Improving Dependability of Real-Time Communication with Preplanned Backup Routes and Spare Resource Pool*

Songkuk Kim and Kang G. Shin

University of Michigan
Ann Arbor, MI 48109-2122
{songkuk,kgshin}@eecs.umich.edu

Abstract. Timely recovery from network component failures is essential to the applications that require guaranteed-performance communication services. To achieve dependable communication, there have been several proposals that can be classified into two categories: reactive and proactive.
The reactive approach tries to reroute traffic upon detection of a network link/node failure. This approach may suffer from contention and resource shortage when multiple connections need to be rerouted at the same time. The proactive approach, on the other hand, prepares a backup channel in advance that will be activated upon failure of the corresponding primary channel due to a broken link or node. The proactive approach, although it offers higher dependability, incurs higher routing overhead than the reactive approach.
We propose a hybrid approach that reduces signaling overhead by decoupling backup routing from resource provisioning. We also propose an efficient backup routing algorithm for the hybrid approach. Our in-depth simulation results show that the proposed approach can achieve the ability of failure recovery comparable to the proactive scheme without the need for broadcasting the routing information.

1 Introduction

Though the Internet has infiltrated into our everyday lives over the last couple of decades, its use has been limited to non-critical applications. Mission/safety-critical applications, which need real-time communication service, cannot use the Internet as their communication media. One chief reason for this is that the Internet cannot preserve real-time performance in case of link/router failures.

Paxson [1] measured the stability of routes between selected nodes, and reported that up to 2.2% of probes experienced connection outages. Labovitz *et al.* [2] showed that about 50% of routes were available for less than 99.9% of the time. This kind of route failures either fail the real-time applications or significantly degrade the performance of real-time communication.

* The work reported in this paper was supported in part by the Office of Naval Research under Grant No. N00014-99-1-0465.

To achieve guaranteed performance, real-time communication schemes rely on off-line resource reservation and runtime traffic scheduling. Resources are reserved along a specific path before data packets are sent. If a failure occurs on the path, a new route should be found and resources need to be reserved along the new path. So, the instability of routes will disable the real-time communication service for a long time. According to Labovitz *et al.* [2], it may take tens of minutes for the intermediate routers to converge on a new topology. Furthermore, one link failure may disable many real-time connections, and their end systems will try to set up new channels at the same time. Because one real-time connection chooses a route without knowing others' choices, this re-establishment of real-time channels may experience severe contention. Many re-establishment attempts may result in signaling failures even when the network has sufficient resources.

Banerjea [3, 4] explored use of the reactive scheme to provide dependable communication service. The reactive scheme deals with failures only after their occurrence. To restore a real-time connection affected by a failure, the connection's source node selects a detour path around the faulty component. The merit of the reactive scheme is that it requires no extra work in the absence of failures. However, the reactive scheme cannot guarantee success of recovery because it does not reserve resources *a priori* for backup channels. When the network load is high, most resources are consumed by primary channels, and it is difficult to find a path that has bandwidth available for an additional real-time channel.

Another drawback of the reactive scheme is that it may suffer from contention when the source node tries to set up a detour channel around the faulty link/node. Because each source node selects a detour path based on its *local* information that does not reflect other nodes' detour path choices, the decision may result in conflicts over a link that does not have enough available bandwidth. Banerjea showed that sequential signaling is effective to alleviate the contention. Sequential signaling means that connection setup requests are served one by one. However, it is practically impossible to coordinate the order of setup requests in a distributed manner, and the sequential signaling takes a long time to reroute the broken real-time channels.

To provide fast and guaranteed recovery, proactive schemes have been proposed [5–8]. In Han and Shin's scheme, a dependable (D-) connection consists of one primary channel and one or more backup channels. When a primary channel is established, a backup channel(s) is also set up with (spare) resource reservation. To reduce the amount of spare resources, they developed a backup multiplexing scheme. If primary channels do not share any network component, their backups can be multiplexed (and hence overbooked) over the same resources. Though the proactive scheme can guarantee dependability, it comes with high resource overhead. To establish a backup channel, the signaling procedure should be performed and the intermediate routers on the backup path should keep the information about backup channels for their multiplexing. Establishing a new backup channel can change the amount of available bandwidth for primaries because backups and primaries compete for same resources. Thus, backup signaling

hastens the broadcasting of link status. Furthermore, for backup multiplexing, the information on the amount of spare resource needs to be conveyed to other nodes, which incurs additional routing overhead.

To provide dependable communication service with low overhead, we propose a hybrid scheme in which a D-connection is composed of a *primary* channel and a preplanned *backup path*. To establish a detour channel immediately without contention upon failure of a link, we pre-select a backup route when we establish a primary channel. However, we do not perform any signaling along the backup route. When a failure occurs to a link, the disabled real-time connections try to reserve resources and set up backup channels along the pre-selected backup routes. To prevent bandwidth shortage, we set aside a certain amount of bandwidth on each link as spare resources. The spare resources can be used to deliver best-effort traffic until backup channels are activated. Because the amount of spare resources is fixed, we need not broadcast the information of spare resources.

2 Pool of Spare Resources

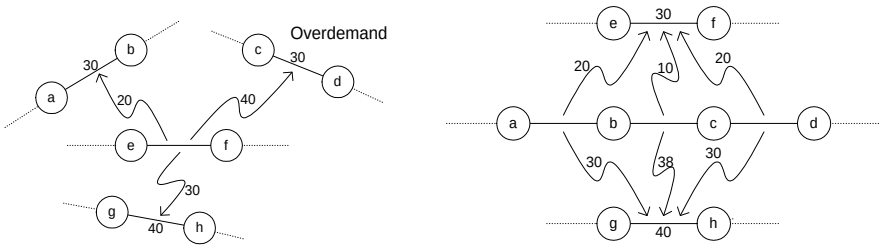
When almost all of the network resources are occupied by real-time channels, the network is said to be *saturated*. If a link is broken when the network is saturated, it will be impossible to find a detour path. Thus, dependability decreases dramatically as the network load increases. To cope with this problem, we need to set aside some resources. For the proactive approach, the backup channel establishment makes resource reservation, so it can prevent any drastic decrease of dependability. Because our proposed scheme does not reserve bandwidth for each connection, we need a separate mechanism to set aside some resources.

