Active Network Management and Kolmogorov Complexity

Amit B. Kulkarni and Stephen F. Bush, GE Corporate R&D, Niskayuna, NY, USA
kulkarni@crd.ge.com, bushsf@crd.ge.com

This paper seeks to describe new and better ways to represent network health and thus attempts to explore concepts other than those based on network topology-based representations of network management. It examines the manner in which active network management can benefit from Algorithmic Information Theory. Due to the new paradigm and enhanced capabilities of active networks, this work proceeds along the lines that a new perspective that incorporates Algorithmic Information Theory can provide superior, innovative solutions for network management.

Openarch 2001 – Short Paper Session “Ghosts of the Net!”

1.1 Introduction

Actives Networks [2, 5] enable the propagation of active packets, that is, network packets that carry executable code in addition to data. The executable code should be designed in a more compact form than transmission of the equivalent static, non-executable data in a piecemeal fashion. Clearly, the algorithmic nature of the active packet allows for more compression. As a simple example, a million digits of pi can be transmitted, or more compactly and simply, the description of a circle and the command to divide the circumference by the diameter. This paper seeks to describe new and better ways to represent network health and thus attempts to explore concepts other than those based on network topology-based representations of network management. It examines the manner in which active network management can benefit from Algorithmic Information Theory. Due to the new paradigm and enhanced capabilities of active networks, this work proceeds along the lines that a new perspective that incorporates Algorithmic Information Theory [3] can provide superior, innovative solutions for network management.

The proposed approach uses Kolmogorov Complexity and the science of Algorithmic Information Theory (sometimes called Complexity Theory) to build self-managed networks and vulnerability analysis techniques that draw on fundamental properties of information to identify, analyze, and correct faults as well as security vulnerabilities in an information system. [1] introduced the use of Streptichrons in the form of algorithmic information transfer in an active network. This paper proposes an approach for network management that uses complexity measures to detect and analyze problems in the network before applying self-composition techniques to remedy the problem. From an implementation perspective, these approaches will be deployed in an active network environment.

1.2 Proactive Network Management

The primary resources in a network are computing (CPU), memory, bandwidth and storage. Other critical elements of the network that contribute to the health of the network are the hardware interfaces (the network interface cards), the links between the network nodes, and the control software. Our premise for analyzing faults in the network is based on the notion that the shared resources have an "operating range" of behavior within which the network functions normally. Whenever one or more of these shared resources exceeds its normal range, the health of the network is said to be compromised. The irrational behavior of one of the resources can have a cascading effect on other resources, creating a ripple effect of problems that may mask the original problem. To quickly isolate and identify a problem, one has to understand the overall behavior of the network given different root causes. Observing the effects of root causes on the operating ranges of the shared resources enables rapid identification of a problem. Understanding emergent behavior induced by root causes enables us to design and implement remedies that recognize emergent behavior and understand the root cause and implement solutions to alleviate the problem. In fact, if sufficiently distinguishing emergent behavior exists, problems can be identified in very early stages of their manifestation.

The objective of this work is a step towards communication networks whose inherent state, or natural tendency, is optimum performance. Faults should naturally attract the entities required to eliminate faults. The proposed mechanism to accomplish the attraction of solution entities can be summarized in three steps. The first step is detecting the problem. Current approaches use reactive methods in which devices in the network notify a management station of its current status. Specialized software at the management station sifts through all the data and determines if a fault has occurred. If a fault has indeed occurred or if certain Management Information Base (MIB) [4] values appear out-of-range, an alarm is raised and the network manager is notified. This approach is reactive and cannot respond quickly to faults. On the other hand, we utilize a proactive approach in which the fault must identify itself to the entities capable of eliminating it. Notice that this does not necessarily require a human to identify or understand the fault, only that the solution components of the management application are capable of recognizing it. Once the fault is identified, the next step requires that the information be efficiently and accurately propagated to the solution entities with minimal overhead to the network. Once a fault occurs, the network may already be impaired; adding additional overhead in trying to solve the problem would only exacerbate the

problem. To re-state this step from another viewpoint, the necessary and sufficient solution entities should be attracted to the proper locations to solve the problem. The final step in this vision is that only the necessary and sufficient entities required to correct the fault arrive, and that these solution entities act in a cooperative manner to correct the fault quickly and accurately. In this paper, we describe techniques and means to achieve automated fault identification.

