Link Selection Schemes for Avoiding Channel Contention

Fang Yu[1], Rakesh K. Sinha[2], Robert Doverspike[2], Bruce Cortez[3], John Strand[2]
[1] Department of EECS, University of California, Berkeley
[2] AT&T Labs (Research), Middletown, NJ 07748,
[3] AT&T Labs, Middletown NJ 07748
E-mail: fyu@eecs.berkeley.edu, {sinha, rdd, jls}@research.att.com, bcortez@att.com

Abstract – **This paper describes methods to reduce channel contention during distributed setup of high-speed connections in a mesh network of Cross-Connects (XCs) [1]. A particular aspect of contention addressed here is when two different connection requests arrive on a pair of adjacent XCs and they choose the same restoration channels from spare inventory. Such a contention problem is often referred to as "glare". This paper gives channel selection methods that considerably reduce glare, while at the same time maintaining large groups of channels (for future high speed connections) across multiple, parallel links. We establish the benefit of our approach in a typical intercity backbone network. As shown in the simulation results, our channel selection mechanism can eliminate almost all the glare compared to the previously known algorithms, "Best-fit" and "Hi-Lo".**

Index Terms – **Channel selection algorithm, restoration, mesh restoration.**

I. Introduction

The current generation of transport network equipment provides the ability to set up explicit paths in the network by employing various kinds of signaling mechanisms. The specific architecture we will be considering uses XCs with STS-1 or STS-3 granularity electrical fabrics that are interconnected with OC-48 or OC-192 "links". These in turn are transported as wavelengths on fibers, each of which normally carries many wavelengths. There may be multiple links between two XCs; in this case the grouping of "similar" links (same XC end points, same fiber routing) is called a "logical link". The XC fabric allows the subdivision of each link into multiple TDM (STS-1) channels. Figure 1 illustrates a pair of XCs and two links connecting them. Examples of XCs include IP routers with MPLS capabilities and Digital cross connects. A few examples of signaling mechanisms for path set-up are RSVP, CR-LDP, and ATM signaling.

A connection (connection-oriented data streams or TDM circuits) is routed over a sequence of XCs and adjacent links. Each XC along the path routes the data contained in a connection from a set of incoming channels to a set of outgoing channels. The choice of which output channels to transfer the data depends on a combination of factors such as identity of incoming channels, header bits, timeslots, wavelengths, etc. We consider a SONET-like network where each channel belongs to at most one connection and all the channels assigned to a connection within a single aggregated link must be on a single link. However, channels belonging to a connection within a link do not have to be contiguous. To set up a connection, first the routing or the management system computes a sequence of XCs as a path. We use the term "path" to refer to the selection of a sequence of logical links, but not the selection of the specific link and channels to use within each logical link, which is most efficiently done by the XC's when they receive a path setup signaling message. That is, each XC on the path needs to select a link on its outgoing interface (e.g. Link 2 in the Figure 1) and a set of channels (e.g. the shaded channel in the Figure 1) within that link. This link and channel information is conveyed to the incoming interface of the next XC on the path for reservation.

The link selection problem is essentially a distributed version of one dimensional bin packing problem. Bin packing algorithms have been extensively studied in the literature [6, 7, 8, 9] and studies show that the "Best-fit" algorithm is generally most efficient for online applications. (The Best-fit algorithm chooses the link with the least amount of spare bandwidth that can accommodate the connection.) Therefore, most XC software adopts a Best-fit algorithm for link selection.

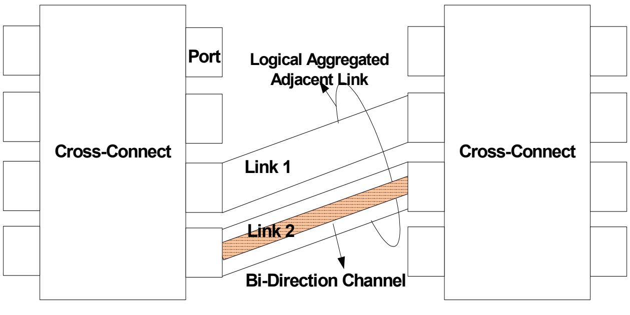

Figure 1 – Parallel links between two XCs.

A well-known protocol problem (known as "**glare**" [5].) arises when two connection-setup messages traveling in opposite directions may try to reserve the same bi-directional channels (even if other channels and links are available). The first setup attempt reserves a set of channels on one end of the link and meanwhile, the message traveling in the opposite direction reserves some of these channels on the other endpoint of the same link. Both messages, when they try to reserve the channels on their other endpoints, find that the channels have already been reserved by another connection and therefore return unsuccessful.