In general, the bandwidth of a link is divided into two parts when the network deploys a real-time service based on resource reservation. One part of bandwidth serves real-time traffic, and a proper amount of bandwidth needs to be reserved before the real-time data is delivered. The other part of bandwidth must be left unreserved to prevent best-effort traffic from starvation.

In our scheme, the bandwidth is divided into three parts. An additional, third part is reserved for backup channel traffic. In other words, we set aside a certain amount of bandwidth for an aggregate of backups. The spare resources of a link are used for best-effort traffic in the absence of failure.

This pre-reservation has several important differences from the per-channel-based resource reservation of the proactive scheme. Our proposed scheme does not require any complex signaling procedure for backups. Thus, our approach does not involve the intermediate routers on a backup path until the backup is activated, whereas the proactive approach demands the intermediate routers to maintain information about backup channels and to change the amount of spare resources as backup channels are added or released.

Another advantage of the hybrid scheme is that the backup preparation does not affect primary routing. As described in the previous section, the backup channel establishment under the proactive approach affects the primary channels



(a) (b)

Fig. 1. The number next to each link is the amount of spare bandwidth and the number next to each arrow is the amount of backup bandwidth required to activate backups. The unit is Mbps.

that will be established later. The backup resource reservation of the proactive approach requires other nodes to update the link states, because it changes the amount of available link resources for real-time channels. This results in frequent exchanges of routing messages, which are usually expensive. Our proposed hybrid scheme does not incur these additional routing messages because it does not change the amount of spare resources.

The hybrid scheme affects the best-effort traffic only when backups are activated, whereas the proactive scheme changes the total available bandwidth for best-effort traffic whenever the spare resource changes. In fact, we can think of the backup channels as borrowing some bandwidth temporarily from the best-effort portion of bandwidth.

3 Selection of Backup Routes

The dependability of a D-connection is determined by its pre-selected backup route. When a link failure disables multiple primary channels, their backups that traverse a link without enough spare bandwidth will fail to be activated. To make the backup activation more likely to succeed, the backup routes should be chosen very carefully.

A link failure triggers activation of multiple backups that run through different links. If the demand of backup bandwidth on each of these links is smaller than the amount of spare bandwidth, all the activation will succeed. Let b_ℓ^k denote the bandwidth demand on link ℓ when link k fails. In other words, $b_\ell^k = \sum_{d \in \mathcal{D}_\ell^k} r_d$, where $\mathcal{D}_\ell^k$ is the set of D-connections whose primaries traverse link k with backups running through ℓ and r_d is the bandwidth requirement of D-connection d.

Figure 1 (a) shows an example of spare and backup bandwidth. The number next to a link is the amount of spare bandwidth and the number next to each

arrow denotes the amount of backup bandwidth required to activate backups. When link (e,f) breaks down, backups will be activated along pre-selected routes. Because links (a,b) and (g,h) have enough spare bandwidths, the backups routed over them will be activated successfully. In link (c,d), the backups require more bandwidth than its spare bandwidth. So, some of the backups on the link may not be activated. Because the backups can use the primary portion of link bandwidth, if available, the overdemand of backup bandwidth does not always cause activation failures.

The goal of backup routing is to choose a route such that the new backup does not overdemand spare resources along its route. Because a D-connection prepares an end-to-end backup, we attempt to construct a backup route with links that does not overdemand resources from any link of the corresponding primary route. Figure 1 (b) shows an example of choosing the link of a backup route. We try to find a backup route for the primary channel that traverses links (a,b), (b,c), and (c,d). The new backup requires 5 Mbps. The backups, already routed, require bandwidth on each link as shown in the figure. Links (e,f) and (g,h) do not overdemand from links (a,b), (b,c), and (c,d).

If we select either link (e,f) or (g,h) to construct a backup route, each link of the primary route will demand more backup bandwidth on the selected link. If link (g,h) is selected, the backup bandwidth demand on link (b,c) will increase to 43 Mbps because the new backup requires 5 Mbps. This exceeds the spare bandwidth of 40 Mbps. If we choose link (e,f), however, we can avoid the overdemand because the spare bandwidth of link (e,f) is still larger than the increased demand of backup bandwidth of each link on the primary route. Thus, link (e,f) is a better choice than link (g,h) for the new backup.

As shown in the above example, when we choose a backup route for a D-connection d, link ℓ is an appropriate choice of the backup route if $s_\ell \geq b_\ell^k + r_d, \forall k \in PR_d$, where s_ℓ is the spare bandwidth of link ℓ and PR_d denotes the set of links of d's primary route. To maximize the probability of successful backup activation, a backup route should minimize the number of links that overdemand from its corresponding primary route. Also, a backup should be as disjoint as possible from its primary route. To find the shortest route among those that satisfy the requirements, we use Dijkstra's algorithm after assigning a cost, C_ℓ, to link ℓ:

$$C_\ell = \begin{cases} M & \text{if } \ell \in PR_d, \\ m & \text{if } \exists k, \text{ such that } k \in PR_d \text{ and } s_\ell < b_\ell^k + r_d, \\ \varepsilon & \text{otherwise} \end{cases} \tag{1}$$

where $M \gg m \gg \varepsilon$. Because the backup route is the least desirable when it overlaps with the primary, the largest cost is given to the link which belongs to the primary route. ε is given to a link to choose the shortest route when the other conditions are the same.

When a source node establishes a D-connection, the node selects a backup route after establishing a primary channel. So, the source node can easily check the first condition of the above equation. However, it is not trivial for the node

to check the second condition because the node must know b_ℓ^k's. One possible way is that every node maintains all the b_ℓ^k's for every pair of links. However, this approach is not practical because it involves a huge amount of information exchange between nodes. To cope with this problem, we devised a new protocol that examines the second condition in a distributed manner.

4 The Protocol for a Hybrid Scheme

4.1 Notation and Data Structures

In a network $G(\mathcal{N}, \mathcal{L})$ with $|\mathcal{N}|$ nodes and $|\mathcal{L}|$ links, each link has a unique id between 1 and $|\mathcal{L}|$. Let l_i be the link whose id is i.

