University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

CSE Conference and Workshop Papers | Computer Science and Engineering, Department of

1-1-2004

A load-balancing spare capacity reallocation approach in service-rich SONET metro mesh networks

Lu Shen
University of Nebraska - Lincoln, lshen@cse.unl.edu

Xi Yang
University of Nebraska - Lincoln, xyang@cse.unl.edu

Byrav Ramamurthy
University of Nebraska - Lincoln, bramamurthy2@unl.edu

Follow this and additional works at: http://digitalcommons.unl.edu/cseconfwork

Part of the Computer Sciences Commons

Shen, Lu; Yang, Xi; and Ramamurthy, Byrav, "A load-balancing spare capacity reallocation approach in service-rich SONET metro mesh networks" (2004). *CSE Conference and Workshop Papers*. Paper 79.
http://digitalcommons.unl.edu/cseconfwork/79

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Conference and Workshop Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

A load-balancing spare capacity reallocation approach in service-rich SONET metro mesh networks†

Lu Shen, Xi Yang and Byrav Ramamurthy
Department of Computer Science and Engineering
University of Nebraska – Lincoln
Lincoln NE 68588-0115
Email: {lshen, xyang, byrav}@cse.unl.edu)

Abstract

The next-generation SONET metro network is evolving into a service-rich infrastructure. At the edge of such a network, multi-service provisioning platforms (MSPPs) provide efficient data mapping enabled by Generic Framing Procedure (GFP) and Virtual Concatenation (VC). The core of the network tends to be a meshed architecture equipped with Multi-Service Switches (MSSs). In the context of these emerging technologies, we propose a load-balancing spare capacity reallocation approach to improve network utilization in the next-generation SONET metro networks. Using our approach, carriers can postpone network upgrades, resulting in increased revenue with reduced capital expenditures (CAPEX). For the first time, we consider the spare capacity reallocation problem from a capacity upgrade and network planning perspective. Our approach can operate in the context of shared-path protection (with backup multiplexing) because it reallocates spare capacity without disrupting working services. Unlike previous spare capacity reallocation approaches which aim at minimizing total spare capacity, our load-balancing approach minimizes the network load vector (NLV), which is a novel metric that reflects the network load distribution. Because NLV takes into consideration both uniform and non-uniform link capacity distribution, our approach can benefit both uniform and non-uniform networks. We develop a greedy load-balancing spare capacity reallocation (GLB-SCR) heuristic algorithm to implement this approach. Our experimental results show that GLB-SCR outperforms a previously proposed algorithm (SSR) in terms of established connection capacity and total network capacity in both uniform and non-uniform networks.

1. Introduction

The next-generation SONET metro network will serve as a service-rich infrastructure that responds quickly and efficiently to emerging data services [1][2][3]. In such a network, multi-service provisioning platforms (MSPPs) are deployed at the edge where efficient data mapping is enabled by Generic Framing Procedure (GFP) and Virtual Concatenation (VC) technologies [1][2][3][4]. Virtual concatenation, which is standardized and widely accepted in industry [2][3][5][6], is an inverse-multiplexing technology that groups multiple base-rate SONET circuits (e.g., VT-1.5 or STS-1) and eliminates the continuous constraints on conventional SONET multiplexing hierarchy. Equipped with multi-service switches (MSSs) [1], the metro core network is evolving from ring to mesh topologies that provide enhanced routing flexibility and resource utilization. However, due to high traffic uncertainty in metro networks, leveraging new technologies to achieve efficient network resource utilization remains a challenge.

Provisioning in real-world networks is usually implemented without the knowledge of how network resources will be used in the future. As connections are added and deleted from the network, the network resource utilization becomes sub-optimal [7][8][9]. This problem becomes more severe in the next-generation metro networks, where traffic is data-dominant and tends to be more unpredictable. Bandwidth reallocation, a method of reconfiguring the existing connections, is an attractive solution to fill in the gap between current and optimal of network resource utilization. [8][9].

As the metro network becomes a bottleneck in the entire network, carriers feel pressed to provide more network capacity. When bandwidth demands exceed network capacity, carriers need to upgrade their network by deploying extra equipment such as fibers, wavelengths, transceivers and switching fabrics. Due to uncertainty of future traffic volume and equipment costs, carriers need a flexible, incremental capacity upgrade solution to avoid over-investment on equipment [10][11]. Taking capacity planning into account, we propose a load-balancing spare capacity reallocation approach to postpone network

† This work was supported in part by NSF grants (ANI-0074121 and EPS-0091900).

upgrades, resulting in substantial cost savings. Because our approach provides a flexible and efficient capacity upgrade solution to handle traffic uncertainty, carriers may incorporate it into their network planning tools to maintain robustness against unpredictability, which is an important requirement for network planning tools in next-generation multi-service networks [11]. To the best of our knowledge, our work, for the first time, studies the bandwidth reallocation problem from a capacity planning and network upgrade perspective. Our proposed spare capacity reallocation approach only changes protection paths, and hence does not interfere with the ongoing services on the working paths and is therefore risk-free. Our approach is particular suitable for the networks with shared-path protection, which has proven to be cost-efficient in the next generation metro mesh networks [12][13].

Unlike previous spare capacity reallocation approaches whose objective is minimizing the total spare capacity, our load-balancing approach reconfigures the existing protection paths with the objective of minimizing the network load vector (NLV). NLV is a new metric defined in this paper to reflect the network load distribution. By minimizing NLV, the network load distribution is changed to accommodate more future connections before a network upgrade is needed. Network upgrade is usually triggered based on carriers' policies, e.g., when the number of congested links exceeds a threshold or when a provisioning attempt fails on a connection request. After an upgrade, the network link capacity increases. If all the links in a network have the same amount of capacity, the network is called a *uniform network*; otherwise, it is called a *non-uniform network*. Metro networks tend to be non-uniform for two reasons: (a) the links between major traffic-aggregation locations (e.g., major inter-offices) need more capacity, and (b) carriers may choose to upgrade only congested links due to budget crunch, resulting in non-uniform distribution of link capacities. (This kind of upgrade is called a *non-uniform upgrade* in this paper.) The conventional spare capacity reallocation approaches aim at minimizing spare capacity, but cannot deal with the non-uniform distribution of link capacity. In contrast, the definition of NLV takes into consideration the non-uniform distribution of link capacity. Therefore our load-balancing approach can benefit both uniform and non-uniform networks.

