Controlled Use of Excess Backbone Bandwidth for Providing New Services in IP-Over-WDM Networks

Antonio Nucci, *Member, IEEE*, Nina Taft, *Member, IEEE*, Chadi Barakat, and Patrick Thiran, *Member, IEEE*

***Abstract*—We study an approach to quality-of-service (QoS) that offers end-users the choice between two service classes defined according to their level of transmission protection. The fully protected (FP) class offers end-users a guarantee of survivability in the case of a single-link failure; all FP traffic is protected using a 1:1 protection scheme at the wavelength-division multiplexing (WDM) layer. The best effort protected (BEP) class is not protected; instead restoration at the IP layer is provided. The FP service class mimics what Internet users receive today. The BEP traffic is designed to run over the large amounts of unused bandwidth that exist in today's Internet. The goal is to increase the load carried on backbone networks without reducing the QoS received by existing customers. To support two such services, we have to solve two problems: the off-line problem of mapping logical links to pairs of disjoint fiber paths, and an on-line scheduling problem for differentiating packets from two classes at the IP layer. We provide an algorithm based on a *Tabu Search* meta-heuristic to solve the mapping problem, and a simple but efficient scheduler based on *weighted fair queueing* for service differentiation at the IP layer. We consider numerous requirements that carriers face and illustrate the tradeoffs they induce. We demonstrate that we can successfully increase the total network load by a factor between three and ten and still meet all the carrier requirements.**

***Index Terms*—Internet, provisioning, scheduling, services, Tabu Search (TS), wavelength-division multiplexing (WDM).**

I. Introduction

THE Internet backbone contains a large amount of capacity that is currently not being used. Carriers today are very interested in carrying additional load on their networks in order to generate additional revenue, however, they are concerned about not reducing the quality-of-service (QoS) received by existing customers. The three main reasons why the Internet contains unused capacity are because of *equipment redundancy*, *overprovisioning*, and *the link upgrade process*. Equipment redundancy typically leads to a multiplicity of links and/or nodes. Overprovisioning usually implies that network links are run at low utilization levels. Redundancy of equipment and overprovisioning are used to protect the backbone against failures. Some recent research studies have started to uncover the nature and extent of failures in today's Internet protocol (IP) backbones [2], [3]; these findings have revealed that failures of one type or another occur almost on a daily basis [2], and roughly 12% of failures are related to optical-layer failures [3]. With technologies such as wavelength-division multiplexing (WDM), a single-fiber failure can bring down a large number of IP paths. Most large carrier networks today use highly meshed topologies to prevent network partitioning in the event of widespread failures involving multiple links.

Upgrading the links (e.g., converting an OC-48 to an OC-192 link) in a large backbone is a time consuming process. For example, upgrading a single large inter-POP backbone link can take a few months, while upgrading a sizeable portion of the entire network can take over a year. Each time a link is upgraded, a "pocket" of additional bandwidth is opened up, but this is not really available to users because: 1) for some users the shortest paths they use may not traverse the new fast link; 2) for other users the sequence of links their packets follow may traverse the new link, but the other links in the sequence will be older slower ones and these slower links determine the end-to-end throughput; and 3) if the upgraded link is in the main working path, it may not be used if the backup path has not also been upgraded at the same time. When a network is partially upgraded, and has many pockets of bandwidth scattered over the topology, a potentially large number of users could indeed profit from this new capacity, if it is properly managed.

In this paper, we propose that carriers provide two classes of service, one of which would mimic today's service and a second one that would provide a lower QoS. The idea is for the lower grade service to be carried on the "excess" bandwidth in the backbone in such a way that has no impact on the service level agreements (SLAs) promised to the higher grade service. The majority of time this excess bandwidth is unused, hence, the lower grade service will experience good performance and can support a good SLA. When this excess bandwidth becomes needed in a failure scenario, we drop as many packets as necessary from the lower grade service in order to ensure there is enough bandwidth to protect the higher grade service.

In order to achieve this, the two classes of service should be differentiated by their *level of protection* against failures and the packets need to be marked according to their class of service. The first class, called fully protected (FP), offers users the insurance that they will not suffer service interruption in the case

Manuscript received February 20, 2004. The work of N. Taft was done while with Sprint Advanced Technology Laboratories, Burlingame, CA. The work of C. Barakat was done while with the Swiss Federal Institute of Technology at Lausanne (EPFL), Lausanne, Switzerland. The work of P. Thiran was done while at EPFL and was supported in part by a grant from Sprint and in part by the Hasler Foundation, Bern, Switzerland under Grant DICS 1830.

A. Nucci is with Sprint Advanced Technology Laboratories, Burlingame, CA 94010 USA (e-mail: anucci@sprintlabs.com).

N. Taft is with Intel Research Berkeley, Berkeley, CA 94704 USA (e-mail: nina.taft@intel.com).

C. Barakat is with INRIA, Sophia Antipolis FR-06902, France (e-mail: cbarakat@sophia.inria.fr).

P. Thiran is with the Laboratory for Computer Communications and their Applications, Faculty of Computer Sciences and Communication Systems (LCA-I&C), Swiss Federal Institute of Technology at Lausanne (EPFL), CH-1015 Lausanne, Switzerland (e-mail: Patrick.Thiran@epfl.ch).

Digital Object Identifier 10.1109/JSAC.2004.836368

of a single failure. Protection is provided at the WDM layer via a 1:1 protection scheme that guarantees fast recovery after a single failure. The second class of service, called best-effort protected (BEP) is new. It does not provide a specific guarantee on service disruption. Instead, in the case of failure, it offers to restore as much of the affected traffic as possible. For a survey on protection and restoration strategies, see, e.g., [13].

Recently, the problem of service management has gained a lot of attention in the optical community [5], [9], [14]–[17]. Proposals for different service classes in optical networks were introduced by Gerstel and Ramaswami [14]. Ramamurthy and Mukherjee [15] study the traditional $1 + 1$ and 1:1 protection strategies at the WDM layer for a single class of traffic. They formulate the corresponding integer linear program (ILP) optimization problem applicable to small networks. Mohan and Somani [16] propose a class of service that offers a minimal level of protection to every connection. They claim that if the demands are highly dynamic, it is possible to select routes whose (shared) backup paths have a specified maximal nonzero probability of being unavailable if a failure occurs. Sridharan and Somani [17] formulate the ILP problem when three different service classes coexist. They try to minimize the capacity requested by all working and backup paths, weighted by the traffic class to which it belongs (since each class brings in a different amount of revenue). Ramamurthy and Mukherjee [15] prove that the general problem is NP-complete for a single class of traffic. Hence, the recent proposal for three classes of traffic at the WDM layer may be too complex to apply to real networks.

In an IP/WDM network, survivability can be provided at the IP layer or at the WDM layer. Each layer presents different advantages and drawbacks [4], [8]. Some multilayer protection/restoration schemes can adequately combine the advantages of each layer and still avoid most of their disadvantages [7]. They raise another challenge, however, namely the complexity of coordinating the different restoration schemes at the various layers (some solutions are proposed in [7]). In this paper, this race between the layers for restoring traffic is circumvented by allocating this task to a different layer for each traffic class. FP traffic is rapidly and completely protected at the WDM layer, whereas BEP traffic is restored (at a slower scale) at the IP layer.

In [18], we initially presented the idea of two classes of service differentiated by their level of protection, and we have shown that networks can safely carry a much larger load (in scenarios without failures) if they support these two service classes. In that work, we proposed an ILP model to find the primary and backup paths (sequence of fibers in the physical topology) for each logical link and to maximize the BEP traffic carried by the network in the no-failure scenario. This problem is known in the literature as a **mapping problem**. We did not study the restoration of the BEP traffic after the occurrence of a physical link failure. This is an important issue because it is clearly preferable for BEP traffic to experience a smooth, gradual degradation rather than a sudden, total disruption during failure episodes. By considering single failure events in our solutions, we can reduce the likelihood of total disruption and instead push the solutions toward ones that will yield smooth degradations. In this paper, we thus extend our work by incorporating the impact of single failure events. We incorporate additional constraints that carriers face, consider fairness in the excess bandwidth repartition, and provide a heuristic solution based on Tabu Search (TS) methodology that can scale to large networks. In order to provide a complete solution to supporting our two proposed services, we also **design a scheduler** that is needed at the IP layer to distinguish packets from the two services during failure episodes. The scheduler is based on *weighted fair queueing* mechanism and is transparent to both classes when the network is in normal operation (i.e., no failures). Our scheduler helps ensure that FP packets continue to experience the same SLA after failures, while BEP packets may experience a degradation. An appealing advantage of our scheduler is that it does not require any particular signaling to switch between the no-failure and failure modes, instead this switching is driven by the change in the available bandwidth at the WDM layer. The heuristic algorithm and scheduler also constitute extensions to our earlier work.

The goals of this paper are: 1) to quantify how much BEP traffic can be carried on the network without impacting the FP service; 2) to determine how to allocate the BEP traffic load among all the logical connections such that the partition of the BEP traffic is as fair as possible; 3) to maximize network-wide load carried, while simultaneously balancing the tradeoffs of designing for normal operating conditions versus for failure modes; 4) to assess the service degradation during failure episodes; and 5) to evaluate the success of the composite mapping and scheduling solutions by examining the performance of each class of service in terms of throughput, delay and losses at the IP layer. Task 5) is carried out using *ns* simulation. The output of the mapping problem solution is used to establish the physical and logical topologies, that are in turn used as inputs to the simulator.

The remainder of this paper is organized as follows. The FP and BEP classes of service are fully defined in Section II. In Section III, we explain which components of the overall problem belong to which layer (physical or logical), give a formal problem statement and describe our approach. A heuristic solution based on TS methodology is introduced in Section IV. The scheduler is described in Section V. Performance results for both medium and large-sized networks are presented and discussed in Section VI, along with a validation of our heuristic. Section VII concludes the paper.

II. Definition and Provisioning of Classes of Service

The **FP** service guarantees its customers that their traffic is protected against any single point of failure in the backbone. FP traffic is protected via precomputed, dedicated backup paths at the WDM layer, using a 1:1 protection strategy. Fiber failures are transparent to the IP layer for this class of traffic. In a 1:1 protection scheme, the FP traffic is transmitted only on one path (called the *working* or *primary* path). If this path fails, the sender and receiver both switch to the other path (called the *backup* path). Our idea is to take advantage of 1:1 protection because the reserved but unused capacity on the backup path can be given to unprotected traffic whose packets would be dropped in the case of a failure.

The **BEP** service is one whose traffic runs on the excess backbone capacity during *normal* operation (i.e., a network state with no failures). The BEP traffic of each logical connection can be routed on either the primary or the backup path, but not on both (i.e., it cannot be split over two paths). When a failure occurs, the available bandwidth drops on all logical links that share this fiber. Our IP scheduler enters into action and discards the BEP packets as needed, while protecting the FP packets. Thus, the SLA performance, in terms of packet drop rate, received by the BEP traffic depends upon the amount of overprovisioning that exists after both FP and BEP traffic have been accomodated.

We point out that in an environment in which each logical connection is protected via a 1:1 scheme at the WDM layer, and in which failures happen one at a time, the logical topology will always be connected. Thus, the logical topology will be always able to apply a restoration strategy at the IP layer, and does not suffer from the *failure propagation* problem described in [20]–[22].

