A Scalable Framework for Wireless Network Monitoring

Camden C. Ho, Krishna N. Ramachandran, Kevin C. Almeroth, Elizabeth M. Belding-Royer

Department of Computer Science
University of California, Santa Barbara
{camdenho, krishna, almeroth, ebelding}@cs.ucsb.edu

ABSTRACT

The advent of small form-factor devices, falling hardware prices, and the promise of untethered communication is driving the prolific deployment of wireless networks. The monitoring of such networks is crucial for their robust operation. To this end, this paper presents VISUM, a scalable framework for wireless network monitoring. VISUM relies on a distributed set of agents within the network to monitor network devices and store the collected information at data repositories. VISUM's key features are its extensibility for new functionality, and its seamless support for new devices and agents in the monitoring framework. These features enable network operators to deploy, maintain, and upgrade VISUM with little effort. VISUM can also visualize collected data in the form of interactive network topology maps as well as real-time statistical graphs and reports. These visualizations provide an intuitive, up-to-date, and useful overview of a wireless network. We have implemented VISUM and used it to monitor a wireless network deployment at UC-Santa Barbara. In this paper, we describe the architecture of VISUM and report on the performance of the monitored network using information collected by VISUM.

Categories and Subject Descriptors

C.2.3 [**Computer-communication Networks**]: Network Operations—*network monitoring, network management*

General Terms

Management, Design, Measurement

1. INTRODUCTION

The marketplace is witnessing an explosive growth in wireless technology. With the advent of small form-factor devices, falling hardware prices, and the promise of untethered communication, wireless networks are undergoing prolific deployment in private homes, corporate offices, whole communities, and even entire cities.

For the robust operation of wireless networks, it is crucial that monitoring complements increasing deployment. Monitoring offers several benefits to network operators, system designers, and researchers. Monitoring can provide network operators with valuable insight into the state of the network, which can in turn increase understanding of the network's topology and usage. It can also enable operators to perform critical tasks, such as site surveying, billing/accounting, and fault detection/isolation, all necessary for the robust operation of the network. With monitoring, operators can check for compliance of system implementations with set standards. Compliance checks are important because wireless networks are typically formed by users who carry devices with heterogeneous hardware and software supplied by different vendors. System designers and researchers can use monitoring to improve protocols and systems through the analysis of collected network state. Furthermore, this state can help designers to develop realistic data traffic [4], user mobility [7], and wireless propagation models [7]. Network simulators, such as NS-2 [8] and GloMoSim [21], can then apply these models to simulate real-world network behavior more accurately [5].

The monitoring of wireless networks, however, is challenging. This is because of the rapid pace of development of wireless technologies and the short time-to-market of the developed products. The result is that wireless hardware vendors implement proprietary solutions with little standardization. To monitor networks with proprietary solutions, tools are required that are "tailored" specifically for such solutions. Consequently, operating and maintaining a set of such tools becomes cumbersome and scales poorly with increasing network size. Furthermore, the tools typically use proprietary information formats for representing collected data. This makes the correlation of monitoring information, represented in various formats, for network analysis particularly difficult.

For the monitoring of wireless networks, a framework is required that enables network operators to monitor heterogeneous devices in a manner that is generic across all devices supplied by different vendors. Moreover, this framework should scale well to cope with increasing network size yet require minimal maintenance by network operators. Our goal is to address the need for such a framework with *VISUM*. VISUM is based on a distributed architecture for monitoring wireless networks. VISUM delegates the monitoring functionality to a distributed set of agents that monitor devices and send collected information to monitoring repositories. Its generic architecture allows the seamless integration of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WMASH'04, October 1, 2004, Philadelphia, Pennsylvania, USA.
Copyright 2004 ACM 1-58113-877-6/04/0010 ...$5.00.

new devices and agents with minimal configuration. It does this by using a novel XML (eXtensible Markup Language) based framework to abstract device idiosyncrasies. In addition to the collection and storage of monitoring information, VISUM processes the stored information to create a number of interactive and graphical real-time representations of the network.

We have implemented VISUM using Java. As a result, our implementation is easily portable across various operating systems. To demonstrate its utility, we have used VISUM to monitor a wireless local area network (WLAN). This WLAN, consisting of sixteen access points, is deployed in a typical office building on the University of California, Santa Barbara campus. The information gathered using VISUM has given us valuable insight into the performance of the network.

The rest of this paper is organized as follows. In Section 2, we review work related to network monitoring. Section 3 describes the various challenges in meeting our goal of developing a generic framework for wireless network monitoring. In Section 4, we provide a detailed description of the design and the architecture of VISUM. Section 5 describes our VISUM implementation. Section 6 presents observations from our analysis of the WLAN deployment at UC Santa Barbara based on information collected by VISUM. Finally, we conclude the paper in Section 7.

2. RELATED WORK

A wide array of monitoring tools are available for wired networks. Early tools developed are *traceroute* and *ping*. Visualization extensions for these tools such as LACHESIS[15] and GTrace[13] synthesize additional information, i.e. historical and geographical data, to provide intuitive representations of collected information. Other recent tools use standardized management protocols such as the Simple Network Management Protocol (SNMP) [2] and syslog [10] to achieve sophisticated monitoring requirements. These tools come in two main flavors: tools that rely on information from within the network, such as information collected from network routers (e.g. Border Gateway Protocol state); and tools that rely on end-to-end data collection to monitor the state of the network. Examples of the former are Rocketfuel [16] and MANTRA[14]. Examples of the latter are ScriptRoute [17] and King [6]. Such tools have led to several studies that give valuable insight into the performance of deployed protocols and networks [20, 11, 12].

In the area of monitoring single-hop wireless networks, there is a general lack of tools that are easily accessible to the community. Some proprietary tools are supplied by access points vendors such as Cisco, Netgear, and Lucent. These tools are typically installed on the device itself and allow information to be accessed via SNMP or the Hypertext Transfer Protocol (HTTP). Their effectiveness, however, is generally limited by insufficient documentation and the proprietary nature of such tools. Nevertheless, numerous studies have analyzed the performance of such networks using these tools [18, 19, 3, 9, 1]. There have been efforts to study the usage and mobility patterns of wireless networks in several environments, namely university campuses [3, 9], metropolitan areas [18], and public areas [1]. Each study monitored its specific wireless network environment for a predetermined period using a collection of some of the aforementioned tools.

