I. Introduction

In 1962, an MIT professor named J.C.R Licklider published a paper entitled “On-Line Man Computer Communication” [10]. The ideas he expressed in this paper were a vision of something very similar to today’s internet. Shortly after publishing this paper, Licklider became the director of ARPA’s program to develop a network that would ensure communication between various government locations in case of an emergency [10]. In 1972, ARPA’s computer network was demonstrated at the International Conference on Computers and Communications as ARPAnet, showing the world the potential computer networking had [11].

In 1969, when ARPAnet was first used, there were 4 hosts. As the Internet grew, the number of hosts grew also. As of August 2002, there were an estimated 162,128,493 hosts on the Internet [12]. This huge increase in the number of hosts was necessarily accompanied by an increase in infrastructure to handle the increased traffic. Combined with the increasing number of people who are connected via broadband service, the increase in the number of hosts demands that the Internet’s backbone (and eventually subnets) be as fast and wide as possible.

A fast, wide network path such as those that are utilized presently in fiber optic lines provides much increased bandwidth and data rate, thus improving the amount of data that can be transmitted between hosts. In a single optical fiber, data rates of 320 Gbps have been demonstrated [2]. There is, however, a downside to the great speeds that new technology allows. If a fiber carrying data at a rate of 320 Gbps failed, the amount of connections lost would be tremendous. A network that addresses and attempts to fix this problem of potential connectivity loss due to failure is called a survivable network.

II. The Problem of Survivability

To understand why survivability of networks has become an issue only in the past couple of years, it helps to look at an example of a high speed line compared to a low speed line. Suppose there was a line that sustained a data rate of 1Mbps. If this line failed for a minute, 7.5 Megabytes (about the size of 5 floppy disks) of data would be blocked from transmitting on this line. Now, suppose there was a line that sustained a data rate of 1 Gbps. If this line failed for only a minute, there would be 7.5 Gigabytes (about the size of a typical hard drive 5 years ago) of data that were blocked from transmitting on it. If the network the line was in had no survivability measures in place, many connections would be lost and much data would have to be retransmitted.
SONET and other protocols utilizing optical networks possess the ability to provide link restoration capabilities, which at first seems to make any lower level survivability a moot point [5]. The reason why survivability should be added to the optical layer rather than the network layer is that the higher layers take time “on the order of seconds,” with the exception of SONET, compared to time “on the order of milliseconds,” which is the expected delay to restore links on the optical layer [5]. While switching all networks over to SONET may seem like an alternative to developing survivability on the optical layer, doing so would be very expensive and arduous. It is desirable and necessary (as with almost all network improvements) that survivability be available to most protocols in use today, especially IP. For this reason, survivable networks must be implemented in the lower layers.

II.B. Importance of Disjoint paths

Comprehensive network survivability is dependent on the existence of disjoint paths between pairs of nodes in a network. There are two different types of disjoint paths: edge-disjoint paths and node-disjoint paths. Edge-disjoint paths are paths that share no common arc. Node-disjoint paths are paths that share no common node. All node-disjoint paths are also edge-disjoint paths. Disjoint paths are necessary for survivable networks because they are guaranteed to recover from single failures, no matter what link or node fails (assuming the end nodes do not fail).

Node-disjoint paths are more desirable than edge-disjoint paths, because no nodes (pieces of hardware like routers in the case of networks) are shared except the end nodes. This means that a router can fail on one path and the other path will still be connected. For example, Figure 1 below shows two graphs. Figure 1(a) shows a graph in which there are two edge-disjoint paths between nodes A and C. The graph in Figure 1(b) has two node-disjoint paths between nodes A and C. If the node B fails in Figure 1(a), there will be no path between nodes A and C. If either node B1 or B2 fails in Figure 1(b), there will still be a path between nodes A and C, assuming no other failures. It should be noted that both graphs depicted in Figure 1 will still have a path between nodes A and C should any one of the edges fail.
The edge-disjoint paths and node-disjoint paths problems are both NP-hard problems [14]. This means that they are “decision problems that are intrinsically harder than those that can be solved by a nondeterministic Turing machine in polynomial time” [3]. In other terms, they are hard to solve because there is no way to be certain that a proposed solution is the best possible one.

