MINIMIZING PROTECTION COST FOR HIGH-SPEED RECOVERY
OF MISSION CRITICAL TRAFFIC IN WDM MESH NETWORKS

Sun-il Kim, Xiaolan J. Zhang, and Steven S. Lumetta
CS and ECE Departments, Coordinated Science Laboratory
University of Illinois at Urbana-Champaign, Urbana, IL

ABSTRACT

Network survivability depends on the choice of recovery scheme and offers challenging tradeoffs in terms of recovery speed, cost, and management complexity. To allow more efficient operation of backbone networks, service differentiation based on reliability requirements have been proposed. However, the cost for protecting mission critical data using high-speed recovery schemes can still be expensive. We previously described an online protection scheme called Streams Protection that allows fast recovery and low data loss with little extra capacity. In this paper, we present a capacity optimization technique using the Streams Protection scheme for use with static traffic demands consisting of high-priority traffic. Unlike the use of flexible, dynamic provisioning, which is desirable for non-critical traffic (may be blocked depending on resource availability), high-priority, mission critical traffic must be guaranteed a high level of reliability and stability, which lends itself better to more static provisioning models. Our results show that using the proposed technique, we can achieve significant savings in protection capacity while providing the recovery speed and low data loss figures of a fully dedicated protection scheme.

INTRODUCTION

Network survivability is increasingly important in optical networks as the huge amount of bandwidth supported by the use of wavelength division multiplexing (WDM) technology exacerbates the impact of failures (more data loss in shorter amount of time). To guarantee reliability in the communications infrastructure, protection schemes have been used to provide recovery services. However, protection schemes with very fast recovery are expensive in terms of the amount of network capacity required to reserve backup for protection. In [1], classification of traffic based on different survivability requirements for the Department of Defense was specified to allow more flexible and efficient operation of optical networks, and to reduce this cost. Similar work in survivability based service classification that aims at reducing overall cost by differentiating services based on the level of reliability required by different classes of traffic can be found in [2, 3]. However, high-priority, mission critical traffic requires minimal downtime in the event of failures and utilizing dedicated protection to solve this problem can still be expensive and may adversely affect the available budget (and often a compromise is made by utilizing cheaper protection schemes that are slower with more data loss) [1].

A protection scheme that solves this problem by allowing fast recovery while reducing capacity cost was independently developed and introduced in [4, 5] under different contexts. In [5], we showed that this scheme, called Streams, can provide high-speed recovery and low data loss similar to dedicated protection in the context of online and dynamic provisioning scenarios while allowing savings in capacity cost. Another major advantage of the Streams protection scheme is that it can co-exist in the network with other standard protection schemes (such as dedicated or shared path protection) allowing service differentiated operation of networks similar to the methods outlined in [1, 2, 3] without additional equipment. Dynamic provisioning, as discussed in [2], is attractive because it allows flexible setup and tear down of connections as communication needs vary and is going to play a key role in how future networks are operated. However, for high-priority connections, it may be essential to perform more static allocation of resources to guarantee reliable, non-blocking communication. With the goal of minimizing capacity cost while providing fast recovery (and low data loss), we present an attractive and practical solution using Streams.

In this paper, we introduce a novel capacity optimization technique for use in provisioning high-priority traffic. We first introduce an integer linear programming (ILP) based optimization technique to minimize capacity cost. We then
present a novel heuristic to reduce the problem space and show that our heuristic is effective in achieving its goal. Simulation results show that protection cost can be reduced by a significant amount (36% or more) using our technique.

We first start with a brief overview of network survivability. We then provide a short description of the Streams protection scheme followed by a detailed description of our novel optimization technique. We then present performance evaluation results and provide some concluding remarks.

NETWORK SURVIVABILITY

The most common failure mode in optical networks is channel failures, often caused by the failure of a linecard at a port of an optical switch. Link failures, such as fiber cut, amplifier failures, etc., are also common, and lead to failure of all channels on the link. There is also the possibility of attack scenarios disabling links in a communications backbone. In this paper, we consider link failures, which cover channel failures scenarios as well. There are mainly two types of survivability schemes—protection and restoration—that are designed to address these failures [6]. The main tradeoff between the two are capacity cost and recovery speed. Restoration allocates resources after the failure occurs and therefore only uses the minimal amount of capacity. Protection on the other hand, utilizes pre-planned, pre-allocated resources to provide recovery, and therefore is more expensive, as protection capacity must be allocated for all connections. However, when network downtime and data loss can be critical, as is the case for optical connections carrying huge amount of bandwidth and for mission critical traffic, it is important to achieve high-speed recovery. Because protection schemes are preplanned, it is possible to achieve recovery in 50ms versus the slow recovery time of restoration schemes, which can take a few seconds or more [7, 8].

