NETWORK RELIABILITY AND FAULT-TOLERANCE

When we make a telephone call, the call is connected through a communication network to the receiving party. Similarly, when we send an e-mail using the Internet, the message is sent through a communication network to the recipient. Such communication networks are made up of nodes and links that connect the nodes by hardware as well as the software components that allow for the functionality to communicate through such networks. Network reliability refers to the reliability of the overall network to provide communication in the event of failure of a component or components in the network. The term *fault-tolerant* is usually used to refer to how reliable a particular component (element) of a network is (e.g., a switch or a router.) The term *fault-tolerant network,* on the other hand, refers to how resilient the network is against the failure of a component.

Communication network reliability depends on the sustainability of both hardware and software. A variety of network failures, lasting from a few seconds to days depending on the failure, is possible. Traditionally, such failures derived primarily from hardware malfunctions that result in downtime (or "outage period") of a network element (a node or a link). Thus, the emphasis was on the element-level network availability and, in turn, the determination of overall network availability. However, other types of major outages have received much attention in recent years. Such incidents include accidental fiber cable cut, natural disasters, and malicious attack (both hardware and software). These major failures need more than what is traditionally addressed through network availability. For one, these types of failures cannot be addressed by congestion control schemes alone because of their drastic impact on the network. Such failures can, for example, drop a significant number of existing network connections; thus, the network is required to have the ability to detect a fault and isolate it, and then either the network must reconnect the affected connections or the user may try to reconnect it (if the network does not have reconnect capability). At the same time, the network may not have enough capacity and capability to handle such a major simultaneous "reconnect" phase. Likewise, because of a software and/or protocol error, the network may appear very congested to the user

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

(1–3). Thus, network reliability nowadays encompasses more than what was traditionally addressed through network availability.

In this article, we will use the term *network reliability* in a broad sense and cover several subtopics. We will start with network availability and performability and then discuss survivable network design, followed by fault detection, isolation, and restoration as well as preplanning. We will conclude with a short discussion on recent issues and literature.

NETWORK AVAILABILITY AND PERFORMABILITY

Network availability refers to some measure of the reliability of a network. Thus, network availability analysis considers the problem of evaluating such a measure. [Note that in current literature, this is often termed as the network reliability analysis (4)]. Moore and Shannon did early work in this area (5). We discuss network availability through an example. Figure 1 shows that two telephones are connected by distribution segments (A) to local switches (S), while the switches are connected by the facility (B). The following allocation of outage/downtime percentage is assumed for the different elements: S 0.01%; A, 0.01%; B, 0.03%. Then, the availability of this connection is $(1 - 0.0001)^4(1 - 0.0003) = 99.93\%$; this translates to the maximum downtime of 368 min per year.

In general, network availability computation addresses the availability of a network in operational states, and discrete probability models are often used in analysis. Let $\mathscr{E}$ denote the set of elements of a network (for examples all the nodes and links). Each element may be in up or down state, where up refers to fully operational and down refers to total loss of the element. Let p_e denote that probability that element $e \in \mathscr{E}$ is up—this is also referred to as the availability of element e. Now consider the subset E_i of $\mathscr{E}$ consisting of the up elements of state i. Then, the probability that the network is in up state E_i is given by

$$Pr(E_i) = \prod_{e \in E_i} p_e \prod_{e \in \mathscr{E} \setminus E_i} (1 - p_e) \tag{1}$$

Note that there are $2^{|\mathscr{E}|}$ possible states (where $|\mathscr{E}|$ denotes the cardinality of the set $\mathscr{E}$); thus, usually network availability computation needs to deal with the problem of this exponential growth in states. A variety of algorithms for efficient computation have been developed over the years for different availability measures; the interested reader is directed to Ref. 4 and the references therein for additional information.

