Using Grid Technologies to Optimise a Wireless Sensor Network for Flood Management

Danny Hughes, Phil Greenwood, Barry Porter, Paul Grace, Geoff Coulson, Gordon Blair, Francois Taiani
Computing Department, InfoLab21,
South Drive, Lancaster University
Lancaster, LA1 4WA, UK.
danny@comp.lancs.ac.uk

Florian Pappenberger, Paul Smith, Keith Beven
Environmental Science Department,
Lancaster University,
Lancaster, LA1 4YQ.
f.pappenberger@lancaster.ac.uk

1. INTRODUCTION

Current approaches to flood monitoring (e.g. in river valleys) involve statically deploying depth and ultrasound-based flow sensors across flood-prone areas, and feeding the collected data off-site (e.g. using GSM) to grid-based computational models which predict flooding trends [1] [2]. We believe that there is considerable scope for improvement in such scenarios. In particular, we are investigating selectively shifting the execution of prediction models *to the sensor network itself*, which thus acts as a 'mini-grid'. Computations organised in this way can be used not only to provide more timely flood warnings, but also to help to *dynamically adapt* the wireless sensor network (WSN) and thus optimise it for current or predicted environmental conditions. For example, the network can employ a low power, low throughput organisation in quiescent conditions, and switch to a high power, resilient, high throughput organisation when flooding is imminent. To achieve this vision we have developed a sensor network framework based on an appropriate combination of software and hardware.

Software The software consists of our component-based run-time reconfigurable GridKit middleware [4]. This provides the key functionality required to develop both WSN and grid applications: i.e. flexible networking support, service binding, resource discovery, resource management, and security. Gridkit is based on our language-independent OpenCOM component model [5] with each area of functionality being implemented as an independent component framework. As it is component based, GridKit is inherently configurable and extensible, allowing us to build rich support for grid and WSN applications, or, conversely, to build minimal deployments that are suitable for execution on embedded hardware (even on hardware such as Berkeley motes [6]). Of particular significance for the flood monitoring WSN scenario is GridKit's 'Open Overlays' framework, which supports 'pluggable' application level networks which can be instantiated dynamically (see more on this below).

Hardware Our hardware platform is based on the *Gumstix* embedded computing platform [3]; we call the combined hardware and software platform 'GridStix'. The Gumstix—so named because they are roughly the size of a pack of chewing gum—are significantly more powerful than highly-embedded devices such as Berkeley Motes. Each features an Intel XScale CPU running at up to 400MHz, and has 64MB of RAM, 16MB of flash memory and Bluetooth radio. Additionally, each node has 802.11b networking hardware and a subset of nodes are equipped with serial GPRS modems for off-site data dissemination. As might be expected, this computational power and flexibility comes at the expense of power consumption. During typical operation each Gumstix consumes up to 1 watt with a maximum theoretical power draw of up to 3 watts. This is not a significant problem however in our target domain, as it is quite feasible for these power requirements to be met using a high-capacity battery backed by a mid-sized solar panel.

Networking and Adaptation As noted, GridStix nodes are capable of maintaining an ad-hoc on-site networking infrastructure using either Bluetooth or 802.11b or a combination of the two. This is useful as these networks have quite different properties and one can be selected over the other to reflect current environmental conditions, changing application requirements or failures in the network. For example, to detect the rate-of-flow of rivers, we employ network cameras and a locally-executed image analysis algorithm to identify and track naturally occurring tracer particles on the water surface. Individual nodes can detect coarse-grained changes in surface velocity in a timely fashion; however, for more precise measurements, images need to be distributed to a number of nodes. To support this, we can switch network type from Bluetooth to 802.11b as Bluetooth does not have sufficient bandwidth to distribute the large image data-set in a timely fashion. The specific power consumption and performance of the GridStix and their various networking technologies is discussed in more detail in [7].

Other, more fine-grained, adaptations can be performed at the overlay level, in that different overlays can be substituted for each other at run-time depending on environmental conditions. For example, the nodes might initially be structured using a power-efficient shortest-path tree for off-site data dissemination. However, trees of this type have poor resilience to failure. Therefore, when flooding is predicted we can increase resilience (at the expense of power consumption), by dynamically substituting a more resilient fewest-hop spanning tree. Thanks to GridKit's component-based nature, this can be achieved by simply replacing the 'forwarding' component of the shortest-path spanning tree.