- SV (spare bandwidth vector): $|\mathcal{L}|$-dimensional integer vector whose i^{th} element denotes s_i, the amount of spare bandwidth of l_i. Every node keeps SV.
- BV_i (backup bandwidth vector): $|\mathcal{L}|$-dimensional vector whose j^{th} element is b_j^i. BV_i is maintained by node n if l_i is adjacent to n.
- APV (accumulated properness vector): $|\mathcal{L}|$-dimensional one-bit vector. This vector is calculated before selecting a backup route to check which link is appropriate to compose a backup route. If the i^{th} bit is set to 1, l_i is suitable for the backup route.

4.2 Establishing a Primary Channel and Selecting a Backup Route

When a source node sets up a D-connection, the node establishes the primary path first. It can use any routing method for primary channels, because our backup routing is orthogonal to the primary routing. After selecting the primary path, the source node sends a QUERY message along the primary channel route.

When an intermediate node receives the QUERY message, it relays the message to the next node. When the QUERY message arrives at the destination node, the node prepares a RESULT message. The RESULT message has a field that contains APV. At the beginning, every bit of APV is set to 1. The destination node sends the confirmation message to the source node along the primary channel path in the reverse direction.

When an intermediate router receives the RESULT message through link l_i, it computes the links that overdemand resources on l_i and sets the corresponding bits of em APV to 0. In other words, if $b_j^i + r_d > s_j$, the j^{th} element of APV is set to 0. Because this intermediate node maintains SV and BV_i, the node can easily update APV. After updating APV, the node relays the RESULT message to the next node.

When the RESULT message arrives at the source node, each bit of the APV in the RESULT message is as follows:

$$APV_i = \begin{cases} 0, & \text{if } \exists k, \text{ such that } k \text{ belongs to the primary links and } s_i < b_i^k + r_d, \\ 1, & \text{otherwise} \end{cases}$$

where APV^i denotes the i^{th} element of APV. The above equation is equivalent to the second condition of Eq. (1). APV^i is 1 if and only if l_i has enough bandwidth to accommodate backups when any link on the corresponding primary route fails. The source node selects a backup route using Dijkstra's algorithm after assigning link cost C_l:

$$C_\ell = \begin{cases} M & \text{if } \ell \in PR_d, \\ m & \text{else if } APV^l = 0 \\ \varepsilon & \text{otherwise} \end{cases}$$

4.3 Maintaining BV

As described above, every router should maintain BV for each link attached to it to choose links suitable for a backup route and exchange BV during the signaling of the primary channel setup. BV_i represents the resource demand on each link to activate backups when link l_i is broken. More precisely, the j^{th} element of BV_i is the sum of bandwidths required by all backups on l_j. If ℓ is link $(v1, v2)$, both $v1$ and $v2$ maintain BV_ℓ.

To keep BVs up-to-date, each node should know backup routes of D-connections whose primaries go through its links. After a source node chooses a backup route for a given primary channel, the source node informs nodes on the primary route of its decision. Also, when the backup route is no longer needed, the source node notifies the intermediate nodes to decrease BV. An intermediate node updates BV based on the data sent by the source node.

Because we cannot assume that the source node tears down the real-time connection gracefully, we take a soft state approach to maintaining BV. In the case of RSVP, the connection information is invalidated if it is not refreshed before a timer expires. However, maintaining BV as a soft state is more complex because BV is in summation form of information about backups, whereas RSVP keeps the information on a per-connection basis.

The basic idea is that a source node periodically sends Backup Bandwidth Demand (BBD) message carrying link *id*s of a backup route and its bandwidth requirement to intermediate nodes on the corresponding primary route. Intermediate nodes add the backup bandwidth requirements into a temporary BV when it receives BBD. An intermediate node updates BVs periodically by replacing them with temporary BVs in which the node has accumulated bandwidth demands of backups. To make this operation idempotent, we incorporate a version number into BV and BBD messages. The detailed procedure is given below.

1. A source node sends a BBD message along with the primary path. The BBD message constitutes the resource requirement of a backup route, the list of nodes that the message will visit, a version number for each router, and a backup route. All the version numbers are set to 0 when the end node sends the BBD message for a backup route for the first time.
2. When an intermediate router receives the BBD message, it compares the corresponding version number in the message with that of its BV. If the

version of the message is less than that of BV, the bandwidth requirement is added into both BV and temporary BV. If the two version numbers are equal, the resource requirement is added only into temporary BV. If the message has a higher version than BV, the bandwidth requirement is not added into either BV or temporary BV. The intermediate node relays this BBD message to the next node after increasing the corresponding version number in the message by one.

3. When the destination node receives the BBD message, it generates a BBD-ACK message that has updated version numbers and sends BBD-ACK to the source node along the primary route in reverse direction.
4. The intermediate nodes relay the BBD-ACK message until the message reaches the source node.
5. The end node records the version number in the real-time connection table and sets up a timer with interval T. When the timer expires the node resends the BBD message.
6. All the nodes have a timer with interval T. When the timer expires, the node replaces BVs with temporary BVs, resets temporary BV to 0, and increases the version number of BV by 1.

The above algorithm ensures two important features: (1) BV and temporary BV are not increased more than once during one period for the same real-time connection, even when the end node sends the BBD message more than once before the node's timer expires; (2) BV will be properly decreased within two periods even when the source node does not tear down the connection gracefully.

5 Scalability and Deployment

Every node maintains one BV per each link attached to the node and one SV. Because the vectors have as many elements as the number of links in the network, the space complexity is $O(|\mathcal{L}| \times (nd+1))$ where nd is the node degree. To set up a new D-connection, $O(|\mathcal{L}| \times h)$ messages need to be exchanged, where h is the hop count of the primary route. Because our scheme does not keep per-connection information in intermediate nodes and does not broadcast routing messages, it is highly scalable when the network has a limited number of links. Thus, overlay networks and VPNs are good candidates to deploy our scheme.

To deploy our scheme in the Internet, which has millions of links, we propose a hierarchical approach. The Internet is composed of Autonomous Systems (AS). Instead of an end-to-end backup, we can prepare an ingress-to-egress backup within an AS. ASs too are organized hierarchically. An AS is composed of POPs and a backbone. Recently, Spring *et al.* [9] measured ISP topologies. According to their results, an ISP has hundreds of backbone links, and each POP has less than one hundred routers. Thus, each POP can use our scheme to protect D-connections within the POP and an AS sets up ingress-to-egress backups in the backbone, which connects POPs, with a reasonable amount of storage and message overhead.

