COMPNW 3627
25 August 2007 Disk Used
ARTICLE IN PRESS
No. of Pages 13, Model 3+

Available online at www.sciencedirect.com
ScienceDirect
Computer Networks xxx (2007) xxx–xxx
Computer Networks
www.elsevier.com/locate/comnet

A restorable MPLS-based hose-model VPN network

Jian Chu, Chin-Tau Lea *

Electronic and Computer Engineering Department, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

Received 2 January 2007; received in revised form 29 May 2007; accepted 26 July 2007

Responsible Editor: J. Sole-Pareta

Abstract

Design of a restorable MPLS-based Layer-3 VPN network with QoS guarantee is a new and important subject that has not been widely studied before. The main challenge arises from the fact that the Service Level Agreements (SLAs) of a L3-VPN usually only specify the maximum ingress and egress traffic rate, and provide no point-to-point traffic matrix information (i.e., a hose-model VPN). Conventional restoration and traffic engineering techniques do not apply to this type of traffic model. In this paper, we present a restoration network architecture and present two algorithms for solving the routing problem of this type of restoration networks. We demonstrate the effectiveness of our proposed restoration architecture by comparing the throughput performance with other approaches.

Keywords: Layer-3 VPNs; MPLS network; Traffic engineering; Restoration

1. Introduction

1.1. MPLS-based L3-VPN

MPLS-based VPN services can help an enterprise to converge existing disparate networks onto a consolidated, end-to-end infrastructure that can support combined data, voice, and video services. The underlying concept for VPN implementation is MPLS's *label stacking*. We can use a two-layer label stack: an external label and an internal label. The external label is the routing label and is used by the core router to route a packet to its destination edge router. The internal label is the VPN label and it is used by the edge router to separate traffic from different VPNs (Figs. 1 and 2).

Depending on the protocol levels involved in constructing a VPN, we can divide VPNs into two types: Layer-2 (L2) and Layer-3 (L3) VPNs. A L2 VPN provides a secure point-to-point transport service. But if an enterprise needs connectivity among n points, full connectivity among them requires $n(n-1)$ L2 VPN links. It is obviously not economical unless n is small. A L3 VPN (MPLS-based), on the other hand, is a set of sites of which the connectivity is provided by a provider's MPLS routing network and there are no virtual links set up directly among these sites. L3 VPNs provide three key benefits to enterprises: any-to-any connectivity through the use of forwarding tables, the ability to retain

* Corresponding author. Tel.: +852 23587090.
E-mail addresses: chujian@ust.hk, eejchu@ust.hk (J. Chu), eelea@ust.hk (C.-T. Lea).

1389-1286/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2007.07.009

Please cite this article in press as: J. Chu, C.-T. Lea, A restorable MPLS-based hose-model VPN network, Comput. Netw. (2007), doi:10.1016/j.comnet.2007.07.009

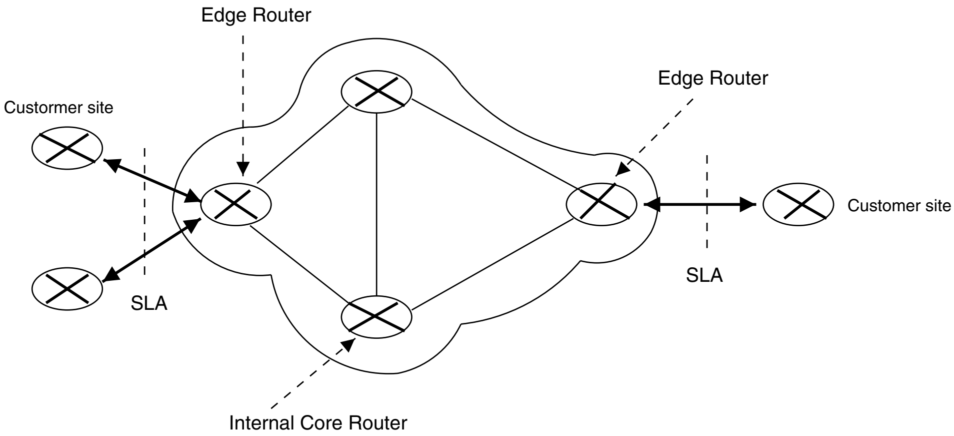


Fig. 1. An MPLS backbone network.

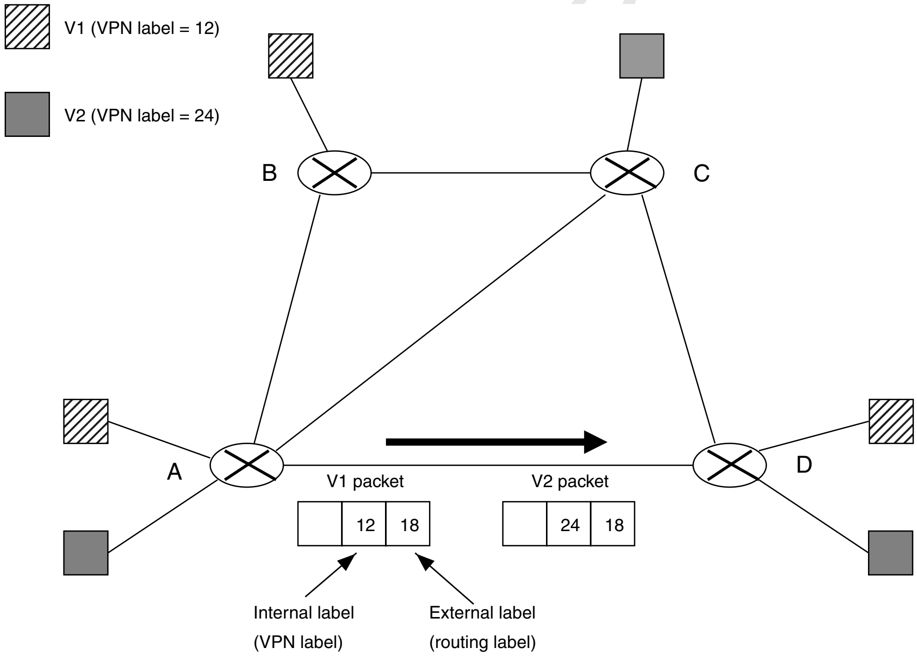


Fig. 2. There are two labels in a VPN packet. The internal label is the VPN label, and the external label is the routing label.

existing IP addressing plan by supporting overlapping IP addresses, and greater scalability at the site-to-site and data center levels [1].

To support the QoS of a L3 VPN, customers are required to sign a Service Level Agreement (SLA) with the provider (Fig. 1). The SLA of a L3-VPN usually only specifies the ingress and egress rate, but provides no destination information. This type of VPNs are usually called hose-model VPNs [2]. The other type of rate specification is the pipe-model that requires the customer to specify the traffic rate between each source-destination pair. Hose-model VPNs obviously are much easier to use for customers.

This paper will focus on hose-model L3 VPNs. Conventional MPLS traffic engineering tools usually assume that traffic matrix $T = \{d_{ij}\}$ of the VPN to be set up is given, where d_{ij} represents the average traffic intensity from node i to node j. However, as pointed above a hose-model L3-VPN does not always provide that information. Instead, only the *row* sums $\sum_j d_{ij} = \alpha_i$, where α_i (the *ingress* band-

Please cite this article in press as: J. Chu, C.-T. Lea, A restorable MPLS-based hose-model VPN network, Comput. Netw. (2007), doi:10.1016/j.comnet.2007.07.009

width constraint) is the maximum rate of traffic that node i can send into the network, and *column* sums $\sum_i d_{ij} = \beta_j$, where β_j (the *egress* bandwidth constraint) is the maximum rate of traffic that node j can receive from the network, are given. To guarantee the QoS of a hose-model L3-VPN is a challenging task because for a given set of traffic constraints α_i and β_i, there are many traffic matrices that can satisfy the constraints. Therefore, we must provide enough bandwidth for any traffic matrix that meets the ingress and egress constraints. The uncertainty inherited in a hose-model traffic pattern makes finding efficient routing and capacity planning algorithms a difficult task.

