

A Configuration Algorithm for Provisioning of Cell-Based MPLS Network

Daniel Won-Kyu Hong¹, Choong Seon Hong², and Dong-Sik Yun¹

¹ Operations Support System Lab., KT

463-1 Jeonmin-Dong Yuseong-Gu, Daejeon 305-811 Korea

{wkhong,dsyun}@kt.co.kr

² School of Electronics and Information, Kyung Hee University

1 Seocheon-Ri Kiheung, Yongin, Kyungki-Do 449-701 Korea

cshong@khu.ac.kr

Abstract. This paper proposes a configuration algorithm of multipoint-to-point connection for facilitating VC merge concept in an MPLS over ATM network. This configuration solves two major problems: determining the most reasonable candidate ATM switch served as an ATM-LSR under the complex ATM network topology; and providing the most reasonable multipoint-to-point connections with regards to network resource utilization. This algorithm is therefore subsequently composed of two procedures: ordering and backtracking. The ordering algorithm is for determining the orders of link and node taking the routing metrics of available bandwidth delay and hop count into account. The backtracking algorithm is to determine the optimal multipoint-to-point route based on the order information allocated by ordering algorithm. We evaluate the proposed algorithms under the operational network environment of KT.

1 Introduction

The Internet Engineering Task Force community recently established a working group on multi-protocol label switching (MPLS) aimed at standardizing the label-swapping paradigm that integrate layer 3 routing with layer 2 switching [1]. At the outset, MPLS is designed to be scalable and capable of being deployed in the core network with a large number of nodes. VC merging is an attractive feature that allows MPLS to scale to a very large network. Although layer 2 switching in MPLS can be applied in principle, the use of asynchronous transfer mode (ATM) switches in the backbone network is considered the most attractive solution as ATM hardware switches have been extensively studied and are widely available in various architectures [4,5,6,8]. This paper defines layer 2 switching that uses the ATM technology and refer to the associated label switched router (LSR) as an ATM-LSR. In this case, each IP packet may be segmented to multiple 53-byte cells before being switched. Traditionally, ATM adaptation layer 5 (AAL5) has been used as the encapsulation method in data communications since it is simple, efficient, and possesses a powerful error-detection mechanism.

For the ATM-LSR to forward incoming cells to the correct output, the IP-route information needs to be mapped to the ATM labels that are kept in the virtual path identifier (VPI) or/and virtual channel identifier (VCI) fields. The relevant route information that is stored semi-permanently in the IP-routing table contains the tuple {ingress, next-hop router}.

Several methods for mapping the route information to ATM label exist. In the simplest method, each ingress-ingress pair is mapped to a unique VCI at a switch. The non-VC merge scheme, as this method is called, allows receivers to easily reassemble cells into respective packets as the VCI can be used to distinguish between senders. However, for n ingresses and ingresses, each AT-LSR is potentially required to manage $O(n^2)$ VC labels for full-meshed connectivity. This method is not scaleable for large networks. On the other hand, in the VP merge scheme the VP labels of cells intended for the same ingress are translated to the same outgoing VP value, thereby reducing VP consumption downstream. The VCI identifies the sender for each VP so that the receiver can reconstruct packets, even though cells from different packets are allowed to interleave. Although it can remarkably reduce the number of label entries compared with the non-VC merge scheme, VP merge is limited because the VP space is limited to only 4096 entries at the network-to-network interface (NNI). The VC merge scheme does not have the space-constraint problem as in VP merge. However, it cannot distinguish the sender at the output port of ATM-LSR. Therefore, cells belonging to different packets for the same ingress cannot interleave with each other or the receiver will not be able to reassemble the packets. Currently there have been several attempts to solve the problem of indistinguishable senders of VC merge and to propose the ATM-LSR architecture to accommodate VC merge scheme [4,5,6,7,8]. Using the results of these studies [4,5,6,7,8], scalable MPLS network using VC merge scheme in MPLS over ATM network can be configured. However, our proposal gives less emphasis to deployment of MPLS over ATM network. In newly-deployed ATM-LSR under the pre-provisioned ATM network in which some network resources are already allocated to ATM service provision, the most reasonable merge point where the existing ATM switch should be replaced with ATM-LSR to provide efficient MPLS service using VC merge concept must be determined.

