

# DATA COLLECTION: PUTTING IT ALL TOGETHER. THE SCORPION STATISTICS COLLECTION SYSTEM.

**Daniel Olivares**

dolivares@humboldt.edu  
Department of Computer Science  
Humboldt State University  
1 Harpst Street, Arcata, CA 95521

**Dr. Katia Obraczka**

katia@soe.ucsc.edu  
Department of Computer Engineering  
University of California, Santa Cruz  
1156 High Street, Santa Cruz, CA 95064

## Abstract

As the technology and needs of today's networks grow, transmission and collection of information remains a vital focus for those managing these networks. The Santa Cruz mObile Radio Platform for Indoor Outdoor Networks (SCORPION) testbed was created to test current and up-coming technology and applications to insure continuing functionality and growth as networking needs change. At the core of the testbed is the storing and collection of data. The SCORPION statistics collection system is designed to streamline the process of data storage and collection and compliment the unique testing features the testbed provides by promoting a smooth transition between testing and analysis of test data.

## Keywords

Sensor networks, testbed, networking, internet, logging, statistics, data collection.

## Introduction

SCORPION [1] is a heterogeneous wireless networking testbed consisting of various mobile nodes. The SCORPION project is intended to be used to test and evaluate wireless network protocols such as multi-radio, multi-channel medium access control, multi-hop wireless ad-hoc routing, disruption-tolerant routing, disruption-tolerant message delivery, and protocols with varying connectivity. The unique features of the testbed network allow testing of current and next-generation network applications in real world conditions. As the size of the network grows, manual collection of logs for each node becomes a time-consuming process. Also, due to their mobile nature, connectivity becomes an issue as nodes move in and out of signal range. To overcome these issues and improve the data collection process

logging on the SCORPION testbed will be standardized by using an implementation of syslog to store logs, which include GPS, network, and experimental protocol data, in an easy to parse and human readable format.

## Background

The core structure of the SCORPION testbed consists of various nodes which range from mobile briefcases (which can be carried by people or attached to automated vehicles) to mounted on buses or installed in aerial vehicles (model plane or helicopter nodes), or even mounted on mobile iRobot nodes. Currently, there can potentially be up to 88 active nodes at a given time. Various projects using the testbed include the Epidemic Routing Protocol for Delay Tolerant Networks, UCSC's Bus Tracking System (BTS), and RTT estimation using the Experts Algorithm all of which make use of the testbed nodes in order to test and collect statistics of interest.

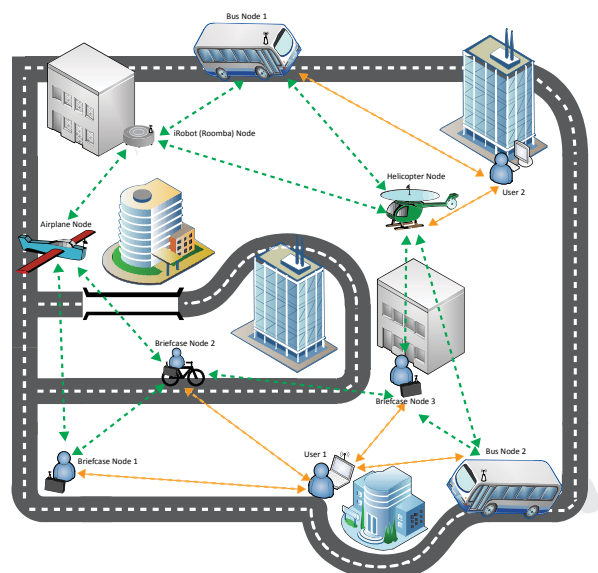


Figure 1. The SCORPION testbed consists of many mobile nodes.

### Data Collection Obstacles

Due to the scope of SCORPION, manual collection of logs from each node can be a time-consuming process due to currently up to 88 nodes active at any given time and the ability to increase in size with the addition of new hardware. Originally, the process of collecting log information from the testbed nodes required each node to be brought back to the lab in order to connect and transfer the logs for analysis.