A related issue to availability is the performability. Most availability measures deal only with the connectivity aspect of the network; for example, what is the availability of a path from a source node to a destination node. However, when a failure occurs, the network may not be able to perform at the same level as when there were no failure. For example, the average network blocking in voice telephone networks (circuit-switched networks) is typically the measure for grade-of-service (GoS). A common value of GoS is 1% blocking under the normal operational mode, but under a specific outage, this may increase to more than 10% blocking; similarly, in a packet-switched network, the average packet delay may increase by an order of magnitude during a major failure compared to under the normal circumstances. Thus, the network failure performability addresses the performance of the network under various failure states. Consider a network with m elements that can each be either in operational or in a completely failed state; then, the total number of states is 2^m. The performability measure $\mathscr{P}$ is given by

$$\mathscr{P} = \sum_{k=1}^{2^m} Pr(k)X(k) \tag{2}$$

where $Pr(k)$ is the probability of state k, and $X(k)$ is the measure (e.g., network blocking in circuit-switched networks or average delay in packet-switched networks) in state k. Again, we face the issue of the exponential number of states. This can, however, be bounded by considering most probable t states as was first shown by Li and Silvester (6). Often, with the proper choice of t, the performability measure can be quite accurately computed. For example, if in a network, multiple simultaneous link failure scenarios are extremely unlikely, then the most probable states are the failure of each link independently. Accordingly, one may limit the computation to these states.

SURVIVABLE NETWORK CAPACITY DESIGN

While network availability and performability address important measures for evaluating the reliability of a network, designing the networks for survivability is extremely important for overall network reliability. In this section, we address this topic for the capacity design problem using separate examples for circuit-switched networks and packet-switched networks.

Circuit-Switched Traffic Networks Example

Consider the three-node circuit-switched network (Fig. 2) for which we are given that the availability of each link is 99.9%. We assume that the network has symmetric offered traffic (or load) and capacity. Offered load in circuit-switched networks is given in erlangs; this load is the product of the average call arrival rate and the average call holding time. For example, if the average call arrival rate is 200 calls/h and the average call holding time is 3 min, then the offered load is 10 erlangs ($=3 \times 200/60$).

For the symmetric three-node network, offered load between any pair of nodes is assumed to be 10 erlangs, and the link capacity on each link is given to be 21 trunks (or circuits). We assume that the traffic between each pair of nodes is

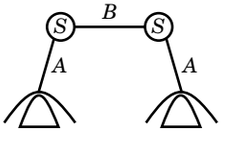

Figure 1. Network view for availability example.

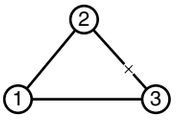

Figure 2. Three-node network.

routed on the direct link that connects the end nodes of the pair, and we would like to know the call-blocking probability. For an offered load of a erlangs, and c trunks, and under the assumption that call arrival follows a Poisson process, Erlang-B loss formula can be used for computing the blocking probability, which is given by

$$E(c, a) = \frac{a^c/c!}{\sum_{k=0}^{c} a^k/k!} \tag{3}$$

Thus, in our example, we have $E(21, 10) = 0.000889 \approx 0.001$. That is, the network is providing a service quality (grade-of-service) of 0.1% blocking. (In actuality, the blocking for each pair of nodes is slightly different because any traffic blocked on the direct link can try the alternate route.)

Now, suppose that the link 2–3 fails; in this case, the network is still connected because node 1 is connected to node 3 via node 2. Assuming that the network still has the same amount of offered load, the load between node 2 and node 3 is now required to be routed through node 1; thus, the load offered to each link is 20 erlangs, whereas the capacity on each link is still 21 trunks. Thus, the blocking seen by traffic on each link is $E(21, 20) = 0.13144$, and the blocking seen by pair 2–3 traffic going through node 1 is even higher. Under the link independence assumption, the blocking on a path consisting of two links is given by $1 - (1 - b)^2$, where b is the link blocking probability. Thus, in our example, the blocking for traffic between 2–3 going through node 1 is $1 - [1 - E(21, 20)]^2 = 0.24558$.