Solving NP problems is one of the most daunting tasks set to scientists. The Clay Mathematics Institute is offering $1 million dollars to anyone who can solve the problem “P = NP?” [7]. In simpler terms, the problem is: can a NP-complete problem be solved in polynomial time? NP-complete problems can be difficult enough to warrant a $1 M to understand them better, and NP-hard problems are even more difficult, so approximating a solution to an NP-hard problem is the most anyone can do when approaching the disjoint-paths problems.

II.C. Approximating Shortest Disjoint Paths

Survivable networks not only require disjoint paths, but the also should try to utilize the shortest disjoint-paths possible. This allows the least amount of degradation in network performance. Several parties have approached the problem of finding the shortest disjoint-paths possible in a graph (representing a network). As noted before, finding a solution for optimal disjoint paths is an NP-hard problem, so it is a very difficult undertaking.

In a paper [14] on disjoint path algorithms by Stavros G. Kolliopoulos and Clifford Stein of Dartmouth College, two previous approaches to the disjoint path problem are reviewed and two new approaches based on the previous approaches are proposed.
The two previous approaches discussed by Kolliopoulos and Stein are the rounding approach and the routing approach. The rounding approach “consists of solving a fractional relaxation and then use [sic] rounding techniques to obtain an integral solution” [14]. They note that this approach has yielded the best results so far and has even produced “the first approximation algorithm for uniform unsplittable flow” [14]. The routing approach is one in which “a commodity is never split, i.e. routed fractionally along more than one path during the course of the algorithm” [14]. Kalliopoulos and Stein note that the routing approach has not been show to be very applicable, except in the case of the “on-line Bounded Greedy Algorithm … whose approximation guarantee depends also on the diameter of the graph” [14].

The new approaches presented by Kolliopoulos and Stein are “a simple deterministic greedy algorithm” for the routing approach and a packing integer program for the rounding approach [14]. The greedy algorithm tries to “keep routing commodities along sufficiently short paths” so that the lowerbound of the number of commodities is close to the optimum [14]. One problem with the greedy algorithm as they propose it is that it is only applicable to edge-disjoint or vertex-disjoint paths in which the edge-disjoint paths are unweighted. They do not discount weight altogether though. They propose using a packing integer program to approximate optimum disjoint-paths for both weighted and unweighted paths. Packing integer programs are programs that are able to model several NP-complete problems by utilizing “randomized rounding techniques” [14]. With the disjoint paths problem being NP-complete and having a related NP-hard optimization problem, they applied techniques gleaned from packing integer programs to model the disjoint paths problem.

Another pair of researchers, Petr Kolman and Christian Scheideler, favors the bounded greedy algorithm over other algorithms to approximate an optimized solution to the disjoint-paths problem [8]. They contest that most of the “randomized algorithms suffer from the drawback that only the expected competitive ratio is good” [8]. To help demonstrate their findings, Kolman and Scheideler present a parameter termed the “$D$-bounded routing number $R$” [8]. Basically, for a graph with a routing number $R$ there exists “a path collection for any permutation routing problem” for which the maximum congestion (“the maximum number of paths that share an edge”) is $R$ and the maximum length of any path is at most $D$ [8].

The bounded greedy algorithm rejects a request if no path with a length shorter than or equal to a chosen parameter $L$ is free. If a path with a length $L$ or less is free, then the request is granted [8]. A simple version of the bounded greedy algorithm lets $L=2*R$, but applying the concept of the $D$-bounded routing number lets $L=2*D$ [8]. This decreases the upper bound of the maximum path length, which provides a more optimized approximation.
This algorithm is refined even further with the “Shrewd algorithm,” which imposes a maximum limit on the heaviness of edges (usually a sign of a bottlenecks in the network) used in a path [8].

No matter how disjoint-paths are found in a network, the existence of those paths is required for a comprehensive survivable networks. The optimized disjoint-paths problem is a very difficult problem in its own right, but it composes just a part of the problem of survivable networks. Indeed, one could say that it did not matter if the shortest disjoint paths were used in terms of connectivity loss, but it is imperative that short paths are chosen to provide the fastest connection possible to the end nodes of a network.