This paper proposes an innovative multipoint-to-point algorithm that is able to identify the most reasonable VC merge point under the complex ATM network. It is composed of two subsequent algorithms: ordering and backtracking. The ordering algorithm trims the unreasonable nodes or links and assigns the orders of nodes and links with routing metrics of hop count and available bandwidth. The performance of ordering is unrelated to the number of connecting points in multipoint-to-point connection, as will be demonstrated by performance comparable to the original topology. The backtracking algorithm finds the most reasonable merge point based on the order information of nodes and links allocated by the ordering algorithm. Backtracking performs the connection admission control (CAC) and resource reservation. In addition, some LSRs can be added to the ATM-LSR according to the network expansion. This paper also

proposes the addition of LSRs mechanisms, also using ordering and backtracking algorithms, to the existing multipoint-to-point connection to add new edge LSRs to PoPs. The proposed multipoint-to-point algorithm finds the optimal merge point under the complex ATM network composed of 400 ATM switches and 1,078 links connecting per node. In addition, the performance of ordering and backtracking algorithms under the above network topology is measured.

2 Determination of an Optimal Merge Point

MPLS networks can use traditional ATM equipment to migrate and introduce MPLS to an existing ATM network. ATM network providers may prefer to keep an existing ATM infrastructure. On the other hand, building a new MPLS infrastructure allows customers to access both the MPLS network and services and along with the old ATM network, even from a single access link. There are several considerations in choosing ATM LSRs, such as type of trunks, number of trunks, number of connections, whether or not VC merge is required, and requirements for redundancy and reliability. This paper focuses only on the support of VC merge facility while maintaining traditional ATM services. Fig. 1 shows a simplified network configuration for introducing MPLS service to an existing ATM network. There are three ingress label switched routers (LSR1, LSR2, and LSR3) and one egress LSR (LSR4).

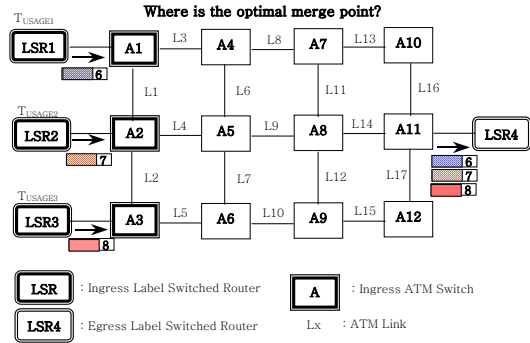


Fig. 1. Determination of optimal VC merge point

In order to accommodate MPLS service using the VC merge scheme, the most efficient location should be determined. There are three intermediate ingress ATM switches and one intermediate ingress ATM switch. In such an environment, a novel multipoint-to-point algorithm is needed to find the most reasonable merge point, particularly with regards to the maximization of network resource utilization and end-to-end transit delay. Determining the optimal merge point is vital in introducing MPLS service to a traditional ATM network as the ATM

switch identified as the optimal merge point should be replaced with the ATM-LSR having MPLS control functions. To find the most reasonable merge point under the complex ATM network topology, taking into account network resource utilization and end-to-end delay, we propose an innovative multipoint-to-point algorithm composed of ordering and backtracking algorithms.

