Comparative Study of Multicast Protection Algorithms Using Shared Links in 100GET Transport Network

Samer Sulaiman, Abdelfattah Haidine, Ralf Lehnert, and Stefan Tuerk

TU Dresden, Chair for Telecommunications
{Sulaiman,lehnert,tuerk}@ifn.et.tu-dresden.de,
Abdelfatteh.haidine@signalion.com

Abstract. In recent years new challenges have emerged in the telecommunications market resulting from the increase of network traffic and strong competition. Because of that, service providers feel constrained to replace expensive and complex IP-routers with a cheap and simple solution which guarantees the requested quality of services (QoS) with low cost. One of these solutions is to use the Ethernet technology as a switching layer, which results in using the cheap Ethernet services (E-Line, E-LAN and E-Tree) and to replace the expensive IP-routers. To achieve this migration step, new algorithms that support the available as well as the future services have to be developed. In this paper, we investigate the multicast protection issue. Three multicast protection algorithms based on the shared capacity between primary and backup solutions are proposed and evaluated. The blocking probability is used to evaluate the performance of the proposed algorithms. The sub-path algorithm resulted in a low blocking probability compared with the other algorithms.

1 Introduction

In recent years new services like IPTV, Video on Demand, Distance Learning, etc. appeared making the network traffic growths faster. Some of these services need a high bandwidth when they are unicasted to each customer. In this case, multicast technology can reduce the required bandwidth through distributing the traffic over a multicast tree rooted by the source. Basically, IP-multicast uses the UDP protocol to forward the multicast data. Because of that and because the dynamic behavior of multicast groups, it is difficult to avoid packet loss and to keep the multicast distribution tree optimum. Multicast routing protocols can be classified into two classes. On one hand, protocols use own routing information to build the distribution tree (e.g. Distance Vector Multicast Routing Protocol "DVMRP" and Multicast Open Shortest Path First "MOSPF") [RFC1075]. On the other hand, protocols use the existing unicast routing information to build the distribution tree (e.g. Protocol Independent Multicast "PIM" and Core Based Tree "CBT") [Wil02] [RFC2201] [RFC4601].

Due to the increasing cost pressure in the telecommunication market and the slump in the telecommunication services, service providers feel constrained to find a new solution which guarantees the request quality of services with low cost. Carrier-Grade Ethernet solution is proposed to replace the expensive and complex IP-router with a cheap and simple Ethernet switch. However, this replacement has to fulfill the

M. Oliver and S. Sallent (Eds.): EUNICE 2009, LNCS 5733, pp. 158–167, 2009.

existing QoS and to increase the network capacity. Many investigations are done to improve the capacity of the Ethernet switch. Additionally, several approaches as well protocols are proposed to realize this migration step [AH08] [FED07] [WB08].

In view of using the Carrier Ethernet for increasing the network capacity, new challenges arise, like: 1) keeping the new technique as simple and cheap as possible comparing to the available ones, 2) developing protocols working with existing ones in different layers (IP/Eth/WDM, IP/WDM, Eth/WDM, Eth/SDH/WDM, etc.), 3) supporting the available services (IP and Ethernet, Point-to-Point "P2P" and Point-to-Multipoint "P2MP"), 4) scalability, etc. P2MP or multicast services are used to transport the same data to a group of customers simultaneously through a so-called distribution tree. Therefore, the available QoS and resilience algorithms used in unicast are not suitable for multicast. By using the unicast algorithms for multicast, the distribution tree used for forwarding the multicast data has to be subdivided into unicast paths for each receiver of the multicast group. That results in multiplying the required capacity of the shared links in the distribution tree.

Our contribution focuses on the comparison of three protection algorithms proposed to solve the multicast protection issue. Different calculation scenarios are implemented by MATLAB to evaluate the performance of the investigated algorithms.

The rest of the paper is organized as follows. An overview on the multicast protection requirements is given in section II. In section III, three protection algorithms are described. Results and comparisons between the investigated algorithms are shown in section IV.

