Impact Analysis of Faults and Attacks in Large-Scale Networks

Monitoring and quantifying component behavior is key to making networks reliable and robust. The agent-based architecture presented here continuously monitors network vulnerability metrics, providing new ways to measure the impact of faults and attacks.

Salim Hariri, Guangzhi Qu, Tushneem Dharmagadda, and Modukuri Ramkishore
University of Arizona, Tucson

Cauligi S. Raghavendra
University of Southern California

Most of the Internet's infrastructure was designed to withstand physical failures—such as broken wires or computers—rather than attacks launched by legal network users.[1–3] The Internet's rapid growth, however, coupled with its cost-effective ability to move data across geographically dispersed heterogeneous information systems, has made it a virtual breeding ground for attackers. Furthermore, the improvisation and sophistication of hackers' attack strategies and methods have overshadowed progress in security systems development.

A sustained attack on the Internet could cause a catastrophic infrastructure breakdown. According to a study on the Internet's structure, its reliance on a few key nodes makes it especially vulnerable to organized attacks by hackers and terrorists.[4] According to that report, if 1 percent of the key nodes were disabled, the Internet's average performance would be reduced by a factor of two; if 4 percent were shut down, the Internet's infrastructure would become fragmented and unusable.

To make networked systems reliable and robust, we need vulnerability metrics that let us monitor, analyze, and quantify network and application behavior under a range of faults and attacks. Here, we present an agent-based framework for analyzing network vulnerability in real time. The framework lets us quantify how attacks and faults impact network performance and services, discover attack points, and examine how critical network components behave during an attack or system fault.

Attack analysis

In most network attacks, attackers overwhelm the target system with a continuous flood of traffic designed to consume all system resources (such as CPU cycles, memory, network bandwidth, and packet buffers). These attacks degrade service and can eventually lead to a complete shutdown.[5]

There are two common types of attacks:

- *Server attacks.* There are many types of server attacks,[6] including TCP SYN, Smurf IP, ICMP flood, and Ping of Death attacks. In some attacks, the attacker makes overwhelming connection requests to a victim server with spoofed source IP addresses. Due to TCP/IP protocol stack vulnerabilities, the victim server cannot complete the connection requests and wastes all of its system resources. As a result, the server cannot service legitimate traffic, which severely impacts network performance.
- *Routing attacks.* Distributed denial-of-service (DDoS) attacks increasingly focus on routers. Once a router is compromised, it will forward traffic according to the attackers' intent. Similar to server attacks, the attackers aim to consume all router resources, forcing the router to drop all incoming packets, thus negatively affecting network performance and behavior.

Analyzing vulnerability in networks and in Internet infrastructure still is in its infancy, and there's much room for improvement. Several existing tools, which are based on modeling network specifications, fault trees, graph models, and performance models, analyze vulnerability by checking logs of system software and monitoring performance metrics.[5,7] In what follows, we briefly high-

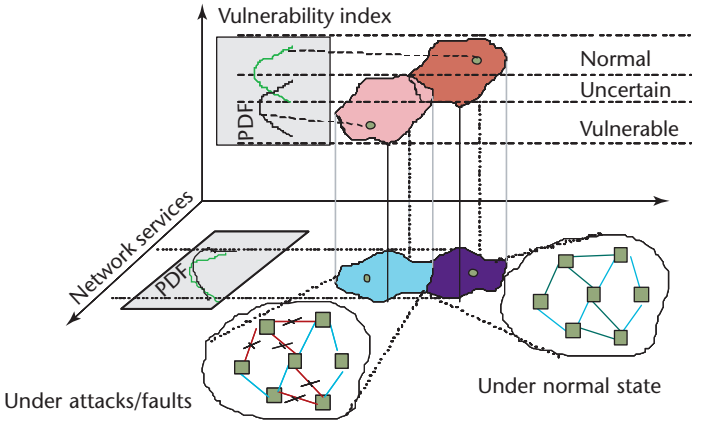

Figure 1. The vulnerability index. A network's state is monitored in relation to the VI, which measures the state in relation to normal operational thresholds and thus provides a mathematical basis for real-time response to faults and attacks.

