Int. J. Engng Ed. Vol. 17, Nos. 4 and 5, pp. 460–467, 2001
Printed in Great Britain.

0949-149X/91 $3.00+0.00
© 2001 TEMPUS Publications.

Educating Designers of Complex Systems: Information Networks as a Testbed*

STEPHEN J. LUKASIK
Center for International Security and Cooperation, Stanford University and Harvey Mudd College, 1714 Stone Canyon Road, Los Angeles, CA 90077, USA. E-mail: stephen.j.lukasik@cpmx.saic.com

MICHAEL ERLINGER
Computer Science Department, Harvey Mudd College, 301 E. 12th Street, Claremont, CA 91711, USA

The needs of modern societies, coupled with the ever increasing power of technology, encourage the development of systems of enormous complexity. Examples are contemporary infrastructures such as transportation, energy, and information, often described as 'systems of systems'. Coping with such complexity is a central problem for their architects and operators. Complex systems behave in unanticipated ways and they have failed spectacularly. The factors influencing system failure go beyond the purely technical, to matters of organization, management, operations, training, regulation, and market incentives, all concerns of the engineering profession. Failures often result from subtle combinations of such factors. Public policy has been directed to the vulnerabilities of national infrastructures, especially as they can be disrupted by both inadvertant failure and malicious attack. If engineers are to increase the robustness of the systems they design and operate, they must recognize the phenomenon of emergent properties rooted in the scale and 'depth' of the system. The Internet, while an operational infrastructure, is also capable of supporting experimentation without the knowledge of, or interference with, its users. Harvey Mudd College's CS 125 course, Computer Networking, covers principles and practices of computer networking. The course has a significant project component that relies on the Internet as an experimental facility. Oversight of such educational and research activities is required if network performance is not to be degraded or user privacy violated. Several examples are presented. An appreciation of emergent properties of systems should be a baccalaureate-level goal, not simply for computer science curricula, but for all of systems engineering.

COMPLEX SYSTEMS

THE NEEDS of modern societies, coupled with the ever increasing power of technology, encourage the development of systems whose complexity can be virtually without limit. The North American electric power infrastructure of generation, transmission, and distribution facilities is one example. Another is the air transport system, consisting of aircraft designers, manufacturers, owners, and operators; ground and satellite-based air-traffic control and communication facilities; air operations and maintenance support, and pilot and crew training.

The complexity of such systems is not designed-in, for no one would be foolhardy enough to undertake such constructions from scratch, or wealthy enough to afford the capital investment. Such systems arise in some cases from the merger of regional enterprises, as with the telephone and rail infrastructure. In others the complexity simply accumulates over long periods, with newer technology layered over older technology, as in large enterprise software systems. Some systems are self-organizing, illustrated in biological and ecological systems. Others arise from the working of the blind hand of the market, as in the case of material recycling. Government regulatory policies encouraging competition and the breakup of previously centrally-managed monopolies result in organizations that interact across a multitude of newly defined interfaces.

Thus, in critical aspects of society we depend on systems that were not 'designed'; for which no single person or organization is 'in charge'; and where the details of their various parts are unknown, continually changing, and thus effectively unknowable. Our most important systems consist of a very large number of separate parts, are large in geographical extent, span numerous legal and political jurisdictions, grow over time, and have so great a replacement value that their systematic modernization poses substantial practical difficulties.

It is systems such as these, typically public, but in some cases private or mixed public/private, infrastructure systems that are the focus of this paper. Transportation, communication, information, energy storage and delivery, emergency services, health care, and the like are current instantiations. They are the result of both accretion and fragmentation processes. They are often described as systems of systems.

Our starting point is to argue that while one is, in principle, able to understand in substantial

* Accepted 3 December 2000.

detail a single 'system' at some sufficiently low level, the joining together of smaller systems into larger entities results in assemblages whose behavior is neither known nor predictable. This may be because systems of systems exhibit properties in the large that are not exhibited by their smaller parts. Or it may be the result of their large geographical and organizational extent, and the multiplicity of internal interfaces that are either unknown or only imperfectly controllable.

