INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

PUBLICATION
INTERNE
N° 1548

USING MPLS FOR MULTICAST TRAFFIC ENGINEERING

ALI BOUDANI AND BERNARD COUSIN

ISSN 1166-8687

IRISA
CAMPUS UNIVERSITAIRE DE BEAULIEU - 35042 RENNES CEDEX - FRANCE

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES
Campus de Beaulieu – 35042 Rennes Cedex – France
Tél. : (33) 02 99 84 71 00 – Fax : (33) 02 99 84 71 71
http://www.irisa.fr

Using MPLS for Multicast Traffic Engineering

Ali Boudani and Bernard Cousin[*]

Thème 1 — Réseaux et systèmes
Projet Armor

Publication interne n ° 1548 — July 2003 — 14 pages

Abstract: In this paper, we present a framework for multicast traffic engineering. We define first the multicast traffic engineering and we study after its particularity comparing to unicast traffic engineering. We study merging multicast and MPLS as traffic engineering tool. We present also a taxonomy of different MPLS proposals for multicast TE. We describe briefly our approach, the MPLS multicast tree (MMT) protocol which utilizes MPLS LSPs between multicast tree branching node routers in order to reduce forwarding states and enhance scalability. We discuss also implementing a simulator for MMT and finally we present some simulation results.

Key-words: MPLS, Multicast, Routing, Simulation, Network

(Résumé : tsvp)

---

[*] { Ali.Boudani}{Bernard.Cousin}@irisa.fr

L'utilisation de MPLS pour l'ingénierie de trafic *multicast*

Résumé : Dans cet article, nous étudions l'ingénierie de trafic *multicast*. Nous définissons d'abord l'ingénierie de trafic *multicast* et sa particularité par rapport à l'ingénierie de trafic *unicast* (IT). Nous étudions la combinaison du *multicast* et de MPLS en tant qu'outil d'ingénierie de trafic. Nous présentons également une taxonomie de différentes propositions MPLS pour l'ingénierie de trafic *multicast* (ITM). Nous décrivons brièvement notre approche, le protocole MMT (*MPLS Mutlticast Tree*) qui utilise les chemins LSP (*Label Switched Path*) entre les routeurs de branchement de l'arbre *multicast* afin de réduire des états de routage et d'améliorer la résistance au facteur d'chelle. Nous discutons également la mise en oeuvre d'un simulateur pour MMT et nous présentons finalement quelques résultats de simulation.

Mots clés : MPLS, PIM-SM, NS, Multicast

Contents

1 Introduction

The best-effort service model existing in the current Internet is inadequate in meeting the growing demands of the next generation applications, most of which demand QoS assurances for effective data delivery. The network will need to provide a different type of service for applications sensitive to network congestion, delay, jitter and packet loss such the video conferencing where video and audio quality are essential for the communication.

In order to provide QoS to users across the Internet, we either increase the bandwidth available such that the extra capacity of the network allows all users to meet their appropriate QoS and we avoid by that network congestion, or we suppose that bandwidth can never be considered unlimited and therefore resource allocation should be appropriately prioritized among users. As the available bandwidth to end users increases, new applications are continually being developed which erode gains in network capacity. Thus, for the foreseeable future, some form of resource provisioning is necessary to provide QoS across the Internet. One of the more promising models for providing QoS across the Internet is the traffic engineering model which attempts to appropriately provision limited network resources in order to provide QoS.

Traffic engineering (TE) and quality of service (QoS) are related since TE deals to manage traffic within a network while QoS deals to reserve resources to guaranty the service. The factors driving the need for better TE tools include in addition to the quality of service, interdependent tunable parameters, network growth, and traffic variability. In the best effort model, this means that available network resources are not being used well, resulting in higher delays while there is a potential for providing better quality of service (QoS) by reducing delay and packet losses and increasing throughput experienced by end users using the same network infrastructure. This results in a minimization of the vulnerability of the network to service outages arising from errors, faults, and failures occurring within the infrastructure.

