Improving Scalability with the Use of the Overlay Multicast Protocol (OMP)

Hind O. Al-Misbahi and Arwa Y. Al-Aama*

College of Computing and Information Technology,
King Abdul Aziz University, Jeddah, Saudi Arabia
E-mail: ha.almis@gmail.com E-mail: aalaama@kau.edu.sa
*Corresponding author

Abstract: Demands to increase network usage efficiency and scalability do exist. IP and overlay multicast provide multicasting services by enabling data to be sent to several recipients simultaneously over a network. Although MPLS, a high performance method for forwarding packets, has many benefits, implementation of multicast on MPLS still suffers from IP multicast limitations. This paper proposes the Overlay Multicast Protocol (OMP) in which the overlay approach is applied on MPLS networks to improve the scalability of multicasting. Comparisons of OMP with Protocol Independent Multicast-Sparse Mode and Protocol Independent Multicast-Source Specific Multicast are presented, showing improved scalability when using OMP.

Keywords: overlay multicast; internet protocol; MPLS; PIM-SM; protocol independent multicast-sparse mode; PIM-SSM; protocol independent multicast-source specific multicast.

Reference to this paper should be made as follows: Al-Misbahi, H.O. and Al-Aama, A.Y. (xxxx) 'Improving Scalability with the use of the Overlay Multicast Protocol (OMP)', *Int. J. Internet Protocol Technology*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Hind O. Al-Misbahi has been a Lecturer at the Department of Information Technology at the College of Computing and Information Technology, King Abdul Aziz University (KAU), Jeddah, Saudi Arabia, for two years. She carries a BSc and an MSc in Computer Science. Her research interests include networking in general and MPLS in specific. She plans to pursue her PhD in that field.

Arwa Al-Aama is an Assistant Professor at the Computer Science Department at King Abdul Aziz University, Jeddah, Saudi Arabia, and the Director of e-Communication at the Jeddah Municipality, received her MSc and DSc in Computer Science from the George Washington University and has several published papers. She was awarded several research grants and has won several awards, including a Microsoft and ACM international research award and the Microsoft Arabia Most Distinguished Faculty Member.

1 Introduction

Data transfers over a network in one of the three known ways: unicast, in which traffic is sent to a single destination; broadcast, in which traffic is sent to all users of a network; or multicast, which lies between unicast and broadcast methods, where traffic is sent only to specific users of the network.

With the availability of improved technologies and the phenomenal growth in the number of online users worldwide, more group communication applications exist today than ever before. Examples include content distribution, teleconferencing, media streaming, distance learning, online gaming and collaborative workspaces. Multicasting enables the transmission of information to several receivers at the same time efficiently using one-to-many or many-to-many models. In IP multicasting, multicast is implemented in the IP layer. However, IP multicast has not yet been widely adopted owing to concerns related to scalability, deployment and network management.

To address the issues of IP multicast services, ALM or Application Level Multicast is used, in which the multicast functions are implemented at the application layer rather than at the IP layer. This is also known as the overlay multicast. In ALM, the multicast tree is constructed on the top of a virtual network, which is composed of some nodes.

Furthermore, MPLS is an advanced forwarding scheme that extends routing with respect to packet forwarding and path controlling. MPLS addresses several network issues such as speed, Quality-of-Service (QoS) management and traffic engineering. Implementing multicast on MPLS also suffers from the scalability problem, which limits the concurrent number of groups that can be served and the group sizes. The following is a description of both multicast and MPLS.

1.1 Multicast

IP multicast is the first created model of multicasting (Almeroth, 2000). In any IP multicast, there is a need to maintain a forwarding tree for each multicast group. Each tree requires keeping state information at each router at that tree. As the number of groups and the group sizes increases, the amount of state information that must be kept also increases, which in turn leads to the scalability problem. Despite the early invention of the IP multicast service, it is still far from being widely deployed. This is due to several concerns related to scalability, deployment, network management and the lack of appropriate charging models.

1.1.1 PIM-SM and PIM-SSM

Protocol Independent Multicast-Sparse Mode (PIM-SM) is an IP multicast routing protocol designed to be used in Wide Area Networks (WANs), where groups are sparsely distributed. It is called protocol independent because it can use the route information of any unicast or multicast routing protocol. Sparse mode means that the protocol is used for situations where multicast groups are lightly populated across a large region. In this mode, the number of the subnets with receivers (i.e., group members) is significantly smaller than the whole number of subnets at the WAN.

Protocol Independent Multicast-Source Specific Multicast, or PIM-SSM, is a subset of PIM-SM. Any router implementing PIM-SM can also implement PIM-SSM. PIM-SSM is a source-specific protocol, which builds a Shortest Path Tree (SPT) between the source and the receivers, i.e. there is one tree for each source, unlike PIM-SM, in which all the sources of one group share the same tree (Fenner et al., 2006).

1.1.2 Overlay Multicast

Overlay multicast was originally introduced to address IP multicast limitations. The overlay is a virtual topology built above the physical network. It is composed of the nodes that are proxies or end hosts that need to participate in the multicast group. Table 1 compares overlay and IP multicasting.

Table 1 Comparing overlay and IP multicasting

	Overlay multicast	*IP multicast*
Scalability	Less pressure on network core	Higher pressure on network core
Deployment	Install proxies/Install ESM application	Update network infrastructure
Network Management	Easier to support security and access control	Harder to support security and access control

In overlay multicast, the connections between the nodes are unicast paths and may go through several routers. There are several methods to classify overlay multicast. One of them is based on the place where the multicast services are implemented. Depending on this criterion, overlay multicast can be classified into either: End System Multicast (ESM) or Proxy-Based Multicast (PBM) (Zhu et al., 2005). In ESM, the multicast functionalities shift from core routers to end systems. While in PBM, the multicast functionalities shift from core routers to proxies, which are called Multicast Service Nodes (MSNs). While ESM has more flexibility, it places a substantial burden on the end systems and does not scale well in terms of large group sizes (Zhu et al., 2005). As this research uses PBM, throughout this paper, any reference to the term overlay multicast refers to PBM.

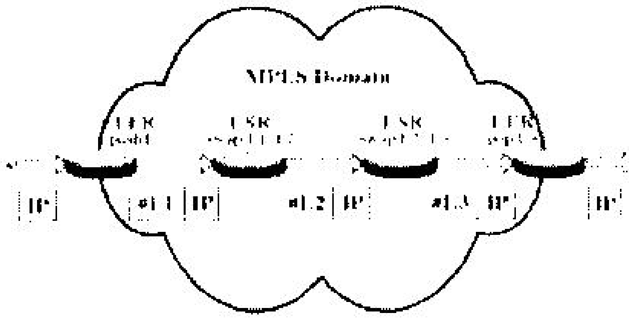
1.2. MPLS

MPLS is a technology in which each MPLS node in the route between the source and the destination forwards data packets using a label attached to the packet. This process is called label switching. The goal is to switch a packet between routers depending on a small fixed format label rather than performing a lookup on the destination address, which requires more time. Currently, MPLS is gaining more popularity and is being used in more applications. An MPLS capable router is called a Label Switching Router (LSR).

In an MPLS network, a label is inserted in a packet header when it enters the network. At each hop, the packet is routed based on the value of the incoming interface and label and dispatched to an outwards interface with a new label value. The path in which data travels in a network is defined by the transition in label values, as the label is swapped at each LSR. This path is called the Label Switching Path (LSP). Since the mapping between labels is constant at each LSR, the path is determined by the initial label value (Rosen et al., 2001). At the ingress to an MPLS network, each packet is examined to determine which LSP it should use and what label to assign to it. This decision is based on factors including the destination address, the QoS requirements, and the current state of the network.