Information technology, especially the technology of networking of computer-based systems, plays a critical role in enabling and even encouraging the creation of systems of virtually unlimited complexity. The exchange of digitally-encoded information across an interface defined both by technical standards at the physical level and by agreed upon protocols for how the transferred information is to be interpreted has made possible a world of unprecedented technical and social complexity.

FAILURES IN COMPLEX SYSTEMS

This ability to build enormous systems, capable of addressing increasingly large and complex tasks, is proving to be both a blessing and a curse. Coping with complexity is a central problem for the architects and operators of such systems. Principles such as hierarchical organization; partitioning into manageable subunits; the provision of redundant capacity; establishing and encouraging adherence to technical standards; reuse of proven designs; requiring that designs 'fail-soft;' and employing systematic fault-tree analysis techniques help to some extent and reduce the likelihood of catastrophic system failure.

But complex systems nevertheless behave in ways not anticipated by their owners and operators and can fail in spectacular ways. Ships collide in clear weather; aircraft fly into mountains in daylight; power blackouts cover major areas of the country; and during the Three Mile Island nuclear reactor accident, operators were powerless for thirty-six hours because they lacked an understanding of what was happening within the containment vessel. These and other cases of pathological behaviors of complex systems have been studied to understand what characteristics lead to them [1].

Perrow concludes, based on analyses of failures in a number of different types of systems, that unanticipated behavior can be expected to arise from the interplay of two system properties. *Complexity* is one, and this property is characterized by:

- tight spacing of equipment;
- many common-mode connections of components not in production sequence;
- limited isolation of failed components;
- personnel specialization that limits awareness of interdependencies;
- limited substitution of supplies and materials;
- unfamiliar or unintended feedback loops;
- many control parameters with potential interactions;
- indirect or inferential information sources;
- limited understanding of some processes (typically transformational processes).

A second, and equally important property, is that of *tightness of coupling* between the parts of the system. Tight coupling in systems is characterized by:

- delays in processing are not possible;
- sequence of processes is invariant;
- goals can be achieved in only one way;
- little slack in supplies, equipment, and personnel is available or possible;
- buffers and redundancies are deliberate and must be designed-in;
- substitutability of supplies, equipment, and personnel are limited and designed-in.

The causal factors influencing system failure go beyond the purely technical, and include matters of organization, management, operations, training, regulation, and market incentives. While sometimes ascribed to 'operator error,' failures in complex systems are often preordained by combinations of engineering design and operational procedures. In short, the full range of concerns of the engineering profession is involved.

Recent concern in the public policy sector has been directed to the vulnerabilities of national infrastructure systems, especially as their operation can be disrupted by accidental failures of their exquisitely complex information-based subsystems and their control software, or as they may be targets of malicious attacks on their control subsystems [2]. The trend to relying increasingly on information technology in all aspects of system operation is driven by the need to become more competitive by reducing cost, by enhancing labor productivity, by integrating operations to achieve economies of scale, by reducing inventories, and by increasing the rate of asset turnover. Thus pressure to adopt information technology ever more widely has the result of increasing the vulnerability of ever larger parts of our national infrastructures.

Accidental failures reflect our inability to deal with system complexity. Increasingly, commercial and industrial operations depend on the global Internet, linking a million organizations, ten million computer systems, and over a hundred million users. Internet growth has been so rapid that some of its operators fear collapse. The Internet is a combination of hardware and software systems whose interplay increases the complexity of its individual parts. One of the primary attributes of the design of Internet protocols was to make it possible to move information over almost any type of hardware medium, such as twisted pair Ethernet, ATM fiber, and satellites. The resulting infrastructure is extremely complex, especially at the interfaces between different transmission media. These multiple complex interfaces lead to many difficulties. In terms of software, Internet protocols are specified in word descriptions that, at some level, are ambiguous. These ambiguities in turn lead to incompatibilities between protocol implementations that result in impediments to improving network performance.