2 Requirements for Multicast Resilience

Let us consider the use of unicast protection algorithms for the multicast case. In this case, each path of the distribution tree has to be protected with a unicast backup path. However, it is difficult to realize this idea because the multicast address is used to identify a group of unicast addresses. This results also in duplicating the packet processing and sending. Figure 1 presents an example network with 8 nodes and 15 links.

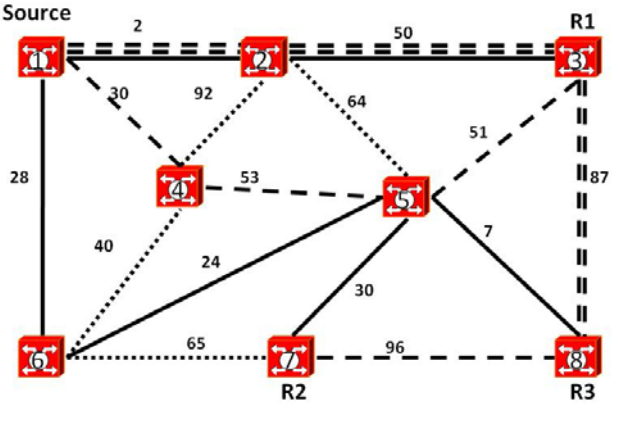

Fig. 1. Multicast resilience using unicast protection algorithm

The solid paths {(1,2,3); (1,6,5,7); (1,6,5,8)} represent the active multicast distribution tree based on source based tree algorithm, where node 1 is the source and node 3, 7 and 8 are the multicast receivers. The dashed paths {(1,4,5,3); (1,2,3,8); (1,2,3,8,7)} stand for the unicast backup paths found for each path in the distribution tree. The number above each link represents the link weight. Now we assume that a failure occurs over the link between node 2 and node 3, the unicast backup path (1,4,5,3) will be used. In this case, node 5 receives the multicast packet twice (from node 2 and node 4). The node has to process both duplicates and to send them further. Furthermore, multiple bandwidth is reserved on the links {(1,2); (2,3); (3,8)} to realize this protection process. Because of that, it is necessary to apply a multicast protection algorithm resulting in protecting the whole distribution tree and not to protect each path of this tree separately; such is the case in unicast.

3 Investigated Multicast Resilience Algorithms

The simple solution to guarantee the arrival of packets to each multicast group member is to build two separate distribution trees like using two paths in the unicast case. This solution is simple and guarantees a fast rerouting. However, it is not efficient and results in increasing the required bandwidth. Different tree protection algorithms are proposed assuming that all the network nodes are member in the tree [MBG99] [XLT03]. However, in the multicast case, the distribution tree consists of only some network nodes. Additionally, the structure of this tree can be changed dynamically according to the dynamical behavior of the multicast group. Therefore, these algorithms have to be adjusted to protect the distribution tree. Three improved algorithms based on building two distribution trees, sharing some links, will be discussed in this section. The backup tree can be activated as soon as a failure occurs in the primary tree. On the other hand, the backup tree can be used to reroute some paths or links of the primary tree. The resilience algorithms described in this section are: a) preplanned tree based, b) sub-path based and c) dual forest tree.

3.1 Preplanned Tree Based Protection

The main idea of the preplanned tree based algorithm is to find the shortest path tree from the red/blue tree constructed by the MEBG algorithm developed by Médard et al. [MBG99] [XLT03]. Let us assume that all the network nodes are member in the multicast tree. There are different algorithms used to build the distribution tree. We will here present the MEBG algorithm which guarantees fast recovery from any single link/node failure as long as the failed node is not the source node. The basic idea of this algorithm is to construct two redundant trees called blue tree and red tree. Figure 2 shows an undirected graph with 8 nodes (bridges) and 15 links. The source node is node 1. At the beginning both trees (T^B for blue tree and T^R for red tree) contain only the source node. Then we try to find a ring consisting of at least 3 nodes in which the source node is the start as well the end of this ring. Different criteria can be specified for selecting the ring depending on the design objective, such as minimizing average delay or reducing the total cost.

