ABSTRACT

WASHINGTON, ALICIA NICKI *Performance Analysis of Traffic-Groomed Optical Networks.* (Under the direction of Dr. Harry Perros)

Wavelength division multiplexing (WDM) has emerged as a viable solution to the increasing bandwidth demands of current backbone networks. Traffic grooming allows the resources available on a single wavelength to be utilized by multiple traffic streams. These networks will support large amounts of traffic from various sources. In view of this, our research focuses on the analysis of such types of networks, and in particular on a traffic-groomed tandem optical network that supports multiple sources each with unique resource requirements. We developed a decomposition algorithm for the analysis of this network using a modified version of Courtois' method. This work was extended to analyze tandem networks with a large number of wavelengths and resources per wavelength using a modified version of the recursion developed by Nilsson et. al. Finally, we extended our work to a traffic-groomed tandem optical network employing alternate routing and supporting traffic from multiple sources. For each method, numerical results show that, overall, our approximation provides good accuracy.

PERFORMANCE ANALYSIS OF TRAFFIC-GROOMED OPTICAL NETWORKS

by

ALICIA NICKI WASHINGTON

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Doctor of Philosophy

COMPUTER SCIENCE

Raleigh
2005

Approved By

Dr. Michael Devetsikiotis	Dr. Rudra Dutta
Dr. Arne Nilsson	Dr. Harry Perros Chair of Advisory Committee

DEDICATION

This work is dedicated to my parents, Reginald and DeLois Washington, for their constant support.

PERSONAL BIOGRAPHY

Alicia Nicki Washington is a native of Durham, NC. She received her Bachelor of Science in Computer Science from Johnson C. Smith University in Charlotte, NC in 2000. She received her Master of Science in Computer Science from North Carolina State University in 2002. She is a 2000 David and Lucille Packard Fellow and a 2003 NASA Harriet G. Jenkins Pre-doctoral Fellow.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Research Motivation

The ever-increasing demands of the Internet are due, in part, to the emergence of applications such as video conferencing, television channel distribution, multimedia graphics, and Internet browsers. Recent advances in telecommunication networks have allowed wavelength division multiplexing (WDM) to emerge as a viable solution to these demands. In WDM, multiple wavelengths can be transmitted simultaneously over a single optical fiber. Each wavelength is capable of carrying traffic streams at Gb/s rates, thereby providing a single fiber with a total throughput in the range of Tb/s.

In a wavelength-routed optical network, traffic is transported over lightpaths. A lightpath exclusively occupies an entire wavelength on each hop of the source-destination

path. This is the same wavelength through out the optical network, or different wavelengths if converters are available at the optical nodes. However, many traffic streams do not require the bandwidth of an entire wavelength. In fact, a traffic stream may need only a small fraction of the bandwidth. This results in a large amount of underutilized resources, as the majority of the lightpath's bandwidth is unused but unavailable for use by other traffic streams.

To resolve this, the bandwidth of a lightpath is divided into lower sub-rate units so that it can carry traffic streams transmitted at lower rates. A traffic stream uses one or a multiple of these sub-rate units. This allows the unused bandwidth to be available to other traffic streams in the network. This technique is known as *traffic grooming*. In traffic grooming, lower-rate traffic streams are multiplexed and de-multiplexed onto higher capacity WDM wavelengths, in order to improve wavelength utilization [8] and meet network design goals such as cost minimization [29]. The lowest sub-rate stream carried, hereafter referred to as a *sub-wavelength unit*, defines a lightpath's granularity.

Because these networks carry large amounts of traffic, alternate routing methods are designed in order to allow traffic to be properly re-routed from source to destination in the event of certain events, such as link blocking, or even link failure. All traffic in the network is assigned a primary path, upon which it initially attempts to be routed. If the call cannot be routed on this path, then it is re-routed to an alternate path. If there are multiple alternate paths available, then they are attempted in a specific order until the traffic is either accepted or blocked from the network.

In this study, we focus on the analysis of traffic-groomed tandem optical networks supporting multi-rate traffic. We begin with a tandem network composed of a single wavelength, which supports traffic from multiple sources each with unique resource requirements. The optical network is modeled as a tandem queueing network of multi-rate Erlang loss nodes with simultaneous resource possession. The queueuing network was decomposed into two-node subsystems, and each subsystem was analyzed using a modified version of Courtois' method. The results from each subsystem were appropriately combined to analyze calls in the original network.

Next, we extended our research to tandem optical networks with a large number of wavelengths and a large number of servers per wavelength. We assumed traffic bifurcation was permitted, which allowed us to view each link of the network as a single pool of resources. The network was decomposed into two-node subsystems as before, and each subsystem was analyzed using a modification of the recursion developed by Nilsson et. al [19].

Finally, we extended our research to the analysis of a tandem employing alternate routing techniques. The network was decomposed into subsystems based on the lightpaths existing on a wavelength. Each subsystem was then analyzed using an extension of Basharin and Kurenkov's approximation and the Nilsson et. al recursion.

1.2 Organization of the Dissertation

The remainder of this dissertation is organized into five chapters. In Chapter 2, we review in detail the most important literature pertaining to the calculation of call blocking probabilities in WDM networks, traffic-groomed WDM networks, and WDM overflow systems. In Chapter 3, we introduce the decomposition algorithm developed for the analysis of a traffic-groomed tandem WDM network. The algorithm uses a modified version of Courtois' method. We also provide some numerical examples and compare the results obtained to both simulation data and the single-node decomposition method assuming link independence. In Chapter 4, we extend the decomposition algorithm to a large number of wavelengths and sub-wavelength units. We introduce a modified version of the recursion developed by Nilsson et. al for the analysis of the traffic-groomed tandem optical network. Numerical examples are provided and we compare the results obtained to the simulation data and single node decomposition data. In Chapter 5, we present a decomposition method for the analysis of networks employing alternate routing, i.e. overflow systems. Finally, in Chapter 6, we conclude our research and present future areas of research.

Chapter 2

Literature Survey

In this chapter, we review the literature for the analysis of traffic-groomed WDM networks and overflow systems. The literature is grouped into logical sections according to the subject matter.

2.1 Call Blocking Probabilities in WDM Networks

The calculation of call blocking probabilities in a WDM network where a single call occupies an entire wavelength has been studied extensively. Birman[4] compared the blocking probabilities of a circuit-switched network and a WDM network, and determined that the blocking experienced by a call in the WDM network is greater. In addition, an approximation algorithm was developed, which determined the blocking probability of traffic

on all source-destination paths within the WDM network, assuming random wavelength assignment. The results were compared for fixed and least-loaded routing.

Conway et. al [6] developed three decomposition methods for the exact analysis of large multi-facility networks that exploit the sparse nature of the demand matrix associated with the network. The first algorithm decomposes the network into two separate subsystems. Each subsystem is analyzed using the normalization constant recursion developed by the authors. The results from the two subsystems are then used to determine the blocking probability of all calls in the network.

The second decomposition algorithm extends the work of the previous to M subsystems. The normalization constant recursion is used to analyze each subsystem and the results are combined appropriately to analyze the original network. The third decomposition algorithm is an incremental decomposition method that successively analyzes each subsystem. The proposed methods are shown to have lower computational and storage requirements than previous algorithms.

Zhu et. al [29] developed a decomposition method for the analysis of WDM networks with fixed or alternate routing. The blocking experienced by an individual path in the network was determined by decomposing the path into a set of tandem segments. The effective arrival rates of calls within a segment account for the link correlation and blocking in the original path. Each segment was analyzed in isolation using an approximate closed-form solution. The results of each segment were combined to determine the blocking probability of calls in the original network traversing more than one segment. The effective

arrival rates of each segment were then updated and each segment was again analyzed in isolation until the blocking probabilities of the path converged.

Ramesh et. al[20] extended the work of Zhu et. al [29] to analyze a WDM network with multiple classes of calls as well as multicast traffic. To analyze the multi-class traffic, the authors created an equivalent system that aggregates the multi-class traffic into a single-class. The analysis of the equivalent system is then performed using the path decomposition method described in [29]. The authors extended this work to measure the performance of multi-class networks with both multicast as well as unicast traffic.

The problem of calculating call blocking probabilities in a traffic-groomed WDM network has gained attention in recent years. Thiagarajan and Somani [24] modeled a dynamic traffic-groomed WDM network assuming both link independence and wavelength independence. The network consisted of two node types: Wavelength-Selective Crossconnect (WSXC) nodes and Wavelength-Grooming Crossconnect (WGXC) nodes. Both nodes have the capability to groom traffic streams to (from) added (dropped) wavelengths. WSXC nodes contain Optical Crossconnects (OXC) that can only switch full wavelengths from an input port to a specific output port. WGXC nodes have the added functionality of switching low-rate traffic streams from an input port at one wavelength to an output port at another. It is assumed that, if a path traverses through one or more intermediate WGXC nodes, then the path contains more than one lightpath.

The analysis of the model provides the overall network blocking probability of a class r call in the network, r=1,2, …R. The analysis was performed iteratively, beginning with the calculation of the link arrival rates for each link in the network. Each link was then analyzed

as a single-wavelength link, by creating the generator matrix Q^*, corresponding to all state transitions within the link. Solving the set of equations $XQ^*=0$, $Xe=1$ provided the stationary probability of all states within the link. The results from each link were used to determine the blocking probability of a single source-destination path with multiple wavelengths. The overall blocking probability of a class r call in the network was determined using the results of each individual path analysis. The authors found that, although the method is simple, efficient, and, for most cases, converged within a few iterations, it did not always converge.

In [23], Thiagarajan and Somani extended the work in [24] to account for the capacity correlation that exists between links. The authors developed a model for the analysis of a multi-hop path consisting of a single wavelength. Beginning with a two-hop model, the state of the system is defined as the number of traffic streams of a specific class that are present on the one-hop links as well as the path continuing from the first link to the second. The steady-state probability of the two-hop model was determined by numerically solving the generator matrix, Q^*.

The analysis of an h-hop path was determined by first dividing the path into a set of two-hop paths, such that each successive two-hop path originates at the first node and terminates at node i, where $2 \leq i \leq h$. The steady-state probability of calls in each two-hop path was determined using the numerical solution, and the blocking probability of the h-hop path is recursively calculated using the results of each successive two-hop path. The authors found that the technique provided a better accuracy than the link independence algorithm.

Srinivasan and Somani [22] developed an analytical model to measure the call blocking probabilities without accounting for the exact call distribution. The network was

modeled as a homogenous Trunk Switched Network. Within this two-level network model, each trunk represented a wavelength. The analysis was performed in three steps. First, assuming a two-link path model, the state space was constructed based on the number of busy channels due to calls on the first link, second link, and those continuing from the first to second link. The steady-state probability of the system was then calculated recursively.

Second, the trunk distribution is recursively calculated using the channel distribution. The state space of the trunk distribution is defined as the number of busy trunks on the first link, second link, and the path continuing from the first to second link. Finally, the blocking probability of a *z*-link path is recursively calculated by viewing the path as a two-hop path consisting of the first *z*-1 links in the first hop and the last two links in the second hop. The trunk distribution is used in this recursion.

Mosharaf et. al [18] developed an analytical model to determine the number of wavelengths necessary to provide a specific grade of service for dynamic traffic-groomed WDM networks. For a single link consisting of multiple wavelengths, the state of the system is determined as the number of class *r* customers present on each wavelength. The product-form solution is provided for the steady state probability of calls within the system. The authors showed that the calculation of the normalizing constant requires extensive computation. As a result, an approximation to the normalizing constant was developed, assuming low traffic loads. The results were compared, for both random and first-fit allocation policies, where it is determined that the approximation provides better results for the case of first-fit allocation. The approximation was then used to determine the minimum number of wavelengths required to provide a specific grade of service for a link.

Xin et. al [26] measured the end-to-end blocking probability of a class *r* call in the network using the Erlang-fixed point approximation, assuming link independence. Within the network, the arrival of a new call may trigger the establishment of a new lightpath. The authors modeled a single-wavelength link as a multi-rate link queue consisting of a number of servers. The arrival of each call is converted into micro calls, such that each micro call can be carried by one server of the multi-rate link queue. The individual link blocking probabilities were then determined using the number of call types, offered load, total number of servers, and the probability mass function.

Zhu et. al [28] developed a novel graph model to analyze the blocking probability of a traffic-groomed WDM network. First, an auxiliary graph is constructed according to the current network configuration. For each arriving call, the shortest path from the source to destination node is determined. The call is appropriately routed along either a pre-existing or newly formed lightpath of the route. The auxiliary graph is then updated to reflect the appropriate changes within the network. The algorithm is compared against different grooming policies. By comparing the average utilization and the blocking probabilities of each policy, the authors determined that an adaptive grooming policy, which switched between the two best performing policies according to the network state, provided better performance than any of the individual grooming policies.

2.2 Call Blocking Probabilities in Erlang Loss Queues

There are several elegant solutions for the analysis of a single multi-rate loss node. Kaufman [14] and Roberts developed a one-dimensional recursion for the analysis. This work was

preceded by the work of Fortet and Grandjean. The recursion, which is applicable in the case of the complete-sharing policy, tracks the total number of busy servers, thereby reducing the amount of storage required. It is shown that the storage requirements are dependent on the maximum demand of all call types in the system.

Nilsson et. al [19] observed that the work by Kaufman and Roberts suffered from potential numerical stability problems. As a result, the authors developed a recursive function that was derived from the work of the previous authors' and the single-rate Erlang-B recursion. It was found that the recursion is of the same complexity as the Kaufman-Roberts recursion and the results are numerically stable.

Dziong and Roberts [9] developed a recursive procedure for the joint distribution of the number of busy servers in each Erlang loss node. While providing an exact solution, the use of the joint distribution increased the computational complexity of the analysis. As a result, the authors proposed an approximation based on the multivariate normal distribution. The approximate method took into account the correlation between links.

Iversen et. al [12] use the generalization of the recurrence developed for the Erlang-B equation to create a recursive algorithm for the analysis of a multi-rate Erlang loss node. The algorithm developed determines the probability of k servers busy on a single link. At each step of the recursion, the algorithm computes the normalized values. As a result, the recursion is both numerically stable as well as simple, and requires a small amount of storage.

2.3 Overflow Analysis in Erlang Loss Queues

The analysis of overflow systems, where blocked calls arriving to a primary Erlang loss system overflow to a secondary system, has been extensively studied. The majority of this work has focused on systems where all calls require a single resource. In [13], Iversen surveyed the various techniques for the analysis of overflow systems. The Equivalent Random Theory (ERT) method, also known as the Wilkinson method, assumes a number of traffic streams, characterized by their individual mean and variance, are offered to a common Erlang loss system consisting of a number of channels. The total mean and variance of the traffic in this system is defined as the sum of the individual means and variances. It is assumed that the total overflow of this system can be represented by an equivalent Poisson flow arriving to an equivalent system consisting of a number of channels. The values for the equivalent flow and equivalent system are approximately calculated. The overall blocking of the equivalent system, determined using the Erlang-B equation, approximates the overall blocking in the original system. The individual blocking probabilities of each traffic stream are determined using parcel blocking probability techniques.

In addition to these methods, the BPP traffic model which uses the mean and peakedness to analyze overflow traffic, the Sander's method which changes the scale of the state probabilities so that the peakedness becomes 1, and Berkeley's method which fixes either the number of channels or the offered traffic and uses the Erlang-B formula to determine the correct mean of the overflow traffic, are presented as approximation methods for overflow analysis. The work in [11] extends Iversen's survey to include approximations for the calculation of the mean and variance of each individual overflowing traffic stream. In

addition to the IPP method, the Equivalent Capacity method, Approximate Wilkinson Wallenstrom method, Equivalent Random Traffic method, and Equivalent Poisson Stream method are presented.

Kuckzura [15] determined that the Interrupted Poisson Process (IPP) can be used to represent overflow traffic. The IPP is a Poisson process that is interrupted by a random switch that turns the process on and off. This on/off state of the process represents the bursty nature of the traffic arriving to the overflow system. Using this model, the overflow traffic is characterized by three parameters: the arrival process, the on duration, and the off duration of the arrival process. The author determined that the blocking in the secondary system was numerically determined using the appropriate state transitions of the IPP model.

Frederick and Hayward's [10] equivalence method is presented as a simpler alternative to the Wilkinson method. The approximation assumes that a system with n servers, offered traffic A, and peakedness $Z \neq 1$ has the same blocking probability as a system with n/Z servers, offered traffic A/Z, and peakedness $Z=1$.

Alyatama [2] presented an approximate method for the analysis of a WDM network supporting single-rate traffic in an overflow system consisting of multiple wavelengths. No traffic grooming was considered. As a result, a single call occupies an entire wavelength. The network was modeled as an aggregate group of disjoint, single-wavelength subsystems, with a common physical topology. Each subsystem is analyzed in iteration using a numerical method that extends the generalized Erlang-B equation. The mean and variance of the overflow traffic were determined using Riordan's formula. The equivalent Poisson traffic and equivalent number of servers are determined using Frederick and Hayward's

approximation. This process was repeated for each wavelength using both a first-fit as well as random wavelength assignment strategy.

Scherer [21] presented an ERT method for the analysis of an overflow system with finite source traffic. The author first determined an explicit and recursive solution for the analysis of an overflow system with a single primary group. An approximate solution for this system was also determined. The mean and variance of the overflow traffic is determined using an equivalent single group such that the overflow from the equivalent group approximates the overflow from the primary group. The variance of the overflow traffic is determined using an equivalent system to the original. The calculated variance is then appropriately weighted to approximate the variance of the offered traffic from the original primary system. The ERT method used assumes traffic from two primary groups overflow to a common secondary group. The mean and variance of each primary group were determined using either the exact or approximate method. The total equivalent offered traffic from both systems was determined and the overflow of calls from the secondary equivalent group was analyzed as an approximation for the total overflow from the original overflow group.

Akimaru and Takahashi [1] derived an approximation of individual call blocking probabilities in overflow systems, using an IPP model. This method is a suitable approximation to the exact solution developed by Kuczura. A traffic stream is described by its mean offered traffic and peakedness. It is offered to a system consisting of a number of servers. The traffic overflowing from this system is offered to a secondary group, which also services random primary traffic. The total offered traffic and peakedness of the traffic

offered to the secondary system is approximated by an IPP. Using the total blocking of the system, the blocking of each individual stream is determined by assuming the call congestion of the random traffic arriving to the secondary node equals the time congestion. The individual blocking probabilities are then calculated under this assumption. The analysis of this network is extended to multi-overflow systems, where multiple identical overflow traffics are offered to a secondary system, in addition to a single random Poisson traffic. It is assumed that the input traffic is divided into two groups, such that the analysis of each is then performed using the same IPP approximation as with the single overflow case.

Brochin and Pradel [5] used the $MMPP^{(1,\ldots n)}$ model to simultaneously match the first two moments of each individual overflow stream in an integrated services digital network. Multi-class traffic overflows from a direct route to a final route, where the bandwidth on the direct and final route may differ. The overflow process of each traffic stream is represented by a $MMPP^{(1,\ldots,n)}$ process, which accounts for the correlation between both overflows from the same class as well as different classes. The analysis of the system is performed assuming the overflowing streams are serviced by an infinite number of servers. Moment-matching techniques are then used to model each overflow stream as a two-state $MMPP^{(1,\ldots,n)}$ process.

Listanti et. al [16] modeled a switchless network as a number of Passive Optical Networks (PON) interconnected by a passive wavelength router node, with a dedicated wavelength for each PON pair. An additional number of wavelengths, defined as dummy ports, are designated as a common overflow group for all PON pairs. Each PON pair supports traffic from multiple sources with multiple demands. The analysis of the dedicated wavelength between a PON pair is determined using either the generalized Erlang-B equation

or a recursive solution. The average blocking probability of a dedicated primary wavelength is determined from these results.

The analysis of the dummy ports is performed using Frederick and Hayward's approximation. The total mean and variance of the overflow traffic is determined using the individual traffic characteristics from each PON pair in the network. The average blocking probability of a call in the network is then determined recursively. The overall blocking probability of the network is determined, assuming both loss events in the primary and dummy ports are disjoint.

Liu et. al [17] developed an approach for the analysis of the individual moments of the overflow traffic using the combination method, previously developed by the author, and the moment-matching technique. Individual calls, each with a unique resource requirement, arrive to an individual trunk group. The overflow from each trunk group, described by its first two moments, arrive to a common secondary trunk group. The network model allows for both overflow and Poisson traffic to arrive to the common trunk group.

The convolution method is used to determine the blocking experienced by each traffic stream in the network. The equivalent is determined, using Basharin and Kurenkov's method. The analysis of the equivalent system is used to determine the binomial moments, for each traffic stream. The binomial moments are then used to determine the mean and variance of each traffic stream overflowing from the common group.

Chapter 3

Analysis of Traffic-Groomed Optical Networks

In this chapter, we present a decomposition algorithm used to calculate call blocking probabilities in a traffic-groomed tandem optical network. The algorithm decomposes the network into subsystems each consisting of two adjacent nodes. Each subsystem is analyzed exactly using a modification of Courtois' algorithm. The subsystems are analyzed iteratively, and the results are appropriately combined to analyze calls in the original network. The remainder of the chapter is organized as follows. Section 3.1 describes the queueing model used to calculate the call blocking probabilities in the network. Section 3.2 describes the decomposition algorithm, and Section 3.3 presents the numerical comparisons between the algorithm and simulation model.

3.1 The Queueing Network Model

We consider a traffic-groomed optical network composed of N nodes linked in series. Each optical node is an add/drop multiplexer (ADM). That is, it can be configured so that some of the traffic continues through, some is dropped, i.e. delivered to the local users, and some new locally originated traffic is added. The network supports traffic from multiple sources, each with a resource requirement measured in sub-wavelength units. We focus our analysis on a single wavelength that forms a lightpath terminated at each ADM, and we assume that traffic cannot be switched between wavelengths.

