PAPER

On Constraints for Path Computation in Multi-layer Switched Networks*

Bijan JABBARI†, **Shujia GONG**†, ***Nonmembers***, *and* **Eiji OKI**††, ***Member***

SUMMARY This paper considers optical transport and packet networks and discusses the constraints and solutions in computation of traffic engineering paths. We categorize the constraints into prunable or non-prunable classes. The former involves a simple metric which can be applied for filtering to determine the path. The latter requires a methodic consideration of more complicated network element attributes. An example of this type of constraints is path loss in which the metric can be evaluated only on a path basis, as opposed to simply applying the metric to the link. Another form of non-prunable constraint requires adaptation and common vector operation. Examples are the switching type adaptation and wavelength continuity, respectively. We provide possible solutions to cases with different classes of constraints and address the problem of path computation in support of traffic engineering in multi-layer networks where a set of constrains are concurrently present. The solutions include the application of channel graph and common vector to support switching type adaptation and label continuity, respectively.

key words: *Multi-layer Networks, path computation, channel graph, routing, optical networks, Ring networks, ROADM, DWDM.*

1. Introduction

Packet networks using an underlying optical transport network are to provide a high degree of flexibility and economy if multi-layer switching capability is deployed. The switching technologies include but are not limited to layer 1 switching such as wavelength (lambda) waveband and fiber switching; layer 2 switching such as Ethernet and cell switching; sub-layer 3 or label switching such as Multi-Protocol Label Switching (MPLS), and layer 3 switching such as packet (e.g., Internet Protocol, IP) switching. All the above switching types of layer 1 may be modeled in space and time, while all switching types above layer 1 may be viewed as packet filtering. Nonetheless, in general a realization of these switching types within a network requires consideration of the specific type of switching on a per port basis at each node, hence adding a level of complexity to the network for multi-layer switching. This complexity to some extent can be mitigated through use of a control plane; one such control plane is the IETF developed set of standards under Generalized MPLS (GMPLS) [1].

GMPLS has the potential to become an integral part of Internet core networks [2] by providing end-to-end control, provisioning, protection and restoration in heterogeneous transport networks. The transport networks are already incorporating to a degree network automation and self-actualization. This is indeed needed to simplify bandwidth procurement, provisioning and management. Due to close interaction between these functions in transport network and packet networks to establish a traffic engineering (TE) path, a common control plane becomes increasingly an attractive proposition.

Link state protocols such as Open Shortest Path First (OSPF) has been enhanced to provide all the resource availability information of a TE-link. Therefore a single Traffic Engineering Database (TED) that integrates the latest topological and network state information is available for the Path Computation Element (PCE) [3] to search an optimal end-to-end path efficiently.

Multi-Region and Multi-Layer networks (MRN/MLN) are introduced in [4]. A region is defined as a switching technology domain and MRN is defined as a network of multiple switching types. MRN/MLNs bring in the switching capability constraint. In MRN/MLN, multiple switching technologies coexist and an LSP will traverse networks with different switching capabilities. Packet Switch Capable region (PSC), Layer-2 Switch Capable region (L2SC), Time Division Multiplex capable region (TDM), Lambda Switch Capable (LSC), and Fiber Switch Capable (FSC) have so far been defined. All the resource availability information is integrated into a single Traffic Engineering Database (TED) and GMPLS provides a comprehensive framework to control the cross-layered Label Switched Path (LSP) setup through vertical and horizontal interaction and integration in MRN/MLN. Therefore, efficient and optimal path computation across the whole MRN/MLN is enabled.

MRN/MLN may consist of single-switching-type-capable Label Switching Routers (LSR) and multi-switching-type-capable LSRs as defined in [4]. Simplex node and hybrid node are two types of multi-switching-type-capable LSRs, where simplex node is defined as a network element with different switching capabilities, but each interface has only one switching capability,

Manuscript received October 9, 2006.
Manuscript revised February 15, 2007.
Final manuscript received March 9, 2007.
†The author is with George Mason University.
††The author is with NTT.
*This work was partially supported by the National Science Foundation (NSF) under grant no. 0335230.

