

Towards an IP-centric Optimization: On the Topology Design of WDM-mesh Networks

Jian Wang^{*+}, Rao Vemuri⁺ and Biswanath Mukherjee[⊗]

⁺Department of Applied Science, University of California, Davis, CA 95616, USA

[⊗]Department of Computer Science, University of California, Davis, CA 95616, USA

*Correspondence Author: Tel: +1 530-752-5129, Fax: +1 530-752-4767, Email: jnwang@ucdavis.edu

Abstract: We investigate the topology design in IP backbone networks while exploiting both the advantages of IP routing and optical networking. The emerging WDM and optical switching technologies enable the use of optically switched circuits to interconnect routers, which may not have direct fiber links with each other, in an IP-over-WDM network. All these “light circuits” form the underlying topology of the IP network. In this study, we optimize this WDM-layer topology to minimize the packet-loss rate under dynamic IP traffic. The dimensions of this optimization include finding the right connectivity pattern and allocate limited capacity to each light circuit. Giving the fact that the total interface capacity of a router is bounded by its internal packet-switching speed, we found the problem manifests itself as a trade off between the interface count and the interface speed. Dense connectivity leads to low average interface speed and utilization, vice versa for sparse connectivity. Interestingly, low interface utilization alone do not guarantee low packet-loss rate since higher speed pipes carry traffic more efficiently. Using both analytical and experimental methods, we verified that there are two critical loads for any network. When the network load is higher than the high critical load, the best topology is always the clique. When the network load is lower than the low critical load, the best topology is always the ring. When the network load is between these two extremes, which we believe to be the normal operating point of a backbone network, the best topology has a moderate connectivity density.

1. Introduction

Moor’s law holds that the semiconductor performance doubles every 18 months. A similar law seems to hold for Internet traffic. A recent study shows that the Internet traffic has continued to grow at roughly a uniform rate of 3 folds a year from the early 90s to early 2002 (<http://www.caspiannetworks.com/pressroom/press/>). Driven by this tremendous increase in data traffic, Internet service providers (ISPs) and network operators are forced to investigate backbone network architecture alternatives for high capacity, scalability and cost effectiveness.

IP-over-WDM [1, 2] simplifies the existing backbone architecture by connecting IP routers directly with optically switched circuits. In an IP-over-WDM network, the typical networking elements are IP routers and optical crossconnects (OXC). IP routers are the packet-switching elements that take clients’ (e.g., the routers’ subnets) packets to and from the network; they also forward traffic among other routers on the same network. The OXCs, on the other hand, are used to connect wavelength channels from different fibers to form optical circuits (referred to as “lightpaths”). Two routers that do not have physical connections can directly talk with each other over a lightpath. All the lightpaths in a network form the underlying topology (referred to as the light topology) of the IP network. Rapid development in OXC hardware and

controlling software (e.g., generalized multi-protocol label switching, or GMPLS) will eventually automate the lightpath setup and teardown (provisioning) process. Fast lightpath provisioning presents new opportunities and challenges for building optimized network topology in future IP-over-WDM networks.

A large body of work on virtual topology design already exists in the literature [3-6]. Most of these studies either directly assume circuit traffic, or only consider the first order character (i.e., mean) of packet traffic. The design objective usually is to minimize the load of the most congested link, or, similarly, the average hop distance of IP traffic. As a result of this linear simplification, these studies are always in favor of dense (or even clique, if the optical resource is sufficient) connectivity. A more practical optimization goal would be to minimize the packet loss (possibility on the worst link). For realistic Internet traffic, the packet-loss rate is not a linear function of link load.

A good analogy of this piece of work is designing a road system among cities. Each city has interfaces to a limited number of lanes. If the road system is densely connected, each city is directly connected with most other cities and most roads are narrow. On the contrary, if the road system is sparsely connected, each city is connected to less number of cities (so traffic may be expected to traverse more cities on average); consequently, most roads will be multilane highways. Congestions only happen in cities when traffic from multiple roads competes for a single road, so the entrance of a narrow road is often problematic even though it has lower average utilization. A good design should strike a balance between the road size and reachability. As we will show later, when the overall network load is higher than some critical point, the optimal topology is clique, which is the same as what is given in previous studies. When the network load is below this critical point, however, the connectivity density of the optimal “light topology” could be much lower than that of clique.

This paper is organized as follows: In section 2, we present several concepts regarding IP router architecture and dynamic multiplexing of IP traffic. In section 3, we give a sample network design example to illustrate why an IP-centric network design yields different result than traditional design. In section 4, we outline a problem specification. In section 5 we provide a heuristic approach to solve the topology-searching problem. In section, we present our results and give detailed analysis. In the last section, we discuss future works.

2. Router Architecture and Dynamic Multiplexing of IP Traffic

In this work, we primarily use the packet-loss rate as the gauge for the QoS measurement since the upper layer TCP performance is largely determined by this parameter (although other parameters, e.g., average delay, jitter, can be potential candidates for future study). Packet-loss rate of a network is determined by several factors, including the architecture of IP routers, traffic load of each lightpath, as well as the lightpath capacities. We will describe our network assumptions and how packet-loss rate is determined in this section.

2.1. IP Router Architecture

The architecture of IP router has evolved several generations to what it is today. The general architecture of today’s multigigabit router (MGR) can be described as the following [7].

“The MGR consists of multiple line cards (each supporting one or more network interfaces) and forwarding engine cards, all connected to a high-speed (crossbar) switch as shown in Figure 1. The design places forwarding engines on boards distinct from line cards. When a packet arrives at a line card, its header is removed and passed

through the switch to a forwarding engine. The remainder of the packet remains on the inbound line card. The forwarding engine reads the header to determine how to forward the packet and then updates the header and sends the updated header and its forwarding instructions back to the inbound line card. The inbound line card integrates the new header with the rest of the packet and sends the entire packet to the outbound line card for transmission.”