We model the optical network under study with a tandem queueing network of multi-rate Erlang loss models with simultaneous resource possession. Figure 1 illustrates a seven-node optical network. Two adjacent nodes are linked by a single fiber with W wavelengths. In our queueing model, we only consider a single wavelength.

As shown in Figure 1, we consider all possible source-destination paths. That is, we assume that calls may arrive at any node i, i=0,1,…6, and be destined for any node j, j=1,2,…6, where $j > i$. All calls, irrespective of their source-destination path, are grouped into R classes. Each class r, $r=1, 2, \ldots R$, has an associated demand d_r measured in sub-wavelength units. A class r call arriving at node i with destination j, $j>i$, will be accepted if d_r sub-wavelength units are available on links $i+1, i+2,..,j$. These sub-wavelength units are assumed to be allocated simultaneously on all the links at the moment the call arrives. When the call departs, all d_r sub-wavelength units on all the links along the source-destination path are simultaneously released.

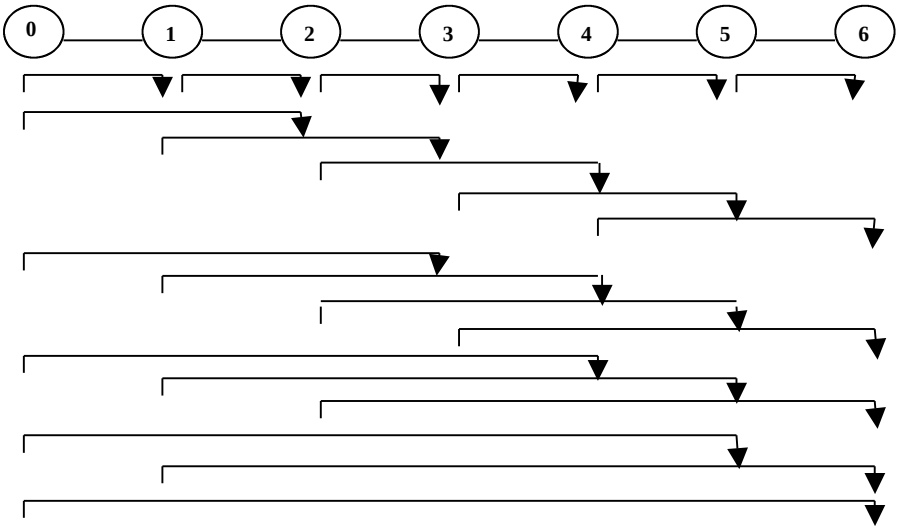


Figure 1. A seven-node optical network with all source-destination paths

Each link of the optical network is represented in the queueing network model by a multi-rate Erlang loss model with *s* servers, where each server represents a sub-wavelength unit. In view of this, the queueing network consisting of six nodes, as in Figure 3, represents the seven-node optical network shown in Figure 1. A customer in the queueing network represents a call. Customers arrive at any node i=0,1,…,5 and terminate at any node j=1,2,...,6, where $j > i$. Class r customers with a source-destination path (i,j) arrive at node i in a Poisson fashion at the rate of $\lambda_{ij,r}$, and they have a holding time of rate $\mu_{ij,r}$. A class r customer simultaneously occupies d_r servers at nodes i, $i+1,\ldots,j$ of the queueing network model. The d_r servers on all these nodes are released simultaneously at the time the customer leaves the queueing network, i.e. it is terminated. A class r arriving customer is accepted into the queueing network, if the required d_r servers on each node along the source-destination path of the customer are available. Otherwise, the customer is blocked, i.e., lost.

We note that in the optical network shown in Figure 1, we assumed that any two adjacent nodes are linked by a single fiber, and each node is an ADM. However, these nodes may not necessarily be physically adjacent. Rather, they can be logically adjacent. That is, there may be other nodes in-between, through which the groomed wavelength passes transparently, i.e., without being terminated.

3.2 The Decomposition Algorithm

In this section, we present the decomposition algorithm developed for the analysis of the above described queueing network. The algorithm decomposes the queueing network into subsystems, each consisting of two adjacent links. Each subsystem is analyzed numerically using a modified version of Courtois' method [7]. As will be seen, in order to analyze a subsystem, we need information from the previous subsystem and the next subsystem. In view of this, our algorithm is an iterative scheme, where the subsystems are analyzed one at a time continuously, until a convergence criterion is satisfied. The results for each subsystem are then combined to analyze the original network.

Below, we first describe how we analyze a single subsystem in isolation, using both Courtois' method and the modified Courtois' method. We then describe the overall iterative scheme.

3.2.1 Courtois' Method

In this section, we first present Courtois' method and then we describe how this method is modified to analyze a single subsystem.

Let P be a transition probability matrix. The state space associated with P is partitioned into groups, often referred to as *lumps*, such that any state within a group contains the same number of customers in a specific portion of the state descriptor. The transition probability matrix P, written out according to these groups consists of blocks of transition probabilities along the diagonal and of other blocks along the off-diagonals above and below the diagonal. P is said to be *nearly completely decomposable* (NCD) if the sum of the non-zero transition probabilities on each row that lie within the diagonal block is close to 1. That means, that the sum of the off-diagonal probabilities along a row is extremely small. This occurs when most of the transitions are between states of the same group, with very few transitions between states of different groups.

Let $\underline{p} = (p_1, p_2,\ldots,p_W)$ be the stationary probability vector associated with P, where p_i is the probability of being in state i and W is the total number of states associated with P. Then $\underline{p}$ can be obtained by solving the linear system of equations $\underline{p}\,P=\underline{p}$, $\underline{p}\,\underline{e}=1$, where $\underline{e}$ is a column vector of 1's.

Assuming that P is NCD, $\underline{p}$ can also be solved approximately following Courtois' method. In this case, the off-diagonal transition probabilities of each row are re-distributed along the row inside the diagonal block in an arbitrary fashion. As a result of this, P is reduced to a matrix consisting of blocks along the diagonal. Let D be the diagonal block matrix obtained from P by allocating the off-diagonal elements. We

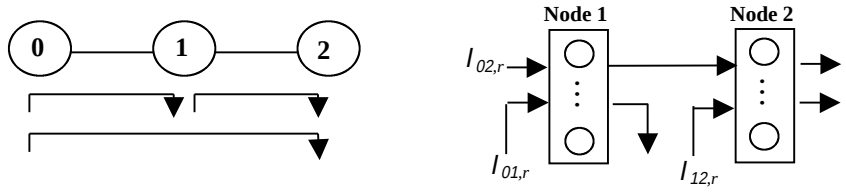
partition the state vector $\underline{p}$ conformally to the blocks of D, i.e., $\underline{p}=(\underline{p}_1, \underline{p}_2, \ldots \underline{p}_L)$ where L is the total number of groups. The state vector $\underline{p}$ is then obtained from $\underline{p}\ D=\underline{p}$ where $\underline{p}\underline{e}=1$, or since D is a diagonal matrix, each sub-vector $\underline{p}_i$ can be obtained by solving numerically the following system of equations: $\underline{p}_i\ D_i=\underline{p}_i$, $\underline{p}_i\ \underline{e}=1$.

Let $\underline{p}_i\ =(p_{i1}, p_{i2}, \ldots p_{il(i)})$, where $l(i)$ is the total number of states within the i^{th} group. Then, p_{ij} is the conditional probability of being in state j given that we are in block i. However, what we need to obtain is the unconditional probability of being in any state. These probabilities can be obtained by un-conditioning on p_{ij} ,$i=1,2,\ldots,L$, and $j=1,2,\ldots,l(i)$. This is done as follows. We first construct the aggregate matrix $Q=(q_{ij})$, where q_{ij} is the transition probability of going from block i to block j. This is obtained using the expression: $q_{ij}=\underline{p}_i P_{ij}\ \underline{e}$, where P_{ij} is the block of transition probabilities in the original matrix P that corresponds to the transitions from any state in group i to any state in group j. Solving the system $\underline{x}Q=\underline{x}$, $\underline{x}\underline{e}=1$ gives $\underline{x}=(x_1,x_2,\ldots x_L)$, where x_i is the unconditional probability of being in the i^{th} block. Finally, the unconditional probability of being in state j is given by $p_j=p_{ij}\ x_i$, $j=1,2,\ldots,W$.

3.2.2 Modified Courtois' Method

As mentioned above, the queueing network is decomposed into a number of subsystems, each representing two adjacent links. Figure 2(a) shows the first subsystem consisting of ADMs *0, 1* and *2*, and all the possible source-destination pairs, i.e., *(0,2), (0,1),* and *(1,2)*. Figure 2(b) shows the two-node queueing network model of the optical subsystem. Each

node consists of *s* servers, with each server representing a sub-wavelength unit. Node *1* represents the link between ADMs *0* and *1*, and node *2* represents the link between ADMs *1* and *2*. There are three different types of customers: *cross traffic*, *local traffic 1*, and *local traffic 2*. These customers are associated with the source-destination pairs *(0,2)*, *(0,1)*, and *(1,2)*, respectively. Customers of class *r* for source-destination pair *(i,j)* requiring d_r servers arrive in a Poisson fashion with rate $\lambda_{ij,r}$ and have an exponentially distributed holding time with rate $\mu_{ij,r}$.



a) Nodes 0,1, and 2 **b) The two-node queueing network**

Figure 2. The two-link subsystem

Let $\underline{n}=(\underline{n}_{02}, \underline{n}_{01}, \underline{n}_{12})$ be the state of the system, where $\underline{n}_{ij}=(n_{ij,1}, n_{ij,2}, \ldots n_{ij,R})$ and $n_{ij,r}$ is the number of class *r* customers on source-destination path *(i,j)*. For each link of the system, a state is valid if

$$0 \leq \sum_{\substack{all \\ (i,j),r}} n_{ij,r} d_r \leq s$$

where d_r is the demand, measured in the number of servers required by a class *r* customer.

We now proceed to describe the algorithm for the analysis of the two-node sub-system. For presentation purposes, we assume only two classes of customers with demands $d_1=1$, $d_2=2$, and $s=4$ servers per node. The state space associated with all feasible states $\underline{n}=(\underline{n}_{02}, \underline{n}_{01}, \underline{n}_{12})$ is enumerated by first fixing the value of $\underline{n}_{02}$ and then generating all the feasible values for $\underline{n}_{01}$ and $\underline{n}_{12}$. For example, we first assume that $\underline{n}_{02}=(0,0)$. This means that there is no cross traffic, and therefore the local traffic on either node can use all 4 servers. The following are the feasible states: *{(0,0,0,0,0,0), (0,0,0,1,0,0), (0,0,0,1,0,1), (0,0,2,1,1,1)}.* Subsequently, we fix the cross traffic to be such that only 1 server is used, i.e., $\underline{n}_{02}=(1,0)$. For this state, the following are feasible states: *{(1,0,0,0,0,0), (1,0,0,1,0,0), (1,0,0,1,0,1), (1,0,0,1,1,1)}.*

In general, the following states are possible for $\underline{n}_{02}$: *(0,0), (0,1), (0,2), (1,0), (1,1), (2,0), (2,1), (3,0), (4,0).* For each of these states, we generate a block of the feasible states for the local traffic on nodes *1* and *2*. We can now construct the rate matrix *R* by generating all possible transitions from each feasible state $\underline{n}$ to $\underline{n}'$, where $\underline{n}'$ represents a valid state that can be entered from state $\underline{n}$, due to a single arrival or departure. *R* is then transformed into the transition probability matrix *P* as follows. For each row, the negative sum of all row entries is placed in the diagonal element, and subsequently, all elements of the resulting matrix are divided by the absolute value of the largest diagonal element.

Assuming that the resulting matrix is NCD, we can follow the steps of Courtois' method outlined above to solve it numerically, and obtain the steady-state probability vector $\underline{p}$. However, we note that, for a fixed value of $\underline{n}_{02}$, the random variables $\underline{n}_{01}$ and $\underline{n}_{12}$ are independent of each other. That is, the number of local traffic *1* customers on node *1*

does not affect the number of local traffic *2* customers on node *2* and vice versa. Furthermore, we observe that, for fixed $\underline{n}_{02}=(n_{02,1}, n_{02,2})$, the first node is a multi-rate Erlang loss model with $s-(n_{02,1}d_1+ n_{02,2}d_2)$ servers. Likewise, the second node is also a multi-rate Erlang loss model with $s-(n_{02,1}d_1+ n_{02,2}d_2)$ servers. Consequently, given a fixed value for $\underline{n}_{02}=(n_{02,1}, n_{02,2})$, we have that $p(\underline{n}_{01},\underline{n}_{12})=p(\underline{n}_{01})^{x}p(\underline{n}_{12})$ where $p(\underline{n}_{ij})$ can be obtained using the following expression from the multi-rate Erlang loss model:

$$p(\underline{n}_{ij})=G^{-1}\prod_{r=1}^{R}\frac{A_{ij,r}^{n_{ij,r}}}{n_{ij,r}!} \tag{1}$$

where $A_{ij,r} = l_{ij,r}/m_{ij,r}$, is the offered traffic of a class r customer on source-destination path (i,j), and G is the normalizing constant defined as

$$G = \sum\prod_{r=1}^{R}\frac{A_{ij,r}^{n_{ij,r}}}{n_{ij,r}!} \tag{2}$$

Expression (1) gives the exact solution for the conditional probability $p_{il(i)}$ of a state in block i, $i= 1,2,\ldots,L$. Therefore, we can by-pass the first step of Courtois' algorithm that requires the solution of the system of equations $\underline{p}_i\ D_i=\underline{p}_i$, $\underline{p}_i\ \underline{e}=1$, $i= 1,2,\ldots,L$. Having solved for $p_{il(i)}$ we can proceed as above to construct the aggregate matrix Q and solve the system $\underline{x}Q=\underline{x}$, $\underline{x}\underline{e}=1$ numerically in order to obtain $\underline{x}=(x_1,x_2,\ldots x_L)$, where x_i is the unconditional probability of being in the i^{th} block. Finally, the unconditional probability of being in state j is given by $p_j=p_{ij}x_i$, where $j=1,2,\ldots,W$ represents all valid states, $\underline{n}$, for the network.

Courtois' method typically gives an approximate solution. However, in this case it gives the exact solution due to the fact that it calculates exactly the conditional

probabilities $p_{il(i)}$ using (1). This can be easily demonstrated by considering the balance equations of the two-node sub-system. Let p($\underline{n}$) be the probability of network state $\underline{n}$ of a subsystem. Then, the balance equation is as follows:

$$p(\underline{n})[\sum_{\substack{r=1\\(i,j)}}^{2} l_{ij,r} + \sum_{\substack{r=1\\(i,j)}}^{2} m_{ij,r}\ ] =$$

$$\begin{aligned}
&p(n_{02,1}\text{-}1, n_{02,2}, n_{01,1}, n_{01,2}, n_{12,1}, n_{12,2})\, l_{02,1}\, I(n_{02,1}\geq 1)\\
+\ &p(n_{02,1}, n_{02,2}\text{-}1, n_{01,1}, n_{01,2}, n_{12,1}, n_{12,2})\, l_{02,2}\, I(n_{02,2}\geq 1)\\
+\ &p(n_{02,1}, n_{02,2}, n_{01,1}\text{-}1, n_{01,2}, n_{12,1}, n_{12,2})\, l_{01,1}\, I(n_{01,1}\geq 1)\\
+\ &p(n_{02,1}, n_{02,2}, n_{01,1}, n_{01,2}\text{-}1, n_{12,1}, n_{12,2})\, l_{01,2}\, I(n_{01,2}\geq 1)\\
+\ &p(n_{02,1}, n_{02,2}, n_{01,1}, n_{01,2}, n_{12,1}\text{-}1, n_{12,2})\, l_{12,1}\, I(n_{12,1}\geq 1)\\
+\ &p(n_{02,1}, n_{02,2}, n_{01,1}, n_{01,2}, n_{12,1}, n_{12,2}\text{-}1)\, l_{12,2}\, I(n_{12,2}\geq 1)\\
+\ &p(n_{02,1}\text{+}1, n_{02,2}, n_{01,1}, n_{01,2}, n_{12,1}, n_{12,2})\, (n_{02,1}\text{+}1)\, m_{02,1}\, I(n_{02,1}< \lfloor s/d_1 \rfloor)\\
+\ &p(n_{02,1}, n_{02,2}\text{+}1, n_{01,1}, n_{01,2}, n_{12,1}, n_{12,2})\, (n_{02,2}\text{+}1)\, m_{02,2}\, I(n_{02,2}< \lfloor s/d_2 \rfloor)\\
+\ &p(n_{02,1}, n_{02,2}, n_{01,1}\text{+}1, n_{01,2}, n_{12,1}, n_{12,2})\, (n_{01,1}\text{+}1)\, m_{01,1}\, I(n_{01,1}< \lfloor s/d_1 \rfloor)\\
+\ &p(n_{02,1}, n_{02,2}, n_{01,1}, n_{01,2}\text{+}1, n_{12,1}, n_{12,2})\, (n_{01,2}\text{+}1)\, m_{01,2} I(n_{01,2}< \lfloor s/d_2 \rfloor)\\
+\ &p(n_{02,1}, n_{02,2}, n_{01,1}, n_{01,2}, n_{12,1}\text{+}1, n_{12,2})\, (n_{12,1}\text{+}1)\, m_{12,1} I(n_{12,1}< \lfloor s/d_1 \rfloor)\\
+\ &p(n_{02,1}, n_{02,2}, n_{01,1}, n_{01,2}, n_{12,1}, n_{12,2}\text{+}1)\, (n_{12,2}\text{+}1)\, m_{12,2} I(n_{12,2}< \lfloor s/d_2 \rfloor)
\end{aligned}$$

where the indicator functions $I(*)$ are defined as follows:

$$I(n_{ij,r}\geq 1) = \begin{cases} 1; & n_{ij,r} \geq 1 \\ 0 & else \end{cases}$$

$$I(n_{ij,r}< \lfloor s/d_r \rfloor) = \begin{cases} 1; & n_{ij,r} < \lfloor s/dr \rfloor \\ 0 & else \end{cases}$$

In addition, we have the normalizing equation,

$$\sum_{\underline{n}} p(\underline{n}) = 1.$$

We note that $p(\underline{n})$ can be expressed as follows: $p(\underline{n})=p(\underline{n}_{01}, \underline{n}_{12}|\, \underline{n}_{02})p(\underline{n}_{02})$, where $p(\underline{n}_{01}, \underline{n}_{12}|\, \underline{n}_{02})$ is given by (1). Let $E(\underline{n}_{01}, \underline{n}_{12})$ represent (1). That is, $p(\underline{n}_{01}, \underline{n}_{12}|\, \underline{n}_{02}) = E(\underline{n}_{01}, \underline{n}_{12})$. Substituting $E(\underline{n}_{01}, \underline{n}_{12})$ into the previous balance equation, we have

$$p(n_{02,1}, n_{02,2})\, E(\underline{n}_{01}, \underline{n}_{12})[\sum_{\substack{r=1\\(i,j)}}^{2} l_{ij,r} + \sum_{\substack{r=1\\(i,j)}}^{2} m_{ij,r}\ ]$$

$$\begin{aligned}
&= p(n_{02,1}\text{-}1, n_{02,2})\, E(\, n_{01,1}\, , n_{01,2}, n_{12,1}, n_{12,2})\, l_{02,1}\, I(n_{02,1}\geq 1)\\
&+ p(n_{02,1}, n_{02,2}\text{-}1)\, E(\, n_{01,1}\, , n_{01,2}, n_{12,1}, n_{12,2})\, l_{02,2}\, I(n_{02,2}\geq 1)\\
&+ p(n_{02,1}, n_{02,2})\, E(\, n_{01,1}\text{-}1\, , n_{01,2}, n_{12,1}, n_{12,2})\, l_{01,1}\, I(n_{01,1}\geq 1)\\
&+ p(n_{02,1}, n_{02,2})\, E(\, n_{01,1}\, , n_{01,2}\text{-}1, n_{12,1}, n_{12,2})\, l_{01,2}\, I(n_{01,2}\geq 1)\\
&+ p(n_{02,1}, n_{02,2})\, E(\, n_{01,1}\, , n_{01,2}, n_{12,1}\text{-}1, n_{12,2})\, l_{12,1}\, I(n_{12,1}\geq 1)\\
&+ p(n_{02,1}, n_{02,2})\, E(\, n_{01,1}\, , n_{01,2}, n_{12,1}, n_{12,2}\text{-}1)\, l_{12,2}\, I(n_{12,2}\geq 1)\\
&+ p(n_{02,1}\text{+}1, n_{02,2})\, E(\, n_{01,1}\, , n_{01,2}, n_{12,1}, n_{12,2})\, (\, n_{02,1}\text{+}1)\, m_{02,1}\, I(n_{02,1}< \lfloor s/d_1 \rfloor)\\
&+ p(n_{02,1}, n_{02,2}\text{+}1)\, E(n_{01,1}\, , n_{01,2}, n_{12,1}, n_{12,2})\, (\, n_{02,2}\text{+}1)\, m_{02,2}\, I(n_{02,2}< \lfloor s/d_2 \rfloor)\\
&+ p(n_{02,1}, n_{02,2})\, E(\, n_{01,1}\text{+}1\, , n_{01,2}, n_{12,1}, n_{12,2})\, (\, n_{01,1}\text{+}1)\, m_{01,1}\, I(n_{01,1}< \lfloor s/d_1 \rfloor)\\
&+ p(n_{02,1}, n_{02,2})\, E(\, n_{01,1}, n_{01,2}\text{+}1, n_{12,1}, n_{12,2})\, (\, n_{01,2}\text{+}1)\, m_{01,2} I(n_{01,2}< \lfloor s/d_2 \rfloor)\\
&+ p(n_{02,1}, n_{02,2})\, E(\, n_{01,1}, n_{01,2}, n_{12,1}\text{+}1, n_{12,2})\, (\, n_{12,1}\text{+}1)\, m_{12,1} I(n_{12,1}< \lfloor s/d_1 \rfloor)\\
&+ p(n_{02,1}, n_{02,2})\, E(\, n_{01,1}, n_{01,2}, n_{12,1}, n_{12,2}\text{+}1)\, (\, n_{12,2}\text{+}1)\, m_{12,2} I(n_{12,2}< \lfloor s/d_2 \rfloor)
\end{aligned}$$

The values of $p(\underline{n}_{02})$ can be determined by solving the above system of linear equations. We note that this system of linear equation is in fact the system $\underline{x}Q=\underline{x}$, $\underline{x}e=1$, obtained through the aggregation step in Courtois' method, which means that the solution for $\underline{x}$ is exact.

