Traffic Engineered Multicast Content Delivery Without MPLS Overlay

Ning Wang, *Member, IEEE*, and George Pavlou, *Member, IEEE*

***Abstract*—Multicast traffic engineering (TE) has recently attracted significant attention given the emergence of point-to-multipoint multimedia content delivery over the Internet. Existing multicast resource provisioning solutions tend to use explicit-routing based TE with multiprotocol label switching (MPLS) tunnels. In this paper, we shift away from this overlay approach and address native IP multicast traffic engineering based on link state routing protocols. The objective is that, through plain Protocol Independent Multicast-Sparse Mode (PIM-SM) shortest path routing with optimized multitopology IGP (MT-IGP) link weights, the resulting multicast trees are geared towards minimal consumption of bandwidth resources. We apply genetic algorithms (GA) to the calculation of optimized MT-IGP link weights that specifically cater for engineered PIM-SM routing with statistical bandwidth guarantees in multimedia content delivery. Our evaluation results show that GA-based multicast traffic engineering consumes significantly less bandwidth in comparison to conventional IP approaches while also exhibiting higher service availability.**

***Index Terms*—Link weight optimization, multicast routing, multimedia content delivery, Steiner tree, traffic engineering.**

I. Introduction

TRAFFIC engineering (TE) is an effective resource provisioning mechanism for improving the service capabilities of operational IP networks. In [2], TE is defined as a “large-scale network engineering for dealing with IP network performance evaluation and optimization”. Its key task is to enhance network performance while at the same time optimizing resource utilization. In recent years, TE has been extensively used by Internet service providers (ISPs) as a mechanism to provide good quality of service (QoS) for critical traffic such as real-time multimedia content delivery (e.g., MPEG video streaming). Traffic engineering approaches can be classified into multiprotocol label switching (MPLS) based and pure IP-based. With MPLS-based TE, packets are encapsulated with labels at ingress points, which are then used to forward these packets along a chosen explicit label-switched path (LSP). In addition, with its resource reservation mechanisms, MPLS is able to support stringent end-to-end bandwidth guarantees for multimedia content delivery. While MPLS is a powerful technology for creating overlay networks to support any specific routing strategy, it is also expensive and suffers potentially from scalability problem in terms of LSP state maintenance. On the other hand, the advent of pure IP-based TE solutions challenges MPLS-based approaches in that Internet traffic can also be effectively tuned through native hop-by-hop routing, without the associated complexity and cost of MPLS LSPs or “tunnels”. Some research works have indicated that OSPF/IS-IS link weights can be intelligently pre-assigned to achieve near-optimal path selections with respect to the expected traffic demand [3]–[5].

IP multicast [6] has always been regarded as an efficient paradigm for real-time multimedia group communication. Unfortunately, traffic engineering for multicast resource provisioning remains largely a dark area even today. In the past few years, MPLS-based multicast TE has become a subject of interest, with a number of relevant research works becoming available [7]–[11]. While MPLS is an attractive mechanism for delivering real-time multicast content, ISPs seem to be reluctant to deploy it at large scale given the cost and scalability considerations in terms of *point-to-multipoint* (p2mp) [10] LSP maintenance. Apart from the well-known issue in LSP state overhead, mature solutions are also missing for aggregating real-time multicast flows from different groups into a single point-to-multipoint LSP, as different groups tend to have different sets of egress routers [11]. On the other hand, native IP-based multicast resource provisioning without pre-configured MPLS overlay is considered as a promising alternative, but relevant solutions have not yet been explored. The reasons for this situation can be summarized as follows. First, the Protocol Independent Multicast–Sparse Mode (PIM-SM) [12] uses the underlying IP unicast routing table for the construction of multicast trees, and hence it is difficult to decouple multicast traffic engineering from its unicast counterpart. Bandwidth optimization for multicast traffic is generally formulated as the directed Steiner tree problem, which is NP-complete. The enforcement of Steiner trees can be achieved through packet encapsulation and explicit routing mechanisms such as MPLS tunneling. However, this approach lacks support from hop-by-hop protocols, due to reverse path forwarding (RPF) in the IP multicast routing protocol family. Given the inherent difference between the shortest path tree used by PIM-SM and the optimized Steiner tree, engineered multicast traffic for bandwidth optimization through Steiner tree heuristics could result in RPF check failures. In summary, it is some of the multicast control plane mechanisms that hamper the efficient dimensioning of network resources for multicast content delivery.

In this paper, we aim to break this barrier, and investigate the feasibility of engineering real-time multicast traffic for the pur-

Manuscript received February 15, 2006; revised August 8, 2006. An earlier and shorter version of this paper was published in the *IEEE/IFIP International Conference on Management of Multimedia Networks and Services* (MMNS) 2004. The associate editor coordinating the review of this paper and approving it for publication was Dr. Anna Hac.

The authors are with the Center for Communication Systems Research (CCSR), University of Surrey, Guildford BU2 7XH, U.K. (e-mail: n.wang@surrey.ac.uk; g.pavlou@surrey.ac.uk.)

Color versions of Figs. 1, 10, and 11 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2006.888016

pose of optimal resource provisioning based on plain IP routing protocols. Assuming we cannot touch the existing multicast control plane mechanisms, we confine our effort to the management plane and, more specifically, to *offline* multicast TE. *Our objective is to optimize the overall bandwidth consumption while maximizing the service availability for multicast content delivery with bandwidth requirements.* By satisfying the bandwidth demand of multimedia applications, both queuing delay/jitter and packet loss can be bounded, thus enabling potentially end-to-end QoS guarantees for real-time multimedia content delivery. In order to achieve these objectives, we perform offline network dimensioning so as to optimally constrain the behavior of control plane IP multicast routing. More specifically, the enforcement of engineered PIM-SM path selections is effected via setting optimized link weights for the underlying link state routing protocols. In our proposed approach, PIM-SM follows the shortest path according to the pre-set link weights, whereas the resulting multicast tree is in effect a hop-count Steiner tree with minimum number of links, which implies that minimum bandwidth resources are consumed. The advantage is that, through suitable link weight setting as calculated by off-line network dimensioning, conventional IP routers are able to construct optimized multicast trees by simply using Dijkstra's shortest path algorithm. As a result, MPLS support for explicitly building Steiner trees for bandwidth optimization is not necessary. Until recently, one difficulty in realizing such a scheme is that plain unicast routing protocols such as OSPF and IS-IS do not provide independent set of link weights for different types of flows. It is obviously undesirable to set link weights exclusively for multicast traffic and, at the same time, it is difficult to try to optimize unicast and multicast traffic through one set of link weights due to their potentially conflicting TE objectives. In order to decouple multicast from unicast path selection, our approach is based on recent multitopology enabled IGPs (MT-IGPs), e.g., Multitopology extensions to the IS-IS protocol (M-ISIS) [13] and OSPF protocol (MT-OSPF [14]), which are able to populate dedicated Multicast Routing Information Bases (M-RIBs, i.e., RPF tables) for PIM-SM routing. This multitopology routing feature provides a mechanism to separate multicast and unicast traffic engineering. For the rest of the paper we will use M-ISIS as a typical MT-IGP example for illustration. Another important consideration has to do with the fact that both multicast and unicast traffic use the same physical network. Given this situation, we assume the following offline TE scenario. First, unicast traffic engineering is performed based on the relevant TE objectives (e.g., load balancing using the scheme proposed in [3]). After that, the bandwidth resources allocated for the unicast traffic are deduced from the link capacities, and our proposed multicast traffic engineering solution is applied to the residual bandwidth. In this paper, we assume for simplicity that only multicast traffic exists in the network, which means that we work with the residual bandwidth after offline unicast TE has been performed.