Previous studies on the spare capacity reallocation problem include [8][13][15]. The work in [13] proposed a distributed spare capacity optimization approach in WDM optical networks. The authors in [15] presented a matrix-based spare capacity allocation approach called successive survivable routing (SSR). Both the approaches in [13] and [15] re-provision their protection paths iteratively to minimize total spare capacity. Each iteration in their approaches uses the dynamic protection path selection approach to reroute protection path. State-of-the-art dynamic protection path selection schemes include partial information restoration (PIR) [14], full information restoration (FIR) [18], and distributed partial information management (DPIM) [16][17]. All of these previous approaches focus on minimizing spare capacity. However, this objective does not directly relate to cost savings because minimizing spare capacity does not necessarily maximize the capability of accommodating future connections. Thus, service providers may still need to pay for early network upgrades. Our previous study in [19] proposed a load-balancing shared-protection-path reconfiguration approach for wavelength-routed networks, but did not take into consideration the non-uniform link load distribution.

In this study, we develop a greedy load-balancing spare capacity reallocation (GLB-SCR) heuristic algorithm to minimize NLV. We conduct substantial experiments to simulate the service-rich environment in next-generation SONET metro networks and compare GLB-SCR with SSR, a conventional approach introduced in [15]. The experimental results show that GLB-SCR can establish more connections under the same amount of total network capacity in uniform networks. In non-uniform networks, GLB-SCR obtains constant delays in network upgrades, resulting in significant cost savings on network capacity investments.

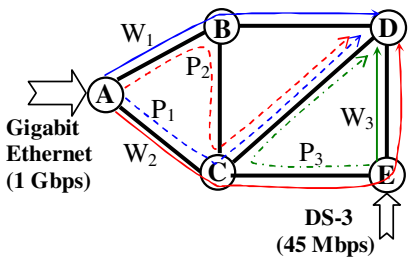
The remainder of this paper is organized as follows. Section 2 presents the network model and gives the problem formulation. Section 3 describes our approach to solve this problem. Section 4 presents the experimental design and numerical results. Section 5 concludes this paper.

2. Network model and problem formulation

2.1 Network model

In this paper, we consider next-generation SONET mesh networks. In such a network, nodes are interconnected by a set of fibers, each having a certain amount of link capacity. Different fibers are allowed to have different link capacities. Each network node is capable of cross-connecting circuits at a base rate (e.g., STS-1), which is called switching granularity. Capacity is represented by an integer with the unit of switching granularity. For example, in a network with an STS-1 switching granularity, link capacity 48 represents a capacity of 48 STS-1, i.e., OC-48. Each link in the network is associated with a working capacity, used for working services, and a spare capacity, used for protection. The sum of the working capacity and the spare capacity of a link is the used capacity of this link. The free capacity on a link can be obtained by subtracting the used capacity from its link capacity. To calculate spare capacity, we maintain two arrays, called FailOther and FailSelf, for each link. FailOther records the amount of

IEEE COMPUTER SOCIETY



Routing Information					
Connection	Bandwidth Requirement	ID of Sub-connection	Bandwidth Capacity	Working Path	Protection Path
Gigabit Ethernet	1 Gbps	1	15 STS-1	W1 (A-B-D)	P1 (A-C-D)
		2	6 STS-1	W2 (A-C-E-D)	P2 (A-B-C-D)
DS-3	45 Mbps	1	1 STS-1	W3 (E-D)	P3 (E-C-D)

Spare Capacity Information			
Link	Spare Capacity (STS-1)	Link	Spare Capacity (STS-1)
A-B	6	C-D	15
A-C	15	E-C	1
B-C	6	other	0

Fig. 1 An illustration of the network model.

spare capacity needed on the link to restore working services upon failure of other links. FailSelf records the amount of spare capacity needed on other links to restore working services upon failure of this link. These two arrays were originally introduced in [18] for protection path selection. The definition of each member in these arrays will be given in Section 3.1.

A user service on the network is represented by a connection. A connection comprises a source, a destination, and a bandwidth requirement. We assume that every node has virtual concatenation functionality. In other words, the source node is capable of dividing a connection into multiple circuits at the rate of switching granularity, and the destination node is capable of combining these circuits to reconstruct the original connection. These circuits are called virtual concatenation (VC) group members. They can be routed, cross-connected, and protected independently in the network. Each VC group member has shared-path protection against single-link failures. In particular, the spare capacity is assigned such that, under a single-link failure, all the disrupted working circuits that traverse the failed link can be restored. The VC group members of a connection that traverse through the same pair of working and protection paths are called a sub-connection of this connection. In summary, a connection is divided into multiple of sub-connections, each of which contains multiple VC group members (circuits) that traverse through the same pair of link-disjoint protection and working paths. Each of these VC group members (circuits) operates at the rate of switching granularity.

Fig. 1 illustrates our network model. The example network consists of 5 nodes and 14 unidirectional links. Two connections exist in the network: a Gigabit Ethernet (GbE) connection with bandwidth requirement of 1 Gbps, and a DS-3 connection with bandwidth requirement of 45 Mbps. Under virtual concatenation at node A, the GbE connection is decomposed to 21 STS-1 VC group members (i.e., STS-1-21v). These 21 VC group members are further divided into two sub-connections that traverse two working paths, W1 with 15 STS-1 circuits and W2 with 6 STS-1 circuits, each having a protection path (P1 and P2 respectively). They are combined at node D to reconstruct the original GbE connection. Fig. 1 also shows the spare capacity information. Note that link (C, D) only needs to reserve a spare capacity of 15 STS-1, although all three protection paths traverse through it. This is because the three working paths will not break down simultaneously under any single-link failure.

