IPSA: a novel architecture design for integrating IP and sensor networks

Min Chen*

Department of Electrical and Computer Engineering,
University of British Columbia,
Vancouver BC V6T 1Z4, Canada
E-mail: minchen@ece.ubc.ca
*Corresponding author

Shiwen Mao

Department of Electrical and Computer Engineering
Auburn University,
200 Broun Hall, Auburn, AL 36849-5201, USA
E-mail: smao@ieee.org

Yang Xiao

Department of Computer Science,
University of Alabama,
Tuscaloosa, AL 35487, USA
E-mail: yangxiao@ieee.org

Ming Li

Department of Computer Science,
California State University,
Fresno, CA 93740, USA
E-mail: mingli@csufresno.edu

Victor C.M. Leung

Department of Electrical and Computer Engineering
University of British Columbia,
Vancouver BC V6T 1Z4, Canada
E-mail: vleung@ece.ubc.ca

Abstract: Recently, there is a growing interest in the design, development and deployment of sensor systems for applications of high-level inference, which leads to an increasing demand on connecting Internet Protocol (IP) network users to wireless sensor networks and accessing the available services and applications. In this paper, we first identify key requirements of designing an efficient and flexible architecture for integrating IP and sensor networks. Based on the requirements, we outline an IP and Sensor Network Integration Architecture (IPSA), which provides IP mobility support and a universal interface for IP mobile users to access sensor networks while considering application-specific requirements. With on-demand processing code assigned to the middleware layer of an IP mobile user, IPSA provides the flexibility for enabling diverse applications and manages network resources efficiently without additional requirements on sensor nodes, except for the limited additional hardware requirement at IP mobile nodes.

Keywords: sensor networks; integration architecture; internet protocol.

Reference to this paper should be made as follows: Chen, M., Mao, S., Xiao, Y., Li, M. and Leung, V.C.M. (2009) 'IPSA: a novel architecture design for integrating IP and sensor networks', *Int. J. Sensor Networks*, Vol. 5, No. 1, pp.48–57.

Biographical notes: Min Chen is a Post-Doctoral Fellow in the Department of Electrical and Computer Engineering, University of British Columbia, Canada since 2006. Before joining UBC, he worked as a Post-Doctoral Fellow in School of Computer Science and Engineering at Seoul National University for one and half years. He is the recipient of the Best Paper Runner-up Award in ICST QShine'08. He is a co-chair of PCSI'09. He has served technical programme committees of several international conferences such as WRECOM'07, ICCCN'08 and ChinaCom'08. He received a PhD degree in electronics engineering from South China University of Technology in 2004.

Shiwen Mao is an Assistant Professor in the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL. He was a research scientist at Virginia Tech, Blacksburg, VA, USA from 2004 to 2006. He received the PhD degree in electrical and computer engineering from Polytechnic University, Brooklyn, NY, USA in 2004. His research interests include cross-layer optimisation in multihop wireless networks, cognitive networks and multimedia communications. He received the 2004 IEEE Communications Society Leonard G. Abraham Prize in the field of communications systems and co-authored textbook *TCP/IP Essentials: A Lab-Based Approach* (Cambridge University Press, 2004).

Yang Xiao is currently with Department of Computer Science at The University of Alabama. His research areas are security, telemedicine, sensor networks and wireless networks. He has published more than 300 papers in major journals (more than 60 in various IEEE journals/magazines), refereed conference proceedings, book chapters.

Ming Li has been a faculty in the Department of Computer Science, California State University, Fresno, since August 2006. He received his MS and PhD degrees in Computer Science from The University of Texas at Dallas in 2001 and 2006, respectively. His research interests include QoS strategies for mobile ad-hoc networks and multimedia streaming over wireless networks. He is the recipient of the Best Student Paper Award in IEEE WoNGeN'05. He is the track co-chair of 'Multimedia Networking' in ICCCN'08.

Victor C.M. Leung is a Professor of Electrical and Computer Engineering and holder of the TELUS Mobility Research Chair at The University of British Columbia. He is a Fellow of IEEE and serves on the editorial boards of the *IEEE Transactions on Wireless Communications and Transactions on Computers*. His research work focuses on wireless networks and mobile systems.

1 Introduction

Wireless Sensor Networks (WSNs) (Al-Karaki and Kamal, 2004) have attracted remarkable attention from the research community, as driven by a wealth of theoretical and practical challenges and an increasing number of practical civilian and military applications. Most of the existing work focuses on the WSN itself, with many algorithms/protocols proposed for solving various problems such as energy efficiency, load balancing, coverage and data dissemination. However, on the other side of the story, effective integration of WSNs with the IP network provides several key technical advantages including significant improvement of ubiquitous access of physical information and involvement of people (e.g. PDAs and/or mobile phone users) in the process of information gathering and sharing. In such an integration architecture, WSNs will collect environmental data on demand to facilitate mobile users to access and exploit various WSN services and applications. We believe that such integration is an important research problem and should be addressed to leverage and harvest the power of WSNs.

The integration of WSNs with the IP networks poses many challenges. Seamlessly, integration of heterogeneous terminals (e.g. sensors, mobile devices and internet hosts) requires innovative network architectures, data services and communication protocols. Sensors should be able to detect events and forward the information to one or more mobile users operating with the TCP/IP protocols. In an integrated system, mobile users can access sensory environments physically to retrieve sensed data on demand. Such users who carry mobile IP devices (e.g. mobile IP phones, PDAs or portable computers) can be deemed as IP mobile users. When an IP mobile user enters WSNs to collect sensed data, the problem in terms of IP mobility should be addressed. The problem of how to support the IP mobile user to communicate with IP network transparently within the visited sensor networks, has been studied in a few papers, by exploiting a gateway (Marco and Bhaskar, 2003), a new 'bundle layer' (Fall, 2003) or taking an overlay approach (Dunkels et al., 2004; Dai and Han, 2004) (see Section 2). It is worth noting that these existing approaches all require an additional interface layer for addressing and context-aware translation. Considerable additional storage and processing overhead are required for constructing such an interface layer on top of the protocol stack of sensor nodes and/or IP nodes.