II.D. Absence of Disjoint-paths

Despite the desirability of disjoint paths between two communicating nodes, there may not always be disjoint paths. This does not mean that no survivability is possible in the network. Survivability may also be provided in some sense by reducing the scope of the disjoint-paths problem to the intermediate nodes that compose the path from one end node to the other end node. That is, if disjoint-paths between the end nodes do not exist, disjoint paths between intermediate nodes may still exist. This provides something called “link-protection” [5].

III. Solutions for Survivability

As the need for survivable networks increases, the number of people working on the problem of survivability increases. There are a few popular algorithms for survivable networks, namely: 1 + 1 dedicated protection, shared-path protection, 1 + N protection, path restoration, and shared protection using rings. There are also a few papers on new methods that can provide survivability such as distributed partial information management, wavelength division multiplexed (WDM) self-healing rings, and book-ahead guaranteed services.

III.A. 1 + 1 Dedicated-Path Protection

In the 1 + 1 dedicated protection scheme, whenever two nodes are connected by any number of wavelengths on an active path, an identical number of wavelengths are reserved on a disjoint path [13]. These reserved wavelengths cannot be used by any other nodes, although the paths that they are on may be used on different wavelengths by other nodes [13]. R. Drew Davis, et al. propose a method for choosing which paths to use as active and protection paths based on demand and the costs of everything from the ports on the nodes to the amplifiers needed to propagate the light through the fibers [13].

A drawback to the dedicated protection scheme is that every active path must have a backup path reserved that is necessarily inactive. Theoretically, this
would cause a loss of half the possible capacity of all paths used in the network. Using this scheme would be a huge blow to performance, and if the mean time to failure of a link was long enough, it might be better to have no protection at all.

### III.B. Shared-path Protection

Expanding on the idea of 1 + 1 dedicated protection, shared-path protection lets multiple active paths share a single protection link and wavelength [5]. The chances that multiple active paths between different nodes fail at the same time is unlikely, so by letting multiple paths share a certain wavelength on a certain link, other wavelengths can be used for other connections. This scheme uses the capacity of the network links more efficiently than the 1 + 1 dedicated protection scheme. A drawback to this shared protection scheme is that in the unlikely event that two active paths sharing the same protection path do fail, one of the connections will be severed. In a dedicated protection scheme, all active paths could fail simultaneously and they would each be guaranteed service through the respective protection paths.

### III.C. Dedicated-link Protection

One issue with path protection is that when a link or node breaks on the primary path, the connection is not restored on the protection path until the information that a failure has occurred propagates back to the originating end node [6]. Link protection is a scheme that can recover from failures without the propagation delay involved in path protection [5].

Dedicated-link protection is much like dedicated-path protection. When a connection is demanded, dedicated paths are reserved for each link in the path [5]. When a link in the path is broken, the traffic is simply routed around the broken link on the dedicated path that was reserved for the broken link. It would seem that this scheme would have a lot more overhead work involved since it essentially poses the shortest disjoint-path problem with regard to each link in the path rather than the whole path. This scheme does protect against multiple link failure in a single path as well as providing survivability to connections that do not have disjoint-paths between end nodes.

### III.D. Shared-link Protection

Shared-link Protection is analogous to shared-path protection in the same way dedicated-link protection is analogous to dedicated-path protection. The wavelengths of the protection paths of the links may be shared with other protection paths [5]. This is more efficient than using dedicated link-protection while still providing protection in the case of multiple link failures in a single path.
III.E. Shared Protection Using Rings

A scheme that builds on the concept of shared-path protection is shared protection using rings. In the graph in Figure 2 there exists a ring i.e., a path that can start and end at the same node without passing through any link or node more than once. For any demand that is made for a connection between two nodes on the ring, an active path can be established along the shortest (least cost) path within the ring while a backup path is established along the longest (greatest cost) path on the ring [13]. For example, assume there is demand for a connection between nodes A and B in the graph in Figure 2. Assuming that all edges have the same capacity and are currently unused, an active path will be established between nodes A and B on the path A-B and a backup path will be established on path A-E-D-C-B. If the link between A and B fails, the backup path will recover the connection.

