A Ubiquitous System for Secure Management of Critical Rescue Operations

Suleyman Kondakci
Faculty of Engineering & Computer Sciences
Izmir University of Economics,
Izmir, Turkey
Email: suleyman.kondakci@ieu.edu.tr

***Abstract*—Application of real world sensor networks to critical rescue and aid operations is still a challenging research and development area for scientists and engineers. A ubiquitous monitoring, aid, and rescue management system, its requirements, architecture, communication protocols, algorithms, and design details are presented here. The system is designed for use by various operational purposes, organizations, individuals, and operation teams, especially by critical rescue and aid forces. With its light weight and secure communication abilities, it can be used in a variety of critical operations, ranging from disaster management to anti–terror operations. The proposed system consists of three interoperating subsystems; ubiquitous agents, intermediate message passing agents, and command control centers. The paper considers first the basic requirements and design of the overall system, and then presents a prototype implementation consisting of the three subsystems.**

***Keywords*-WSN application; cryptography; security protocol;ubiquitous system.**

I. INTRODUCTION

Historically, it has been witnessed that controlling natural disasters threatening human life has not been considered as series as it should deserve. Especially, populations do steadily proliferate in underdeveloped countries, which makes it even harder to overcome critical situations. This is mainly due to the limitation in economical resources and lack of technologies needed to safely manage critical operations. Especially in such countries, developing inexpensive systems needed to ubiquitously monitor and manage subjects is a challenge area for many researchers. It is still an open problem to design adequate mobile systems needed to rapidly identify critical situations and communicate related data with management points in order to efficiently manage dynamically growing emergency situations.

By going through the history of disaster outbreaks throughout the world, numerous reasons can be listed for employing science and technology as one of the primary aids in the management of disasters and rescuing of human lives. Thus, there is a growing demand for developing effective measures to manage critical public safety and disaster management operations, environmental monitoring, and various object tracking operations. In order to elaborate the management of such critical events, we have designed a ubiquitous monitoring system (UBIMOS), [1], which monitors real-time environmental conditions (gas, smoke, humidity, temperature), locations of subjects (e.g., humans, animals, and system) as well as their vital conditions known as heart rate, oxyhemoglobin saturation [2], thermal stress factors such as the body temperature, [3]. Vital human conditions are noninvasively measured by wearable sensors. As the main unit, a wearable micro-computer collects and processes sensor data, and transmits the sensor data to a remote center for the coordination of necessary emergency operations. An important contribution of UBIMOS is such that the results gained by its prototype implementation will allow us to refine assumptions and requirements made when designing hardware, software, protocols and mechanisms for critical ubiquitous operations.

As will be detailed later, one of the major application areas of UBIMOS is the secure wireless sensor network (WSN) operation, e.g., management of anti-terror operations. Related to this, we have incorporated a secure communication protocol for secure authentication of the nodes and for cryptographic data exchange between the nodes. There is an increase in terrorist actions undertaken in several countries. UBIMOS also aims at mitigating the potential damage from such terrorist attacks by two ways: (i) by equipping public transportation vehicles and other terrorist targets with the early detection and warning units of UBIMOS, and (ii) by equipping anti–terror team members with wearable UBIMOS units that monitor and exchange data about locations, environmental conditions, and vital body conditions of the team members located at the field of operation. Terrorism, policies, and anti–terrorism legislation issues have been seriously considered throughout th world, and a vast amount of organizations, e.g., Council of Europe Convention on the Prevention of Terrorism [4].

A. Outline of the Paper

In the following, Section II gives a brief overview of related work, Section III outlines the major requirements of UBIMOS, Section IV presents the overall architecture and subsystems of the presented system, Section V details the structure and operation modes of the UBIMOS subsystems. Section VI deals with the construction of UBIMOS domains, their configurations, routing algorithm, Section VII considers

secure communication and structure of the transport protocol. Section VIII reviews the prototype implementation of UBIMOS, Section IX concludes the paper.

II. Related Work

The work presented here contains the application of several new topic areas such as WSNs, topology control and routing in WSNs, security, interconnection between IP and sensor networks, programming, and embedded sensor hardware design and integration. WSNs [5], as the fundamental communication technology of the UBIMOS project, have a broad application spectrum with an enormous variety of designs [6]. A location aware WSN architecture called Disaster Aid Network dealing with real-time patient localization is presented in [7]. Another rescue related work considering the application of WSNs for fire rescue operations together with its requirements and challenges is given in [8].

In recent years, ubiquitous computing has become very popular as it delivers truly useful solutions in pervasive application domains. Especially, WSNs, mobile wireless technologies based on General Packet Radio Service (GPRS), Digital RF, WLAN 802.15.4, and Global Positioning System (GPS) are becoming today's must-have technologies for a variety of ubiquitous applications.

As already known, security threats [9], [10] to WSNs and countermeasures against those threats are of many faceted. Implementing security in small hand-held devices is a challenging task, because of the strengthened requirements such as the requirement for extreme low power consumption, high–mobility, and real–time operations. There exist a number of hardware and software design proposals for providing effective communication approaches. For security critical operations add on security functionality is a must. Adding security functionality to existing communication protocols (e.g., the IEEE 802.15.4 and TCP/IP protocol suits) is a demanding issue, and can even become a speed burden on real-time operations. In conjunction with this we have designed a secure transport protocol, based on the RSEP protocol [11], for use by UBIMOS's secure communications. An architecture for secure communication in mobile wireless networks is presented in [12]. An implementation and analysis of a lightweight cryptographic algorithm suitable for WSNs is presented in [13], and a reliable synchronous transport protocol for wireless image sensor networks is considered in [14].

A typical wireless sensor node has limited protection against radio jamming. The situation becomes worse if energy-efficient jamming can be achieved by exploiting knowledge of the data link layer. Encrypting the packets may help to prevent the jammer from taking actions based on the content of the packets, but the temporal arrangement of the packets induced by the nature of the protocol might unravel patterns that the jammer can take advantage of, even when the packets are encrypted. Several jamming attacks that allow the jammer to jam S-MAC, LMAC, and B-MAC are discussed in [15], where the algorithms are described in detail and simulated for the analysis of energy efficiency. Another work dealing with the denial of sleep attacks is presented in [16].

Another important aspect of WSNs is the power consumption [17]. In order to reduce the power consumption in sensor nodes, the nodes must go into a sleep mode during idle periods. Adaptive approaches to the solution of power consumption problems are important, which provide dynamic relocations of sensors and resizing of sensor networks. To achieve higher scalability and adaptability, a network architecture composed of self-organizing entities is presented in [18].

Design of energy-efficient wireless sensor networks with censoring and on-off sensors is considered in [19]. Recall that censoring is a statistical analysis method used for reliability testing of systems. A framework for the study of power consumption and bit error rate performance of non-coherent impulse radio ultra wideband correlation receivers conforming to the IEEE 802.15.3a is considered in [20]. The work from [21] examines the performance of differential positioning using both the GPS and GLONASS satellite systems for vehicle positioning. Another work, [22], addresses the problem of GPS signal tracking processes in low signal-to-noise ratio (SNR) and multi–path interference environments.

Routing [23] is a relatively more energy consuming process in WSNs, which is also a challenging research area growing with a great interest. Two different energy– and security–aware routing approaches for the real-time communication in WSNs are presented in [24] and [25]. Finally, an extensive work dealing with the estimation of mobile user's trajectory in mobile wireless networks is presented in [26].