As of summer 2009 nodes could be accessed, including transferring of log information (figure 2), wirelessly and configured using a suite of node management tools. The management tools drastically improved data collection, but collection still remains a time-consuming process. An additional problem is that because of nodes' mobile nature they may not be in range, or move out of range during transfer of information.

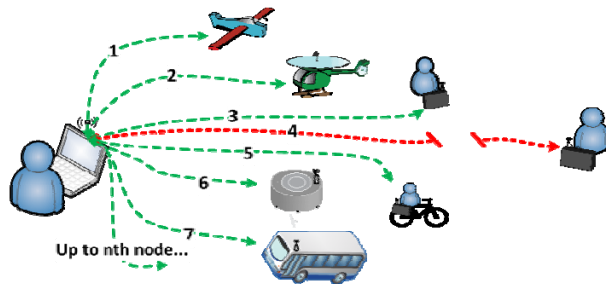


Figure 2. Manual Collection by wirelessly connecting to each node.

### Related Work

SCORPION is not the only testbed network currently implemented. Each testbed network focuses vary to different degrees. To name a few, Harvard's sensor network testbed, called MoteLab [2], is designed for the testing of sensor network applications by users. MoteLab is unique in that it allows users to upload software to nodes and schedule tests to be ran and are given results upon job completion. The network emulation testbed, emulab [3], is part of the School of Computing at the University of Utah and is designed to allow emulated experiments and live internet experimentation. Last, the Orbit-Lab Testbed [4] is a collaborative testbed project designed to

be a laboratory emulator and field trial testbed with goals of achieving reproducible testing as well as evaluating experimental protocols in real-world settings.

### Methods

The work done to help streamline the process of data collection on the SCORPION testbed included modifying how logging was handled as well as incorporating a new management tool which will be used to collect node logs in an automated fashion (figure 3), eliminating the time-consuming manual process currently in place.

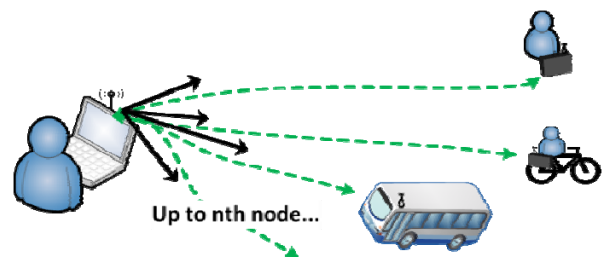


Figure 3. Automated collection using the Collection Tool.

### Logging

Logging of information on the SCORPION testbed can be a difficult task. Currently, there are three types of information of interest which includes GPS information, Network state information, and experimental protocol information. Handling of this information is broken up into three modules. First is the GPS syslogger module; responsible for collecting GPS time, location (latitude and longitude), raw altitude, number of satellites present, and speed (in knots).

Next, the network state monitoring module logs the MAC address of neighbors, connection durations, signal strengths, and number of bytes sent and received. The network statistics are collected by using the libtrace library [5], using C code, to decode and parse the 802.11 wireless and Radiotap headers, and then logged using the syslog custom module.

Lastly, the custom protocol logging module can be used to log any information of interest and used with a simple method called logInfo. logInfo was designed to be used much like the standard C method printf. For example, logInfo("formatted message: %d, ...", args, ...)

where the formatted message can contain any information relevant to the user, and the test values are passed in as arguments. The output log would be in the format of <System date and time> <computer-name> <PROTOCOL\_NAME>[<PID>]: <User Formatted message>. This method of formatting allows the user to customize the logs in a way that will simplify parsing and analysis of the desired data.

Each node runs a build of Debian Linux and has server daemon software running in the background listening for node management signals. The logging software, written in C, also runs in the background logging statistics from the GPS device attached to each node.

