

UCLA

Posters

Title

Tenet: An Architecture for Tiered Sensor Networks

Permalink

<https://escholarship.org/uc/item/3s94w7v2>

Authors

Gnawali, Omprakash

Jang, Ki-Young

Paek, Jeongyeup

et al.

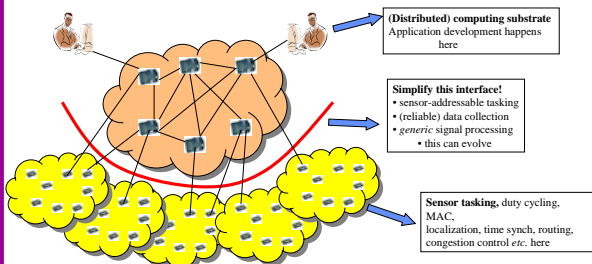
Publication Date

2007-10-10

Tenet: An Architecture For Tiered Sensor Networks

Omprakash Gnawali, Ki-Young Jang, Jeongyeup Paek, Marcos Vieira, Karen Chandler, Vinayak Naik
 Deborah Estrin, Ramesh Govindan, Eddie Kohler
<http://tenet.usc.edu>

Design Principles of Tenet



Multi-node data fusion functionality and complex application logic should be implemented only on the masters, since the cost and complexity of implementing this in motes outweighs the performance benefits of doing so.

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.

Addressability

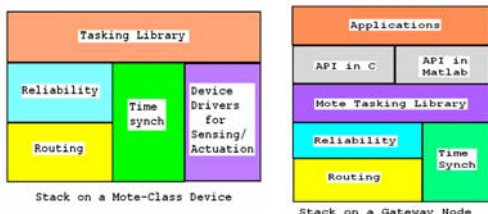
Any master in a Tenet can communicate with any mote or master in that Tenet. Any mote in a Tenet can communicate with at least one master in that Tenet.

Task Library

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

Software and Tools

The Tenet Stack



Tasking API

Tasking API provides functions to describe a task to run on the motes.

Example: `sample()`, `actuate()`, `send()`, `count()`, `avg()`

Transport API

Transport API provides functions to disseminate tasks and collect data from the network:

`send_task(task_description)`
`attr* read_response(wait_interval)`

Deployment and Testing tool: Tenetrun

- To simplify deployment and testing, “Tenetrun”:
- Daemonizes the processes that constitute the master stack and restarts them if they fail.
- Instantiates multiple copies of the master stack to emulate multiple masters in a single physical host.

Centralized Routing: Centroute

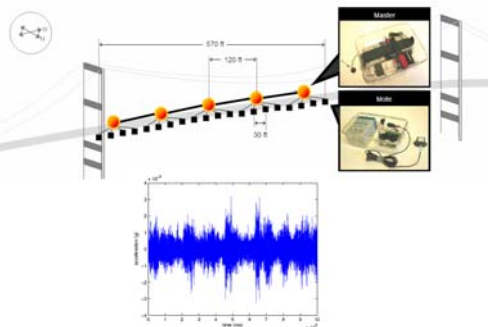
Tenet supports Centroute for mote-to-master and master-to-mote routing. Centroute features:

- Low control and state overhead on the motes
- Low churn – routes are determined during initialization or to recover from packet loss
- State and route management on the masters thereby simplifying the code on the motes

Applications and Deployments

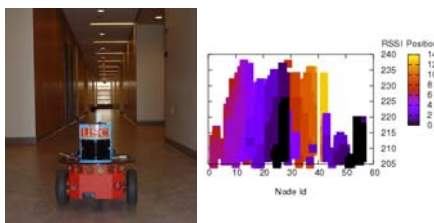
Ambient Structural Vibration monitoring

Continuous structural monitoring and event detection
`“sample(3 channels, 20 Hz) → send(stream)”`
<http://enl.usc.edu/projects/bridge/>



Pursuit Evasion Game

Pursuer robots estimate the location of evaders and corral them.
`“sample(0xaa, RSSI) → compare(LT, 0xaa, 125, 0xbb) → deleteactivetaskif(0xbb) → send()”`
<http://enl.usc.edu/projects/peg/>



Wildlife (Lizard/snake) monitoring

Cyclops take pictures of the trap, run image processing algorithms to detect the animals, and the result directs the biologists to the trap that has captured lizards or snakes.
`“periodic(2 mins) → detect_lizard(0xbb, 0xaa) → not(0xaa) → deleteactivetaskif(0xaa) → send(stream)”`