2.2 Problem formulation

Our problem is formulated as folllows: *Given a network topology, the capacity information of each link, and a set of existing connections, determine how to reallocate spare capacity with the objective of minimizing the network load vector (NLV).*

Network load vector (NLV) is a novel network performance metric introduced in this paper. Let $G(V,E)$ denote the network topology, where V is the set of nodes and E is the set of links. Let C_{ab} denote the link capacity of the link (a,b), $C = \max_{\forall(a,b)\in E}\{C_{ab}\}$ denote the maximum link capacity among all the links, and W_{ab} and S_{ab} denote the working capacity and spare capacity on the link (a,b) respectively. Note that C_{ab}, W_{ab}, and S_{ab} are integers with the unit of switching granularity. We define the NLV of network G as a vector: $L_G = [l_0,\ldots,l_i,\ldots,l_C]$, where i is an integer, and $l_i = |\{(a,b) : \lfloor C\cdot(W_{ab}+S_{ab})/C_{ab}\rfloor = i\}|$. In other words, l_i denotes the number of links that satisfies the equation $\lfloor C\cdot(W_{ab}+S_{ab})/C_{ab}\rfloor = i$, where $(W_{ab}+S_{ab})$ is the used capacity on the link (a,b). $\lfloor C\cdot(W_{ab}+S_{ab})/C_{ab}\rfloor$ is the normalized used capacity that is scaled to C levels according to the percentage of the used capacity over the link capacity (i.e., $(W_{ab}+S_{ab})/C_{ab}$) on the local link. In this paper, link load refers to this normalized capacity. The index ($0 \le i \le C$) of a NLV represents a specific value of the link load. The value of a member in NLV is the number of links whose load equals its index. The definition of NLV applies to both uniform and non-uniform networks. Note that in the uniform networks, l_i becomes the number of links with used capacity $(W_{ab}+S_{ab}) = i$ because $C = C_{ab}, \forall(a,b)\in E$.

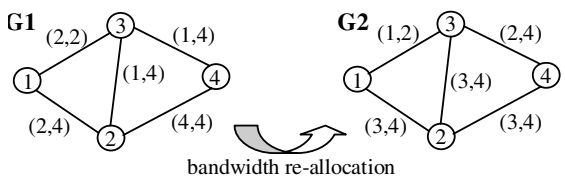
We define the relationship between two NLVs, L_{G1} and L_{G2}, as follows:

- $L_{G1} = L_{G2}$, if $L_{G1}[i] = L_{G2}[i]$, $\forall i : 0 \le i \le C$.

IEEE COMPUTER SOCIETY

- $L_{G1} > L_{G2}$, if $\exists j: 0 \le j \le C$, such that $L_{G1}[j] > L_{G2}[j]$ and $L_{G1}[i] = L_{G2}[i]$, $\forall i: 0 \le j < i \le C$.
- $L_{G1} < L_{G2}$, if $\exists j: 0 \le j \le C$, such that $L_{G1}[j] < L_{G2}[j]$ and $L_{G1}[i] = L_{G2}[i]$, $\forall i: 0 \le j < i \le C$.

Note that $L_G[i]$ denotes l_i in L_G and i is an integer. By this definition, reducing an NLV means moving some used capacity on the links with higher loads to the links with lower loads, resulting in a less congested network. Thus, the load-balancing objective can be achieved by minimizing the NLV of a network through re-provisioning the existing protection circuits. In addition, this definition can facilitate the non-uniform network upgrade, because it reduces the number of congested links that may need to be upgraded, resulting in CAPEX savings.



NLV changes from: $L_{G1} = [l_0 = 0,\ l_1 = 2,\ l_2 = 1,\ l_3 = 0,\ l_4 = 2]$

to: $L_{G2} = [l_0 = 0,\ l_1 = 0,\ l_2 = 2,\ l_3 = 3,\ l_4 = 0]$.

$L_{G2} < L_{G1}$, because $L_{G2}[4] < L_{G1}[4]$ ($L_G[4]$ denote l_4 in L_G).

Fig. 2 An example of NLV.

Fig. 2 shows an example on how to calculate the NLV of a network, how to evaluate the relationship between two NLVs, and how a network benefits from reducing its NLV. For simplicity of presentation, we assume that the network is bi-directional in this example. Each link in the network is associated with two numbers, $(W_{ij} + S_{ij}, C_{ij})$, which denote the used capacity and link capacity respectively. By reallocating the used capacity on some links, we change G1 to G2. After this change, although the total number of used capacity increases from 10 to 12, NLV is reduced (note that $L_{G2} < L_{G1}$). In the new graph G2, we can find a pair of link-disjoint working and protection paths (i.e., $(1-2-4)$ and $(1-3-4)$) from node 1 to node 4 with some free capacities, which cannot be found in G1 before the reallocation.

3. Load-balancing spare capacity reallocation approach

We propose a greedy heuristic algorithm to achieve our load-balancing objective (i.e., minimizing NLV) through spare capacity reallocation. The heuristic operates on sub-connections because all the VC group members (circuits) belonging to a sub-connection traverse the same pair of working and protection paths. This is also to satisfy the constraint on the number of diverse paths (i.e., sub-connections) of each connection, which is usually imposed by the implementation of virtual concatenation [5]. By treating each sub-connection as one operational unit, the number of sub-connections (diverse paths) remains the same after reallocation. In this section, we first define the notations to be used. We then introduce the spare capacity allocation and release procedures for operating one sub-connection, which are the basic operations in our heuristic. In the remaining two subsections, we present the design of our heuristic algorithm and discuss some operational considerations for our approach.

3.1 Notations

- $G(V,E)$ denotes the network topology with a set of nodes V and a set of unidirectional links E.
- $N = |V|$ denotes the number of nodes in the network.
- C_{ij} is an integer that denotes the link capacity of the link $(i,j) \in E$.
- $C = \max_{(i,j)\in E}\{C_{ij}\}$ is an integer that denotes the maximum link capacity among all the link in the network.
- W_{ij} is an integer that denotes the working capacity on the link $(i,j) \in E$.
- S_{ij} is an integer that denotes the spare capacity reserved on the link $(i,j) \in E$.
- F_{ij} is an integer that denotes the free capacity on the link $(i,j) \in E$, where $F_{ij} = C_{ij} - W_{ij} - S_{ij}$.
- $FailSelf_{ij}(x,y)$ is an integer that denotes the spare capacity needed on the link (x,y) in order to protect the working services upon failure of the link (i,j), where $(i,j) \in E$ and $(x,y) \in E$. In other words, $FailSelf_{ij}(x,y)$ is the sum of the capacity of all the sub-connections whose working paths traverse the link (i,j) and protection paths traverse the link (x,y).
- $FailOther_{ij}(x,y)$ is an integer that denotes the spare capacity needed on the link (i,j) in order to protect the working services upon failure of the link (x,y), where $(i,j) \in E$ and $(x,y) \in E$. In other words, $FailOther_{ij}(x,y)$ is the sum of the capacity of all the sub-connections whose working paths traverse the link (x,y) and protection paths traverse the link (i,j). Note that $S_{ij} = \max_{(x,y)\in E}\{FailOther_{ij}(x,y)\}$ in order to protect the working services against any single-link failure.