Many algorithms have been proposed for hose-model VPN provisioning [3–6]. But there is little work on hose-model VPN restoration. This is the focus of the paper. When a link or node fails, traversing traffic needs to be rerouted through alternate paths. There are two restoration approaches: link restoration and path restoration. In *link restoration* (also referred to as local restoration or fast restoration), each link of the network is protected by a set of pre-determined detour paths that connect the two endpoints of that link. Upon a link failure, traffic of the link is switched to the detour protecting paths. In *path restoration*, each path carrying working traffic is protected by a diverse backup path. Link restoration can be activated immediately when link failures are detected. In contrast, path restoration can be activated only after the failure information propagates to the source node. Link restoration is the preferred method for providing fast restoration in MPLS and optical networks [7,8] and we will focus on link restoration in the paper.

1.2. Prior works and our contributions

There are many papers on restoration, but few are related to hose-model traffic patterns. Ref. [9] presented the linear programming based approaches to solve the optimal capacity and flow assignment problem with link and path restoration strategies. Refs. [10,11] tackled a similar problem in mesh-based WDM optical networks. In all these papers, traffic matrix T is assumed given. Consequently they do not apply to a hose-model problem. For hose-model VPN protection, Ref. [12] proposed a restoration algorithm for a single hose-model VPN construction. It constructs a tree with each link protected by a detour path. In case of a link failure, traffic will be rerouted and the overall topology is still a tree. This is obviously not an efficient protection scheme as we need to do it for every VPN. A recent paper [13] proposed to create a fully-connected virtual topology on top of the physical topology. It then routes packets across two-hop logical links in the fully-connected virtual topology. Some obvious drawbacks of the approach include longer paths, and thus longer end-to-end delays, and low efficiency. These problems become more severe for networks with sparse links as creating a fully-connected topology out of these networks will be more expensive than networks with dense links.

In this paper, we propose a restorable MPLS VPN network architecture with QoS guarantees. Our contributions include the following:

1. We present a restorable MPLS-based network architecture for supporting L3 VPNs with QoS guarantees. The protection is done for all VPNs, not just for one VPN.
2. We present an efficient decomposition algorithm to compute the optimal routes for this restoration network. This algorithm can include a hop-count limit on the restoration paths.
3. We compare the throughput performance of the new restoration architecture with that of conventional architectures. The results show that our proposed architecture can achieve a better performance for hose-model VPNs under different bandwidth requirements.

The rest of the paper is organized as follows. Section 2 presents the new restorable network architecture and the linear programming formulation for designing the working and restoration paths for the network. Section 3 presents an efficient decomposition algorithm. Section 4 presents performance evaluation of the proposed schemes. We conclude our discussion in Section 5.

2. Restorable L3-VPN network with QoS guarantee

We consider single-link failures [10]. Single-node failures can be analyzed similarly. The analysis in [5] showed that splitting traffic among multiple paths will have a much better performance than a single path approach for supporting hose-model VPNs. We will use the multi-path approach in this paper. Multiple paths for a given source-destination pair will be set up and load-balancing among them is done according to a set of pre-determined splitting ratios derived from the routing computation.

Please cite this article in press as: J. Chu, C.-T. Lea, A restorable MPLS-based hose-model VPN network, Comput. Netw. (2007), doi:10.1016/j.comnet.2007.07.009

Load-balancing can be done such that no out-of-sequence transmissions occur for packets belonging to the same flow [14].

2.1. Network architecture

Our architecture is based on the non-blocking network approach outlined in [15]. The advantages of this approach for restoration will be discussed later. Let $(\theta\tilde{\alpha}_i, \theta\tilde{\beta}_i)$ represent the maximum amount of *total* ingress and egress VPN traffic allowed to enter and leave the network at the edge router i, where $\tilde{\alpha}_i$ and $\tilde{\beta}_i$ are given constants, describing the degree of unevenness of amount of traffic at each node. For example, suppose a network specifies $(\tilde{\alpha}_1 = 4, \tilde{\beta}_1 = 4)$ and $(\tilde{\alpha}_2 = 12, \tilde{\beta}_2 = 12)$, then it means that the amount of traffic allowed at edge router 2 (both ingress and egress) is three times that of router 1. Note that only the relative – not absolute – magnitudes of $\tilde{\alpha}_i$ and $\tilde{\beta}_i$ have significance as the real amount of admissible traffic is determined by θ. We need to design the network such that as long as the ingress and egress traffic of node i is below $(\theta\tilde{\alpha}_i, \theta\tilde{\beta}_i)$, traffic routed through a link is always below its link capacity. A network with this property is called non-blocking in [15]. If the network is non-blocking, we only need to check if there is enough bandwidth left at the edge routers to which endpoints of the VPN are connected. There is no need to check the internal paths' available bandwidths. The decision of admitting a VPN is greatly simplified.

2.2. Optimal routing and link restoration

Our goal is to compute the working and link-restoration paths to maximize the admissible amount of traffic (i.e., θ). Recall that $\tilde{\alpha}_i$ and $\tilde{\beta}_i$ are given constants and their values are determined from past traffic demands (if there is no prior information about the network, we can simply assume the ingress–egress capacity at an edge node is proportional to the total capacity of network links incident at that node). The network is described as a directed graph $G(V, E)$, where V is the set of vertices (nodes) and E is the set of links. Let $Q \subseteq V$ be the set of edge routers through which traffic is admitted into the network. We first introduce the following notations:

- c_e The capacity for link $e \in E$
- χ^e_{ij} The routing variable, representing the portion of working traffic from node $i \in Q$ to $j \in Q$ routed through link e.
- $A(e)$ the capacity on link e reserved for working traffic. When link e fails, $A(e)$ amount of traffic has to be rerouted along a set of restoration (detour) paths that connect the two endpoints of this link.
- y^e_f The amount of restoration traffic that will be routed through link e in case link f fails.

Given the ingress and egress traffic constraints at all the edge nodes, a traffic matrix is called *valid* if it satisfies the specified traffic constraints. Let $H = [(\tilde{\alpha}_1, \tilde{\beta}_1), \ldots, (\tilde{\alpha}_n, \tilde{\beta}_n)]$ and $\widetilde{H} = [(\theta\tilde{\alpha}_1, \theta\tilde{\beta}_1), \ldots, (\theta\tilde{\alpha}_n, \theta\tilde{\beta}_n)]$. Let M be the set of valid traffic matrices $T = \{d_{ij}\}$ constrained by H. If $T \in M$, then θT will be a valid traffic matrix for the constraint $\widetilde{H}$. In the following discussion, we consider $T \in M$ and use the form θT to indicate a valid traffic matrix constrained by $\widetilde{H}$.