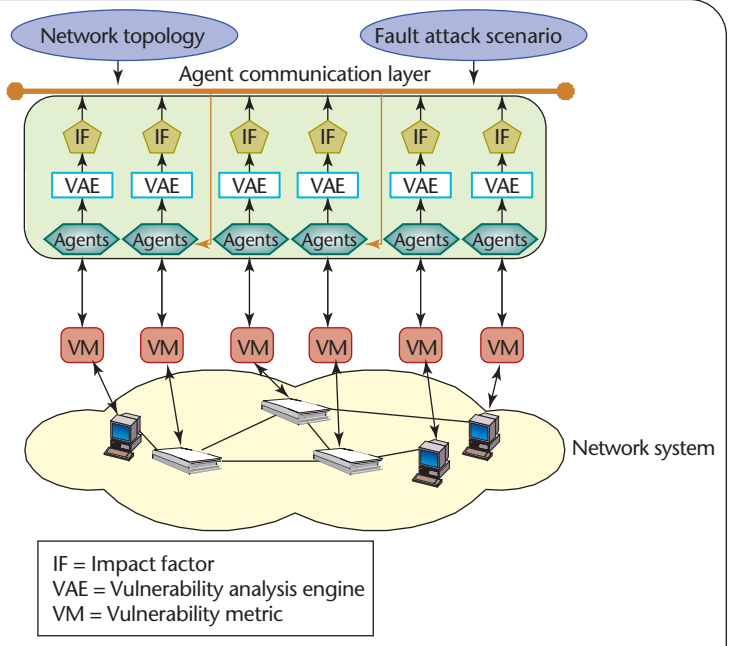

Figure 2. Vulnerability analysis framework. Agents calculate vulnerability and impact factors in real time, and generate events if the metrics significantly change. The vulnerability analysis engine (VAE) correlates events and computes vulnerabilities.

light each of these techniques.

- *Survivability analysis of network specifications.* In this approach, a system architecture injects fault and intrusion events into a given network specification, and then visualizes the effects in scenario graphs.[8] Using model checking, Bayesian analysis, and probabilistic analysis, it provides a multifaceted network view of a desired service.
- *Attack trees.* This approach determines which attacks are most feasible and therefore most likely in a given environment, and quantifies vulnerability by mapping known attack scenarios into trees.[9] Attack trees assume that all vulnerability paths are known and can be defined as possible or impossible. This can change as new attacks are discovered, however, to sudden render a previously impossible node possible.
- *Graph-based network-vulnerability analysis.* This approach analyzes risks to specific network assets and examines the possible consequences of a successful attack.[10] As input, the analysis system requires a database of common attacks (broken into atomic steps), specific network configuration and topology information, and an attacker profile. Nodes identify an attack stage, such as the machine class the attacker has accessed and the user privilege level he or she was compromised. Using graph methods lets you identify the attack paths with the highest probability of success.

Each of these methods uses offline analysis after the attacks have occurred. None quantifies the vulnerability or the attack's impact with certainty. From a network management perspective, it's critical to analyze faults and attacks during runtime to detect and protect against them proactively. Also, all these methods focus on software attacks, but, as we saw on 9/11, physical attacks are also possible.

Our online monitoring and analysis framework is general, and can be used to analyze the system operational state and performance degradation for any given fault or attack scenario. Our approach does not use knowledge about specific types of faults or attacks, because an attack's ultimate goal is to force a particular component (server, router, or link) to operate in an unacceptable manner. We designed our metrics to detect these events once they occur.

Impact analysis approach

Our approach's overall goal is to formulate a theoretical basis for constructing global metrics. System administrators then can use the framework to analyze and proactively manage the effects of complex network faults and attacks, and recover accordingly.

Measuring vulnerability

Our vulnerability index (VI) metric quantifies network system states (such as *normal, uncertain,* and *vulnerable*), much like a biological system uses metrics such as temperature and blood pressure to quantify an organism's health.[11] As Figure 1 shows, online monitoring determines whether a network's VI exceeds normal operational thresholds when it encounters an attack or faults.

This provides the mathematical basis for proactively responding to faults and attacks in real time.

