# VFMAs, Virtual-flow Multipath Algorithms for MPLS

**Dario Pompili***, **Caterina Scoglio**†, **Vehbi C. Gungor***

*School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332

†Department of Electrical and Computer Engineering
Kansas State University, Manhattan, KS 66506
e-mails: *{dario, gungor}@ece.gatech.edu, †caterina@ksu.edu

*Abstract*— **This paper deals with IP traffic engineering (TE) for multipath selection in MPLS networks. A centralized and a distributed routing algorithms are proposed, which aggregate IP flows entering the MPLS domain, and optimally partition them among virtual flows that are forwarded on multiple paths according to their quality of service (QoS) requirements. The virtual-flow multipath routing problem is formulated as a multicommodity network flow (MCNF) problem, and is solved by implementing on-line the Dantzig-Wolfe decomposition method, which is proven to converge to the optimal solution through an iterative procedure that divides the complex optimization problem into a tractable subproblem. The proposed multipath algorithms are shown to outperform single-path routing solutions by means of extensive simulation experiments.**

*Index Terms*— **MPLS, IP Traffic Engineering, QoS Multipath.**

## I. 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 *over-provisioning*. However, this approach is neither efficient nor practical. Hence, IP traffic engineering (TE) has become an essential requirement for ISPs in order to optimize the utilization 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 [1][2]. Although these routing algorithms are simple to be implemented and may provide an effective solution under some network conditions [3], in most cases shortest path routing algorithms cannot efficiently utilize the network resources, and offer limited control capabilities for traffic engineering. Moreover, they hardly provide *load balancing* in a network where the traffic to be accommodated has heterogeneous characteristics in terms of required bandwidth and QoS [4]. In addition, the lack of load balancing may also impair *fairness* among connections.

One effective approach to prevent network bottlenecks is to keep the average link utilization low by distributing data flows among the least-loaded links. In the literature, it is shown that the problem of minimizing the maximum link utilization in the network can be efficiently solved through the *multicommodity network flow* formulation, whose objective is to optimally split the traffic over multiple paths between source-destination pairs [5]. Since multipath routing provides network load balancing, network resources are more efficiently utilized 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. In fact, in multipath routing, the network can split the data traffic into smaller data flows, which can then be routed on different paths. However, if we consider one single connection, 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. 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, the total number of out-of-order packets should be limited.

To address the discussed challenges, in this paper we propose two virtual-flow multipath routing algorithms, a centralized (VFMA-C) and a distributed (VFMA-D) solution, in the context of Multi Protocol Label Switching (MPLS) [6][7]. MPLS is one of the most prominent technology that can improve the routing efficiency of IP networks through its intrinsic traffic engineering capabilities on heterogeneous network infrastructures. Both algorithms formulate the virtual-flow multipath routing problem as a *multicommodity network flow (MCNF)* problem [5], 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 guaranteing their QoS requirements and respecting the network constraints.

We introduce the *virtual-flow* concept, which allows the 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. Moreover, 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 quantization problems - the MCNF optimization problem is not cast as an integer linear problem (ILP), which is proven to be NP-complete [5]. Another advantage of the virtual-flow concept is that many IP packets that are assigned to the same virtual flow can now be encapsulated into few

1-4244-0355-3/06/$20.00 (c) 2006 IEEE

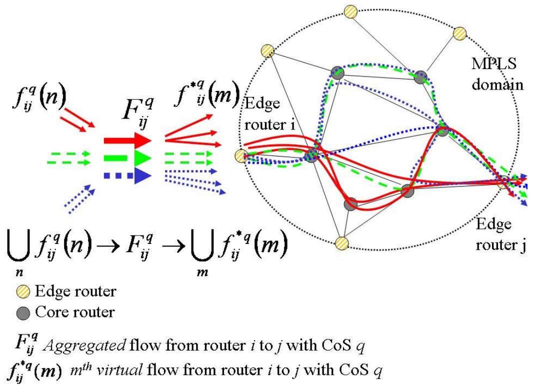

$f_{ij}^{q}(n)$
$F_{ij}^{q}$
$f^{*q}_{ij}(m)$
Edge router i
MPLS domain
Edge router j
$\bigcup_n f_{ij}^{q}(n) \rightarrow F_{ij}^{q} \rightarrow \bigcup_m f^{*q}_{ij}(m)$
Edge router
Core router
$F_{ij}^{q}$ *Aggregated* flow from router *i* to *j* with CoS *q*
$f^{*q}_{ij}(m)$ *m*th *virtual* flow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

Fig. 1. Enhanced MCNF-based virtual-flow multipath partitioning (VFMA).

large MPLS packets, which decreases the MPLS overhead in the data transmission.