3.2 Spare capacity allocation and release procedures

Spare capacity allocation (SC-Allocation) and release (SC-Release) procedures are the basic operations performed on a sub-connection in our heuristic. We use (s,d,b,W,P) to present a sub-connection, where s denotes its source

IEEE COMPUTER SOCIETY

node, d denotes its destination node, b denotes its capacity (or the number of VC group members), W denotes its working path, and P denotes its protection path. The spare capacity allocation and release procedures update the link information along the working and protection paths. The information includes spare capacity, free capacity and the members in the arrays FailSelf and FailOther.

SC-Allocation($G,(s,d,b,W,P)$)

1) Update the working path
   for each link $(i,j) \in W$, **do**
   - **for** each link $(x,y) \in P$, **do**
     - $FailSelf_{ij}(x,y) \leftarrow FailSelf_{ij}(x,y) + b$;

2) Update the protection path
   for each link $(i,j) \in P$, **do**
   - **for** each link $(x,y) \in W$, **do**
     - $FailOther_{ij}(x,y) \leftarrow FailOther_{ij}(x,y) + b$;
   - $S_{ij} \leftarrow \max_{(x,y)\in E} \{FailOther_{ij}(x,y)\}$;

(a) Spare capacity allocation procedure

SC-Release($G,(s,d,b,W,P)$)

1) Assign bandwidth on the working path
   for each link $(i,j) \in W$, **do**
   - **for** each link $(x,y) \in P$, **do**
     - $FailSelf_{ij}(x,y) \leftarrow FailSelf_{ij}(x,y) - b$;

2) Assign bandwidth on the protection path
   for each link $(i,j) \in P$, **do**
   - **for** each link $(x,y) \in W$, **do**
     - $FailOther_{ij}(x,y) \leftarrow FailOther_{ij}(x,y) - b$;
   - $S_{ij} \leftarrow \max_{(x,y)\in E} \{FailOther_{ij}(x,y)\}$;

(a) Spare capacity release procedure

Fig. 3 SC allocation and release procedures.

Fig. 3 (a) and (b) present the SC-Allocation and SC-Release procedures respectively. The input for both procedures is the network topology along with the link capacity information and a sub-connection. In the SC-Allocation procedure, $FailSelf_{ij}(x,y)$ is incremented by the value of b if the link (i,j) is on the working path and the link (x,y) is on the protection path of this sub-connection. $FailOther_{ij}(x,y)$ is incremented by the value of b if (i,j) is on the protection path and (x,y) is on the working path. Then, the spare capacity of the link (i,j) along the protection path is set to the maximum value of $FailOther_{ij}(x,y)$ among all the links $(x,y) \in E$. The SC-Release procedure has steps similar to the SC-Allocation procedure, but it decreases the corresponding members in the arrays by the value of b .

3.3 Greedy load-balancing spare capacity reallocation (GLB-SCR) heuristic

We develop a greedy load-balancing spare capacity reallocation (GLB-SCR) heuristic, which iteratively reallocates all the sub-connections one by one until the NLV of the network cannot be reduced. Fig. 4 presents the procedure of the GLB-SCR heuristic, in which a spare capacity reallocation (SCR) procedure is called iteratively.

Greedy Load-Balancing Spare Capacity Reallocation (GLB-SCR)

repeat←1.
while *repeat* = 1 **do**

1. *repeat*←0.
2. insert all the sub-connections to a list according to the non-decreasing order of the maximum link load on their protection path and their capacities (the link load takes priority over capacity when sorting).
3. **for** each sub-connection in the list, denoted by (s,d,b,W,P) , **do**
   a. SCR($G,(s,d,b,W,P)$) /* spare capacity reallocation */
   b. **if** the *SCR* procedure returns success, **then** *repeat*←1.
4. empty *the* list of sub-connections.

Fig. 4 Greedy load-balancing spare capacity reallocation heuristic.

SCR($G,(s,d,b,W,P)$)

1) $NLV_O \leftarrow$ the NLV of the network G

/ find the maximum link load on the protection path */*

2) $\max_load \leftarrow \max_{(i,j)\in P} \{(W_{ij} + S_{ij}) \cdot C / C_{ij}\}$

3) *SC-Release*(G,s,d,b,W,P); */* in Fig. 3 (b) */*

*/*construct a weighted graph (shown in* Fig. 6*)*/*

4) $G_W \leftarrow$ least-NLV-weight(*max_load*, (s,d,b,W,P));

5) $P_N \leftarrow$ the shortest path on the weighted graph G_W .

6) *SC-Allocation*($G,(s,d,b,W,P_N)$); */* in Fig. 3 (a) */*

7) $NLV_N \leftarrow$ the NLV of the network G

8) **if** $NLV_N < NLV_O$, **then**
   a. update the sub-connection by changing its protection path P to P_N ;
   b. return *success*;

9) **if** $NLV_N \geq NLV_O$, **then**
   a. *SC*-Release($G,(s,d,b,W,P_N)$);
   b. *SC*-Allocation($G,(s,d,b,W,P)$);
   c. return *failure*;

Fig. 5 Spare capacity reallocation (SCR) procedure.

SCR is the key procedure in our approach. It tries to reallocate the protection path of an existing sub-connection to reduce the NLV of the network. Its input is the network

IEEE COMPUTER SOCIETY

topology with link capacity usage information, and an existing sub-connection. If it successfully reduces NLV, it returns success. Otherwise it returns failure. In the SCR procedure, the spare capacity on the existing protection path of the sub-connection is released by the SC-Release procedure (shown in Fig. 3). Then, without changing its working path, a new protection path (P_N) is selected by calculating the shortest path on a weighted graph. In the weighted graph, which has the same topology as the input graph G, each link is assigned a weight according to the least-NLV weight function (shown in Fig. 6). Then, the spare capacity of the sub-connection is re-allocated based on its original working path and the new protection path. If the NLV of the network is reduced, the sub-connection is changed by replacing the new protection path with its original protection path and a success is returned. If the NLV is not reduced, the sub-connection is not changed, the protection path of this sub-connection is restored to its original one and a failure is returned. The time complexity of SCR is $O(N^2)$, where N is the number of nodes in the network.