Figure 1 shows the packet forwarding in an MPLS network. When an unlabelled packet reaches the ingress Label Edge Router (LER), which is an MPLS LSR that connects an MPLS domain with a node, which is outside the domain, it determines the Forwarding Equivalence Class (FEC) of that packet and pushes the suitable label on the packet. FEC is a group of IP packets, which are forwarded in the same manner (e.g., over the same path, with the same forwarding treatment). Then, the subsequent LSRs swap the label. Finally, the egress LER pops the label and forwards the IP packet outside the MPLS domain value (Rosen et al., 2001).

Figure 1 Packet forwarding across an MPLS network (see online version for colours)



Source: Adopted from Pointurier (2002)

1.3 MPLS and the overlay multicast

The fundamental idea of MPLS involves assigning short fixed length labels to the packets at the ingress point of the network. In an ATM environment, the label is encoded in the VCI/VPI field. In an IP network, a 32-bit 'shim' header is inserted between the network layer header and the data link layer header. When packets are forwarded within an MPLS domain, the MPLS capable routers, or the LSRs, only examine the label rather than the IP header. In MPLS networks, the routing needs less time because it depends on the label instead of looking up to the destination address in the IP header.

Some of the network applications need to implement multicast services in MPLS networks to achieve the desired performance. Multicast traffic has specific characteristics owing to the nature of the IP multicast routing protocols. Indeed, the multicast routing is based on multicast IP address and this is why it is very difficult to aggregate multicast traffic since receivers belonging to the same group can be located at multiple localisations. In the IP multicast, the multicast tree structure requires Point-to-MultiPoint (P2MP) LSP or even MultiPoint-to-MultiPoint (MP2MP) LSP establishing.

However, implementation of multicast on MPLS still suffers from some of the IP multicast limitations because the P2MP LSP or MP2MP LSP tree requires storing the forwarding states in each LSR in the path between the source and the receivers. With the overlay multicast, a virtual topology is built above a physical network using the proxies or end hosts. The connection between the proxies or the end hosts is unicast connections. So, the overlay multicast can be implemented on MPLS using the Point-to-Point (P2P) LSPs, i.e., without the need to P2MP LSP or MP2MP LSPs.

This paper proposes the Overlay Multicast Protocol (OMP) (Al-Misbahi and Al-Aama, 2007), which applies an overlay multicast model on MPLS networks. The goal of the protocol is to improve the scalability of multicasting in MPLS networks. The paper also compares OMP performance with PIM-SM and PIM-SSM.

The remainder of this paper is organised as follows: Section 2 presents the related work. Section 3 explains the proposed OMP. The methodology used to evaluate OMP performance is presented in Section 4. The results of the evaluation are discussed in Section 5. And, the conclusion is presented in Section 6.

2 Related work

A framework for IP multicast deployment in an MPLS environment is offered by Ooms et al. (2002). It provides a general overview of the issues arising when MPLS techniques are applied to IP multicast services. An approach described in Farinacci et al. (2000) explains how the label advertisement is piggybacked on multicast routing messages using Protocol Independent Multicast (PIM). Although this approach advertises the labels without the need for additional control messages beyond those needed to support the multicast routing, it suffers from several disadvantages. It is suitable only with sparse mode protocols such as PIM-SM and Core Based Tree (CBT), which have explicit join messages. The dense mode protocols such as Protocol Independent Multicast-Dense Mode (PIM-DM) have no control messages to allow the piggybacking. In addition, this approach suffers from all the limitations of the IP multicast mentioned above.

With regard to the scalability problem, the aggregated multicast is used in Rosen and Aggarwal (2008), which explains the implementation of aggregation on the VPNs that are built using MPLS. The idea of aggregated multicast is that instead of constructing a tree for each individual multicast group, multiple multicast groups can share a single aggregated tree to reduce multicast states. With this scheme, it is more likely that some routers will receive multicast data for which they have no need, thus reducing the optimality of the forwarding trees.

Some protocols reduce the forwarding by reducing the number of routers needed to store the forwarding state. For example, in a protocol called MPLS Multicast Tree (MMT) (Boudani and Cousin, 2002), only routers that act as multicast tree branching node routers for a group need to keep a forwarding state for that group. The reduction obtained from this protocol depends on the spread of the members, i.e., if the members are sparse and spread out, the branching points are few and the reduction is high. So, it may be suitable only for limited applications such as video conferencing.

Minei et al. (2008) describe the setup of P2MP and MP2MP LSPs in MPLS networks. These LSPs are referred to as MultiPoint LSPs (MP LSPs). The solution relies on the Label Distribution Protocol (LDP) without requiring a multicast routing protocol in the network. These MP LSPs are used to apply IP multicast on MPLS networks. Hence, it suffers from all the limitations of IP multicast mentioned earlier.

On the other hand, recently several overlay multicast models were introduced such as ALMI (Pendarakis et al., 2001), Overcast (Jannotti et al., 2000), and OMNI (Banerjee et al., 2003). The overlay multicast has several advantages. First, it does not need support from the network routers, which leads to easier deployment than the IP multicast.

Second, the state information is kept only in the member proxies rather than the core network routers, which improves the scalability in term of the number of the concurrent groups. In addition, since overlay multicast is an application layer, it permits the implementation of high layer services such as security and access control (Pendarakis et al., 2001).

3 Overlay Multicast Protocol (OMP)

As explained earlier, the overlay is a virtual topology constructed above a physical network using a set of proxies. These proxies are connected to the physical network through access links. The connections between the proxies are unicast paths. The clients or the receivers subscribe to the closest proxies.

The following subsections illustrate the operations of OMP as described in Al-Misbahi and Al-Aama (2007).

3.1 Group identification

Each multicast group is identified by a group ID, which consists of owner proxy IP and group number. The first part is the IP address of the proxy where the group was initialised. The second part is a local unique number at the owner proxy.

3.2 Session initialisation

When a source node wants to distribute data to a set of receivers, it must obtain a group ID that identifies the new session from its proxy. Then, it announces the group ID to the receivers through a method such as email or a URL site.

3.3 Joining the group

When a proxy has one or more clients that request to join a multicast group, it sends a *join* message towards the owner proxy. The owner proxy collects the join requests that have reached before the beginning of the session, then, computes the Minimum Spanning Tree (MST), and distributes the routing information to the member proxies using *response* messages.

The *response* message informs each member about its parent and children in the tree. If a new proxy wants to join the group during the session, it sends a *join* message towards the owner proxy. The owner proxy connects that new member to an existing proxy in the current MST and sends the routing information to that member. MST is computed periodically to reflect the frequent modification of the members.

When the member receives the *response* message, it sends a *connect* message to its parent to establish a connection between them. The parent returns a *connect-ack* message to the child.

The computation of MST needs the owner proxy to know the delay between the member proxies. This knowledge is obtained from the members themselves. Each member measures the delay between its node and all the other proxies using *ping* messages. Then, the members send the measurements towards the owner using a *probe* message. This process must be repeated periodically to reflect the change of the paths. With respect to the first computation of MST, each member must add the delay measurements to the join message when it joins the group.

The connections between the proxies are bidirectional as explained in the following section. The owner proxy is the administrator of the group, which means that it is responsible for the tree building and maintenance, but does not mean that it is the unique source of the data. Any member proxy can send the multicast data because MST is a shared tree.

MST is similar to the MP2MP LSP (Minei et al., 2008) in the building such that when the leaf members receive the *response* messages, they establish both a downstream and an upstream LSP; propagate the request towards their parents, which are transit nodes. Transit nodes (which are non-leaf members) support the setup by propagating the downstream and upstream LSP setup towards the root and installing the necessary MPLS forwarding state. The root node installs a forwarding state to map traffic into the MP2MP LSP.

