

Adaptive Active Network Control and Management System (AANCMS)

Morteza MOSAVI* and Mahdi JALILI-KHARAAJOO**

Islamic Azad University of Arak Branch

Young Researchers Club, Islamic Azad University, Tehran, Iran

P.O. Box: 14395/1355, Tehran, Iran

Abstract. In this paper, the general architecture of adaptive control and management in active networks is presented. The proposed Adaptive Active Network Control and Management System (AANCMS) merges technology from network management and distributed simulation to provide a unified paradigm for assessing, controlling and designing active networks. AANCMS introduces a framework to assist in managing the substantial complexities of software reuse and scalability in active network environments. Specifically, AANCMS provides an extensible approach to the dynamic integration, management, and runtime assessment of various network protocols in live network operations.

Keywords: Active networks, network management, control theory, network control.

1. Introduction

Active Networking (AN) is an emerging field which leverages the decreasing cost of processing and memory to add intelligence in network nodes (routers and switches) to provide enhanced services within the network [1,2]. The discipline of active networking can be divided into two sub- fields: Strong and Moderate AN. In Strong AN, users inject program carrying *capsules* into the network to be executed in the switches and routers. In Moderate AN, network provides provision code into the routers to be executed as needed. This code can provide new network based services, such as active caching and congestion control, serve as a mechanism for rapidly deploying new protocol versions, and provide a mechanism to monitor, control, and manage networks [3]. Note, however, that a provider can provision an interpreter, which executes user code, providing a Strong AN service in a Moderate AN context. The OpenArch and OpenSig forums aim to provide, respectively, open network architectures and signaling interfaces, which are necessary but not sufficient to provide AN. Active Networking is related to IN (intelligent networking), which provides intelligence and service creation mechanisms in the PSTN (public switched telephone network) [4].

The most significant trends in network architecture design are being driven by the emerging needs for

global mobility, virtual networking, and active network technology. The key property common to all these efforts is *adaptability*: adaptability to redeploy network assets, to rewrite communication rules, and to make dynamic insertion of new network services¹ a natural element in network operations. Critical to the deployment and management of these future networks is the need to provide consistency and control over dynamic changes, and to limit the impact that such changes have on performance and stability, as required for robust communication. Adaptive computing environments could benefit greatly from several ongoing research efforts. Active network research [5-11], in particular, seeks to pursue this concept of adaptive computing by providing network protocols that are more flexible and extensible. Active networking is motivated by the notion that the improvement and evolution of current networking software is greatly hindered by slow and expensive standardization processes. Active networking tries to accommodate changes to network software by facilitating the safe and efficient dynamic reconfiguration of the network. Adaptive computing environments may be seen as the composition of the

¹ In this paper, the term *network service* refers to a resource made available through the network that provides a well-defined interface for its utilization.

two main orthogonal approaches to active network design discussed in [12]:

Discrete Approach: Administrators issue explicit commands that load, modify, or remove networking software. With this approach a network is active in the sense that it can be dynamically changed administratively.

Integrated Approach: The network is modified by the data packets that travel through it. When packets travel through the network, they automatically cause required software resources to be loaded on demand.

In this paper, an Adaptive Active Network Control and Management System (AANCMS) is proposed. Our architecture is designed to actively control, monitor, and manage both conventional and active networks, and be incrementally deployed in existing networks. The AANCMS is focused on an active monitoring and control infrastructure that can be used to manage networks. The rest of the paper is organized as follows: in Section 2 we explain the architecture of an active network node. Section 3 presents the basic structure of AANCMS. Some comments on distributed simulation are made in Section 4. Section 5 presents some possible future works on this subject. Finally, the paper is concluded in Section 6.

