Supporting IPv6 Interaction with Wireless Sensor Networks Using NP++

Matthew Jakeman[1], Danny Hughes[1], Geoff Coulson[1], Gordon Blair[1], Steve Pink[1]
and Kevin Lee[2]

[1] Computing Department, InfoLab 21, Lancaster University, Lancaster, UK. LA1 4WA
m.jakeman@lancaster.ac.uk | {danny, geoff, gordon, pink}@comp.lancs.ac.uk

[2] School of Computer Science, University of Manchester, Manchester, UK. M13 9PL
klee@cs.man.ac.uk

Abstract. There is growing interest in exploiting standard Internet protocols such as IPv6 in wireless sensor networks. Support for IPv6 has the potential to facilitate application development, increase the flexibility of sensor node interaction, and better integrate sensor nodes into the 'Internet of things'. Unfortunately, IPv6 is poorly suited for resource-constrained environments and is particularly wasteful for typical wireless sensor network data flows. This paper presents NP++, a flexible network protocol that provides efficient mapping of IPv6 onto heterogeneous physical networks. The performance of NP++ is evaluated in the context of a deployed WSN-based flood monitoring and warning system.

Keywords: Wireless Sensor Networks, IPv6, IPHC, TSMP.

1 Introduction

The vision of nodes in wireless sensor networks (WSNs) as first class Internet entities - a part of the 'Internet of things' [1] - holds significant promise in terms of facilitating access to WSN data, facilitating the development of WSN applications, and supporting tighter integration between modeling/control facilities and WSN deployments.

In order to realize this vision, sensor nodes must interoperate with IPv6 [2]. Unfortunately, IPv6 is poorly suited to resource-constrained environments such as wireless sensor networks, where power, bandwidth and computational resources are extremely scarce. Specifically IPv6 is poorly suited for supporting typical WSN data flows as it introduces significant overhead due to its large packet headers. Existing gateway-based approaches to WSN/Internet integration limit the flexibility of interaction with sensor nodes via the Internet and increase the burden on developers who must develop using both WSN-specific protocols as well as standard IP-based protocols. NP++ addresses these problems using a layer of indirection which allows developers to write applications using a standard IPv6 *logical specification*, which is transparently mapped onto optimised *physical specifications* tailored to suit different network media and environments.

This paper introduces NP++, a flexible network protocol that uses a layer of indirection to efficiently map IPv6 onto heterogeneous network media, while offering a consistent representation of IPv6 to the upper layers of the network stack and allowing WSN motes to be addressed as standard IPv6 nodes. In the paper we evaluate NP++ in the context of a multi-network flood modeling and warning scenario [3]. In this scenario, which is currently deployed and operational, NP++ allows direct IPv6 interaction with sensor nodes, while optimizing the protocol's performance to suit each of the three network media employed. In a general sense, this illustrates the power of NP++ for integrating diverse network technologies while offering a common interface to application developers.

The remainder of this paper is structured as follows: Section 2 describes our 'GridStix' flood monitoring platform and deployment environment. Section 3 introduces the NP++ protocol. Section 4 describes the physical specifications that are used to optimize NP++ for the different media types used in this scenario. Section 5 provides an initial evaluation. Section 6 places NP++ in the context of related work. Finally, Section 7 discusses avenues of future research and concludes.

2 GridStix Flood Monitoring Platform

Each GridStix node is based on the Gumstix [4] embedded computing platform, so named as each device is roughly the same size as a pack of gum. Despite their small-size, each of these devices is equipped with a 400 MHz Intel XScale PXA255 CPU, 64Mb of RAM and 16MB of flash memory. These hardware resources support the execution of a standard Linux kernel and Java Virtual Machine along with our Open Overlays WSN middleware [15]. In the field, each GridStix is connected to a variety of sensors including pressure-based depth sensors to monitor water levels, conductivity sensors to monitor pollution, and digital cameras which are used to support image-based flow measurement [16]. In terms of networking, each device is equipped with a Dust Networks mote [12], which acts as a low-power 802.15.4 time synchronized network interface. Furthermore, a small number of the devices are equipped with a GPRS uplink and DVB satellite downlink for transmitting and receiving data from off-site. The devices are powered by solar arrays of four 15CM^2 2.5W solar panels in combination with a 12V 7AH battery, which ensures reliable operation even during the dark British winter months. To minimize the effects of harsh weather,

