MPLS+ : A Scalable Label Switching Network
Shai Benjamin and Izhak Rubin
University of California, Los Angeles

Abstract - We introduce MPLS+, an extension to Multiprotocol Label Switching (MPLS) that improves the scalability of MPLS networks by allowing any node on an MPLS path to send labeled packets to any downstream node, and not only to the one node at the end of the path. This paradigm results in a reduction in the number of labels required in the label switching network and thus improves its scalability. We define conditions that allow a set of paths to provide all-to-all shortest-path connectivity in an MPLS+ network. We give a heuristic algorithm for the construction of efficient sets of paths in general topology networks and describe optimal examples for some special topologies (tree, grid, concentric rings).

I. INTRODUCTION

In [2] we introduced the paradigms of Path Merging and Intermediate Destination Removal. These paradigms increase the utilization of labels in a label switching network and thus allow for reduction in the size of routing tables and increased scalability of the network. In brief, Path Merging allows for paths carrying traffic to the same destination node to merge into an intree structure and to share the same label that identifies that destination. This paradigm is already in use today in MPLS networks ([5],[6]). Intermediate Destination Removal, used by Rubin & Ling in the context of optical and electrical cross-connect meshed-ring networks ([9],[10]), allows a labeled path to carry traffic from each node along the path to any downstream node, not only to the one node at the end of the path. It involves appending a sub-label or hop-count to each labeled packet so it can be terminated by the intended destination node.

In this paper we consider a third paradigm, namely Label Swapping, already in use in MPLS, in conjunction with the other two paradigms. With this paradigm, whenever a packet is forwarded from one node to another, the value of the label identifier may be swapped with another value. We shall use MPLS+ to refer to a network that employs all three paradigms: Path Merging, Destination Removal and Label Swapping.

Each node in an MPLS+ network is required to maintain a routing table and a switching table. The routing table is used for unlabeled packets that enter the network. It is keyed by destination node identifiers and includes three entries for every node in the network: an output label identifier, a label hop-count and a next-hop node. When an unlabeled packet enters the network, a header is appended to it with the label and hop-count taken from the routing table and it is then forwarded to the next-hop node. The switching table is used for handling packets that are already labeled. This table is keyed by input label identifiers and includes two entries for each key: the next-hop node and the output label identifier. Each labeled packet carries in its header the label identifier and a label hop-count. When a node receives a labeled packet it first checks if the label hop-count is zero. If this is the case then the packet is terminated. Otherwise, the label hop-count is decremented by 1, the label identifier is swapped with the corresponding output label extracted from the switching table and then the packet is forwarded to the next-hop node associated with the input label identifier.

The size of the routing table is always the number of nodes in the network (minus one, assuming a node does not need to route packets to itself). However, the size of the switching table is determined by the constructed set of label switched paths to be used in the network. The objective of this work is the synthesis of a set of label switched paths for an MPLS+ network that minimizes the size of the switching tables, while providing all-to-all shortest-path connectivity in the network. Each packet in the network requires one query of the routing table at the beginning of its route and then one query of the switching table at every node it visits along that route. Therefore minimizing the size of the switching tables may significantly improve performance of the network.

Afek & Bremler-Barr [1] claim that in order to achieve all-to-all shortest-path connectivity in MPLS+ (which they call Trainet), the number of labels needed to be recognized by a node equals to the number of leaves in the tree of shortest-paths rooted at that node. We first demonstrate that further reduction in the number of labels for MPLS+ may be achieved by allowing structures other than trees of shortest paths. Furthermore, not every collection of paths can be used in MPLS+. In order for such a collection to be compatible with the MPLS+ paradigms it needs to satisfy some consistency conditions that we define.

In section II of this paper we define the consistency conditions needed to be satisfied by a set of paths in order to be efficiently used in MPLS+. In section III we describe the construction of a minimal set of shortest-paths (not necessarily trees) emanating from one node (the root) and reaching all other nodes of a network. We conjecture that the construction of an optimal MPLS+ consistent collection of paths is an NP-Hard problem. In section IV we show that such a collection is feasible for some canonical networks (e.g. tree, grid, concentric rings) and we describe a distance-vector heuristic for the construction of sub-optimal consistent structures for general topology networks. We conclude the paper in section V.
II. CONSISTENCY IN MPLS+

We model the underlying communication network as an undirected connected graph $G = (V,E)$ where $V$ denotes the set of nodes and $E$ denotes the set of links or edges. We will use the terms network and graph interchangeably.