Our problem is to determine the working flow χ^e_{ij} and the restoration flow y^e_f that can maximize θ. We formulate it as the following:

$$\max \theta \tag{1a}$$

$$\text{s.t.} \sum_{e\in\Gamma^+(v)} \chi^e_{ij} - \sum_{e\in\Gamma^-(v)} \chi^e_{ij} = 0, \quad i,j \in Q, \quad v \in V, \ v \neq i,j \tag{1b}$$

$$\sum_{e\in\Gamma^+(v)} \chi^e_{ij} - \sum_{e\in\Gamma^-(v)} \chi^e_{ij} = 1, \quad i,j \in Q, \quad v \in V, \ v = i \tag{1c}$$

$$\sum_{e\in\Gamma^+(v)} \chi^e_{ij} - \sum_{e\in\Gamma^-(v)} \chi^e_{ij} = -1, \quad i,j \in Q, \quad v \in V, \ v = j \tag{1d}$$

$$\sum_{i,j\in Q} \chi^e_{ij} \cdot (\theta d_{ij}) \leqslant A(e), \quad e \in E, \ T \in M \tag{1e}$$

$$\sum_{e\in\Gamma^+(v)} y^e_f - \sum_{e\in\Gamma^-(v)} y^e_f = 0, \quad f = (o,t) \in E, \ v \neq o,t \tag{1f}$$

$$\sum_{e\in\Gamma^+(v)} y^e_f - \sum_{e\in\Gamma^-(v)} y^e_f = A(f), \quad f = (o,t) \in E, \ v = o \tag{1g}$$

$$\sum_{e\in\Gamma^+(v)} y^e_f - \sum_{e\in\Gamma^-(v)} y^e_f = -A(f), \quad f = (o,t) \in E, \ v = t \tag{1h}$$

$$A(e) + y^e_f \leqslant c_e, \quad e,f \in E, \ e \neq f \tag{1i}$$

$$y^e_e = 0, \quad e \in E \tag{1j}$$

$$\chi, y, \theta, A \geqslant 0 \tag{1k}$$

where $\Gamma^+(v)$ and $\Gamma^-(v)$ are the set of outgoing and incoming links of node v, and o and t represent the originating and terminating nodes of link f.

Please cite this article in press as: J. Chu, C.-T. Lea, A restorable MPLS-based hose-model VPN network, Comput. Netw. (2007), doi:10.1016/j.comnet.2007.07.009

Constraints (1b)–(1d) represent the flow conservation constraints for working traffic at intermediate, source, and destination nodes and constraints (1f)–(1h) represent the flow conservation constraints for restoration traffic. Constraint (1e) ensures that the total amount of working traffic on any link does not exceed the working capacity $A(e)$. Constraint (1i) ensures that the sum of working traffic and the restoration traffic that appears on a link due to failure of any other link does not exceed the link capacity. Constraints (1j,1k) provide the ranges for the variables.

Constraint (1e) is not a linear constraint. But we can introduce a new routing variable $x^e_{ij} = \chi^e_{ij} \cdot \theta$ and rewrite Eq. (1) as the following:

$$\max \theta \tag{2a}$$

$$\text{s.t.} \sum_{e\in\Gamma^+(v)} x^e_{ij} - \sum_{e\in\Gamma^-(v)} x^e_{ij} = 0, \quad i,j\in Q, \quad v\in V, \; v\neq i,j \tag{2b}$$

$$\sum_{e\in\Gamma^+(v)} x^e_{ij} - \sum_{e\in\Gamma^-(v)} x^e_{ij} = \theta, \quad i,j\in Q, \quad v\in V, \; v=i \tag{2c}$$

$$\sum_{e\in\Gamma^+(v)} x^e_{ij} - \sum_{e\in\Gamma^-(v)} x^e_{ij} = -\theta, \quad i,j\in Q, \quad v\in V, \; v=j \tag{2d}$$

$$\sum_{i,j\in Q} x^e_{ij} d_{ij} \leqslant A(e), \quad e\in E, \; T\in M \tag{2e}$$

$$\sum_{e\in\Gamma^+(v)} y^e_f - \sum_{e\in\Gamma^-(v)} y^e_f = 0, \quad f=(o,t)\in E, \; v\neq o,t \tag{2f}$$

$$\sum_{e\in\Gamma^+(v)} y^e_f - \sum_{e\in\Gamma^-(v)} y^e_f = A(f), \quad f=(o,t)\in E, \; v=o \tag{2g}$$

$$\sum_{e\in\Gamma^+(v)} y^e_f - \sum_{e\in\Gamma^-(v)} y^e_f = -A(f), \quad f=(o,t)\in E, \; v=t \tag{2h}$$

$$A(e) + y^e_f \leqslant c_e, \quad e,f\in E, \; e\neq f \tag{2i}$$

$$y^e_e = 0, \quad e\in E \tag{2j}$$

$$x,y,\theta,A \geqslant 0 \tag{2k}$$

Although Eq. (2) is a linear programming formulation, it cannot be solved directly because constraint (2e) lists every valid T in M and there are too many of them. The problem is solved with the following property. Different forms of this property have been given in [16,17].

Property 1. *Given $H = [(\tilde{\alpha}_1, \tilde{\beta}_1), \ldots, (\tilde{\alpha}_n, \tilde{\beta}_n)]$, routing x^e_{ij} and working capacity reservation $A(e)$ can satisfy constraint (2e) for all traffic matrices in M if and only if there exist non-negative weights $\pi_e(i)$ and $\lambda_e(i)$ for each $e\in E$ and $i\in Q$ such that*

(i) $\sum_{i\in Q}\tilde{\alpha}_i\pi_e(i) + \sum_{i\in Q}\tilde{\beta}_i\lambda_e(i) \leqslant A(e)$ for each $e\in E$.
(ii) $x^e_{ij} \leqslant \pi_e(i) + \lambda_e(j)$ for each $e\in E$ and every $i,j\in Q$.

Proof. The proof is provided in Appendix A. □

Property 1 allows us to replace constraint (2e) in Eq. (2) with requirements (i)–(ii) in Property 1 and transform the formulation into the following:

$$\max \theta \tag{3a}$$

$$\text{s.t.} \sum_{e\in\Gamma^+(v)} x^e_{ij} - \sum_{e\in\Gamma^-(v)} x^e_{ij} = 0, \quad i,j\in Q, \quad v\in V, \; v\neq i,j \tag{3b}$$

$$\sum_{e\in\Gamma^+(v)} x^e_{ij} - \sum_{e\in\Gamma^-(v)} x^e_{ij} = \theta, \quad i,j\in Q, \quad v\in V, \; v=i \tag{3c}$$

$$\sum_{e\in\Gamma^+(v)} x^e_{ij} - \sum_{e\in\Gamma^-(v)} x^e_{ij} = -\theta, \quad i,j\in Q, \quad v\in V, \; v=j \tag{3d}$$

$$\sum_{i\in Q}\tilde{\alpha}_i\cdot\pi_e(i) + \sum_{i\in Q}\tilde{\beta}_i\cdot\lambda_e(i) \leqslant A(e), \quad e\in E \tag{3e}$$

$$x^e_{ij} \leqslant \pi_e(i) + \lambda_e(j), \quad i,j\in Q, \; e\in E \tag{3f}$$

$$\sum_{e\in\Gamma^+(v)} y^e_f - \sum_{e\in\Gamma^-(v)} y^e_f = 0 \quad f=(o,t)\in E, \; v\neq o,t \tag{3g}$$

$$\sum_{e\in\Gamma^+(v)} y^e_f - \sum_{e\in\Gamma^-(v)} y^e_f = A(f), \quad f=(o,t)\in E, \; v=o \tag{3h}$$

$$\sum_{e\in\Gamma^+(v)} y^e_f - \sum_{e\in\Gamma^-(v)} y^e_f = -A(f), \quad f=(o,t)\in E, \; v=t \tag{3i}$$

$$A(e) + y^e_f \leqslant c_e, \quad e,f\in E, \; e\neq f \tag{3j}$$

$$y^e_e = 0, \quad e\in E \tag{3k}$$

$$x,y,\pi,\lambda,\theta,A \geqslant 0 \tag{3l}$$

The above linear programming (LP) problem can be solved by standard LP solvers like Cplex [18]. Then we can derive the set of working paths and link restoration paths from the flow variables x^e_{ij} and y^e_f.

3. Adding hop-count limit to restoration paths

If the goal is only to maximize network throughput, some of the computed restoration paths may be long and this makes the restoration latency unacceptable. In the following we present a decomposition scheme that will lead to a path formulation for the design of restoration paths. The path formulation allows us to add a hop count limit on the restoration paths. Another benefit of the approach is that the new approach is faster than the one in Section 2.