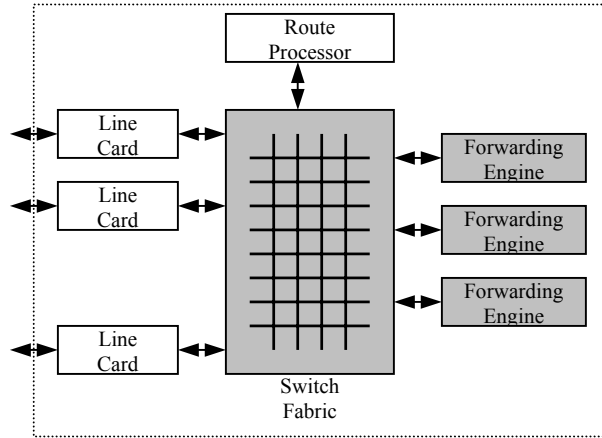


Figure 1. Switch-based router architecture with multiple forwarding engines

For the purpose of our study, we assume that the system is purely output buffered, i.e., upon arrival, all packets are immediately placed in the output buffers. The bandwidth of the switch fabric puts a limit on the aggregated port capacity. The output buffers for different output ports are independent.

We also assume that each node in the backbone network contains a logical router (although in reality, backbone node usually consists of multiple routers, they are interconnected to form a logical router). Some interface ports on a router are used for connecting with clients so they are referred to as edge-side ports. Other ports are used for peer connection with other backbone router so they are referred to as backbone-side ports. In general, the total capacity of edge-side ports is at the same order of magnitude as that of the backbone-side ports. We are interested in the packet loss starting from the point where traffic enters the backbone network, to the point before it leaves the backbone network, i.e., packet loss on buffers of backbone-side ports. For example, in the case of a clique shaped backbone network, packet loss may happen on the buffer of backbone-side ports of an IP router because the aggregated capacity of edge-side ports is much larger than the capacity of any backbone-size port. We do not count the packet loss on edge-side ports because it is independent of the backbone architecture. If we compare two alternatives of backbone architecture, as long as packet loss in the backbone network is small, traffic reached to an edge-side port should be roughly the same for both the networks.

2.2. Burstiness of Internet Traffic and Packet-loss Estimation

Large amount of traffic measurements from various network environment have shown that the Internet traffic is statistically self-similar [8-10]. One of the several equivalent mathematical manifestations of the self-similar property of Internet traffic is its long-range dependency (LRD) [11, 12]. A LRD process is characterized by an autocorrelation function that decays as a power of the lag time, implying that the sum (over all lags) of the autocorrelations diverges. The degree of

LRD is often indicated by the Hurst parameter H . A fractional Brownian Motion (FBM) model was developed by Norros [11] to describe the LRD process as follows:

$$A(t) = mt + \sqrt{am}Z(t) \quad t \in (-\infty, \infty) \quad (1)$$

The process has three parameters m , a and H with the following interpretations: $m > 0$ is the mean input rate; $a > 0$ is a variance coefficient; and $H \in [1/2, 1)$ is the Hurst parameter of $Z(t)$. For more details of this model and why it is an exactly self-similar model, we direct readers to the original papers. The following formula shows the probability that the queue-length X exceeds x , which can be used to estimate the packet-loss rate for buffer size x .

$$P(X > x) \sim \exp\left(-\frac{(C-m)^{2H}}{2\kappa(H)^2 \alpha m} x^{2-2H}\right), \quad \text{where } \kappa(H) = H^H (1-H)^{1-H} \quad (2)$$

In this formula, the explanations for m , a and H are the same as those in formula 1. The C is the capacity of the output link. This formula is shown by Duffield and O'Connell [13] to be asymptotically tight for large x . Comparison with queueing performance of real traffic tracing can be found in [12].

In a strict sense, the above formulation assumes that the input data rate of the queue to be infinity. However, for a real IP router, the ratio of router capacity over the capacity of its port can be smaller than 10. We were curious to know how small the difference between the input and output port data rate can be before equation (2) fails to apply any longer. We conducted queueing simulations and found that as long as the input data rate is equal to or larger than 2 times the output data rate, equation (2) is fairly accurate; however, when input data rate is even closer to the output data rate, the packet-loss rate quickly drops to zero.

2.3. Dynamic Multiplexing Gain

The effective bandwidth of a traffic stream is defined as the service speed (μ) of a single server queue (with buffer size k) that can ensure the overflow probability, when the questioned traffic stream is fed through, equal to some ε . If we assume the maximum delay bound on the buffer is δ , then $k = \mu\delta$. Similarly, if multiple traffic streams are fed through a single server queue, the effective bandwidth of the combined flow is the service speed (μ' , and $k' = \mu'\delta$) of the system that can ensure the packet-loss rate equal to ε .

Dynamic multiplexing gain is the phenomenon that effective bandwidth of the combined traffic stream is smaller than the sum of the effective bandwidths of each individual traffic stream, i.e., $\mu' < \mu_1 + \mu_2 + \dots + \mu_n$. Sometimes, this gain is quantified by the following ratio: $gain = \left(\sum_n \mu_i\right) / \mu'$.

In the case of dynamic multiplexing of heterogeneous self-similar traffic streams, the mean of the aggregated traffic is the sum of the means of the individual streams. Since the traffic streams are independent, the variance of the aggregated

traffic is the sum of the variances of the individual sources; hence the variance coefficient α is a constant. The Hurst parameter H of the combined traffic stream is determined by the largest H of the tributary streams.

There is a common misunderstanding about the dynamic multiplexing of self-similar traffic. The following quotation from a papers [12] published in 1996 clarifies the doubts.

“The interpretation of the Hurst parameter as a measure of burstiness, along with this nondecreasing property, have lead some to conclude that multiplexing does not 1) reduce the burstiness of traffic, and 2) as such multiplexing gains are not feasible with long-range dependent traffic. On the contrary, the FBM model does predict significant multiplexing gains with a large number of independent sources are multiplexed. This is because burstiness is characterized not just by the correlations in the fluctuations parameterized by H , but also by their relative magnitude characterized by $\sqrt{a/m}$.”

The traffic aggregation gain is widely proven both in experiments and theory [14]. Our original motivation for this research work was to bring out the effect of this “local phenomenal” on “network-wide performance”. When we use some small number of hi-speed links in the network instead of a large number of lower-speed links, in effect, we are using the dynamic multiplexing gain to achieve lower packet-loss rates.

How to take the advantage of high-speed connections in IP-over-WDM design may not be obvious in physical implementations, since the interfaces of IP routers are industry standardized and the choices of speed granularities are very limited. Implementation details are given in appendix A. In the following discussion, we assume that the connection speed is a quantumized variable with certain unit capacity.