What BEP offers to users, as a tradeoff for a lower amount of protection, is either a larger throughput, or a cheaper price. The wide variety of applications that exist today do not all need the same level of protection. Some applications, such as IP telephony, video-conferencing, and distance surveillance require 100% availability and, hence, full protection against network failures. Others, like on-line games, Web surfing, and Kazaa downloads are likely to be willing to tradeoff a partial and slower protection for increased throughput (or a lower price). Such tradeoffs are attractive as long as the probability of a service becoming unavailable is very small. Applications like e-mail can fall into either one of these service classes.

III. Mapping: Problem Statement

The main problem we address is to find a mapping of IP-layer logical links to physical fibers such that 1) the FP traffic, specified by an FP traffic matrix, is protected and 2) we maximize network-wide load (including both traffic classes) subject to a constraint imposing a fairness policy on the allocation of BEP load among all the logical connections. Our intent is to add BEP traffic into the system such that there is no impact at all on the protection quality received by the FP traffic in the case of either a single failure or even multiple failures as long as none of them is a *critical* failure. In this context, a *critical failure* is a multiple failure scenario that brings down a set of links such that both the working and backup paths of the same logical link are interrupted.

We focus on PoP-to-PoP (point-of-presence) topologies at the IP layer, rather than on router-to-router topologies that consist of hundreds of routers. A PoP is an ensemble of core and access routers that usually reside in a single building in a metropolitan area. PoPs are interconnected via inter-PoP links attached to the core routers. The access routers are used to connect customers to the backbone. With this topology, the logical links we map capture the inter-PoP backbone links. Access routers can be ignored because they do not connect directly to other PoPs or other routers in the backbone.

The block diagram in Fig. 1 clarifies the inputs and outputs of the mapping problem. A number of inputs to our problem,

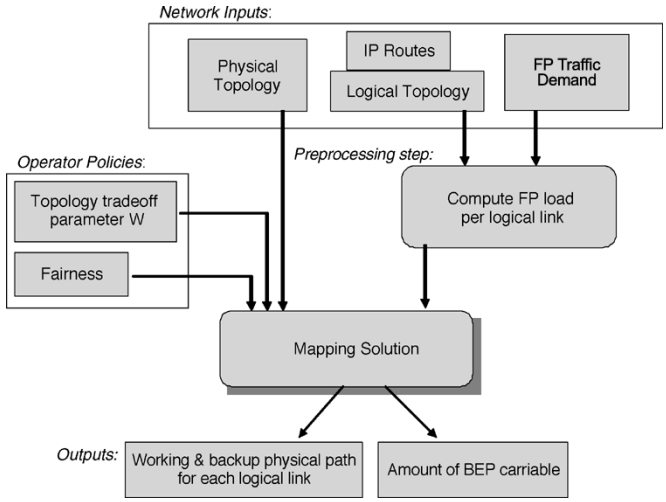


Fig. 1. Block diagram of method.

which define requirements and constraints, come from the IP layer (labeled "network inputs" in the figure). Two of the features we support, fairness and the topology tradeoff parameter (explained below), would be specified by an operator as they essentially define policies (hence, labeled as "policy inputs" in the diagram). We next discuss each of the elements in this diagram and try to clarify which components of the problem are related to the logical (IP) layer and which are part of the physical (WDM) layer. To be clear, we state some definitions of basic terms. We use the expression *logical link* to refer to a single link between two PoPs at the IP layer. We use the term *logical connection* to refer to a sequence of logical links. Each logical link corresponds to a sequence of one or more physical links interconnected via optical cross-connects (OXCs).

The **FP traffic matrix** is a part of the logical layer. We decided to focus on maximizing the amount of BEP traffic carried, while letting the FP traffic be specified by an input demand matrix. The reason for this is because capacity planning in the Internet is typically done using an IP layer traffic matrix that specifies the average amount of bandwidth that needs to flow between any two PoPs or PoPs in a domain. After we choose an initial matrix, we scale the entire matrix up, in order to load the maximum amount of FP onto our network. By "scaling up," we mean that we multiply all elements in the matrix by a constant factor that is as large as possible. The limit on how much the matrix can be scaled up is defined by the maximum amount we can protect.

The **IP routes** are those given by either the OSPF or IS-IS protocol that operates at the IP layer. These protocols usually compute shortest-path routes between PoPs, based on a set of preassigned link costs. A path specified by OSPF (or IS-IS) is, thus, a sequence of *logical links*.

Both the FP traffic matrix and the IP routes are inputs to our problem. Using these two inputs, together with the logical topology, we can calculate the aggregate load for each logical link by routing the *FP traffic matrix* over the *logical topology* according to the OSPF *IP routes*. As depicted in the block diagram, this is considered as a preprocessing step to the optimization problem. Three of these things—the FP traffic matrix, the IP routes, and the logical link FP load (all coming from the IP

layer)—constitute the network inputs needed for our optimization problem at the physical layer.

The optimization procedure needs to find a pair of disjoint fiber paths for each logical link. One fiber path is for the working path and the second is for the backup path. There are typically a large number of such possible pairs for each logical link. We choose among the many candidate solutions by evaluating the corresponding amount of BEP traffic that maximizes our objective function after we have satisfied the demands for FP traffic. Since each logical connection is allowed to carry a certain amount of BEP traffic, the network-wide view of the total BEP traffic carried can also be expressed as a **BEP traffic matrix** with the same rows and columns as the FP traffic matrix.

We should mention at this point that we cannot solve the mapping problem separately for the FP and BEP traffic. Indeed, the BEP traffic takes the same routes as the FP traffic at the logical layer, and is mapped on the working or backup path of the logical link for the FP traffic at the physical layer. The two traffic classes need, therefore, to be considered simultaneously in the mapping problem. In particular, we cannot consider the FP traffic matrix as a simple "bias" on the capacity of the logical links available for BEP traffic.

We now explain our objective function more carefully. We want to select a mapping that is good under two types of scenarios: the normal network state in which no link has failed, and the network state in which a single link has failed. If the optical layer is composed of L physical links, then the number of failure scenarios is L. The network, or topology for each of the L failure scenarios, is the original topology with one link missing. Since we want to consider L single failure scenarios (we also used the term "failure modes") and one normal scenario with no failures, we essentially need to do an optimization over $L+1$ images of the backbone topology.

A mapping that considers the no-failure mode could assign a large amount of the spare capacity to BEP traffic. Since the BEP traffic is completely unprotected at the WDM layer, this could produce very bad performance, in terms of BEP traffic lost, when some physical links fail. Thus, by focusing on the no-failure mode alone, we would be able to carry a large amount of BEP traffic but experience potentially very poor performance during failures. By considering the failure modes, we can mitigate the performance degradation at the time of failures. A mapping that considers only failure modes would encourage the use of small amounts of BEP traffic as it would only load up an amount of BEP that could survive the particular failure. We, thus, define a *topology tradeoff parameter*, called W with $W \in [0,1]$, that balances the amount of emphasis put on the normal topology versus those (with a link missing) that represent failure modes.

Our objective function contains two terms; the first term specifies the amount of BEP traffic carried by the network in the normal operating state (i.e., no-failure-mode), while the second one is the BEP load still carried by the network after the occurrence of a single failure, and averaged over all the possible single failures. We state this more formally as follows. Let $d^{kh}_{S_0}(\mathrm{BEP})$ denote the BEP traffic carried by the connection $(k,h) \in \mathcal{C}$ in the no-failure mode (denoted by S_0), where $\mathcal{C}$ denotes the set of all logical connections flowing at the IP layer. Let $d^{kh}_{S_{mn}}(\mathrm{BEP})$ denote the BEP traffic carried by the connection $(k,h) \in \mathcal{C}$ when fiber (m,n) has been involved in a failure. The notation S_{mn} refers to the failure mode for link (m,n), i.e., it indicates a network state in which the physical topology is missing link (m,n). Let E^0 denote the set of edges in the optical-layer topology (graph) and, thus, $(m,n) \in E^0$. Our objective function $\mathcal{F}$, that we want to maximize, is now given by

$$\mathcal{F} = (1-W) \sum_{(k,h)\in\mathcal{C}} d^{kh}_{S_0}(\mathrm{BEP}) + \frac{W}{|S_{mn}|} \sum_{(k,h)\in\mathcal{C},(m,n)\in E^0} d^{kh}_{S_{mn}}(\mathrm{BEP}). \quad (1)$$

Note that by modifying the weight W, we are able to reach solutions with different characteristics. When more importance is given to the first term (smaller W), more BEP is carried in the no-failure mode but the average amount of BEP lost is larger when failures occur. On the other hand, if more importance is given to the second term (larger W), less BEP traffic is lost during the failures but less BEP traffic is carried by the network during normal conditions. The topology tradeoff parameter W could be chosen as a function of the probability of a link failure. If the link failure probability is very low, then clearly, we want a small W so that the topology under normal operating conditions is given a very large weight. Conversely, if the probability of failure is high, more importance should be given to the failure modes.

There are a multitude of ways in which BEP can be added to the spare capacity because there are many combinations of bandwidth that can be given to each connection, and each connection can route its BEP traffic on either the working or backup paths. If the goal is to add the maximum amount of BEP possible (in general) or in particular, according to the objective function $\mathcal{F}$, then the result is likely to be a very unbalanced distribution of the BEP load—giving large amounts of traffic to some connections and close to zero to others. In particular, single-hop connections would tend to receive a large amount of BEP, while longer multihop connections would receive nothing or very little. It is intuitive that this would lead to the largest total amount of allocated BEP bandwidth network-wide.

We believe that carriers would find this unappealing because of the unfairness. For this reason, we include in our problem the concept of a **fairness policy**. In our scheme for allocating BEP bandwidth among all the logical connections, we consider two different fairness policies. The first policy is called **minimum guaranteed fairness policy (MinG)**. According to this policy, each logical connection must receive a minimum bandwidth for its BEP traffic, denoted as $Z_{\min}$. After having met this even distribution, there is no further fairness mechanism implemented and each logical connection is free to get as much as it can. This policy is a first step toward the second policy presented, called **Maximum-Minimum fairness policy (MaxMin)**. This policy forces each logical connection sharing a bottleneck logical link to receive the same share of the bandwidth left for BEP traffic. The second policy introduces more fairness among all the logical connections. We will show that the more fairly the BEP bandwidth is distributed, the less BEP load the network will be able to carry.

We now give the formal problem statement, incorporating all of the elements above.

GIVEN:

i) a physical topology (which must be at least biconnected), whose nodes are OXCs interconnected by optical fibers that support a limited number of wavelengths and have limited capacity;
ii) a logical topology whose nodes are IP PoPs interconnected by logical links; these links have a finite limit on the total amount of traffic they can carry (including both FP and BEP); the limit comes from the capacity of their line cards;
iii) an FP traffic matrix, denoted $D_{\mathrm{FP}} = [d^{kh}(FP)] \geq 0$, that defines the FP traffic demand for each pair of PoPs (k, h) at the IP layer, we call these pair origin-destination (OD) pairs;
iv) the routing paths selected at the IP layer for each OD pair of PoPs; this set of routes is denoted by $\mathcal{R}$;
v) a 1:1 FP protection strategy at the WDM layer;
vi) a fairness policy to allocate BEP traffic among all the logical connections;
vii) the objective function F defined above;

FIND

Primary and backup paths for each logical link and the BEP traffic matrix $D_{\mathrm{BEP}} = [d^{kh}(\mathrm{BEP})] \geq 0$ for each pair of PoPs (k, h) at the IP layer in the regular condition in such a way that the network is able to:

i) carry the amount of FP traffic defined as an input by the FP traffic matrix D_{FP};
ii) the objective function F is maximized.