III. Requirements

UBIMOS is mainly a domain-specific communication infrastructure designed to exchange critical event information for the management of critical operations. Exchange of the critical event data is performed via ubiquitous agents (UAs) and other agents having higher communication capabilities, which are called intermediate message passing nodes (IMPs). The management of the critical operations is coordinated by command control centers (3Cs). These centers also provide functionalities for long-range (RF and satellite) communications among different UBIMOS domains. Obviously, the UBIMOS infrastructure contains three different types of subsystems each with different technical and procedural capabilities, which require consistent policies for building mechanisms that make the nodes interoperate efficiently and securely. Thus, the policy of the UBIMOS infrastructure consists of the following major aspects:

Power consumption. Configurable modes of operations are necessary both for different operational circumstances

and for different modes of operations. For example, security operations require more energy sustainability. Thus, if a node operates in secure mode then precautions are needed for the provision of redundancy in the power source and reduced coverage both in the *distance* among the nodes having sensors (UAs and IMPs) and in the *size* of the operational domains. The size depicts the number of non–idle nodes in a given domain. The distance between two UA-nodes is depicted by the number of active hops between them, while the distance between a UA and an IMP, a UA and a 3C, and an IMP and a 3C is given by the time delay a packet takes to travel to its destination. As a result of the reduction in the domain size and shortened node distances, the number of instructions required by security and routing algorithms will decrease, which will substantially reduce the energy consumption. Secure exchange of data always requires larger packet sizes compared to insecure data exchanges. This is due to the secure node authentication and cryptographic data contents. Therefore, improvising on the energy constraints of wireless sensor networks is crucial. Considering the aspect of system availability, WSNs differ considerably from other existing networked systems. Due to extreme system availability and energy constraints, the design of WSNs requires a proper understanding of the interplay between network protocols, energy-aware design, signal-processing algorithms, embedded and distributed programming techniques, [27]. By dynamically configuring coverage of a domain, we can maximize the domain's lifetime instead of minimizing the energy consumption or maximizing the residual energy. There have been proposed similar models to minimize energy consumption of sinks in a WSN, [28] and [7]. With the UBIMOS policy, relocating of the nearest multi-hop nodes is periodically done so that the effective route path of messages diminish to a minimum level. Besides, in the secure mode, security functionality of the IEEE 802.15.4 will be bypassed by the secure transport protocol of UBIMOS. This will substantially eliminate the required power consumption in the physical layer.

Security. Achieving secure data exchange between small wireless systems is a challenging issue, which requires the design of interoperable, robust, time–efficient, easily applicable, scalable, and configurable systems. Based on the operation type, each node may change its operation mode to be secure. Especially, operations dealing with confidential operations must perform encrypted data exchange and secure authentications. There also exist situations where sensor data are transmitted securely to command control centers over insecure channels (e.g. Internet) or via dedicated RF channels. For example, one may continuously monitor, oil pipes, bridges, and critical passage points used for smuggling of drugs or immigrants. There are several security threats if the data from such nodes are transmitted in plain form. For example, critical data may be intercepted, hijacked, changed, and transmitted to unauthorized principals during a confidential operation or during the observation and exchange of real-time data. Launching the so called "man in the middle attack" can help adversary to intercept data transmitted in plain form, and the intercepted data can be disclosed, misused, and redirected to unauthorized principals. Security related problems and requirements for adequate solutions are considered separately in the Secure Communication section.

Scalability. A WSN node, regardless of the spread of geographical positions among its nodes, should be easily relocatable and scalable for configuring different coverage spaces by applying techniques for optimum relocation, shrinking, and augmenting. To achieve higher scalability and adaptability, design of a ubiquitous network should contain the composition of self-organizing elements. Adaptive approaches to the solution of power consumption problems in sensor networks are important means for effective scalability. Adaptive solutions may provide dynamic relocations in the organization of sensors while they are mobile and hence can drop out of the coverage or they may run out of power. These problems are considered under routing (VI-A) and self-optimization (VI-B).

Self-optimization. Nodes are arranged to apply dedicated routing algorithms, intelligent clustering approaches, and algorithms for sleep/idle mode operations. These algorithms vary from node to node depending on the node type. For examples, UAs are intensively active and require frequent use of intelligent clustering and modified multi-path routing algorithm. These are effective approaches needed for reducing the power consumption and diminished WSN traffic. A self-optimization algorithm comprised of the intelligent clustering, a modified multi-path routing algorithm, and idle period management algorithm is used to minimize the average consumed energy for active sensors during the data transmission and sensor data processing. Details are given in Section VI-B: Optimization Strategy.

Self-organization. During the initial setup and also periodically under operations, nodes should learn their relative distances to available sinks, mainly the nearest IMP domains. Here, the nearest neighbor algorithm applying the Minkowski metric is used to cluster UAs into routing groups in order to build a table on each node that contains shortest paths within its domain. Further, for locating and effectively communicating with the nearest IMP, each UA will be dynamically clustered around its nearest IMP using the hierarchical clustering algorithm. Since the requirements for scalability, self-optimization, and self-organization are closely related to the dynamics of the communication of the nodes, they are commonly

considered in Section VI, Setting Up a UBIMOS Domain.

IV. Overall Architecture

UBIMOS is a mobile operation management system with many actors composed of a secure communication infrastructure, mobile human operators carrying wearable computers and sensors, and stationary computer nodes with server capabilities that can securely communicate over long range RF and satellite communication channels. These elements can be configured to manage various types of emergency operations, ranging from natural disaster managements to anti-terror operations. Therefore, the entire system is organized around a secure communication infrastructure, which can serve many types of operational architectures each designed for a specific type of operation. For example, an architecture for a medical care center can be designed to aid ambulatory operations within a geographical region. In this domain patients are equipped with UA's sensors for monitoring and exchanging vital body parameters with the medical center. Another architecture can be setup to manage critical border security of a part of a country. Tracking of individual sports activities exercised under harsh conditions can also be managed by UBIMOS. For example, an operation for rescuing buried mountaineers by a snow avalanche or an accident occurred during a rafting sports activity.

The design of an operational architecture consists mainly of a secure communication protocol, back-end services, and a variety of wireless agents used by operation teams on the field. The secure communication protocol enables users to set up domains (operational architectures), create communication channels among the UBIMOS nodes and make them securely communicate with each other. The back–end services, comprised of computer servers and human operators, perform tasks related to the emergency response management, logging of sensor and communication data, and management of remote operations carried out by rescue/operation teams. Agent nodes, implemented as wearable computers with sensors, are responsible for collecting and transmitting sensor data both from team members and environments to emergency response locations (3Cs) within a given WSN segment (UBIMOS domain). The communication between an agent node and a 3C point is carried out in a full duplex form so that instantaneous conditions of the team members and interventions from the 3C point are accomplished safely.

As already mentioned, UBIMOS is designed to conduct operations in a number of domains varying from individual rescue operations to anti–terror missions. Members of the operations are equipped with UBIMOS agents, while the communication of the systems are organized as wireless sensor networks each with a specific operational domain. Some of the operational domains can be defined as civil defense, public transportations, maritime security, railroad security, traffic control, rescuing and anti–terror operations, and management of disasters (e.g., flooding, earthquake, fire). Accordingly, the domains are organized in WSNs, which can be geographically dispersed depending on the operation type.

Figure 1 shows the general architecture of the operational domains, where each domain consists of three major elements, command control centers, first level agents (IMPs), and second level agents (UAs). Both the IMPs and UAs are

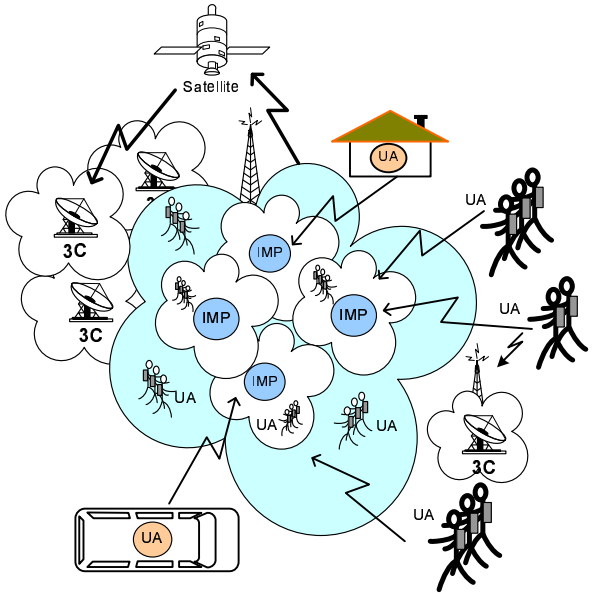

Figure 1. General architecture and components of the UA and IMP nodes

mobile embedded nodes each equipped with sensors and a wearable computer. The main difference between an IMP and UA is such that the IMP has much higher functionality for communication and routing. In addition to all functions a UA has, the IMP acts also as a gateway between its domain and the dedicated command control center. An IMP can also function as a bridging unit (and a sink also) between the UAs of its own domain and external UBIMOS domains. For this reason, depending on the size of a domain, fewer IMPs are harnessed in a given domain. Because, the UA nodes operate in much shorter communication ranges performing most of the intensive operations on the field. On the other hand, the IMP nodes can be configured to operate with reduced mobility taking care of mostly gateway operations among the UAs and the related command control center. At least one IMP must be assigned to a domain, otherwise long range communications between UAs and their command control center can be inefficient, or completely interrupted depending on the geographical distance. As shown in Figure 1, each cloud denotes an operation domain (actually a WSN), where each domain can work independently of others as well as cooperating with some other WSNs as required. As will be detailed later, each WSN is organized as a hierarchical communication system, in which the UA nodes

gather data and route the data to IMPs, which further forward the data to a command control center, and vice versa.