3. A Simple Network Design Example

In this section, we are going to use a very simple network design example to illustrate how an IP oriented view can actually lead to a solution that is very different from conventional wisdom.

The example is to design a virtual topology for a 5-node IP-over-WDM network so that the total packet-loss rate in the network is minimized. The typical application of such size networks is Metropolitan Area Network (MAN), where the central offices are physically connected as a ring by optical fibers. Suppose each IP router is equipped with four OC-48 line cards for backbone-side connections (and about the same capacity for edge-side connections). With the presence of optical-switching capability, we can connect the IP routers differently from the physical ring topology. In Figure 3, we show two possible logical topologies, which are clique (Fig. 2.1) and ring (Fig. 2.2). In Fig. 2.1, the capacity of each link (could be a lightpath) is OC-48; while in Fig. 2.2, the capacity of each link is equal to OC-96.

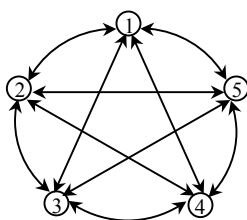


Fig. 2.1 Clique logical topology.

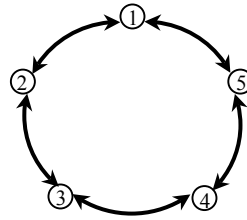


Fig. 2.2 Ring logical topology.

Figure 2. Clique and ring virtual topologies of a 5-node network.

Suppose all the routers use the shortest-hop-distance routing for data forwarding, a packet need only to travel one hop on the clique network to get to their destinations. In the ring network, however, some packets may need to travel two hops to get to their destination. Suppose the traffic load on this network is uniformly distributed, i.e., the mean traffic volume between any node pair is the same, it is trivial to verify that the average hop distance a packet needs to travel on the ring network is 1.5. This average hop distance (denoted as \bar{h}) is an important concept for the following discussions and it is a function of both the network topology and the traffic distribution.

Since the \bar{h} is 1.5 for the ring and 1.0 for the clique networks in this example, the link-load of the ring network is 1.5 times that of the clique networks. For example, if the link load is 20% (which corresponds to an average traffic volume of 0.5 Gbps) on the clique network, then the link load on the ring network should be 30% (which corresponds to an average traffic volume of 1.5 Gbps).

So far, we have figured out the relation of link capacity and link load in these two networks. The numbers, together with the constants H , α , and δ , can be plugged into equation (2) to get the packet-loss rate on each backbone-side port. Notice that directly comparing the packet-loss rates of ports on different networks is partial to the ring network. If the packet-loss rates are the same for all ports in both networks, then the ring network loses 1.5 times more packets than the clique network. What makes sense to compare is the packet-loss rate of a port on the clique network with 1.5 times the packet-loss rate (referred to as the weighted packet-loss rate) of a port on the ring network. Figure 3 shows the comparison of the weighted packet-loss rates in these two networks.

In Fig. 3, the x-axis is the network load, which is defined as the ratio of total offered traffic volume to the total capacity of backbone-side ports. The y-axis in Fig. 3 is the weighted packet-loss rate in logarithmic scale. Since the equation (2) is only asymptotically correct, we use it primarily for relative packet-loss performance study. In Fig. 3, the y-axis is presented without given specific value and unit. Details on the parameters used to get Fig. 3 are shown in table 1.

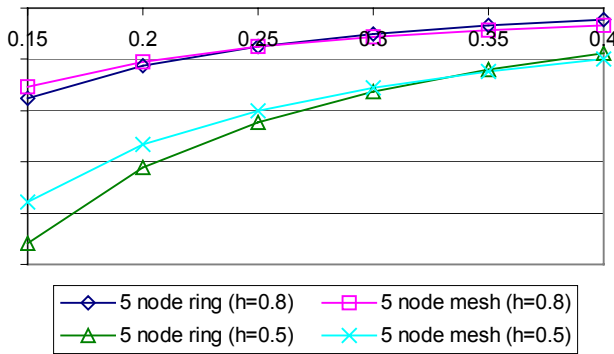


Table 1. Parameters used in getting Figure 4.

	Clique networks	Ring network
C	2.5 Gbps	5 Gbps
$\alpha^{[3]}$	0.0003 Gb·s	0.0003 Gb·s
δ	2 s	2 s

Figure 3. Weighted packet-loss rates in clique and ring networks. The x-axis is the network load; the y-axis is the weighted packet-loss rate in logarithmic scale.

The upper two curves in Fig. 3 are the packet-loss rate for LRD traffic with $H=0.8$. The results show that the ring connectivity actually has lower packet loss than the clique connectivity when the network load is lower than 24% (referred to as the “critical load”); where as the network load is larger than 24%, the clique network has lower packet loss. For traffic

with smaller H , the packet loss decreases dramatically, and the critical load also moves up. When the traffic is pure memory less ($H=0.5$), the turning point is about 34%.

Our network-wide queuing simulation study, which captures the effects such as tandem of packet loss, also confirmed the theoretical analysis results. We build these two networks with $ns-2$ (<http://www.isi.edu/nsnam/ns/>), feed them with LRD traffic with $H=0.8$. The results show that when the network load is 30%, which corresponding to 30% link load on the clique network and 45% link load on the ring network, the total packet loss in the ring network is lower than that in the clique network.

Besides packet-loss behavior, ring and clique are also different in the follow aspects: 1) in the ring virtual topology, routers are point to point connected so the WDM layer can be simplified; 2) in the ring virtual topology, wavelength resource can be saved i.e., at least 3 wavelengths are needed to deploy the clique connectivity on a physical ring; while, 2 wavelengths are needed for the ring virtual topology. 3) A clique has lower average packet delay than ring.

It is important to note that we are not advocating ring topology for general backbone network. When network size is large, the choices are no longer between ring and clique. The best virtual network normally has an average node degree between 2 (the ring case) and $N - 1$ (clique where the N is the network size).