For high-priority traffic, protection is clearly more attractive, as fast recovery is one of the primary goals in designing survivability into the network for mission critical traffic. There are two basic types of end-to-end protection schemes that allow recovery time of 50ms or less—1+1 and 1:1 dedicated protection. 1+1 actively sends data over two diverse paths so that in the event of a failure on the primary path, the receiving nodes can simply listen on the backup path. It takes about 20ms to detect the failure and switch to the backup path. However, it is expensive to send out and actively drive two live paths. The 1:1 scheme, on the other hand, sends the data over the backup path only after the failure is detected. Because of the added propagation delay, it takes about 40-50ms for the data to reach the destination from the time of the failure. The recovery is considered to be complete when the data reaches the receiving end, but it is important to note that 1+1 and 1:1 both incur the same amount of data loss, which occurs during the time which the network is unaware of the failure (about 10ms detection) and until data is physically placed on the backup path (another 10ms). They both therefore have the same data loss characteristic [9, 10]. It is important to understand that exact timing is not as important, as the relative timing will remain the same since they utilize the same detection and switching mechanisms.

To reduce capacity cost, protection schemes that share backup resources can be used. Shared protection schemes offer significant reduction in capacity cost, but suffer from slow recovery times, incurring twice the amount of data loss (and slower recovery at around 100ms) [10]. In this paper, we focus on Streams protection, which can be thought of as virtually shared 1:1 protection with the advantages of both dedicated and shared protection schemes. We next discuss the details of the Streams protection Scheme.

PROTECTION WITH STREAMS

The original focus of Streams in [5] was on all-optical networks (AON) because of the key advantages AON’s offer over optically opaque networks. Faster data rates can be achieved with the absence of electronic/photonic processing that become a bottleneck on transmission time. At the same time, the absence of high-speed electronics may offer a significant reduction in equipment costs. In addition, unlike opaque networks, AON’s can handle signals with different data rates, communication/security protocols and formats, making it more suitable for supporting different classes of traffic with significantly different protocol requirements as well as future changes and upgrades. On the other hand, they suffer from limited functionality in wavelength conversion, signaling capabilities and detailed performance monitoring [11]. These limitations reduce the efficiency of many of the conventional protection schemes and make implementation more challenging [12]. We showed that Streams can be effectively utilized in the context of AON’s. We expect AON technology to play a key role in building reliable and cost-effective communications infrastructure in the future and focus on AON’s in this paper. However, it is important to note that our work is easily applicable to opaque networks as well. The only difference, in our algorithm, is the wavelength continuity constraint, which can be relaxed or eliminated.

The Streams protection scheme can be applied to any two-link connected network. It is like dedicated protection in the sense that all nodes are preconfigured when connections are provisioned, and in the event of a failure, backup traffic is simply sent over along the pre-established backup path, termed a stream. Preconfiguration enables the traffic to
be placed on the backup paths without any reconfiguration along the path, allowing faster recovery. All nodes along a stream are thus preconfigured to simply forward the backup traffic along the reserved wavelength channels ($\lambda$-channels) in a specific ingress to egress port setting (identical to 1:1 dedicated protection). Therefore, recovery with Streams is much faster than with protection algorithms that use soft-reserved backup capacity (such as shared path protection). The key difference between 1:1 and Streams is that we allow sharing of a stream across different connections. Each connection has an assigned backup path, which must lie entirely on a single stream and backup paths that are on the same stream cannot merge or diverge.

![Diagram](image)

Figure 1: (a) three primary paths. (b) Dedicated backup for primaries in (a). (c) Backup allocation using a stream.

Figure 1 illustrates how three connections, A-D, B-E and B-F (represented by primary paths shown in 1(a)) are protected by dedicated path protection (in 1(b) by A-B-C-D, B-F-E and B-E-F) and by Streams protection (in 1(c) A-B-C-D, B-C-D-E and B-C-D-E-F all on a single stream). Note that, in this example, the primaries cannot fail simultaneously due to link failures. It takes 3 $\lambda$-channels to allocate the primary paths (we assume each $\lambda$-channel has a cost of one). Likewise, the cost for using dedicated protection is 7 and the cost for Streams is 5. The bottom part of Figure 1(c) shows how the intermediate nodes along the stream are preconfigured to forward data. It is the same as dedicated protection in the sense that, upon failure, only the end nodes need to perform switching. For example, switch reconfiguration only need to take place at nodes B and E upon failure of connection B-E.