The optimization of link weights through shortest path routing for indirectly obtaining one single Steiner tree in terms of hop-counts is NP-complete, since this is an adapted version of the classical Steiner tree problem. In effect, a more practical problem that concerns an ISP in terms of multicast traffic engineering is how to assign a set of unified link weights so that all the multicast trees within the network consume minimum bandwidth resources. In addition, given that bandwidth resources play a key role for overall QoS guarantees, we also introduce the additional constraint that the total bandwidth allocated on each link for the overlapping multicast trees should not exceed the link capacity. In this paper we adopt a genetic algorithm (GA)-based approach, which is a sophisticated global search tool for optimization problems, for addressing the problem of optimizing overall bandwidth consumption for multiple multicast flows. More specifically, the MT-IGP link weights are adjusted in each GA generation so that the overall fitness is geared towards optimized (i.e., minimized) network resource consumption with the constraint of link capacity. As already mentioned, the key novelty of this work is that optimized multicast resource provisioning with statistical bandwidth guarantees can be achieved through hop-by-hop IGP routing without relying on MPLS tunneling. Our results confirm that the proposed approach outperforms significantly conventional IP-based solutions and its capability of conserving network resources is comparable to Steiner tree schemes that need MPLS support.

II. Related Works

In [4], the authors proved that, any arbitrary set of *loop-free* routes can be represented with shortest paths with respect to a set of positive link weights, and [15] presented further analysis on the relevant issues in shortest path representability. The contribution from these works is of great significance, as they indicate the feasibility of transforming general routing optimization problems into shortest path routing. As a typical application, the authors of [3] claimed that by offline optimizing OSPF/IS-IS link weights for the purpose of load balancing, link congestion can be effectively avoided for unicast traffic. The key idea of the proposed algorithm is to intelligently adjust the weight of a certain number of links that depart from one particular node, so that new paths with equal weight are created from this node towards the destination. As a result, the traffic originally traveling through one single path can be split into other paths with equal OSPF/IS-IS weights. These emerging schemes of MPLS-free traffic engineering have enabled ISPs to dimension their networks without considering any scalability issue in LSP maintenance while performing optimized path selection for unicast traffic.

More recently, research efforts have also addressed traffic engineering for multicast content delivery, particularly for QoS and bandwidth optimization purposes. One common aspect of those schemes is that they are based on explicit routing, typically through MPLS tunneling for multimedia content delivery. The problem of bandwidth optimization in multicast routing is formulated as a Steiner tree problem, which has been extensively studied in the literature, with Takahashi and Matsuyama's (TM's) heuristic [16] being a near-optimal solution. Steiner tree routing for multiple multicast flows with bandwidth constraint was also investigated in [17] using the TM heuristic. As already mentioned, Steiner trees can be enforced through point-to-multipoint LSPs [10]. In [7], Steiner-tree-based heuristics are applied for computing multicast paths only at the edge of MPLS domains, so that multicast TE within the network can be reduced to a unicast routing problem. In [8], the authors propose an online multicast TE scheme using Steiner tree heuristics, which addresses also the issue of minimizing multicast flow interferences. In [9] and [10], the authors discuss general issues for multicast TE in MPLS environments.

III. M-ISIS Based Multicast Traffic Engineering

Before describing our proposed IP multicast traffic engineering scheme, we first describe the interactions between the parties involved in this TE-aware multicast content delivery. We assume the following business relationship: first, receivers subscribe to the multicast services offered by multimedia content providers (MCPs), and they may also express their QoS requirements during the subscription procedure. Receivers make an all-in-one payment to MCPs, including the cost of both multicast content and the associated QoS-based delivery. Each MCP needs to establish a multicast service level agreement (mSLA) and a multicast service level specification (mSLS) with the ISP in order to have its QoS aware multicast service deployed over the ISP's physical network. First, the MCP will allocate part of the revenue from its own customers for the ISP payment regarding the consumption of network resources. Technically, individual MCPs should pass necessary traffic information to the ISP, in order for the latter to derive multicast traffic matrix as an input to the offline TE procedure. Specifically, the following information should be included when negotiating a basic mSLA/mSLS: 1) traffic characteristics (e.g., bandwidth demand, QoS requirements, availability etc) and 2) root node (i.e., ingress router) and a set of egress routers with attached subscribers. After the ISP has provisioned optimally the resources through offline multicast TE, receivers may start to join their subscribed groups.

In the control/data plane, the traditional OSPF and IS-IS protocols only have a uni-dimensional viewpoint on the weight of each link, and this influences path selections for both unicast and multicast traffic. In contrast, M-ISIS and MT-OSPF provide the original IS-IS/OSPF protocols with additional ability of viewing the weight of each link for different logical IP topologies independently. For example, in M-ISIS the field of multi topology identifier (MT-ID) with value 3 is dedicated to the multicast reverse path forwarding topology. With this multitopology capability, it becomes possible that PIM-SM based multicast routing is completely decoupled from the underlying routing table for unicast traffic.

Fig. 1 shows how M-ISIS link weight optimization in the management plane can enforce the behavior of the underlying multicast routing protocol for path selections. Once the multicast traffic matrix and the network topology have been obtained, the optimized link weights are computed through off-line algorithms and configured in the routers that run the M-ISIS routing protocol with MT-ID equal to 3, which, as explained, is dedicated to the multicast RPF table construction. On receiving link state advertisements (LSAs), each M-ISIS aware router computes shortest path trees according to this set of link weights and decides the NEXT_HOP router for a specific IP address/prefix. When a PIM-SM join request is received, the router simply looks up the RPF table and finds the proper NEXT_HOP for forwarding the packet. In this scenario, the delivery of PIM-SM group join requests follows an engineered path, thus the resulting multicast distribution tree from the root to individual members conforms to the TE requirement. In addition, the multicast forwarding information base (M-FIB) is dynamically updated for the incoming interface (*iif*) and outgoing interface (*oif*) list of each group.

