CHAPTER 3

Failure Impacts, Survivability Principles, and Measures of Survivability

In this chapter we will look at causes of fiber cable failures, identify the impacts of outage, and relate these to the goals for restoration speed. We then provide an overview of the different basic principles and techniques for network survivability. This provides a first overview appreciation of the basic approaches of span, path and *p*-cycle based survivability which we treat in depth in later chapters. The survey of basic mesh-oriented schemes in this chapter also lets the reader see these schemes in contrast to ring-based schemes that are 100% or more redundant, and which we do not consider further in the book. The chapter concludes with a look at the quantitative measures of network survivability, and the relationships between availability, reliability and survivability.

3.1 Transport Network Failures and Their Impacts

3.1.1 Causes of Failure

It is reasonable to ask why fiber optic cables get cut at all, given the widespread appreciation of how important it is to physically protect such cables. Isn't it enough to just bury the cables suitably deep or put them in conduits and stress that everyone should be careful when digging? In practice what seems so simple is actually not. Despite best-efforts at physical protection, it seems to be one of those large-scale statistical certainties that a fairly high rate of cable cuts is inevitable. This is not unique to our industry. Philosophically, the problem of fiber cable cuts is similar to other problems of operating many large-scale systems. To a lay person it may seem baffling when planes crash, or nuclear reactors fail, or water sources are contaminated, and so on, while experts in the respective technical communities are sometimes amazed it doesn't happen more often! The insider knows of so many things that *can* go wrong [Vau96].

Indeed some have gone as far as to say that the most fundamental engineering activity is *the study of why things fail* [Ada91] [Petr85].

And so it is with today's widespread fiber networks: it doesn't matter how advanced the optical technology is, it is in a cable. When you deploy 100,000 miles of any kind of cable, even with the best physical protection measures, it *will* be damaged. And with surprising frequency. One estimate is that any given mile of cable will operate about 228 years before it is damaged (4.39 cuts/year/1000 sheath-miles) [ToNe94]. At first that sounds reassuring, but on 100,000 installed route miles it implies more than one cut *per day* on average. To the extent that construction activities correlate with the working week, such failures may also tend to cluster, producing some single days over the course of a year in which perhaps two or three cuts occur. In 2002 the FCC also published findings that metro networks annually experience 13 cuts for every 1000 miles of fiber, and long haul networks experience 3 cuts for 1000 miles of fiber [VePo02]. Even the lower rate for long haul implies a cable cut every four days on average in a not atypical network with 30,000 route-miles of fiber. These frequencies of cable cut events are hundreds to thousands of times higher than corresponding reports of transport layer node failures, which helps explain why network survivability design is primarily focused on recovery from span or link failures arising from cable cuts.

3.1.2 Crawford's Study

After several serious cable-related network outages in the 1990s, a comprehensive survey on the frequency and causes of fiber optic cable failures was commissioned by regulatory bodies in the United States [Craw93]. Figure 3-1 presents data from that report on the causes of fiber failure. As the euphemism of a "backhoe fade" suggests, almost 60% of all cuts were caused by cable dig-ups. Two-thirds of those occurred even though the contractor had notified the facility owner before digging. Vehicle damage was most often suffered by aerial cables from collision with poles, but also from tall vehicles snagging the cables directly or colliding with highway overpasses where cable ducts are present. Human error is typified by a craftsperson cutting the wrong cables during maintenance or during copper cable salvage activities ("copper mining") in a manhole. Power line damage refers to metallic contact of the strain-bearing "messenger cable" in aerial installations with power lines. The resulting i^2R (heat dissipation) melts the fiber cable. Rodents (mice, rats, gophers, beavers) seem to be fond of the taste and texture of the cable jackets and gnaw on them in both aerial and underground installations. The resulting cable failures are usually partial (not all fibers are severed). It seems reasonable that by partial gnawing at cable sheaths, rodents must also compromise a number of cables which then ultimately fail at a later time. Sabotage failures were typically the result of deliberate actions by disgruntled employees, or vandalism when facility huts or enclosures are broken into. Today, terrorist attacks on fiber optic cables must also be considered.

Floods caused failures by taking out bridge crossings or by water permeation of cables resulting in optical loss increases in the fiber from hydrogen infiltration. Excavation damage reports are distinct from dig-ups in that these were cases of failure due to rockfalls and heavy

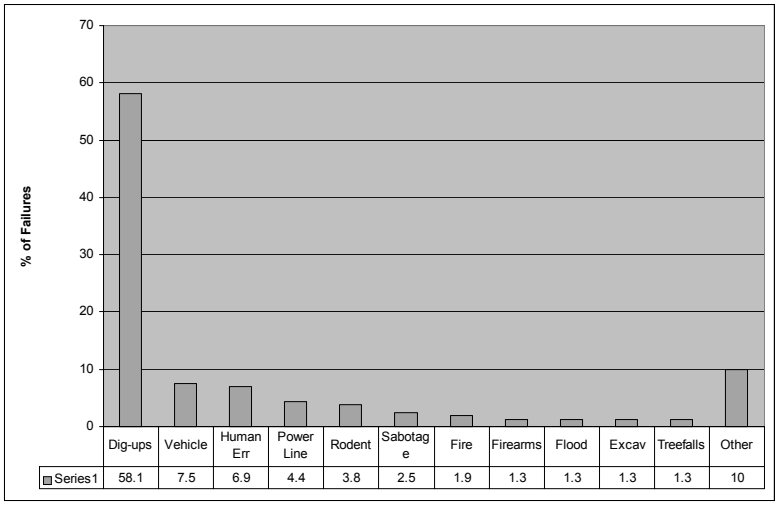


Figure 3-1 Immediate cause breakdown for 160 fiber optic cable cuts ([Craw93]).

vehicle bearing loads associated with excavation activities. Treefalls were not a large contributor in this U.S. survey but in some areas where ice storms are more seasonal, tree falls and ice loads can be a major hazard to aerial cables. Conduits are expensive to install, and in much of the country cable burial is also a major capital expense. In parts of Canada (notably the Canadian shield), trenching can be almost infeasible as bedrock lies right at the surface. Consequently, much fiber cable mileage remains on aerial pole-lines and is subject to weather-related hazards such as ice, tree falls, and lightning strikes.

Figure 3-2 shows the statistics of the related service outage and physical cable repair times. Physical *repair* took a mean time of 14 hours but had a high variance, with some individual repair times reaching to 100 hours. The average *service* outage time over the 160 reported cable cuts was 5.2 hours. As far as can be determined from the report, all 160 of the cable failures reported were single-failure events. This is quite relevant to the applicability and economic feasibility of later methods in the book for optimal spare capacity design.

In 1997 another interesting report came out on the causes of failure in the overall public switched network (PSTN) [Kuhn97]. Its data on cable-related outages due to component flaws, acts of nature, cable cutting, cable maintenance errors and power supply failures affecting transmission again add up to form the single largest source of outages. Interestingly Kuhn concludes that human intervention and automatic rerouting in the call-handling switches were the key factors in the systems's overall reliability. This is quite relevant as we aim in this book to reduce the dependence on human intervention wherever possible in real-time and effectively to achieve the adaptive routing benefits of the PSTN down in the transport layer itself. Also of interest to readers is [Zorp89] which includes details of the famous Hinsdale central-office fire from which many lessons were learned and subsequently applied to physical node protection.

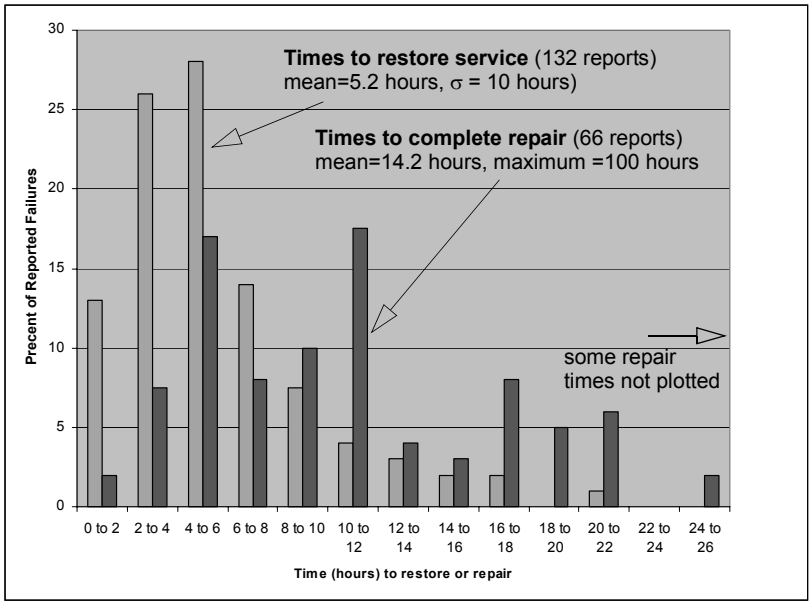


Figure 3-2 Histogram of service restoration and cable repair times (data from [Craw93]).

3.1.3 Effects of Outage Duration

There are a variety of user impacts from fiber optic cable failures. Revenue loss and business disruption is often first in mind. As mentioned in the introduction, the Gartner research group attributes up to $500 million in business losses to network failures by the year 2004. Direct voice-calling revenue loss from failure of major trunk groups is frequently quoted at $100,000/minute or more. But other revenue losses may arise from default on service level agreements (SLAs) for private line or virtual network services, or even bankruptcies of business that are critically dependent on 1-800 or web-pages services. Many businesses are completely dependent on web-based transaction systems or 1-800 service for their order intakes and there are reports of bankruptcies from an hour or more of outage. (Such businesses run with a very finely balanced cash-flow.) Growing web-based e-commerce transactions only increase this exposure. Protection of 1-800 services was one of the first economically warranted applications for centralized automated mesh restoration with AT&T's FASTAR system [ChDo91]. It was the first time 1-800 services could be assured of five minute restoration times. More recently one can easily imagine the direct revenue loss and impact on the reputation of "dot-com" businesses if there is any outage of more than a few minutes.

When the outage times are in the region of a few seconds or below, it is not revenue and business disruptions that are of primary concern, but harmful complications from a number of network dynamic effects that have to be considered. A study by Sosnosky provides the most often cited summary of effects, based on a detailed technical analysis of various services and

signal types [Sosn94]. Table 3-1 is a summary of these effects, based on Sosnosky, with some updating to include effects on Internet protocols.

The first and most desirable goal is to keep any interruption of carrier signal flows to 50 ms or less. 50 ms is the characteristic specification for dedicated 1+1 automatic protection switching (APS) systems. An interruption of 50 ms or less in a transmission signal causes only a "hit" that is perceived by higher layers as a transmission error. At most one or two error-seconds are logged on performance monitoring equipment and data packet units for most over-riding TCP/IP sessions will not be affected at all. No alarms are activated in higher layers. The effect is a "click" on voice, a streak on a fax machine, possibly several lost frames in video, and on data services it may cause a packet retransmission but is well within the capabilities of data protocols including TCP/IP to handle. An important debate exists in the industry surrounding 50 ms as a requirement for automated restoration schemes. One view holds that the target for any restoration scheme must be 50 ms. Section 3.1.4 is devoted to a further discussion of this particular issue.

As one moves up from 50 ms outage time the chance that a given TCP/IP session loses a packet increases but remains well within the capability for ACK/NACK retransmission to recover without a backoff in the transmission rate and window size. Between 150-200 ms when a DS-1 level reframe time is added, there is a possibility (<5% at 200 ms) of exceeding the "carrier group alarm" (CGA) times of some older channel bank[1] equipment, at which time the associated switching machine will busy out the affected trunks, disconnecting any calls in progress.

Table 3-1 Classification of Outage Time Impacts

Target Range	Duration	Main Effects / Characteristics
Protection Switching	< 50 ms	No outage logged: system reframes, service "hit", 1 or 2 error-seconds (traditional performance spec for APS systems), TCP recovers after one errored frame, no TCP fallback. Most TCP sessions see no impact at all.
1	50 ms - 200 ms	< 5% voiceband disconnects, signaling system (SS7) switch-overs, SMDS (frame-relay) and ATM cell-rerouting may start.
2	200 ms - 2 s	Switched connections on older channel banks dropped (CGA alarms) (traditional max time for distributed mesh restoration), TCP/IP protocol backoff.

1. A channel bank is the equipment that digitizes and interleaves 24 analog voice circuits into a DS-1.

Table 3-1 Classification of Outage Time Impacts

Target Range	Duration	Main Effects / Characteristics
3	2s - 10 s	All switched circuit services disconnected. Private line disconnects, potential data session / X.25 disconnects, TCP session time-outs start, web page not available errors. Hello protocol between routers begins to be affected.
4	10s - 5 min	All calls and data sessions terminated. TCP/IP application layer programs time out. Users begin attempting mass redials / reconnects. Routers issuing LSAs on all failed links, topology update and resynchronization beginning network-wide.
"Undesirable"	5 min - 30 min	Digital switches under heavy reattempts load, "minor" societal / business effects, noticeable Internet "brownout."
"Unacceptable"	> 30 min	Regulatory reporting may be required. Major societal impacts. Headline news. Service Level Agreement clauses triggered, lawsuits, societal risks: 911, travel booking, educational services, financial services, stock market all impacted.

With DS1 interfaces on modern digital switches, however, this does not occur until 2.5 +/ - 0.5 seconds.[2] Some other minor network dynamics begin in the range from 150-200 ms. In Switched Multi-megabit Digital Service (SMDS) cell rerouting processes would usually be beginning by 200 milliseconds. The recovery of any lost data is, however, still handled through higher layer data protocols. The SS7 common channel signaling (CCS) network (which control circuit-switched connection establishment) may also react to an outage of 100 ms at the SONET level (~150 ms after reframing at the DS-1 level). The CCS network uses DS-0 circuits for its signaling links and will initiate a switchover to its designated backup links if no DS-0 level synch flags are seen for 146 ms. Calls in the process of being set up at the time may be abandoned. Some video codecs using high compression techniques can also require a reframing process in response to a 100 ms outage that can be quite noticeable to users.

In the time frame from 200 ms to two seconds no new effects on switched voiceband services emerge other than those due to the extension of the actual signal lapse period itself. By two seconds the roughly 12% of DS0 circuits that are carried on older analog channel banks (at the

2. Whether at 230 ms or 2.5 s, it is reasonable to ask why a switch deliberately drops calls at all. One reason is that the switch must "busy out" the affected trunks to avoid setting up new calls into the failed trunk group. Another is to avoid processing the possibly random supervisory signaling state bits on the failed trunks. Doing so can threaten the switch call-processing resources (CPU, memory and real-time) causing a crash.

time of Sosnosky's study) will definitely be disconnected. In the range from two to 10 seconds the effects become far more serious and visible to users. A quantum change arises in terms of the service-level impact in that virtually all voice connections and data sessions are disconnected. This is the first abrupt perception by users and service level applications of *outage* as opposed to a momentary hit or retransmission-related throughput drop. At 2.5 +/- 0.5 seconds, digital switches react to the failure states on their transmission interfaces and begin "trunk conditioning"; DS-0, *(n)*xDS-0 (i.e., "fractional T1"), DS-1 and private line disconnects ("call-dropping") occur. Voiceband data modems typically also time out two to three seconds after detecting a loss of carrier. Session dependent applications such as file transfer using IBM SNA or TCP/IP may begin timing out in this region, although time-outs are user programmable up to higher values (up to 255 seconds for SNA). X.25 packet network time-outs are typically from one to 30 seconds with a suggested time of 5 seconds. When these timers expire, disconnection of all virtual calls on those links occurs. B-ISDN ATM connections typically have alarm thresholds of about five seconds.

In contrast to the 50 ms view for restoration requirements, this region of 1 to 2 second restoration is the main objective that is accepted by many as the most reasonable target, based largely on the cost associated with 1+1 capacity duplication to meet 50 ms, and in recognition that up until about 1 or 2 seconds, there really is very little effect on services. However, two seconds is really the "last chance" to stop serious network and service implications from arising. It is interesting that some simple experiments can dramatically illustrate the network dynamics involved in comparing restoration above and below a 2 second target (whereas there really are no such abrupt or quantum changes in effects at anywhere from zero up to the 2 second call-dropping threshold).

Figure 3-3 shows results from a simple teletraffic simulation of a group of 50 servers. The servers can be considered circuits in a trunk group or processors serving web pages. The result shown is based on telephony traffic with a 3 minute holding time. The 50 servers are initially in statistical equilibrium with their offered load at 1% connection blocking. If a call request is blocked, the offering source reattempts according to a uniform random distribution of delay over the 30 seconds following the blocked attempt. Figure 3-3(a) shows the instantaneous connection attempts rate, if the 50 trunk group is severed and all calls are dropped, then followed by an 80% restoration level. Figure 3-3(b) shows the corresponding dynamics of the same total failure, also followed by only 80% restoral, but *before* the onset of call dropping. Figure 3-3(c) shows how the overall transient effect is yet further mitigated by adaptive routing in the circuit-switched service layer to further reduce ongoing congestion. This dramatically illustrates how beneficial it is in general to achieve a restoration response before connection or session dropping, even if the final restoral level is not 100%.

The seriousness of an outage that extends beyond several seconds, into the tens of seconds, grows progressively worse: IP networks begin discovering "Hello" protocol failures and attempt to reconverge their routing tables via LSA flooding. In circuit-switched service layers, massive connection and session dropping starts occurring and goes on for the next several min-

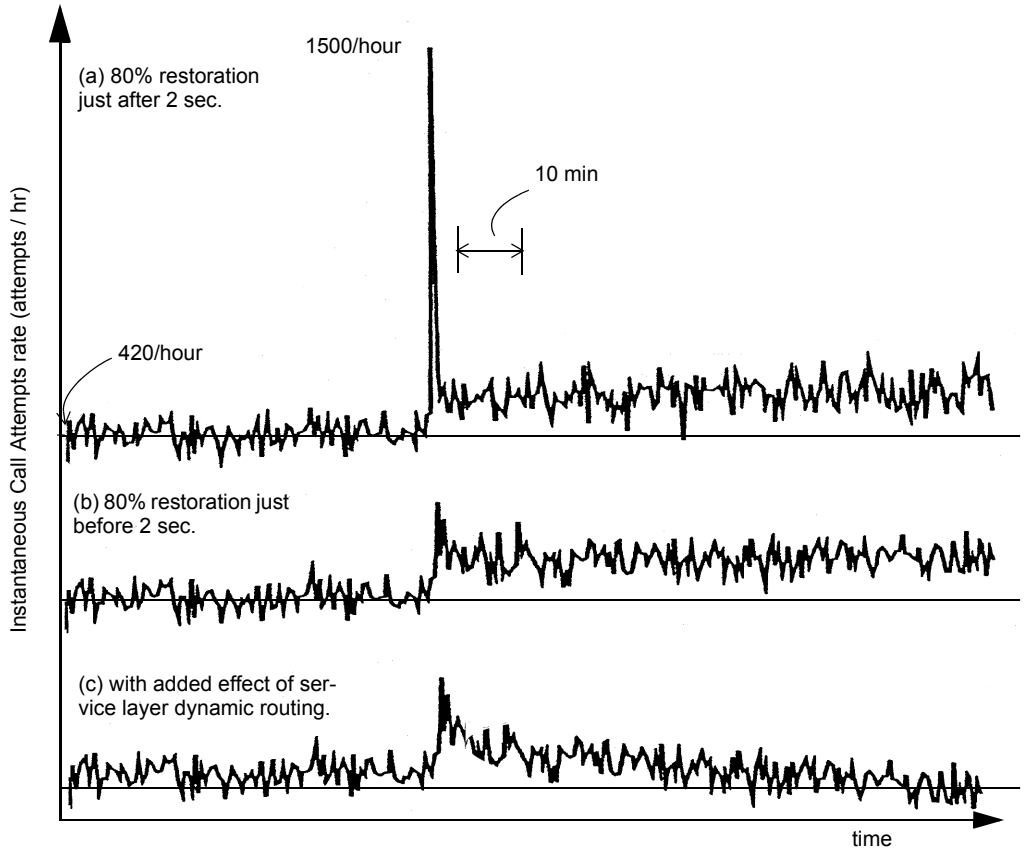


Figure 3-3 Traffic dynamic effects (semi-synchronized mass re-attempts) of restoration beyond the call-dropping limit of ~2 seconds (collaboration with M. MacGregor).

utes. Even if restoration occurred at, say, 10 seconds, there would by then be millions of users and applications that begin a semi-synchronized process of attempting to re-establish their connections. There are numerous reports of large digital switching systems suffering software crashes and cold reboots in the time frame of 10 seconds to a few minutes following a cable cut, due to such effects. The cut itself might not have affected the basic switch stability, but the mass re-attempt overwhelms and crashes the switch. Similar dynamics apply for IP large routers forwarding packets for millions of TCP/IP sessions that similarly undergo an unwittingly synchronized TCP/IP backoff and restart. (TCP/IP involves a rate backoff algorithm called "slow start" for response to congestion. Once it senses rising throughput the transmit rate and window size is multiplied in a run up to the maximum throughput. Self-synchronized dynamics among disparate groups of TCP/IP sessions can therefore occur following the failure or during the time routing tables are being updated). The same kind of dynamic hazards can be expected in MPLS-based networks as label edge routers (LERs) get busy (following OSPF-TE resynchronization)

with CR-LDP signaling for re-establishment of possibly thousands of LSPs simultaneously through the core network of LSRs. Protocols such as CR-LDP for MPLS (or GMPLS) path establishment were not intended for, nor have they ever been tested in an environment of mass simultaneous signaling attempts for new path establishment. The overall result is highly unpredictable transient signaling congestion and capacity seizure and contention dynamics. If failure effects are allowed to even enter this domain we are ripe for "no dial tone" and Internet "brown outs" as switch or router O/S software succumbs to overwhelming real-time processing loads. Such congestion effects are also known to propagate widely in both the telephone network and Internet. Neighboring switches cannot complete calls to the affected destination, blocking calls coming into themselves, and so on. If anything, however, the Internet is even more vulnerable than the circuit switched layer to virtual collapse in these circumstances. [3]

Beyond 30 minutes the outage effects are generally considered so severe that it is reportable to regulatory agencies and the general societal and business impacts are considered to be of major significance. If communications to or between police, ambulance, medical, flight traffic control, industrial process control or many other such crucial services break down for this long it becomes a matter of health and safety, not just business impact. In the United States any outage affecting 30,000 or more users for over 30 minutes is reportable to the FCC.

3.1.4 Is 50 ms Restoration Necessary?

Any newcomer to the field of network survivability will inevitably encounter the "50 ms debate." It is well to be aware that this is a topic that has been already argued without resolution for over a decade and will probably continue. The debate persists because it is not entirely based on technical considerations which could resolve it, but has roots in historical practices and past capabilities and has been a tool of certain marketing strategies.