Capacity cost optimization
We now present our optimization technique based on an ILP formulation and also present our stream selection heuristic. For static traffic demands, protection capacity provisioning for Streams can be optimized by solving an ILP problem. In this section, we first present the ILP formulation and address several practical issues in finding optimal solutions. The protection capacity sharing optimization problem is in general difficult to solve as the problem space grows exponentially given the number of choices of backup paths for each connection. For practical applications, limiting this choice to few shortest backup paths while utilizing shortest primary paths has been proposed in the past [6]. Shared protection schemes are less constrained in the way backup paths need to be arranged compared to Streams, and therefore can relatively efficiently utilize the technique mentioned above. However, from our experience in working with Streams in the context of online routing, we noticed that allowing longer backup paths allowed for a greater reduction in capacity cost relative to shared protection. This phenomenon is true also for slower shared protection schemes, but they are affected much less by it due to their flexibility in sharing.

To address this problem, we utilize an approach that is somewhat opposite what has been done in the past—in the context of Streams, instead of merging backup paths into streams, we start with some number of streams and fit backup paths onto them. This approach also can suffer from the same problem as the number of possible streams (paths and cycles in a network) can be huge depending on the network. However, using our Q-stream selection algorithm, we are able to determine the potential efficiency of the streams and limit the problem by utilizing only a small percentage of the total number of the streams. This technique is presented later in the section.

![Diagram](image)

Figure 2: (a) Simple path-stream. (b) Simple cycle-stream. (c) Complex cycle-stream. (d) Complex path-stream.

In a simple graph (i.e., there is at most one edge between any two nodes), there are four types of streams—simple paths and cycles (nodes are repeated at most once for paths and at most once for all nodes except for the start/end node for cycles), and complex paths and cycles (nodes may be visited many times)—as shown in Figure 2. Whether a stream is simple or complex is determined by the topology of the subgraph that is induced by the stream. We denote a stream simple if and only if the induced subgraph is either a simple path or a simple cycle. The number of paths and cycles can also quickly grow as the network size and complexity grows, but in practice it is relatively easy to find for backbone network topologies. However, finding all complex streams adds a significant amount of complexity to this problem. In this work, we only utilize simple streams.

ILP Formulation
$E$ set of edges in the network.
$W$ set of available $\lambda$-channels on each edge.
$P$ set of primary paths. $p \in P, p \subseteq E$
A set of streams, \( S \), is defined such that \( s \in S, s \subseteq E \).

- **Feasible set for each primary path** \( F_p \) contains all streams that can be used to protect path \( p \in P \).
- **Feasible set for each stream** \( F_s \) contains all streams that can be used to protect stream \( s \in S \).

A set of tuples \( CS_e \) is defined as \( \{ (p \in P : s \in F_p) \} \).

- **Cost function** \( \text{len}_s \) is the length of the stream \( s \) in hops, which is simply the number of \( \lambda \)-channels required to allocate a stream.
- **Indicator for association between a primary and a stream** \( X_{ps}^{sw} \) is a binary variable that is set to 1 if stream \( s \) is allocated on wavelength \( w \) to protect the primary path \( p \) and 0 otherwise.
- **Indicator for allocation of a stream** \( Y_{sw} \) is a binary variable that is set to 1 if stream \( s \) is allocated on wavelength \( w \) and 0 otherwise.

The objective is to minimize the total cost for allocating protection capacity using Streams. Note that we assume that each \( \lambda \)-channel has a cost of one, but the cost can be easily modified otherwise by replacing \( \text{len}_s \) with an appropriate cost function. Ultimately, the solution to the ILP returns a set of \( Y_{sw} \)'s such that all primaries are covered with minimum total capacity cost.

\[
\text{Minimize } \sum_{s \in S} \sum_{w \in W} \text{len}_s Y_{sw} \tag{1}
\]

It is subject to the following constraints. The first constraint enforces that each primary path is assigned to one and only one stream for backup.

\[
\forall p \in P, \sum_{s \in F_p} \sum_{w \in W} X_{ps}^{sw} = 1 \tag{2}
\]

Each stream is considered occupied if at least one primary path has been assigned to it. The binary variable enforces that such assignment is only counted once when multiple primary paths utilize the same stream. Note that the objective function rules out solutions for which \( Y_{sw} = 1 \) and all \( X_{ps}^{sw} \) are 0 for some \( s, w \) pair.

