GNOSIS: Global Network Operations Status Information System

Jessica A. Kornblum and Jonathan M. Smith
{jkornblu,jms}@dsl.cis.upenn.edu
Distributed Systems Laboratory
Computer and Information Science Department
University of Pennsylvania

September 10, 2001

Abstract

Monitoring the global state of a network is a continuing challenge for network operators and users. It has become still harder with increases in scale and heterogeneity. Monitoring requires status information to be near real-time and displayed continuously. A particular challenge is obtaining status information for each node and constructing the global picture at a monitoring point. GNOSIS, the Global Network Operations Status Information System, achieves a global view by careful extraction and presentation of locally available node data. The GNOSIS model improves on the traditional polling model of monitoring schemes by 1.) collecting accurate data 2.) decreasing the granularity with which network applications can detect change in the network and 3.)displaying status information in near real-time.

We define the *Network Snapshot* as the basic unit of information capture and display in GNOSIS. A Network Snapshot is a visualization of locally available state collected during a common time interval. A sequence of these Network Snapshots over time represent the evolution of network state.

In this paper, we motivate the need for a network monitoring system that can detect global problems, in spite of both scale and heterogeneity. We present three design criteria, *Accuracy, Continuity and Timeliness* for a global monitoring system. Finally, we present the GNOSIS architecture and demonstrate how it better detects network problems which are currently of concern. The goal of GNOSIS is to present a stream of consistent, accurate local data in a timely manner.

1 Introduction

The complexity and scale of modern networks have increased, while the capabilities of tools for monitoring and managing them have not. This has a variety of causes, but our belief is that the local autonomy which makes the Internet scalable serves to make a scalable monitoring system difficult. In addition, the heterogenous nature of network devices provides a difficult platform for extracting a stream of consistent, accurate local data as well as representing this data.

A network is a set of *nodes* joined together by *links* over which they communicate. The primary functions of the nodes may differ as well as the link technologies that connect them. At any given time, many nodes and many links are active, moving packets for a wide variety of applications. The behavior of nodes is complex, and can affect performance of applications as well as other nodes in the network.

Network applications use networks as if they are a utility charged with moving data. Our monitoring, however, is of individual nodes and links rather than the holistic view taken by applications. *Traceroute*, for example, derives the route packets travel to a specific destination by causing TTL expiry reports to be returned from nodes on the path by iteratively incrementing the TTL. The expiry reports consist of an ICMP TIME_EXCEEDED response to the sender, indicating the address of the node. From a network-wide perspective, traceroute is used to detect where packets are being dropped along a specific path, but neither provides information about other paths, nor any causal data for the packet losses. Thus, traditional methods for monitoring networks are unacceptable for detecting and diagnosing global phenomena in the network's behavior because they do not visualize a global view. In addition, the granularity with which they represent the network is not acceptable for detecting current network problems.

The most important principle in designing a network monitoring system is the choice of its correctness criteria. We define three major criteria: Accuracy, Continuity and Timeliness, abbreviated ACT. A system which meets these criteria can then be optimized to meet scalability or overhead requirements.

The remainder of this paper is organized as follows. The next section motivates the need for global monitoring and explains the ACT criteria. Section 3 discusses previous work, and Section 4 provides an example of a traditional Network Management System to outline its approaches. We then present the GNOSIS architecture, designed to satisfy the ACT criteria while scaling to large networks in Section 5. Section 6 highlights a particular element of GNOSIS, the ***network snapshot***, while Section 7 covers some GNOSIS implementation strategies and applications. Section 8 concludes the paper.

2 Global Monitoring Principles

This section presents definitions that differentiate global monitoring from local measurement and local monitoring. It outlines desiderata for a global monitoring system, and discusses architectural issues for ACT criteria.

2.1 Local versus global properties - an example

The scale and heterogeneity of modern networks such as the Internet creates a difficult environment to monitor globally (network wide). However, with increases in scale, the need to achieve a global view becomes even more important because problems arise that are better dealt with on global-scale.

