

A Self-Configuring Sensing System for Data Centers

Malena Mesarina, Cyril Brignone, Tim Connors, Mehrban Jam, Geoff Lyon,
Salil Pradhan, and Bill Serra

Hewlett-Packard Laboratories
Palo-Alto, CA 94304, USA

Abstract. Wiring sensors in a data center is extremely expensive in comparison to the wiring of computing equipment. This is due to the central architecture of traditional sensing systems, which requires long wires to be connected between racks and a central box. In addition, a re-layout of the racks after the sensor wires are deployed is practically impossible. We propose using a wireless self-configuring network of smart sensing nodes to alleviate these problems. We explore how to design the sensor control software to be self-reconfiguring when nodes relocate. The software is divided in three layers: network organization, data aggregation and visualization. In this paper, we identify several insights into the thermal monitoring requirements, design issues and initial design solutions for these layers.

1 Introduction

We propose a self-configuring, wireless sensing network to eliminate the labor cost of wiring a sensor¹ and facilitate the re-layout of racks. The nodes in the network are able to locate themselves, using a time-of-flight method, with a precision of less than 5 cm [1]. The thermal monitoring software is structured in layers of network, data aggregation and visualization functions. Self-configuration means automatic adaptation of these layers to node moves or failures. The network layer adapts by updating data routes; the aggregation layer by reselecting clusters of sensors and the visualization layer by relocating objects in its model. The adaptive algorithms are distributed for the network and centralized for the aggregation and visualization layers. In this report on a first experimental implementation, we identify requirements, design problems, tradeoffs, and provide insights into initial algorithmic solutions.

2 Software Layers

The layers mentioned above should meet the application's requirements: to avoid IP address assignment to sensors, the wireless and IP networks should be separate and bridged by access points. Sensor reading is periodic or event triggered.

¹ \$1000 according to customers

The data flows from sensors to access points, and between immediate neighbor sensors. Data bit rate is low (5-10 kbps) and sensor density is high ($5\text{-}7/m^3$). The following assumptions are made: a node knows its location and its neighbors' locations and ids; nodes report their locations to access points, and the radio has a fixed transmission power.

2.1 Network Configuration, Data Aggregation and Visualization

The network layer solves three problems: determining the number of access points, finding routes from sensors to access points, and updating routes when nodes move or fail. The number of access points depends on the maximum number of sensors that talk to an access point, bandwidth, and cost. We use a source routing approach for route discovery. The access points broadcast discovery packets to nodes within a delineated area. A receiving node appends its IDs and location and forwards the packet. Nodes construct multipaths using information from the discovery packets. Updating routes consists of deleting paths from memory when a neighbor sensor moves or fails, and adding a new path when a new neighbor node appears. The underlying localization algorithms in the node detect when a node has moved or failed and notifies the network layer. A node computes a new path with the new node as an intermediate node based on the new node's location and existing knowledge of paths.

The network aggregates packets to reduce the number of packet transmissions, saving power and bandwidth. We use a 1-hop clustering scheme, where a cluster-head acts as the data aggregator. The problems to solve are: cluster formation, cluster-head selection, de-synchronization of data collection between channel sharing clusters, and selection of non-colliding paths from cluster-heads to access points. Cluster formation should be based on proximity, line of sight between sensors, and historical data correlation. The cluster formation and cluster algorithms are centralized and controlled by the access points, since they require global knowledge of nodes locations and failures.

Given the location of sensors, our visualization tool, Geoview, creates a geometric model of the room in 3D, allowing the user to program the sensors by clicking on the display.

2.2 Conclusions

We propose a self-configuring sensing infrastructure and control software that adapts to sensor mobility. We identify several system requirements, design problems, tradeoffs and initial solution approaches. We plan to deploy our system in a data center at HP Labs in Palo Alto in the fourth quarter of 2003. Low implementation costs and reconfiguration latency will validate the efficiency of our design in comparison to traditional methods.

References

- [1] Brignone, C., Lyon, G.: Smartlocus: A location aware system for adaptive environments, hpl-2003-41. Technical report, Hewlett-Packard (2003) 200