3.2.3 The Decomposition Algorithm

We describe the decomposition algorithm using a seven-node optical network modeled by the six-node queueing network shown in Figure 3. For presentation purposes, we do

not show the customer flows for all the possible source-destination paths. The queueing network is decomposed into 3 subsystems as shown in Figure 3, where each subsystem contains 2 nodes that represent two links in the original network, as in Figure 2(a). Subsystem *1* represents the links connecting ADMs *0* and *1*,and *1* and *2* in the original network, subsystem *2* represents the links connecting ADMs *2* and *3*, and *3* and *4*, and subsystem *3* represents the links connecting ADMs *4* and *5*, and *5* and *6*.

For simplicity, the paths of each subsystem will be defined as *(0,2), (0,1), (1,2),* representing the two-node path, one-node path on the first node, and one-node path on the second node, respectively. Let $\underline{\lambda}_{02}^{(x)}$, $\underline{\lambda}_{01}^{(x)}$, and $\underline{\lambda}_{12}^{(x)}$ denote the effective arrival rates of customers on each path of each subsystem, where $\underline{\lambda}_{ij}^{(x)}=(\lambda_{ij,1}^{(x)}, \lambda_{ij,2}^{(x)}, \ldots \lambda_{ij,R}^{(x)})$ and $\lambda_{ij,r}^{(x)}$ is the effective arrival rate of a class r customer on path *(i, j)* of subsystem *x* as seen by the subsystem. In addition, let $\underline{b}_{02}^{(x)}$, $\underline{b}_{01}^{(x)}$, and $\underline{b}_{12}^{(x)}$ denote the blocking probability on each path of subsystem *x*, where $\underline{b}_{ij}^{(x)}=(b_{ij,1}^{(x)}, b_{ij,2}^{(x)}, \ldots b_{ij,R}^{(x)})$ and $b_{ijr}^{(x)}$ is the blocking probability of a class r customer on path *(i,j)* of subsystem *x*.

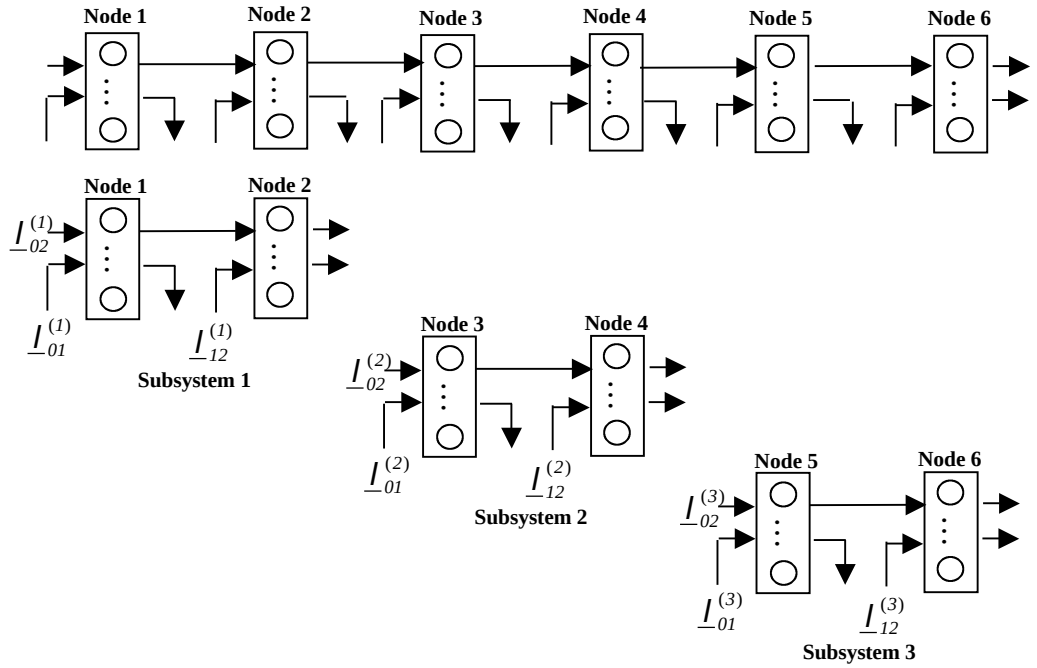


Figure 3. The queueing network and the individual sub-systems

The decomposition algorithm iteratively analyzes each subsystem. An iteration consists of analyzing subsystems *1*, *2*, and *3* once, in the respective order. At the beginning of the first iteration, we assume that all blocking probabilities are zero. In order to analyze each subsystem we need to know the effective arrival rates for each class *r* customer. For subsystem *1*, these arrival rates are determined as follows:

$$l_{02,r}^{(1)} = l_{02,r} + l_{03,r}(1-b_{01,r}^{(2)}) + l_{04,r}(1-b_{02,r}^{(2)}) + l_{05,r}(1-b_{02,r}^{(2)})(1-b_{01,r}^{(3)}) \qquad (3)$$

$$+ l_{06,r}(1-b_{02,r}^{(2)})(1-b_{02,r}^{(3)}),\ r=1,2,\ldots,R$$

$$\lambda^{(1)}_{01,r} = \lambda_{01,r}, \; r=1,2,\ldots,R \qquad (4)$$

$$\lambda^{(1)}_{12,r} = \lambda_{12,r} + \lambda_{13,r}(1-b^{(2)}_{01,r}) + \lambda_{14,r}(1-b^{(2)}_{02,r}) + \lambda_{15,r}(1-b^{(2)}_{02,r})(1-b^{(3)}_{01,r}) \qquad (5)$$

$$+ \lambda_{16,r}(1-b^{(2)}_{02,r})(1-b^{(3)}_{02,r}), \; r=1,2,\ldots,R$$

The effective arrival rates for subsystem *1* account for all traffic that originates in the system and terminates either inside the system or in subsystem *2* or *3*, as seen by subsystem *1*. Expression (3) determines the effective arrival rate of all traffic traversing both nodes of subsystem *1*. This corresponds to all customers that originate at ADM *0* of the original seven-node optical network and terminate at ADM *i*, where *i* . For example, the term $\lambda_{03,r}(1-b^{(2)}_{01,r})$ is the effective arrival rate of traffic on path *(0,3)* of the original network. Expression (5) follows the same idea, where the effective arrival rate is calculated for traffic traversing the second node of subsystem *1*. This corresponds to traffic originating at ADM *1* of the original network, and terminating at ADM *2* or higher. Customers traversing only the first node of subsystem *1* do not leave the subsystem. Therefore, the effective arrival rate in (4) is equal to the original arrival rate of traffic originating at ADM *0* of the original network and terminating at ADM *1*.

Using (4), (5), and (6) the two-node subsystem is analyzed as described in section 3.2.2. From the results obtained, we calculate the blocking probability, $b^{(1)}_{ij,r}$, of each class *r* customer in subsystem *1* by summing the probabilities of every state for which the number of free servers available for a class *r* customer on path *(i,j)* is less than its respective demand, d_r .

Next, we analyze subsystem *2*. For this subsystem, the effective arrival rates are calculated as follows:

$$\lambda^{(2)}_{02,r} = \lambda_{24,r} + \lambda_{04,r}(1-b^{(1)}_{02,r}) + \lambda_{14,r}(1-b^{(1)}_{12,r}) + \lambda_{05,r}(1-b^{(1)}_{02,r})(1-b^{(3)}_{01,r}) \quad (6)$$
$$+ \lambda_{15,r}(1-b^{(1)}_{12,r})(1-b^{(3)}_{12,r}) + \lambda_{25,r}(1-b^{(3)}_{12,r}) + \lambda_{06,r}(1-b^{(1)}_{02,r})(1-b^{(3)}_{02,r})$$
$$+ \lambda_{16,r}(1-b^{(1)}_{12,r})(1-b^{(3)}_{02,r}) + \lambda_{26,r}(1-b^{(3)}_{02,r});\ r=1,2,\ldots,R$$

$$\lambda^{(2)}_{01,r} = \lambda_{23,r} + \lambda_{03,r}(1-b^{(1)}_{02,r}) + \lambda_{13,r}(1-b^{(1)}_{12,r});\ r=1,2,\ldots,R \quad (7)$$

$$\lambda^{(2)}_{12,r} = \lambda_{34,r} + \lambda_{35,r}(1-b^{(3)}_{01,r}) + \lambda_{36,r}(1-b^{(3)}_{02,r});\ r=1,2,\ldots,R \quad (8)$$

The most recent $\underline{b}^{(1)}_{ij}$ values are used in (6), (7), and (8). The paths of subsystem *2* include traffic from subsystems *1* and *3*. As a result, (6) accounts for all traffic originating or terminating in, and passing through ADMs *2*, *3*, or *4* of the original seven-node optical network. Expression (7) accounts for traffic on the first node of subsystem *2*. This corresponds to traffic originating at ADM *2* or lower of the original network and terminating at ADM *3*. Expression (8) accounts for traffic on the second node of subsystem *2*. This corresponds to traffic originating at ADM *3* of the original network, and terminating at ADM *4* or higher. Subsystem *2* is analyzed using the procedure described above in section 3.2.2, and the conditional probabilities $\underline{b}^{(2)}_{02}$, $\underline{b}^{(2)}_{01}$, and $\underline{b}^{(2)}_{12}$ are subsequently calculated in the same manner as in subsystem *1*.

Next we analyze subsystem *3*, for which the effective arrival rates are calculated as follows:

$$\lambda^{(3)}_{02,r} = \lambda_{46,r} + \lambda_{06,r}(1-b^{(1)}_{02,r})(1-b^{(2)}_{02,r}) + \lambda_{16,r}(1-b^{(1)}_{12,r})(1-b^{(2)}_{02,r}) \quad (9)$$
$$+ \lambda_{26,r}(1-b^{(2)}_{02,r}) + \lambda_{36,r}(1-b^{(2)}_{12,r});\ r=1,2,\ldots,R$$

$$I^{(3)}_{01,r} = I_{45,r} + I_{05,r}(1-b^{(1)}_{02,r})(1-b^{(2)}_{02,r}) + I_{15,r}(1-b^{(1)}_{12,r})(1-b^{(2)}_{02,r}) + I_{25,r}(1-b^{(2)}_{02,r}) \quad (10)$$

$$+ I_{35,r}(1-b^{(2)}_{12,r}); r=1,2,\ldots,R$$

$$I^{(3)}_{12,r} = I_{56,r}; r=1,2,\ldots,R \quad (11)$$

The most recent $\underline{b}^{(1)}_{ij}$ and $\underline{b}^{(2)}_{ij}$ values are used in (9), (10), and (11). Expression (9) accounts for traffic originating at ADM *4* or lower in the original seven-node optical network and terminating at ADM *6*. Similarly, (10) accounts for traffic originating at ADM *4* or lower in the original network and terminating at ADM *5*. Expression (11) does not account for any traffic from lower subsystems because all traffic on the second node of the subsystem, i.e. path *(5,6)* of the original network, originates and terminates inside the subsystem. Subsystem *3* is analyzed using the procedure described in section 3.2.2, and subsequently the unconditional probabilities $\underline{b}^{(3)}_{02}$, $\underline{b}^{(3)}_{01}$, and $\underline{b}^{(3)}_{12}$ are determined in the same manner as in the previous subsystems.

At this point, we have completed the first iteration where each subsystem was analyzed once and new blocking probabilities were calculated. The algorithm executes successive iterations until the blocking probabilities have all converged. That is, the absolute difference of each blocking probability between two successive iterations is less than e, where e is set to 10^{-4}.

After the algorithm has converged, the blocking probabilities from each subsystem are used to calculate the blocking probabilities of the original network as follows. Let $b_{ij,r}$ denote the blocking probability of a class r call on path *(i,j)* of the

original network. For paths in the original network that map directly to a path of a subsystem, the blocking probabilities are equal to the blocking probabilities of the corresponding paths in the respective subsystem. For example, the blocking probability of a class *r* call on path *(0,1)* of the original network is $b_{01,r} = b_{01,r}^{(1)}$. Likewise, the blocking probability of calls on path *(4,6)* of the original network is $b_{46,r} = b_{02,r}^{(3)}$, and so forth. For the remaining source-destination paths the blocking probability is calculated using the expression *1-p{not blocked on each subsystem path traversed}*. For example, the blocking probability of a class *r* call on the four-link path *(1,5)* is $1-[(1-b_{12,r}^{(1)})(1-b_{02,r}^{(2)})(1-b_{01,r}^{(3)})]$ which accounts for the respective paths on each subsystem that the original path traverses.

3.3 Numerical Results

In this section, we discuss the accuracy of the decomposition algorithm by comparing results obtained to simulation data. In the experiments reported below we considered a seven-ADM optical network modeled by the six-node queueing network shown in figure 3. Each link between two adjacent ADMs is composed of a single wavelength that consists of 12 sub-wavelength units. In view of this, the number of servers in each node of the queueing network is 12. All possible source-destination paths are considered, and for each path we assume four classes of traffic, i.e. R=4, with demands of d_1=2, d_2=4, d_3=6, and d_4=8 servers.

The accuracy of the decomposition algorithm was evaluated under two cases, referred to as case 1 and case 2, where the cross traffic, i.e. the traffic traversing more than one link, represented 25% and 50% respectively of the total offered traffic to the network.

In case 1, $l_{01,1}$ is assigned the highest arrival rate, and the arrival rate of each successive local call type is decreased by a small amount from the previous as follows. Referring to Figure 1, all calls that traverse one link are assigned first, starting with the link between ADMs *0* and *1*, and then proceeding to the right. Within the calls of a given path, the assignment of the arrival rates is from class *1* to class *4*. The following values were used. $l_{01,1}$ was set to 1 and each successive local call type was decreased by 0.01. All two, three, four, five, and six link calls were assigned arrival rates of 0.1, 0.1, 0.1, 0.1, and 0.1, respectively. The service rate of all calls was 500.

Figures 4 through 8 show the results for all one-hop, two-hop, three-hop, four-hop, five-hop, and six-hop paths, respectively. Each graph plots the approximate and simulated blocking probability of a class *r* call, *r=1,2,3,4*, as a function of the call path, identified by the pair (source, destination). The approximate (simulated) results are indicated by app (sim) followed by the class number. The simulation results are plotted with a 95% confidence interval, indicated by the '+' values in each graph. The blocking probabilities on the vertical axis are given in the logarithmic scale.

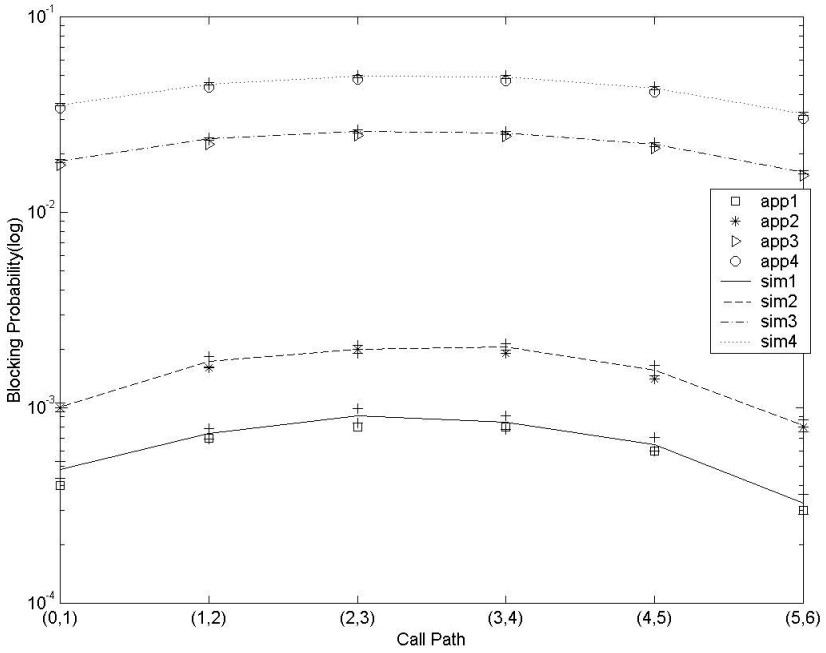


Figure 4. One-hop paths for class r, r=1,2,3,4

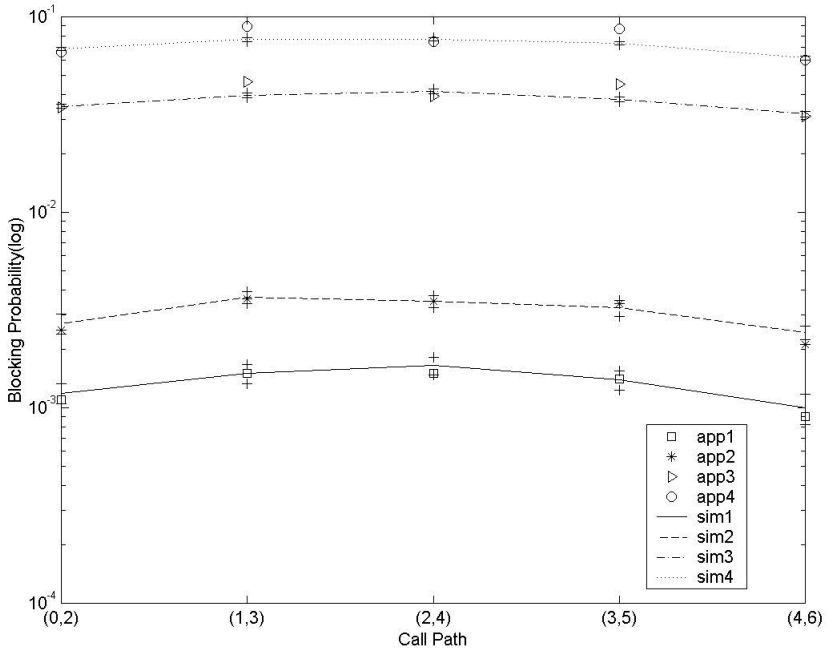


Figure 5. Two-hop paths for class r, r=1,2,3,4

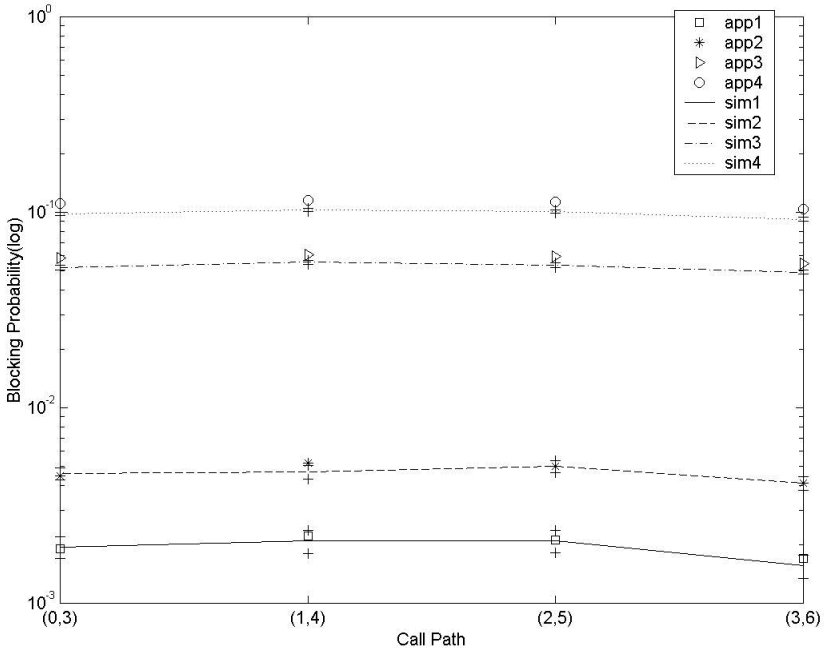


Figure 6. Three-hop paths for class r, r=1,2,3,4

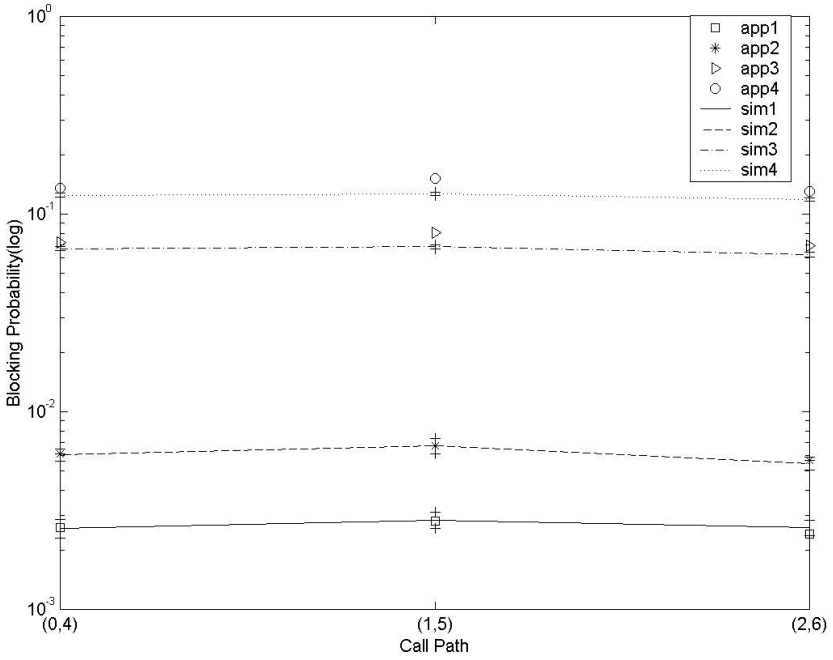


Figure 7. Four-hop paths for class r, r=1,2,3,4

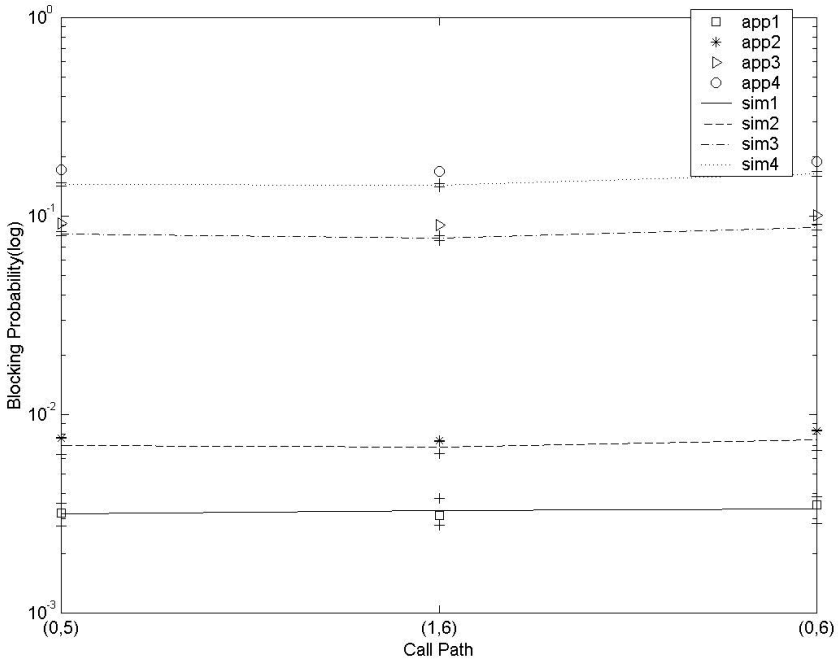


Figure 8. Five and six-hop paths for class r, r=1,2,3,4

We observe in all figures that the blocking probability increases with the call demand. This is expected, and it is due to the fact that calls requiring a greater number of servers are more likely to be blocked due to a lack of available resources, thereby lowering the blocking of calls requiring fewer servers.