3.1. Two-stage decomposition algorithm

The computation of the working flow and the restoration flow in Eq. (3) can be partitioned into two separate stages. At the first stage, corresponding to constraints (3b)–(3f), we assume the working capacity vector Λ is given, where

$$\Lambda = [A(1), A(2), \ldots, A(m)] \tag{4}$$

and m is the number of links (assuming the set of links is labeled from 1 to m). We can compute the optimal routing. The process also generates a new working capacity vector $\widetilde{\Lambda}$. A working capacity vector is called *feasible* if it satisfies constraints (3g)–(3k), meaning that the network has enough capacity left to protect it. At the second stage, we will test if the newly generated $\widetilde{\Lambda}$ from stage 1 is feasible or not. If not, we will modify $\widetilde{\Lambda}$ in the 2nd stage to make it feasible and pass the result back to stage 1 for another round of iteration.

Stage 1: Assume Λ (i.e., all $A(e)$) is given. We determine routing and maximum θ by solving the following linear programming problem.

$$\max \theta \tag{5a}$$

$$\text{s.t.} \sum_{e\in\Gamma^+(v)} x_{ij}^e - \sum_{e\in\Gamma^-(v)} x_{ij}^e = 0, \quad i,j\in Q, \quad v\in V, \ v\neq i,j \tag{5b}$$

$$\sum_{e\in\Gamma^+(v)} x_{ij}^e - \sum_{e\in\Gamma^-(v)} x_{ij}^e = \theta, \quad i,j\in Q, \quad v\in V, \ v=i \tag{5c}$$

$$\sum_{e\in\Gamma^+(v)} x_{ij}^e - \sum_{e\in\Gamma^-(v)} x_{ij}^e = -\theta, \quad i,j\in Q, \quad v\in V, \ v=j \tag{5d}$$

$$\sum_{i\in Q} \tilde{\alpha}_i \cdot \pi_e(i) + \sum_{i\in Q} \tilde{\beta}_i \cdot \lambda_e(i) \leqslant A(e), \quad e\in E \tag{5e}$$

$$x_{ij}^e \leqslant \pi_e(i) + \lambda_e(j), \quad i,j\in Q, \ e\in E \tag{5f}$$

$$x, \pi, \lambda, \theta \geqslant 0 \tag{5g}$$

We use $\theta(\Lambda)$ to denote the optimal value of θ in Eq. (5) since it is a function of the working capacity vector Λ. Let R denote the set of feasible Λ. Property 2 in Appendix B shows that $\theta(\Lambda)$ is a concave function on R. This allows us to use the subgradient scheme [19] to compute a new working capacity vector $\widetilde{\Lambda}$ to improve θ. $\widetilde{\Lambda}$ can be computed as $\widetilde{\Lambda} \leftarrow \Lambda + \tau\gamma$, where γ is a subgradient vector at point Λ and τ is the step size. γ is called a *subgradient vector* of $\theta(\Lambda)$ at the point $\overline{\Lambda}$ if

$$\theta(\Lambda) - \theta(\overline{\Lambda}) \leqslant \gamma \cdot (\Lambda - \overline{\Lambda}), \quad \Lambda \in R \tag{6}$$

holds. We show how to compute γ below.

Property 3. *Suppose $\overline{\Lambda}$ is a working capacity vector as defined by Eq. (4). Let $\overline{\Lambda} \in R$ and γ be a subgradient of $\theta(\Lambda)$ at $\overline{\Lambda}$. Then*

$$\gamma = [\bar{\omega}_1, \bar{\omega}_2, \ldots, \bar{\omega}_m]. \tag{7}$$

where $\bar{\omega}$ are the corresponding optimal dual variables for constraint (5e).

Proof. From the definition of subgradient, γ at $\overline{\Lambda}$ can be computed as follows. For $\theta(\Lambda)$, let ω be the corresponding optimal dual variables for constraint (5e). Then from linear programming theory,

$$\begin{aligned}\theta(\Lambda) - \theta(\overline{\Lambda}) &= \sum_e \omega_e A(e) - \sum_e \bar{\omega}_e \overline{A}(e) \\ &\leqslant \sum_e \bar{\omega}_e A(e) - \sum_e \bar{\omega}_e \overline{A}(e) \\ &= \sum_e \bar{\omega}_e [A(e) - \overline{A}(e)]\end{aligned} \tag{8}$$

We can rewrite Eq. (8) as $\theta(\Lambda) - \theta(\overline{\Lambda}) \leqslant [\bar{\omega}_1, \bar{\omega}_2, \ldots, \bar{\omega}_m] \cdot (\Lambda - \overline{\Lambda})$. From the definition of subgradient, we thus have $\gamma = [\bar{\omega}_1, \bar{\omega}_2, \ldots, \bar{\omega}_m]$. □

Stage 2: At stage 2, we check if the new working capacity vector $\widetilde{\Lambda} = [\widetilde{A}(1), \ldots, \widetilde{A}(m)]$ produced by stage 1 is feasible or not. If not, we will modify it and make it feasible. This is done with the following LP formulation.

$$\max r \tag{9a}$$

$$\text{s.t.} \sum_{e\in\Gamma^+(v)} y_f^e - \sum_{e\in\Gamma^-(v)} y_f^e = 0, \quad f=(o,t)\in E, \ v\neq o,t \tag{9b}$$

$$\sum_{e\in\Gamma^+(v)} y_f^e - \sum_{e\in\Gamma^-(v)} y_f^e = A(f), \quad f=(o,t)\in E, \ v=o \tag{9c}$$

$$\sum_{e\in\Gamma^+(v)} y_f^e - \sum_{e\in\Gamma^-(v)} y_f^e = -A(f), \quad f=(o,t)\in E, \ v=t \tag{9d}$$

$$A(e) + y_f^e \leqslant c_e, \quad e,f\in E, \ e\neq f \tag{9e}$$

$$y_e^e = 0, \quad e\in E \tag{9f}$$

$$A(e) \geqslant \widetilde{A}(e) \cdot r, \quad e\in E \tag{9g}$$

$$y, r, A \geqslant 0 \tag{9h}$$

Please cite this article in press as: J. Chu, C.-T. Lea, A restorable MPLS-based hose-model VPN network, Comput. Netw. (2007), doi:10.1016/j.comnet.2007.07.009

Constraints (9b)–(9e) ensure that the computed working capacity is feasible. Constraint (9g) implies that $r = \min_e\{A(e)/\widetilde{A}(e)\}$. Thus, if $r \geqslant 1$, $\widetilde{A}$ is obviously feasible and it will be put back into Eq. (5) for the next iteration. If $r < 1$ ($\widetilde{A}$ is not feasible), then $(r\widetilde{A})$ will be feasible and we will pass this vector back to stage 1 for further iterations.