4. Network Design Problem Formal Definition

The virtual topology design problem can be formally described as follows: given a certain number (N) of IP routers and the traffic statistics among them, and also an unlimited optical transmission capability; the goal is to find a virtual topology so that the highest packet-loss rate among all lightpaths, which will be referred to as the network's packet-loss rate bound, is minimized when all the traffic is carried. The reason we choose to minimize the packet-loss rate bound instead of the overall packet-loss is that this objective better reflects QoS guarantee to clients, while at the same time gives a good estimation on the network's overall packet-loss.

The total interface capacity a router uses for peer connection is limited. This capacity constraint is represented by the vector: $Cap = [cap_i]$, where $i = 0, 1, \dots, N - 1$. Note that, in a realistic scenario, the interface capacity of a node cannot be arbitrarily divided, i.e., they are the sum of some discrete value, e.g., OC-12, OC-48, etc.

The traffic demand is represented as the following traffic matrix: $T = \begin{bmatrix} 0 & \dots & t_{ij} & \dots \\ & 0 & \dots & \\ & & \dots & \\ & & & 0 \end{bmatrix}$, where $t_{ij} = t_{ij}(m, \alpha, H)$

is a stochastic process that models the traffic from node i to j . In this model, m is the mean traffic volume between node i and j . Since α and H are the property of IP traffic, they are not so heavily depending on the values of i and j as m does.

What we want to find out is a capacity matrix which can be represented as: $C = \begin{bmatrix} 0 & \dots & C_{ij} & \dots \\ & 0 & \dots & \\ & & \dots & \\ & & & 0 \end{bmatrix}$, where C_{ij} is

the lightpath capacity from node i to j . If there is no connection from node i to j , then C_{ij} is 0.

The problem is formulated as follows:

Objective:	$\text{minimize}(\text{Max}(\varepsilon_{ij}(L_{ij}, C_{ij}, \delta_{ij})))$	
Subject to:	$\sum_j C_{ij} \leq \text{cap}_i$	Router capacity constraint
	$\sum_i C_{ij} \leq \text{cap}_j$	
	$L_{ij} = L_{ij}(m, \sigma, H) = \sum_{(ij) \in (sd)} t_{sd}(m, \sigma, H)$	Flow conservation constraint

In the above formulation, the packet-loss rate on lightpath C_{ij} is calculated from $\varepsilon_{ij}(L_{ij}, C_{ij}, \delta_{ij}) \approx P_{ij}(X > x)$, which is the equation (2) mentioned in section 2.2. The $L_{ij}(m, \sigma, H)$ is the traffic load on lightpath C_{ij} and it is determined by both the network topology and IP routing algorithm (shortest path is usually used in real IP routers). The δ_{ij} is the queue delay bound. The flow conservation constraint is applicable when the packet-loss rate is small on all lightpaths. When multiple traffic streams are dynamically multiplexed onto a single lightpath, we denoted the aggregated stream as a summation of all its tributary streams. The summation of two heterogeneous traffic flows is calculated by the following formula:

$$t(m_1, \alpha_1, H_1) + t(m_2, \alpha_2, H_2) = t(m_1 + m_2, (\alpha_1 m_1 + \alpha_2 m_2) / (m_1 + m_2), \max(H_1, H_2)) \quad (3)$$

There are two differences between this problem statement and previous studies [3, 4]. First, a non-linear objective function is used so this is no longer a linear programming problem. Second, previous studies often assumed that the virtual topology design and the traffic routing on virtual topology could be optimized simultaneously; however, in reality, we have no control over the upper layer routing mechanisms during topology design stage. IP forwarding can use any strategy; in particular, we picked shortest-hop-distance routing in the following discussion.

5. Heuristic Solution

5.1. Basic Hunches

A solution to this optimization problem can be divided into the following three steps: 1) randomly pick a valid connectivity pattern among the N nodes, 2) route traffic demands on the selected connectivity pattern, and 3) allocate capability to lightpaths according to the IP routers' capacity constraint so that the packet-loss rate bound is minimized on this topology. These three steps are repeated, and a new topology is tried in each iteration, until a satisfactory solution is

reached. The problem of enumerating all connectivity patterns in a N -node network is NP-hard, so the topology design problem is also NP-hard.

One special case of this problem is to have uniformly distributed traffic demand on the network. We suspected that the best virtual topology should also have certain symmetric property, which may be useful for accelerating the searching procedure. Unfortunately, our effort at searching for such a property turned out to be unsuccessful. In Appendix B, we consider a 7-node network example. We show that an asymmetric topology actually gives better packet-loss performance than any symmetric solutions under uniformly distributed traffic demands. Interestingly enough, this is also a counter-example that shows minimizing average hop distance is not equivalent to minimizing the load of the most congested link.

Recall the 5-node example given in section 3. When reducing the number of lightpaths on the virtual topology, we observe two effects. The good effect is that we can (logically) increase the buffer size and bandwidth of the remaining lightpaths so they can be used more efficiently; the bad effect is that the network traffic is forced to go through more hops on the average, so the average load of lightpaths is increased. Intuitively, our goal is similar to minimizing the average hop distance \bar{h} in some sparsely connected network (i.e., the number of lightpaths in the network is limited). Toward this direction, some well known results can help us to establish a feeling on how our solution should be. Stated in [16], it is possible to build a 128-node bidirectional shufflenet so that the average hop count for a connection is 2.25. The uniformed node degree in this network is 16. Compared with clique, the lightpath capacity in this bidirectional shufflenet is about 7 times fatter. Readers of this paper can try to mimic the way Figure 4 is drawn to find out what the critical network load is (about 30% for LRD traffic with $H=0.8$). Unfortunately, it is difficult to use these special topologies directly in virtual topology design for the following reasons: 1) they usually require the network to possess certain properties, e.g., bidirectional shuffle network has special requirements on number of nodes; and 2) they are not guaranteed to give a good solution. In the next section, we are going to give a heuristic algorithm for general solutions.

5.2. Genetic Algorithm Based Heuristic

In the previous sub-section, we break the virtual topology design problem into three steps. To exhaust the searching space of step 1 is NP hard, however, steps 2 and 3 are not that expensive. In step 2, we simply route all the packet traffic on the topology (suppose shortest path routing and traffic bifurcation) found in step 1. In step 3, we use a greedy algorithm to allocate capacity on the virtual topology. Since the traffic on each lightpath is determined in step 2, so we simply repeatedly increase the capacity on the lightpaths that have the highest packet-loss rate. A greedy algorithm can find the capacity distribution that gives the lowest packet-loss rate bound.