6 Performance Evaluation

We evaluated the proposed scheme by simulation. We implemented the reactive scheme, the proactive scheme, and the hybrid scheme using the network simulator **ns**. We generated scenario files where D-connection requests are listed. Each D-connection request consists of source node, destination node, bandwidth requirement, start and end times. The scenario files are fed to each simulator. Each simulator attempts to establish a D-connection based on the source node, the destination node, and the bandwidth requirement. If it succeeds, the simulator records the path taken by the primary and backup channels on a trace file, and terminates the D-connection at the specified end time. We analyzed the trace files thus generated, to derive performance metrics.

For simplicity, we assume that each D-connection requires 1 Mbps. The running time of a D-connection was uniformly distributed between 20 and 60 minutes so that the average running time may be 40 minutes. We conducted simulation with three network topologies: an 8×8 mesh, an 8×8 torus, and a random topology.

6.1 Load Index and Performance Metrics

To evaluate the performance of each scheme at various network loads, it is important to choose the load index that represents the load imposed on the network and the performance metrics to compare different schemes. For best-effort traffic, the overall resource usage or the amount of traffic can be a good load index. However, in real-time communication, the resource usage and the amount of traffic are the performance metrics because they can be affected by the routing and scheduling schemes. The path of a real-time channel is not always the shortest. Depending on the routing scheme used, a real-time channel traverses a different path, and the overall bandwidth consumption is different.

The amount of resources consumed by real-time channels can be expressed as:

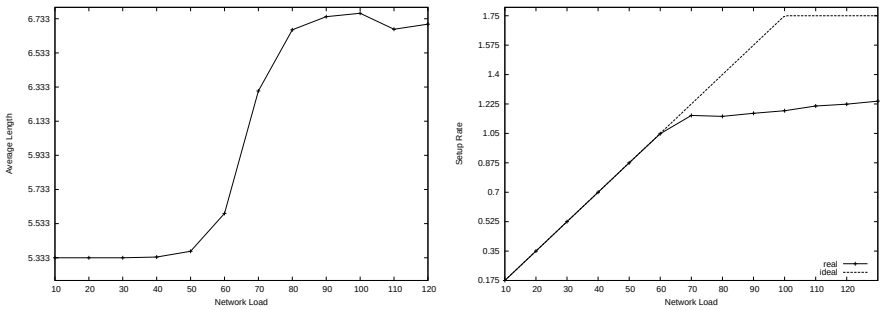
$$\lambda' \times \overline{RT} \times \overline{h'} \times \overline{b'}$$

where λ' is the setup rate of real-time channels, $\overline{RT}$ is the average running time, $\overline{h'}$ is the average hop count of established real-time channels, and $\overline{b'}$ is the bandwidth requirement of established real-time channels. For our simulation, we assume that every real-time connection requires the same amount of bandwidth, b, so $\overline{b'} = b$. Because the network resources are limited, the setup rate has an upper limit:

$$\lambda' \leq \frac{B}{\overline{RT} \times \overline{h'} \times b} \leq \frac{B}{\overline{RT} \times \overline{D} \times b}$$

where B is the total amount of bandwidth of a network and $\overline{D}$ is the average hop count of the shortest paths between all pairs of nodes and $\overline{h'} \geq \overline{D}$. We define the maximum setup rate of real-time connections that the network can accommodate as:

$$\lambda'_{max} = \frac{B}{\overline{RT} \times \overline{D} \times b}.$$



(a) Average Length Connections (b) Acceptance Rate

Fig. 2. Example performance metrics and load index

Because the network cannot accept the setup request at a rate higher than λ'_{max}, we increase the request rate of real-time connections λ up to λ'_{max} to see how each scheme performs. We defined *network load* as the ratio of λ to λ'_{max}:

$$NetworkLoad(\%) = \frac{\lambda}{\lambda'_{max}} \times 100.$$

As the network load increases, the bandwidths of some links are used up by real-time channels. So, real-time channels traverse longer routes than the shortest routes, i.e., $\overline{h'}$ becomes larger than $\overline{D}$. Thus, the network becomes saturated before the network load reaches λ'_{max}.

Figure 2 shows how $\overline{h'}$ and λ change in a sample network. The average distance, $\overline{D}$, is 5.33 and the maximum setup rate, λ'_{max}, is 1.75. We established real-time connections without backup channels. The line *real* represents the simulation results; in Figure (b), the line *ideal* shows the theoretical setup rate where the $\overline{h'}$ remains equal to $\overline{D}$. After the network load reaches 80%, the network is saturated and cannot accept more requests. When this happens, increasing the network load does not make a considerable difference in the network.

$\overline{h'}$ is one of performance metrics. We use *normalized average hop count* $\overline{nh'}$, defined as $\frac{\overline{h'}}{\overline{D}}$, to make the metric independent of the network characteristics. $\overline{nh'}$ of primaries shows the bandwidth each primary channel consumes, on average, in proportion to bandwidth usage of the shortest path. Usually, $\overline{nh'}$ of backups is longer than that of primaries. As $\overline{nh'}$ of backups becomes longer, the backups consume more bandwidth when they are activated, and the end-to-end delay of backups is increased.

The dependability is the probability that a D-connection can continue the service when a link on the primary channel of the D-connection is broken. To measure the dependability, we broke every link alternately and tried to activate the corresponding backups. We calculated the probability of successful backup

activation of each link and computed the average over all links. More precisely, the dependability is defined as:

$$Dependability = \frac{\sum_{\ell \in \mathcal{L}} \frac{|\mathcal{S}_\ell|}{|\mathcal{D}_\ell|}}{|\mathcal{L}|}$$

where $\mathcal{S}_\ell$ is the set of backups successfully activated when link ℓ is broken, and $\mathcal{D}_\ell$ is the set of D-connections whose primaries traverse ℓ.

Another important metric is the *average acceptance rate*, $\alpha = \frac{\lambda'}{\lambda}$. Because the acceptance rate is closely related to bandwidth consumption, α implies a *capacity overhead*.