Given a network $G$ we want to synthesize a set of paths that will be used for routing packets using the MPLS+ paradigms. We define for each node $u \in V$ a set of directed paths emanating from $u$. Let $P^u = \{ p^u_1 , p^u_2 , ..., p^u_{l_u} \}$ be this set of paths where $l_u$ is the number of paths in $P^u$. Path $p^u_i = x^u_{i,0}x^u_{i,1}x^u_{i,2}...x^u_{i,j} \delta(p^u_i)$ is the $i$th path in $P^u$, $x^u_{i,j} \in V$ is the $j$th node along that path ($x^u_{i,0} = u$), $\delta(p^u_i)$ is the path length and $x^u_{i,j-1}x^u_{i,j} \in E$ for all $u \in V$, $i = 1...l_u$ and $j = 1...\delta(p^u_i)$. For all-to-all communication we want each node in the network to be reachable from $u$ by at least one of these paths. In order to provide shortest-path connectivity we require that each path $p^u_i$ is a shortest-path between $u$ and its endpoint $x^u_{i,\delta(p^u_i)}$. This is obtained when $d(u,x^u_{i,j}) = j$ where $d(\cdot,\cdot)$ is the distance (hop-count) between two nodes. We define $\sigma(p)$, the suffix of a length $d$ path $p$, to be the length $d-1$ path resulted from removing the first node of $p$. We also define $\psi(p,k)$, the $k$-prefix of $p$, to be the length $k$ path resulting from removing the $d-k$ last nodes of $p$. A path may have length zero, in which case it consists of a single node. Finally, we say that a collection of paths $P$ is consistent if for each path $p \in P$ there exist a path $q \in P$ such that $\sigma(p) = \psi(q,\delta(p) - 1)$.

Consider the collection $P = \{ p^u_1 , p^u_2 , ..., p^u_{l_u} \}$. Assume that each path is a shortest path and that for each pair of nodes $u,v \in V$ there is a path $p^u_v \in P$ that reaches $v$. We have the following:

Theorem 1: If $P$ is consistent then $l_u$ labels are sufficient at node $u$ in order to provide all-to-all shortest-path connectivity using the MPLS+ paradigms.

Proof: We will prove this by constructing a switching table with $l_u$ entries, as well as a routing table, for each node $u$ in the network. Assume $P$ is consistent. The switching table for node $u$ is constructed as follows: entry $i$ has $w$ as the next-hop node and $j$ as the output label identifier, where $w = x^u_{i,j}$ and $\sigma(p^u_i) = \psi(p^u_j,\delta(p^u_i) - 1)$, the existence of which is guaranteed by the consistency assumption. The entries for destination node $v$ in the routing table of node $u$ include output label identifier $j$, next-hop node $w$ and label hop-count $d - 1$, whereas $j$ and $w$ are the output label identifier and next hop-node associated with label $i$ in the switching table such that $v$ is a node along $p^u_i$, $x^u_{j,d} = v$. Using the principle of optimality, it can be verified that this provides all-to-all shortest-path connectivity. □

Internet routing protocols that support hop-by-hop routing (without labels), such as BGP [7] and RIP [8], only specify the exchange of reachability information between peers. Once a node has this information, it needs to choose for each destination node what would be the next-hop node used to reach that destination. The choice is limited to the set of neighbours that are along a shortest path to that destination. Once each node decides what the next-hop node to reach each destination node in the network is, based on some local policy, a set of paths is implicitly determined. Assume a global, fully ordered, set of distinct node identifiers. We have the following theorem, for which the proof is omitted:

Theorem 2: If the policy used by each router is to choose the next-hop node to reach a destination as the node with the smallest node identifier among the set of neighbours that are along a shortest path to that destination, then the resulting set of paths is consistent.