A useful analogy is that of air traffic control systems. If air traffic controllers monitored the airspace at the same level we monitor our networks, at "airplane-level", the radar systems would determine positions of each plane in the air space, but never realize their (more important) relative positions. The latter is necessary to detect a possible collision and is accomplished by constructing a global view of the airspace.

We can engineer a network monitoring system to provide a global view of the network as a real air traffic control system does, by integrating data (*e.g.*, positions and velocities for individual airplanes) together in a global coordinate system. Thus, global network monitoring systems must monitor the "network airspace" to determine how nodes and links interact and behave as a system. Monitoring a single point in the system does not provide a global perspective. We must collect a set of data from various points in the network. Data can be gathered using both active and passive monitoring techniques [32]. However, the following criteria must apply to a set of data to reflect the whole network.

2.2 Principles for Global Network Monitoring

Independent of any optimizations such as scalability and overhead requirements, three principal criteria define a global network monitoring system:

1. **Accuracy**, meaning that the data are precise, complete and consistent when treated as a set from a point in time.
2. **Continuity**, meaning that a sequence of data sets are presented as a stream, to reflect the time-varying nature of network status.
3. **Timeliness**, meaning that the sequence of data sets presented is "near" real time (while this could as well be considered an accuracy property, we prefer to separate time from other data set properties).

Reflecting their initials, we call these the *ACT criteria.*

We can use these criteria to differentiate global network monitoring from other tasks such as local network monitoring and network measurement. Local monitoring may satisfy timeliness and continuity constraints (*e.g.,* a typical use of the `ping` command) but are not easily composed into a network wide picture without (significant) off-line analysis. Network measurements might have requirements of precision, and if they are global, they may have other consistency requirements which allow them to achieve accuracy. However, since the measurements are generally intended to be used offline, they need not meet our timeliness requirement.

A global network monitoring system must present collected status information in a near real-time manner, and do so continuously. Three basic steps are necessary for a system to meet this requirement: (1) gathering status information from the network; (2) converting gathered information into a logical set; and (3) representing this logical set visually and displaying it to the system user. Depending upon the user's needs, the metrics for accuracy, continuity, and timeliness may vary. For example, if a global network monitoring tool is to be used by managers to detect and diagnose problems with a sudden onset, such as the increasingly common denial of service attacks, timeliness may be emphasized, with the system optimized accordingly. Other network problems may require metrics more suited to rare events, such as exception monitoring, rather than aggregate measures. In GNOSIS, we focus on optimizing for rapidly changing network environment.

2.3 Architectural Issues and the ACT Crieria

This section discusses how the correctness criteria of ACT can be achieved while maintaining acceptable performance.

2.3.1 Accuracy

Precision is a measure of the detail at which monitoring information could be collected. Units might include bits, bytes, packets, flows and connections, for example, or other sorts of structured events. When timestamps are used, the precision of the clock will be important. Status information is considered accurate if it deviates only within acceptable limits.

Completeness is a measure of whether we gather status information from every node in the network or from a representative subset. A monitoring system might require that each visual representation include status information from every node in the network. This might pose significant barriers to the timeliness and continuity criteria due to the latency to collect data from every node. A subset of nodes might represent a better design. The former approach may be necessary for applications that want to detect complex behaviors among nodes in the network that are configuration-determined. Thus information is needed from each node to completely understand how the nodes are interacting. Alternatively, an application requiring sampled network characteristics needs a less complete set of information.

Heterogeneity, the diversity of the nodes and links in the network, is a major challenge to gathering of precise status data, since not all devices will have equally precise clocks and equally complete data gathering capabilities. Variation in both interfaces and information available introduces additional complexity into the monitoring system. Heterogeneity is simply a fact of life, *e.g.*, specialized devices such as [9, 20, 30, 21].

To provide accurate global information, status data gathered at various points in the system (i.e. individual nodes and/or links) must be *correlated* into a whole. If this information is to be displayed as a picture, it must be correlated in some functional or qualitative way, such as being of a common type, gathered in a particular locale, or from a single point in time. Otherwise the picture is useless for detecting and diagnosing global phenomena.