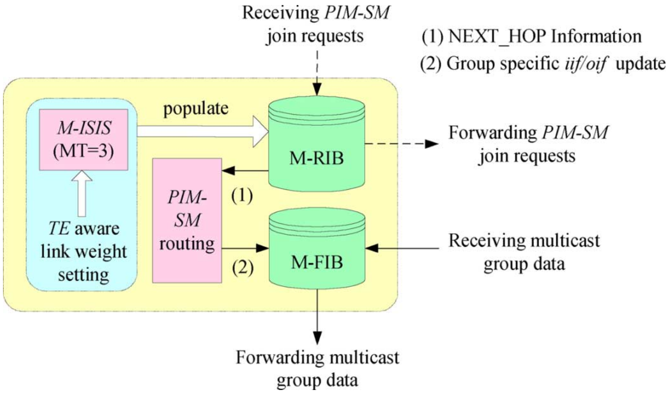

Receiving *PIM-SM* join requests
(1) NEXT_HOP Information
(2) Group specific *iif/oif* update
populate
M-ISIS (MT=3)
TE aware link weight setting
PIM-SM routing
(1)
(2)
M-RIB
M-FIB
Forwarding *PIM-SM* join requests
Receiving multicast group data
Forwarding multicast group data

Fig. 1. M-ISIS based multicast traffic engineering.

IV. Problem Formulation

A network topology is represented as a graph $GR = \langle V, E \rangle$, where V and E denote the node and link set of the topology graph respectively. The following is the integer-programming formulation for computing explicit bandwidth constrained Steiner trees with the objective of minimizing overall bandwidth consumption. By setting the group-specific *binary* variables $x_{ij}^{g,k}$ and y_{ij}^{g} for each link $(i,j) \in E$, a set of explicit multicast trees with minimum number of links is obtained, which implies that minimum bandwidth consumption is achieved. We first present some definitions.

G Total number of active multicast groups.
r_g Root node of group $g (g = 1 \ldots G)$.
V_g Multicast member (receiver) set for group g.
T_g Multicast tree spanning active group members in V_g.
D_g Bandwidth demand for group g traffic on each link.
C_{ij} Bandwidth capacity of link (i,j).
y_{ij}^{g} Equal to 1 if link (i,j) is included in the multicast tree for group g, equal to 0 otherwise.
$x_{ij}^{g,k}$ Equal to 1 if link (i,j) is on the multicast tree branch from the root node r_g of group g to the group member node k in the multicast tree, equal to 0 otherwise.

The integer-programming problem of computing a set of bandwidth constrained Steiner trees with minimum overall bandwidth consumption is formulated as

$$\text{Minimize} \qquad \sum_{g=1}^{G} \sum_{(i,j)\in E} D_g \times y_{ij}^{g}$$

subject to

$$\sum_{h\in V} x_{ih}^{g,k} - \sum_{j\in V} x_{ji}^{g,k} = \begin{cases} 1, & i = r_g \\ -1, & i = k, k \in V_g \\ 0, & i \neq r_g, i \notin V_g \end{cases} \quad (1)$$

$$x_{ij}^{g,k} \le y_{ij}^{g}, \quad (i,j) \in E, k \in V_g \quad (2)$$

$$x_{ij}^{g,k} = 0,1, \quad (i,j) \in E, k \in V_g \quad (3)$$

$$y_{ij}^{g} = 0,1, \quad (i,j) \in E \quad (4)$$

$$\sum_{g=1}^{G} y_{ij}^{g} \times D_g \le C_{ij}, \quad (i,j) \in E. \quad (5)$$

The variables to be determined are $x_{ij}^{g,k}$ and y_{ij}^{g} for every link $(i,j) \in E$. Constraint (1) ensures the same unit of multicast flow from r_g to every group member node $k \in V_g$. Constraint (2) guarantees that the amount of flows along link (i,j) must be zero if this link is not included in the multicast tree for group g. $x_{ij}^{g,k}$ and y_{ij}^{g} are confined to zero-one variables in constraints (3) and (4) for nonsplitting of multicast flows. It is required in (5) that the total bandwidth consumption on each link should not exceed its capacity.

As we have mentioned before, the enforcement of the above set of bandwidth constrained Steiner trees can be achieved through an explicit routing overlay, e.g., through MPLS, on a per-group basis. However, the paths in the Steiner tree from r_g to individual group members $k \in V_g$ might not completely overlap with the shortest paths between them. This means that, in case of hop-by-hop routing, multicast traffic flowing on the Steiner tree will be discarded due to the network RPF check failure, if the packets are not received from the correct interface on the shortest path back to the source. In order to apply the above programming model to IP-layer solutions, we introduce a unified M-ISIS link weight w_{ij} for each link (i,j), and by properly setting those link weights it is guaranteed that the tree branch from r_g to any receiver $k \in V_g$ is the shortest path according to this set of weights. In other words, our strategy is to represent this set of explicit Steiner trees with shortest path trees through intelligent configuration of a unified set of link weights. Formally, the problem is to calculate a set of positive link weights $W = \{w_{ij}\} : w_{ij} > 0$, such that for each optimized multicast tree $T_g(g = 1 \ldots G)$ with bandwidth constraints, the following inequality holds.

For any on-tree path $P_{r_g\to k} \subseteq T_g$ (i.e., for each link $(i,j) \in P_{r_g\to k}, x_{ij}^{g,k} = 1$), $\forall P'_{r_g\to k} \not\subset T_g$

$$W(P_{r_g\to k}) \le W(P'_{r_g\to k})$$

where $k \in V_g$.

According to [15], finding one set of unified link weights for converting an arbitrary group of explicit routes exactly into shortest paths is an NP-complete problem. This reveals one of the reasons behind why MPLS explicit routing is able to outperform pure IP based approaches, which lack flexibility in path selection. In this paper we address this issue within the scope of multicast traffic engineering where one set of M-ISIS link weights is optimized for controlling multiple multicast flows. Although it is not always possible to apply this type of shortest-path representability to an arbitrary set of explicit trees with one set of link weights, there always exists an approximation that can be geared towards the TE requirement.