An obvious heuristic for step 1 is pure random search. We developed a simple random-topology generator to generate bi-connected connectivity patterns with given lightpath counts. Since the packet-loss information cannot be fed back to the topology generation process, this approach is not very efficient.

We improved the pure random heuristic by using the genetic algorithm. In this approach, we randomly generate some virtual topologies and use them as the first generation of population. Then we calculate the networks' packet-loss bounds. Giving each virtual topology an opportunity proportional to the inverse of its packet-loss bound, we randomly pick virtual topologies, two at a time, to do a so-called "crossover". In the crossover procedure, we simply exchange part of the

lightpaths of these two networks. A “mutation” procedure is also defined as replacing an individual virtual topology with a newly generated virtual topology. After certain numbers of crossovers and mutations, we have a new generation that is the same size of the old generation. Then, we use the new generation to generation the third generation, and so on. The best solution through out the generations is recorded as the final result.

Once played around with this optimization problem for a while, we start to develop a feeling on how many lightpaths are needed for a certain size network. For example, we know that about 250 lightpaths are needed for a 50-node network to achieve best packet loss performance when network load is 20%. If we restrict the search space to, say, 200~300 lightpaths, then the searching speed can be increased (or the solution’s quality can be improved if we choose to keep the computational time unchanged). So we developed an alternative version of this genetic algorithm, which searches the best virtual topology with given lightpath count. In this version, all individuals in the first generation are having the same desired lightpath count (which can be specified in our random topology generator). We specially designed a “two-point crossover” procedure that can guarantee the offspring to have the same lightpath count as their parents. The procedure is as follows: 1) in our topology representation (which is actually a matrix, although other representations are also possible), we randomly pick a “crossover starting point” and a “crossover length” (which tells us how many lightpaths will be moved from one parent to another); 2) start from the crossover starting point, the first “crossover length” number of lightpaths that exit in the original “father topology” but not in the original “mother topology” are moved to the “mother topology”; 3) repeat step 2 but switch the role of “father topology” and “mother topology”.

6. Numerical Results

6.1. Packet-loss Rate as a Function of Network Topology

In this section, we show how the packet-loss bound changes with a network’s lightpath count (Figure 4). The results are obtained by using our genetic algorithm, and they are compared with the pure random search results. The network is assumed to be the following. Each IP router has aggregated capacity of 320 Gbps, and half of it (160 Gbps) is used for peer communication. We can have both OC-192 and OC-48 ports on a router. The traffic on the network is uniformly distributed and H is assumed to be 0.8 for all traffic flows. Notice that our heuristic is designed and implemented for arbitrary traffic pattern. The uniform traffic distribution assumption is just to make the results easy to understand and compare.

We tested our network design algorithms under network loads of 20% and 30%. In Fig. 4.1, we show the packet-loss bound vs. average node degree. The constants used in this computation are the same as those shown in Table 2. Note that average node degree is directly proportional to the total number of lightpaths in the network since each bidirectional lightpath consumes exactly two degrees. Also notice that equation (2), which we used to obtain the curves, does not directly give the exact values of packet-loss rate. The values labeled on the y-axis of Fig. 4.1 are estimated by comparing theoretical results with queueing simulation results, and the procedures are described as follows. When the network load is 30% and average node degree is 7 (the left most point on the curves), we take the solution given by our genetic algorithm and observe the load (which is actually close to 100%) on the link that has the worst packet loss rate. We then did queueing simulation to estimate the packet-loss rate at the observed load. Then we normalized all other values accordingly.

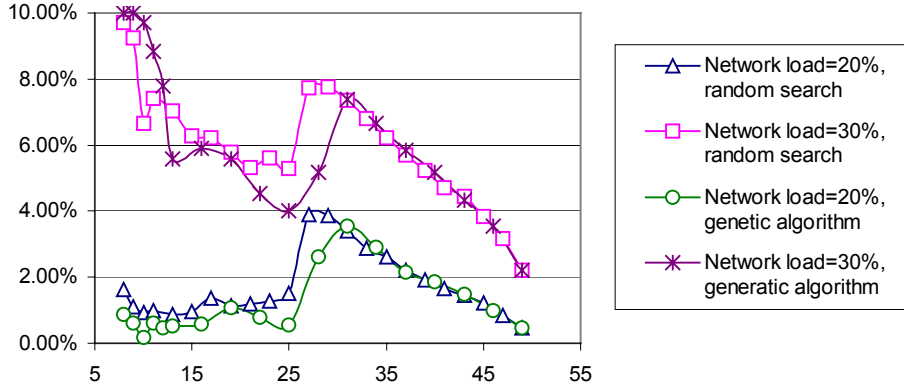


Figure 4.1. Packet-loss bound vs. average node degree in the 50-node IP-over-WDM network. The x-axis is the average node degree, the y-axis is the packet-loss bound.

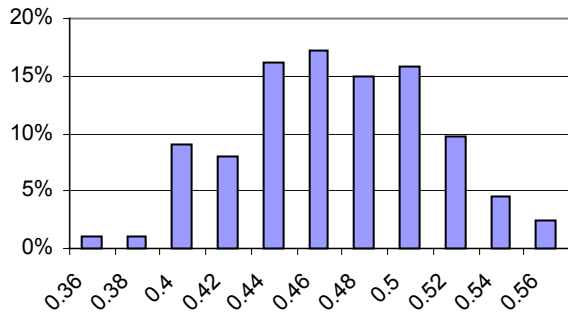


Figure 4.2. Link load distributions for the best “known topology” when network load is 20%. The x-axis is the link load, the y-axis is the density.

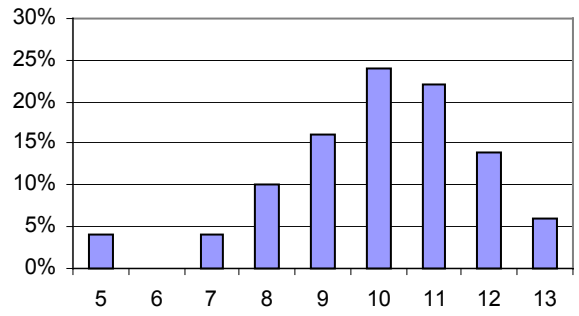


Figure 4.3. Node degree distributions for the best “known topology” when network load is 20%. The x-axis is the node degree, the y-axis is the density.