The construction given in theorem 2 is useful in providing a consistent set of paths with very little effort at each node. We denote by $L_{\text{RIP}}$ the vector whose $i$th element is the number of paths emanating from node $i$, $\in V$ in the set of paths resulted from this construction. Notice that $L_{\text{RIP}}$ depends on the actual numbering of the nodes.

III. OPTIMAL PATHS FROM A SINGLE NODE

As mentioned before, a tree of shortest-paths is not necessarily the optimal structure for paths in MPLS+. We demonstrate this by using the “fish” graph in Figure 1. Table 1 lists the optimal MPLS+ consistent set of shortest-paths for this graph. Notice that the paths emanating from node 1 do not form a tree. In fact, if we had limited ourselves to shortest-paths that do form a tree, the number of labels required at node 1 would have been 3, while our construction shows that 2 labels suffice.

We define an algorithm to find the minimum number of shortest-paths, rooted at a single node, that reach all other

Figure 1: “Fish” Graph

2
nodes in the graph. We first define $R_G(u,v)$, the radial subgraph of $G$ from node $u$ to node $v$, as the subgraph of $G$ that includes all nodes and edges of $G$ that are along any shortest-path from $u$ to $v$.

**Algorithm 1**: Let $u$ be the root node. In stage 0 of the algorithm we tag $u$ with the number 1. In the subsequent stages we shall tag nodes of distance $d$ from $u$, incrementing $d$ from 1 until all nodes are tagged. The tagging is determined as follows: At stage $d$ we construct a bipartite graph $H$ with node sets $V_d$ and $V_r$. $V_d$ is a subset of $V$ that includes all nodes of distance $d$ from $u$. $V_r$ includes a node for each tag already used to mark nodes of distances up to $d-1$ in previous stages. $H$ includes an edge between $v \in V_d$ and $t \in V_r$ iff $R_G(u,v)$ contains all nodes that are already tagged with $t$. Next we run a maximum matching algorithm [3] on $H$ and tag each matched node $v \in V_d$ with the tag $t$ it was matched to. If any nodes in $V_d$ remain untagged (unmatched) then we tag them distinctively with new tags. Then a new shortest-path is extended from $u$ to each node in $V_d$ which is unmatched, while for each node $v \in V_d$ that is matched in $H$ we take the path that reaches some node $v' \in V$ that has the same tag as $v$ and is closest to $v$, and stretch this path (i.e. elongate it) with a shortest-path to $v$. The algorithm ceases when all nodes are tagged. At that point, the number of paths equals the number of used tags.

**Theorem 3**: Algorithm 1 produces (in polynomial time) a minimal set of paths from $u$ to all other nodes in $G$.

**Proof**: Since we construct shortest-paths, the number of paths reaching all nodes up to distance $d$ is independent of the topology of $G$ beyond distance $d$ from $u$. Therefore, it is enough to show that the number of newly introduced paths in each stage is minimized. This is true because the number of newly introduced paths in stage $d$ equals the number of new tags in that stage, which is $|V_d|$ minus the number of stretched paths. Since the number of stretched paths is maximized by the matching algorithm, the number of newly introduced paths is minimized. Since matching is a polynomial time algorithm [3], and so is the construction of $R_G(u,v)$, the whole algorithm can be executed in polynomial time.

Let $\lambda(u)$ be the minimum number of shortest-paths rooted at $u$ that reach all other nodes in the graph, then $\lambda(u)$ is a lower bound for the number of labels required at node $u$ in an MPLS+ network. It is also a lower bound on the number of paths emanating from $u$ in a consistent set of paths for $G$ that provides all-to-all shortest-path connectivity. Let $\Lambda(G)$ be the vector whose $i$th element is $\lambda(v_i)$, $v_i \in V(G)$.

### IV. MPLS+ Optimization

In this section we combine the requirement for consistency together with the objective of minimizing the size of the switching table at each node.

