Managing Heterogeneous Data Flows in Wireless Sensor Networks Using a 'Split Personality' Mote Platform

Danny Hughes[1], Mickaël Daudé[2], Geoff Coulson[1], Gordon Blair[1],
Paul Smith[3], Keith Beven[3], Wlodek Tych[3]

[1] Computing Department, InfoLab21, South Drive, Lancaster University, Lancaster, UK. LA1 4WA
{danny, geoff, gordon} @comp.lancs.ac.uk

[2] INP ENSEEIHT, Toulouse, France. 31071
mickael.daude@etu.enseeiht.fr

[3] Environmental Science Department, Lancaster University, Lancaster, UK. LA1 4YQ
{k.beven, p.j.smith, w.tych}@lancaster.ac.uk

Abstract

Remote sensing applications are increasingly making use of sensors that require significant computational and networking resources (e.g. digital imaging devices). The data flows generated by such sensors are radically different than those generated by simple voltage or current-based sensors and require better network performance and more computational resources. This paper presents the design of GridStix2, a novel wireless sensor network platform that offers alternative high-performance and low-power 'personalities'. This two-tier approach allows GridStix2 to provide power efficient support for sensors which produce simple data flows, and rich computational and network support for more demanding sensor types. Our approach is evaluated in the context of a deployed flood monitoring and warning system.

1. Introduction

Nodes in wireless sensor networks (WSNs) are increasingly required to support complex data flows and in-network computation [1] [11]. Examples of such complex sensor nodes are digital cameras, audio sensors and sensors based on spectrum analysis. In the flood monitoring scenario featured in this paper (see section 2), significant computing and networking resources are employed to support image-based flow measurement and on-site flood modeling. But our system also supports more 'common-place' WSN functionality such as relaying readings from pressure transducers and conductivity sensors to a GSM uplink [1].

Image based flow measurement and on-site flood modeling require support that is not typically present in WSN environments: specifically, network overlay support for data-flows between motes, support for sporadic high-bandwidth communications, and in-network computation. These requirements are discussed in more detail in section 2.

Our previous-generation GridStix platform supported flood warning and prediction using a combination of our Open Overlays middleware platform [2] and an extended version of the GumStix [3] embedded computing platform. Unfortunately, while this platform was capable of meeting significant computational and networking demands, it did so at the expense of power consumption. This paper presents the GridStix2 platform which addresses the energy consumption shortcomings of GridStix v1 by providing two discrete 'personalities': a *power-saving* personality and a more computationally capable *high performance* personality. In the context of the high-performance personality, the Open Overlays platform provides rich networking support, while in the context of the low power personality, Open Overlays is used to support a more basic configuration.

The remainder of this paper is structured as follows: Section 2 discusses system requirements of next-generation sensor nodes in the context of complex data flows and in-network computation; Section 3 describes the GridStix2 platform; Section 4 describes the deployment environment and initial evaluation; Section 5 provides context on related work; and Section 6 offers our conclusions and proposes directions for future research.

2. System Requirements

Our flood monitoring system [1] supports three distinct processes that impose quite different system requirements: i) reporting of sensor data; ii) in-network computational flood modeling; and iii) image based flow measurement. The computational and networking requirements of these processes are discussed in

Sections 2.1 and 2.2 respectively. Reporting of sensor readings occurs continually while in-network computational flood modeling and image-based flow measurement are performed only when dictated by environmental conditions [1].

2.1 Data Flows

The networking requirements imposed by the reporting of depth readings, in-network computational flood modeling and image-based flow analysis are as follows:

Reporting of depth readings Our system sends data from pressure-based depth sensors to a GSM uplink for dissemination off-site. Each sensor reading comprises a 1 byte node identifier, a time-stamp and two 12 bit ADC readings, giving a total size of 5 bytes per sensor reading. Pressure sensors are sampled at intervals of five minutes and during sampling, one sensor reading is taken per second for a period of 16 seconds. Thus depth sensors generate a predictable data flow rate of 80 bytes at intervals of five minutes which must be relayed from 15 sensor nodes to a single gateway. This data has no significant latency requirements and the bandwidth requirements of the data flow can easily be met by low power, medium range networking hardware such as a 433MHz serial radio [4].