Such overhead may not be affordable for memory-constrained sensor nodes. Furthermore, 'one deployment, multiple applications' is an emerging trend in the development of WSNs due to the high cost of deploying numerous sensor nodes over a large region, as well as the application-specific nature of tasking a WSN (Chen et al., 2007a). This trend requires that sensor nodes have various capabilities to handle multiple applications. However, it may be impractical to store all the programs needed to run every possible application in the local memory of embedded sensors, as these devices generally have tight storage constraints.

To address the above challenges, we present a novel integration architecture, termed IPSA in this paper. The main idea is to shift major functionalities of network integration from IP devices and sensor nodes to enhanced sink node(s) as much as possible, thus effectively keeping IP devices compatible with ubiquitous IP networks as well as relieving the processing, communication and storage burdens from sensor nodes. The enhanced sink in IPSA includes the following key functional components: register

manager, processing code manager, pricing manager, inquiry manager and event database. We also propose an extension of IPSA with additional functionalities to broaden the design space of IPSA.

Compared with the existing work (Marco and Bhaskar, 2003; Fall, 2003; Dunkels et al., 2004; Dai and Han, 2004), IPSA has the following four desirable features.

- It supports IP users to access sensory data, either from the IP network or when roaming into the WSN. The latter involves the IP mobility management problem: when an IP mobile user roams into WSNs, he/she may need to deliver the collected information to another IP user within the IP network.
- It allows designed applications in an on-demand fashion while hiding the details of the WSNs and IP networks. Thus, IP mobile users only make use of the services at the sensor networks and are not concerned with the details of how the service is being provided. The use of middleware and designing, corresponding processing codes dynamically for new applications in WSNs has been proved to be an effective method (Chen et al., 2007b) and the introduction of middleware layer gives up the luxury of designing the overlay or 'bundle layer' protocol at either sensor node or IP nodes (Fall, 2003; Dunkels et al., 2004; Dai and Han, 2004).
- It does not require any change to the sensor nodes. The only requirement is that IP mobile users are equipped with an add-on middleware to obtain on-demand services in various WSNs. Thus, IPSA is fully compatible with various WSNs and current IP network architecture and this feature makes IPSA flexible and extensible to interconnect multiple WSNs and IP networks.
- IP mobility management and application-specific sensory data collection are transparent to IP users and sensor nodes. The IP mobile user and IP network do not need to know any operations of WSNs and how this kind of service is provided in detail. The responsibility is shifted to the sink and the processing code which makes IPSA work in an application-oriented manner.

The remainder of this paper is organised as follows. We discuss related work in Section 2. We then present the key functional entities of IPSA in Sections 3 and the IPSA extension in Section 4. We conclude this paper and discuss future directions in Section 5.

2 Related work

The problem of integrating IP networks and WSNs has been investigated in a few works. Marco and Bhaskar (2003) proposed a gateway-based approach, in which gateways with application-level interface are deployed to connect WSNs and IP networks. However, this approach needs an additional hardware component to provide application-level interface for both sensor nodes and IP nodes. Furthermore, it is not flexible and suitable for the application-specific nature of tasking a WSN.

The gateway-based approach is also adopted in delay tolerant networks (Fall, 2003), where a 'bundle layer' is deployed in both IP networks and non-IP network protocol stacks to store and forward packets. However, it may not be feasible to deploy a 'bundle layer' in the memory-const rained embedded sensors.

Dunkels et al. (2004) proposed an overlay-based solution to implement IP protocol stack on sensor nodes, thus enabling IP nodes to task sensor nodes in WSNs via IP addresses. This solution requires that sensor nodes have enough processing capabilities to run IP protocol and is not a cost-effective architecture when we consider the limited power consumption of sensors. Dai and Han (2004) presented a scheme for sensor networks to overlay IP protocol. Each IP node can be deemed as a virtual sensor node by deploying WSN protocol stack on top of TCP/IP. Although IP node can directly communicate with sensor nodes, it increases protocol overhead to IP networks and sacrifices the consistency with current IP-based operation model.

By comparison, this paper exploits capabilities at the network edge (i.e. an edge-based approach) (Mao and Hou, 2007). Mao and Hou (2007) showed that exploiting edge capability provides a new dimension of freedom for WSN and is effective in relieving the processing, communication and storage requirements from sensor nodes.

Cho et al. (2007) present a Sensor and RFID Integration Framework (SARIF). Although several similar architectural components are included in SARIF and in the IPSA support module (see Section 4), our proposed architecture is remarkably different from SARIF since it mainly concerns the integration of RFID networks and sensor networks.

The IP mobility management problem is inherent in such integration architecture. Although mobile IP (Perkins, 2002) provides a viable approach to mobility management, it is originally designed for IP hand-off between two IP networks. How to support seamless hand-off for an IP mobile user at its visiting sensor networks is still a challenging problem. Note that in this paper, we sometime use the terms of 'mobile user' and 'mobile device' interchangeably.

