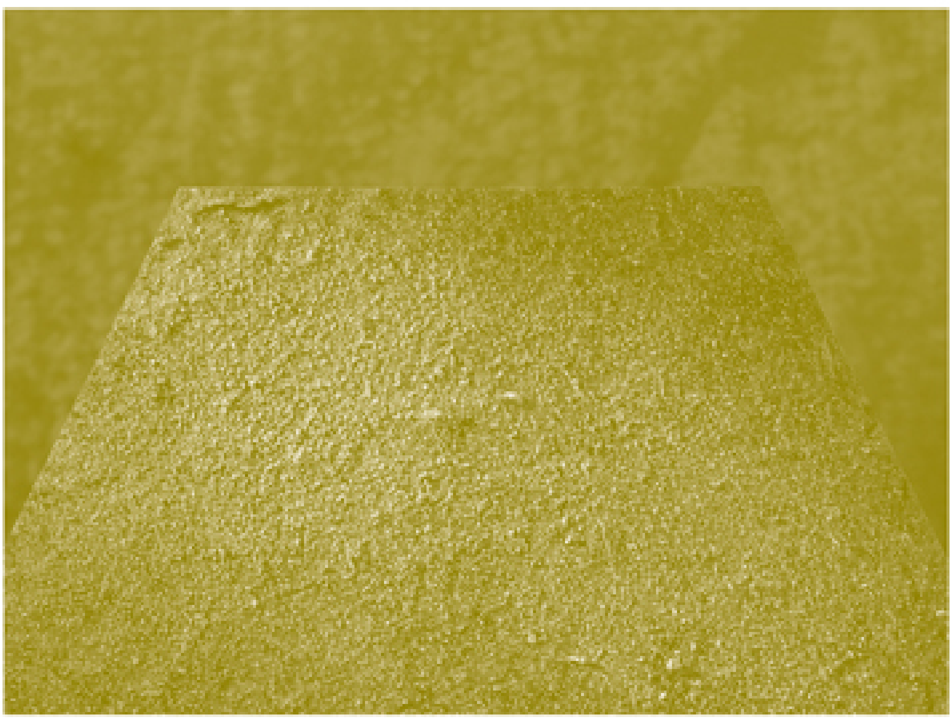
An official publication of
the Information Resources
Management Association

INTERNATIONAL JOURNAL OF Grid and High Performance Computing



IGI PUBLISHING
Publisher of IT books, journals and cases since 1988
www.igi-global.com

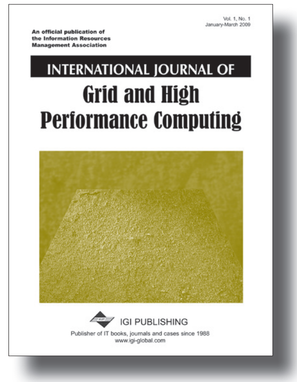
IGIP IGI PUBLISHING
WWW.IGI-GLOBAL.COM

CALL FOR ARTICLES

International Journal of Grid and High Performance Computing

An official publication of the Information Resources Management Association!

The Editors-in-Chief of the *International Journal of Grid and High Performance Computing* (IJGHPC) would like to invite you to consider submitting a manuscript for inclusion in this scholarly journal. The following describes the mission, coverage, and guidelines for submission to IJGHPC.



ISSN 1938-0259
eISSN 1938-0267
Published quarterly

MISSION:

The primary mission of the *International Journal of Grid and High Performance Computing* (IJGHPC) is to provide an international forum for the dissemination and development of theory and practice in grid and cloud computing. IJGHPC publishes refereed and original research articles and welcomes contributions on current trends, new issues, tools, societal impact, and directions for future research in the areas of grid and cloud computing. This journal is targeted at both academic researchers and practicing IT professionals.

COVERAGE/MAJOR TOPICS:

Topics can include (but are not limited to) the following:

- Advanced collaboration techniques and scaling issues
- Bio-inspired grid resource management
- Cloud architectures
- Cloud business process integration
- Cloud foundation concepts
- Cloud platforms and infrastructures
- Cloud reliability and security
- Cloud services
- Cloud standards
- Cloud types
- Combating global terrorism with the world wide grid
- Emerging standards for organizations and international projects
- Future of grid, trends, and challenges
- Grid and software engineering aspects
- Grid architecture, resources, and data management
- Grid education and applications - science, engineering, and business
- Grid evolution, characterization, and concepts
- Grid fundamentals, algorithms, and performance analysis
- Grid impact, scientific, and industrial and social implications
- Grid instrumentation, measurement, and visualization
- Grid portals and security
- Grid programming, models, tools, and API
- Grid services, concepts, specifications, and frameworks
- Grid uses and emerging technology
- New initiatives, SOA, autonomic computing, and semantic grid
- Simple API for grid applications (SAGA)
- Software and hardware support for HPC
- Test, evaluation, and certificate presentation
- Wireless and optical grid, characteristics, and applications
- Work flow management

All submissions should be emailed to:
udohe123@yahoo.com

For full submission guidelines, please turn to the back of this journal or visit the IGI Global website at www.igi-global.com.

