Title
SYS6: Tenet: An Architecture for Tiered Embedded Networks

Permalink
https://escholarship.org/uc/item/13w231kr

Authors
Krishna Chintalapudi
Deborah Estrin
Om Gnawali
et al.

Publication Date
2005
Tenet: An Architecture For Tiered Embedded Networks

Krishna Chintalapudi, Deborah Estrin, Om Gnawali, Ramesh Govindan, Eddie Kohler, Jeong Paek, Sumit Rangwala, Thanos Sthathopoulos

Introduction

Traditional view of Sensor Networks

- Thousands of impoverished nodes distributed over vast regions
  Traditionally sensor networks are viewed as a collection of thousands of resource constrained devices collaborating as a collective.
- Application specific code potentially resides on every node
  Energy, bandwidth and processing constraints on the nodes require finely tuned application specific code and limits reusability across applications.

Challenges faced in building such networks

- Programming, debugging is hard and time-consuming
  - Developing and debugging distributed collaborative applications on the resource constrained motes is hard
  - To do this afresh for every application is even harder
- Too much complexity as a system
  - Such an architecture leads to fragile and unmanageable systems
  - Resource management in such systems is very hard

How do we build sensor network systems that are robust and manageable?

Large scale sensor networks will be tiered

- Two classes of nodes – mote-class nodes, gateway-class nodes
  - Mote-class nodes (eg. MicaZ) interact with the real world i.e sense/actuate
  - Enable flexible deployment of instrumentation
- Gateway-class Nodes (Masters)
  - Needed for scaling overall network capacity
  - Have relatively more plentiful resources

The Tenet Architecture

Design Principles of Tenet

- Asymmetric Task Communication
  - Any and all communication from a master to a mote takes the form of a task. Any and all communication from a mote is a response to a task; motes cannot initiate tasks themselves.

Addressability

- Any master in a Tenet can communicate with any mote or master in that Tenet, as long as there is possibly multi-hop physical layer connectivity between them. Furthermore, any node in a Tenet can communicate with at least one master in that Tenet, unless no master is reachable.

Task Library

- Motes provide a limited library of generic functionality, such as timers, sensors, simple threads, data compression, and FFT transforms. Each task activates a simple subset of this functionality.

Robustness and Manageability

- Robustness and manageability are primary design goals.

Tenet : An Architectural Principle for Tiered Embedded Networks

- A programmable Structural Health Monitoring (SHM) system
  - SHM techniques detect and localize damages in structures
  - Allows an SHM engineer to program in Matlab/C through an API
  - Reliable delivery, time synchronization and routing are provided as basic services
  - Mica-Zs form the mote-class nodes, PC/Stargates form the gateway-class nodes
  - Has been implemented and deployed on scaled models as well as real structures

NetSHM – A Precursor to Tenet

Sample Matlab Damage Detection Code in NetSHM

```
function shifts = getModalShiftsFromBuilding()
    % Create a group for sensors
    gidSensors = NetSHMCreateGroup([1,2,3,4]);
    % Create a group for actuators
    gidActuators = NetSHMCreateGroup([5]);
    % Actuate after 22 seconds
    NetSHMCmdActuate(gidActuators,22);
    % Collect structural response starting 20 seconds from now, 4000 samples at 200Hz, along x-axis only
    samples = NetSHMCmdGetSamples(gidSensors,20,200,1,4);
    % Find modal frequencies
    modes = findModes(samples);
    % Read original modes
    load OriginalModes;
    shifts = findModalFreqShifts(modes,OriginalModes);
end
```

Scaled model of a 6-story building for testing damage detection and localization algorithms