For all \( p \in P, s \in S, w \in W \),

\[
Y_{sw} \geq X_{ps}^{sw} \tag{3}
\]

Next, conflicting primaries incompatible with a stream cannot be allocated on the same \( \lambda \)-channel of that stream. Given a set of conflicting primaries and a stream \( s, w \), it should not be used by all of the primaries in the set. At most \( n - 1 \) connections can share the same stream, where \( n \) is the number of total primaries in the set. Note that it does not imply that any strict subset of the primaries is incompatible with the given stream. A subset of the primaries can share the stream if this subset and the stream are not in the set \( C \).

\[
\forall (p_1, p_2, \ldots, p_n, s) \in C, w \in W, \sum_{s \in C \setminus CS_e} Y_{sw} \leq n - 1 \tag{4}
\]

Conflicting streams that have any edge(s) in common cannot utilize the same \( \lambda \)-channel for any of their edges. In other words, each wavelength can only be assigned to at most one of the streams in the conflicting set.

\[
\forall e \in E, w \in W, \sum_{s \in C \setminus CS_e} Y_{sw} \leq 1 \tag{5}
\]

Wavelength continuity constraints are implicitly satisfied by the use of the binary variables. Allocation of each stream is done on a per-wavelength basis instead of allocating capacity on a per-edge basis. This constraint forces all edges on the stream to utilize the same wavelength.

**Finding the Incompatibility Set \( C \)**

In most cases, a single stream can protect multiple primary paths individually (i.e., all primaries in \( F_s \) are protected by \( s \)). However, not all primaries in \( |F_s| \) can be protected simultaneously by \( s \). A set of primaries that can share stream \( s \) simultaneously are said to be compatible (formal definition is provided in the Appendix). Determining the compatibility of a stream for sets of primaries is important for solving the ILP.

![Figure 3](image-url)

(a) Two compatible primaries. (b) Incompatible primaries. (c) Incompatible primaries.
be protected simultaneously by the stream shown in solid gray curve. This stream can protect the two primaries under any single link failure, including the common edge (1,4) since they utilize different parts of the stream. However the primaries shown in Figure 3(b) are not compatible, as the failure of edge (1,4) requires the two primaries to utilize the same part of the stream. The stream shown in Figure 3(c) cannot protect the three primaries (0-4-6, 1-4-6 and 1-4-3) together even though they are pair-wise compatible. In other words, any two combination of the three primaries can be simultaneously protected by the stream, but not all three can be protected.

It is possible to find all incompatible set $C$ for each stream $s$ by checking the compatibility of all subsets of primaries in $F_s$. For each subset, an exhaustive search can be performed recursively. Since a compatible set must have all its subsets compatible, a bottom up approach can be used with memoization to keep track of the states of the subsets. First, for every pair of primary paths in $F_s$, we check the pair-wise compatibility on the stream. If they are not compatible, the set is added to $C$. If at least one two-primary combination passes the check, we can check the compatibility of every three primary combination that includes this two-primary combination that passed the check. The same step is repeated for all combinations of primaries. The number of primary paths a single stream can protect is at most $|E|$, but in practice this number is smaller. However, the search space can still grow rapidly when the set size is large. This naive approach requires $O(2^k), k = |F_s| \leq |E|$ checks in the worst case.

For simple streams we found that, for compatibility checks of a set of primaries $R \subset F_s, |R| > 3$, checking all subsets $T \subset R, |T| \leq 3$ is sufficient. Therefore, after checking all 2 and 3 combinations of primaries in $F_s$, a set of primaries of size 4 or greater is compatible with the stream if and only if any subset of size 3 is compatible. This result directly follows Theorem 1 presented in the Appendix along with a proof. By using this property of simple streams, the complexity can be reduced to polynomial $O(k^3)$.

**Stream selection and Q-streams Heuristic**

The ILP solution time depends strongly on the number of streams considered. Thus we tried to efficiently reduce this number without impacting solution quality. After finding all streams in a network, we first to discard streams that do not provide protection for any primary paths for the given traffic demand. Streams that are left after this step are called *valid streams*. However, given a reasonable number of demands, this step does not eliminate a significant number of streams.