2. Active network node architecture

Active networking technology signals the departure from the traditional store-and-forward model of network operation to a store-compute-and-forward mode. In traditional packet switched networks, such as the Internet, packets consist of a header and data. The header contains information such as source and destination address that is used to forward the packet to the next element that is closer to the destination. The packet format is standardized and processing is limited to looking up the destination address in the routing tables and copying the packet to the appropriate network port. In active networks, packets consist not only of header and data but also of code. This code is executed on the active network element upon packet arrival. Code can be as simple as an instruction to re-send the packet to the next network element toward its destination, or perform some computation and return the result to the origination node. Additionally, it is possible for these packets to install code whose lifetime exceeds the time that is needed for the active packet to be processed. Software modules that are installed in this fashion are called active extensions. Active extensions facilitate for software upgrades, new protocol implementations, system and network monitoring agents. Other

potential applications that need control functionality to be installed on demand are also made possible. This is a major breakthrough compared to the current situation where network elements come with a set of configurable, yet pre-installed options at the time the element is shipped. To install new functions, one has to bring the infrastructure off-line to manipulate its functionality.

Apart from obvious practical advantages such as those described above, there are several properties which make active networks attractive for the future of global networking as a form of agreement on network operation for interactions between components that are logically or physically distributed among the network elements. A number of reasons have been contributing to a very long standardization cycle, as observed in the activities of the Internet Engineering Task Force (IETF). Most importantly, the high cost of deploying a new function in the infrastructure, required extreme care and experimentation before the whole community would to agree that a standardized protocol or algorithm is good enough. Diversity, competition and other conflict creating conditions also contribute to severe delays in standardization and thus deployment of new services. In active networks, functionality can be deployed in the infrastructure dynamically, and can be easily removed or replaced. This offers more flexibility in deploying early implementations (without having to stall on the standardization process), protocol bridged (that translates between different revisions/generations of a service as in the active bridge [13]), and most importantly, services themselves: users are free to customize the network infrastructure to fit their needs, when such needs emerge. This means that network operation forms the low layers of the architecture up to the application layer, could be dynamically customized to provide CPU and packet scheduling to suit application needs. Value-added services can be installed on network elements by appropriately authorized users to create a true open-service market. From a business perspective, one couple even think of scenarios where a network operator might want to rent out a router or contract another organization to manage the network. Secondly, the "end-to-end argument", one of the Internet's key design principles, being attributed both to its success and some of the most serious problems, is being questioned by the active networking paradigm. In its original manifestation, the idea was to have the least possible functionality (and thus complexity) within the network and push the intelligence to the end-systems.

The key component enabling active networking is the *active node*, which is a router or switch containing the capabilities to perform active network processing.

The architecture of an active node is shown in Fig. 1, based on the DARPA active node reference architecture [14].

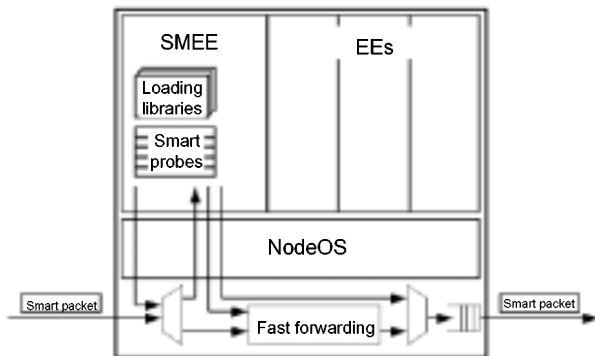


Fig. 1. Architecture of an Active Node

The conventional node hardware and software provide the basic forwarding, routing, and signaling capability of the node. Packets typically pass through the normal forwarding fast path of the node. Packet filters detect *smart packets* that need active processing, which are demultiplexed and sent up the active path. The NodeOS provides operating system services for the active node. Execution Environments (EEs) provide the language and execution environment for active code. Active Applications (AAs) execute to provide active services. The Smart Environment for Network Control, Monitoring and Management (SENCOMM) is an EE that provides an environment for the execution, monitoring, and control of AAs for node management (MAAs), which is shown by SMEE in the figure. MAAs consist of *smart probes* and *loadable libraries*, which will be discussed in detail in Section III. For now, it is sufficient to note that loadable libraries consist of code and data that is likely to be shared and reused among multiple MAAs and smart probes are monitoring and control programs dynamically installed by smart packets.