History of the 50 ms Figure The 50 ms figure historically originated from the specifications of APS subsystems in early digital transmission systems and was not actually based on any particular service requirement. Early digital transmission systems embodied 1:N APS that required typically about 20 ms for fault detection, 10 ms for signaling, and 10 ms for operation of the tail-end transfer relay, so the specification for APS switching times was reasonably set at 50 ms, allowing a 10 ms margin. Early generations of DS1 channel banks (1970s era) also had a Carrier Group Alarm (CGA) threshold of about 230 ms. The CGA is a time threshold for persistence of any alarm state on the transmission line side (such as loss of signal or frame synch loss) after which all trunk channels would be busied out. The 230 ms CGA threshold reinforced the need for 50 ms APS switches at the DS3 transmission level to allow for worst-case reframe times all the way down the DS3, DS2, DS1 hierarchy with suitable margin against the 230 ms CGA deadline. It was long since realized that a 230 ms CGA time was far too short, however.

3. The following unattributed quote in the minutes of a task force on research priorities makes the point about Internet reliability rather imaginatively: (paraphrasing) "What would you do if I grabbed my chest and fell down during a meeting—Dial 911? Imagine opening a browser and typing in http://www.911.org instead?"

Many minor line interruptions would trigger an associated switching machine into mass call-dropping because of spurious CGA activations. The persistence time before call dropping was raised to 2.5 +/- 0.5 s by ITU recommendations in the 1980s as a result. But the requirement for 50 ms APS switching stayed in place, mainly because this was still technically quite feasible at no extra cost in the design of APS subsystems. The apparent sanctity of 50 ms was further entrenched in the 1990s by vendors who promoted only ring-based transport solutions and found it advantageous to insist on 50 ms as the requirement, effectively precluding distributed mesh restoration alternatives which were under equal consideration at the start of the SONET era. As a marketing strategy the 50 ms issue thus served as the "mesh killer" for the 1990s as more and more traditional telcos bought into this as dogma.

On the other hand, there was also real urgency in the early 1990s to deploy some kind of fast automated restoration method relatively immediately. This lead to the quick adoption of ring-based solutions which had only incremental development requirements over 1+1 APS transmission systems. However, once rings were deployed, the effect was to only further reinforce the cultural assumption of 50 ms as the standard. Thus, as sometimes happens in engineering, what was initially a performance *capability* in one specific context (APS switching time) evolved into a perceived *requirement* in all other contexts.

But the "50 ms requirement" is undergoing serious challenges to its validity as a ubiquitous requirement, even being referred to as the "50 ms myth" by data-centric entrants to the field who see little actual need for such fast restoration from an IP services standpoint. Faster restoration is by itself always desirable as a goal, but restoration goals must be carefully set in light of corresponding costs that may be paid in terms of limiting the available choices of network architecture. In practice, insistence on "50 ms" means 1+1 dedicated APS or UPSR rings (to follow) are almost the only choices left for the operator to consider. But if something more like 200 ms is allowed, the entire scope of efficient shared-mesh architectures become available. So it is an issue of real importance as to whether there are any services that truly require 50 ms.

Sosnosky's original study found no applications that require 50 ms restoration. However, the 50 ms requirement was still being debated in 2001 when Schallenburg [Schal01], understanding the potential costs involved to his company, undertook a series of experimental trials with varying interruption times and measured various service degradations on voice circuits, SNA, ATM, X.25, SS7, DS1, 56 kb/s data, NTC digital video, SONET OC-12 access services, and OC-48. He tested with controlled-duration outages and found that 200 ms outages would not jeopardize any of these services and that, except for SS7 signaling links, all other services would in fact withstand outages of two to five seconds.

Thus, the supposed requirement for 50 ms restoration seems to be more of a techno-cultural myth than a real requirement—there are quite practical reasons to consider 2 seconds as an alternate goal for network restoration. This avoids the regime of connection and session timeouts and IP/MPLS layer reactions, but gives a green light to the full consideration of far more efficient mesh-based survivable architectures.

3.2 Survivability Principles from the Ground Up

As in many robust systems, "defence in depth" is also part of communication network survivbility. We will now look at various basic techniques to combat failures and their effects, starting right at the physical layer. Table 3-2 follows the approach of [T1A193] and identifies four levels at which various survivability measures can be employed. Each layer has a generic type of demand unit that it provides to the next higher level. As in any layering abstraction, the basic idea is that each layer exists to provide a certain service to its next higher layer, which need know nothing about how the lower layer implements the service it provides. Here it is capacity units of various types that each layer provides to the next to bear aggregations of signals or traffic formed in the next higher layer. It is important to note that although a layered view is taken it is not implied that one or more methods from each layer must necessarily be chosen and all applied on top of each other. For instance, if rings are implemented at the system layer, then there may be no survivability measures (other than against intra-system circuit-pack level of failures) implemented at the logical layer, and vice-versa. Additionally, certain service layer networks may elect to operate directly over the physical layer, providing their own survivability through adaptive routing. In contrast, however, certain physical layer measures must always be in place for any of the higher layers to effect survivability. In this framework it is usually the system and logical layers, taken together, that we refer to when we speak of "transport networking" in general.

Table 3-2 Layered view of networks for survivability purposes

Layer	Elements	Service and Functions	Demand Units Generated	Capacity Units Provided	Generic Survivability Techniques
Service	IP routers, LSRs telephone switches, ATM switches, smart channel banks	Circuit-switched telephony and data, Internet, B-ISDN private networks, multi-media	OC-3, OC-12, STS-1s, DS-1s, DS-3s GbE, etc.	n/a	Adaptive routing, demand splitting, application reattempt
Logical	OXC DCS ATM VP X-connects	Services grooming, logical transport configuration, bandwidth allocation and management	OC-48, OC-192, wavelength channels, wavebands	OC-3, OC-12, STS-1s, DS-1s, DS-3s GbE, etc.	Mesh protection or restoration DCS-based rings *p*-cycles

Table 3-2 Layered view of networks for survivability purposes

Layer	Elements	Service and Functions	Demand Units Generated	Capacity Units Provided	Generic Survivability Techniques
System	SONET OC-n TM, LTE, ADMs, OADMs WDM transmission systems	Point-to-point bit-transmission at 10 to 40 Gbs/s Point-to-point fiber or wavelengths	fibers, cables	OC-48 OC-192 wavelength channels, wavebands	1:N APS 1+1 DP APS, rings
Physical	Rights-of-way, conduits, pole-lines, huts, cables, ducts	Physical medium of transmission connectivity	n/a	Fibers, cables	Physical encasement, physical diversity

3.3 Physical Layer Survivability Measures

The physical layer, sometimes called Layer 0, is the infrastructure of physical resources on which the network is based: buildings, rights-of-way, cable ducts, cables, underground vaults, and so on. In this layer, survivability considerations are primarily aimed at physical protection of signal-bearing assets and ensuring that the physical layer topology has a basic spatial diversity so as to enable higher layer survivability techniques to function.

3.3.1 Physical Protection Methods

A number of standard practices enhance the physical protection of cables. In metropolitan areas PVC tubing is generally used as a duct structure to give cables a fairly high degree of protection, albeit at high cost. Outside of built up areas, fiber cables are usually direct-buried (without the PVC ducts), at 1.5 to 2 meters depth, and a brightly colored marker ribbon is buried a foot above the cable as a warning marker. There is usually a message such as "Warning: Optical Cable—STOP" on the tape. It is standard practice to also mark all subsurface cable routes with above-ground signs, but these can be difficult to maintain over the years. In some cases where the water table is high, buried cables have actually been found to move sideways up to several meters from their marked positions on the surface. "Call before you dig" programs are often made mandatory by legislation to reduce dig-ups. And hand digging is required to locate the cable after nearing its expected depth within two feet. Locating cables from the surface has to be done promptly and this is an area where geographical information systems can improve the operator's on-line knowledge about where they have buried structures (and other network assets). Cable locating is also facilitated by application of a cable-finding tone to the cable, assuming (as is usual) that a metallic strength member or copper pairs for supervisory and power-feeding are present. Measures against rodents include climbing shields on poles and

cable sheath materials designed to repel rodents from chewing. On undersea cables the greatest hazard is from ship anchors and fishnets dragging on the continental shelf portions. Extremely heavily armored steel outer layers have been developed for use on these sections as well as methods for undersea trenching into the sea floor until it leaves the continental shelf. Beyond the continental shelf cables are far less heavily armored and lay on the sea floor directly. Interestingly, the main physical hazard to such deep sea cables appears to be from shark bites. Several transoceanic cables have been damaged by sharks which seem to be attracted to the magnetic fields surrounding the cable from power-feeding currents. Thus, even in this one case where it seems we might not have to plan for cable cut, it is not so.

Underground cables are either gel-filled to prevent ingress of water or in the past have been air pressurized from the cable vault. An advantage of cable pressurization is that with an intact cable sheath there will normally be no flow. Any loss of sheath integrity is then detected automatically when the pressurization system starts showing a significant flow rate. In addition to the main hazards to "aerial" cables of vehicles, tree falls and ice storms mentioned by Crawford, vandalism and gunshots are another physical hazard to cables. A problem in some developing countries is that aerial fiber optic cables are mistaken for copper cables and pulled down by those who steal copper cable for salvage or resale value. Overall, however, aerial cables sustain only about one third as many cable cuts as do buried cables from dig-ups. And (ironically), while buried cable routes are well marked on the surface, experience with aerial cables shows it better *not* to mark fiber optic cables in any visibly distinct way to avoid deliberate vandalism.

3.3.2 Reducing Physical Protection Costs with Restoration Schemes

The cost of trenching to bury fiber cable can be quite significant and is highly dependent on the depth of burial required. An interesting prospect of using active protection or restoration schemes is that an operator may consider relaxing the burial depth (from 2 m to 1.5 m, say), relative to previous standards for point-to-point transmission systems. An (unpublished) study by the author found this to be quite a viable strategy for one regional carrier. The issue was that existing standards required a 2 meter burial depth for any new cable. It would have been very expensive to trench to 2 meters depth all the way through a certain pass in the Rocky Mountains. But since these cables were destined to be part of either a restorable ring or mesh network, the question arose: "Do we really still have to bury them to two meters?" Indeed, availability calculations (of the general type in Section 3.12) showed that about a thousand-fold increase in the physical failure rate could be sustained (for the same overall system availability) *if* the fibers in such cables were parts of active self healing rings. Given that the actual increase in failure rate from relaxing the depth by half a meter was much less than a thousand-fold, the economic saving was possible without any net loss of service availability. Essentially the same trade-off becomes an option with mesh-based restorable networking as well and we suggest it as an area of further consideration whenever new cable is to be trenched in.

3.3.3 Physical Layer Topology Considerations

When a cable is severed, higher layers can only restore by rerouting the affected carrier signals over physically diverse surviving systems. Physically disjoint routes must therefore exist in Layer 0. This is a basic requirement for survivability that no other layer can provide or emulate. Before the widespread deployment of fiber, backbone transport was largely based on point-to-point analog and digital microwave radio and the physical topology did not really need such diversity. Self-contained 1:N APS systems would combat fading from multipath propagation effects or single-channel equipment failures but there was no real need for restoration considerations in the sense of recovery from a complete failure of the system. The radio towers themselves were highly robust and one cannot easily "cut" the free space path between the towers. National scale networks consequently tended to have many singly connected nodes and roughly approximated a minimum length tree spanning all nodes. Fiber optics, being cable-based, however forces us to "close" the physical topologies into more mesh-like structures where no single cut can isolate any node from the rest. The evolution this implies is illustrated in Figure 3-4.

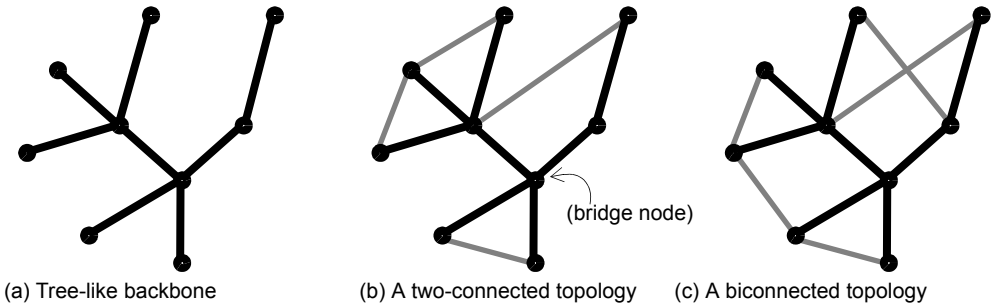


Figure 3-4 For survivability against failures that overcome physical protection measures, the physical layer graph must be either two-connected or biconnected.

Technically, the physical route structure must provide either two-connectedness or biconnectedness over the nodes. In a biconnected network, there are at least two fully disjoint paths between each node pair. Two-connectedness implies that two span-disjoint paths exist between all node pairs, but in some cases there may be a node in common between the two paths. Algorithmic tests for this property are discussed in Chapter 4, although these properties are readily apparent to a human viewing a plan diagram of the network. Note that this topological evolution to a closed graph of some form has by itself nothing to do with the speed or type of restoration scheme to be employed. It is just topologically essential to have at least two physically disjoint routes between every node pair for automatic restoration by diverse routing to even be an option.

In practice, however, the acquisition of rights-of-way to enhance physical layer diversity can be very costly. Whereas a spanning tree requires as few as *N-1* spans to cover *N* nodes, and can do so efficiently in terms of total distance, a biconnected graph requires at least *N* spans (which makes a single but long physical ring) and more typically up to 1.5 *N* for a reasonably well-connected and distance-minimized topology to support either ring- or mesh-based surviv-

able transport networking. Thus, depending on the legacy network or starting point for evolution to mesh-based operation, a major expense may be incurred in the physical layer to ensure that higher layer survivability schemes can operate. Thus, optimization and evolution of the physical layer topology is one of the fundamental problems faced by modern network operators. This problem is treated further in Chapter 9.

3.3.4 The Problem of Diversity Integrity

Related to the creation of physical layer diversity is the need also to be able to validate the details of physical structures that underlie logical protection or restoration routes to ensure integrity of the mapping from physical to logical diversity. For instance, how does one know that the opposite spans of a SONET ring correspond to cables that are on different sides of the street? Maybe they are on one street but two blocks later they share the same duct or bridge-crossing. This is the issue of shared risk link groups mentioned in Chapter 1. It is one thing to recognize that we will have to take SRLGs into account, but the further point here is that even knowing with certainty what the mapping of each logical path into physical structures is (hence defining the SRLGs) is itself a difficult and important aspect of the physical network. This general issue is one of being able to correlate logical level service or system implementations to the ultimate underlying physical structures. This is a significant administrative challenge because cables and ducts may have been pulled in at different times over several decades. The end points may be known, but correlating these to different outside plant conduits and pole structures, etc., is the problem. Many telcos are investing heavily in geographic information systems and conducting ground-truth audits to keep tabs on all these physical plant details. Without assured physical diversity (or at least knowledge of the SRLGs on a given path pair), attempts to provide redundancy to enable active protection or restoration are easily defeated. More about the problem of ensuring physical diversity follows after our review of protection options at all layers.

The "Red and White" Network One interesting proposal to address this physical diversity assurance problem, and provide a very simple and universal strategy for survivability, is the concept of a "red and white" network.[4] The suggestion, not made frivolously, is to purchase and install every physical item of a network in duplicate and classify one of each pair as either "red" or "white," and literally paint every item accordingly. Only one rule then ever need be applied network-wide: always keep red and white apart, whether cables, power supplies, equipment bays, etc. When followed through the result would be an entirely dual-plane network with complete physical disjointedness between planes. Every application warranting protected service would then be realized once in the red plane and again in the white plane, network-wide. The result is assured physical diversity and the operations and planning simplicity of 1+1 tail-end selection as the network-wide survivability principle, for a 100% investment in redundancy

4. While the concept is easy to remember, the author's best recollection is only that this proposal was made in a verbal presentation by T. Sawyer of Southern Bell Communications at an IEEE DCS Workshop in the mid-1990s.

in both node and span equipment. Lest it seem that the idea of completely duplicating the network is unrealistic, it should be pointed out that ring-based networks often embody 200 to 300% capacity redundancy, and although the nodal equipment elements are not completely duplicated in ring-based transport, it is normal to have 1+1 local standby redundancy on high speed circuit packs, processors and power supplies built into these network elements. In contrast each plane of the "red and white" network would use fully non-redundant individual equipment items. Importantly, however, we will see that mesh-based networking can achieve survivability with much less than 100% capacity redundancy and can also provide differentiated levels of service protection.

3.4 Survivability at the Transmission System Layer

Next up from the bottom in Table 3-2 is the "system layer," named in reference to the transmission systems found at this layer. This is the level at which, historically and prior to consideration of mesh-based survivability, almost all active measures to react against single-channel failures or cable cuts have been implemented. This includes mainly linear APS schemes, ring schemes, and the recent *p*-cycle technique, of which we give an overview in this section.

It is important to note that survivability techniques at the system layer also include basic equipment-level design redundancy. Built-in equipment redundancy includes dual power feed connections and power converters, usually dual maintenance and control processors and may often include 1+1 redundant high-speed optical transmit/receive interface cards as well. Standard design methods may also include error-correction coding, in-service bit-error-rate monitoring, loss of signal detectors, laser bias current monitors, and so on. These are all measures that increase the system availability by ensuring its ability to carry on performing its function in the face of faults that may arise *within the system itself*. This kind of designed-in redundancy is extensive in telecommunications equipment: redundant power supplies, processors, tape drives, frequency allocations, antennas, lasers, etc. All this goes in to achieve basic operational availability levels that are often matched only in space, military, and nuclear applications.

Generally system layer survivability schemes for combating cable cuts are *protection* schemes (as opposed to later *restoration* schemes). The main characteristics of a protection scheme is that the protection route and standby capacity are predefined and the mechanism is self-contained within the transmission system layer. It usually involves redirecting the composite optical line signal as a whole without processing or identifying any of its constituent tributaries. The distinction between protection and restoration is not wholly black and white, however, and we later refine the basic categorizations.

3.4.1 "Linear" Transmission Systems

Historic examples of linear transmission systems include point-to-point digital carrier systems operating at the DS-1 rate on twisted pair, up to DS-4 on coaxial cable, and PDH-based transmission systems operating at 12 and 24 x DS-3 rates (565 Mb/s and 1.2 Gb/s) per fiber using proprietary higher-than-DS3 framing structures. Such systems included a 1:N automatic

protection switching (APS) subsystem to protect single-fiber or regenerator failures, but had no designed-in measures to combat a complete cable cut. SONET not only defined standards for the higher-than-DS3 line rates but also extended the capabilities of the transmission systems to include a "nested" 1:N APS configuration which allowed for add/drop multiplexing within the span of the protection channel. Effectively the one protection system would be shared by all the subtending add/drop segments by breaking into and out of the protection channel where needed, as opposed to switching end-to-end over to protection. [Wu92] provides more details. Using any of these so-called "linear" transmission systems (in contrast to later ring systems), the only way to withstand cable cuts was to use 1+1 APS and route the protection system over a physically diverse route. This would establish a "1+1 DP" APS arrangement, DP standing for diverse protection. 1+1 DP is effectively two parallel instances of the same linear transmission system.

3.4.2 Automatic Protection Switching (APS)

Let us now give meaning to the notations 1:1, 1+1, etc. which are widely used in the context of APS systems. 1+1 denotes a dedicated standby arrangement: one working system and a completely reserved backup system in which the transmit line signal is copied (called head-end bridging) and drives both signal paths. 1+1 DP implies that the protection channel is routed over physically diverse rights-of-way from the working system. The fastest possible switching speed is obtained with 1+1 because the receivers need only monitor both receive signal copies and switch from one to the other if either fails. The receiver switch-over is called "tail-end-transfer." 1:1 APS is like 1+1 but the transmit signal is not kept in a bridged state. The resources needed are the same as in 1+1, but 1:1 can allow other uses for the protection channel when not needed by the working channel. 1:1 operation can be of advantage in providing "extra traffic" services (below) or in maintenance and trouble testing because one of the two signal paths can be taken off line for intrusive testing, then the signal "force switched" and the other path separately tested. The switching spped of 1:1 APS is, however, slightly slower than 1+1 because the transmit signal is not bridged at all times and receiver detection of a working channel failure must be signalled back to the head-end to request the head-end bridge establishment to effect protection.

1:N APS denotes that N working systems share one standby "protection" system in an arrangement such as in Figure 3-5. The intent and ability of a 1:N APS system is only to protect against single channel or fiber failures by using a standby channel within the system itself. The standby need not be diverse routed because there is no ability or intent with 1:N APS to protect against a complete cable cut. The protection or spare channel system in 1:N APS is inherently shared since $N > 1$. In 1:N APS the receiving end of a failed channel detects the failure and checks if the spare span is available. If so, it signals to the other end of the system to request a head-end bridge of the failed channel onto the spare span. Return signaling confirms the number of the selected channel and once the spare-span receiver is in-lock at suitably low BER on the new spare-span signal, a tail-end transfer relay substitutes the spare span signal for the original working channel signal at the system output port. A converse head-end bridge and tail-end transfer is also set up for the other direction of transmission on the failed channel. In SONET this is

implemented in a simple state-driven protocol using the K1-K2 overhead bytes on the protection channel. The same protocol also drives the SONET BLSR ring switching mechanism. Although the K1-K2 byte signaling protocol is simple, it is significant conceptually in that it stands as an alternate paradigm to explicit inter-processor data messaging to perform certain time-critical rerouting functions in the most robust and reliable real-time way possible. The distributed restoration algorithms (DRAs) mentioned later for span restoration (Chapter 5) use an extension of this form of signaling to realize generalized mesh rerouting as opposed to APS. The K1-K2 byte protocol involves two finite state machines at each end of the APS system, one associated with its transmit direction, for which it has head-end bridge responsibility, the other for its receive direction, for which it has tail-end transfer responsibility. If the spare span is not already in use, the protocol is:

1:N APS Protocol

```
Tail-end role:
{state = idle; event= receive failure on Ch x ;
action = transmit "x" on K1 byte on spare span; (bridge request)
next state: wait}
{state = wait; event= receive "x" on spare K2; (bridge confirm)
action = tail-end transfer (substitute spare span output for Ch x
system output);
next state = protected}

Head-end role:
{state = idle; event= receive "x" on spare span K1 byte;
action = {set up head-end bridge for Ch x;
         transmit "x" on K2 byte on spare span (bridge confirm);
         transmit "x" on K1 byte on spare span (bridge request);}
next state: wait}
{state = wait; event= receive "x" on spare K2 byte(bridge confirm);
action = tail-end transfer (substitute spare span output for Ch x
system output);
next state = protected}
```

The extension of 1:N APS protection to *k:*N APS where $k > 1$ follows directly except that the protection control logic must then manage allocation of single channel failures on N working channels to k available protection channels.