Figure 1 shows the enhanced MCNF-based virtual-flow multipath partitioning (VFMA) in MPLS. While the standard multipath traffic partitioning in MPLS domains keeps the structure of the $N_{ij}^{q}$ incoming IP flows $f_{ij}^{q}(n), n = 1, ..., N_{ij}^{q}$, and routes them from an ingress router $i$ to an egress router $j$ on different paths according to their class of service (CoS) $q = 1, ..., Q$, our enhanced version 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 Fig. 1. This aggregate flow is then optimally split into $M_{ij}^{q}$ *virtual flows* $f_{ij}^{*q}(m), m = 1, ..., M_{ij}^{q}$, which are routed on $M_{ij}^{q}$ different paths towards the destination router $j$. To this end, the virtual-flow multipath algorithms 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, however, would not be a heavy-computational task, since, in general, few IP packets should be re-ordered, provided that they are in-order when they arrive at the ingress router.

In this paper, we compare the performance of our distributed and centralized routing solutions by means of extensive simulation experiments. We show that both the proposed multipath algorithms outperform single-path routing solutions [1][2], such as the constraint shortest path first (CSPF) and the bandwidth-based shortest path (BSPR) routing algorithms, since they can more flexibly exploit network resources.

The remainder of the paper is organized as follows. In Section II, we briefly review the literature. In Section III, we present the arc-path form of the MCNF problem. In Section IV, we present the proposed centralized and distributed multipath algorithms, whose performance is evaluated in Section V. Finally, in Section VI we draw the main conclusions.

## II. Related Work

MPLS is a connection-oriented label swapping technology that supports constraint-based routing (CBR) algorithms [8], thanks to its ability to implement explicit route functionality. A MPLS domain is constituted of label switching routers (LSRs), i.e., *edge routers* (*ingress* and *egress routers*) and *core routers*. In the literature, there are several research efforts addressing traffic engineering (TE) in MPLS networks. Traditional MPLS TE frameworks have been assumed to use single LSPs [1][2]. However, single LSPs in the network may result in network load unbalancing. In order to provide efficient network resource utilization, multiple LSPs between an ingress router and an egress router are proposed in several papers. Specifically, in [4] a multi-objective formulation of the traffic engineering problem for the minimization of the maximum link utilization and the minimization of the total network resource usage is proposed. However, in [4], only the centralized approach to the TE problem is presented, which, in practical applications, may not be feasible due to inaccurate information on network traffic. In [9], the authors develop a TE method utilizing multiple multipoint-to-point LSPs, in which multiple routes are used as backup routes in case of network failures. In [10], a network load balancing protocol called MATE (MPLS Adaptive Traffic Engineering) 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. However, MATE is not designed for bandwidth guaranteed services, and does not scale well when many ingress-egress pairs are considered. In [11], the authors propose MPLS-OMP, an Optimized MultiPath algorithm in which the distribution of network load among multiple paths is determined by utilizing a hash computation for each path. Finally, in [12], a stochastic framework for the 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 minimizing the overall traffic latency. However, the proposed model is difficult to implement and relies on many assumptions that may not hold in realistic network environments.

## III. Multicommodity Network Flow Problem

The multicommodity network flow (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. The objective of the MCNF problem is to minimize the total cost to "ship" all the commodities to their destinations, while satisfying the capacity constraints associated with the underlying network resources. In this paper, 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 Section III-A, we introduce the network and cost models that will be used in Section III-B to cast the MCNF arc-path formulation.

### A. Network and Cost Models

In this section, we introduce the network and cost models, as well as their notations and variables, used in the MCNF problem formulation:

- $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is a *directed graph* modeling 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 $q = 1, ..., Q$, e.g., $F_{ij}^q$, $i, j \in \mathcal{N}$, in Fig. 1. These commodities will be indexed with $k = 1, ..., K^q$, 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 connection $k \in \mathcal{K}^q$, respectively;
- $u_{ij}^{q,tot}$ 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 the following;
- $B_k^q$ is the bandwidth demanded by connection request $k \in \mathcal{K}^q$;
- $\mathcal{P}_k^q$ is the set of all feasible paths from source $s_k^q$ to destination $d_k^q$ 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 connection request $k \in \mathcal{K}^q$ assigned to path $p \in \mathcal{P}_k^q$.