Thus, we can see that, under no failure, the network provides a grade-of-service of 0.1%, whereas under a single link failure, the worst traffic pair blocking is 24.558%, although the network connectivity is still maintained. Recall that the link availability was assumed to be 99.9%; this means that the link can possibly be down for as long as 8 hours in a year. If we assume one event per link per year, then this link could conceivably be down for up to 8 hours straight! In some networks, this may be unacceptable given that the worst traffic pair blocking jumps to 24.558% from 0.01%. If we assume that the network should still provide a 0.1% blocking grade even under a single failure for every traffic pair, then to accommodate for the worst path blocking, we need link blocking on each of the remaining links to be b such that the path blocking for traffic between node 2 and node 3 using links 2–1 and 1–3 needs to satisfy $1 - (1 - b)^2 = 0.001$; this translates to $b = 0.0005$ for each link. Because, now we have an offered load of 20 erlangs on each link, we need to find the smallest c such that $E(c, 20) = 0.0005$. Solving for integral c, we find that c needs to be at least 36 (i.e., we need to have 36 units of capacity on links 1–2 and 1–3 each). By the same argument, if we consider the failure of a different link independently, then the other two links each need 36 trunks. Thus, to cover for failure of each link independently, each link needs 36 trunks to provide the same level of blocking as was originally wanted for the network in the nonfailure mode. In other words, the network needs 80% more capacity to cover for a link failure compared to the no-failure case although network availability requirement was met.

Packet-Switched Networks Example

Consider this time a three-node packet-switched network. We will use Fig. 2 again. In packet networks, the offered traffic is usually given by the average packet arrival rate (packets per second, pps in short). If the average packet arrival rate to a network link is λ and follows a Poisson process, the average packet size is exponentially distributed with mean $1/\hat{\mu}$ kilobits, and the link speed is C kilobits per second (kbit/s), then the average packet delay (caused by the queueing phenomenon) can be obtained from the M/M/1 queueing system and is given by

$$T(\lambda, C, \hat{\mu}) = \frac{1}{\hat{\mu}C - \lambda} \tag{4}$$

For the three-node example, we assume unit mean packet size (i.e., $\hat{\mu} = 1$), in addition to assuming that the average arrival traffic between each pair of nodes is 10 packets per second and that the capacity of each link is 30 kbit/s. If all traffic between each node-pair is routed on the direct link, this provides an average delay of $T(10, 30, 1) = 0.05$ s, or 50 ms. Now suppose that the link 2–3 fails, then the traffic between node 2 and node 3 is routed through node 1; this induces an offered traffic of 20 pps on each remaining link. Thus, the average delay on each link (1–2 and 1–3) is 100 ms which is observed by traffic between nodes 1 and 2 and between nodes 1 and 3. On the other hand, the traffic between nodes 2 and 3 will go over two links and will thus experience a delay of $2 \times 100 = 200$ ms; this delay is four times more than under the no-failure situation.

If the network goal is to provide the average delay for any pair to be less than or equal to 50 ms under a single link failure, then to meet this condition we need link capacity C such that $2\ T(20, C, 1) = 2/(C - 20) = 0.05$ which implies that C needs to be 60 kbit/s on each of the remaining links. Similarly, if we consider the independent failure of a different link, then the other two links will require 60 kbit/s to provide the same level of service. Thus, in this network, we see that we need to double the capacity to provide the same level of service obtained under a single-link failure.

Discussion

We can see from these examples that if the network is *not* provided with additional capacity, then the traffic blocking can be very high in circuit-switched networks, which can result in excessive retry by users, or the packet backlog (queue) can build up in packet-switched networks. Thus, a transient effect can take place. From these two examples for two different networks, we can also see that, in some circumstances, the network capacity needs to be 80% to 100% more to provide the same level of service under a single link failure. This of course depends on the network objective (in our examples, we have used the objective that worst-pair traffic blocking or delay is minimized). In some networks, this near doubling of capacity can be cost-prohibitive; thus, the network performance requirement under failure may be relaxed. For example, under a single-element failure, it may be acceptable to have 5% blocking under a single link failure for the circuit-switched network case, or the average delay is acceptable to be 100 ms for the packet-switched network case. It is easy to see that this will reduce the additional capacity requirements in both cases.

Even though additional capacity can meet GoS requirement under a failure, the actual network topology layout and

routing are also critical for survivable design (7). Thus, we also need to understand the network connectivity requirement for the purpose of survivability. For instance, a network needs to be minimally two-edge connected to address a single-link failure; this means that there must be two links connected to each node so that if one of them fails, a node can still be connected to the rest of the network through the other link; this avoids isolation of a node or a part of a network from the rest of the network. If a network is prone to multiple link failures at a time, this would require the network to have a higher degree of connectivity, which, in turn, would usually mean more network resource requirement to address for such failure situations. Survivable design for different node and edge connectivity level is extensively discussed in Ref. 8; the interested reader is directed to this reference for additional information.