3. CHALLENGES

Broadly, our goal is to develop a generic framework for monitoring single-hop wireless networks. In this section we discuss some of the challenges in achieving our goal. These challenges are as follows:

- **Network Size**: Because of the rapid pace of WLAN deployments and their universal appeal, it is not uncommon for networks to consist of hundreds of access points. As examples of networks of such scale, companies such as Boingo Wireless and Wayport have deployed networks that span whole communities and even entire cities. Networks of such scale can consist of devices supplied by different hardware vendors. Because of device idiosyncrasies, monitoring such networks is challenging. Furthermore, several of the solutions typically rely on a centralized infrastructure for monitoring. This can make the monitoring techniques scale poorly.

- **Device Integration**: As networks grow in size, it is crucial that new devices are seamlessly integrated into the monitoring solution. This can be difficult because of device idiosyncrasies in different vendor products and even across various products released by a single vendor. To overcome these differences, network operators may be required to manually configure and maintain monitoring tools.

- **Information Retrieval**: Information collected from heterogeneous devices arrive in inconsistent proprietary formats, making retrieval and utilization of the accumulated information challenging. The architecture should be general enough to make the collected information easily accessible. Higher level applications should not require detailed knowledge of network components for data mining and analysis. Therefore, the method of storing accumulated information must be considered carefully, keeping in mind the need for a robust and scalable structure that allows efficient data retrieval.

- **Extensibility:** Because of the rapid pace of development in wireless networking technology, it is difficult for a monitoring solution to remain useful for monitoring newly developed network devices. It is critical for a wireless network monitoring system to be easily extensible for the purposes of collecting information from newly developed devices. Similarly, the system should provide the necessary facilities to retrieve new information that is deemed important to WLAN performance.

In the next section, we describe the design of VISUM to overcome the challenges described above.

4. VISUM DESIGN

VISUM is based on a distributed architecture for monitoring large scale wireless networks. It delegates the monitoring task to a set of agents distributed throughout the network. These agents collect monitoring information from network devices using SNMP and store the collected information at a centralized repository. In this paper, our discussion of the VISUM architecture assumes that VISUM uses a centralized repository. However, it is easily extensible to support a

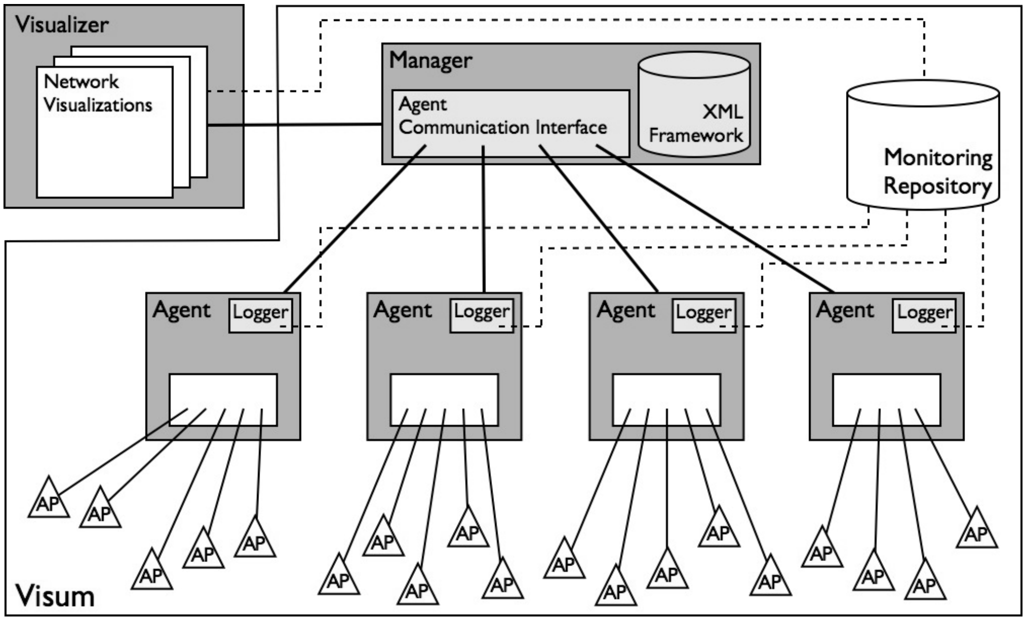


Figure 1: VISUM Architecture.

distributed set of repositories. VISUM is designed such that agents can be added seamlessly to the monitoring framework. This makes VISUM easily scale to large networks. Moreover, VISUM supports the seamless integration of new devices into its monitoring framework. Seamless integration of agents and devices is achieved using a novel XML-based framework that abstracts device idiosyncrasies.

Figure 1 illustrates a conceptual representation of the VISUM framework. The figure shows three main components in the framework. The *Manager* coordinates the operation of the distributed set of *Agents*. It also serves as a central location for the configuration and the maintenance of Agents. The *XML framework* is used to support the seamless integration of new agents and devices.

The remainder of this section describes the aforementioned components in more detail. The XML framework forms the foundation of our generic architecture. It is described first, followed by a detailed description of the remaining modules.