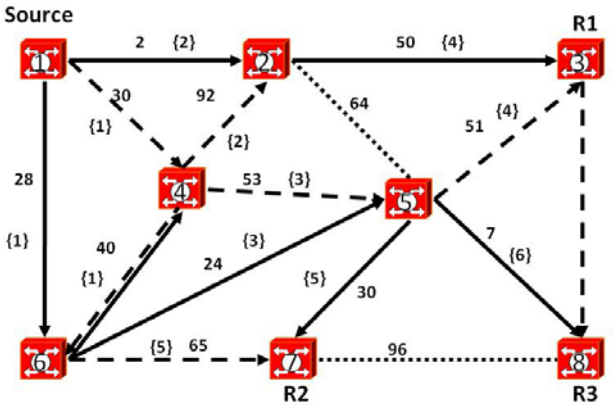

Fig. 2. Protection scheme based on preplanned tree

In this example, minimizing the average delay is used as objective to build the tree. Thus, the first found ring from Figure 2 is (1,6,4,1). All the links on this ring selected during the first iteration are labeled with label {1}. From this ring we can define two paths starting at the source node. According to the objective function the path (1,6,4), referred as solid links, has a lower cost than the path (1,4,6) with dashed links. Thus, it is added to T^B and the path (1,4,6) is added to T^R. Now we look for a new path connecting two distinct nodes (e.g. node 1 and 4) in T^B and at least one additional node not in T^B (e.g. node 2). In the same way, the link (1,2) is added to T^B and the link (4,2) to T^R. Because all links of this path are selected during the second iteration, they are labeled with {2}. The algorithm will continue until T^B and T^R span all the network nodes. In this case, two trees are constructed for the primary tree T^B and backup tree T^R. Several algorithms have been proposed to protect either each link or each path of the primary tree using the backup tree [XLT03].

As mentioned above, MEBG protects all the network nodes. This results in an unnecessary bandwidth reservation in some nodes. Since only the distribution tree nodes of a multicast group have to be protected, we improve the MEBG algorithm to build two distribution trees from T^B and T^R. The shortest path tree algorithm has been used to construct the primary as well as backup distribution tree.

3.2 Sub-path Based Protection

In this section, we propose a new protection algorithm. The basic idea of the sub-path algorithm is to divide the primary tree into sub-paths, and to find a backup path for each part. To understand the division of the primary tree, a set of protection nodes has to be defined first. It consists of all receivers and switching nodes of the primary tree. A switching node is a node which has more than one downstream in the distribution tree. Now we can define a set of sub-paths between sender and receiver, sender and switching point, receiver and receiver and switching point and receiver. To make this algorithm easy to understand, we explain it using an example. Figure 3 presents a network of 8 nodes and 13 links. Let us assume a multicast group consisting of node 1 as a sender and the nodes 2, 3, 7 and 8 as receivers. Firstly, the primary distribution

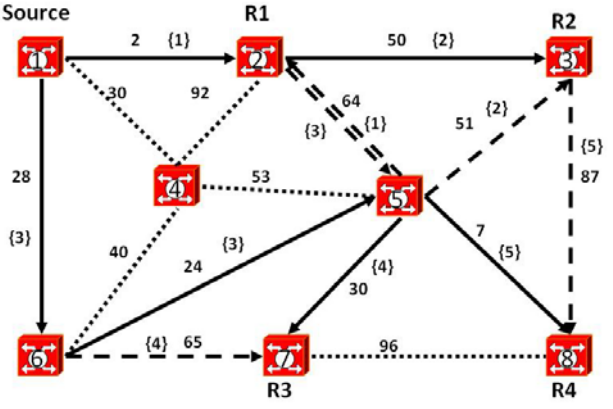

Fig. 3. Sub-path based protection

tree will be constructed as a shortest path tree (solid lines). The set of the Protection Nodes (PN) consists of the nodes 2, 3, 7, 8 and 5, where node 5 is a switching node. From this set we can define the following sub-paths: SP={(1,2); (2,3); (1,6,5); (5,7); (5,8)}. Each sub-path has to be composed of only two nodes from the set PN.