3 The integration architecture

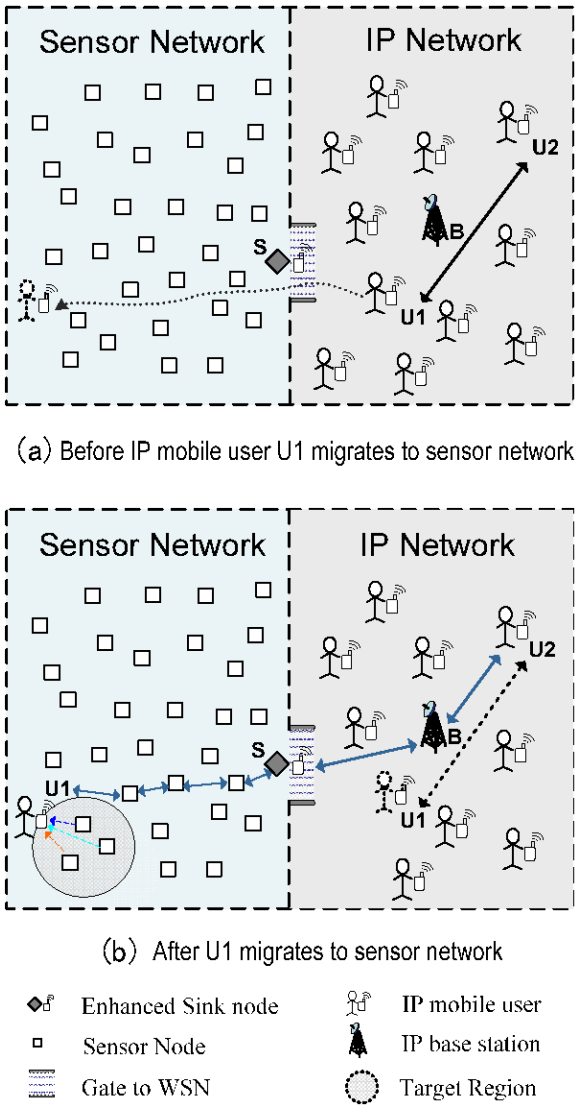
In this section, we present the architecture and design issues of the proposed integration architecture. We first give an overview of the network scenario and basic applications, where new business models and incentives should be sought to allow mobile users to participate in the creation and provision of sensor-based services. Then, we describe our component-based IPSA design and facilitate the creation of a flexible, efficient and highly configurable integration system, which provides mobility support and on-demand service provision while keeping sensor nodes and mobile devices as simple as possible. With a different realization for the combination of the key components, it is expected that flexible trade-offs (e.g. between add-on device complexity

and service quality) can be achieved according to specific application requirements. We present an implementation of the middleware layer and finally examine the problem of supporting multiple WSNs.

3.1 Overview

The basic integrated system comprises one IP network and one sensor network. As shown in Figure 1, the IP network includes a base station (denoted by node *B)* and IP mobile users (i.e. the users carrying IP mobile devices, such as IP mobile phones or portable computers), while a WSN consists of sensor nodes and an enhanced sink node (denoted by *S*) which has an additional interface to connect IP network. We assume the WSN does not overlap with the IP network geographically, due to the vast region to monitor.

Figure 1 An example of an IP mobile user migrating from an IP network to a WSN (see online version for colours)



In Figure l(a), both IP mobile users *U1* and *U2* are within the IP network and can communicate with each other by IP. Figure l(b) shows a mobility case of *U1* moving from the IP network to the WSN for on-demand sensory data querying. As an illustrative example, we can consider *U1* manages an orchard. Periodically, *U1* enters the gate of the orchard to check the status of fruit trees over a wide geographical area, in which there are possibilities that some of fruits grow abnormally. Hence, once observing exceptional phenomena, *U1* collects the environmental conditions such as temperature, humidity and light intension (or any other physical conditions) and deliver the sensed data to a worker (e.g. *U2)* outside the orchard in an attempt to seek timely control and actions. In addition to passively receiving data from *U1*, an IP connection is also likely initiated by *U2* which is not aware of *U1's* moving away. In contrast with the above mobility case, the orchard manager *U1* may also query sensory information without moving to WSN. In the following, we describe how IPSA handle the above three events, respectively.

3.1.1 U1 moves to WSN and sends packet to U2

The sink *S* advertises its existence by periodically sending a beacon signal which can cover the proximity of the gate. As shown in Figure l (a), when *U1* approaches the gate and receives the beacon signal, it sends a registration request to *S.* Then *S* responds to *U1* with a registration reply message carrying the registration result with an assigned sensor address mapped from its IP address, and notifies IP base station *B* the mobility binding for *U1*. By using the mapped sensor address, *U1* can reach the sink or other sensor node(s) based on the gradient (Intanagonwiwat et al., 2000) set-up in WSN. The registration procedure will be described in detail in Section 3.2.

As shown in Figure l(b), when *U1* arrives at the target region, it collects sensed data from the neighbouring sensor nodes, processes the sensor data and then delivers the data to *U2*.[1] From *U1* to *S*, the sensor nodes forward the sensed packets in a hop-by-hop style by the gradient in the WSN, until the data reaches *S.* When *S* receives a sensed packet, the sensory payload is detached from the sensed packet and packetised into an IP packet (or IP packets) by tagging the IP address of *U2.* Then, the IP packet is forwarded to *B* in the IP network and finally is delivered to *U2* by *B.*

During above procedure, although *B* knows the fact that *U1* moves out of the IP network and enters the WSN, it is ignorant of *Ul*'s detailed location in the WSN. Only *S* is aware of the location of *U1*. Thus, both *U2* and *B* are ignorant of the detailed communication mechanism within the WSN. In other words, *U1* is transparent to *U2* and *B.* The realisation of above operations in terms of IP mobility management will be detailed in Section 3.3.

3.1.2 U1 moves to WSN and U2 sends packets to U1

With the mobility binding described as the above, packets sent by *U2* will arrive *S* via *B*. When *S* receives a packet from *B*, *S* will forward it to *U1* by the routing function.