2.3.2 Continuity

The *consistency* of status information is a measure of the interarrival delay of consistent data (that is, for example, time correlated). Network status information is constantly changing over time. The level of consistency determines what kinds of behavior can be detected by the monitoring system. Status information that changes infrequently, such as node IP Addresses, requires little work by the monitoring system to keep consistent. On the other hand, if the system wanted to display process load, a rapidly changing metric, a monitoring station would need to quickly gather and collate this information to achieve consistency. Consistency is closely related to the granularity with which a network is monitored (described below).

We consider the *monitoring granularity* to be a measure of the smoothness achievable in a global network monitoring system. This is limited by either the rate at which consistent samples of the system status can be constructed and displayed, or the rate of change in the system. In a real network, the former will almost certainly be the limit to system performance. In the latter case, however unlikely, a Nyquist-like sampling rate limit could optimize polling behavior. The bottom line is that a global network monitoring system must gather and display status information efficiently to be able to detect rapidly changing network behaviors.

2.3.3 Timeliness

An essential question in achieving acceptable timeliness is the *overhead* imposed by global network monitoring on the monitored system. A network monitoring system that is used to detect problems in real-time must impose a small overhead on network resources. Often, network resources are a scarce commodity when the network exibits problematic behavior. Such is the case during a denial of service (DoS) attack. A DoS attack aims to consume network resources, thus rendering the network unable to provide services. Obviously, a monitoring system geared towards detecting this kind of problem must consume a low amount of bandwidth and processing time. On the other hand, network monitoring systems that have high overhead (Complex Event Processing [22], for example) can be used while the network is under-utilized, since the processing and bandwidth required have little effect on the ultimate timeliness with which data are delivered.

An issue strongly related to overhead is the *cost of infrastructure* necessary to gather information in a timely fashion. Monitoring systems can have a severe impact on network resources. One approach to limiting this impact is to construct a dedicated network for the monitoring system. This could be done by simply dedicating a processor in each node and connecting every node by an additional link. However, the cost of adding such dedicated resources is high.

Finally, there is the important issue of *scalability*, which is the limit in the number of nodes monitored accoring to the ACT criteria. As an example, assume the system overhead is a constant factor of the number nodes in the system. A small constant would render our system more scalable than a larger constant if we were trying to achieve a monitoring system with low overhead. The required scalability is derived from the metrics required of a specific system.

3 Previous Work

In the following text, we categorize existing monitoring tools and systems into two categories: 1.) *Node and Link level*; 2.) *Network level.* For those systems that do monitoring at the network level, we describe how they fail to achieve the ACT criteria for global monitoring.

Node and Link Level *Ping* and *Traceroute* are the most widely used tools for detecting problems. Ping is used to determine host reachability. Traceroute exports the route packets take from a source to a destination. Round trip time values are given for each intermediate node along the path. Both of these tools are useful for determining node level characteristics but do not provide insight into the health of the global network because they collect data from only one point in the network. Similarly, packet sniffers used for link measurement, such as [3, 31, 24, 17], capture and analyze link characteristics. This information is not plotted into a global framework. However, it can be used to extract statistical models of traffic characteristics.

Network Level Recently, several Internet monitoring and management projects have created architectures and frameworks for aggregating data compiled from the node and link monitoring tools mentioned above to discern end-to-end performance measurements in the Internet. PingER sends a succession of ping requests to a pre-determined number of hosts, collects the RTT of each and computes the average end-to-end delay [23]. Surveyor project is similar, but synchronizes each end host to determine unidirectional delay and loss [15]. Other projects archive data generated from monitoring and measurement tools include [8, 27, 38, 26]. Data collected by these projects are correlated together and sometimes visualized, but not in near real-time.

The time spent gathering status information affects the granularity with which the network can be monitored. Active network [10] and mobile agents [34, 35] research have reduced the latency to gather status information from nodes in the network and thus improved the monitoring granularity. ABLE, SmartPackets and Distributed Management by Delegation delegate management functions to the managed network element [12, 18, 37]. However, the set of data collected does not have the accuracy criterion needed to correlate the data points into a global space. Data cannot be collect at the same point in time with any precision.