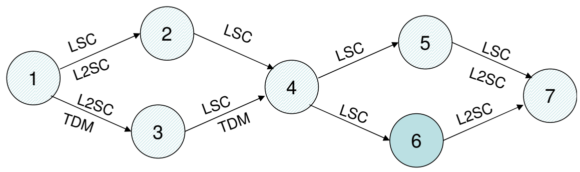

Fig. 1 Sample Multi-Region Network Topology

and hybrid node means that one interface of a network element have multiple switching capabilities, and thus can adapt between different switching types.

Figure 1 is a sample network topology which consists of three regions, i.e., L2SC, LSC and TDM. All nodes are hybrid nodes except that node 6 is a simplex node.

A Network Graph (NG) with n nodes and m links is defined by a digraph $G = \langle V, E \rangle$, where V is the node set $\{v_1, v_2, \ldots, v_n\}$ and E is the arc set. If a directed link connects the the head node $v_j \in V$ to the tail node $v_k \in V$ and $v_j \neq v_k$, we define the link as $e^i_{jk} \in E$, where index of the link $i = \{1, \ldots, m\}$ and we use the ordered pair $\langle v_j, v_k \rangle$ to represent the connectivity. Each switching type s_x on $\langle v_j, v_k \rangle$ can be denoted as $\langle v_j, v_k, s_x \rangle$.

A Channel Graph (CG) can be viewed as the dual of an NG. A node in the NG will be translated into several arcs in the CG, and $\langle v_j, v_k, s_x \rangle$ in an NG will be translated into a node N_{jkx} in the CG. The construction of the CG yields a $one - one$ mapping of an LSP in NG into a path in the CG. We define $H = \langle N, A \rangle$ as the CG transformed from NG, where N is the node set and A is the arc set.

In this paper we consider multi-layer networks and focus on constraints in traffic engineering path computation and solutions to support them. In the next two sections we discuss constraint classes of prunable and non-prunable and their solutions. This is followed by a discussion of channel graph and common vector for a non-prunable category of constraints referred to as additive and non-additive. Subsequently, we discuss the solution based on channel graph for path computation and its complexity. Finally, we provide some concluding remarks.

2. Constraints in Path Computation

Cross-layered search of path and the nature of the optical network add new constraints to the PCE. We summarize the constraints into two areas, i.e., prunable and non-prunable constraints.

Prunable metric means that the solution to such constraints is a simple filtering. All the network elements that do not have required features will be pruned before a path searching process. Bandwidth requirement is an example of prunable constraints.

Non-prunable constraints include the following two categories: additive constraints and non-additive constraints. Examples of additive constraints are attenuation and dispersion request of an optical signal. Wavelength continuity and its more general form referred to as label continuity and switching type adaptation are examples of non-additive constraints.

Label continuity constraint is different from the switching type adaptation in nature. In Multi-Region Networks (MRN), some network elements are multi-switching capable and they can adapt from one switching type to another. Unlike the wavelength conversion that can generally translate any incoming wavelength on any incoming fiber to any outgoing wavelength on any other fiber, adaptation function of multi-switching capable LSRs is interface specific and generally, the adaptation cannot be done from any particular switching type to any other particular type on an interface.

3. Prunable Constraints and the Solution

Prunable constraints include but are not limited to bandwidth requirement, policy constraint or protection requirement.

Some administrative policy may require that certain network resources should not be used for path setup. SRLG can also be a prunable constraints in such a way that the nodes and links of the same SRLG should not be used when searching disjoint paths.

Ref. [5] discusses a disjoint path selection scheme with Shared Risk Link Group (SRLG). Suppose that two optical paths, which are treated as different IP links in IP layer, share the same fiber using different wavelengths. We say that the two IP links belong to the same SRLG. If failure occurs at the fiber, both IP links will fail at the same time. Therefore, when we compute node/link disjoint paths, SRLG, which includes the lower-layer information, must be taken into account.

A solution to prunable constraints is known as CSFP (Constrained Shortest Path First), which is an extension of Dijkstra's algorithm. Link state protocol such as OSPF and IS-IS can broadcast the resource availability, such as residual bandwidth information of a link, based on which Path Computation Element (PCE) can build the TED. When an LSP setup request with the bandwidth requirement is made, the PCE can prune the links without sufficient bandwidth before running Dijkstra algorithm. CSPF can also be used to solve policy constraints or SRLG constraints, i.e., to pruned the restricted resource before path searching. Ref. [6] also uses a k-shortest path algorithm as the solution to SRLG.