The standard node reference model does not capture the details of high performance router and switch platforms, which have distributed hardware architecture rather than a single central processor and memory. In particular, the forwarding function is implemented in hardware as a switch fabric, with packet processing distributed across the input and output interfaces. Thus, the location of active service functionality and implementation of active data paths is more complex, and must be distributed among the *per port* processing functions [15,16]. Monitoring and control MAAs, however, will be located in the control processor of a high performance router, even though the packet filters are distributed.

3. AANCMS structure

In addition to work in the active network community, new standards are being proposed to assist in the distribution and maintenance of end-user applications [17-19]. These standards attempt to introduce more timely and cost-effective mechanisms for distributing and maintaining application software via the network, allowing users to install or update software components by simply accessing HTML-like pages. However, extending such mechanisms to include the deployment and maintenance of system-level software is more difficult. The addition of system-level networking software must be done carefully to avoid potentially costly mistakes, and must also be properly coordinated with the management infrastructure if such changes are to be properly monitored and controlled.

While the trend toward adaptable protocol and application-layer technologies continues, the control and assessment of such mechanisms leaves open broader questions. Future networks could greatly benefit from simulation services that would allow network engineers to experiment with new network technologies on live network operations, without compromising service. Live traffic-based simulation services would provide engineers insight into how a proposed alteration would affect a network, without committing the network to potentially disruptive consequences.

Finally, the management of adaptive networks would greatly benefit from sophisticated monitoring tools to help assess the effects of runtime alterations and detect when those effects result in activity outside a boundary of desired operation. AANCMS is intended to streamline and, at the same time, enrich the management and monitoring of active networks, while adding new support to the network management paradigm to assist network designers. The AANCMS is pursuing a unified paradigm for managing change in active network computing environments. Underlying this framework is a conceptual model for how elements of technology from network management, distributed simulation, and active network research can be combined under a single integrated environment. This conceptual model is illustrated Fig. 2.

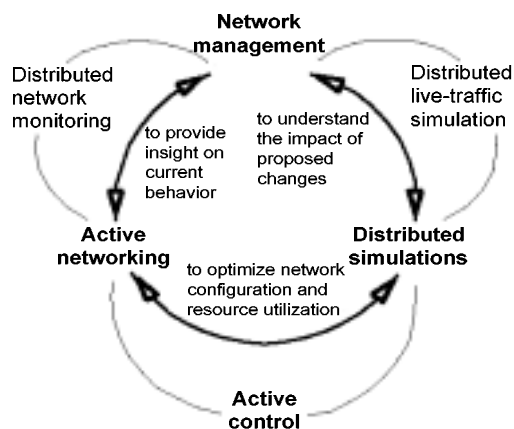


Fig. 2. Conceptual Framework of AANCMS

AANCMS gains from *discrete* active networking the ability to dynamically deploy engineering, management, and data transport services at runtime. AANCMS leverages this capability with network management technology to (1) integrate network and system management with legacy standards (SNMP, CMIP) to provide a more flexible and scalable management framework, (2) dynamically deploy mechanisms to collect network statistics to be used as input to network engineering tools and higher-level assessment tools, and (3) assist network operators in reacting to significant changes in the network.

AANCMS targets an active network environment, where powerful design and assessment capabilities are required to coordinate the high degree of dynamism in the configuration and availability of services and protocols. To this end, we have formulated architecture of a network management and engineering system that, while inheriting some components from current NM technology, introduces distributed simulation as an additional tool for design and performance assessment. Some components of the AANCMS architecture map very well to already existing technology. Recognizing this, the architecture has been explicitly designed to accommodate other network management engineering solutions. The AANCMS architecture is divided into data, assessment, and control layers. Fig. 3 shows how the data and information flow through the layers. The data layer operates at the data packet level and offers a set of services for the manipulation of network data. The assessment layer performs analytical reviews of network behavior to extract relevant semantic information from it. The control layer performs higher-order functions based on expert knowledge.