Apart from QoS assurances, another important component aspect of the Internet usage is bandwidth utilization. Several evolving applications like WWW, video/audio on-demand services, and teleconferencing consume a large amount of network bandwidth. Multicasting is a useful operation for supporting such applications. Using multicast services, data can be sent from a source to several destinations by sharing the link bandwidth. A single transmission is needed for sending a packet to n destinations, while n independent transmissions would be required using unicast services. By reducing the information being transmitted across the network, multicast essentially increases the QoS given to other users of the network due to the additional available bandwidth in the network.

MPLS [1] as a traffic engineering tool has emerged as an elegant solution to meet the bandwidth-management and service requirements for next generation Internet protocol (IP) based backbone networks. We think that multicast and MPLS are two complementary technologies and merging these two technologies where multicast trees are constructed in MPLS networks will enhance the network performance and present an efficient solution for multicast scalability and control overhead problems. Multicast attempts to conserve network bandwidth, while traffic engineering attempts to provision the bandwidth in an appropriate fashion to users.

2 Traffic Engineering

TE is the process of controlling how traffic flows through a network in order to facilitate efficient and reliable network operations while simultaneously optimizing network resource utilization and traffic performance [2]. Besides, individual backbone networks are growing rapidly in size, speed, and scope. As a result, network management functions that could once be handled by a small group of people, based on intuition and experimentation, must now be supported by effective traffic engineering tools that unite configuration and usage information from a variety of sources. The level of manual intervention involved in the traffic engineering process should be minimized whenever possible.

Otherwise, TE is needed in the Internet mainly because current interior gateway protocols (IGPs) always use the shortest paths to forward traffic. While the approach of shortest paths is very simple to scale to very large networks and conserves network resources, but it does not always make good use of these resources and may also cause the following problems:

- The shortest paths from different sources overlap at some links, causing congestion on those links.
- The traffic from a source router to a destination router exceeds the capacity of the shortest path, while a longer path between these two routers is underutilized.

The first problem can be solved by expansion of link capacity, or by application of classical congestion control techniques, or both. Classical congestion control techniques attempt to regulate the demand so that the traffic fits onto available resources. Classical techniques for congestion control include: rate limiting, window flow control, router queue management, schedule-based control, and others [3]. The second problem, namely congestion resulting from inefficient resource allocation, can usually be addressed through TE.

A constraint-based routing (CBR) and an enhancement of existing IGPs may be needed to permit unicast forwarding through explicit routes.

2.1 Multicast TE

Multicast TE means that a multicast forwarding tree is built through policies and explicit routes, instead of topology. It can be use as same as unicast TE to achieve efficient network resource utilization.

Multicast traffic has some specific characteristics due to the multicast routing protocols nature [4]. The most of the currently proposed multicast routing protocols are based on reverse path forwarding (RPF) to setup forwarding states on intermediate routers between the source and the destinations. But RPF is based on the idea that paths are symmetric in the network. However, when routing constraints are introduced, there is no guarantee that this is the case. Hence, RPF will cause forwarding on a sub-optimal path (in QoS routing) or might even prevent receivers from receiving traffic from certain (or all) sources (in policy routing). This check must turned off or the multicast routing protocol must be able to obtain the constraint RPF via a constraint based routing (CBR) API. Multicast trees should be constructed taking into consideration the dynamism in the receiver set, and the receiver's heterogeneity, that is, receivers with different service requirements in terms of delay or jitter. This will introduce important modifications to CBR and conventional multicast routing protocols. Indeed, fast recovery for paths failure is very important in multicast TE since this failure may influence all the tree and not only the link in failure. Otherwise, load balancing should be carefully used since a packet should not pass by the same link more than one time. Finally, multicast forwarding is done based on the multicast IP address and that's why it is very difficult to aggregate multicast traffic since receivers can be located anywhere in the Internet.