Malicious attacks on this critical information infrastructure are common. There are various reasons behind Internet attacks, all of which compound the technical problems. First and foremost, the Internet is vulnerable because it was never designed with a consideration for adversaries. It was created by and for researchers to enable them to develop network technology and to work collaboratively. At the beginning of the ARPA network research program, security was viewed as an issue to be deferred. The vulnerabilities then tolerated now pose a challenge to both attackers and defenders. Upsetting Internet operation, breaking into networks and computers connected to the Internet, and creating viruses has become a game. Hacker tools, readily available on the Internet itself, enable unauthorized users to break into systems for fun and to leave evidence of their presence. While in one sense this is annoying but harmless, hacker activities result in the expenditure of time and effort to repair the breached systems. More serious in their impact are the crackers who recognize the importance of the Internet for commercial and government applications and attempt to steal, destroy, and confuse.

LEARNING ABOUT COMPLEX SYSTEMS

If engineers are to increase the robustness of the systems they design and operate, they need to become better acquainted with at least some of their essential features. In particular, the fact of emergent properties that arise, not as easily deducible consequences of the fundamental design models on which the system is based, but as somehow rooted in the size, scale, and 'depth' of the system constitutes a minimum undergraduate exposure to the deep issues involved. Perhaps an example from fluid dynamics can illustrate the point. While one can look long and hard at the Navier-Stokes equations when seeking analytical solutions, the phenomena of stability and turbulence are unlikely to suggest themselves. On the other hand, direct observation of physical fluids is a simple and direct way of adding these characteristics of viscous flow to the student's mental toolkit.

This paper argues that gaining such an appreciation of the emergent properties of systems should be included in undergraduate engineering education. Just as understanding fluid dynamics is greatly aided by experiment, so also, we maintain, are complex systems. This is not to argue that theory has no role to play. But from the earlier description of complex systems, and as elaborated in Ref. 1, their behavior is heavily influenced by non-analytical features such as management policies, human cognition limits, and economic and market factors. Thus as is so often the case in the early stages of science and technology, direct observation is likely to lead the way initially to understanding.

A previous paper suggested indirect ways to accomplish this [3]. But there is also an important role for direct observation and experiment, because learning is more effective the closer it can be brought to personal experience. While 'capturing' a complex operational system and introducing it into a college environment is obviously difficult, one such system is under the jurisdiction, and to some extent the control, of educational institutions: the Internet-based campus information system and its supporting communication network. This can provide a useful testbed and it is to this that attention is directed here.

What might engineering students gain through a practical exposure to system complexity? In the early discussions of the design of the ARPANET, the use of simulated traffic as a basis for network development and performance verification was abandoned because it would not capture the unpredictable variability of actual user demands. This 'real world' aspect of an operational network is what one would capture under this proposal. A second aspect is that network pathologies, as opposed to application-level pathologies, would be directly observed and measured. And third, the adversarial challenge of dealing with 'active' users who game, rather than simply load the network will add missing social dimensions to analytical models. Complex system failures frequently derive from unanticipated user behavior not contained in the models on which the system design rests.

There are various possibilities for student data collection, analysis and modeling of the Internet, and even, under appropriate conditions, for direct experimentation. Thus traffic analyses at the packet, message, and session level can be performed; the network can be probed to determine what hardware devices are connected to it and what software is running on it; audits to assist in detecting the presence of unauthorized users on the network, and to understand what data have been, or are being collected by them, or what software they have introduced into the network, can be performed; and levels of 'unusualness' can be established to serve as baselines to help detect, and possibly prevent or limit, future undesirable system behavior.

From a practical standpoint, how can this idea be accommodated in current engineering curricula that already put severe time stress and information overload on students and faculty? There would seem to be three threshold questions. First, are the matters presented here amenable to formal instruction, or do they involve subjects that are so area-specific they are better learned in a post-baccalaureate professional environment? Second, if they are matters for inclusion in engineering education, at what point in the cycle should they be addressed? And third, if curricula are relatively inelastic, how are these issues to be accommodated?