4.1 XML Framework

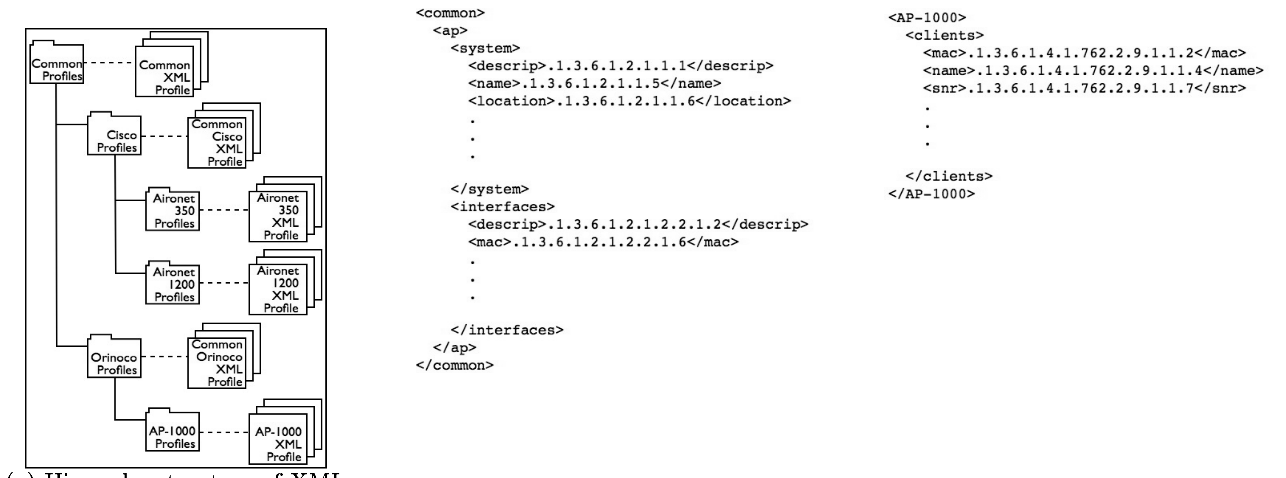
To enable VISUM to be easily extensible to a heterogeneous set of devices, we use an XML-based framework for abstracting device idiosyncrasies in device-specific XML profiles. These profiles map high-level monitoring information or *identifiers* that need to be retrieved to device-specific SNMP Object Identifiers (OIDs). By abstracting vendor-specific OIDs to vendor-independent identifiers, VISUM decouples the information retrieval process from the representation of information and its subsequent analysis. The XML profiles are organized according to generality in a hierarchical structure; profiles at the root level are more general, and profiles at lower levels becomes increasingly device-specific. Using this hierarchy structure, VISUM aggregates replicated OID mappings and enables partial data collection from network devices without specific XML profile definitions. A sample hierarchy is illustrated in Figure 2a. As examples of profiles stored in this XML hierarchy, Figures 2b and 2c show common- and device-specific XML definitions. The OID mappings defined in XML profiles are organized according to device and information type. Each mapping consists of a single OID enclosed with its corresponding high-level identifier.

4.2 Agent-Manager Interaction

The Manager is used for the configuration and maintenance of the distributed set of agents. The only initial configuration needed when an agent is first deployed is the address of its Manager. Once the agent is installed and the Manager is aware of its presence, the Manager delegates a subset of network devices that need to be monitored. In this way, the Manager enables seamless integration of additional Agents without interrupting normal operation. The Manager also maintains the XML framework described above to provide device-specific OIDs for Agents to use for data retrieval. The Manager is responsible for notifying its Agents when changes in the XML framework occur.

4.3 Agent

Agents do the actual collection of monitoring information from network devices. Figure 3 shows the conceptual architecture of an Agent. It consists of three modules to collect monitoring information: the *InfoGather*, *NodeInfo*, and *Query Scheduler*. An additional *Logger* module is used for storing the collected information at the repository. Each Agent module is briefly described below. *InfoGather:* The InfoGather module retrieves monitoring information from network devices. The data retrieval process is illustrated in Figure 4. Information is gathered in the following manner: The InfoGather module uses the device description OID (system.sysDescr) to retrieve the description of the device it is querying (Steps 1 and 2). The device description OID is a standardized OID used by all device vendors to identify their devices. The InfoGather then uses the resulting device



(a) Hierarchy structure of XML framework.

```
<common>
  <ap>
    <system>
      <descrip>.1.3.6.1.2.1.1.1</descrip>
      <name>.1.3.6.1.2.1.1.5</name>
      <location>.1.3.6.1.2.1.1.6</location>
      .
      .
      .

    </system>
    <interfaces>
      <descrip>.1.3.6.1.2.1.2.2.1.2</descrip>
      <mac>.1.3.6.1.2.1.2.2.1.6</mac>
      .
      .
      .

    </interfaces>
  </ap>
</common>
```

(b) Sample common XML profile definition.

```
<AP-1000>
  <clients>
    <mac>.1.3.6.1.4.1.762.2.9.1.1.2</mac>
    <name>.1.3.6.1.4.1.762.2.9.1.1.4</name>
    <snr>.1.3.6.1.4.1.762.2.9.1.1.7</snr>
    .
    .
    .

  </clients>
</AP-1000>
```

(c) Sample AP-specific XML profile definition.

Figure 2: VISUM XML Framework.

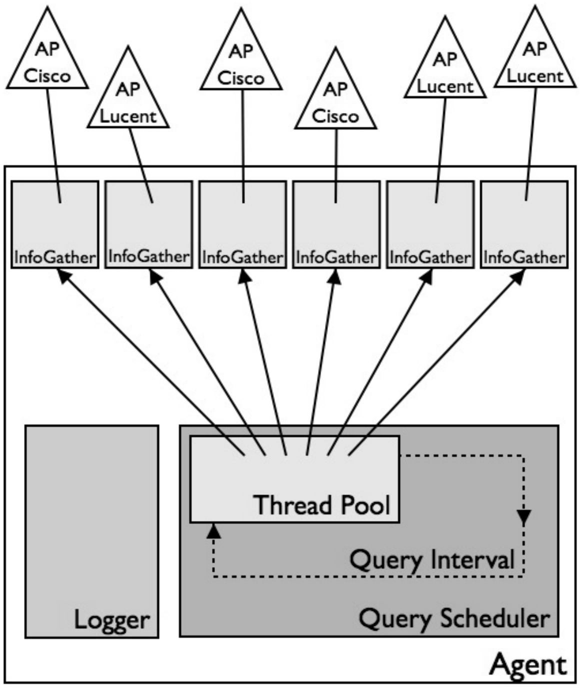


Figure 3: The Agent Architecture.

description to identify the appropriate set of XML profiles that contain the OIDs for the specific device. If the device-specific OIDs are not in the local OID lookup cache, the InfoGather module requests the device-specific OIDs from the Manager (Step 3). The manager retrieves the OIDs by traversing the profile hierarchy, first by the vendor type, then by specific model number. If a profile for the particular AP has been defined, the corresponding OIDs are retrieved and sent to the requesting Agent (Step 4). The Agent, in turn, stores the OIDs in a lookup cache to be used by the InfoGather module to monitor the AP (Step 5).