4. Non-prunable Constraints

Not every network element in an optical network provides O-E-O conversion along the lightpath. In such a network, Routing and Wavelength Assignment (RWA) is subject to many constraints categorized either as additive, or as non-additive. Certain constraints, such as wavelength continuity, switching capability and bandwidth requirements, are non-additive, while constraints such as attenuation and dispersion are additive. These constraints make the lightpath setup in an optical network very different from circuit setup in a traditional circuit-switched network.

4.1 Additive Constraints

In DWDM networks, with the increase of bit-rate and the number of wavelength, and with the reduction of the channel spacing, optical impairments have a more significant impact on the routing scheme.

Typical constraints are attenuation, in which signal strength is reduced due to absorption and scattering; dispersion caused by dissimilar speed of wavelengths during light ray propagation; and inter-channel cross-talk due to the reduction of signal spacing. Constraints on attenuation, dispersion and delay are additive. Optimization of all the additive constraints at the same time is an NP-hard problem.

4.2 Non-additive Constraints

In this section we discuss the non-additive constraints. These include constraints such as label continuity (for example wavelength continuity in all-optical networks) and switching capability.

4.2.1 Label Continuity Constraint

Label continuity is important at least in two cases. First, wavelength continuity which is a non-additive constraint in all-optical networks where wavelength translation is not present at every or certain nodes across a path. For nodes without wavelength translation capability, the incoming wavelength and outgoing wavelength along the same path should be the same. Another case is the problem of global VLAN tag where a path should maintain its VLAN tag at the nodes across a path. In such networks, the label swapping is not available at every or certain nodes.

This constraint is considered to be non-prunable because we cannot delete those links without the required wavelength. Such links can also be part of the lightpath after wavelength translation occurs in some intermediate nodes.

4.2.2 Switching Capability Constraint

The switching capabilities of a node will be advertised in its TE-link LSA. This LSA contains Interface Switching Capability Descriptor (ISCD) and Interface Adaptation Capability Descriptor (IACD). IACD describes how different switching types can be adapted in this node.

An LSP request can specify 10 Gbps end-to-end connection with L2SC from node 1 to node 7. In this case, we cannot simply prune those links without L2SC because we can adapt the switching type in the intermediate nodes to LSC or TDM. We must also guarantee that when the flow reaches destination node, it must have been adapted back to L2SC.

An optimal path in MRN/MLN can be non-elementary, i.e., a simplex node A may need to be visited more than once in a path, but each visit will take a different switching type on a different interface.

5. Solution to Path Computation with Non-prunable Constraints

In this section, we consider the non-prunable constraints and discuss the solutions in support of path computation for when the constraints are additive and non-additive.

5.1 Additive Constraints

K Shortest Paths (KSP) is a solution to address additive constraints such as attenuation, dispersion and delay. Clearly, we cannot delete any intermediate results even if a threshold of a constraint is exceeded because it is hardly possible to determine which segment of the path caused constraint violation. Meanwhile that segment could appear in another path that satisfies all the constraints. These additive constraints only need to be satisfied rather than optimized. Therefore, we can find K Shortest Paths from the source to the destination and choose the one that can satisfy all the additive constraints at the same time.

Numerous publications have discussed the KSP which computes K shortest paths in the order of increasing length. Reference [7], [8] and [9] are three among many KSP algorithms. The algorithms provided in [7], and [8] allow cycles. Reference [8] provides a straightforward solution, *Recursive Enumeration Algorithm (REA)*, which does not delete any link or node to avoid cycles.

Yen's algorithm searches loopless K Shortest Paths. A new implementation of Yen's algorithm is presented in [9]. Yen's algorithm is more complicated than REA. Because it needs to delete nodes and arcs before the next shortest path is found, its efficiency is worse than REA. However, the path we can find is loopless. The complexity of YEN's algorithm, according to [9], is $O(Kn(m+nlogn))$, where n is the number of nodes and m is the number of arcs in the network graph.

Yen's algorithm cannot be applied directly to routing in MRN because it will produce unnecessary loops.

