Virtual-Flow Multipath Algorithms for MPLS

Dario Pompili*

Broadband and Wireless Networking Laboratory,
School of Electrical and Computer Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332, USA
E-mail: dario@ece.gatech.edu
*Corresponding author

Caterina Scoglio

Department of Electrical and Computer Engineering,
Kansas State University,
Manhattan, KS 66506, USA
E-mail: caterina@eece.ksu.edu

Charles A. Shoniregun

School of Computing and Technology,
University of East London,
Docklands Campus, University Way,
London E16 2RD, UK
E-mail: C.Shoniregun@uel.ac.uk

Abstract: Multiprotocol Label Switching (MPLS) can improve the routing efficiency of Internet Protocol (IP) networks through its intrinsic Traffic Engineering (TE) capabilities. In this paper, a centralised and a distributed virtual-flow routing algorithms are proposed, which aggregate IP flows entering the MPLS domain and optimally partition them among virtual flows that are routed on multiple paths. The routing algorithms dynamically select multiple Label Switched Paths (LSPs), taking into account the available bandwidth of links in the network to balance the traffic load and avoid network congestion. The multipath routing problem is formulated as a Multicommodity Network Flow (MCNF) problem, and is solved by implementing online the Dantzig–Wolfe decomposition method. The proposed multipath algorithms are shown through simulations to outperform single-path routing solutions, such as the Constraint Shortest Path First (CSPF) and the Bandwidth-Based Shortest Path (BSPR) routing algorithms.

Keywords: MPLS; IP traffic engineering; multi-commodity network flow problem; quality of service.

Reference to this paper should be made as follows: Pompili, D., Scoglio, C. and Shoniregun, C.A. (2007) ‘Virtual-Flow Multipath Algorithms for MPLS’, *Int. J. Internet Technology and Secured Transactions*, Vol. 1, Nos. 1/2, pp.1–19.

Biographical notes: Dario Pompili graduated in Telecommunications Engineering (summa cum laude) from the University of Rome 'La Sapienza', Italy, in 2001. In 2004, he earned from the same university the PhD Degree in System Engineering. In 2003, he worked on sensor networks at the Broadband and Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta, as a Visiting Researcher. Currently, he is pursuing the PhD Degree in Electrical Engineering at the Georgia Institute of Technology, under the guidance of Dr. I.F. Akyildiz. His main research interests are in wireless ad hoc and sensor networks, underwater acoustic sensor networks and satellite networks.

Caterina Scoglio graduated in Electrical Engineering (summa cum laude) from the University of Rome 'La Sapienza', Italy, in 1987. In 1988, she received a Post-graduate Degree in Mathematical Theory and Methods for System Analysis and Control from the same university. From 1987 to 2000, she was with Fondazione Ugo Bordoni, Rome, as a research scientist. From 2000 to 2005, she was with the Broadband and Wireless Networking Laboratory, Georgia Institute of Technology, as a Research Engineer. Currently, she is an Associate Professor in the Electrical and Computer Engineering Department at Kansas State University. Her research interests include optimal design and management of overlay networks.

Charles A. Shoniregun is reader in Computing. He has taught in many universities and colleges in the UK and abroad. He is a member of the research committee at the School of Computing and Technology and an elected member of the University of East London Academic Board. He is also a member of many professional bodies and the Editor-in-Chief of the *International Journal for Infonomics* (*IJI*), *International Journal of Internet Technology and Secured Transactions* (*IJITST*), an author, co-author, adjunct and distinguished Visiting Professor in 'Applied Internet Security and Information Systems' and External Assessor/Examiner to Manchester Metropolitan University, UK; London Metropolitan University, UK; Wessex Institute of Technology, UK; and Florida International University, USA.

1 Introduction

With the rapid growth of the Internet and the emergence of new demanding services, Internet Service Providers (ISPs) are facing the challenge of providing Quality of Service (QoS) to end users. To this end, the simplest approach is network bandwidth *overprovisioning*. This approach, however, is neither efficient nor practical for the following reasons:

- it is not economically viable, especially in Wide Area Networks (WANs) where the cost of bandwidth is higher than in Local Area Networks (LANs)
- it can not prevent short-term congestion due to the unpredictable nature of IP traffic
- it can not meet the QoS requirements of heterogeneous applications, e.g., it can not guarantee delay bounds for time-sensitive applications
- it can not distinguish network *transparency*, i.e., the capability of the network not to impair the quality of user connections, and network *accessibility*, i.e., the capability of the network to offer guaranteed services to end users (Bonald et al., 2002).

For these reasons, IP TE has become an essential requirement for ISPs in order to optimise the utilisation of existing network resources and provide QoS to end users. Several researchers have proposed to integrate TE capabilities to IP networks using single shortest path routing algorithms based on measured link cost metrics, e.g., available bandwidth, link delay, and delay jitter (Kar et al., 2000; Awduche et al., 1999). Although these routing algorithms are simple to implement and may provide an effective solution under some network conditions (Fortz and Thorup, 2000), in most cases shortest path routing algorithms can not efficiently utilise the network resources, and offer limited control capabilities for TE, since they rely on a single path between source-destination pairs. Moreover, they hardly provide *load balancing* in the network when the traffic to be accommodated has heterogeneous characteristics in terms of required bandwidth and QoS (Lee et al., 2002). The lack of load balancing may also impair *fairness*[1] among connections.

An effective approach to prevent network bottlenecks is to keep the average link utilisation low by distributing data flows among the least-loaded links. The literature reminds us that the problem of minimising the maximum link utilisation in the network can be efficiently solved through the *MCNF* formulation, whose objective is to optimally split the traffic over multiple paths between source-destination pairs (Wang and Wang, 1999; Bertsekas and Gallager, 1992). Since multipath routing provides network load balancing, the network resources are more efficiently utilised than in the case of single-path routing. Moreover, multipath routing can satisfy end user demands that a single-path strategy would not be able to, e.g., the available capacity on a link may not be sufficient to accommodate the user demand. In multipath routing, in fact, the network can split the data traffic into smaller data flows, which can then be routed on different paths. This way, efficient network resource utilisation can be ensured, and the overall required capacity may be also reduced.

If we consider, however, one single connection,[2] multipath routing algorithms may require more network bandwidth capacity than single-path routing algorithms, because they may use paths that are not the shortest ones. Therefore, a multipath routing strategy is to be preferred to a single-path strategy if overall a higher number of connections are admitted into the network, thus compensating for the possible higher amount of bandwidth utilised to route single connections. To achieve this, a maximum path-cost constraint should be incorporated into multipath TE schemes in order to maximise the admitted connections into the network, i.e., to minimise the *rejection ratio*.[3] Another issue in multipath routing algorithms is that out-of-order packet delivery may occur during the data transmission. In order to prevent the impairment of throughput performance, and to bound the complexity of ad hoc algorithm aiming at reordering packets, the total number of out-of-order packets should be limited.

One of the most prominent technology that can improve the routing efficiency of IP networks through its intrinsic TE capabilities on heterogeneous network infrastructures is *MPLS* (Xiao et al., 2000; Awduche and Jabbari, 2002). MPLS offers sophisticated path management, traffic assignment, and network management functionalities through appropriate path selection mechanisms that enable fast data forwarding through the network. However, the existing path selection mechanisms in MPLS networks are not efficient enough to solve the above mentioned problems.