**The MPLS+ Optimization Problem**

Let $L(P)$ be a vector whose $i$th element is the number of paths in $P$ emanating from node $v_i \in V$. The MPLS+ optimization problem is the problem of minimizing some norm of $L(P)$ given $P$ is a consistent set that provides all-to-all shortest-path connectivity. A lower bound for the norm of $L(P)$ is the norm of $\Lambda(G)$. A graph is said to be MPLS+ perfect if there is a set of paths $P$ such that $L(P) = \Lambda(G)$.

**Examples**

Here we give some examples of graphs that are known to be MPLS+ perfect and describe the construction of their optimal consistent set of paths.

**a. Graphs with unique shortest-paths**: These include trees, as well as any network that is made of a tree plus one edge that forms an odd size cycle, and other special cases. For trees, the size of the switching table for each root is the number of leaves in the tree (minus one if the root is itself a leaf).

**b. Graphs with diameter up to 2**: For each node we independently construct a minimal set of shortest-paths that reaches all other nodes, using, for example, algorithm 1. Each path in the set is of length 1 or 2. If it is of length 1, say $x_0x_1$, then its suffix is a 0-prefix of any path leaving $x_1$. If it is of length 2, say $x_0x_1x_2$, then since $x_2$ must have a path to its neighbour $x_2$ then the suffix $x_1x_2$ is a 1-prefix of such a path. Therefore the set is consistent.

**c. Rectangular grids**: See, for example, the 4 by 7 rectangular grid in Figure 2. The consistent set of paths is described in Figure 3 as a collection of 28 trees of shortest-paths where the sets of unique paths from the root of each tree (marked as a dark node) to its leaves make up the complete set of 158 paths. It can be verified that this set is consistent and that the number of paths for each node achieves the lower bound, therefore the grid is indeed MPLS+ perfect. Similar constructions for any arbitrary $m \times n$ grid can be made, showing that all rectangular grids are MPLS+ perfect. The details of these constructions are omitted.

The size of the switching table for an $m \times n$ grid, denoted $R_{\text{swt}}$, ranges, from node to node, between $\min(m,n)$ and $2 \cdot \min(m,n)$, compared to $m \cdot n$ in MPLS.
**d. Concentric Rings:** See, for example, 4 concentric rings with 6 nodes in each ring in Figure 4. A subset of the consistent set of paths is shown in Figure 5, the rest of the paths are symmetric rotations of those given. Again, any arbitrary graph of concentric rings can be shown to be MPLS+ perfect.

The size of the switching table for $r$ rings with $n$ nodes per ring, denoted $C_{r \times n}$, ranges between $\min(r, n)$ and $2 \cdot \min(r, n)$, compared to $r \cdot n$ in MPLS. When $n \geq 2r$ the size of the switching table is exactly $2r$ at each node. For example, in a single ring topology the size of the switching table is 2, one label used for routing packets in the clockwise direction and the other for counter-clockwise.

**Consistent paths in general topology graphs**

Here we describe a distributed heuristic algorithm for the construction of a consistent set of paths in general topology graphs. The algorithm is based on synchronous advertisements made by nodes to their neighbours, and it operates in a fashion similar to distance-vector algorithms for the construction of trees of shortest paths.

**Algorithm 2**: Cycle $d$ of the algorithm consists of two stages. During the first stage, each node $u$ advertises to its neighbours the set of paths it has constructed so far, denoted $P_d^u$. Paths in $P_d^u$ have lengths up to $d$. The second stage starts at each node once the node has received advertisements from all of its neighbours. During this stage, the node constructs a new set of paths by copying and stretching paths it had in the previous cycle and, when necessary, by introducing new paths. Node $u$ builds a bipartite graph $H_d^u$ with node set $V(H_d^u) = V_{d+1}^u \cup \Pi_d^u$. The set $V_{d+1}^u$ is a subset of $V$ that includes all nodes of distance $d + 1$ from $u$. The set $\Pi_d^u$ includes one node for each path in $P_d^u$ (we refer to nodes in $\Pi_d^u$ and paths in $P_d^u$ interchangeably). The bipartite graph $H_d^u$ includes an edge $(p, v)$ between $p \in \Pi_d^u$ and $v \in V_{d+1}^u$ iff there is a path $p'$ advertised by a neighbour of $u$ that reaches $v$ such that $\sigma(p') = \psi(p', d - 1)$. Next, a maximum matching $M_d^u$ is found for each bipartite graph $H_d^u$. The matching $M_d^u$ has node-set $V(M_d^u)$ and edge-set $E(M_d^u)$. Finally, each node $u$ constructs a new set of paths $P_{d+1}^u$ as follows:

$$P_{d+1}^u = P_{d+1, a}^u \cup P_{d+1, b}^u \cup P_{d+1, y}^u \cup P_{d+1, \delta}^u \cup P_{d+1, e}^u$$

where

- $P_{d+1, a}^u$ includes one path of the form $pv$ for each edge $(p, v) \in E(M_d^u)$ (matched pairs);
- $P_{d+1, b}^u$ includes all paths $p \in P_d^u$ that are not connected in $H_d^u$ to any node in $V_{d+1}^u$ (unmatched paths that cannot be stretched);
- $P_{d+1, y}^u$ includes any path in $V_{d+1}^u$.

**Figure 2:** $R_{4 \times 7}$

**Figure 3:** Optimal consistent paths for $R_{4 \times 7}$

**Figure 4:** $C_{4 \times 6}$

**Figure 5:** (Subset of) optimal consistent paths for $C_{4 \times 6}$
\( P_{d+1,\gamma} \) includes one path of the form \( pv \) for each path 
\( p \in \Pi^u_d \) which is not in \( V(M'_{d}) \), where \( v \) is picked arbitrarily such that \( (p,v) \in E(H^u_d) \) (unmatched paths that can be stretched);
\( P_{d+1,\delta} \) includes one path of the form \( pv \) for each node \( v \in V_{d+1} \) which is not in \( V(M'_{d}) \), where \( p \) is picked arbitrarily such that \( (p,v) \in E(H^u_d) \) (unmatched distance \( d+1 \) nodes); and
\( P_{d+1,\varepsilon} \) includes one path of the form \( up' \) for each node \( v \in V_{d+1} \) which is disconnected in \( H^u_d \), where \( p' \) is a path from a neighbour of \( u \) that reaches \( v \).

In cycle zero each node advertises itself as a length zero path and then paths of length 1 are constructed.

Notice that \( P_{d+1,\gamma}^u \cap P_{d+1,\delta}^u = \emptyset \) because of the maximality of \( M'_{d} \). Also, only paths constructed in \( P_{d+1,\delta}^u \) and \( P_{d+1,\varepsilon}^u \) are new paths, while the rest of the paths in \( P_{d+1,\gamma}^u \) are paths utilized from \( P_{d}^u \) by either copying or stretching. It is therefore desired to minimize the cardinality of these two sets of paths. For \( P_{d+1,\varepsilon}^u \) this is achieved by the maximal matching. Minimizing the cardinality of \( P_{d+1,\varepsilon}^u \) can only be done with a “forward looking” algorithm, i.e. one that uses information on the structure of \( G \) beyond distance \( d \) from \( u \). We leave this for further study in the future. And last, while paths constructed in \( P_{d+1,\gamma}^u \) are stretched to nodes already reached by paths in \( P_{d+1,\delta}^u \), these paths do not increase the cardinality of \( P_{d+1,\gamma}^u \) and they may become useful in subsequent cycles.

**Theorem 3**: Algorithm 2 terminates with a set of paths 
\( \bigcup_{u \in V} P_{d}^u \) which is consistent and provides all-to-all shortest-path connectivity.

**Proof**: We show that \( \bigcup_{u \in V} P_{d}^u \) is consistent and provides shortest-path connectivity between all nodes of distance up to \( d \), for \( d = 1,2,...,\text{diam}(G) \), using induction on \( d \).

For \( d = 1 \) the proof is trivial. For \( d > 1 \) the paths are shortest-paths and the set is consistent by construction. The only issue is the existence of a path \( p' \) for each disconnected node in the construction of \( P_{d+1,\varepsilon}^u \), which is guaranteed by the induction assumption.