The multicast traffic engineering trees can be built by expanding the existing protocols. There are two categories of protocols depending on the tree setup:

- Sender initiated tree setup: this kind of tree can have limited number of receivers with very rare join and prune action. Multicast trees are computed by the first-hop router from the source (root), based on sender traffic advertisements.
- Receiver initiated tree setup: this kind of tree can have a large number of receivers and they join and prune quite frequently. Multicast trees are computed from receivers to the root. Each receiver-side router independently computes a QoS-accommodating path from the source, based on the receiver reservation. This path can be computed based on unicast routing information only, or with additional multicast flow-specific state information. In any case, multicast path computation is broken up into multiple, concurrent unicast path computations.

Finally, MPLS label switching can be used to forward unicast traffic through explicit routes and multicast traffic down the explicit tree to avoid RPF checking.

MPLS shows several advantages over conventional network layer forwarding [2, 1, 5]. Focusing on the advantages of the layer two switching protocol over , Multicasting over MPLS networks can benefit from the multicast reduce of traffic on one hand, and MPLS flexibility, speed and quality of service on the other hand.

3 Taxonomy of MPLS proposals for multicast TE

IP multicast protocols have different characteristics (scalability, computational complexity, latency, control message overhead, tree type, etc...). A framework for IP multicast deployment in an MPLS environment is proposed in [4]. Issues arising when MPLS techniques are applied to IP multicast are overviewed. Following characteristics are considered: aggregation, flood and prune, co-existence of source and shared trees, uni/bi-directional shared trees, encapsulated multicast data and loop free ness, and RPF check. The pros and cons of existing IP multicast routing protocols in the context of MPLS are described and the relation to the different trigger methods and label distribution modes are discussed. The framework did not lead to the selection of one superior multicast routing protocol but it concluded that different IP multicast routing protocols could be deployed in the Internet.

Using PIM-SM [6] *join* messages to distribute MPLS labels for multicast routes is proposed in [7] (called hereinafter PIM-MPLS). A piggy-backing methodology is suggested to assign and distribute labels for multicast traffic for sparse-mode trees. The PIM-SM *join* message is expanded to carry an MPLS label allocated by the downstream LSR. MPLS is not used with all its efficiency as a TE tool since the multicast tree still constructed using the RPF tree checking without constraints. In [5], we proposed a simulator for this methodology by using the MNS [8] (MPLS network simulator). We think that the *join* message in PIM-SM should be expanded to carry the explicited routed path towards the RP. A PIM-SM router always sends *join/prune* towards the upstream router listed in the explicited routed path. It can also carry other constraints, such as color or bandwidth. A new message *join-nak* can be sent from upstream to downstream if the upstream can not satisfy the constraints listed in the *join* message.

In [9], authors consider the problem of supporting Ip multicast efficiently within MPLS environment for both PIM dense mode and sparse mode. They suggest a data-driven, per source assignment of labels to traffic on the shared tree and they present a common scheme for implicitly distributing and binding labels to multicast FECs. Authors suppose also like the previous proposal that multicast trees will be constructed using the RPF tree checking without constraints.

In [10], authors propose to engineer paths for IP multicast traffic in a network by directing the control messages to setup multicast trees on engineered paths. This proposal partitions the multicast traffic engineering problem such that multicast routing protocols do not have to be modified to allocate resources for multicast traffic nor do resource allocation protocols such as RSVP or CR-LDP have to be able to setup forwarding states (in this case labels) like multicast routing protocols. Resources are allocated on the same trip that paths are selected and setup. This prevent the problem of data being forwarded on branches of the tree where resources have not being allocated yet. An important aspect of this proposal is that it enables multicast paths to be engineered in an aggregatable manner, allowing this solution to scale in the backbone. But while this proposal uses MPLS (label and explicit route object) to cause engineered paths to be selected, it forwards data using multicast routing.