To address the discussed challenges, we propose two Virtual-Flow Multipath routing Algorithm, a centralised (VFMA-C) and a distributed (VFMA-D) solution. Both algorithms formulate the virtual-flow multipath routing problem as a *MCNF* problem

(Ahuja et al., 1993), whose objective is to aggregate the IP traffic entering the MPLS domain at the ingress router and optimally split it into multiple virtual flows. These flows are then separately routed towards the egress routers, while guaranteeing their QoS requirements and complying with the network constraints.

We also introduce the *virtual-flow* concept, which allows our proposed routing algorithms to have a smaller packet-level granularity, in contrast to the coarser flow-level granularity of traditional approaches. This packet-level granularity allows the network to smooth the heterogeneity of traffic flows, which leads to better leverage the network resources and avoid bottleneck. In addition, since there are no constraints on the *grooming* of the virtual flows, i.e., their bandwidth is not forced to be selected among a discrete set of predefined values – which would lead to quantisation problems – the MCNF optimisation problem is not cast as an Integer Linear Problem (ILP), which is proven to be NP-complete (Ahuja et al., 1993). Another advantage of the virtual-flow concept is that many IP packets that are assigned to the same virtual flow are encapsulated into few large MPLS packets. This results in decreasing the MPLS overhead in the data transmission.

Figures 1 and 2 show the standard multipath traffic partitioning in a MPLS domain and the enhanced MCNF-based virtual-flow multipath partitioning (VFMA), respectively. While the former keeps the structure of the N_{ij}^{q} incoming IP flows $f_{ij}^{q}(n), \forall n = 1,\ldots,N_{ij}^{q},$ and routes them from an ingress router *i* to an egress router *j* on different paths according to their Class of Service (CoS) $q = 1, \ldots, Q$, the latter dynamically *pre-aggregates* all the flows with the same CoS into an *aggregate flow* $F_{ij}^{q} = \bigcup_{n} f_{ij}^{q}(n),$ as shown in Figure 2. This aggregate flow is then optimally split into M_{ij}^{q} *virtual flows* $f_{ij}^{*q}(m), \forall m = 1,\ldots,M_{ij}^{q},$ which are potentially routed on M_{ij}^{q} different paths towards the destination router *j*. The VFMAs determine the optimal number of paths, the best set of paths to split the aggregate flow, and the optimal share of traffic to be routed on each path by solving an ad hoc MCNF problem. Note that IP packets belonging to the same virtual flow $f_{ij}^{*q}(m)$ may come from different IP traffic flows; consequently, packets from the same IP flow before the aggregation phase may end up in *different* virtual flows, i.e., may be switched on different paths inside the MPLS domain. Therefore, if packets are needed to be in-order when they exit the MPLS domain, re-ordering is required at the destination egress router before decapsulation takes place. This would not be a heavy-computational task, however, since in general few IP packets should be re-ordered, provided that they are in-order when they arrive at the ingress router; in fact, all the selected paths must be compliant with the QoS contract of the aggregated flow, e.g., must respect the maximum delay allowed to each IP packet flowing through the MPLS domain. Moreover, the minimisation of the utilisation of each link in the network, which is the key objective of our routing algorithms to achieve load balancing, leads to the minimisation of the packet queueing delays as well, as will be shown in the following sections. For the reasons stated, the total number of out-of-order packets can be kept under control, thus bounding the complexity of ad hoc algorithms aiming at reordering packets at the egress routers.

Figure 1 Standard multipath routing partitioning in a MPLS domain

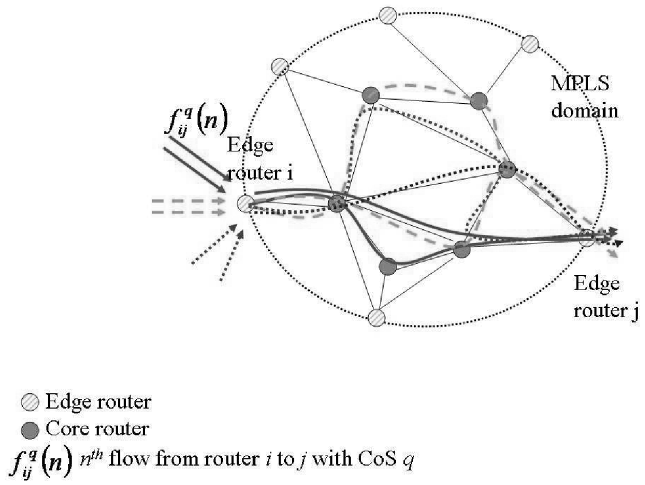
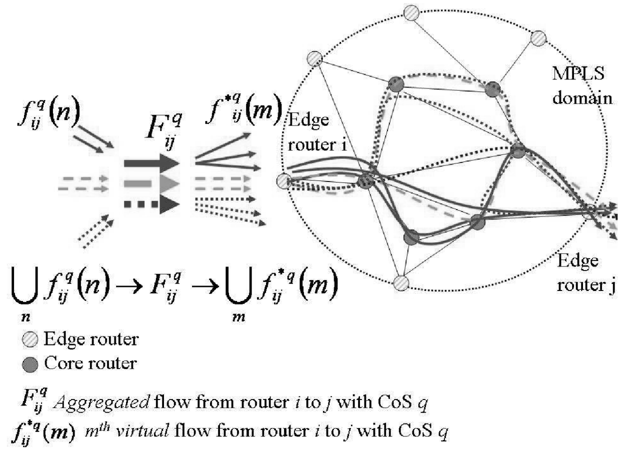


Figure 2 Enhanced MCNF-based virtual-flow multipath partitioning (VFMA)


$f_{ij}^{q}(n)$
F_{ij}^{q}
$f_{ij}^{*q}(m)$
Edge router i
MPLS domain
Edge router j
$\bigcup_{n} f_{ij}^{q}(n) \rightarrow F_{ij}^{q} \rightarrow \bigcup_{m} f_{ij}^{*q}(m)$
▨ Edge router
● Core router
F_{ij}^{q} *Aggregated* flow from router i to j with CoS q
$f_{ij}^{*q}(m)$ m^{th} *virtual* flow from router i to j with CoS q

We compare the performance of a distributed and centralised routing solutions for multipath routing in MPLS domains by means of extensive simulation experiments. We show that both the proposed multipath algorithms outperform single-path routing solutions (Kar et al., 2000; Awduche et al., 1999) such as the CSPF and the BSPR routing algorithms, since they can more flexibly exploit the available network resources.

The main objectives of our VFMAs are:

- guarantee efficient accommodation of heterogenous flow requests, which may have consistently different bandwidth and QoS requirements
- provide load balancing in order to efficiently exploit the available network resources and avoid bottlenecks
- minimise the MPLS overhead by aggregating IP data flows with the same ingressegress pair and CoS, and encapsulating their packets in few large MPLS packets.

The remainder of the paper is organised as follows. In Section 2, we provide the background and review the literature. In Section 3, we introduce the network and cost models, and present the arc-path form of the MCNF problem. In Section 4, we describe the Dantzing–Wolfe decomposition method. In Section 5, we present the proposed centralised and distributed multipath algorithms, whose performance is evaluated in Section 6. Finally, in Section 7, we draw the main conclusions and point future research directions.