This configuration scheme has the advantage that Agents only require knowledge about each device's IP address and SNMP community string to configure itself to monitor the assigned devices. Furthermore, if the configuration of the network devices is modified, the Manager notifies the Agent of the change and the Agent adapts by retrieving the description of the new device and repeating the configuration process. *Query Scheduler:* The Query Scheduler is responsible for the scheduling of the InfoGather modules to retrieve information from network devices. To capture the mobile nature of a wireless network, it is necessary for the Query Scheduler to schedule queries to all the monitored devices in the network both concurrently and with relatively high frequency. The scheduler maintains a pool of threads used to

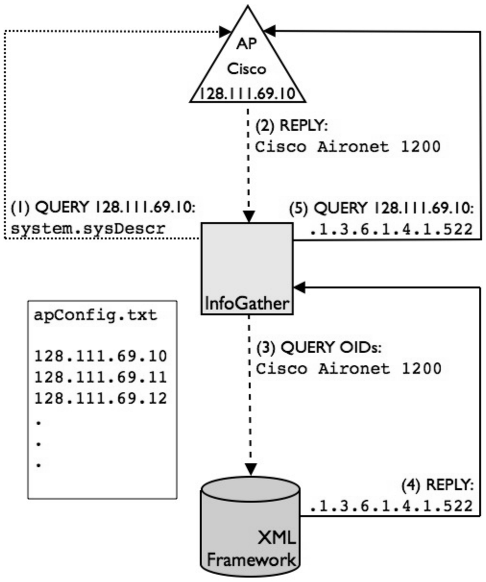


Figure 4: VISUM XML profile-based InfoGather configuration.

collect data from devices. In this way, the interval at which data is retrieved from the monitored devices is adjustable. The frequency is configured at the Manager by the network operator. There is a tradeoff between using high frequency querying, and the resulting increase in processing load on the device themselves and the resulting increase in network traffic.

NodeInfo: The data successfully retrieved by an InfoGather module is passed to the Logger module in the form of a NodeInfo object. NodeInfo is a generic representation of the data retrieved from the device. The NodeInfo object uses data types that can represent the values of various counters and gauges that are monitored in the wireless network devices. The Infogather module does the necessary type conversions so that data retrieved from the device can be represented using NodeInfo data types.

Logger: The Logger module's primary purpose is to process collected information before storing it. The Logger module exports a standard interface that can be used by the InfoGather modules. The Logger module decouples the storage of data from the interface used to pass NodeInfo objects to the Logger module. Because of this, specific Logger modules can be used for storing the monitoring information in different storage formats including flat files and databases. Logger modules can also implement custom interfaces between VISUM and external applications or visualization tools.

5. IMPLEMENTATION

This section describes our implementation of VISUM. Our goal is to demonstrate the feasibility of developing a system based on the framework we propose. Our implementation of VISUM was developed completely in Java. The goal was to make Visum portable across various platforms. We have also leveraged Java's database support to implement our LoggerDB module. Descriptions of the implemented features are organized in the three phases: data collection, logging and presentation.

5.1 Data Collection Stage

Although we implemented an InfoGather module that utilizes the SNMP protocol to retrieve information from the APs in our test network, SNMP support in APs is far from absolute. If SNMP support is not available, additional components of the InfoGather module can be "plugged in" to monitor an AP that requires a different means of data collection. The NodeInfo modules are implemented as Java objects that encapsulate queried data. Each object contains specific information about a particular wireless network node. For example, a NodeInfo object representing an AP has information about the name, location, description and uptime. Similarly, a NodeInfo object describing a WLAN mobile host contains information about its MAC address, bytes sent and received, and average signal-to-noise ratio measured from the AP to the host.

5.2 Data Logging Stage

Our implementation has two Logger modules. The first module is for logging collected data in flat files. The second is for logging the collected data into a database. Although storing logged data in flat files is simple and efficient with regards to the data logging process, it limits the potential for the system to be distributed. In this case, log files would be generated locally and would have to be later merged if each monitor is responsible for a subset of the entire network. The database Logger module is called LogDB. It supports remote database connections through the use of the Java Database Connectivity (JDBC) interface. Logging collected data into a robust database is a way to ensure that data remains intact, organized and always available for retrieval. Specifically, by storing monitoring information in a scalable and highly efficient database, VISUM can monitor networks for an extended period of time (possibly continuously given the reduced cost of storage) as well as provide up-to-date statistical graphs and reports efficiently.

