Dynamically Adaptive Mulitipoint to Point LSPs

Constantinos Neophytou and Chris Phillips

Queen Mary University of London

Abstract: Multi-Protocol Label Switching provides a flexible forwarding mechanism within communication networks. Since its inception at the end of the last century it has been recognised that multipoint-to-point label switched paths offer a valuable way of further simplifying the forwarding process particularly within IP networks. Whilst work has been carried out into schemes for the formulation of such multipoint-to-point LSPs, only a couple of schemes have been proposed. Furthermore resilience in such configurations has only been partially addressed. This paper describes an improved method for the formulation of multipoint-to-point LSPs. In addition, the new notion of *Dynamically Adaptive Mulitipoint to Point* LSPs is presented in order to address the issue of resilience.

1 Introduction

Multi-Protocol Label Switching (MPLS) provides a mechanism for fast layer-2 forwarding together with flexible path selection, minimising the need for the cumbersome longest prefix match used in traditional IP routing. Virtual connections are pre-established and are realised in the form of label switched paths (LSPs). However, by using this connection-oriented approach, the resilience and flexibility provided by the connectionless forwarding of traditional hop-by-hop IP routing is to some extent lost. Conventional use of MPLS requires the use of point-to-point LSPs and thus the maintenance of individual connections throughout the network. The label swapping capability of MPLS facilitates this by allowing for labels to be bound to each leg of the path making up the complete point-to-point LSP. Particularly for large networks, where the greatest benefits of layer-2 forwarding can be derived, such an arrangement is difficult to maintain and does not scale well without aggregation through label stacking. What is required is a way in which the benefits of MPLS can be bought to the core of the network whilst still maintaining qualities normally associated with traditional IP routing. Identified in even the earliest drafts of the IETF's MPLS architecture [1] is the notion of multipoint-to-point (MP2P) LSPs and this (we believe) is the key to achieving the goal of fast forwarding coupled with resilience and flexibility. A multipoint-to-point LSP can be considered as an inverted tree routed at the destination whereby all sources are connected to a particular destination via a single MP2P tree. Such a configuration replaces each point-to-point LSP connection with a single LSP for each destination. The path to all destinations is thus realised as an overlay of these MP2P trees. Naturally a destination can have more than one tree associated with it as service classification and path constraints dictate, providing for an efficient method of implementing different classes of service. Although at first glance MP2P LSPs are an elegant way to provide a more scalable approach within MPLS networks the key problem, as identified by Saito et al [2], is the difficulty in creating them. Consequently only a few schemes have been proposed for this and the issue of resilience in this research area has only partially been addressed. In this paper an improved method for the creation MP2P LSPs is presented. Furthermore, the scheme addresses the issue of resilience in such configurations in an attempt to bring to layer-2 forwarding what is taken for granted with traditional IP routing.

Saito et al [2] propose a scheme whereby routes are initially selected between each ingress/egress node pair and then the MP2P LSP is designed to include the selected the routes. The actual design or creation of this MP2P LSP is a consequence of 0 – 1 integer programming and this is formulated for each individual egress node. Failure and recovery is addressed in this scheme by means of a complete MP2P LSP acting as a backup to the working MP2P LSP such that, should a route between an ingress/egress pair be severed, the backup MP2P LSP is chosen by the ingress as it is already set up. Bhatnagar et al [3] state that this scheme cannot be used frequently for large MPLS domains as integer programming is in general NP–hard and propose another scheme for the formulation of MP2P LSPs. The scheme proposed in [3] is similar to that proposed in [2] in that both use P2P LSPs as a basis in for the formation of MP2P LSPs. Bhatnagar et al [3] describe their scheme as a simple polynomial time heuristic based algorithm and have shown that a benefit of using MP2P LSPs over P2P LSPs is a reduction in the required label space. Their scheme is a merging algorithm that takes as its input P2P LSPs and merges them to create an MP2P LSP routed at the egress. It does so by merging the LSPs at a contiguous set of common nodes starting at the egress whilst providing constraints on merging for loop prevention. Another approach for the creation of MP2P trees has been proposed in [4]. It uses a colony of mobile agents to search out and discover the MP2P tree. The use of mobile agents in this field relative to existing technologies is still in its infancy and the overhead in terms of processing and congestion for what is a comparatively low level task is questionable. Whilst both of the schemes proposed in [2] and [3] have their merits in the creation of MP2P LSPs, both rely on the existence of an appropriate selection of P2P LSPs that can be subsequently merged. Neither scheme has fully addressed the issue of resilience in MP2P LSPs. The scheme proposed in this paper differs from the aforementioned schemes in that, rather than proposing a new algorithm for the creation of MP2P trees, it leverages existing ones that are employed by a relatively mature family of protocols, namely, multicast protocols. In addition to this, the presented scheme provides for resilience in the MP2P tree once it has been formed. The remainder of this paper describes our proposed scheme and goes on to identify future work in the development of this field and further refinement of our scheme.