IV. Solution to Mapping Problem

We develop two solutions to this problem. This first one uses optimization techniques to find an optimal solution based on formulating the problem as an ILP. Although this approach can find optimal solutions, it is limited in its applicability since even for moderate size networks, obtaining an optimal solution to this problem becomes quite cumbersome due to the large number of variables and constraints involved in its formulation. Indeed, a simpler version of this problem, in which one tries to optimize the network load for only one class of service, was already proven to be NP-complete [15]. U.S. backbone carriers can have upwards of 30 OXCs and 50 fibers in a physical topology, and upwards of 20 PoPs and 40 bidirectional logical links at the IP layer. It is, thus, clear that heuristic solutions are the only practical candidate solutions that carriers can consider using. Our second solution defines a heuristic algorithm based on the TS methodology that can be used in practice for actual carrier backbone networks. Due to lack of space, we do not include our optimal ILP solution in this paper. We refer the interested reader to [24]. We provide our heuristic solution herein and use our optimal solution to validate the heuristic algorithm on a medium-sized network (see Section VI-A-3).

TS is based on a partial exploration of the space of admissible solutions, starting from an initial solution usually obtained with a greedy algorithm, and ending when a stopping criterion is satisfied. The algorithm returns the best solution it found during the entire search. For each admissible solution, the algorithm defines a class of neighboring solutions (the *neighborhood*) obtained from the current solution by applying an appropriate transformation, called a *move*. At each iteration of the TS algorithm, all solutions in the neighborhood of the current solution are evaluated, and the best one is selected as the current new solution.

In order to efficiently explore the solution space, the definition of neighborhood may change during the exploration of the solution space; this enables a *diversification* of the search in different solution regions. The TS algorithm can be seen as an evolution of the classical local optimum solution search algorithm called steepest descent [28]. It can avoid getting trapped in local minima due to the TS mechanism that allows limited excursions toward solutions that appear worse than the current one.

The TS method introduces the use of a *Tabu list* to prevent the algorithm from cycling among already visited solutions. The Tabu list stores the latest accepted moves; as long as a move is stored in the Tabu list, it cannot be used to generate a new one. The choice of the Tabu list size is a key parameter of the optimization procedure: too small a size could cause the cyclic repetition of the same solutions, while too large a size can severely limit the number of applicable moves, thus preventing a good exploration of the solution space. The TS heuristic ends when a stopping criterion is reached. A common stopping criterion is simply to stop after some fixed number of iterations has been carried out.

A. Our Algorithm

We now state our algorithm by specifying how we implement each of the elements of a TS heuristic. We have added a preprocessing step that speeds up the rest of the search procedures.

Step 1) *Preprocessing Step*. Generate the set of all admissible pairs of disjoint physical paths that could be used for each logical link. This determines the *admissible* solutions, each of which contains a particular mapping for each logical link. Admissibility here only refers to the fiber paths being disjoint.

Step 2) *Initial Solution*. For each logical link, randomly select one pair of disjoint physical paths. Choose randomly within the pair which physical path is assigned as working path and which one is assigned as backup path. The aggregated BEP traffic flowing on each logical link can be sent either on the working path or on the backup physical path. The path leading to the largest value of the objective function is chosen.

Step 3) *Create Neighborhood*. Select a logical link at random. Keep the working path fixed and change the physical backup path. The set of all the admissible backup paths for the selected logical link defines the neighborhood of the current solution.

Step 4) *Evaluation of Solutions in Neighborhood*. We need to evaluate each solution in the neighborhood and pick the best one. Only the solutions generated

by selecting a *logical link* and *the pair of physical paths* not present in the Tabu list are analyzed during this step.

a) Check the capacity of each solution to ensure that the protection requirements for the FP matrix are satisfied. If enough resources are not available on the two physical paths to protect the FP traffic, then the solution is discarded as infeasible.

b) Determine an allocation of BEP traffic onto the spare bandwidth that maximizes our objective function. Consider putting BEP traffic on either the working or backup paths.

c) Elect the best solution found in the neighborhood as new current solution.

Step 5) *Update*. Update the Tabu list by adding the latest move used to generate the new current solution and removing the oldest. Update the best-solution-seen-so-far if the new current solution analyzed shows a larger value of the objective function F.

Step 6) *Repeat*. If number of iterations is less than some predefined threshold, go to Step 3), else stop.

We now comment on some of these steps in more detail. The move we apply to create the neighborhood has two nice properties. The first one is the guarantee that all solutions in this neighborhood are admissible.[1] The second property is that this kind of move makes it easy to implement a *diversification* step. For example, we can select a different number of logical links at each iteration, which will move up rapidly to another region of the solution space. We apply *diversification* only when a certain number of successive iterations fail to yield improvement. In our simulations, this number is set to 50; when this number is reached, we build a new solution by selecting a random number of logical links between three and five. Note that after the diversification move has been done once, we return to the regular move based on perturbing a single logical link.

We check the feasibility of a solution [Step 4a)] by routing all the logical connections onto the logical topology using the standard OSPF IP routing protocol. Then, each logical link (s,t) is routed over the physical topology using the physical paths selected by the TS metaheuristic. If sufficient resources are not available to protect the entire aggregated FP traffic on each logical link, then the solution is discarded. The next solution is then analyzed. Once we find a feasible solution, we move to Step 4b).

After a solution is claimed *admissible* for the FP traffic, then the BEP traffic needs to be assigned to each logical connection. As we mentioned before, two fairness policies are implemented. The *MinG* policy requires that a minimum bandwidth $Z_{\min} \geq 0$ is assigned to each logical connection for its BEP traffic. The algorithm starts by routing all the logical connections into the logical topology using the OSPF IP routing algorithm, and by assigning $Z_{\min}$ BEP traffic to each connection. Then, the algorithm verifies if the available bandwidth for each logical link is larger than the aggregated FP and BEP traffic flowing on it. If this test is passed, all the single-hop logical connections will get as much as they can, i.e. a further amount of BEP bandwidth equal to the remaining available capacity.

The second fairness policy *MaxMin* is implemented by a water-filling type algorithm as follows. The algorithm starts by routing all the logical connections into the logical topology using the OSPF IP routing algorithm, and by assigning zero BEP traffic to each connection. Then the amount of BEP traffic allocated to each connection is increased in equal increments until a logical link gets saturated. At this point, the BEP bandwidth allocated to all logical connections sharing this bottleneck is frozen (at an equal level for all of them). All the other connections, which do not share this bottleneck, can still receive additional BEP traffic, without impacting the bandwidth allocated to the frozen connections. We then proceed to increase in equal increments the bandwidth to all remaining unfrozen connections, until a new logical link becomes a bottleneck (i.e., saturated). The bandwidth assigned to connections traversing the new bottleneck are now frozen. The algorithm repeats until all the logical connections are frozen. At this point, the bandwidth of each logical connection is determined by its own bottleneck.

We fix the size of the *Tabu list* to be 7. This number was chosen based upon our experience running simulations for different kinds of network topologies and FP traffic matrices. The searching procedure is stopped when a given number of iterations is reached. The number of iterations should be chosen relative to the size of the network and to achieve a good tradeoff between computational time needed and the quality (distance from the optimal solution) of the solutions reached. We set this parameter to 1500 for the medium-sized network and 5000 for the large-sized network.

B. Complexity

We now discuss the complexity of the proposed heuristics. First, we look at the BEP allocation algorithm that distributes the excess bandwidth to BEP connections according to a max-min fair strategy. Let H be the number of logical connections, M the number of physical, and C their capacity. For each logical connection $O(H)$, its BEP load is successively increased by one unit on all the fibers belonging to its path $(O(M))$ until each fiber has reached capacity $O(C)$. This algorithm has complexity $O(HMC)$ since at most $O(C)$ iterations are executed, while at each iteration at most $O(HM)$ operations are required. We now focus on the complexity of TS algorithm. Let T be the maximum number of visited solutions in each neighborhood. For each of them, we have to route the fixed FP traffic and verify the admissibility of the solution. If the solution is admissible, we run the BEP allocation algorithm. The first step requires $O(HM)$ operations since for each logical connection $(O(H))$ we have to route its FP flow on both the working and backup paths, and the maximum length of each path in the worst case is equal to $O(M)$. Then, the complexity to evaluate each neighborhood is equal to $O(T(HM+HMC))$ that is upper bounded by $O(THMC)$. If I is the number of iterations before stopping the algorithm, the complexity of the proposed the proposed TS is equal to $O(ITHMC)$. When we ran our heuristics on a

[1]Note that we distinguish between *admissibility* that refers to two fiber paths being disjoint, and *feasibility* that refers to a set of paths having enough capacity to satisfy the protection needs.

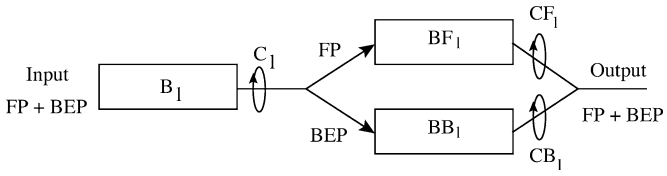


Fig. 2. Scheduler.

550-MHz linux machine with two processors, the running time was approximately 15 min for the large network described in Section VI.

V. Class Based Scheduling at the IP Layer

In this section, we present a scheduler that is able to differentiate the two service classes in the event of a failure at the optical layer. The scheduler is depicted in Fig. 2. The main advantage of this scheduler is that it treats the packets of both classes equally in the no-failure mode, and it protects the FP packets from the BEP packets in the failure mode. The scheduler does its best to provide the FP packets the same service they were getting in the normal mode when a failure happens. This switch between the two modes of operation is automatic and is driven by the drop in the available bandwidth at the physical layer. Many schedulers exist in the literature to provide service differentiation (e.g. WFQ [32], FRED [30], and LQF [31]), however, none of these provide two such modes of operation. In the prior work, packets continuously receive differentiated treatment.

Let C_l denote the capacity of logical link l during normal operation. A buffer of size B_l is available at the input of link l and is served at rate C_l. The space of this buffer can be managed by any policy (Drop-Tail, RED, etc.). By definition of our service classes, the simple best effort service is provided without any guarantees at this stage to any of the two classes.

When a fiber fails at the optical layer, all the logical links sharing this fiber will be switched from their primary paths to their backup paths. Each logical link l, affected by this failure, and whose backup path has a smaller available capacity $C_l^* < C_l$, will experience a drop in bandwidth from C_l to C_l^*. Let CF_l and CB_l denote, respectively, the portion of C_l^* devoted to the aggregated FP and BEP traffic flowing on link l, and computed by the algorithms of Section IV.

After the drop in bandwidth from C_l to C_l^*, packets of FP and BEP classes have to be served at rates CF_l and CB_l respectively. To maintain the same order for FP packets before and after the drop in bandwidth, we keep the buffer B_l (virtually) served at a rate C_l. We also place two parallel queues (one for each class) in between buffer B_l and the link l. After leaving the buffer B_l, a packet goes to its corresponding queue based on its class. The two queues are served in a weighted round-robin way with rates CF_l and CB_l (or with weights $WF_l = CF_l/C_l^*$ and $WB_l = CB_l/C_l^*$). The round-robin scheduler ensures a fine-granularity distribution of the bandwidth C_l^* between the two queues. The outputs of the two queues are connected to the link l whose bandwidth has dropped.