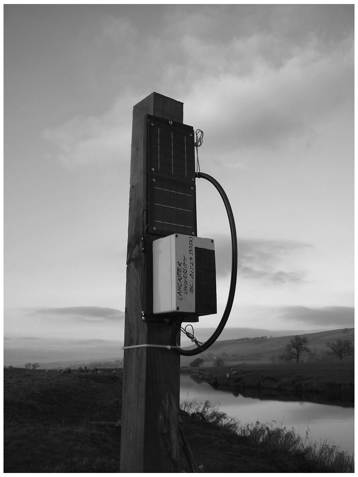

Figure 1 – A Deployed GridStix

flooding, vandalism etc., the devices are housed in durable, water-tight containers, and all external wiring is enclosed in resilient piping. A first-generation GridStix node is shown in Figure 1 (current versions have a larger solar array and more resilient cable-housing).

Between 2005 and 2007, a network of 15 GridStix was deployed along a 3KM stretch of the River Ribble in North West England, and a similar deployment is currently being rolled out on the River Dee in North Wales.

3 NP++

The problem of integrating networked embedded devices, such as the nodes in a wireless sensor network, with the Internet has typically been tackled through the use of specialized gateways, as in the Arch Rock Primer Pack [14]. A gateway-based approach supports the external addressing of sensor nodes, while allowing nodes within the sensor network to use specialized protocols [13], [5] that are specifically designed for dynamic and resource constrained WSN environments. Unfortunately, a gateway-based approach has two major disadvantages. Firstly, it reduces the flexibility of interaction between sensor nodes and other Internet devices. Secondly, it increases the burden on developers of end-to-end WSN systems, who must develop using both WSN-specific protocols as well as standard IP-based protocols.

NP++ addresses these problems by using a layer of indirection which separates the *logical specification* of a network protocol, as seen by developers, from the underlying *physical specification* which defines the control information and data that are actually transmitted on the media. This approach reduces the burden on developers, who may develop applications using a single logical specification, while the underlying physical specification is transparently modified to suit different network environments. This transformation is accomplished through a *mapping function*, which translates the logical specification into one of a larger number of physical specifications. In this paper, we specifically focus on the ability of NP++ to facilitate the interoperation of WSNs with the Internet by offering a common logical specification (IPv6) to developers while at the same time tailoring the performance of this protocol to suit the underlying network using per-media physical specifications.

The physical specifications which are presented and evaluated in this paper represent just a few examples of how the use of different physical mappings can optimize NP++ for different network media. Additional examples include providing support for label switching, field ordering and error detection. In all cases, NP++ offers developers a consistent logical specification.

3.1 Naming, Addressing and Routing

NP++ uses IPv6 as its logical specification as well as its default physical specification. As naming, addressing and routing functionality are inherited from IPv6, each NP++ node has an IPv6 address along with addresses for each physical specification that requires one (e.g. the 8 bit address used in TSMP [13]). For each node in its routing table, NP++ maintains addresses for all physical specifications along with the node's logical address. When required to route a message to a given logical address, NP++ scans its routing table for a match and then uses the associated physical address to create and forward a packet using the appropriate physical specification.

3.2 Physical Mappings and Conflict Resolution

Each NP++ node maintains a list of available physical mappings, which are associated with links in order of priority. When a node joins a network, the node negotiates with its neighbours on which mapping to use. This negotiation is performed on a per-link basis and the highest priority mapping known to both nodes is selected as the physical specification. This priority-based mechanism allows the choice of just one mapping per link and thus avoids the conflicts that can arise in IP (e.g. attempting to use Network Address Translation (NAT) on an encrypted header). In order to ensure that NP++ nodes can always communicate, NP++ requires that each node also implement the default IPv6 physical specification.

In our GridStix scenario, three physical mappings are used to optimize the performance of IPv6 for each of the network media used in the scenario: the GSM uplink, DVB satellite downlink and low power on-site TSMP networking. The physical specifications used to achieve this are described in section 4.