2. Inverted Multicast Trees

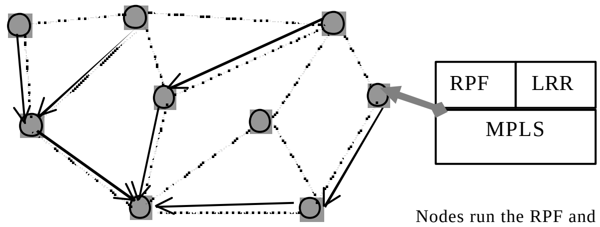
One of the concepts upon which our scheme is based is that a MP2P tree is a special case of a Point-to-Multipoint (P2MP) tree. Simply put, a MP2P tree can be considered to be an inverted P2MP tree. The formation of P2MP trees as part of multicast routing protocols is well established and our scheme proposes to use this in the creation of MP2P trees. In multicast routing, the creation of the multicast delivery tree is undertaken by a multicast forwarding algorithm. There are two main algorithms for this: Reverse Path Forwarding (RPF) and Centre Based Tree algorithm (CBT). The majority of multicast routing protocols are based on either of the two, with the exception of Protocol Independent Multicast-Sparse Mode (PIM-SM) that operates with either algorithm and Multicast OSPF (MOSPF) that uses the OSPF Link-State database. In CBT a centre point is chosen in the multicast group and each recipient sends a join request to that point the result is a shared tree for that multicast group. There exists a many to one mapping of sources to a particular group, the tree is identical for all sources so in real terms the result is a Multipoint-to-Multipoint (MP2MP) tree or shared tree. In contrast to this, RPF builds a single tree for each source, there is a one to one mapping between source and group resulting in a P2MP tree or source tree. It is clear that any multicast routing protocol that creates a source tree clearly lends itself to being adapted and used to create MP2P trees. The protocols that meet this criterion are Protocol Independent Multicast-Dense Mode (PIM-DM), Distant Vector Multicast Routing Protocol (DVMRP), MOSPF and Source Specific Multicast (SSM). Future iterations of our proposed scheme may be developed with each of these protocols in mind in order to ascertain which is best suited for use within this context. As a starting point and for the initial proposal of our scheme PIM-DM has been chosen, as it is relatively simple and is based on the RPF algorithm using a flood and prune mechanism to create the tree.

3. Dynamically Adaptive MP2P (DAM) Trees

The proposed scheme is described in this section. The operation of the scheme can be considered to be in two parts, these are the creation of MP2P LSPs and then their subsequent maintenance.

3.1 MP2P Creation - The tree is initially created using RPF and for our initial implementation we use PIM-DM. The algorithm works as follows: trees are calculated and updated dynamically. The algorithm maintains a reverse path table used to reach each source. In our application of the algorithm the source in this table eventually becomes the sink in our MP2P configuration. This table maps every known source node to the preferred interface used to reach that source. When forwarding data, if the datagram arrives through the preferred interface used to transmit datagrams back to the source, the datagram is forwarded through every appropriate downstream interface. Otherwise, if the datagram has arrived through a sub-optimal path it is subsequently discarded. Using this process, duplicate packets caused by network loops are filtered out. This corresponds to the flood part of the *flood and prune* process used in multicast protocols, the prune simply involves the pruning of branches of the tree for nodes which do not wish to receive data from a multicast group and in our application do not require to send data to a particular sink node. The manner in which the preferred interface is chosen directly impacts on how optimal the tree is, for example if used in conjunction with a routing protocol like OSPF the preferred interface as a result of Dijkstra's algorithm is found to be on the shortest path leading to the least cost delivery tree. RSVP - TE could be used to traffic engineer constrained paths forming a sub optimal tree where required. Unlike the scheme proposed in [2], where the MP2P LSPs are pre-assigned and are created on the basis of network topology, the proposed scheme forms the MP2P dynamically on the basis of preferred interfaces allowing for adaptive tree growth based on chosen criteria.