We also observe that the blocking probability increases as the number of hops per path increases. We attribute this to the fact that calls with longer paths are less likely to find sufficient resources on every link traversed. This results in calls with shorter paths, such as one, two, and three-hops, experiencing lower blocking. In addition, for each figure, the blocking probability increases, for the most part, on paths requiring links *3* and *4* of the original network. These two links support up to half of the total network traffic

each. Ultimately, calls traversing these links experience more blocking than other calls with the same number of hops that traverse other links.

In case 2, the cross traffic represents 50% of the total offered network traffic. Referring to Figure 1, all calls that traverse one link are assigned first, as with case 1, starting with the link between ADMs *0* and *1*, and then proceeding to the right. For case 2, $l_{01,1}$ was set to 1 and each successive local call type was assigned a rate which decreased by 0.01 from the previous. All two, three, four, five, and six link calls were assigned arrival rates of 0.5, 0.4, 0.3, 0.2, and 0.1, respectively. The service rate of all calls was 500. The average relative errors are given in Tables 1 and 2 under the app-2 header. In addition, Tables 1 and 2 give the average relative errors of the single-node decomposition algorithm based on the assumption of link independence, under the app-1 header. As can be seen, our algorithm provides more accurate results than the single-node decomposition algorithm.

	Class 1		Class 2		Class 3		Class 4	
	app-2	**app-1**	**app-2**	**app-1**	**app-2**	**app-1**	**app-2**	**app-1**
Case 1	0.0755	0.1439	0.0546	0.1451	0.1170	0.2376	0.1176	0.2378
Case 2	0.0772	0.1515	0.0937	0.1914	0.2013	0.3918	0.2001	0.3985

Table 1. Per-class relative errors for the two-node (app-2) and single-node (app-1) decomposition algorithm

	1 Hop		2 Hops		3 Hops	
	app-2	**app-1**	**app-2**	**app-1**	**app-2**	**app-1**
Case 1	0.0438	0.0560	0.0905	0.1915	0.1150	0.2299
Case 2	0.0425	0.0442	0.1235	0.2207	0.1891	0.3649
	4 Hops		**5 Hops**		**6 Hops**	
	app-2	**app-1**	**app-2**	**app-1**	**app-2**	**app-1**
Case 1	0.1106	0.2900	0.1429	0.3107	0.1269	0.3088
Case 2	0.2070	0.4809	0.2493	0.5199	0.2436	0.5531

Table 2. Per-hop relative errors for the two-node (app-2) and single-node (app-1) decomposition algorithm

Overall, we observe that the proposed two-node decomposition algorithm has good accuracy. However, its accuracy tends to decrease as the path length and call demand increase. In addition, we observe that the decomposition algorithm is limited to a small number of servers per node. This is due to the fact that the analysis of each subsystem requires knowledge of the entire state space of the subsystem. As the number of servers per node increases, so does the size of the state space. This leads to much longer computation times.

Chapter 4

Analysis of Large Traffic-Groomed Optical Networks

In Chapter 3, we proposed a decomposition algorithm whereby the traffic-groomed optical network is decomposed into two-node subsystems. Each subsystem is analyzed exactly using a modification of Courtois' method. Although the proposed method was shown to be an improvement over the single-node decomposition algorithm, the analysis was limited to networks of small size.

In this chapter, we extend the algorithm presented in Chapter 3 to the case where the link between two adjacent nodes consists of W wavelengths, each with s sub-rate streams. Bifurcation of traffic is permitted, which means that, on each link, a call may use sub-rate traffic streams that belong to different wavelengths. This allows us a view to analyzing a

large number of servers per link. We present an approximation method for the analysis of the two-node subsystem

4.1 The Queueing Network Model

We model the optical network shown in Figure 1, which is a seven-node optical network. Adjacent nodes are linked by a single fiber consisting of W wavelengths. Each wavelength carries s sub-rate streams. We assume that traffic bifurcation is permitted. This means that, on each link, a call may use sub-rate traffic-streams belonging to different wavelengths. Because of this assumption and also due to the fact that each node is an opaque add/drop multiplexer, we consider the Ws sub-rate streams as a single pool from which sub-rate streams are allocated to each call

As in 3.1, calls may arrive to any node i, $i=0,1,\ldots 6$, and be destined for any node j, $j=1,2,\ldots 6$, where $j>i$. All calls, irrespective of their source-destination path, are grouped into R classes. Each class r, $r=1, 2, \ldots R$, has an associated demand d_r measured in sub-wavelength units. An arriving class r call is accepted if there are at least d_r sub-wavelength units available on each link between the source and destination node. Otherwise, the call is blocked, i.e. lost. An accepted class r call simultaneously occupies d_r sub-wavelength units on each link along the source-destination path, and simultaneously releases them upon call completion.

Each link of the optical network is represented in the queueing network model by a multi-rate loss queue with $m=Ws$ servers, where each server represents a sub-wavelength unit. We refer to the six-node queueing network in Figure 3 representing the optical network

in Figure 1. Each customer in the queueing network represent a call. Customers arrive at any node $i=1,2,\ldots 6$ with Poisson rate $\lambda_{ij,r}$, and terminate at any node $j=1,2,...,6$, where $j \neq i$, with rate $\mu_{ij,r}$. A class r arriving customer is accepted into the queueing network if the required d_r servers are available on each node along the source-destination path of the customer. Otherwise, the customer is blocked, i.e., lost.

4.2 The Decomposition Algorithm

In this section, we present the decomposition algorithm developed for the analysis of the above described queueing network. This is the same decomposition algorithm as in Chapter 3. The algorithm decomposes the queueing network into subsystems, each consisting of two adjacent links. Each subsystem is analyzed numerically using the approximation algorithm. As before, in order to analyze a subsystem, we need information from the previous subsystem and the next subsystem. In view of this, the algorithm is an iterative scheme, where the subsystems are analyzed one at a time continuously, until a convergence criterion is satisfied. The results for each subsystem are then combined to analyze the original network.

We first describe the recursion developed by Nilsson et. al [19] and the modified recursion, used for the analysis of a single subsystem. Following this, we describe the decomposition algorithm.

4.2.1 Nilsson et. al Recursion

Consider a single multi-rate loss queue with R sources, each with demand d_r, $r=1,2,..R$. Let A_r be the offered traffic of a class r customer, that is, $A_r = l_r/m_r$. Let m and k denote the maximum number of servers in the node and the number of free servers in the node respectively. Define $b(m, k)$ as the probability that there are m-k busy servers in the loss queue. Nilsson et al [19] showed that:

$$b(m,k) = b(m-1,k-1)/[1+(1/m)\sum_r A_r d_r b(m-1, d_r-1)] \quad (12)$$

with

$$b(m,0) = (1/m)\sum_r A_r d_r b(m-1, d_r-1)\,[/1+(1/m)\sum_r A_r d_r b(m-1, d_r-1)] \quad (13)$$

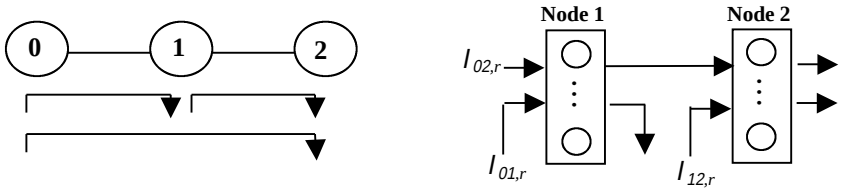
Using these probabilities, the blocking probability of a class r customer, B_r, can be computed as follows:

$$B_r = \sum_{j=0}^{dr-1} b(m, j) \quad (14)$$

4.2.2 Modified Nilsson et. al Recursion

As before, the queueing network is decomposed into a number of subsystems, each representing two adjacent links as in Figure 2, which is shown again below. Figure 2(a) shows the first subsystem, consisting of ADMs *0*, *1* and *2*, and all the possible source-destination pairs. Figure 2(b) shows the two-node queueing network model of the optical subsystem. Each node consists of m servers, with each server representing a sub-wavelength

unit. Node *1* represents the link between ADMs *0* and *1*, and node *2* represents the link between ADMs *1* and *2*. There are three different types of customers: *cross traffic*, *local traffic 1*, and *local traffic 2*, associated with source-destination pairs *(0,2)*, *(0,1)*, and *(1,2)*, respectively. Customers of class r destined for source-destination pair *(i,j)* requiring d_r servers arrive in a Poisson fashion with rate $l_{ij,r}$ and have an exponentially distributed holding time with rate $m_{ij,r}$.



a) Nodes 0,1, and 2 **b) The two-node queueing network**

Figure 2. The two-link subsystem

The subsystem was previously analyzed using a modified version of Courtois' method. It was determined that, while the accuracy of the modified Courtois' method was good, the analysis of each subsystem required knowledge of the entire state-space and, as a result, was limited to networks with a small number of servers per node. In order to analyze this subsystem assuming a large number of wavelengths and servers, we use the recursive equations developed by Nilsson et. al [19] which allow for a quick and stable analysis of a multi-rate loss queue with a large number of servers.

Consider the subsystem shown in Figure 2(b). Each source-destination path in the subsystem is analyzed separately. Beginning with the local traffic *1* associated with path

(0,1), the local traffic is iteratively analyzed as follows. We assume *2R* classes of traffic arrive to node *1*, corresponding to the *R* classes associated with path *(0,2)*, and the *R* classes associated with path *(0,1)*. Node *1* is offered traffic from path *(0,2)* with rate $A_{02,r}(1-b_{2,r})$, where $A_{02,r}= \lambda_{02,r}/\mu_{02,r}$ and $b_{2,r}$ is the blocking probability of the class *r* traffic on node *2* of path *(0,2)*. Traffic from path *(0,1)* is offered with rate $A_{01,r}= \lambda_{01,r}/\mu_{01,r}$. Node *1* is analyzed using (12) and (13). We note that the analysis of local traffic *1* accounts for all traffic in the subsystem that uses the link. As a result, (12) and (13) include the offered traffic and demand of each class *r* call on path (*0,2*) as well as path (*0,1*). The blocking probability of local *1* traffic, $b_{01,r}$, is determined using the following expression:

$$b_{01,r} = \sum_{k=0}^{b_r-1} b_{01}(m,k) \qquad (15)$$

which is the sum of the probabilities where the number of free servers on path (*0,1*), *k*, is less than the demand of an arriving class *r* call, b_r.

The local traffic *2* is analyzed in the same manner as the local traffic *1*. Traffic from path (*1,2*) as well as path (*0,2*) must be included in the analysis. As a result, node 2 is offered traffic from path (0,2) with rate $A_{02,r}(1-b_{1,r})$, where $b_{1,r}$ is the blocking probability of the class *r* traffic on node *1*, and traffic from path (*1,2*) with rate $A_{02,r}= \lambda_{02,r}/\mu_{02,r}$. Node *2* is then analyzed using (12) and (13), and the blocking probabilities of calls on path (*1,2*) are calculated using (15). At this point, the blocking probabilities for the local traffic on nodes 1

and 2 are tested for convergence. If the results have converged, the blocking experienced by local traffic on each node is stored.

For the cross traffic, we assume that the traffic arrives to a single node, representing path (*0,2*). In addition, we assume an aggregate traffic flow exists which represents the local traffic on both links *1* and *2*. This aggregate traffic flow, which we refer to as $A_{agg,r}$, is calculated as follows:

$$A_{agg,r}=A_{01,r}* [\, l_{01,r}/\sum_{r}(l_{01,r}+l_{02,r})\,]+(A_{12,r})*[\, l_{12,r}/\sum_{r}(l_{12,r}+l_{02,r})\,] \qquad (16)$$

It is assumed that *2R* classes of traffic are offered to the node, *R* classes from path (*0,2*) and *R* classes from the aggregate flow. The cross traffic is then analyzed using (12) and (13). The blocking probability, $b_{02,r}$, is obtained as follows:

$$b_{02,r}=[\sum_{c=0}^{m}\sum_{l_1=0}^{c}\sum_{l_2=0}^{\min(d_r-1,c)} b_{02}(m,c)\,b_{01}(m,l_1)\,b_{12}(m,l_2)\,]+[\sum_{dr=0}^{m}\sum_{l1=0}^{dr-1}\sum_{l2=dr}^{c} b_{02}(m,c)\,b_{01}(m,l_1)\,b_{12}(m,l_2)\,] \quad (17)$$

We consider the number of busy servers due to each call type, which would result in a blocked call of class *r* on the cross path (*0,2*). As a result, (17) sums the probability of having *c*, l_1, and l_2 free servers available for the cross path, local path *1*, and local path *2*, respectively.

4.2.3 The Decomposition Algorithm

The decomposition algorithm is similar to the one described in 3.2.3. We define $\underline{l}_{02}^{(x)}$, $\underline{l}_{01}^{(x)}$, and $\underline{l}_{12}^{(x)}$ as the effective arrival rates of customers on each path of each subsystem, where $\underline{l}_{ij}^{(x)}=(l_{ij,1}^{(x)}, l_{ij,2}^{(x)}, \ldots l_{ij,R}^{(x)})$ and $l_{ij,r}^{(x)}$ is the effective arrival rate of a class *r* customer on path *(i, j)* of subsystem *x* as seen by the subsystem. In addition, $\underline{b}_{02}^{(x)}$, $\underline{b}_{01}^{(x)}$, and $\underline{b}_{12}^{(x)}$ denote the blocking probability on each path of subsystem *x*, where $\underline{b}_{ij}^{(x)}=(b_{ij,1}^{(x)}, b_{ij,2}^{(x)}, \ldots b_{ij,R}^{(x)})$ and $b_{ij,r}^{(x)}$ is the blocking probability of a class *r* customer on path *(i,j)* of subsystem *x*.

Beginning with subsystem *1*, the effective arrival rates are determined using (3), (4), and (5). The subsystem is then analyzed using the approximation previously described. The blocking probabilities $b_{ij,r}^{(1)}$ are calculated and the algorithm shifts to subsystem *2*.

At subsystem *2*, the effective arrival rates are determined using (6), (7), and (8), and the most recently calculated blocking probabilities from subsystem *1*. The subsystem is analyzed using the approximation method, and new results for $b_{ij,r}^{(2)}$ are calculated. The algorithm then shifts to subsystem *3*.

At subsystem *3*, the effective arrival rates are determined using (9), (10), and (11), and the most recent results for subsystems *1* and *2*. Subsystem *3* is analyzed using the approximation method, and new results for $b_{ij,r}^{(3)}$ are determined.

At this point, a single iteration of the entire decomposition algorithm has completed, whereby the algorithm tests the blocking probabilities from each subsystem for convergence. If the results from each subsystem have not converged, the algorithm repeats another iteration, beginning with subsystem *1*, using the most recently calculated results. If the results have converged within a specific value, e, which is set to 10^{-4}, then the algorithm stops.

The blocking of paths in the original network is determined as follows. For paths equivalent to a path in a specific subsystem, the blocking probabilities are equal. For paths traversing more than one subsystem, the blocking probability is calculated as *1-{probability the call is not blocked on each subsystem path traversed}*.

4.3 Numerical Results

In this section, we discuss the accuracy of the decomposition algorithm, using the approximation method from 4.2, by comparing the results obtained to simulation data as well as the results of the single-node decomposition method assuming link independence. For each case below, we considered the seven-ADM optical network of Figure 1 modeled by the six-node queueing network shown in Figure 2. Each link between adjacent ADMs is composed of W wavelengths, with each wavelength containing s servers. Bifurcation of traffic is permitted. All possible source-destination paths are considered and, for each path, we assume four classes of traffic, i.e. R=4, with demands of d_1=2, d_2=4, d_3=6, and d_4=8 servers.

We note that, in our model, the accuracy of the approximation method is usually established by comparing the results with simulation results. However, due to the very low blocking probabilities, standard Monte-Carlo simulation techniques cannot be applied without introducing a very large error relative to the blocking rate itself. Therefore, applying rare-event simulation techniques become essential for validating our results. We use the Adaptive Importance Sampling using Standard Clock (A-ISSC) method, developed by Hsu and Devetsikiotis [25] to measure the accuracy of the approximations.

The accuracy of the decomposition algorithm was evaluated under the four cases: *Ws*=128, 256, 512, and 1024 servers. For each case, $\lambda_{01,1}$ was assigned the highest arrival rate, and the arrival rate of each successive local call type was decreased by a small amount from the previous as follows. Referring to Figure 1, all calls that traverse one link were assigned first, starting with the link between ADMs *0* and *1*, and then proceeding to the right. Within calls of a given path, the assignment of arrival rates was from class *1* to class *4*. Table 3 shows the arrival rates for each case.

Case	Ws	$\lambda_{01,1}$	One-hop	Two-hop	Three-hop	Four-hop	Five-hop	Six-hop
1	128	500	-5	250	200	150	100	50
2	256	1000	-10	500	400	300	200	100
3	512	2500	-25	1250	1000	750	500	250
4	1024	5000	-50	2500	2000	1500	1000	500

Table 3. Per-case arrival rates of all network call types

We note that the arrival rate of each successive one-hop call, after $\lambda_{01,1}$ is equal to the arrival rate of the previous one-hop call minus the value indicated in the one-hop column of Table 3. For each case, the service rate of all calls was set to 500.

Figure 9 shows the results, for 128 servers, for all one to six-hop paths. A complete set of results for all cases is found in the Appendix. Each graph plots the approximate results, simulation results (sim), and the link independence results (ind) of a class *r* call, *r*=1,2,3,4, as a function of the call path, identified by the pair (source, destination). The blocking probabilities are given in the logarithmic scale.

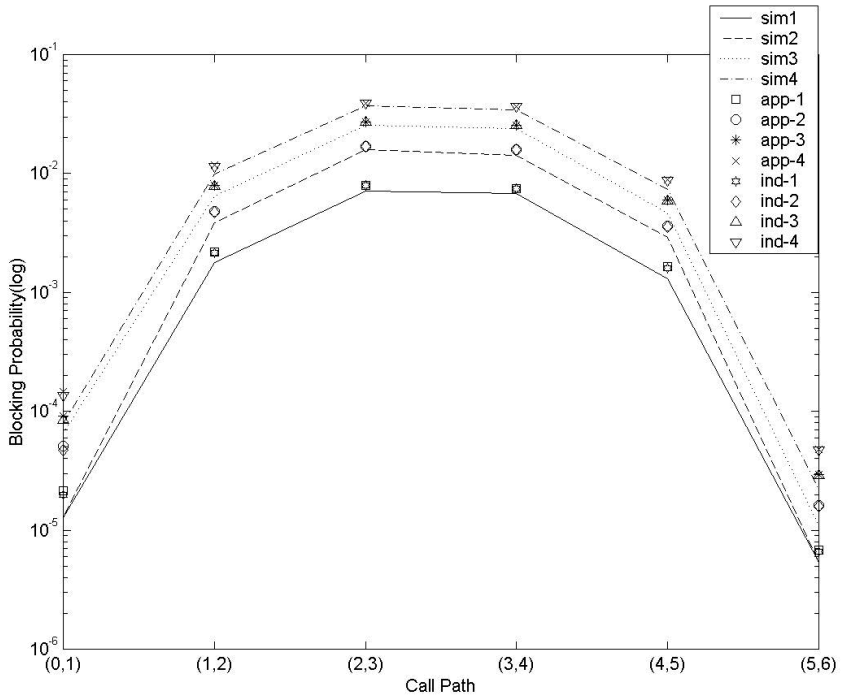


Figure 9a. One-hop path results for class *r*, *r*=1,2,3,4 and *Ws*=128 servers.