3.1.3 U1 queries sensor network without moving to the WSN

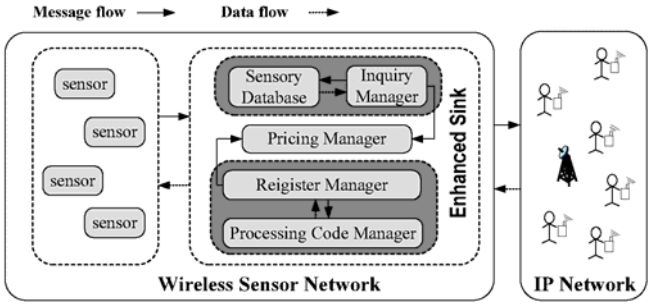
In this case, the whole sensor network can be deemed as a database and the IP user sends an interest packet with specific attributes to *S* which will feedback the required sensed data after the corresponding operation detailed in Section 3.2.

The enhanced ability of *S* incorporated with the processing code of *U1* provides mobility support and on-demand service provision. IPSA provides a service view dependent on the user's current perspective. An IP mobile user can obtain various services as well be connected through a variety of sensor routing protocols by simply changing the processing code imported into middleware of the mobile device.

3.2 The enhanced sink

The enhanced sink node *S* is a key element that embodies IPSA and plays a vital role in the provision of IP mobility support and registration-based services. *S* has both a WSN interface and an IP interface (e.g. IP-based wireless LAN/wired LAN). Figure 2 shows the functional components of the enhanced sink: register manager, processing code manager, inquiry manager and database which are described as follows.

Figure 2 Functional components of the enhanced sink in IPSA



3.2.1 Register manager

The register manager maintains and updates an IP mobile user information table which mainly includes user IDs/profile, log-in/log-off time of the users, IP address entries, mapped sensor address entries, requested application/services and application types. Figure 3 shows an example for the aforementioned 'orchard' application, where the manager *111* and the worker *U2* may require the application of 'orchard managing' which collects the data about environmental conditions. In addition, a tourist U_3 may require the application of 'image retrieving' to obtain pictures of interest, taken by the camera sensors located in inaccessible area for tourists. A fruit buyer U_4 can also request the application of 'price querying' since he/she is more careful about fruit prices. The information about prices and/or the description of each kind of fruits can be stored in some sensor nodes in advance.

Figure 3 An example of register entry in sink node while IP mobile user crosses network border (see online version for colours)

User ID	User Profile	Login Time	IP Address	Application Type	Logoff Time	Expiration Time
U1	Manager	07:54:23/2	192.168.1.100	Orchard Managing	08:15:17/2	17:30:17/2
U2	Worker	08:04:55/2	192.168.1.200	Orchard Managing	N/A	17:30:17/2
U3	Tourist	09:00:17/2	192.168.1.111	Image Retrieving	10:30:17/2	11:00:17/2
U4	Fruit Buyer	10:28:17/2	192.168.1.222	Price Querying	11:36:17/2	12:00:17/2
U5						
U6						
. . .	. . .	. . .	. . .	. . .	. . .	. . .

An example of information entries in the register manager

The information entries are soft state, in other words, after some certain expiration time, shown in Figure 3, they are not valid any more. This mechanism is necessary because if an IP mobile user loses the connection with *S* due to failure of notifying the leaving (e.g. the user's IP mobile device is powered off, etc.). Then, the expiration mechanism can avoid garbage information entry still being kept in the register manager.

3.2.2 Processing code manager

When an IP mobile user enters the sensor network and its information is registered to the enhanced sink, the processing code manager takes charge of processing code assignment for an IP mobile user. When a control message is issued by the register manager, the processing code manager generates on-demand code, considering the application type and user's requirements.

Given Figure 3 as an example, the processing codes assigned to *111* and/or *U2* might be used to aggregate/analyse the environment-sensitive readings from several sensor nodes. The processing code assigned to U_3 is more complicated. Since the amount of data generated by an image sensor is generally very large, transmitting complete pictures not only consumes much bandwidth and energy but also is

unnecessary, if U_3 and/or a remote IP user needs to evaluate only a certain region of interest in the picture. Here, a specific image segmentation code can be used as the processing code for *113*. Even for the same application (e.g. image retrieving service for the tourists), when the circumstances surrounding the environment being sensed have changed substantially, the processing code should be adjusted accordingly. For example, different image segmentation algorithm(s) can be assigned to different tourists to keep the image processing code working efficiently for different image sensors.

In addition to adapt sensor service to various application-specific requirements, the benefits of applying processing code can potentially reduce bandwidth consumptions by moving the data processing elements to the locations of the sensed data, whose transmissions in the raw data would otherwise incur most of the nodes' energy expenditures. The idea of shifting the application-specific processing intensive functions from sensor nodes to the IP mobile device is feasible since it has rich capability and available resources compared to the sensor nodes. This is highly appealing when large amounts of data have been collected and need to be disseminated to the sink.

Furthermore, the functionality of processing code also includes (1) encapsulating IP packet into sensor routing packet and (2) explaining the IP address to the unique sensor address which can be identified in WSNs. The illustration of basic functions included in the processing code will be given in Section 3.3.

3.2.3 *Pricing manager*

In the future, WSNs will be open to civilians to facilitate/enrich their lives. On the other hand, the agency which invests the WSNs will get profit for managing/ maintaining the WSNs. For each instant sensor service requested by an IP user, if the service is approved, the pricing manager will charge the price according to the application type, service time, the size of sensory data the user retrieved, etc.