Based on this methodology, we developed an agent-based vulnerability analysis framework (see Figure 2). Our framework's primary goals are to identify critical network resource points whose failure would severely impact overall system behavior, and proactively configure the network to provide quality of service in case of attacks and faults.

The framework's client, server, and router agents calculate vulnerability impact factors in real time (that is, they measure the ratio between the changes a fault or attack causes against the minimum change required to move the component from a normal to abnormal state). They also generate events whenever metrics significantly change, using *metric collectors* to monitor individual metrics. The agents interact via a layer that supports data and control communications. The vulnerability analysis engine (VAE) statistically correlates the agent-generated events and computes component or system vulnerability and impact metrics.

Component and system impact analysis

We can define the impact of a fault or attack at either the component or system level.

Component impact factor. The CIF characterizes and quantifies impact on individual network components, such as a client, server, or router. For example, as Equation 1 shows, we can define the CIF on a client for a given fault scenario (FS_k) as the ratio between the decrease in data transfer rate due to a fault scenario FS_k ($TR_{norm} - TR_{fault}$) and the minimum decrease in data transfer rate ($TR_{norm} - TR_{min}$). We define the minimum decrease to be the minimum reduction in the normal data transfer rate that will make a client operate in an abnormal state.

$$CIF(Client, FS_k) = \frac{|TR_{norm} - TR_{fault}|}{|TR_{norm} - TR_{\min}|}, \quad (1)$$

where TR_{norm} is the transfer rate during normal network operation, TR_{fault} is the transfer rate due to a fault or attack scenario, and TR_{min} is the minimal transfer rate threshold at which users can acceptably operate the system. We assume, for example, that given normal network operation, the client's data transfer rate is 100 kilobits per second and that the TR_{min} is set to 5 Kbps. If a fault or attack decreases the client's data transfer rate below 5 Kbps, the operational state is unacceptable (vulnerable).

Similarly, we can compute a router's CIF using buffer utilization as the metric to quantify a fault scenario's impact on router behavior as

$$CIF(Router, FS_k) = \frac{|B_{fault} - B_{norm}|}{|B_{\max} - B_{norm}|}, \quad (2)$$

where B_{norm} is a normal operation's average buffer utilization, B_{fault} is the buffer utilization during a fault scenario, and B_{max} is a router's maximum buffer utilization during normal operation.

We compute server CIF based on the connection queue length

$$CIF(Server, FS_k) = \frac{|CQ_{fault} - CQ_{norm}|}{|CQ_{\max} - CQ_{norm}|}, \quad (3)$$

where CQ_{norm} is a normal operation's connection queue length, CQ_{fault} is the connection queue length during a fault scenario, and CQ_{max} is the server's maximum connection queue length during normal operation.

In Equations 1 through 3, we use data transfer rate, buffer utilization, and connection queue length to quantify fault impact, but we can use other metrics as well. To compute a router's CIF, for example, we could use the number of flows open or in process, the total number of flows, or the request-processing rates. We can also dynamically adjust the normal and abnormal behavior thresholds to accurately characterize any network component's operational state.

System impact factor. The SIF identifies how a fault affects the whole network or a subnetwork. For any given fault, we obtain the SIF by evaluating the weighted impact factors of all network components. That is, we evaluate SIF by determining, in relation to the total number of components, the percentage of components operating in vulnerable states (those with CIFs that exceed normal operational thresholds). We compute the overall impact of a given fault or attack on clients and routers as

$$SIF_{Client}(FS_k) = \frac{\sum_{\forall j, CIF_j > d} COS_j}{total\_number\_clients}, \quad (4)$$

and

$$SIF_{Routers}(FS_k) = \frac{\sum_{\forall j, CIF_j > d} COS_j}{total\_number\_routers}, \quad (5)$$

respectively, where d denotes the upper threshold of normal operating conditions and the binary variable COS denotes the component's operation state. COS is equal to 1 when the client operates in an abnormal state (that is, $CIF_i > d$), and 0 when it operates in a normal state (that is, $CIF_i < d$).