3.2. Path formulation for restoration paths

The formulation in stage 2 is link based. But we now transform it into a *path-flow* formulation so that we can impose a hop-count limit on the restoration paths. Also, a faster computation algorithm is available for the new form. Let P_e denote the set of all paths, except e, from the originating node to the terminating node of link e. Let $y(p)$ denote the restoration traffic on path p if its protected link fails. To restore the traffic for any failed link e, we must have $A(e) = \sum_{p\in P_e} y(p)$ for all $e \in E$. The path-flow formulation of Eq. (9) is as follows:

$$\max\ r \tag{10a}$$

$$\text{s.t.} \sum_{p\in P_e} y(p) \geqslant r\cdot \widetilde{A}(e) \quad e \in E \tag{10b}$$

$$\sum_{p\in P_e} y(p) + \sum_{p:p\in P_f, e\in p} y(p) \leqslant c_e \quad f \neq e, e,\ f \in E \tag{10c}$$

$$y, r \geqslant 0 \tag{10d}$$

The above path-flow formulation can be solved efficiently with a primal-dual approach adapted from the technique developed for the maximum concurrent flow problem in [20]. In addition, the hop-count limit can be easily included in the algorithm. The dual formulation of Eq. (10) is to associate a variable σ_e, for each link e, corresponding to constraint (10b) and a non-negative variable $w(e,f)$, for each pair $e,f \in E$, $e \neq f$, corresponding to constraint (10c). The dual formulation can be written as

$$\min \sum_{e\in E} c_e \sum_{f\in E, f\neq e} w(e,f) \tag{11a}$$

$$\text{s.t.} \sum_{e'\in p} w(e',e) + \sum_{f\in E, f\neq e} w(e,f) \geqslant \sigma_e p \in P_e, \quad e \in E \tag{11b}$$

$$\sum_e \widetilde{A}(e)\sigma_e \geqslant 1 \tag{11c}$$

If we set z_e to the minimum value of the left-hand-side (LHS) of constraint (11b), then $w(e,f)$ will be a dual feasible solution that satisfies constraint (11b). In addition, constraint (11c) can be easily satisfied if we divide all weights $w(e,f)$ by $\sum_e \widetilde{A}(e) z_e$.

The algorithm proceeds iteratively. At each iteration, for each link e, the *shortest* path $p \in P_e$ that minimizes the LHS of constraint (11b) is computed, flow is sent on the path, and the primal and dual variables are updated accordingly. Note that we can impose a hop-count limit in this step when we compute the shortest paths (e.g., we can use the Bellman–Ford algorithm [21]). This may reduce the working capacity a little bit (see Section 4), but the restoration latency can be restricted by adding this constraint.

4. Performance evaluation

In this section we compare the performance for different schemes. The primary performance measure is the maximum admissible bandwidth of traffic the network can sustain. In the following experiments, we assume the preference parameters $\tilde{\alpha}_i$ and $\tilde{\beta}_i$ at edge node i is set proportional to the total capacity of network links incident at node i. This is a logical assumption because if there is more traffic demand from a node, more links will be added to that node. As we mentioned in the introduction, most existing restoration algorithms assume the traffic matrix is given, and they can not be applied to problems with hose-model traffic patterns. In the following, we only compare our scheme with those that can be applied to hose-model traffic patterns. The schemes we compare include the following:

(a) *Linear programming, no link protection (LP_NP)*: the optimal scheme by solving Eq. (5) by setting the working capacity = link capacity (i.e., $A(e) = c_e$ for all $e \in E$).
(b) *Linear programming, with link protection (LP_P)*: the optimal scheme by solving Eq. (3) with standard LP solvers.
(c) *Decomposition and Iterative scheme (DI)*: the working and restoration flows are computed by the decomposition algorithm discussed in Section 3.
(d) *Non-iterative scheme (NI)*: the working capacity is computed by the link partition scheme that will be described below.
(e) *Shortest path routing and restoration (SPRR)*: Use the shortest paths for the working traffic. If multiple shortest paths exist, traffic will be evenly split among them. SPRR also uses the

Please cite this article in press as: J. Chu, C.-T. Lea, A restorable MPLS-based hose-model VPN network, Comput. Netw. (2007), doi:10.1016/j.comnet.2007.07.009

shortest restoration paths to protect a link. Once working and restoration paths are given, we can easily compute the maximum θ.

Two non-iterative schemes of (d) are described below. Without considering what type of traffic patterns to be supported in the network, we just compute the amount of working capacity under the condition that it can be protected.

Non-Iterative Scheme 1 (NI1):

$$\max \sum_{e\in E}\sum_{p\in P_e} y(p) \qquad (12a)$$

$$\text{s.t.} \sum_{p\in P_e} y(p) + \sum_{p:p\in P_f, e\in p} y(p) \leqslant c_e, \quad f \neq e, \; e,f \in E \qquad (12b)$$

$$y \geqslant 0 \qquad (12c)$$

The objective function (12a) is to maximize the sum of all links' working capacity. Once we have $y(p)$, we can derive $A(e)$. We then put $A(e)$ back into Eq. (5) to find the maximum θ for the hose-model pattern. This is done in one iteration.

Non-Iterative Scheme 2 (NI2):

Similar to the previous non-iterative scheme, we compute the working capacity first. But we change the objective function in (12a) to $\bar{r} = \min_{e\in E}\{r_e\}$, where r_e is the fraction of the capacity of link e reserved for the working traffic.

$$\max \bar{r} \qquad (13a)$$

$$\text{s.t.} \sum_{p\in P_e} y(p) \geqslant \bar{r}\cdot c_e \quad e \in E \qquad (13b)$$

$$\sum_{p\in P_e} y(p) + \sum_{p:p\in P_f, e\in p} y(p) \leqslant c_e, \quad f \neq e, e, \; f \in E \qquad (13c)$$

$$y, \bar{r} \geqslant 0 \qquad (13d)$$

4.1. Speed of convergence of the decomposition scheme

We first evaluate the effectiveness of the decomposition algorithm with the Sprint IP backbone topology shown in Fig. 3 [22]. We assume all nodes are edge nodes and all links have the same capacity of 1000 U. Although the theoretical rate of convergence for basic subgradient algorithm is linear [23], its convergence speed is much better in practice [24]. Fig. 4 shows how many iterative phases the decomposition algorithm needs to perform before getting a near-optimal solution. The straight line in Fig. 4 indicates the maximum admissible bandwidth computed by the LP_P scheme. As we can see, after 10 phases, the maximum admissible bandwidth computed by the decomposition scheme is very close to the optimal value. For larger networks presented later, we normally can get a near-optimal solution in less than a hundred phases for the decomposition scheme.

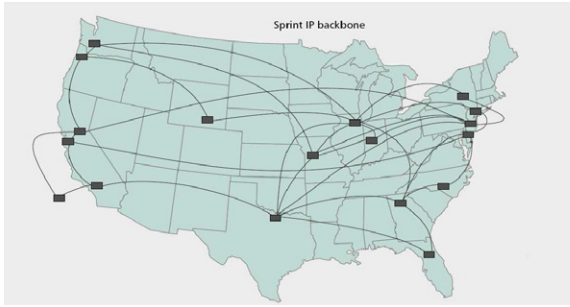


Fig. 3. The Sprint US backbone topology used for performance evaluation.

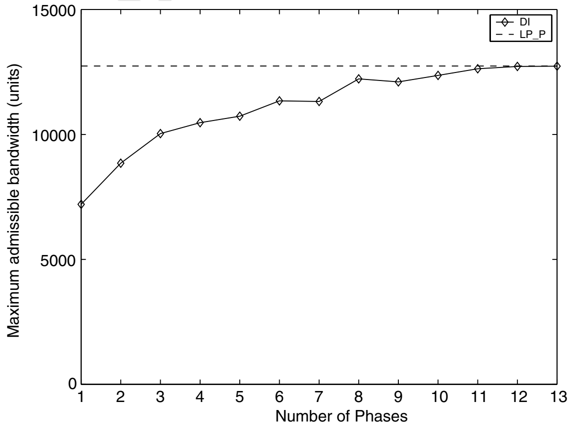


Fig. 4. Maximum admissible bandwidth for the decomposition scheme in the Sprint topology.