Figure 4. Searching for the best topology for a 50-node network. When the network load is 20%, we found the best topology to have an average node degree of 10.

When the network load is 20%, we see from Fig. 4.1 that the best topology has an average node degree of 10, which corresponds to 250 lightpaths. With the increase of network load, the entire packet-loss curve moves up, and the low average-node degree end increases faster. When the network load is 30%, the best topology we can find is the clique with the packet-loss bound equal to 2%.

To understand the best topology given by our heuristic algorithm when network load is 20%, we show the link load and node degree distribution of this network in Fig. 4.2 and 4.3. While the average link load is about 47%, there are 2.4% of the links have load as high as 58%. The node degree spreads from 5 to 13, with the average being 10.

The curves in Fig 4.1 look bumpy because of the discrete nature of the lightpath capacity. The packet-loss bound is the packet-loss rate on the bottleneck lightpath. If we randomly pick a curve (say 20% load curve) and a node-degree window around a dip (say around the average node degree of 25), then we examine each point in the window from left to right. At the beginning, we see the topologies with some under utilized connections. When we increase the average node degree and re-optimize the topology, the added connections help to diversify the traffic so the packet-loss rate on the bottleneck connection is reduced. At the bottom of the dip is the point where the traffic is fairly balanced among most connections, so there is not much bandwidth redundancy on the non-bottleneck connections. If we continue to increase the average node degree, some links will have to reduce its capacity by one unit and lead to dramatic increase in the packet-loss rate. In our

experiment, when the average node degree is 25 in this 50-node network, we observed that the lowest lightpath capacity is 5 Gbps in our solutions. While if the average node increase from 25, we start to see lightpaths with capacity of only 2.5 Gbps, which become the bottlenecks in the network and draw the packet-loss bound up. One very important conclusion we can draw from the result is: very dense mesh connectivity could lead to very bad network performance since it creates low capacity lightpaths just like clique, but lack the ability of spread traffic evenly on lightpaths.

6.2. What is the Desired Node Degree?

If the best topology is not a clique for practical IP-over-WDM networks, then what should it be? Our heuristic is designed exactly to answer this question on a case-by-case basis; however, we are keen for a more intuitive and comprehensive answer. Unfortunately, we have not been able to reach this understanding yet. The following results are the outcome of our preliminary attempt.

We run our genetic algorithm based heuristic for different networks with size ranging from 30 to 200. For each network, we give each router the capacity of $1.4 \times \text{Network_size} \times 2.5 \text{ Gbps}$ for peer communication. The network load for each network is 20% and traffic is uniformly distributed in the network. How the average node degree change with respect to the network size is shown in Figure 5. Since our algorithm is heuristic, this curve is not very smooth. We performed a simple curve fitting and results a logarithmic function.

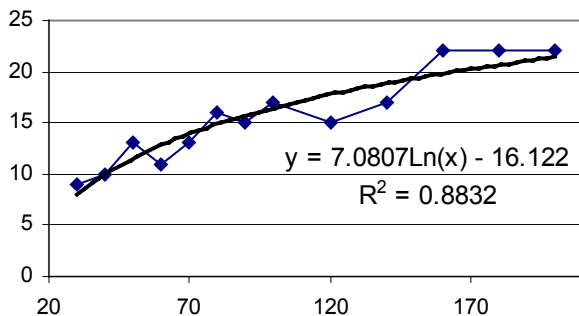


Figure 5. Best average node degree for different size networks.

7. Implications and Possible Future Work

In this work, we studied the virtual-topology-design problem in IP-over-WDM networks from an IP-centric perspective. We show that dense and clique connectivities may NOT give the best packet-loss performance under normal network operation conditions. Since sparse connectivity also has potential advantage on cost (both equipment and management) and scalability when compared with dense connected virtual topology, it could become the desired operation model for future IP-over-WDM network.

Our genetic algorithm based heuristic has achieved good results so far, however, there is still room for improvement. With the introduction of new technologies and services, such as MPLS and voice-over-IP, more study may need to be done on how these may affect traffic dynamic multiplexing. One of the reasons that may hinder the use of sparse connectivity in real network deployment is the complexity of the optimization process. We are particularly interested in seeing how big the difference is between the “optimally designed” and the “naturally grown” sparse networks.

Appendix A:

The traffic aggregation gain can affect network architecture. However, it is not straightforward to take advantage of this in real network implementations. The interfaces of IP routers are industry standardized and the choices of speed are very limited. For example, in a typical backbone IP router today, one can choose interfaces among OC-192 (10Gbps), OC-48 (2.5Gbps), Gigabit Ethernet (1Gbps), etc. If you want to have a 7.2 Gbps pipe, the closest one you can get is 7.5 Gbps, which is constructed from 3 OC-48 pipes. Questions arise when we use multiple parallel pipes to emulate one fat pipe. A fat pipe is a high-speed port that has the two properties: 1) its capacity is equal to the sum of all low-speed ports, and 2) its buffer size is equal to the sum of the buffer size of all low-speed ports. Since the queues of the interfaces are not physically shared, we need to verify whether the queueing performance is close to a fat pipe or not.

The answer to the above doubt depends on how the traffic balancing is done. Traffic balancing is done by the switching fabric under the control of software policy. Typically, there are two policies implemented in real IP routers: per-packet balancing and per-destination balancing. In per-packet balancing, packets are thrown to queues of equal-cost paths sequentially; in per-destination balancing, packets with the same destination address are always sent to the same path. We compared the queueing performance of these two mechanisms with the fat pipe model. Figure 2 shows one set of our result. In these experiments, we feed synthesized LRD traffic through three queueing systems. We used the traffic generator reported in [15] with some minor improvements, e.g., use the packet size distribution of IP backbone networks (<http://www.nlanr.net/NA/Learn/packetsizes.html>), etc., so that the results can be easily compared with existing Internet traffic trace. The parameters used to perform these experiments are given in table 1.