2 Background and related work

MPLS is a connection-oriented label swapping technology that supports Constraint-Based Routing (CBR) algorithms (Davie and Rekhter, 2000), thanks to its ability to implement explicit route functionality. MPLS offers sophisticated path management, traffic assignment and network management functionalities through appropriate path selection mechanisms that enable fast data forwarding through the network. A MPLS domain is constituted of Label Switching Routers (LSRs), i.e., *edge routers* (*ingress* and *egress routers*) and *core routers*, as shown in Figure 1. The ingress router encapsulates IP packets with labels that are then used to forward the packets along the LSP, passing through the core routers. The egress router decapsulates IP packets before sending them out from the MPLS domain. Before mapping packets to a LSP, LSPs are setup between an ingress router and an egress router by a signalling protocol such as the Label Distribution Protocol (LDP) or the extended resource reservation protocol (RSVP-TE) for LSP tunnels.

In the literature, there are several research efforts addressing TE in MPLS networks. Traditional MPLS TE frameworks have been assumed to use single LSPs (Kar et al., 2000; Awduche et al., 1999; Juttner et al., 2000; Girish et al., 2000). However, the single LSPs in the network may result in network load unbalancing. In order to provide efficient network resource utilisation, multiple LSPs between an ingress router and an egress router are proposed in Lee et al. (2000), Saito et al. (2000), Elwalid et al. (2001), Villamizar (1999) and Dinan et al. (2000). Specifically, in Lee et al. (2000), a multiobjective formulation of the TE problem for the minimisation of the maximum link utilisation and the minimisation of the total network resource usage is proposed. In Lee et al. (2000), however, only the centralised approach to the TE problem is presented, which, in practical applications, may not be feasible due to inaccurate information on network traffic. In Saito et al. (2000), the authors develop a TE method that utilises multiple multipoint-to-point LSPs, in which multiple routes are used as backup routes in case of network failures. In Elwalid et al. (2001), a network load balancing protocol called MATE (MPLS Adaptive TE) is presented. The main objective of MATE is to avoid network congestion by balancing the network load among multiple LSPs between an ingress and an egress LSR. In MATE, the ingress router transmits probe packets periodically to the egress router, which sends the probe packets back to the ingress router. Based on the information in the receiving probe packets, the ingress router can determine the LSP characteristics and balance the network load among them. MATE is not designed, however, for bandwidth guaranteed services, and may not scale well when many ingress-egress pairs are considered. In Villamizar (1999), the authors propose MPLS-OMP, an Optimised MultiPath algorithm in which the distribution of network load among multiple paths is determined by utilising a hash computation for each path.

Finally, in Dinan et al. (2000), a stochastic framework for traffic partitioning problem among LSPs is presented. In this framework, network load balancing is provided using a set of pre-established parallel edge-disjoint LSPs, with the objective of minimising the overall traffic latency. The proposed model, however, is difficult to implement, and relies on many assumptions that may not hold in realistic network environments.

3 Multicommodity network flow problem

The MCNF problem arises in a wide variety of important real-world applications such as communications, logistics, manufacturing, and transportation. A commodity represents the entity that needs to be 'shipped' from the source to the destination node using the underlying network, and is defined based on the application context. The objective of the MCNF problem is to minimise the total cost to 'ship' all the commodities to their destinations, while satisfying the capacity constraints associated with the underlying network resources. In this work, the problem of selecting multiple LSPs at each ingress LSR is formulated as a MCNF problem. In this formulation, the commodity represents connection requests of a particular Forwarding Equivalence Class (FEC), which maps the CoS of the packets of the connection between a source node (ingress LSR) and a destination node (egress LSR). In Sections 3.1 and 3.2, we introduce the network and cost models, respectively, which will be then used in Section 3.3 to cast the arc-path formulation of the MCNF problem.

3.1 Network model

In this section, we introduce the notations and variables that are used in the MCNF problem formulation:

- $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is a *directed graph* modelling the MPLS network, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of links
- $\mathcal{K}^q$ is the set of the commodities representing the aggregated connection requests with CoS $q = 1, \ldots, Q$, e.g., $F_{ij}^q, i, j \in \mathcal{N}$, in Figure 2. These commodities will be indexed with $k = 1, \ldots, K^q$, where K^q is the cardinality of $\mathcal{K}^q$, i.e., $K^q = |\mathcal{K}^q|$
- s_k^q and d_k^q are the source (ingress LSR) and the destination (egress LSR) nodes of the requested LSP for the connection $k \in \mathcal{K}^q$, respectively
- $u_{ij}^{q,\text{tot}}$ accounts for both the total bandwidth allocated to CoS q on link (i, j), and the total resources required by node i to handle packets belonging to CoS q
- c_{ij}^q is the cost of link (i, j) associated with CoS q, which will be detailed in Section 3.2
- B_k^q is the amount of bandwidth demanded by the connection request $k \in \mathcal{K}^q$
- $\mathcal{P}_k^q$ is the set of all feasible paths from the source s_k^q to the destination node d_k^q for connection request $k \in \mathcal{K}^q$

- $\delta_{ij}^{k,q}(p)$ is a binary variable equal to 1 iff path $p \in \mathcal{P}_k^q$ includes link (i,j), and 0 otherwise
- $f_k^q(p)$ is the fraction of the demanded bandwidth B_k^q of the connection request $k \in \mathcal{K}^q$ assigned to path $p \in \mathcal{P}_k^q$.

3.2 Cost model

A cost c_{ij}^q is associated to link (i,j) for each CoS $q = 1, \ldots, Q$ the network can support, as introduced in Pompili et al. (2004),

$$c_{ij}^q = \begin{cases} \dfrac{1}{(1-\rho_{ij}^q)+\in} & \text{if } u_{ij}^q \geq B_k^q \\ \infty & \text{if } u_{ij}^q < B_k^q \end{cases}, \tag{1}$$

where u_{ij}^q is the available bandwidth of link (i,j), also *arc*(i,j) in the following, associated to CoS q, and ρ_{ij}^q is the *link utilisation* associated with CoS q, which is defined as,

$$\rho_{ij}^q = \frac{u_{ij}^{q,\text{tot}} - (u_{ij}^q - B_k^q)}{u_{ij}^{q,\text{tot}}}. \tag{2}$$

Note that in equation (1) $\in >$ (maxHop/δ) forces a path to be feasible in the case link (i,j) has $\rho_{ij}^q = 1$, i.e., $u_{ij}^q = B_k^q$, where maxHop is the maximum number of hops for each path and δ is the greatest number that can be represented by a practical software implementation.

It is worth observing that minimising the cost-metric in equation (1) automatically leads to a *twofold objective*:

- minimising the link utilisation
- minimising the average queueing delay associated with link (i,j), if a $\mathcal{M}/\mathcal{M}/1$ queue model is assumed (Kleinrock, 1975).