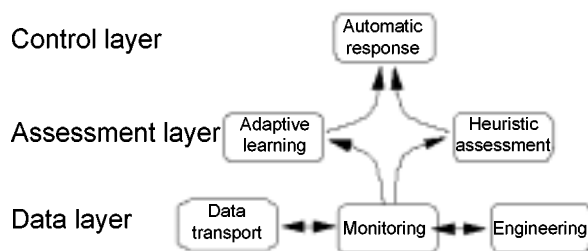


Fig. 3. AANCMS architecture

The AANCMS architecture has been designed to reuse and integrate software components derived from significant advances in network alarm correlation, fault identification, and distributed intrusion detection. In particular, the assessment and control layers of the AANCMS architecture perform tasks analogous to alarm correlation and fault analysis of the types currently proposed by network management expert systems [20-28]. All the components constituting these logical layers may be independently deployed and configured on machines throughout the network using common system management support. The implementation of each of these logical layers may use (1) existing non-active technology properly fitted to be dynamically deployed (thus implementing the discrete active networking approach) or (2) new active networking technology. AANCMS may distribute data-layer services on machines across domains,² but deploys assessment and control layer services in machines within the domain they manage. Depending on the amount of resource sharing resulting from the deployment of active networking services, the assessment layer may also be distributed across machines in multiple domains. Because the control layer must possess a significant amount of authority to perform changes in the network, it should be deployed only within a single domain. Several control services may then cooperate at the inter-domain level to exchange information for making better control decisions about their respective domains.³ The following sections describe the data layer, which embodies the most innovative features of our architecture. The assessment and control layer will not be further discussed in this paper. The foundation of the AANCMS architecture is the data layer, which is composed of engineering, monitoring, and data transport services. Although presented as a single layer, it is useful to recognize and distinguish the various modules that may populate this layer. For this reason, we decompose the data layer into three

² In this context, a domain consists of a collection of software and hardware objects managed by a single administrative authority.

³ A discussion about inter-domain information exchange between control services is beyond the scope of this paper.

distinct data service types, all of which may benefit from dynamic deployment in the network.

3.1. Data transport services

Data transport services offer communication protocols that are either quite general or extensible as proposed by active networking concepts (EEs), or that support more traditional services derived from those available today. Although data transport services are an integral part of the AANCMS architecture, the research does not focus on the development of new transports. Rather, the architecture is tailored to accommodate both new and legacy data transport services. In all cases, we assume that data transport services may be dynamically deployed. The scope of this paper does not permit us to give a detailed description of these services and their possible applications. We instead focus on the monitoring and engineering services.

3.2. Monitoring

Monitoring services perform tasks analogous to those of today's network management agents. In general, monitoring services record the operation of other network services, perform analytical reviews of the network traffic (either directly or through the review of data collected by legacy monitoring agents), and report relevant information to higher-layer components in the AANCMS architecture. In [29] the architecture and specific composition mechanisms that allow the configuration and management of such functionality are discussed. Using such mechanisms a variety of legacy SNMP- or CMIP-based agents, such as RMON, may be directly integrated into the AANCMS framework. In addition, specialized network monitoring services may be dynamically deployed to perform user-defined targeted analyses. The use of active networking to dispatch user definable monitoring capabilities gives AANCMS two major advantages: (1) it permits selective monitoring of particular phenomena, such as new network requirements and new usage patterns that emerge over time, and (2) it improves monitoring scalability (as suggested in [30,31]) through various degrees of sophistication in the monitoring agents, thus allowing, as suggested in [32], a fluid tradeoff in the amount of computation performed in the services distributed throughout the network and the amount of computation performed at the management station.

3.3. Engineering services

Engineering services aim in the design and testing of network services before their deployment. In practice, engineering services may mimic the behavior and performance of all other network services, but differ in the following ways: (1) they live in a separate

address space and are for the exclusive use of the network designers, (2) they execute protocols in a virtual time scale that may differ from physical time,⁴ and (3) in general, they generate synthetic network traffic that does not contain user data.