2. THE GRIDSTIX DEMO

Our demonstration will use a mixture of physical props, audience participation and on-screen visualisations to illustrate how a GridStix-based WSN, deployed to perform flood monitoring, can adapt its behaviour to best suit changing environmental conditions. The demo installation (see Figure 1, left) comprises a display table covered with a satellite map showing the actual site at which the WSN is being deployed. Five Gumstix nodes are 'deployed' on this map in positions reflecting their real-world locations. The physical installation is accompanied by two on-screen visualizations: The first of these shows the current overlay network topology, and the second shows the current software component configuration on each node (see Figure 1, right).

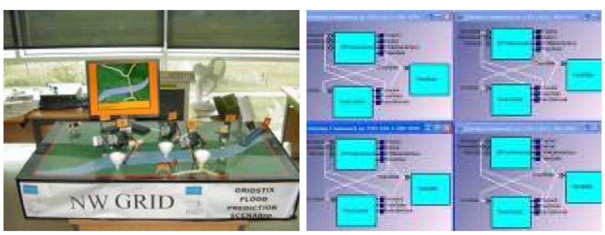

Figure 1 – Demo Installation and Component Visualisation

At the demo, the audience is invited to interact with the demonstration. This happens in *two* ways. *First*, participants are invited to simulate fast river flow conditions by rolling marbles down a chute located at the upper-right hand corner of the installation (see Figure 2). The motion of the marbles is detected by a network camera and this causes the network to switch from Bluetooth to 802.11b, simulating the adaptation that would occur when an on-site camera detected increased flow rates (see above). When the user stops rolling marbles, this simulates river flow returning to normal conditions and the WSN switches back to the Bluetooth network to conserve power. This again is reflected in the network visualization.

In the *second* mode of audience interaction, participants are invited to simulate increasing water depth. In a real-world deployment, depth predictions are made using a 'point-based' flood prediction algorithm executed on a 'mini-grid' formed from the collected computational resources of the GridStix nodes. Increased flood risk causes the WSN to adapt the spanning tree being used to disseminate data off-site. In particular, as outlined above, the configuration changes from a low-power shortest-hop spanning tree to a more reliable fewest-hop spanning tree by swapping the 'forwarding' component of the 'overlay' component framework running on each node. For the purposes of the demo, to simulate increasing water depth, users are invited to pour water into one of three funnels located in front of each node (see Figure 2). This triggers a water sensor, which causes the WSN to perform the above reconfiguration. This is reflected both in the component visualization (which shows the forwarding components of the spanning tree overlay being swapped) and in the network visualization (which shows the rebuilding of the tree to conform to the fewest hop topology). Following this adaptation sequence, users are invited to continue adding water, which, at a set level, simulates node failure due to immersion. This again is reflected in the network visualization as the node is removed from the network topology.

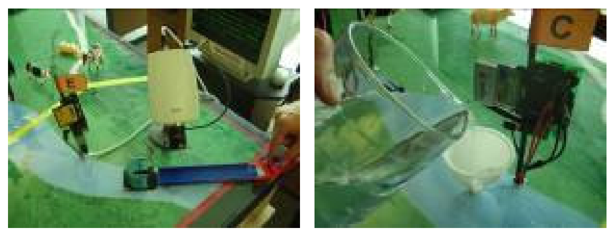

Figure 2 – Audience Interaction with the Demo Installation

3. REFERENCES

1. F. Pappenberger, K. Beven. et al., "Cascading model uncertainty from medium range weather forecasts (10 days) through a rainfall-runoff model to flood inundation predictions within the European Flood Forecasting System (EFFS)", Hydrology and Earth System Science, 9(4), pp381-393, 2005.

2. F. Pappenberger, P. Matgen and K. Beven, "The influence of rating curve and structural uncertainty on flood inundation predictions.", Advances in Water Resources. (in press) 2006.