On the question of formal versus informal and general-principle versus area-specific instruction, we believe that the concept of unanticipated properties of complex systems is a minimum to be addressed as part of the undergraduate experience. The bulk of engineering graduates do not continue their formal technical education beyond the baccalaureate level. Thus, the alternatives are on-the-job training and continuing-education certificate programs. Both are feasible and important for the development of the understanding needed to cope with technology in our increasingly complex world. But at least providing students with a 'placeholder' in 'concept space' for what are inevitable implications of their technologies as they are applied in society would seem to be important.

Saying that is the easy part. The hard part is how to deal with the subject in the real world of courses, budgets, degree requirements, faculty positions, and time. Following from the idea that the issues involve more matters of experimentation than of theory, at least at this point in our understanding of the subject, one is directed to laboratory and project teaching rather than to formal course content. Viewed in this light, these issues can often be addressed through appropriate choices for laboratory instruction rather than the addition of new material *ab initio*. In essence, the suggestion is to provide opportunities for students to uncover unusual aspects of complex system behavior and, thus motivated, to seek a deeper understanding of them.

Further, while recognizing the pitfalls of proposing interdisciplinary solutions in an environment where the academic department and its tenure discipline is the dominant fact of life, it is nevertheless the case that for the proposal presented here to be effective, it requires the skills and facilities of engineering departments and computer science departments taken together. Local academic organization will be a strong factor in implementing this idea. Engineering schools and departments treat computer engineering differently. Computer science departments embrace various mixtures of theory and experiment, with some having stronger links to mathematics and others, in matters of robotics and computer architecture, to engineering.

Complicating the treatment of system complexity in academic environments is the increasingly pervasive role of information systems in all kinds of operations. Systems involving the physical movement of people and material are controlled through information and communication systems, both as integrated subsystems and as linked networks. For example, complexity and vulnerability to failure involves not only the proper design of ships, but of the information networks on which their operators depend for efficient routing, weather information, and response to emergency situations. Thus the proposal made here, for an understanding of system complexity based on direct observation, is directed both to information system performance as well as the performance of physical systems whose information component is a critical, and often troublesome, part.

THE INTERNET AS AN INFRASTRUCTURE

The Internet is usually described as a network of networks. While this description is correct, there are a number of features that enable the Internet to function and grow. Among these are the availability of underlying communication media, addressing, and common protocols and message formats. The existing and developing global communication system is an infrastructure that has allowed the Internet to develop. It is a feature of the Internet that it has been able to use this symbiosis, but without the existence of the international communication system the Internet would not have attained its current stage of development.

Each device connected to the Internet is assigned an IP address, consisting of a network number and a device address. These addresses are unique and span international boundaries. It is the network number part of the Internet address that allows networks to exist as autonomous entities, thus enabling the creation of 'network of networks.'

The most significant feature that enables the Internet to function is the set of common protocols. Each node on the Internet is required to support what has become known as the TCP/IP protocol stack. This protocol stack has several features that have facilitated the growth of the Internet: hierarchy of responsibilities and standardization. TCP/IP is a hierarchical set of protocols, each protocol complements the others. Usually, TCP/IP is organized into four levels. The lowest level is the communication medium, which can vary widely in capability. This level includes the basic network technologies including Ethernet and the universal communication infrastructure such as the telephone network.

The next higher level is the network level. At this level there is only the IP protocol (Internet Protocol). IP is responsible for moving network packets over communication facilities, from network to network, and eventually to the destination device. IP has shown it self to be easily adaptable to different communication media. IP is a 'best effort' protocol. While it makes a best effort to deliver a packet, there is no guarantee of delivery. Its job is solely to guide packets from source to destination.

The source node combines the application message with the destination IP address, breaks the message into packets, and requests delivery by entering the packets into the Internet. Various network nodes and routers are then responsible for forwarding a packet until it reaches its destination. A router checks the packet and forwards it. In checking the packet the router looks to see if the network is congested or if the packet contains a transmission error. If the router sees a problem with the packet, it drops it but tells no one. If all goes well, the router then uses the IP address to determine the best 'next hop' for the packet.