Given the network topology, a sub-connection (s,d,b,W,P) and the maximum link load (i.e., *max_load* calculated in the step 2 of SCR) among all the links on the original protection path of this sub-connection, the least-NLV weight function returns a weighted graph which is used for protection path selection in SCR. In the weighted graph, a weight, denoted by $weight(i,j)$, is assigned to each link (i,j) to facilitate selecting a new protection path that tends to reduce NLV. Before giving the definition of the least-NLV weight function, we further define the notation: $T(i,j) = \max_{(x,y)\in W}\{FailSelf_{xy}(i,j)\}$ is the spare capacity needed on the link (i,j) to protect the working path W. Note that the weighted graph is computed after the spare capacity on the original protection path is released (i.e., after the step 3 in SCR). Thus, $T(i,j)$ is the spare capacity needed on the link (i,j) to protect the working path of this sub-connection if the new protection path traverses through the link (i,j). When $T(i,j) > S_{ij}$, an amount of $(T(i,j) - S_{ij})$ extra spare capacity needs to be reserved on the link (i,j).

The least-NLV weight function is defined in Fig. 6, where $weight(i,j)$ is computed according to the following principles: (1) if the link (i,j) is on the working path, its weight is set to infinity to satisfy the link-diversity constraint on the working and protection path pairs; (2) if the link (i,j) is not on the working path, and its spare capacity needs to be increased, but the free capacity is not sufficient to satisfy the incremental spare capacity, its weight is set to infinity because of lack of the spare capacity; (3) if the link is not on the working path, and its spare capacity needs to be increased, and the free capacity is sufficient for the incremental spare capacity, and its future link load is greater or equal to max_load, and the link (i,j) is not on the original protection path, its weight is set to infinity. This weight is set to detour the new protection path from the links with maximum link load on the original protection paths, resulting in a reduced NLV; (4-5) for all other links, their weight is set to the power of the number of the nodes in the graph (N) to its future link load.

The link weight, which is the power of N to the link load, leads to a minimum NLV when selecting the shortest path. This is because the maximum number of links on the shortest path is $N-1$, resulting in the inequality $N^i > (N)^{i-1}\cdot(N-1)$. Although such a weight is exponential to N, the time complexity of computing integer exponentiation is polynomial under the square-and-multiply method. However, this kind of weight is a large number that may overflow any conventional data type in computer languages. In our implementation, we use the GNU Multiple Precision (GMP) arithmetic library to manipulate these large numbers [20].

3.4 Operational considerations

In Section 3.3, we have presented a load-balancing spare capacity reallocation heuristic. In this sub-section, we discuss how to implement our spare capacity reallocation approach in real-world networks based on this heuristic. We focus on a centralized implementation, but our approach can be both centralized and distributed. A centralized network operator maintains the network

$$weight(i,j) = \begin{cases} \infty, & if\ (i,j)\in W \quad (1) \\ \infty, & if\ (i,j)\notin W \text{ and } T(i,j) > S_{ij} \text{ and } (T(i,j)-S_{ij}) > F_{ij} \quad (2) \\ \infty, & if\ (i,j)\notin W \text{ and } T(i,j) > S_{ij} \text{ and } (T(i,j)-S_{ij}) \le F_{ij} \text{ and } \max_load \le \lfloor (T(i,j)+W_{ij})\cdot C/C_{ij} \rfloor \text{ and } (i,j)\notin P \quad (3) \\ N^{\left\lfloor \frac{(W_{ij}+T(i,j))\cdot C_{ij}}{C} \right\rfloor}, & if\ (i,j)\notin W \text{ and } T(i,j) > S_{ij} \text{ and } (T(i,j)-S_{ij}) \le F_{ij} \text{ and } \max_load \le \lfloor (T(i,j)+W_{ij})\cdot C/C_{ij} \rfloor \text{ and } (i,j)\in P \quad (4) \\ N^{\left\lfloor \frac{(W_{ij}+S_{ij})\cdot C_{ij}}{C} \right\rfloor}, & others \quad (5) \end{cases}$$

Fig. 6 The least-NLV weight function.

IEEE COMPUTER SOCIETY

topology, capacity information on each link, and the connection information. When the spare capacity reallocation is triggered, the operator runs the GLB-SCR heuristic and records a list of sub-connections to be changed. After the heuristic ends, the operator starts to reallocate the protection paths of the sub-connections stored in the list one by one. The NLV of the network is reduced after every reallocation step due to the greedy nature of GLB-SCR. Therefore, there is no risk even if the reallocation procedure does not finish and only a portion of the sub-connections in the list is reallocated.

The GLB-SCR procedure can be triggered on a periodic basis or by a special event, e.g., a connection blocking occurs or the number of links whose link load becomes greater than a pre-specified value reaches a threshold. However, the reallocation procedure should not be invoked too frequently because it increases operational costs. Carriers can choose their own operational policies to trigger the GLB-SCR procedure.

Our proposed approach can be integrated into carriers' network planning tools, which help carriers design network topology, optimize network utilization, and plan for network upgrade [10][11]. By reallocating spare capacity, our approach forms a load-balancing network, which can accommodate more future connections to postpone network upgrades. In the case of a non-uniform upgrade, which only increases the capacity of a portion of links in a network, our approach brings extra benefits by taking the normalized link load into account.

4. Illustrative Numerical Results

We conduct experiments in a mesh network, which has 24 nodes and 86 unidirectional links as shown in Fig. 7 In the experiments, we compare the GLB-SCR approach with the successive survivable routing (SSR) approach proposed in [15]. SSR also reallocates spare capacity iteratively, but aims at minimizing the total spare capacity of the network. At the beginning of our experiments, each link in the network has an initial capacity (IC). During the experiments, we provision connection requests one-by-one until blocking occurs. Then, the spare capacity reallocation procedure is triggered to operate on all existing connections. After the reallocation, we continue to provision the connection request that was previously blocked. If it is still blocked, we upgrade the link capacity by the amount of upgrade capacity (UC). We consider both cases of uniform and non-uniform networks. In uniform networks, every link is upgraded by the amount of UC capacity. In non-uniform networks, only the links with maximum link load are upgraded by the amount of UC.