In-network computational flood modeling Our system also supports in-network computational flood modeling, which allows the system to provide flood warning functionality without the necessity of connection to off-site computational facilities [1]. The required per-node computation requires data (this is a sequence of sensor readings, along with predictions from the computation on the remote node) from a small number of other nodes. The flood models can tolerate latency of multiple seconds and require a maximum throughput of no more than 10kbps [5]. As with transmission of sensor readings, these requirements can be met by low power radio hardware.

Image-based flow measurement Our system also supports image-based flow measurement. This is an emerging technique which uses cheap off the shelf digital cameras to measure flow rates [1]. Image-based flow measurement requires the dissemination of multiple high resolution images among sensor nodes which must be distributed in a timely fashion. This requires up to 1MBPS of bandwidth and thus cannot be supported by low power radio hardware.

2.2 Computation Requirements

Each of the three identified areas of system functionality also has distinct computational requirements. Transmission of sensor readings requires very little computational power and can easily be supported on a low power microcontroller such as the Atmel ATMEGA128 MCU [6]. In-network computational flood modeling and image-based flow measurement require significantly greater computational power than is available on low power microcontrollers. However, these processes can be executed in a timely fashion on more powerful embedded processors such as the Intel PXA255 [3].

The requirements of each area of functionality are summarized in table 1 below.

Table 1 – Data Flow and Computation Requirements

	Computation	*Data*
Sensor Transmission	LOW	LOW
On-site Flood Modeling	HIGH	LOW
Image Based Flow Analysis	HIGH	HIGH

The following section describes how the GridStix2 platform uses adaptation to effectively manage the heterogeneous computational and data-flow requirements of our system.

3. The GridStix2 Platform

The original GridStix platform is described in detail elsewhere [1]. In brief, each GridStix v1 node consists of a GumStix embedded computer with a 400 MHz Intel XScale PXA255 CPU, 64Mb of RAM and 16Mb of flash memory and 802.11b [3]. Each node runs the above mentioned Open Overlays middleware. This is a light-weight component based WSN software platform that supports run-time reconfiguration [2] and provides rich networking support from the MAC level to the application level. Unfortunately, while GridStix v1 supported the requirements outlined in section 2, it did so at the expense of high power consumption which significantly reduced the duration of deployments.

The GridStix2 platform implements a 'low-power' personality which allows the node to support power-efficient transmission of sensor readings during quiescent conditions, and also support for more complex applications as demanded by environmental conditions. As discussed, high computational and networking performance are only required for in-network computational flood modeling and image-based flow analysis, which only execute during flood conditions. The hardware and software extensions which comprise GridStix2 are described in sections 3.1 and 3.2 respectively.

3.1 The GridStix2 Hardware Platform

GridStix2 adds a custom-built low power mote platform to the original GridStix platform (the 'high-performance' personality). The mote platform comprises an Atmel ATMEGA128 [5] with 4KB RAM, low power 433MHz radio module [4], 512KB flash memory and battery backed real-time clock. The low power mote is linked to the GridStix via the I2C bus and can operate in either 'master' or 'slave' mode. In slave mode, the low-power mote acts as an expansion board for the GridStix, adding low power radio, flash memory and real-time clock. In master mode, the GridStix is deactivated and the low-power mote implements basic wireless sensor network functionality.

3.2 The GridStix2 Software Platform

The low power mote implements basic sensing and networking functionality which may be easily configured from the 'high-performance' personality.