The number of iterative phases needed for achieving convergence given in Fig. 4 does not depend on the type of CPUs we use, but the real computation time for each phase will be machine dependent. Fig. 5 compares the running-times (in seconds) of the two approaches on randomly generated topologies measured on a 3-GHz Pentium-4 PC with 2 GB of memory. The results clearly show that the DI scheme is much faster than the LP_P scheme. The running-time of the LP_P scheme grows quickly with the size of the network. In con-

Please cite this article in press as: J. Chu, C.-T. Lea, A restorable MPLS-based hose-model VPN network, Comput. Netw. (2007), doi:10.1016/j.comnet.2007.07.009

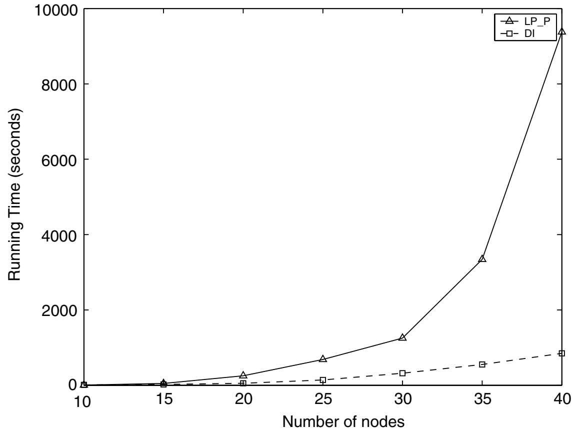


Fig. 5. The running time of the LP_P and DI schemes in various randomly generated topologies with different number of nodes.

trast, the running-time of the DI scheme grows in a much slower pace.

4.2. Throughput comparison

We evaluate the throughput of LP_NP, DI, NI1, NI2, and SPRR, based on randomly generated topologies with the following varying parameters: (1) number of links in the network, (2) number of nodes in the network, and (3) number of edge nodes.

- *Experiment 1*: 20 node topologies with 40–120 bidirectional links. The number of edge nodes is set to 10.
- *Experiment 2*: 10–50 node topologies. The number of links is twice the number of nodes in the topology. The number of edge nodes is set to 10.
- *Experiment 3*: 40 node topologies with 80 bidirectional links. The number of edge nodes is varied from 6 to 20.

The link capacity is 100 U. Figs. 6–8 are the average results of ten independent runs. LP_NP is presented only to show how much traffic we need to sacrifice to ensure restoration. Comparing LP_NP and DI, we find that DI reduces the admissible traffic by 7.4–25.8% in Experiment 1, 11–35.1% in Experiment 2, and 11.8–19.9% in Experiment 3.

DI performs much better than NI1 and NI2 in all the experiments. The performance gap between DI and NI1 (which achieves the secondary high performance among the restoration schemes) ranges from 10.8% to 40% in Experiment 1, from 23.5% to 32.8% in Experiment 2, and from 19.8% to 33.3%

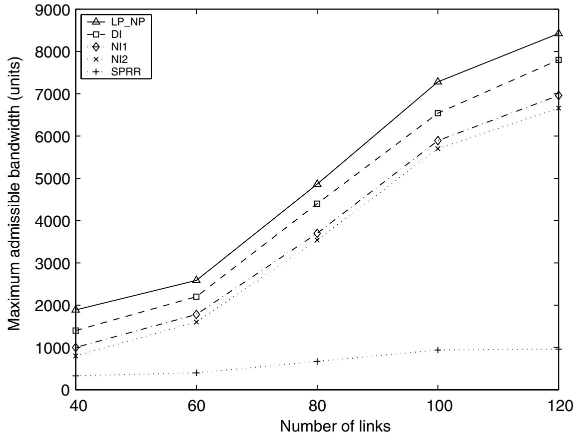


Fig. 6. Maximum admissible bandwidth vs. the number of links in Experiment 1.

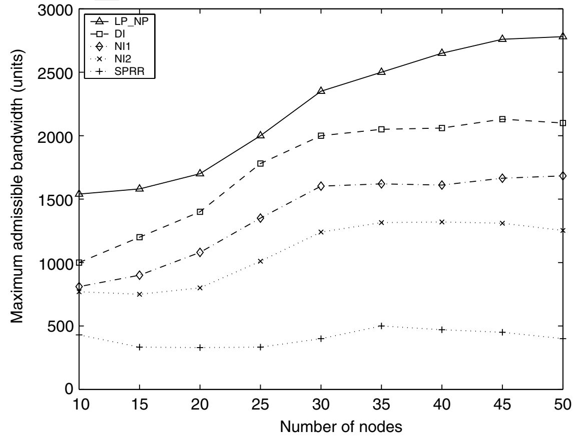


Fig. 7. Maximum admissible bandwidth vs. the number of nodes in Experiment 2.

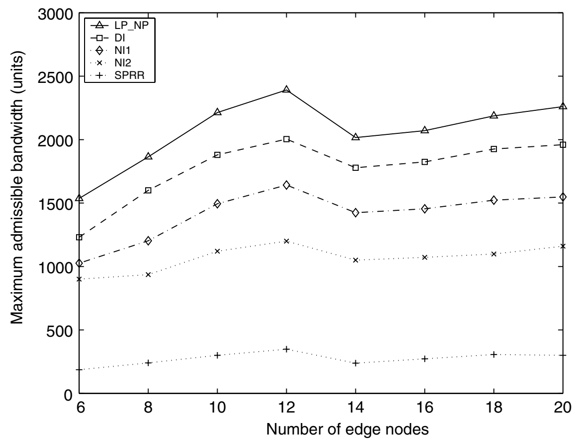


Fig. 8. Maximum admissible bandwidth vs. the number of edge nodes in Experiment 3.

Please cite this article in press as: J. Chu, C.-T. Lea, A restorable MPLS-based hose-model VPN network, Comput. Netw. (2007), doi:10.1016/j.comnet.2007.07.009

in Experiment 3. The reason is because the two NI schemes partition the link capacity without taking in account the hose-model traffic pattern. This results in the reduction of admissible capacity. Among the two, NI1, which maximizes the working capacity for the working traffic, performs better than NI2. We also observe that SPRR performs much worse than the other schemes. This is because SPRR uses shortest paths for working and restoration traffic. Load-balancing can not be done as efficiently as the other schemes.

4.3. Impact of hop-count limit on throughput

We use the Sprint backbone topology (Fig. 3) to study the throughput degradation due to adding a hop-count limit. Fig. 9 plots the maximum admissible bandwidth as the maximum allowable hop count of the restoration paths. As can be seen, the maximum admissible bandwidth does not change much beyond a hop-count of 6. The results allow us to make an intelligent tradeoff between the throughput and the restoration latency.

4.4. Dynamic construction of hose-model VPNs

The performance of the proposed approach shown in the previous sections can be further improved as described in this section. We compare the performance of the proposed approach with that of the conventional approaches in a dynamic environment where VPNs come and go. The performance measure we use is the *rejection ratio* which is defined as the percentage of the total VPN requests that is rejected.

In a dynamic VPN environment, previously proposed VPN provisioning algorithms, as pointed out in [15], have the drawback of computing working and link-restoration paths every time a new VPN is added. This is time consuming and can create a scalability problem if the frequency of adding and deleting VPNs is high. The non-blocking network approach does not have the same problem. For our approach, we will use one additional measure to further improve the performance presented in the previous sections. We use a server to record how much bandwidth of each link has been taken for existing VPNs. Recall that in the proposed approach, the paths are fixed. When a new VPN arrives, we use the formulation of Eq. (A.1) (given in Appendix A) to find the maximum amount of bandwidth required (i.e., the worst-case traffic pattern) along the paths for this new VPN. The computation required for this is much less than finding the optimal set of working and restoration paths of each new VPN. The throughput presented in the previous sections does not track this information and only uses the information of the ingress and egress nodes of the VPN to decide if the VPN can be admitted.