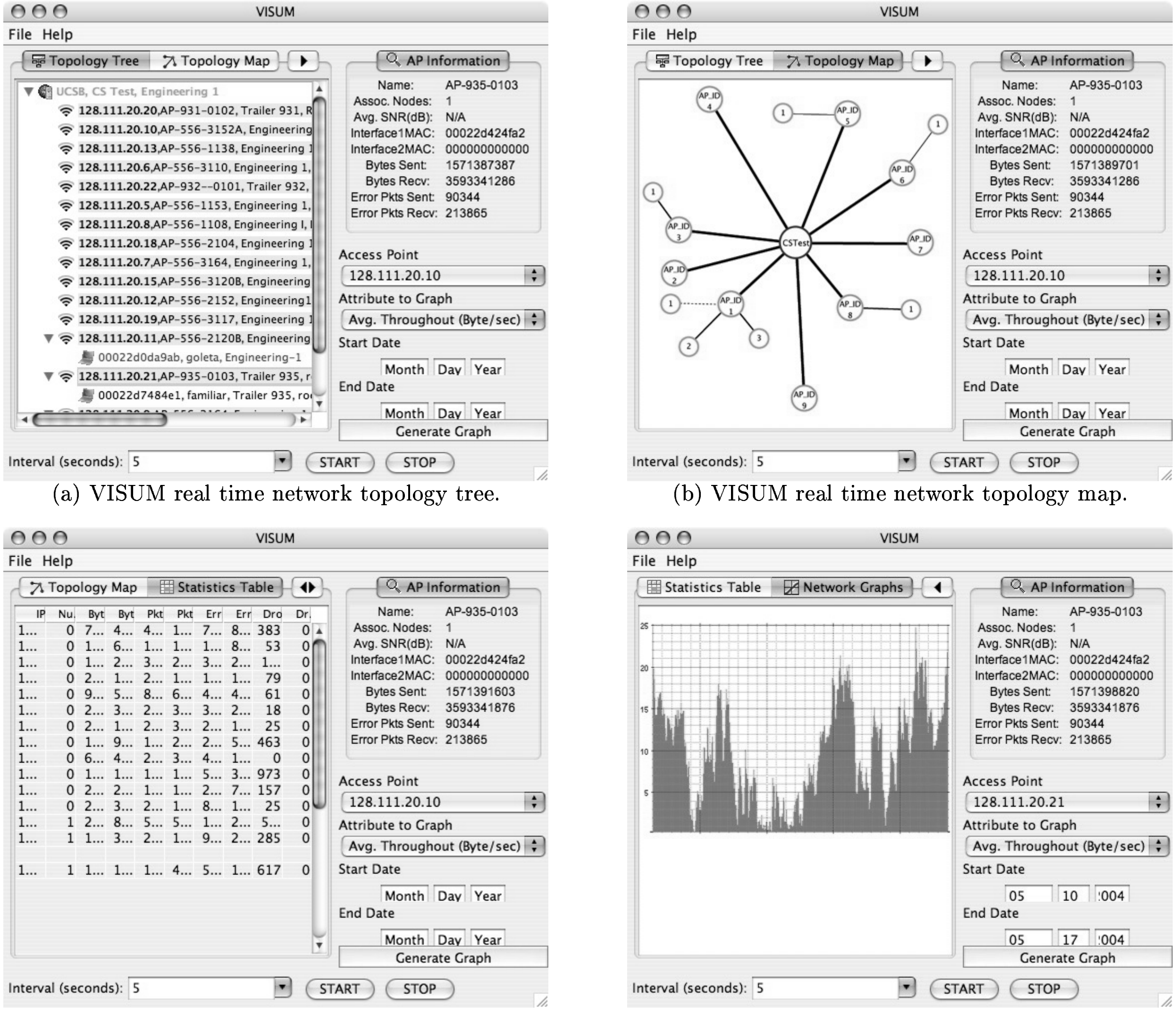
5.3 Data Presentation Stage

The VisumGUI module handles the visualization of monitored data. It allows users to control VISUM system settings and interact with the real time visualizations implemented. The VisumGUI module leverages Java's graphical functionality to provide several options for the visualization of monitored data. Figure 5 shows screen shots of the visualization modules we implemented.

The real-time network topology visualization shown in Figure 5a represents a WLAN topology using a tree hierarchy. The root of the tree corresponds to the entire WLAN network, while the children of the root node represent each of the AP's actively monitored by VISUM. By expanding the AP nodes, the mobile hosts associated with the AP can be seen. The network topology tree is updated in real time. A user is able to view the details of a particular node by clicking and selecting the node of interest. Another network topology view maps the APs of a network together with its associated nodes in a graphical representation. Figure 5b is a snapshot of this visualization. The wireless links between an AP and its associated nodes are represented using various thicknesses to represent the quality of the links. The quality is determined using the signal-to-noise ratio (SNR) of the links as measured by each AP.

Using the real-time statistical table view shown in Figure 5c, operators can observe the activity of the APs in the monitored network in detail. Each row in the table represents an AP in the network identified by its IP address. The statistics given by the table include: number of associated hosts, number of bytes and packets sent and received, number of packets in error and the number of dropped packets. This view is useful for quickly determining which APs are being heavily used and also to identify any APs that are not functioning properly.

While real-time visualizations provide current information about the state of a network, long term visualizations allow researchers and network operators to study trends and patterns that occur over a long period. These modules typically do not obtain information directly from the Agents. Instead they retrieve information from stored logs in a database. VISUM produces static graphs of overall network statistics continuously while monitoring a network for up-to-date views of network performance in the last day, week, month, and year. Figure 5d is a screenshot of the interactive graph visualization. Using interactive graphs the user can select a specific device to be graphed over any period of which monitored data is available. The graphs are generated dynamically from the data logged in the database. As such, the temporal range for which the logged data is processed



UCSB, CS Test, Engineering 1
128.111.20.20,AP-931-0102, Trailer 931, R
128.111.20.10,AP-556-3152A, Engineering
128.111.20.13,AP-556-1138, Engineering 1
128.111.20.6,AP-556-3110, Engineering 1,
128.111.20.22,AP-932--0101, Trailer 932,
128.111.20.5,AP-556-1153, Engineering 1,
128.111.20.8,AP-556-1108, Engineering I, I
128.111.20.18,AP-556-2104, Engineering 1
128.111.20.7,AP-556-3164, Engineering 1,
128.111.20.15,AP-556-3120B, Engineering
128.111.20.12,AP-556-2152, Engineering1
128.111.20.19,AP-556-3117, Engineering 1
128.111.20.11,AP-556-2120B, Engineering
00022d0da9ab, goleta, Engineering-1
128.111.20.21,AP-935-0103, Trailer 935, r
00022d7484e1, familiar, Trailer 935, ro

Name: AP-935-0103
Assoc. Nodes: 1
Avg. SNR(dB): N/A
Interface1MAC: 00022d424fa2
Interface2MAC: 000000000000
Bytes Sent: 1571387387
Bytes Recv: 3593341286
Error Pkts Sent: 90344
Error Pkts Recv: 213865

Access Point: 128.111.20.10
Attribute to Graph: Avg. Throughput (Byte/sec)
Start Date: Month Day Year
End Date: Month Day Year
Interval (seconds): 5

(a) VISUM real time network topology tree.

Name: AP-935-0103
Assoc. Nodes: 1
Avg. SNR(dB): N/A
Interface1MAC: 00022d424fa2
Interface2MAC: 000000000000
Bytes Sent: 1571389701
Bytes Recv: 3593341286
Error Pkts Sent: 90344
Error Pkts Recv: 213865

Access Point: 128.111.20.10
Attribute to Graph: Avg. Throughput (Byte/sec)
Start Date: Month Day Year
End Date: Month Day Year
Interval (seconds): 5

(b) VISUM real time network topology map.