V. A Genetic-Algorithm-Based Solution

Compared to MPLS based traffic engineering solutions, the task of link weight optimization is more complicated in that the search space for optimal path selection is much larger due to the wide range of possible weights for individual links. In effect, a set of optimal multicast trees can be enforced potentially through different sets of link weights. To deal with this high complexity, meta-heuristics (other than dedicated heuristics) are often adopted, e.g., Tabu search for unicast traffic engineering in [3]. In this paper we use another popular meta-heuristic, GA, which is considered as a very good global search tool for complex optimization problems. The basic idea behind genetic algorithm based approaches is as follows. First, a series of random solutions are obtained as the initial generation of *chromosomes* in the population. Thereafter, improved offsprings evolve iteratively from the parents by calculating their fitness. Chromosomes with higher fitness have higher probabilities of being inherited by the next generation. In each iteration, a new generation of chromosomes is created through the process of parent selection and reproduction. This is specifically achieved through genetic operators such as crossover and mutation. Finally, after a predefined number of generations, or if the performance of fitness has reached convergence, the resulting chromosome with the best fitness is selected as the final solution.

A. Encoding and Initial Population

In our GA approach, each chromosome is represented by a link weight vector $W = \langle w_1, w_2, \ldots w_{|E|}\rangle$ where $|E|$ is the total number of links in the network. The value of each weight is within the range from 1 to $\mathrm{MAX\_WEIGHT}$. In our experiments we define the value of $\mathrm{MAX\_WEIGHT}$ to be 64 for reducing the search space. On the other hand, the population size is set to 100, with the initial values inside each chromosome randomly varying from 1 to $\mathrm{MAX\_WEIGHT}$.

B. Fitness Evaluation

Chromosomes are selected according to their fitness. In our approach, the bandwidth constraint is embedded in the fitness function as a penalty factor, so that the search space is explored with the potential feasible solutions. The fitness of each chromosome can be defined as a two-dimensional function of the overall network load ($l1$) and excessive bandwidth allocated to overloaded links ($l2$), i.e.,

$$\text{fitness} = f(l1, l2) = \frac{\mu}{\alpha \times l1 + \beta \times l2} \quad (6)$$

where α, β, μ are manually configured coefficients.

In (6), $l1$ and $l2$ are expressed as follows:

$$l1 = \sum_{g=1}^{G} \sum_{(i,j)\in E} D_g \times y_{ij}^{g} \quad (7)$$

$$l2 = \sum_{(i,j)\in E} \omega_{ij} \times \left(\sum_{g=1}^{G} D_g \times y_{ij}^{g} - C_{ij} \right) \quad (8)$$

```
Procedure Computing_Fitness(Chromosome i)
Begin
Set the weight of each link in the network according to the gene values in chromosome i;
For each multicast group g
    Compute the shortest path tree $T_g$ rooted at $r_g$, and spanning to all the members in $V_g$;
    For each link $(i, j)$ in $T_g$
        Update link load $L_{ij}$ according to the bandwidth demand $D_g$ of group g;
End For
Load1 = 0;   Load2 =0;
For each link $(i, j)$ in the network
    Load1= Load1 + $L_{ij}$;
    If $L_{ij} > C_{ij}$ then
        Load2= Load2 + ($L_{ij}$-$C_{ij}$);
End For
Return fitness= $f(Load1, Load2)$;
End
```

Fig. 2. Fitness calculation.

where

$$\omega_{ij} = \begin{cases} 0, & \text{if } \sum_{g=1}^{G} D_g \times y_{ij}^g \le C_{ij} \\ 1, & \text{otherwise} \end{cases}. \tag{9}$$

The objective of the fitness function (6) is 1) chromosomes of the new generations should converge towards a set of Steiner trees with minimum bandwidth consumption and 2) solutions obtained from the offspring should be feasible in that the total bandwidth allocated to the multicast flows traveling through each link should not exceed its capacity. The tuning of α, β can be regarded as a tradeoff between overall bandwidth conservation and load balancing. Fig. 2 presents the logic for fitness evaluation.

C. Crossover and Mutation

According to the basic principle of GAs, chromosomes with better fitness values havea higher probability of being inherited in the next generation. To achieve this, we first rank all the chromosomes in descending order according to their fitness, i.e., the chromosomes with high fitness are placed on the top of the ranking list. Thereafter, we partition this list into two disjoined sets, with the top 50 chromosomes belonging to the upper class (*UC*) and the bottom 50 chromosomes to the lower class (*LC*). During the crossover procedure, we select one parent chromosome C_U^i from *UC* and the other parent C_L^i from *LC* in generation i for creating the child C^{i+1} in generation $i+1$. Specifically, we use a crossover probability threshold $K_c \in [0, 0.5)$ to decide the genes of which parent to be inherited into the child chromosome in the next generation. We also introduce a mutation probability threshold K_M to randomly replace some old genes with new ones. In addition to this type of conventional mutation, we also find the *congested* link with the highest load in the chromosome of the new generation, and we randomly raise its link weight in order to avoid hot spots; this is, of course, not necessary in uncongested conditions. Fig. 3 presents the crossover and mutation logics.

```
Procedure Crossover($C_U^i$, $C_L^i$)
Begin
For all genes $j = 1,...,|E|$
    Generate $r$ = random [0,1];
    If $r > K_C$ then
        $C^{i+1}(j) = C_U^i(j)$;
    Else if $r > K_M$
        $C^{i+1}(j) = C_L^i(j)$;
    Else
        $C^{i+1}(j)$ = random[1, MAX_WEIGHT]
End For
Find gene (link) t with the highest load in $C^{i+1}$;
If $L_t$ (link load) > $Cap_t$ (link capacity) then
        $C^{i+1}(t)$ = random [$C^{i+1}(t)$, MAX_WEIGHT ]
Return $C^{i+1}$;
End
```

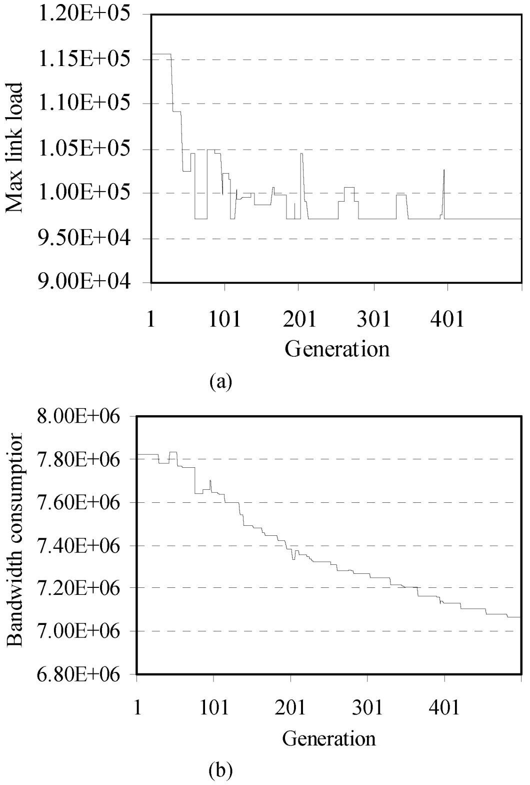
Fig. 3. Crossover and mutation.