In this way, packets travel across the Internet to their final destination. The whole process of determining the 'next hop' is facilitated by a special set of router communication protocols. A weak analogy is the postal system where mail is forwarded and eventually delivered based on address. But in the postal system errors in addressing result in a returned message. IP simply drops a packet containing an error.

The next level in the protocol stack is the transmission level. Two protocols reside here, TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). While IP is only responsible for delivering packets to end devices, transmission level protocols are responsible for reliability and organization of the packets into messages. TCP on the source and destination nodes cooperate to handle lost packets and message organization. Note that transmission protocols do not care about message content, but only that the message as sent by the source is exactly the message received at the destination. This model is sometimes described as TCP riding on top of IP because the only way to move TCP messages between nodes is IP.

At the highest level are the application protocols, which do relate to message content. It is these protocols that the user is most concerned with, e.g. telnet, ftp, http, etc. Each of these protocols is defined to solve a particular application need.

The format of TCP/IP packets is well defined so that each node can parse the packet according to its needs. The usual packet format has a transmission medium header, followed by the IP header with source and destination addresses, followed by a TCP header with parameters to keep track of the underlying message format, followed by the application header and body (the real message being communicated). Routers only concern themselves with media and IP headers while end nodes must be concerned with all the headers.

Standardization underlies the success of the Internet. Each of the Internet protocols is defined as explicitly as possible and each of the protocols carries a label as to whether it is required or recommended. Thus those interested in using the Internet have well-defined protocols and procedures that each of their nodes must support. The Internet standardization process is maintained by an international organization, the Internet Engineering Task Force that seeks the implementation of proposed protocols and demonstrations of their interoperability. A new protocol is standardized only if there are multiple implementations and interoperability has been demonstrated. This process ensures that new protocols are well understood and that implementations are available prior to their general release.

It is the combination of the above attributes that have enabled globalization of the Internet. That is, an existing communication infrastructure, an addressing scheme based on hierarchy and uniqueness, a well-defined and standardized set of protocols, an open process for standardization, and protocol implementation availability. Thus any country, vendor, user, etc., who wants to join the Internet has available the technical foundation to do so.

THE INTERNET AS A LABORATORY

The Internet, while now very much a fixture as a national and international communication and information infrastructure, still has an experimental character. In some cases experiments are carried out without any overt knowledge of users. For example, IPv6 (the next version of IP) experiments are currently conducted on a network (6Bone) that is overlaid on the existing Internet. The approach taken is non-intrusive in all aspects but performance, since additional traffic does increase Internet traffic overall. 6Bone is implemented through a methodology called 'tunneling'. In tunneling the particular experimental protocol is encapsulated in a normal IP packet, leaving it to the end points to encapsulate and unencapsulate the particular experimental protocol message.

Another current approach to experimental use of the Internet involves restricting experiments to particular parts of its infrastructure, e.g. a subnet, a network, or a set of interconnected networks. One first contains experiments for software testing and only later allows them to move onto larger segments of the network. This is made possible by the ability of routers to control traffic based on IP address or packet contents, e.g. not forwarding packets from a particular source network or not forwarding telnet packets.

Performance and privacy are two central concerns when considering experiments to be performed on critical infrastructures. Within a single computer system performance measurement tools have been known to be the largest consumer of resources and experience has shown that it is possible to swamp a network with management traffic alone. One approach to mitigating performance impacts of system experimentation is to restrict experiments to use currently excess resources only, i.e. first testing the network for current utilization before starting an experiment.

Another major issue is privacy. Even the perception of invasion of privacy can be detrimental.

Thus, any experiments that involve packet sniffing (capturing all network traffic) require that the experimenters observe appropriate restrictions and that the users understand what can be exposed. But with prior review of experimental protocols, much like the processes employed in clinical experimentation in other areas of science, and with careful monitoring, auditing, and supervision, network performance need not be degraded nor user privacy compromised.