A cost $c_{ij}^q$ is associated with link $(i, j)$ for each CoS $q = 1, ..., Q$ the network can support, as introduced in [13],

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

where $\epsilon$ is a small positive constant, $u_{ij}^q$ is the available bandwidth of link $(i, j)$, also *arc* $(i, j)$ in the following, associated with CoS $q$, and $\rho_{ij}^q$ is the *link utilization* associated with CoS $q$, which is defined as,

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

It is worth observing that minimizing the cost-metric in (1) automatically leads to the two-fold objective of minimizing i) the link utilization, and ii) the average queueing delay associated with link $(i, j)$, if a $\mathcal{M}/\mathcal{M}/1$ queue model [14] is assumed.

### B. Arc-path Form of the MCNF Problem

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* [5], 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 $\mathcal{G} = (\mathcal{N}, \mathcal{E})$; moreover, let $f_k^q(p)$ be the units of flow on path $p \in \mathcal{P}_k^q$ 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.

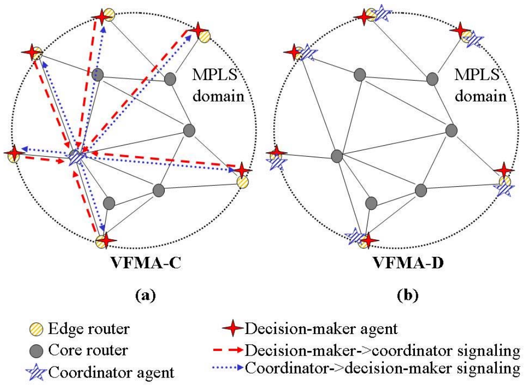


Fig. 2. *Coordinator* and *decision-maker agents* in the MCNF centralized (VFMA-C) (a) and distributed (VFMA-D) (b) routing algorithms.

**$\mathbf{P_{Arc-path}}$: Arc-path Form of the MCNF Problem**

$$\begin{aligned} Given: &\quad \mathcal{K}^q,\ c_{ij}^q,\ u_{ij}^{q,tot};\ \mathcal{P}_k^q,\ s_k^q,\ d_k^q,\ B_k^q,\ \forall k \in \mathcal{K}^q \\ Find: &\quad \delta_{ij}^{k,q}(p),\ f_k^q(p),\ \forall p \in \mathcal{P}_k^q,\ \forall (i,j) \in \mathcal{E} \\ Minimize: &\quad \textstyle\sum_{k \in \mathcal{K}^q} \sum_{p \in \mathcal{P}_k^q} c^{k,q}(p) \cdot f_k^q(p) \\ Subject\ to: & \end{aligned}$$

$$c^{k,q}(p) = \sum_{(i,j) \in \mathcal{E}} \delta_{ij}^{k,q}(p) \cdot c_{ij}^q,\ p \in \mathcal{P}_k^q,\ \forall k \in \mathcal{K}^q; \qquad (3)$$

$$\sum_{k \in \mathcal{K}^q} \sum_{p \in \mathcal{P}_k^q} \delta_{ij}^k(p) \cdot f_k^q(p) \leq u_{ij}^{q,tot},\ \forall (i,j) \in \mathcal{E}; \qquad (4)$$

$$\sum_{p \in \mathcal{P}_k^q} f_k^q(p) = B_k^q,\ \forall k \in \mathcal{K}^q; \qquad (5)$$

$$f_k^q(p) \geq 0,\ \forall p \in \mathcal{P}_k^q,\ \forall k \in \mathcal{K}^q. \qquad (6)$$

This formulation has a collection of $K^q = |\mathcal{K}^q|$ *bundle constraints* (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 $u_{ij}^{q,tot}$, and of $\sum_{k \in \mathcal{K}^q} |\mathcal{P}_k^q|$ network constraints (5), which state that for each commodity the total flow on all the paths connecting source $s_k^q$ and destination node $d_k^q$ equal the demand $B_k^q$. Finally, constraints (6) assure that each path flow be not negative. Overall, for each $q = 1, ..., Q$, $\mathbf{P_{Arc-path}}$ contains $|\mathcal{E}| \cdot |\mathcal{K}^q| + \sum_{k \in \mathcal{K}^q} |\mathcal{P}_k^q|$ constraints, in addition to the nonnegativity restrictions imposed on the path flow values (6).