The next step is to find a backup path of each one in the set SP. To do that, we delete the sub-path from the network topology. Then, we try to find the shortest path between the start and the end node of the deleted sub-path. The link between node 1 and 2 will be firstly deleted. The shortest path from node 1 to node 2 is then the path (1,6,5,2). The shared links between the primary tree and the found backup path (1,6,5) will be deleted. This results in avoiding the duplication of the required reserved bandwidth. This sub-path and the rest of the found backup path will be labeled with {1}. This label is used to guarantee a fast recovery process from any single link/node failure as long as the failed node is not the source or a switching node. This is because the links with the same label will be immediately reactivated when information about a failure is received. The algorithm continues until all the sub-paths of the set SP are protected and labeled. In this case, we do not get a separate backup tree (dashed lines). If a link or a node of the primary tree fails, the algorithm will activate the links of the same label from the backup set.

3.3 Dual Forest Tree Algorithm

The dual forest tree algorithm starts with finding the primary tree as a shortest path tree. After that it continues finding the shortest path between the leaf nodes of this tree. A leaf node can be each node of the primary tree, which has only one connection. Figure 4 shows the primary (solid lines) as well the backup links (dashed lines) for a multicast group consisting of the node 1 as a source and the nodes 5, 7 and 8 as receivers. Because the source has only one connection in this example, it will be selected as a leaf node, too. From Figure 4, the Leaf Nodes set (LN) is composed of the nodes 1, 7 and 8. To find the shortest path between the leaf nodes, the links and the nodes of the primary tree except the leaf nodes are deleted. The shortest path between

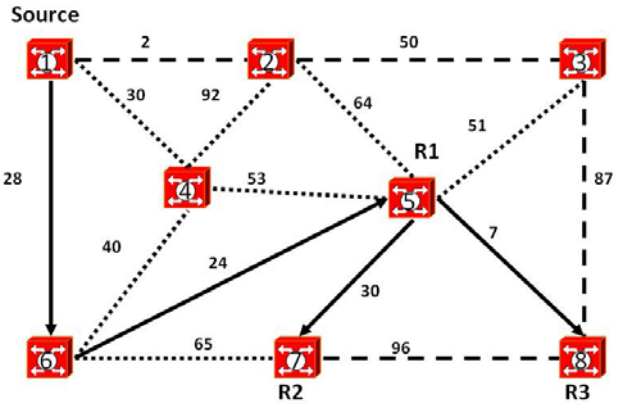

Fig. 4. Dual forest tree protection

the leaf nodes is (1,2,3,8,7). The main challenge of this algorithm is to find the shortest path between the leaf nodes without sharing any link and node of the primary tree. To solve this issue, we choose each leaf node as a root and try to find its shortest path tree. The tree with lowest cost will be then selected as a backup tree. The basic idea of this algorithm has been proposed in [FCG01] [SCM06].

4 Performance Evaluation

4.1 Reference Networks

To evaluate and compare the investigated algorithms, they are implemented in Matlab [MW08]. Primary tree cost, backup cost and blocking probability are used for comparing the performance of these algorithms. Two network topologies from the NOBEL-project [NOB08] are used for the performance evaluation. The first one is the *"German network"* with 17 nodes and 26 links and the second one is the *"European network"* with 28 nodes and 41 links.´