The primary goal of a link and channel selection mechanism is to arbitrate the process so that different setup attempts do not compete for the same channel. Another consideration is to minimize bandwidth fragmentation, that is, we want to preserve links with large amount of channels available so that they can accommodate large requests later. There are two well-known schemes for link selection. The ***Best-fit*** algorithm picks the link within an aggregate link with the smallest available capacity that can accommodate this connection. It minimizes fragmentation but exaggerates glare since the two ends are likely to select the same link. In the ***Hi-Lo*** algorithm [1], every pair of adjacent XCs predetermines an ordering of the links connecting them. Then one XC examines the links in the decreasing order of indices and picks the first link that can accommodate the connection. The other XC examines the links in the increasing order of indices and picks the first link that can accommodate the connection. This scheme works very well for glare but makes no attempt to minimize fragmentation.

Typically, there are two types of connection setup requests: (1) initial provisioning of a connection (2) restoration of failed connections over alternate paths when a network element (e.g., switch, fiber span, or DWDM equipment) fails. Usually, glare is not an issue for initial connection provisioning because the volume of provisioning/switched connection setups is quite low in transport networks when compared to the time to process a single setup. The main consideration during provisioning is to minimize bandwidth fragmentation. Thus we recommend using Best-fit algorithm for provisioning setup requests. However, Glare is most acute during restoration. This is because a node or fiber span failure may cause the reroute of a large number of connections in a small time frame, e.g., under 100ms, thereby creating a lot of contention.

This paper gives channel selection methods that considerably reduce glare, while at the same time maintains large groups of contiguous channels for future high speed connections.

This paper is organized as follows: Section 2 describes the channel setup process in detail. Section 3 introduces our new link selection algorithms and compares them to the Best-fit and the Hi-Lo schemes. Finally we present simulation results in Section 4 and conclusions in section 5.

II. A REPRESENTATIVE CONNECTION SETUP PROCESS

Both the service and restoration paths of a connection consume one or more channels (equal to its bandwidth) on each link along its route. The same set of channels needs to be reserved on both XCs that are adjacent to this link. As an example, Figure 2 shows setting up of a connection of 2 channels from XC 1 to XC 2. At XC 1, the signaling mechanism needs to do the following:

Step 1: It selects an outgoing link (L2 from port D) to the next XC (XC 2) on the path (either from all the links between the XCs or from a subset of links between the XCs).

Step 2: It selects enough channels (e.g., channels 1&2 from port D) inside the link (L2) to meet the bandwidth requirement of the connection and reserves those channels on the XC 1. If there are not enough channels for this connection, the path setup attempt fails.

Step 3: The signaling message then proceeds to the other endpoint of this link. On the other end point XC 2, it first needs to reserve corresponding channels (Channel 1&2 from port G) on the incoming link. If the channels are not available on the incoming link, this path setup attempt fails. If the channels are available, it makes the reservation and then selects an outgoing channel to the next XC on the path and so on.

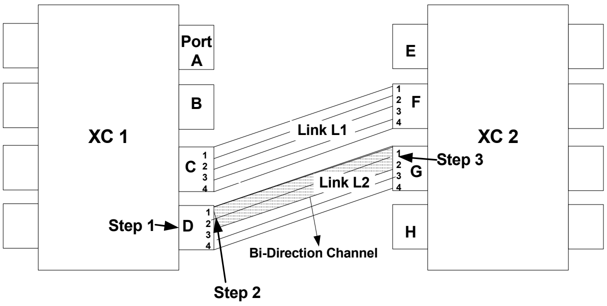

Figure 2 – Channel setup process between two XCs.

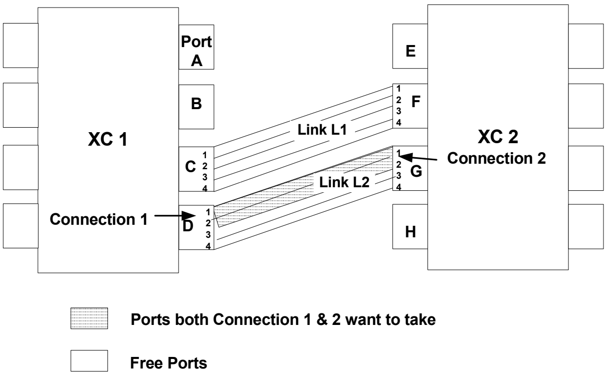

Figure 3 – Glare happens when two connections reserve the same channel from different ends.

Glare happens when two path setup messages traveling in opposite directions try to reserve the same channels. As shown in the above Figure 3, the first set-up attempt for connection 1 reserves channel 1 and channel 2 of link L2. While the first message is in transit, the message traveling in the opposite direction for connection 2 reserves one or both of the same channels on the other endpoint of the link L2. Both messages, when they try to reserve the channels on their opposite endpoints, find that the channels have already been taken by another setup process and therefore fail. Glare can happen even if there is enough capacity on the link (link L1 in the above example) for both connections.