We assume the network has an STS-1 switching granularity. To simulate traffic in the real-world metro networks, we provide various connection requests with bandwidth requirements ranging from 45Mbps to 1 Gbps. Table 1 lists different connection requests with their bandwidth requirements, service type, capacity values with the unit of switching granularity (i.e., STS-1), and arrival distribution. The service types and their arrival distributions are close to those in practical networks [1][2][6]. In the experiments, connection requests are uniformly distributed among all node pairs and arrive in a random order. For simplicity of experiments, we assume that connections are semi-dynamic, i.e., a connection will not be released once provisioned. To provision a connection request, we use the working path selection approach proposed in [5] and use the protection path selection scheme (called FIR) introduced in [18]. We set the number 16 as the limit on the number of sub-connections (diverse-paths) for each connection (our experiments show that the maximum number of sub-connections does not exceed 6 under our provisioning method). We use two metrics for comparison: established connection capacity and total network capacity. Established connection capacity is the sum of the capacities of all the established connections. The total network capacity is sum of the link capacities of all the links in the network.

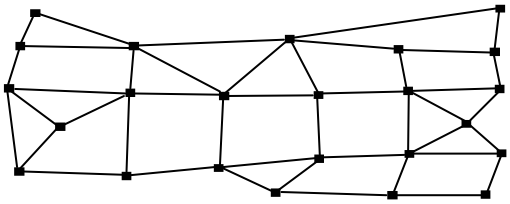


Fig. 7 The 24-node and 86-link network.

Table 1 Service types and their distribution

Bandwidth Requirement	Service Type	# of STS-1	Distribution
45Mbps	DS-3	1	100 (41%)
	Aggregated lower streams, e.g., DS-1		
100 Mbps	Fast Ethernet	2	50 (20%)
150 Mbps	STS-3c	3	50 (20%)
160 Mbps	ESCON	4	20 (8%)
400 Mbps	Gigabit Ethernet (at a lower average rate)	8	10 (4%)
600 Mbps	ATM or Fast Internet Router Connection	12	10 (4%)
850 Mbps	Fiber Channel or FICON	18	2 (0.8%)
1 Gbps	Gigabit Ethernet (at its peak rate)	21	1 (0.4%)

In this section, we first present the results obtained for the cases of a uniform network. Then, we compare these two approaches for the cases of a non-uniform network. At the end of this section, we compare the network utilization of uniform and non-uniform upgrade

4.1 Results for uniform networks

In this sub-section, we present experimental results for the cases of a uniform network. We conduct two sets of experiments, one with an initial capacity of 48 (i.e., line rate of OC-48) and an upgrade capacity (UC) of 48, and the other with an initial capacity of 192 (i.e., line rate of OC-192) and an upgrade capacity of 192. Table 2 and Table 3 show the results of the two cases respectively. Both tables

IEEE COMPUTER SOCIETY

list the link capacity of the network, the established connection capacity (of all existing connections) before the link capacity is upgraded to the next higher level, the improvement of GLB-SCR over SSR in terms of established connection capacity under the same amount of link capacity, and the number of triggered reallocation events. The results show that the network with GLB-SCR can establish more connections in terms of the established connection capacity than the one with SSR. On average, the network with GLB-SCR can accommodate 2.4% and 1% more connections (in terms of the established connection capacity) in the two cases respectively. As shown in the tables, the network with GLB-SCR invokes a few more reallocation events to obtain this improvement, which translates into increased revenue for carriers.

Table 2 The results for the uniform network when IC=48 and UC=48

Initial Capacity (IC)=48 & Upgrade Capacity (UC)=48					
Link Capacity (STS-1)	Established connection capacity (STS-1)		Improvement of GLB-SCR over SSR	# of Reallocation Triggers	
	SSR	GLB-SCR	Average=2.4%	SSR	GLB-SCR
48	504	508	0.8%	1	3
96	1026	1038	0.2%	1	2
144	1432	1508	5.3%	1	2
192	2028	2158	6.4%	1	2
240	2480	2480	0%	1	1
288	3250	3308	1.8%	1	2

Table 3 The results for the uniform network when IC=192 and UC=192.

Initial Capacity (IC)=192 & Upgrade Capacity (UC)=192					
Link Capacity (STS-1)	Established connection capacity (STS-1)		Improvement of GLB-SCR over SSR	# of Reallocation Triggers	
	SSR	GLB-SCR	Average=1%	SSR	GLB-SCR
192	2036	2143	5.3%	1	2
384	4855	4870	0.3%	3	3
576	7475	7475	0%	1	2
768	9969	9985	0.2%	2	2
960	12105	12133	0.2%	1	2
1152	14661	14682	0.1%	1	4

4.2 Results for non-uniform networks

In Section 4.1, we have presented the experimental results for uniform networks. In non-uniform networks, the improvement of GLB-SCR over SSR becomes more significant because the design of GLB-SCR takes into consideration the non-uniform link capacity distribution. In this subsection, we present the experimental results for non-uniform networks. We conduct four sets of experiments: (a) IC=48 and UC=48; (b) IC=48 and UC=96; (c) IC=192 and UC=192; (d) IC=192 and UIC=384. During the experiments, the links with maximum link load are upgraded in the network upgrade. Fig. 8, Fig. 9, Fig. 10, and Fig. 11 plot the network upgrade trace and reallocation trace in these cases respectively. The X-axis is the established connection capacity of all the existing connections and the Y-axis is the total network capacity (i.e., the sum of capacity of all the links). The upgrade trace line shows the footprint of the network upgrade as connections are provisioned. The points of the reallocation trace indicate when a reallocation event is triggered. Note that the Y-axis values of each reallocation trace (the symbols × and + in these figures) have no meaning.

In all cases, the network with GLB-SCR needs less total network capacity to provision the same amount of connections most of the time. During a few short periods, the total network capacity required by the network with GLB-SCR may temporarily reach the same or a slightly higher level, compared to the network with SSR. However, the network with GLB-SCR shows constant savings in total network capacity over the long run. The tables listed following the figures (i.e., Table 4, Table 5, Table 6, and Table 7) show the improvement of GLB-SCR over SSR in terms of total network capacity with the same amount of established connection capacity. The data in each table are obtained by sampling the cross-points of the upgrade trace and five equal-distance vertical lines (as shown in the corresponding figures). The tables show that the average improvement of GLB-SCR over SSR in terms of total network capacity is 2.9%, 7.9%, 5.7%, and 9% for each case respectively. There are two observations: (1) the higher the initial capacity, the more improvement is obtained; and (2) the greater the ratio of upgrade capacity over initial capacity, the more the improvement. This is because more free capacity creates more space for the GLB-SCR approach to balance the network load distribution. Improvement (or reduction) in total network capacity translates into savings in equipment investment. The reallocation trace in these figures shows that this improvement is achieved by triggering slightly more reallocations.