Simulation results: Validation and analysis

To validate our approach and demonstrate its capabilities in online monitoring, analysis, fault–attack detection, and recovery, we developed an instrumented simulation

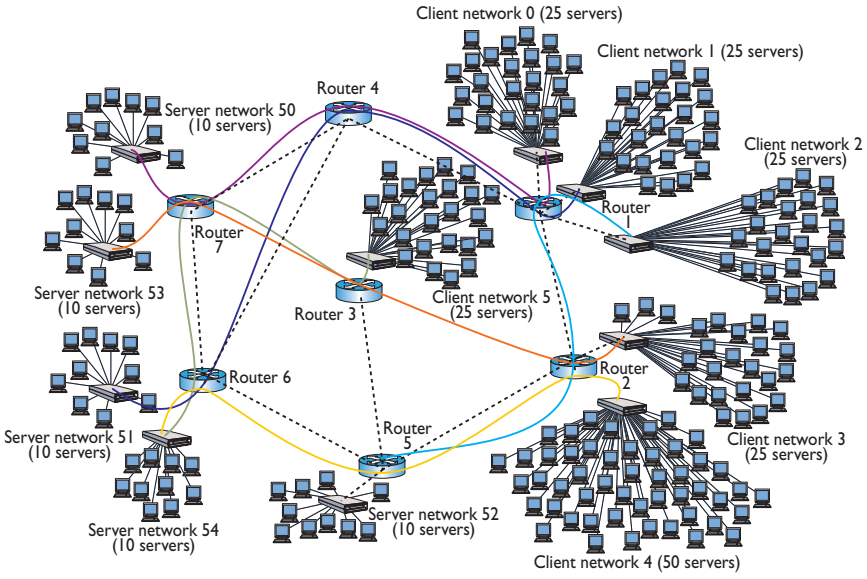

Figure 3. The simulated environment for vulnerability impact analysis. The network consists of six client and five server networks configured for simple file transfer using TCP/IP.

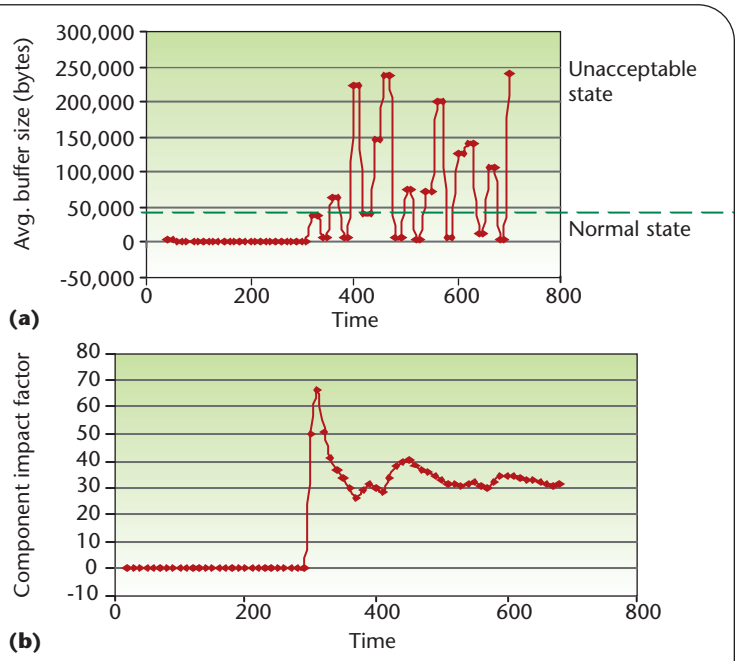

Figure 4. Single router failure. (a) Router 3's average buffer size and (b) router 3's component impact factor.

environment using the Scalable Simulation Framework Network tool (SSFNet; see www.ssfnet.org). Here, we focus on monitoring, analysis, and detection; we discuss how these metrics facilitate proactive recovery mechanisms elsewhere.[12]

Our simulation environment lets us inject faults or attacks into a simulated network system. Once we start the simulation, VAEs continuously compute CIFs based on monitored vulnerability metrics. Agents exchange these CIFs through the agent communication layer. When an agent receives CIF values, it computes the overall system impact factor (see Equations 4 and 5).