6.2 Topology Characteristics

As mentioned earlier, we used 8×8 mesh, 8×8 torus, and random network topologies. We generated the random topology using the Waxman 2 model with Georgia Tech Internetwork Topology Models (GT-ITM) [10]. Initially, we generated a network with 100 nodes and pruned nodes that have only one link to make every node have at least two links attached to it.

These three topologies have several different characteristics that affect the performance of fault-management schemes. Table 1 shows the characteristics of three topologies.

Though the torus and the mesh have the same number of nodes and similar numbers of links, the average distance of the torus network is only about $\frac{3}{4}$ of that of the mesh network. By "distance," we mean the length of the shortest path between two nodes. The random network has the shortest average distance although its node degree is the lowest.

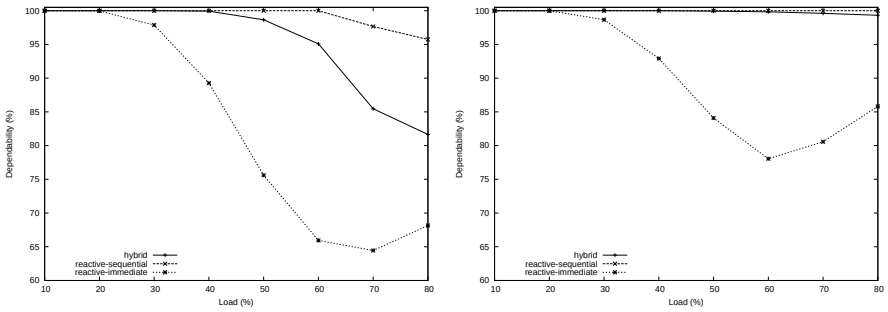
Though the three topologies have a similar average node degree, the distribution of node degrees is different. Table 2 shows the distribution. In the torus, every node has the same node degree of 4. In the random topology, 28 nodes have only two links, so there are a small number of choices for detour routes.

	8×8 mesh	8×8 torus	random
Nodes	64	64	78
Links	112	128	129
Avg. Dist.	$\frac{16}{3}$	$4 \times \frac{64}{63}$	3.706
Node Degree	3.5	4	3.308

Table 1. Characteristics of the topologies used for simulation

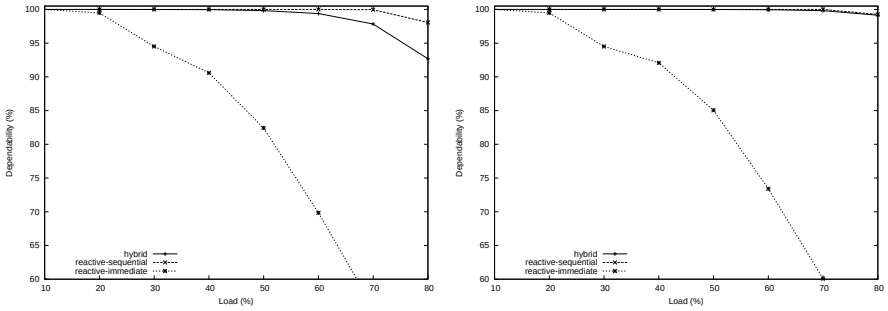
Node Degree	2	3	4	5	6	7
8×8 mesh	4	24	36			
8×8 torus			64			
random topology	28	21	13	12	1	3

Table 2. Distribution of node degrees



(a) Extra 6% Spare Bandwidth (b) Extra 16% Spare Bandwidth

Fig. 3. Comparison with the reactive scheme on an 8× 8 mesh network.



(a) Extra 6% Spare Bandwidth (b) Extra 10% Spare Bandwidth

Fig. 4. Comparison with the reactive scheme on an 8×8 torus network

6.3 Comparison with the Reactive Schemes

Figure 3 shows the dependability of the two reactive schemes and the hybrid scheme. The reactive-immediate scheme tries to reroute all D-connections at the same time. The reactive-sequential scheme reroutes D-connections one by one.

The reactive schemes do not utilize spare resources and suffer from resource shortage when the network load is high. To make a fair comparison and to evaluate the effects of spare resources, we provisioned spare bandwidth for reactive schemes. In this simulation, the bandwidth for primary channels on each link is 100 Mbps, and we provisioned an additional spare bandwidth for backups. We changed the spare bandwidth to see how each scheme performs with various spare bandwidths.

As shown in Figure 3, the reactive-immediate scheme provides considerably low fault-tolerance. This scheme suffers from backup conflicts because each node tries to set up a backup channel independently without considering the backup route selection of other nodes. This is why the reactive-immediate scheme does not perform well with 16% spare bandwidth, with which the hybrid scheme and the sequential scheme provide 99% fault-recovery.

Both the hybrid scheme and the sequential scheme improve the dependability as the spare bandwidth increases. Because both schemes select backup routes avoiding backup conflicts by consideration of backup routes of other D-connections, they can take a full advantage of spare bandwidth.

The reactive-sequential scheme shows slightly better fault-tolerance than the hybrid scheme. The sequential scheme utilizes all the network resources excluding the broken link, whereas the hybrid scheme cannot use the entire primary path.

However, the sequential rerouting is practically impossible to implement and incurs a very long recovery delay. To reroute the D-connections one by one in a distributed manner, it is necessary to decide the order in which each node recovers. Moreover, each node, which wants to establish a backup channel, waits for the new link status that reflects the recently-established backups. Because fast recovery is one of the most important requirements for real-time communication, the sequential scheme is not applicable to real-time communication.