IP	Nu.	Byt	Byt	Pkt	Pkt	Err	Err	Dro	Dr.
1...	0	7...	4...	4...	1...	7...	8...	383	0
1...	0	1...	6...	1...	1...	1...	8...	53	0
1...	0	1...	2...	3...	2...	3...	2...	1...	0
1...	0	2...	1...	2...	1...	1...	1...	79	0
1...	0	9...	5...	8...	6...	4...	4...	61	0
1...	0	2...	3...	2...	3...	3...	2...	18	0
1...	0	2...	1...	2...	3...	2...	1...	25	0
1...	0	1...	9...	1...	2...	2...	5...	463	0
1...	0	6...	4...	2...	3...	4...	1...	0	0
1...	0	1...	1...	1...	1...	5...	3...	973	0
1...	0	2...	2...	1...	1...	2...	7...	157	0
1...	0	2...	3...	2...	1...	8...	1...	25	0
1...	1	2...	8...	5...	5...	1...	2...	5...	0
1...	1	1...	3...	2...	1...	9...	2...	285	0
1...	1	1...	1...	1...	4...	5...	1...	617	0

Name: AP-935-0103
Assoc. Nodes: 1
Avg. SNR(dB): N/A
Interface1MAC: 00022d424fa2
Interface2MAC: 000000000000
Bytes Sent: 1571391603
Bytes Recv: 3593341876
Error Pkts Sent: 90344
Error Pkts Recv: 213865

Access Point: 128.111.20.10
Attribute to Graph: Avg. Throughput (Byte/sec)
Start Date: Month Day Year
End Date: Month Day Year
Interval (seconds): 5

(c) VISUM real time network statistics table.

Name: AP-935-0103
Assoc. Nodes: 1
Avg. SNR(dB): N/A
Interface1MAC: 00022d424fa2
Interface2MAC: 000000000000
Bytes Sent: 1571398820
Bytes Recv: 3593341876
Error Pkts Sent: 90344
Error Pkts Recv: 213865

Access Point: 128.111.20.21
Attribute to Graph: Avg. Throughput (Byte/sec)
Start Date: 05 10 :004
End Date: 05 17 :004
Interval (seconds): 5

(d) VISUM interactive network graphs.

Figure 5: VISUM Data Visualization

for a given graph is also flexible. This is helpful to identify trends and patterns in network usage.

Each of the views we have implemented provides the user with a different view of the monitored information accumulated by VISUM. Although we have implemented four methods of presenting collected data, extending VISUM to provide additional methods for data visualization and presentation within the VISUM framework is straightforward. This is possible because of the object-oriented nature of Java and the modular design of VISUM.

6. CASE STUDY: UCSB COMPUTER SCIENCE DEPARTMENT WLAN

In this section we present a case study describing the use of the VISUM implementation in the UCSB Computer Science department WLAN. First, we describe the wireless network and the VISUM deployment. We then discuss the results we collected using the VISUM implementation.

6.1 Network Deployment

The UCSB Computer Science Department wireless network consists of sixteen Orinico AP-1000 APs throughout a typical office building and in several laboratories and classrooms located outside the building. This network is primarily intended as a research network and therefore access to it is restricted. Consequently, the number of users and amount of traffic on the network is low.

For this study, the VISUM agent ran continuously on one machine monitoring all sixteen APs for a period of 30 days between April 29, 2004, and May 28, 2004. We used only one agent because the size of our network is still small. Information was queried from APs at an interval of fifteen seconds. The collected data was then logged into a remote mySQL server. Initially, the configuration information, which includes description, location, and interface MAC addresses, of each AP was queried. Each subsequent log entry consisted of dynamic AP statistics, such as sent and received bytes, number of packets, error packets, dropped packets, AP uptime, and number of associated nodes, including the

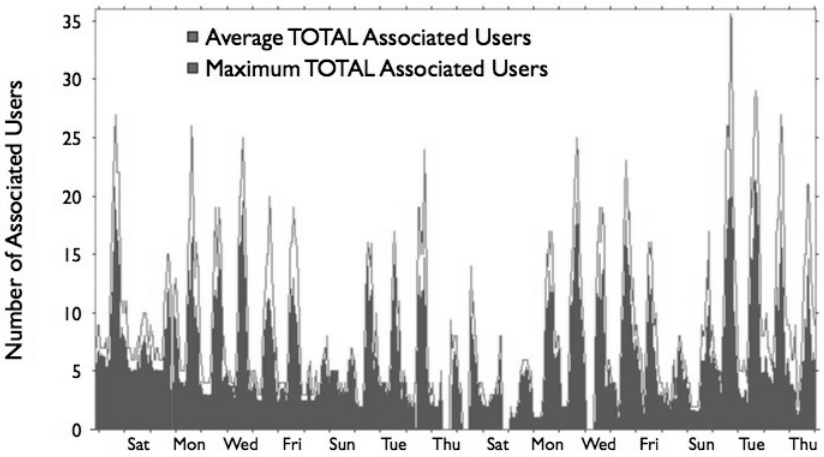


Figure 6: Average number of total associated users.

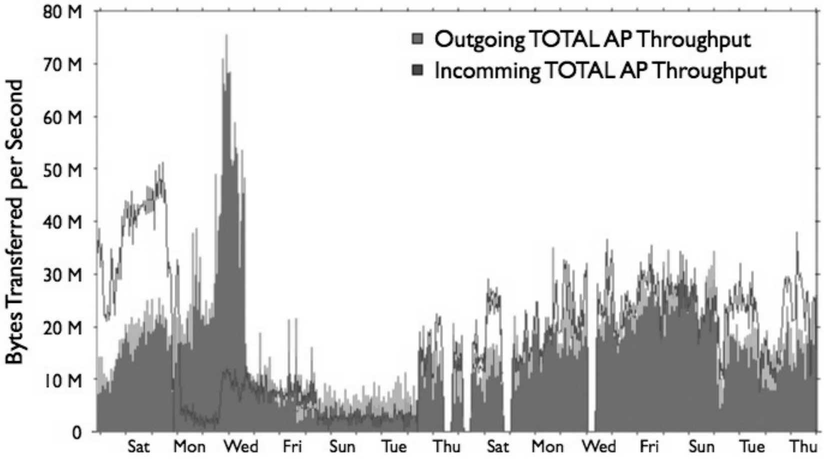


Figure 7: The average total network throughput (bytes/sec).

individual MAC addresses and traffic statistics of each associated node.