3.4 Leaving the group

When a proxy wants to leave the group, it sends a *leave* message towards the owner proxy. This happens when the proxy has no clients that want to receive the multicast data. But, if this member proxy does not represent a leaf node in the tree, it must continue the forwarding of the multicast data to its neighbour proxies until it stops receiving the *response* messages from the owner proxy for a specified time.

3.5 Tree modification

Owing to the frequent joining and leaving during the session, the tree may have some nodes that are connected but are not members of the group. The tree may also have some nodes that are connected to a non-optimal position in the tree because they were added to the tree after completion of the MST computation. To address this problem, MST is computed periodically. The member who leaves the group must continue to forward data packets to its neighbours until it sees that there are no *response* messages reaching to it. At that point, the member will realise that the owner assigned a new parent to its children. The waiting period must be longer than the *response-interval* taking into account the time needed by the *leave* and the *response* messages to travel on the network. A short *response-interval* increases the tree optimality because it reflects the dynamic changes immediately but it increases the control overhead. So, there is a tradeoff between the tree optimality and the control overhead.

It is obvious that the owner proxy can fail during the session. As found in Pendarakis et al. (2001), multiple back-up nodes of the owner can be used to make the service fault-tolerant. These back-up nodes must contain all the required information to deliver the service to the receivers such that they can be in place of the original owner proxy if it fails. The addresses of the back-up nodes must be known to the members. The *response* messages, which are sent periodically from the owner, allow the members to detect the owner failure.

It is clear that there is much work to be done by the owner proxy for each session. If a proxy is an owner of a large number of sessions, it is preferable to transfer the new requests of establishing multicast sessions to another proxy, which has a light load. This can improve the performance and balance the load especially when there is a high load on the network.

The tree may also be modified owing to a member failure. If that member is not a leaf node, the connectivity of the tree will be affected. To detect the member failure, the messages *connect* and *connect-ack* must be sent periodically. When a child member does not receive the *connect-ack* message for a specific time taking into account the time needed by the messages to travel, it detects that the parent failed. In this case, it must rejoin the group by sending a new *join* message towards the owner proxy. If a parent proxy detects that its child failed, it does not do anything but stop forwarding the data to that child.

In case of a member failure, some of the packets are lost in some member proxies. When a member detects a data loss, and at the same time detects a neighbour failure, it requests the lost data from the sender proxy. In this case, the failing member is the proxy that delivers the data from the sender, i.e. the member who detects the data loss but does not detect a neighbour failure does not request the lost data. This reduces the requests that reach to the sender. After receiving the lost data, the member who sent the request sends the lost data to its neighbours other than the failing one.

4 OMP performance evaluation

This section provides a performance evaluation of OMP through simulation. The performance of OMP is compared with PIM-SM, which uses the piggybacking methodology to assign and distribute labels found in Farinacci et al. (2000). It is also compared with PIM-SSM. Two simulations were performed for each comparison using a C++ built simulator. The simulators take as input a scenario, which is a description of network topology and control parameters. The simulation results provide information about the scalability and other measurements that help to analyse the difference between implementing IP multicast and overlay multicast on MPLS.

The duration of each simulation was 2 h. The sending periods of PIM-SM and PIM-SSM control messages are taken in accordance with the PIM-SM and PIM-SSM specifications (Fenner et al., 2006). The sending periods of OMP control messages were 5 min for ping, probe, and response messages and 60 s for connect messages. Two topologies were used in the simulations. One is a topology of 16 nodes (Figure 2) and the other is a topology of 71 nodes (Figure 3). The first topology is a small grid mesh topology.

Figure 2 The mesh topology

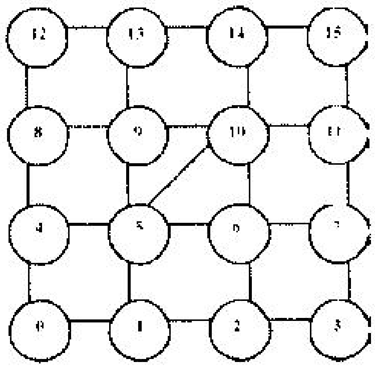
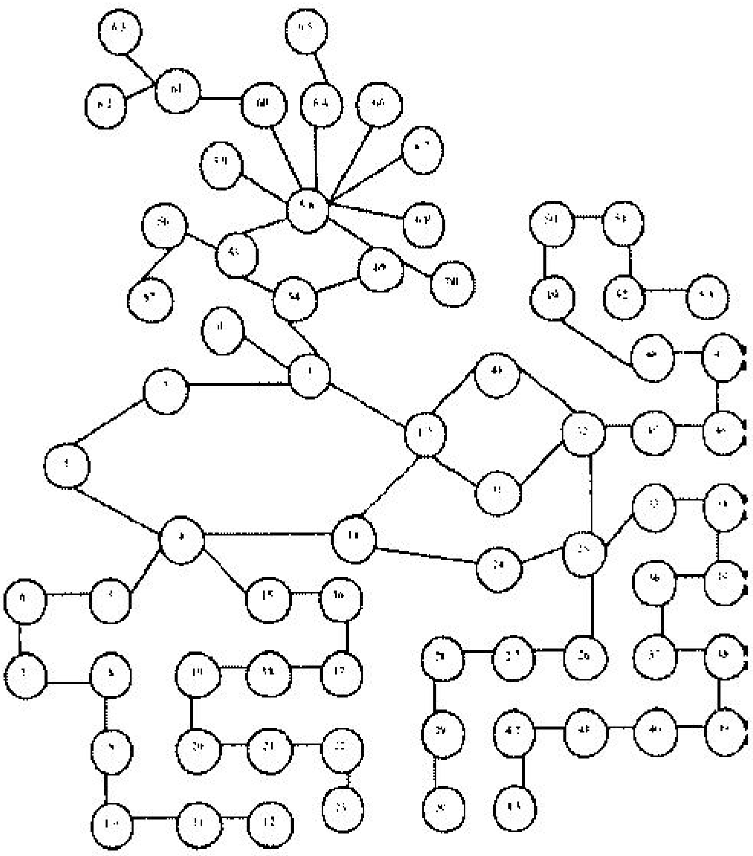


Figure 3 The traceroute topology



The second topology is formed using the traceroute utility. The method is based on a research by Paxson (1996) in which a real topology was constructed using the traceroute utility. This topology was used for simulations by other researchers such as Tian and Neufeld (1998) who proposed a multicast protocol that reduces the forwarding state by the tunnelling approach.

The traceroute was tried out on 18 Saudi university sites. However, in some of these, the traceroute could not be completed and a *Request timed out* message was produced. Therefore, the topology used 14 leaf nodes instead of 18. The produced routes were used to construct the traceroute topology. All the links in the two topologies were bidirectional links with 1 s delays and cost equal to one. Each node was represented in the simulator using a record. Each record consisted of several counters used to count the number of the forwarding states and the number of the join/leave messages.

The two simulations ran the protocols on 1000 concurrent groups each. Four different group sizes were used as follows: 250 groups with 10 members, 250 groups with 30 members, 250 groups with 50 members and 250 groups with 70 members. The owner and the members were selected randomly. The following metrics were used in the simulation:

Average table size of each node: The table size is the number of forwarding states in a node's table. First, the total number of the forwarding states is computed. Then, it is divided by the number of the topology nodes to obtain the average value.

Total control messages for each protocol: This metric presents the total number of the control messages needed to build the multicast trees.