2.1 Ordering Algorithm

The ordering algorithm is for determining the orders of link and node taking the routing metrics of available bandwidth delay and hop count into account. Network topology is represented by Graph, $G(V, E)$, where Vertex (V) denotes a set of nodes and Edge (E) denotes a set of links among nodes. Vertex contains three kinds of information: order (O), visit flag ($VISIT$), and availability (AVA); while edge maintains three kinds of information: order (O), available bandwidth (B_{AVA}) and availability (AVA). Availability represents the possibility of cell transfer capability of each vertex or edge, which depends mostly on fault status or performance degradation of nodes or links.

```

1.  $G(V, E)$ , where  $V$ : Vertex,  $E$ : Edge
2.  $V = \{O, VISIT\}$ ,  $E = \{O, B_{AVA}\}$  //  $O$ : Order,  $B_{AVA}$ : Available Bandwidth
3.  $B_{REQ}$ : Requested Bandwidth
4.  $E_{ADJ}$ : Adjacent Vertex of  $E$ 
5. for all  $E(O) \leftarrow \infty$ ; //  $\infty$ : Infinite
6.  $V_{VISIT}(O, VISIT) \leftarrow (0, YES)$ ; //  $V_{VISIT}$ : Source Vertex
7. Procedure  $Ordering(V_{VISIT}, B_{REQ})$  {
8.   for all  $E \in V_{VISIT}$  {
9.      $E(O) \leftarrow V_{VISIT}(O) + 1$ ;
10.     $V_{NEIGH} \leftarrow E_{ADJ}$  // Adjacent Vertex of  $E$ 
11.    if( $V_{VISIT}(VISIT) == NO$  &&  $V_{NEIGH}(O) < E(O)$  &&  $E(B_{AVA}) \geq B_{REQ}$ ) {
12.       $V_{NEIGH}(O, VISIT) \leftarrow (E(O), YES)$ ;
13.      if( $V_{NEIGH}(O) \neq \infty$ )
14.         $Ordering(V_{NEIGH}, B_{REQ})$ ; // Recursive Call
15.    } else {
16.       $E(O) \leftarrow \infty$ ;
17.       $V_{NEIGH}(O) \leftarrow \infty$ ;
18.    }
19.  } // end of for
20.} // end of Ordering

```

Fig. 2. Ordering algorithm

Fig. 2 illustrates the ordering algorithm. Fig. 3 shows the ordering procedure for determination of orders of nodes and links from ingress nodes of A and B to egress node E with the requested bandwidth of 7Mbps. In the ordering procedure, the ingress and egress vertexes are reversed. In Fig. 3, there are two ingress vertexes of A and B and one egress vertex E. Traversing the network topology to determine the orders of vertexes and the orders of edges, we define two terminologies of the visiting vertex (V_{VISIT}) and the neighbor vertex (V_{NEIGH}). The V_{visit} is an active vertex that allocates the proper orders of all neighbor edges (V_{NEIGH}) connected to it and sets its visit flag to YES. On the other hand, V_{NEIGH} is a passive vertex in that its order is determined by the order of edge

connected to it and its visit flag is not changed to YES until it can be V_{VISIT} . For example, when we start to traverse network topology from ingress vertex E, it can be V_{VISIT} , and the vertexes of C, and D connected to the edges (L1 and L3) of vertex E can be V_{NEIGH} s in Fig. 3.

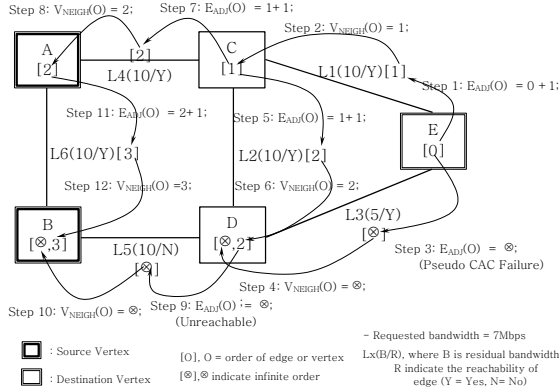


Fig. 3. Ordering procedure

The ordering algorithm trims the unusable network resource from whole network topology in terms of lack of resource and abnormalities and determines the appropriate orders (O) of links and nodes. In order to assign proper order to each edges or vertexes, initially, we assign infinite (∞) to all orders of vertexes and edges and assign NO to visit flags of all vertexes, as show in line 5 of Fig. 2 (for all $E(O) \leftarrow \infty$;). We assign zero to the order of egress vertex E that can serve as the first V_{VISIT} , as shown in line 6 of Fig. 2 ($V_{VISIT}(O, VISIT) \leftarrow (0, YES)$;). From here on, we traverse the network topology with the ordering algorithm of Fig. 2 until the visit flag of all vertexes is set to YES and the order of all edges and vertexes according to the hop count, availability and available bandwidth is assigned with the following rules:

The determination of the next V_{VISIT} is done in line with the depth first search manner. One important scheme can determine the efficient branch point for multipoint-to-point route. It is normal that one of the V_{NEIGH} can serve as the next V_{VISIT} in terms of the depth-first search scheme. We hereby define one exception where the V_{NEIGH} cannot serve as the next V_{VISIT} if its order is infinite (∞). For example, the next V_{VISIT} vertex of E is not D but C because the order of edge L3 connected to vertex E is infinite (∞) in the case of ordering algorithm of Fig. 3.