As Figure 3 shows, our network topology consists of six client networks and five server networks, and we assume a node-pair bandwidth of 100 Mbps. We configured all network clients and servers for simple file transfers using TCP/IP. The network has 243 clients and servers running TCP/IP protocol stack. The core backbone consists of seven routers, and we use Open Shortest Path First (OSPF) as the routing protocol.

To quantify the impact of single and multiple router failures, we analyzed several vulnerability metrics including router buffer size and client data transfer rate. We consider

- core router operation unacceptable when buffer size exceeds 50,000 bytes, and
- client operation unacceptable when the data transfer rate drops below 70 Kbps.

To characterize a network component's state, we use CIF: if a router's CIF is greater than 35 percent, its working state is unacceptable; a client's working state is unacceptable if its CIF is greater than 30 percent.

Single router failure

In this scenario, we separately failed routers 1 through 5 and 7. Router 4's failure occurred at 300 seconds. As Figure 4a shows, the buffer size for interface 2 of router 3 increased drastically after the failure; it reached 250,000 bytes at 400 seconds. We observed the same behavior using CIF. As Figure 4b shows, router 3's CIF is about 35 percent. Figure 5a shows how router 4's failure affected network 0's clients: their transfer rate increased to 70 Kbps, and each client's CIF dropped below 30 percent after the 300-second failure.

We can use CIF metrics to obtain global system impact metrics due to faults or attacks. Router 4's failure, for example, will significantly impact the core routers. As Figure 6a shows, according to the SIF for this failure scenario, more than half of the core routers will operate in an unacceptable buffer utilization state. We can also quantify how the failure affects the overall client population; the

failure rate can reach 25 percent (see Figure 6b).

Multiple router failures

Our approach also lets us analyze the impact of multiple failures on overall system behavior and identify critical resources and vulnerabilities. To illustrate this, we failed a pair of routers and studied the network impact. We used the same thresholds as in the single router failure. We failed multiple routers at 300 seconds of simulation time, and set the total simulation time at 700 seconds.

We failed routers 4 and 5 simultaneously, which affected routers 2, 3, and 7. All traffic for routers 4 and 5 was rerouted through router 3, which created congestion and severely degraded its performance. For this fault scenario, 70 percent of core routers and 82 percent of the clients operated in an unacceptable state (see Figures 7a and 7b).

Single and multiple router failures let us determine the network's most critical routers (indicated by the boldface percentages in Table 1). We evaluate router importance based on how failures impact routers' and clients' SIF. As Table 1 shows, router 4 was the most critical router: its failure affected more than 54 percent of the network's core routers and 31 percent of its clients. The impact on clients was less severe because—as Figure 3's topology shows—when router 4 failed, client traffic was rerouted to servers through other routers (in this case, router 2).

As Table 1 shows, router 2's failure affected more than 47 percent of the clients, because this failure isolated clients from networks 3 and 4. Finally, the simultaneous failure of routers 4 and 5 had the most impact: 71 percent of core routers and 82 percent of clients were affected.

We are currently building a testbed that consists of 10 routers and 40 workstations at the ITL laboratory of The University of Arizona to evaluate the online monitoring and vulnerability metrics discussed in the paper. In addition, we will be evaluating vulnerability metrics with respect to multiple attributes (number of unsuccessful sessions, packet transfer rate per destination, for example) in order to improve the accuracy of our approach to quantify the impact of faults and attacks. We are developing a Quality of Protection (QoP) routing protocol based on our vulnerability metrics to mitigate and eventually eliminate the impact of attacks on networks and their services. □

Acknowledgments

This research was supported in part by DARPA/SPAWAR contrct no. N66001-02-C-6024.

Table 1. Core router criticality.