Average delay of the receiver: The delay of sending data to a receiver is measured in terms of the number of physical hops. While each link has a 1 sec delay, the number of hops represents the delay in seconds. To compute this metric, the delay of each receiver in the tree is calculated. Next, the summation of all the receivers' delay of the tree is calculated. And finally, the average delay of the receiver of that tree is calculated. Then, the average is computed in term of all the groups.

Average cost of each tree: The tree cost is the number of links of that tree. First, the cost of each tree is computed. Then, the average is computed by dividing the cost by the number of trees.

Average stress of the tree links: Link stress is the number of identical copies of a packet carried by that link. Using IP multicast, every link in the network has a stress of exactly one and this is the ideal value. Using OMP, there is a chance to carry more than one copy of a packet by a link. The average stress is computed as $\sum_{i \in L} s_i / |L|$ where L represents all the links of a tree, $|L|$ represents the number of the tree links. s_i represents the stress of link i, where i is the link number.

All the metrics take into account only the relation between the proxies in case of OMP and between the designated routers in case of PIM-SM and PIM-SSM. So, the relation with the clients is excluded.

5 Results and discussion

The following sections provide the comparison of performance between OMP and PIM-SM and PIM-SSM.

5.1 *Average table size*

Figures 4 and 5 show the average table size when comparing PIM-SSM and OMP using the mesh topology and the traceroute topology, respectively. In the mesh topology, the average table size using OMP was smaller than when using PIM-SSM. OMP reduced more than half (69%) of PIM-SSM tables, since the average of the difference between the two protocols was nearly 14 entries.

Figure 4 Table size in mesh topology (PIM-SSM vs. OMP) (see online version for colours)

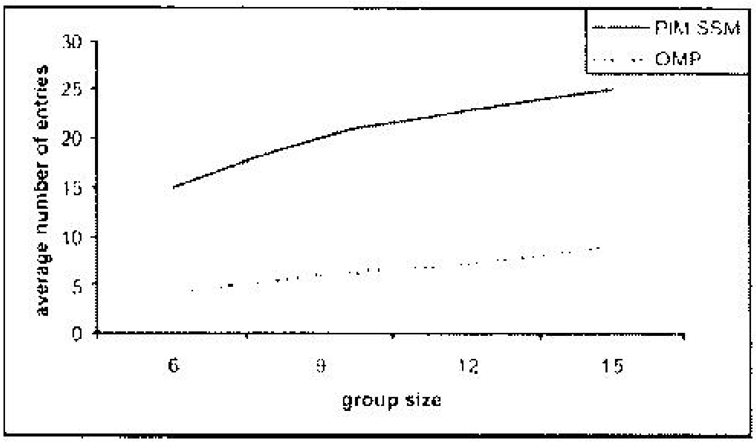
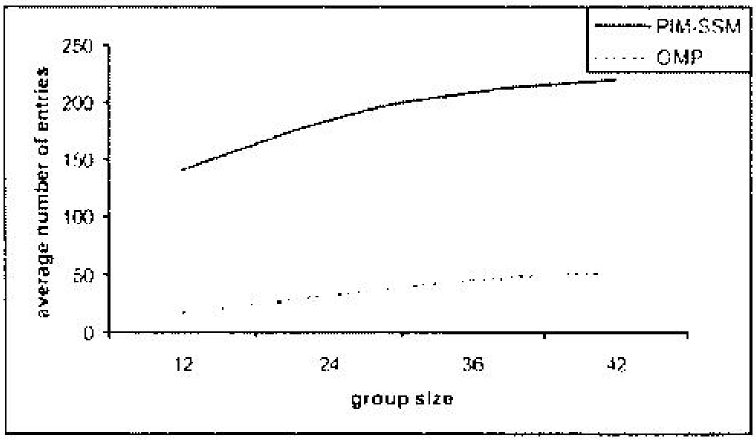


Figure 5 Table size in traceroute topology (PIM-SSM vs. OMP) (see online version for colours)



In the traceroute topology, the average table size using OMP was smaller than when using PIM-SSM. The average of the difference between the two protocols was nearly 152 entries, which means that OMP reduced nearly 81% of PIM-SSM tables.

The average table size when comparing PIM-SM and OMP using the mesh topology and the traceroute topology is shown in Figures 6 and 7, respectively. In the mesh topology, the average table size using OMP was nearly 16 entries smaller than when using PIM-SM, leading to OMP reducing more than half (72%) of PIM-SM tables.

Figure 6 Table size in mesh topology (PIM-SM vs. OMP) (see online version for colours)

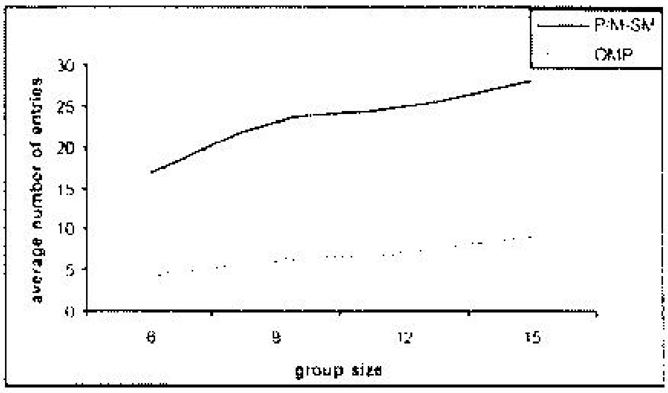
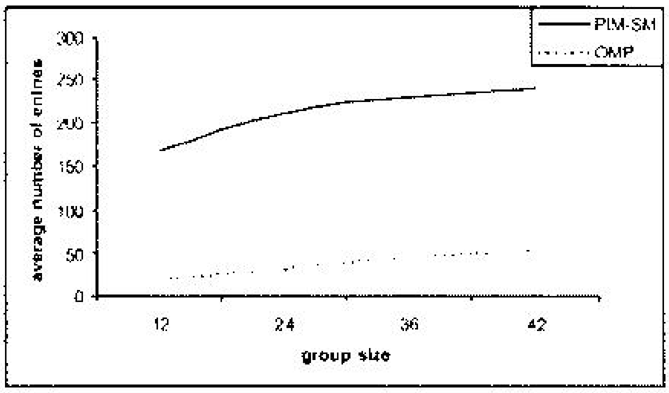


Figure 7 Table size in traceroute topology (PIM-SM vs. OMP) (see online version for colours)



In the traceroute topology, the average table size using OMP was nearly 176 entries smaller than when using PIM-SM. Hence, OMP reduced nearly 83% of PIM-SM tables.

5.2 Total number of control messages

The total number of control messages when comparing PIM-SSM and OMP using the mesh topology and the traceroute topology is shown in Figures 8 and 9, respectively. In the mesh topology, the total number of control messages using OMP was less than when using PIM-SSM. The average of the difference between the two was nearly 29,180 control messages. In the traceroute topology, the total number of control messages using OMP was less than when using PIM-SSM. The average of the difference between them was nearly 687,061 control messages.