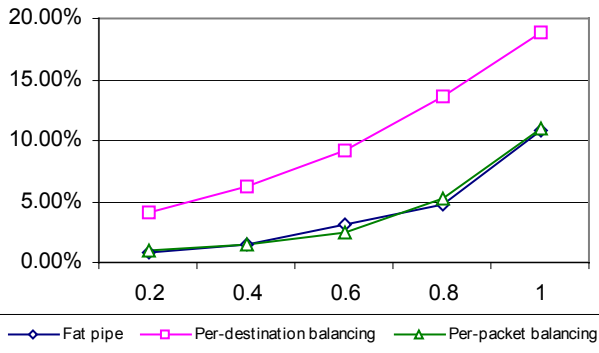


Figure A1. Comparison of two traffic balancing policy with fat-pipe model on queueing performance. The x-axis is the link load and the y-axis is the packet loss rate.

Table 1. Parameters used in getting Figure 1.

	Fat pipe	Per-packet & per-destination balancing
Source data rate	10^7 bps	10^7 bps
Link capacity	5×10^6 bps	2.5×10^6 bps
Buffer size per link	2×10^4 bytes	1×10^4 bytes
Num. of links	1	2

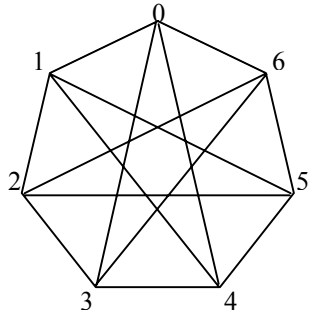
From figure A1 we can see that there is negligible difference on packet-loss rate between the per-packet balanced multiple pipes and the fat pipe model. The packet-loss rates from both of these models are significantly lower than that of per-destination balancing, assuming the destination addresses are random. We have done more experiments, and the results show that the correctness of this conclusion is independent of the Hurst parameter (the same conclusion is also true for memory-less traffic), link capacity and buffer size (unless the buffer size is close to the size of packet size).

In IP-over-WDM network, equal-cost paths can be implemented with multiple equal-length lightpaths. To use per-packet traffic balancing is not likely to cause problems such as packet re-ordering, so we believe it is safe to assume that per-packet traffic balancing feature of IP router can be used to emulate faster links in IP backbone networks.

Appendix B:

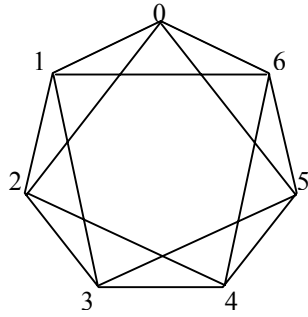
Here is an example shows that seemingly balanced topology may not lead to balanced link load, even with the presents of uniform traffic. Given 7 nodes and uniformly distributed traffic request (say, one unit) among them, we are asked to design a 14-link network so that the traffic load on the most congested link is minimized. Additional assumptions are: all the links have the same capacity, shortest hop-distance routing is used and traffic is bifurcated.

Intuitively, the best solution should be a symmetric graph, such as Figure A1.1 or A1.2; however, the correct answer is shown in figure A1.3. Traffic load on each link are given in the corresponding link-load matrices.



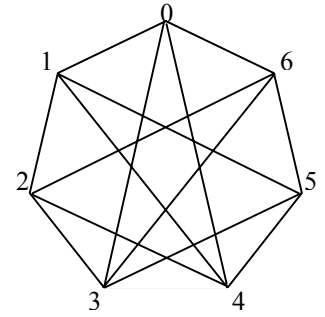
$$\begin{bmatrix} 0 & 2\frac{1}{3} & 0 & 1\frac{2}{3} & 1\frac{2}{3} & 0 & 2\frac{1}{3} \\ 2\frac{1}{3} & 0 & 2\frac{1}{3} & 0 & 1\frac{2}{3} & 1\frac{2}{3} & 0 \\ 0 & 2\frac{1}{3} & 0 & 2\frac{1}{3} & 0 & 1\frac{2}{3} & 1\frac{2}{3} \\ 1\frac{2}{3} & 0 & 2\frac{1}{3} & 0 & 2\frac{1}{3} & 0 & 1\frac{2}{3} \\ 1\frac{2}{3} & 1\frac{2}{3} & 0 & 2\frac{1}{3} & 0 & 2\frac{1}{3} & 0 \\ 0 & 1\frac{2}{3} & 1\frac{2}{3} & 0 & 2\frac{1}{3} & 0 & 2\frac{1}{3} \\ 2\frac{1}{3} & 0 & 1\frac{2}{3} & 1\frac{2}{3} & 0 & 2\frac{1}{3} & 0 \end{bmatrix}$$

Fig. A1.1



$$\begin{bmatrix} 0 & 1\frac{2}{3} & 2\frac{1}{3} & 0 & 0 & 2\frac{1}{3} & 1\frac{2}{3} \\ 1\frac{2}{3} & 0 & 1\frac{2}{3} & 2\frac{1}{3} & 0 & 0 & 2\frac{1}{3} \\ 2\frac{1}{3} & 1\frac{2}{3} & 0 & 1\frac{2}{3} & 2\frac{1}{3} & 0 & 0 \\ 0 & 2\frac{1}{3} & 1\frac{2}{3} & 0 & 1\frac{2}{3} & 2\frac{1}{3} & 0 \\ 0 & 0 & 2\frac{1}{3} & 1\frac{2}{3} & 0 & 1\frac{2}{3} & 2\frac{1}{3} \\ 2\frac{1}{3} & 0 & 0 & 2\frac{1}{3} & 1\frac{2}{3} & 0 & 1\frac{2}{3} \\ 1\frac{2}{3} & 2\frac{1}{3} & 0 & 0 & 2\frac{1}{3} & 1\frac{2}{3} & 0 \end{bmatrix}$$

Fig. A1.2



$$\begin{bmatrix} 0 & 2\frac{1}{6} & 0 & 2\frac{1}{6} & 2\frac{1}{6} & 0 & 2\frac{1}{6} \\ 2\frac{1}{6} & 0 & 2\frac{1}{3} & 0 & 1 & 2\frac{1}{6} & 0 \\ 0 & 2\frac{1}{6} & 0 & 2\frac{1}{6} & 2\frac{1}{6} & 2\frac{1}{6} & 0 \\ 2\frac{1}{6} & 0 & 2\frac{1}{6} & 0 & 0 & 2\frac{1}{6} & 1 \\ 2\frac{1}{6} & 1 & 2\frac{1}{6} & 0 & 0 & 2\frac{1}{6} & 0 \\ 0 & 2\frac{1}{6} & 0 & 2\frac{1}{6} & 2\frac{1}{6} & 0 & 2\frac{1}{6} \\ 2\frac{1}{6} & 0 & 2\frac{1}{6} & 1 & 0 & 2\frac{1}{6} & 0 \end{bmatrix}$$

Fig. A1.3

Figure A1. Three virtual topologies and their link-load matrices. The three topologies are having the same link count of 14. Fig.A1.3. is not symmetric but gives best load balancing among all links.