One way to avoid glare is to use a master-slave relationship: for each pair of neighbouring XC interfaces, select one XC interface as "master" and the other as "slave". The master XC interface is responsible for selecting channels for connections traveling in either direction. If the connection is traveling from master towards the slave node, the master node can select links and channels and convey this to the slave node. For connections traveling from slave to the master node, an extra message conveying the request from slave to master and another message conveying the link/channel selection from master to slave are needed. Because of this additional signaling overhead, many equipment providers do not support this.

III. LINK AND CHANNEL SELECTION SCHEMES

For each connection setup request, we first select a link and then channels within that link. The choice of link and channel selection scheme can significantly affect the performance of the path set up process.

For channel selection, we recommend a previously known scheme.. Under our assumption, since the channels belonging to a connection do not have to be contiguous, there are no fragmentation issues in channel selection. To reduce "glare", one side of the link selects from the top, while the other side select from the bottom. This is clearly an optimal channel-selection scheme because "glare" will happen only if two connections from different ends are using up all the free bandwidth of the link.

For link selection, even if done in a centralized manner, the problem is equivalent to the (NP hard) bin packing problem. The distributed nature of the selection makes the problem even harder. So, we need to consider a heuristic solution. In this section, we first discuss Bestfit and Hi-lo schemes in detail, then present two new link selection mechanisms to reduce glare and bandwidth fragmentation.

3.1 Best-fit link Selection Algorithm

The canonical Best-fit algorithm is an algorithm to minimize the bandwidth fragmentation. Whenever new connection request comes, it always selects a link with the smallest available capacity that can accommodate this request. In this manner, the large contiguous channels on other links are intact and thereby reduce fragmentation. However, when there are connection requests from both end of the link simultaneously, choosing the minimal available link would greatly enhance the chances of both ends selecting the same link and causing a glare.

The Best-fit algorithm works as follows. Suppose there are K links between two XCs. Each link i has available bandwidth $aval[i]$. When a connection request arrives of size b bandwidth units, select the lowest indexed link, j, with $aval[j] \geq b$.

3.2 Hi-Lo link Selection Scheme

Contrary to the Best-fit algorithm which tries to minimize fragmentation, the Hi-Lo link scheme aims to minimize glare [1]. The basic idea of Hi-Lo is: Between any pair of XCs connected by one or more links, pre-select one of them as "Hi XC" and the other as "Lo XC". We also predetermine an ordering of the links. The Hi XC selects the highest indexed link, j, with $aval[j] \geq b$. The Lo XC selects the lowest indexed link, k, with $aval[k] \geq b$. Therefore, glare only happens if two connections are using up most of the channels in the all the links.

Note that the Hi-Lo XC selection involved in this scheme is different from the master-slave scheme described at the end of section 2. The Hi-Lo scheme does not require any signaling extensions and both XCs can make channel selection decisions at the same time. The selection of which node should be Hi can be simply based on the higher node id.

3.3 BestAndNextBest(BANB) link Selection Scheme

The Hi-Lo algorithm can help reduce blocking probability but may introduce fragmentation. The Best-fit algorithm can reduce fragmentation, but increase the probability of glare or contention.

An intuitive solution would be for one side to choose the best-fit link, while the other side chooses the second best-fit link. By using best-fit and second best-fit (BANB), we are avoiding fragmentation. This scheme can also eliminate glare if the granularity (bandwidth size) of all connections are the same, because the best-fit of one side can not be the second best-fit of the other side. Unfortunately, if the connections traveling in two directions have different granularity, then the two sides may still end up selecting the same link. E.g., in a SONET network, there can be multiple granularities of requests, e.g., STS-1, STS-3, STS-12, etc. For example, suppose the available channels, $\{avail[i]\}_{i=1,\ldots,3} = \{2,4,10\}$ STS-1 channels on three parallel links . Then the best-fit for STS-3 and the second best-fit for STS-1 are both link 2, resulting in a collision.

In our simulations, BANB scheme outperformed the Best-fit algorithm but did not do as well as our next proposed scheme when requests are of various granularities. However, we feel that this may still be a very good choice, especially if reducing fragmentation is the primary concern and the requests are of the same sizes.