5.2 Non-additive Constraints

5.2.1 Label Continuity Constraint

A solution for this type of constraints is to consider a common vector where the elements of the vector represent the availability or the lack of the labels at a node across the path. The set of available labels can simply be determined by taking the logical AND across the vectors at each node on the path. A method such as Extended Indexing Ref. [10] can be used to facilitate the distribution of the labels.

The applicability of common vector to label continuity for VLAN tags is justified as there are 4096 tags available. A vector comprised of 4096 bits, with each bit indicating the status of one of the 4096 VLAN tags used at each node is an attractive solution for large Ethernet networks in practice.

Another approach proposed especially for wavelength continuity, when wavelength translation is present at certain nodes of a network, is wavelength graph. Reference [11] provides an approach to solve the wavelength continuity constraint by converting a Network Graph (NG) to a Wavelength Graph (WG). In WG, one plane is generated for each wavelength, and each node in the network graph is duplicated at each wavelength plane. For those nodes that have wavelength conversion capability, additional links are created to connect the replications of each node on corresponding wavelength plane. Virtual nodes are generated as dummy originating node and destination node respectively. They are connected to the replications of the true originating node and destination node respectively, and metric on those virtual links is assigned to be zero.

5.2.2 Adaptation Constraints

Our solution to the adaptation of interface switching type in multi-layer networks is based on the Channel Graph (CG). In this subsection we describe Channel Graph construction. We defer the discussion of path computation and its computational complexity to a separate section.

Let us denote the number of switching types on link $\langle v_j, v_k\rangle$ by s^{jk}. Denoting the size of N and A in $H =< N, A >$ as p and q respectively, we have

$$p = \sum_{i=1}^{n}\sum_{j=1}^{n} s^{ij}, \text{ where } s^{ij} = 0 \text{ if } \langle v_i, v_j\rangle \notin E \quad (1)$$

$$q \le \sum_{i=1}^{n}\left(\sum_{\langle v_j,v_i\rangle\in E} s^{ji} \times \sum_{\langle v_i,v_k\rangle\in E} s^{ik}\right) \quad (2)$$

The arc set of H is constructed as follows: If node v_k in G can transport or adapt switching type s_x in incoming link $\langle v_j, v_k, s_x\rangle$ to switching type s_y in outgoing link $\langle v_k, v_l, s_y\rangle$, an arc $\langle N_{jkx}, N_{kly}\rangle$ is added to arc set A. Here, s_x may be or may not be equal to s_y. Arc $\langle N_{jkx}, N_{kly}\rangle$ is mapped to node v_k in G.

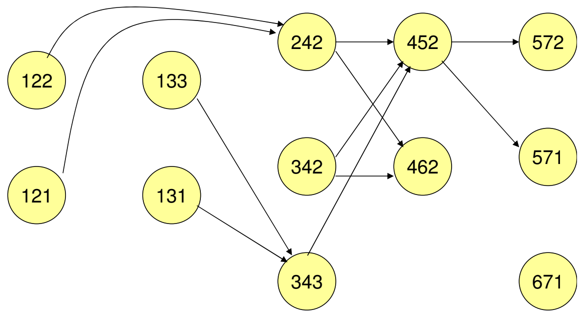

Fig. 2 A channel graph constructed from the network graph as shown in Fig. 1

Based on Fig. 1, Channel Graph H is obtained as in Fig. 2 as an example, where $V = \{v_1, \ldots, v_7\}$. We define s_1 as L2SC, s_2 as LSC and s_3 as TDM. Here, $e^1_{12} = \{\langle v_1, v_2, s_1\rangle, \langle v_1, v_2, s_2\rangle\}$, $e^2_{13} = \{\langle v_1, v_3, s_1\rangle, \langle v_1, v_3, s_3\rangle\}$, $e^3_{24} = \{\langle v_2, v_4, s_2\rangle\}$, $e^4_{34} = \{\langle v_3, v_4, s_2\rangle, \langle v_3, v_4, s_3\rangle\}$, $e^5_{45} = \{\langle v_4, v_5, s_2\rangle\}$, $e^6_{46} = \{\langle v_4, v_6, s_2\rangle\}$, $e^7_{57} = \{\langle v_5, v_7, s_1\rangle, \langle v_5, v_7, s_2\rangle\}$ and $e^8_{67} = \{\langle v_6, v_7, s_1\rangle\}$.