3.2 MP2P Maintenance - The maintenance mechanism of our proposed scheme implements fast reactive healing of the tree. Running along side RPF we propose the use of the Link Reversal Routing (LRR) Algorithm normally used in ad-hoc networking, as shown in Figure 1. Please refer to [5] for an explanation of LRR and related protocols. The LRR algorithm has two variants: the Full Reversal method and the Partial Reversal method [6]. In this scheme we propose the use of the Partial Reversal method. There are a number of protocols based on this algorithm the most recent being the Temporally Ordered Routing Algorithm (TORA) [7]. A directed acyclic graph (DAG) is formed running from all nodes to the sink node. This is a loop free graph of multipath routes to a particular sink node. A "height" metric is used to establish the DAG; links between nodes are assigned a direction based on the relative height metric of neighbouring nodes i.e. the direction points towards a neighbouring node of a lesser relative height. When a node (except the destination node) loses its last outgoing (downstream) link it generates a new reference level. The effected nodes new height triggers a localised structured reaction to the failure resulting in the reversal of one or more of its incoming links. This takes place until all nodes have a route to the sink node and the DAG is complete again. As part of our scheme a distributed check of the preferred interface in the reverse path table used by RPF is made against the DAG. If the preferred interface is not one that corresponds to one of the multipaths on the DAG a change is made to the effected area re-establishing its connection to the sink and subsequently the healing of the MP2P tree. Note that this path re-establishment does not involve the re invoking the RPF flood process changes are only made to the effected nodes.



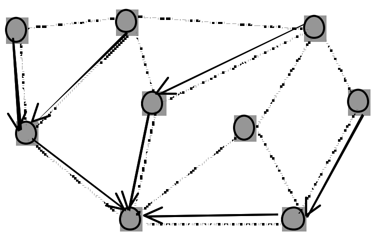
Destination
Node

Figure 1:- An MP2P Tree

Nodes run the RPF and LRR algorithms together to form the MP2P tree used by MPLS to create the MP2P LSP

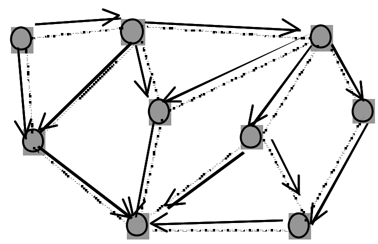
The result of many failures can lead to a deformed – sub optimal tree so we propose a periodic flood to counter this. We suggest a time of thirty minutes between floods but this time is readily configurable. In order to avoid network disruption during the course of and immediately after the periodic flood, the existing tree is not discarded until the new tree is fully established i.e. the branches of the tree have extended to the last upstream nodes (the ingress nodes). This is a gradual prune process, as the paths belonging to the new tree establish connections to the sink their respective paths in old tree wither away. To avoid isolation of nodes from the sink because a downstream node has cut an artery to a branch still using the old tree a node can only prune itself from a tree once it has lost its last incoming path otherwise it assumes other nodes are still using that branch and does nothing. The last upstream node (normally an ingress node) in any branch always initiates this gradual pruning process.