Fig. 8, Fig. 9, Fig. 10, and Fig. 11 also show that each upgrade in the network with GLB-SCR needs fewer capacity increments (due to the design of NLV). This property is especially important to carriers who have strict budget limits and want to achieve short-term cost savings. On the other hand, the results also show that carriers using the GLB-SCR approach can benefit from tremendous savings in capacity investments over a long-term period.

The merit of the GLB-SCR approach comes from the design of NLV and its objective of minimizing NLV. Fig. 12 shows the link load distribution before and after a GLB-SCR reallocation in the case with IC=48 and UC=48. It gives a direct vision for the value of each member in a NLV and explains how the link load distribution changes after a reallocation. The number of congested links is reduced after the reallocation by moving some of the load to the links with lower load. For example, the number of links that have more than 90% capacity used (i.e., have loads greater than 43 in this example) is reduced after the reallocation.

IEEE COMPUTER SOCIETY

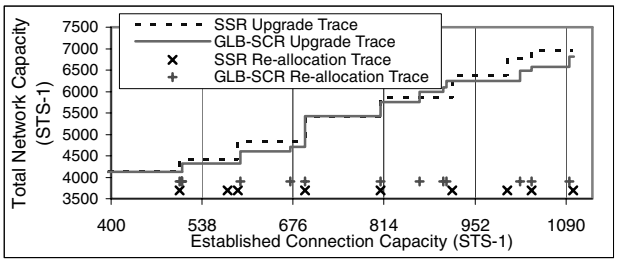


Fig. 8 Upgrade trace and reallocation trace for the non-uniform network when IC=48 and UC=48.

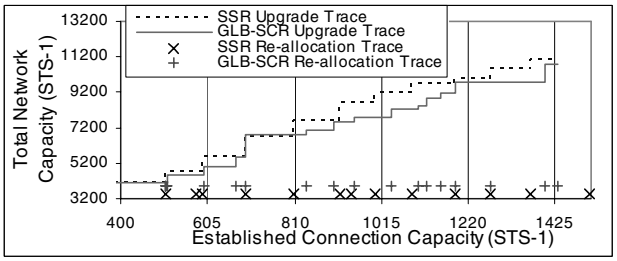


Fig. 9 Upgrade trace and reallocation trace for the non-uniform network when IC=48 and UC=96

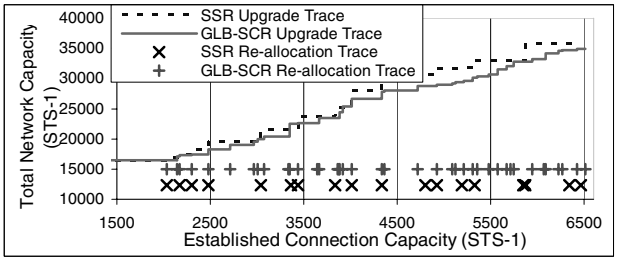


Fig. 10 Upgrade trace and reallocation trace for the non-uniform network when IC=192 and UC=192.

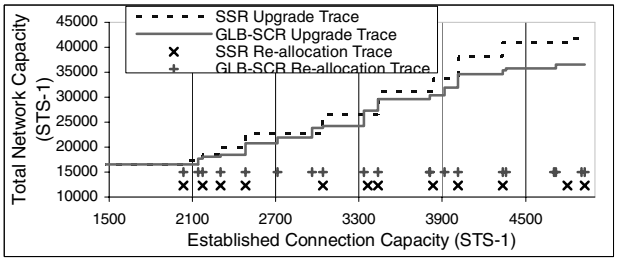


Fig. 11 Upgrade trace and reallocation trace for the non-uniform network when IC=192 and UC=384.

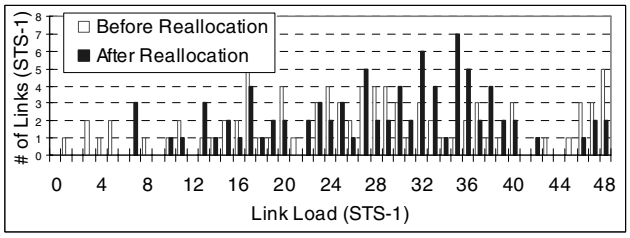


Fig. 12 Link load distribution before and after one GLB-SCR reallocation for the case when IC=48 and UC=48.

Table 4 The improvement of GLB-SCR over SSR for the non-uniform network when IC=48 and UC=48.

EstablishedConnection Capacity (STS-1)	Total Network Capacity (STS-1)		Improvement
	SSR	GLB-SCR	Average = 2.9%
538	4416	4320	2.2%
676	4848	4704	3.0%
814	5856	5760	1.6%
952	6384	6240	2.3%
1090	6960	6576	5.5%

Table 5 The improvement of GLB-SCR over SSR for the non-uniform network when IC=48 and UC=96.

EstablishedConnection Capacity (STS-1)	Total Network Capacity (STS-1)		Improvement
	SSR	GLB-SCR	Average = 7.9%
605	5568	4992	10.3%
810	7584	6816	10.1%
1015	9120	7776	14.7%
1220	9984	9792	1.9%
1425	11040	10752	2.6%

Table 6 The improvement of GLB-SCR over SSR for the non-uniform network when IC=192 and UC=192.

EstablishedConnection Capacity (STS-1)	Total Network Capacity (STS-1)		Improvement
	SSR	GLB-SCR	Average = 5.7
2500	19584	18240	6.9%
3500	23808	22656	4.8%
4500	30336	28032	7.6%
5500	33204	30720	7.0%
6500	35712	34944	2.2%

Table 7 The improvement of GLB-SCR over SSR for the nonuniform network when IC=192 and UC=384.