Our scheduler can be seen as the original buffer B_l extended with a weighted fair queue (WFQ) buffer. The original buffer is always (virtually) served at the original rate C_l whereas the WFQ buffer is served at the real rate of the link. When the bandwidth of the link is equal to C_l, the WFQ buffer is transparent; packets of both classes are only queued in buffer B_l and they are served at a rate C_l.[2] This transparency is the result of the fact that the WFQ buffer is implemented in such a way as to be work conserving. When the bandwidth drops, the WFQ buffer is automatically activated and starts to provide the differentiated service. If the peak rate of the FP traffic is less than CF_l, FP packets will only be queued in buffer B_l and get the same service as before the drop in bandwidth. BEP packets will be queued in both buffers B_l and WFQ, except if their peak rate is less than the available bandwidth.

Denote by BF_l and BB_l the two queues of the WFQ buffer for link l. We choose their sizes in a way that they absorb a full buffer B_l. That is

$$BF_l = \frac{C_l - CF_l}{C_l} B_l, \quad BB_l = \frac{C_l - CB_l}{C_l} B_l. \tag{2}$$

These two queues are managed according to the drop-tail policy. Other sizes and policies can also be used for these two buffers.

To illustrate the functioning of our scheduler, we simulate (using the *ns* simulator) a simple scenario where a link of 10 Mb/s is crossed by an FP and a BEP traffic stream of 4 Mb/s each. Both traffic flows are generated by user datagram protocol (UDP) Poisson sources. The size of the buffer B_l is set to 50 packets and all packets are of 500 bytes. We start the simulation in the no-failure mode, then after 500 s, we emulate a failure that drops the bandwidth from 10 to 5 Mb/s. We stop the simulation after 1000 s. The weights of the scheduler are set as follows: $CF_l = 5$ Mb/s, $CB_l = 0$. We plot as a function of time the throughput of the FP and the BEP traffic averaged over 1 s intervals, and we also plot the length of the queue in the three buffers of our scheduler. The plots are shown in Figs. 3 and 4. For the throughput, we see clearly that the FP traffic is not affected by the failure and see how the BEP is penalized. For the queue length, the buffer at the first stage shows the same occupancy before and after the failure, whereas the buffers at the second stage are empty and transparent before the failure. After the failure, the FP buffer remains almost empty since the rate of the FP traffic is on average less than the available bandwidth of 5 Mb/s. The BEP buffer overflows after the failure since the BEP traffic is on average more than the bandwidth not used by FP. The FP traffic is then protected in terms of throughput and delay whereas the BEP traffic is penalized (less throughput, more delay, and losses). This shows that our scheduler is achieving its goals.

VI. Numerical Results

We now evaluate the performance of our two service proposal on the medium-sized network shown in Fig. 5, and on the large-sized Sprint backbone shown in Figs. 9 and 10. We first solve the mapping problem. Then, to study the performance degradation of both classes in case of physical link failures, we simulate each failure scenario in a network whose logical and physical

[2]In order not to make a packet suffer two transmission times in B_l and on the link, we implement the buffer B_l in such a way as to deliver a packet at the beginning of its service time and not at the end of service, as in classical queueing systems.

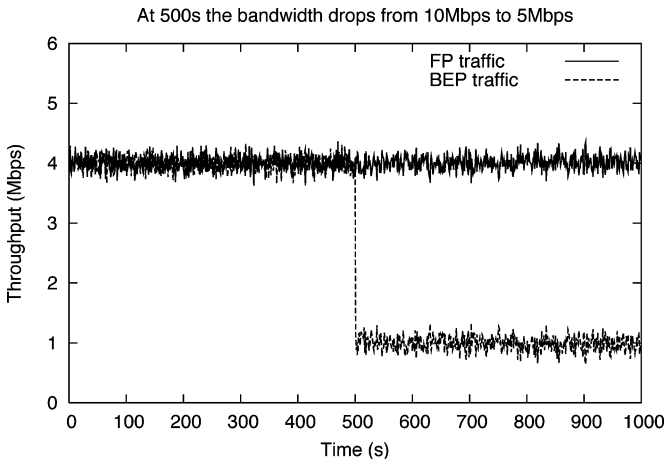


Fig. 3. Throughput for FP and BEP.

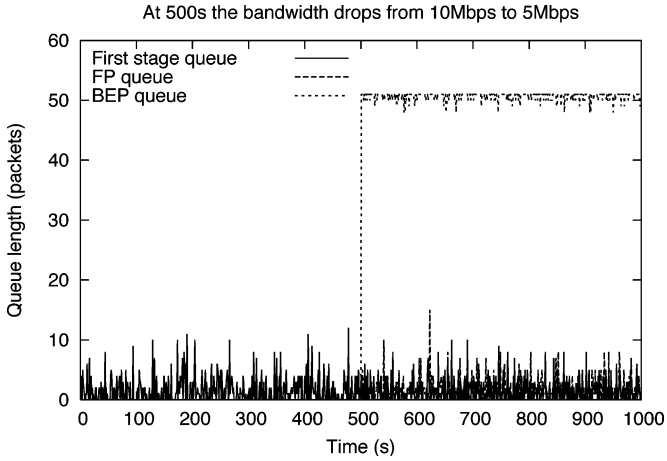


Fig. 4. Occupancy of the three buffers of the scheduler.

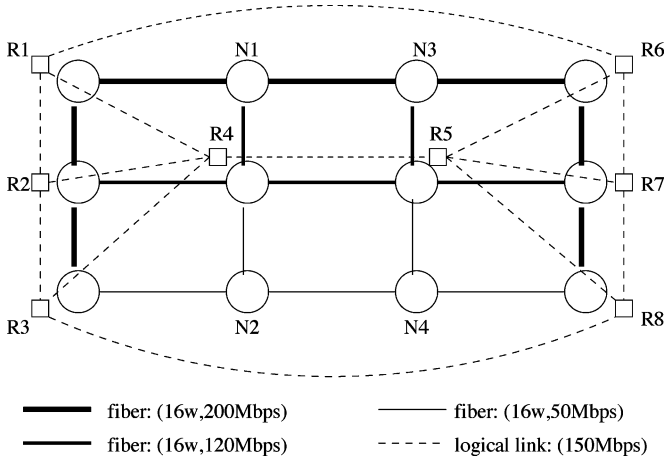


Fig. 5. Medium-sized network composed by 12 OXCs, 17 WDM fibers at the WDM layer, 8 routers, and 13 logical links at IP layer.

topology are connected according to the output of our mapping solution. We use the *ns* simulator with our own implementation of our scheduler in each router. We remind the reader that for large networks, such as the Sprint backbone, we collapse all the intra-PoP routers in one single PoP-node, and we consider the PoP as a large backbone router. For a medium-size network, we study the real router-to-router topology.

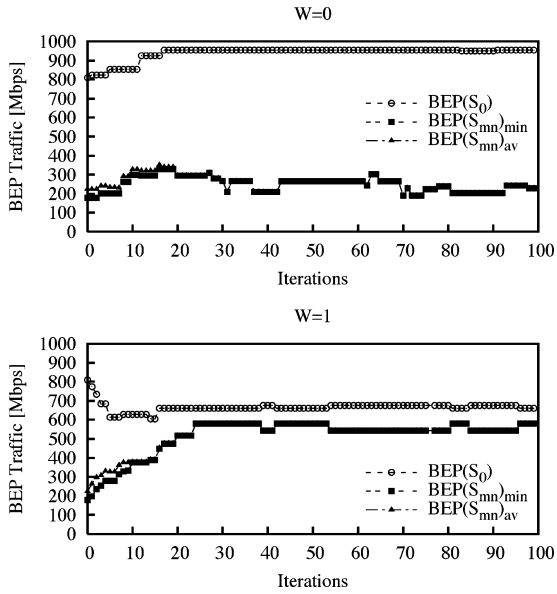


Fig. 6. TS evolution for $W = 0$ (Optimization in the no-failure state only) and $W = 1$ (Optimization in the single failure state only). No fairness policies.

A. Mapping: Medium-Sized Heterogeneous Networks

We use the medium-size network shown in Fig. 5 whose WDM layer is quite heterogeneous. Three different WDM systems are implemented: some fibers are equipped with 16 channels at 200 Mb/s, some with 16 channels at 120 Mb/s, and others with 16 channels at 50 Mb/s. The capacity of each channel is marked on the figure via the thickness of the line as described in the legend. The line card speed limit for each logical link is set to 150 Mb/s.

1) Topology Tradeoff Issue: We now quantify this tradeoff between optimizing for the no-failure mode alone versus finding a good solution for single-failure modes. We use an FP traffic matrix in which each element in the matrix (each logical connection) is assigned a random value uniformly between 0 and 100 Mb/s. (We remind the reader that after we choose an initial matrix, we scale up the entire matrix, in order to load the maximum amount of FP onto our network.) We look at three performance metrics: the amount of BEP traffic carried by the network in the no-failure mode S_0 (denoted $\mathrm{BEP}(S_0)$), the minimum $(\mathrm{BEP}(S_{mn})_{\min})$ and the average BEP traffic $(\mathrm{BEP}(S_{mn})_{\mathrm{av}})$ carried by the network where the minimum and average are computed over all the single failure modes S_{mn}. These metrics are plotted in Fig. 6. This figure includes two graphs for two extreme values of the topology tradeoff parameter W, namely $W = 0$ (maximize the BEP traffic only for the no-failure mode) and $W = 1$ (maximize the average BEP traffic over all single failure scenarios without any consideration of the no-failure mode). These graphs are plotted against the number of iterations executed by the TS heuristic. Before commenting on our performance metrics, we make an observation about the convergence of our heuristic algorithm. Although we limited the number of iterations of our algorithm to 1500 for the medium-sized network (Section IV), we see here that in all cases it typically takes no more than 30–40 iterations for our heuristic to converge.

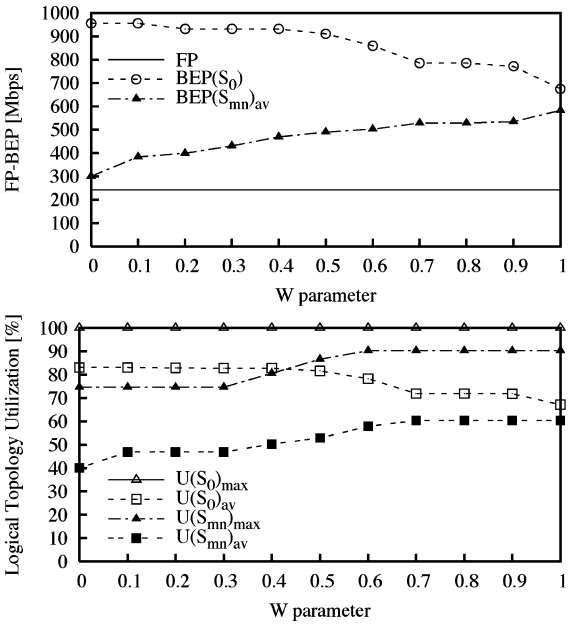


Fig. 7. FP-BEP traffic carried by the network and logical link utilization in all the failure states as a function of the weight W. No fairness policies.