A. Organizing a UBIMOS Domain

UBIMOS domains are constructed according to the type of operations required, each varying from the single disaster management (e.g., mountaineer) to squadron based anti-terror operation. Even with fewer rescue members UBIMOS can be used to aid rescue squadrons if a domain contains all functional parts.

Implementation of hardware and software parts of the UBIMOS infrastructure are described here. Functionally, such an infrastructure is composed of three main subsystems: (i) agents (UAs), (ii) intermediate message passing units (IMPs), (iii) command control centers (3Cs). Both the agents and IMPs are mobile units carried by humans, mounted on vehicles or carried by other subjects, as necessary. On the other hand, 3Cs are stationary decentralized systems that remotely conduct all operations of a given domain and coordinate emergency response units/forces that are mobile on the field.

Based on a major technical requirement, nodes of a UBIMOS network are designed to run effectively on ARM–based small systems. An agent is, indeed, a powerful small computer (or a microcontroller) that remotely collects and processes sensor data from individuals and operation environments and transports the aggregated data to a command control center. That is, agents collect and process sensor data and broadcast them to other UAs and IMPs, which than forward the data to the emergency center. In general, an agent is a mobile client system that monitors some specific parameters of a given subject and transmits them via IMPs to a remote emergency center and obeys instructions received from that center, i.e., the command control center.

Software and hardware modules of both IMPs and UAs are designed to run on embedded systems, having small sizes, but powerful enough for collecting, processing, exchanging the necessary sensor data, and communicating securely with other nodes. Figure 2 shows the structural organization of an ARM-based node specifically designed for the UA and IMP nodes. The UBIMOS runtime environment is designed as a virtual machine with limited functionality, which manages all SW modules that receive sensor data, process the data, and prepare data packets for the transportation. Most obviously, the virtual machine performs scheduling and dispatching of processes needed for all type of data processing and communication software. The virtual machine is also responsible for system diagnoses, debugging, maintenance, and for the execution of UBIMOS processes on the tiny Linux kernel. The sensor interface module is responsible for collecting the physical sensor data and passing them to the sensor data processing module. The transport and security module is a light weight transport

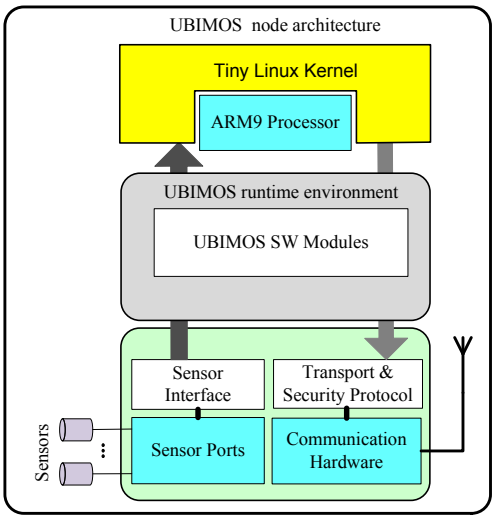

Figure 2. The structural organization of the wearable computer used as UAs and IMPs

protocol implemented with data encryption, decryption, and secure node authentication functions.

Both the UA and IMP nodes are small hand-held devices that are embedded with sensors, while the 3C nodes are general purpose stationary computer systems with server capabilities. A 3C node receives the sensor data from UAs and IMPs, analyses them, displays the locations of UAs and IMPs on a map. Furthermore, vital conditions of the remote users and environmental conditions of the operation field are also analyzed and necessary actions are taken by coordinating necessary instructions with the operation forces. The command control centers also encrypt and store the communication and log information about all events into a database.

In short, a UBIMOS agent works as a mobile system observing, collecting, and distributing vital body conditions of its user and environmental conditions wherever the user and/or objects might be ubiquitously located. As mentioned later, IMPs operate both as an agent and a gateway, while the command control center coordinates and manages the domain operations of a given set of UBIMOS domains. Indeed, an IMP works as a sink that collects the sensor data form agents of a given domain and forwards the data to a dedicated command control center. UAs also do routing, but not as comprehensive as IMPs do. UAs use simple hop–by–hop multi–path routing that only select and flood domain specific sensor data while discarding unrelated, duplicate, and obsolete sensor data packets.

B. Sensor Data Processing

In order to facilitate prioritizing in routing and effectively processing of sensor data, a task based classification of the

sensor data is designed. Sensor data are aggregated and packed following a special packet format shown in Figure 3. The first field of the packet holds an urgent flag, the second field contains a four–bit code identifying 16 different sensor types. The third field contains the sensor data to be queued for broadcasting, the Memory address field contains the address of the first memory location of the sensor data. Finally, the last field contains the length of the sensor data residing in the memory. The urgent flag can be used by

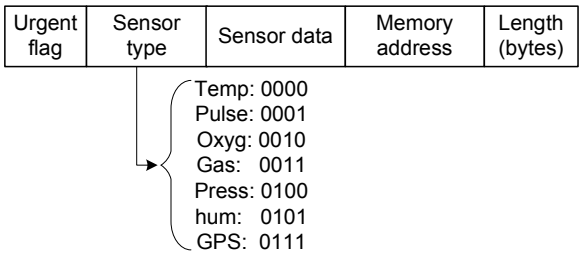

Figure 3. Sensor data format for the process queue

sensor applications, where depending on the application type, each application can explicitly set the urgent flag. For example, some of the ambient data or vital body signs, e.g., temperature or blood pressure may increase unduly, then the related sensor reading code will set its urgent flag. The sensor data processing module checks first whether the urgent flag is set, if so the related sensor data are packed and put in front of the queue for an immediate processing.

There are several reasons to set the urgent flag, (i) if an interrupt is received from the urgent button (hardware interrupt), (ii) the value read from a sensor exceeds its predefined normal range, (iii) the node goes into sleep mode for power saving mode, (iv), the node has reached minimum power level, (v) the node voluntarily changes its state from operative to idle in order to facilitate the self-optimization of the domain it belongs to. If a node announces itself as being idle, then the neighboring nodes delete the respective entry in their routing table in order to reduce the power consumption. This action is taken by the self-optimization algorithm invoked periodically. During the data aggregation from the sensors, excessive values are detected and packed accordingly with the urgent flag set, and then put in front of the process queue for broadcasting. The pseudo–code shown in Algorithm 1 describes the polling of sensor ports, data aggregation from the sensors, and queuing and broadcasting of sensor data.

As observed, we use the notation of object–oriented programming paradigm, where Obj $\rightarrow$ Val denotes a referral to parameter Val of object Obj. As usual, an object defines a conglomerate data structure associated with a given class. For example, Sensor $\rightarrow$ Value := readPort (SP) means that the algorithm reads sensor port SP and assigns the value read to Value parameter of the sensor object Sensor. Thus,

Algorithm 1: Sensor data aggregation and packet generation

```
Algorithm Readsensors()
input : Port addresses of sensors (SP) from 1 to 7
output: Sensor data packets into broadcast queue
for SP ← 1 to 7 do
    if Ready(SP) then
        Sensor := createSensor(SP) ;
        Sensor → Value := readPort(SP) ;
        Sensor → Type := assignType(SP) ;
        Sensor → Urgent := isUrgent(Sensor → Value) ;
        Enqueue(Sensor) ;
    end if
end for

Algorithm Broadcast()
input : Sensor data queue
output: Broadcast sensor data via RF module
while (¬ eof(sensorQueue)) do
    Sensor toSend := Dequeue(sensorQueue);
    Packet Pb := makeBroadcastPacket(toSend);
    Send(RFport, Pb);
end while
```

the above code first generates a sensor object for each sensor, reads and stores the sensor data into the associated sensor objects, checks the urgent flag, and puts the sensor data into a processing queue. This queue is then traversed, sensor data objects (packets) are converted to transport packets, and queued on the RF port for broadcasting. Since most sensors return analog data, the readPort() function reads the analog quantity, converts it to binary form, and stores the result into a memory location. The isUrgent() function reads the binary sensor data from the associated memory location and makes a quick computation (threshold masking) to determine the current value of the urgent flag.