The key element of algorithm 2 is the fact that a node may not retrace paths it already advertised, as those may be used by its neighbours when they stretch their own paths. Therefore when a node advertises a path, this path may either be stretched or remain unchanged in the following cycle.

We denote by \( L_h \) the vector whose \( i \)th element is the number of paths emanating from node \( v_i \in V \) in the set of paths resulted from algorithm 2. Notice that \( L_h \), like \( L_{RIP} \), depends on the actual numbering of the nodes, because of the nature of the matching algorithm.

**Complexity and Performance**

The second stage in each cycle, which is the most time consuming stage, involves maximum matching in a bipartite graph and has complexity \( O(\sqrt{|V_d| \cdot |E(H)|}) \leq O(|V_d| \cdot |E(H)|) \) [3], therefore the whole algorithm is performed in a distributed fashion in time \( O(\sum_{d=1}^{diam(G)} |V_d| \cdot n^2) = O(n^3) \).

We tested the algorithm on several graphs. For each graph we examined the median and maximum value of the vectors \( \Lambda \), \( L_h \) and \( L_{RIP} \). In Table 2 we present the ratio between those and the size of the switching table for MPLS, when Destination Removal is not utilized, namely \( n \). As mentioned, the results for \( L_h \) and \( L_{RIP} \) vary for different numberings of the nodes. Here we present the results obtained with one random numbering for each graph. Recall that grids and concentric rings are MPLS+ perfect.

<table>
<thead>
<tr>
<th>Graph</th>
<th>diam(G)</th>
<th>( \Lambda/n )</th>
<th>( L_h/n )</th>
<th>( L_{RIP}/n )</th>
<th>#labels/n</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_{5,20} )</td>
<td>24</td>
<td>0.10</td>
<td>0.12</td>
<td>0.17</td>
<td>0.16</td>
</tr>
<tr>
<td>( R_{5,20}^b )</td>
<td>25</td>
<td>0.14</td>
<td>0.17</td>
<td>0.25</td>
<td>0.24</td>
</tr>
<tr>
<td>( R_{5,20}^c )</td>
<td>28</td>
<td>0.08</td>
<td>0.12</td>
<td>0.19</td>
<td>0.16</td>
</tr>
<tr>
<td>( C_{5,25} )</td>
<td>19</td>
<td>0.06</td>
<td>0.06</td>
<td>0.12</td>
<td>0.18</td>
</tr>
<tr>
<td>( G_{500,0.02} )</td>
<td>5</td>
<td>0.66</td>
<td>0.70</td>
<td>0.66</td>
<td>0.70</td>
</tr>
</tbody>
</table>

### Table 2: Performance of MPLS+ vs. MPLS

\( ^a \) MPLS+ perfect.

\( ^b \) \( R_{5,20}^b \) is a 5x20 grid where edges are randomly pruned with probability 0.1 per edge.

\( ^c \) \( G_{500,0.02} \) is a typical instance of a random graph with 500 nodes and 0.02 probability of an edge between every pair of nodes. Random graphs tend to have very small diameters [4].

In Table 2 we are most interested in the comparison between the two columns of \( L_h/n \) and those of \( \Lambda/n \) and \( L_{RIP}/n \), in order to assess the performance of algorithm 2, and between all these columns and the MPLS column, for the performance of MPLS+. The results demonstrate that MPLS+ is very efficient in reducing the size of the switching tables (compared to MPLS) when the diameter of the graph is large. Reduction factors ranging from 1:4 to 1:15 are obtained in these examples. This is because long paths may utilize the same label for many (source, destination) pairs of nodes.
V. CONCLUSIONS

In this paper we introduce MPLS+, an extension to MPLS that allows for significant reduction in the size of the switching tables, and consequently for improved performance and scalability, in label switching networks. We present lower bounds on the size of the switching table at each node in the network and show that this bound is strict. We introduce a heuristic algorithm for the construction of efficient MPLS+ consistent sets of paths.

We intend to further investigate Quality of Service and traffic engineering issues in MPLS+, as well as extensions of the MPLS+ optimization problem to hierarchical labeling (using a stack of labels instead of a single label).

REFERENCES