3.2.4 *Sensory database*

A sensory database is used to store up-to-date sensory information. The sensory database is especially appealing for some applications, such as temperature and/or humidity surveillance. For such applications, the sensor nodes periodically wake up from sleeping-mode to collect environmental sensed data, which are delivered to the sink and stored in the sensory database.

3.2.5 *Inquiry manager*

The introduction of inquiry manager is used to provide a timely service for the users within the IP network. When receiving a request from an IP user, the inquiry manager accesses the sensory database to retrieve the information needed by the user or returns the result of the queries. If the result shows that no available information is stored in sensory database at this moment and/or the information is out of date, the IP user may further decide whether to request remote sensory data collection service to the sink by charging expensive fee of tasking the WSN.

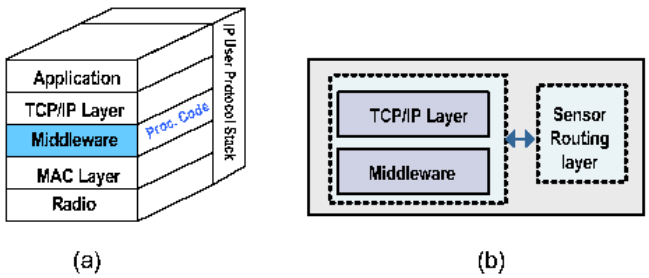
3.3 *Processing code design in middleware layer*

Recent advances in hardware have allowed the implementation of IP mobile device that supports the use of specialised add-on modules for IPSA support (e.g. through USB port). Generally, an IP mobile device with WSNs accessing extension (e.g. the device used by *111* in Figure 1) has no big difference with a normal IP mobile device. The only difference is that *111* plugs a small add-on hardware (e.g. USB stick) in his/her IP mobile device and the add-on hardware can be put off when the service from WSNs is not needed. In the IPSA, such add-on hardware works as a middleware layer, as shown in Figure 4(a). Middlewares are widely used in computing systems to bridge the gap between the operating system and high-level components and to facilitate the development and deployment of applications (Boulis et al., 2003; Fok, 2005; Gonzalez, 2006).

When a normal IP network user *111* enters the WSN where IP does not run, the IP network layer plus the middleware layer becomes equivalent to sensor routing layer, as shown in Figure 4(b). That is, one could view IP network layer and middleware as a single layer or a joint protocol for this 'super layer of sensor routing'.

This design ensures *111* always use IP address to communicate with other nodes in sensor and/or IP networks. Also, the middleware layer provides a universal interface for IP mobile users to obtain various application-specific services, which are realised through processing code. A processing code is a special kind of software or computer program that can be run in the corresponding middleware environment. Though additional resource is needed to run processing code, the IP mobile device is reasonable to be equipped with enough processing ability which is higher than a sensor node. The sink will change the processing code intelligently in response to the user-specific requirements and/or the changing conditions in the network environment.

Figure 4 The introduction of middleware layer at IP mobile nodes: (a) protocol stack; (b) merging of adjacent IP and middleware layers (see online version for colours)



With the support of the middleware layer, the processing code can work efficiently. Sensor applications are highly user specific and the information needed to be collected is also changed from time to time. However, there are some basic functions which should be considered in the design of the processing code.

- *Address translation between IP and sensor address*: The translation function to convert IP address into integer identification as the sensor network address.

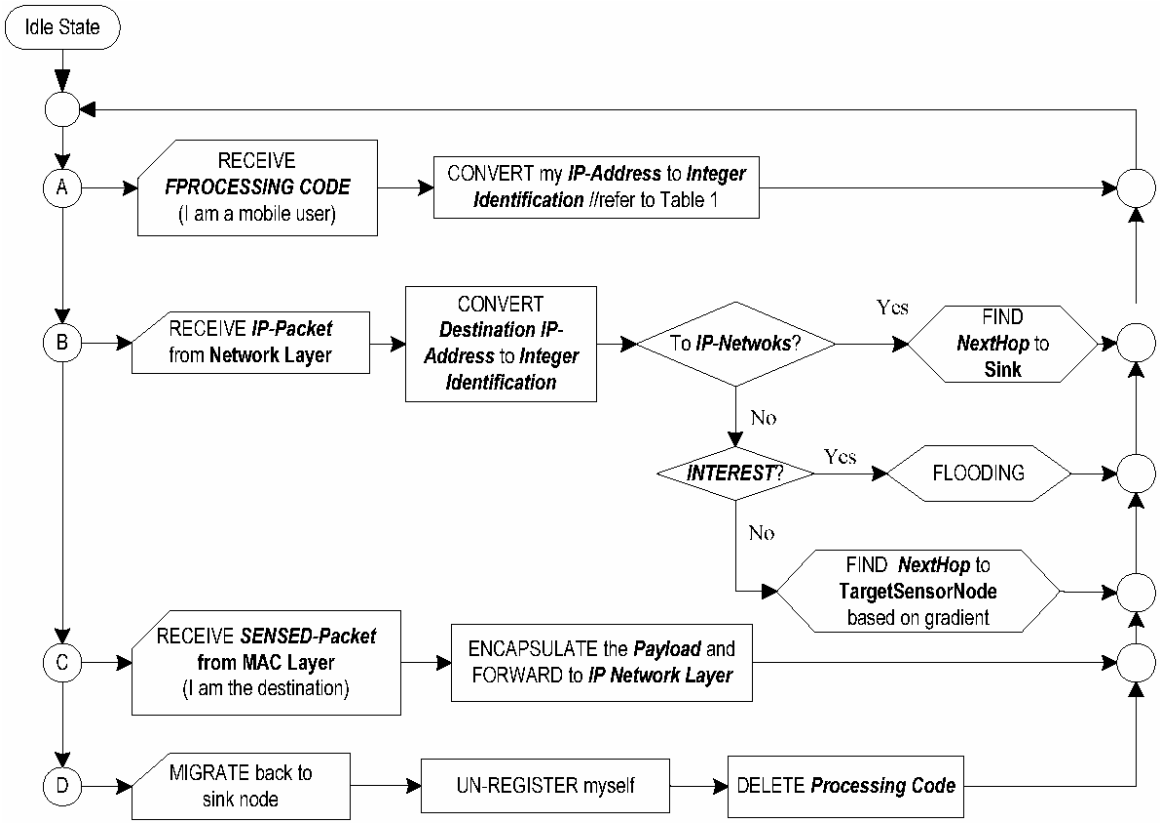
- *Code-assisted local processing:* The raw data generated by individual source nodes are reduced by IP mobile devices. Then, only relevant information is stored and forwarded to other IP user(s).
- *Code-assisted data aggregation:* It has been shown that when nodes are in close geographical proximity, their measurements display a high degree of correlation. Thus, data aggregation is performed to reduce the redundancy of the sensory data from a single event, when the IP mobile user visits source nodes in the vicinity of the event one by one.
- *Adapting to network dynamics:* the network status is dynamic so as to design different algorithms to achieve load-balance and energy-efficiency; also whether geo graphical routing can be supported should depend on the GPS hardware of sensor node. Being adaptive to such dynamics, the processing code could be very different from one WSN to another WSN.