V. Structural View of the UBIMOS Subsystems

As described below, the ubiquitous emergency management system realized so far is designed in three separate subsystems: ubiquitous agent (UA), intermediate message passing (IMP) subsystem, and command control center (3C). Based on internationally recognized standards, the implementation of these subsystems confirms to emerging technologies regarding the software formats and capabilities of small-sized hardware.

A. Agent Subsystem

An agent subsystem is responsible for collecting sensor data, preprocessing the data, and transmitting them to the nearest IMP that has bridging capabilities between the sensor agents and a 3C node. An agent is also able to receive instructions from command control centers and act upon the contents of the instructions. Several agents and IMPs together with a 3C node are organized into a WSN (so called UBIMOS domain) in order to facilitate the application of the multi–path routing algorithm with controlled power constraints.

Each agent subsystem is implemented in five dependable modules: Communication, Sensor, System diagnose, Control, and Security module. The Communication module of UAs is responsible for transmitting preprocessed sensor data from the environment and/or from the team member to its base station, 3C, via other UAs and IMPs that are reachable throughout its routing path. The data exchange is accomplished by UBIMOS's secure transport protocol, while routing of the sensor packets are carried out by a special hop–by–hop multi–path routing algorithm.

Since the UA nodes are constrained for minimizing the energy consumption and data transfer bandwidth, the routing algorithm must, in addition to the classification of the nodes in a WSN, consider the rejection of duplicate packets. The classification of the sink nodes within a WSN is defined by a flag (*DST_type*) in the transport protocol header. When an IMP node sees a *DST_type* flag set (*DST_type = 1*) it disables flooding the related packet, instead, it forwards the packet to its 3C node. By default *DST_type = 0* on UAs and IMPs in a domain, i.e., multi-path routing is enabled regardless of an unreachable sink node.

Sensor module of an agent subsystem is responsible for collecting and processing data available at sensors' ports. Since the sensors are connected to hardware ports (analog and digital) they are addressed by the respective port addresses, which are denoted as SP to identify a given port.

Security module of the agent subsystem is responsible for providing secure authentication, confidentiality, integrity of data, and availability of its services. Secure authentication, confidentiality, and integrity functions are provided by using a light weight implementation of the RSEP protocol, [11], however availability of services is done by self-organization algorithms and by the frequency hopping technique [29], [30], which is a built in functionality of the RF transceiver module. As already noticed, the RF communication module of the UA subsystem is responsible for the communication between UAs, IMPs, and with the related 3C node. It is clear that the self-optimization algorithm can also enable higher availability of the operational nodes, since the amount of flooding and the number of packets during routing are dynamically reduced. As considered later in this paper, the power consumption is also reduced by periodically running the self-optimization algorithm.

System diagnose module is responsible for probing the sensors and other system parts (including the software modules) whether they are properly functioning. The diagnose module is always invoked once during the system startup, however, it can be executed whenever an overall system diagnose is required by its user.

Control module of the agent subsystem manages all other software modules mentioned above. It is mainly involved in process scheduling and dispatching of the system modules. Checking of urgent flags and priority handling of the sensors are also done by this module.

B. IMP Subsystem

Intermediate Message Passing subsystem is an extended agent unit having some additional features, such as routing to 3C nodes via a satellite, long range RF link, or a GSM link. It uses exactly the same modules for gathering and broadcasting of the sensor data as that of the agent subsystem. Though an IMP can be configured to function as a sink in a predefined UBIMOS WSN, it provides bridging between each UA and the control center (3C) within a given WSN. Thus, it has an additional module for the 3C communication to bridge UAs of a WSN to their 3C nodes and to other WSNs as necessary. Since the modules of the IMP subsystem are extended replicas of the UA subsystem, we omit detailed description of them here, and rather refer to the description of the UA subsystem.

C. 3C subsystem

As mentioned earlier, a 3C node receives the sensor data form UAs and IMPs, analyses them, displays the locations of UAs and IMPs on a map. Furthermore, vital conditions of the remote users and environmental conditions of the operation field are also analyzed and necessary actions are taken to intervene critical situations. Moreover, it encrypts and stores the communication data and log information about all events during an operation into a secure database. Hence, the 3C subsystem is responsible for decision–making operations related to the contents of sensor data received from UAs and IMPs. The decision–making operations include processing of the sensor data, creating and submitting alerts, managing emergency situations, coordinating the operational domains, logging, and securely saving the processed sensor data. Data records for the remote team members, alerts, UAs, and IMPs are kept in an encrypted database. In short, overall management of a given set of operational domains is carried out by a 3C node. Figure 4 illustrates a 3C console, where the sample domain contains two different operation teams, Team 1 and Team 2. The locations shown on the map are built using the GPS data received from the team members. The screen also illustrates the vital body information and some environmental data (e.g., pressure, humidity, and ambient temperature) gathered from the remote users (UAs and IMPs).

Regarding the software components, the 3C subsystem is composed of five software modules, Security, Monitoring, Communication, Logging, and Control module. Additionally, it maintains a database to store event information in an encrypted database.

In cooperation with the Communication module, the Security module of 3C subsystem performs secure authentication and encrypted data exchange with its domain nodes. The Communication module of 3Cs is also responsible for data exchange among the 3C nodes and other systems over the Internet. It differs from the corresponding modules of UAs and IMPs in a way that the 3C nodes can also communicate

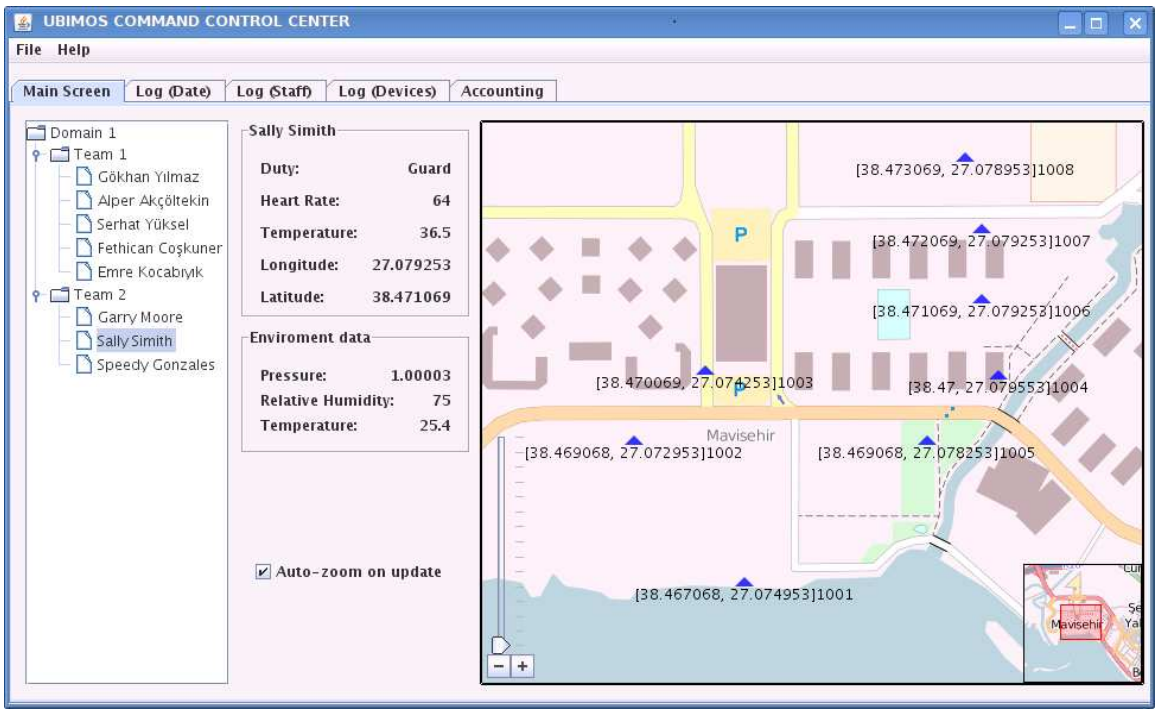

Figure 4. A screenshot of the command control center managing the domain Domain 1

with IMPs via satellite and other long range RF and GSM links.