Three further suggestions for protecting privacy have been offered: (a) only allow header information to be accessed, (b) archived rather than real-time data be used, and (c) actual IP addresses be protected. Users would be made aware of the use of the network as a testbed, and means would have to be provided to satisfy user concerns over unwarranted intrusion. On the other hand, what is suggested is not different, in principle, from what system administrators and network managers already do to provide the best possible service to users. The new idea is to use the network as a tool to enable students to experience system behavior and come to appreciate the nature of complex systems.

HMC EXPERIMENTS USING THE INTERNET AS A LABORATORY

Harvey Mudd College CS 125, Computer Networking, covers principles and techniques for computer networking and analysis of networking models and protocols. This course has a significant project component, much of which uses the Internet as an experimental testbed. The following paragraphs describe some of the experiments and how they were performed on the department network, the campus network, and the Internet.

Two projects are based on developing applications which use TCP/IP and UDP/IP as communication protocols between two hosts. Students develop a complete, but simple, application and use the networking libraries of the source and destination hosts to build communications infrastructure. Students are then responsible for the application operation and for analysis of the network traffic. The major concern for these projects is network performance and not privacy. Poorly formed applications can affect the source or destination node and a poor implementation can affect the number of packets being transmitted between source and destination. In general these projects have had no performance effect on the local department network, nor on the campus network when moved there. These projects have no privacy concerns because students are only working with their own code and their own network messages. But to capture their message traffic, students need to run a network sniffer– a software tool that captures all packets seen by a particular host. In an Ethernet environment this can be all packets addressed to any host on the local net. Setting appropriate parameters for the sniffer restricts its behavior, and versions can be installed that force such limitations. A restricted sniffer has been used to allow students to investigate message contents and each of the headers in the messages associated with their application.

Other class experiments have involved looking at various protocol headers that described individual protocols. Students have been able to search for network packets containing certain protocols, and to measure overall statistical presence of various protocols, e.g. the occurrence of http traffic. Again, sniffers were setup to restrict access to packet contents beyond specific headers. A related project required students to set up a long-running sniffer to look at packets between their specific personal machine and the department server. They were then asked to analyze these packets in great detail, i.e. headers and packet contents. This project demonstrated the lack of privacy in current Internet traffic. Since it is easy to acquire a sniffer for any computer system, it is easy for any user to sniff packets. This latter project was an attempt to get students to understand the need for privacy policies within a network, and the need for enforcement of those policies.

A fundamental problem in networking is determining all the hosts on a network and the characteristics of each host. Various Internet documents describe information that should be available, but this information is inconsistently supported by network nodes. Students were asked to create a solution for this problem. Basically, given an IP address within an IP class (address range), the student application was to find all hosts on that network, print out a table of the hosts with all available information about each host, e.g. host IP address, host name, etc. This is an interesting project from a number of viewpoints. First, from a protocol point of view, there are a number of possible approaches. Thus students must have a good understanding of many Internet protocols to solve the problem. Some of the approaches are brute force and can have disastrous effects on the network. In fact, they can appear to be a network attack, e.g. 'pinging' every valid node address within a network address range. At the very least pinging can have a large performance effect. Thus, students were also given parameters on impacting network performance. Their grade depended partly on controlling their impact on network traffic.

Second, there is a privacy concern with this project. While students have a good understanding of the nodes on the department network, their knowledge of campus and other department networks is limited. Thus, moving their experiment to other networks raises an issue as to how valuable the knowledge about each host is. Numerous intrusion attacks are based on knowledge of the type of node being attacked. Thus, the mapping experiment could be viewed as a prelude to such an attack. The intent of this project was to demonstrate two things to the students: the solution of the problem and the related performance and privacy issues. While few of the students were able to finish the project, the class was able to discuss and understand both the technical issues and the related performance and privacy concerns. In the future this project will be moved to a wider network, e.g. the Harvey Mudd campus network or the Claremont Colleges network.

IN SUMMARY

Important and quite complex systems on which society depends are frequently only designed piecewise and are 'integrated' with varying degrees of success. Complex systems display emergent properties, and as a result, behave in unanticipated ways. Their accidental failures can be spectacular and, being unanticipated, are difficult to prevent. Of increasing public policy concern are criminal acts and malicious attacks directed against complex infrastructure systems.