The NIMI infrastructure was designed as a control framework for large-scale data collection in a secure environment. Access control policies determine what monitoring tools are allowed to execute. The architecture separates management tools for collecting data from the important aspect of managing which tools *can* gather information in the network [32]. We are considering the NIMI architecture for control and security in the GNOSIS implementation.

HP Openview uses a more centralized approach to gathering status information. The application sends a series of SNMP requests to a set of managed nodes in the network. State is collected and visually represented to the user [13]. This method lacks the timeliness criterion. Latency for gathering status information, collating it into a global space and displaying the global space is too high to monitor rapidly changing network characteristics and problems, and collected data lacks temporal accuracy.

In the next section (4), we further motivate the need for a global monitoring system that satisfies the ACT criteria by presenting a traditional centralized polling model for network monitoring. We describe two network problems, *ICMP sweeps* and *Denial of Service* attacks and demonstrate the weakness of the traditional model for detecting them.

4 Centralized Polling Model

Current network monitoring systems attempt to gather and visualize status information at the network level, but fail because the models do not optimize for Accuracy, Continuity and Timeliness (as described in Section 2). There are three basic steps required to visually display network status

information into a global view: *gathering data, converting the data to a picture,* and ***displaying the picture.*** A common model for achieving this is called the Centralized Polling Model and is shown in Figure 1.

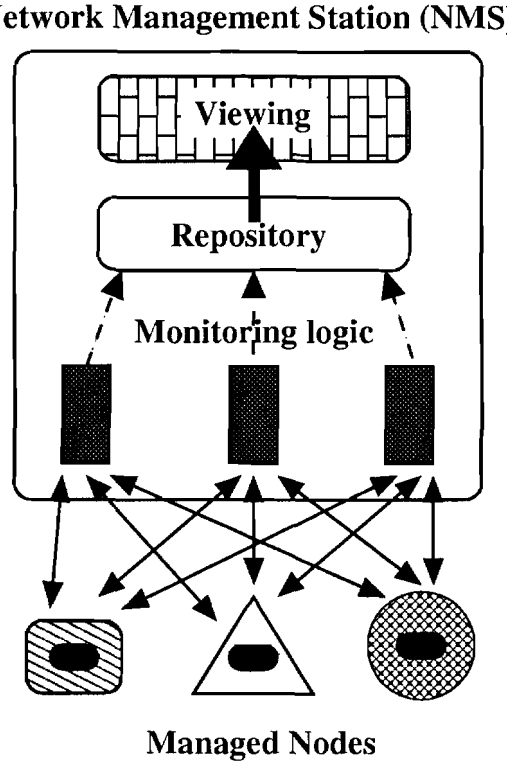

Figure 1: Centralized Polling Model

A Network Management Station (NMS) gathers status information (usually stored in a MIB [25]) by sending a series of SNMP *get* commands [6] to the node individually. Local status information is retrieved, collated and displayed to the user. HP OpenView is an example of such a system [13]. This model has a number of drawbacks when expected to detect current network problems such as *ICMP Sweeps* [14, 4, 33] and *Denial of Service* attacks [1, 36, 7].

ICMP Sweeps The Internet Control Message Protocol (ICMP) was designed to send network control information between nodes. It can be used to learn and study a target network, gaining topology and configuration information [33]. Such information is ammunition for serious network attacks [4]. For example, determining the IP address of a webserver could be used later in a denial of service attack.

During an ICMP Sweep, a series of ICMP packets are sent to various nodes in the network. Simply by using *ping* in broadcast mode, a sweep can be done with a single packet. Discerning the difference between normal ICMP traffic and Sweep traffic is impossible without some mechanism to correlate data temporally. In the Centralized Polling Model, a NMS requests ICMP packet counters from each node, detects that each packet counter, supported by MIB-II ICMP table, has incremented by 1 (from receiving the single broadcast ping packet), but cannot determine if the increase is due to a sweep. This model is missing a mechanism to correlate data into a global space.