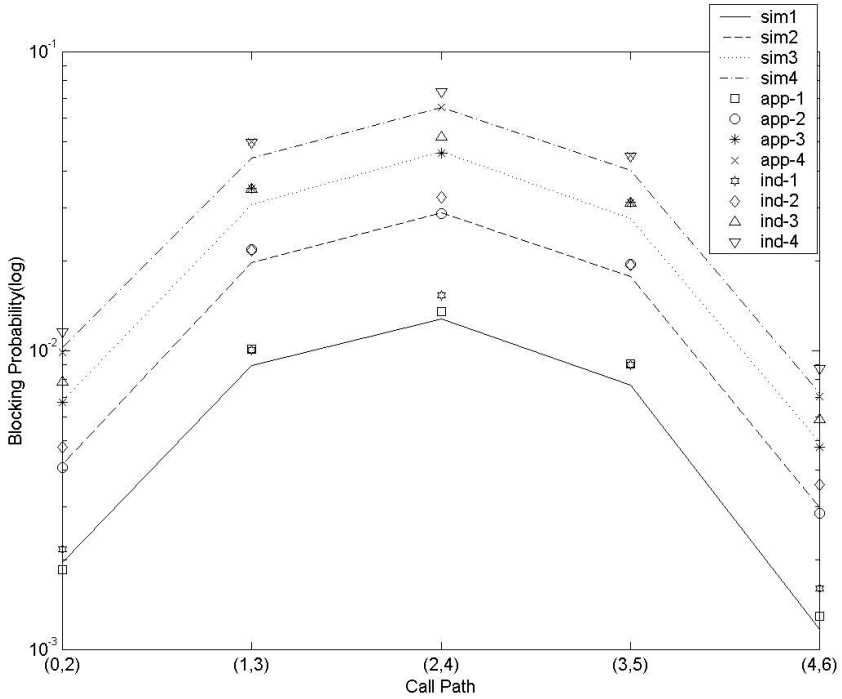


Figure 9b. Two-hop path results for class *r*, *r*=1,2,3,4 and *Ws*=128 servers.

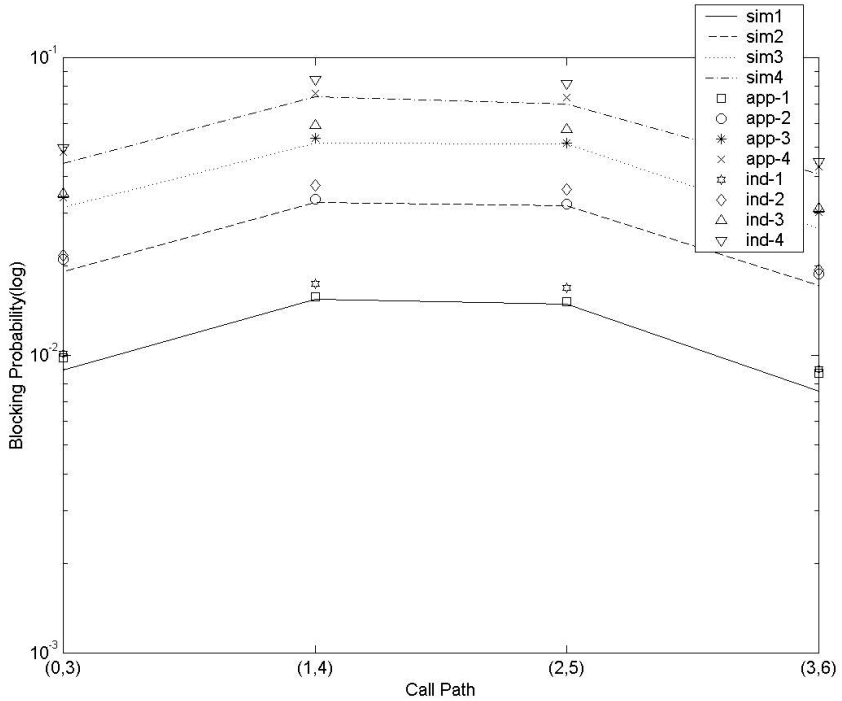


Figure 9c. Three-hop path results for class *r*, *r*=1,2,3,4 and *Ws*=128 servers.

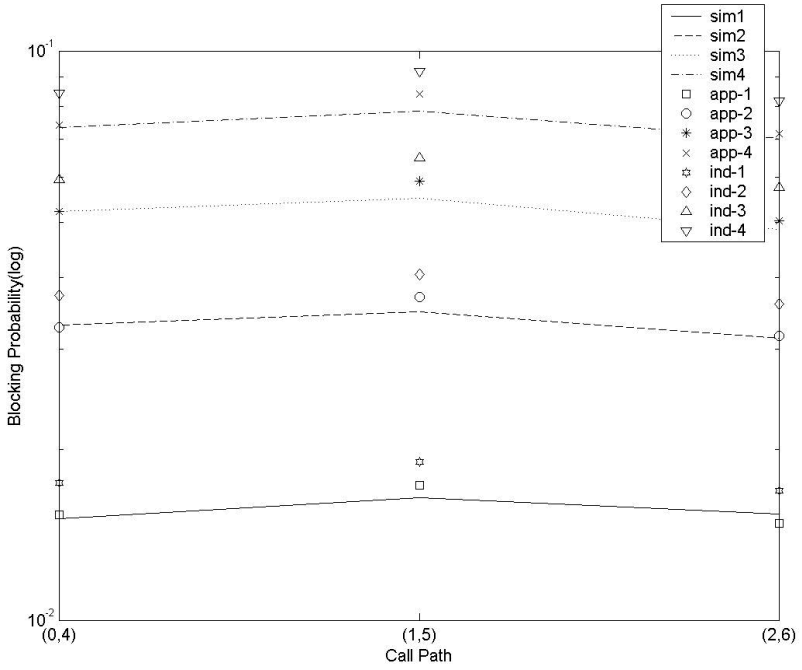


Figure 9d. Four-hop path results for class *r*, *r*=1,2,3,4 and *Ws*=128 servers.

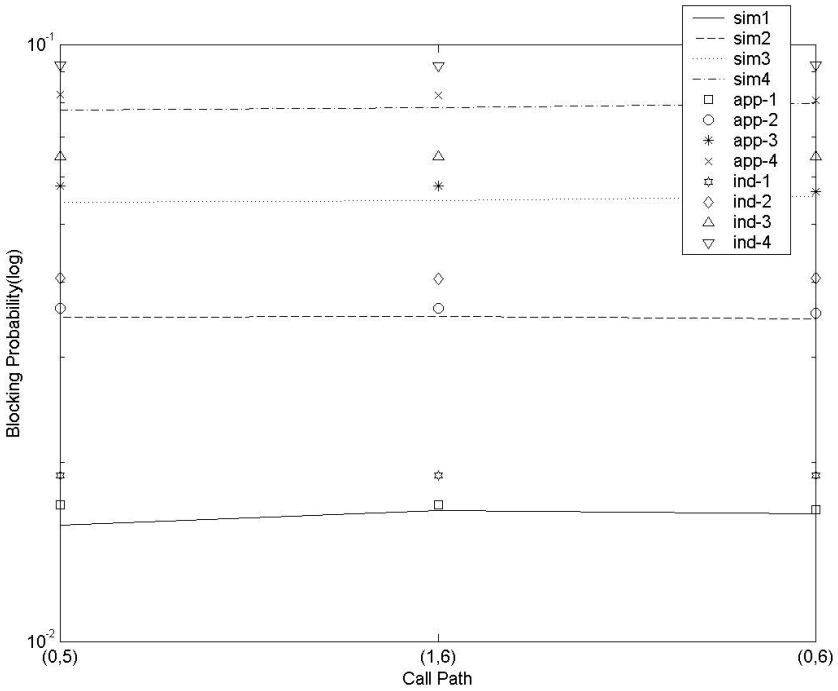


Figure 9e. Five-hop and six-hop path results for class *r*, *r*=1,2,3,4 and *Ws*=128 servers.

For all cases, we observed that the blocking probability increases with the call demand. In addition, we observed that the blocking probability increases with the number of hops per path. These results were noted in Chapter 3. Also, we note that the approximation results were more accurate than the link independence algorithm when the total number of servers, *Ws*, was equal to 128. As the total number of servers increased, the accuracy of the approximation results became closer to those of the link independence algorithm.

Chapter 5

Analysis of Traffic-Groomed Optical Networks with Alternate Routing

5.1 The Queueing Network Model

We consider a traffic-groomed optical network composed of N nodes linked in series. Each link between adjacent nodes is composed of W wavelengths. Each wavelength contains s sub-wavelength units, upon which traffic may be carried on any. The wavelengths are used to establish a number of lightpaths. The network supports traffic from multiple sources, each with a unique resource requirement measured in sub-wavelength units.

Figure 10 illustrates a four-node optical network with W=3 wavelengths on each link. The following lightpaths have been established: *0-3* on wavelength *1*, *0-2* and *2-3* on wavelength *2*, and *0-1*, *1-2*, and *2-3* on wavelength *3*. Calls arrive at any node i, i=0,1,…,3 and are destined for any node j, j=1,2,3, where $j>i$. All calls, irrespective of their source-

destination path, are grouped into R classes. Each class r call, r=1,2,…,R, has an associated demand, d_r, measured in sub-wavelength units.

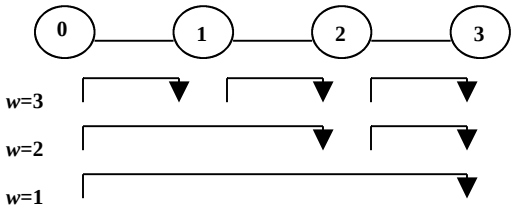


Figure 10. A four-node, three-wavelength optical network and all source-destination paths

A class r call arriving at node i with destination j, $j>i$, is initially offered to the direct lightpath between nodes i and j, if there is any. The call will be accepted on the direct lightpath, which we will refer to as the primary path, if d_r sub-wavelength units are available on the same wavelength along links i+1, i+2, …j. These sub-wavelength units are simultaneously allocated on all links upon call arrival. Upon the call departure, all d_r sub-wavelength units are simultaneously released along each link of the source-destination path. If a call cannot be serviced on its primary path, the call is offered to its pre-determined alternate paths in a sequential fashion. The alternate paths may traverse one or more lightpaths. This process repeats until the call is either accepted on one of its alternate paths or is blocked.

We model the optical network of Figure 10 as a W-level overflow system, where each level represents the lightpaths existing on a wavelength and each lightpath is represented by a multi-rate loss node with n servers. Figure 11 illustrates the queueing network model. The solid lines represent random Poisson traffic arriving to the node. The dashed lines represent

overflow traffic arriving to the node from a previous wavelength. A server represents a sub-wavelength unit. Each customer in the queueing network represents a call. Customers arrive at any node on any wavelength w, w=1,2,…W, with Poisson rate $\lambda_{w,r}$ and exponential service rate $\mu_{w,r}$. In addition to the local arriving traffic, traffic overflowing from wavelength w-1 may also arrive to the node.

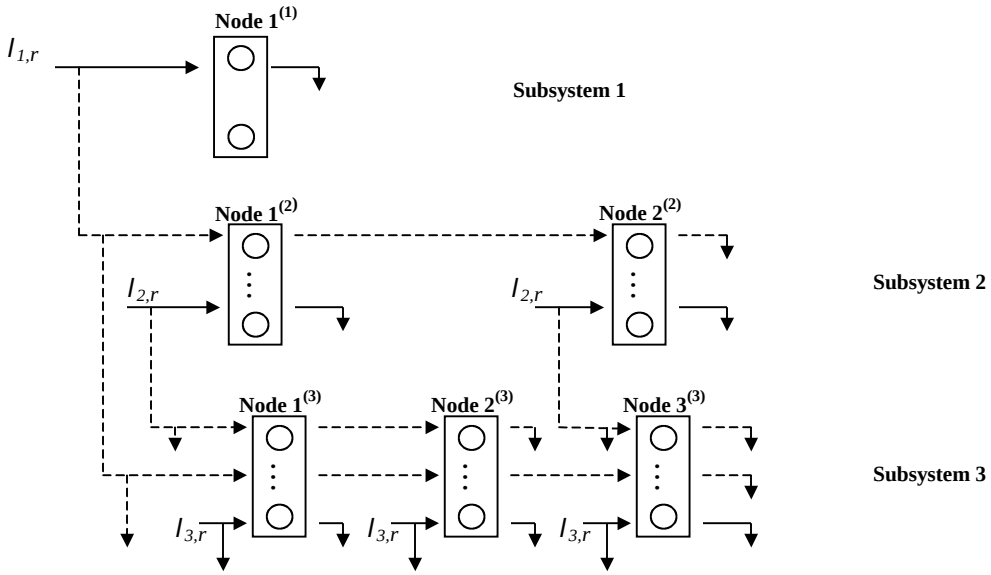


Figure 11. The queueing network and the individual subsystems

A class r local customer arriving to a node at wavelength w simultaneously occupies d_r servers and simultaneously releases them at the time the customer leaves the network. If there are not enough free servers available, the customer overflows to the pre-determined alternate node(s) on wavelength w+1. If the customer is accepted on the alternate path, then it simultaneously occupies and releases d_r servers on each node of the alternate path. If there are not enough free servers available at wavelength w+1, the customer overflows to the

alternate node(s) on wavelength w+2, and so forth. Upon reaching wavelength W, if the class r customer finds an insufficient number of servers available, the customer is blocked, i.e. lost. We note that, at wavelength W, random Poisson customers arriving to any node for service are blocked, i.e. do not overflow, if they cannot be serviced at the node.

5.2. The Decomposition Algorithm

In this section, we present the decomposition algorithm developed for the analysis of the above-described queueing network. The algorithm first decomposes the network into W subsystems. Each subsystem is then analyzed as a single-wavelength network composed of the nodes that correspond to the lightpaths established over the wavelength. Within each subsystem, each node is iteratively analyzed in isolation using an approximation method derived from the recursive method developed by Nilsson et. al [19] as well as the method developed by Basharin and Kurenkov. The details of the recursion by Nilsson et. al can be found in section 4.2.1. In order to analyze a node within a subsystem, we need information from the previous node as well as the next node. As a result, the approximation method within a subsystem is an iterative scheme where each node is analyzed one at a time continuously until a convergence criterion is satisfied.

In addition, the analysis of each node within a subsystem requires information from the previous subsystem. As a result, the decomposition method analyzes each subsystem, one at a time starting from wavelength *1* and moving downwards. Each subsystem is analyzed horizontally using the iterative scheme developed in Chapter 3. Below, we describe the approximation developed for the analysis of overflow systems by Frederick and

Hayward, as well as the extension by Basharin and Kurenkov. Following this, we present our approximation method.

5.2.1. Frederick and Hayward's Approximation

Consider a single class of customers arriving to a loss system, referred to as the primary system, consisting of N servers. All customers arrive to the node with Poisson rate λ and are serviced with rate μ. Each arriving customer requires a single server for service. Customers blocked in the primary system overflow to a secondary system of N servers where, if the customers cannot be serviced, they are blocked, i.e. lost. We define the offered traffic of an arriving customer as $A=\lambda/\mu$. The blocking probability, b_1, of an arriving customer in the primary system is determined using the Erlang-B equation

$$b_1=E(A, N) \tag{18}$$

where $E(A,N)= AE(A, N-1)/(N+AE(A, N-1))$ and $E(A,0)=1$. The mean and variance, M and V respectively, of the traffic offered to the secondary system are given by the expressions:

$$M=A*b_1 \tag{19}$$

$$V=1-M+(A/(N+1-A+M)) \tag{20}$$

We note that, while the traffic offered to the primary system is Poisson, the traffic offered to the secondary system is bursty, i.e. non-Poisson. By assuming that this traffic is

Poisson, the blocking probability of the secondary system is underestimated. Frederick and Hayward [10] developed the following approximation to address this issue.

We define the peakedness, $Z=V/M$. For Poisson traffic, $Z=1$. Frederick and Hayward determined that a non-Poisson traffic stream with mean M and $Z\neq1$, arriving to a secondary system of N servers experiences the same blocking as an equivalent system of N/Z servers offered traffic with mean M/Z and peakedness Z=1. The analysis of the secondary system is performed using the equivalent Poisson flow and equivalent maximum servers. The blocking probability of the customers in the secondary system, b_2, is approximated using

$$b_2=E_r(\mathrm{A}/Z,\ N/Z) \tag{21}$$

We note that $E_r(A, N)= \lambda^N e^{-\lambda}(\Gamma(N+1, \lambda))^{-1}$ is the generalized Erlang-B equation that allows for non-integer values. $\Gamma(N+1, \lambda)$ is the incomplete Gamma function.

Basharin and Kurenkov [3] extended the work of Frederick and Hayward to the analysis of networks supporting multi-rate calls with simultaneous resource possession. Assume that customers arrive to a primary system requesting d servers, which are simultaneously allocated and released upon call arrival and departure, respectively. The blocking of calls in the primary system is determined using the product-form solution

$$p(n)=G^{-1}(A^n/n!) \tag{22}$$

where n is the number of customers present in the network and G is the normalizing constant, defined as

$$G=SA^n/n! \quad (23)$$

The authors noted that the peakedness, Z, is d times bigger as a result of the simultaneous resource possession, and subsequent increased mean and variance. The secondary system is analyzed in one of two manners. First, the demand is fixed and number of servers is decreased by Z. That is, an equivalent system of N/Z servers is offered traffic with mean M/Z and demand d. Second, the maximum number of servers is fixed and the demand is increased by Z. That is, an equivalent system of N servers is offered traffic with mean M/Z and demand $d*Z$. We note that, the second case results in non-integer values for the call demand, which must be appropriately addressed.

5.2.2 The Approximation Method

As previously stated, the queueing network consists of levels of nodes, representing lightpaths on specific wavelengths in the original network. There are two types of nodes that exist in the network. The analysis of each is described below.

5.2.2.1 Nodes Supporting Poisson Traffic

First, we consider the case where the node supports Poisson traffic only. That is, there is no overflow traffic arriving to the node. This is the case for traffic arriving to node *1* on

wavelength *1*, in Figure 11. The node supports traffic from *R* classes, each with a demand of d_r servers, r=1,2,…,*R*. The analysis of the node is performed using the recursion developed by Nilsson et. al [19]. Let *m* denote the maximum number of servers in the node. In addition, let *k* denote the number of free servers in the node. The function $b(m,k)$, defined as the probability that *m-k* servers are busy in the node, is determined as

$$b(m,k)= b(m-1,k-1)/[1+(1/m)\sum_r A_r d_r b(m-1, d_r-1)] \quad (24)$$

with

$$b(m,0)= (1/m)\sum_r A_r d_r b(m-1, d_r-1)\,[/1+(1/m)\sum_r A_r d_r b(m-1, d_r-1)] \quad (25)$$

$A_r=A_{w,r}=\lambda_{w,r}/\mu_{w,r}$ is the offered traffic of a class *r* customer. The blocking probability is determined as

$$b_r=\sum_{j=0}^{d_r-1} b(m, j) \quad (26)$$

5.2.2.2 Nodes Supporting Overflow and Poisson Traffic

The second type of node supports both local arriving Poisson traffic and traffic overflowing from a previous subsystem, such as those on wavelength *2* of Figure 11. If either traffic cannot be serviced on the node, it overflows to the subsequent wavelength until it is blocked. We note that there may be overflow traffic from more than one node arriving.

Consider, for example, the traffic arriving to node $1^{(2)}$ in Figure 11. The node supports *R* classes of traffic overflowing from wavelength *1*, as well as *R* classes of local

Poisson traffic. As noted before, the overflow traffic is not Poisson. In order to analyze the node, we must determine an equivalent Poisson flow to represent the overflow traffic. We propose the following approximation.

The mean of the offered traffic of each class *r* overflow customer, M_r, is determined using (19). However, the variance, V_r, cannot be determined using Expression (20). This is due to the fact that the overflow of each class *r* customer at node *1* on wavelength *1* is not independent of the overflow from the other classes. To determine V_r, we assume that each class *r* customer arrives to an independent set of primary servers, N_r, on wavelength *1*. Each customer, regardless of class and original demand, requires one server. The blocking experienced by the class *r* customer is equal to the blocking experienced by a class *r* customer in the original node. This analysis allows us to analyze each class *r* customer using the Erlang-B equation, such that we can determine an equivalent number of servers that would produce the same blocking probability, assuming single-server demands, as in the original system.

The value of N_r is determined as follows. For each class *r*, N_r is initialized to zero. We define b_{test} as the current value of the blocking probability of the single-flow equivalent system, such that

$$b_{test}=E_r(A_r, N_r) \tag{27}$$

Beginning with N_r=0, b_{test} is determined using (27). The difference between b_{test} and b_r is calculated. If the difference is less than a pre-specified value, then the current value of N_r is

a suitable approximation for the single-flow system. Otherwise, if b_{test} is greater than b_r, the value of N_r is too low, and it is increased by a specific amount, referred to as *delta*. If b_{test} is lower than b_r, N_r is too high, and it is decreased by *delta*. The value of *delta* is initialized to 1 and successively halved, depending on the difference between b_{test} and b_r. While the difference is higher, the value of *delta* remains unchanged. As the difference decreases, the value of *delta* is halved. This process is repeated until the difference between b_{test} and b_r is less than the pre-specified value.