Providing engineering students firsthand experience with emergent properties of complex systems has a place in undergraduate education. Information systems, both local and remote, provide easily accessible testbeds to provide such experience. Strict oversight of such educational activities is required if network performance is not to be degraded or user-privacy violated. This can be done by restricting exploration to defined subnets and to lower levels of the TCP/IP protocol stack.

One may question whether the insights gained from information systems of the sort suggested here are extendible to other types of physical systems. It will be important for engineering departments to decide this for themselves, perhaps after experience in using the local information network as a student testbed. This proposal is intended to be of general utility, and not simply a part of computer science curricula. Using complex information systems as a testbed can, we believe, provide an important link between burgeoning information science and technology and other parts of mainstream engineering practice that are becoming increasingly dependent on it.

REFERENCES

1. Charles Perrow, *Normal Accidents: Living with High Risk Technologies*, Basic Books, 1984; Scott D. Sagan, *The Limits of Safety: Organizations, Accidents, and Nuclear Weapons*, Princeton University Press, 1993.
2. *Critical Foundations: Protecting America's Infrastructures*, Report of the President's Commission on Critical Infrastructure Protection, Washington DC, October 1997. See: http://www.pccip.gov
3. S. J. Lukasik, Systems, systems of systems, and the education of engineers, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **12** (1998) pp. 55–60.

Stephen J. Lukasik received a BS in physics from Rensselaer Polytechnic Institute and an MS and Ph.D. in physics from the Massachusetts Institute of Technology. His early research work at Stevens Institute of Technology was devoted to the physics of fluids and plasmas. While a member of the Department of Defense Advanced Research Projects Agency (ARPA), he was responsible for research in support of nuclear test ban negotiations and subsequently served from 1967–1974 as Deputy Director and Director of the Agency. Later government service was as Chief Scientist of the Federal Communications Commission, 1979–1982, where he was responsible for advising the Commission on technical issues in communication regulation and for the management of non-government use of the electromagnetic spectrum. He has taught physics and engineering at Stevens Institute of Technology, technology policy at the RAND Graduate Institute, and in the Master of Science in Technology Management program at the Pepperdine University School of Business and Management. He currently holds appointments as Visiting Scholar at the Stanford University Center for International Security and Cooperation, and as Visiting Professor of International Affairs at the Sam Nunn School of International Affairs, Georgia Institute of Technology, where his research is directed to technical and policy issues related to critical infrastructure protection. In addition to academic and government positions, Dr Lukasik has extensive business and management experience in industry. This includes positions as Vice President and Manager of the Systems Development Division at the Xerox Corporation, Vice President for National Security Research and Chief Scientist at the RAND Corporation, Vice President and Manager of the Northrop Research and Technology Center, Corporate Vice President for Technology at Northrop, and Vice President for Technology at the TRW Space and Defense Sector. He has served on numerous advisory committees of the federal government, of several universities, and of the National Research Council. He was a founder of the Software Productivity Consortium and served as the Chairman of its Board of Directors in 1988. Dr Lukasik was awarded the Department of Defense Distinguished Service Medal in 1973 and 1974, and a D. Eng. (Hon.) from Stevens Institute of Technology. He is a founder of *The Information Society: An International Journal*, a consultant to SAIC, a technical advisor to serveral start-up companies engaged in applying information technologies to the entertainment industry, and a member of the Board of Trustees of Harvey Mudd College.

Michael A. Erlinger is a professor in the computer science department at Harvey Mudd College. Mike devotes his non-teaching time to research in high speed networking and to creating a computer science program Currently Mike is the co-chair of the IETF Intrusion Detection Working Group which is developing protocols for the communication of intrusion information. Previously he was the chair of the IETF Remote Network Monitoring Working Group, which developed the SNMP based RMON MIB (RFC 1271) which has gained gain wide marketplace acceptance. Today, Mike continues his hands-on approach to network management while researching network security issues as a member of a joint Aerospace Corp. and UD Davis DARPA project team.