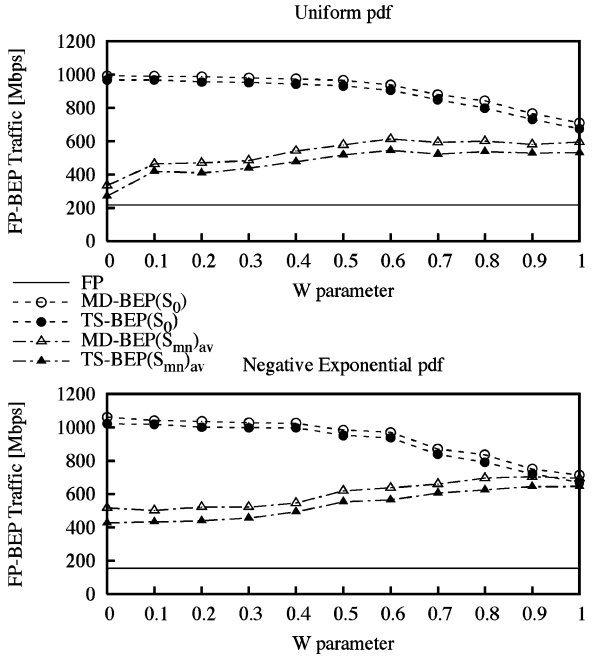


Fig. 8. Comparison between Model and TS. Two different traffic matrices are analyzed, the first with its entries uniformly distributed, and the second with entries following a negative exponential distribution. No fairness policies.

In the case of $W = 0$, our algorithm would enable around 950 Mb/s of BEP traffic to be carried in the network during no-failure modes. The BEP traffic values in the figure are summed over all logical links and, thus, represents a network-wide BEP load. The load generated by FP traffic in this example was roughly 240 Mb/s; hence, our two-service class proposal combined with a good mapping solution, enables a network to increase its total carried load by a factor between 3 and 4. Since we optimized for the no-failure mode only, when failures do happen, the average amount of BEP carried after a failure typically drops to around 250–300 Mb/s. Some solutions lose 63% of the BEP traffic they enjoyed before the failure, while others can lose as much as 77%.

When we optimize for the failure modes ($W = 1$), we can see that during normal operation, the network carries roughly 650 Mb/s of BEP traffic, and when a failure occurs this number typically drops to around 550 Mb/s. Overall, we carry approximately 21% less BEP traffic in normal operating conditions (S_0) when we optimize for failure modes instead of optimizing only for the no-failure mode. On the other hand, the BEP loss in the event of a failure is limited to around 23% when $W = 1$ as opposed to the 60%–75% loss incurred when $W = 0$. This clearly indicates the tradeoff between optimizing for failure modes as opposed to nonfailure modes.

2) Setting the Value of the Topology Tradeoff Parameter W: We now examine how the performance varies as a function of W as it ranges from 0 to 1. The metrics we examine here are the total network load carried including both FP and BEP traffic (shown in the top portion of Fig. 7), and the utilization of the links at the logical level (shown in the bottom portion of Fig. 7), where the utilization numbers again include both FP and BEP traffic.

We observe that by increasing W, the amount $\mathrm{BEP}(S_0)$ of BEP traffic carried, under no-failure conditions decreases, whereas the average amount $\mathrm{BEP}(S_{mn})_{\mathrm{av}}$ of BEP carried by the network in failure modes increases. This is what we would expect given our understanding of the topology tradeoff issue. The same behavior is true for the metric of logical link utilization—with the exception of the maximum utilization under the no-failure mode. This makes sense; the corresponding curve $(U(S_0)_{\max})$ is always at 100% because there is always at least one link in the network at 100% utilization. We point out that without BEP traffic, the average logical link utilization would be around 18%. This is in the typical range at which carriers load their networks today. Carrier's do this as part of their overprovisioning approach, which provides additional robustness to large failure events. Hence, these results for our two-service proposal indicates that carriers could run their networks at much higher load levels (e.g., between 40%–80% on average) *without* impacting today's clients who essentially use an FP service.

3) Validation of Heuristic: In this section, we compare the performance of our heuristic algorithm to that of our optimal ILP solution (presented in [24]). To do this over a multiplicity of cases, we first examined 50 different FP traffic matrices, each of which was generated using a uniform distribution. Then, we generated another 50 traffic matrices whose entries were drawn from a negative exponential distribution. Both traffic matrix types used an average of 50 Mb/s. Again, each FP traffic matrix is scaled up as much as possible until some FP traffic connections reach their limit, and would no longer be protected on a 1:1 basis if we would continue to increase their allocated rate.

Results from this comparison are given in Fig. 8. The notation $\mathrm{MD-BEP}(..)$ refers to the amount of BEP traffic carried in the solution found by our ILP model, while the notation $\mathrm{TS-}$

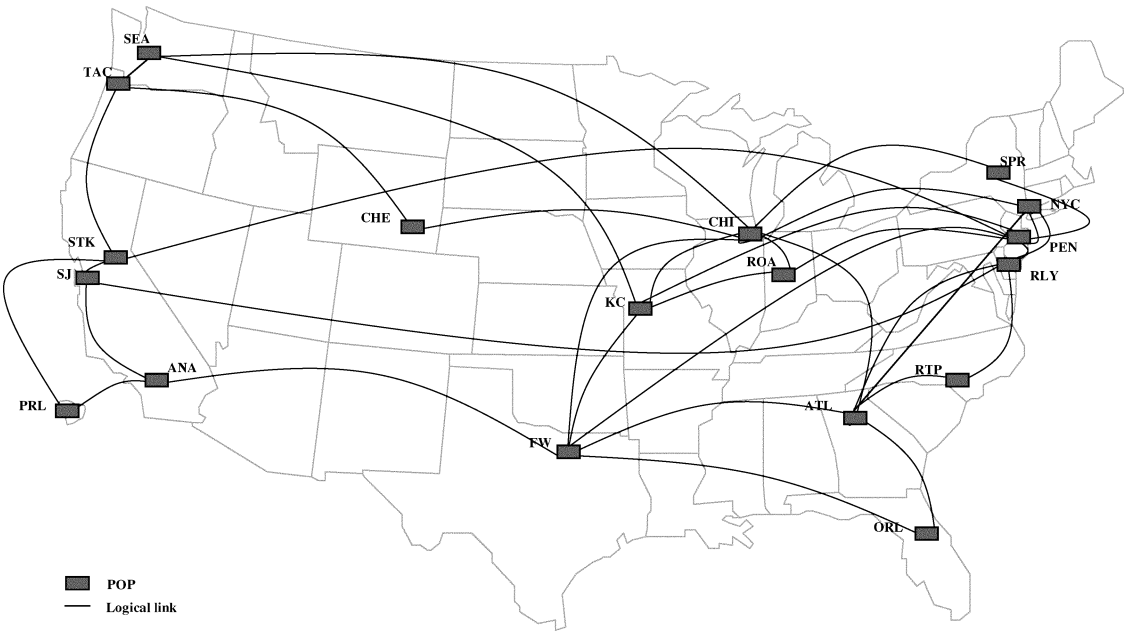


Fig. 9. Sprint logical topology: 18 IP routers with 36 bidirectional logical links.

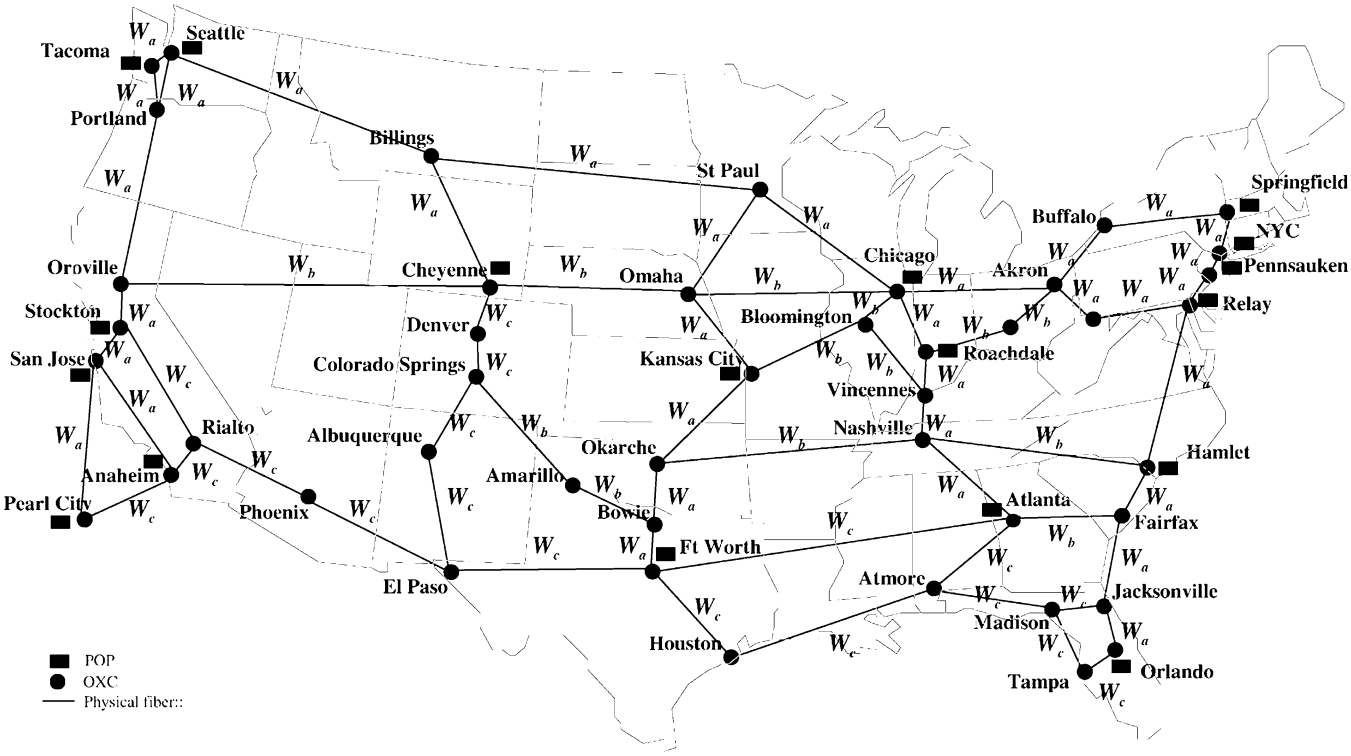


Fig. 10. Sprint WDM topology: 36 OXC with 55 WDM fibers. Heterogeneous backbone: W_a represents an OC192 system equipped with 40 channels at 10 Gb/s, W_b represents an OC48 system with 40 channels at 2.44 Gb/s, and W_c an OC12 system with 40 channels at 622 Mb/s.

BEP(..) refers to the amount of BEP carried in the solution found by our TS heuristic algorithm. In these figures, we plot the FP and BEP loads separately. We can see for that all values of W and for both types of FP traffic matrices (uniform and negative exponential), the performance of the heuristic and the model are very close. For $W = 0$, the gap between the TS heuristic and the ILP model is less than 3%, while for $W = 1$, the gap is less than 5.8% for both distributions.