## IV. VFMAs: Virtual-flow Multipath Algorithms

The proposed virtual-flow multipath algorithms 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 [5], 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 (DW) method, 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 class of service $q$ 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 $c_p$ do not exceed the cost $c_{sh}$ of the shortest path from $i$ to $j$ by more than a fixed percentage $r$, i.e., $c_{sh} \leq c_p \leq c_{sh} \cdot (1 + r)$, $\forall p \in {\mathcal{P}_{ij}^q}^*$. This is done to reduce the complexity of the algorithm. In particular, the lower $r$, the lower the complexity of the restricted problem, but 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. Moreover, since at each iteration the DW decomposition method maintains a lower bound on the optimal value of the problem [5], 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 Fig. 2, the centralized 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 centralized 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 signaling. 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. Thus, the extra information exploited by VFMA-C is expected to lead to better performance. On the other hand, although key feature of the centralized solution is to keep the information exchanged to a minimum, this information gathered from the ingress routers comes to the cost of extra signaling. In addition, this information may be delayed when mapped into standard communications through the UNI, which further degrade the performance of VFMA-C. In the next section, we evaluate this trade-off, and show under which conditions our centralized solution outperforms the distributed solution (VFMS-D).

## V. 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 [1][2], namely the constraint shortest path first (CSPF) routing and the bandwidth-based shortest path routing (BSPR) algorithms. While the former simply computes feasible source-destination shortest paths that minimize the minimum number of used links, the latter computes source-destination paths taking into account the *residual* available bandwidth of each link in order to distribute data flows among the most under-utilized links. The optimization problem presented in Section III-B was implemented with AMPL [15], and solved with CPLEX [16], 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* [13]. 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 $j$,

$$P_e(i,j) = \beta \cdot exp\left(-\frac{d_{ij}}{\alpha \cdot D}\right), \tag{7}$$

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 $\alpha$ and $\beta$ are parameters in the interval $(0, 1]$. We assume that the cost of each link $(i, j)$ is computed according to the model in Section III-A, and that the bandwidth capacity $u_{ij}^{q,tot}$ is uniformly distributed in $[50, 150]\ Mbps$, with mean equal to $\overline{u^{q,tot}} = 100\,Mbps$, for each CoS $q$. In general, the bandwidth capacity on link $(i, j)$ may be different from the capacity on link $(j, i)$, i.e., $u_{ij}^{q,tot} \neq u_{ij}^{q,tot}$. We call this model *modified Waxman's model.*

To better evaluate the performance of the proposed algorithms in different operating conditions, two network scenarios are considered. In each scenario, many network topologies are generated according to the modified Waxman's model. In particular, in Scenario I, $\alpha = 0.2$ and $\beta = 0.4$, while in Scenario II, $\alpha = 0.3$ and $\beta = 0.6$. 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. 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. 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 $[0.1, 30]\ Mbps$.

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 an unloaded network, LSPs are setup according to the four competing routing algorithms, until a fixed rejection rate is achieved, the so-called *maximum rejection rate*. CSPF, BSPR, VFMA-C, and VFMA-D algorithms are compared and evaluated using three network metrics: the *Rejection Rate*, the *Network Utilization*, and the *Overhead Ratio*.

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

$$\overline{\mathfrak{R}} = \frac{\sum_{q=1}^{Q} \sum_{h \in \mathcal{H}_{rej}^{q}} b_h^q}{\sum_{q=1}^{Q} \sum_{h \in \mathcal{H}^{q}} b_h^q}, \tag{8}$$

where the numerator in (8) is the sum over all CoS $q$ of the IP flow bandwidth requests $b_h^q$ that cannot 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 Utilization* $\overline{\rho}_{\mathcal{E}}$ is defined as,

$$\overline{\rho}_{\mathcal{E}} = \frac{\sum_{(i,j)\in\mathcal{E}} \sum_{q=1}^{Q} \rho_{ij}^q}{|\mathcal{E}| \cdot Q}, \tag{9}$$