3.3 Example Operation - This section provides a step through of the various processes that make up a DAM. In the following illustrations, figure 2 shows a new tree that has just been created as a result of RPF. This could be a brand new tree or a tree resulting from the periodic flood mechanism described in the previous section. Figure 3 then shows the initial DAG created by LRR



Destination
Node

Figure 2:- MP2P Tree Formed Through the Use of RPF



Destination
Node

Figure 3:- A DAG of the Same Network in Figure 2

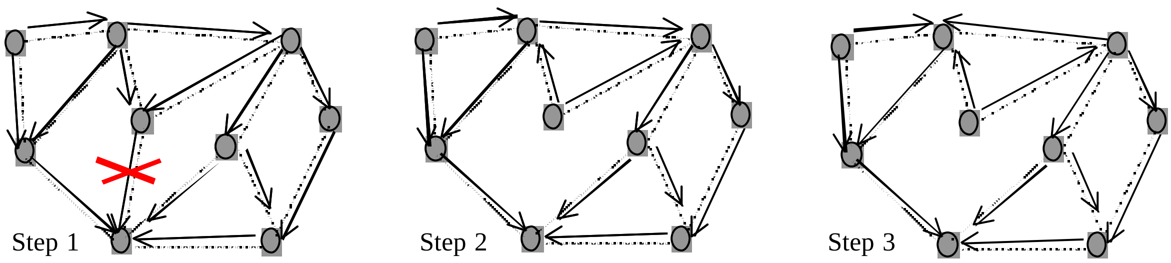


Figure 4:- LRR Reforming the DAG After a Link Failure

Figure 4 shows its subsequent revision after a link failure. Step1 shows the failure of a link. Step 2 shows the effected node reversing its incoming links after recalculating its reference level as a result of the failure. This triggers the structured reaction and reversal of other links by nodes in the immediate locality. Step 3 shows the reformed DAG.

Finally, figure 5 shows the healed tree after comparison between the new DAG in figure 4 and the RPF reverse path table.

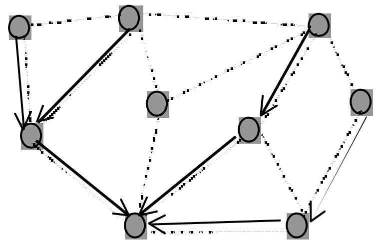


Figure 5:- The Repaired MP2P Tree Formed by Reactive Healing After a Link Failure

4. Summary & Conclusions

In this paper we have proposed an improved scheme for the creation and maintenance of MP2P LSPs. We have identified the use of Multicast forwarding algorithms and protocols for the creation of MP2P trees. Our scheme provides for distributed discovery and formation of trees and through our maintenance scheme adaptive growth and contraction. The scheme provides fast reactive healing through the use of LRR that is a robust algorithm originally designed for adhoc networks. We believe our approach to be simpler and more robust than that proposed in [2]. We do not believe backup trees are necessary as in [2] because our fast healing scheme makes changes only within the immediate locality of a failure nevertheless separate class of service trees can coexist. We are currently evaluating this scheme using OPNET™, and aim to demonstrate its utility with a variety of network scenarios in the near future.

Acknowledgments

The authors graciously acknowledge the EPSRC for their support in this research.

References

[1] RFC 3031 “Multiprotocol Label Switching Architecture”, E. Rosen, A. Viswanathan and R. Callon, January 2001

[2] “Traffic Engineering using Multiple Multipoint-to-Point LSPs”, Hiroyuki Saito, Yasuhiro Miyao and Makiko Yoshida, IEEE INFOCOM 2000

[3] “Label Space Reduction in Multipoint-to-Point LSPs for Traffic Engineering”, Sudeept Bhatnagar, Samrat Ganguly and Badri Nath, 2nd European Conference on Universal Multiservice Networks, ECUMN2002

[4] “QoS Routing for MPLS Networks Employing Mobile Agents”, S. Gonzlez-Valenzuela and V.C.M. Leung, IEEE Network, Volume: 16 Issue: 3, May/Jun 2002, pages 16 - 21

[5] “Ad Hoc Networking”, Charles E.Perkins, Addison-Wesley December 2000, Chapter 8 - Link Reversal Routing

[6] “Distributed Algorithms for Generating Loop-free Routes in Networks with Frequently Changing Topology”, E. Gafni and D. Bertsekas, IEEE Personal Communications January 1981

[7] “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks”, Vincent D. Park and M. Scott Corson, Proceedings of INFOCOM 97, IEEE 1997