A possibly confusing industry trend is to refer to shared mesh restoration schemes as achieving *"m for n protection,"* and even denoting this *m:n*. In this context, however, people do not mean to suggest that *m:n* APS systems are literally being employed. Rather they are referring in general to the attribute that mesh networks achieve a certain characteristic sharing of protection capacity an *overall* network basis; that is to say that for every n units of working capacity there are m units of spare capacity ($n > m$) in the network *on average*. This will come to have more meaning as we look at mesh restoration schemes and capacity design; it will become

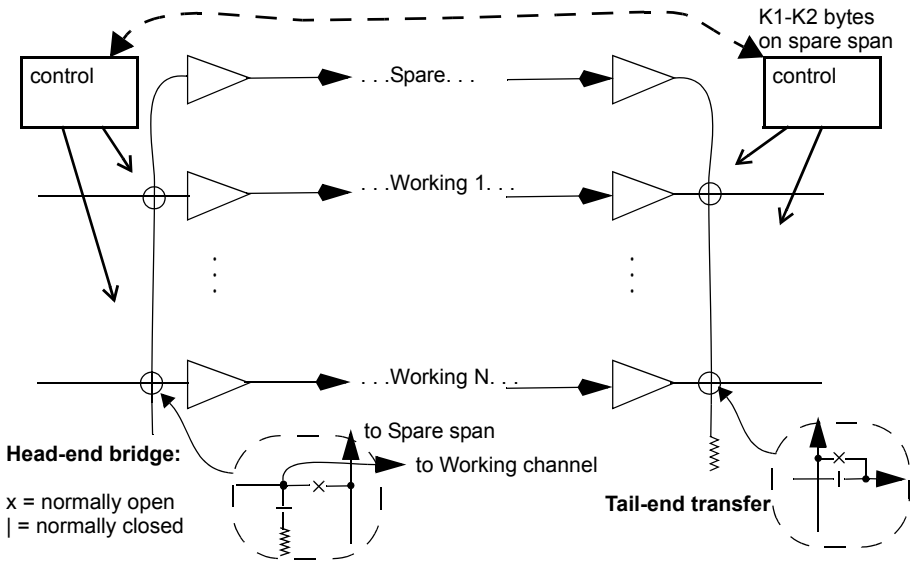


Figure 3-5 Head-end bridge and tail-end transfer functions illustrated in a 1:N APS system.

apparent how such sharing of all protection channels occur over all working channels of the network without implying specifically established *m:n* APS subgroups or systems.

Note that if APS methods are intended to protect against cable cuts they have to be either 1+1 or 1:1 DP APS, not 1:N APS. 1:N APS is usually employed as a high-availability equipment level system design method to protect against *internal* failures of the system and all channels—the N working and one standby—are routed together. But protection against *externally imposed* failures requires 1+1 or 1:1 DP (or other schemes which follow). Typically therefore one finds equipment designs employing combinations of 1:1 APS on the high-speed "line" side (where the risk is of a complete cable cut) and something like 1:7 APS on lower-speed circuit packs or cross-office interconnection interfaces etc. (where the risk is primarily of a single electronics or connector failure). Figure 3-6 illustrates 1:N APS versus 1+1 DP APS.

3.4.3 Reversion

In a 1:N protection switching system, and more generally in restoration or protection of any kind using spare capacity sharing, the protected signal path must be returned to its normal working path (or another working path) following physical repair so that the protection capacity is accessible again for the next failure that might arise. This is usually done with care to minimize subsequent "hits" on the customer payload. To minimize such a reversion hit the usual procedure is to set up a head-end bridge to supply a copy of the signal to the repaired signal path. This does not interrupt the restored signal path. Tests are done while in the bridged state to validate the receive signal quality and then a tail-end transfer switch substitutes the repaired work-

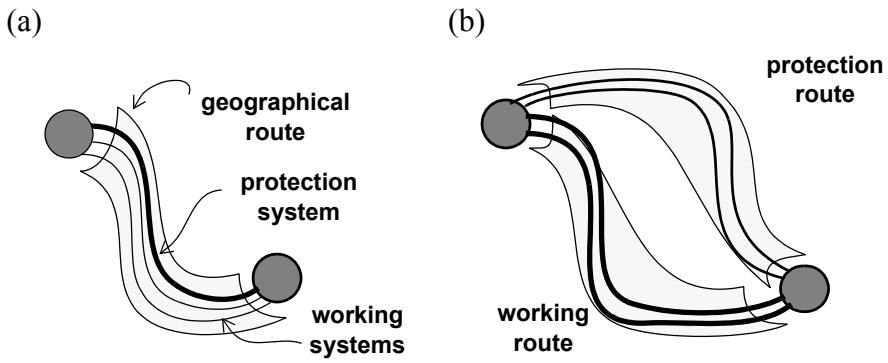


Figure 3-6 (a) 1:N (N=2) co-routed APS and (b) 1+1 diverse-protection (DP) APS.

ing path signal for the restored signal. A hit, usually only of 10 ms or so in duration, arises only due to the tail-end transfer. This is called a "bridge and roll" process.

In systems such as 1+1 APS or a UPSR, reversion is not actually necessary. If done at all it is only for administrative reasons. Generally in any mesh protection or restoration scheme reversion will be required after physical repair, because we always want to return the system to a state of readiness for a next failure. Since we are always using shared capacity for efficiency, this means returning signals to their pre failure routing. This practice also avoids the accumulation of longer-than-required working routes that would result from a non-reversion policy in a mesh restorable network. Unlike the failure that triggered a restoration event, however, reversion itself is a process whose timing and pace the network operator controls following physical repair and can be scheduled late at night and/or coordinated directly with the service users to minimize customer impact.

3.4.4 "Extra Traffic" Feature

1:N or 1:1 APS systems, including their extensions into BLSR rings (to follow shortly) support a practice called "extra traffic." This is a feature that allows the network operator to transport any other lower-priority traffic (in compatible format for the APS or ring's line-rate signal) over the protection channel. Extra traffic is bumped off if the APS or ring system switches to protect its own working channels.[5] The ability to access the protection channels of a ring via the extra traffic inputs of an APS or ring terminal is later—in Section 11.7 (and Section 11.8)—employed a part of a strategy called "ring-mining" for ring to mesh (or ring to *p*-cycle) evolution.

5. An amusing, possibly apocryphal, account about the use of the extra traffic feature is the story that "Hockey Night in Canada" used to be distributed nation-wide during Stanley Cup playoffs as extra traffic on protection channels of backbone digital radio and fiber systems. Hockey enthusiasts would not have been pleased.

3.4.5 AIS Concept

One of the features of transmission systems that helps isolate the location of failures is the generation of Alarm Inhibit Signal (AIS), also called Alarm Indication Signal. The idea is to suppress the propagation of the loss of valid signal in one section of a transmission system from setting off the alarms all the way down the rest of the path. OXC nodes can also perform AIS insertion. To illustrate the concept, consider two OXC nodes in a mesh network connected by a WDM transmission system with several OAs. If one of the OAs has a catastrophic failure, or if a cable cut occurs, the adjacent OXC nodes will register the physical loss of signal and insert an "AIS" on the surviving directions on the failed paths. AIS is a standardized dummy payload structure such as a fully framed SONET signal, but with "all-ones" payload that is easily recognized. As a dummy payload of proper framing, timing, power level, etc., it suppresses downstream alarms and indicates to each downstream node that the signal path has failed upstream but that another node has already realized this. AIS is relevant to survivability strategies in at least three regards:

1. For span restoration AIS techniques ensure that only one pair of custodial nodes are activated.
2. In end-to-end path-oriented survivability schemes, the appearance of AIS at path end nodes is what initiates restoration.
3. In some strategies for mesh restoration, the appearance of AIS on a working channel anywhere in the network can indicate that the channel can be taken over as equivalent to spare capacity for restoration. This is part of the later concept of "stub release" in Chapter 6.

3.4.6 Hitless Switching

"Hitless switching" is a special technique that can be used in conjunction with 1+1 DP APS so that a cable cut would not even cause a single *bit* error. This is not often implemented in practice as the costs are high and seldom is such an extreme performance guarantee really required. However, hitless switching has often been specified in a "wish list" sense in at least the first drafts of the APS subsystem requirements in transmission system designs. More often it has been implemented on digital radio to hide the effects rather frequent 1+1 APS switching actions in combatting multipath fading. It is of interest to recognize this scheme, as it defines the ultimate quality of system-level hiding of physical layer disruptions.

The technique uses an adaptive delay buffer switched into the shorter path of the two (1+1) signal paths at each receiver. In conjunction with a suitably long masterframe alignment sequence, the receiver in Figure 3-7 frames on both incoming signals and adaptively delays the signal copy that is arriving with less propagation delay (A) to bit-align it in time with the later arriving signal copy (B). If signal A is considered the normal working signal and if the buffer delay is greater than the alarm detection time for a loss of signal on either signal feed, then it is possible to switch from signal A to signal B at the buffer output *before damaged bits from A reach the output*. Alternately an error checking code on each signal path can give a byte, col-

umn, or frame-by-frame level selection between delay aligned A and B outputs. Both of these schemes would be realizations of hitless switching in which not one bit error occurs during protection switching. The delay alignment process is called differential absolute delay equalization (DADE). DADE has been employed in digital radio systems to combat fading but seems not to have been used on fiber systems to date.

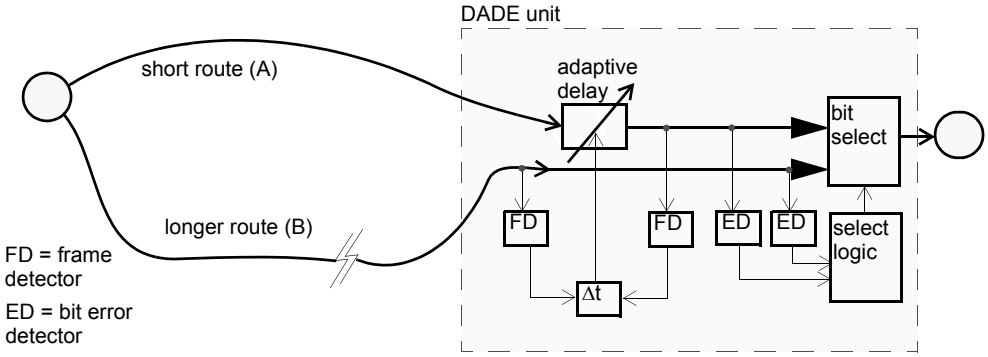


Figure 3-7 The ultimate: 1+1 DP / DADE: “Hitless switching” (only one direction shown).

3.4.7 Unidirectional Path-Switched Rings (UPSR)

Ring-based transmission systems are an evolution from APS systems. It is perhaps easiest to see this with the UPSR which can be seen as packaging up of several logical 1+1 DP APS systems to share a common higher-speed transmission system. A drawback of “standalone” 1+1 DP APS is that the optical line signal (single-fiber OC-n or a DWDM waveband for instance) on each fiber is delivered to the other end as a complete unit, without the possibility to add or drop individual STS or wavelength channels at intermediate locations. Another issue is that unless a large point-to-point demand exists, a dedicated 1+1 APS transmission system may not be justifiable economically. We need some way to fill these large transmission capacities to exploit the economy of scale in cost vs. capacity for transmission. This is in essence what the UPSR [Bell95] does. To understand the role of the UPSR, imagine a number of nodes exchanging, say, single STS-3c demands. Conceptually one could serve each demand with its own 1+1 DP arrangement with OC-3 transmission systems. But if an OC-48 transmission system costs, say, only four times that of an OC-3 system, then it would be better to combine all the individual 1+1 DP requirements to take advantage of a single proportionately cheaper OC-48 transmission technology. This is precisely the idea of the *unidirectional path-switched ring* (UPSR).

Thus UPSRs comprise a number of logical 1+1 APS systems on a set of nodes aggregated onto a common closed-loop path that provides each with the disjoint A and B signal feeds for 1+1 DP APS operation. Nodes in a ring are connected by equal-capacity working and protection fibers (or fiber pairs) in a closed loop or cycle. The diverse route for every working system is the remaining part of the ring of which it is a member. Each unit-demand is transmitted in opposite directions around the ring on both the working and protection fibers. As in 1+1 APS, the receiv-

ing node independently selects the better of the two received signals and has no need for signaling to any other nodes. A UPSR also requires just two fibers. Nodes X,Y exchange a bidirectional working demand pair by virtue of X sending clockwise to Y and Y doing likewise, sending to X around the remainder of the same-direction ring. Thus, each bidirectional signal exchange (X to Y plus Y to X) completes a path around a whole unidirectional ring. ADM nodes are connected by a single working and protection fiber, each of which transmits the line signal in the opposite direction. (Really under UPSR the distinction between one fiber as working the other as protection is arbitrary and it is also a per-channel attribute, not an overall system attribute.) Figure 3-8 illustrates. Under normal conditions, the demand between pairs of nodes in the ring is transmitted on the working fiber in one direction around the ring. A copy of each demand is also transmitted on the protection fiber in the opposite direction. At the receiving node, a path selector continuously monitors the working and protection signals and switches from the working to the protection fiber when the working signal is lost or degraded. Protection switching decisions are made individually for each path rather than for the entire line. SONET standards call for a protection switching time less than 50 ms after detection of signal loss or degradation in the UPSR. This UPSR specification seems to be the general source of the belief that *all* rings give 50 ms switching notwithstanding that BLSR systems in general do *not* necessarily provide 50 ms switch times.

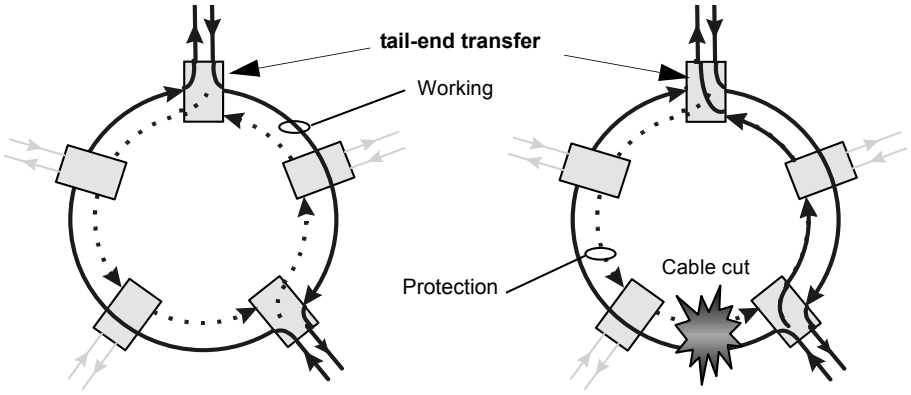


Figure 3-8 UPSR protection switching operation.

An important capacity-planning principle about the UPSR is that because (when considering both A and B signal feeds) the working signal for each demand pair is transmitted all the way around a UPSR, the total demand on any span is the sum of all the demands between all nodes on the ring. This implies that the UPSR line transmission rate must be greater than the sum of all demands served by the ring (regardless of the specific demand exchanged by each node-pair). That is:

$$UPSR\_Line\_rate \geq \sum_{\forall (i,j) \in \boldsymbol{D},\, i \neq j} d_{ij} \tag{3.1}$$

where $\boldsymbol{D}$ is the demand matrix and d_{ij} is the demand from node i to node j. Conversely this means the total demand served cannot exceed the number of channels (i.e., time-slots or wavelengths) provided by the ring. SONET OC-48 and OC-192 UPSR rings have been fairly widely deployed for metro area customer access applications where the hubbed nature of the demand pattern on such a ring makes it basically as efficient as otherwise generally more efficient the BLSR. Recently the UPSR logical structure has been implemented in DWDM technology where each channel comprised a lightpath or a waveband. The optical version is called an Optical Path Protected Ring (OPPR). In Europe the UPSR logical structure is often called an SNCP-ring standing for subnetwork connection protection ring.

3.4.8 Bidirectional Line-Switched Rings (BLSR)

A more efficient arrangement under general demand patterns is obtained when a linear multi-point SONET 1:1 nested APS system is closed on itself. One then obtains what was initially called a shared protection ring (SPring) and now referred to in North America as the bidirectional line-switched ring (BLSR) and in Europe as multiplex-section protected ring (MSPRing). A basic reference documenting the BLSR is [Be1195b]. Unlike the UPSR which uses receive path selection, any *line-switched* ring protects demands by looping the entire working line signal back onto the protection fiber system at both nodes adjacent to a failure. This is a bit more like a 1:N APS in that access to the protection facility must be coordinated at both ends of the failure and signaling is involved. Unlike 1:N APS, however, the protection fiber system has to have equal capacity to the working system so as to be 100% restorable. In a 4-fiber bidirectional line switched ring (BLSR/4), a separate pair of bidirectional fibers is used for working and for protection. Working demands are not permanently bridged to the protection fiber. Instead, service is restored by looping back the working demand from the working fiber to the protection fiber at the nodes adjacent to the failed segment, as shown in Figure 3-9.

A failed segment may include a span, a node, or several spans and nodes. The SONET K1, K2 line-level overhead bytes perform signaling in the SONET BLSR. Because the protection fiber passes through one or more intermediate nodes before reaching its destination, addressing is required to ensure that the APS message is recognized by the proper node and protection switching is initiated at the right pair of nodes. For this purpose, the SONET reserves four bits in the K1 byte for the destination node's ID and four bits in the K2 byte for the originating node's ID. Thus, the maximum number of nodes in a SONET BLSR is 16. A two-fiber bidirectional line switched ring (BLSR/2) operates in the same logical fashion as a BLSR/4 except over predefined working and protection channels groups on each bidirectional fiber pair. As an example which also shows the direct extension to DWDM, a logical BLSR/2 system using 12 wavelengths per fiber might define wavelengths 1 through 6 to protect wavelengths 7 through 12 on the reverse direction fiber. Standards for DWDM optical rings can be expected to emerge just as

for SONET rings but the DWDM version of the BLSR has so far been called the optical shared protection ring (OSPR).

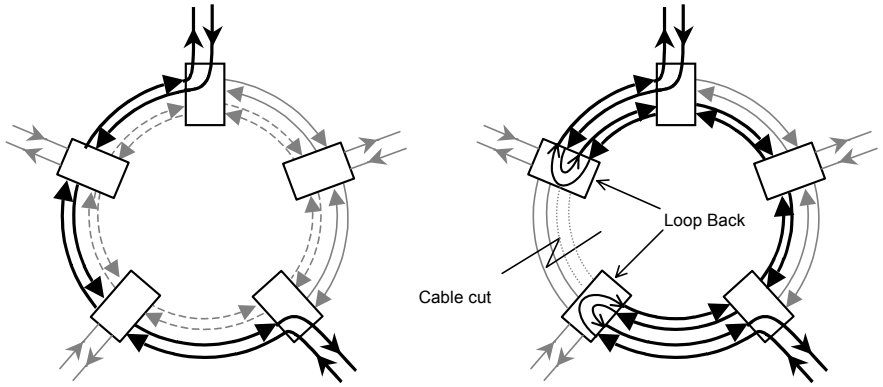


Figure 3-9 BLSR protection switching operation.

An advantage of BLSRs over UPSRs is that channels can be reused around the ring and the protection bandwidth is shared among all working span sections. Because a demand occupies a channel only between its entry and exit nodes, and is usually routed on the shortest path between nodes on the ring, the same channel can be reused for other demands on unused spans of the given path. Since the demands travel directly between the entry and exit nodes (and not all the way around the ring, as in a UPSR), the load on any one span is the sum of demands that are routed over that span. Thus, because the line rate of the BLSR is the same on all spans, the required capacity (i.e., the line transmission rate) of the working and protection rings (or channel groups in the case of a BLSR/2) must be:

$$BLSR\_Line\_rate \geq max_{k \in S}(w_k) \tag{3.2}$$

$$\text{where} \quad w_k = \sum_{\forall (i,j) \in \boldsymbol{D},\, i \neq j} d_{ij} \cdot \delta_{ij}^{k} \quad .$$

w_k is the total demand routed over span k based on the choice of bidirectional routing made for each demand, i.e., either clockwise or counterclockwise around the ring. For Equation 3.2, the indicator parameters δ_{ij}^{k} allow calculation of these span-wise working capacity totals, which unlike the UPSR, may differ on each span. $\delta_{ij}^{k} = 1$ if the route chosen for demands on node pair *(i,j)* crosses span k, otherwise it is zero. Simply put, the capacity of the protection ring has to meet or exceed the largest total of the demand flow crossing any span of the ring. A consequence of this is that (unlike the UPSR) the total demand serving capacity of a BLSR *is* dependent on the demand pattern and the routing choices for each demand on the ring. In a pure-hubbed demand pattern, BLSR efficiency is no better than a UPSR. At the other extreme, the

ideal (but essentially fictitious) pattern of demand for a BLSR is where all non-zero demands are exchanged only between adjacent nodes, and all exchange equal demand totals. Under these ideal conditions the BLSR reaches its best possible redundancy of 100%. In random mesh-like demand patterns BLSRs can be between 200 to 300% redundant (i.e., the ratio of total protection plus unused working span capacity to the total working capacity required using the shortest paths on the graph for the demands served).

Two key issues in planning and operating networks based on BLSR rings are the capacity-exhaustion of a span and the related concept of "stranded capacity." When a single span exhausts (i.e., reaches full working capacity utilization) the whole ring is in effect exhausted with respect to its ongoing usefulness to take up more growth. The condition of equality in Equation 3.2 is then reached on at least one span of the ring. At that point no further demands can be routed through the ring in a way that would cross that span. The side effect of this can be that other spans that have remaining working capacity which is "stranded."

For more details of both UPSR and BLSR as well as APS systems, in both SONET and ATM conexts, see [Wu92] [WuNo97].

3.4.9 Resilient Packet Rings (RPR)

With the dominance of IP packet data as the primary traffic type, a packet-oriented evolution of the BLSR has been developed [Cisc99]. When considering packet data flows, the channelized nature of the line capacity of SONET rings, where capacity units are parceled out to each node pair, restricts the maximum burst rate available to any one data connection on a rigidly channelized ring. It is also fairly management intensive to establish a full set of logical point-to-point OC-12 connections within, say an OC-192 BLSR ring. Data-centric applications may also want to have many more than 16 nodes on the ring and leave the total line transmission capacity available for sharing among all data sources. The 100% reservation for protection capacity is also seen as inefficient when data applications could be using that capacity as well, simply for added performance, during non-failure times. These are some of the motivations for the IEEE 802.17 initiative to develop a Resilient Packet Ring (RPR) standard.