EstablishedConnection Capacity (STS-1)	Total Network Capacity (STS-1)		Improvement
	SSR	GLB-SCR	Average = 9.0
2100	17280	16512	4.4%
2700	22656	20736	8.5%
3300	26496	24192	8.7%
3900	33792	30336	10.2%
4500	41088	35712	13.1%

4.3 Comparison of uniform upgrade and non-uniform upgrade

In Section 4.1 and 4.2, we have shown that the GLB-SCR approach outperforms the SSR approach in terms of established connection capacity for uniform networks and in terms of total network capacity for non-uniform networks. From a network planning perspective, uniform networks are formed by uniform upgrade, which increments the same amount of capacity for every link in the network. An alternative is to upgrade a portion of the links in the network, which is called a non-uniform upgrade. This section compares the uniform upgrade and non-uniform upgrade schemes under the GLB-SCR approach. We only show the results of the case with IC=48 and UC=48. Other cases have similar results. Table 8 gives the total network capacity needed for the uniform upgrade and for the non-uniform upgrade respectively. The data are sampled at the time when a uniform upgrade occurs (or the same amount of connection capacity is established for the case of non-uniform upgrade). It shows that the non-uniform upgrade scheme saves, on average, 29% total network capacity compared to the uniform upgrade scheme. However, this saving is obtained at the expense of more

IEEE COMPUTER SOCIETY

network upgrades, resulting in increased operations expenditure (OPEX) for more frequent equipment installation. Carriers can choose their own scheme based on estimates of CAPEX and OPEX.

Table 8 Comparison of the uniform upgrade and non-uniform upgrade for the case when IC=48 and UC=48

Established connection capacity (STS-1)	Total Network Capacity (STS-1)		Improvement of uniform upgrade over non-uniform upgrade
	Uniform Upgrade	Non-uniform Upgrade	Average = 29%
1038	8256	6576	20%
1508	12384	8976	28%
2158	16512	11904	28%
2480	20640	13104	37%
3308	24768	16416	34%

5. Conclusions

In this paper, we proposed a load-balancing spare capacity reallocation approach in next-generation SONET metro networks. We defined a novel network load vector (NLV), which reflects the network load distribution and takes into consideration the non-uniform link capacity distribution. Our proposed approach aims at minimizing this NLV. For the first time, we proposed a spare capacity reallocation approach from a capacity planning and network upgrade perspective. We developed a greedy heuristic (GLB-SCR) to implement our approach and conducted experiments to compare GLB-SCR with a previous approach (SSR). Our experiments show that GLB-SCR outperforms SSR in terms of established connection capacity and total network capacity in both uniform and non-uniform networks. This improvement translates into increased revenue for carriers. Because of the way the NLV is designed, the advantage of GLB-SCR over SSR is more significant in non-uniform networks, which are more likely to be the case in real-world metro networks. In the experiments, we further showed that the non-uniform upgrade can bring significant savings on capacity investment at the expense of more network upgrades.

This paper focuses on the next-generation SONET metro networks. However, our approach can be applied to any circuit-switching network. Although our approach only operates on protection paths, it can be slightly modified to reallocate both working and protection paths when some connections allow their working paths to be rerouted under their service level agreements (SLAs).

6. References

[1] H. Graber and H. Roche, “Multi-service switches and the Service IntelligentTM optical architecture for SONET/SDH metro networks,” Bell Labs Technical Journal, vol. 8, no. 2, pp. 111-127, September 2003.

[2] D. Cavendish et al., “New transport services for next-generation SONET/SDH systems,” IEEE Communications Magazine, vol. 40, no. 5, pp. 80-87, May 2002.

[3] E. Hernandez-Valencia, “Hybrid transport solutions for TDM/Data networking services,” IEEE Communications Magazine, vol. 40, no. 5, pp. 104-112, May 2002.

[4] P. Bonenfant and A. Rodriguez-Moral, “Generic framing procedure (GFP): the catalyst for efficient data over transport,” IEEE Communications Magazine, vol. 40, no. 5, pp. 72-79, May 20002.

[5] K. Zhu, H. Zang, and B. Mukherjee, “Exploiting the benefit of virtual concatenation technique to the optical transport networks,” Proceedings, Optical Fiber Communication Conference (OFC), WI2, March 2003.

[6] C. Ou, K. Zhu, N. Singhal, and B. Mukherjee, “Survivable virtual concatenation for Data-over-SONET/SDH networks,” Proceedings, Optical Fiber Communication Conference (OFC), FH2, February 2004.

[7] P. Charalambous et al., “A national mesh network using optical cross-connect switches,” Proceedings, Optical Fiber Communication Conference (OFC), FR6, March 2003.

[8] E. Bouillet et al., “Lightpath re-optimization in mesh optical networks,” Proceedings, European Conference on Networks & Optical Communication (NOC), June 2002.

[9] N. Geary et al., “Analysis of the potential benefits of OXC-based intelligent optical networks,” Optical Networks Magazine, vol.4, no.2, March/April 2003.

[10] N. Geary et al., “Analysis of optimisation issues in multi-period DWDM network planning,” IEEE INFOCOM, pp. 152-158, April 2001.

[11] B. Doshi and P. Harshavardhana, “Broadband network infrastructure of the future: roles of network design tools in technology deployment strategies,” IEEE Communications Magazine, May 1998.

[12] M. Goyal, G. Li, and J. Yates, "Shared mesh restoration: a simulation study," Proceedings, Optical Fiber Communication Conference (OFC), 2002.

[13] B. Doshi et al., "Optical network design and restoration," Bell Labs Technical Journal, Vol.4, No.1, 1999.

[14] M. Kodaliam and T. Lakshman, "Dynamic routing of bandwidth guaranteed tunnels with restoration," IEEE INFOCOM, pp. 902-911, 2000.

[15] Y. Liu, D. Tipper, and P. Siripongwutikorn, "Approximating optimal space capacity allocation by successive survivable routing," IEEE INFOCOM, pp. 699-708, 2001.

[16] C. Qiao and D. Xu, "Distributed partial information management (DPIM) schemes for survivable networks--part I," IEEE INFOCOM, pp. 302-311, June 2002.

[17] D. Xu, C. Qiao and Y. Xiong, “An ultra-fast shared path protection scheme - distributed partial information management, Part II,” IEEE International Conference on Network Protocols (ICNP), pp. 344-353, November 2002.

[18] G. Li, D. Wang, C. Kalmanek, and R. Doverspike, “Efficient distributed restoration path selection for shared mesh restoration,” IEEE/ACM Transactions on Networking, vol. 11, no. 5, pp. 761-771, October 2003.

[19] L. Shen, X. Yang and B. Ramamurthy, "A load-balancing shared-protection-path reconfiguration approach in WDM wavelength-routed networks," Proceedings, Optical Fiber Communication Conference (OFC), ThO3, February 2004.

[20] GNU Multiple Precision Arithmetic Library, http://www.swox.com/gmp/