B. Mapping: Large-Sized Heterogeneous Networks

We now examine how the previous results extend to a large-size network, such as the Sprint backbone. Figs. 9 and 10 display the two simplified versions of the WDM and IP layers actually used in the Sprint backbone. The WDM layer consists of 36 OXC and 55 WDM fibers, while 18 PoPs and 36 logical links are present at the IP layer. Three different WDM systems are used, which we call W_a (40 channels at 10

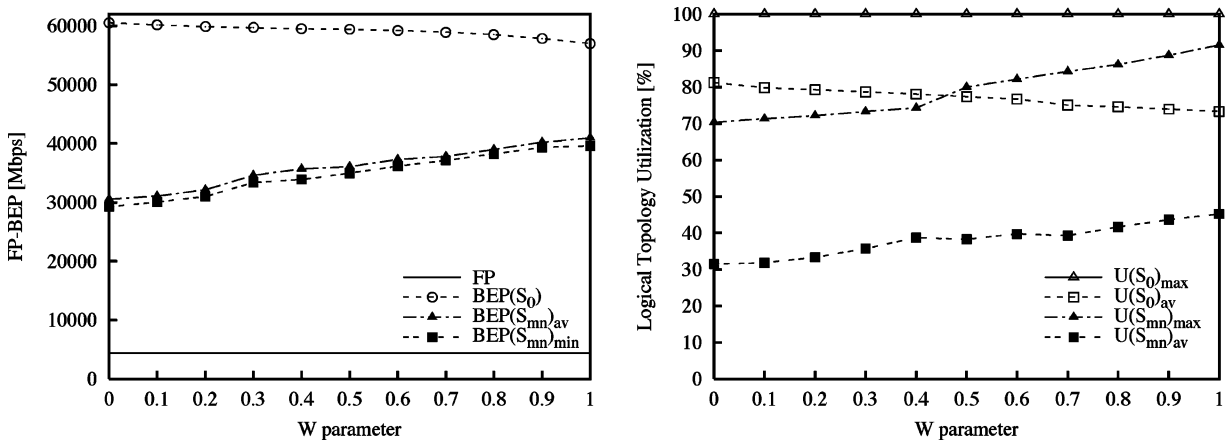


Fig. 11. Load and utilization performance on Sprint backbone for a uniform FP traffic matrix.

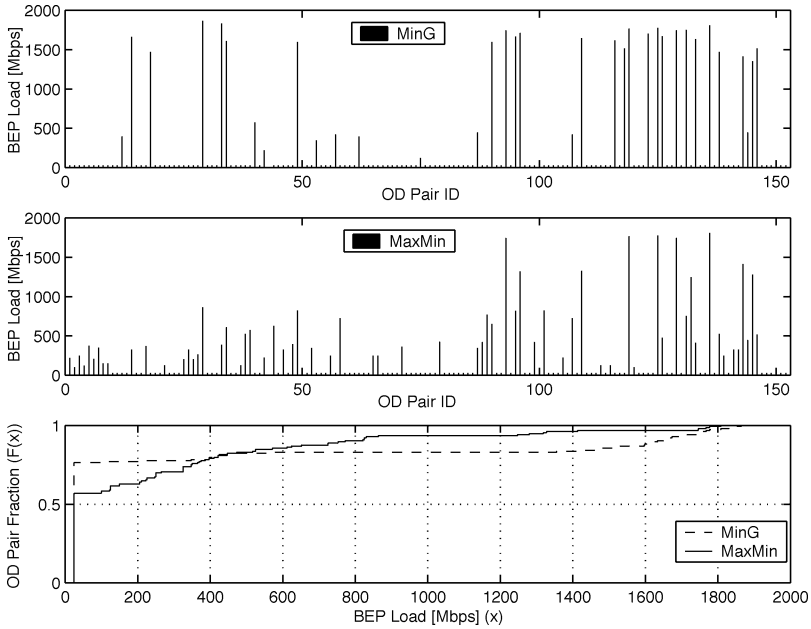


Fig. 12. BEP load distribution among the logical connections for the two fairness policies introduced with a $Z_{\min} = 25$ Mb/s.

Gb/s), W_b (40 channels at 2.44 Gb/s), and W_c (40 channels at 622 Mb/s). Each IP PoP has an electronic speed equal to 2.4 Gb/s.

1) Basic Results: We ran our TS heuristic for the Sprint backbone using 15 traffic matrices randomly generated from a uniform distribution between 0 and 100 Mb/s as described previously. Our two metrics of load levels carried and logical link utilization are shown in Fig. 11. The general results are similar to those obtained for the medium-sized network.

For all values of W, the amount of BEP traffic carried during no-failure scenarios ranges from 55 to 60 Gb/s. This corresponds to an increase in the carried load of a factor of 9–10, as compared with a network carrying FP alone. In the event of a failure, the average amount of BEP lost ranges from 30% to 50%. Even in the most conservative case ($W = 1$), we can support a BEP service carrying approximately 55 Gb/s of traffic, and the performance degradation suffered by BEP during failure events is approximately the loss of 1/3 of its traffic. In this case, the average logical link utilization is around 70% during normal operation and drops to roughly 40% during failure modes.

2) Impact of the Fairness Policies: We now examine the difference in terms of BEP network load carried by each logical connection when the two fairness policies are implemented. On the top and middle of Fig. 12, we show the BEP bandwidth in megabits per second (y axis) assigned to each logical connection (x axis) by using respectively the *MinG* policy and the *MaxMin* policy. The bottom of Fig. 12 shows the cumulative distribution of the two fairness policies, i.e., the fraction of logical connections (y axis) with an assigned bandwidth less than or equal to a specific value (x axis). The case shown is for $W = 0.7$ and $Z_{\min} = 25$ Mb/s.

First, note that the minimum BEP bandwidth assigned to each connection is greater than or equal to $Z_{\min} = 25$ Mb/s.[3] By looking at the number of origin-destination (OD) pairs with a BEP load larger than 25 Mb/s, we can see from these figures

[3]This characteristic is not visible from the top and the middle of Fig. 12 because of the large y axis range, but is clear by looking at the bottom of Fig. 12.

that the *MinG* policy assigns almost 80% of the logical connections to the minimum value, while the *MaxMin* policy assigned only 60% of its logical connections to 25 Mb/s. This can also be seen by looking at the bottom plot for the case when the x axis value is at 25 Mb/s. While the *MinG* policies successfully avoids assigning some connections zero bandwidth, it is still prone to a tendency to give each connection either a minimum or some maximum value (in this case roughly 1600 Mb/s) with very few connections receiving some intermediate value. We can observe that very few OD pairs have values in the range of 25 to 1600 Mb/s by looking at the bottom plot, in which the *MinG* policy is nearly flat in the range of 25 to 1600 Mb/s. The *MaxMin* curve, however, does have gradual change and growth in that bandwidth range. It is also clear from the top two plots that the *MaxMin* policy has more OD pairs with values in the 100–1000 Mb/s range. As expected, the *MaxMin* policy yields better fairness than a *MinG* policy. We computed the total load carried in the two fairness policies, and the *MaxMin* policy carries 14% less load than the *MinG* policy. Hence, the tradeoff between these two policies is that increasing fairness leads to a reduction in overall total load carried.

C. IP Scheduler: Simulation Results

We now examine the on-line performance of our proposed schemes. We study the medium-sized network shown in Fig. 5 and implement our scheduler in each of the routers. The performance of both classes of service was evaluated using the *ns simulator*. We remind the reader that to assess the performance of the two classes of service in case of a physical link failure, we need to know exactly which sequence of physical links are used by each logical link. For this purpose, we implement the solution obtained by solving the mapping problem for this topology, using the heuristic proposed in Section IV. Between each pair of routers, we set the two *average* traffic flow rates (for FP and BEP traffic) according to the values used in the previous uniform traffic matrix. We use this average rate for each logical connection as the mean of a Poisson distribution so that packet arrivals are generated according to a Poisson process. We take Poisson traffic for its simplicity and for its good approximation of Internet traffic in IP backbone networks [29]. The traffic is symmetric in that two routers exchange the same amount of traffic in both directions. We assume each logical link to have 150 Mb/s card speed. We take all logical link delays equal to 10 ms, and we set the packet size of FP and BEP packets to 1500 bytes. All simulations are run for a long duration of 1000 s.

First, we run a simulation for the no-failure case. Between each pair of routers, we measure the throughput, loss and delay of FP and BEP traffic. With eight IP layer routers, we have $8 * 7 = 56$ router pairs. Since the logical connections are symmetric, we group bidirectional traffic into a single router pair. We have, thus, 28 such pairs. We also measure the aggregate throughput on each logical link. Next, we run a simulation for each failure scenario. Seventeen failure scenarios are considered in total, numbered from 1 to 17, with the no-failure scenario numbered 0. Every failure causes a drop of the total bandwidth available for logical links. Logical links are symmetric in all failure scenarios. Fig. 13 summarizes these drops in bandwidth. The lines in this figure correspond to the logical links (13 in total) as defined in Fig. 5. The x axis represents the index of the failure scenario considered. The y axis represents the total bandwidth available on a logical link in a failure scenario. We have split the 13 logical links over three plots for ease of readability.

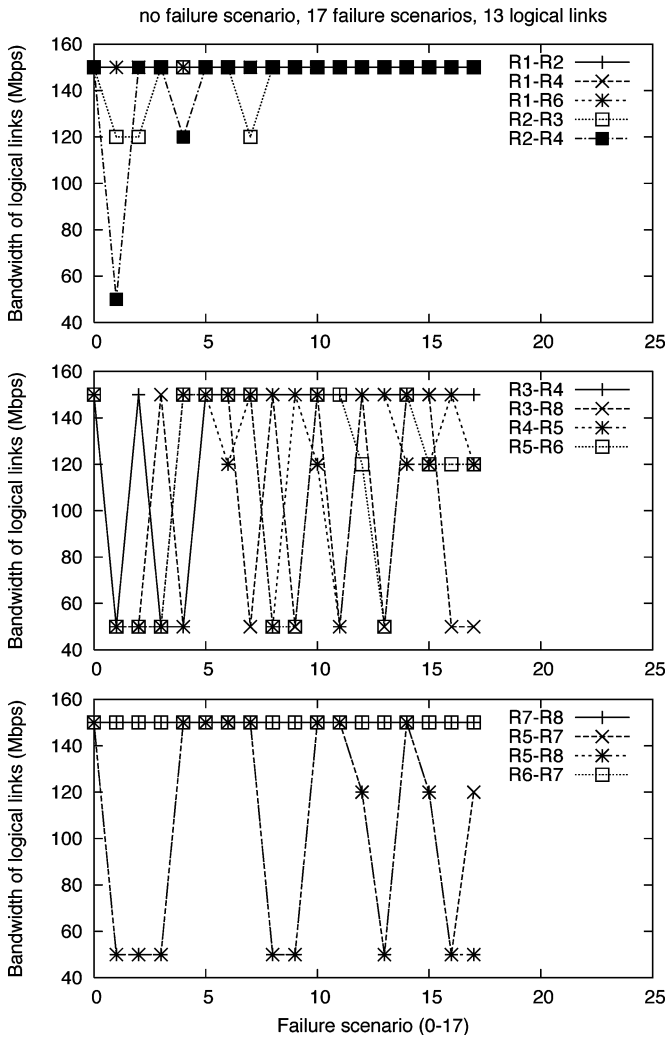


Fig. 13. Bandwidth of each logical link (13 in total) in the normal operation state (failure id 0) and for any physical link failures (failure id from 1 to 17).

For all failure scenarios, we take the following measurements: 1) throughput, delay, and losses between router pairs and 2) aggregate throughput on every logical link. Using these measurements, we can study the impact of a fiber failure on each class of service at the IP layer in terms of throughput (Figs. 14 and 15), delay (Figs. 16 and 17), and loss (Figs. 18 and 19). For all these figures, the x axis shows the performance of the traffic in the no-failure mode and the y axis shows the performance of the traffic in the failure mode. The number of points in each figure is equal to the number of failure scenarios (17) times the number of router pairs (28). Thus, each point represents the end-to-end performance between one pair of routers for one failure scenario.