If the total number of streams is large, we utilize the metric shown in Equation 6 to determine the quality of each stream and use a subset of streams called Q-streams based on their computed quality (Q).

$$Q_s = \frac{1}{\text{len}_s} \left( \sum_{i=1}^{|T_s|} i^2 v_{s,i} \right)$$

where $T_s$ is the set of primaries that stream $s$ can individually protect, $v_{s,i}$ is the number of $i$-compatible sets for stream $s$. Our metric basically measures how many combinations of primaries in $T_s$ a stream can protect with more weight given to compatible sets with higher number of primaries.

![Networks](image)

![Q-stream efficiency](image)

Three of the five networks shown in Figure 4 were small enough to allow the ILP to complete with all simple streams as input in a short amount of time (less than about an hour using the CPLEX software on HP Blade servers with AMD Opteron 2000 series processors and 2GB of RAM). The total number of simple streams are shown on the left in Figure 5 for all five networks. The right side of the figure shows the efficiency of our Q-stream selection heuristic. The horizontal axis represents the fraction of the total number of simple streams with the highest Q values used as input to our ILP, and the vertical axis represents protection capacity cost normalized to the optimal case where all simple streams were used. We also added the shortest backup paths to the set of streams used for the ILP when the shortest backup paths are not already included by using the Q-stream selection. This method allows the ILP to find more efficient solutions using a small fraction as it has the freedom to leave out some connections while attempting to protect as much primaries as possible with longer streams. The connectivity of NJLATA network is much higher compared to the vBNS and NSFNET networks, and the performance of Q-streams is better for NJLATA. For all cases, however, it shows that...
Table 1: Simulation Results.

<table>
<thead>
<tr>
<th>node deg.</th>
<th>vBNS</th>
<th>NSFNET</th>
<th>ARPA.NET</th>
<th>COST</th>
<th>NILATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.83</td>
<td>2.86</td>
<td>3.20</td>
<td>3.89</td>
<td>4.18</td>
</tr>
<tr>
<td># streams</td>
<td>1209</td>
<td>7370</td>
<td>198013</td>
<td>552105</td>
<td>9510</td>
</tr>
<tr>
<td># valid streams</td>
<td>±35.9</td>
<td>6169.1</td>
<td>±270.2</td>
<td>±7641.4</td>
<td>±24301.4</td>
</tr>
<tr>
<td># streams used for ILP</td>
<td>993</td>
<td>6169.1</td>
<td>±270.2</td>
<td>@5%</td>
<td>@2%</td>
</tr>
</tbody>
</table>

SUMMARY OF RESULTS

We used five well-known networks, shown in Figure 4, that are representative of optical backbone network topologies. 20 different provisioning scenarios for each network consisting of 30 randomly selected bi-directional connections for each scenario were performed. We used shortest robust primary paths—when topological shortest paths do not have a disjoint path, the shortest of paths that have disjoint paths must be used. vBNS required the use of shortest robust primaries that are longer than topological shortest paths. Our results are summarized in Table 1 along with 95% confidence intervals. The top portion of the table shows the average node degree, total number of simple streams, total number of valid streams, and the total number of streams actually used in the ILP. For the ARPANET and COST network, we utilize the Q-stream heuristic to reduce the number of streams using only 5% and 2% of the total number of valid streams respectively. The lower portion of the results table first show the capacity cost for provisioning primary paths only, followed by cost for using dedicated path protection, online greedy Streams with shortest backup path only and arbitrarily long backup paths (shown for reference only using algorithm presented in [5]). The results for greedy Streams are obtained by taking 30 random demands in routing them in random orderings, provisioning streams that greedily minimize cost. The results are averaged over 50 random orderings for each of the 30 random demands. Finally, capacity cost for using ILP optimized Streams is shown along with the amount of improvement over the greedy approach as well as the improvement over dedicated protection. The results for ARPANET and COST networks are obviously not optimal (given utilization of only 5% and 2% of the available streams), but our heuristic is efficient in that it allows the ILP solver to find good solutions that are comparable to improvements shown in other networks. Overall, it shows that a significant savings in capacity cost can be achieved using the technique presented in this paper.

CONCLUSION

In this paper, we presented a novel cost optimization technique for the Streams protection scheme that can be used to protect high-priority, mission critical traffic. Our results showed that a significant amount of capacity can be saved while maintaining the same level of recovery speed and data loss by using our optimization technique and Q-stream heuristic along with the Streams protection scheme. Although we focused on link failure protection, node failure protection can be covered by simply selecting node-disjoint paths for backup paths that lie on streams as shown in [5] rather than selecting link-disjoint paths.