Another interesting proposal is aggregated multicast [11]. The key idea of aggregated mulicast is that, instead of constructing a tree for each individual multicast session in the core network, one can have multiple multicast sessions share a single aggregated tree to reduce multicast state and, correspondingly, tree maintenance overhead at network core. In this proposal there is two requirements: (1) original group addresses of data packets must be preserved somewhere and can be recovered by exit nodes to determine how to further forward these packets; (2) some kind of identification for the aggregated tree which the group is using must be carried and transit nodes must forward packets based on that. In group to aggregated tree matching, complication arises when there is no perfect match or no existing tree covers a group (leaky matching). The disadvantage in leaky matching is that certain bandwidth is wasted to deliver data to nodes that are not involved for the group. Thus, bandwidth can be crucial factors for provisioning QoS in multicast networks and even for best effort Internet. To handle aggregated tree management and matching between multicast groups and aggregated trees, a centralized management entity called tree manager is introduced.

In [12], extensions to CR-LDP are proposed to construct multicast trees immediately on L2. Thus the mapping of L3 trees onto L2, as described in [6] and [9] is not needed. All of the elements of the entire tree must be carried in the initial label request. Given this and given that it is highly undesirable to fragment such requests, this style of tree building is pimarily applicable to trees with a small number of receivers.

In [13], [14], extensions to LDP and RSVP for MPLS multicasting services are proposed. In these two proposals multicasting functions of LDP and RSVP are independent of traditional IP-based multicast routing protocols such as DVMRP, MOSPF, PIM, etc and multicast trees are already calculated by a special entity.

To enable MPLS based multicasting, the tree formation with *join, leave, destroy* and *RPF* messages should be directly implemented in LDP and RSVP. New messages and extending of existing messages are studied to introduce to the LDP and the RSVP. Multicasting message (for *join, leave* and *destroy* operations) is created. Extensions to *hello, notification, path* messages, the label request, the label mapping and the multicast forwarding table are introduced. These two proposals require a MPLS and multicast routing protocols to be merged, an exercise which tend to increase the complexity of multicast traffic engineering while not providing any means of aggregating multicast traffic engineering.

The complete tree information should be stored in all LSR-RP (branching nodes in the tree). Multicast *hello* messages are used to inform the LSRs of the multicasting source and group IP address of the multicasting tree. When the number of group grows the number of *hello* messages grows also. And since we will send for every source and group a *notification* message, when the number of group grows the number of *notification* messages grows also. It should be noted that a new table for multicast should be created independent of the existing unicast table. It is not very clear how the source will choose the LSR-RP.

3.1 The Multicast MPLS Tree Proposal

Using MPLS with multicast has many benefits not only for reducing multicast forwarding states but also for traffic engineering and QoS issues. In this paper, we only focus on the scalability problem. We propose a novel approach that uses MPLS LSPs between multicast tree branching node routers in order to reduce forwarding states and enhance scalability.

In [15], we proposed a new approach to construct multicast trees in MPLS networks. Each domain contains a network information manager system (NIMS) for each group, charged to collect *join* and *leave* messages from all group members in that domain. The NIMS is elected through a mechanism similar to the one used to elect the RENDEZ-VOUS router in PIM-SM [6]. After collecting all *join* messages, the NIMS computes the multicast tree for that group in the domain (using the Dijsktra's algorithm). The computation for a group means discovering all branching node routers for that group. The NIMS sends then *branch* messages to all branching node routers to inform them about their next hop branching node routers. On receiving this message, a branching node router creates a multicast forwarding state for the multicast session. Once branching node routers and their next hops are identified, packets will be sent from a branching node router to another until achieving their destinations.

Already established MPLS LSPs are used between multicast tree branching node routers in order to reduce forwarding states and enhance scalability. When a multicast packet arrives to the ingress router of an MPLS domain, the packet is analyzed according to its multicast IP header. The router determines who are the next hop branching node routers for that packet. Based on this information, multiple copies of the packets are generated and an MPLS label is pushed in to the multicast packet according to next hop branching node router. When arriving to a next hop branching node router, the label is pulled up and again the same process is repeated. This process should be repeated until the packet arrives to its destination (see Fig.1).