VI. The GA Process Analysis

In the following examples, we assume that the scaled bandwidth capacity of each link is 10^5. Detailed information on the experimental configuration used is provided in Section VII.A. To illustrate how GA optimization improves the performance step by step in each generation, we study the following two scenarios. From both of them, we can clearly observe the tradeoff between our objectives of conserving network resources and guaranteeing feasible solutions.

In the first scenario, we set the maximum group traffic demand $\text{Max } D_g$ to be 4000, so that none of the initial solutions in the first generation can satisfy the constraint of bandwidth capacity. From Fig. 4(a), we can see that the maximum link load computed by the best chromosome in the initial generation is 1.16 $\times 10^5$, which means that at least one link is overloaded by 16%. Starting from this set of infeasible solutions, the GA approach first manages to eliminate the overloaded links by decreasing the load of those congested links. We can see that the maximum link load decreases drastically within the first 50 generations, and feasible individuals emerge from then on. Thereafter, the overall bandwidth consumption starts to drop significantly with the maximum link load varying just below the bandwidth capacity for most of the period. Finally in the 500th generation the overall bandwidth consumption converges to the lowest value (7.08 $\times 10^6$), while the link with the highest load becomes near saturated (9.7 $\times 10^4$) but not overloaded (i.e., the link load is still below the capacity). We regard this scenario as a case of *marginal resource over-provisioning* for the full projection of the multicast traffic matrix.

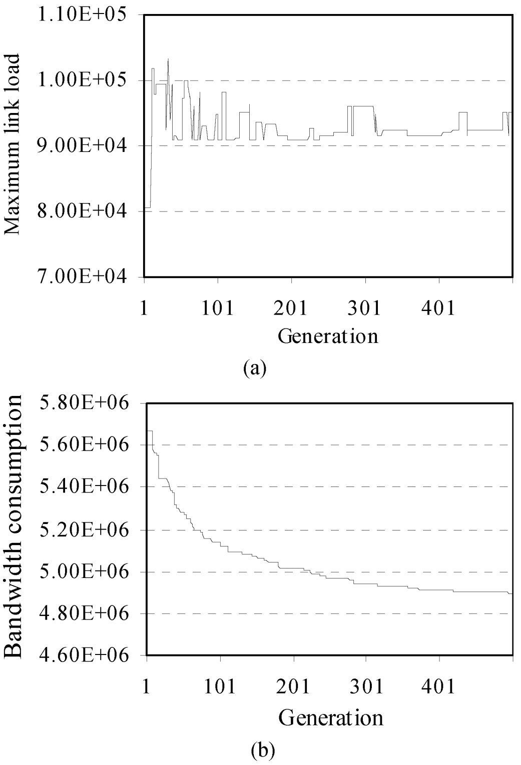
In the second scenario, we set $\text{Max } D_g = 3000$, so that feasible solutions already exist in the initial population. In Fig. 5(a), we can see that the load of the highest link is about 80% of the capacity in the first generation. When the GA optimization starts, the overall bandwidth consumption decreases significantly, as shown in Fig. 5(b). During this period, the traffic



(a)

(b)

Fig. 4. GA optimization process (Scenario I).



(a)

(b)

Fig. 5. GA optimization process (Scenario II).

distribution becomes less balanced, as the highest link load increases sharply within the first ten generations. Although this value exceeds the bandwidth capacity occasionally, for most of the period the highest utilization varies between 90% and 100%, thus feasible solutions are guaranteed in each generation.

From the description in Section V, it is obvious that the computing of chromosome fitness takes up most processing time in each generation, in comparison to crossover and mutation. From Fig. 2 it is easy to figure out that the time complexity of fitness calculation is $O(G|N|^2 + |E|)$, where G is the total number of active groups, and $|N|$ and $|E|$ are the number of nodes and links in the network. Given the fact that $|N|^2 \gg |E|$, the overall time complexity of the GA algorithm is $O(MPG|N|^2)$, where M is the predefined maximum generation and P is population size. We ran the algorithm on a PC with a Pentium IV 1.4-G processor, and it took about 8 min to compute converged solutions for a network with 100 nodes and 100 groups (M and P are set to 500 and 100, respectively). Given that this is offline optimization, the time taken to calculate the weights is not critical and the optimization problem is computationally tractable without requiring significant computing resources.

VII. Simulation Results

A. Simulation Configuration

In this section, we evaluate the proposed approach through simulation. We adopt the GT-ITM topology generator [18] (Waxman's flat model), which is widely used for the simulation of large size networks (e.g., [3], [17]), for constructing our network models. This approach distributes the nodes randomly on the rectangular grid and nodes are connected with the probability function

$$P(u,v) = \lambda \exp\left(\frac{-d(u,v)}{\rho L}\right)$$

where $d(u,v)$ is the distance between node u and v and L is the maximum possible distance between any pair of nodes in the network. The parameters λ and ρ ranging (0, 1) can be modified to create the desired network model. A larger value of λ gives a node with a high average degree, and a small value of ρ increases the density of shorter links in comparison to longer ones. In our simulation, we set the values of λ and ρ to be 0.2 respectively, and generate a random network of 100 nodes, out of which 50 are configured as designated routers (DRs) with attached group sources or receivers. The total number of multicast groups is set to 100 in our experiment. As far as bandwidth demand is concerned, the D_g range is based on the fact that multicast sources may send the same content in different quality levels, each associated with specific bandwidth demand, to individual groups subscribed by receivers with similar capacities. A destination set grouping (DSG [19]) is one typical example. In order to cover a wide range of over-provisioning scale for bandwidth resources, we change the maximum bandwidth demand D_g in our simulation so as to reflect different ratios between the overall bandwidth demand and the link capacity. For the purpose of simulation accuracy, we run simulations with the same configuration ten times for each data point and use the mean value to plot Figs. 7–11, and 20 times for calculating the success rate in Fig. 6.

The simulation parameters of the proposed Genetic Algorithm are illustrated in Table I. As we mentioned previously,

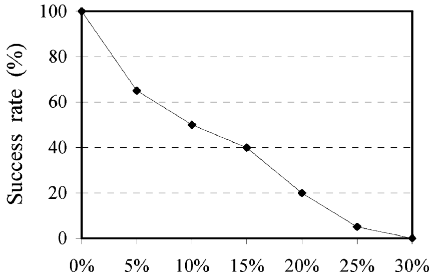


Fig. 6. GA success rate versus $\mathrm{MLOR}_{\mathrm{SPH}}$.