AANCMS incorporates a distributed simulation engineering service to help operators explore and select the optimal deployment and configuration of network assets. Like monitoring services, the AANCMS simulation component can be distributed to key traffic arteries of the network, and can perform high-fidelity protocol and network component simulations based on the content of selective live traffic. The results of these simulations can be used to better understand the affects of various alterations of network behavior (which may become commonplace under the active network paradigm), before committing the network to these alterations. In the next section, we will present an overview of the AANCMS distributed simulation services.

4. Distributed simulation

Adaptable and configurable networks will require code repositories to store and retrieve deployable applications. This idea has already appeared in several network management designs where deployable monitors can be dynamically inserted to key points in a network. Under AANCMS we are reusing and extending these concepts in the development of generic and reusable simulation models, which are deliverable as part of an AANCMS simulation service. In particular, we are developing simulation models that allow network engineers to compose and design experiments dynamically, which may then use traffic models derived from network traffic observed from spatially distributed points in a network. The traffic models may be (more traditionally) derived at the NM station and then re-exported to the simulation nodes or derived in the network itself through a distributed modeling approach (i.e. deploy a specialized monitoring application that creates and feeds the models to the network engineering services). The following briefly summarizes the benefits of extending simulation into the network management framework, and how issues of resource utilization can be controlled and balanced against the fidelity of simulation results.

4.1. Distributed simulation and network management

⁴ The passage of time is explicitly controlled by predefined time synchronization algorithms.

As with any distributed application, network simulation experiments will require some form of remote management. In our paradigm, the network management infrastructure and protocols is reused for this task. This work differs significantly from current simulation work in that the current model of simulation typically involves the generation of synthetic workloads derived from statistical models. Inevitably, these high-level models of network traffic fail to capture important phenomena of the real load experienced by the real network. By including simulation in the network management infrastructure, it is possible to feed real workloads to the simulation system and thus greatly improve its fidelity. In some cases the workload abstracted from monitoring agents may be directly piped to the simulation, while in other cases this may not be possible because of differences in time-scale. In either case, network management tools can be used to first define the workload parameters required by the simulation and then feed collected data to the design experiments.

4.2. Controlling resource utilization

Simulation is typically used to predict the performance of a design by abstracting the behavior and performance of the design. The abstraction is usually performed in a way that optimizes the use of computing and communication resources to address very specific design problems. Distributed simulation offers a way to divide the execution of the software across a network to exploit the model's parallelism. The amount of abstraction used in the simulated models has a huge impact on the amount of resources required to simulate a given system. For example, if one wanted to study the performance of a particular transport protocol, at one extreme one could simulate the protocol by producing the actual packets of the protocol and transmitting actual simulated payload between the simulated hosts, or at the other extreme only transmit high-level digests of the packets to abstract connections or the number of bytes to transfer in each session.

4.2.1. Balancing fidelity with resource availability

We are currently focused on high-fidelity protocol development and prototyping. This kind of engineering service may require a substantial amount of communication and computing resources to be effective. This use of engineering services should either be relegated to dedicated portions of the network so as not to interfere with normal operations, or should be executed slower than as fast as possible (AFAP) to limit the amount of resources used. It is important to realize that even when the simulation is relegated to specific designated areas of the network (for example, a LAN of low-cost PCs), the

engineering support can still benefit from being part of an integrated system like AANCMS.

4.2.2. Non-AFAP simulations

Most simulators run as fast as possible to accommodate the designers' needs. In some situations, however, as fast as possible may not be the best solution or may not be desirable. In some cases, simulation, although it could execute faster than the target system (i.e. the simulated time advances faster than the physical time) must be slowed down to the speed of physical time to allow humans to interact with it (for example, flight simulators). In some other cases, as in network engineering, the simulation may be slowed to prevent consuming too much of the computing and communication resources. The degree of slowdown is intimately tied to the amount of the resources one wants to dedicate to simulation relative to other network functions and should therefore be set accordingly.