An RPR is a kind of "hollowed out" 2-fiber BLSR. It uses OC-n, wavelength, dark fiber or other physical layer options for bidirectional transmission on each span and a line-level loop-back mechanism for protection. It is "hollowed out" in the sense that circuit-like channelization of the line capacity is abandoned and the entire line-rate transmission capacity is available at each node for packet access. The BLSR attribute of capacity reuse on spans is also achieved by the RPR under a spatial reuse protocol (SRP). Under SRP, destination nodes strip packets off the ring rather than the source node following a full ring transit, as in prior token-ring and FDDI ring standards.

The nodes of an RPR are essentially routers that connect LANs, enterprise data centers, web server farms, etc. to the ring via an SRP Media Access Control (MAC) interface that terminates each bidirectional span. Figure 3-10 illustrates. The access routers can decide to send new packets onto the ring via either of its line transmit interfaces. This inserts new packets onto the

ring in either the long or short directions to their destination so load leveling is facilitated. In the receive direction each MAC unit can either receive a packet destined for it or forward packets en route. Normally a received packet is stripped off the ring but for multicast it can be received and forwarded as well. A pass-through mode also allows express flows to be defined that are essentially invisible to the RPR node, passing directly through at the link layer.

Failures are handled by "wrapping" the ring at both nodes adjacent to the failure. This is conceptually the same as the BLSR's loopback mechanism but the wrapping happens by redirection at the packet level, inboard of the MAC interfaces (rather than at the line signal level as in the BLSR). This allows protection status to be allocated selectively by priority or by other traffic attributes. Obviously, if both counter-rotating fibers are in use for traffic in normal times, the wrapping for protection imposes a sudden additional packet load on the surviving ring spans. Unlike a BLSR, the performance of an RPR under a fiber cut is therefore a matter of *oversubscription* based planning of protection capacity, a topic introduced in Chapter 7. At first instance, it appears that there could be up to 100% oversubscription of capacity in the wrapped state, but the packet-level congestion effects depend on the actual utilization of capacity at the failure time, the mix of protected and unprotected flows, and the adjustments that the SRP fairness protocol will make, as well as the backoff effects that user application protocols may undertake. RPR can thus exploit the soft degradation of services to avoid needing a 100% reservation of strictly unused protection capacity.

For the same reasons, RPR is not easily classified as purely a system layer survivability scheme. It uses a link-level packet loopback mechanism as well as service-level means to accomplish the overall recovery from failure. In reference to the classical data protocol layering it is therefore often referred to as a combined layer 2 and 3 (L2/3) scheme.

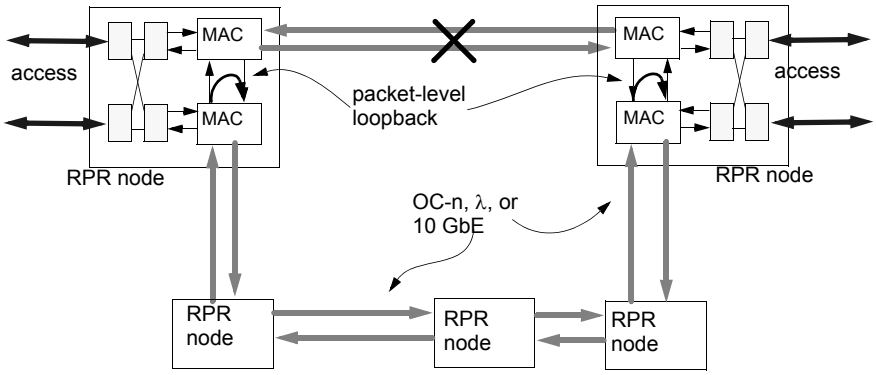


Figure 3-10 Resilient Packet Ring: a combined Layer 2–Layer 3 packet ring survivability technique (amenable to the over-subscription based capacity planning method of Chapter 7).

3.4.10 Ring Covers

A single BLSR ring is often said to be 100% redundant in the sense that each pair of bidirectional working fibers (or channels) is exactly matched by a protection fiber or channel pair. This simple definition is not entirely adequate because it disregards the fact that because of the ring construct asserted on the routing of demands, it may not be possible to usefully fill each span of such working fiber. In practice, when whole transport networks are designed with multiple interconnected rings, the total installed capacity is usually much more than two times the capacity needed only to route all demands via shortest paths over the graph (i.e., the standard working capacity in the sense of Section 1.5.3). Such networks are thus much more than 100% redundant as a whole. Three of the main reasons for this are:

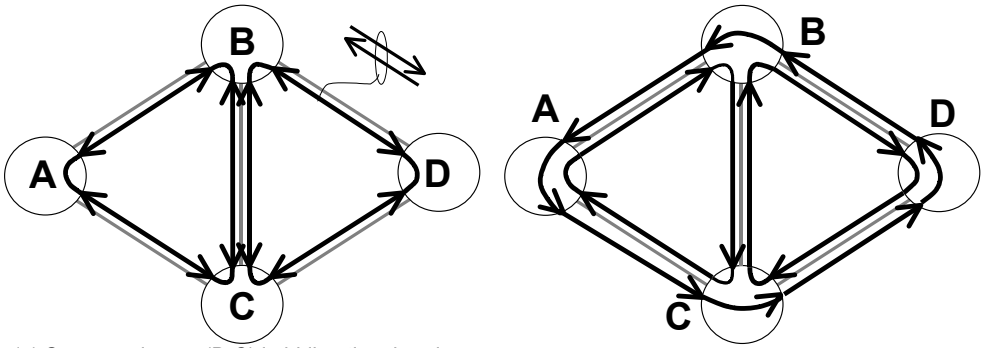
1. Demand routing has to follow ring-constrained paths, not shortest paths over the graph, so demands take longer routes in the first place than they otherwise would, even before their subsequent matching with 100% protection is considered.
2. A set of rings that overlies each span with at most one ring only on each span is usually not possible. In other words, ring covers usually involve some span overlaps.
3. When the working capacity on one span of a ring is filled, it blocks still available working capacity on other spans of the ring from being used for routing.

Unidirectional ring covers are a way of addressing the second of these contributing factors: the problem of span overlaps in trying to lay out a set of rings on the graph. A span overlap is a span whose working capacity could be handled by one ring alone but for purely topological layout reasons the solution requires two rings (each with their own protection) to both overlie that span. The inefficiency of such overlaps can be avoided using unidirectional rings instead of bidirectional ring covers. Formally the technique involves finding an *oriented cycle double-cover (O-CDC)* of the graph [ElHa00]. If we temporarily ignore the other two factors above which bear on the true redundancy of ring networks, the O-CDC principle can achieve ring-type network protection with exactly 100% redundancy in the restricted sense we mentioned of that being the simple ratio of working to protection fibers or channels over the network as a whole.

To explain this let us start by considering an ordinary (i.e., bidirectional) cycle cover, which essentially represents a ring network design based on the span coverage principle. The network demands are routed over working fibers on each span and an overlay of dedicated protection bidirectional fiber pairs, connected in cycles, "covers" each span. If that span fails, the working fiber pair is looped back onto the protection fiber pair, just as in a BLSR. Figure 3-11 shows an example. To cover every span with at least one BLSR ring (or more generally any type of bidirectional cycle which would include a UPSR), the two required cycles unavoidably overlap on span (B-C), in Figure 3-11(a). It is easy to see in general that anywhere an odd-degree node is involved, an ordinary bidirectional cycle cover (where the two directions are locked together on the same cycle) will not be possible without at least one span overlap. The problem with such overlaps is they lay down two working fibers and two protection fibers on a span

where (as previously postulated) working demand flow requires at most one of each. With a single overlap, such a span is effectively 300% redundant instead of 100% redundant (in the simple sense of counting working to non-working fiber or channel ratios).

The motivation behind *directed* or *"oriented" cycle double covers* is to at least improve the situation relative to conventional ring covers, by avoiding such overlaps in planning the cycle covers, so as to get to exactly 100% redundancy (in the sense above) over a network as a whole. Figure 3-11(b) shows how the overlap can be avoided if a set of three unidirectional cycles are used instead of two bidirectional cycles. The total capacity provisioned in (a) is 12 fiber-hops whereas in (b) it is only 10 because the double coverage of span (B-C) is avoided by the unidirectional cycles. Thus with no overlaps and exactly one protection link for each working link on each edge of the graph (or wavelength or waveband level as applicable) we arrive at a class of networks that are exactly 100% redundant in terms of protection to needed working fiber counts. If planned at the whole-fiber level, this gives rise to "4-fiber" networks which have exactly two working and two protection fibers on each span. The practical idea behind this is that with, say, up to 128 or 256 wavelengths per fiber it becomes possible to consider networks that are based entirely and uniformly on just four fibers per span. The 100% redundancy is not retained, however, if some spans need multiple fiber pairs, while others do not.



(a) Span overlap on (B-C) in bidirectional cycle cover, > 100% redundant.

(b) Span (B-C) only covered once under set of 3 oriented (i.e., unidirectional) cycles. 100% redundant.

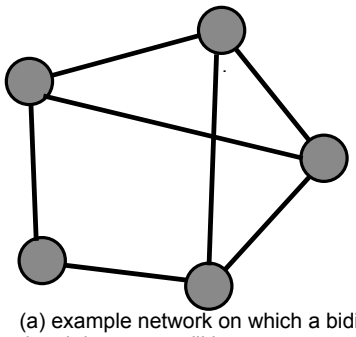
Figure 3-11 Showing how oriented unidirectional cycle covers can avoid the span overlaps that occur in bidirectional cycle covers or BLSR multi-ring network designs.

The key result in [ElHa00] is to show that with an *oriented* cycle double cover the resulting designs can be *exactly* 100% redundant at the fiber level. More detailed explanation is deferred to Chapter 10 where OCDCs are compared to *p*-cycles. For reference, other works on the problem of ring-covers include [GHS94], [KeNa97].

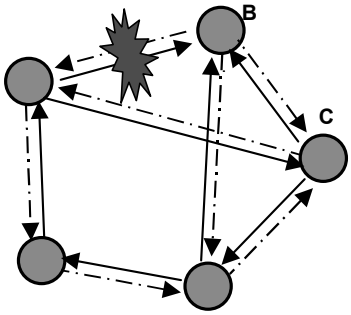
3.4.11 Generalized Loopback Networks

"Generalized loopback" networks (GLBN) were introduced in [MeBa02] and are conceptually related to OCDCs. As with OCDCs the basic idea is to eliminate the use of bidirectional

rings or bidirectional cycle covers, while arriving at an overall design that is exactly 100% redundant at the fiber level (or waveband or wavelength level) on every span. Like OCDCs, a GLBN also assumes a uniform "4-fiber" logical span model: one bidirectional working fiber pair is assumed adequate for all capacity on each span, and a matching protection fiber pair is also provided on each span. Thus, each span level cross-section is 100% redundant in the same (limited) sense as we would say a 4-fiber BLSR is 100% redundant. The difference from OCDCs is that the protection and working fibers of a GLBN are not preconnected into (unidirectional or bidirectional) rings. Instead, a simple flooding-type protocol finds and cross-connects a single replacement path through the protection fibers upon failure. Figure 3-12 shows a small network on which it is impossible to avoid having at least one span-overlap under a bidirectional ring cover, and illustrates how a GLBN works without any predefined cyclic structures.

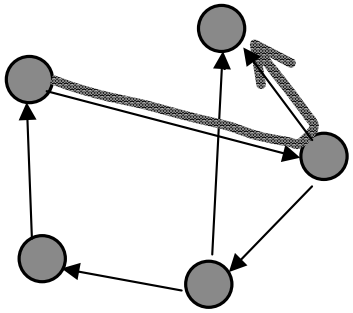


(a) example network on which a bidirectional ring cover will have a span-overlap, exceeding 100% redundancy.

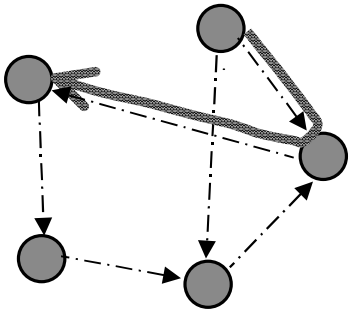


(b) directional decomposition of the working bidirectional transmission on each span.

Working fiber pairs



(c) "Dashed" unidirectional working fiber is protected by a path through the "solid" protection digraph.



(d) "Solid" unidirectional working fiber is protected by a path through the "dashed" protection digraph.

Protection fiber digraphs

Figure 3-12 How generalized loopback works to avoid needing more than 100% redundancy in a 4-fiber span protection environment.

The idea is to divide the bidirectional flow of working demands over the basic graph into two directed graphs (digraphs) where each graph has only one directed working flow on each of

its edges. To do this each direction must be assigned (appropriately, not arbitrarily) to one or the other working digraphs, as in Figure 3-12(b), where the digraphs are denoted "dashed" and "solid." Once this is done, a protection copy of each working digraph is identically defined. Each of the directed "primary" (i.e., working) graphs is then protected by the protection copy of the other primary digraph. For the failure in Figure 3-12(b) the node that was normally transmitting on the "dashed" working link, now loops back onto the "solid" protection digraph (Figure 3-12(c)), and vice-versa at the other end of the failed span (Figure 3-12(d)). The transmission from the nodes next to the failure is actually a flooding copy of the working signal into all outgoing fibers of the anti-directional protection digraph. Other nodes also flood but under a protocol that stems off arrivals of duplicate copies so that the single shortest replacement route in the directed protection graph is all that results.

Not every initial assignment of the two directional working links of each original bidirectional link into the two primary digraphs works to protect all failures, however. In fact the directional decomposition of Figure 3-12(b) works for the failure shown, but not for span (B-C). The key to assignment of the working link directions into the two primary digraphs is that each digraph must remain a connected graph, meaning that at least one *directed* path must exist between all nodes. In Figure 3-12(b) the "solid" digraph is not connected because node B has no path from itself to other nodes on "solid" edges. Medard et al. [MeBa02] give an algorithm for the assignment of directions that ensure the required properties based on finding a directed cycle that visits all nodes.

The result is a network with four fibers on every span with exactly two working and two protection fibers which are like the spans of a single 4-fiber BLSR in that every span consists of a bidirectional working fiber pair and a matching pair of backup fibers. Unlike a BLSR, however, protection inherently takes more generalized routes over the equal-capacity backup network, rather than being restricted to following a particular ring structure. Generalized loopback networks are thus in effect "BLSN"s, where "N" stands for *network* instead of ring and the efficiency they offer relative to a typical 4-fiber BLSR-based network is the removal of the overlapping ring spans that are usually unavoidable in ring-planning. In other words, exactly 100% matching of working and protection resources is achieved, but not worse.

Table 3-3 summarizes the schemes so far discussed, all of which are "ring-like," as defined by virtue of having 100% or higher redundancy. In Table 3-3 logical redundancy refers to the ratio of total non-working fibers or channels to working.

Table 3-3 Overview of ring-like schemes for network protection at the system layer

Logical Redundancy	Scheme or Principle	Logical Equivalences	Notes
> 100%	1+1 DP APS		Basic parallel hot standby redundancy model (head-end bridged)
> 100%	1:1 DP APS	UPSR, SNCP	Permits extra traffic on standby

Table 3-3 Overview of ring-like schemes for network protection at the system layer

Logical Redundancy	Scheme or Principle	Logical Equivalences	Notes
> 100%	UPSR	OPPR, SNCP	Modularized assembly of 1+1 DP APS arrangements
> 100%	BLSR	OSPR, SRING, MSCP ring	Nested linear APS arranged in a closed loop
> 100%	FDDI	ULSR	Unidirectional ring LAN with "wrapping"
> 100%	Cycle cover	Protection fiber pair ring overlay	BLSRs of dark-fiber pairs protecting working fiber pairs
exactly 100%	Generalized Loopback	Unidirectional fiber-level span protection	Like directionally planned SR (without flow spreading) in 1+1 ("4-fiber") span model
exactly 100%	Oriented CDC	Unidirectional planning of shared dark-fiber rings	Assumes "4-fiber" network model

3.4.12 System Layer Protection Without 100% Redundancy: *p*-Cycles

p-Cycles provide another technique that is applicable at the system level using nodal devices that are counter-parts to the ADMs of conventional rings, but *p*-cycles break below the 100% "redundancy barrier" that characterizes other system level techniques. We introduce it here as a system level protection option, but due to its mesh-like capacity efficiency and its applicability at either logical or system layers, we later summarize it with the family of mesh-type schemes. *p*-Cycles seem to be a unique approach in the sense that they have applicability as a system layer solution, but are mesh-like, not ring-like, in their spare capacity requirements. It is the only system level protection technique that does not require a direct matching of working and protection fibers on each span and is typically well under 100% redundant in both logical and true redundancy measures (i.e., the "standard redundancy" of Section 1.5.3).

The easiest way initially to think of *p*-cycles is as a BLSR to which the protection of *straddling* failure spans is added. A straddling span is one that has its end-nodes on the *p*-cycle, but is not itself part of the *p*-cycle. Rather, it is like a chord on a circle. The usual ring-like loopback protection of spans on the ring itself still applies but are called *"on-cycle"* failures, to distinguish them from *straddling span* failures. Upon the failure of a straddling span the *p*-cycle (which is undamaged in such circumstances) provides two protection paths. The simplest and most important distinguishing feature of a *p*-cycle-based network, compared to rings or cycle covers of any kind and generalized loopback networks, is the protection of straddling spans

which themselves may bear *two* units of protected working capacity for each unit of capacity on the *p*-cycles and require *zero* spare capacity on the same spans.

Figure 3-13 illustrates how a *p*-cycle provides single-span failure protection to a small network without covering all spans directly. Figure 3-13(a) shows the example network and we assume that all spans shown bear working capacity to be protected. Figure 3-13(b) shows how rings or any (conventional) fiber-level cycle cover can provide such protection, although we note in this example that it is impossible to use fewer than three simple cycles to do this. The particular ring cover shown is actually an efficient one because it matches up odd-degree nodes to share the same span overlaps (four odd-degree nodes are handled with two span overlaps). An O-CDC or GLBN solution can do better than the bidirectional cycle cover shown but will still be 100% redundant because each span failure is protected by a cycle in a ring-like way or by a directional loopback that forms a cycle with the failed span. In each case the protection is by a direct covering of cycles.

But the same set of single-span failures can be protected by a single *p*-cycle, as in Figure 3-13(c). (Optimal *p*-cycle designs will not in general use only one cycle, although that is adequate using the Hamiltonian cycle shown in this example.) Should any of the spans on the *p*-cycle fail, the *p*-cycle acts just like a BLSR, protecting against on-cycle failures through loopback to protection on the same cycle. The failed signals reverse away from the break and go the other way around the cycle. If any of the three straddling spans shown fail, the same *p*-cycle is broken into at the end points of the straddling failure span and actually can provide a protection path in *both* directions around the *p*-cycle. For that reason, the efficiency of protecting straddling failures is twice that of an on-cycle failure.

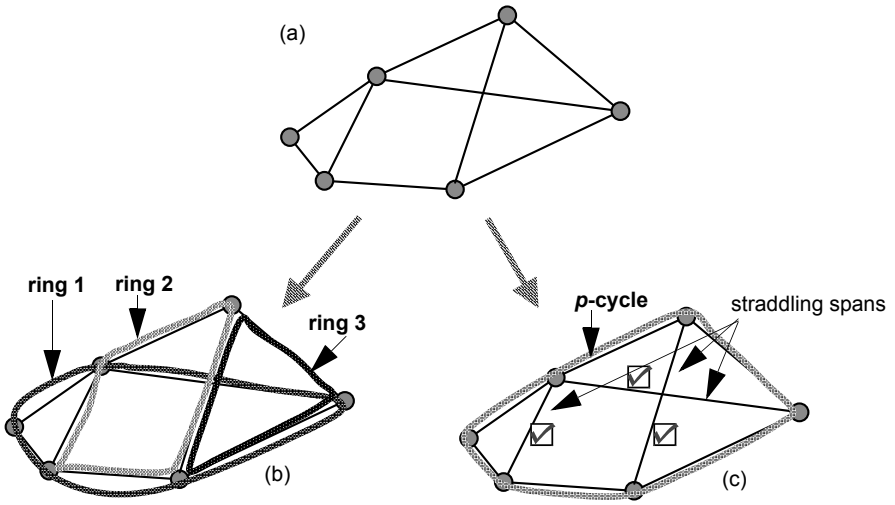


Figure 3-13 A network (a) with a (bidirectional) cycle cover or ring cover (b) in comparison to a single *p*-cycle providing the same protection against single-span failures (c).

In a sense the addition of straddling failure protection is only a minor technical variation on BLSR rings: Chapter 10 shows the nodal structure in more detail, but the nodal elements remain almost as simple as ring ADMs and the switching function needed is exactly that of the BLSR. Nonetheless, even in a network as simple as Figure 3-13(a) it can be seen that the difference in protection efficiency can be dramatic. The ring cover is assuredly over 100% redundancy, because it has three (unavoidable) span overlaps. The ring cover shown actually protects nine working hops with 5 + 4 + 3 = 12 ring protection hops. Purely on a fiber-count basis the cycle cover is then 12 / 9 = 133% redundant.

In contrast the *p*-cycle uses six hops of protection capacity, but protects up to 6 + 2(3) = 12 hops of working capacity making it only 50% redundant. The true efficiency relative to rings is actually higher than in this simple comparison because with *p*-cycles the working demands also go via shortest paths over the graph; only the protection structure itself is formed in a cycle. It has been shown in general that *p*-cycle based networks are essentially as efficient as span-restorable mesh networks to follow in Chapter 5 yet, as will be detailed further in Chapter 10, they require only ring-like, fully pre-configured switching actions known in advance of any failure. Chapter 10 is dedicated to *p*-cycle based network planning and implementation at either the system or logical layer where they can be hosted on OXC-nodes and dynamically reconfigured for demand pattern adaptation.

3.5 Logical Layer Survivability Schemes

System layer protection schemes all rely on essentially fixed transmission and/or protection structures. An advantage of this is that once installed and tested, such systems are discrete identifiable network substructures, their operation is relatively simple and self-contained (i.e., they don't involve a highly general reaction over the network at large), and the restoration path taken for any failure is clearly known in advance. The difference (relative to mesh protection at the logical layer) was once explained to the author that "transmission people are comfortable with self-healing *systems*, but self-healing *networks* are (perceived to be) too general and unpredictable for their liking."

On the other hand, system layer implementations of protection are essentially static, and this can be more to the dislike of the services, planning, and business people within the same companies. If the configuration of as-built systems turns out not to match forecast demand well, it is not at all easy to change the configuration because it is essentially determined by the hardware installation. In addition, if a first-failure has occurred, there is nothing that can be done during the period of physical repair to particularly enhance the readiness of the network to withstand a possible second failure. Fixed system layer protection schemes also do not easily support differentiated quality of protection classes: when whole systems switch at the line rate to protect failures, everyone gets the same class of protection.