3.4 *Interleave link Selection Scheme*

In this section, we propose a link selection policy that adopts different algorithms based on the type of path setup request. Because of the longer time scales, initial connection set-up (also called *provisioning*) is not usually subject to contention among multiple connections and, therefore, glare is not an issue; but, we want to minimize bandwidth fragmentation and we recommend using the Best-fit link selection scheme. For path setup requests during restoration, which have a stringent time requirement, we propose an Interleave channel selection algorithm to considerably reduce glare, while at the same time maintaining large groups of contiguous channels for high speed connections across multiple, parallel links.

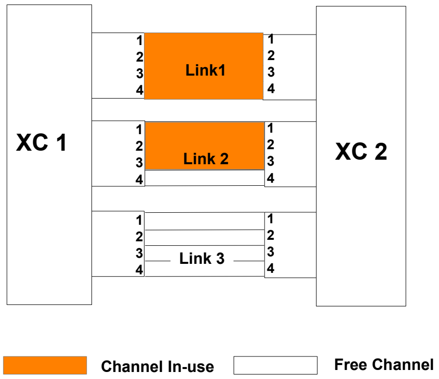

Figure 4 — Free bandwidth of all links after provisioning.

Before we introduce the Interleave scheme in detail, let's look at an example of how the free capacity of different links between a pair of XCs might evolve during the connection provisioning phase. For simplicity we assume that all links have the same initial available capacity in the following example. However, our scheme can be generalized as long as we know the initial distribution of the available capacities. For the first provisioning request, all links provide identical packing so it picks the first link. Now the first link has the smallest available capacity. By the property of Best-fit algorithms, the next several connections will all go to the first link. When the available capacity of the first link can't support a new request, the next connection provisioning request needs to go to the second link. The next several connections will be distributed among the first two links and so on. At the end of all connection provisioning, the available capacity of links will be roughly an increasing function of link index as shown in Figure 4. It is highly likely that the highest indexed links have no connections on them (e.g., Link3 in the example).

For connection restoration, if we run the Hi-lo scheme at this stage to avoid glare, we will fragment these high indexed links. A better solution would be to accommodate all connections in the first few links while still avoiding collision. The intuition behind the Interleave scheme is to run a modified Hi-Lo method on an ordering of links, where both nodes are likely to select among the first few links. Thus we order the links as 1,3, 5, …, *M*, *N*,…,6,4,2, where $M = 2*\lceil K/2 \rceil -1$ and $N = 2*\lfloor K/2 \rfloor$. The Interleave scheme pre-selects one XC as "Even" and the other as "Odd". The "Odd-XC" examines the links {1,3,5,..,*M*} and selects the link with the smallest available capacity that can still accommodate this request. If it cannot find a link, it examines the links in the order of {*N* ,...6,4,2} and selects the *first* link that can accommodate the request. The "Even-XC" examines the links {2,4,6, …, *N* } and selects the link with the smallest available capacity that can still accommodate this request. If we can not find a link, it examines the links in the order of {*M* ,...,5,3,1} and selects the *first* link that can accommodate the request.

The two XCs are considering links in opposite order while making a conscious effort to avoid high-indexed links. Following is the pseudo code for Interleave scheme.

```
Interleave Link Selection Algorithm for Restoration Path Setup
int link_selection( int b)
{
        if "Odd XC"
                up = 1; down = N;
        else if "Even XC"
                up = 2; down = M;

COMMENT: Try to find a best-fit link among the first K/2 links

        int Min = MAX_INT;
        int Opti_link = -1;
        for( i = up; i ≤K; i +=2)
          if(avail[i] ≥ b && avail[i]<Min){
                Min = avail[i];
                Opti_link = i;
                }
        if(Opti_link ≥ 0)
                return Opti_link;

COMMENT: Find first available link among the last K/2 links

        for( i = down ; i >= 1; i -= 2)
                if(avail[i] ≥ b)
                        return i;
        return -1;
}
```

Interleave Link Selection Algorithm

Figure 5 shows an example of how the Interleave scheme works. After the provisioning phase, which uses the Best-fit algorithm for link selection, the remaining capacities of all links are left in an increasing order. Suppose two connections come from different ends, each requesting 1 channel. According to the Interleave algorithm, connection 1 will try links in the order of 1, 3, 2, and pick link 3 since link 1 is full. Connection 2 will try link in the order of 2, 3 1 and pick link 2. In this way, two connections are arbitrated to different links. Otherwise, if using Best-fit algorithm, both connections will select link 2 thus potentially causing glare. So, Interleave heuristic is optimized to work with a Best-fit provisioning link selection scheme.