Going back to the three-node examples, recall that the routing choice was limited to taking the *only* two-link path in the event of a failure. In a larger network, usually multiple routes between each origin and destination nodes are available; in the event of a failure, traffic can be sent on any of the unaffected paths. However, the actual flow on each path would depend on the actual routing rule in place as well as the availability of network capacity. Thus, it is not hard to see that the actual capacity requirement to address a failure in the network depends also on the actual routing schemes available in the event of a failure.

In any case, the overall network survivability and reliability depends on a number of issues. Network capacity design for survivability, as we see from these examples, plays an important part. In the next section, we discuss fault detection and isolation as well as network restoration—another key piece in network reliability.

FAULT DETECTION, ISOLATION, AND RESTORATION

Usually, different elements in a network are equipped with alarm generation capability to indicate the occurrence of any abnormal condition, which may cause the reduction or complete loss of the element. This abnormal condition is sometimes labeled as a fault. When an actual failure occurs, depending on the triggers set by various elements in the network, multiple alarms may be generated by a number of network elements—this is the fault-detection phase. Then, the network management system that monitors the network needs to determine the root cause of the fault. Fault isolation is the process of identifying the root cause of the fault. Thus, an issue that first needs to be addressed is correlation of alarms (9) to determine and isolate the actual point of failure in the network. Such fault-detection systems are needed to determine the cause of a fault quickly so that appropriate action can be taken. It is easy to see the relation of fault isolation to network reliability. The longer it takes to detect the cause of a fault, the longer it takes to fix it, and thus, conceivably the network is affected for a longer period of time, which decreases the performability of the network. Rule-based and model-based systems are used for fault isolation. Both centralized and distributed fault localization can be used; see Ref. 10 for a survey of different techniques.

Along with the fault-isolation phase, the restoration/repair phase begins. First, the network may be provided with additional capacity. If the additional capacity is provided so that even after failure the quality of service is met, then from the user's viewpoint, the failure is not perceived! Thus, a way of "restoring" the network is through additional capacity in the network (although, in actuality, the fault is not physically repaired yet). As we have already seen, to address for a single failure, the network may need twice the capacity, which may be sometimes cost prohibitive. Thus, the network may be provided with less than full spare capacity to address for a failure. In such cases, if the network has adaptive routing capability, then some of the traffic can be rerouted around the failure; thus, the users may not perceive the full impact of a failure.

Sometimes, the spare capacity can be provided in a different layer in the network because of cost and technological considerations. In the simplest architectural view of the communication network infrastructure, services such as voice or Internet are provided over *logical* switched or router-based networks; the capacity required for these logical networks is then provided over the physical transmission network, which may be connected by the digital cross-connect systems or SONET (Synchronous Optical Network) rings. For example, if a network is equipped with fast automated digital cross-connect system and/or SONET self-healing ring capability at the transmission network, the network where the services are provided may not perceive any failure because of fast automated restoration (11,12). At the same time, the transmission network level restoration schemes do not address failures such as a line card failure, or a switch or router failure; thus, restoration at the logical network level also needs to be triggered; this may include rerouting and automatic reconnection of affected connections. It is clear from this discussion that to restore from a failure, the network should be equipped with capacity as well as the proper network management system and software components to detect, isolate, and recover from a failure.

Other types of failures such as a software attack or a protocol operation failure cannot be addressed through the restoration process discussed earlier. An example is the SYN attack (2) in transmission control protocol (TCP), which severely affected an Internet service provider (TCP is the transport layer protocol on which services such as email, file transfer and web browsing are provided in the Internet). In this case, the mechanism is needed to identify where such attacks are coming from so as to stop such attacks.

ADVANCED PREPARATION FOR NETWORK RELIABILITY

To provide network reliability, it is also important to do preplanning and/or advanced preparation. Of course, one way is to have additional spare capacity in the network. However, there can be a failure in the network that can actually take away the spare capacity if the network is not designed properly because of dependency between the logical network and the physical network (7). Thus, it is necessary to audit the network and find the vulnerable points in the network and then to equip the network with additional capabilities to avoid such vulnerabilities. For example,

1. The network may be provided with transmission-level diversity so that for any transmission link failure there

is at least another path not on the path of the failure.