![Figure 2: Graph with a ring](image)

Since this ring protection scheme uses the principles of shared protection paths, the backup paths in the ring can share wavelengths. Dedicated protection would be impossible in the ring scheme because once a failure occurs in a link in the ring, all protection paths in the ring are necessarily severed.

III.F. 1 + N Protection

1 + N Protection builds on the concept of 1 + 1 dedicated protection. 1 + N protection involves establishing active connections on all but the longest disjoint-paths between a pair of nodes (for a single demand) and reserving the longest disjoint-path as the protection path for that demanded node pair [13]. Utilizing multiple active paths means that the demand can be divided among them [13]. When a single path fails, the protection link will become an active link and take over for the part of the demand that was served by the failed path. If more than a single path fails, the demand can be re-divided among the paths that remain. The more disjoint-paths that exist between a node pair, the more failures this scheme can recover from. A downside to this scheme is that there
is a good deal of overhead involved in finding and establishing multiple paths for a single demand.

III.G. Path Restoration

Another approach to survivability that removes the overhead of involved in setting up backup paths in advance is path-restoration. The path restoration scheme does not allocate any backup paths at the time of the connection request. Instead, a backup path is found dynamically by end nodes participating in a distributed algorithm, after a path fails [5]. If no backup path exists for any connection using the failed path, then the connection is completely blocked [5].

There is also a link restoration scheme in which the end nodes using a path with a failure look for a route around the failed link [5]. This scheme seems unnecessary because the largest advantage of treating failure on the scope of a link rather than the scope of the path is that the end nodes do not have to be involved. Using the end nodes to solve the problem of link restoration requires the propagation delay of communication between the node after which failure occurred and the end node.

III.H. Distributed Partial Information Management

Distributed partial information management (DPIM) is a “novel framework” proposed to address:

“several major challenges in achieving efficient shared path protection under distributed control with only partial information, including (1) how much partial information about existing active and backup paths (or APs and BPs respectively) is maintained and exchanged; (2) how to obtain a good estimate of the bandwidth needed by a candidate BP, called BBW, and subsequently select a pair of AP and BP for a connection establishment request so as to minimize total bandwidth consumption and/or maximize revenues; (3) how to distributively allocate minimal BBW (and deallocate maximal BBW) via distributed signaling; and (4) how to update and subsequently exchange the partial information” [1].

It has been shown that integer linear program can be used to determine the best active path and backup path given access to complete information about existing active paths and backup paths, but it is not practical to provide every node complete information [1]. There has also been a scheme that uses partial information at the cost of a decrease in the optimization of paths and a scheme that uses an “active path first heuristic” with “complete but aggregated information,” which has the same scaling problem that complete information with integer linear programming has [1]. The proposed distributed partial information management framework uses the advantage that complete
information offers by distributing the information across all of the nodes in the network, thus easing the scalability problem that occurs when complete information is used [1].

III.I. Book-Ahead Guaranteed Services

In an essay by S.R. Thirumalasetty and D. Medhi the concept of book-ahead guaranteed (BAG) services (that are handled along side best-effort services) is examined in the context of Multi-protocol Label Switching (MPLS), which should have the capability for traffic engineering that current standards do not provide [4]. Their envisioned system would allow users on a network utilizing MPLS to buy guaranteed bandwidth to another node over the network for a certain period of time. They suggest that three levels of BAG services be made available in terms of survivability: zero-survivable, fractional-survivable, and fully-survivable [4]. A zero survivable service would guarantee bandwidth for a service only when the network was operating normally. The fractional-survivable level would guarantee service under normal conditions and reduced bandwidth service when there is a major link failure, while the full-survivable level would guarantee bandwidth for the service even in the event of a major link failure [4].

Since BAG services must be requested ahead of time (Medhi and Thirumalasetty suggest more than one hour ahead of time), the algorithm to provide survivability does not have to be on-line or real-time [4]. They employ a linear programming relaxation in their pursuit of a solution to solve the traffic engineering problem (making sure BAG services are indeed guaranteed while not starving best-effort services) that providing BAG services presents [4].