We conduct experiments on the Sprint backbone topology (Fig. 3). The VPN requests are generated following a Poisson process and the holding time is exponentially distributed. The VPN endpoints are randomly attached to the edge nodes, and the number of endpoints of each VPN is chosen randomly between 5 and 15. The ingress and egress

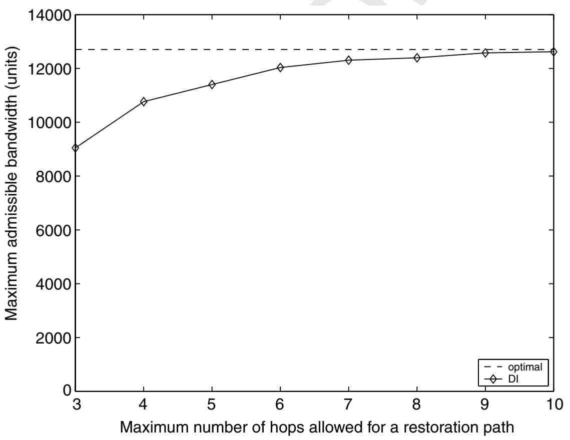


Fig. 9. Maximum admissible bandwidth vs. maximum hop count of a restoration path.

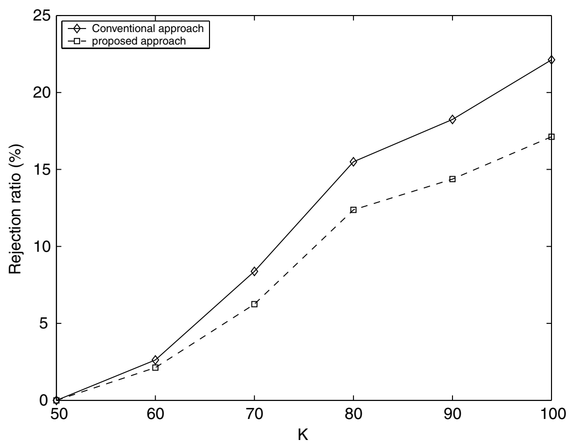


Fig. 10. The comparison of rejection ratio of the conventional and our proposed VPN construction approaches.

Please cite this article in press as: J. Chu, C.-T. Lea, A restorable MPLS-based hose-model VPN network, Comput. Netw. (2007), doi:10.1016/j.comnet.2007.07.009

bandwidth requirement of a VPN endpoint are assumed to be the same, and its value is chosen randomly between 1 and K, where K is a selected parameter that indicates the degree of variability of VPN bandwidth requirements. The results are shown in Fig. 10. As can be seen, the rejection ratios of the two approaches are very close if K is small. However, as K becomes larger, which implies greater variability of the bandwidth requirements of each VPN, our proposed approach has a better performance than the conventional approach.

5. Conclusions

In this paper, we presented a new restoration architecture for L3-VPN networks. We also presented a linear programming formulation for computing the optimal routing and link restoration for this architecture. Furthermore, we showed an efficient decomposition algorithm that can compute a near-optimal solution with much less computational overhead. The proposed architecture has many advantages, including no need to set up an external routing table, no need to set up restoration paths for a new VPN, and high throughput performance. The proposed decomposition algorithm is computation efficient and allows us to include a hop-count limit to bound the restoration latency of the VPN in case restoration occurs. The techniques developed in this paper can also be applied to other restoration networks.

Appendix A. Proof of Property 1

Property 1. Given $H = [(\tilde{\alpha}_1, \tilde{\beta}_1), \ldots, (\tilde{\alpha}_n, \tilde{\beta}_n)]$, routing x^e_{ij} and working capacity reservation $A(e)$ can satisfy constraint (2e) for all traffic matrices in M if and only if there exist non-negative weights $\pi_e(i)$ and $\lambda_e(i)$ for each $e \in E$ and $i \in Q$ such that

(i) $\sum_{i\in Q}\tilde{\alpha}_i\pi_e(i) + \sum_{i\in Q}\tilde{\beta}_i\lambda_e(i) \leqslant A(e)$ for each $e \in E$
(ii) $x^e_{ij} \leqslant \pi_e(i) + \lambda_e(j)$ for each $e \in E$ and every $i,j \in Q$

Proof. (*"only if" direction*): Let routing x^e_{ij} and working capacity reservation $A(e)$ satisfy constraints (2e) for all traffic matrices in M (i.e., $\sum_{ij}x^e_{ij}d_{ij} \leqslant A(e)$ for all $e \in E$ and $T \in M$). Consider a link e. The problem of finding $T = \{d_{ij}\}$ that maximizes link load on e can be formulated as the following linear programming problem.

$$\max \sum_{ij} x^e_{ij} d_{ij} \tag{A.1a}$$

$$\text{s.t.} \sum_{j\in Q} d_{ij} \leqslant \tilde{\alpha}_i, \quad i \in Q \tag{A.1b}$$

$$\sum_{i\in Q} d_{ij} \leqslant \tilde{\beta}_j, \quad j \in Q \tag{A.1c}$$

$$d_{ij} \geqslant 0, \quad i,j \in Q \tag{A.1d}$$

where constraints (A.1b) and (A.1c) are the ingress and egress bandwidth constraints. The *dual* of the above LP problem for link e is:

$$\min \sum_i \tilde{\alpha}_i \pi_e(i) + \sum_i \tilde{\beta}_i \lambda_e(i) \tag{A.2a}$$

$$\text{s.t. } \pi_e(i) + \lambda_e(j) \geqslant x^e_{ij}, \quad i,j \in Q \tag{A.2b}$$

$$\pi, \lambda \geqslant 0 \tag{A.2c}$$

Since $\sum_{ij}x^e_{ij}d_{ij} \leqslant A(e)$, the dual for any link e must have optimal value $\leqslant A(e)$. Therefore, the objective function of the dual satisfies (i). Requirement (ii) is trivially satisfied by the dual problem constraint (A.2b).

(*"if" direction*): Let x^e_{ij} be a routing, and $T = \{d_{ij}\}$ be any valid traffic matrix. Also let $\pi_e(i)$ and $\lambda_e(i)$ be the weights satisfying requirements (i)-(ii). Consider a link e. From (ii), we have

$$x^e_{ij} \leqslant \pi_e(i) + \lambda_e(j)$$

Summing over all node pairs (i,j), we have

$$\begin{aligned}\sum_{i,j\in Q} x^e_{ij} d_{ij} &\leqslant \sum_{i,j\in Q} [\pi_e(i) + \lambda_e(j)] d_{ij} \\ &= \sum_{i\in Q} \pi_e(i) \sum_{j\in Q} d_{ij} + \sum_{j\in Q} \lambda_e(j) \sum_{i\in Q} d_{ij} \\ &\leqslant \sum_{i\in Q} \tilde{\alpha}_i \pi_e(i) + \sum_{i\in Q} \tilde{\beta}_i \lambda_e(i)\end{aligned}$$

The last inequality comes from the constraints imposed by H (i.e., $\sum_j d_{ij} \leqslant \tilde{\alpha}_i$ and $\sum_i d_{ij} \leqslant \tilde{\beta}_j$). From (i), we have

$$\sum_{i,j\in Q} x^e_{ij} d_{ij} \leqslant \sum_{i\in Q} \tilde{\alpha}_i \pi_e(i) + \sum_{i\in Q} \tilde{\beta}_i \lambda_e(i) \leqslant A(e)$$

This means that for any traffic matrix constrained by H, the working traffic on any link is at most $A(e)$. □

Appendix B. Concavity of $\theta(\Lambda)$

Property 2. Let R denote the set of feasible Λ. Then $\theta(\Lambda)$ is a concave function on R.

Please cite this article in press as: J. Chu, C.-T. Lea, A restorable MPLS-based hose-model VPN network, Comput. Netw. (2007), doi:10.1016/j.comnet.2007.07.009

Proof. The property can be proven from the dual form of Eq. (5).