In this example, suppose v_1, v_2, v_3, v_4, v_5 and v_7 are hybrid nodes. Node v_2 can adapt from $\langle v_1, v_2, s_1\rangle$ to $\langle v_2, v_4, s_2\rangle$, v_3 can adapt from $\langle v_1, v_3, s_1\rangle$ to $\langle v_3, v_4, s_3\rangle$, v_4 can adapt from $\langle v_3, v_4, s_3\rangle$ to $\langle v_4, v_5, s_2\rangle$, and v_5 can adapt from $\langle v_4, v_5, s_2\rangle$ to $\langle v_5, v_7, s_1\rangle$. All the links are directional. Link e^1_{12} with switching type L2SC and LSC in G can be mapped to node N_{121} and N_{122}, respectively, in H. Similarly, we can add N_{131}, N_{133}, N_{241}, N_{342}, N_{343}, N_{451}, N_{461}, N_{571}, N_{572} and N_{671}. Links are created in H based on the connectivity and adaptation possibility in G as shown in Fig. 2.[†]

6. Channel Graph for Adaptation Constraints

6.1 Path Computation using Channel Graph

We now proceed to search for a path in H. The source node ID and destination node ID in an LSP request are node IDs in G. Therefore, to find a path in H, we need to convert node IDs in G into node IDs in H. Suppose we want to find a path from v_i to v_j with switching capability s_k. The source node in H can be any element in $N_s = \{\langle v_i, \cdot, s_k\rangle\}$, and destination node in H can be any element in $N_t = \{\langle \cdot, v_j, s_k\rangle\}$. We should notice that N_s is a set of nodes in H with the first component v_i and third component s_k. N_t is also a set of nodes in H with the second component v_j and

[†]Generally, GMPLS will setup bidirectional LSPs. To simplify the illustration, we assume that the LSP from node v_1 to node v_7 can only be set up along the directed links in Fig. 2.

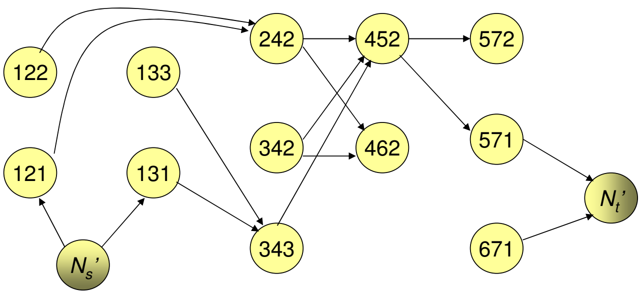

Fig. 3 Adding virtual nodes and links to Channel Graph

third component s_k. To simplify the path computation, we need to add two virtual nodes $N_s^{'}$ and $N_t^{'}$. We also add virtual links from $N_s^{'}$ to each element in N_s and from $N_t^{'}$ to each element in N_t to H.

Suppose we are seeking a path from v_1 to v_7 in NG in Fig. 1 with switching capability L2SC, we just need to run CSPF to find a path from $N_s^{'}$ to $N_t^{'}$ on H as shown in Fig. 3.

There are cases that may require KSP in multi-layer networks. An example is when there are optical impairment constraints.

If we take Figure 3 to run KSP, we can find two paths. The first path will be $L_1 = N_s^{'} - N_{121} - N_{242} - N_{452} - N_{571} - N_t^{'}$. The second path will be $L_2 = N_s^{'} - N_{131} - N_{343} - N_{452} - N_{571} - N_t^{'}$.

In YEN's KSP algorithm, if L_1 is the parent path, we need to find alternative path from N_{571} to $N_t^{'}$. We may find one if the network topology gets complicated enough. However, the newly found path will contain an unnecessary loop. This is because when we reach node N_{571}, we have already reached the destination node. Therefore, we need to modify the KSP algorithm in such a way that we stop searching alternative path from the predecessor of $N_t^{'}$.

The second reason that could cause an unnecessary loop when we run YEN's algorithm is that we may visit a node in G more than once with the same switching type on different interfaces. This is because different outgoing links with the same switching type in G will become different nodes in H. The solution is that when a node in H is deleted, all the nodes in H that map to the links with the same head node and switching type in G must also be deleted.