Figure 8 Control messages in mesh topology (PIM-SSM vs. OMP) (see online version for colours)

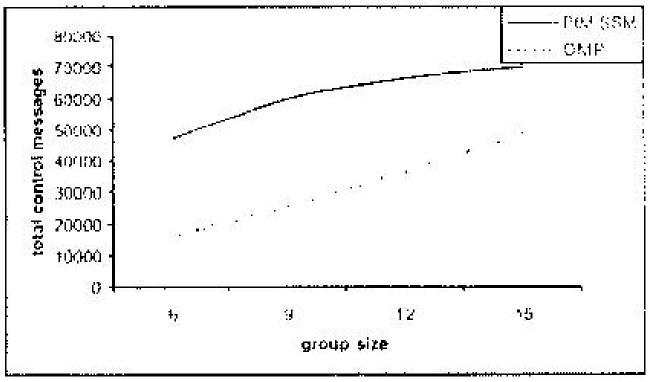
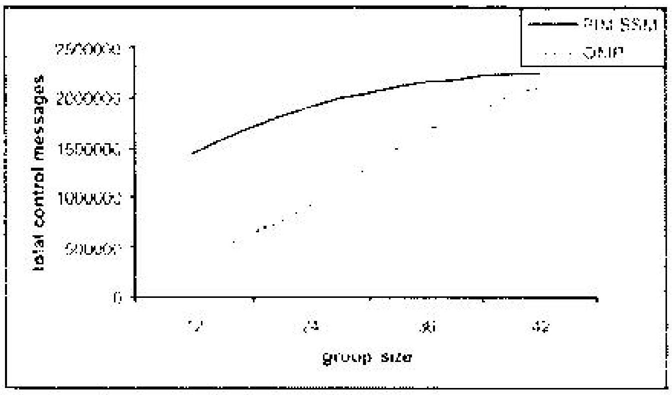


Figure 9 Control messages in traceroute topology (PIM-SSM vs. OMP) (see online version for colours)



The total number of control messages when comparing PIM-SM and OMP using the mesh and the traceroute topology is shown in Figures 10 and 11, respectively. In the mesh topology, the total number of control messages using OMP was less than when using PIM-SM, with an average of the difference nearly 32,600 control messages.

Figure 10 Control messages in mesh topology (PIM-SM vs. OMP) (see online version for colours)

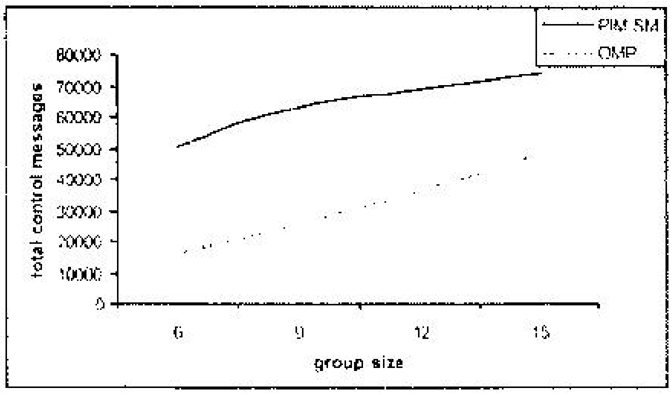
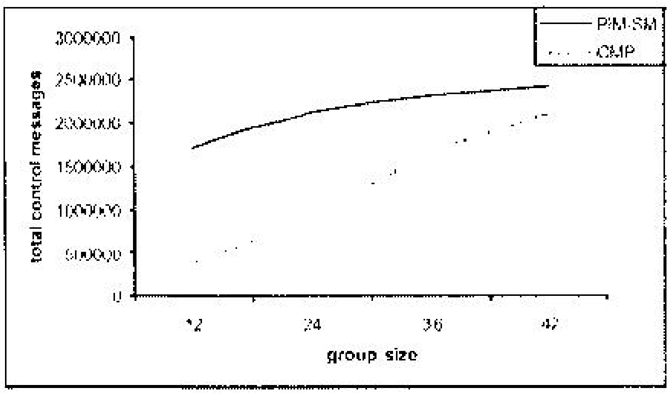


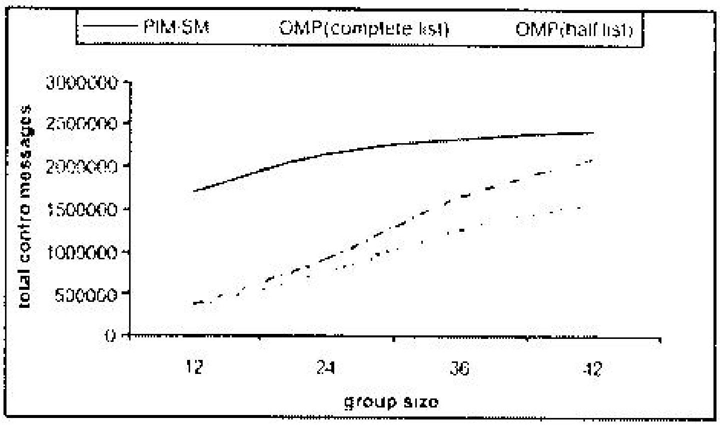
Figure 11 Control messages in traceroute topology (PIM-SM vs. OMP) (see online version for colours)



In the traceroute topology, the total number of control messages using OMP was nearly 892,333 control messages less than when using PIM-SM.

The control overhead of OMP was calculated in two cases in the traceroute topology. In the first case, each member monitored all the other members. In the second case, each member monitored half of the members. The two cases are shown in Figure 12 in comparison with the control messages of PIM-SM. It is clear that the second case provides less control overhead. The average of the difference between the control messages of the two cases of OMP was 286,500 control messages. The average of the difference between the control messages of PIM-SM and OMP with monitor list of half of the members was 1,178,833 messages.

Figure 12 Control messages when changing the monitor list (PIM-SM vs. OMP) (see online version for colours)



In addition, Figure 13 shows the total number of control messages when OMP used a complete monitor list and used ping interval equals to 10 min. The average of the difference between the control messages in case of 10 min ping intervals and in case of 5 min ping interval was 315,000 messages such that the monitor list is complete. The average of the difference between the control messages of PIM-SM and OMP with 10 min ping interval was 1,207,333 messages. Also, Figure 14 shows the total number of control messages when OMP used half the monitor list and used ping intervals equal to 10 min. The average of the difference between the control messages in case of a complete monitor list and 5 min ping intervals and in case of half monitor list and 10 min ping intervals was 458,250 messages. The average of the difference between the control messages of PIM-SM and OMP with 10 min ping interval and half the monitor list was 1,350,583 messages.

Figure 13 Control messages when changing the ping interval of OMP (PIM-SM vs. OMP) (see online version for colours)

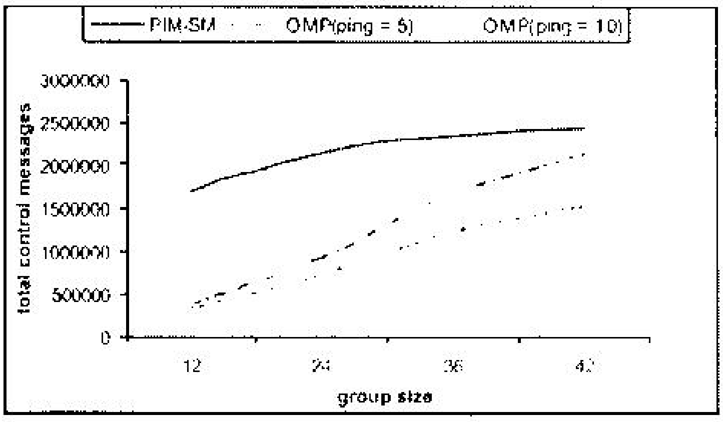
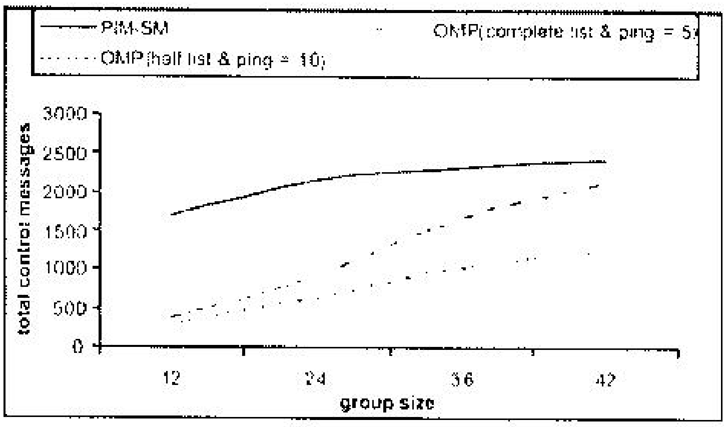


Figure 14 Control messages when changing the monitor list and ping interval of OMP (PIM-SM vs. OMP) (see online version for colours)



5.3 *Average delay for the receiver*

In the mesh topology, the average delay for the receiver using PIM-SSM was less than when using OMP, as shown in Figure 15. The average of the difference between the two protocols was nearly 1.5 hops.