The Logging module of the 3C subsystem is responsible for securely logging of every single event in detail during the communication. The event data logged so far will also be used to backtrack and compose accountability information on the team members and other subjects found in a given domain. Operations such as insertion, search, and retrieval of the event data are also accomplished by the Logging module in cooperation with the Control module.

The Monitoring module manages real–time monitoring of the remote activities and alerts. For example, the screenshot shown in Figure 4 is processed and displayed by the Monitor module. This module also provides the necessary data to the related command control center needed for its procedural operations such as team organization, coordination, and cooperation of operational domains.

Similar to UAs and IMPs, the Control module of 3C subsystem is responsible for providing a unified interface to underlying modules in order to manage the remaining software modules in a multi–threaded process execution environment.

VI. Setting Up a UBIMOS Domain

For each ubiquitous operation a WSN must be setup by a dedicated command control center, where depending on the type of the operation, either secure or non-secure mode can be initially chosen. For the sake of reliability of the dynamic relocation of the domain nodes, each domain must first specify at least one IMP for relying of data from its UAs. If only one IMP node is available, then the IMP node must have a high availability feature. We can also define some UAs to operate as IMPs if the operation allows. Initial setup of a UBIMOS domain is completed in three main steps:

1) neighbor discovery and authentication of neighbors,
2) self-optimization: *k*–nearest neighbor clustering for preventing nodes from duplicate packet flooding to heavily loaded paths and farther destinations,
3) self-organization: hierarchical clustering of nodes for the detection and configuration of IMPs and 3Cs in a hierarchical task structure.

Figure 5 shows a sample topology of three UBIMOS WSNs organized for a specific operation. Here, basically, the hop–by–hop multi–path routing algorithm is issued with some modifications. The modifications are applied to processing of the duplicate packets and using IMP and 3C nodes as sinks when routing the data packets to their ultimate destinations within a communication pathway. Within these domains, an agent (UA) can only route using flooding within its own domain, while IMPs and 3Cs can perform inter–domain routing. Since IMPs have also limited distance coverage compared to 3C nodes, they do best-effort routing during the data exchange with external domains, either directly to a peer (UA or IMP) or to a 3C node.

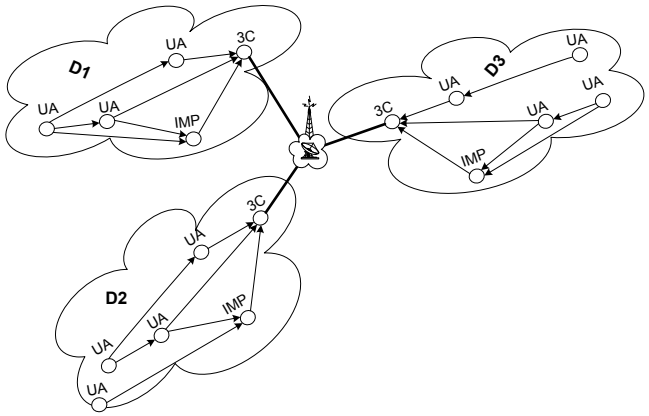

Figure 5. A topological view of three UBIMOS domains configured to interoperate

That is, prior to an operation a UBIMOS domain is constructed (clustered) around at least one IMP node and one 3C node. Clustering of the nodes (UAs, IMPs, and 3Cs) is based on node IDs, functionality, and distances that are measured as time delays for the delivery of data packets. Each node stores a list of node IDs and their cryptographic hash values in a simple data structure. Using the list of these node IDs, each node runs the nearest neighbor clustering (based on Delaunay Triangulation) algorithm for discovering and clustering neighbors, and for assigning appropriate IMPs and 3Cs to their domains, see Figure 6. Details about other localization algorithms can be found in [31].

Figure 6 (a) shows that a clustering of UAs around higher level nodes (IMPs and 3Cs) is realized first and then the IMPs cluster themselves around the 3C nodes, Figure 6 (b). It is important to note that if a UA finds a 3C node, it immediately connects itself to that node without making a distinction between the IMP and 3C. This case is illustrated in Figure 6 (a), where for example, in domain D1 two UAs cluster themselves around a 3C, even though they have an IMP in their domain. Initially, the nodes always search for higher level nodes with shorter distances in the hierarchy for finding an appropriate path.

A. Routing

Power constraints, self-organization, and self-optimization are the main aspects considered when designing the routing policy of UBIMOS. Intensive mobility, long distances between the nodes, and physical obstacles always degrade signal quality of WSNs, which in turn, causes packet delays and frequent packet drops at the nodes. This leads to extreme use of both the channel bandwidth and power consumption at the nodes. To cope up with these and several other known constraints, routing policy for such critical systems should be carefully designed and applied. For example, one item of the UBIMOS roting policy states that an agent (UA node) can only route within its own domain, while IMPs and 3Cs can additionally perform inter–domain routing. This policy prevents a UA from flooding its packets to exterior domains, and hence, significantly reduces the power consumption both for itself and for the flooded nodes. With the inter–domain routing the nodes can exchange sensor data among diverse domains and involve in operations of other domains, which are coordinated by any authorized 3C. Agents are only responsible for collecting and flooding the sensor data within a given operation domain. The part of the algorithm taking care of packet duplicates and hop–by–hop packet forwarding to a specific type of nodes is described by the pseudo–code given in Algorithm 2.

Algorithm 2: Hop-by-hop multi-path routing at the UA nodes

```
Algorithm UAreceive(Packet received)
input : Received packet
output: UA Hop-by-hop Routing
if (received → DSTaddr = thisAddr) then
    Enqueue(processQueue, received) ;
    exit ;
end if
if (received → hopCount > MaxHop) ∨
(Duplicate(received)) then
    drop(received);
    exit ;
end if
if (received → DSTtype = IMP) ∨
(received → DSTtype = CCC) then
    Packet P := repackTosink(received) ;
    Hash h := computeHash(P → ID, P → seqNo) ;
    save(hashTable,h) ;
    Broadcast(P,DSTtype) ;
    exit ;
else
    Hash h := computeHash(received → ID, received →
    seqNo) ;
    save(hashTable,h) ;
    send(RFport,received);
end if
```

```
Algorithm Duplicate(Packet P)
input : Received packet
output: Check the duplication of packets
Hash h := computeHash(P → ID,P → seqNo) ;
if search(h,hashTable) then
    return true ;
else
    return false ;
end if
```

B. Optimization Strategy: Self-Organization for Energy Optimization

During any operation UAs may become quite hectic while exchanging data with each other under various physical conditions. Overall efficiency of an operation mainly depends on faster and reliable communication among the nodes. Periodic self-optimization is required to reduce network traffic bandwidth by decreasing distances and time delays during the data exchange among the active nodes. The self-optimization