Only those routers acting as multicast tree branching node for a group need to keep forwarding state for that group. Other routers between two branching nodes do not need to store multicast states. Unicast is used between two branching node routers. This way the total number of multicast forwarding states may be significantly reduced. In our approach we will use the same MPLS label for multicast traffic that follows the same path than unicast traffic. Other approaches use different labels for multicast and unicast traffic which mean the need of encoding techniques and additional overheads in routers. When arriving to a LAN, the packet unlabeled can be delivered by conventional multicast protocols according to IGMP [16] informations.

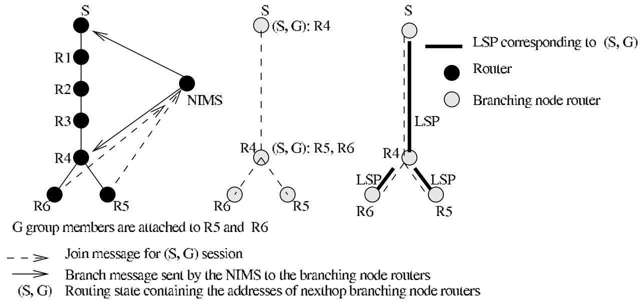

Figure 1: The MPLS multicast tree construction

4 Implementing the simulator for MMT

MPLS code in NS does not work with multicast routing, particularly because (1) there is no label setup mechanism for multicast groups, (2) there is no multicast replicator to cooperate with MPLS classifier, and (3) MPLS header contains pointers, which do not work with multicast replicator. In this section, we describe the modifications needed to allow multicast packet transmission in MPLS networks without implementing a new protocol. Two main points are to be considered: information tables of MPLS nodes, and multicast packet transmission. Our major objectif was implementing MMT without major modifications of the unicast MPLS code already exist in NS.

4.1 Information tables of MPLS nodes

As mentioned in [7], an MPLS node contains three information tables: LIB (Label Information base), PFT (Partial Forwarding Table), and ERB (Explicit Routing information Base). To apply the MMT proposition, a mapping of the (S, G) session to more than one FEC on one hand, and a mapping of each FEC to one <incoming label, incoming interface> and thus to one <outgoing label, outgoing interface>, on the other hand, are needed. The information base at the MPLS nodes must be modified as shown in Fig. 2.

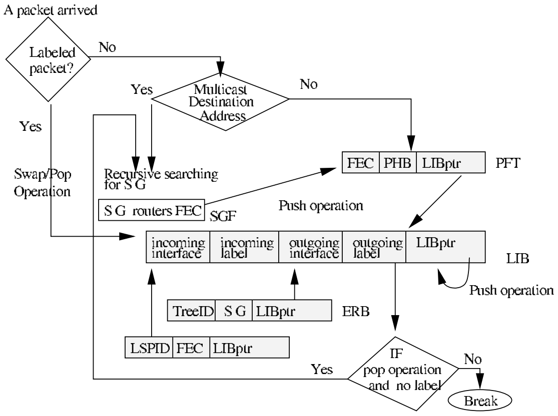

Figure 2: Structure of tables for MMT packet switching

For the first mapping, the SGF (Source Group Fec) table is defined. This table exists only in branching node routers and includes three fields: Source, Group, and FEC. This table is filled at each node after receiving the *branch* message sent by the NIMS. In branching node routers, more than one FEC will result from the recursive searching in the SGF table. Each FEC is mapped to a couple of <incoming label, incoming interface> in teh LIB table. As for the second mapping, each <incoming label, incoming interface> is mapped to exactly one <outgoing label, outgoing interface>. The LIB table remains unchangeable with one <incoming label, incoming interface> for one <outgoing label, outgoing interface>.

4.2 Multicast packet transmission

Data is transmitted exactly as in unicast MPLS packets with only one difference at branching nodes. The procedure is done as follows: When a labeled packet arrives, a swap/pop operation is executed and the LIB table is examined. If as a result of a pop operation, the packet remains without label, a recursive search is done in the SGF table to attribute the packet to FECs and thus to <incoming label, incoming interface> couples. For each FEC, a packet copy is created, and then label swapped with the corresponding outgoing label, and then transmitted to the outgoing interface.