4 Physical Specifications

As described, distinct physical specifications are used to optimize NP++ for each of the 3 network media used in our scenario. The mappings are as follows:

GSM Uplink: For this bandwidth-constrained link type, NP++ uses IP Header Compression (IPHC) [5] as its physical specification. This is capable of compressing both IPv4 and IPv6 headers. IPHC on average reduces an IPv6 packet header from 40 bytes to just 4 bytes. This leads to significant bandwidth savings and also reduces packet loss (as packet loss tends to increase as a function of packet size [6]). While the use of IPHC does not allow for enhanced IPv6 features such as extensions headers and security, the highly resource constrained nature of the GSM uplink renders these features infeasible costly.

DVB Satellite Downlink: For this high performance satellite downlink NP++ uses the default IPv6 physical mapping. This allows the features of IPv6 such as optional extensions headers, support for mobility and enhanced security to be fully exploited. Furthermore, as the satellite downlink offers relatively high throughput and low loss the overhead incurred by running IPv6 is quite acceptable.

On-site TSMP Network: In the case of the on-site 802.15.4 network, NP++ maps onto a specially developed protocol known as Peer-to-peer Time Synchronized Mesh

Protocol (P-TSMP). P-TSMP builds on the core TSMP protocol [13], which is a commercial time-synchronized protocol for WSNs implemented by the Dust Networks motes [12] used in our scenario. While TSMP is efficient and has very low power requirements, it only supports the transmission of messages between motes and a centralised manager. P-TSMP extends this by adding support for peer-to-peer messaging between motes, and for network-wide broadcast. To support these features, simple routing functionality has been added to the Dust manager, and the Dust packet format has been extended as shown in Figure 2.

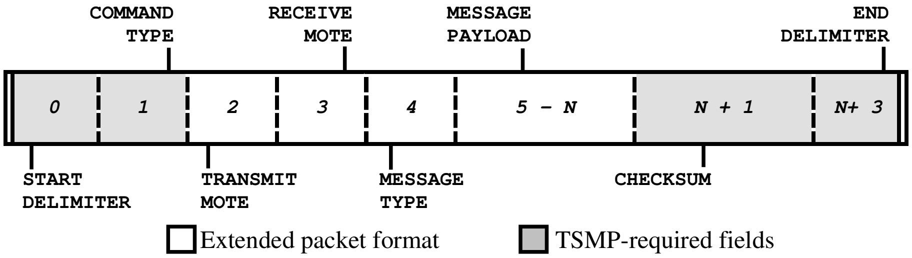

Figure 2 – P-TSMP Packet Structure

Of those fields not required by the Dust mote's implementation of TSMP, the P-TSMP packet structure is as follows:

- *Byte 2* - Address of the originating Dust mote.
- *Byte 3* - Address of the destination Dust mote.
- *Byte 4* - Message type: The extended packet format supports four message types: reliable/unreliable mote-mote messaging and reliable/unreliable broadcast.

P-TSMP thus provides flexible and low power mote-to-mote messaging with an overhead of just 8 bytes per packet. This reduction in packet size compared to IPv6 is expected to have a number of benefits including better compatibility with the smaller frame sizes of 802.15.4, increased throughput, and reduced packet loss. Furthermore, while it may appear that IPHC (described in section 4.1) has even lower packet overhead, the underlying TSMP implementation offered by the Dust motes results in an additional overhead of 5 bytes per packet (as shown in figure 2). Thus, running IPHC over TSMP would result in an overhead of 9 bytes while offering neither broadcast nor reliable routing. As with the GSM connection, the use of a low overhead physical mapping prevents use of enhanced IPv6 features, though the small frame size of the TSMP network makes this highly infeasible.

5 Evaluation

The below evaluation was performed using the GridStix implementation of NP++ at the gateway of our flood monitoring WSN. As described previously, the gateway node is connected to a GSM uplink and a DVB satellite downlink [3] as well as the on-site TSMP network. The below experiments were conducted under 'typical' gateway conditions, with satellite signal strength at ‘good’, GSM signal strength at 67%, and in fair weather. The on-site TSMP network was configured with its default settings: 31.25ms time-slots and a frame length of 200 slots.

Section 5.1 evaluates the extent to which NP++ can optimize IPv6 for the GSM link by reducing packet size and hence loss; Section 5.2 investigates how NP++ can optimize IPv6 for the on-site TSMP network; and Section 5.3 discusses the benefits of a unified logical specification. Throughout, optimized physical mappings are compared to the IPv6 physical specification which is used on the higher performance DVB satellite downlink.