3.3 Arc-path form of the Multicommodity Network Flow problem

The MCNF problem extends the definition of the Single Commodity Network Flow (SCNF) problem, whose objective is to ship one commodity from its origin node to its destination node through the network in some primal fashion, e.g., along a shortest path or via a minimum-cost flow. A MCNF problem, in fact, can be viewed as several independent SCNF problems, if we disregard the bundle constraints, i.e., the arc capacity constraints that tie together flows of different commodities passing through the same arcs. The MCNF problem is a *Linear Programming* (LP) problem, which can be formulated in a arc-path form. The arc-path form of the minimum-cost MCNF problem is based on the *flow decomposition theorem of network flows* (Ahuja et al., 1993), which states that any arc-flow solution can be decomposed into path and cycle flows.

For each commodity $k \in \mathcal{K}^q$, which represents the aggregated incoming flow F_{ij}^q with CoS q, let $\mathcal{P}_k^q$ denote the set of all feasible paths from the source node s_k^q

(ingress router *i*) to the destination node d_k^q (egress router *j*) in the underlying MPLS network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, finally, let $f_k^q(p)$ be the units of flow on path $p \in \mathcal{P}^q$ and $c_k^q(p)$ the per-unit cost of flow on path *p* using c_{ij}^q as the arc cost. We can now formulate the arc-path form of the MCNF problem as follows.

$\mathbf{P}_{\text{Arc–Path}}$: *Arc-path form of the MCNF problem*

$$\text{Given}: \mathcal{K}^q, c_{ij}^q, u_{ij}^{q,\text{tot}}, \mathcal{P}_k^q, s_k^q, d_k^q, B_k^q, \quad \forall k \in \mathcal{K}^q, \quad \forall q \in \mathcal{Q}$$

$$\text{Find}: \ \delta_{ij}^{k,q}(p), f_k^q(p), \quad \forall p \in \mathcal{P}_k^q, \quad \forall q \in \mathcal{Q}, \quad \forall (i,j) \in \mathcal{E}$$

$$\text{Minimise}: \sum\nolimits_{k \in \mathcal{K}^q} \sum\nolimits_{p \in \mathcal{P}_k^q} c^{k,q}(p) \cdot f_k^q(p)$$

Subject to:

$$c^{k,q}(p) = \sum\nolimits_{(i,j) \in \mathcal{E}} \delta_{ij}^{k,q}(p) \cdot c_{ij}^q, p \in \mathcal{P}_k^q, \quad \forall k \in \mathcal{K}^q, \quad \forall q \in \mathcal{Q} \tag{3}$$

$$\sum\nolimits_{k \in \mathcal{K}^q} \sum\nolimits_{p \in \mathcal{P}_k^q} \delta_{ij}^{k,q}(p) \cdot f_k^q(p) \le u_{ij}^{q,\text{tot}}, \quad \forall (i,j) \in \mathcal{E}, \quad \forall q \in \mathcal{Q} \tag{4}$$

$$\sum\nolimits_{p \in \mathcal{P}_k^q} f_k^q(p) = B_k^q, \quad \forall k \in \mathcal{K}^q, \quad \forall q \in \mathcal{Q} \tag{5}$$

$$f_k^q(p) \ge 0, \quad \forall p \in \mathcal{P}_k^q, \quad \forall k \in \mathcal{K}^q, \quad \forall q \in \mathcal{Q} \tag{6}$$

This formulation has a collection of $K^q = |\mathcal{K}^q|$ *bundle constraints* (equation (4)), which state that for each arc (i, j) the sum of the path flows passing through it be at most the capacity of the arc $u_{ij}^{q,\text{tot}}$ and of $\sum\nolimits_{k \in \mathcal{K}^q} |\mathcal{P}_k^q|$ network constraints (equation (5)), which state that for each commodity the total flow on all the paths connecting the source s_k^q and destination node d_k^q equal the demand B_k^q. Finally, constraints (equation (6)) assure that each path flow be not negative.

Complexity: For each $q = 1, \ldots, Q$, $\mathbf{P}_{\text{Arc-Path}}$ contains $|\mathcal{E}| \cdot |\mathcal{K}^q| + \sum\nolimits_{k \in \mathcal{K}^q} |\mathcal{P}_k^q|$ constraints, in addition to the nonnegativity restrictions imposed on the path flow values (equation (6)).

4 Dantzing–Wolfe decomposition method

In this section, we describe the *Dantzing–Wolfe decomposition method*, which our VFMAs apply to the arc-path form of the MCNF problem to decrease its computational complexity. The Dantzig–Wolfe decomposition method (DW) is a general-purpose approach to decompose problems, such as the MCNF, that have a set of 'easy' constraints (*network flow constraints*) and a set of 'hard' constraints (*bundle constraints*) (Ahuja et al., 1993). This decomposition is based on an iterative process where, in each iteration, a new instance of the problem is solved. The DW method involves $\sum\nolimits_{q=1}^{Q} K^q$ different *decision maker* agents and one *coordinator* agent; the coordinator solves the Restricted Master Problem (RMP), which is the arc-path form of the MCNF problem restricted to a subset $\mathcal{P}^*$ of the paths instead of using all the feasible paths $\mathcal{P}$, as in $\mathbf{P}_{\text{Arc-Path}}$, i.e., in general $|\mathcal{P}^*| < |\mathcal{P}|$. These paths in $\mathcal{P}^*$ are initially selected by the coordinator so that their costs do not exceed the cost of the shortest path by more than a fixed percentage *r*. This is done to reduce the complexity of the algorithm. In particular, the lower *r*, the lower the complexity of the restricted problem, but the higher the number of iterations required to converge to the optimal solution. The coordinator agent solves the restricted master problem by applying the DW decomposition, and then, according to the feedback from the decision-maker agents, it determines whether the solution to this restricted problem is optimal for the master problem as well.

In order to determine whether the solution is optimal, the coordinator communicates to the decision makers the optimal set of *simplex multipliers* of the restricted master problem, i.e., it broadcasts an *arc price* w_{ij}^q associated with each arc $(i,j) \in \mathcal{E}$ and CoS $q \in \mathcal{Q}$, and a *path cost* σ_k^q associated with each commodity $k \in \mathcal{K}^q$. After receiving these multipliers, the decision maker agent responsible for commodity $k \in \mathcal{K}^q$ determines the least-cost way of shipping B_k^q units from the source s_k^q to the destination node d_k^q for commodity $k \in \mathcal{K}^q$, assuming that each arc (i,j) has an associated price of w_{ij}^q in addition to its arc cost c_{ij}^q. Therefore, each of the $\sum_{q=1}^{Q} K^q$ decision maker agents solves a shortest-path problem using an additional arc cost of w_{ij}^q on each arc (i,j). This shortest-path problem can be cast as follows.