Figure 5 shows the flowchart of the basic functions included in the universal processing code. When an IP mobile user arrives the WSNs border, it will register the user information at the sink and informs the sink which kind of sensory data it needs. Then, the sink will send corresponding processing code to the mobile user. On receiving the processing code, the IP address of the mobile user will be converted to an integer ID which can be recognised by other sensor nodes as sensor address. In our experiment, the IP to sensor address mapping code is shown in Table 1, while sensor to IP address mapping is the inverse procedure. When the middleware layer receives an IP packet from the IP layer, it explains the destination IP address. If the IP packet is targeted to IP networks, the packet will be delivered to the sink. Otherwise, if the packet is an interest packet which includes the requirement of sensory information, the packet will be controlled flooding to locate a target region. If the packet needs to be routed to a sensor node, it will be unicast to the corresponding next hop node. When the mobile user finishes the task and migrates back to the sink, it will un-register itself and delete the processing code from its memory. The sink also deletes the register entry of the IP mobile user.

Table 1 Pseudo-code for address converting

```
procedure address_converting(ip_string)
begin
01 ip_addr = ip_address_string_to_value(ip_string);
02 ip_addr_tmp = ip_addr >> 8;
03 ip_addr_tmp1 = ip_addr_tmp;
04 ip_addr_tmp = ip_addr << 8;
05 ip_fourth_suffix = ip_addr - ip_addr_tmp;
06 ip_addr_tmp = ip_addr_tmp1 >> 8;
07 ip_addr_tmp2 = ip_addr_tmp;
08 ip_addr_tmp = ip_addr_tmp1 << 8;
09 ip_third_suffix = ip_addr_tmp1 - ip_addr_tmp;
10 ip_addr_tmp = ip_addr_tmp2 >> 8;
11 ip_addr_tmp = ip_addr_tmp2 << 8;
12 ip_second_suffix = ip_addr_tmp2 - ip_addr_tmp;
13 ip_prefix = ip_addr_tmp2 >> 8;
14 sensor_addr = ip_prefix*1000000000
          + ip_second_suffix*1000000
          + ip_third_suffix*1000 + ip_fourth_suffix;
end
  ">> and << are bitwise operators."
```

Figure 5 Basic functions included in the universal processing code for IP mobility management and sensory inquiry



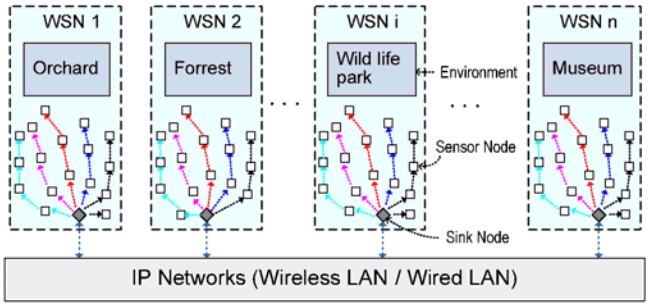
3.4 Supporting multiple WSNs

In future, with sensor technologies highly developed, various WSNs will become easily accessible to human society which is covered by IP wireless/wired networks. The WSNs will become closer to their resident area. Furthermore, there is not only one kind of WSNs, many kinds of WSNs are possibly located around civilian resident area which is overlapped with wireless IP networks. Thus, it is possible for IP mobile user to obtain the various information from different WSNs.

Given Figure 6 as an example, an IP mobile user can go to an orchard to check the status of fruit, as illustrated in Section 3.1; he can go to a forest for camping; he can go to a wild life park to watch or study animal; he can also go to a museum to collect information for his research, etc.

Different WSNs will use different protocols and target different applications. The behaviour of sensed data collection cannot be predetermined in advance, since the fixed protocol might only suit for one or few WSNs. If an IP mobile user needs to collect information from a new deployed WSN, the fixed protocol does not work. Thus, again, it is important to introduce a middleware layer for providing an interface to import flexible and intelligent processing code, which can achieve the goal of 'one universal add-on hardware, multiple services supporting'.