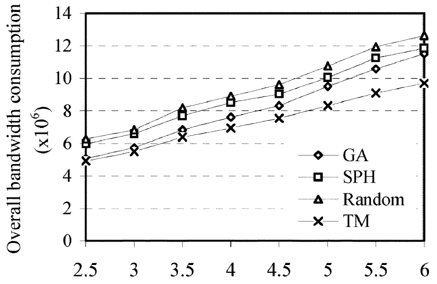


Fig. 7. Total bandwidth consumption versus $\mathrm{Max}\ D_g(\times 10^3)$.

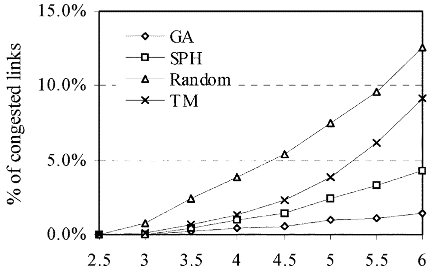


Fig. 8. Link congestion ratio versus $\mathrm{Max}\ D_g(\times 10^3)$.

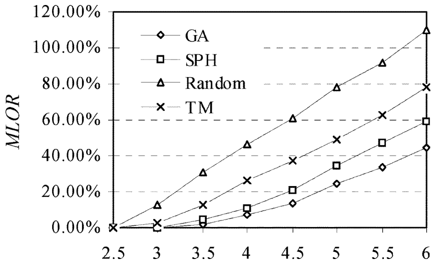


Fig. 9. MLOR versus $\mathrm{Max}\ D_g(\times 10^3)$.

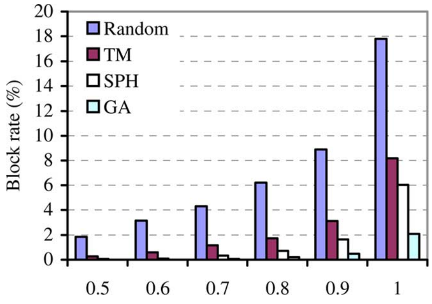


Fig. 10. Join blocking rate versus invocation ratio ω.

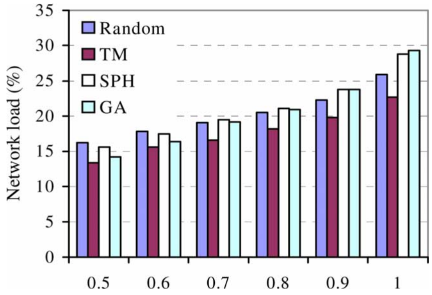


Fig. 11. Network load versus invocation ratio ω.

TABLE I
GA PARAMETER CONFIGURATION

Parameter name	Value
Population size (*P*)	100
Maximum number of generations (*M*)	500
Maximum link weight (*MAX_WEIGHT*)	64
Crossover probability threshold (K_C)	0.3
Mutation threshold (K_M)	0.01
μ	10^7
α	1.0
β	10

this work is the first attempt towards pure IP-based multicast traffic engineering through multitopology IGP link weight optimization. Hence, apart from our proposed GA approach, we also implemented two non-TE-based hop-by-hop IP routing approaches, and also one explicit routing approach with TE awareness that needs MPLS support, specifically 1) shortest path routing with random link weight setting (Random); 2) shortest path routing in terms of hop-counts (SPH); and 3) Steiner tree approach using the TM heuristic. For this TM-based Steiner tree algorithm, we aimed to construct one explicit multicast tree per group, each containing minimum number of tree links (i.e., total number of hops). More specifically, if link (i,j) is included in the multicast tree for group g, then the binary variable y_{ij}^g is set to 1, otherwise y_{ij}^g equals 0. Explicit multicast trees are computed using the TM algorithm with the objective of minimizing $\sum_{g=1}^{G}\sum_{(i,j)\in E} y_{ij}^g$. This is equivalent to the computation of multicast trees with minimum bandwidth consumption, as a multicast tree consisting of minimum number of links (hop counts) indicates also that minimum bandwidth resources are consumed. As TM has been regarded as near-optimal in minimizing the overall cost, it is used as the reference to indicate the bandwidth conservation capability of the proposed IP based approach (see Fig. 7). On the other hand, we do not aim to compute optimal Steiner trees for large-size networks due to the fact that the problem is NP-complete. In the next section we will show that the TM heuristic has the best performance among the four algorithms in terms of bandwidth conservation. Nevertheless, it should be emphasized that this solution requires the setting up of MPLS tunnels for explicit routing on a per-group basis, and this cannot be achieved in a pure IP environment. Hence, the inclusion of the TM algorithm is to use its performance as a lower bound for comparison with the other three hop-by-hop oriented approaches in terms of bandwidth conservation.

B. Performance Evaluation

We classify our simulation experiments into two categories: 1) static provisioning performance with respect to the full projection of the multicast traffic matrix and 2) real-time performance based on group membership dynamics, including both group joins and leaves.

Figs. 6–9 illustrate the provisioning performance where the entire multicast traffic matrix is mapped onto the physical network. We found that SPH has higher capability in finding feasible solutions than random link weight setting approaches (shown later). Hence, we will start from the comparison between GA and SPH in the capability of exploring feasible solutions. Fig. 6 presents the ratio of successful instances obtained by GA but failed to be found in SPH. We define the maximum link overload rate (MLOR) as follows:

$$\mathrm{MLOR} = \max_{(i,j)\in E}\left(\frac{\sum_{g=1}^{G} D_g \times y_{ij}^{g} - C_{ij}}{C_{ij}}\right).$$

From this definition we can see that MLOR reflects the overloading scale of the most congested link (if any, i.e., $\mathrm{MLOR} > 0$). In the figure, when the value of MLOR computed by SPH is below 5%, the GA approach can obtain feasible solutions (i.e., $\mathrm{MLOR}_{\mathrm{GA}} \leq 0$) for 65% of these instances. We can also see that, with the increase of max demand D_g, the capability of GA in finding feasible solutions is decreasing. When the MLOR value of SPH grows up to 25% (due to higher bandwidth demand D_g from individual multicast groups), the success rate of the GA approach drops to 5%. From this figure, it can be inferred that, when the group traffic demand is at the brink of causing network congestion, the GA approach has higher capability of avoiding link overloading in comparison to the other approaches. This result is expected because SPH always uses shortest paths in terms of hop count, regardless of the forecasted traffic matrix. In contrast, GA aims to intelligently compute link weights, by taking into consideration the multicast traffic matrix, so that multicast flows can be distributed more evenly, thus reducing the chance of creating bottleneck links with congestion. Obviously, it may be the case that no feasible solution is produced by GA if the traffic demand from individual groups exceeds a certain threshold. For example, if we jointly observe Figs. 6 and 9 (presented later), we can see that when D_g reaches 5000, resulting in $\mathrm{MLOR}_{\mathrm{SPH}} > 30\%$, the GA approach is not able to find any feasible solution at all.