P$_{\text{Dantzig–Wolfe}}$: *Dantzig–Wolfe decomposition method*

$$\text{Given}: s_k^q, d_k^q, B_k^q, \quad \forall k \in \mathcal{K}^q, \quad \forall q \in \mathcal{Q}$$

$$\text{Find}: \quad \forall p \in \mathcal{P}_k^q, \quad \forall q \in \mathcal{Q}$$

$$\text{Minimise}: \sum\nolimits_{p \in \mathcal{P}_k^q} c_w^k(p) \cdot f_k^q(p)$$

$$\text{Subject to}:$$

$$c_w^{k,q}(p) = \sum\nolimits_{(i,j)\in\mathcal{E}} \delta_{ij}^{k,q}(p) \cdot (c_{ij}^q + w_{ij}^q), p \in \mathcal{P}_k^q, \quad \forall k \in \mathcal{K}^q, \quad \forall q \in \mathcal{Q} \tag{7}$$

$$\sum\nolimits_{p \in \mathcal{P}_k^q} f_k^q(p) = B_k^q, \quad \forall k \in \mathcal{K}^q, \quad \forall q \in \mathcal{Q} \tag{9}$$

$$f_k^q(p) \geq 0, \quad \forall p \in \mathcal{P}_k^q, \quad \forall k \in \mathcal{K}^q, \quad \forall q \in \mathcal{Q}. \tag{10}$$

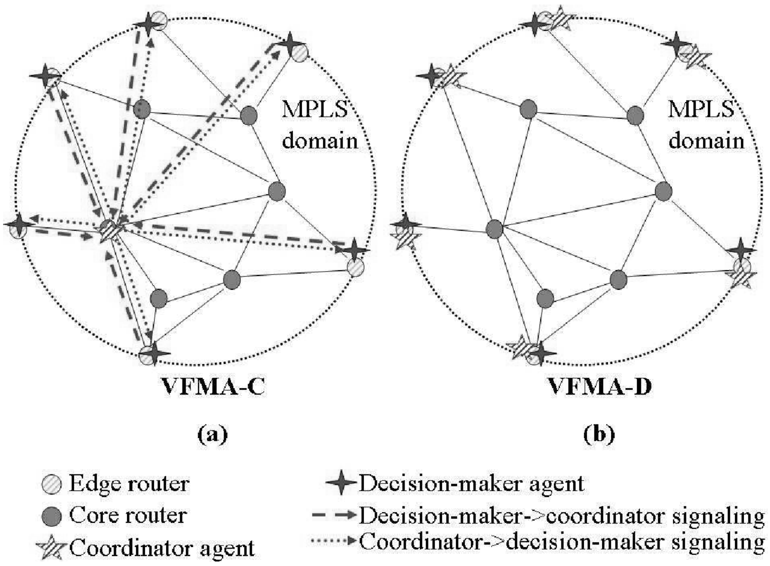
If the cost defined in equation (7) of the shortest path $\hat{p}$ found by solving **P**$_{\text{Dantzig–Wolfe}}$ is less than σ_k^q, which is the cost of the path associated with each commodity $k \in \mathcal{K}^q$, the associated decision maker agent will report this new path $\hat{p}$ to the coordinator agent, as an improving solution path to be added to the initial subset $\mathcal{P}_k^{q*}$. Otherwise, if the cost of the shortest path $\hat{p}$ is equal to σ_k^q, the decision maker will not need to report it to the coordinator since the coordinator is already using a path for commodity $k \in \mathcal{K}^q$ whose cost is σ_k^q. Note that the cost will never be greater than σ_k^q. The coordinator agent will continue solving iteratively the restricted master problem for each commodity $k \in \mathcal{K}^q$, including in the set of possible paths $\mathcal{P}_k^{q*}$ the new paths generated by the decision makers implementing the DW method, until the optimality condition is reached, i.e., until the cost of the new paths is not smaller than the cost of the path associated by the coordinator to each commodity $k \in \mathcal{K}^q$. In other words, the coordinator agent will continue solving the restricted master problem using the information from the decision makers until no improving solution paths need to be added in the solution. This assures that the optimal solution is reached, as it is rigorously proven in Ahuja et al. (1993).

5 Virtual-Flow Multipath Algorithms (VFMAs)

The proposed VFMAs are based on multiple agents, i.e., the coordinators and the decision makers, which iteratively exchange information in order to decompose the original MCNF problem, the so-called master problem, into a tractable subproblem, the so-called restricted master problem. This is done because the master problem is a NP-hard problem (Ahuja et al., 1993), i.e., it is computationally 'intractable' in most realistic network conditions due to its very high complexity. The coordinator and the decision-maker agents implement the Dantzing–Wolfe method, as detailed in Section 4, which provides the mathematical foundation of our framework, to decompose the MCNF master problem into a tractable restricted problem. Whilst the former problem optimally partitions the aggregated LSP requests with the same CoS q from ingress router i to egress router j into multiple independent flows by taking into account all possible paths $\mathcal{P}_{ij}^{q*}$ compliant with the CoS requirements and connecting i with j, the latter initially restricts the partitioning to a subset of paths, namely $\mathcal{P}_{ij}^{q*}$. These paths $p \in \mathcal{P}_{ij}^{q*}$ are initially selected in such a way that their costs cp do not exceed the cost c_{sh} of the shortest path from i to j by more than a fixed percentage r, i.e., $c_{sh} \le c_p \le c_{sh} \cdot (1 + r)$; $\forall p \in \mathcal{P}_{ij}^{q*}$. This is done to reduce the complexity of the algorithm. In particular, the lower r, the lower the complexity of the restricted problem, the higher the number of iterations required to converge to the optimal solution. The coordinator agent will continue solving iteratively the restricted master problem for each commodity $k \in \mathcal{K}^q$, including in the set of possible paths $\mathcal{P}_{ij}^{q*}$ the new paths generated by the decision makers implementing the DW method, until the optimality condition is reached, i.e., until the cost of the new paths is not smaller than the cost of the path associated by the coordinator with each commodity $k \in \mathcal{K}^q$. Moreover, since at each iteration the Dantzing–Wolfe decomposition method maintains a lower bound on the optimal value of the problem (Ahuja et al., 1993), the coordinator agent can terminate the algorithm at any iteration, not only with a feasible solution, but also with a guarantee of how far, in objective function value, that solution is from the optimality.

As it is depicted in Figure 3, the centralised algorithm (VFMA-C) relies on one coordinator agent, inside the MPLS domain, and on *several* decision-maker agents, one in each ingress router of the MPLS domain, while the distributed algorithm (VFMA-D) relies on several coordinator/decision maker pairs, one in each ingress router, i.e., one coordinator and one decision maker agents are *co-located* in each ingress router. A restricted master problem is iteratively solved by the coordinator and the decision-maker agent(s) for each CoS q and source-destination pair. Note that the centralised algorithm (VFMA-C) avails of more information to solve the MCNF multipath routing problem than its distributed counterpart (VFMA-D), since it gathers from each ingress router in the MPLS domain their LSP setup requests through signalling, as it is depicted in Figure 3(a). This allows VFMA-C to have a *global view* of all the LSP setup requests sent to the ingress routers and, thus, solve the multipath routing problem *jointly* for all the ingress router requests, whereas the distributed algorithm finds the multipath route for each ingress router considering only the incoming request to one particular ingress router. Hence, the extra information exploited by VFMA-C is expected to lead to better performance. On the other hand, although key feature of the centralised solution is to keep the information exchanged to a minimum, this information gathered from the ingress routers comes to the cost of extra signalling. In addition, this information may be delayed when mapped into standard communications through the User Network Interface (UNI), which further degrade the performance of VFMA-C. In the next section, we evaluate this trade-off, and show under which conditions our centralised solution outperforms the distributed solution (VFMS-D).