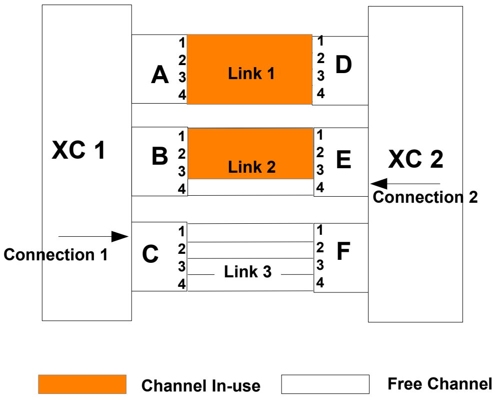

Figure 5 – An example of Interleave Scheme.

The Interleave scheme can reduce the chance of both ends selecting the same link, thus reducing glare. However, nothing short of a full handshaking scheme, like master/slave, can eliminate glare completely: For example, glare will still happen when both requests are relatively large, e.g., STS-12 and only one link between the two XC's has free bandwidth to accommodate these requests. Therefore, these two connections both select this link and glare may happen. In this case, although there is sufficient bandwidth for one of the two connections, both will get rejected because of the glare condition. However, this condition only occurs under many simultaneous conditions: 1) large bandwidth requests from different directions, 2) connection requests very close in time, and 3) insufficient bandwidth. Therefore, the probability is low.

The Interleave scheme can also achieve relatively low fragmentation. This is because when the restoration requests come in, the network typically has already set up a set of service paths using the Best-fit link selection algorithm. The natural effect of the Best-fit algorithm will be to leave the free bandwidth of the link in roughly increasing order and the interleave scheme roughly simulates two Best-fit selection among the low-indexed links. The advantages of the Interleave algorithm over Hi-Lo lies in the assumption that the Best-fit link selection scheme has been used during the service provisioning phase, which leaves available bandwidths in a certain order. This observation can be generalized. As long as we know the link selection mechanism used during the service provisioning phase and the expected distribution of available bandwidths, we can tailor the link selection mechanism during restoration to reduce fragmentation in addition to reducing glare.

The Interleave link selection scheme does not require any signaling extensions. It does examine more links than the Hi-Lo scheme. However, the number of links between a pair of adjacent nodes is typically of the order of ten and the overall restoration time is determined largely by number of restoration attempts, processing and transmission of signaling messages, and cross-connect set up time.

IV. PERFORMANCE EVALUATION

We ran two types of simulations. In the first set of simulations, we focused on one aggregate logical link. We report the performance of various link selection algorithms by varying several parameters, such as the number of links in the aggregate link. In the second set of simulation, we included these link selection algorithms as part of a large simulator that implements an MPLS like signaling protocol. In most of our studies, the Interleave scheme greatly outperformed Best-fit and BANB algorithm. In all cases, it performs at least as well as the Hi-Lo and Best-fit schemes. Since these schemes have roughly the same implementation complexity, the Interleave algorithm can easily replace the widely deployed Best-fit algorithm to achieve better performance.

5.1Evaluation Matrix

We use three figures of merits to study the efficiency of the link selection algorithms: bandwidth of rejected connections due to fragmentation, bandwidth of rejected connections due to glare, and the total failed bandwidth.

Note that the first two are mutually exclusive. When a connection gets rejected at the ingress interface of a XC, there are two possibilities. If the cumulative available bandwidth of the logical link cannot accommodate this connection, this is a result of bad capacity planning or poor routing decisions. This situation never arises in our simulation because we use the hybrid approach for capacity planning and restoration path selection. The other possibility is that the cumulative bandwidth is enough to accommodate this connection but is fragmented across multiple links. In this case, we mark this rejection as resulting from "bandwidth fragmentation." If the connection gets accepted at the ingress interface but then gets rejected at the other endpoint of this logical link, we mark this rejection as resulting from "glare". To evaluate the overall efficiency of an algorithm, we add the above two cases to get the total failed bandwidth.

5.2 Performance of link selection algorithms on one aggregated link.

In this simulation, we test efficiency of different algorithms on one aggregated link, which may consist of multiple OC-48 links. Requests are generated with different granularity: STS-1, STS-3, STS-12, STS-24, or STS-48. The distribution of requests is given as an input. During the simulation, we keep generating the requests until the total bandwidth of the requests exceeds the total logical link capacity and we discard the last request. Among these requests, we select the first 80% as service path setup requests and set them up according to best-fit link selection algorithm. The remaining 20% are taken as restoration requests. We generate less restoration request than service request because typical failure in the network won’ fail all service connections.

There are other three parameters that may affect the performance: *communication delay, processing delay*, and *average arrival interval*. Communication delay denotes the time stated as the time from when a message is put on the output queue at one XC until it is handed to the processing software at the receiving XC. We set it to 3 *ms*, except in the last simulation, where we study the effect of varying communication delay. Processing delay is the time used to process each request. In our simulation, we use 1 *ms* as processing delay. Average arrival interval records the time difference between two adjacent restoration requests. During a node or fiber span failure, a large number of connections must be rerouted to alternate paths to meet a network reliability objective. In order to model these “message storms”, we vary the average arrival interval to be 0 to 5 ms.

