Background and Goals

Traditional network monitoring systems are based on client/server paradigm and have two drawbacks: (1) static agents: their interface, functionality and alarms (if any) are statically defined and implemented, and thus cannot be modified or enhanced on-line. (2) centralized processing: all the static and low-level agent information has to be collected to the network control center and get processed centrally, which leads to poor scalability and one-spot failure.

We aim to build an intelligent, hierarchical network monitoring architecture using Java that can solve the above-mentioned problems. The significance of our work relies on the assumption that JVM will become prevalent in network devices in the near future.
Methods

The project is done in pure Java:

◆ Take advantage of Java’s run-time class loading to facilitate dynamic enhancement to the agent interface and functionality

◆ Use Java Remote Method Invocation (RMI) mechanism for class shipping over the network and hierarchical distribution of the monitoring system.
Features

◆ Intelligent monitors, object attributes and operations are all Java objects, and therefore are dynamically changeable

◆ Classes can be loaded locally or from network via Java RMI

◆ Uniform interface facilitates hierarchical distribution of JAMA over the managed network

Significance

Provides one possible platform on which the next-generation intelligent network management systems can be built
Future Work

◆ Use directory service for system naming and object management

◆ Incorporate control objects into the system to build network performance or fault management systems

◆ Integrate JAMA with Web techniques to provide a friendly and cross-platform user interface (UI) to human managers