Figure 3 Coordinator and decision-maker agents in the MCNF: (a) centralised (VFMA-C): and (b) distributed (VFMA-D) and routing algorithms



6 Performance evaluation

In this section, we present the performance results of our virtual-flow multipath routing algorithms, and compare them with two single-path routing solutions (Kar et al., 2000; Awduche et al., 1999), namely the CSPF routing and the BSPR routing algorithms. While the former simply computes feasible source-destination shortest paths that minimise the minimum number of used links, the latter computes source destination paths taking into account the *residual* available bandwidth of each link, i.e., using the cost metric presented in Section 3.2, in order to distribute data flows among the most under-utilised links.

The optimisation problems presented in Sections 3.3 and 4 were implemented with AMPL (Fourer et al., 2002), and solved with CPLEX (http://www.cplex.com/), which uses a branch and bound algorithm to solve mixed linear problems. To ensure a fair evaluation of the performance of the competing routing algorithms, and to capture several network scenarios, random networks have been generated using a modified version of the Waxman's model (1988). According to this model, network nodes are randomly distributed across a Cartesian coordinate grid, and links are statistically added to the graph by considering all possible node pairs (i, j), using the following function, which accounts for the probability to have a link between nodes i and j,

$$P_e(i, j) = \beta \cdot \exp\left(-\frac{d_{ij}}{\alpha \cdot D} \right), \tag{11}$$

where d_{ij} is the Euclidean distance between the two nodes, D is the diameter of the network, i.e., the maximum possible distance between a pair of nodes in the network, and α and β are parameters in the interval (0, 1]. In particular, a high value of α increases the probability to have links between nodes further away, while a high value of β increases the average node degree. We assume that the cost of each link (i, j) is computed according to the model in Section 3.2, and that the bandwidth capacity $u_{ij}^{q,\text{tot}}$ is uniformly distributed in [50, 150] Mbps, with mean equal to $\overline{u^{q,\text{tot}}} = 100$ Mbps, for each CoS q. Note that, in general, the bandwidth capacity on link (i, j) may be different from the capacity on link (i, j), i.e., $u_{ij}^{q,\text{tot}} \neq u_{ij}^{q,\text{tot}}$. We call this model *modified Waxman's model.*

To better evaluate the performance of the proposed algorithms in different operating conditions, two different network scenarios are considered. In each scenario, many network topologies are generated according to the modified Waxman's model using the parameters reported in Table 1. Note that in Scenario II, the network has the same number of nodes as in Scenario I ($|\mathcal{N}| = 100$), but on average a higher number of links. Also, nodes further apart have a higher probability to be connected.

Table 1 Simulation parameters

Scenario	α	β	$\overrightarrow{u^{q,tot}}$ (Mbps)	$\lvert\mathcal{N}\rvert$	*LSP* (Mbps)
I	0.2	0.4	100	100	[0.1, 5]
II	0.3	0.6	100	100	[0.1, 30]

It is assumed that LSP requests arriving at each ingress router belong to the CoS q, which is an integer uniformly distributed in $[1, Q]$, where Q is the number of classes of service supported by the network. For each LSP, one ingress and one egress LSR are randomly chosen among the edge nodes of the MPLS network. In addition, the amount of bandwidth demanded by an IP flow and the length of its packets are randomly chosen. Moreover, each packet belonging to the same data flow is assumed to have the same length. Specifically, the length of packets is uniformly distributed in the interval [20, 2000] Bytes, where 20 Bytes is the minimum length of an IP header. As shown in Table 1, two traffic scenarios are considered: in Scenario I, the amount of LSP bandwidth requested by each IP flow is uniformly distributed in the interval [0.1, 5] Mbps; in Scenario II, the LSP bandwidth requested is uniformly distributed in [0.1, 30] Mbps. This second scenario is much more demanding for the network since the order of magnitude of the incoming LSP bandwidth requests and the link capacities are comparable.

The CSPF, BSPR, VFMA-C and VFMA-D algorithms are compared and evaluated using three network metrics: the *Rejection Rate* and the *Network Utilisation*, defined in Section 6.1, and the *Overhead Ratio*, defined in Section 6.2. For each simulation, several experiments, each with different traffic conditions and network topologies, have been run to ensure 95% relative confidence intervals smaller than 5%. Starting from a completely unloaded network, LSPs are setup according to the four competing routing algorithms, until a fixed rejection rate is achieved (*maximum rejection rate*).

6.1 Rejection rate and network utilisation

In this section, we define the rejection rate and the network utilisation, and analyse the results in terms of these metrics.

The *Rejection Rate* $\overline{\mathcal{R}}$ is defined as,

$$\overline{\mathcal{R}} = \frac{\sum_{q=1}^{Q} \sum_{h \in \mathcal{H}_{\text{rej}}^{q}} b_h^q}{\sum_{q=1}^{Q} \sum_{h \in \mathcal{H}^{q}} b_h^q}, \tag{12}$$

where the numerator in equation (12) is the sum over all CoS q of the IP flow bandwidth requests b_h^q that can not be accommodated in the network due to lack of resources and, thus, are rejected $(\mathcal{H}_{rej}^{q})$, and the denominator is the sum of bandwidth requests of all the incoming IP flows $f_{ij}^q \in \mathcal{H}^q, \forall i, j \in \mathcal{N}$.

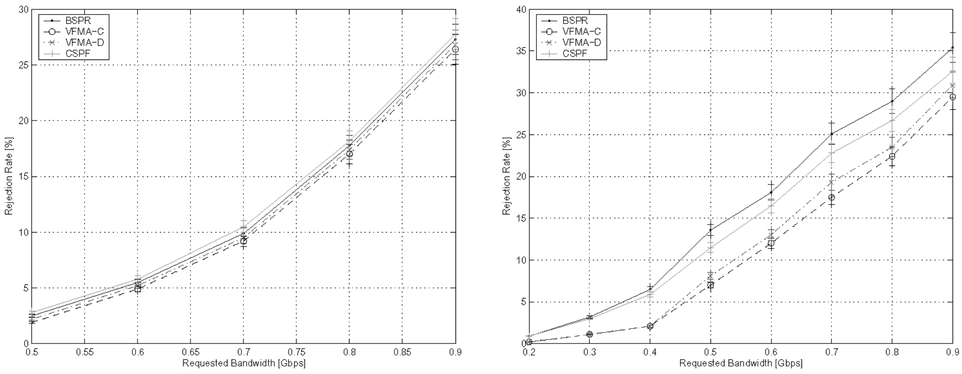
The *Network Utilisation* $\overline{\rho}_{\varepsilon}$ is defined as,

$$\overline{\rho}_{\varepsilon} = \frac{\sum_{(i,j) \in \mathcal{E}} \sum_{q=1}^{Q} \rho_{ij}^q}{|\mathcal{E}| \cdot Q}, \tag{13}$$

where $\mathcal{E}$ is the set of existing links, $|\mathcal{E}|$ its cardinality, i.e., the total number of links in the network, and $\rho_{ij}^q = (u_{ij}^{q,\text{tot}} - u_{ij}^q) / u_{ij}^{q,\text{tot}}$ is the utilisation of link (i, j) for CoS $q = 1, \ldots, Q$.