In the following simulation, we test the performance of three link selection algorithms under different scenarios. For each scenario, we report the average of 1000 independent runs.

5.2.1 Different distribution of requests

In this scenario, we generate request with different distribution of requests. The parameter values we used are: communication delay = 3 *ms*, average arrival interval= 0, processing time = 1 *ms*, logical link size =10 OC-48s. We first test the case when the requests are evenly distributed among all request granularities which is shown in the first line of table 1, 2 and 3. Note that the distribution is based on the ratio of the bandwidth request size, so 1:1:1:1:1 means all five bandwidth requests have the same probability. Then we change the distribution and put more requests of one kind at one time. For example, in line 2, we put 3 times more STS-1 than other request categories.

A. Bandwidth (Multiple of STS-1) of Failed Connections due to Glare

From Table 1 it is not surprising that the Best-fit algorithm has the largest number of glares because two connection setup requests will both select the link with smallest remaining channel to accommodate the request. Thus it is very likely that their selections will overlap and cause glare. The BANB is better than best-fit, but still it creates a lot of contention because the best-fit for STS-3 can still be the second Best-fit for STS-1.

Both Hi-lo and Interleave schemes have low glare rate, because connection from two sides will not select the same link unless necessary to accommodate both requests. However, as shown in Table 1, Interleave results in less failure due to glare compared with Hi-lo. This is because it creates less fragmentation, so the chances that only one link can accommodate the request are decreased. For example, suppose there is a logical link with 4 links with remaining capacity of (2, 12, 48, 48) after Best-fit service provisioning. Suppose there are three connections: connection 1 is an STS-12 request which comes to the LO end, and connections 2 and 3 are both STS-48 requests arriving at the same time at different ends. If we use Interleave, the first connection will take the 12 units of link 2 and the remaining bandwidth would be {2, 0, 48, 48}. When connections 2 and 3 come together at different ends, one will select link 3 and the other will select link 4. So, all the connections can go through. If we use Hi-lo algorithm, the first connection will select the last link since it arrives at the LO end and cause the remaining bandwidth to be {2, 12, 48, 36}. When connections 2 and 3 arrive at different ends, they both can only choose link 3 and thus cause glare.

STS-1:STS-3:STS-12:STS-24:STS-48	Best-fit	BANB	Hi-Lo	Interleave
1:1:1:1:1	31.264	21.151	16.606	14.122
3:1:1:1:1	30.137	20.336	13.227	11.422
1:3:1:1:1	31.03	20.377	14.037	11.565
1:1:3:1:1	35.966	18.28	12.236	11.359
1:1:1:3:1	35.463	19.362	13.325	10.53
1:1:1:1:3	28.568	20.55	19.288	14.212

Table 1 – Bandwidth (multiple of STS-1) of failed connections due to glare.

STS-1:STS-3:STS-12:STS-24:STS-48	Best-fit	BANB	Hi-Lo	Interleave
1:1:1:1:1	0	0.912	1.296	0.804
3:1:1:1:1	0	1.824	3.96	3.54
1:3:1:1:1	0	1.68	2.82	2.784
1:1:3:1:1	0	0.624	0.6	0.579
1:1:1:3:1	0	0.384	0.336	0.192
1:1:1:1:3	0	1.32	2.352	1.344

Table 2 – Bandwidth (multiple of STS-1s) of failed connections due to fragmentation.

STS-1:STS-3:STS-12:STS-24:STS-48	Best-fit	BANB	Hi-Lo	Interleave
1:1:1:1:1	31.264	22.063	17.902	14.926
3:1:1:1:1	30.137	22.16	17.187	14.962
1:3:1:1:1	31.03	22.057	16.857	14.349
1:1:3:1:1	35.966	18.904	12.836	11.938
1:1:1:3:1	35.463	19.746	13.661	10.722
1:1:1:1:3	28.568	21.87	21.64	15.556

Table 3 – Total bandwidth (multiple of STS-1s) of failed connections.

B. Bandwidth of Failure Due to Fragmentation

During the simulation, we generate just enough requests to fill the capacity of all links. So, if there is no fragmentation, all the connection should be able to find a link which has enough remaining bandwidth for it. Table 2 records the total bandwidth of failed connection due to fragmentation. In all our simulations, Best-fit does not reject any connections because of fragmentation. BANB will cause a bit more fragmentation because it does not necessarily select the best link. Interleave scheme and Hi-lo scheme will cause more fragmentations. When there are higher portion of large requests (last two lines), Interleave performs much better than Hi-Lo.