5.1 Optimizing NP++ for GSM Using an IPHC Physical Mapping

The flood monitoring system generates a predictable upstream data flow during normal operation due to its periodic reporting of depth and conductivity readings [3]. This data flow consists of 100 bytes of data per node and is transmitted at intervals of 1 minute. When NP++ is configured to use the IPHC [5] physical mapping, the header size for TCP traffic such as sensor readings is reduced from 40 bytes to just 4 bytes - a reduction of 90%.

As the packet payloads generated during the reporting of sensor readings are relatively small (100 bytes), the IPHC physical specification results in a significant reduction in total packet size: from 140 bytes using an IPv6 physical mapping to just 104 bytes using an IPHC physical mapping (the effects of this reduction on packet loss and power consumption are explored in section 5.2 and 5.3 respectively).

In the context of the DVB satellite down-link, such optimisations are unnecessary due to the higher bandwidth and better quality of service offered by this link. Furthermore, the use of an IPv6 physical mapping allows the full flexibility of IPv6 networking to be exploited.

Research has shown that packet loss on radio links is strongly correlated with packet size [6]. We therefore specifically analyzed the relationship between packet size and loss on our GSM uplink and DVB downlink using iPerf [7]. iPerf was configured to send long sequences of UDP datagrams in sizes ranging from 10bytes to 160bytes (at intervals of 10 bytes) and the rate of packet loss was recorded. Each experiment was repeated 10 times and the results logged. As can be seen from figure 3, there is indeed a strong correlation between packet size and packet loss.

Considering the reduction in packet size achieved using IPHC on the GSM uplink (from 140 bytes for IPv6 to 104 bytes for IPHC), we would expect packet loss to be significantly reduced: from 6.4% using IPv6 to 4.7% using IPHC - a net reduction of in packet loss of 27%. Conversely on the DVB satellite downlink, packet size has a relatively low impact on packet loss and thus the rich features of the IPv6 physical mapping may be exploited at minimal cost (Reducing packet loss also has significant implications for power consumption on the GSM connection, as discussed in Section 5.1.1).

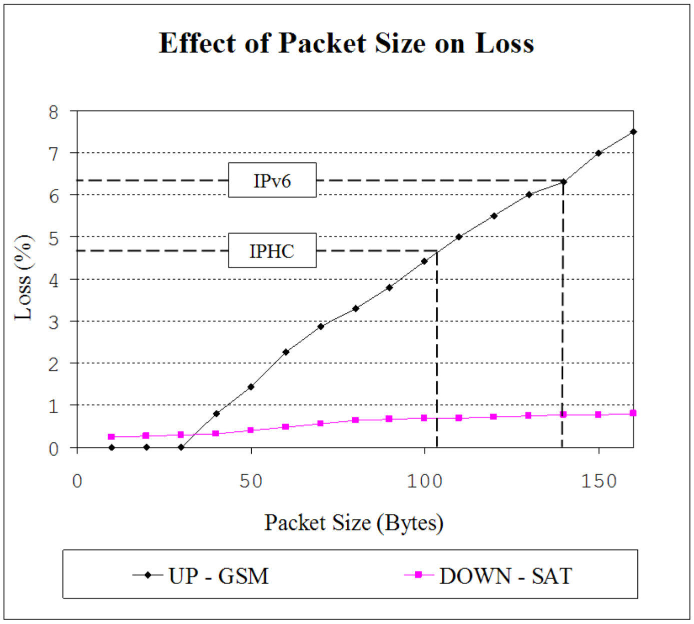

Figure 3 – Effect of Packet Size on Packet Loss on GSM uplink

5.1.1. Implications for Power Consumption

Power consumption is a critical factor in any WSN application and the IPHC physical mapping is expected to have a significant impact on the power consumption of the GSM connection (as the lower volume of data being transmitted means that networking hardware is not active for as long). The GSM uplink and Satellite downlink may remain physically switched on, but they will use less power as they are not active. Average power consumption figures for the GSM uplink and the DVB satellite downlink are provided in Table 1 below.