4.2.3. Software emulation versus simulation

Emulation reproduces the behavior of a design by substituting some of the components of the system that may not be available or enable a better development environment. Emulation only tests the behavioral semantic of a system but does not provide (in most cases) metrics that are related to time. For example, emulating a transport protocol may help in the development and prototyping of the finite state machinery but may not allow the measurement of how long each of the operations may take or the quantitative effects of contention on the transmission lines. Simulation, on the other hand, has the notion of virtual time and can be used to estimate the time at which different operations take place in the design. Simulation can therefore be used to perform detailed quantitative analysis of a design's performance. Distributed simulation is typically much more expensive than distributed emulation because it must keep track of a global notion of virtual time among the simulating entities. This resource requirement gap between emulation and simulation can therefore be exploited to tailor the amount of resources dedicated to network engineering. The aim of this paper is not providing the prototype of the proposed method. Indeed, the paper is focused on the theoretical points of adaptive control and management in active networks.

5. Future works

As a future work, a real implementation of AANCMS on a real prototype can be considered. Also, AANCMS's engineering support services should be

described and some synchronization protocols for the high-fidelity distributed simulation of TCP/IP software should be evaluated.

6. Conclusion

In this paper, an adaptive control and management system for active networks (termed AANCMS) was proposed. The AANCMS is focused on an active monitoring and control infrastructure that can be used to manage networks. As the dynamic deployment of network services becomes standard technology to support user applications, network operators will require an efficient and flexible infrastructure to assist them in network design, configuration, and monitoring. The quality of future network management, monitoring, and engineering tools and standards will be crucial in determining the speed at which networking will evolve toward a more dynamic architecture. In AANCMS, network monitoring, control, and design can coexist in an integrated paradigm. The synergy of combining distributed simulation, network monitoring, and active networking will dramatically increase the power of network management and engineering. Network-wide management is a significantly more difficult problem than node management. While we believe that we have provided useful tools to be used in the management community, it is important to understand that network management in general, and active network management in particular, is a long way from being a solved problem.

7. Reference

- [1] D. L. Tennenhouse and D. J. Wetherall. Towards and active network architecture. *ACM Computer Communication Review*, 26(2):5–18, Apr. 1996.
- [2] K. L. Calvert, S. Bhattacharjee, E. Zegura, and J. P. Sterbenz. Directions in active networks. *IEEE Communications Magazine*, 36(10), Oct. 1998.
- [3] A. W. Jackson, J.P.G. Sterbenz, M. N. Condell, R. R. Hain. Active network monitoring and control, Proc. DARPA Active Networks Conference and Exposition (DANCE.02), 2002.
- [4] A.V. Vasilakos, K.G. Anagnostakis, W. Pedrycz, Application of computational intelligence techniques in active networks, *Soft Computing* (5), pp. 264-271, 2001.
- [5] D. S. Alexander, M. Shaw, S. M. Nettles, and J. M. Smith. Active bridging. *Proceedings of the ACM SIGCOMM'97 Conference*, Cannes, France, September 1997.
- [6] J. Hartman, U. Manber, L. Peterson, and T. Proebsting. Liquid software: A new paradigm for networked systems. Technical Report 96-11, University of Arizona, 1996.
- [7] U. Legedza, D. J. Wetherall, and J. V. Guttag. Improving the performance of distributed applications using active networks. *IEEE INFOCOM'98*, 1998.
- [8] J. Smith, D. Farber, C. A. Gunter, S. Nettle, M. Segal, W. D. Sincoskie, D. Feldmeier, and S. Alexander. Switchware: Towards a 21st century network infrastructure. <http://www.cis.upenn.edu/~switchware/papers/switchware.ps>, 1997.
- [9] D. J. Wetherall, J. V. Guttag, and D. L. Tennenhouse. ANTS: A toolkit for building and dynamically deploying network protocols. *Proceedings of IEEE OPENARCH'98*, 1998.
- [10] Y. Yemini and S. da Silva. Towards programmable networks. *Proceedings IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, L'Aquila, Italy, October 1996.
- [11] L. Ricciulli. Anetd: Active NETWORK Daemon. Technical Report, Computer Science Laboratory, <http://www.csl.sri.com/anchors/Anetd>, SRI International, 1998.
- [12] G. Minden, W.D. Sincoskie, J. Smith, D. Tennenhouse, D. Wetherall, "A survey of Active Network Research", *IEEE Communications*, Vol. 35, No. 1, pp 80-86, January 1997.
- [13] Alexandr D.S., Show M., Nettles S.M. and Smith J.M., Active bridging, in *Proc, 1997 ACM SIGCOMM Conference*, pp.231-237, 1997.
- [14] K. Calvert, ed. Architectural Framework for Active Networks. AN draft, AN Architecture Working Group, 1998.
- [15] D. Decasper, B. Plattner, G. Parulkar, S. Choi, J. DeHart, and T. Wolf. A scalable, high-performance active network node. *IEEE Network*, 13(1):8–19, Jan.-Feb. 1999.
- [16] J. P. Sterbenz. Intelligence in future broadband networks: Challenges and opportunities in high-speed active networking. In *Proceedings of the IEEE International Zurich Seminar on Broadband Communications (IZS 2002)*, Feb. 2002.
- [17] A. van Hoff, J. Giannandrea, M. Hapner, S. Carter, and M. Medin. The HTTP Distribution and Replication Protocol. <http://www.marimba.com/standards/drpf.html>, August 1997.
- [18] A. van Hoff, H. Partovi, and T. Thai. Specification for the Open Software Description (OSD) Format. <http://www.microsoft.com/standards/osd/>, August 1997.