In the throughput plot for the FP traffic (Fig. 14), all the points lie around the diagonal. This indicates that the throughput for FP traffic is not impacted by single-link failures. In the case of delay and loss (Figs. 16 and 18), there are just a few points that are a bit above the diagonal. Note that this would happen even without the addition of BEP traffic. When the bandwidth drops

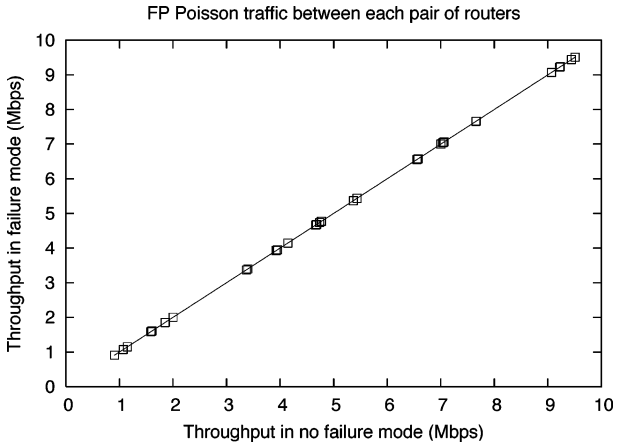


Fig. 14. Throughtput for FP traffic. Failure mode versus no-failure mode.

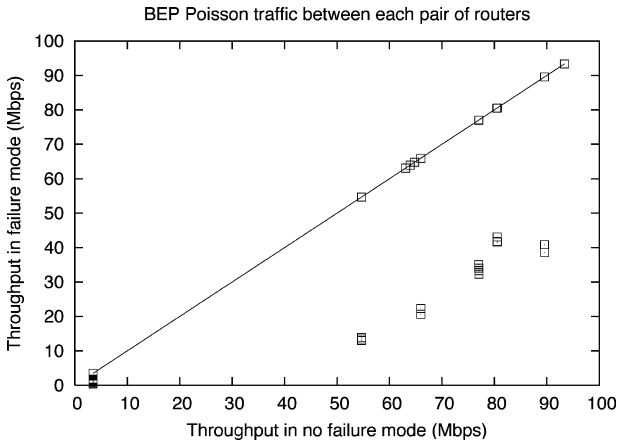


Fig. 15. Throughtput for BEP traffic. Failure mode versus no-failure mode.

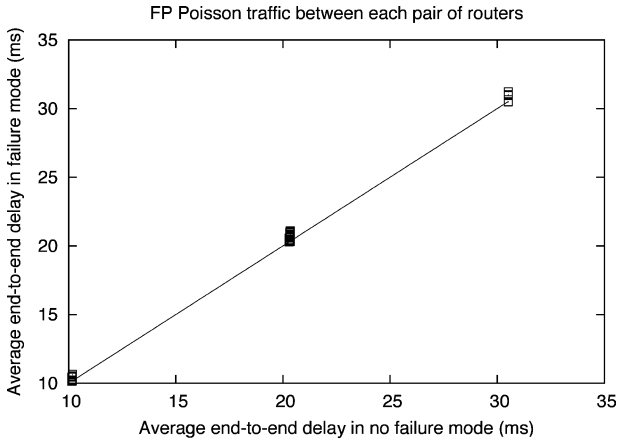


Fig. 16. End-to-end delay for FP traffic. Failure mode versus no-failure mode.

during a failure, the transmission time of FP packets increases, so we cannot avoid an increase in the packet delay even if the average FP traffic is less than the available bandwidth in the failure mode. For the loss, it is the same thing since buffers are finite and the traffic at the packet level is Poisson (more bursty than constant bit rate). These figures show that our mapping solution and scheduler are working properly in that they achieve their goal of adding BEP traffic into the network without impacting the SLA of the FP traffic.

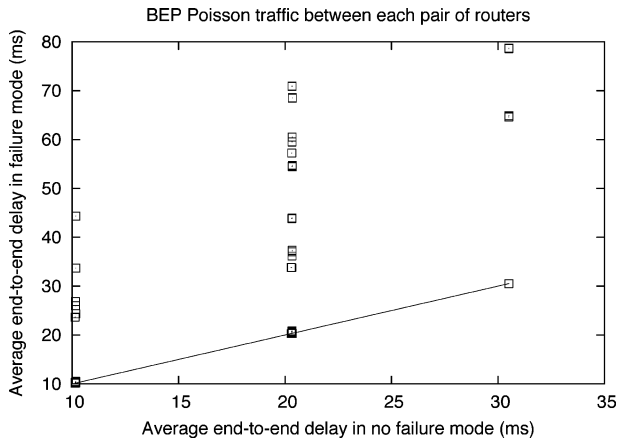


Fig. 17. End-to-end delay for BEP traffic. Failure mode versus no-failure mode.

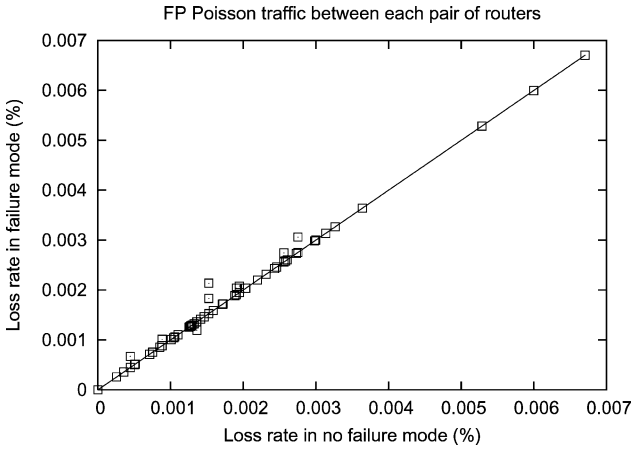


Fig. 18. Losses for FP traffic. Failure mode versus no-failure mode.

For the BEP traffic there is clearly a degradation of service in the failure modes. This is evidenced by the points below the diagonal in the throughput plot[4] and by the points above the diagonal in the delay and loss plots. When throughput drops occur during failure periods, the overall throughput of BEP load is reduced between 30%–60% depending upon the particular failure scenario. Many points in the BEP figures continue to lie around the diagonal which means that some BEP flows are not affected by the corresponding failure and they continue to receive the same service as in the no-failure mode.

Although the BEP traffic can experience a serious degradation at times, we remind the reader of two things. First, the kinds of failures we are talking about are fiber cuts and, thus, it is reasonable to assume that such failures should not happen too often; thus, most of the time the BEP service experiences top quality. Second, some failures are worse than others. The fraction of points above 30 ms (the maximum delay under no failures) for the delay (Fig. 17) is 27%; thus, for the majority of failures, there is little degradation in BEP service. The drawback of occasionally having poor BEP performance for some failure scenarios, is the tradeoff to pay for having a cheaper service. In

[4]The throughput values in this plot range from 0 to 10 Mb/s, while those in Figs. 7 and 8 range up to 1000 Mb/s because in Fig. 15, we plot throughputs per router pair, while in Figs. 7 and 8, BEP traffic is given network-wide (the sum of all router pairs).

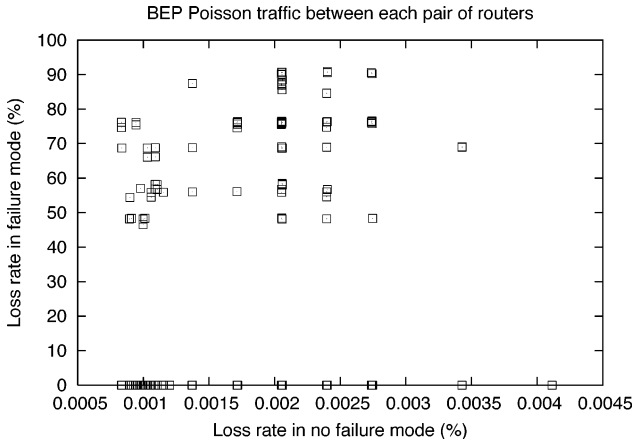


Fig. 19. Losses for BEP traffic. Failure mode versus no-failure mode.

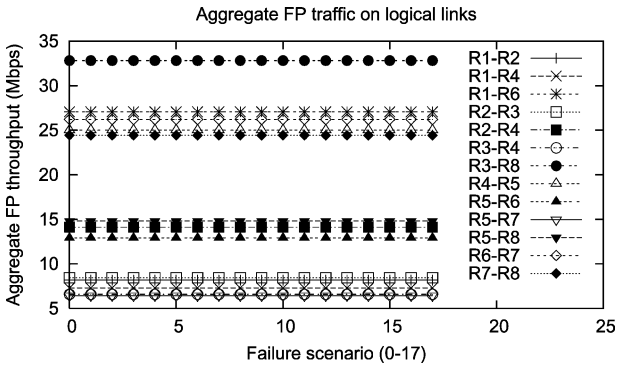


Fig. 20. Average throughput for FP traffic for each failure scenario.

Section II, we mentioned some applications for which BEP is a viable service. We also point out that we used UDP sources in our simulations. If an application were using transmission control protocol (TCP), then it would self-regulate according to the available bandwidth and the loss would be much less than what is shown here. So the BEP performance would be superior in terms of loss and end-to-end delay to what is shown here for TCP-based applications.

We also show the aggregate throughput of FP and BEP traffic on logical links and compare it between the failure mode and the no-failure mode. For each logical link between two neighboring routers and for each failure scenario, we measure the aggregate throughput for both FP and BEP. We plot the results in Fig. 20 for FP traffic and in Fig. 21 for BEP traffic. The x axis in the figures shows the failure scenario number and the y axis the aggregate throughput in megabits per second. The lines in the figures correspond to logical links of the network topology in Fig. 5. Although there are many lines in Fig. 20, it is clear that the aggregate FP throughput remains constant on all logical links for all failure scenarios, and is equal to its value in the no-failure mode (obtained by looking at the y axis for the scenario numbered). This is another metric indicating the success of our mapping and scheduler solutions in terms of not impacting existing customers using the FP service. The 13 logical links in Fig. 21 are displayed over two plots for ease of readability. The aggregate BEP throughput degrades only on some logical links in some failure scenarios, and the amount of BEP degradation

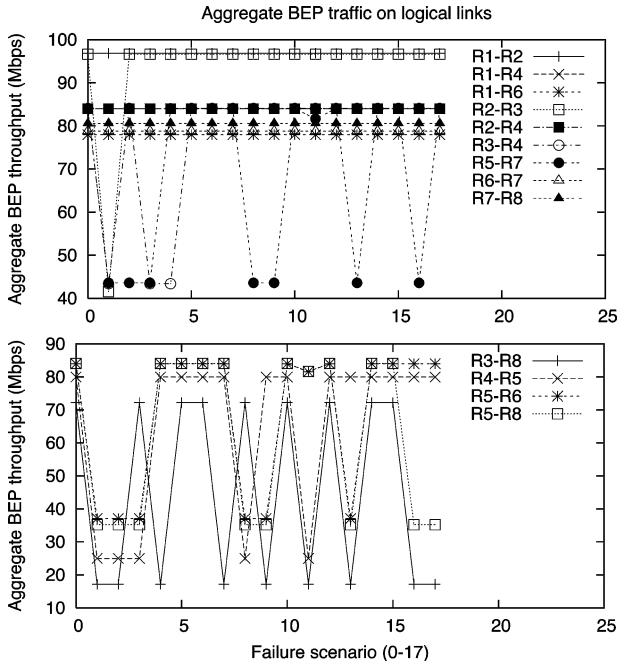


Fig. 21. Average throughput for BEP traffic for each failure scenario.

is dictated by how much bandwidth is available on a logical link after failure.