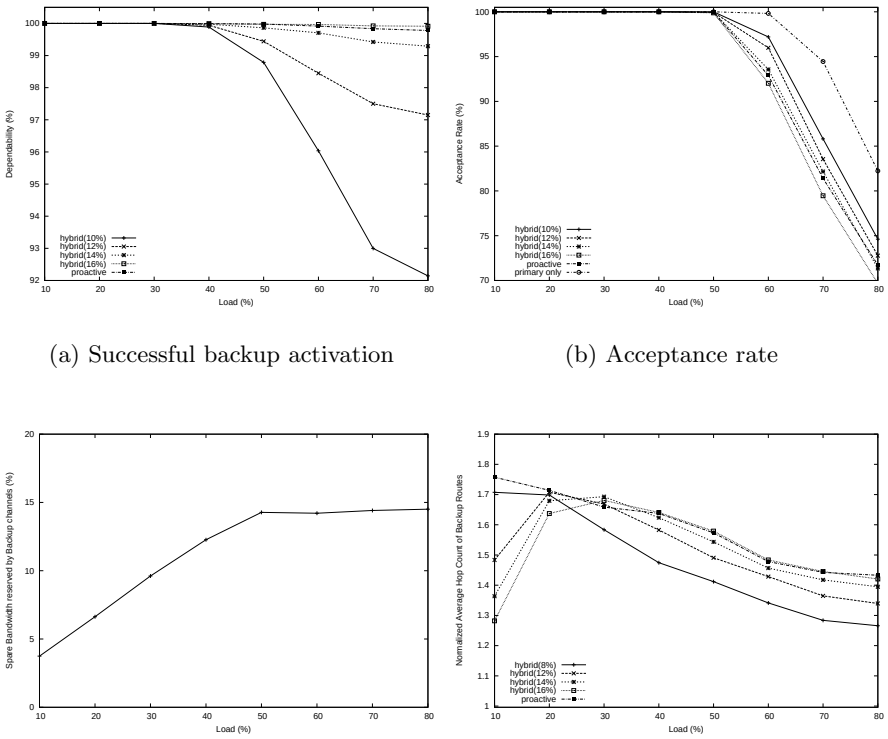
Figure 4 shows the performance of each scheme in the 8×8 torus network. Both the hybrid and the sequential schemes perform much better in this network compared to the 8×8 mesh network. As stated in the previous section, the torus has a shorter average distance between two nodes. The average length of backups is shorter in the torus than in the mesh and backups use less bandwidth. Thus, with less spare bandwidth the torus can accommodate more backups. To provide 99% dependability, the hybrid scheme needs only 10% extra bandwidth.

However, the reactive-immediate scheme performs worse in the torus than in the mesh topology. A shorter average distance means that the starting end nodes of the D-connections start are closer to the broken link. So, the end nodes that reroute D-connections are more closely located to each other in the torus and the end nodes are more likely to choose the same links for backup routes. The reactive-immediate scheme suffers from more contention in the torus.

As shown in this comparison with the reactive schemes, careful selection of backup routes improves dependability dramatically. To avoid contention, backups need to be distributed over a large area. With knowledge about backups of other D-connections, the hybrid scheme distributes backups and utilizes the spare bandwidth efficiently.

6.4 Comparison with the Proactive Scheme

We compared the hybrid scheme with the proactive scheme. Because the proactive scheme reserves spare bandwidth dynamically, it does not need separate spare bandwidths. To compare the two schemes under the same condition, we provided each link with 100 Mbps bandwidth for D-connections. Because the hybrid scheme requires separate spare bandwidth, we reserved a certain amount of



(a) Successful backup activation

(b) Acceptance rate

(c) Spare bandwidth in proactive scheme

(d) Average hop count of backups

Fig. 5. Comparison with the proactive scheme on an 8×8 mesh network

bandwidth for backups out of the 100 Mbps bandwidth. To see the performance of the hybrid scheme with various amounts of spare bandwidth, we changed the spare bandwidth from 4 to 20% of the total bandwidth.

Figure 5 shows the performance of the proactive and hybrid schemes. The hybrid scheme improves dependability as the spare bandwidth increases. When 16% of the total bandwidth is provisioned as spare bandwidth, the hybrid scheme shows dependability compatible to the proactive scheme.

However, as more bandwidth is reserved for backups, less bandwidth is available for primaries. Figure 5 (b) shows the acceptance rate of requests for D-connections. In the figure, the 'primary only' represents the acceptance rate when we establish real-time channels without backups. The difference between the primary only and each scheme is the *capacity overhead*.

The proactive scheme incurs a capacity overhead similar to that of the hybrid scheme with 14% spare bandwidth. We can find the reason from Figure 5 (c). The figure shows the amount of bandwidth reserved by backup channels in

the proactive scheme. Because the proactive scheme reserves spare bandwidth according to the network load, more bandwidth is reserved for backups as the network load increases. After the network load reaches 50%, the spare bandwidth does not increase, because the primaries occupy the remaining bandwidth.

The proactive scheme reserves a maximum of about 14% bandwidth for backups. This is the reason why the proactive scheme shows an acceptance rate similar to the hybrid scheme with 14% spare bandwidth. However, the proactive scheme provides higher dependability than the hybrid scheme with 14% spare bandwidth. To match the dependability of the proactive scheme, the hybrid scheme requires a little more bandwidth. This is because the proactive scheme utilizes much more information when it selects backup routes.

Figure 5 (d) shows the average hop count of backups. When load is low, the backups of the proactive scheme is longer than those of the hybrid scheme with a similar amount of spare bandwidth. Because the hybrid scheme sets aside a certain amount of bandwidth in advance irrespective of the network load, the hybrid scheme has more room for backups and can find backup routes within a near area. The proactive scheme starts without any spare bandwidth and increases it. As the backup bandwidth increases, it can find a shorter route for a given D-connection using spare bandwidth reserved for other backups.

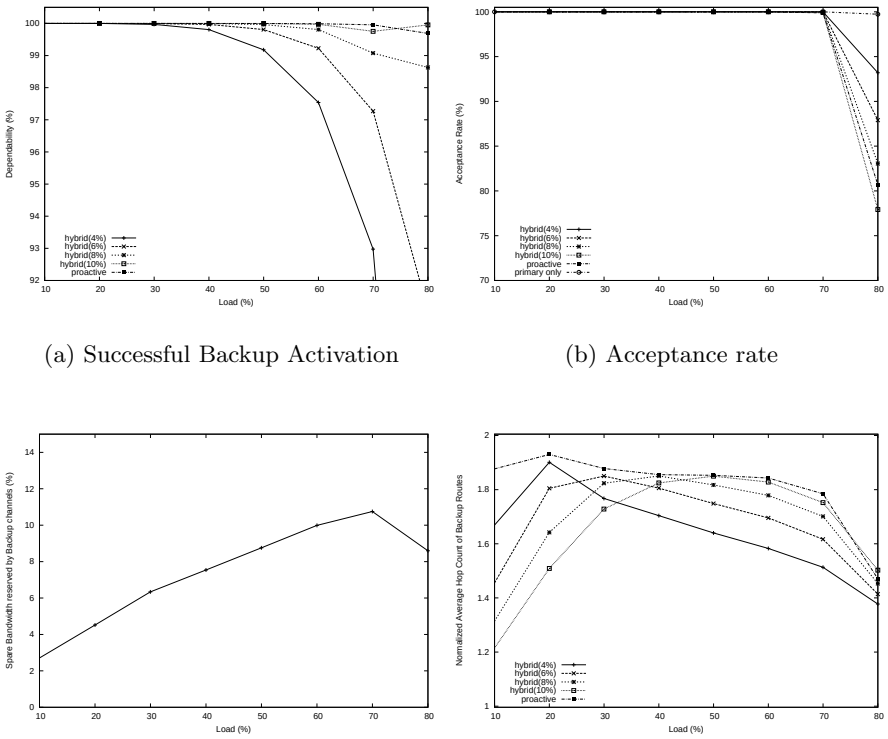
To see how the topology affects the performance of each scheme, we conducted simulation with the torus and the random topology. Figure 6 shows the performance in the 8×8 torus topology.