Dual Problem of Eq. (5):

$$\min \sum_{e \in E} \omega_e A(e) \tag{B.1a}$$

$$\text{s.t. } \sigma_{ij}^{u} - \sigma_{ij}^{v} + \mu_{ij}^{e} \geqslant 0, \quad i,j \in Q, \;\; e = (u,v) \in E \tag{B.1b}$$

$$\sum_{i,j \in Q} \sigma_{ij}^{j} \geqslant 1 \tag{B.1c}$$

$$\sigma_{ij}^{i} = 0, \quad i,j \in Q \tag{B.1d}$$

$$\tilde{\alpha}_i \omega_e - \sum_{j \in Q} \mu_{ij}^{e} \geqslant 0, \quad i \in Q, \;\; e \in E \tag{B.1e}$$

$$\tilde{\beta}_j \omega_e - \sum_{i \in Q} \mu_{ij}^{e} \geqslant 0, \quad j \in Q, \;\; e \in E \tag{B.1f}$$

$$\omega, \mu, \sigma \geqslant 0 \tag{B.1g}$$

Let $\widetilde{\Lambda}, \overline{\Lambda} \in R$, and ρ be in the range $0 \leqslant \rho \leqslant 1$. Let $\Lambda = \rho\widetilde{\Lambda} + (1-\rho)\overline{\Lambda}$. Then according to the dual form of Eq. (5),

$$\begin{aligned}\theta(\Lambda) = \min\{\sum_{e \in E} \omega_e A(e) : \\ \text{s.t. B.1b–B.1g}\} \\ = \min\{\sum_{e \in E} \omega_e[\rho\widetilde{A}(e) + (1-\rho)\overline{A}(e)] : \\ \text{s.t. B.1b–B.1g}\} \\ = \min\{\rho\sum_{e \in E} \omega_e\widetilde{A}(e) + (1-\rho)\sum_{e \in E} \omega_e\overline{A}(e) : \\ \text{s.t. B.1b–B.1g}\}\end{aligned}$$

Let

$$\begin{aligned}\theta(\widetilde{\Lambda}) = \min\{\sum_{e \in E} \omega_e\widetilde{A}(e) : \\ \text{s.t. B.1b–B.1g}\} \\ \theta(\overline{\Lambda}) = \min\{\sum_{e \in E} \omega_e\overline{A}(e) : \\ \text{s.t. B.1b–B.1g}\}\end{aligned}$$

From linear programming theory, $\theta(\Lambda) \geqslant \rho\theta(\widetilde{\Lambda}) + (1-\rho)\theta(\overline{\Lambda})$ obviously holds. Hence, $\theta(\Lambda)$ is a concave function on R. □

References

[1] E. Rosen, Y. Rekhter, BGP/MPLS VPNs, RFC 2547, March 1999.

[2] N.G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan, J.E.V. der Merwe, A Flexible Model for Resource Management in Virtual Private Networks, ACM Sigcomm, San Diego, CA, USA, 1999. August.

[3] A. Kumar, R. Rastogi, A. Silberschatz, B. Yener, Algorithms for Provisioning Virtual Private Networks in the Hose Model, ACM Sigcomm, Cambridge, MA, USA, 2001. August.

[4] Alpar Jüttner, Istvan Szabo, Aron Szentesi, On bandwidth efficiency of the hose resource management model in virtual private networks, IEEE Infocom 2003, San Francisco, April 2003.

[5] Thomas Erlebach, Maurice Rüegg, Optimal bandwidth reservation in hose-model VPNs with multi-path routing, IEEE Infocom 2004, March 2004.

[6] G. Italiano, S. Leonardi, G. Oriolo, Design of networks in the hose model, in: Proceedings of the 3rd Workshop on Approximation and Randomization Algorithms in Communication Networks (ARACNE), pp. 65–76, 2002.

[7] G. Swallow, P. Pan, A. Atlas, Fast reroute extensions to RSVP-TE for LSP tunnels, RFC 4090, May 2005.

[8] V. Sharma et al., Framework for multi-protocol label switching (MPLS)-based recovery, RFC 3469, February 2003.

[9] K. Murakami, H.S. Kim, Optimal capacity and flow assignment for self-healing ATM networks based on line and end-to-end restoration, IEEE/ACM Trans. Network. 6 (1998).

[10] S. Ramamurthy, B. Mukherjee, Survivable WDM mesh networks, part I – protection, IEEE Infocom 1999.

[11] S. Ramamurthy, B. Mukherjee, Survivable WDM mesh networks, part II – restoration, IEEE ICC 1999.

[12] G. Italiano, R. Rastogi, B. Yener, Restoration algorithms for virtual private networks in the hose model, IEEE Infocom 2002, June 2002.

[13] M. Kodialam, T.V. Lakshman, Sudipta Sengupta, Throughput guaranteed restorable routing without traffic prediction, ICNP '06, 2006.

[14] Load balancing with cisco express forwarding, Cisco application note.

[15] Jian Chu, Chin-Tau Lea, New architecture and algorithms for fast construction of hose-model VPNs, IEEE/ACM
Q1 Transactions on Networking, in press.

[16] Jian Chu, Chin-Tau Lea, Routing and restoration in networks with hose-model traffic patterns, HPSR'05, May 2005.

[17] A. Altin, E. Amaldi, P. Belotti, M.C. Pinar, Provisioning virtual private networks under traffic uncertainty, Tech. Report, no. 16, DEI, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy, January 2005.

[18] CPLEX, ILOG CPLEX Division, <http://www.cplex.com/>.

[19] P. Wolfe, H. Crowder, Validation of subgradient optimization, Math. Program. 6 (1974) 62–68.

[20] N. Gary, J. Könemann, Faster and simpler algorithms for multicommodity flow and other fractional packing problems, in: Proceedings of the 39th Annual Symposium on Foundations of Computer Science, pp. 300–309, 1998.

[21] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows, Prentice-Hall, 1993.

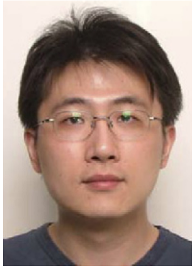
[22] Sprint IP Webpage, <http://www.sprint.net>.

[23] A. Nemirovsky, D. Yudin, Problem Complexity and Method Efficiency in Optimization, John Wiley and Sons, 1983.

[24] C. Beltran, F.J. Heredia, An effective line search for the subgradient method, J. Optimizat. Theory Appl. 125 (2005).

Please cite this article in press as: J. Chu, C.-T. Lea, A restorable MPLS-based hose-model VPN network, Comput. Netw. (2007), doi:10.1016/j.comnet.2007.07.009

COMPNW 3627 ARTICLE IN PRESS No. of Pages 13, Model 3+
25 August 2007 Disk Used



Jian Chu received the B.S. degree in Electronic Engineering from Shanghai Jiao Tong University, P.R. China, and Ph.D. degree in Electronic and Computer Engineering from Hong Kong University of Science and Technology.

He works as a postdoctoral fellow in the Internet Switching Technology Laboratory, Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology. His research interests include routing protocol design and analysis, network survivability, and algorithms for related optimization problems.

Chin-Tau Lea received a B.S and a M.S. degree from the National Taiwan University in 1976 and 1978, and a Ph.D. degree from the University of Washington, Seattle, in 1982, all in electrical engineering. He is now a professor at the Hong Kong Univ of Science and Technology which he joined in 1996. Prior to that, he was with AT&T Bell Labs from 1982 to 1985 and with the Georgia Institute of Technology from 1985 to 1995.

His research interests are in the areas of switching and networking. He is on the editorial board of IEEE JSAC and of Computer Networks. He received the DuPont Young Faculty Award from Georgia Tech in 1987, the IEEE Jack Neubauer Paper Award in 1998, and the School of Engineering Teaching Award from HKUST in 1998. He also holds five U.S. patents.

Please cite this article in press as: J. Chu, C.-T. Lea, A restorable MPLS-based hose-model VPN network, Comput. Netw. (2007), doi:10.1016/j.comnet.2007.07.009