VII. Conclusion

In this paper, we have solved both a mapping problem and a scheduling problem that carriers would need to resolve in order to support two classes of service differentiated by their level of protection. We illustrated that our heuristic solution, that scales to large networks, performs within 3%–5% of an optimal solution. The multifaceted version of the problem we considered, engenders a variety of important tradeoffs that we illustrated and quantified. For example, we showed that in order to provide service degradation rather than total service disruption, one needs to incorporate failure scenarios inside the optimization steps. However, ensuring that the throughput drops for BEP traffic during failure are limited, also implies that during normal operation the total BEP throughput carried is less than would be if we did not consider failures inside the optimization solution. In the large network scenario we examined, when we include failures in the optimization we carry roughly 8% less BEP traffic than if we do not. However, the gain is that we also drop 22% less BEP during failure episodes than if we did not consider failures. This is clearly worth the tradeoff because even when including failure events, the total load carried by a network (with both FP and BEP services) is roughly a factor of ten more than the load carried by a network supporting FP alone.

Because the pockets of additional bandwidth in carrier networks are usually unevenly distributed, straightforward solutions for offering BEP bandwidth to logical connections would lead to unfair partitions of bandwidth. To compensate, we enforced a max-min fairness policy and showed that this does improve the fairness of the BEP bandwidth partition over simple fairness policies such as a minimum bandwidth allocation. More

importantly, we illustrated that this carrier requirement also induces an important tradeoff on the amount of BEP a network can carry. The more fairness that is required, the less total BEP traffic can be carried. For the two fairness policies we examined, providing max-min fairness instead of a minimum guarantee, means that the BEP traffic load carriable drops by 14%.

Our approach is both practical and complete, because we provide a scalable heuristic that converges quickly and because we provide a scheduling solution for on-line usage. Our combined solution to the mapping and scheduling problems yields a system in which the SLAs of the FP traffic are not affected by the addition of BEP, and the total load carried on backbone networks is increased by a factor from 3 to 10 (depending upon the network scenario considered). We avoided a total disruption in the BEP traffic and limited the degradation to be in the range of a 30%–60% drop in throughput. Thus, BEP users will experience slower connections but not a complete disruption.

In summary, we have illustrated that carrier requirements often lead to restrictions in the total amount of BEP traffic than can be carried. The good news is that even when one meets these load limiting policies, there is still a great deal of BEP traffic than can be carried and, hence, carrier networks contain a large potential to increase their current carried load.

REFERENCES

[1] S. Casner and A. Alaettinoglou, "Detailed analysis of ISIS routing protocol in the QWEST backbone," in *Proc. NANOG Presentation*, Feb. 2002.

[2] G. Iannaccone, C. Chuah, S. Bhattacharyya, and C. Diot, "Analysis of link failures in an IP backbone," in *Proc. ACM Sigcomm Internet Measurement Workshop*, Nov. 2002.

[3] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot, "Characterization of failures in an IP backbone," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004.

[4] A. Autenrieth and A. Kirstdter, "Fault-tolerance and resilience issues in IP-Based networks," in *Proc. 2nd Int. Workshop on the Design of Reliable Communication Networks (DRCN)*, Apr. 2000.

[5] P. Bonenfant, "Optical layer survivability: a comprehensive approach," in *Proc. OFC '98*, vol. 2, San Jose, CA, Feb. 1998, pp. 270–271.

[6] B. Van Caenegem, W. Van Parys, F. De Turck, and P. Demeester, "Dimensioning of survivable WDM networks," *IEEE J. Select. Areas Commun.*, vol. 16, no. 7, pp. 1146–1157, Sept. 1998.

[7] D. Colle *et al.*, "Data-centric optical networks and their survivability," *IEEE J. Select. Areas Commun.*, vol. 20, no. 1, pp. 6–20, Jan. 2002.

[8] P. Demeester *et al.*, "Resilience in a multi-layer network," *IEEE Commun. Mag.*, vol. 37, no. 8, pp. 70–76, Aug. 1999.

[9] H. Zhang and A. Durresi, "Differentiated multi-layer survivability in IP/WDM networks," in *Proc. NOMS'02*, Apr. 2002, pp. 681–694.

[10] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, F. Tobagi, and C. Diot, "Analysis of measured single-hop delay from an operational backbone network," in *Proc. IEEE INFOCOM*, New York, June 2002, pp. 535–544.

[11] J. Moy, "Open shortest path first version 2," *RFC 2178*, July 1997.

[12] *Intermediate System to Intermediate System (IS-IS) Intra-Domain Routing Exchange Protocol*, International Standard 10 589:2002, 2002.

[13] A. Fumagalli and L. Valcarenghi, "IP restoration versus WDM protection: Is there an optimal choice?," *IEEE Network*, pp. 34–41, Nov./Dec. 2000.

[14] O. Gerstel and R. Ramaswami, "Optical layer survivability: A services perspective," *IEEE Commun. Mag.*, vol. 38, pp. 104–113, Mar. 2000.

[15] R. Ramamurthy and B. Mukherjee, "Survivable WDM mesh networks," in *Proc. INFOCOM 1999*, New York, Mar. 1999, pp. 744–751.

[16] G. Mohan and A. K. Somani, "Routing dependable connections with specified failure restoration guarantees in WDM networks," in *Proc. INFOCOM 2000*, Tel-Aviv, Israel, Apr. 2000, pp. 1761–1770.

[17] M. Sridharan and A. K. Somani, "Revenue maximization in survivable WDM networks," in *Proc. Opticomm 2000*, Dallas, TX, Oct. 2000.

[18] A. Nucci, N. Taft, P. Thiran, H. Zhang, and C. Diot, "Increasing link utilization in IP over WDM networks," in *Proc. Opticomm 2002*, Boston, MA, July 2002.

[19] J. Armitage, O. Crochat, and J. Y. Le Boudec, "Design of a survivable WDM photonic network," in *Proc. INFOCOM 1997*, Boston, MA, Apr. 1997, pp. 244–252.

[20] O. Crochat and J. Y. Le Boudec, "Design protection for WDM optical networks," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1158–1165, Sept. 1998.

[21] O. Crochat, J. Y. Le Boudec, and O. Gerstel, "Protection interoperability for WDM optical networks," *IEEE Trans. Networking*, vol. 8, pp. 384–395, June 2000.

[22] E. Modiano and A. Narula-Tam, "Survivable routing of logical topologies in WDM networks," in *Proc. INFOCOM 2001*, vol. 1, Anchorage, AK, Apr. 2001, pp. 348–357.

[23] F. Giroire, A. Nucci, N. Taft, and C. Diot, "Increasing the robustness of IP backbones in the absence of optical level protection," in *Proc. INFOCOM 2003*, San Francisco, CA, Mar. 2003.

[24] A. Nucci, N. Taft, P. Thiran, and C. Diot, "Exploiting failure recovery for the robust support of two service classes in IP over WDM networks," INRIA, Res. Rep. RR-5286, [Online]. Available: http://www.inria.fr/rrrt/rr-5286.html., Aug. 2004.

[25] A. Nucci, B. Sanso, T. G. Crainic, E. Leonardi, and M. A. Marsan, "Design of fault-tolerant logical topologies in wavelength-routed optical IP networks," in *Proc. GLOBECOM 2001*, San Antonio, TX, Nov. 2001.

[26] CPLEX. ILOG CPLEX software optimization suite. [Online]. Available: http://www.ilog.com/products/cplex/

[27] F. Glover and M. Laguna, *Tabu Search*. Norwell, MA: Kluwer, 1997.

[28] G. L. Nemhauser, A. H. G. Rinnoy Kan, and M. J. Todd, *Optimization—Handbooks in Operations Research and Management Science*. Amsterdam, The Netherlands: North-Holland, 1989, vol. 1.

[29] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun, "On the nonstationarity of internet traffic," in *Proc. ACM SIGMETRICS*, 2001, pp. 102–112.

[30] D. Lin and R. Morris, "Dynamics of random early detection," in *Proc. ACM SIGCOMM*, Sept. 1997.

[31] B. Suter, T. V. Lakshman, D. Stiliadis, and A. K. Choudhary, "Design considerations for supporting TCP with per-flow queueing," in *Proc. IEEE INFOCOM*, Mar. 1998, pp. 299–306.

[32] S. Keshav, "A control-theoretic approach to flow control," in *Proc. ACM SIGCOMM*, Sept. 1991.

Antonio Nucci (M'99) received the Dr.Ing. degree in electronics engineering and the Ph.D. degree in telecommunications engineering from the Politecnico di Torino, Turin, Italy, in 1998 and 2002, respectively.

In 1999, he spent four months at the CRT of the Université de Montréal, Montreal, QC, Canada, where he worked with Prof. Theodor Gabriel Crainic and Prof. Brunilde Sansò. He has been a member of the IP research group at the Sprint Advanced Technology Laboratories, Burlingame, CA, since September 2001. His research interests are in traffic characterization, performance evaluation, traffic engineering, and network design.

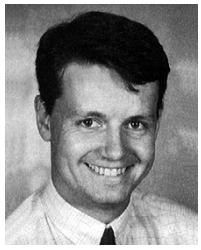
Nina Taft (M'94) received the B.S. degree from the University of Pennsylvania, Philadelphia, in 1985, and the M.S. and Ph.D. degrees from University of California, Berkeley, in 1990 and 1994, respectively.

She is a Senior Researcher with Intel Research Berkeley, Berkeley, CA. From 1999 to 2003, she was a Member of the IP Group, Sprint Advanced Technology Laboratories, Burlingame, CA, working in traffic measurement, characterization and prediction, traffic matrix estimation, IP-over-WDM networks, and performance evaluation. From 1995 to 1999, she was with SRI International working in the areas of congestion control and routing in ATM networks. She is the Tutorial Co-Chair for ICNP 2005, and has served on many program committees including SIGCOMM, INFOCOM, IMC, IWQOS, SIGMETRICS, and Hot Interconnects. Her current interests lie in overlay networks and large-scale statistical security analysis.

Dr. Taft is on the Editorial Board of the IEEE TRANSACTIONS ON NETWORKING.

Chadi Barakat received the electrical and electronics engineering degree from the Lebanese University, Beirut, Lebanon, in 1997, and the M.S. and Ph.D. degrees in networking from the University of Nice, Sophia Antipolis, France, in 1998 and 2001, respectively. His Ph.D. has been done in the Mistral Group, INRIA, Sophia Antipolis.

Since March 2002, he has been a Research Scientist in the Planete Research Group, INRIA. From April 2001 to March 2002, he was with the Laboratory for Computer Communications and their Applications (LCA), Swiss Federal Institute of Technology at Lausanne, Lausanne, Switzerland, where he held a Postdoctoral position, and from March to August 2004, he was a Visiting Faculty Member at Intel Research, Cambridge. He was the General Chair of PAM 2004 and serves on the program committees of many international conferences such as INFOCOM, PAM, WONS, ASWN, and GLOBECOM. His main research interests are congestion and error control in computer networks, the TCP protocol, voice-over-IP, wireless LANs, Internet measurement and traffic analysis, and performance evaluation of communication protocols.



Patrick Thiran (S'88-M'90) received the electrical engineering degree from the Université Catholique de Louvain, Louvain-la-Neuve, Belgium, in 1989, the M.S. degree in electrical engineering from the University of California, Berkeley, in 1990, and the Ph.D. degree from the Swiss Federal Institute of Technology at Lausanne (EPFL), Lausanne, Switzerland, in 1996.

He became a Professor at EPFL in 1998, and was on leave with Sprint Advanced Technology Laboratories, Burlingame, CA, from 2000 to 2001. His research interests are in communication networks, performance analysis, and dynamical systems.

Dr. Thiran received the 1996 EPFL Doctoral Prize. He served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1997 to 1999.