4.2 Evaluation Scenarios

Three scenarios are used in this work. The results of these scenarios are averaged over 100 calculation runs. The first one uses the proposed protection algorithms to find the primary tree as well the backup set for a multicast group whose size falls within the range [3, N-2], where N is the number of the network nodes. The members of these groups are randomly chosen. The first node of each group is selected as a source of this group. Because of that, a group of two members represents a unicast case. Therefore, we use at least three nodes to build a multicast group. Furthermore, a group of N members represents a broadcast case. The range [3, N-2] is chosen to evaluate the performance of the proposed algorithms in different group density (sparse and dense mode). The results of this scenario are defined as a function of multicast group size. The second scenario is similar to the first one. However, a

defined number of multicast groups are randomly created for each calculation run. In this case, the link capacity will play a major role. The link capacity is 10 capacity units; however, the demand of each multicast service is one capacity unit. The results of the last scenario are defined as a function of the number of the multicast groups for each calculation run. The number of multicast groups falls within the range [1, 15]. In this scenario, each protection algorithm is used to find the primary tree and the backup set for a defined multicast group size with different number of multicast groups (within the range [1, 15]) for each calculation run. Therefore, the link capacity plays also a major role in this scenario. We have chosen this range [1, 15] to investigate the proposed algorithms with a low and a high network load.

4.3 Results Discussion

To evaluate the performance of the proposed algorithms, the blocking probability is used. The blocking probability is the number of multicast groups whose backup set cannot be found divided by the total number of created multicast groups. This criterion shows clearly the performance of the proposed algorithms. A 95% confidence interval is used to show the accuracy of the averaged value. The primary as well as backup tree cost will be discussed in this paper, too. However, their results are not presented because of the paper limit.

4.3.1 German Network with 17 Nodes and 26 Links

The results presented in Figure 5 are defined as a function of the multicast group size, while the results in Figure 6 are defined as a function of the multicast groups number. Because of using the shortest path algorithm to find the primary tree, the cost of this tree is a suboptimum compared to the minimum spanning tree. However, we get the optimum solution of the end-to-end cost in this tree. Therefore, the end-to-end cost of the primary tree constructed by the sub-path and dual forest tree algorithms is the optimum. However, the backup cost depends on the method used by the algorithms. The dual forest tree algorithm finds the shortest path tree between the leaf nodes of a primary tree. Thus, its backup cost is the lowest. On the other hand, the sub-path algorithm tries to find a backup set of defined sub-paths from a primary tree. The shared links between the primary tree and the backup set will be deleted to avoid duplicating the reserved link capacity in the same direction. This results in reducing the backup cost compared to the preplanned tree algorithm. Figure 5 shows that the dual forest tree algorithm is the worst one and the sub-path algorithm is the best one in finding the backup tree. The preplanned tree algorithm tries to find two separate trees. In the case of a large number of multicast groups and high network load, it becomes difficult to find two separate trees. However, the sub-path algorithm tries to find a set of backup paths sharing with the primary tree. This results in a network load reduction. From Figure 5(a) we can see that the blocking probability of the dual forest tree algorithm for multicast group size 14 and 15 is one. This is because the algorithm cannot find any backup set for all the investigated groups. We can infer that the sub-path algorithm is the best one in both small as well as large multicast group sizes (see Figure 6-(a) and (b)). While the dual forest tree algorithm shows the worst result in both small as well as large multicast group sizes where in large multicast group sizes, the blocking probability is almost one. This is because this algorithm tries to find the

shortest path between the leaf nodes without sharing the primary links and nodes. Hence, this algorithm is not applicable for large group sizes. From Figure 6 we can see that the preplanned tree algorithm performs similarly to the sub-path algorithm for large group sizes (right diagram), while the difference becomes larger in small group sizes (left diagram). We can explain this difference by the fact that the sub-path algorithm needs less backup capacity than the preplanned tree algorithm. As mentioned above, the dual forest tree cannot find any backup tree. Furthermore, its blocking probability differs between zero and one in small multicast group size and low load. Because of that the precision of the calculated mean (Figure 5-(a) and Figure 6-(a)) becomes worse.