The first step in streamlining information gathering on each node was to use syslog to standardize data logging on the SCORPION testbed. Syslog was chosen for use due to the syslog subsystem being fully developed and ready to use in all versions of Linux. With this in mind, it allows the custom logging modules to be dropped into any Linux system without having to make changes to the system core, allowing for greater portability for future use. As data is logged over time, storage becomes a major issue when logs cannot be moved off nodes in a timely manner. Syslog handles this problem by compressing log automatically which drastically decreases storage needed for log files.

In addition to standardizing logging, a custom method (loginfo) was created to allow easy logging of experimental and protocol-specific information and designed to abstract the details of syslog use. Using the loginfo documentation, the method can be easily dropped into existing code with minimal changes to take advantage of syslog's features and logging capabilities.

### Management

Once the logging of information is handled, collection of the logs is handled by an easy to use module, created using existing management tools in addition to custom code, called node get stats. This module collects all logs related to running experiments in an automated manner.

Log collection is achieved by having the user log onto the "Testbed" network. The user then issues the nodegs command which then

messages all available nodes to send SCORPION data logs to the command issuing client.

The nodegs tool is a bash script which operates by polling the Testbed network for all available nodes using the nodels node management command, issuing the noderun command to set file permissions allowing the SCORPION logs to be copied by other users, and finally by issuing the scp (secure copy) command to copy all SCORPION logs from each node to the client machine. Currently the nodegs command issues these commands sequentially but simplifies the log collection process by automating the process.

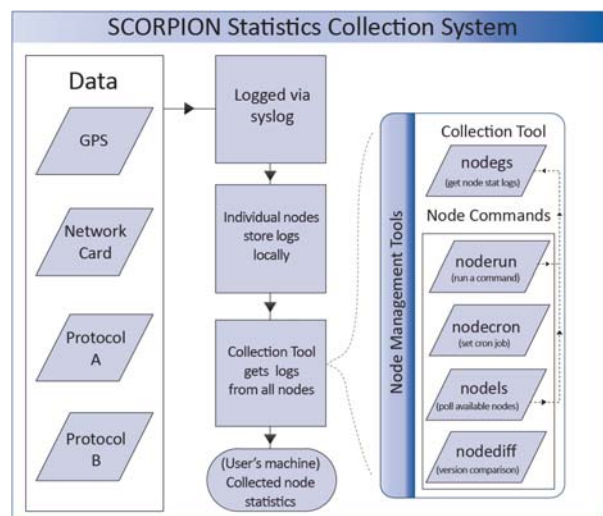


Figure 4. An overview of the SCORPION Statistics Collection System.

### Future Work

The next big step for the management tool suite would be to incorporate handling of disruption tolerant network functionality. Currently it is assumed that all nodes within range will remain connected during the entire collection process. Also, the current collection tool only works for nodes within range and connected to the same Testbed network. Future functionality should be expanded to allow nodes to communicate with each other and pass the collection message as well as the desired logs to the user. Finally, once these tools are complete they will need to be integrated into all testbed nodes and not just the few used for testing.

## Conclusion

The combination of logging, management, and collection tools allows the SCORPION testbed data to be analyzed in both current and future projects for improving and creating reliable network applications.

## Acknowledgments

This work was sponsored by the National Science Foundation, SURF-IT (surf-it.soe.ucsc.edu) Research Experience for Undergraduates Program. NSF grant Award No. CNS-0852099. We would also like to thank the University of California, Santa Cruz. Mentor: Professor Katia Obrazcka, and graduate mentors: Vladislav Petkov, Kerry Veenstra, and Bruno Nunes.

## References

- [1] SCORPION Testbed network  
<http://inrg.cse.ucsc.edu/inrgwiki/Scorpion%20Testbed>
- [2] Harvard Sensor Network Testbed  
<http://motelab.eecs.harvard.edu/>
- [3] Network Emulation Testbed  
<http://www.emulab.net/>
- [4] Orbit-Lab Testbed  
<http://www.orbit-lab.org/>
- [5] Libtrace library  
<http://research.wand.net.nz/software/libtrace.php>