In the traceroute topology, the average delay for the receiver using PIM-SSM was less than when using OMP with nearly one hop, as shown in Figure 16.

Figure 15 Delay for receiver in mesh topology (PIM-SSM vs. OMP) (see online version for colours)

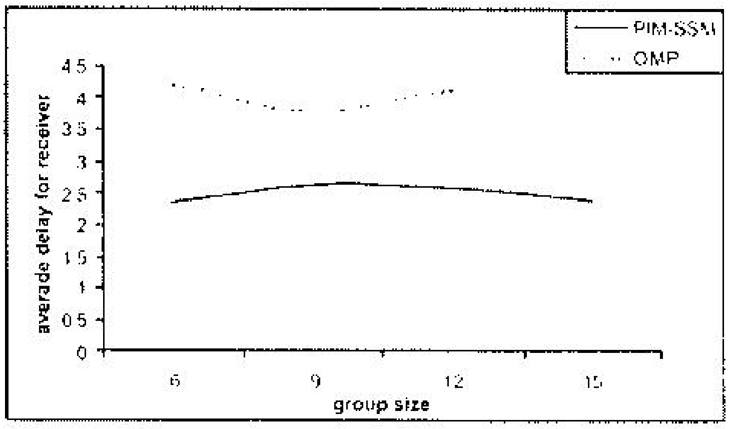
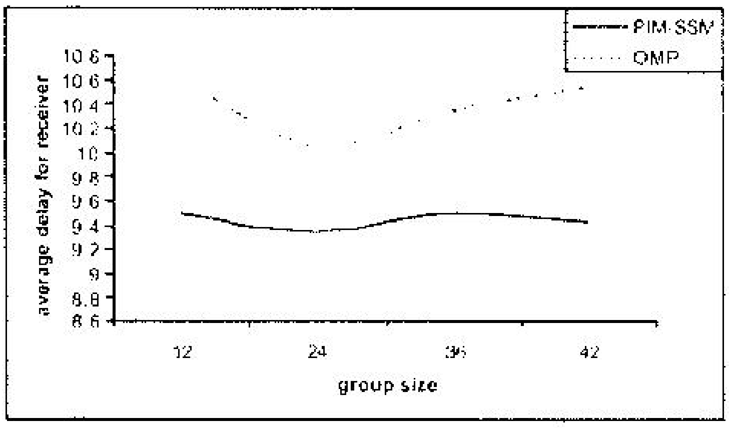


Figure 16 Delay for receiver in traceroute topology (PIM-SSM vs. OMP) (see online version for colours)



The average delay for the receiver, when comparing PIM-SM and OMP, using the mesh topology and the traceroute topology, is shown in Figures 17 and 18, respectively.

Figure 17 Delay for receiver in mesh topology (PIM-SM vs. OMP) (see online version for colours)

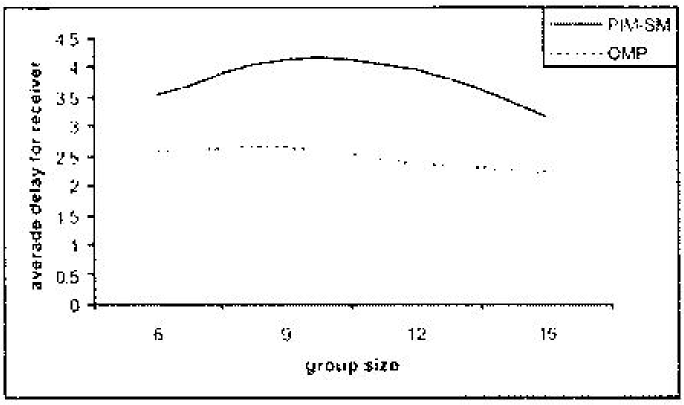
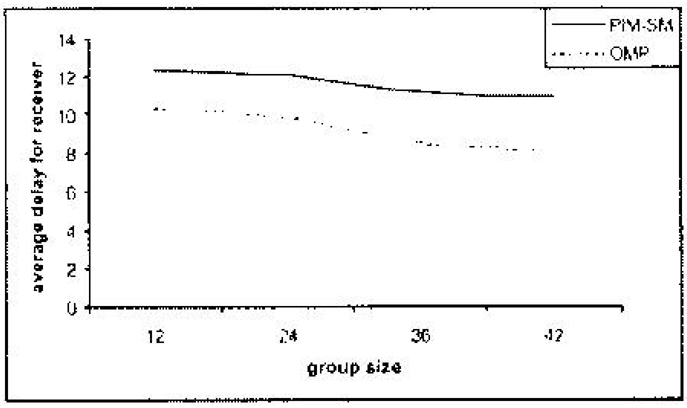


Figure 18 Delay for receiver in traceroute topology (PIM-SM vs. OMP) (see online version for colours)



In the mesh topology, the average delay for the receiver using OMP was less than when using PIM-SM, where the difference between the two protocols was nearly one hop. And, in the traceroute topology, the average delay for the receiver using OMP was less than when using PIM-SM. The average difference was nearly two hops.

5.4 *Average cost of the tree*

In the mesh topology, the average cost of the tree using OMP was less than when using PIM-SSM. The average of the difference between the two protocols was nearly 4 links, as depicted in Figure 19. In the traceroute topology, however, the average of the difference between the two protocols was nearly 15 links, as shown in Figure 20, resulting in the cost of the tree using OMP being less than when using PIM-SSM.

The average cost of the tree when comparing PIM-SM and OMP using the mesh topology and the traceroute topology is shown in Figures 21 and 22, respectively. In the mesh topology, the average cost of the tree using OMP was less than when using PIM-SM. The average of the difference between the two protocols was nearly seven links. In the traceroute topology, the average cost of the tree using OMP was less than when using PIM-SM, with an average difference of nearly 25 links.

Figure 19 Cost of the tree in mesh topology (PIM-SSM vs. OMP) (see online version for colours)

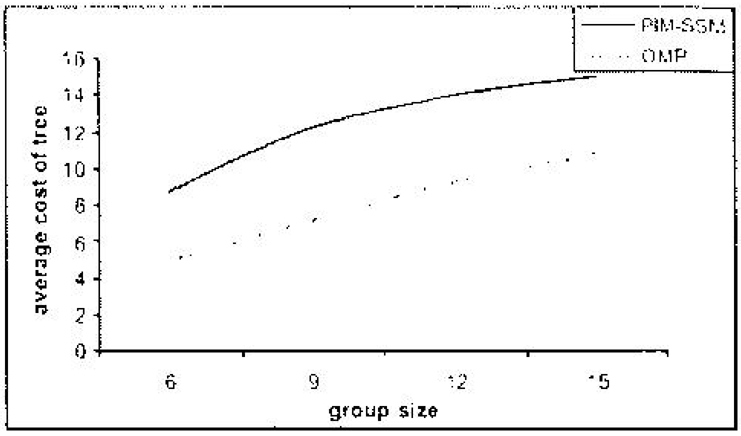


Figure 20 Cost of the tree in traceroute topology (PIM-SSM vs. OMP) (see online version for colours)