This brings us to consider logical layer protection (or restoration) schemes. The flexible ability of the logical layer to create paths on demand between desired end points, out of a general inventory of uncommitted channels on transmission system channels, makes it the natural

domain of a number of survivability schemes with features that are not provided by the ring or APS system layer schemes. Foremost among these considerations is the higher capacity efficiency which can be achieved by "mesh" restoration schemes which permit extensive sharing of protection capacity over non-simultaneous failure scenarios. Capacity efficiently arises not only through sharing of spare capacity but also because cross-connects in the logical layer manage capacity at a finer granularity than in the system layer. As an example, a SONET DCS might manipulate STS-1 and STS3c signals whereas a OC-192 BLSR manipulates the entire 10 Gb/s line signal as a unit for protection purposes. Similarly in an optical network the logical layer OXC will manipulate single lightpaths for routing or restoration, whereas protection actions in the system layer (and capacity allocations to go with those actions) will probably be based on whole-fiber or large waveband levels of manipulation.

3.5.1 Concepts of Protection, Restoration and Distributed Preplanning

System layer schemes are inherently all of the class we will define as protection, while logical layer schemes can be either restoration or protection schemes. Let us now make the distinction. The term *protection* derives from its origin in APS systems. In a 1+1 DP APS system, the switching actions are completely predefined and the protection system is fully connected between its end nodes and in a pre-tested, ready-to-use state. The working system is said to be protected, as opposed to restorable, in these circumstances. If a 1:1 APS is involved, then signaling is required to request the head-end bridge and to bump any "extra traffic" off of the spare span. In addition the protection system has to be tested on the fly for correct transmission of the bridged signal. However, the protection route is completely pre-defined and no cross-connections are needed to create the signal path. It follows that UPSR is the same as 1+1 DP APS and BLSR is the same as 1:1 DP APS in these regards. The term protection is generally used for all these as a category. The main difference in *restoration* is really only that the replacement paths that will reroute the payload signals may have to be found and/or cross-connected in real-time when the failure occurs. Thus, one can say that in a *pure protection* scheme, the backup paths are completely dedicated and ready to bear rerouted working demand flow. And in a *pure restoration* scheme all redundant resources are held in a shared pool until configured on demand for restoration against a specific failure that arises.

Having identified these general distinctions it is important to stress that logical layer implementations of mesh-based survivability can be *either* protection schemes or restoration schemes. Possibly for competitive reasons, the classification of schemes as either "protection" and "restoration" has become rather over-emphasized, and over-simplified and coupled with an almost axiomatic assertion that "protection is fast and restoration is slow" and that the most efficient OXC-based mesh-survivability schemes are all "restoration" schemes. All of these points need to be sorted out. We hope to convey in this overview of the issues, and further in Chapters 5 and 8, that these views are overly simplified and dogmatic. In fact the best possible arrangement for survivability may be the combination of a distributed restoration mechanism embedded in the logical layer which self-generates efficient mesh network protection preplans to withstand

any first-failure and then executes directly providing best-efforts state-adaptive restoration to a second failure, should it arise.

The basic assumption that needs to be challenged is that a process of finding paths in real-time is always slow and that if the replacement paths are known in advance they will always be fast. Neither are necessarily true as generalizations, especially with distributed preplanning to identify paths with a restoration mechanism *in advance of* any actual failure. Moreover, there are really at least *three* basic categories of scheme to consider of which pure protection and pure restoration are only the extremes. The two extremes and intermediate possibilities are detailed in Table 3-4. A somewhat similar categorization appears in [ElBo03] which also surveys a taxon-

Table 3-4 Three Fundamental Classes of Survivability Scheme

Type	Description	Examples	Generic Term
(a)	**Pure Protection:** Protection routes are known in advance and cross-connection is not required to use them: spare capacity is preconnected and needs only be accessed at end-points.	1+1 APS, SNCP ring, UPSR, OPPR, BLSR, OSPR, *p*-cycles	Protection
(b)	**Pure Restoration:** Restoration routes are found adaptively based on the failure and the state of the network at the time of failure; connections to assemble the restoration paths are also made in real-time.	Distributed or centralized adaptive restoration algorithms or distributed restoration algorithms (DRA).	Restoration
(c)	**Intermediate:** Replacement routes are known in advance and cross-connection maps for fast local action are in place at all nodes but cross-connection is required to assemble the restoration path-set in real-time.	Distributed preplanning with Span restoration (SR-DPP), ATM Backup VP, shared backup path protection (SBPP).	Preplanned Restoration

omy of variations between pure protection and pure restoration. The intermediate schemes are in some ways the most promising in terms of combining efficiency and speed and how fast or slow these schemes are is dependent on whether path finding or path cross-connection time dominates, not whether these intermediate schemes are *classified* as protection or restoration. In the *intermediate* category are schemes where the restoration paths *are* fully known before a failure, but spare channels are not cross-connected until a specific failure arises. In this regard neither span restoration using *distributed preplanning* (SR-DPP) nor shared backup path protection (SBPP) can be classified as simply a protection or a restoration scheme.

SR-DPP is an especially powerful technique in that, even if path finding is slow, *distributed preplanning* (DPP) can create (and frequently update) protection preplans that are already in place in the nodes in advance of failure. DPP works by using a series of mock-failure trials

responded to by a distributed restoration algorithm (DRA) embedded in the network or other restoration protocol. For each "dress rehearsal" nodes simply record the local set of cross-connections that constituted their participation (if any) in the assembly of restoration paths for each failure trial. The concept is described more fully in [Grov94] or [Grov97] and in Chapter 5. It is a simple technique that retains all of the generality and database freedom of a distributed restoration algorithm, but provides a "protection" scheme of the intermediate type in Table 3-4. This always-present relationship between restoration and a corresponding preplanned "protection" scheme, derivable through DPP, must be kept in mind. Moreover, it is fundamental that if one solves any variety of spare capacity planning problems for different classes of mesh survivability schemes, that spare capacity can either be accessed adaptively by a restoration algorithm in the real network (which gives certain extra tolerances for error) or, the same capacity planning solution for the restorable network can be used to provide a set of preplanned protection arrangements to be used in each node. Finally, regardless of whether any mesh-based survivability scheme operates in real-time with preplanned protection reactions, or with an adaptive restoration algorithm, there is no difference at all in the capacity required or in the definition of the capacity-planning problem (assuming the restoration algorithm is fully capable in the required path finding role).

The relative speed of the intermediate schemes (following a failure) depends on what dominates the real-time performances: *path-finding* time or *cross-connecting* time. Schemes of type (c) can approach the speed of pure protection if OXC cross-connection is fast, occurs in parallel at all nodes, and, through distributed preplanning (DPP), the protection routes and all local switching actions are completely known in advance. Upon failure, real-time is consumed only for failure notification. All nodes put their most recently preplanned actions into effect, in parallel. In span restoration with distributed preplanning on fast OXC nodes, the most dominant time delay could be the simple dissemination of fault notification. As soon as nodes learn the failure identity they assert an already known, locally stored, spare capacity cross-connection map into effect. This happens in parallel at all networks nodes as soon as notification arrives. In another type of intermediate scheme route-finding can take significant time but "assembly" of the path is virtual and takes essentially zero time. e.g., CR-LDP "redial": once label distribution is complete, there is essentially zero subsequent path establishment delay per se.

Thus, we need to appreciate the range of possibilities between pure protection and pure restoration, but avoid the oversimplifications associated with these categorizations, particularly regarding speed and availability. SR-DPP and SBPP are in particular important intermediate schemes for which no categorical statement about relative speed is really justified other than if based on a detailed implementation study. Depending on the relative speed of path finding to cross-connection either scheme may even approach the speed of pure protection in the same network. A final point related to this discussion is how to refer to spare capacity designed into a network for either protection or restoration purposes. For brevity we will make no further distinction and refer simply to *spare* capacity whether used for protection or restoration.

Let us now return to our overview of restoration or protection schemes that operate in the logical layer. To guide the overview we introduce Table 3-5. Because the book itself is devoted to in-depth treatment of the mesh-based survivability schemes, we will not go into the same depth introducing them here as we did for rings, OCDCs and GLBNs, which we will not be covering further.

Table 3-5 Overview of Logical Layer Mesh-type Survivability Schemes

Scheme or Principle	Short Description or Equivalences	Notes
Span restoration (SR)	Dynamic *k*-shortest paths	Uses a "DRA" (Chapter 5) or [Grov94]
Span protection (SP)	Shared-protection routes pre-planned	Centrally controlled or self-organized by distributed preplanning with a DRA
Meta-mesh (MM)	SR in meta-mesh graph, loop-back in chain subnets	A hybrid between span and path restoration (Chapter 5)
λ–based *p*-cycles	Like a BLSR that also protects straddling spans	ADM-like system level or OCX managed *p*-cycles (Chapter 10)
Path restoration (PR) (with stub release)	MCMF with limited commodity requirements	Theoretically most efficient possible scheme (Chapter 6)
λ–based SBPP	1:N APS sharing arranged over disjoint failures	(Chapter 6)
λ–based SLSP	SBPP on redefined sub-path segments	"Short leap" shared backup protection: overlapped SBPP sub-path setups
GMPLS: OSPF-TE / CR-LDP	Independent path reprovisioning attempts by all affected pairs	No assured speed or recovery level

3.5.2 Span Restoration or Span Protection

In span restoration, restoration paths (or preplanned protection paths) reroute locally around the break, between the nodes of the failed span. In pure span restoration the paths are both found and connected in real-time. Span protection refers to a network operating either as outlined above with DPP-based protection preplans or through centrally computed and downloaded preplans. This type of scheme is sometimes also called link protection or line restoration.

3.5.3 Meta-mesh

Meta-mesh is a variation on span restoration that enhances the spare capacity efficiency of span restoration in sparse network graphs. It involves a combination of ring-like ADM loopback within subnetworks that are chains of degree-2 nodes and mesh-like planning of capacity for restoration flows over the logical higher-degree skeleton of a network containing many chain sub-

networks. It represents a specific partial step toward path restoration. Chapter 5 is devoted to in depth treatment of span restoration including meta-mesh.

3.5.4 *p*-Cycles

p-Cycles were introduced as a system layer technique where they would use modular-capacity nodal elements similar to an ADM and implemented at the whole-fiber or waveband level. However, because the *p*-cycle concept separates the routing of working flows from the configuration of protection structures (not locking these two together as in rings), *p*-cycle based protection is also amenable to logical layer implementations. In this context the OCX nodes can set-up and take-down service paths as demand requires and separately configure and maintain a set of span-protecting *p*-cycles. Such *p*-cycles are established and managed at the logical channel, rather than system level and can be easily changed to adapt to shifting demand patterns. Multi-service priority schemes for access to *p*-cycles for protection can also be fairly easily implemented in a logical layer implementation of *p*-cycles but not in the system layer. Chapter 10 covers *p*-cycles.

3.5.5 Path Restoration

In path restoration (PR) the capacity design and corresponding rerouting problems are posed as multi-commodity maximum flow-like rerouting problems to replace affected paths end-to-end following removal of the failed span from the graph. This may or may not involve conversion of surviving working capacity of failed paths into capacity that is available for use in restoration, an aspect called "stub release." This type of path restoration with stub release is of special theoretical significance because it represents the most efficient possible class of survivable network.

3.5.6 Shared-Backup Path Protection (SBPP)

A related method that is particularly amenable to IP-centric control of optical networks is called shared-backup path protection (SBPP). A prior scheme for ATM VP-based transport networking works in the same logical manner and can also be used for MPLS path protection. The approach in SBPP is simplified relative to path restoration by defining a single fully disjoint backup path for each working or "primary" path. In effect, a 1:1 DP APS arrangement is established at the level of each service path. This simplifies real-time operation as the protection response is independent of where the failure occurs on the corresponding working path (whereas the PR response is failure-specific). Relatively high efficiency is still achieved even though a 1-for-1 APS setup exists because spare capacity is shared over failure-disjoint backup paths. Chapter 6 is devoted to path restoration and SBPP schemes. Chapter 7 treats the application of SBPP to the MPLS layer or ATM VP layer transport where oversubscription-based planning of protection capacity is possible to considerably reduce overall capacity requirements.

3.5.7 Segmented or Short-Leap Shared Protection (SLSP)

This is a variation on SBPP in which SBPP-like shared backup protection paths are set up over several segments or sections of the path, rather than end-to-end over the entire length of the working path. This accommodates a working path that may need to travel through several predefined protection domains. More generally, division of any working path into segments for protection produces a family of options between the extremes of pure span protection and SBPP. When a primary is protected with segment-wise disjoint paths, not end-to-end with a single backup path, availability is improved and protection arrangements can be managed locally within each domain the primary crosses between entry and egress nodes of that domain. By further defining protection domains to overlap, single point exposures to node failure are avoided where the SBPP segments would be otherwise connected in tandem through a single node. The idea of segmented interlaced backup paths was introduced in [KrPr00] and later applied to lightpath protection in [HoMo02b], [SaMu02], [GuPr03]. The concept of *segment-based rerouting* was also studied in [JoSa98] where it was similarly recognized that with ATM backup-VP protection, availability would degrade for long service paths. The methods for SBPP design in Chapter 6 cover SLSP when used on a transformation of the initial demand matrix which converts end-to-end path requirements into apparent demand between the designated segment protection nodes.

3.5.8 GMPLS Automatic Reprovisioning as a Restoration Mechanism

For completeness we should also recognize that GMPLS is being viewed by some as offering a network restoration mechanism in addition to its primary role of provisioning transport paths. The thinking is that since OSPF-TE will eventually produce an updated global view of the topology and available capacity following a failure, then each node will be in a position to begin re-establishing those paths which it lost in the failure using GMPLS to simply "redial" each of their lost connections, over the shortest route following the failure.

It is important to note in this regard that SBPP uses GMPLS to establish a working path, and a corresponding disjoint backup path, but it does so for each path as it is provisioned and ahead of the failure. It thus effects a preplanned protection arrangement that is cognizant of the physical spare capacity present and of failure-coordinated contention or sharing relationships that have been established on each unit of spare capacity. This is significantly different than the direct reliance on OSPF-TE/CR-LDP in real-time following a failure to attempt simultaneous re-establishment of all failed paths. Direct use of OSPF-TE/CR-LDP following a failure involves no preplanned reservation or sharing arrangements for capacity for a backup path. There are therefore considerable drawbacks to direct reliance on GMPLS auto reprovisioning for restoration. All other schemes involve considerations to coordinate or preplan the access to spare capacity for an effective and fast recovery following failure. By effective we mean that guarantees about the restorability level can be made by design. While GMPLS auto reprovisioning would usually be an effective response to isolated failure of a single lightpath, relying on the same method for recovery from a cable cut would seem almost irresponsible. A recovery

response of some form would result but there is essentially no control or assurances that can be given about the duration and effectiveness of the overall recovery pattern that results.

When a cable is cut a large number of independent asynchronous instances of the topology update and path redialing protocol will be triggered. Recovery actions must first wait until the OSPF-TE global view is synchronized in each node. As soon as it is, there will be a mass onset of CR-LDP signaling instances as individual end-node pairs attempt to re-establish their failed paths. Each acts without coordination with the others doing so concurrently. OSPF-TE updates will continue to be generated as the available capacity state changes on each link as CR-LDP seizures of capacity occur. This causes other CR-LDP instances to fail or be initiated with out of date resource information and destined to have to crank back. The overall dynamics of possibly thousands of concurrently activated signaling, capacity seizure, and update dissemination protocol instances, and how they will interact to allocate the available capacity for protection, and how fast the whole process would settle down is quite uncertain. Even without considering signaling contention and fall-back dynamics, it is theoretically impossible to say what restoration level will be achieved because a finite-capacity multi-commodity rerouting problem is being attempted by a greedy (and mutually interfering) set of routing instances. The theoretical issue, treated in Chapter 4 and further in Chapter 6, is that of "mutual capacity"—where one path-finding instance with several choices may, when acting independently, take capacity that makes paths for many other pairs infeasible. This can occur even when there is sufficient capacity for full restoration under more coordinated routing. Thus, GMPLS auto reprovisioning may provide a useful built-in reaction for isolated path failures, but because there can be no assurances about the overall level or distribution of the recovery pattern for the set of paths that fail simultaneously under a cable cut, we do not treat it as a restoration method for use in the logical layer. This technique is more suited to use in the service layer.

3.6 Service Layer Survivability Schemes

Service layer techniques are the last safeguards before physical failures become apparent to user applications and are usually worth having in addition to a lower layer scheme. Unlike lower layer schemes in which costs are incurred for extra ports and explicit protection capacity, service layer schemes are usually software-based implementations that attempt rerouting within the working, but only partly utilized, capacity that is visible at the service layer. A service layer rerouting response can also complement a lower layer response if the latter is incomplete, by logical reconfiguration of its paths, and/or application of service priorities to reduce delay or packet loss.

In addition, a service layer *node* failure, or interface failure on a switch or router, can be best dealt with among the peer layer network elements in the same service layer itself. Unlike methods at the logical or system layers which tend to be very fast-acting but all-or-nothing in terms of their benefit for any given path i.e., they either fully protect the traffic-bearing signals (so the effect is invisible) or do not (so the effect is total outage), service layer methods are generally more gradual and provide a shared "graceful degradation"-type of network response. Typ-

ically, blocking or congestion and delay levels may rise, but a basic functionality continues. Thus, except for extreme cases, service layer restoration methods tend to prevent hard outage per se, trading a performance degradation instead.

Table 3-6 identifies a number of options for service layer survivability. Dynamic routing in circuit switched networks and link-state adaptive routing schemes, such as OSPF in the Internet are the two most traditional service layer schemes. With the advent of an IP-centric control plane, several of the logical layer schemes, in particular SBPP and *p*-cycles, have direct correspondents for use in the service layer as well. The main difference is only that a physical circuit like path entity in the data plane is replaced with a virtual path construct such as a VP or LSP. With IP-centric protocols an essentially identical control plane implementation can establish these service layer constructs, just as GMPLS constructs transport layer constructs. In an MPLS/IP service layer, label-switched paths just replace lightpaths in the prior descriptions of logical layer SBPP and *p*-cycles. Other more service-specific forms of restoration are also possible in the services layer. For instance, circuit switched telephony networks have long-used centralized adaptive call routing (called dynamic routing or dynamic non-hierarchical routing (DNHR)) to re-calculate routing plans in the face of congestion [WoCh99], [Topk88], [Ash91], [IEEE95].

Table 3-6 Schemes for Service-Layer Restoration or Protection

Scheme or Principle	Short Description	Notes
MPLS *p*-cycles	IP-link protecting *p*-cycles formed using LSPs	Conceptually same as span-protecting *p*-cycles in logical layer but formed in MPLS layer and amenable to oversubscription based planning (Chapters 7, 10)
Node-encircling *p*-cycles	*p*-cycles formed as LSPs to protect against node (router or LSR) failure	*p*-cycles for which all flows through a node are straddling flows hence restorable in the event of *node* loss (Chapter 10)
MPLS SBPP	Equivalent to ATM Backup VP Protection	Oversubscription based capacity design (Chapter 7)
MPLS SLSP	SBPP on redefined sub-path segments	Short leap shared protection on LSPs with overlapped SBPP sub-path setups
OSPF (for routed IP flows)	Routing table reconvergence	No assured recovery level if used without a lower layer scheme, uncontrolled oversubscription
OSPF-TE / CR-LDP (for label-switched paths)	Independent LSP "redial"	No assured recovery level if used without a lower layer scheme, uncontrolled oversubscription
Dynamic call routing	Centrally recomputed alternate routing policies	Minimizes circuit-switched trunk group blocking

And, of course, in all data networks, message retransmission and adaptive routing protocols apply. These, and the basic ability of OSPF to update its routing tables following link withdrawal LSAs, are all possible forms of service layer restoration mechanisms, as well as GMPLS auto reprovisioning of MLPS paths. The same basic proviso applies that, by itself, mass independent reprovisioning attempts by every affected end-node pair will have no assured or predictable outcome. But when used in the services layer to complement a logical layer restoration response (if needed), there is much less concern, because any auto reprovisioning activity in the service layer is, in that context, understood as only a best-efforts activity to improve performance following a logical layer response.

Dynamic routing schemes for circuit switched networks are an evolution of alternate routing in teletraffic networks wherein a direct "high-usage" trunk group would be supplemented by shared overflow "final routes." The routing of individual calls is determined at call setup time by first testing the direct route and then possibly one or more alternate routes, subject to loop-avoidance constraints. Dynamic routing schemes today follow this basic pattern but are centrally managed with a typical period of about 10 seconds between updates to a central site on the traffic levels on each trunk group from each site. Centralized algorithms can then update the outgoing first and/or second choice trunk group recommendation at each node based on the destination of calls it is handling. The centralized recommendations are able to take into account the current congestion states in various parts of the network, thereby inherently diverting traffic flows around areas of failure. A main benefit of adaptively updating the routing tables is exploitation of the non-coincidence of busy hour loads in the network.

Note that such updates to the routing plan do not imply rerouting existing connections. The aim is to improve the situation for *new* call (or packet or LSP) arrivals only. This is a natural approach in a pure data or telephony service layer where calls or sessions come and go on a minute-by-minute time-scale, and where users can re-establish their calls if need be and where data protocols retransmit lost data packets. Thus, the aim and approach is to seek adaptations that improve aggregate performance, without too much concern about the fate of any particular call or session. This is in contrast to the lower layer restoration environment where the emphasis is on re-establishment of existing paths, which may be in existence for years, and which may bear the entire traffic between two cities, rather than a single call or data session.

In circuit-switched services efforts may also be made to split the realization of the single logical trunk group between two nodes over physically diverse paths. In addition, optimization algorithms can be used to slightly overprovision the trunk quantities in each group as a general margin against failures or congestion. Other attractive aspects of service layer restoration in general is that different priority statuses for various users or services may be much more easily established and finely assigned. In addition, capacity is managed at a much finer scale so that small amounts of available capacity in larger working channels units manipulated by the lower layer schemes can be accessed to enhance performance.

Service layer schemes may also involve establishment of a full logical mesh of trunk groups or MPLS paths, or a full mesh over a subset of key nodes, so that routing of any connec-

tion is not through more than one other intermediate node of the same service layer. Such high-degree logical connectivity is possible because the resources to support this in service layer networks are essentially virtual, i.e., VPI numbers or LSP labels, etc., not physical cables and routes. A consequence of this, however, is that when a high-degree logical network is established over a sparse physical network there can be escalation or expansion of one physical cut into a large and hard to predict number of logical link failures in the service layer, making it rather uncertain to rely solely on a service layer restoration scheme. This is when it is especially useful to make sure a system or logical layer scheme is in place to hide the whole event from the service layers. This is later referred to as the "fault escalation" issue.