As described in Section 6.2, the torus has a shorter average distance between two nodes and a higher node degree. A shorter average distance means that each backup requires less backup bandwidth. As shown in Figure 6 (a), with 10% spare bandwidth, the hybrid scheme provides more than 99% dependability.

Figure 6 (c) shows the spare bandwidth reserved for backups in the proactive scheme. The maximum spare bandwidth is about 10%. In the torus, both the proactive and hybrid schemes need less spare bandwidth than in the mesh network. This is because the torus has more node degree in addition to a shorter average distance. A higher node degree means that there are more disjoint routes within a certain boundary between two nodes. So, backups are distributed over more routes without conflicts and less spare bandwidth accommodates more backups.

When the network load is 80%, the spare bandwidth of the proactive scheme decreases. Because the primary channel reserves the bandwidth before the backup channel, whenever the bandwidth is available, the free bandwidth is allocated to a primary channel and the corresponding backup channel squeezes into the spare bandwidth shared by other backups. This results in less allocation of bandwidth to backup channels and more backup conflicts. The contraction of the backup bandwidth accompanies the degradation of dependability. As the spare bandwidth of the proactive scheme decreases below 10%, the dependability of the proactive scheme also decreases.

Because less bandwidth is reserved for backups in the torus, more bandwidth is available for primaries. Figure 6 (b) shows the acceptance rate. Until the net-



(a) Successful Backup Activation

(b) Acceptance rate

(c) Spare bandwidth in proactive scheme

(d) Average hop count of backups

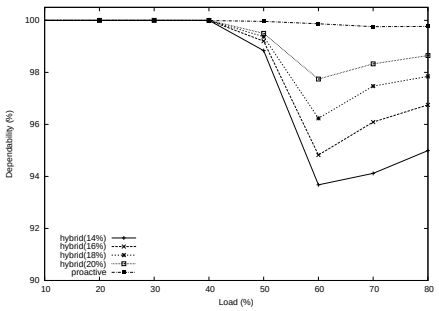
Fig. 6. Comparison with the proactive scheme on an 8×8 torus network

work load reaches 70%, the acceptance rate is 100%. Compared to the acceptance rate in the mesh topology, the torus improves the acceptance rate considerably.

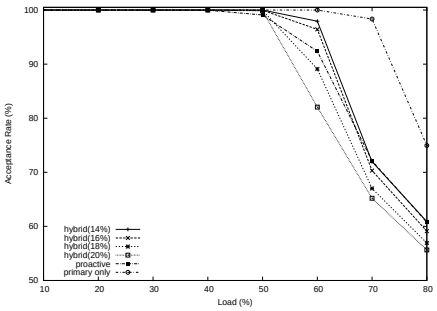
The average hop count of backups shows a similar pattern. As the network load increases, the hybrid scheme selects longer backup routes to avoid overdemands. After the network load reaches a certain point, it is impossible to build a route without links that do not overdemand. Then, the length of backup routes decreases because a shorter path incurs less conflicts and the routing algorithm selects the shortest path when several paths have the same number of conflicts.

The 8×8 torus has 16 more links than the 8×8 mesh. The additional 16 links improve the performance of D-connections dramatically. The torus uses 30% less spare bandwidth, provides higher dependability, and shows lower capacity overhead.

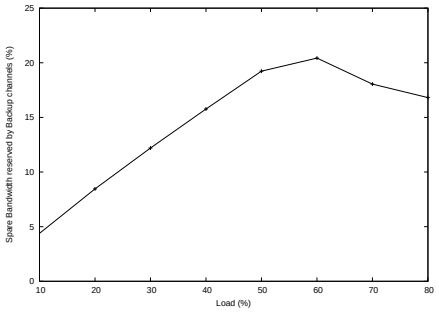
The random topology is similar to the torus in the average distance. However, in the node degree, the random topology is considerably different. On average, the node degree of the random topology is just a little smaller than the mesh.



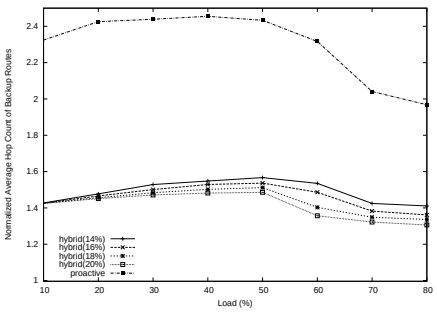
(a) Successful Backup Activation



(b) Acceptance rate



(c) Spare bandwidth in proactive scheme



(d) Average hop count of backups

Fig. 7. Comparison with the proactive scheme on a random network

However, there is a large deviation in the node degree of the random topology. More than a half of the nodes have 2 or 3 links, whereas 4 nodes have 6 or 7 links.

The diverse node degrees introduce some disadvantages to the D-connections. Because many nodes have a small number of links, there are fewer detour routes and more backups conflicts occur. Though the average node degree is smaller, the average distance of random topology is shorter than the mesh and the torus. This is because a few nodes with many links act as crossroads. This results in nonuniform traffic flows in the network.