At this point, we have determined the equivalent number of servers, N_r, for each class r customer, that would produce the blocking probability, b_r, experienced by the node of the previous subsystem, assuming each class r customer arrived to an independent group of servers with rate $A_{w,r}$ and requesting a single server. N_r is used to approximate the variance of each overflowing traffic stream as

$$V_r=1-M_r+(A_r/(N_r+1-A_r+M_r)) \tag{28}$$

The peakedness of each overflowing customer, Z_r, is defined as

$$Z_r=V_r/M_r \tag{29}$$

At this point, the equivalent Poisson traffic for the overflow traffic offered to the node is defined as M_r/Z_r. We now proceed to analyze the node using the equivalent and local Poisson traffic. Basharin and Kurenkov's approximation requires either the scaling of the

maximum number of servers in the node or the demand of each class *r* call offered to the node. We note that, because multiple classes are considered in addition to the arrival of both overflow and local traffic to the node, the analysis requires the scaling of the call demands of the overflow traffic by the respective peakedness values, Z_r. The node is analyzed using (24), (25), and (26). It is to be noted that, the demands of each class *r* customer of the overflow and Poisson traffic may no longer be identical. As a result, we assume that *2R* classes of traffic, with individual demands d_r, r=1, 2, …*2R*, arrive to the node.

5.2.3 The Decomposition Algorithm

We describe the decomposition algorithm using the four-node optical network modeled by the queueing network shown in Figure 11. The queueing network is decomposed into three subsystems, corresponding to the three wavelengths shown in Figure 10. Subsystem *1* represents the lightpath on wavelength *1*, subsystem *2* represents the lightpaths on wavelength *2*, and subsystem *3* represents the lightpaths on wavelength *3*. $A_r^{(x)} = I_r^{(x)} / m_r^{(x)}$ is the offered traffic of a class *r* customer on subsystem *x*, where *x*=1,2,3. $M_{y,r}^{(x)}$ and $Z_{y,r}^{(x)}$ denote the mean and peakedness, respectively, of the offered traffic of a class *r* customer overflowing from node *y* of subsystem *x*. $\underline{A}_{y,r}^{(x)}$ and $\underline{d}_{y,r}^{(x)}$ denote the equivalent offered traffic and equivalent demand, respectively, of a class *r* customer overflowing to node *y* on subsystem *x*. Finally, $b_{y,r}^{(x)}$ is the blocking probability of a class *r* customer at node *y* of subsystem *x*. All blocking probabilities are initialized to zero.

Beginning with subsystem *1*, each subsystem is analyzed in iteration. Within a subsystem, each node is analyzed in iteration as well using the single-node decomposition method assuming link independence. For subsystem *1*, the only traffic arriving to the node is Poisson traffic. The node is analyzed using (24), (25), and (26). The blocking probabilities, $b_{1,r}^{(1)}$, are stored and the algorithm shifts to subsystem *2*.

At subsystem *2*, the algorithm begins with node *1*. The equivalent Poisson traffic for each class *r* customer overflowing from subsystem *1* is first determined, using (19), (28), and (29), and the most recently calculated values for $b_{1,r}^{(1)}$. We note that the overflow traffic arriving to this node continues on to node *2*, if accepted. As a result, the equivalent Poisson traffic is

$$\underline{A}_{1,r}^{(2)} = M_{1,r}^{(1)} \ (1 - b_{2,r}^{(2)}) / Z_{1,r}^{(1)} \tag{30}$$

Expression (30) assumes that the overflow traffic is not blocked at the downstream nodes of the alternate path, in this case, node *2*. This allows for the calculation of the blocking experienced by a customer, due to an insufficient number of servers on node *1* of the alternate path. The equivalent demands of the overflow traffic are determined as

$$\underline{d}_{1,r}^{(2)} = d_r * Z_{1,r}^{(1)} \tag{31}$$

Because there is both Poisson and overflow traffic arriving to the node, the maximum number of servers per node is fixed and the demands of the overflow traffic are scaled by their corresponding peakedness.

At this point, we have a total of *2R* classes arriving to node *1* on wavelength *2*, *R* classes of overflow traffic with equivalent offered traffic $\underline{A}_{1,r}^{(2)}$ and equivalent demands $\underline{d}_{1,r}^{(2)}$, and *R* classes of local Poisson traffic with offered traffic $A_r^{(x)}$ and demands d_r. The nodes are analyzed using (24), (25), and (26) assuming *2R* classes. The algorithm then shifts to node *2* on wavelength *1*.

At node *2*, the overflow traffic arriving to the node is the amount of traffic that is not blocked at node *1*. The equivalent Poisson offered traffic and equivalent demand is

$$\underline{A}_{2,r}^{(2)} = M_{1,r}^{(1)} \ (1 - b_{1,r}^{(2)}) / Z_{1,r}^{(1)} \tag{32}$$

and

$$\underline{d}_{2,r}^{(2)} = d_r * Z_{1,r}^{(1)} \tag{33}$$

We note that the mean and peakedness of the traffic overflowing from node *1* on subsystem *1* is the same. The node is analyzed using (24), (25), and (26).

At this point, a single iteration has completed within subsystem *2*. The algorithm executes successive iterations until the blocking probabilities of each node within the subsystem have converged within a criterion, ε, which is set to 10^{-3}. Once the results have converged, the blocking probabilities of all source-destination paths within the subsystem are calculated by appropriately combining the results from each node. For example, the blocking

probability of the class *r* renewal traffic on path (*0,3*) of subsystem *2* is $1-(1-b_{1,r}^{(2)})(1-b_{2,r}^{(2)})$. The algorithm then shifts to subsystem *3*.

At subsystem *3*, each node is analyzed in iteration using the most recently calculated blocking probabilities from each node in subsystem *2*. Beginning with node *1* on wavelength *3*, the equivalent Poisson traffic for each class *r* customer overflowing from node *1* on wavelength *2* is determined. We note that there is a total of *2R* classes of overflow traffic offered to node *1* on wavelength *3*, *R* classes from the overflow path traversing both nodes *1* and *2* on wavelength *2*, and *R* classes from the local traffic traversing node *1*. The mean and peakedness of the overflow traffic is determined using (19), (28), and (29). The equivalent Poisson traffic is determined as

$$\underline{A}_{1,r}^{(3)} = M_{1,r}^{(2)} \ (1-b_{2,r}^{(3)})(1-b_{3,r}^{(3)})/Z_{1,r}^{(2)}; \qquad r=1,2,\ldots,R \tag{34}$$

$$\underline{A}_{1,r}^{(3)} = M_{1,r}^{(2)} \ (1-b_{2,r}^{(3)})/Z_{1,r}^{(2)}; \qquad r=R+1,R+2,\ldots,2R \tag{35}$$

The traffic overflowing from node *1* on subsystem *2* traverses different nodes in subsystem *3*. As a result, (34) and (35) account for the alternate path of each, assuming the customer is not blocked on its downstream nodes. The equivalent demands of the overflow traffic are

$$\underline{d}_{2r}^{(3)} = d_r * Z_{1,r}^{(2)}; \ r=1,2,..,2R \tag{36}$$

At this point, we have a total of *3R* classes arriving to node *1* on subsystem *3*, *2R* classes overflowing from node *1* of subsystem *2*, and *R* classes of local Poisson traffic. The node is analyzed using (24), (25), and (26). The algorithm then shifts to node *2*.

At node *2*, the equivalent Poisson offered traffic and equivalent demands are

$$\underline{A}_{2,r}^{(3)} = M_{1,r}^{(2)} \ (1-b_{1,r}^{(3)})(1-b_{3,r}^{(3)})/Z_{1,r}^{(2)}; \qquad r=1,2,...,R \tag{37}$$

$$\underline{A}_{2,r}^{(3)} = M_{1,r}^{(2)} \ (1-b_{1,r}^{(3)})/Z_{1,r}^{(2)}; \qquad r=R+1,R+2,...,2R \tag{38}$$

$$\underline{d}_{2,r}^{(3)} = d_r * Z_{1,r}^{(2)}; \ r=1,2,..,2R \tag{39}$$

We note that (37) assumes the overflow traffic is not blocked on any downstream or upstream node of its alternate path. Similarly, (38) does the same. Node *2* is analyzed using (24), (25), and (26).

At node 3, the equivalent Poisson traffic and equivalent demands are determined as

$$\underline{A}_{3,r}^{(3)} = M_{1,r}^{(2)} \ (1-b_{1,r}^{(3)})(1-b_{2,r}^{(3)})/Z_{1,r}^{(2)}; \qquad r=1,2,..,R \tag{40}$$

$$\underline{A}_{3,r}^{(3)} = M_{2,r}^{(2)} \ /Z_{2,r}^{(2)}; \qquad r=R+1,R+2,..,2R \tag{41}$$

$$\underline{d}_{3,r}^{(3)} = d_r * Z_{1,r}^{(2)}; \ r=1,2,..,R \tag{42}$$

$$\underline{d}_{3,r}^{(3)} = d_r * Z_{1,r}^{(2)}; \ r=R+1,R+2,..,2R \tag{43}$$

We note that (40) and (42) account for the overflow traffic originating at node *1* in subsystem *2* and traversing all three nodes. Expressions (41) and (43) account for the overflow traffic

overflowing from node *2* on wavelength *2*. The node is then analyzed using (24), (25), and (26).

At this point, a single iteration within subsystem *3* is complete. The algorithm executes successive iterations until the blocking probabilities of each node within the subsystem have converged within e. Once the results have converged, the blocking probabilities of all source-destination paths within the subsystem are calculated by appropriately combining the results from each node. Because this is the last subsystem, the blocking is determined as follows. For local traffic on node *1* of wavelength *3*, the blocking probability is simply $b_{1,r}^{(3)}$, $r=2R+1,2R+2,\ldots,3R$. The blocking experienced by a call in a previous subsystem, such as subsystem *1*, is $M_{1,r}^{(2)}\ [1-(1-b_{1,r}^{(3)})(1-b_{2,r}^{(3)})(1-b_{3,r}^{(3)})]/A_r^{(1)}$, $r=1,2,\ldots,R$. We note that the blocking for this path uses the offered traffic arriving from subsystem *2*, the original offered traffic to subsystem *1*, and the blocking probability of the call in subsystem *3*.

5.3 Numerical Results

In this section, we discuss the accuracy of the decomposition algorithm by comparing the results obtained to simulation data. For each case, we consider the four-ADM optical network of Figure 10, modeled by the *W*-level queueing network shown in Figure 11. Each link between two adjacent ADMs is composed of 3 wavelengths, each wavelength containing *s* servers.

All possible source-destination paths are considered and, for each path, the call is first offered to the direct lightpath between the source and destination node. If the call cannot be

serviced on the direct lightpath, it overflows to its alternate paths sequentially, until the call is either accepted or blocked. We assume four classes of traffic, i.e. R=4, with demands of d_1=2, d_2=4, d_3=6, and d_4=8 servers.

The accuracy of the decomposition algorithm was evaluated under different cases, where the arrival rates and the number of servers vary. We assume, for presentation purposes, that the arrival rate of all class r calls, regardless of source-destination path, is λ_r. For each case, λ_1 was assigned the highest arrival rate, and the arrival rate of each successive class r call was decreased by a small amount from the previous as follows. We assume s=128, 256 servers. For s=128 servers, cases 1, 2, and 3, set λ_1 to 5000, 8000, and 10000, respectively. For s=256, cases 1 and 2 set λ_1 to 8000 and 10000, respectively. For both cases, the arrival rate of each successive class r call was decreased by 500, and the service rate of all calls was set to 500.

Figures 12, 13, 14, and 15 show the results for the wavelengths *1* and *2*, and wavelengths *3* for 128 and 256 servers, respectively. Each figure plots the blocking probability of each source-destination path, labeled as (source, destination), as a function of the call class. In addition, the simulation results are plotted with a 95% confidence interval.