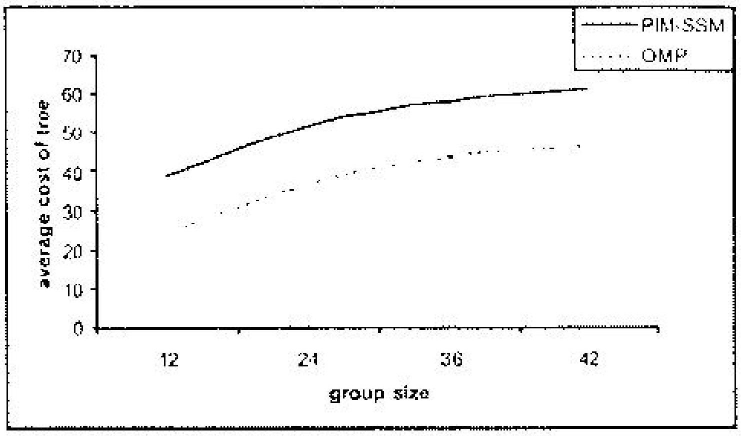


Figure 21 Cost of the tree in mesh topology (PIM-SM vs. OMP) (see online version for colours)

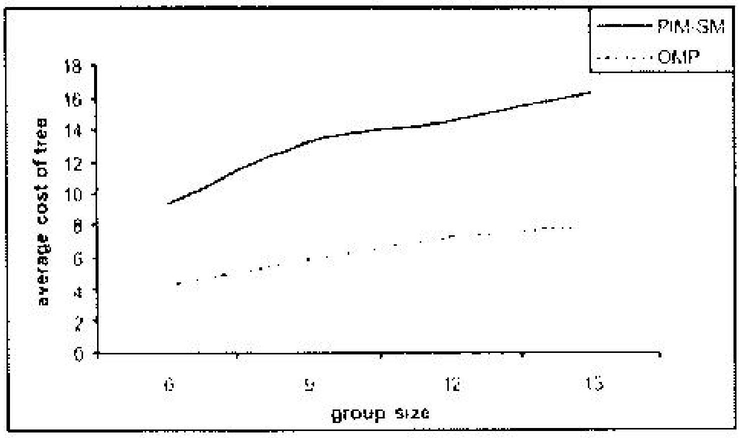
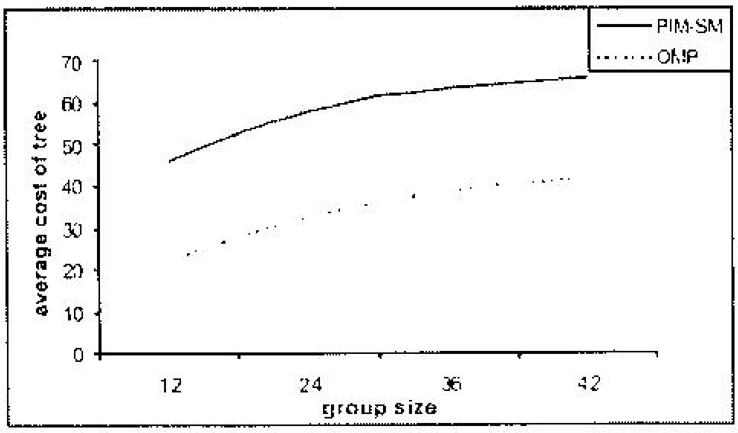


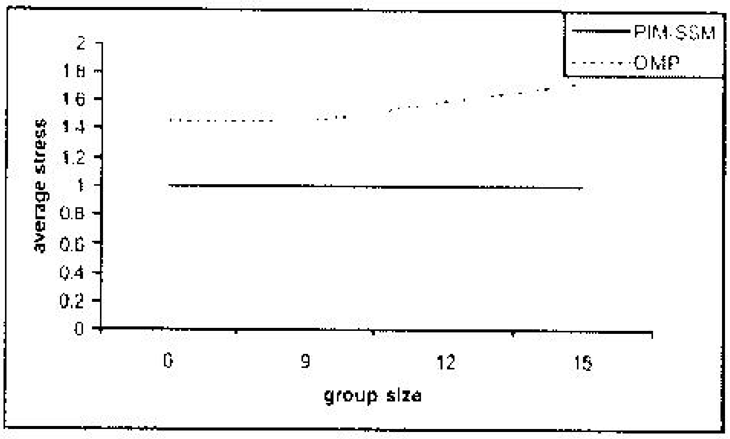
Figure 22 Cost of the tree in traceroute topology (PIM-SM vs. OMP) (see online version for colours)



5.5 *Average stress of the link*

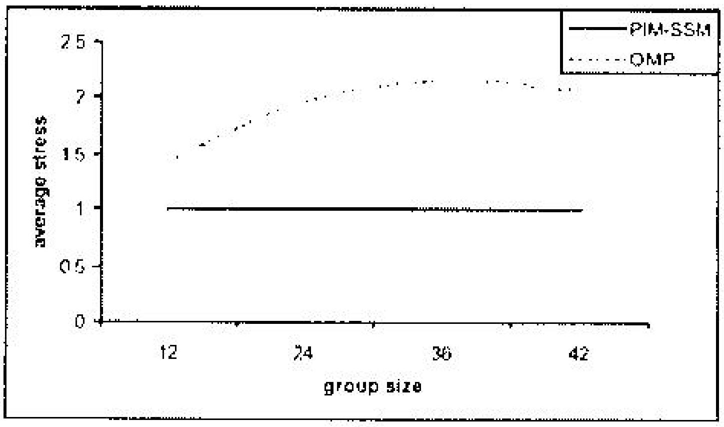
In the mesh topology, the average stress of the link using PIM-SSM was less than when using OMP, as shown in Figure 23. The average number of the identical copies of the same packet that can be carried by the link using OMP was nearly 1.55 packets while it was exactly one packet using PIM-SSM.

Figure 23 Stress of the link in mesh topology (PIM-SSM vs. OMP) (see online version for colours)



In the traceroute topology, as shown in Figure 24, the average stress of the link using PIM-SSM was less than when using OMP. The average number of the identical copies of the same packet that can be carried by the link using OMP was nearly 1.89 packets.

Figure 24 Stress of the link in traceroute topology (PIM-SSM vs. OMP) (see online version for colours)



The average stress of the link, when comparing PIM-SM and OMP, using the mesh topology and the traceroute topology, is shown in Figures 25 and 26, respectively. In the mesh topology, the average stress of the link using PIM-SM was less than when using OMP. The average number of the identical copies of the same packet that can be carried by the link using OMP was nearly 1.27 packets while it was exactly one packet using PIM-SM. In the traceroute topology, the average stress of the link using PIM-SM was less than when using OMP. The average number of the identical copies of the same packet that can be carried by the link using OMP was nearly 1.68 packets.