As can be seen from the table, the GSM uplink consumes significantly less power when the connection is inactive compared to when the connection is active (under 250mA compared to over 400mA). Reducing the volume of data that must be transmitted, and in turn the time period that the GSM connection is active, thus leads to significant power savings. In addition, as fewer packets are lost using the IPHC physical mapping (see section 5.2), power consumption due to packet retransmission is also minimized.

Table 1 – Power Consumption

	GSM	*SAT*
Inactive	248 mA	806 mA
TX / RX	424 / 416 mA	849 mA

In the case of the DVB satellite downlink, there is little difference in power consumption whether the link is inactive or receiving (806mA v 849mA). This further supports our argument that an IPv6 physical mapping can be used to provide rich networking support on the satellite downlink at minimal cost in terms of packet loss, or power consumption.

5.2 Optimizing NP++ for 802.15.4 Using a TSMP Physical Mapping

Section 5.1 highlighted the benefits of using an IPHC header to reduce loss on the GSM uplink of the site gateway. This section will now investigate the benefits of using a P-TSMP mapping to minimize the number of packets transmitted via the on-site 802.15.4 network.

Each TSMP node [13] is allocated a *time-slot* (31.25ms) for packet transmission. A complete sequence of time-slots is referred to as a *frame*. By default each TSMP frame contains 200 time-slots (giving a frame length of 6,250ms). While strict scheduling makes TSMP extremely power-efficient, it results in high latency, as in the default configuration, motes transmit only one packet in each network frame. Thus, it is particularly critical that packet overhead be minimized to prevent the transmission of unnecessary packets, each of which has a high lag due to the wait for an available time-slot.

Consider the encapsulation and transmission of 100 byte sensor readings over TSMP using IPv6. TSMP packets have a maximum size of 80 bytes, of which the IPv6 header consumes 40 bytes. Thus the transmission of 100 bytes of sensor data using IPv6 requires the transmission of three IPv6 encapsulated packets over TSMP. In comparison, encapsulating sensor data using our P-TSMP packet format consumes just 3 additional bytes per packet, and thus the same 100 bytes of sensor data can be transmitted using two P-TSMP packets. This significantly reduces the latency of data transmission, from 18,750ms .to 12,500ms – a reduction of 6,250ms or 33%.

While this latency improvement derives from the data-flow pattern of this specific application, these results serve to illustrate the importance of reducing packet overhead in networks with small frame sizes or high lag. As with the IPHC physical mapping, which is used for the GSM uplink, reduction in packet size is expected to result in a significant reduction in power consumption. This is discussed in section 5.2.1.

5.2.1. Implications for Power Consumption

As previously discussed, power consumption is a critical factor in all WSN applications, and the reduction in packet transmissions that can be achieved using P-TSMP will lead to significant power savings for any application (as radios are activated less frequently). Power consumption data for the Dust motes is shown in Table 2 below [12].

Table 2 – Power Consumption of Dust Motes

	Average	*Maximum*
Transmit	50mA	90mA
Receive	22mA	30mA
Sleep	10µA	15µA

As described in the previous section, based on the data distribution requirements of this application, transmitting sensor data using our P–TSMP physical mapping, reduces the number of packets that are transmitted during each reporting period by one third. Assuming the maximum power consumption for each state enumerated in Table 2, the default IPv6 physical mapping will result in an average power draw of 3mA, while the using the P-TSMP physical mapping results in a power draw of 2.1mA. Thus the use of the P-TSMP physical mapping is expected to increase battery life by approximately 30%.

5.3 Benefits of a Unified Logical Representation

Sections 5.1 and 5.2 have shown that through the use of specific physical mappings it is possible to optimize NP++ for various network media. In the case of our GSM uplink, the use of an IPHC physical mapping significantly reduces average packet sizes and thus loss. In the case of the on-site 802.15.4 network, the use of a P-TSMP physical mapping significantly reduces the number of transmissions and therefore power consumption.

Perhaps an even greater advantage is that, when using NP++, the developer is shielded from the complexity inherent in using multiple network protocols tailored for different environments. The application developer simply addresses nodes using standard IPv6, and NP++ efficiently maps this onto the underlying network.

By separating the physical and logical representations of a network protocol, NP++ allows the most recent advances in network protocols and network media to be exploited with no effort from application developers. Moreover, as protocol developers are expected to develop physical mappings, systems built using NP++ are expected to be of higher quality than those where application developers must also be concerned with the low-level details of different protocol implementations.