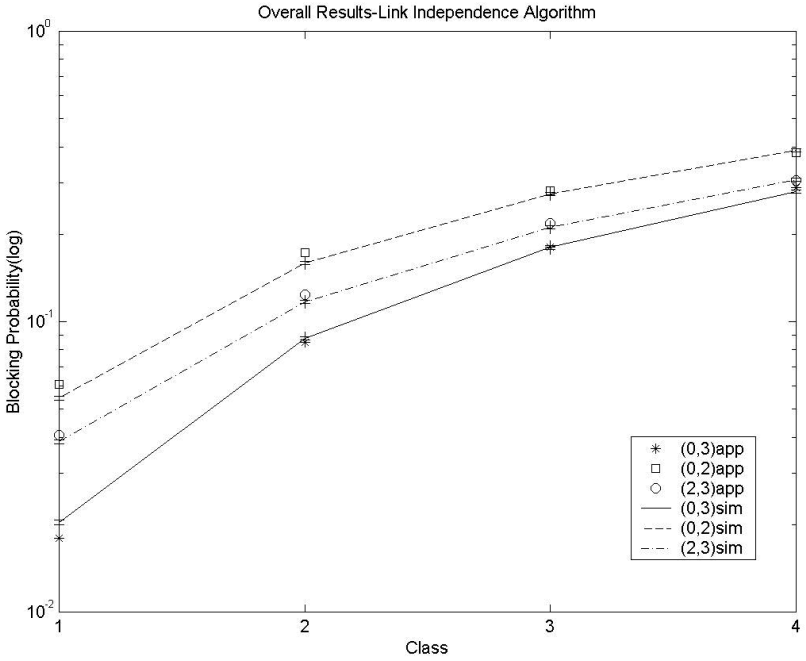


Figure 12a. Wavelength 1 and 2 results for *s*=128 servers and l_1=5000

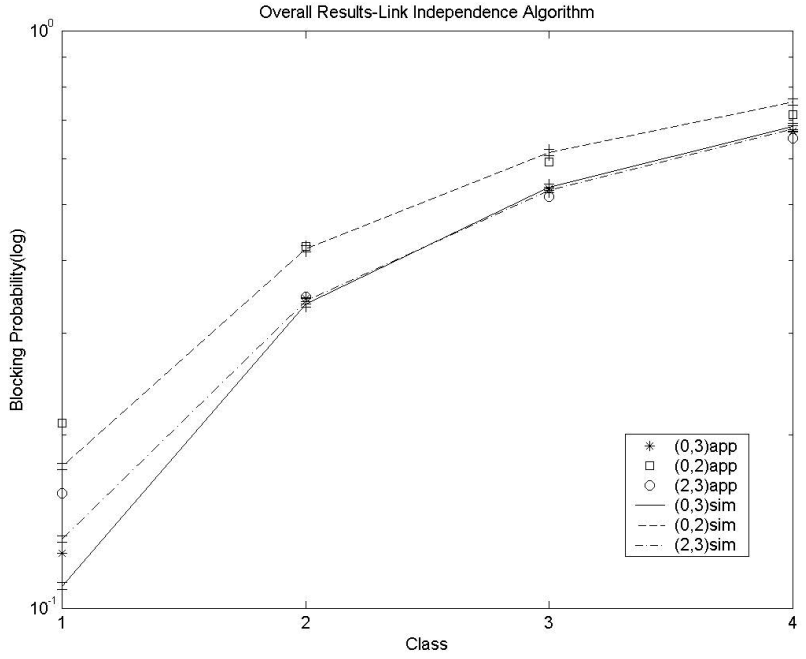


Figure 12b. Wavelength 1 and 2 results for *s*=128 servers and l_1=8000

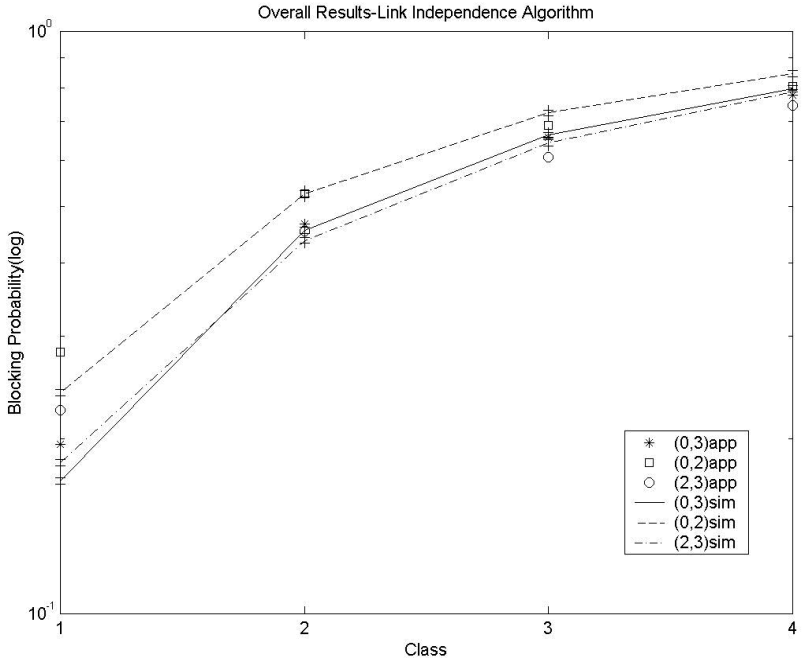


Figure 12c. Wavelength 1 and 2 results for *s*=128 servers and l_1=10000

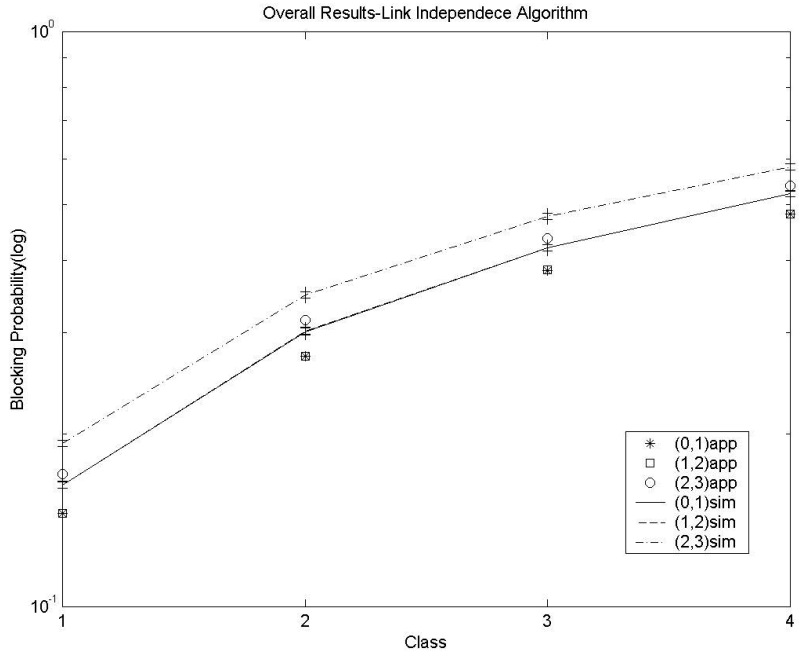


Figure 13a. Wavelength 3 results for *s*=128 servers and l_1=5000

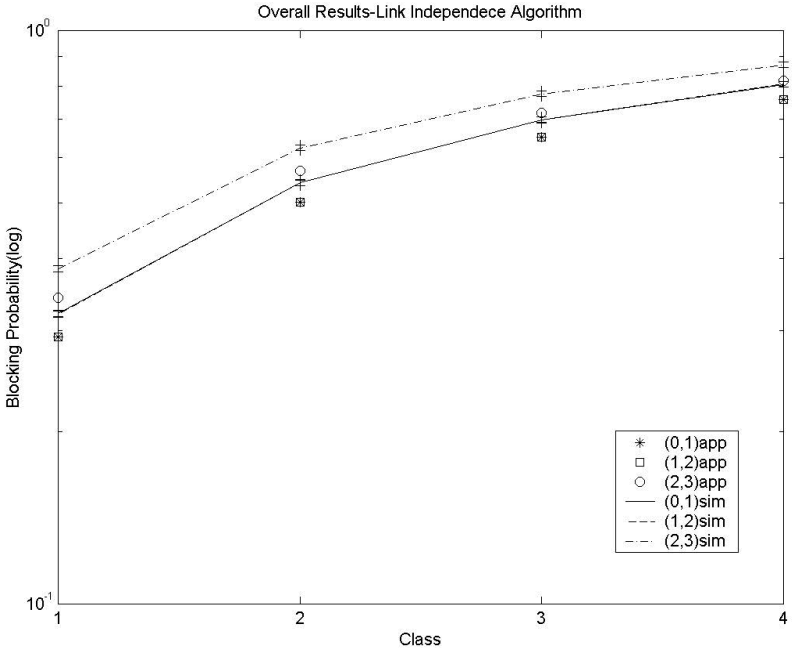


Figure 13b. Wavelength 3 results for *s*=128 servers and l_1=8000

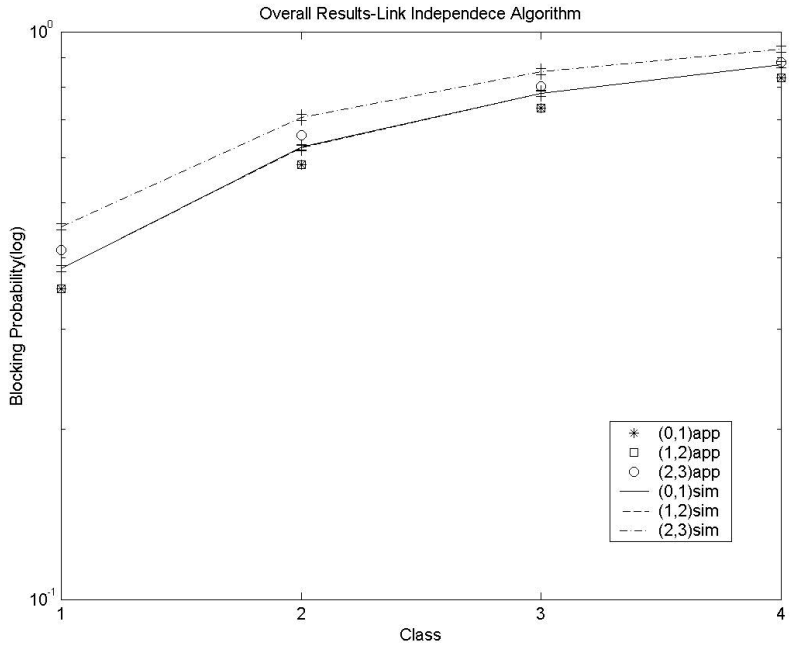


Figure 13c. Wavelength 3 results for *s*=128 servers and l_1=10000

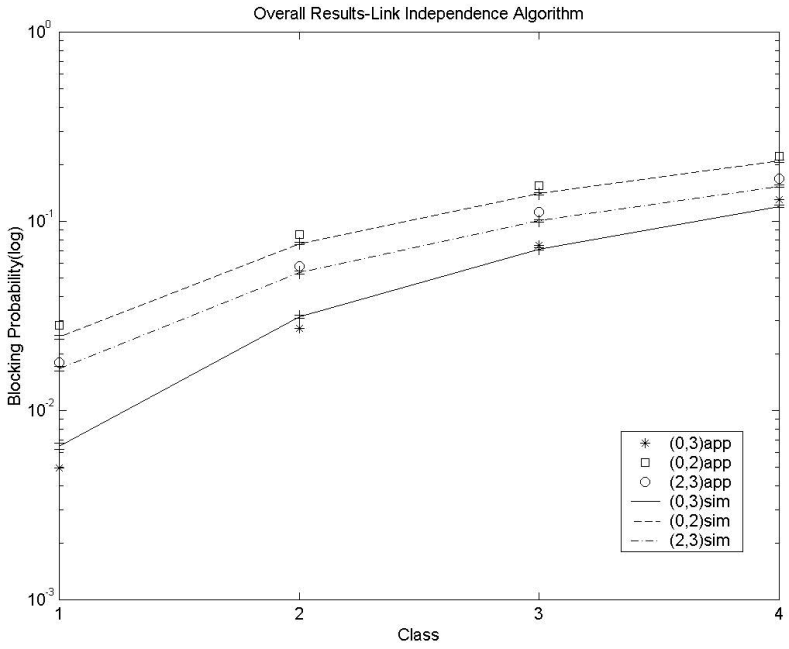


Figure 14a. Wavelength 1 and 2 results for *s*=256 servers and l_1=8000

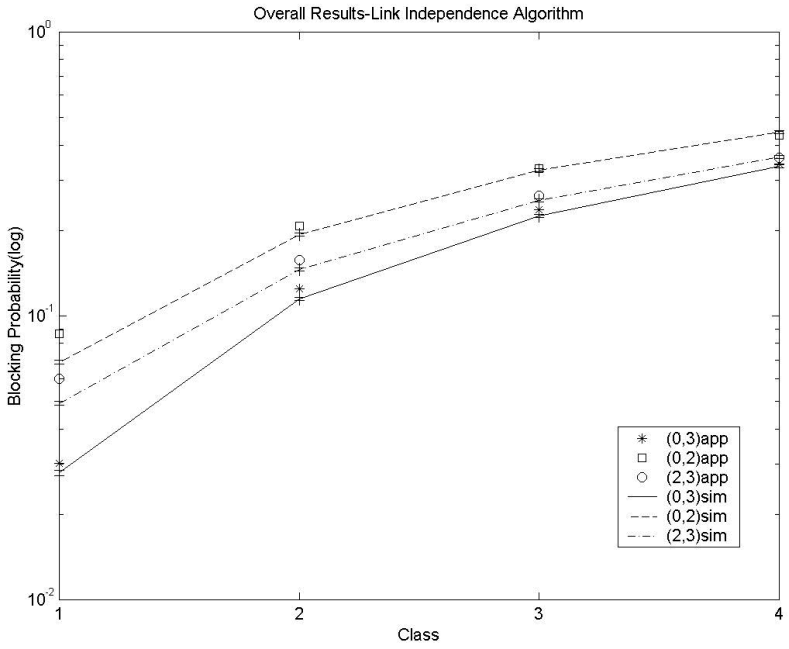


Figure 14b. Wavelength 1 and 2 results for *s*=256 servers and l_1=10000

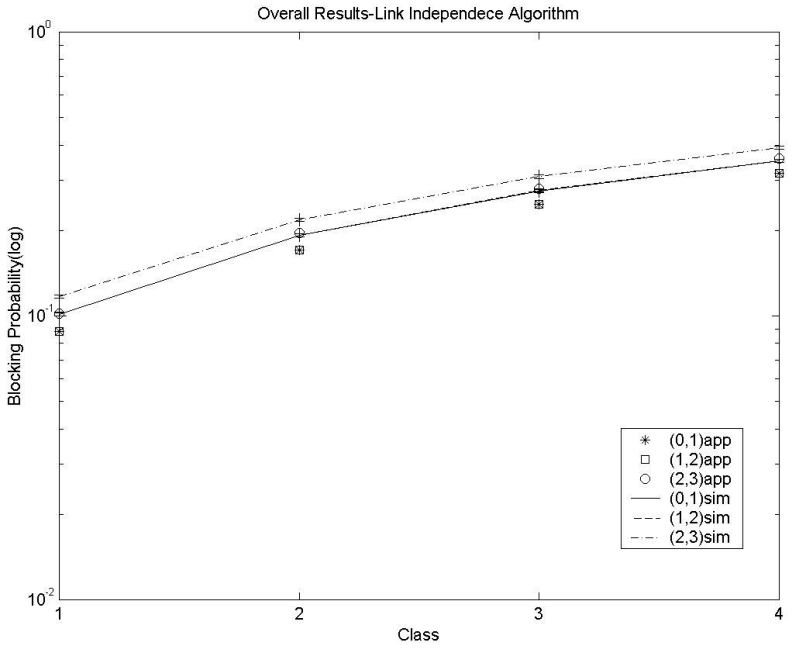


Figure 15a. Wavelength 3 results for s=256 servers and l_1=8000

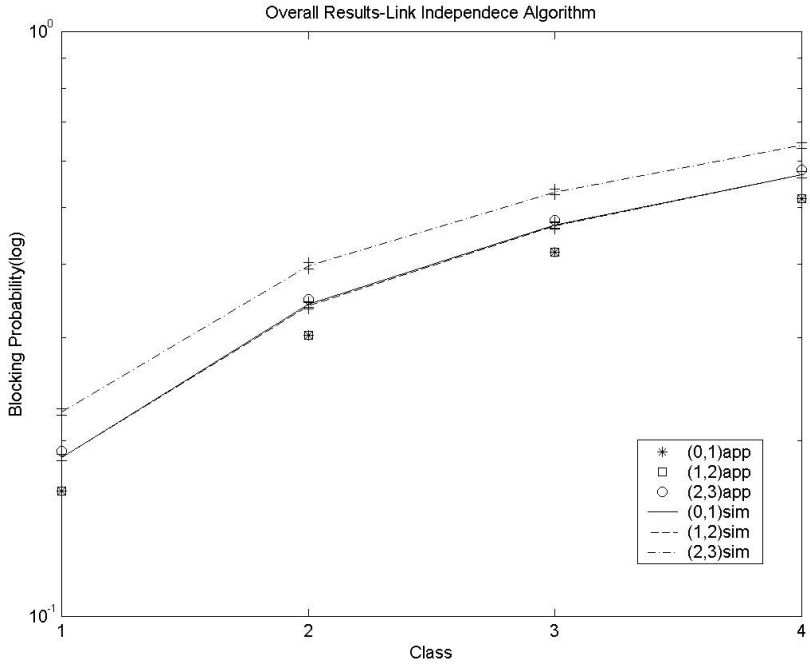


Figure 15b. Wavelength 3 results for s=256 servers and l_1=10000

We observe that, overall, the decomposition provides good accuracy for all cases. Tables 4 and 5 show the per-class and per wavelength relative errors for each case. As is shown, the average relative errors are fairly low, meaning that the decomposition algorithm, using the single-node decomposition algorithm assuming link independence has good accuracy.

		Class 1	**Class 2**	**Class 3**	**Class 4**
128	**Case 1**	0.029	0.022	0.015	0.013
	Case 2	0.039	0.014	0.013	0.014
	Case 3	0.038	0.013	0.014	0.013
256	**Case 1**	0.041	0.031	0.027	0.025
	Case 2	0.046	0.027	0.019	0.015

Table 4. Per-class relative errors

		Wave 1	**Wave 2**	**Wave 3**
128	**Case 1**	0.044	0.048	0.092
	Case 2	0.047	0.070	0.076
	Case 3	0.054	0.081	0.065
256	**Case 1**	0.125	0.099	0.108
	Case 2	0.058	0.090	0.109

Table 5. Per-wavelength relative errors

Chapter 6

Conclusion and Future Research

WDM networks appear to be a viable solution to the bandwidth limitations experienced by current backbone networks. As the amount of traffic supported and their subsequent demands continue to increase, traffic grooming allows for a closer match between the resource requirements and the amount of resources allocated. Our focus in this research is the performance analysis of traffic-groomed tandem WDM networks with varying network architectures supporting multi-rate traffic.

From Chapter 3, we determined a decomposition algorithm that provided an approximate analysis of a traffic-groomed tandem WDM network. Assuming a single-wavelength existed per link, we used a modified version of Courtois' method in order to

analyze each subsystem. It was determined that, while providing good accuracy, the algorithm was limited to small networks.

From Chapter 4, we extended the work of Chapter 3 to a large number of wavelengths and a large number of servers per wavelength. A modified version of the recursion developed by Nilsson et. al was developed to analyze each subsystem. We found that overall, the algorithm provided a good approximation to simulation results. However, the approximate results were comparable to the link independence results as the number of servers increases.

Finally, from Chapter 5, we determined a decomposition algorithm for the analysis of a traffic-groomed tandem WDM network with alternate routing. The decomposition algorithm used a modified version of the Frederick and Hayward's approximation as well as the Nilsson et. al recursion, assuming link independence, in order to analyze the network. It was determined that the decomposition algorithm provided good accuracy overall.

6.1 Future Research

Our work can be extended in two directions. First, as stated before, the decomposition algorithm based on the modified Courtois' method was limited to small networks. In addition, the extension of this work allowed for traffic bifurcation. A good extension for this problem is the removal of the traffic bifurcation assumption in the analysis of links with a large number of wavelengths.

Second, the analysis of the traffic-groomed network employing alternate routing focused on a tandem network. Analyzing arbitrary mesh topologies as well as the addition of multiple flows to each node can extend this work.

References

[1] Akimaru, H. and Takahashi, H., "An Approximate Formula for Individual Call Losses in Overflow Systems", IEEE Transactions on Communications, 31(6)(1983), 808-811.

[2] Alyatama, A., "Wavelength Decomposition Approach for Computing Blocking Probabilities in WDM Optical Networks Without Wavelength Conversion", Journal of Computer Networks, (To appear).

[3] Basharin, G. P. and Kurenkov, B. E., "Analysis of Overflow in Circuit Switching Networks", Problems of Information Transmission, 23(3)(1987), 216-223.

[4] Birman, A., "Computing Approximate Blocking Probabilities for a Class of All-Optical Networks", IEEE Journal on Selected Areas in Communications, 14(5)(1996), 852-857.

[5] Brochin, F. and Pradel, E., "A Call Traffic Model for Integrated Services Digital Networks", IEE Global Telecommunications Conference, (5)(1992), 1508-1512.

[6] Conway, A.E., Pinsky, E., and Tridandapani, S., "Efficient Decomposition Methods for the Analysis of Multi-Facility Blocking Models", Journal of the ACM, 41(4)(1994), 648-675.

[7] Courtois, P.J., Decomposability: Queueing and Computer System Applications, Academic Press, NY, 1977.

[8] Dutta, R. and Rouskas, G. N., "Traffic Grooming in WDM Networks: Past and Future", IEEE Network, 16(6)(2002), 46-56.

[9] Dziong, Z. and Roberts, J.W., "Congestion Probabilities in a Circuit-Switched Integrated Services Network", Performance Evaluation, 7(1987), 267-284.

[10] Frederick, A.A., "Congestion in Blocking Systems-A Simple Approximation Technique", Bell System Technical Journal, 59(6)(1980), 805-827.

[11] "Overflow Approximations for Non-Random Inputs", ITU-T Recommendation E.524, 1999.

[12] Iversen, V., Stepanov, S.N., and Kostrov, V.O., "The Derivation of a Stable Recursion for Multi-Service Models", International Conference on Next Generation Teletraffic and WiredWireless Advanced Networking, 2004.

[13] Iversen, V., Teletraffic Engineering and Network Planning, COM, DTU, 2005.

[14] Kaufman, J.S., "Blocking in a Shared Resource Environment", IEEE Transactions on Communications, 29(10)(1981), 1474-1481.

[15] Kuczura, A., "The Interrupted Poisson Process as an Overflow Process", Bell System Technical Journal, 52(1973), 437-448.

[16] Listanti, M, Parisi, D., and Sabella, R., "Optical Data Network Based on the Switchless Concept: Analysis and Dimensioning", Photonic Network Communications, 3(4)(2001), 363-375.

[17] Liu, Y., Iversen, V., Dickmeiss, A., and Larsen, M., "Individual Overflow Characteristics for Loss System with Multi-Slot Traffic and State-Dependent Poisson Input Processes", 11th Nordic Teletraffic Seminar, 1993.

[18] Mosharaf, K., Talim, J., and Lambadaris, I., "Network Dimensioning in WDM Networks with Dynamic Traffic Grooming", Manuscript, Computer Science Department, Carleton University, 2003.

[19] Nilsson, A., Perry, M., Iversen, V., and Gershtc, A., "On Multi-Rate Erlang-B Computations", 16th ITC, 1999.

[20] Ramesh, S., Rouskas, G.N., and Perros, H.G., "Computing Blocking Probabilities in Multiclass Wavelength-Routing Networks with Multicast Calls", IEEE Journal on Selected Areas in Communications, 20(1)(2002), 89-96.

[21] Schehrer, R., "On the Calculation of Overflow Systems with a Finite Number of Sources and Full Available Groups", IEEE Transactions on Communications, 26(1)(1978), 75-82.

[22] Srinivasan, R. and Somani, A.K., "Analysis of Multi-Rate Traffic in WDM Grooming Networks", 11th International Conference on Computer Communications and Networks, 2002, 295-301.

[23] Thiagarajan, S. and Somani, A.K., "Capacity Correlation Model for WDM Networks with Constrained Grooming Capabilities", IEEE International Conference on Communications, 5(2001), 1592-1596.

[24] Thiagarajan, S. and Somani, A.K., "Performance Analysis of WDM Optical Networks with Grooming Capabilities", SPIE Technical Conference on Terabit Optical Networking: Architecture, Control, and Management Issues, 2000.

[25] Washington, A.N., Hsu, C., Perros, H.G., and Devetsikiotis, M., "Approximation Techniques for the Analysis of Large Traffic-Groomed Tandem Optical Networks", To Be Published in the Proceedings of the 38th Annual Simulation Symposium, April 2005.

[26] Washington, A.N. and Perros, H.G., "Call Blocking Probabilities in a Traffic Groomed Tandem Optical Network", Journal of Computer Networks, 45(2004).

[27] Xin, C., Qiao, C. and Dixit, S., "Traffic Grooming in Mesh WDM Optical Networks-Performance Analysis", OPTICOMM, 2003.

[28] Zhu, K. and Mukherjee, B., "A Review of Traffic Grooming in WDM Optical Networks: Architectures and Challenges", Optical Networks Magazine, 2003, 55-64.

[29] Zhu, H., Zang, H., Zhu, K., and Mukherjee, B., "Dynamic Traffic Grooming in WDM Mesh Networks Using a Novel Graph Model", IEEE Global Telecommunications Conference, 3(2002), 2681-2685.

[30] Zhu, Y., Rouskas, G.N., and Perros, H.G., "A Path Decomposition Approach for Computing Blocking Probabilities in Wavelength-Routed Networks", IEEE Transactions on Networking, 8(6)(2000), 742-762.

.

Appendix

In this section, we present further numerical results obtained by the decomposition algorithm of Chapter 4, using the modified version of the recursion developed by Nilsson et. al. The network was analyzed under four cases, where the number of servers per link, *Ws*=128, 256, 512, and 1024 servers.

The analysis of the each subsystem was performed using the approximation algorithm described in 4.2. Figures 16, 17, and 18 show the results for 256, 512, and 1024 servers, respectively.

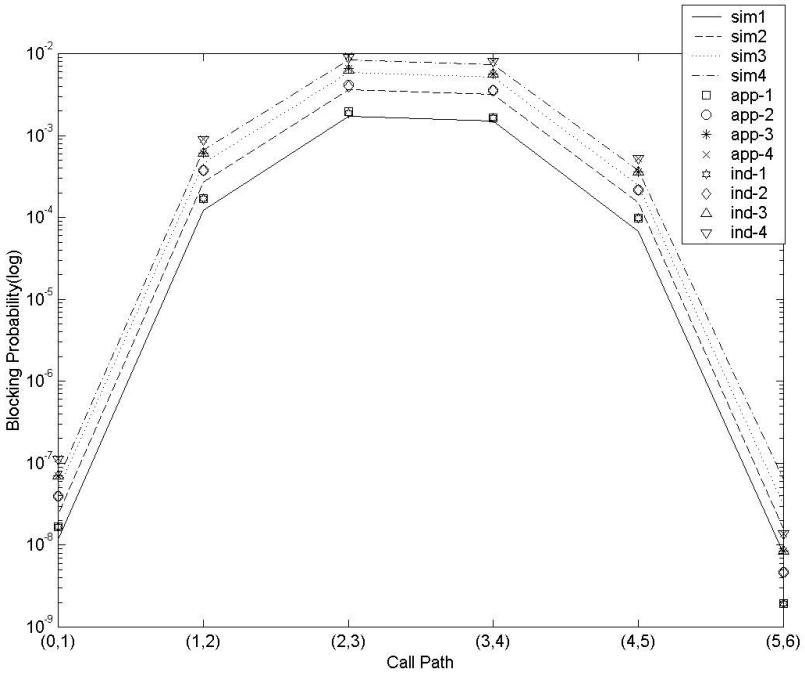


Figure 16a. One-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=256 servers.

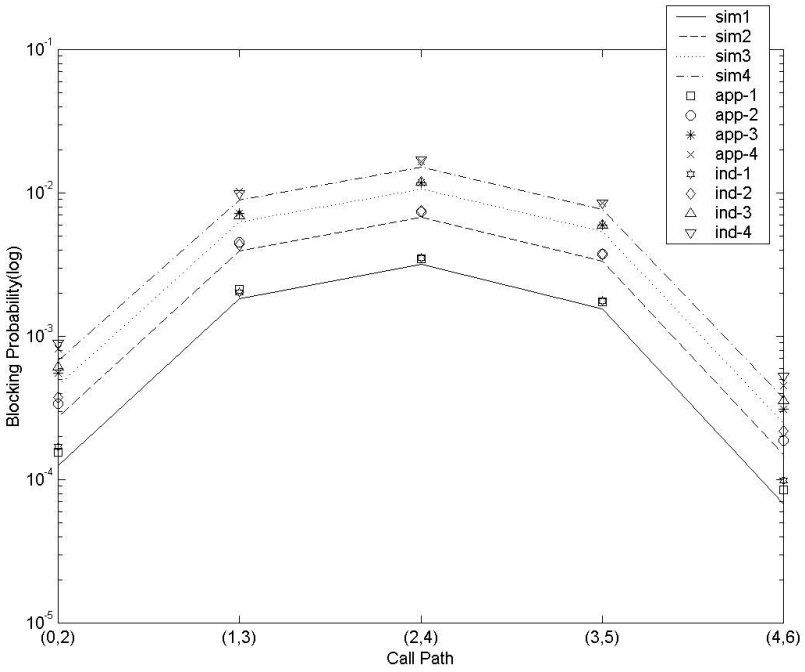


Figure 16b. Two-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=256 servers.

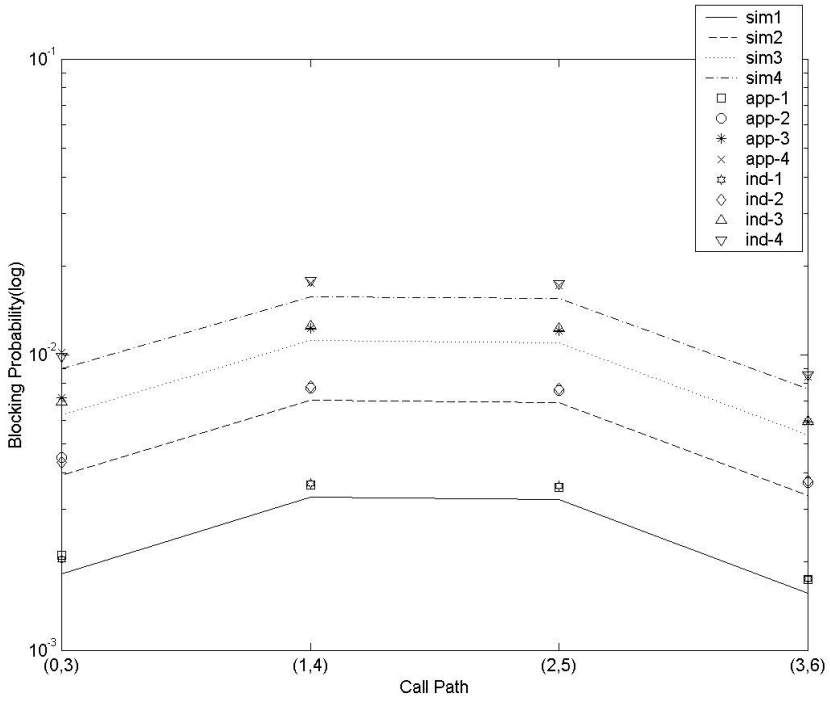


Figure 16c. Three-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=256 servers.