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

The *Overhead Ratio* allows evaluating the average MPLS overhead introduced by the considered routing algorithms. The overhead has to be computed separately for the single-path and virtual-flow multipath routing algorithms, although the definition is the same. In particular, the overhead ratio $\overline{\mathfrak{O}}$ associated with the single-path routing algorithms (CSPF and BSPR) is,

$$\overline{\mathfrak{O}} = \frac{\sum_{n=1}^{M_{flow}} \frac{H_{MPLS}}{H_{MPLS}+N_n \cdot L_n}}{M_{flow}}, \tag{10}$$

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

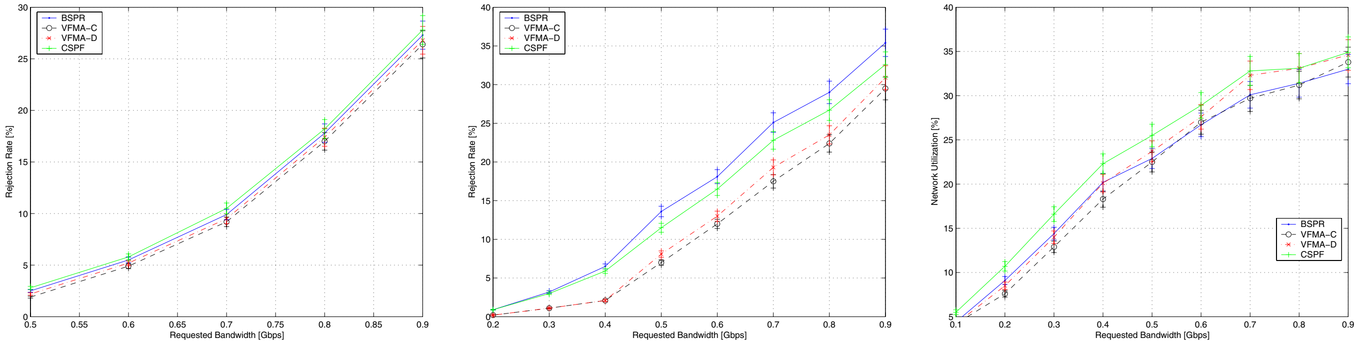
$$\overline{\mathfrak{O}}_{VFMA} = \frac{H_{MPLS}}{H_{MPLS} + \sum_{n=1}^{M_{flow}} N_n \cdot L_n}. \tag{11}$$

In (10) and (11), $H_{MPLS}$ is the header of the MPLS packet ($H_{MPLS} = 4\ Bytes$ in MPLS over SONET), $M_{flow}$ is the number of the IP data flows arriving at the ingress LSRs, $L_n$ is the length of the packets of the $n^{th}$ flow, and $N_n$ is the average number of consecutive packets from the $n^{th}$ 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 $n^{th}$ 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 $N_n = \left\lfloor \frac{T_A}{L_n/b_n} \right\rfloor$. By adjusting the value of $T_A$, we can modify the MPLS overhead efficiencies of the competing routing algorithms in (10) and (11). In particular, 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.

Figures 3(a) and 3(b) show the Rejection Rate $\overline{\mathfrak{R}}$ (8) experienced by all the competing algorithms in Scenarios I and II, respectively. As can been seen, 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 virtual-flow multipath algorithms 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. 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. Interestingly, 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, Fig. 3(a) shows that in Scenario I the BSPR rejection rate is slightly lower than the CSPF rejection rate. On the other hand, Fig. 3(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 paths that minimize 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 characterizes BSPR, overcomes the benefits of a better load balancing.