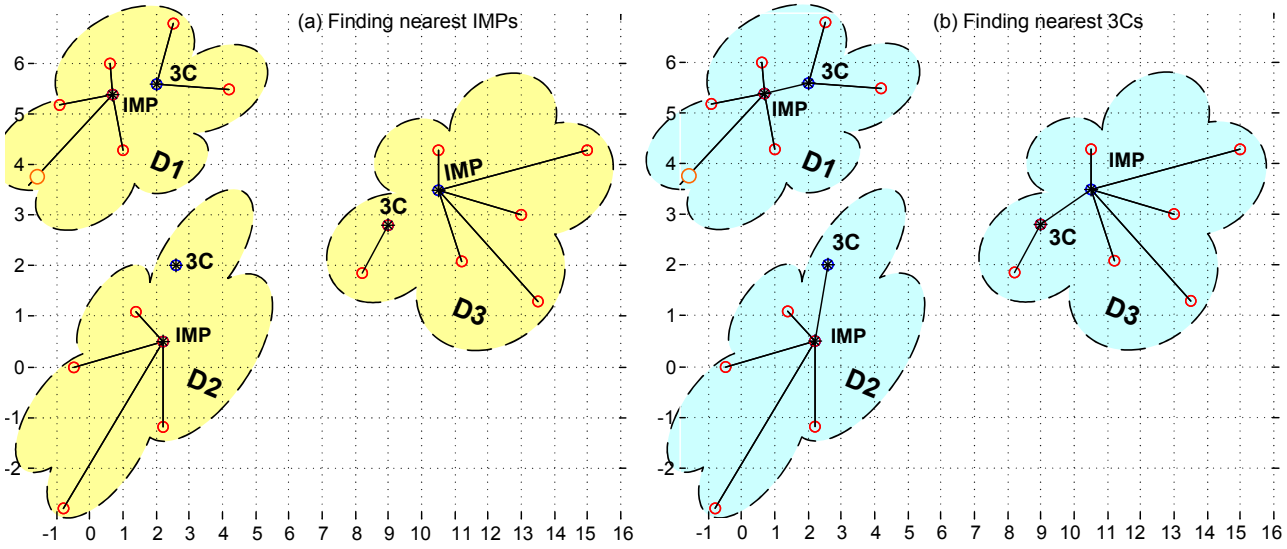

Figure 6. Initialization of UAs and IMPs: UAs cluster around nearest IMP and 3C nodes, while IMPs cluster around the nearest 3C nodes only

policy is mainly realized by dynamic organization/relocation of the active nodes. At the startup of the UA nodes, a self-organization algorithm is run to colonize (insert) nodes into one of the active neighborhoods governed by an IMP. This is generally performed in $O(n \log n)$ running time if a Delaunay Triangulation algorithm is used. Here n denotes the number of nodes traversed for the triangulation. During an active operation, the nodes periodically recompute their new locations in order to dynamically insert themselves into more effective communication paths. The effectiveness is measured by the degree of packet delays among the nodes, rate of packet loss, and size of the convex hull of the domain under consideration. If these constraints can be adequately managed then we can achieve highly effective packet routing and hence substantially reduced energy consumption.

Regarding the periodic relocation, during an operation some UAs may become idle, fall outside of the communication coverage of its operational domain, or switch off (go into sleep mode). In these cases, the domain must reduce its routing coverage and hence flooding space in order to preserve more energy. The nodes are configured to periodically run the optimization algorithm for clustering around *k*–nearest neighbors. The idle nodes set their *IDL* flag in the transport protocol header (next section) to indicate the out of operation status, while other ”unheard” nodes are assumed to be idle if within a given time duration they have not shown any activity. Obviously, *the number of IDL flags that are unset within the newly computed domain boundary* gives us the number of the nearest k nodes for the new cluster size. Thus, clustering with this threshold size gives a new active set of operational domains. Figure 7 illustrates relocating of active nodes around the domains D1, D2, and D3. Blue dots (UA nodes) are assigned to D1, black nodes to D2, and red nodes are assigned to D3. Encircled nodes are clustered around the related IMPs, while the others are distant nodes, which are also able to communicate with their domains if they are in the communication coverage. However, some of the previously active nodes are now out of the operational domains, which can be interpreted as if they were either become voluntarily idle or forced to be off. Most

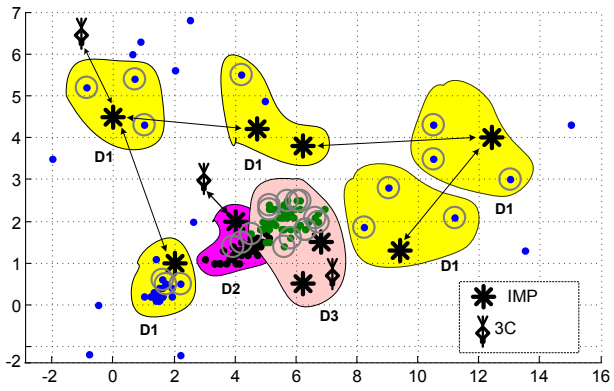

Figure 7. Clustering around *k*–nearest neighbors

importantly, during the initial setup of a UBIMOS domain, the nodes are clustered into a WSN based on the nearest neighbor algorithm. That is, the nodes that are close to each other are organized into a dedicated routing level. The UA nodes are first clustered into nearest neighbor clusters among themselves, where each cluster is assigned a routing level with other clusters. These clusters are than hierarchically clustered towards the nearest IMP. Further, the nearest IMPs in the hierarchy are also clustered towards their nearest 3Cs. Figure 8 shows the clustering of three UBIMOS domains, D1, D2, and D3. By this clustering, the number of active routing (flooding) nodes reduces to the number of joint

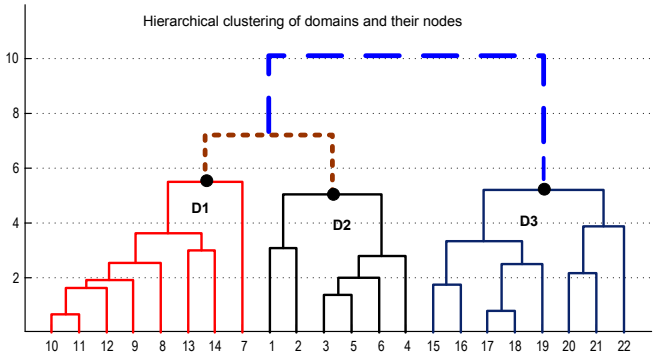

Figure 8. A dendrogram showing the hierarchical clustering around nearest IMPs and 3Cs using the Minkowski metric

points of the entire domain. For example, as shown in Figure 8, after clustering the UAs, the number of flooding nodes of the domain D1 reduces to 6, which is 8 for the plain multi-path flooding algorithm. The clustering of the nodes in this example are done binary-wise, however, since we use average distance to compute the routing level of a cluster we can organize each cluster as a splay tree. This can further reduce the number of flooding points significantly compared to the binary tree.

VII. Secure Communication

Some operations require secure authentication during the connection establishment, and confidentiality and integrity of data during the data exchange. Especially, communications needed for anti–terror operations, public transportations, and civil defense must be securely carried out. Rescue operations related to public safety operations such as urgent health-care services, earthquake, snowslide, mugslide, fire, flooding, and other natural disaster rescue operations need not be confidentially done, however, integrity must always be ensured. For example, during a rescue operation dealing with flooding or fire catastrophes, operation team member accountability and information on vital signs, including body temperature, pulse rate, respiration rate, and blood pressure must be reliably and quickly transmitted to the control center. However, during anti–terror or highly confidential operations data should be exchanged both confidentially and reliably.

It is important to note that, security mechanisms, such as integrity, peer authentication, and confidentiality, are applied to all types of nodes. However additionally, the command control centers must encrypt before saving the sensor data and communication information that are logged during the domain operations. That is, prior to a connection establishment, each node must use secure authentication regardless of the type of the domain operation. However, during the data exchange phases, nodes may perform unencrypted data transport, depending on the operation type. Unless specified, the command control center saves the data in the encrypted form. Nevertheless, during the initial setup of a UBIMOS domain, nodes do authenticate each other using a light weight implementation of the RSEP protocol, [32].

A. Transport Protocol

Transport protocol of UBIMOS nodes is thus accordingly designed to ensure integrity, confidentiality, and secure authentication. Packet format of the transport protocol is shown in Figure 9. One octet is reserved for flags, where

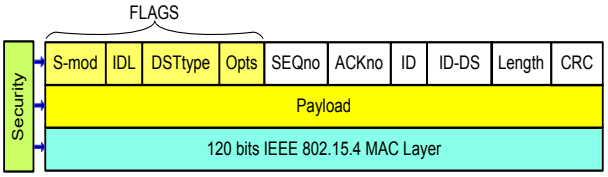

Figure 9. The structure of the Transport Protocol of UBIMOS.

three of the flags are predefined for security, idle nodes, and for destination type, and the remaining five flags are optional and available to other applications. The first flag, *S-mod*, is used for switching to secure mode, *IDL* is used to inform destinations if the source node is in one of idle, sleep, or in off mode. *DSTtype* identifies the type of the destination, whether the destination is a sink, i.e., IMP or 3C. We have two 16-bit fields storing the packet sequence number (*SEQno*) and acknowledgment (*ACKno*), which are comparable to that of the TCP fields. The *Length* field contains a 16-bit length of the entire packet defined as number of octets. The final field, *CRC*, is indeed a hash value of either the entire packet or the header only. This field is used to ensure the integrity of the received data. In the secure mode, *CRC* contains the hash value of only the header, however, for the insecure mode this contains the hash value of both the header and the payload (data). In the secure mode the CRC of payload is not necessary since the payload will always be hashed by the security protocol.