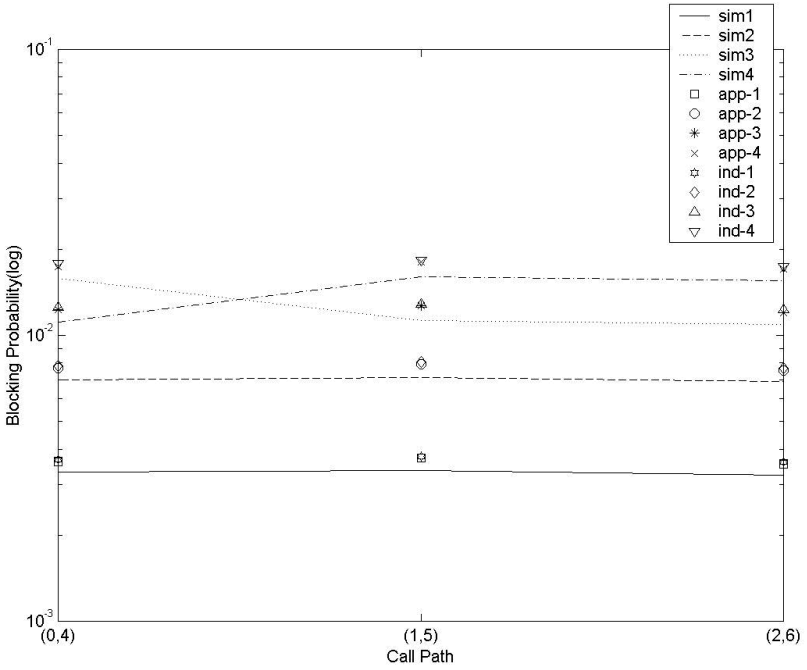


Figure 16d. Four-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=256 servers.

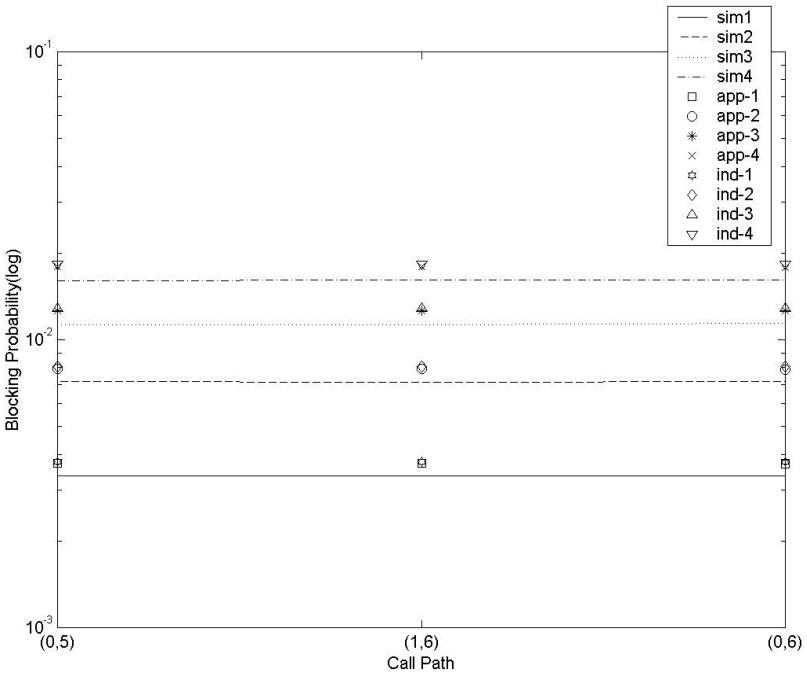


Figure 16e. Five-hop and six-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=256 servers.

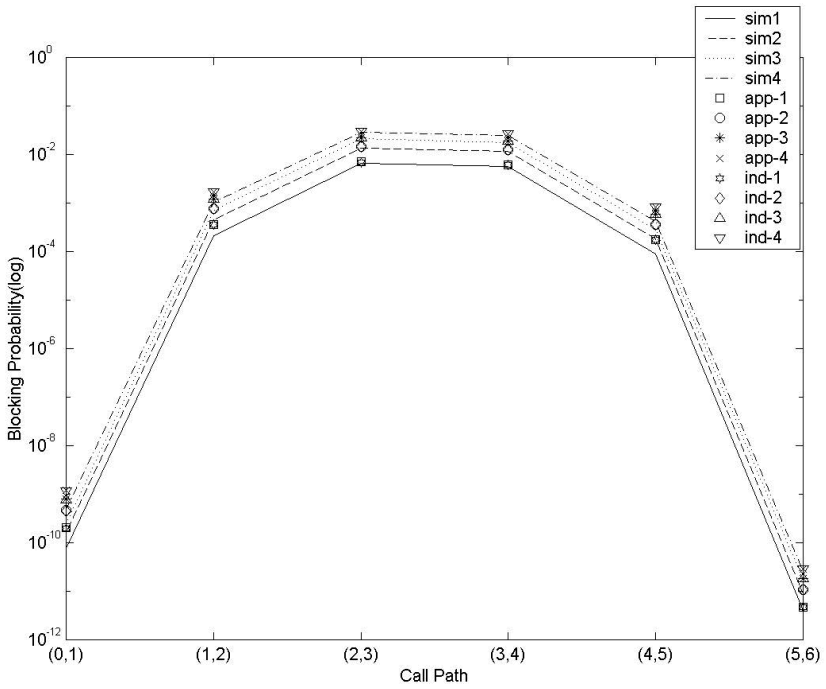


Figure 17a. One-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=512 servers.

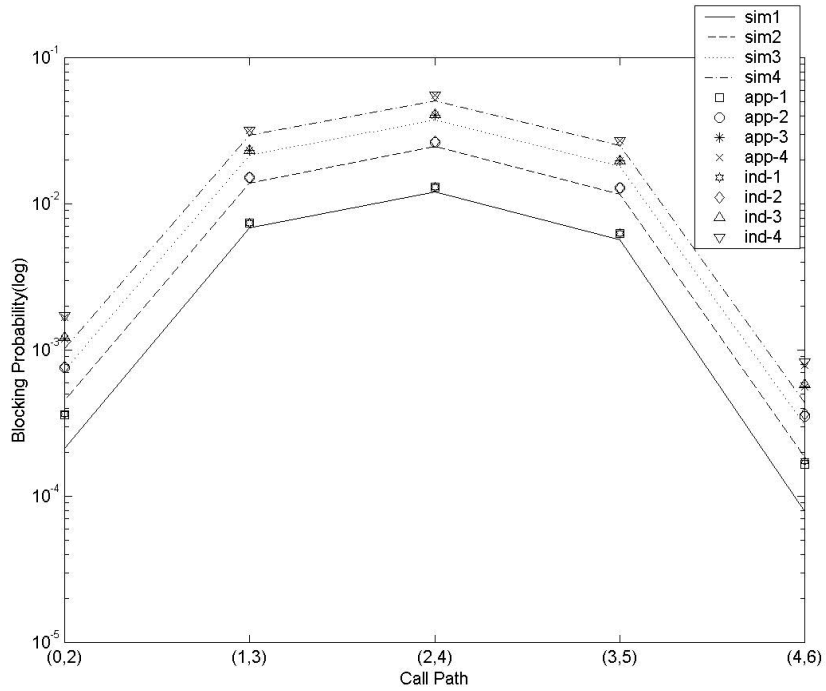


Figure 17b. Two-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=512 servers.

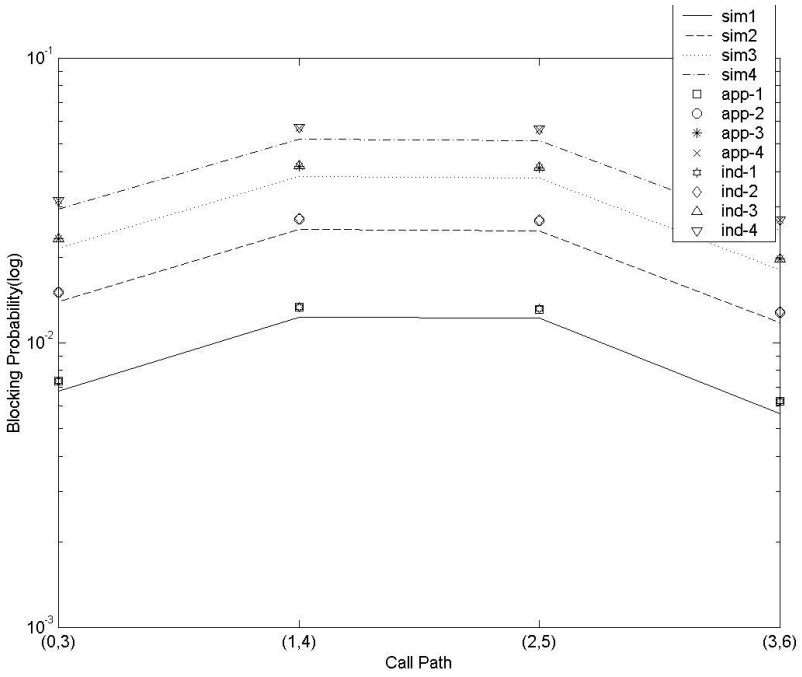


Figure 17c. Three-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=512 servers.

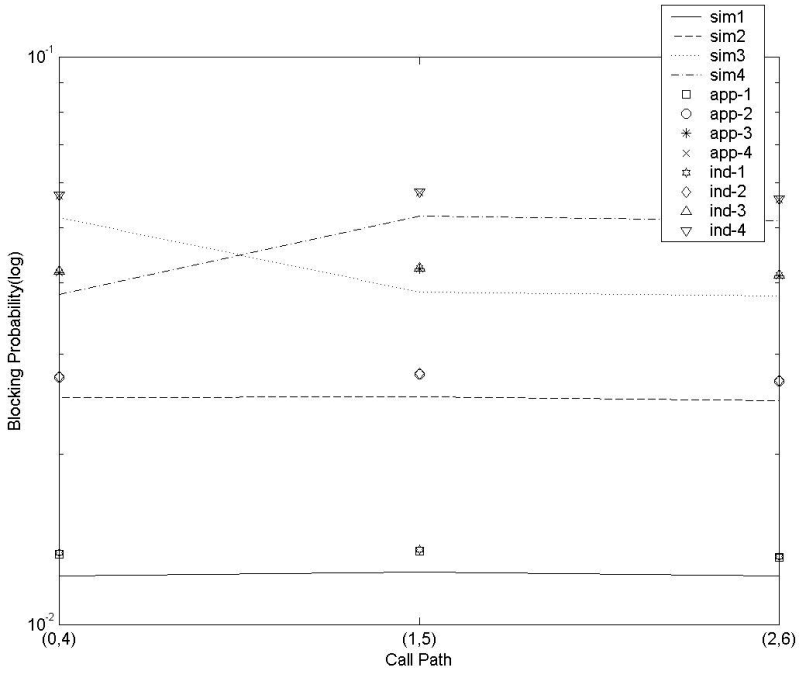


Figure 17d. Four-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=512 servers.

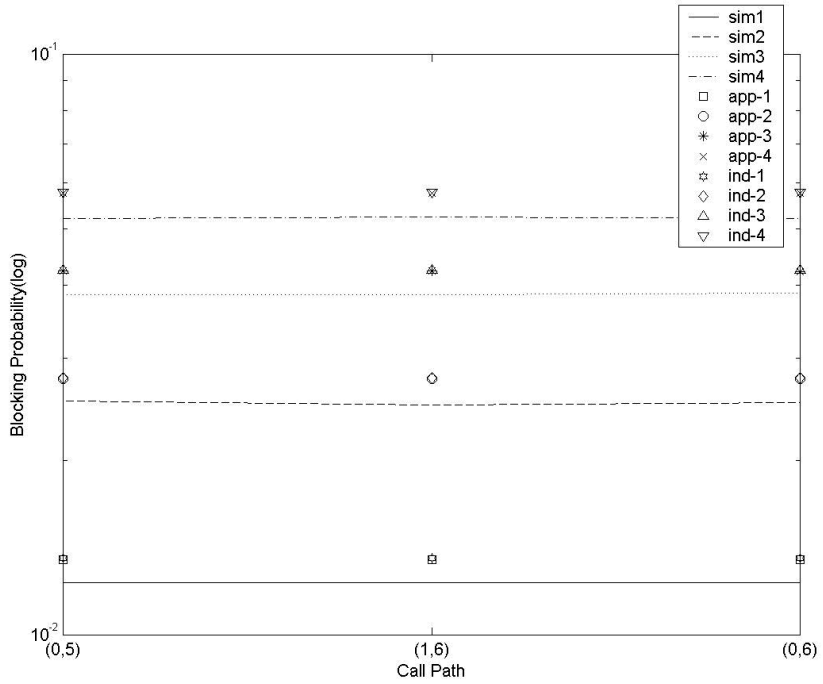


Figure 17e. Five-hop and six-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=512 servers.

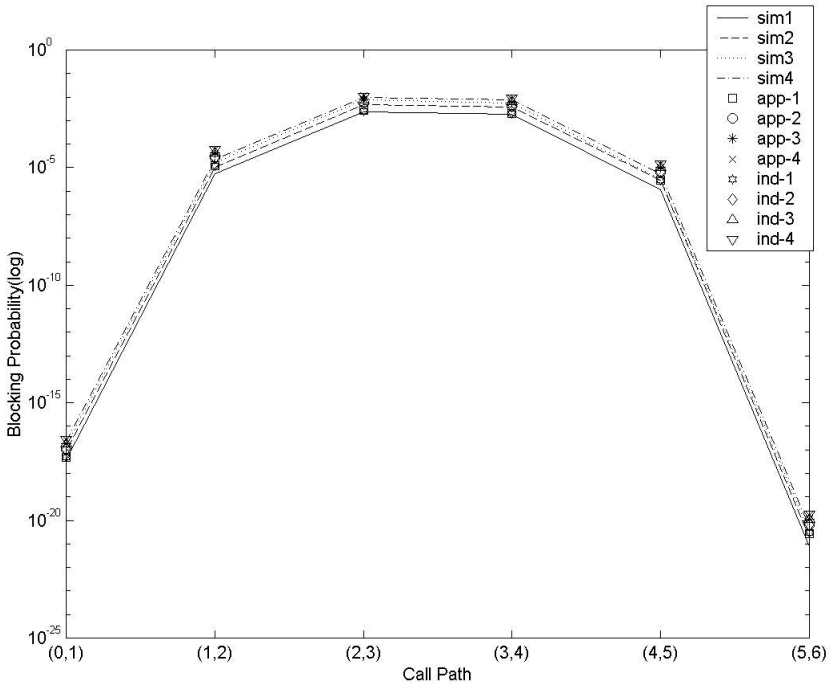


Figure 18a. One-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=1024 servers.

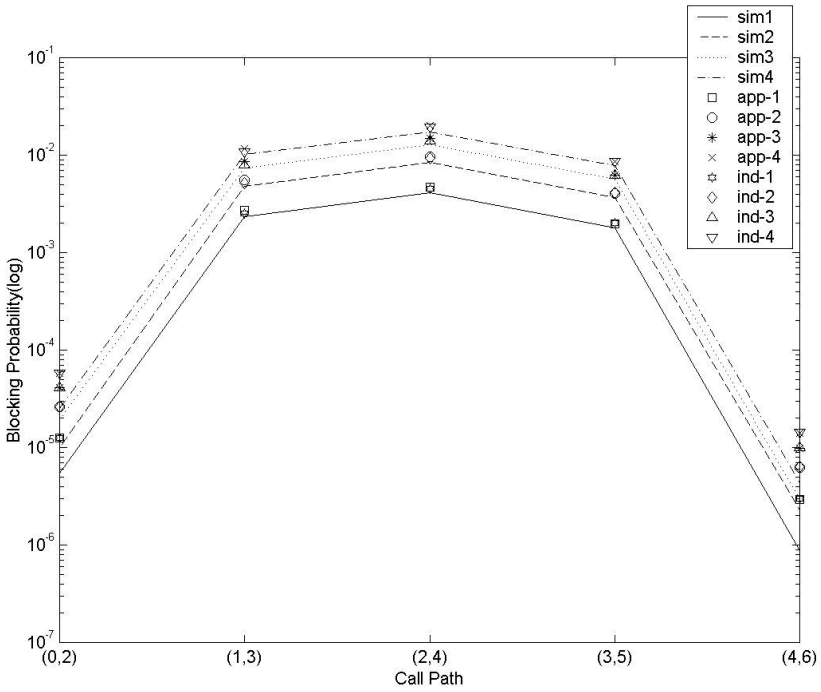


Figure 18b. Two-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=1024 servers.

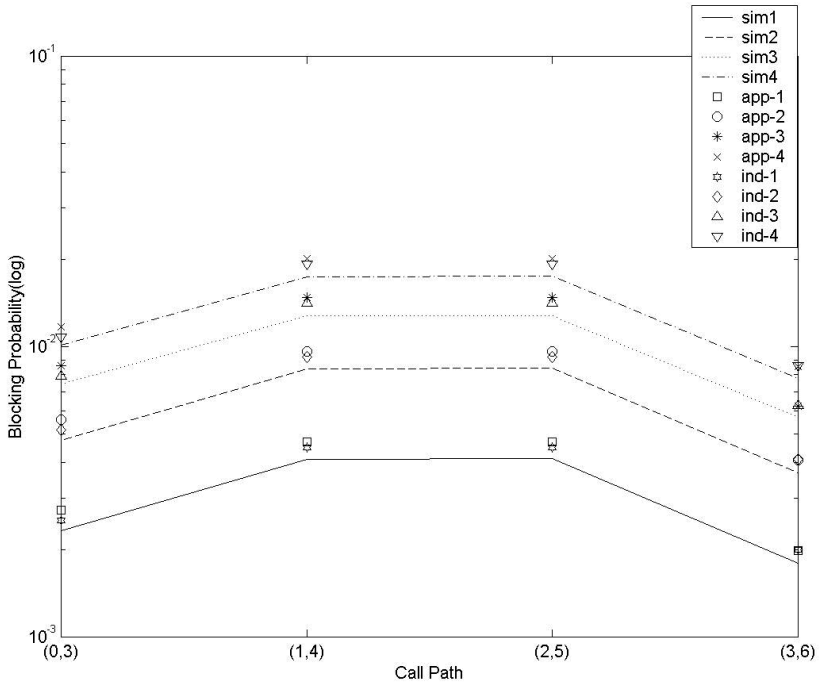


Figure 18c. Three-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=1024 servers.

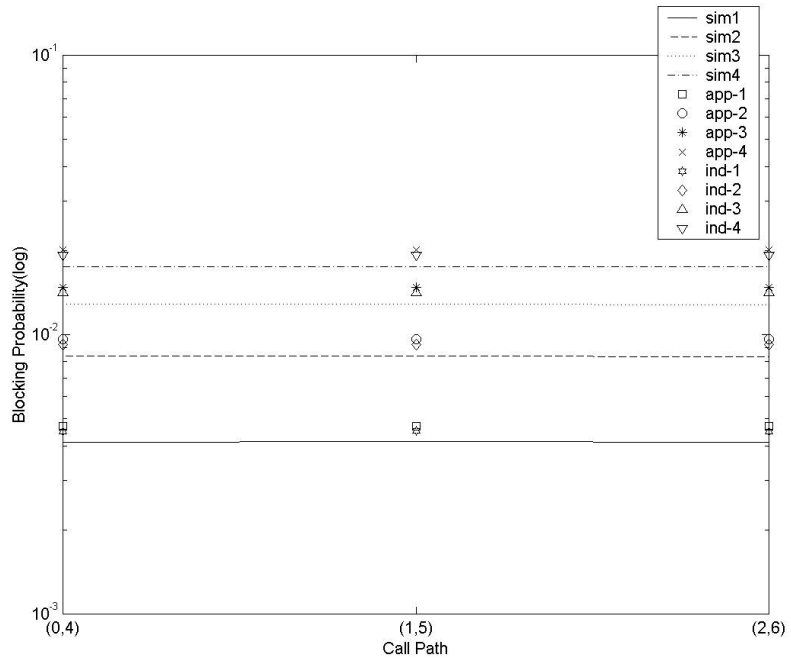


Figure 18d. Four-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=1024 servers.

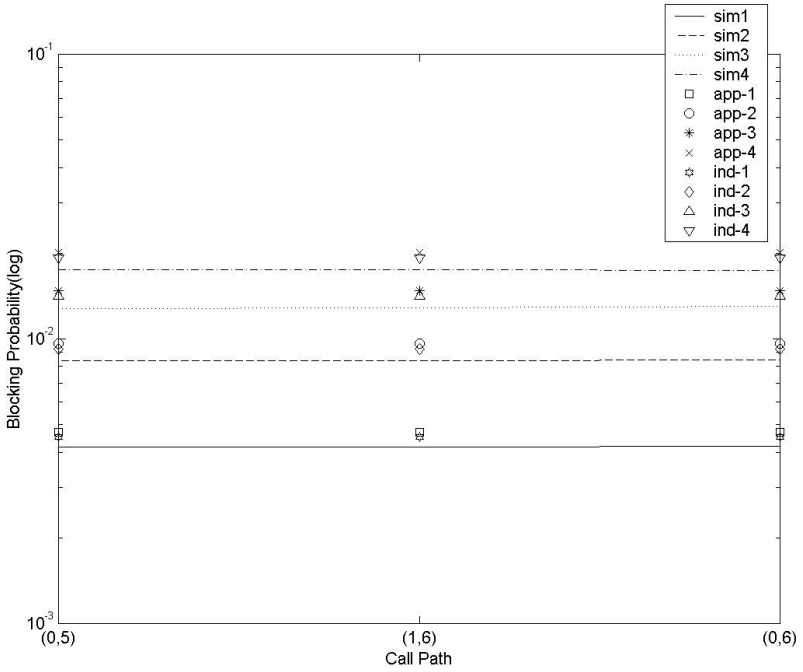


Figure 18e. Five-hop and six-hop paths for class *r*, *r*=1,2,3,4 and *Ws*=1024 servers.

We note that the approximation algorithm is comparable to the link independence algorithm as the number of servers increases. Ultimately, when the number of servers is high, the link independence is a good approximation for the network.