Figure 4(a) and (b) show the Rejection Rate $\overline{\mathcal{R}}$ experienced by all the competing algorithms in Scenarios I and II, respectively. The VFMA-C and VFMA-D rejection rate curves are always lower than those referring to the single-path routing algorithms. This is because the proposed VFMAs aggregate incoming IP flows with the same CoS and source-destination pair, and distribute these aggregated flows among multiple paths. The advantage of this multipath approach is more evident in Scenario II where it is more demanding accommodating the incoming LSP requests due to their higher average bandwidth requests (see Table 1). In fact, if there is no feasible path between two desired nodes that satisfies the request of bandwidth of an incoming data flow, VFMA-C and VFMA-D algorithms can still split the aggregate flow among those paths that satisfy a lower request of bandwidth, while single-path strategies, such as CSPF and BSPR, must reject the request, thus failing in accommodating the flow.

Figure 4 Rejection rate in: (a) Scenario I and (b) Scenario II

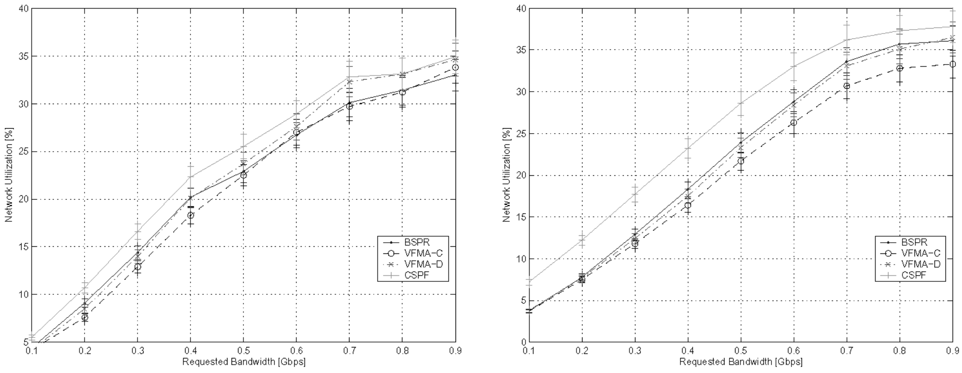


(a) (b)

Interestingly enough, in both scenarios the VFMA-D rejection rate is slightly higher than the VFMA-C rejection rate – but still lower than the CSPF and BSPR rejection rates – since VFMA-D lacks a global knowledge of the incoming IP flows at each ingress router, as stressed in the previous sections. In addition, Figure 4(a) shows that in Scenario I the BSPR rejection rate is slightly lower than the CSPF rejection rate. On the other hand, Figure 4(b) shows that in Scenario II the BSPR rejection rate is the highest among all the depicted rates. This is because BSPR generally selects LSPs with a number of links greater than CSPF, which chooses the paths that minimise the number of links, although it achieves a better load balancing. However, in the case of high bandwidth requests, as in Scenario II, the effects of higher drain of network resource, which characterises BSPR, overcomes the benefits of a better load balancing.

Figure 5(a) and (b) depict the Network Utilisation $\overline{\rho}_{\varepsilon}$ achieved by all the competing algorithms in Scenarios I and II, respectively. In both figures, the proposed virtual-flow algorithms achieve a lower network utilisation than their single-path counterparts. This is because they can obtain a better load balancing than single-path routing algorithms. This result also explains and corroborates their lower rejection rates previously shown in Figure 4(a) and (b).

Figure 5 Network utilisation in: (a) Scenario I and (b) Scenario II



(a) (b)

6.2 Overhead ratio

In this section, we define the overhead ratio, and compare the competing routing algorithms according to this metric.

The *Overhead Ratio* allows evaluating the average MPLS overhead introduced by the considered routing algorithms. The overhead has to be computed separately for the singlepath and virtual-flow multipath routing algorithms, although the definition is the same. This is because in the former algorithms only the IP packets belonging to the same flow can be encapsulated in one MPLS packet, while in the latter this constraints is removed by exploiting the virtual-flow concept, i.e., all the IP packets with the same CoS that are routed on the same MPLS path may be encapsulated in the same MPLS packet, no matter the flow they belong to. In particular, the overhead ratio $\bar{\mathcal{O}}$ associated with the single-path routing algorithms (CSPF and BSPR) is,

$$\bar{\mathcal{O}} = \frac{\sum_{n=1}^{M_{\text{flow}}} H_{MPLS} / (H_{MPLS} + N_n \cdot L_n)}{M_{\text{flow}}}, \tag{14}$$

while the overhead ratio $\bar{\mathcal{O}}_{VFMA}$ associated with the proposed multipath routing algorithms (VFMA-C and VFMA-D) is,

$$\bar{\mathcal{O}} = \frac{H_{MPLS}}{H_{MPLS} + \sum_{n=1}^{M_{\text{flow}}} N_n \cdot L_n}. \tag{15}$$

In equations (14) and (15), HMPLS represents the header of the MPLS packet (HMPLS = 4 Bytes in MPLS over SONET), M_{flow} accounts for the number of IP data flows arriving at the ingress LSRs, L_n is the length of the packets of the nth flow, and N_n is the average number of consecutive packets from the nth flow that are aggregated and encapsulated into one MPLS packet. In particular, if we assume that b_n is the average bit rate of the nth data flow, given the *enqueuing time* T_A, which is defined as the time that IP packets must be enqueued in the ingress LSR queue before they are encapsulated into a MPLS packet, then,

$$N_n = \left\lfloor \frac{T_A}{L_n / b_n} \right\rfloor. \tag{16}$$

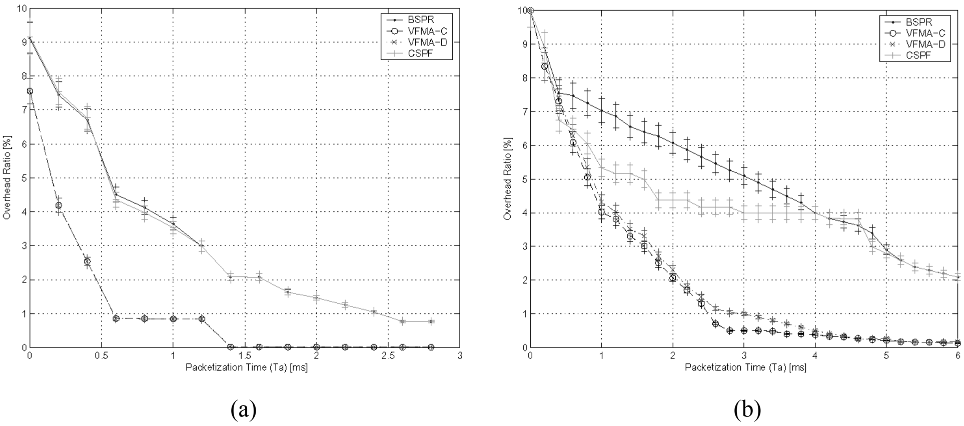
In equations (14) and (15), HMPLS represents the header of the MPLS packet (HMPLS = 4 Bytes in MPLS over SONET), M_{flow} accounts for the number of IP data flows arriving at the ingress LSRs, L_n is the length of the packets of the nth flow, and N_n is the average number of consecutive packets from the nth flow that are aggregated and encapsulated into one MPLS packet. In particular, if we assume that b_n is the average bit rate of the nth data flow, given the *enqueuing time* T_A, which is defined as the time that IP packets must be enqueued in the ingress LSR queue before they are encapsulated into a MPLS packet, then,

$$N_n = \left\lfloor \frac{T_A}{L_n / b_n} \right\rfloor. \tag{15}$$

By adjusting the value of T_A in equation (16), we can modify the MPLS overhead efficiencies of the competing routing algorithms in equations (14) and (15). Specifically, by increasing T_A we decrease the overhead of the MPLS header, since on average we are encapsulating a higher number of IP packets in one MPLS packet. On the other hand, we are delaying a higher number of IP packets in the ingress router, thus increasing their average queueing delays, before they can be encapsulated in the MPLS packet.