6.2 Performance Study

During the 30 day monitoring period, we recorded 237 unique MAC addresses associated at some time to the APs. Figure 6 is a graph of the average number of total associated users for the entire network over the 30 day monitoring period. Each "average total" point in the graph represents the average of 240 data samples for a period of one hour. Each "maximum total" point represents the maximum value recorded over a period of one hour. The Y-axis of the graph represents the number of associated users, and each labeled interval of the X-axis represents a duration of two days. The average total associated users for the entire network each day was only about 8 people. The graph shows that the wireless users of the network follow a weekly working schedule, in which weekdays generally see more associated users than weekends. Furthermore, daily patterns show that the number of associated users peak after 12:00 pm each day.

The average total network throughput generated by WLAN users over the 30 day monitoring period is shown in Figure 7. The Y-axis in this graph indicates the network throughput measured in bytes per second, and the X-axis is identical to Figure 6. In this study the terms *incoming* and *outgoing* are AP-centric, i.e., incoming refers to traffic received by the wireless interface of an AP, and outgoing refers to traffic sent from the wireless interface. There are four distinct gaps in the data on the 13th, 14th, 15th and 19th of May that are attributed to power outages caused by construction of nearby buildings and maintenance reboots of the VISUM server. The peculiar characteristic of the graph, however, is the distinct lack of network activity for almost four days between Saturday May 8th and Wednesday May 12th. Re-examination of the average total users graph in Figure 6 did not indicate a significant drop in total associated users during the four day period. To further investigate the unusual behavior, we examined graphs of average throughput for individual APs. We found that the average network throughput profile of one AP in particular (AP13, shown in Figure 8) matched that of the overall network. Moreover, it was evident that AP13 was disabled during the same four day period and that the absence of traffic on this AP was responsible for the drop in network activity for the entire WLAN.

To verify our initial findings we analyzed the distribution of average throughput during the monitoring period across each of the sixteen APs. The distribution is presented in Fig-

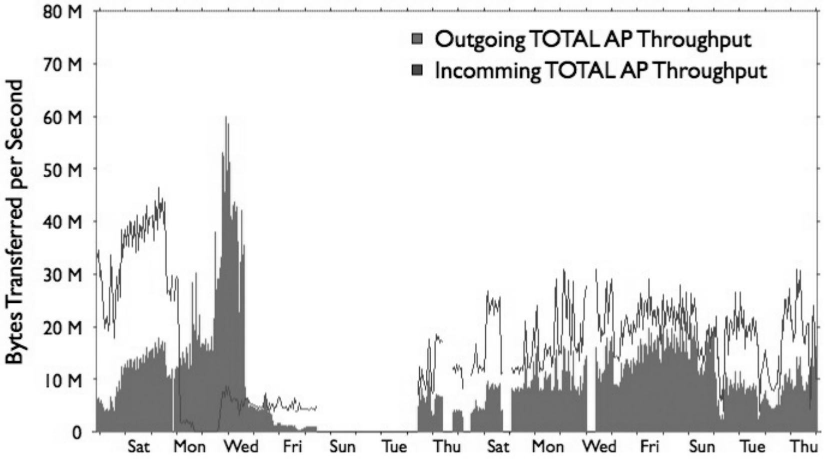
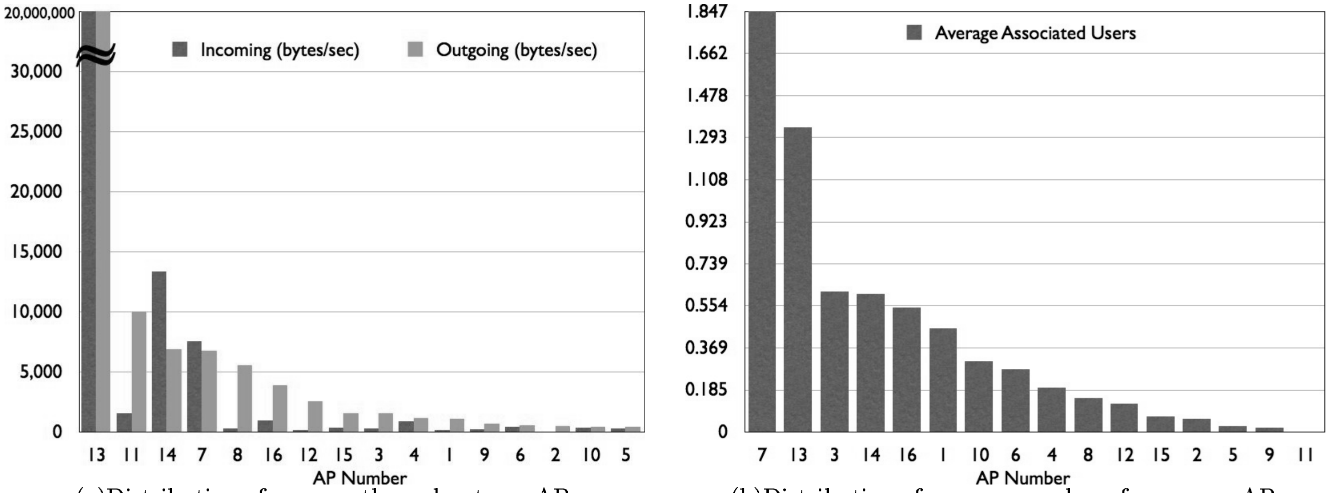


Figure 8: The average throughput for AP13 (bytes/sec).



(a)Distribution of average throughput per AP.

(b)Distribution of average number of users per AP.

Figure 9: The UCSB CS Department WLAN distribution graphs.

ure 9a. The APs are ordered by average outgoing throughput from greatest to least. The graph indicates that AP13 clearly dominates all other network traffic and supports our initial findings that the traffic trends of the overall network is defined by a single AP. Figure 9b shows the distribution of the average number of associated users per AP over 30 days. Both distribution graphs suggest that the Department wireless network is over-provisioned and utilization of the network is unevenly distributed. We believe that the observations made from analyzing VISUM monitoring results are valuable for planning future expansion of the Department's wireless network infrastructure and possible integration with other university WLAN deployments.

