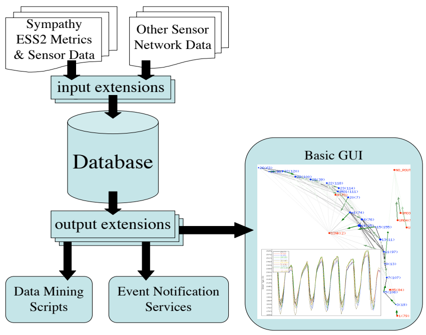
Demo Abstract: D.A.S. – Deployment Analysis System

Kevin Chang, Nithya Ramanathan, Deborah Estrin, Jens Palsberg
UCLA Center for Embedded Network Sensing
{kchang, nithya, destrin, palsberg}@cs.ucla.edu

ABSTRACT

Understanding how a sensor network system works requires running the system, extracting system log files, and manually interpreting data. When interpreting system data, we often try to correlate behavior over multiple modalities. For example, if a node is exhibiting strange behavior, the cause may be due to weak battery, geographically bad placement, collision, interference, sensor failure, algorithmic faults, or a combination of the above. While the approach of interpreting log files is adaquate for closed systems such as the ones run in simulations for limited duration, it is difficult, laborious, and error-prone for big and complex sensor network systems that have already been deployed for weeks or even months in the fields. Thus, a suite of tools to help analyze complex sensor network system is desirable. We have implemented Deployment Analysis System (DAS), a centralized data mining suite designed to better understand sensor networks. It supports visualization and deployment-related queries that allow the user to inspect historical system metrics, environmental data, geographical placement, and system status.

The graphical user interface displays real-time and historical information in an easy and intuitive format. The topological map feature allows users to have both a historical and immediate view on the status of the system with metrics such as routing table, neighbor tables, and the number of neighbors heard, among others. The historical map helps users view and narrow down past points of failure and for future deployment refinements. For example, it may help pinpoint bottlenecks in the network or bad connectivity due to environmental conditions, resulting in unexpected low throughput at the sink. In addition to the static map, DAS also provides historical routing replay animation. This feature allows programmers to validate and to make assessments to routing algorithms. For example, using the replay animation, a user can determine route stability, and overall efficiency of mote placements and route choices. In one instance, we used DAS to inspect one of the environmental deployments in James Reserve at San Jacinto and observed that in many cases, motes that had high elevation were frequently used as a next-hop neighbor even though they were much farther away from the final destination. We also observed unexpected routing dependencies such that when a mote became inoperative, the communication for a cluster



of other motes were cut-off, even though their proximities to each other should have provided redundancy.

In addition to the topological map, DAS generates different charts that include single metric multiple nodes graph, multiple metrics single node graph, and link quality graph. Using graphs users can easily identify spatial-temporal correlations and events over system metrics and environmental data. For instance we used DAS to easily generate two graphs in two different instances (using a total of less than 4 mouse clicks) which show that the battery voltage of nodes tends to degrade simultaneously. In both cases the entire sensor network was broken within 5 days. We used this observation to apply philosophically to future deployments– if the battery level of a few motes starts to degrade, we should change batteries quickly as a preventive measure from total system failure. Likewise, we use DAS to easily generate graphs that show the temperature affects the battery level directly, and occasionally happens when route-flapping occurs. This correlation opens up the possibility of fine tuning routing and link estimation algorithms based on past data and even predicted trends.

The back-end of DAS is completely detached from its CGI based front-end. It stores DAS environmental data with internal system metrics in a traditional database to allow interfaces with input extensions and output extensions. Currently DAS has a pluggable input extension to Sympathy system metrics and general sensor data from ESS2, and the infrastructure allows other data types to be easily added or changed to accommodate for other sensor networks. Some of the output extensions that have been implemented include graphical user interface, status display, and event notification facilities. Since the framework uses open tools it can be easily interfaced with other systems including GNU Plot, Java, and Visual Studio tools.

Keywords

deployment, analysis, data mining, visualization, graph generation, status notification

SenSys'05, November 2–4, 2005, San Diego, California, USA.
ACM 1-59593-054-X/05/0011.