- [19] S. Crane and N. Dulay and H. Fossa and J. Kramer and J. Magee and M. Sloman and K. Twidle, Configuration Management For Distributed Software Services, Integrated Network Management IV, 1995.
- [20] A. Bouloutas, A. Calo, and A. Finkel. Alarm correlation and fault identification schemes in communication networks. Technical Report RC 17967, IBM, 1992.
- [21] N. Shroff and M. Schwartz. Fault detection/identification in linear lightwave. Technical Report CU/CTR/TR 243-91-24, CTR, 1989.
- [22] C. Wang and M. Schwartz, Identification of Faulty Links in Dynamic-Routed Networks, IEEE JSAC, 1993.
- [23] S. K. Kilger and S. Yemini and Y. Yemini and D. Ohsie and S. Stolfo, A coding Approach to Event Correlation, Integrated Network Management IV, 1995
- [24] G. Jakobson and M. Weissman, Real-time telecommunication network management: extending event correlation with temporal constraints, Integrated Network Management IV, 1995.
- [25] I. Rouvellou and G. W. Hart, Automatic Alarm Correlation for Fault Identification, INFOCOM '95.
- [26] Y. Peng and J. A. Reggia, A probabilistic Causal Model for Diagnostic Problem Solving -Part {I}: Integrating Symbolic Causal Inference with Numeric Probabilistic Inference, IEEE Trans. on Systems, Man, and Cybernetics, March-April, 1987.
- [27] Y. Peng and J. A. Reggia, A probabilistic Causal Model for Diagnostic Problem Solving -Part {II}: Diagnostic Strategy, IEEE Trans. on Systems, Man, and Cybernetics, May-June, 1987.
- [28] P. A. Porras and P.G. Neumann, EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances, Proceedings of the National Information Systems Security Conference, Baltimore, MD, 1997.
- [29] L. Ricciulli, Service Configuration and Management in Adaptable Networks, DSOM, 1999.
- [30] K. Meyer, M. Erlinger, C. Sunshine, G. Goldszmidt, and Y. Yemini. Decentralized control and intelligence in network management. *Integrated Network Management IV*, 1995.
- [31] Y. Yemini, G. Goldszmidt, and S. Yemini, Network management by delegation. *Second International Symposium on Integrated Network Management*, Washington DC, April 1991.
- [32] I. Katzela, A.T. Bouloutas, and S.B. Calo. Centralized vs. distributed fault localization. *Integrated Network Management IV*, 1995.