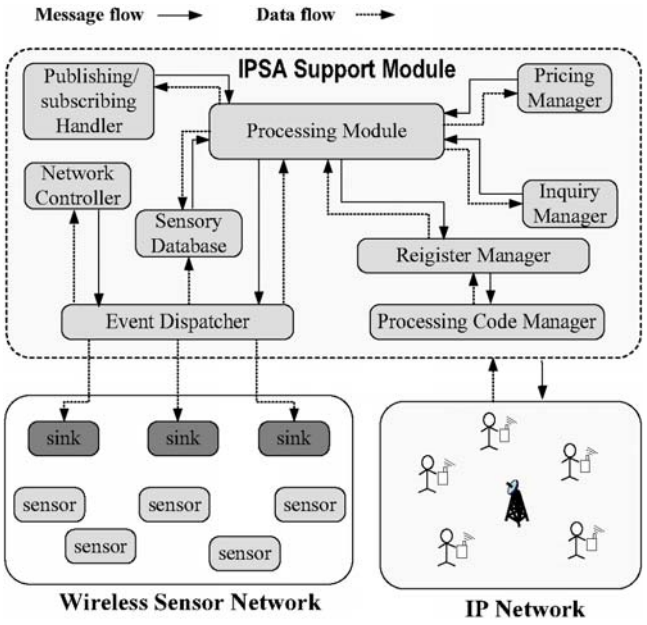
Figure 6 The architecture for integrating IP networks with multiple sensor networks (see online version for colours)



4 IPSA extensions

IPSA support module is used to shift the responsibility from the sink to a special add-on hardware module. If all the functionalities are realised in the sink, it is required that enough capacity and processing ability be equipped, which is reasonable in the case that the sink is realised as an infrastructure. However, with the increasing number of service requests from IP users, the sink becomes a bottleneck and may not be able to provide timely services. To address this acalability problem, we propose to integrate all of the functional entities in a special hardware called IPSA support module, which also includes several other add-on components, such as event dispatcher, network controller, processing module and publishing/subscribing handler, as shown in Figure 7.

Figure 7 Functional components of the enhanced sink in IPSA



When multiple sinks are working simultaneously, the *event dispatcher* plays a role of coordinating among them. A suitable scheduling algorithm can achieve load-balancing among multiple sinks and provide differentiated services for IP users with various priorities. The primary responsibility of the *network controller* is to provide the means for organising the WSNs and to keep track of the states of the sensors. The *processing module* deals with incoming sensory data by accessing the sensory database. The publishing/subscribing handler advertises the available sensed data information to IP network actively, in order to provide faster data delivery services for IP users.

4.1 Event dispatcher

The event dispatcher takes charge of task (re)assignment to manage sensors' operations. When a task is issued by the information processing manager or the network manager, the event dispatcher processes the event task and adapts it to the WSN environments considering sensor types and locations. If the task is complicated, it can be divided into two or more subtasks that will be assigned to a specific sensor(s) or sensor group(s) (Chen, 2005). The subtasks are transferred to the WSN through multiple sinks, which relay (or broadcast) each subtask to specific sensors (or sensor groups) using a task assignment protocol. When sensory data are reported by sensors, the event dispatcher stores them to the sensory database to enable other components to access them. If the energy level of network is reported, it is transferred to the network controller for network load balance or maintenance.

If WSN does not have the ability to provide corresponding services required by the users, the event dispatcher will negotiate with the users. For example, when the network is working for a long time, some of the sensor

nodes are depleted, the path becomes longer, which indicates the lack of ability to delivering the packet in timely fashion. Then, the event dispatcher may inform processing code manager to change code to enhance the ability to satisfy quality of service requirement by the IP user (e.g. embed more powerful code by compromising energy efficiency, such as exploiting concurrent multipath transfer (Chen, 2007) and cooperative routing (Chen et al., 2008), etc.)

4.2 Network controller

The network controller, depending on the energy level of sensors, requests the event dispatcher to adjust sensors' operations (e.g. such as the data reporting rate and the sensor nodes' duty cycle). By periodically flooding a network-wide 'heartbeat' message, network controller can realise residual energy monitoring (Bhattacharyya et al., 2007) and failure detection. When any region cannot be sensed, a notification message should be forwarded to users.

Network controller also provides enhanced service for a special group of IP users. For example, if user needs to enter a region of interest to check the environmental conditions, network manager will send a notification to the sensor nodes in that region to sense the environment in advance before the user arrives the region, so as to provide faster service.

4.3 Processing module

For each query request from users, the processing module provides advanced processing techniques such as searching, filtering and pricing, etc.

When it receives a query from the IP network and finds that the query was requested before, the processing module will access the sensory database and return results directly. When the query requires the present or future data, it will dispatch tasks to the sink(s) through the event dispatcher.

4.4 Publishing/subscribing handler

For the IP users who stay in the IP network, the whole WSN can be virtually deemed as a big 'source node'. Similar to directed diffusion (Intanagonwiwat et al., 2000), publish/subscribe provides an application's view to the IP network and attribute-based naming specifies which sensory information is needed, by specifying service types, desired data rate and possibly some geographical region, etc.

Exploiting IP technology makes publishing sensory information more convenient, as the publishing/subscribing handler can publish up-to-date sensory information by providing entry points for external applications (or users) in various ways, e.g. HyperTtext Transfer Protocol (http) and Short Message Service (SMS).

5 Conclusion

In this paper, we propose a novel architecture, termed IPSA (IP and Sensor Networks Integration Architecture) that allows an application to be designed flexibly while hiding the details of the WSNs and IP networks. In IPSA, an IP mobile user only needs to carry a plug-in middleware in an add-on fashion if it needs to collect sensed data in WSNs. The sink will assign the corresponding processing code to the IP mobile user according to the application requirements when the user enters the sensor network, where IPSA will manage IP mobility and data collection transparently. IPSA provides an efficient way for IP users to access and deliver sensory data based on their own application-specific requirements, either staying in the IP network or roaming into the WSN. In Figure 8, we categorise the existing work in terms of core design components.