1.3 Complexity and Network Health

According to Complexity Theory, the complexity of a piece of information is the size of the smallest program capable of producing that piece of information. There are strong ties between Algorithmic Information Theory and Complexity. A truly random piece of information cannot be compressed, and its length is its complexity. Complexity is, in general, uncomputable. However, bounds on complexity can be derived, unfortunately page limitations do not allow us to discuss this aspect in more detail.

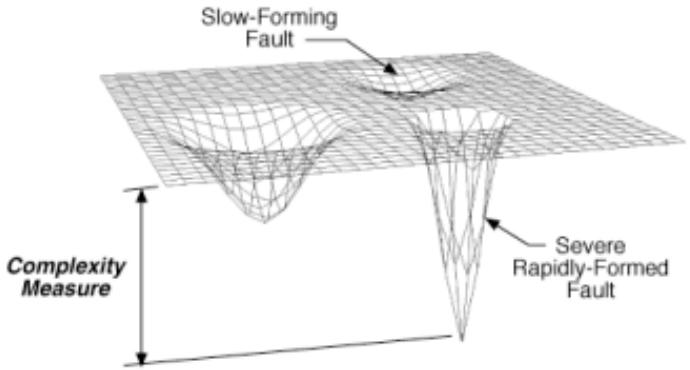

Figure 1.1: Complexity-Based View of Network Health.

Complexity-based approach to network management provides certain advantages over a topology-based representation. A complexity-based paradigm can exploit relationships between MIB variables on different branches of the MIB tree and those that are spatially scattered in the network. These relationships can be used to compute complexities of the network in different dimensions. These different dimensions could relate to availability of network services, health of devices/components/sub-components in the network, performance of user applications, vulnerability analysis, and other relevant network services and applications. Applying results from Algorithmic Information Theory, one can compute the complexities of the various dimensions. We use the assumption that greater complexity implies more randomness, which in turn implies that the system is healthy. Regions of low complexity indicate a problem that has occurred or may occur. The measure of complexity indicates the severity of the problem. Linking problem severity to complexity measures has further advantages. This enables mechanical quantification of problems occurring in the network, enabling development of sophisticated management applications that can enable self-healing within the network. Furthermore, the slope of the complexity curve can be monitored to indicate future problem occurrences. When a region of complexity in a certain dimension changes its measure from high to low, it indicates that the related network component is likely to develop a fault as illustrated in 1.1.

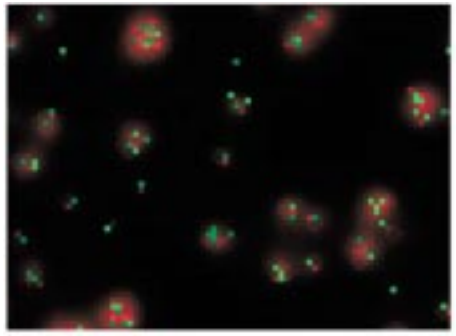

Figure 1.2: Nebulae of MIB Objects Formed from Fault (Heat).

1.4 Complexity Model and Experimental Validation

To visualize the concept of complexity-based representation of network health, imagine a space filled with entities that represent the values of various monitored objects from the managed system. For instance, each entity could be an SNMP MIB Object and its corresponding value. Initially, all entities are randomly located in that space. Furthermore, each entity can move in a random fashion in an area around its initial location. By complexity theory, a truly random sequence is incompressible; therefore, the sequence representing the location of these entities cannot be compressed.

Each entity has a normal operating range within which its value should fall during ”normal” operation. As the operating range is exceeded, heat is generated as shown in 1.2. We use the characteristics of heat diffusion provided by the Swarm library package to represent dissemination of the information about the network service or component in the network. As heat diffuses into space its energy is dissipated. Similarly, information dissipates upon the passage of time finally making it obsolete. Other entities are attracted to the heat, forming circular patterns, or clusters. This represents the formation of relationships between the different entities that are affected by the problem.