C. Total Bandwidth of All Failures

To combine the above two results, Table 3 presents total bandwidth failure either because of glare or fragmentation. Since glare is the dominant source of failure, Table 3 is similar to Table 1. It shows that Best-fit performs the worst at all request distributions and Interleave performs the best.

5.2.2 Different size of aggregated link

In this simulation, we vary the number of OC-48 links in an aggregated link from 1 to 50. Similar to previous simulation, we set the communication delay to 3 *ms*, average arrival interval to 0 *ms*, and processing time to 1 *ms*. The distribution of requests in all categories is 1:1:1:1:1. When there is only 1 OC-48 link, all the schemes performs the same because it is the only choice. When there are two links, BANB, Hi-Lo and Interleave perform similar because all three schemes will let one side choose one link and the other side choose another link.

Figure 6 shows the total bandwidth of failed connections. Interleave and Hi-Lo scheme greatly outperform the widely used Best-fit algorithm. In all the cases, interleave behaves no worse than Hi-Lo. When link is of medium size (8-15), a typical size in today's inter-city transport networks, the Interleave scheme reduces an average of 14% of failed bandwidth compared to Hi-lo scheme.

5.2.3 Different arrival interval

In all previous simulations, we use an average arrival rate of the restoration request as 0. In this section, we study the performance of different scheme when the average request arrival interval varies from 0 to 5 *ms*. Although 5 *ms* seems to be a very small time interval, note that there are hundreds of connections being rerouted. A 5 *ms* average arrival interval will cause the first connection to arrive hundreds of milliseconds or even seconds earlier than last connection.

Similar to previous sections, we set communication delay to 3 *ms* and processing time to 1 *ms* and then study the performance on an aggregated link with 10 OC48s with distribution of requests in all categories as 1:1:1:1:1. As shown in Figure 7, Best-fit performs poorly when the average arrival interval is small. This is because it will cause a lot of glare. However, when the average arrival intervals increase, Best-fit out-performs interleave because in this case, the chances of two connections coming to different end of the link simultaneously becomes much lower. So, the chance of glare is dramatically decreased. This case is similar to the connection provisioning where Best-fit is the best choice. Overall, the performance of Interleave algorithm is stable under all arrival intervals.

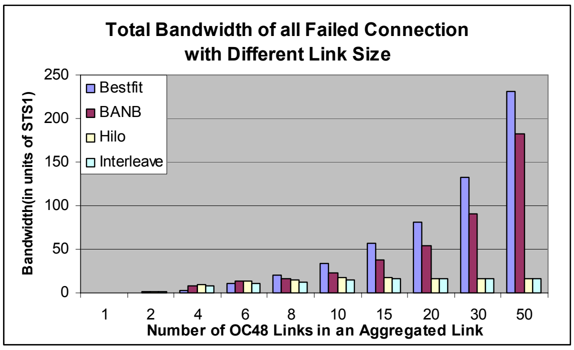

Figure 6 — Total bandwidth of all failed connections on aggregated links with different sizes.

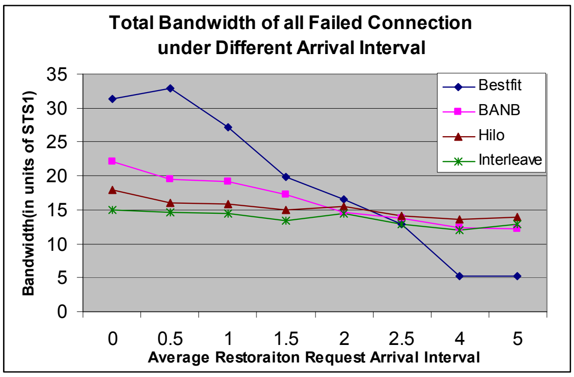

Figure 7 — Total bandwidth (multiple of STS-1) of all failed connections at different arrival intervals.

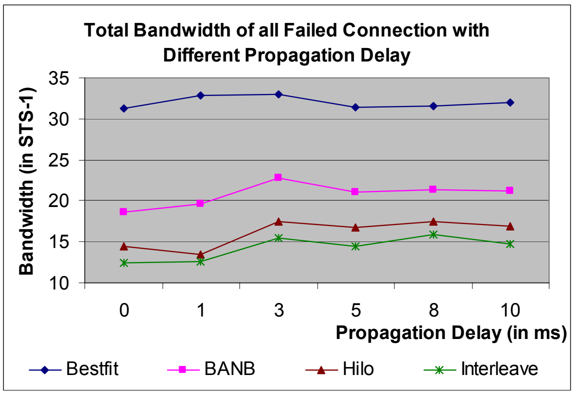

Figure 8 — Total bandwidth (multiple of STS-1) of all failed connections with different communication delay.