3.7 Comparative Advantages of Different Layers for Survivability

The layered view we have just worked through allows us to see that survivability measures at each layer are for the most part complimentary, not competitive. Physical layer measures are essential and service layer measures always help. And we should always have at least one technique implemented at the system or logical layers but there is really no need to employ both, especially if cost is considered. An important planning decision is thus whether to employ a system layer *or* a logical layer recovery scheme. Two of the main factors in this decision are flexibility and efficiency. With rings, or 1+1 diverse routing, there will be an investment of *over* 100% in redundant transmission capacity because (by definition) both diverse routes cannot be equally shortest routes. With logical layer mesh alternatives this may often be reduced to 50-70% redundancy. In addition, complete flexibility exists with an OXC-based (i.e., logical layer) implementation (1) to adapt the protection stance to changing demand patterns, (2) to evolve the entire protection strategy from one scheme to another and/or, (3) to implement prioritized protection service classes. A summary of other comparative aspects is offered in Table 3-7.

Table 3-7 Comparative Strengths and Weaknesses of Layers for Survivability

Attribute	Transmission System Layer	Logical Cross-Connection Layer	Services (or IP Transport) Layer
example:	***BLSR Rings***	***Span Restoration***	***MPLS SBPP***
Capacity Required	Highest	Middle	Least
Speed	Highest (~50 ms)	High (~ 100–300 ms typ.)	Slowest (seconds–minutes)
Certainty / predictability	Highest	High	Lower
Multiple Quality of Protection (QoP)	None	Easily supported on per path basis	Easily supported

Table 3-7 Comparative Strengths and Weaknesses of Layers for Survivability

Attribute	Transmission System Layer	Logical Cross-Connection Layer	Services (or IP Transport) Layer
Provisioning view (working)	Ring-constrained shortest path	Shortest path	Shortest path coordinated to be disjoint with protection
Provisioning view (survivability)	Inherent once routed	Checked upon shortly after routing	Coordinate protection sharing arrangements network-wide
Degradation characteristics (if restoration fails)	Abrupt and total outage	Abrupt on affected channels, may be partial	More graceful degradation; congestion not outage
Oversubscription strategies	No	SONET or WDM: no ATM VP: yes	Yes
Customer control	Least	Through VPN services	Most
Database and protocol dependencies	Least: a "hardwired" implementation	Little: event-driven protocols in firmware interacting on overhead bytes, network state is database	Highest–large databases of global network state, dissemination protocols, software dependent
Susceptibility to SRLG effects and fault escalation	Least, controlled during planning	Low, especially with adaptive distributed restoration	Highest vulnerability to SRLG effects and physical-to-logical fault expansion

Multi-Layer Protection: Containing the Inheritance of Dependencies In thinking about the different layers where we can implement survivability, the issue of physical to logical fault multiplication is critical. Adequate knowledge of SRLG relationships may be extremely difficult to obtain (or maintain) if there are several steps of the emergence and inheritance of failure dependencies, to use the terms introduced in [OePu98]. At every layer of routing abstraction, new fault dependencies *emerge* and are *inherited* by all higher levels. The growth in complexity of determining physical diversity between paths as one goes higher up the hierarchy from physical toward service layers is conveyed in Figure 3-14, based on [OePu98]. Graph *G* shows the layout of cables which in this case involves some degree 1 nodes. As mentioned, a first step is to create a biconnected physical graph. Doing so in this example would remove some of the dependencies in *G'* emerging from *G,* but not all. Even when *G* is biconnected, dependencies between transmission systems are impossible to avoid as long as the systems are allowed to pass through nodes without terminating. They are especially frequent if least-cost routing of each system is desired. For example, we could close *G* with respect to stub-node 7 by adding a cable (7-8) but transmission systems (6-5) and (7-5) would likely remain dependent because span (5-7) in *G* is on both of their shortest routes. Observe also that node 11 is a junction in the

cable graph but has no corresponding appearance in the higher level logical graphs. This is the classic case of a common duct (here, 5-11) creating dependency between what are otherwise viewed as separate transmission spans (6-5) and (7-5) in *G'*. When one routes lightpaths over theses transmission systems, still further dependencies emerge where lightpaths share transmission systems and the prior dependencies from the cables to systems layer are inherited.

The example in Figure 3-14 goes up only two layers above the cables and considers only eight top-layer nodes. In practice if the *G"* shown is the lightpath service layer, then service paths at the STS-3c level routed over them have at least one or two more layers of dependency emergence and inheritance. The overwhelming impression, extrapolating from Figure 3-14 as a simple example, is that it may be difficult to give a robust assurance of full survivability against a cable cut if operating higher up in the hierarchy. Diverse STS-3c level paths would be able to protect the corresponding service against same-level failures or a single lightpath failure (one layer down), or perhaps also a single transmission system failure (two layers down). For example, STS-3c level 1+1 diversity can protect against an STS-3c interface port failure on the host router or against a lightpath failure (including access multiplexing) one layer below. But at three layers of reach-down (to the cables) it seems far less plausible that we would always be certain that STS-3c primary and backup paths would have no inherited dependencies.

In practice this a compelling reason to use protection strategies at the service layer and at either the system or logical layers. Diversity measures at one level can realistically be expected to protect against single failures with known dependencies one or two levels below, or at the same level, but it is probably unrealistic to expect a services layer diversity mapping to retain complete physical disjointness more than two layers below. One set of options to consider is rings, *p*-cycles, or mesh protection implemented with whole-fiber cross-connects directly over a biconnected physical cable graph, *G*. In this case there *are* no emergent dependencies to be inherited by higher layers since each cable span becomes a directly protected single-failure entity. This is simple, robust, and requires relatively low-cost devices for whole-fiber protection switching. It is not very fine-grained, however, and the devices used for protection have no secondary use such as for dynamic service provisioning (other than provisioning dark fiber services).

Alternately, logical layer measures implemented in *G'* using cross-connects for mesh-protection at the channel level are more agile, multi-purpose, and fine-grained in capacity-handling and only have to cope with one level of known dependencies, such as arise in *G'* from the cable junction node 11 in *G*. Using SBPP in *G"* (instead of a protection scheme in *G'*) is not infeasible, but we see a major complexity associated with this alternative because now we are two layers above the level at which physical faults occur—so the complete map of dependencies is far more complex. To go yet another layer up and rely on MPLS autoreprovisioning or MPLS-level SBPP, it becomes hard to imagine that we could support a claim of protection (implemented at that level) against failures stemming from the physical layer, because *G* is a full three levels below the MPLS layer. This all suggests a practical principle that the emergence and inheritance of SRLG-like effects may need to be "contained" by an appropriate protection arrangement

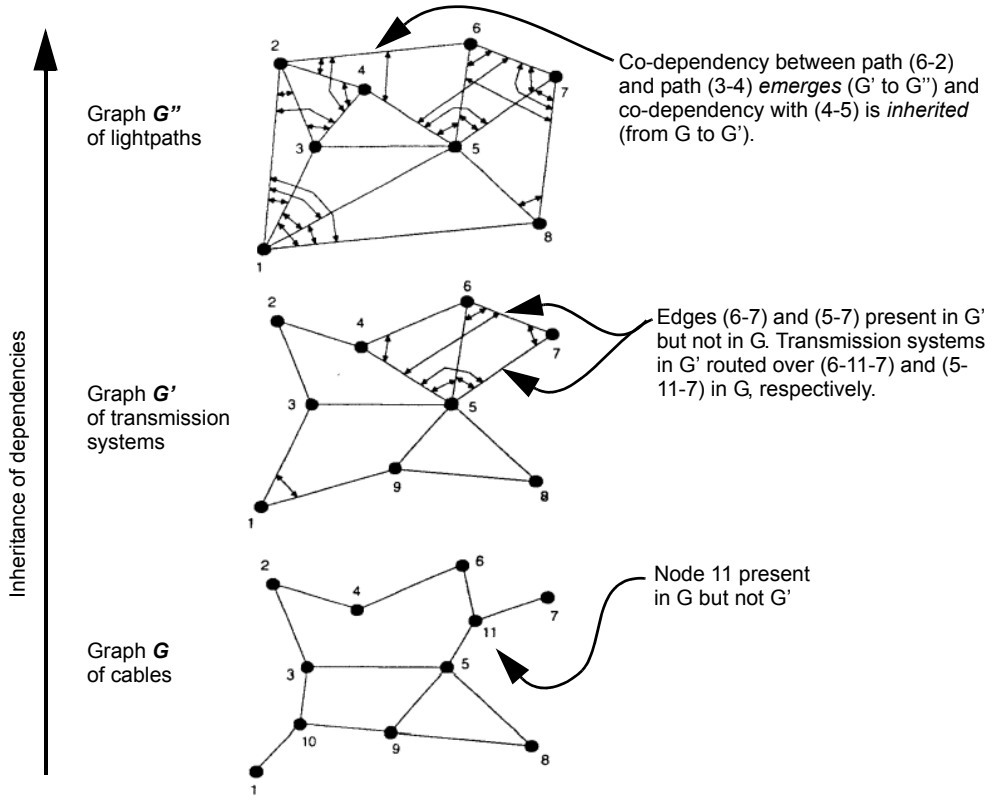


Figure 3-14 Illustrating the fault dependencies that emerge and are inherited by higher levels (adapted from [OePu98]).

every *two* layers. If followed, this leads to a strategy of choosing some basic "infrastructure" protection scheme at the system or logical layers and complimenting it (possibly only for high priority service paths) with an additional technique at the corresponding service layer itself. For example system level *p*-cycles and service level 1:1 APS would be one combination. Lightpath level SBPP complimented by MPLS layer *p*-cycles would be another viable combination, and so on. Note in this regard that even in an ideal "IP over WDM" network, the *three* layers (*G, G', G")* in Figure 3-14 all still exist. Where the reduction of levels occurs in IP over WDM is actually in the levels above the lightpath layer.

3.8 Measures of Outage and Survivability Performance

Let us now introduce various quantitative measures of failure impact, given a failure occurs, and of intrinsic survivability performance in terms of the ability to resist failures in the first place. Given the impact of failures, there is growing regulatory interest in attempts to quantify the magnitude of the impact of various failures that occur. The notion is that hurricanes, tor-

nadoes and earthquakes each have a system for classification of their severity, so why not network failures too? Network operators are also interested in such standardized measures for quality improvement and competitive processes. A second sense of "measuring survivability" is to ask about those intrinsic properties of a network that by design make it less likely to sustain an outage in the face of failures within itself. These are the basic notions of reliability and availability and what we define as the restorability. Let us touch on these in sequence.

3.8.1 McDonald's ULE

McDonald [McDo94] was perhaps the first to advocate development of quantifiable measure of network outages. McDonald's argument was that any drive toward a standard method for quantification for network failures would focus attention on the issue and inevitably lead to improvements in avoiding outage. His proposed measure is the "User-Lost Erlangs" (ULE) defined as:

$$ULE \equiv \log_{10}(E \cdot H) \tag{3.3}$$

where E = average historical traffic intensity (in Erlangs) through the outage period and H = outage duration in hours. The measure is logarithmic, like the Richter scale. 10 Erlangs blocked for one hour is 1 ULE. 10 ULE is equivalent to an hour-long outage affecting 100 Erlangs of normally offered traffic, or 6 minutes of outage on 1000 Erlangs, etc. The logarithmic nature is a key idea for its utility. McDonald argues a logarithmic measure discriminates well between events of major and minor consequences. And it reflects a plausible belief that the overall societal impact somehow scales with the exponent of the total outage. We would add that it is also appropriate to avoid false precision: the data going into a ULE calculation will at best be estimates, so what is important is indeed the order of magnitude, not linear differences.

3.8.2 The (U,D,E) Framework for Quantifying Service Outages

The ULE notion was developed further with an eventual aim toward standardization in [T1A193]. In this framework the impact of a failure is assessed in terms of: Unservability (U), Duration (D) and Extent (E), called a (U, D, E) triple. The three parameters of the (U, D, E) framework are:

Unservability (U): is defined in terms of a basic capability and unit of usage appropriate to the application. For example, in a circuit switched network, this would be the ability to establish connections with acceptable blocking and transmission performance. The unit of usage is a call attempt and the unservability is the percentage of call attempts that fail. In a packet network, the unit of usage is a packet and the unservability is the percentage of packets that were not delivered within a stipulated delay. In a leased line network, unservability is defined as the percentage of DS-0, DS-1 or DS-3 leased signals that are *not* available.

Duration (D): is the elapsed time interval during which performance falls below the threshold for defining unservability.

Extent (E): reflects the geographic area, population affected, traffic volumes, and customer traffic patterns, in which the unservability exceeds a given threshold.

The idea is not to operate on U, D, E values further boiling them down to a single measure but to preserve them as a three-dimensional characterization of any outage event. A *(U, D, E) triple* can thus be plotted in the corresponding 3-space for classification of the event as catastrophic, major, or minor, depending on which predefined "volume shell" the (U, D, E) vector enters. Figure 3-15 illustrates. It seems reasonable that some vector weighting scheme might

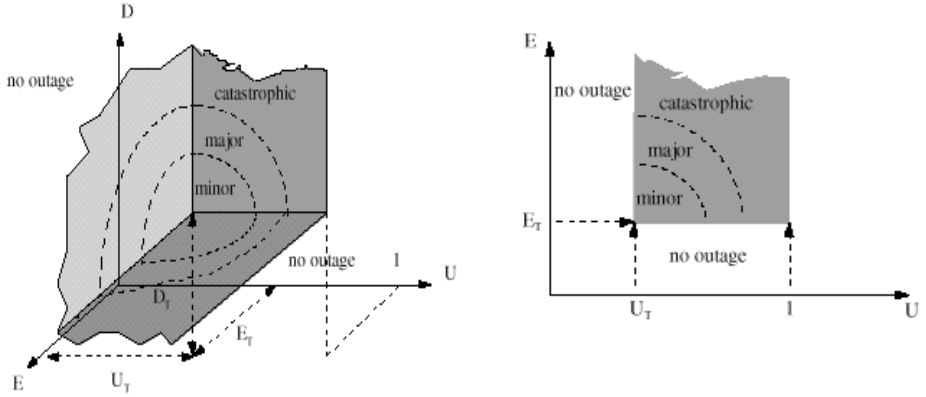


Figure 3-15 (U, D, E) concept for classification of network outages (source: [T1A193]).

also be agreed upon for definition of the qualifying regions. Or, conversely, a general (U, D, E) classification model would not necessarily have simple spherical shells for defining classifications unless the intent is to give strictly equal weight among U, D, E.

(U, D, E) shells can be used both to categorize events as well as to lay down prescriptive policy for what might constitute an event requiring a company review of an incident or methods. For example, a Local Switch failure may be defined to have occurred whenever 500 subscriber lines (the extent E) are totally isolated (the definition of unservable, U) for 2 minutes or more (the duration, D), i.e., (U, D, E) = (100,2,500).

Outage Index: Later work in the T1A1.2 committee that produced [T1A193] considers an approach leading to a single *Outage Index*. It is conceptually equivalent to formation of a vector weighted magnitude of the (U, D, E) triple but involves predefined nonlinear weighting curves for D, E and discrete multipliers for time of day, type of trunk affected (inter- or intra-LATA, 911 etc.) Such weightings are ultimately arbitrary but nonetheless can be fully detailed in a standardized method and then be of valuable service when applied industry-wide.

3.9 Measures of Network Survivability

Measures of outage are different than measures of survivability. An outage is an event that arises from a failure that has actually occurred. Survivability is, however, the ability of a network to continue to provide service in the event of a failure that might arise. Survivability is,

thus, an inherent attribute of the network design or technology employed regardless of if, or how often, failures actually occur. The term survivability itself is not usually given quantitative meaning but is used in general to refer to the quality of being able to keep on functioning in the face of either internal failures or externally imposed damage. The quantitative measures of survivability are necessarily more specific.

One class of such measures are called *conditional* or *Given Occurrence of Failure (GOF)* models. In these measures, each relevant failure scenario is first postulated to have occurred, then an assessment of survivability is made. These tend to be design-oriented measures since they reflect the merit of a survivable design over a pre-specified set of failure scenarios against which coverage is to be ensured but do not depend on knowledge or assumptions about how often such failures may actually occur. They deal with questions such as, "*If* failure x occurs, how well are network services protected from it?"

The other general class of survivability measures aim to take into account the probability of failure onset as well as the survivability response or capability of the network. These are called *Random Occurrence of Failure (ROF)* models. In contrast to the GOF orientation, ROF measures typically ask questions such as: "How likely is it that a path between nodes has an outage over x minutes in any given year?" ROF models are usually based on the assumption that failures can be characterized by random variables with given probability distribution functions and are thus closely related to the fields of reliability and availability which we will review.

3.10 Restorability

A simple GOF-type measure that is widely used in design and characterization of transport networks is the *restorability,* also sometimes called the restoration ratio. Restorability is the most basic indication of survivability because it directly reflects the extent to which "single points of failure" have been removed as outage-causing circumstances. The biggest single step toward survivability is to eliminate single-span failures as a cause of service outage. This has a quantum effect on improving service availability as service outage can then only arise from much less frequent dual failures or node failures. As most commonly used the restorability is defined as the fraction of payload-bearing (i.e., "working") signal units that are subsequently restored, or that are topologically capable of being restored by replacement routes through the network. That is, for a specific failure scenario X,

$$R_X \equiv \frac{\sum_{(i,j) \in X} min(w_{i,j}, k_{i,j})}{\sum_{(i,j) \in X} w_{i,j}} \tag{3.4}$$

where (most generally) $w_{i,j}$ is the number of service paths between nodes *i,j* that are failed in the failure scenario X. This way of stipulating a failure scenario is totally general; any number of span and/or node failures can be represented in terms of the set X of *i,j* node pairs that simulta-

neously have one or more failed paths in scenario X. Thus the denominator of Equation 3.4 can be thought of as a "total damage" sum in terms of the number of transport signal units that are severed in the failure scenario *X*. The numerator is the sum of what is restored (or can be restored) for each subset of failed signal units corresponding to a damaged span. $k_{i,j}$ represents the number of replacement (restoration) paths that can be provided for *(i,j)*. The *min(-)* operator ensures that no credit is given for providing more restoration than is actually needed for any subgroup of failed working signals.

One set of failure scenarios that is of particular practical interest is the set of all single and complete span failures. That is the set of all *X* which just one *(i,j)*. In this case the restorability for any one scenario *m = (i,j)* simplifies to:

$$R_m = min(w_m, k_m)/w_m \tag{3.5}$$

and the *network restorability* is defined as the average restorability of all working paths that are failed under each single-span failure scenario. That is:

$$R_n = \sum_{m \in S} min(w_m, k_m)/\sum_{m \in S} w_m = \sum_{m \in S} R_m \cdot w_m/\sum_{m \in S} w_m \tag{3.6}$$

where *S* is the set of all spans in the network. $R_n = 1$ is often referred to as a "fully restorable" network. It is the mark of a network that can withstand any single-span failure without any service path outage. As a single figure of merit for network survivability R_n is of considerable practical interest because:

a. The likelihood of failure scenarios containing more than one (independent) span failure at a time is much lower than a single failure.
b. It is generally considered economically feasible (or at least necessary and reasonable) to design for $R_n = 1$ whereas it may be economically infeasible to protect against all possible multi-span or node failures by design.
c. R_n is a property of the network design, or current network state, that is independent of any knowledge or assumptions about actual failure frequencies or mechanisms.
d. Given the much higher failure rate of cables (outside plant structures in general) relative to node failures, achieving $R_n = 1$ by design is the most significant single step that can be taken in practice toward improvement of service availability.

A variety of purpose-specific variants from the basic definition of restorability are common. Examples are the "prompt restorability" which is the restorability level arising before a certain elapsed time from failure onset, or the "dual-failure restorability" which is as the name suggests and is considered further in Chapter 8. Other measures can include prioritized demand weightings R_n. These are all valid measures as long as their specifics are fully stipulated in terms of the specific set of failure scenarios being considered and the criteria being employed to define survivability against those failures.

Restorability, and GOF measures in general, are relatively simple to compute and to understand, because they reflect simple measures of recovery levels for a specific set of assumed failure scenarios. In contrast, ROF measures can be much more involved and/or require simulation. A grounding in reliability and availability theory is required for their appreciation. Let us therefore now cover the basic concepts of reliability and availability which underlie ROF measures, and are also highly relevant to work in network survivability in general.

3.11 Reliability

In ordinary English, "reliable" is a qualitative description, meaning that something or someone is predictable, usually available when needed, follows through on promises, etc. But the technical meaning of reliability is quantitative and much more narrowly defined [BiAl92], [OCon91]:

> Reliability is the probability of a device (or system) performing its purpose adequately for the period of time intended under the operating conditions intended.

In other words, reliability is the probability of a system or device *staying* in the operating state, or providing its intended service uninterrupted, as a function of time since the system started in a fully operating condition at *t=0*. A reliability value can also be thought of as answering a mission-oriented question of the form: If the device or system was started in perfect working order at time *t=0*, and the mission takes until *t=T* (with no opportunity for external intervention or repair), then what is the probability that the system will work failure-free at least until the end of the mission?

Reliability is thus always a *non-increasing function of time* with $R(0) = 1$ and $R(\infty) = 0$. When someone says the reliability of a system or device is a specific number, 0.8, say, there is usually some understood interval of time that is implicitly assumed. They are really saying the the probability of the device or system working that long without any failure is 0.8. More formally, the *reliability function*, also called the survivor function is:

$$R(t) \equiv prob\{\text{ no failure in interval [0,t]}\} \tag{3.7}$$

and the cumulative failure distribution is its probabilistic complement:

$$Q(t) \equiv prob\{\text{ one or more failures in interval [0,t]}\} = 1 - R(t) \tag{3.8}$$

Another way to think of the reliability function is as the complimentary cumulative distribution function (CDF) for the random variable that gives the time between failures. That is:

$$R(t) = 1 - \int_0^t f(t)dt = \int_t^\infty f(t)dt\ ; \qquad Q(t) = 1 - R(t) = \int_0^t f(t)dt \tag{3.9}$$

where *f(t)* is the *failure density function* which is the probability density function of time to failure from a known-good starting point.