Figure 6(a) and (b) show the Overhead Ratios $\bar{O}$ and $\bar{O}_{VFMA}$ obtained with the four competing algorithms in Scenarios I and II, respectively. In both scenarios, the capability of aggregating data flows that characterises the virtual-flow approach of VFMA-D and VFMA-C algorithms allows decreasing the overhead ratio more quickly than with singlepath algorithms, with a small increase of the *enqueuing time* T_A. This is because the two virtual-flow based algorithms can encapsulate into a MPLS packet IP packets belonging to different incoming flows, i.e., the number of IP packets that can be encapsulated in one MPLS packet increases more rapidly than in the case of single-path algorithms. This solves the trade-off between the MPLS overhead minimisation, and the extra delay introduced by the encapsulation process of the IP packets.

Figure 6 Overhead ratio in: (a) Scenario I and (b) Scenario II



(a)

(b)

7 Conclusions and future work

This paper dealt with IP TE mechanisms for multipath selection in MPLS network domains. A centralised and a distributed virtual-flow routing algorithms were proposed, which aggregate IP flows entering the MPLS domain and optimally partition them among virtual flows that are forwarded on multiple paths. The proposed routing algorithms dynamically select multiple LSPs, taking into account the available bandwidth of links in the network in order to balance the traffic load and avoid network congestion caused by bottlenecks. The virtual-flow multipath routing problem was formulated as a MCNF problem, and was solved by implementing online the Dantzig–Wolfe decomposition method. The proposed centralised and distributed routing algorithms were shown to outperform single-path routing solutions such as the CSPF and the BSPR routing algorithms, and to achieve the performance targets by means of extensive simulation experiments. As future work, we plan to develop and evaluate novel effective policies to provide QoS differentiation to the IP flows passing through the MPLS domain.

Acknowledgements

This work was supported by the National Science Foundation (NSF) under Grant No. ANI-0219829. The authors wish to thank Dr. Ian F. Akyildiz for his support and continuing guidance.

References

Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993) *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, USA, February.

Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M. and McManus, J. (1999) 'Requirements for traffic engineering over MPLS', *IETF RFC 2702*, Tech. Rep., September.

Awduche, D.O. and Jabbari, B. (2002) 'Internet traffic engineering using MultiProtocol Label Switching (MPLS)', *IEEE Computer Networks*, Vol. 40, No. 1, September, pp.111–129.

Bertsekas, D. and Gallager, R. (1992) *Data Networks*, 2nd ed., Prentice-Hall, Inc., Upper Saddle River, New Jersey, USA.

Bonald, T., Oueslati, S. and Roberts, J. (2002) 'IP traffic and QoS control: towards a flow-aware architecture', *Proceedings of the World Telecommunications Congress*, Paris, France.

Davie, B.S. and Rekhter, Y. (2000) *MPLS Technology and Applications*, Morgan Kaufmann, Academic Press, San Francisco, CA, USA.

Dinan, E., Awduche, D. and Jabbari, B. (2000) 'Analytical framework for dynamic traffic partitioning in MPLS networks', *Proceedings of IEEE ICC'00*, New Orleans, Louisiana, USA, June, pp.1604–1608.

Elwalid, A., Jin, C., Low, S. and Widjaja, I. (2001) 'MATE: MPLS adaptive traffic engineering', *Proceedings of IEEE INFOCOM'01*, Anchorage, Alaska, USA, April.

Fortz, B. and Thorup, M. (2000) 'Internet traffic engineering by optimizing OSPF weights', *Proceedings of IEEE INFOCOM'00*, Tel Aviv, Israel, March.

Fourer, R., Gay, D.M. and Kernighan, B.W. (2002) *AMPL: A Modelling Language for Mathematical Programming*, Duxbury Press, Cole Publishing Co., Murray Hill, New Jersey, USA.

Girish, M.K., Zhou, B. and Hu, J.Q. (2000) 'Formulation of the traffic engineering problems in MPLS based IP networks', *Proceedings of ISCC'00*, Antibes, France, July, pp.214–219.

Juttner, A., Szviatovszki, B., Szentesi, A., Orincsay, D. and Harmatos, J. (2000) 'On-demand optimization of label switched paths on MPLS networks', *Proceedings of IEEE ICCCN'00*, Las Vegas, Nevada, USA, October, pp.107–113.

Kar, K., Kodialam, M. and Lakshman, T.V. (2000) 'Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering', *IEEE Journal of Selected Areas in Communications (JSAC)*, Vol. 18, No. 12, pp.2566–2579.

Kleinrock, L. (1975) *Queueing System: Theory*, Vol. I, John Wiley & Sons, New York.

Lee, Y., Seok, Y., Choi, Y. and Kim, C. (2002) 'A constrained multipath traffic engineering scheme for MPLS networks', *Proceedings of IEEE ICC'02*, New York City, New Jersey, USA, May, pp.2431–2436.

Pompili, D., Lopez, L. and Scoglio, C. (2004) 'DIMRO, a diffserv-integrated multicast algorithm for internet resource optimization in source specific multicast applications', *Proceedings of ICC 2004*, Paris, France, June.

Pompili, D., Scoglio, C. and Gungor, V.C. (2006) 'VFMAs, Virtual-Flow Multipath Algorithms for MPLS', *Proceedings of ICC 2006*, Istanbul, Turkey, June.

Saito, H., Miyao, Y. and Yoshida, M. (2000) 'Traffic engineering using multiple multipoint-to-point LSPs', *Proceedings of IEEE INFOCOM'00*, Tel Aviv, Israel, March.

Villamizar, A. (1999) *MPLS Optimized Multipath (MPLS-OMP)*, IETF DRAFT, draft-ietf-mpls-omp-00.txt, Tech. Rep., August.

Wang, Y. and Wang, Z. (1999) 'Explicit routing algorithms for Internet traffic engineering', *Proceedings of IEEE ICCCN'99*, Boston, Massachusetts, USA, October, pp.582–588.

Waxman, B.M. (1988) 'Routing of multipoint connections', *IEEE Journal on Selected Areas in Communications (JSAC)*, Vol. 6, No. 9, December, pp.1617–1622.

Xiao, X., Hannan, A., Bailey, B. and Ni, L. (2000) 'Traffic engineering with MPLS in the internet', *IEEE Network Magazine*, Vol. 14, No. 2, pp.28–33.

Notes

[1] Network fairness measures how equally multiple network connections with the same Class of Service share common network resources.

[2] A connection is supported by one single path or multiple paths, depending on the adopted routing strategy, the single-path or multipath, respectively.

[3] The rejection ratio is defined as the total bandwidth of the rejected requests over the bandwidth of all the requests.

Website

CPLEX, http://www.cplex.com/.