5 Evaluation and Simulation

MMT was evaluated in [15] in terms of scalability (state information requirement and control messages overhead) and efficiency (tree cost and data processing). The state information requirement can be measured using the average multicast forwarding table size. The control messages overhead can be measured in terms of average number of control messages sent per link or the total percentage of bandwidth spent on control traffic. MMT allows only the shortest path trees, which are the most efficient for data forwarding. Besides, since we are using MPLS processing at routers, our approach may be considered more efficient in data movement than other schemes.

Multicast address aggregation is important since multicast groups may share some links in their multicast trees. In conventional multicast, it is not possible to aggregate multicast IP addresses. Receivers can be located anywhere in the Internet, there is no other alternative than having one entry by multicast IP address in the multicast routing table. Since in our approach, we are using MPLS, the aggregation problem of multicast IP addresses can be transformed to a simple aggregation of labels.

5.1 Simulation Analysis

We simulate MMT in NS (Network Simulator) [17] to validate the basic approach behavior and its effectiveness in state reduction and tree construction. The performance of MMT is compared to PIM-MPLS. PIM-MPLS in our simulations refers to the simulator described in [5]. In [5] we presented a simulator for multicast routing over an MPLS network where we choosed PIM-SM (source specific tree) as the multicast routing protocol. In this paper we present the MMT protocol simulation which will be compared to PIM-MPLS.

5.2 Simulation Scenario

We present in our simulation two models generated using the GT-ITM generator [18]: each with flat graph of 100 nodes and all the links in the network are identical bidirectional links with 20Mbps bandwidth. The topology of the first model is based on the first Waxman algorithm [19] and used as a dense mode network with 0.3 as the node degree distribution. The topology of the second model is based on a pure random algorithm in 5 domains and used as a sparse mode network. Four domains contain receivers and sources only, while the fifth domain is considered as the core domain. NT sources and Nr receivers are randomly deployed in the network. A receiver join randomly the tree. Table 1 summarizes the parameters used in the simulation.

Table 1: Summary of Simulation Parameters

N	100	number of nodes in the network
NT	10, 20, 30, 40, 50, 60	percentage of sources in the network (number of trees)
Nr	3, 6, 9, 12, 15, 18	number of receivers for each source

The forwarding table size in all routers in the network using the pure random sparse mode model is shown in Fig. 3 and using the waxman model is shown in Fig. 4.

The horizontal axis is the percentage of sources that are active in the network, and the vertical axis is the overall forwarding table size in the network. The poly-lines labeled PIM-x and MMT-x show the overall forwarding table size for PIM-MPLS and MMT protocols respectively when the number of receivers per group is x.

The forwarding table size grows with the number of active groups and the number of receivers. From Fig. 4 and Fig. 3 we can see that the relative state information reduction of MMT is roughly 40% and 80% respectively compared to PIM-MPLS. We deduce also that our protocol is more suitable for sparse mode networks and for groups with few members.

6 Conclusion and Future Works

In this paper, we presented a framework for multicast traffic engineering. We defined first the multicast traffic engineering and we studied after its particularity comparing to unicast traffic engineering. We studied merging multicast and MPLS as traffic engineering tool. We presented also a taxonomy of different MPLS proposals for multicast TE. We described briefly our approach, the MPLS multicast tree protocol which utilizes MPLS LSPs between multicast tree branching node routers in order to reduce forwarding states and enhance scalability. We discussed also implementing a simulator for MMT and finally we presented some simulation results.

Our future work will be complete specification of the protocol (control plane with different messages and router behavior algorithms) and also more simulation comparison between our approach, other MPLS multicast approaches and the conventional multicast protocols.