In the process of traversing the network topology, the order is assigned according to the simple rule that the order of edges connected to the V_{VISIT} is assigned by the order of the V_{VISIT} plus one ($E(O) \leftarrow V_{VISIT}(O) + 1$). If the order of any edge is larger than the order of V_{VISIT} plus one, it is changed to

the order of V_{VISIT} plus one. If the order of any edge is less than or equal to the order of V_{VISIT} plus one, it will not be changed. For example, the V_{VISIT} is E, the links connected to the vertex E are L1 and L3. The requested bandwidth (B_{REQ}) is 7Mbps. The order of the edges connected to the vertex E is replaced with one (the order of vertex E plus one) because the order of the first V_{VISIT} E is zero. If any link can accommodate B_{REQ} and its order is not infinite, its order will be changed to one such as L3. However, if any link cannot accommodate B_{REQ} , its order is changed to infinite (∞), having nothing to do with the order of V_{VISIT} . If the order of any link such as L1 is greater than or equal to the order of the V_{VISIT} plus one, it will not be changed. Fig. 3 shows the sample network topology and its results of order assignment in accordance with the ordering algorithm. There are three ingress vertexes of A1 and A2 and one egress vertex A11 with the requested bandwidth of 20Mbps.

2.2 Backtracking Algorithm

Backtracking aims to determine the optimal multipoint-to-point route based on the order information allocated in the process of order assignment. In multipoint-to-point route selection, the ingress and regress vertexes are not reversed when compared with the ordering procedure. In addition, the CAC on the selected edges is done in multipoint-to-point route selection procedure. Fig. 4 shows the multipoint-to-point route selection procedure based on the order assignment result of Fig. 3.

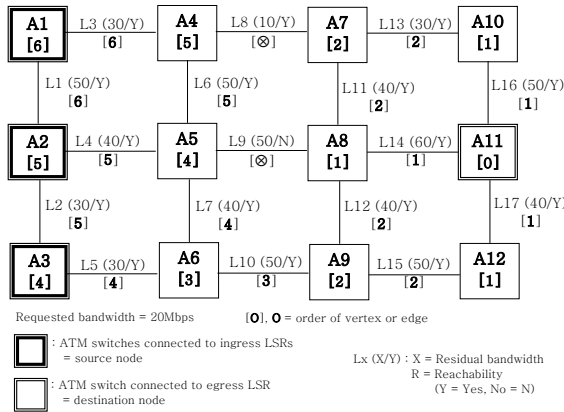


Fig. 4. Results of order assignment

There are three ingress vertexes of A1, A2 and A3 and one egress vertex A11 from the perspective of backtracking procedure, as shown in Fig. 5. Backtracking algorithm is as follows:

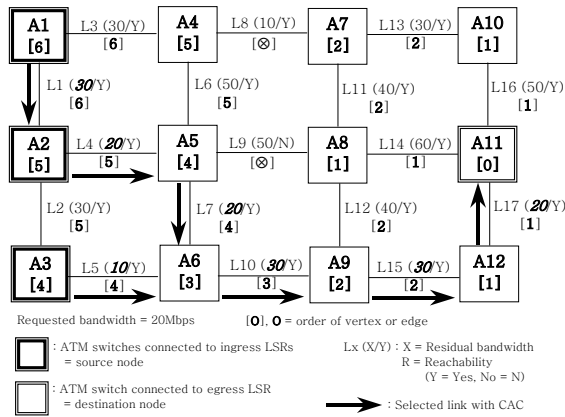


Fig. 5. Backtracking algorithm and procedure

- When it traverses the network topology from the ingress to egress vertexes, it selects the edge having the least order. For example, ingress vertex A3 selects the edge L5 as an optimal edge between two edges of L2 and L5 because the order of L2 is less than that of L5.
- If there are two or more edges with the same order, the edge having the largest residual bandwidth is selected. For example, ingress vertex A1 selects L1 as an optimal edge between the two edge of L1 and L3 because the residual bandwidth (30Mbps) of L1 is larger than that (10Mbps) of L3 after CAC.