6. Related Work

The Flexible Interconnecting Protocol (FLIP) [8] provides support for heterogeneous devices and network links using a ‘meta-header’ which defines the fields present in the packet header. While this design is somewhat flexible, it is limited to field suppression and thus, unlike NP++, it is unable to provide additional features such as label switching, field ordering or error detection.

Braden et al [9] introduce a Role-Based Architecture (RBA) which does not use the OSI network stack. Instead, the system uses a 'heap' of role headers which interact based upon defined rules. These headers are also persistent, such that downstream nodes are aware of previous packet treatment. While this allows for a comparable level of flexibility to NP++, it does so at the cost of considerable overhead, making it unsuitable for resource-constrained environments such as WSNs.

IPv6 over Low Power Wireless Personal Area Networks (6LowPAN) [10] allows IPv6 to be supported on highly resource-constrained networks. Specifically, 6LowPAN allows IPv6 to be supported on 802.15.4 frames which have a maximum transmission unit of just 127 octets. 6LowPAN also implements header compression, including compression of node addresses. While this approach allows easy interoperation between WSN and Internet devices, the approach is not as flexible as NP++, essentially offering a single physical mapping optimised for 802.15.4.

7. Conclusions and Future Research

The 'real-world' evaluation presented in this paper suggests that significant benefits may be achieved by using different physical specifications to tailor the performance of a unified network protocol to fit the requirements of heterogeneous data flows and network media. For example: in the case of our GSM uplink, the IPHC physical specification provides significant bandwidth and power savings while reducing packet loss. Conversely, in the case of the DVB satellite downlink the more flexible IPv6 physical specification can be used while incurring minimal cost in terms of power consumption and packet loss.

Our future research will focus on deploying our flood monitoring network as a new and larger site on the River Dee. Once operational, this will enable a more complete evaluation of our proposed NP++-based approach to WSN networking. For example, we plan to log network performance and power consumption over periods of days in order to take into account varying environmental conditions. We will also investigate the benefits of NP++ in terms of providing rich networking support between WSN monitoring facilities and off-site modeling and control facilities. Finally, we intend to investigate the potential of integrating NP++ with our Open Overlays [11] WSN middleware platform.

References

[1] The International Telecommunication Union, Internet Report: The Internet of Things, available online at: http://www.itu.int/dms_pub/itu-s/opb/pol/S-POL-IR.IT-2005-SUM-PDF-E.pdf

[2] The Internet Society Network Working Group, RFC 2460 - Internet Protocol, Version 6 (IPv6) specification, available online at: http://www.faqs.org/rfcs/rfc2460.html

[3] 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.

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

[5] Degermark M., Nordgren B., Pink S., RFC 2507 - IP Header Compression (IPHC) RFC, available online at: http://www.ietf.org/rfc/rfc2507.txt

[6] Korhonen J., Wang Y., Effect of packet size on loss rate and delay in wireless links, in the proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '05), vol. 3, pp. 1608-1613, New Orleans, USA, March 2005.

[7] NLANR/DAST, iPerf Network Measurement Tool, http://sourceforge.net/projects/iperf

[8] I. Solis, K. Obraczka, and J. Marcos. FLIP: a flexible protocol for efficient communication between heterogeneous devices. In ISCC, pages 100–106. IEEE Computer Society, 2001.

[9] Braden R., Faber T., Handley M., From Protocol Stack to Protocol Heap: Role-Based Architecture, in Computer Communication Review, 33(1):17–22, 2003.

[10] Montenegro G., Kushalnagar N., Hui J., Culler D., RFC 4944 - IPv6 over Low Power Wireless Personal Area Networks (6LowPAN) RFC, available online at: http://www.ietf.org/rfc/rfc4944.txt

[11] 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.

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

[13] Dust Networks, Technical Overview of TSMP, white paper, available online at: http://www.dustnetworks.com/docs/TSMP_Whitepaper.pdf

[14] Arch Rock, Primer Pack: http://www.archrock.com/products/primer_pack_ip.php

[15] 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), March 2008.

[16] Creutin, J.D., Muste, M., Bradley, A.A., Kim, S.C., River Gauging using PIV Techniques: A Proof of Concept on The Iowa River, A. Kruger, Journal of Hydrology 277, 182-194, 2003.