KSP is used in [5] to find disjoint path with SRLG in GMPLS network. To ensure the survivability of the network, we can also apply Oki's algorithm to channel graph. When we prune links $L(i,j)$ on the network graph, we need to prune all the nodes in channel graph that can be mapped to $L(i,j)$ in the network graph.

6.2 Computational Complexity

In this subsection, we discuss the computational complexity associated with applying the channel graph to solve the constrained path computation. We consider the following steps:

1. Construction of CG from NG. The computational complexity is $O(p+q)$, i.e., the number of nodes plus the number of arcs in H;
2. Searching KSP in H. The computational complexity is $O(Kp(q+plog(p)))$ according to YEN's algorithm; and
3. Mapping the path in H back to LSP in G. The computational complexity is $O(p)$.

Therefore, the overall complexity with this method is $O(Kp(q+plog(p)))$.

Breadth-first Search (BFS) can also find a path from source to destination by exhausting all the possible paths from the source until the destination node is found. Suppose we have n nodes and each node has a_i adaptation possibilities, the computational complexity of BFS is $\prod_{i=1}^{n}(a_i O(m+n))$. Clearly, the complexity is exponential and BFS becomes unscalable.

7. Concluding Remarks

We have classified the constraints in path computation in support of routing and traffic engineering. We have provided possible solutions to those cases with different classes of constraints. Our focuss here has been on multi-layer networks. However, the approach on multi-layer path computation described in this paper is also applicable to multi-area/multi-AS networks.

All the optical impairment, wavelength continuity and switching capability requirements can be satisfied. Channel Graph may be considered as a general approach to address the non-additive switching constraints at a node. This method is in particular efficient for switching type adaptation for which the number of switching types is limited. However, for label continuity the common vector method is preferred.

The solutions to path computation as discussed here lend themselves as good candidates for practical implementation. The proposed solutions for switching type adaptation and VLAN tag have been implemented as part of path computation in Dynamic Resource Allocation in GMPLS Optical Networks (*DRAGON*) project, an NSF sponsored project to create dynamic, deterministic, and manageable end-to-end network transport services for high-end e-Science applications.

References

[1] E. Mannie and Ed., "Generalized multi-protocol label switching (GMPLS) architecture," *RFC3945*.

[2] D. Awduche and B. Jabbari, "Internet traffic engineering using multi-protocol label switching (MPLS)," *Journal of Computer Networks*, vol. 40, no. 1, pp. 111–129, Sept. 2002, invited Paper.

[3] A. Farrel, J.-P. Vasseur, and J. Ash, "A path computation element (PCE)-based architecture," Aug. 2006, RFC4655.

[4] K. Shiomoto, D. Papadimitriou, J.-L. L. Roux, M. Vigoureux, and D. Brungard, "Requirements for GMPLS-based multi-region and multi-layer networks (MRN/MLN)," *draft-ietf-ccamp-gmpls-mln-reqs-02.txt*, Oct. 2006, Internet Draft (work in progress).

[5] E. Oki, N. Matsuura, K. Shiomoto, and N. Yamanaka, "A disjoint path selection scheme with shared risk link group constraints in gmpls networks," *IEICE Transaction on Communications*, vol. E86-B, no. 8, pp. 2455–2462, Aug. 2003.

[6] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*. Kluwer Academic Publishers, 1999.

[7] D. Eppstein, "Finding the K shortest paths," *SIAM Journal on Computing*, vol. 28, no. 2, pp. 652–677, Apr. 1999.

[8] V. M. J. Pelayo and A. M. Varó, "Computing the k shortest paths: A new algorithm and an experimental comparison," in *Proc. 3rd Int. Worksh. Algorithm Engineering (WAE 1999)*, ser. Lecture Notes in Computer Science, J. S. Vitter and C. D. Zaroliagis, Eds., no. 1668. Springer-Verlag, 1999, pp. 15–29.

[9] E. Q. V. Martins and M. M. B. Pascoal, "A new implementation of YEN's ranking loopless paths algorithm." *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, vol. 1, pp. 121–134, Jan. 2003.

[10] B. Jabbari, "Extended indexing method for resource assignment," *Internal Report*, Aug. 2006.

[11] I. Chlamtac, A. Faragoe, and T. Zhang, "Lightpath (Wavelength) routing in large WDM networks," *IEEE JSAC*, vol. 14, no. 5, June 1996.