Also useful is the "age-specific failure rate" otherwise known as the "hazard rate", λ, in reliability. Given a population of systems or components that may fail, the hazard rate is the rate

of failure per member of the group *given that the member has already survived this long*. This itself is may be a function of time, $\lambda(t)$. An example is the classical "bathtub curve" of infant mortality, useful life, and wear-out phases for devices, which reflects age-specific failure rates). Much useful work is, however, based on the assumption of a constant hazard rate λ which reflects systems during their useful life phase. The term "failure rate" or "hazard rate" are then equivalent and both terms are often used for λ. But more generally λ is the *age-specific failure rate per unit*, i.e.;

$$\lambda(t) = \lim_{\Delta t \to 0}\left[\frac{\text{number of failures in } [t \pm \Delta t/2]}{\text{number of systems exposed to failure} \cdot \Delta t}\right] \tag{3.10}$$

where Δt is a unit of elapsed time. Thus the hazard rate is strictly only the same as the failure rate if there is no age-dependency. When one is considering a single unit or system it follows that the hazard rate is the derivative of $Q(t)$ (which is $f(t)$) because as soon as there is one failure, there is no remaining pool from which to generate more failures). If there are a group of items being observed, however, we have to reflect the conditional probability nature of the fact that for a failure to arise in time $t \pm \Delta t/2$ the sample of the elements being considered only contains those remaining units that *already have survived* until time *t;* the probability of which is by definition $R(t)$. Therefore, the hazard rate (or *age specific failure rate for per unit*) is in general:

$$\lambda(t) = \frac{\mathrm{d}}{\mathrm{d}t}[Q(t)]/R(t) = -\left(\frac{1}{R(t)}\right)\frac{\mathrm{d}}{\mathrm{d}t}R(t)\,, \tag{3.11}$$

which is a simple differential equation from which it follows that:

$$\log R(t) = -\int_0^t \lambda(u)\cdot du \tag{3.12}$$

$$\Rightarrow R(t) = e^{-\int_0^t \lambda(u)\cdot du}$$

and this applies for *any* hazard rate function $\lambda(t)$. Also, because $R(t)$ is the probability of a unit surviving to time t (i.e., not failing in [0,t]) then over a population of items or a succession of trials where one item is repeatedly repaired and allowed to run again to failure, it is meaningful to think about the expected time between failures or mean time to failure (MTTF).[6] This will be the expected value of the failure density function:

$$MTTF = E(f(t)) = \int_o^\infty t\cdot\left(-\frac{\mathrm{d}}{\mathrm{d}t}R(t)\right)\cdot dt\,. \tag{3.13}$$

Much practical analysis of network or equipment reliability assumes a constant failure rate for equipment items in service. This is not necessarily accurate but it is an accepted practice to

6. Note that MTTF is not exactly the same as the more often used MTBF of a repairable system. Although usually very close numerically, MTBF is strictly MTTF + MTTR because the time between failures includes the time of repair following the last failure, whereas MTTF is the time to the next failure, following completion of the repair.

characterize failures in service paths arising from a large number of possible independent failures over a large pool of operating equipment in service and independent external events each with individually low probabilities per unit time. Early life stress testing of critical components such as lasers helps eliminate the "infant mortality" portion of the non-constant hazard rate, improving the validity of the assumption somewhat. In addition, if external hazard mechanisms such as cable dig-ups are assumed to be unsynchronized with the equipment deployment, the overall hazard rate from cable cuts can reasonably be modeled as constant on average. A practical justification is also that while mathematical methods do exist to take the non-constant hazard rate curves into effect for each piece of equipment, doing so in real network calculations would imply tracking of the exact type, installation date, and every maintenance date in the life of each individual piece of equipment in each specific network path. Finally, there is recognition that what a planner is often doing with reliability or availability methods in the first place is making *comparative* assessments of alternate networking strategies or broad technology assessment studies of adopting new equipment or operating policies. In these contexts it is seldom the absolute numbers that matter, but the relative ranking of alternatives and these are unaffected by idealization of a constant failure rate. Thus, we have a special practical interest in the case where $\lambda(t) = \lambda_0$ (a constant), for which we get the special results from above that:

$$R(t) = e^{-\lambda_0 \cdot t} \tag{3.14}$$

$$MTTF = 1/\lambda_0 \tag{3.15}$$

$$\text{and } P(\text{exactly k failures in } [0,t]) = \frac{(\lambda_0 \cdot t)^k}{k!} \cdot e^{-\lambda_0 \cdot t}. \tag{3.16}$$

The last result is otherwise recognized as the Poisson distribution.

The relationships between reliability and failure density functions, and its complement, the cumulative failure distribution function are illustrated in Figure 3-16. The dashed arrows linking function values on $Q(t)$ and $R(t)$ to areas under $f(t)$ show the integral relationships involved. In effect the fundamental function is the failure density $f(t)$. The cumulative failure distribution $Q(t)$ is its integral and the reliability is just the probabilistic complement of $Q(t)$.

3.12 Availability

In this section we review the basic concept of system availability. In the design of the book, the overall material on availability is split between this section, which gives a generic introductory treatment, and Chapter 8, where the topic of availability is developed further in the specific context of determining the availability of paths through mesh-restorable networks.

To appreciate how availability differs from reliability, notice the "mission-oriented" nature of the definition above. Reliability is concerned with how likely it is for the system to operate for a certain time *without* a service-affecting failure occurring. There is no presumption of external repair or maintenance to recover from failures. A pump in the shuttle booster engine

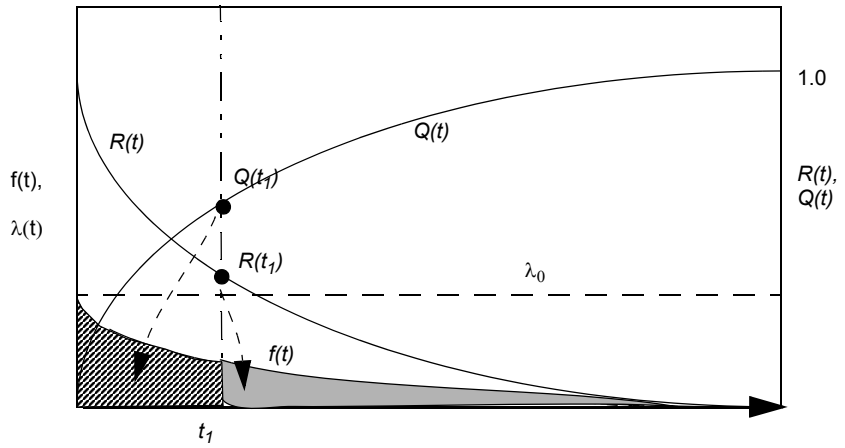


Figure 3-16 Reliability *R(t)*, cumulative failure *Q(t)*, and failure density curve *f(t)* relationships for a constant hazard rate, λ_0.

must run perfectly for three minutes during launch; there is no chance of repair, so the most relevant measure is reliability: what is the probability of operating flawlessly for three minutes?

This is a different orientation than required to characterize continuously operating systems which are subject to repair when failures occur. An Internet user does not care, for instance, about when the ISP had its first server or router failure, nor would he even directly be concerned with how often such failures occur. If such failures are promptly repaired the user will find the probability of being able to get access at any time suitably high enough to be satisfactory. In other words, *availability* can be high, even if failures are frequent. Availability is concerned with the steady state *probability of finding the system in an operating state at any time we want its service*. We are not concerned with whether its history of operation has been flawless to that point or not.

> Availability is the probability of the system being found in the operating state at some time *t* in the future given that the system started in the operating state at time *t=0.* Failures and down states occur but maintenance or repair actions always return the system to an operating state. [BiAl92]

Note that finding a system in the up state at time t_1 is quite different from requiring or expecting that it has stayed continuously in the operating state from $t = 0$ to t_1. When considering availability of repairable systems, a statistical equilibrium is reached between the failure arrivals process and the repair process, both characterized by respective rates, and resulting in a fraction of total time that is "up" time. The fraction of all time that is up time is the system availability, more particularly the steady-state availability. In general a system is biased toward being in the up state for a short time after having a known start in that state. With time, however, failures arise, are repaired, and the system reaches its steady-state equilibrium. Figure 3-17 illus-

trates these relationships for the case of a single repairable component with a constant failure rate and repair rate. It is not the case that reliability is undefined for a repairable system. It con-

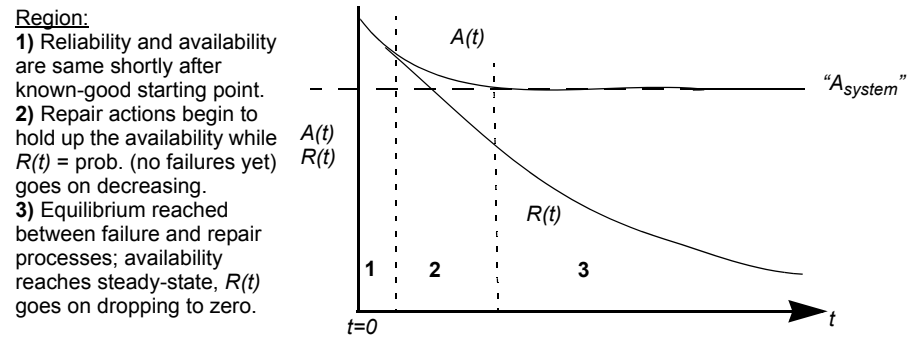


Figure 3-17 Relationship between reliability and steady-state and time-dependent availability for a single (non-redundant) repairable component or system.

tinues to be the probability that the system operates failure-free for the interval $[0,t]$. Whether the system is repaired or not at the point of first failure only makes a difference to the availability. Without repair, however, reliability and availability are identical and both trend monotonically to zero with time. By far in practice it is the steady-state availability that is of interest. Nonetheless, touching on the concept of time-dependent and steady-state availability helps clarify the nature of the phenomenon and also its relationship to the reliability of the same system.

Fundamentally, the steady-state (as opposed to time-dependent) availability is:

$$A \equiv \lim_{T_{obs} \to \infty} \left\{ \frac{\text{Uptime}}{T_{obs}} \right\} \tag{3.17}$$

where T_{obs} is a total observation time. Henceforth, we refer to this simply as the system availabilty. It is the proportion of time that a system is providing its intended service or function observed in the limit as time increases toward infinity.

The most widely known equation for this limit is based on the intuitive picture of the life of a repairable system as a succession of cycles around a loop of operating-failure-repair-operating states. If we assume that each repair episode restores the system to its fully nominal operating condition, the time t in the reliability function is effectively reset to $t=0$. Consequently the expected time to the next failure is the MTTF (Equation 3.15). By definition at $t=MTTF$ following each repair, another failure occurs on average. This is followed by another repair time whose average duration we denote MTTR. Thus the long-term life of a repairable system is comprised of repeated cycles as shown in Figure 3-18. The actual times to failure and actual repair times for each individual cycle are random variables, but for the limiting case we need to know only their averages. In other words, the failure-repair-operating cycle is not regular and repeatably timed as the diagram might seem to suggest. Rather, this is the conceptual limiting average fail-

ure cycle. Once this mental image of the "failure cycle" is obtained, it is easy to remember or

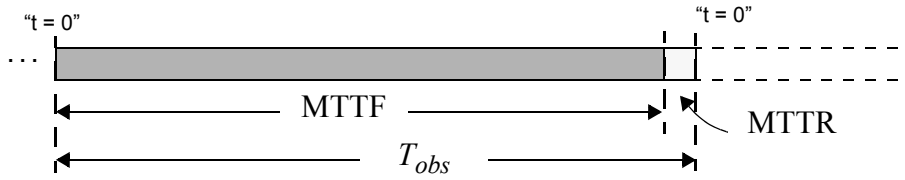


Figure 3-18 Time-line illustration of the failure cycle of a repairable maintained system.

derive the most widely used expression for availability whenever it is needed. This is:

$$A = \frac{\text{uptime}}{T_{obs}} = \frac{MTTF}{MTTF + MTTR} \tag{3.18}$$

Often the term MTBF ("mean time *between* failures") appears in this expression instead of MTTF. As mentioned in the footnote of page 155, it rarely makes a significant numerical difference, but conceptually MTTF is the correct form. If repair times are a significant fraction of average operating times, this distinction can become important, however.

Note that Equation 3.18 applies on the means of the failure density and repair time functions regardless of their distributions as long as the system is statistically stationary (expectations and other moments are not time-varying). Under these conditions Equation 3.18 is also equivalent to

$$A = \frac{\mu}{\lambda + \mu} \tag{3.19}$$

where $\mu = 1/MTTR$ is the "repair rate" and $\lambda = 1/MTTF$ is the "failure rate."

3.12.1 Concept of "Unavailability"

The probabilistic complement of availability A is the *unavailability* U,

$$U \equiv 1 - A\,. \tag{3.20}$$

In a much availability analysis for communication networks, we work with *unavailability* quantities or expressions because of some simplifying numerical assumptions which we will now examine. These assumptions often make the analysis or modeling of complex systems feasible, with acceptable numerical accuracy, in cases where the exact availability model would be intractably complex. The most enabling simplification is the concept of *adding unavailabilities instead of multiplying availabilities* for elements in series. As long as we are considering subsystems or components that each do have relatively high absolute availability, then from Equation 3.18 and Equation 3.20 it follows that

$$U = \frac{MTTR}{MTTF + MTTR} = \frac{\lambda}{\lambda + \mu} \tag{3.21}$$

from which, for the many practical cases of interest in which $MTTF \gg MTTR$ (e.g., years versus

hours is typical), we can write

$$U \approx \frac{MTTR}{MTTF} = \lambda \cdot MTTR . \tag{3.22}$$

In other words, unavailability is approximated as simply the repair time times the frequency of failure, or the failure rate expressed in the appropriate inverse-time units.

FITS: The FIT is an internationally used unit for measuring or specifying failure rates. Because individual components or subsystems are generally highly reliable in their own right, the convention has arisen of using a period of 10^9 hours as a time unit or time scale on which to quantify failure rates (or conversely MTTFs):

$$\text{a failure rate of 1 failure in } 10^9 \text{ hours} = 1 \text{ “FIT”} \tag{3.23}$$

Thus,

$$\text{MTTF} = 10^9 / \text{ FITs} \tag{3.24}$$

gives the MTTF in hours, if the FIT rate is known, and

$$\text{U} = (\text{MTTR} * \text{FITs})/10^9 \tag{3.25}$$

is the unavailability if MTTR is given and the failure rate is given in FITs. The following examples give a feel for some typical failure rates, MTTRs, and common “constants” involved in typical communication network unavailability analyses[7]:

- 1 year = 8766 hours
- 1 failure/year = 114,155 FITs
- 1 FIT = 1 failure in 114,155 years
- Typical FITs for a logic circuit pack of medium complexity = 1500, i.e., MTTF = 76 years
- FITs for an optical Tx circuit pack = 10,867, => MTTF = 10.5 years
- FITs for an optical receiver circuit pack = 4311
- “Three nines availability” $U = 10^{-4}$ -> A= 0.99900 => 8.76 hours per year outage
- “Five nines availability” $U = 10^{-6}$ -> A=0.99999 => 5.26 minutes per year outage
- Typical cable cutting (failure) rate = 4.39/year/1000 sheath miles => 5000 FITs/ mi
- Typical cable physical repair MTTR = 14.4 hours
- Typical plug-replacement equipment MTTR = 2 hours

Note that while failure rates have “units” of FITs, *A* and *U* are inherently dimensionless as both are just time fractions or probabilities of finding the system up or down, respectively.

7. References [ToNe94], [ArPa00], [MaCo03], [Free02], [Free96b] provide additional failure rate and availability data. Selected numbers given here are from [ToNe94].

3.12.2 Series Unavailability Relationships

If a system is comprised of *n* components (or subsystems) in a "series" availability relationship then (like the proverbial Christmas tree lights) *all* components must be operating for the system to be available. Figure 0-1 shows the availability block diagram for elements in a pure series relationship. For elements in series the overall reliability function becomes:

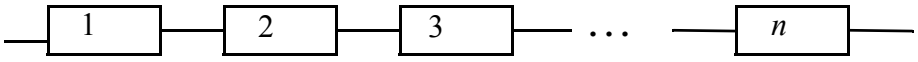


Figure 0-1 Elements in a series availability relationship.

$$R_s(t) = \prod_{i=1}^{n} R_i(t) \tag{3.26}$$

and the exact form for unavailability and availability are

$$A_s = \prod_{i=1}^{n} A_i \; ; \quad Us = 1 - \prod_{i=1}^{n} A_i = 1 - \prod_{i=1}^{n} (1 - U_i) \, . \tag{3.27}$$

As the term "series" implies, the reliability reflects the product of probabilities that any one might fail by time *t*, and the availability of the whole requires that every series element must also be available.

Adding Series Unavailabilities Let us now show why, as mentioned above, one can numerically approximate Equation 3.27 by a corresponding sum of *un*availabilibity values. Specifically the assertion is that:

$$\prod_{i=1}^{n} A_i \approx 1 - \sum_{i=1}^{n} U_i \tag{3.28}$$

where A_i is the availability of the i^{th} set of *N* elements in a series relationship, and U_i is the corresponding unavailability, $1\text{-}A_i$. A simple example with two elements in series will show the basis of this useful numerical approximation. Consider two identical elements *A* and *B* in a series availability relationship, each with elemental unavailability *U*. If we assume A and B fail independently, the exact result would be $A = P(AB) = P(A)P(B)$. Or, therefore $A = (1\text{-}U)\ (1\text{-}U) = 1\text{-}\ 2U + U^2$. In contrast, Equation 3.28 would give us $A = 1\text{-}2U$. Thus we see that in "adding unavailabilities" of two elements in series we are only ignoring the square of an already small number, U^2. Moreover, the error in the approximation is toward being conservative (in an engineering sense) because to the extent we err numerically, it is in the direction of underestimating the actual availability. The typically high accuracy of this approximation is illustrated in [Free96b] with an example having six elements in series with U_i from 10^{-5} to 10^{-3}. The accuracy of the approximation is better than 0.5% which Freeman notes is also "typically far more precise than the estimates of element A_i's".

3.12.3 Parallel Unavailability Relationships

To say that *n* elements or subsystems are arranged in a parallel availability relationship means that only one of *n* has to be working for the system to be available. As long as one element is working, the system as a whole is providing its intended service or function. Figure 3-19 shows the availability block diagram for elements in a pure parallel relationship.

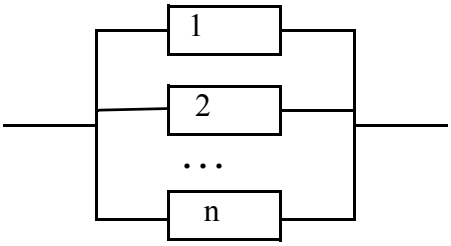


Figure 3-19 Elements in a parallel availability relationship.

For elements in parallel the overall reliability function becomes

$$R_s(t) = 1 - \prod_{i=1}^{n} (1 - R_i(t)), \tag{3.29}$$

and the exact form for the unavailability is

$$U_s = \prod_{i=1}^{n} U_i. \tag{3.30}$$

In summary, two useful basics for simplified availability analysis using the *un*availability orientation are:

1. *Unavailabilities add for elements in series*. This is an approximation (but numerically conservative) and quite accurate for $U_i << 1$.
2. *Unavailabilities multiply for elements in parallel.* This is exact.

3.12.4 Series-Parallel Reductions

The first step in more detailed evaluation of system availability is often to apply repeated series-parallel reductions to the availability block diagram of the system. This involves repeated application of the two main rules: unavailabilities add in series, unavailabilities multiply in parallel. For relatively simple problems, a suitable series of series-parallel reduction steps can completely solve the problem of computing system availability. Figure 3-20 shows an example of this type. As a convenient shorthand in Figure 3-20 we denote element unavailabilities simply by the element numbers and "A + B" means the addition of the unavailabilities of blocks A and B. Similarly the notation "A || B" means the unavailability of element A in parallel with element B, i.e., the product of their unavailabilities. In the example, three stages of reduction lead

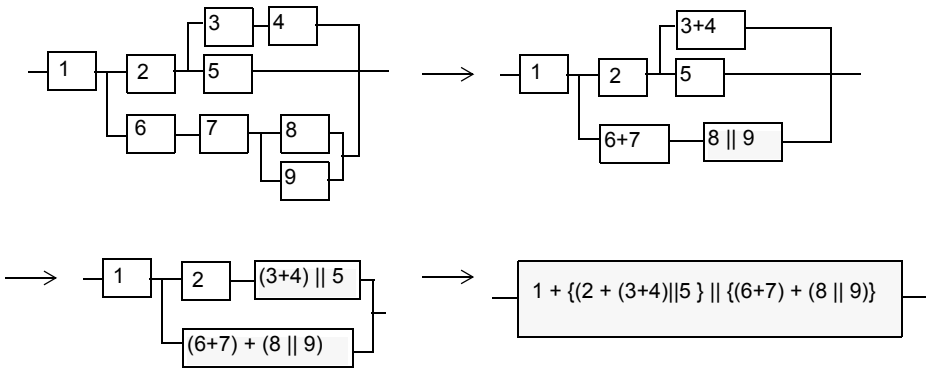


Figure 3-20 Example of series - parallel reductions.

directly to a simple algebraic expression for the system unavailability as a function of the elemental unavailabilities. The problem is simple because there are no cross-coupling paths in the availability model. The approach to calculation of availability for more complex system availability block diagrams is facilitated by first introducing the topic of network reliability.

3.13 Network Reliability

The field of "network reliability" is concerned with questions of retaining graph connectivity in networks where edges have a non-zero probability of being in a failed state. The central issue is simple-sounding but in fact it is quite difficult to exactly compute the probability that a graph remains connected as a whole, or if a path remains between specific nodes or sets of nodes, in a graph with unreliable edges. Specific measures that are studied in this field are questions of *"{s,t}"* or *"two-terminal" reliability*, *k-terminal reliability*, and *all-terminal reliability.* These are all various measures of the purely topology-dependent probability of graph disconnection between pairs of nodes points. Rai and Agrawal [RaAg90] provide a complete survey of this field. Here we try only to extract those basic ideas of network reliability that form part of a grounding for work on transport network survivability and feed into the problem of availability block diagram reduction.

Figure 3-21 illustrates the basic orientation for the network reliability problem. Four equally likely states are drawn for an assumed $p_{link}=0.32$ (i.e., out of 28 links present we expect 9 of them down at any one time on average). A solid line is a working link, dashed is a failed link. If we pick nodes 0-11 we see that in (a)-(c) despite the failures there is always still a route between them. Inspection shows in fact that none of the randomly generated states (a)-(c) contributes any two-terminal unreliability: there is still at least one topologically feasible route between all node pairs. Equivalently, we can say that none of these failure combinations has disconnected the graph. Case (d), however, is an equally likely state but has a dramatically different effect. Four of the nine failure links form a cut of the graph across edges (14-19), (14-9), (6-13) and (13-5). The two-terminal reliability of all node pairs separated by the cut are thus affected

by this failure state. This not only illustrates how abrupt and discontinuous network behavior is in general but it also conveys why numerical enumeration of all link state combinations, followed by tests for graph connectivity, is not feasible for this type of problem on a large network.[8]

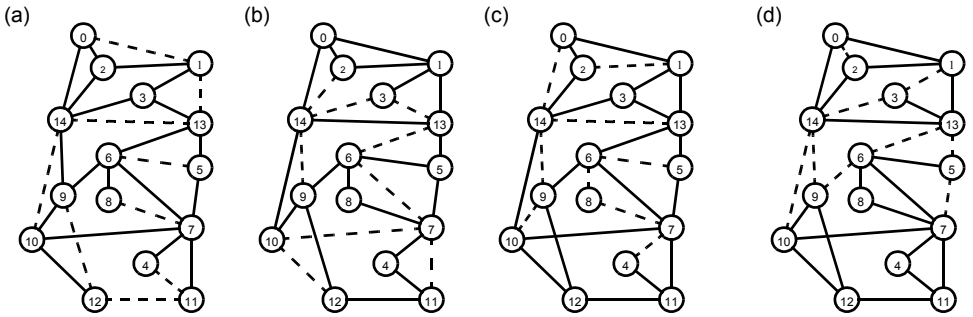


Figure 3-21 Network reliability: How likely is it that at least one route exists between nodes? In this example there are 2^{28} link-state combinations to consider.