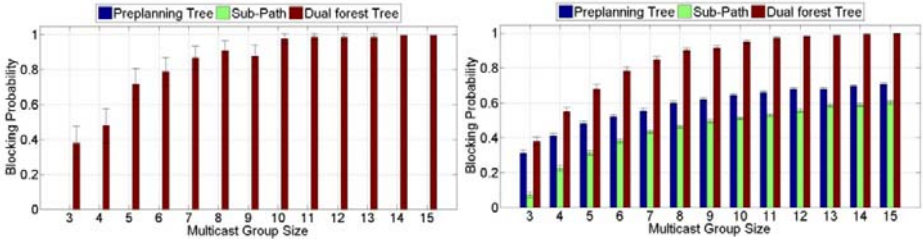

Fig. 5. Blocking probability as function of multicast group size using Germany network topology: a) Number of multicast groups = 1 (left); and b) Number of multicast groups = 15 (right)

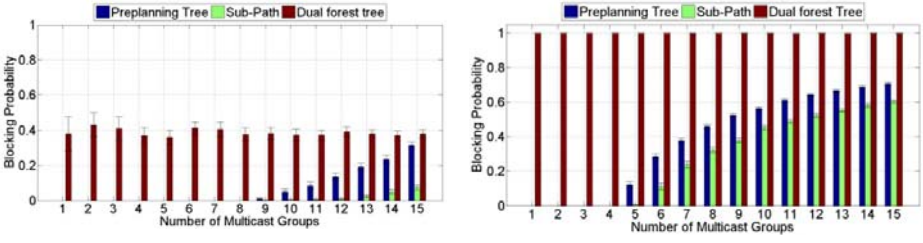

Fig. 6. Blocking probability as function of number of multicast groups using Germany network topology: a) Multicast group size = 3 (left); and b) Multicast group size = 15 (right)

4.3.2 European Network with 28 Nodes and 41 Links

We repeat the calculation process with a larger, but less meshed network topology. In the same way, we present the results in two forms: as a function of multicast group size and as a function of number of multicast groups. The results presented in Figure 7 and 8 give the same conclusions as the results presented previously. However, the performance of the investigated algorithms becomes somewhat worse, compared to the results of the fist network topology. This is because the network mesh level plays a role in finding a backup tree, when the network load and the multicast group size increase. On the contrary, the increasing of the network size can improve the performance of the investigated algorithms in the small group sizes. Form these results we can infer that the performance of the investigated algorithms depend on the network topology, network load and multicast group size.

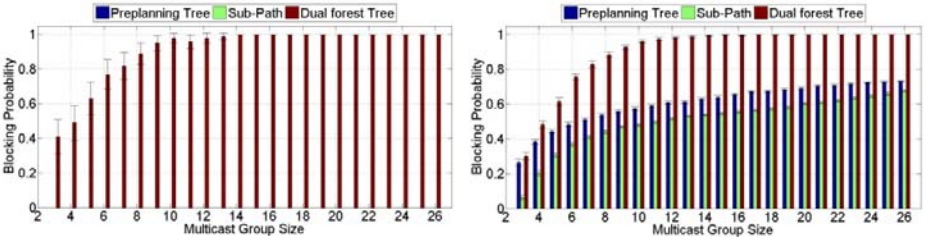

Fig. 7. Blocking probability as function of multicast group size using Europe network topology: a) Number of multicast groups = 1 (left); and b) Number of multicast groups = 15 (right)

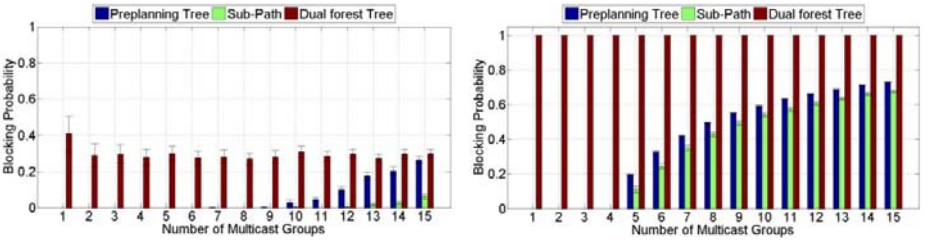

Fig. 8. Blocking probability as function of number of multicast groups using Europe network topology: a) Multicast group size = 3 (left); and b) Multicast group size = 26 (right)

5 Conclusions

In this work we proposed three algorithms used to protect the multicast distribution tree. In IP/MPLS core networks, the 1+1 unicast resilience is used to protect link as well node failures. As mentioned above using unicast algorithms to protect the multicast distribution trees is inefficient and resulting in multiplying the required bandwidth. Recently, several proposals are investigated to improve the MPLS standard to be able to protect multicast trees [WB08].