Thus, the CAC on the selected edges is carried out in multipoint-to-point backtracking. The optimal route between ingress vertex A1 and egress vertex A11 can be {A1-L1-A2-L4-A5-L7-A6-L10-A9-L15-A12-L17-A11}. Furthermore, The optimal route between A2 and A11 can be {A2-L4-A5-L7-A6-L10-A9-L15-A12-L17-A11}. In addition, the optimal route between A3 and A11 can be {A3-L5-A6-L10-A9-L15-A12-L17-A11}.

2.3 Determination of Optimal VC-Merge Point

With the ordering and backtracking algorithms, we can determine the optimal VC-merge point under the complex network topology. The VC-merge point implies that an ATM switch identified as an optimal VC-merge point should be replace with the ATM-LSR having MPLS VC-merge capability. As a result of backtracking, there are duplicated routes among multipoint-to-point routes. In the case of Fig. 6, the duplicated route among three ingress vertexes of A1, A2 and A3 and egress vertex A11 can be {A6-L10-A9-L15-A12-L17-A11}. Hence, the first duplicated vertex can be the optimal VC-merge point, as shown in Fig. 6. Therefore, we can replace the ATM switch corresponding to A6 with the ATM-LSR having VC-merge capability. Of course, there is another candidate

VC-merge point of A2. However, it can only merge MPLS traffics from A1 and A2. We need to identify the optimal VC-merge point where all MPLS traffics from A1, A2 and A3 is simultaneously merged. It can be the first duplicate vertex among the found three routes in terms of multipoint-to-point route selection, as shown in Fig. 6. The ingress vertexes of A1, A2 and A3 are connected to ingress ATM-LSRs and egress vertex A11 is connected to LSR in IP-based MPLS network. In order to configure ATM based MPLS network, we establish ATM virtual path connection corresponding to an Label Switched Path (LSP) between ingress ATM-LSR and egress ATM-LSR having VC-merge capability.

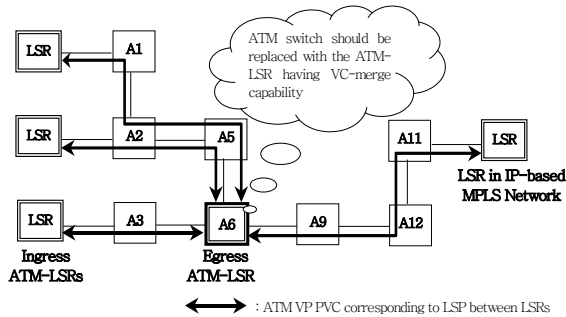


Fig. 6. Optimal VC-merge point

3 Performance Analysis

We implement the multipoint-to-point routing algorithm as a part of network management system (NMS) in accordance with the functional layering concepts of TMN, as applied to the High Speed Information Network (HSIN) of KT. We evaluate the proposed multipoint-to-point algorithm under the KT's HSIN composed of 14 core nodes, 38 edge nodes and 239 access nodes. All the core nodes are connected in nearly full mesh with STM-4, while the edge and access nodes are connected to the core nodes in star topology with STM-1. There are 291 nodes and 908 links, comprising a significantly large network. We measured the performances of ordering algorithm, backtracking algorithm and adding branch algorithm using the SUN E65000 equipped with 8CPUs and 2G main memory. We did not use any simulation tool to measure performance because of some differences between performance measured using any simulation tool and performance measured using the real environment. As a national network carrier, we need real performance measured using a real service provider to guarantee customer's satisfaction. Fig. 7 presents the performance of multipoint-to-point route selection up to one hundred ingresses. There are two major performance metrics: ordering performance and backtracking performance. In ten ingresses, it

took 2.13 seconds to find the optimal multipoint-to-point route. In the worst case among fifty-one ingresses, it took 4.48 seconds to find the multipoint-to-point route. The ordering performance is constant under the same network topology regardless of the number of ingresses. However, backtracking performance including CAC and residual bandwidth manipulation gradually deteriorates until the number of ingresses reaches fifty-one. However, the backtracking performance gradually improves in proportion to the number of ingresses when it is over fifty-one, as shown in Fig. 7.