7. CONCLUSIONS AND FUTURE WORK

Network monitoring is a critical part of many aspects of networking, in particular, protocol development, network deployment, and network management. The advancement of network technology and the maintenance of networks would be more complex and unnecessarily inefficient without effective monitoring tools to obtain network information about its state, operation, and reaction to operating conditions. However, due to a lack of support for standardized monitoring and management protocols, heterogeneity of network components, ever-increasing network deployment, and rapid development of new network technologies, developing mechanisms to monitor networks is critical. Wireless networking is an example of a rapidly evolving network technology that needs effective monitoring for successful research, development, and deployment. Examples of practical wireless network applications that would benefit from monitoring systems are: WLAN deployment, dynamic load balancing, and network billing/accounting.

In this paper we have introduced VISUM, a framework for a distributed generic wireless network monitoring system. Using a modular architecture, VISUM collects MAC layer information from wireless network infrastructure components to provide real time views of network status. The VISUM architecture facilitates monitoring of heterogeneous wireless network components and has mechanisms to seamlessly cope with changes in network configuration. Accumulated data is logged for later analysis, or provided as real time input to a diverse set of visualization tools and high-level applications. To accommodate the monitoring of large wireless networks, VISUM's agents can be distributed among multiple hosts. Their data is easily aggregated to provide a comprehensive view of the entire network. We have demonstrated the feasibility of the architecture with an implementation developed in Java. Using the implementation we have successfully monitored the University of California, Santa Barbara Computer Science department network for a period of a month. For the presentation of monitored information, the implementation provides four distinct views of the collected information, including real time network topology and statistics tables, as well as static and interactive statistical graphs of ongoing collected data. With the capacity to monitor statistics from a varied set of wireless network components, VISUM is a flexible distributed architecture for monitoring wireless networks with diverse options for visualizing and processing the collected data.

Future work on VISUM includes a large scale deployment to monitor a more active conference environment. We also plan to develop an additional wireless network visualization that will leverage the monitoring capabilities of VISUM and incorporate geographic information of network components to provide an intuitive means to study user mobility patterns. This visualization will be useful in determining how accurately existing mobility models represent actual user behavior. Another possible extension is to provide mechanisms to facilitate management of individual wireless network components currently only monitored by VISUM. Finally, we plan to investigate techniques that can pinpoint the reason behind a certain network event, such as a congestion or an outage, through the analysis of collected information.

Ackowledgements

This work is supported by an NSF Networking Research Testbeds (NRT) grant (ANI-0335302), an NSF Infrastructure grant (EIA-0080134) and by Intel Corporation.

8. REFERENCES

[1] A. Balachandran, G. Voelker, P. Bahl, and P. Rangan. Characterizing User Behavior and Network Performance in a Public Wireless LAN. In *Proceedings of ACM Sigmetrics*, Marina Del Ray, CA, June 2002.

[2] J. Case, M. Fedor, M. Schoffstall, and J. Davin. A Simple Network Management Protocol. Internet Engineering Task Force, RFC 1067, August 1988.

[3] F. Chinchilla, M. Lindsey, and M. Papadopouli. Analysis of Wireless Information Locality and Association Patterns in a Campus. In *Proceedings of IEEE Infocom*, Hong Kong, March 2004.

[4] C. Cooper, J. Zeidler, and R. Bitmead. Modeling Dynamic Channel Allocation Algorithms in Multi-BS TDD Wireless Networks with Internet Based Traffic. In *Proceedings of IEEE Vehicular Technology Conference*, Milan, Italy, May 2004.

[5] D. Kotz and C. Newport and C. Elliott. The Mistaken Axioms of Wireless-network Research. In *Technical Report TR2003-467, Dept. of Computer Science, Darmouth College*, July 2003.

[6] P. Gummadi, S. Saroiu, and S. D. Gribble. King: Estimating Latency between Arbitrary Internet End Hosts. In *Proceedings of ACM Sigcomm Internet Measurement Workshop*, Marseille, France, November 2002.

[7] A. Jardosh, E. Belding-Royer, K. Almeroth, and S. Suri. Towards Realistic Mobility Models for Mobile Ad hoc Networks. In *Proceedings of ACM International Conference on Mobile Computing and Networking*, San Diego, CA, September 2003.

[8] K. Fall and E. Varadhan. ns notes and documentation. In *http://www-mash.cs.berkeley.edu/ns/*, 1999.

[9] D. Kotz and K. Essien. Analysis of a Campus-wide Wireless Network. In *Proceedings of ACM International Conference on Mobile Computing and Networking*, Atlanta, GA, September 2002.

[10] C. Lonvick. The BSD syslog Protocol. Internet Engineering Task Force, RFC 3164, August 2001.

[11] N. Spring and R. Mahajan and T. Anderson. Quantifying the Causes of Path Inflation. In *Proceedings of ACM Sigcomm*, Karlsruhe, Germany, August 2003.

[12] P. Rajvaidya and K. Almeroth. Analysis of Routing Characteristics in the Multicast Infrastructure. In *Proceedings of IEEE Infocom*, San Fransisco, CA, April 2003.

[13] R. Periakaruppan and E. Nemeth. GTrace: A Graphical Traceroute Tool. In *Proceedings of USENIX Large Installation System Administration*, Seattle, WA, November 1999.

[14] P. Rajvaidya, K. Almeroth, and K. Claffy. A Scalable Architecture for Monitoring and Visualizing Multicast Statistics. In *Proceedings of IFIP/IEEE International Workshop on Distributed Systems: Operations & Management*, Austin, TX, December 2000.

[15] J. Sedayao and K. Akita. LACHESIS: A Tool for Benchmarking Internet Service Providers. In *Proceedings of USENIX Large Installation System Administration*, Monterey, CA, September 1995.

[16] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Pittsburgh, PA, August 2002.

[17] N. Spring, D. Wetherall, and T. Anderson. ScriptRoute: A Facility for Distributed Internet Measurement. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, Seattle, WA, March 2003.

[18] D. Tang and M. Baker. Analysis of a Metropolitan-Area Wireless Network. In *Proceedings of ACM International Conference on Mobile Computing and Networking*, Seattle, WA, August 1999.

[19] D. Tang and M. Baker. Analysis of a Local-area Wireless Network. In *Proceedings of ACM International Conference on Mobile Computing and Networking*, Boston, MA, August 2000.

[20] V. Paxson. End-to-end Routing Behavior in the Internet. In *Proceedings of ACM Sigcomm*, Palo Alto, CA, August 1996.

[21] X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks. In *Proceedings of Workshop on Parallel and Distributed Simulations*, Banff, Canada, May 1998.