Figures 1.3 and 1.4 show the entities and the heat generated from the entities after one hundred simulated time units. The pattern formation results in a loss of randomness and enables a compressed representation of the problem. As the magnitude and number of heat clusters increases, randomness decreases and compressibility increases. The result is that more severe faults can be represented in smaller and more efficient forms for transmission. Thus, a relationship between network health and complexity is established. Figure 1.3 and Figure 1.4 show the cluster rate for a simulation with a specified probability of fault occurrence, duration, and severity.

In other words, when a fault occurs or can potentially occur, relationships between the various components are exposed due to the information transfer (causing the clusters to coagulate and become dense). This in turn enables information to be transmitted algorithmically (that is, as relationships and data) instead of long sequences of raw data.

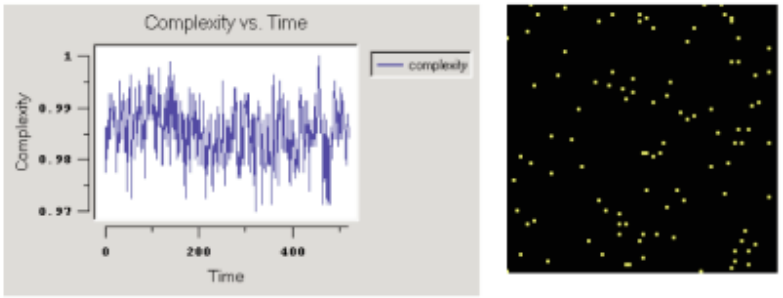

Figure 1.3: Healthy Network.

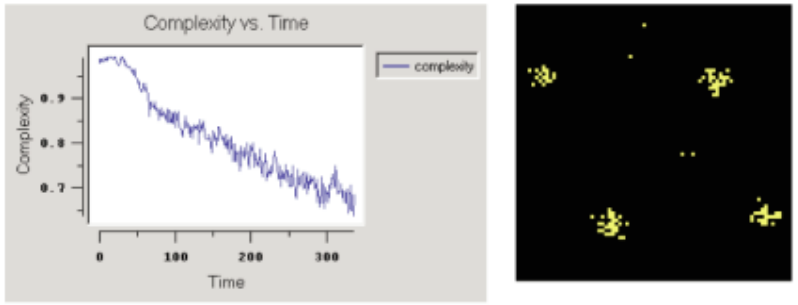

Figure 1.4: Multiple Faults.

One of the goals of this work is to pursue better complexity metrics. Measuring clustering is an attempt to measure complexity. We define the cluster measure to be one minus the proportion of the entity that is surrounded by adjacent entities. Note in 1.4 that clustering clearly occurs. Each entity in this experiment had the same probability of fault occurrence and severity.

1.5 Summary

Proactive network management is an area that has not received much attention in networking research circles. The ideas described in this paper outline a novel approach for an active network that inherently attracts solutions to net-

work problems, even before such problems occur. Proactive management will allow the creation of networks that are robust, as well as resistant to security threats such as intrusion and denial of service. This paper takes the first step in applying algorithmic information theory and techniques to network management to create the next generation of networks that are self-diagnosing, self-managing, and self-healing.

References

[1] Stephen F. Bush. Active Virtual Network Management Prediction. In *Parallel and Discrete Event Simulation Conference (PADS) '99*, May 1999.

[2] Stephen F. Bush and Amit Kulkarni. *Active Networks and Active Virtual Network Management*. Kluwer Academic/Plenum Publishers, Boston, March 2001.

[3] Ming Li and Paul Vitanyi. *Introduction to Kolmogorov Complexity and its Applications.* Springer-Verlag, August 1993.

[4] Marshall T. Rose. *The Simple Book, An Introduction to the Management of TCP/IP-Based Internets*. Prentice Hall, 1991.

[5] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden. A survey of active network research. *IEEE Communications Magazine*, 35(1):80–86, January 1997.