ROUTER FAILED	CORE ROUTERS SIF (PERCENT)	TOTAL CLIENTS SIF (PERCENT)
Critical router (single failure)		
Router 2	27.1428	**47.62623**
Router 5	41.428	38.92857
Router 1	17.857	38.43407
Router 4	**54.28571**	34.079
Router 7	28.21429	31.799
Router 3	23.2142	28.06319
Critical routers (multiple failures)		
Router 4 and 5	**71.4285**	82.417
Router 2 and 4	27.24	**84.5824**
Router 3 and 5	42.87	82.794

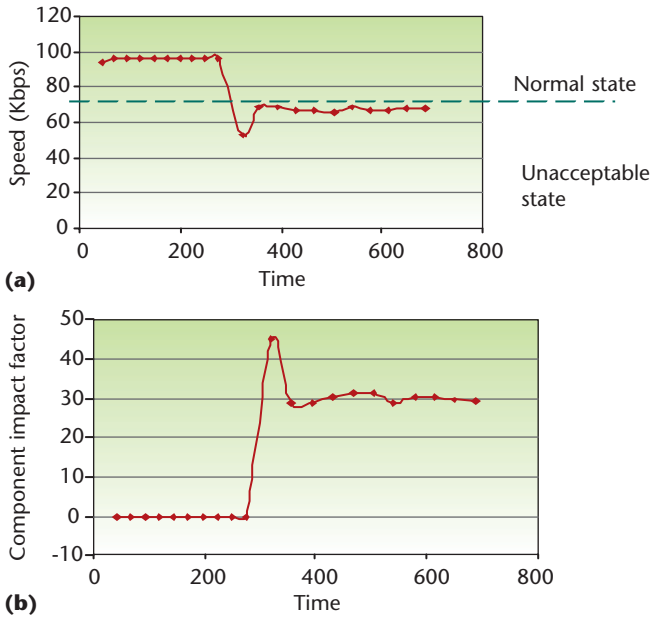

Figure 5. The impact of router 4's failure. (a) The transfer rate for a network 0 client. (b) The component impact factor for a network 0 client.

References

1. *US Infrastructure Assurance Prosperity Game Final Report;* President's Commission on Critical Infrastructure Protection, www.ciao.gov/resource/pccip/ProsperityGames.pdf.
2. G. Huston, "The Unreliable Internet," *ISP Column*, May 2001; www.potaroo.net/ispcolumn/2001-05-reliable.html.
3. Nat'l Research Council, *Committee on the Internet under Crisis Conditions: Learning from the Impact of September11*," Nat'l Academies Press, 2003.
4. R. Albert, H. Jeong, and A.-L. Barabási, "The Internet's Achilles' Heel: Error and Attack Tolerance of Complex

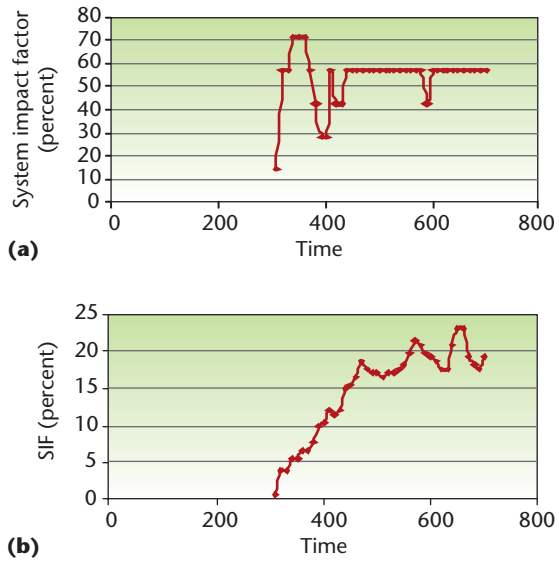

Figure 6. A single failure's system impact. (a) The system impact factor of a router with a single failure. (b) The system impact factor for clients with a single failure.

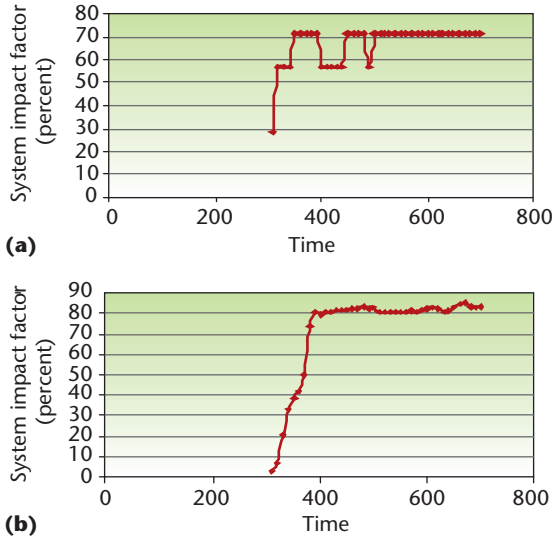

Figure 7. Multiple failure's system impact. (a) The system impact factor of core router with multiple failures. (b) The system impact factor of clients with multiple failures.