REFERENCES

In this section we show that checking compatibility for sets up to size 3 is sufficient for simple streams (i.e., streams formed from simple paths or cycles). Let $s$ be a simple stream.

**Definition 1 (Primary conflict).** Two primary paths are said to conflict if they have an edge in common, and therefore can fail simultaneously when that edge fails.

**Definition 2 (Backup path).** A backup path (BP) of primary path $p$ on stream $s$ is a segment $U \subset s$ of the stream that connects the two end points of $p$. The set of BP's for $p$ is denoted $BP_p$.

**Lemma 1.** Any primary path $p \in F_s$ has exactly one BP on stream $s$, if $s$ is a not a cycle.

**Lemma 2.** Any primary path $p \in F_s$ has exactly two BP on stream $s$, if $s$ is a cycle.

**Definition 3 (Assignment).** Given a simple stream $s$ and a multi-set $R$ of primary paths, $R \subseteq F_s$, an assignment for $R$ is a function associating each path $p \in R$ with one of its backup paths on $s$.

**Definition 4 (Active backup path).** An active backup path (ABP) for primary path $p$ is the BP chosen for protection use among all paths in $BP_p$: $ABP_p$.

**Definition 5 (Compatible).** A multi-set $R$ of primary paths is said to be compatible on $s$ if and only if there exists an assignment of active backup paths for all primary paths in such that for any pair of conflicting primaries, the active backup paths assigned to these two primary paths are disjoint.

**Definition 6 (Compatible Assignment).** Given a stream $s$ consisting of a simple path (not a cycle), and a multi-set $R$ of primary paths, $R \subseteq F_s$, with $|R| > 2$, $R$ is compatible on $s$ if and only if for any subset $T \subset R$, $|T| = 2$, $T$ is compatible on $s$. Since each path has exactly one BP, assignment is unique and induced assignment has same compatibility.

**Theorem 1.** Given a stream $s$ consisting of a simple cycle and a multi-set $R$ of primary paths, $R \subseteq F_s$, with $|R| > 3$, $R$ is compatible on $s$ if and only if for any subset $T \subset R$, $|T| = 3$, $T$ is compatible on $s$.

**Proof.** Assume $R$ is compatible on $s$. Then, there exists a compatible assignment of $R$ on $s$. Let $T \subset R$, $|T| = 3$, and consider the assignment induced on $T$ by such compatible assignment of $R$ on $S$. If the induced assignment is not compatible, there exists two conflicting primary paths $p_1, p_2 \in t$ such that the BP’s assigned to $p_1$ and $p_2$ are disjoint. But, $T \subset R$, so $p_1, p_2 \in R$ and the original assignment of $R$ on $s$ is also not compatible (Contradiction). Thus, $T$ is compatible on $s$.

Assume $\forall T \in R$, $|T| = 3$, $T$ is compatible on $s$. Now assume that $R$ is not compatible on $s$. We will construct a set $T$ of three paths such that $T$ is not compatible on $s$. There must exist some path $p_1 \in R$ such that all of $p_1$’s backup paths overlap with some other conflicting path in $R$ for any assignment. $p_1$ has exactly two backup paths on $s$. If every path $p \in R$ has a backup path that does not conflict with any other backup path for any path in $R$, assigning the non-conflicting backup paths to each path produces a compatible assignment (contradiction). Thus, there exists some $p_1 \in R$ such that both of its backup paths conflict with backup paths for some other path in $R$.

Figure 6: (a) both backup paths of two primaries crossing (b) compatible assignment (c) incompatible assignment

Let $p_2 \in R$ conflict with $p_1$. There must exist an assignment of backup paths to $p_1, p_2$ so that there is no crossing between their ASP’s. Figure 6(a) illustrates the case where there is a crossing between the pair, and just the two primaries are not compatible. Therefore, the assignment must be in some general form of the example illustrated in Figure 6(b), where only one of the two backup path is viable for a compatible assignment due to the conflict between $p_1$ and $p_2$. Now let $p_3 \in R$ conflict with $p_1$, where $p_3$ may or may not conflict with $p_2$. Again, there must exist an assignment of backup paths to $p_1, p_3$ so that there is no crossing to allow $p_1$ and $p_3$ to be compatible on $s$. Again, the assignment must be as illustrated in Figure 6(c) and no viable backup paths exist. Thus, $T = \{p_1, p_2, p_3\}$ is not compatible on $s$. \[\square\]