Figure 7 (a) shows the dependability in the random topology. The hybrid scheme provides a little lower dependability even with 20% spare bandwidth. As shown in Figure 7 (c), the proactive scheme reserves a maximum of 20% spare bandwidth. With a similar amount of spare bandwidth, the proactive scheme provides higher dependability because the proactive scheme reserves different

amounts of spare bandwidth on each link according to network traffic, whereas the hybrid scheme uses the same amount of spare bandwidth on each link.

The proactive scheme shows less capacity overhead in Figure 7 (b). When traffic is concentrated in some links, the proactive scheme reserves spare bandwidth in other links. So, the congested links have more bandwidth for primaries, and backups run through other less congested links. This improves dependability and decreases the capacity overhead.

Figure 7 (d) shows the average hop count of backup routes. The proactive scheme selects significantly longer routes than the hybrid scheme. The poor connectivity of the random topology affects more the backup routing of the proactive scheme because the proactive scheme uses less spare bandwidth.

7 Related Work

Since Han and Shin [6, 7] proposed the backup multiplexing scheme, many proactive schemes have employed this idea. In the backup multiplexing scheme, the same spare resources can be shared by multiple backups, if the corresponding primaries do not traverse through same links. This approach needs to broadcast the information about spare bandwidth on each link to select backup routes. Also, it broadcasts the information about available bandwidth to select primary routes frequently, because changing the amount of spare bandwidth affects the amount of the available bandwidth. Our hybrid scheme is the first approach that does not broadcast the information about the shared spare bandwidth.

Though Han and Shin explored several routing heuristics for backup channels, they did not propose a distributed algorithm for backup routing. Kodialam *et al.* [11] developed a routing algorithm that selects a backup path based on the amount of the aggregate bandwidth used on each link by primary channels, the aggregate bandwidth used on each link by backup channels, and the link residual free bandwidth. The algorithm tries to minimize the amount of backup bandwidth increased by a new backup when selecting a route for a new backup. However, because the algorithm does not have any information about the paths of other backups, it overestimates the spare bandwidth by assuming that every disrupted D-connection has conflicts on the same link.

In [12], three routing schemes are proposed and evaluated. The algorithm with the best performance chooses a backup path based on *backup conflicts* in addition to the amount of free bandwidth. Backup paths are said to have conflicts if they traverse the same link and their corresponding primaries share one or more links. Although this approach utilizes more information, backup paths are still selected without precise information.

Li *et al.* [13] recently proposed a distributed backup route selection algorithm for the proactive scheme. They use full information about spare resources as we do. Though their algorithm is similar to ours, there are several differences. First, their algorithm involves signaling for backup channels. Moreover, though it is not clearly stated, their scheme needs to broadcast information about the amount of spare resources, whereas our hybrid scheme does not need broadcasting.

Li *et al.* use the increment of spare resources as their metric to choose a backup route. In other words, they try to minimize the amount of spare resources. Whereas, we select a backup route to minimize the number of links that do not have enough spare resources. Because two algorithms use different metrics for path selection, the information exchanged for backup selection is also different. In our algorithm, routers exchange a bit-vector that represents a list of links that are suitable for a backup route, while in Li's algorithm routers exchange an integer-vector representing the amount of spare resources that will be needed when a link fails. Because the size of the vector is the same as the number of links in the network, the integer-vector consumes more bandwidth than the bit-vector and may not be delivered in a single packet.

8 Conclusion

In this paper, we presented a hybrid scheme that pre-selects backup routes without reserving bandwidth for each backup channel. Because a certain amount of spare bandwidth is set aside *a priori* for backups, the hybrid scheme does not require global routing messages for spare bandwidth, thus eliminating one of the main drawbacks of the proactive scheme. Also, we devised a novel distributed routing algorithm that does not require nodes to keep information for each D-connection.

We evaluated and compared the effectiveness of the hybrid scheme by simulation. We compared the hybrid scheme with the proactive and reactive schemes for various network topologies. The hybrid scheme offers as high dependability as the proactive scheme without the need for broadcasting the information about spare bandwidth. When the network is homogeneous, the hybrid scheme is more effective. Using the hybrid scheme, we were able to reduce the overhead of the proactive scheme without degrading its performance.

References

1. Paxson, V.: End-to-end routing behavior in the internet. IEEE/ACM Transaction on Networking **5** (1997) 601–615
2. Labovitz, C., Ahuja, A., Jahanian, F.: Experimental study of internet stability and backbone failures. In: Proceedings of IEEE FTCS'99. (1999) 278–285
3. Banerjea, A., Parris, C., Ferrari, D.: Recovering guaranteed performance service connections from single and multiple faults. In: Proceedings of IEEE GLOBECOM'94, San Francisco, CA (1994) 162–168
4. Banerjea, A.: Fault recovery for guaranteed performance communications connections. IEEE Transactions on Computer Systems **7** (1999) 653–668
5. Dovrolis, C., Ramanathan, P.: Resource aggregation for fault tolerance in integrated services networks. Computer Communication Review **28** (1998) 39–53
6. Han, S., Shin, K.G.: Efficient spare resource allocation for fast restoration of real-time channels from network component failures. In: Proceedings of IEEE RTSS'97. (1997) 99–108

7. Han, S., Shin, K.G.: Fast restoration of real-time communication service from component failures in multihop networks. In: Proceedings of ACM SIGCOMM'97. (1997) 77–88
8. Han, S., Shin, K.G.: A primary-backup channel approach to dependable real-time communication in multi-hop networks. IEEE Transactions on Computers **47** (1998)
9. Spring, N., Mahajan, R., Wetherall, D.: Measuring isp topologies with rocketfuel. In: Proceedings of ACM SIGCOMM 2002. (2002) 133–146
10. Calvert, K., Zegura, E.: Gt-itm: Georgia tech internetwork topology models. http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/gt-itm.tar.gz. (1996)
11. Kodialam, M., Lakshman, T.V.: Dynamic routing of bandwidth guaranteed tunnels with restoration. In: Proceedings of INFOCOM 2000. (2000) 902–911
12. Kim, S., Qiao, D., Kodase, S., Shin, K.G.: Design and evaluation of routing schemes for dependable real-time connections. In: Proceedings of DSN 2001. (2001) 285–294
13. Li, G., Wang, D., Kalmanek, C., Doverspike, R.: Efficient distributed path selection for shared restoration connections. In: Proceedings of INFOCOM 2002. (2002) 140–149