Networks," *Nature*, vol. 406, July 27, 2000, pp. 378–382.
5. C. Sample and I. Poynter, "Quantifying Vulnerabilities in the Networked Environment: Methods and Uses," 2000; http://builder.itpapers.com/abstract.aspx?docid=21491.
6. *Denial of Service*, tech. report, Malaysian Computer Emergency Response Team (MyCERT) 2003; www.mycert.org.my/network-abuse/dos.htm.
7. Y. Dong, Y. Hou, and Z.-L. Zhang, "A Server-Based Performance Measurement Tool within Enterprise Networks," *J. Performance Evaluation*, vols. 36–37, Nov. 15, 1999, pp. 233–247.
8. S. Jha et al., "Survivability Analysis of Network Specifications," *Proc. Int'l Conf. on Dependable Systems and Networks* (DSN2000), IEEE CS Press, 2000, pp. 613–622.
9. B. Schneier, "Attack Trees," *Dr. Dobb's J.*, vol. 12, Dec. 1999; www.ddj.com/articles/1999/9912.
10. L.P. Swiler, C. Phillips, and T. Gaylor, "A Graph-Based Network-Vulnerability Analysis System," tech. report, Sandia Nat'l Labs, 1998; www.prod.sandia.gov/cgi-bin/techlib/access-control.pl/1997/973010-1.pdf.
11. M. Parashar, *Interpretive Performance Prediction for High Performance Computing*, PhD thesis, Dept. of Computer Eng., Syracuse Univ., 1994.
12. S. Hariri et al., *Quality of Protection (QoP) Routing Protocol—An Online Network Defense Mechanism*, tech. report, Internet Tech. Lab., University of Arizona, Aug. 2003.

Salim Hariri *is a professor in the Department of Electrical and Computer Engineering at the University of Arizona and director of the Center for Advanced TeleSysMatics (CAT): Next Generation Network Centric Systems. His current research focuses on autonomic computing, high-performance distributed computing, design and analysis of high-speed networks, benchmarking and evaluating parallel and distributed systems, developing software design tools for high-performance computing and communication systems, and network-centric applications. He is editor-in-chief for* Cluster Computing Journal, *founder of the IEEE International Symposium on High-Performance Distributed Computing (HPDC), and cofounder of the NSF Workshop on Active Middleware Services (now the Autonomic Computing Workshop). He received a PhD in computer engineering from the University of Southern California and an MS from The Ohio State University. Contact him at ITL Lab, University of Arizona, Tucson, AZ, 85719; hariri@ece.arizona.edu.*

Guangzhi Qu *is a doctoral student at the ITL Lab, the University of Arizona, Tucson. Contact him at ITL Lab, University of Arizona, Tucson, AZ, 85719; qug@ece.arizona.edu.*

Tushneem Dharmagadda *works at Analog Devices Company, Wilmington, Mass. Contact him at tushneem.dharmagadda @analog.com.*

Modukuri Ramkishore *is a graduate student at the ITL Lab, the University of Arizona, Tucson. Contact him at ITL Lab, University of Arizona, Tucson, AZ, 85719; kishore@ece.arizona.edu.*

Cauligi S. Raghavendra *is a professor in and chairman of the Department of Electrical Engineering-Systems at USC, and was previously the Boeing Chair Professor of Computer Engineering in the School of Electrical Engineering and Computer Science at the Washington State University, Pullman. He received the Presidential Young Investigator Award and became a Fellow of the IEEE. He received his BSc (Hons) in physics degree from Bangalore University, his BE and ME degrees in electronics and communication from the Indian Institute of Science, Bangalore, and his PhD in computer science from University of California, Los Angeles. Contact him at raghu@ halcyon.usc.edu.*