Fig. 7 illustrates the feature of overall bandwidth conservation capability of individual schemes with the variation of maximum group traffic demand D_g. Thanks to the functionality of explicit routing through MPLS support, the TM heuristic achieves the lowest overall network resource consumption, while random link weight assignment results in the worst performance. We can also see in the figure that the GA approach exhibits the best capability in conserving bandwidth among all the hop-by-hop routing schemes. Typically, when the network is under-utilized, our proposed GA approach exhibits significantly higher performance than the conventional IP based solutions without explicit routing. For example when $D_g = 3000$, the overall bandwidth consumption of the Random and SPH solutions are higher than that of GA by 19.3% and 14.9% respectively. In comparison with the TM heuristic that needs support from MPLS overlaying, the gap from the GA approach is below 8%. However, when traffic demand grows, the performance of GA converges towards that of the SPH approach. On the other hand, although the TM algorithm exhibits significantly higher capability in bandwidth conservation when the traffic demand increases ($D_g > 4000$), this does not mean that all the obtained solutions are feasible ones.

In Figs. 8 and 9, we evaluate the capability of alleviating network congestion in our proposed solution. In the same fashion as Fig. 7, Max D_g is set in the range of 2500–6000 in order to cover the spectrum of resource provisioning (with full traffic projection), i.e.,

$$\begin{aligned}&\text{wide over-provisioning}\\&\quad\rightarrow \text{marginal over-provisioning}\\&\quad\rightarrow \text{under-provisioning (over-subscription)}.\end{aligned}$$

From the figures, we can see that in time of overwhelming multicast traffic demand, network congestion will be inevitable. Fig. 8 shows the relationship between the proportion of overloaded links and the maximum group traffic demand D_g in time of congestion. From the figure, we can see that there exist more overloaded links within the network as D_g increases. The most promising result is that, through our GA optimization, the number of overloaded links is significantly lower than all the other routing schemes. In the most congested situation ($D_g = 6000$), the average rate of overloaded links computed by GA is only 1.4%, in contrast to 12.6% by random link weight setting, 8.6% by the TM heuristic, and 4.4% by SPH respectively. On the other hand, the amount of overloaded bandwidth on the most congested links is another important parameter. An ISP should avoid network dimensioning that results in hot spots with high MLOR values. Through our simulations, we also find that the proposed GA approach achieves the lowest MLOR performance. In Fig. 9, the overloading scale is 45% of the bandwidth capacity on the most congested link in the GA approach with D_g equal to 6000, while this value reaches 110% and 59% in random link weight setting and SPH respectively. By using the explicit routing TM heuristic, the overloaded bandwidth is 78% of the original link capacity. The performance of the various approaches in Figs. 8 and 9 can be explained as follows. The random approach always results in the worst performance, as it has pure ad hoc routing logic that may easily introduce hot spots in traffic distribution. Being specifically greedy on bandwidth conservation, the TM algorithm achieves minimum resource consumption (see Fig. 7) at the expense of less-balanced traffic loading, as indicated in both Figs. 8 and 9. Compared to TM, SPH is less greedy on bandwidth conservation, which compensates to some extent its performance in terms of traffic distribution. However, SPH does not specifically cater for load balancing, as routing is not adaptive to specific multicast traffic patterns. Finally, since the GA approach applies a multiobjective policy that considers both resource conservation and load balancing according the traffic matrix, the optimized link weights make it perform the best among all the illustrated approaches.

It should be noted that the above simulation results based on the static traffic matrix have limitations, as multicast traffic is indeed very dynamic with frequent group joins and leaves. In addition, none of the above experiments is able to indicate the *service availability* when an individual join occurs. In order to address this, we emulate a sequence of events for group membership updates based on the static scenario by using the probability function proposed in [20], and we evaluate the real-time traffic condition with the group dynamics derived from the original static multicast traffic matrix. For each event, we first randomly select one group $g \in G$, and then we use the following probability function to decide whether this event is a group join or leave:

$$P_g = \frac{\omega(|V_g| - m_g)}{\omega(|V_g| - m_g) + (1-\omega)m_g}.$$

In the function, m_g indicates the instant number of active members in T_g, while $|V_g|$ identifies the *forecast* size of group g (see Section IV). ω ranging [0, 1] is known as the *invocation ratio* that controls the density of each group. For example, $\omega = 0$ means that no group joins are invoked, while $\omega = 1$ indicates full group membership invocation. In our simulation we use this function for creating 10 000 events for each data point based on the static multicast traffic matrix. When a join request is issued for group g, a node $v \in V_g$ but currently not yet on the instant multicast tree T_g is selected for group joining. Likewise, in case of a leave request for group g, an on-tree node is randomly selected for pruning from T_g.

We assume that any new join request will be blocked once link congestion on its join path has been detected. Fig. 10 illustrates the overall blocking rate with the variation of the invocation ratio ω, while maximum D_g is set to 6000. From the figure we see that more group joins are rejected as the invocation ratio grows. The reason for this is that, bandwidth consumption increases when there are more active members in each group. Once the consumed bandwidth on a link reaches its capacity, any new group join is blocked if its join path includes the congested link. On the other hand, we can see that through sophisticated network dimensioning using the proposed M-ISIS link weight optimization, the ratio of group join blockings is significantly lower than the other approaches, which indicates the drastic improvement of multicast service availability. When ω increases from 0.5 to 1.0, the total number of blockings grows very slowly with our proposed GA solution, which is in contrast to all the other conventional methods. One interesting observation is that, compared to Fig. 9 in the scenario of full traffic projection, although the provisioning performance of the GA approach results in 45% MLOR, with the configuration of the resulted M-ISIS link weights, the actual number of blocked join requests is quite low (2.1%) even in case of full group invocation. When $\omega \leq 0.7$, virtually there are no blocked group join requests. The reason for this is that, while there are overwhelming group joins, group leaves also take place at the same time, with used bandwidth resources being returned to the network. Finally, it is also worth mentioning that the MPLS-based Steiner tree approach does not exhibit strong capability in reducing the blocking rate, as the TM algorithm is greedy on bandwidth conservation and does not address the elimination of congested links.