5.2.4 Different communication delay

In this simulation, we change the communication delay of the link. We set average arrival interval to 0 *ms*, processing time to 1 *ms*, aggregated link size = 10, and STS-1:STS-3:STS-12:STS-24:STS-48 =1:1:1:1:1. As shown in the Figure 8, communication delay doesn't affect the performance a lot. Interleave schemes stays the best schemes for all values of communication delay.

5.3 Performance comparison in a large intercity network

In this simulation, we use a 95-XC, 164-logical-link network, representative of an intercity backbone network. The same network and demand forecast were used in the studies in references [4]. We implemented different port selection algorithms in an MPLS-like protocol simulator. A logical link may consist of multiple OC-48 links. We assume requests for bandwidth occur in units of STS-1, STS-3, STS-12, or STS-48. In addition, we generate 1000 connections based on the private line distribution for a large intercity backbone to evaluate the performance of four different link selection algorithms. In addition, we consider the Shared Risk Groups (SRG [10]) (fiber spans) and identify 160 cases of possible SRG failure. In the simulation, we cycle through all the failure cases and analyze the restoration process under each failure. All the simulation results reported in this section are the average of all SRG failure cases.

As shown in Table 4, the Interleave link selection algorithm can greatly reduce glare compared to Best-fit. It does create more fragmentation, but it can still reduce 50% overall bandwidth of failed connection compared to the widely adopted Best-fit algorithm.

In this simulation, Interleave algorithm performs better than Hi-lo, but the improvement is not dramatic. This is because in the current network, small connections such as STS-1 and STS-3 dominate. Thus, fragmentation will not be a serious problem. Moreover, the average number of links in an aggregate logical link is small, which does not allow us to differentiate between different link selection algorithms. The advantage of Interleave grows with larger networks and greater proportion of high rate connections (as shown in the previous Table 2).

	Fragmentation	Glare	Total
Best-fit	7408	31632	39040
BANB	12558	21638	34196
Hi-Lo	17820	2170	19990
Interleave	17534	2244	19778

Table 4 – Bandwidth (multiple of STS-1) of all failed connections

V. CONLUSION

In this paper, we studied contention during distributed set-up of high-speed connections in a telecommunication network of XCs. In particular, we proposed an Interleave link selection algorithm to reduce glare as well as fragmentation during restoration path setup process. This scheme has the same implementation complexity as the currently known Best-fit and Hi-Lo schemes but can result in significant advantages.

ACKNOWLEDGEMENTS

We would like to thank Professor Randy Katz for his enormous support. We would also like to thank Dongmei Wang, Guangzhi Li, Charles Kalmanek, and anonymous referees for their insightful comments.

REFERENCES

[1] R. D. Doverspike, G. Sahin, J. L. Strand, and R. W. Tkach, "Fast Restoration in a Mesh Network of Optical Cross-connects", *OFC-99*, Feb 1999.

[2] B. Cortez, R.D. Doverspike, R. Sinha, F. Yu, "Novel Link Selection Schemes in XCs", *provisional patent*, Dec 2002.

[3] M. Kodialam and T. V. Lakshman, "Dynamic Routing of Locally Restorable Bandwidth Guaranteed Tunnels using Aggregated Link Usage Information*", in the proceedings of IEEE INFOCOM*, Apr. 2001.

[4] F. Yu, D. Wang, R. K. Sinha, G. Li, R.D. Doverspike and C. Kalmanek, "Hybrid Centralized/Distributed Approach to Optical Network Restoration," *Optical Fiber Communication. Conference*, March 2003.

[5] G. Li, J. Yates, R. D. Doverspike, and D. Wang, "Experiments in Fast Restoration using GMPLS in Optical / Electronic Mesh Networks" *Post-deadline Papers Digest, OFC-2001,* March 2001.

[6] J. Csirik1 and D. S. Johnson, "Bounded Space On-Line Bin Packing: Best is Better than First", *ALGORITHMICA* 31:2, pp 115-138, 2001.

[7] E.G. Coffman, Jr., M.R. Garey, and D.S. Johnson, "Approximation Algorithms for Bin-Packing -- An Updated Survey", *In Algorithm Design for Computer System Design*, 1984.

[8] D. S. Johnson, "Fast Algorithms for Bin Packing", *Journal of Computer and System Sciences* 8, pp 272-314, 1974.

[9] C. Kenyon, "Best-fit bin-packing with random order", *In the proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, 1996.

[10] Robert Doverspike and Jennifer Yates, "Challenges For MPLS in Optical Network Restoration," *IEEE Communications Magazine*, February 2001.