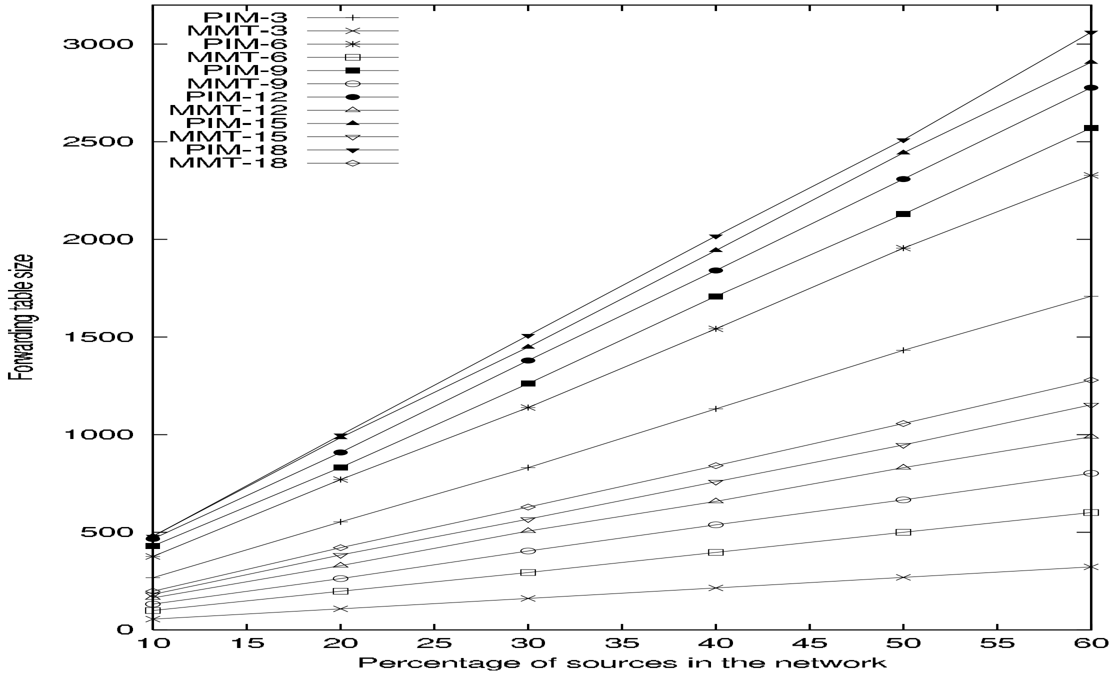

Figure 3: Forwarding table size - pure random sparse mode model

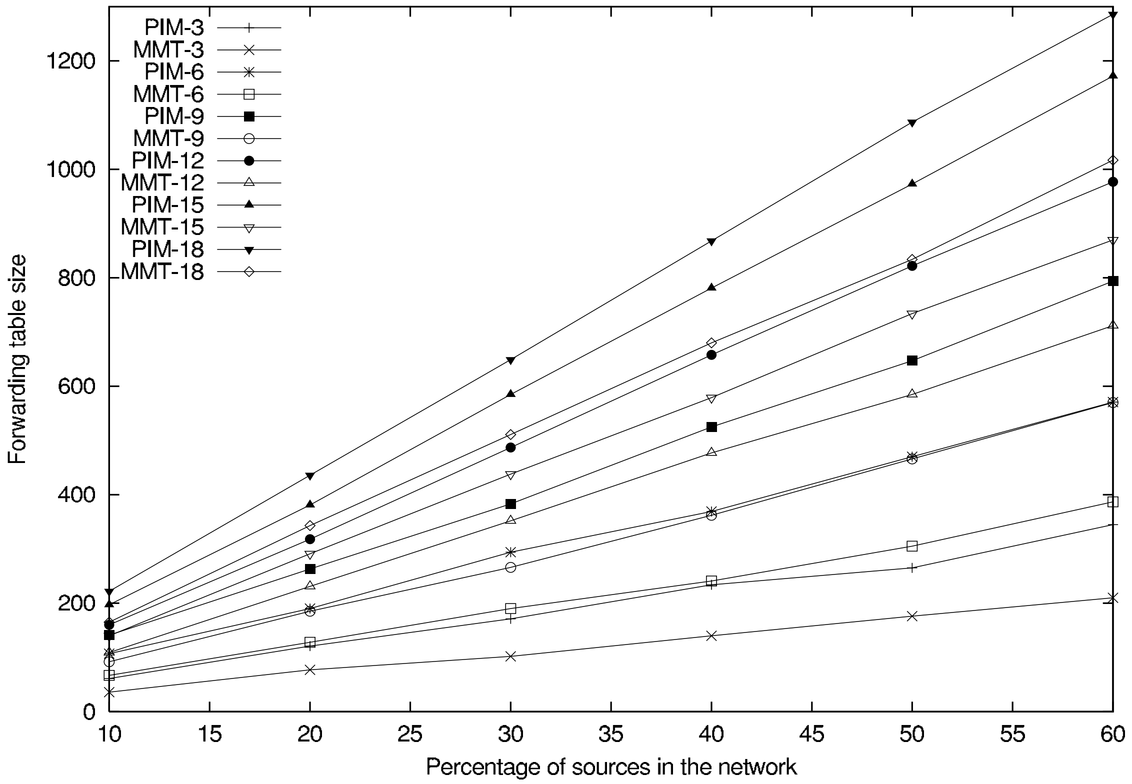

Figure 4: Forwarding table size - Waxman model

References

[1] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. IETF RFC3031, January 2001.

[2] X. Xiao, A. Hannan, B. Bailey, and L. Ni. Traffic engineering with MPLS in the Internet. *IEEE Network*, 14(2):28–33, March/April 2000.

[3] C. Yang and A. Reddy. A taxonomy for congestion control algorithms in packet switching networks. *IEEE Network*, 9(5), July/August 1995.

[4] D. Ooms, W. Livens, A. Acharya, F. Griffoul, and F. Ansari. Framework for IP multicast in MPLS. IETF Internet draft, January 2002.

[5] A. Boudani, C. Jawhar, B. Cousin, and M. Doughan. A simulator for multicast routing over an mpls network. Technical report 1493, IRISA, October 2002.

[6] D. Estrin et al. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. IETF RFC 2362, 1998.

[7] D. Farinacci, Y. Rekhter, and E. Rosen. Using PIM to distribute MPLS labels for multicast routes. IETF Internet draft, November 2000.

[8] G. ahn and W. Chun. Overview of MPLS network simulator: Design and implementation. Chungnam National University, Korea,http://flower.ce.cnu.ac.kr/ fogl/mns.

[9] A. Acharya, F. Griffoul, and F. Ansari. Ip multicast support in mpls networks. IETF Internet draft, February 1999.