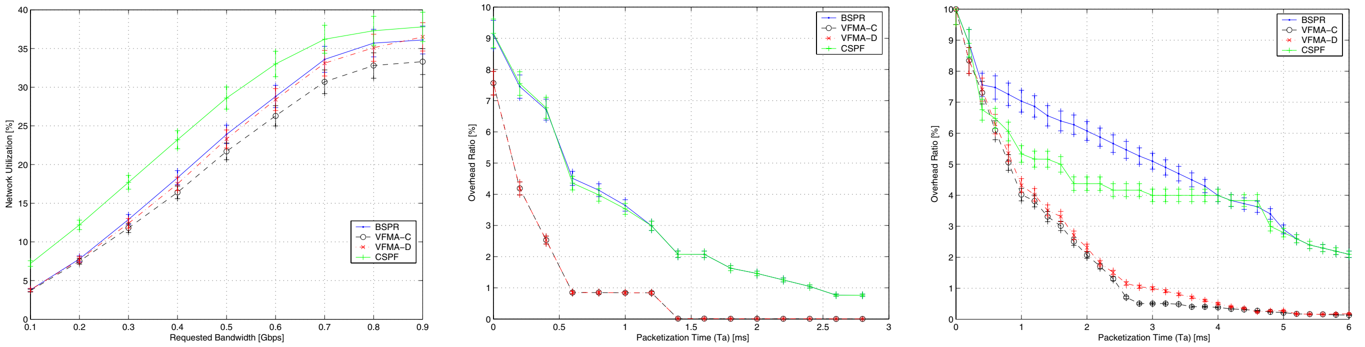
Figures 3(c) and 4(a) depict the Network Utilization $\overline{\rho}_{\mathcal{E}}$ (9) achieved by all the competing algorithms in Scenarios I and II, respectively. In both figures, the proposed virtual-flow algorithms achieve a lower network utilization than their single-path counterparts. This is 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 Figs. 3(a) and 3(b).

Figures 4(b) and 4(c) show the Overhead Ratios $\overline{\mathfrak{O}}$ (10) and $\overline{\mathfrak{O}}_{VFMA}$ (11) obtained with the four competing algorithms in Scenarios I and II, respectively. In both scenarios, the capability of aggregating data flows that characterizes the virtual-flow approach of VFMA-D and VFMA-C algorithms allows decreasing the overhead ratio more quickly than with single-path 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. This solves the trade-off between the MPLS overhead minimization and the extra delay introduced by the encapsulation process of the IP packets.



(a) (b) (c)

Fig. 3. (a): Rejection Rate in Scenario I, (b): Rejection Rate in Scenario II, (c): Network Utilization in Scenario I.



(a) (b) (c)

Fig. 4. (a): Network Utilization in Scenario II, (b): Overhead Ratio in Scenario I, (c): Overhead Ratio in Scenario II.

## VI. Conclusions

This paper dealt with IP traffic engineering mechanisms for multipath selection in MPLS network domains. A centralized 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 virtual-flow multipath routing problem was formulated as a multicommodity network flow (MCNF) problem, and was solved by implementing on-line the Dantzig-Wolfe decomposition method. The proposed centralized (VFMA-C) and distributed (VFMA-D) routing algorithms were shown to outperform single-path routing solutions by means of extensive simulation experiments.

## Acknowledgments

This work was supported by the National Science Foundation (NSF) under grant no. ANI-0219829. The authors wish to thank Dr. I.F. Akyildiz for his valuable support.

## References

[1] K. Kar, M. Kodialam, and T. V. Lakshman, "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, 2000.

[2] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS," IETF RFC 2702, Tech. Rep., September 1999.

[3] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proceedings of IEEE INFOCOM'00*, Tel Aviv, Israel, March 2000.

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

[5] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, February 1993.

[6] X. Xiao, A. Hannan, B. Bailey, and L. Ni, "Traffic engineering with MPLS in the Internet," *IEEE Network Magazine*, vol. 14, no. 2, pp. 28–33, 2000.

[7] D. O. Awduche and B. Jabbari, "Internet Traffic Engineering using MultiProtocol Label Switching (MPLS)," *IEEE Computer Networks*, vol. 40, no. 1, pp. 111–129, September 2002.

[8] B. S. Davie and Y. Rekhter, *MPLS Technology and Applications*. Morgan Kaufmann, Academic Press, 2000.

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

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

[11] C. Villamizar, "MPLS optimized multipath (MPLS-OMP)," IETF DRAFT, draft-ietf-mpls-omp-00.txt, Tech. Rep., August 1999.

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

[13] D. Pompili, L. Lopez, and C. Scoglio, "DIMRO, a DiffServ-Integrated Multicast algorithm for Internet Resource Optimization in source specific multicast applications," in *Proceedings of ICC 2004*, Paris, France, June 2004.

[14] L. Kleinrock, *Queueing system: vol. I, theory*. New York: John Wiley and Sons, 1975.

[15] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, Cole Publishing Co., 2002.

[16] CPLEX, http://www.cplex.com/.