Figure 25 Stress of the link in mesh topology (PIM-SM vs. OMP) (see online version for colours)

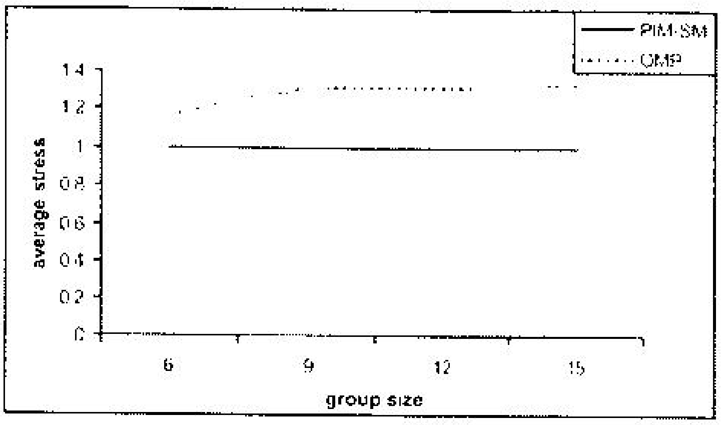
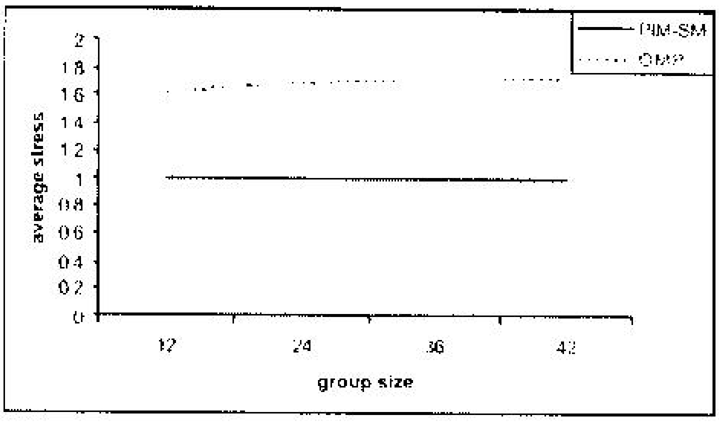


Figure 26 Stress of the link in traceroute topology (PIM-SM vs. OMP) (see online version for colours)



5.6 *Summary*

Table 2 summarises the findings when comparing OMP with PIM-SSM.

Table 2 Comparison of OMP and PIM-SSM

Metrics	*OMP*	*PIM-SSM*
Average table size	Lower	Higher
Total number of control messages	Lower	Higher
Average delay for the receiver	Higher	Lower
Average cost of the tree	Lower	Higher
Average stress of the link	Higher	Lower

And, Table 3 shows the findings when comparing OMP with PIM-SM.

Table 3 Comparison of OMP and PIM-SM

Metrics	*OMP*	*PIM-SM*
Average table size	Lower	Higher
Total number of control messages	Lower	Higher
Average delay for the receiver	Lower	Higher
Average cost of the tree	Lower	Higher
Average stress of the link	Higher	Lower

6 Conclusion

This paper described the OMP protocol, which applies the overlay service in MPLS networks. It is clear that OMP provides a scalable solution for multi-sender multicast communication. The general operations of OMP were explained. The simulation results showed improvement in performance when using OMP. When comparing OMP with PIM-SSM, OMP provides better performance than PIM-SSM in terms of the average table size, the total number of control messages, and the average tree cost. When comparing OMP with PIM-SM, OMP provides better performance in terms of the average table size, the total control messages, the average delay for receiver and the average tree cost.

The scalability degree of the protocol depended mainly on average table size. OMP outperforms PIM-SSM and PIM-SM in this metric, and the difference magnified as the group size was increased. The large difference in the average table sizes was due to that OMP stores the forwarding states only in the member proxy while PIM-SSM and PIM-SM stores the forwarding states in all the routers in the paths between the source and the receivers.

With respect to the total number of control messages, OMP achieved less control overhead but the overhead increased with the group size increase in case of the complete monitoring lists. The use of monitoring lists that included a subset of the members decreased the control overhead especially with the continuous increase in the group size.

PIM-SSM provided less delay by nearly one hop because it builds trees with shortest paths while OMP builds MSTs. PIM-SM builds shared trees with shortest paths between RP and receivers. Although it was expected that this would lead to less delays in PIM-SM, it actually did not. This was because in PIM-SM the paths must go through the RP, which is the core of the tree, causing more delays in PIM-SM than OMP.

The tree type built for each protocol affected the average tree cost. The MST, used in OMP, focuses on building trees of less costly links, which resulted in less tree cost in OMP.

With regard to stress, PIM-SSM and PIM-SM provided less stress than OMP. This is, however, a problem that is common among all overlay protocols and is not specific to OMP. The problem is caused by the fact that when a proxy follows a unicast path to forward packets to other proxies, it may receive and send data over the same link, causing duplicate packets on links close to the proxy. However, the increase in the stress value in OMP was relatively low and reasonably acceptable especially when focusing on the achieved benefits and the several limitations it solves that are found in IP multicasting such as the difficulty of deployment and network management.

These results show that OMP is very promising especially with the increasing demand to deliver multicast services globally. Its value is especially important owing to its support for scalability.

References

Almeroth, K.C. (2000) 'The evolution of multicast: from the MBone to interdomain multicast to Internet2 deployment', *IEEE Network*, Vol. 14, No. 1, January–February, pp.10–20.

Al-Misbahi, H. and Al-Aama, A. (2007) 'The Overlay Multicast Protocol (OMP): a proposed solution to improving scalability of multicasting in MPLS networks', *The IEEE 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007)*, August, pp.79–85.

Banerjee, S., Kommareddy, C., Kar, K., Bhattacharjee, B and Khuller, S. (2003) 'Construction of an efficient overlay multicast infrastructure for real-time applications', *Proceedings of IEEE INFOCOM*, Vol. 2, March–April, pp.1521–1531.

Boudani, A. and Cousin, B. (2002) 'A new approach to construct multicast trees in MPLS networks', *Seventh IEEE Symposium on Computers and Communications*, July, pp.913–919.

Farinacci, D., Rekhter, Y. and Rosen, E. (2000) *Using PIM to Distribute MPLS Labels for Multicast Routes*, IETF Internet Draft, November, Work in Progress, USA.

Fenner, B., Handley, M., Holbrook, H. and Kouvelas, I. (2006) 'Protocol Independent Multicast – Sparse Mode (PIM-SM): protocol specification', *IETF RFC4601*, August, RFC Editor, USA.

Jannotti, J., Gifford, D., Johnson, K., Kaashoek, M. and O'Toole Jr., J. (2000) 'Overcast: reliable multicasting with an overlay network', *Proceedings of USENIX OSDI*, October, pp.197–212.

Minei, I., Kompella, K., Wijnands, I. and Thomas, B. (2008) *Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint label Switched Paths*, IETF Internet Draft, Work in Progress, November, USA.

Ooms, D., Sales, B., Livens, W., Acharya, A., Griffoul, F. and Ansari, F. (2002) 'Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) environment', *IETF RFC3353*, August.

Paxson, V. (1996) 'End-to-end routing behavior in the internet', *ACM SIGCOMM*, Vol. 26, No. 4, October, pp.25–38.

Pendarakis, D., Shi, S., Verma, D. and Waldvogel, M. (2001) 'ALMI: an application level multicast infrastructure', *Proceedings of USENIX USITS*, March, pp.49–60.

Pointurier, Y. (2002) *Link Failure Recovery for MPLS Networks with Multicasting*, Master Thesis, University of Virginia, August, Charlottesville, VA, USA.

Rosen, E. and Aggarwal, R. (2008) 'Multicast in MPLS/BGP IP VPNs', *IETF RFC2547*, July, RFC Editor, USA.

Rosen, E., Viswanathan, A. and Callon, R. (2001) 'Multiprotocol label switching architecture', *IETF RFC3031*, January, RFC Editor, USA.

Tian, J. and Neufeld, G. (1998) 'Forwarding state reduction for sparse mode multicast communication', *INFOCOM '98 Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*, San Francisco, CA, USA, March, Proceedings, Vol. 2, pp.711–719.

Zhu, Y., Shu, W. and Wu, M. (2005) 'Approaches to establishing multicast overlays', *Proceedings of IEEE International Conference on Services Computing*, Vol. 2, July, pp.268–269.