2. A redundant architecture at network nodes can be built to address for a node component or nodal failure; this may include dual- or multihoming to provide for multiple access and egress points to and from the core network.

To address for failures due to a software or protocol operations error or a software attack, different types of preparations are necessary. Several software errors that have occurred on various data networks such as the ARPANET, Internet, and SS7 Network (the data network that carries the signaling information for the public telephone network) (1,2) have caused such severe congestion in the network that it cannot be adequately addressed by normal congestion control schemes. Although enormous efforts go into developing robust software, it is not always possible to catch all possible software bugs (and sometimes bugs in the protocol operation). Should any software errors occur, the network should be provided with the capability to go to a known state in a speedy manner [e.g., speedy manual network reinitialization (1)]. If an error occurs as a result of a new feature, then it should have the ability to disable this feature and go to a known state for which the track record is good (3). To address for a software attack that can take advantage of a protocol's "loop hole", however, requires the development of intrusion-detection schemes.

RECENT ISSUES

Much research remains to be done to address network reliability in today's complex networking environment. We briefly touch on two areas in this regard: multilayered networking architecture and software errors/attacks.

Networking environment is evolving to various services being provided over multiple interconnected networks with different technologies and infrastructure. For example, the voice service is provided over circuit-switched networks, which are carried over the transmission network. Similarly, for Internet, applications such as web, email, and file transfers are carried over internet protocol (IP) layer connected by routers, which can be connected to the same transmission network or carried over an asynchronous transfer mode (ATM) or frame relay layer and then over the same transmission network. Thus, we are moving to an environment that we have coined the *multinetwork* environment. In such environment, in each of these networking layers, different types of failures/attacks and responses are possible. Some work in recent years has addressed this subject to some extent (7,13–17). It remains to be seen the impact of the failure propagation from one network to another, how the restoration process at each of these layers interacts with one another, whether they can make the best use of the network resources, and what type of network management coordination is needed for this purpose. Thus, network reliability in such interconnected multitechnology architecture needs further research.

Software/protocol operations errors and software attacks encompass the other area where mechanisms are needed to provide network reliability. This subject is relatively new—research on intrusion detection mechanisms is currently being explored to determine if an attack has occurred. Also, we need to see more work that helps us understand how severely the network will be affected in terms of network performance if a software attack or protocol failure occurs and how to recover from this anomaly. Also, the network architecture should be revisited to identify if there are ways to reconfigure the network after an attack so that parts of the network remain operational.

BIBLIOGRAPHY

1. B. A. Coan and D. Heyman, Reliable software and communication: III. Congestion control and network reliability, *IEEE J. Select. Areas Commun.,* **12**: 40–45, 1994.
2. S. Dugan, Cyber sabotage, *Infoworld,* **19**(6): 57–58, 1997.
3. D. J. Houck, K. S. Meier-Hellstern, and R. A. Skoog, Failure and congestion propagation through signalling controls, in J. Labetoulle and J. Roberts (eds.), *Proc. 14th Intl. Teletraffic Congr.,* Amsterdam: Elsevier, 1994, pp. 367–376.
4. M. O. Ball, C. J. Colbourn, and J. S. Provan, Network reliability, in M. O. Ball, et al., (eds.), *Network Models, Handbook of Operations Research and Management Science,* Vol. 7, Amsterdam: Elsevier, 1995, pp. 673–762.
5. E. Moore and C. Shannon, Reliable circuits using less reliable relays, *J. Franklin Inst.,* **262**: 191–208, 281–297, 1956.
6. V. O. K. Li and J. A. Silvester, Performance analysis of networks with unreliable components, *IEEE Trans. Commun.,* **32**: 1105–1110, 1984.
7. D. Medhi, A unified approach to network survivability for teletraffic networks: Models, algorithms and analysis, *IEEE Trans. Commun.,* **42**: 535–548, 1994.
8. M. Grötschel, C. L. Monma, and M. Stoer, Design of survivable networks, in M. O. Ball, et al. (eds.), *Network Models, Handbook of Operations Research and Management Science,* vol. 7, Amsterdam: Elsevier, 1995, pp. 617–672.
9. G. Jakobson and M. Weissman, Alarm correlation, *IEEE Netw.,* **7** (6): 52–59, 1993.
10. S. Kätker and K. Geihs, A generic model for fault isolation in integrated management systems, *J. Netw. Syst. Manage.,* **5**: 109–130, 1997.
11. W. D. Grover, Distributed restoration of the transport network, in S. Aidarous and T. Plevyak (eds.), *Telecommunications Network Management into the 21st Century,* Piscataway, NJ: IEEE Press, 1994, pp. 337–417.
12. T.-H. Wu, *Fiber Network Service Survivability,* Norwood, MA: Artech House, 1992.
13. R. D. Doverspike, A multi-layered model for survivability in intra-LATA transport networks, *Proc. IEEE Globecom'91,* 1991, pp. 2025–2031.
14. R. D. Doverspike, Trends in layered network management of ATM, SONET, and WDM technologies for network survivability and fault management, *J. Netw. Syst. Manage.,* **5**: 215–220, 1997.
15. K. Krishnan, R. D. Doverspike, and C. D. Pack, Improved survivability with multi-layer dynamic routing, *IEEE Commun. Mag.,* **33** (7): 62–69, 1995.
16. D. Medhi and R. Khurana, Optimization and performance of network restoration schemes for wide-area teletraffic networks, *J. Netw. Syst. Manage.,* **3**: 265–294, 1995.
17. D. Medhi and D. Tipper, Towards fault recovery and management in communication networks, *J. Netw. Syst. Manage.,* **5**: 101–104, 1997.