Regarding the *ID* field, for the implementation of the required security functionality, each node is associated with an ID defined as a lightweight X–509 certificate, which has the function of describing current ID of the source node, operation team member, and the operation code (or the operation ID). During the authentication, every node (UA, IMP, and 3C) should present its certificate, where each certificate, among others, is comprised of the issuer of the certificate, a legitimate team member ID, and a unique ID of the node. That is, each node is assigned a unique ID, users's public key information, name and digital signature of the issuer, time stamp, serial number, version, last update date, period of validity, and digital signature algorithm (the algorithm used to sign the certificate). The reader is referred to the description of CCITT X.509 v3 [33] for further details, which defines a standard certificate format for public key certificates and required procedures for the certification validation.

A digital signature, which is generated from the sender node ID, is stored in the *ID-DS* field. This field is used by the receiver to securely authenticate the source node. To do so, the receiver, during the domain setup, computes and stores a list of hash values of the IDs of its domain members. Later, during the authentication, the receiver computes a new hash value of the ID field coming from the remote node, and compares the newly computed hash value with the one already registered in the list. If these hash values are equal then the authentication of the remote node will be verified and acknowledged back to the sender.

In order to reduce the energy consumption and avoid redundancy in security mechanisms, the security functionality of the IEEE 802.15.4/ZigBee Protocol is not used. That is, in the secure mode, in order to ensure secure authentication of the UBIMOS nodes and confidentiality of the data exchanged between the nodes, all communications are tunneled with a light weight implementation of the RSEP protocol [11], [34] using the elliptic curve algorithm [35].

VIII. Prototype Implementation

Ubiquitous agents and IMPs are designed to run on ARM–based platforms, whereas mainstream computers are chosen to be used as the 3C servers. In essence, the 3C servers are stationary and relatively high-capacity systems, which must perform resource demanding operations such as encrypted database functions, satellite communication, monitoring, real–time alerts, and real–time rescue management tasks.

As the ARM processor we have used TS-7350® [36], which is a compact full-featured single board computer based on the Cirrus EP9302 200MHz ARM9 CPU, which allows development of multi-function embedded applications through its multiple peripheral interfaces. The ARM processor is a 32-bit reduced instruction set computer (RISC). It was known as the Advanced RISC Machine, and before that it was known as the Acorn RISC Machine. The ARM architecture is a widely used 32-bit RISC processor, which was originally conceived as a processor for desktop personal computers by Acorn Computers. The relative simplicity of ARM processors made them suitable for low power applications. This has made them dominant CPUs in the mobile and embedded electronics market as relatively low cost and small microprocessors and microcontrollers.

The sensors were designed separately and connected to related input connector pins. For the GPS module, we have used Atheros AR1511®, which consists of a tiny CMOS AR1511 GPS IC, a highly-integrated GPS receiver comprised of a single conversion RF front-end and a GPS baseband processor all combined on a single die. Additionally, we have used five other sensors: OTP-538U for body and environment temperature measurement, SDT1-028K for heart rate, MQ135 for gas and air quality, SHT75 for humidity, HP03D for pressure measurement.

A. Reviewing the Implementation

We have focused on the power consumption and the efficiency of the choice of the programming language used for the implementation. Generally, main limitations of nodes in a sensor network relate to power consumptions and necessary energy–saving algorithms. Battery life for such mobile units can be slightly prolonged by use of efficient routing algorithms and denial of sleep protection mechanisms. For the security enhanced operations, UBIMOS nodes will naturally consume more power due to secure authentication, encryption, and decryption algorithms used for the confidentiality of the data exchanged.

Use of the development language has a crucial role in both the transmission speed and the power consumption. Therefore, we have implemented the system both in Java and C++ languages. Following the implementation, we conducted several test and evaluations of the system on two different CPU architectures, ARM and Intel®. Since the software components other than the security and transport modules run in constant time, we evaluated only the implementation of the security and transport modules. Encryption, decryption, and secure authentication algorithms require extensive CPU and data transmission resources, all depending on the size of data blocks being handled. The results of running times versus input block sizes for Java and C++ implementations of the security module are shown in Figure 10. Although the encryption algorithm can use larger key sizes, we have used 116-bit elliptic curve algorithm and 128-bit RC4 stream encryption algorithm in the prototype version of UBIMOS. These key lengths are known to be relatively moderate. As known, the key length in an encryption algorithm increases the strength of the algorithm while decreasing the time-efficiency. Hence, the key length can be increased for the 3C systems, but for the mobile nodes 116-bit elliptic curve and 128-bit RC4 are optimum.

Finally, we have experimented with a secure real–time operation using 5 UAs, 2 IMPs, and 2 3Cs in order to observe the efficiency of the communication speed. The nodes were spread around an area of 50 km/radius. Both the simulation and the real–time operation results with this size of network were shown to be successful. However, simulation results of a real–time operation have shown that increasing the number of UA and IMP nodes beyond 200 nodes caused a running time of 10 ms/packet per UA node, which caused an unacceptable delay in total for the security–enabled operation. This shows that, in order to increase the communication efficiency, a UBIMOS WSN with larger sizes should be segmented and bridged using more IMPs. Although the prototype implementation of UBIMOS gave satisfactory results, in a future work, we need to perform more efficiency analysis regarding the power consumption, communication speed, and strength of the security functions.

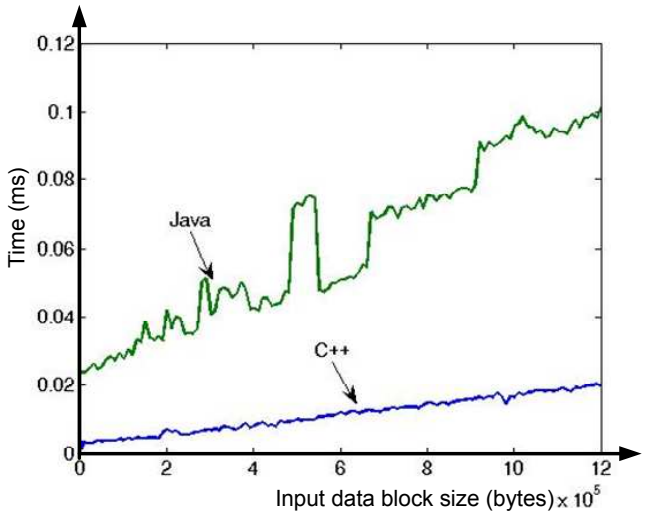

Figure 10. Running times of the secure authentication; C++ implementation versus Java implementation

IX. Conclusions

We have designed and implemented a ubiquitous management system to exchange vital body signs, environmental conditions, and locations of subjects during critical rescue operations. Data on the environmental conditions and vital body signs known as heart rate, oxyhemoglobin saturation, and thermal stress factors were successfully collected by ubiquitous nodes within a WSN, and transmitted to the command control centers for further processing. The results obtained from the simulations and the prototype implementation will allow us to refine new assumptions made when designing future hardware, software, protocols and mechanisms for more critical operations. Although the prototype implementation of UBIMOS gave satisfactory results, in a future work, we need to perform detailed efficiency analysis regarding the power consumption, communication speed, and strength of the security functions.

X. Acknowledgment

We thank our students Gökhan Yilmaz, Emre Kocabiyik, Fethican Coskuner, Alper Akçöltekin, and M. Serhat Yüksel for helping us in coding and testing a prototype system, and special thanks to Computer Sciences Laboratory staff of Izmir University of Economics for their support in facilitating the laboratory experiments during the implementation of the prototype system. Finally, we owe special thanks to the management of the Faculty of Engineering & Computer Sciences for funding the entire project.