The representation of the link-load matrix is straightforward. In Fig. A1.1, the load on the link (0,1) is $2\frac{1}{3}$ because it consists of the following streams: $T_{0 \rightarrow 1} + \frac{1}{3} \cdot T_{0 \rightarrow 2} + \frac{1}{3} \cdot T_{0 \rightarrow 5} + \frac{1}{3} \cdot T_{6 \rightarrow 1} + \frac{1}{3} \cdot T_{3 \rightarrow 1} = 2\frac{1}{3}$ (units). In Fig. A1.3, traffic is better balanced among links. For example, there are four equal length routes from node 0 to node 2, which does not

happened in Fig. A1.1. So the traffic on link (0,1) comes from the following contributors:

$$T_{0 \rightarrow 1} + \frac{1}{4} \cdot T_{0 \rightarrow 2} + \frac{1}{4} \cdot T_{0 \rightarrow 5} + \frac{1}{3} \cdot T_{6 \rightarrow 1} + \frac{1}{3} \cdot T_{3 \rightarrow 1} = 2 \frac{1}{6} \text{ (units).}$$

The summations of all link loads on the three networks are the same. The average hop-distances of the three networks are also the same, which equal to $1 \frac{1}{3}$ units. The lower bound for maximum link load is 2, however, we have done exhaustive search to show that it is not possible to construct a connectivity pattern to achieve this lower bound.

Acknowledgement: We gratefully thank Glen Kramer for providing the LRD traffic synthesizing code for our experiment, Dr. Fei Xue for providing the code to verify the Hurst parameter of synthesized LRD traffic. We also gratefully thank for all the researchers at the Networks research lab of UC Davis, especially, Narendra Singhal, Keyao Zhu and Aysegul Gencata for their insightful advice during the research. Without all these helps, this research will be impossible.

References:

- [1] B. Mukherjee, "WDM optical communication networks: progress and challenges," *IEEE J. Select. Areas Commun.* vol 18, no. 10 , pp 1805 -1809, Oct. 2000.
- [2] S. Baroni, J. O. Eaves, M. Kumar, M. A. Qureshi, A. Rodriguez-Moral, and D. Sugeran, "Analysis and design of backbone architecture alternatives for IP optical networking," *IEEE J. Select. Areas Commun.* vol 18, no. 10 , pp 1980 - 1994, Oct. 2000.
- [3] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," *IEEE/ACM Trans. Networking*, vol.4, no.5, pp 684-696, Oct. 1996.
- [4] R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed optical networks," *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, pp 840-851, 1996.
- [5] Z. Zhang, and A. Acampora, "A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength Routing and Wavelength Reuse", *IEEE/ACM Trans. Networking*, vol. 3, no. 3, pp. 281-288, June 1995.
- [6] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: an approach to high bandwidth optical WAN's," *IEEE/ACM Trans. Networking*, vol. 40, no. 7, pp. 1171-1182, July 1992.
- [7] J. Aweya, "IP router architectures: an overview," *International J. Commun. Systems*, vol.14, no.5, pp.447-75, Wiley, June 2001.
- [8] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol.2, no.1, pp1-15, Feb. 1994.
- [9] V. Paxson, and S. Floyd, "Wide-area traffic: The failure of Poisson modeling," *IEEE/ACM Trans. Networking*, vol.3, no.3, pp.226-244, June 1995.
- [10] W. Willinger, M. S. Taqqu, R. Sherman, D. V. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level," *IEEE/ACM Trans. Networking*, vol.5, no.1, pp. 71-86, Feb. 1997.
- [11] I. Norros, "On the use of fractional Brownian motion in the theory of connectionless networks," *IEEE J. Select. Areas Commun.*, vol.13, no.6, pp. 953-962, Aug. 1995.

- [12] A. Erramilli, O. Narayan, and W. Willinger, "Experimental queueing analysis with long-range dependent packettraffic," *IEEE/ACM Trans. Networking*, vol.4, no.2, pp 209-223, April 1996.
- [13] N.G. Duffield, and N. Connell, "Large deviations and overflow probabilities for the general single-server queue, with applications," *Math. Proc. Cam. Phil. Soc.*, vol 118, pp. 363-374, 1995.
- [14] N. L. S. Fonseca, C. A. V. Neto, and G. S. Mayor, "Statistical multiplexing of self-similar sources," in *Proc. Globecom '00*, vol.3, pp 1788-1792, San Francisco, Nov. 2000.
- [15] Glen's paper?
- [16] M. Gerla, E. Leonardi, F. Neri, and P. Palnati, "Routing in the bidirectional shufflenet," *IEEE/ACM Trans. Networking*, vol.9, no.1, pp. 91-103, IEEE; ACM, Feb. 2001.
- [17] M. Corradi, R. G. Garroppo, S. Giordano, and M. Pagano, "Analysis of f-ARIMA processes in the modelling of broadband traffic," *Proc. of IEEE ICC 2001*, Helsinki, Finland, June 11-14, 2001.
- [18] H. Ahn, J-K Kim, S. Chong, B. Kim, and B. D. Choi, "A video traffic model based on the shifting-level process: the effects of SRD and LRD on queueing behavior," *Proc. IEEE INFOCOM 2000*, vol. 2, p.1036-45, pp 1036-1045, Tel Aviv, Israel, 26-30 March 2000.
- [19] R. G. Addie, T. D. Neame, and M. Zukerman, "Modeling superposition of many sources generating self similar traffic," *IEEE ICC'99*, vol.1, pp. 387-391Piscataway, NJ, USA: 1999.