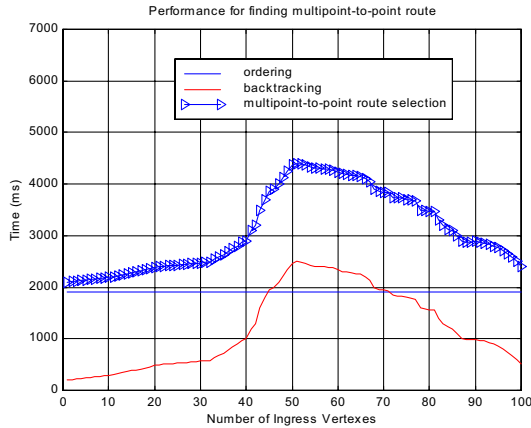


Fig. 7. Performance of multipoint-to-point routing algorithm

We analyze performance under the constraint of one hundred ingresses as the maximum number for purposes of validating scalability. The ordering algorithm manifested a constant performance of 1.90 seconds under the same network topology. However, the backtracking algorithm resulted in higher performance in reverse proportion to the number of ingresses when it is larger than fifty-one. It is mainly because the duplication ratio of intermediate edges and vertexes are increased in proportion to the number of ingress. Our performance evaluation revealed that the proposed ordering and backtracking algorithms for finding the optimal merge point for utilizing ATM VC-merge concept under the MPLS over ATM network environment yielded good scalability.

4 Concluding Remarks

This paper proposed the multipoint-to-point routing algorithm to find the optimal merge point for utilizing ATM VC-merge concept under the MPLS over ATM network. It could identify the ATM switch that should be replaced with ATM-LSR having VC-merge capability. It is composed of ordering algorithm for

determining the orders of link and node taking the routing metrics of available bandwidth delay and hop count into account and backtracking algorithm for selecting optimal multipoint-to-point routes based on the order information of vertexes and edges. In addition, this paper proposed the algorithm for adding extra branches to the existing multipoint-to-point route. This paper showed that the proposed ordering and backtracking algorithms could be applied to large-scale ATM network by means of performance analysis of the multipoint-to-point route selection. It could confirm the multipoint-to-point service provision within 2.13 seconds when the ingress is below ten, with residual bandwidth manipulation of CAC, and optimal route selection included. The ordering algorithm manifested a constant performance of 1.90 seconds under the same network topology. However, the backtracking algorithm resulted in higher performance in reverse proportion to the number of ingress when it is larger than fifty-one. It is mainly because the duplication ratio of intermediate edges and vertexes are increased in proportion to the number of ingress. Our performance evaluation revealed that the proposed ordering algorithm and backtracking algorithms for providing optimal multipoint-to-point service yielded good scalability. It is the most prominent strength of the proposed adding branch algorithm. With the proposed multipoint-to-point algorithm, we can find the most optimal merge point under the complex ATM network for utilizing the ATM VC merge concept under the MPLS over ATM network.

References

1. Rosen, E., Viswanathan, A. and R. Callon, "Multi-Protocol Label Switching Architecture", RFC 3031, January 2001.
2. B. Davie, J. Lawrence, K. McCloghrie, et al, "MPLS using LDP and ATM VC Switching," RFC 3035, January 2001.
3. http://www.cisco.com/warp/public/121/mpls_vcmerge.html
4. Hungkei Keith Chow and Alberto Leon-Garcia, "VC-Merge Capable Scheduler Design," IEEE Proceedings of ATM Workshop, pp. 153–160, 1999.
5. R. Venkateswaran, C.S. Taghvendra, X.Chen and V.P. Kumar, "Support for Multiway Communications in ATM Networks," IEEE Proceedings of ATM Workshop, pp. 339–348, 1998.
6. Peiganf Zhou and Oliver W.W. Yang, "Reducing Buffer Requirement for VC-Merge Capable ATM Switches," Proceedings of IEEE Global Telecommunications Conference, pp. 44–48, Globecom'99, 1999.
7. Indra Widjaja and Anwar I. Elwalid, "Performance Issues in VC-Merge Capable Switches for IP over ATM Networks," IEEE Journal on Selected Areas in Communications, VOL 17, No. 6., pp. 372–380, June 1999.
8. D.W.Lee, T.W. Lee, Y.C.Kim, D.J.Choi and Mike M.O. Lee, "Implementation of a VC-Merge Capable Crossbar Switch on MPLS over ATM," IEEE International Conference on ATM and High Speed Intelligent Internet Symposium, ICATM 23001, pp. 66–69, 2001.