In terms of *sensing*, developers can schedule the logging and transmission of sensor readings. In addition, they may specify simple alarm conditions which are used to activate the high-power personality. For example, they may specify that if depth increases at a certain rate (i.e., flooding may be imminent) then the high-performance personality should be activated.

In terms of *networking*, the low-power mote provides a self organizing and self healing spanning tree overlay implementation suitable for dissemination of sensor readings from many sensor nodes to a single gateway node. The structure of this spanning tree may also be configured from the high-performance personality to best suit environmental conditions as discussed in [2].

Configuration of the low power personality is achieved by writing to the low power mote's flash memory, which is accessible via the shared I2C bus. Configuration functionality is itself wrapped in a standard Open Overlays component framework [2], allowing developers to configure the low-power personality using standard Open Overlays interfaces. Similarly, in slave mode, developers have access to peripherals using standard Open Overlays interfaces.

3.3 Exploiting Split Personalities

The two personalities are appropriately exploited by matching the behaviour of the system to best suit available resources and conditions. For example, during quiescent conditions, when water depth and flow-rate are within normal bounds, there is no benefit in running flood modeling and image-based flow measurement. During such conditions, the GridStix2 nodes enter the low-power personality to conserve battery life. However, when environmental conditions indicate that flooding may occur, GridStix2 enter the high-performance personality (as triggered by the low pwer mote), thereby providing sufficient computational and networking support for in-network computational flood modeling and image-based flow measurement. Similarly, if mote battery levels run low, the system can scale back to the low-power personality, avoiding total system failure.

4. Deployment and Initial Evaluation

Between 2005 and 2007 the flood warning and prediction system was deployed on the River Ribble in North West England. Fifteen GridStix were deployed along a 3Km stretch of river. Each node was connected to a pressure-based depth sensor. One node was connected to an ultrasound-based flow monitor and digital imaging equipment and one node was equipped with a GSM uplink. In order to meet the range requirements imposed by the site, high gain 90° directional antennas were used.

We are currently in the process of re-deploying a similar network of GridStix2 on the River Dee in North Wales. We expect deployment to be completed by late spring 2008. While this deployment in ongoing we have evaluated the GridStix2 platform in the lab in terms of functionality and power consumption. Specifically, figure 1 compares the power consumption of the high-performance and low-power GridStix2 personalities while listening (L) and transmitting (T). The power draw of the radio is shown in black, while other power draw (i.e. CPU, memory etc.) is shown in grey.

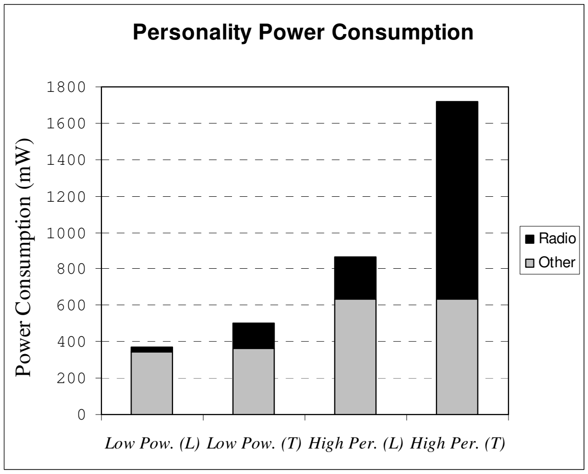

Figure 1 – Power Consumption of Each Personality

As can be seen, the low-power personality consumes significantly less power than the high-performance personality. Power consumption is 57% lower while listening, and 71% lower during message transmission. This allows system deployments to be radically extended during periods when power is scarce or the features offered by the high-performance personality are not needed. Informal testing has shown that the 433MHz radio module [4] achieves similar range to the 802.11b radios [3] using an omni-directional antenna: 250M for the former and 300M for the latter.

