

# **N2N: Network Overlays for Troubleshooting the Network**

## **Abstract**

*The Internet provides minimal support for a network operator to monitor and correlate the network events at a global scale. An infrastructure that provides the facility for correlating events from across the network (e.g. data from a SNMP and other such sources) could help in early detection of configuration errors, network attacks and could enhance the manageability of the internet. In this project we propose to build an infrastructure that will allow network operators to describe and instantiate overlays that acquire and process the network events to deliver a useful and broader view (e.g. the total number of failed login attempts per hour at the routers in the operator's domain) of the network to the operator console. We plan to implement a framework which allows specification of network events of interest and build a correlation engine which processes streams of such events from multiple sources in the network and generate appropriate alerts/notifications for network operators. We plan to use existing middleware to generate the data-flow graph that can then be mapped onto the network as an overlay which supports in-network aggregation and processing of events to reduce the communication and processing overhead imposed by the overlay.*

## **1. Problem Statement**

The network management information that is available to the network operator often comes from source like SNMP and measurement scripts that run locally. However, this information is not enough to detect and diagnose events that happen at a global level - like a DoS attack, or a configuration error. To detect such global events one needs to correlate knowledge from several local event sources. We plan to develop an infrastructure that will help us to achieve this goal. We plan to extend the system to deal with errors too – i.e., detect and correct inconsistent configurations amongst pairs of interconnected devices, thus making the network self-healing.

## **2. Related Work**

In this project we will leverage the work being done by the middleware group at Georgia Tech [6, 5]. The middleware being developed by the group provides support for facilities like in-network deployment of a processing entity that can process multiple incoming streams of data to produce one or more output streams. The entity essentially contains a description (it uses ECode [1], a highly portable subset of C for this purpose) of the operation to be performed on the incoming stream to produce the output tuples. Our project will extend the middleware to provide an infrastructure for network management. Distributed data stream management work by the STREAM [4] group at Stanford, the Borealis [3] group at Brown and MIT have focused on developing techniques for managing streams of network management information. However, most of the work has focused on the datamanagement issues that arise in such systems. Our project will focus on implementing a comprehensive system geared and optimized for dealing with network management issues.

## **3. Evaluation**

The infrastructure that we develop during the course of this project will be deployed on a representative setup at Emulab [2]. We will simulate treat each node in our emulab topology as a router providing the system with local information. The operator node will provide an interface to query such router nodes to correlate the information.

## **4. Deliverables**

We plan to work on the project in the following phases:

Phase 1 – Define network issues that can be detected and monitored using our approach. Design algorithm for placement of correlation engines in the network. Design algorithm for correlation of events and design the correlation engine.

Duration: 4 weeks

Phase 2 – Code and test

Duration: 4 weeks

Phase 3 – Evaluate using Emulab setup with synthetic data sets with characteristic network events.

Duration: 4 weeks.

We plan to use the results obtained from this work towards a term/workshop paper.

Final Demo: May 1<sup>st</sup> 2006.

## References

- [1] EVPath. <http://www.cc.gatech.edu/systems/projects/EVPath/>.
- [2] The Utah Network Testbed. <http://www.emulab.net/>. University of Utah.
- [3] Y. Ahmad and U. Cetintemel. Network-aware query processing for distributed stream-based applications. In *Proceedings of Very Large Databases Conference*, 2004.
- [4] S. Babu and J. Widom. Continuous queries over data streams. *SIGMOD Record*, 30(3):109–120, 2001.
- [5] V. Kumar, B. F. Cooper, Z. Cai, G. Eisenhauer, and K. Schwan. Resource-aware distributed stream management using dynamic overlays. In *Proc. IEEE ICDCS*, 2005.
- [6] V. Kumar, B. F. Cooper, and K. Schwan. Distributed stream management using utility-driven self-adaptive middleware. In *Proc. ICAC*, 2005.