Fig. 11 shows the overall network load versus the invocation ratio ω. The network load is defined as the mean ratio of consumed bandwidth over the link capacity, so that it can directly reflect the performance of overall bandwidth conservation. From the figure, we can see that higher invocation ratio results in higher network load. On the other hand, the TM heuristic using MPLS explicit routing always achieves the lowest network load, which means that the least bandwidth resources are consumed. In addition, the network load of the GA optimization is very close to that of the TM approach when ω is relatively small, and this again indicates that the proposed solution exhibits strong capability in bandwidth conservation in time of light traffic loading (another proof comes from Fig. 7). However, with the growth of ω, the network load by the GA approach increases more sharply than all the other approaches, and this is because more group joins can be accommodated successfully with this approach, while in the other approaches, a large number of join requests have been blocked due to network congestion (see Fig. 10) so that the total bandwidth consumption is relatively low.

VIII. Summary

In this paper, we proposed a novel network dimensioning approach aiming at offline traffic engineering for IP multicast traffic in order to support statistically bandwidth guaranteed multimedia content delivery. Our key target was a lightweight solution with not far from optimal performance and relatively low complexity. Through offline optimization and pre-configuration of MT-IGP link weights, traditional Steiner tree-based multicast traffic engineering can be reduced to PIM-SM shortest path routing that is widely supported by IP routers. This means that MPLS tunneling is not anymore a necessity for multicast traffic engineering, and using suitable MT-IGP weight setting results in reduced operational costs and better scalability in comparison to MPLS. The proposed GA-based approach also results in higher service availability for multicast group receivers, while at the same time reducing network congestion given the optimized use of resources.

As far as we are aware, the proposed approach represents the first attempt to explore multicast traffic engineering targeted to multimedia content delivery based on hop-by-hop routing, and this is in contrast to most of the current multicast TE schemes that require MPLS support. Our work complements the relevant approach for unicast traffic presented in [3], which is used by some ISPs, but also considers statistical bandwidth guarantees for QoS-aware multicast content delivery.

References

[1] N. Wang and G. Pavlou, "Bandwidth constrained IP multicast traffic engineering without MPLS overlay," in *Proc. IEEE/IFIP Management of Multimedia Newtorks and Services (MMNS)*, San Diego, CA, Oct. 2004.

[2] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, Overview and Principles of Internet Traffic Engineering 2002, RFC 3272.

[3] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000.

[4] Y. Wang and Z. Wang, "Internet traffic engineering without full mesh overlaying," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001.

[5] M. Ericsson, M. G. C. Resende, and P. M. Pardalos, "A genetic algorithm for the weight setting problem in OSPF routing," *J. Combin. Optimiz.*, vol. 6, pp. 299–333, 2002.

[6] S. Deering, Host Extensions for IP Multicasting 1989, RFC 1112.

[7] B. Yang and P. Mohapatra l, "Multicasting in MPLS domains," *Comput. Commun.*, vol. 27, no. 2, pp. 162–170.

[8] M. Kodialam, T. Lakshman, and S. Sengupta, "Online multicast routing with bandwidth guarantees: A new approach using multicast network flow," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 676–686, Aug. 2003.

[9] D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul, and F. Ansari, Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment 2002, RFC 3353.

[10] S. Yasukawa, "Requirements for Point to Multipoint Traffic Engineered MPLS LSPs", Internet Draft 2004 [Online]. Available: draft-ietf-mpls-p2mp-requirement-04.txt

[11] J. Cui, J. Kim, A. Fei, A. Faloutsos, and M. Gerla, "Scalable QoS multicast provisioning in diff-serv-supported MPLS networks," in *Proc. IEEE GLOBECOM*, 2002.

[12] B. Fenner, Protocol Independent Multicast – Sparse Mode (PIM-SM): Protocol Specification (Revised) 2006, RFC 4601.

[13] T. Przygienda, N. Shen, and N. Sheth, M-ISIS: Multi Topology (MT) Routing in IS-IS 2005 [Online]. Available: draft-ietf-isis-wg-multi-topology-11.txt

[14] P. Psenak, A. Roy, L. Nguyen, and P. Pillay-Asnault, "MT-OSPF: Multi Topology (MT) Routing in OSPF", Internet Draft 2006 [Online]. Available: draft-ietf-ospf-mt-06.txt

[15] G. Retvari, R. Szabo, and J. Biro, "On the representability of arbitrary path sets as shortest paths: Theory, algorithms and complexity," in *Proc. IFIP Netw.*, 2004.

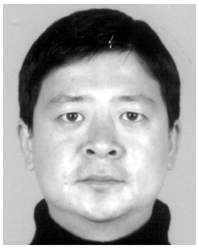
[16] H. Takahashi and A. Matsuyama, "An approximate solution for the Steiner problem in graphs," *Math. Japonica*, vol. 6, pp. 533–577.

[17] C. P. Low and N. Wang, "An efficient algorithm for group multicast routing with bandwidth reservation," *Comput. Commun.*, vol. 23, no. 18, pp. 1740–1746, 2000.

[18] *GT-ITM*, [Online]. Available: http://www.cc.gatech.edu/projects/gtitm/

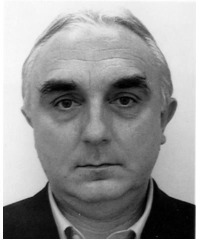
[19] S. Cheung *et al.*, "On the use of destination set grouping to improve fairness in multicast video distribution," in *Proc. IEEE INFOCOM*, San Francisco, CA, Apr. 1996.

[20] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Select. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Sep. 1988.



Ning Wang (M'01) received the B.Eng degree in computing from Changchun University of Science and Technology, Changchun, China, in 1996, the M.Eng. degree in electronic engineering from Nanyang Technological University, Singapore, in 2000, and the Ph.D. degree in electronic engineering from the University of Surrey, Guildford, U.K., in 2004.

He is currently a Postdoctoral Research Fellow in the Center for Communication Systems Research, University of Surrey, working on several IST research projects funded by the European Commission. His major research area includes multicast communication, quality-of-service provisioning, traffic engineering, and network management.



George Pavlou (M'95) received the Diploma degree in electrical and mechanical engineering from the National Technical University of Athens, Athens, Greece, and the M.Sc. and Ph.D. degrees in computer science from University College London, London, U.K.

He is currently a Professor of Communication and Information Systems at the Centre of Communication Systems Research, Department of Electronic Engineering, University of Surrey, Guildford, U.K., where he leads the activities of the Networks Research Group. He has previously been a Senior Research Fellow and Lecturer in the Department of Computer Science, University College London, where he led research activities on network and service management. His research interests focus on network management, networking, and service engineering.

Dr. Pavlou has been a Chartered Engineer and Member of the Technical Chamber of Greece since 1984. He is on the Editorial Board of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, the *IEEE Communication Surveys and Tutorials*, and the *Journal of Network and Systems Management*. He is Network and Service Management Series Editor of *IEEE Communications Surveys*.