This paper focuses on the resilience issue. To reduce the required backup bandwidth, we delete the sharing links between the primary and backup tree. This results also in avoiding the double transition of the multicast data in the same link direction. Blocking probability is used to compare and evaluate the performance of the investigated algorithms. We can clearly see that the sub-path algorithm performs at the best. The preplanned tree algorithm seems to perform similar as sub-path algorithm in the large multicast group sizes. However, it becomes worse in the small group sizes with high network load. That is because the preplanned tree algorithm is based on finding two separate distribution trees that increase the reserved backup capacity. On the contrary, the performance of the dual forest tree is at the worst. Because this algorithm tries to find its backup as a shortest path between the leaf nodes of the primary tree without sharing any link and node of the primary tree, it is also not applicable for large group sizes.

It is also important to investigate the reaction time needed to reconstruct the distribution tree when a single link/node failure occurs. This time depends on the used

recovery method. If we use the 1+1 class, the recovery time is smaller than in case of 1:1 class. The implementation of the 1+1 class is also simpler. However, the network load will increase. This issue will be investigated in future work.

Acknowledgment

The work presented in this paper is a result of the CELTIC project 100GET-E3, which is partially supported by Nokia Siemens Networks GmbH & Co. KG and the German Federal Ministry of Education and Research (BMBF) under grant 01BP0740.

References

[AH08] IEEE 802.1AH Draft 4.2: Provider Backbone Bridges. IEEE 802.1 Working Group, http://www.ieee802.org/1/pages/802.1ah.html

[FCG01] Fei, A., Cui, J., Gerla, M., Cavendish, D.: A "Dual-Tree" Scheme for Fault-Tolerant Multicast. In: ICC 2001. IEEE International Conference, vol. 3, pp. 690–694 (2001) ISBN 0-7803-7097-1

[FED07] Fedyk, D.: Provider Link State Bridging. Nortel Networks (2007), http://www.ieee802.org/1/files/public/docs2007/aq-fedyk-plsb-present-0107.pdf

[MBG99] Médard, M., Finn, S.G., Barry, R.A., Gallager, R.G.: Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs. IEEE/ACM Trans. Networking 7, 641–652 (1999)

[MW08] Documentation for MathWorks Products (R2007b), http://www.mathworks.com/access/helpdesk/help/helpdesk.html

[NOB08] NOBEL 2 Project: D1.1 Architectural vision of network evolution, 01.03.2006-28.02.2008, http://www.ist-nobel.org/Nobel2/imatges/D1.1-Public%20part-final.pdf

[RFC1075] Waitzman, D., Partridge, C., Deering, S.: Distance Vector Multicast Routing Protocol. RFC1075 (1988)

[RFC2201] Ballardie, A.: Core Based Trees (CBT) Multicast Routing Architecture (1997)

[RFC4601] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I.: Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification, RFC4601 (2006)

[SCM06] Saidi, M.Y., Cousin, B., Molnár, M.: Improved Dual-Forest for Multicast Protection. In: NGI 2006. 2006 2nd Conference, vol. 8, p. 378 (2006) ISBN 0-7803-9455-0

[WB08] Ward, D., Betts, M.: MPLS Architectural Considerations for a Transport Profile. ITU-T - IETF Joint Working Team (2008), http://www.ietf.org/MPLS-TP_overview-22.pdf

[Wil02] Williamson, B.: Cisco System: Developing IP Multicast Networks, vol. I (2002) ISBN 1-57870-077-9

[XLT03] Xue, G., Chen, L., Thulasiraman, K.: Quality-of-Service and Quality-of-Protection Issues in Preplanned Recovery Schemes Using Redundant Trees. Selected Areas in Communications, IEEE Journal 21, 1332–1345 (2003)