation problem is NP-Complete, all the existing algorithms for route-computation are heuristics. It is not only important for a heuristic to perform well on some inputs but also to offer a stable performance across a wide variety of inputs. It was shown that the Interest-based algorithm offers more stable performance as compared to other route-computation algorithms over a variety of inputs.

This thesis proposed a simple way to compute the maximum load that should be offered to any route-computation algorithm during simulation on a given network. This was used during the simulation experiments conducted for this thesis.

It was also pointed out in this thesis that using a link's physical distance in a route-computation algorithm is harmful. It is a widely used criteria today and it is hoped that this result will discourage its use.

7.2 FUTURE WORK

This thesis assumed that connections, once established became permanent. The behavior of Interest-based algorithm when connections are established and torn down simultaneously needs to be analyzed. Further simulation is needed with different network topologies in order to instill more confidence in the superiority of the Interest-based algorithm with other algorithms. A further analysis and simulation is necessary to compare the performance of the Interest-based algorithm with max-min based algorithms, in particular with MIRA.

This thesis suggested that a better route-computation algorithm should not only offer a higher usable network capacity in a network but the variability in its performance also should be lower. This emanates from the fact that usable network capacity offered by a route-computation algorithm is a random variable that depends on the network topology, the sequence of connection request arrivals and the route-computation algorithm used to compute the routes. The author plans to investigate if a general theory can be developed to compute the distribution of this random variable.

BIBLIOGRAPHY

[1] B. G. Cortez and J. King, “The emerging intelligent optical network: now a reality,” Optical Fiber Conference 2002.

[2] C. Hamilton, R. Ward and M. Wendel, “Deployment of intelligent optical switches in North America,” Optical Fiber Conference 2002.

[3] G. Ash, “Dynamic routing in telecommunications networks,” McGraw-Hill 1998.

[4] R. J. Gibbens, F. P. Kelly and P. B. Key, “Dynamic alternative aouting – modeling and behavior,” Proc ITC 12, paper no. 3.4A.3, Torino, June 1988.

[5] K. Kompella et. al., “Routing extensions in support of generalized MPLS,” draft-ietf-ccamp-gmpls-routing-04.txt, Internet Draft, Work in Progress, April 2002.

[6] C. Assi, A. Shami, M. A. Ali, R. Kurtz, and D. Guo, “Optical networking and real-time provisioning: an integrated vision for the next-generation Internet,” IEEE Network, July/August 2001.

[7] P. H. Ho and H. T. Mouftah, “A novel routing protocol for WDM mesh networks,” Optical Fiber Conference 2002.

[8] J. Spath, “Resource Allocation for dynamic routing in WDM networks,” Proc of SPIE, OptiComm 2000, Richardson, October 2000.

[9] C. F. Hsu, T. L. Liu and N. F. Huang, “On adaptive routing in wavelength-routed Networks,” Optical Networks Magazine, Jan/Feb 2002.

[10] F. P. Kelly, “Network routing,” Philosophical Transactions: Physical Sciences and Engineering, vol 337, Issue 1647, Complex Stochastic Systems, Dec 15, 1991.

[11] F. P. Kelly, “Blocking probabilities in large circuit-switched networks,” Advances in Applied Probability, vol. 18, no. 2, June 1986.

[12] S. Even, A. Itai and A. Shamir, “On the complexity of timetable and multicommodity flow problems,” SIAM Journal of Computing, vol. 5, no. 4, Dec 1976.

[13] H. Zang, J. P. Jue, L. Sahasrabuddhe, R. Ramamurthy and B. Mukherjee, "Dynamic lightpath establishment in wavelength-routed WDM networks," IEEE Communications Mag., September 2001.

[14] J. Strand, R. Doverspike and G. Li, "Importance of wavelength conversion in an optical network," Optical Networks Mag., vol. 2, no. 3, May/June 2001.

[15] H. Zang, J. P. Jue and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," Optical Networks Mag., vol. 1 , no. 1, January 2000.

[16] M Kodialam and T V Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering," IEEE INFOCOM 2000.