3. Waysmall Computers "Gumstix Embedded Computing Platform Specifications", website: http://gumstix.com/spexboards.html.

4. G. Coulson, P. Grace, G. Blair et al, "Open Overlay Support for the Divergent Grid", in the proceedings of the UK E-Science All Hands Meeting, Nottingham, UK, September 2005.

5. G. Coulson, G. Blair et al "The Design of a Highly Configurable and Reconfigurable Middleware Platform", ACM Distributed Comp Journal, Vol. 15, No 2, pp109-126, April 2002.

6. Crossbow "Mica Motes and Sensors", website: http://www.xbow.com/

7. "An Intelligent and Adaptable Flood Monitoring and Warning System", Hughes D., Greenwood P., Coulson G., Blair G., Pappenberger F., Smith P., Beven K., to be published in the proceedings of the 5th UK E-Science All Hands Meeting (AHM'06), September 2006

APPENDIX

For clarity, please find included below larger photos of the demonstration installation, associated visualizations and interactions.

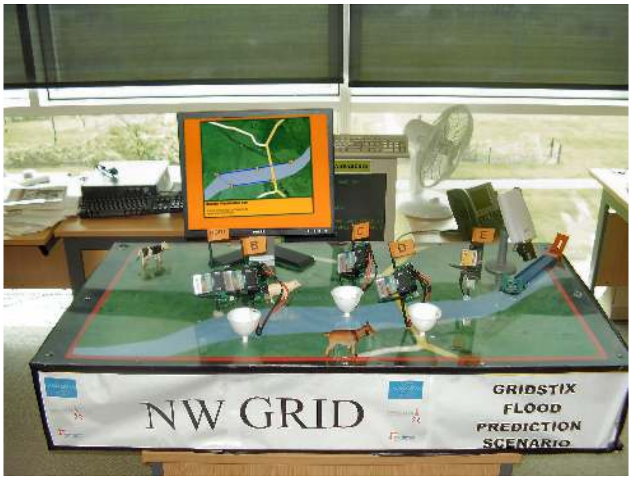

Figure 1 - Demonstration Installation

Figure 2 – Network Topology Visualization

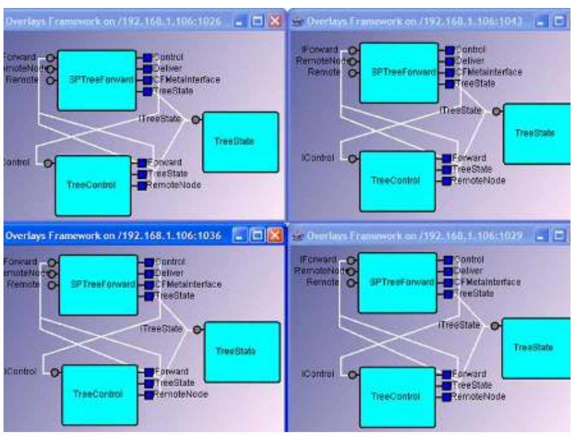

Figure 3 – Real-time Component Visualization

Figure 4 – Triggering 'High Flow' Adaptation

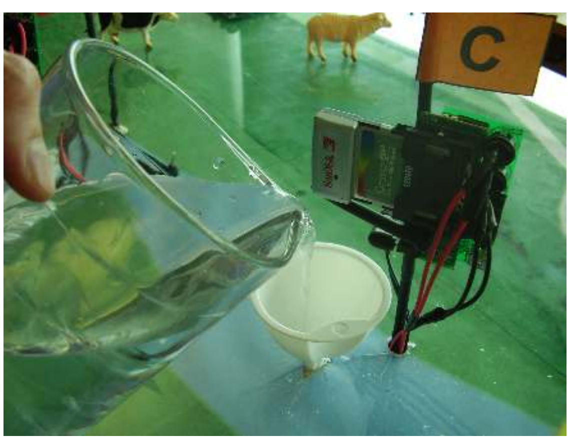

Figure 5 – Triggering 'High Water' Adaptation

A video showing this demonstration in action along with a short presentation on system functionality has also been produced and may be downloaded in Microsoft WMV format from the following address (6 Min / 41MB):

http://www.comp.lancs.ac.uk/computing/users/hughesdr/DemoVideo_Final.wmv