References

[1] S. Kondakci, G. Yilmaz, E. Kocabiyik, F. Coskuner, A. Akcoltekin, and M. S. Yuksel, "Ubiquitous monitoring system for critical rescue operations," in *Proceedings of the 2010 6th International Conference on Wireless and Mobile Communications*, ser. ICWMC '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 515–520. [Online]. Available: http://dx.doi.org/10.1109/ICWMC.2010.103

[2] J. L. W. K. J. Ruskin, "Pulse oximetry: basic principles and applications in aerospace medicine," *Aviation, space, and environmental medicine*, vol. 78, no. 10, pp. 973–978, October 2007.

[3] R. B. Hetnarski and M. R. Eslami, *Thermal Stresses – Advanced Theory and Applications (Solid Mechanics and Its Applications)*, 1st ed. Springer, December 2008. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/1402092466

[4] "Council of Europe Convention on the Prevention of Terrorism," accessed June 2010. [Online]. Available: http://conventions.coe.int/Treaty/EN/Treaties/Html/196.htm

[5] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.

[6] K. Romer and F. Mattern, "The design space of wireless sensor networks," *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 54 – 61, dec. 2004.

[7] A.-K. Chandra-Sekaran, C. Kunze, K. D. Mller-Glaser, and W. Stork, "Self-organizing zigbee network and bayesian filter based patient localization approaches for disaster management," *International Journal on Advances in Intelligent Systems*, vol. 2, no. 4, pp. 446 – 456, 2009. [Online]. Available: http://www.iariajournals.org/intelligent_systems/

[8] K. Sha, W. Shi, and O. Watkins, "Using wireless sensor networks for fire rescue applications: Requirements and challenges," in *Electro/information Technology, 2006 IEEE International Conference on*, May 2006, pp. 239–244.

[9] X. Du and H.-H. Chen, "Security in wireless sensor networks," *Wireless Communications, IEEE*, vol. 15, no. 4, pp. 60 –66, aug. 2008.

[10] H. Kumar, D. Sarma, and A. Kar, "Security threats in wireless sensor networks," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 23, no. 6, pp. 39 –45, june 2008.

[11] S. Kondakci, "A remote IT security evaluation scheme: A proactive approach to risk management," in *IWIA '06: Proceedings of the Fourth IEEE International Workshop on Information Assurance*, vol. 1. Washington, DC, USA: IEEE Computer Society, 2006, pp. 93–102.

[12] M. Ismail and M. Sanavullah, "Security topology in wireless sensor networks with routing optimisation," in *Wireless Communication and Sensor Networks, 2008. WCSN 2008. Fourth International Conference on*, dec. 2008, pp. 7 –15.

[13] W. K. Koo, H. Lee, Y. H. Kim, and D. H. Lee, "Implementation and analysis of new lightweight cryptographic algorithm suitable for wireless sensor networks," in *Information Security and Assurance, 2008. ISA 2008. International Conference on*, april 2008, pp. 73 –76.

[14] A. Boukerche, Y. Du, J. Feng, and R. Pazzi, "A reliable synchronous transport protocol for wireless image sensor networks," in *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, july 2008, pp. 1083 –1089.

[15] Y. W. Law, M. Palaniswami, L. V. Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-efficient link-layer jamming attacks against wireless sensor network mac protocols," *ACM Trans. Sen. Netw.*, vol. 5, no. 1, pp. 1–38, 2009.

[16] M. Brownfield, Y. Gupta, and N. Davis, "Wireless sensor network denial of sleep attack," in *Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC*, june 2005, pp. 356 – 364.

[17] M. Mura, F. Fabbri, and M. Sami, "Modelling the power cost of security in wireless sensor networks : The case of 802.15.4," in *Telecommunications, 2008. ICT 2008. International Conference on*, june 2008, pp. 1 –8.

[18] N. Wakamiya, S. Arakawa, and M. Murata, "Self-organization based network architecture and control technologies for new generation networks," *International Journal on Advances in Intelligent Systems*, vol. 3, no. 1 & 2, pp. 75–86, 2010. [Online]. Available: http://www.iariajournals.org/intelligent_systems/

[19] K. Yamasaki and T. Ohtsuki, "Design of energy-efficient wireless sensor networks with censoring, on-off, and censoring and on-off sensors based on mutual information," in *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol. 2, may-1 june 2005, pp. 1312 – 1316 Vol. 2.

[20] H. Shaban, M. El-Nasr, and R. Buehrer, "A framework for the power consumption and ber performance of ultra-low power wireles swearable healthcare and human locomotion tracking systems via uwb radios," in *Signal Processing and Information Technology (ISSPIT), 2009 IEEE International Symposium on*, dec. 2009, pp. 322 –327.

[21] D. Walsh, S. Capaccio, D. Lowe, P. Daly, P. Shardlow, and G. Johnston, "Real time differential gps and glonass vehicle positioning in urban areas," *Space Comms.*, vol. 14, no. 4, pp. 203–217, 1997.

[22] M. Sahmoudi and M. G. Amin, "Robust tracking of weak gps signals in multipath and jamming environments," *Signal Process.*, vol. 89, no. 7, pp. 1320–1333, 2009.

[23] R. Ennaji and M. Boulmalf, "Routing in wireless sensor networks," in *Multimedia Computing and Systems, 2009. ICMCS '09. International Conference on*, april 2009, pp. 495 –500.

[24] J. Heo, J. Hong, and Y. Cho, "Earq: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks," *Industrial Informatics, IEEE Transactions on*, vol. 5, no. 1, pp. 3 –11, feb. 2009.

[25] S.-C. Jung and H.-K. Choi, "An energy-aware routing protocol considering link-layer security in wireless sensor networks," in *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, vol. 01, feb. 2009, pp. 358 –361.

[26] S. Khokhar and A. A. Nilsson, "Estimation of mobile users trajectory in mobile wireless network: Framework, formulation, design, simulation and analyses," *International Journal on Advances in Intelligent Systems*, vol. 2, no. 4, pp. 387–410, 2009. [Online]. Available: http://www.iariajournals.org/intelligent_systems/

[27] D. Jain and V. Vokkarane, "Energy-efficient target monitoring in wireless sensor networks," in *Technologies for Homeland Security, 2008 IEEE Conference on*, may 2008, pp. 275 –280.

[28] L. B. Saad and B. Tourancheau, "Towards an optimal positioning of multiple mobile sinks in WSNs for buildings," *International Journal on Advances in Intelligent Systems*, vol. 2, no. 4, pp. 411 – 421, 2009. [Online]. Available: http://www.iariajournals.org/intelligent_systems/

[29] C. Ding, R. Fuji-Hara, Y. Fujiwara, M. Jimbo, and M. Mishima, "Sets of frequency hopping sequences: Bounds and optimal constructions," *Information Theory, IEEE Transactions on*, vol. 55, no. 7, pp. 3297 –3304, july 2009.

[30] M. Strasser, C. Pöpper, and S. Čapkun, "Efficient uncoordinated fhss anti-jamming communication," in *MobiHoc '09: Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2009, pp. 207–218.

[31] B. Niehöfer, A. Lewandowski, R. Burda, C. Wietfeld, F. Bauer, and O. Lüert, "Community map generation based on trace-collection for gnss outdoor and rf-based indoor localization applications," *International Journal on Advances in Intelligent Systems*, vol. 3, no. 1 & 2, pp. 1 – 11, 2010. [Online]. Available: http://www.iariajournals.org/intelligent_systems/

[32] S. Kondakci and G. Yilmaz, "Implementation and performance evaluation of the RSEP protocol on ARM and Intel platforms," in *SIN '10: Proceedings of the 3rd international conference on Security of information and networks*. New York, NY, USA: ACM, 2010, pp. 194–202.

[33] CCITT, "The directory authentification framework," Draft Recommendation X.509, 1987, version 7.

[34] S. Kondakci, "A high level implementation of the RSEP protocol," in *ISC'07:Int. Conf. on Information Security & Cryptology*, vol. 1. ISC Turkey, December 2007, pp. 63–69.

[35] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987. [Online]. Available: http://www.jstor.org/stable/2007884

[36] "ARM9 Processor Development Board," accessed June 2010. [Online]. Available: www.embeddedarm.com