5. Related Work

A number of projects have tackled the problem of providing flood support using WSN technology. The Floodnet project [9] uses a platform similar to GridStix v1 (an XScale CPU with 802.11b networking) to implement flood monitoring on a tidal river in South East England, though this system supports neither in-network computation nor computationally demanding sensors. The Hydrowatch project uses low power Telos motes to implement river monitoring in the Sierra Mountains in northern California. This project is currently focused on supporting the planning of micro-solar installations to support environmental monitoring WSNs. The mote platform used in Hydrowatch does not have sufficient resources to support in-network computation or image-based flow measurement [10].

The approach of separating the concerns of networking and sensing from application processing is also used in Dust Networks 'Smart Mesh' products [6] which provide reliable mesh networking and basic sensing functionality with the expectation that developers will add their own application processor; while the GainSpan GS1010 [7] provides separate application and network processors. Such a separation allows power-hungry application processors to be activated only when needed. We are currently evaluating the Dust Networks SmartMesh-XT 2135 alongside the GridStix2 low-power personality for providing low power networking support.

The RUNES project [8] has looked at implementing a middleware similar to Open Overlays on low power microcontrollers. Such a platform would allow significantly greater flexibility for the 'low power' GridStix2 personality.

6. Conclusions and Future Research

We have described a WSN node architecture that employs two personalities: a low power one and high performance one. We have also discussed in outline how the new capabilities enabled by this platform are controlled and managed by our Open Overlays middleware. Based on this combination of hardware and software, we expect that our flood system deployments can survive for significantly longer durations while offering the same or greater functionality.

Our future research will initially focus upon deploying the GridStix2 flood monitoring platform at the new site on the River Dee. The GridStix2 platform will then be evaluated more thoroughly in this deployed environment. As well as evaluating low-level system functionality, we are particularly interested in evaluating the role of adaptation in improving system performance. This includes adaptation between personalities as discussed in this paper, as well as more fine grained adaptation such as the adaptation of networking behaviour [2].

7. References

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

[2] Grace P., Hughes D., Porter B., Blair G., Coulson G., Taiani F., Experiences with Open Overlays: A Middleware Approach to Network Heterogeneity, in the proceedings of the European Conference on Computer Systems (EuroSys'08), Glasgow, UK, March 2008.

[3] GumStix Embedded Computing Platform, http://gumstix.com/, February 2008.

[4] Low Power Radio Solutions, Easy Radio Data Sheet, available online at: www.radio-modules.com/products/radiodatamods/datasheets/er-datasheet-2.3-sept-05.pdf, 2005.

[5] Atmel Corporation, ATMEGA128 Microcontroller Data Sheet, available online at: http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf, 2008.

[6] GainSpan, GS1010 Ultra Low-Power Wireless System-On-Chip Data Sheet, available online at: http://www.gainspan.com/Docs/GS1010_SoC_Product_Brief.pdf, 2008.

[7] Dust Networks, SmartMesh-XT 2135 Data Sheet, available online at: http://www.dustnetworks.com/docs/M2135.pdf, 2008.

[8] Costa, P., Coulson, G., Mascolo, C., Mottola, L., Picco, G.P., Zachariadis, S., "A Reconfigurable Component-based Middleware for Networked Embedded Systems", International Journal of Wireless Information Networks, Vol. 14, No. 2, pp 149-162, June 2007.

[9] DeRoure D., "Improving Flood Warning times using Pervasive and Grid Computing", available online at: http://envisense.org/floodnet/ingenia/ingenia.htm.

[10] Taneja J., Jeong J., and Culler D., Design, Modeling, and Capacity Planning for Micro-Solar Power Sensor Networks, in the proceedings of the Seventh International Conference on Information Processing in Sensor Networks (SPOTS '08), April 2008.

[11] Rahimi M., Baer R., Iroezi O., Warrior J., Estrin D., Srivastava m., Cyclops: in situ image sensing and interpretation in wireless sensor networks, in the proceedings of the 3rd Conference On Embedded Networked Sensor Systems (Sensys 2005), San Diego, California, USA, pp 192 – 204.