Figure 8 Classification and comparison of existing integration architectures (see online version for colours)

Integration architecture	Requirement on sensor nodes	Requirement on IP nodes	Application diversity	Flexibilty and Extensibility	IP Mobility
Proposed in Marco and Bhaskar (2003)	Application-level interface	Application-level interface	N/A	Low	N/A
Proposed in Fall (2003)	Bundle layer	Bundle layer	N/A	Low	N/A
Proposed in Dunkels et al. (2004)	TCP/IP protocol stack	No requirement	N/A	Low	N/A
Proposed in Dai and Han (2004)	No requirement	WSN protocol stack	N/A	Low	N/A
IPSA	No requirement	Plug-in middleware	Multiple	High	Support

Currently, our framework does not support IP mobile user to migrate from one WSN to another WSN without unregistering in original WSN. Since different WSNs have different processing code and routing methods, the processing code designed in one WSN may not work in another. Additionally, if the IP mobile user leaves the original WSN without unregistering, the pricing problem should also be carefully addressed. We will study these interesting problems in our future work.

Acknowledgements

This work was supported in part by the Canadian Natural Sciences and Engineering Research Council under grant STPGP 322208-05. Professor Yang Xiao's work was partially supported by the US National Science Foundation (NSF) under the grant numbers CCF-0829827, CNS-0716211 and CNS-0737325. Professor Shiwen Mao's research is supported in part by the National Science Foundation under Grant ECCS-0802113 and through the Wireless Internet Center for Advanced Technology at Auburn University.

References

Al-Karaki, J.N. and Kamal, A.E. (2004, December) 'Routing techniques in wireless sensor networks: a survey', *IEEE Personal Communications*, Vol. 11, No. 6, pp.6–28.

Bhattacharyya, M., Kumar, A. and Bayoumi, M. (2007) 'A framework for assessing residual energy in wireless sensor network', *International Journal on Sensor Networks,* Vol. 2, No. 3/4, pp.256–272.

Boulis, A., Han, C. and Srivastava, M. (2003, May) 'Design and implementation of a framework for efficient and programmable sensor networks', *ACM Mobi-Sys*, pp.187–200.

Chen, M., Kwon and Choi, Y. (2005, November) 'Data dissemination based on mobile agent in wireless sensor networks', *IEEE LCN,* pp.527–529.

Chen, M., Gonzalez, S. and Leung, V. (2007, December) 'Applications and design issues of mobile agents in wireless sensor networks', *IEEE Wireless Communications Magazine*, Vol. 14, No. 6, pp.20–26.

Chen, M., Kwon, T., Yuan, Y., Choi, Y. and Leung, V. (2007) 'Mobile agent-based directed diffusion in wireless sensor networks', *EURASIP Journal on Advances in Signal Processing: Special Issue on Visual Sensor Network,* Vol. 2007, Article ID 36871, p.13, doi:10.1155/2007/36871.

Chen, M., Kwon, T., Mao, S., Yuan, Y. and Leung, V. (2008) 'REER: reliable and energy-efficient routing protocol in dense wireless sensor networks', *International Journal on Sensor Networks,* Special Issue On Energy-Efficient Algorithm and Protocol Design in Sensor Networks*,* Vol. 3, No. 5/6.

Chen, M., Leung, V., Mao, S. and Yuan, Y. (2007, November) 'Directional geographical routing for real-time video communications in wireless sensor networks', *Elsevier Computer Communications,* Vol. 30, No. 17, pp.3368–3383.

Cho, J., Shim, Y., Kwon, T., Choi, Y., Pack, S. and Kim, S. (2007, December) 'SARIF: a novel framework for integrating wireless sensor and RFID networks', *IEEE Wireless Communications,* Vol. 14, No. 6, pp.50–56.

Dai, H. and Han, R. (2004, November) 'Unifying micro sensor networks with the internet via overlay networking', *IEEE LCN*, pp.571–572.

Dunkels, A., Alonso, J., Voigt, T., Ritter, H. and Schiller, J. (2004, February) 'Connecting wireless sensornets with TCP/IP networks', *Proceedings of the Second International Conference on Wired/Wireless Internet Communications,* Frankfurt, Germany.

Fall, K. (2003, August) 'A delay-tolerant network architecture for challenged internets', *ACM SIG-COMM*, pp.27–34.

Fok, C, Roman, G. and Lu, C. (2005, April) 'Mobile agent middleware for sensor networks: an application case study', *Proceedings of IEEE IPSN*, pp.382–387.

Gonzalez, S., Vuong, S. and Leung, V. (2006, June) 'A mobile code platform for distributed task control in wireless sensor networks', *Proceedings of ACM Mo-biDE*, pp.83–86.

Intanagonwiwat, C., Govindan, R. and Estrin, D. (2000, August) 'Directed diffusion: a scalable and robust communication paradigm for sensor networks', *ACM MobiCom,* Boston, MA, pp.56–67.

Mao, S. and Hou, Y. (2007, November) 'BeamStar: an edge-based approach to routing in wireless sensor networks', *IEEE Transactions on Mobile Computing*, Vol. 6, No. 11, pp.1284–1296.

Marco, Z. and Bhaskar, K. (2003) 'Integrating future large-scale wireless sensor networks with the internet', USC Computer Science Technical Report CS 03-792.

Perkins, C. (2002, August) 'IP Mobility Support for IPv4', *IETF RFC 33U.*

Note

1 We assume that *U1* is equipped with an interface to be able to communicate with the sensor nodes. See Section 3.3.