The idea of BAG services is a novel one, but it is somewhat of a capitalist idea also. Users should not have to pay a premium to have network connections utilizing survivability schemes. Medhi and Thirumalasetty suggest that BAG services can be used for users who “want to request access … to supercomputer sites to conduct high-bandwidth intensive applications with bandwidth and/or survivability guarantee,” but they do not consider that the possibility of providing survivability for every connection [4]. It would be daunting to provide everyone with survivable networks, but given the data rates that are going to be possible in the future, there may be enough network overhead to handle it.

IV. Comparison of Survivability Schemes
There are many schemes for survivability, all of which have advantages and disadvantages to go along with them. The issue of which scheme is the best must be determined on a case by case basis because it depends on network topology, the physical medium of the network (e.g., optical or not), the amount of traffic on the network, the consistency of data rates of the links in the network, the mean time to failure for different network components, and numerous other factors.

If components never failed, there would be no need for survivability in networks. Unfortunately, components do fail, but if mean time to failure is high enough, failures of links and nodes will seldom occur. If it is a given that failures will seldom occur, the best schemes for survivability would seem to be those which only provide protection for one failure on a demanded connection, such as 1 + 1 dedicated protection, shared-path protection, path restoration, and shared protection using rings. 1 + 1 dedicated protection uses a lot of link capacity to apply to a problem that seldom happens. Shared-path protection and shared protection using rings are very similar, with the main difference being that the ring is defined so that that a node does not have to find the best backup path to another node on the same ring. Both of these schemes provide protection against one failure and use significantly less link capacity than the dedicated protection. One draw-back to shared schemes is that they are more complex to implement, but the increased complexity is worth the link utilization improvement. The path restoration is the most complex of the schemes that provide protection against only one fault per connection. Also, path restoration requires the least link capacity at the cost of having the highest reconnection time. The path restoration scheme has a definite advantage over path protection because it takes into account the network topology following a failure.

If components failed often, as they might in a large network, it is important to protect against more than one failure for a single demanded connection. The link-protection, link-restoration, and 1 + N protection schemes all provide protection for more than one link failure, given a proper network topology. The link-protection and link-restoration can recover from the failure of any link or node that has a route around it. Link-protection uses even more link capacity than path-protection, as each link is individually protected. Link-restoration may not be as good as path restoration, because the path that replaces the link may not be as optimal as the possibly disjoint path that path restoration can find. The 1+N protection scheme can protect against N failures while still providing service, but there is a good amount of work to be done in connecting all of the paths it finds useful.

The DPIM and BAG services schemes are more complicated than the rest of the scheme presented here. The DPIM provides the interesting concept of storing complete path information across the network in different nodes, but it would require all nodes on the network to use DPIM. The scheme presented to deal with BAG services seems to be conjecture more than anything. For this scheme to properly function MPLS would have to be used across the network and the whole network would have to support the BAG scheme. BAG service is designed as a way to make money by providing guaranteed bandwidth to those people who can afford it. While
it is true that this may just be the definition of a business (taking money in exchange for a good or service), providing survivability to a few for a cost is not as desirable or simple as providing survivable networks to all users.

V. Conclusion

Ensuring a network’s survivability is both a difficult and necessary task given the high speed networks that are rapidly replacing older technology. There have been many people that have risen to the task of formulating solutions to the problem of survivability (and the accompanying problem of optimal disjoint-paths). There have been both simple and complicated schemes proposed for implementation. One of the big problems some of the schemes such as DPIM and BAG services have is that they will not work properly if the whole network does not use them. In a scheme like dedicated path-protection, the calling node finds and sets up the paths for the connection, eliminating the need for a network overhaul. The largest problem facing the implementation of survivable networks may be standardization. If networks use different schemes it may be problematic to provide a survivable connection between users on different networks and it will be difficult to convince all networks to switch over to one scheme without a standard. That being said, SONET is an ANSI standard that supports network survivability using rings and has been utilized by large telecommunications companies such as Sprint [9]. SONET operates on the physical layer and provides reconnection of broken paths in the range of milliseconds. It also provides a better view of the future of survivability than convoluted schemes like DPIM, so Occam’s Razor is shown to have merit.
REFERENCES


