Distributed Computation in Wireless Ad Hoc Grid Formations with Bandwidth Control

Elisa Rondini and Stephen Hailes
University College London

MSN 2007, 13th July 2007
Overview

- Scenario
- Assumptions
- Challenges
- Research Problem
- Approach to Solution
- Experimental Results
- Conclusion
- Future Work
Emergency Scenario

- Emergency ‘CBRE’ events
- Indoor environment (e.g., building or highly frequented public space)
- Hazardous for the first responders to explore the contaminated area
- Need for fast environmental information gathering for C²
Why Sensors?

• Cheap devices
• Easily deployable
• Capable of working in hazardous environments
• Standard development platform in the research community

Even with the availability of more computational capable devices, it is still worth thinking about an infrastructure for the distribution of the computation (it is easy to find applications that exceed the global capacity of every single node)
TMote Sky devices

- Sensor characteristics:
  - limited battery power
  - small storage capacity (i.e., 10kB RAM, 48kB Flash ROM)
  - low computational power (i.e., MSP430 16-bit microcontroller)
  - limited communication capabilities (i.e., CC2420 radio module supporting IEEE 802.15.4, 250kbps as maximum available bandwidth)
  - sensing capacity (i.e., humidity, light and temperature sensors are supported by the platform - with the ability to interface more)
Ad Hoc Grid Formation

- Toxic Cloud
- Sensor Moving
- Ad Hoc Grid Formation
- Mobile Sensor holding a Computational Intensive Application
- CBRE Event
- Fixed Sensor
(2) The Bandwidth Problem

Congested Area

CPU Availability=80%

Uncongested Area

CPU Availability=30%

= communication messages
A, B = Nodes involved in a main flow passing through the area of congestion
S = Server (Fixed Sensor)
C = Client (Mobile Sensor)
Existing Work Limitations

- They do not take into account communication issues
- They are implemented in simulation environments:
  - In simulators, radio communication model are oversimplified
- The implicit difficulties of real-world applications are not met
- They often use a simplistic model of energy-efficiency as metric (instead of latency)
Research Problem

Explore the convergence of *ad hoc sensor networking* and that of *computational grids*, presenting novel paradigms that take into account both the computational capabilities of the nodes (CPU) and the local network conditions (bandwidth) when distributing computation.
Initial Approach

- We selected two load sharing algorithms (Auction and Lookup List)
- We adapted them to take into account not only CPU but also Bandwidth requirements during the decisional process
- We implemented them on a sensor testbed
- We used a performance evaluation measuring the average latency
Auction Algorithm

Ad hoc grid formation

1. Broadcast a Task Request Message (with task CPU and Bandwidth details)
2. Send a Bid (containing CPU and Bandwidth availability)
3. Select the Best Available Node (computing an Utility Function)
4. Offload the Task to the Winner of the Auction.
5. Upload the Result Back
Lookup List Algorithm

Utility Function

Parameters:
• $i$ = neighbour of a Client ($i=1, ..., N$)
• $N$ = number of a client neighbours
• $C(i)$ = CPU availability of neighbour $i$
• $B(i)$ = Bandwidth availability of neighbour $i$
• $S(i)$ = Score of neighbour $i$
• $w_C$ = Weight CPU
• $w_B$ = Weight Bandwidth

$$S(i) = w_C * C(i) + w_B * B(i)$$

Best Candidate = $\max\{S(i)\}$

$i=1, ..., N$
Small-Scale Testbed

Fixed TMote Sky sensors running the Contiki operating system

(Streaming Node)

(Congested Area)

(Streaming Node)

(Area Affected by Natural Offload Traffic)

Radio Communication Range
Experimental Results

The graph shows the experimental results for different auction algorithms under various conditions. The x-axis represents the number of tasks, while the y-axis indicates the job execution time in seconds. The graph compares:

- Auction Alg. With Congestion Without Bandwidth Control
- Auction Alg. With Congestion With Bandwidth Control
- Auction Alg. Without Congestion

The data points show that adding congestion and bandwidth control significantly affects the job execution time compared to running the auction algorithm without these controls.
Experimental Results (2)

![Graph showing job execution time vs. number of tasks for different algorithms with and without congestion and bandwidth control.]

- Lookup List Alg. With Congestion Without Bandwidth Control
- Lookup List Alg. With Congestion With Bandwidth Control
- Lookup List Alg. Without Congestion
Experimental Results (3)

![Graph showing job execution time versus number of tasks for different algorithms.](image)
Experimental Results (4)
Experimental Results (5)

![Graph showing experimental results with two lines representing different algorithms. The red line represents the Lookup List Algorithm Without Bandwidth Control, and the black line represents the Lookup List Algorithm With Bandwidth Control. The x-axis represents Task Offload Size (Packets), and the y-axis represents Job Execution Time (Seconds).]
Experimental Results (6)

- Lookup List Alg. Without Bandwidth Control
- Lookup List Alg. With Bandwidth Control
- Auction Alg. Without Bandwidth Control
- Auction Alg. With Bandwidth Control

The graph shows the relationship between job execution time (in seconds) and task offload size (in packets), with lines indicating different algorithms and bandwidth control states.
HEN Mote Testbed

http://www.cs.ucl.ac.uk/research/hen/

- 40 Tmote Sky Sensors
- Random Deployment
- Remote Accessed
- Remote Programmed
- Fast Kernel Flashing

http://www.cs.ucl.ac.uk/research/hen/
Large-Scale Testbed - 1

- HEN Mote Testbed (UCL)
- Radio Channel 20
- 32 Tasks Execution
- 50 Offload/Upload Packets
- UDP/TCP Communication
Experimental Results

- Auction Alg. With Congestion Without Bandwidth Control
- Auction Alg. With Congestion With Bandwidth Control
- Auction Alg. Without Congestion
Experimental Results (2)
Large-Scale Testbed - II

- HEN Mote Testbed (UCL)
- Radio Channel 26
- 32 Tasks Execution
- 50 Offload/Upload Packets
- UDP/TCP Communication
Experimental Results

- Auction Alg. With Congestion Without Bandwidth Control
- Auction Alg. With Congestion With Bandwidth Control
- Auction Alg. Without Congestion
- Auction Random Alg.

Job Execution Time (Seconds) vs. Number of Tasks
Experimental Results (2)

- Lookup List Alg. With Congestion Without Bandwidth control
- Lookup List Alg. With Congestion With Bandwidth control
- Lookup List Alg. Without Congestion
- Lookup List Random Alg.

Graph showing job execution time (seconds) vs. number of tasks.
Conclusion

In all different scenarios, although the overall performance was greatly affected by the changing environmental conditions, significant performance improvement (using latency as metric) was always obtained by taking the bandwidth into account during the collaborative distribution process.
Future Work

- Case study on a real-world application (e.g., localisation)
- Introduction of node mobility
- Introduction of node heterogeneity
- Introduction of security issues
Questions?