Of course in a real network, there may also be outage due to finite capacity effects in Figure 3-21 (a) though (c) but this is not in the scope of the basic "network reliability" problem. Basic network reliability (in the sense of [Colb87],[HaKr95],[Shei91]) presumes that there are no routing or capacity constraints on the network graph. If at least one route exists topologically between *{s,t}*, then it is assumed the signal (or packet train) will discover and use it. With this limitation understood, however, its methods and concepts can provide tools for use in other means for more encompassing considerations availability analysis. The problem of most relevance to the availability of a service path through a network is that of *two-terminal* reliability.

3.13.1 Two-Terminal Network Reliability

Two-terminal reliability[9] is the complement to the likelihood that every distinct path between *{s,t}* contains at least one failed (or blocked) link. Exact computation of the two-terminal reliability problem is NP-complete for general graphs even when the link failure probabilities are known. The computational complexity of trying to enumerate all networks states and inspect them for connectivity between nodes *{s,t}* has led to the approach of more computationally efficient bounds. A widely known general form is called the *reliability polynomial:*

8. Note also that with $p_{link}=0.32$, 9 (out of 28) is the only most likely number of failures at any time, but that all states with fewer or greater number of failed links at the same time still occur with lesser but finite probability, and may also contribute to graph disconnection. The complete calculation of network reliability has to consider all such combinations, not just states with exactly the expected number of failures, as used for the example.

9. In addition to the treatment given here, interested readers are referred to the excellent tutorial paper on algorithms for terminal pair availability calculation by A. Iselt [Isel00].

$$R(G, \{s, t\}, p) = \sum_{i=0}^{m} N_i(G, s, t) p^i (1-p)^{m-i} \tag{3.31}$$

where $G = (V, E)$ is the network graph, $m = |E|$ is the number of edges in the graph, $\{s,t\}$ is a specific terminal pair and p is the link *operating* probability.

This form prescribes either exact or approximate (bounding) estimates of $R(\cdot)$, depending on how $N_i(\cdot)$ is obtained. In its exact form $N_i(\cdot)$ is the number of subgraphs of G in which there are exactly $(m-i)$ failed links but the remaining graph contains a route between nodes $\{s,t\}$. Of course this just defers the problem of calculation $R(\cdot)$ to that of counting or estimating $N_i(\cdot)$. Two simple bounds are conceptually evident at this stage. One is to enumerate (for each $i \in 1 \ldots m$) only those $m-i$ failure link combinations that constitute cuts of the graph between $\{s,t\}$. A cut-finding program can thus enumerate a large number of cuts and their associated weights (in terms of number of edges) for insertion into Equation 3.31. Obviously for $p \approx 1$ the smallest cuts are the most likely and hence numerically dominant contributors to $R(\cdot)$. Assuming not all of the highest order cuts are enumerated[10] the result will be an upper (i.e., optimistic) bound on the exact $R(\cdot)$. i.e.,

$$R(G, \{s, t\}, p) \le 1 - \sum_{i=c}^{m} C_i(G, s, t) p^{m-i} (1-p)^i \tag{3.32}$$

where c is the minimum cut of the graph between $\{s,t\}$ and $C_i(\cdot)$ is the number of $\{s,t\}$ cutsets found comprising exactly i edges. The exact reliability will be lower than this because network states involving i failures but containing a cutset of fewer than i edges are connectivity-failure states that are not counted.

A converse viewpoint for assessing $N_i(\cdot)$ is from the standpoint of network states that contain at least one working route among the set of all distinct routes between $\{s,t\}$. (The two are conceptually the same as the notion of "cuts and ties" in more advanced analysis of system availability block diagrams.) Here, all of the k-successively longer distinct (non-looping) routes on the graph between $\{s,t\}$ are generated and each recorded with its associated length (number of edges in series en route). Then a simple upper (i.e., optimistic) bound on $\{s,t\}$ reliability is:

$$R(G, \{s, t\}, p) \le 1 - \prod_{i=1}^{\kappa} (1 - p^{L_i}) \tag{3.33}$$

where L_i is the length of the k^{th} distinct route between $\{s,t\}$. Figure 3-22(a) portrays the basic notion of $\{s,t\}$ reliability being viewed in Equation 3.33 as the probability that *not every* possible route is blocked and implicitly treats routes as independent entities. In contrast, Figure 3-22(b) shows how several distinct routes may actually share single link failures in common, illustrating

10. Note that enumerating all cutsets is no less demanding computationally than the original prospect of generating all link-state combinations, i.e., $O(2^{|E|})$ but each cutset is a direct prescription for connectivity loss so the goodness of the bound and computation time can often benefit from using direct knowledge of most likely failure combinations and can be tailored by systematically increasing the size of the cutsets considered.

why Equation 3.33 is an optimistic bound.

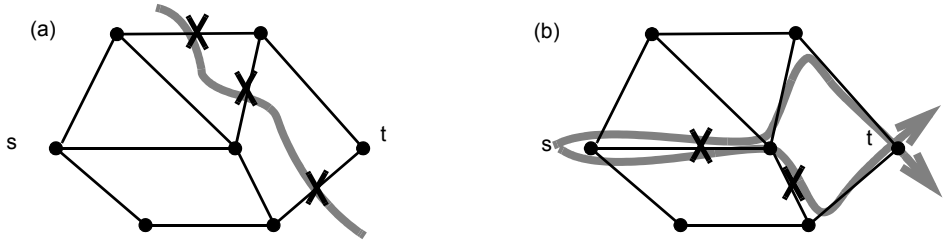


Figure 3-22 Orientations to the network reliability calculation: (a) failures that together create an *(s,t)* cut set, (b) failures that defeat all routes between *(s,t)*.

More precisely the route-based formulation is dependent on union of the probabilities that all edges in route *i* are operating, i.e.,

$$R(G, \{s, t\}, p) = \bigcup_{i=1}^{k} (1 - p^{L_i}) \tag{3.34}$$

which calls for application of the inclusion-exclusion principle for the union of non-disjoint sets [GaTa92] (p.90). Denoting $P(i) = (1 - p^{L_i})$ as the probability that all links in the i^{th} route are operating,

$$R(\ ) = \sum_{i \in 1 \ldots k} P(i) - \sum_{i<j,\, i,j \in 1 \ldots k} P(i)P(j) + \sum_{i<j<n,\, i,j,n \in 1 \ldots k} P(i)P(j)P(n) - \ldots + (-1)^{k+1} \prod_{i \in 1 \ldots k} P(i) \tag{3.35}$$

In [Shei91] the application of the inclusion-exclusion principle for probability union is treated further, showing that there are always certain cancellation effects between terms of the inclusion-exclusion series that give further insights (the concept of irrelevant edges) and that can be exploited to simplify the expansion process.

3.13.2 Factoring the Graph and Conditional Decomposition

Let us now return to the problem of calculating system availability in cases where basic series and parallel relationships do not completely reduce the model. This is where the link to network reliability arises. If a network is completely reducible between nodes *{s,t}* by repeated application of simple reductions into a single equivalent link, the network is said to be *two-terminal series-parallel*. In such a case the resultant single reduced edge probability is *R(G, {s,t},p)*. But many realistic cases are not two-terminal series-parallel in nature because of some edge that cross-couples the remaining relationships in a way that halts further application of the series-parallel reductions. In the approach that follows, which is also based in network reliability, such an edge is used as a kind of pivot point on which the problem is split into two condi-

tional probability sub-versions that apply when the particular edge is in one case assumed available and in the other case where it is assumed to be down.

Figure 3-23 summarizes the basic series-parallel reduction rules in a canonical form on the

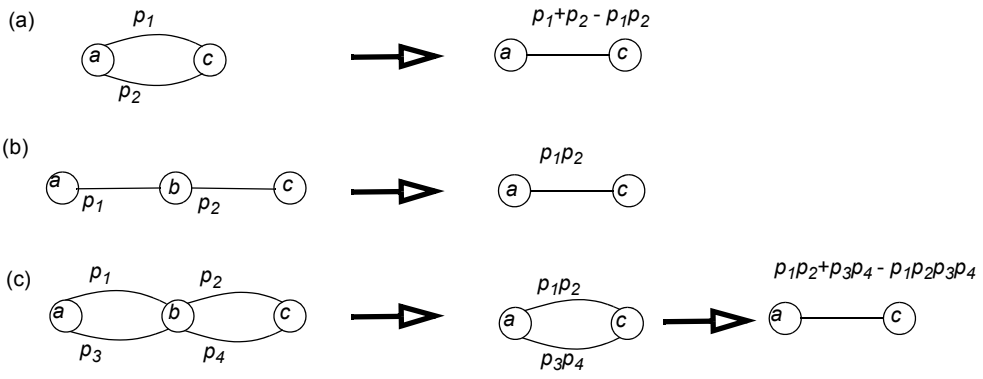


Figure 3-23 Reliability-preserving graph reduction rules: (a) parallel, (b) series, (c) two-neighbor reduction.

edge probabilities (probabilities of the link being up, equivalent to the elemental availability). Cases (a) and (b) are the previous basic parallel and series relationships, to which case (c), called a "two-neighbor reduction," is added. When applied to either a network graph or an availability block diagram, these transformations are exact or, in the language of network reliability, they are "reliability preserving." To use these reductions, element failures must be statistically independent, and in cases (b) and (c) in Figure 3-23 node *b* must have no other arcs incident upon it. Node *b* also cannot be either the source or target. While single arcs are shown the rules apply to any block that is similarly reducible to a single probability expression, so that, for instance, p_1 in Figure 3-23(a) may already be the result of a prior set of series-parallel reductions.

In general the application of series-parallel reduction rules will be exhausted before the original network is completely reduced. This will usually manifest itself through some edge that cross-couples between remaining subgraphs, i.e., one or more nodes will be like node *b* in Figure 3-23(b) but with the presence of more than just two arcs, so that another application of a series reduction is not possible. At this stage the graph can be "factored" to continue the reductions. Graph factoring is based on Moscowitz's *pivital decomposition formula* [Shei91] (p.10). The key idea is that:

$$R(G, s, t, p) = p \cdot R(G|e) + (1 - p) \cdot R(G - e) \tag{3.36}$$

where p is the probability the edge is available, $G|e$ means graph "*G given e*", and *{G-e}* is graph *G* without edge *e*. Thus the whole is considered as the conditional probability decomposition of the two states that the confounding edge *e* may be in, with probability *p* and *(1-p)* respectively. $G|e$ is represented by graph *G* where edge *e* is contracted or "short circuited." The

probability-weighted sum of the two conditional probability decomposition terms is the two-terminal graph reliability. In practice the idea is to recognize a key edge *e* that will decouple the two resulting conditional subgraphs in a way that allows another round of series-parallel reductions. A complete graph may thus be decomposed through a series of series-parallel reductions, splitting to two conditional subgraphs, series parallel reductions on each, splitting again in those as needed, and so on. The real computational advantage of the decomposition steps is to overcome the situations where no further series-parallel reductions are possible. Were it not for this use of decomposition to link between subproblems that are further series-parallel reducible, it would be of little practical value because by itself it is equivalent to state-space enumeration by building a binary tree of all two-state edge combinations. More detailed treatments can be found in [Colb87] (p.77) and [HaKr95].

To illustrate the application to availability problems, however, consider the availability block diagram in Figure 3-24(a). Because of the "diagonal" element, it is not amenable to series-parallel reduction. We do, however, obtain two subgraphs that are each easily analyzed if we presuppose the state of the diagonal element. In (b) we presume it is failed. In (c) we presume it to be in a working state. Thus the resulting subgraphs are conditional probability estimates of the system availability. To get the overall availability we weight the result for each subgraph by the probability of the decomposed link state that lead to that subgraph. Therefore, for this example:

$$A_{sys} = (1 - A_d) \cdot A_{\text{case(b) subgraph}} + A_d \cdot A_{\text{case(c) subgraph}} \quad . \tag{3.37}$$

$$A_{sys} = (1 - A_d) \cdot [1 - \{(1 - A^2)\}^2] + A_d \cdot [1 - (1 - A)^2]^2$$

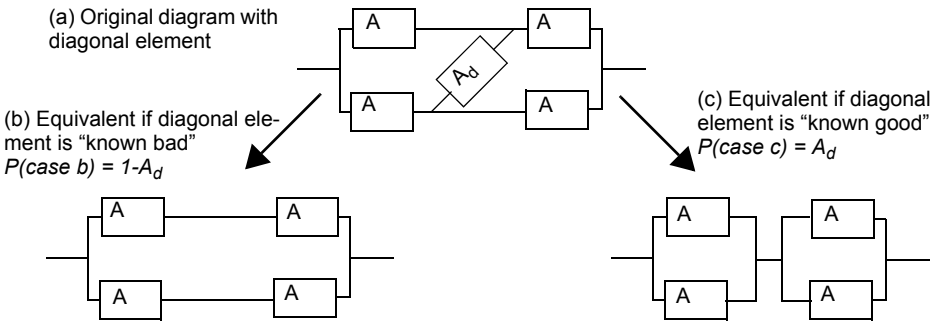


Figure 3-24 Example illustrating conditional decomposition of an availability block diagram.

3.14 Expected Loss of Traffic and of Connectivity

Let us now return to look at the two most commonly used ROF-type measures of survivability. These are the *expected annual loss of traffic (ELT)* [ArPa00] and the *annual expected downtime of connection* (AEDC) [T1A193]. An appreciation of these measures is aided by the background above on availability and network reliability. ELT is like a traffic-weighted path availability and AEDC is a certain type of two-terminal network reliability measure. Both of

these are path-oriented measures in that they pertain to a transport signal path between a stipulated pair of nodes. The path orientation follows the philosophy of using hypothetical reference paths (HRP) for various performance assessments. This type of measure allows comparisons between network designs and/or various restoration techniques to be made on the basis of one or a few path models of specific relevance and that are agreed by stakeholders to be representative of either worst or average cases. In principle the corresponding path measures can be computed for all possible end-node pairs to obtain statistics of all individual paths in the network. A path in this context is constituted by a demand unit between nodes *i,j* that is a contiguous transmission signal.

3.14.1 Expected Loss of Traffic (ELT)

For a given pair of nodes exchanging demand over possibly several paths through a network, ELT asks what the expected number of lost demand-minutes will be over a year. The total demand between a node pair need not follow a single route. The total demand between *i* and *j,* denoted *dij* may be realized by routing over a set of diverse routes *P* as long as $\sum_{p \in P} d^{p}_{i,j} = d_{i,j}$. One way of implementing the ELT calculation is then:

$$ELT_{i,j} = \sum_{p = 1 \ldots P} \left[d^{p}_{ij} \cdot \sum_{\forall k \in (S,N)} \delta^{p}_{i,j}(k) \cdot U_k \right] \cdot M_0 . \qquad (3.38)$$

where d^{p}_{ij} is the amount of *(i,j)* demand assigned to the p^{th} route and $\delta^{p}_{i,j}(k) = 1$ if span *k* is in the p^{th} route for demand pair *(i,j)*, and zero otherwise. The constant $M_o = 5.26x10^5$ gives ELT units of demand-minutes/year of "traffic" loss. Thus ELT is the sum of the demand-weighted unavailability of each distinct (not disjoint) path employed to bear the total demand between nodes *i,j*. In the inner sum every node and span in the network is indexed by *k* and the routing function $\delta^{p}_{i,j}(k)$ answers the 1 / 0 question: is network element *k* a constituent of the p^{th} diverse route used for demands between *i* and *j?* If so, the unavailability of that element, *Uk,* contributes to the ELT. The benefit of ELT beyond a single path availability analysis is that it reflects the size and number of demand units affected, allowing apples-versus-apples comparison across alternatives involving different signal levels, diversity routing and/or restoration techniques. Note that Equation 3.38 has some of the same numerical approximations and assumptions as mentioned above for availability analysis. In particular it strictly overestimates the total by virtue of the addition of unavailabilities for each distinct path as if they were independent.

Also, as written, the ELT formula is most applicable to "passive" point-to-point transmission networks where the network elements in each path directly contribute their true *Uk* values to the total (although the *Uk* for a transmission span may include the built-in benefit of co-routed APS against internal failures). In a network that embodies active restoration mechanisms and designed-in spare capacity, however, the *Uk* values should be those already reflecting any net benefit in equivalent unavailability terms of the designed-in survivability methods. For instance in a span-restorable mesh network, the *Uk* values for spans could be the equivalent unavailability of spans defined in Chapter 8. Alternately the native path availabilities could be presented

but a simple extension to Equation 3.38 used so that unavailability is not contributed to the sum if *either* a path or its known backup are available. This implicitly recognizes the switch-over from working to protection that happens to avoid "lost traffic" in a survivable network with active restoration or protection. Ultimately, however, if the details of modeling the effects of active restoration measures into the framework of Equation 3.38 become unmanageable, Equation 3.38 still guides how one would approach the calculation of ELT by simulation.

3.14.2 Annual Expected Downtime of Connection (AEDC)

AEDC is a Random Occurrence of Failure (ROF) measures of survivability defined in [T1A193]. AEDC is more like a network reliability measure. It asks how often during a year would one expect *total disconnection* of all the paths for communication between nodes i and j. [T1A193] states:

> "Connection between two nodes is lost if, for all paths, there exists no working or protection channels able to carry the demand. The consequences of losing connection between two nodes can be more serious than the consequences of losing an equivalent amount of traffic throughout the network without losing connectivity. Loss of connectivity can lead to the loss of important emergency and high priority traffic or create a situation of isolation."

Thus AEDC specifically relates to the two-terminal reliability of a network. The main difference is that classical network reliability addresses whether there exists *any* topologically possible route between nodes, but the intent with AEDC is to consider details or limitations of the actual restoration mechanism and network capacity constraints that would be involved in determining how many, individual prefailure paths between i and j could feasibly be restored. The point is that while the graph may remain connected it is possible that specific rerouting mechanisms may be starved of capacity or constrained in routing so that they cannot emulate the routing generality in classical network reliability. It is safe to say, however, that the two-terminal reliability of the network graph (multiplied by the number of minutes in a year) would be a lower bound on the AEDC (in minutes).

A more exact evaluation of the AEDC can also be conducted with the availability analysis methods above. The approach is as follows for a given node pair:

1. Identify all distinct routes between the nodes which would be eligible for use for either the normal working path or a protection or restoration path.
2. Represent the set of distinct routes as a series-parallel availability block diagram.
3. Apply series-parallel availability reductions to the current availability block diagram.
4. When step 3 halts, but the reduction is not complete, select a cross-over element in the block diagram and apply a conditional decomposition.
5. Repeat steps 3-4 until the block diagram is completely reduced.
6. The AEDC is the resulting unavailability value times the number of minutes in a year.

The basic process is illustrated in Figure 3-25 for a small network where the AEDC for nodes A to D is calculated. In the availability block diagram the dashed block names (such as

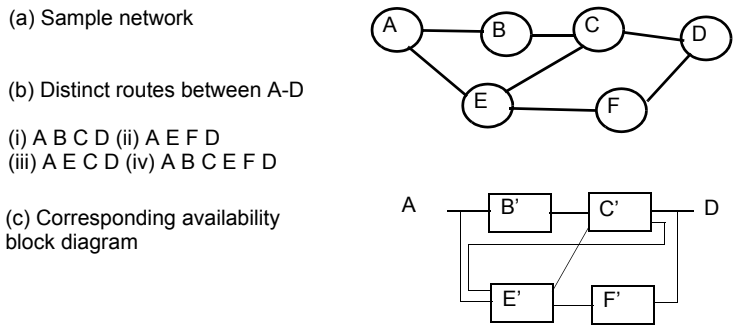


Figure 3-25 Example using availability analysis to estimate AEDC.

node B') represent the unavailability of the network node and the physical span leading up to it. i.e., B' represents node B and link A-B as a single block. A way to develop the availability block diagram (c) from the network diagram (a) is to start from the list all distinct routes. The first route becomes the pure series path model seen at the top (A B C D) in Figure 3-25(c). If the next route listed was fully disjoint from the first then it too is drawn in full, in perfect parallelism to the first route represented. More generally where the routes are not disjoint, one tries to draw the path in parallel but has to obey a rule that if an element has been drawn already it cannot be drawn again. Thus, for the second route listed we drop down from A to represent E and F, then return to up to D. The third route above also adds no new elements, just the vertical link between rows joining the output of E to the input of C. Note this link is in effect unidirectional in that it represents the route option going from E to C but does not imply a corresponding direct linkage between B and F. The last route lays down the link from the joining the output of C back to the input of E to represent the route (ABC)-(EFD).

As the example shows, the primary complication is that the "diverse" paths between A and D are not *disjoint* paths. There is thus a correlation of failures affecting each path. This is true for ELT as well, but ELT is an expected sum of demand weighted outage on all paths, a probability union type of construct which is numerically insensitive to this at typical *Uk* values. In contrast AEDC is a question of probability intersection. That is, all paths down defines the condition for *i,j* "disconnection." In this case any common elements among one or more paths can drastically alter the probability of all paths failing together. Consider for instance a case of $|P|=5$ but all paths sharing on node in common (span disjoint but not node disjoint). Then the AEDC would be almost literally just the *Uk* of the one node common to all paths. The general case of the paths not being fully disjoint does not lend itself to direct analytical expression. Rather, the methods of series-parallel reduction and decomposition can be used to address the question and any specific limitations to the rerouting capability of the restoration method are reflected in the set of distinct routes represented.

If a set of fully disjoint paths between *(i,j)* is identified (or span disjoint paths where node failures are not being considered) then the AEDC can be lower-bounded (i.e., an optimistic bound) as:

$$AEDC_{i,j} = \prod_{p \in P} \left\{ \sum_{\forall k \in S} \delta^{p}{}_{i,j}(k) \cdot U_k \right\} \cdot M_0 \,. \tag{3.39}$$

This is "exact" (i.e., except for the series addition of elemental unavailabilities on each path) for the fraction of time during the year that the set of all mutually disjoint paths between *i* and *j* would all be down simultaneously.