[10] C-Y. LEE, L. Andersson, K. Carlberg, and B. Akyol. Engineering paths for multicast traffic. IETF Internet draft, October 1999.

[11] A. Fei, J. Cui, M. Gerla, and M. Faloutsos. Aggregated multicast: An approach to reduce multicast state. In *Proceedings of the Third International COST264 Workshop (NGC 2001) UCL. London*, number 2233 in LNCS, pages 172–188, november 2001.

[12] D. Ooms, R. Hoebeke, and P. Cheval. Mpls multicast traffic engineering. IETF Internet draft, February 2001.

[13] J. Chung, M. Benito, H. Chhabra, G. Cho, and P. Rasiah. Ldp extensions for mpls multicasting services. IETF Internet draft, February 2002.

[14] J. Chung, M. Benito, H. Chhabra, G. Cho, and P. Rasiah. Rsvp-te extensions for mpls multicasting services. IETF Internet draft, February 2002.

[15] A. Boudani and B. Cousin. A new approach to construct multicast trees in mpls networks. In *Seventh IEEE Symposium on Computers and Communications*, pages 913–919, 2002.

[16] B. Cain, S. Deering, B. Fenner, I. Kouvelas, and A. Thyagarajan. Internet group management protocol, version 3. IETF Internet draft, January 2002.

[17] K. Fall. and K. Varadhan. The NS Manual. UC Berkeley, LBL, USC/ISI, and Xerox PARC, January 2001.

[18] E. Zegura, K. Calvert, and S. Bhattacharjee. How to model an internetwork. In *INFOCOM*, 1996.

[19] B. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, December 1988.