Reading List

This list includes work on network reliability that address different failure and fault issues. This list is by no means exhaustive. This sampling should give the reader some feel for the wide variety of work available for further reading, as well as lead to other work in this subject.

Y. K. Agrawal, An algorithm for designing survivable networks, *AT& T Tech. J.,* **63** (8): 64–76, 1989.

D. Bertsekas and R. Gallager, *Data Networks,* 2nd ed., Englewood Cliffs, NJ: Prentice-Hall, 1992.

C. Colbourn, *The Combinatorics of Network Reliability,* Oxford, UK: Oxford Univ. Press, 1987.

P. J. Denning (ed.), *Computers Under Attack: Intruders, Worms, and Viruses,* Reading, MA: ACM Press & Addison-Wesley, 1990.

B. Gavish et al., Fiberoptic circuit network design under reliability constraints, *IEEE J. Select. Areas Commun.,* **7**(8): 1181–1187, 1989.

B. Gavish and I. Neuman, Routing in a network with unreliable components, *IEEE Trans. Commun.,* **40**: 1248–1258, 1992.

A. Girard and B. Sansó, Multicommodity flow models, failure propagation, and reliable loss network design, *IEEE/ACM Trans. Netw.,* **6**: 82–93, 1998.

W. D. Grover, Self healing networks: *A distributed algorithm for k-shortest link-disjoint paths in a multigraph with applications in real time network restoration,* Ph.D. Dissertation, Univ. Alberta, Canada, 1989.

Fault Management in Communication Networks, Special Issue of *J. Netw. Syst. Manage.,* **5** (2): 1997.

Integrity of Public Telecommunication Networks, Special Issue *IEEE J. Select. Areas Commun.,* **12** (1): 1994.

Y. Lim, Minimum-cost dimensioning model for common channel signaling networks under joint performance and reliability constraints, *IEEE J. Select. Areas Commun.,* **8** (9): 1658–1666, 1990.

C. L. Monma and D. Shallcross, Methods for designing communications networks with certain two-connected survivability constraints, *Oper. Res.,* **37**: 531–541, 1989.

L. Nederlof et al., End-to-end survivable broadband networks, *IEEE Commun. Mag.,* **33** (9): 63–70, 1995.

B. Sansó, F. Soumis, and M. Gendreau, On the evaluation of telecommunication networks reliability using routing models, *IEEE Trans. Commun.,* **39**: 1494–1501, 1991.

D. Shier, *Network Reliability and Algebraic Structures,* Oxford, UK: Oxford Univ. Press, 1991.

D. Tipper et al., An analysis of congestion effects of link failures in wide-area networks, *IEEE J. Select. Areas Commun.,* **12**: 179–192, 1994.

DEEPANKAR MEDHI
University of Missouri—Kansas City