Ideas for Special Theme Issues may be submitted to the Editor-in-Chief.

Please recommend this publication to your librarian. For a convenient easy-to-use library recommendation form, please visit: http://www.igi-global.com/ijghpc and click on the "Library Recommendation Form" link along the right margin.

INTERNATIONAL JOURNAL OF GRID AND HIGH PERFORMANCE COMPUTING

July-September 2010, Vol. 2, No. 3

Table of Contents

G2G:
A Meta-Grid Framework for the Convergence of P2P and Grids

Wu-Chun Chung, National Tsing Hua University, Taiwan

Chin-Jung Hsu, National Tsing Hua University, Taiwan

Yi-Hsiang Lin, National Tsing Hua University, Taiwan

Kuan-Chou Lai, National Taichung University, Taiwan

Yeh-Ching Chung, National Tsing Hua University, Taiwan

ABSTRACT

Grid systems integrate distributed resources to form self-organization and self-management autonomies. With the widespread development of grid systems around the world, grid collaboration for large-scale computing has become a prevalent research topic. In this paper, the authors propose a meta-grid framework, named the Grid-to-Grid (G2G) framework, to harmonize autonomic grids in realizing a grid federation. The G2G framework is a decentralized management framework that is built on top of existing autonomic grid systems. This paper further adopts a super-peer network in a separate layer to coordinate distributed grid systems. A super-peer overlay network is constructed for communication among super-peers, thus enabling collaboration among grid systems. This study proposes the G2G framework for use in a Grid-to-Grid federation and implements a preliminary system as a demonstration. Experimental results show that the proposed meta-grid framework can improve system performance with little overhead.

Keywords: *Convergence, Grid Computing, Grid-to-Grid (G2G), Peer-to-Peer (P2P), Super-Peer*

INTRODUCTION

A grid computing system is a distributed computing system for solving complex or high-performance computing problems as encountered in bioinformatics, healthcare systems, ecosystems, and even experiments involving the use of the Large Hadron Collider. In such a computing environment, a virtual organization is a self-organization and self-management group which shares the computing resources (Foster, Kesselman, & Tuecke, 2001). Grid systems employ the middleware as the abstract interface to integrate large-scale distributed computing resources. Therefore, the aggregated capability for distributed computing and data accessing can be improved by integrating geographical distributed resources.

Various organizations, institutions, and private communities around the world adopt

DOI: 10.4018/jghpc.2010070101

centralized or hierarchical architectures to develop grid systems based on the Open Grid Service Architecture (OGSA) (Foster, Kesselman, Nick, & Tuecke, 2002). Most grid systems, though, cannot be integrated for collaborative computation. This is why grid collaboration for large-scale computing has become a prevalent research topic. The integration of distinctly autonomic grid systems into a grid federation is one prospective approach. What poses a challenge to realizing grid federations, however, is how to harmonize various grid systems without bringing a heavy burden on existing grid infrastructure.

In order to coordinate multiple diverse grid systems, a grid system requires a mechanism for achieving the cross-grid convergence of diversely autonomic grid communities. There are two ways to accomplish cross-grid integration. One is to enhance the grid middleware by modifying the original mechanism in existing grid systems, while the other is to develop a meta-grid framework on top of existent grid systems. The former burdens an existent grid system with lots of efforts to harmonize with other grid systems; moreover, there is no mature cross-grid middleware for integrating with distinct grid systems. On the other hand, the latter increases the extra overhead for the existent grid systems. In this paper, we present a meta-grid framework, named the Grid-to-Grid (G2G) framework, to form a federation consisting of multiple institutional grid systems. The G2G framework can harmonize autonomic grid systems and achieve cross-grid collaborative computing with seamless modification for existing grid systems.

Past studies have applied decentralized approaches to exploit the system scalability of grids in the development of the grid management architecture. In general, a centralized or hierarchical architecture is not suitable for large-scale grids (Mastroianni, Talia, & Verta, 2007). Reasons for this include the potential bottleneck at root, the scalability limit of a grid, and load imbalance problems. Integrating grid systems with the P2P paradigm can improve the scalability of a grid federation. Therefore, we attempt to exploit a decentralized G2G framework to realize synergy between P2P networks and existing grid systems. In consideration of the scalability and the efficiency of the grid management architecture, this study utilizes a super-peer network (Yang & Garcia-Molina, 2003) to develop our G2G framework for the coordination of multiple autonomic grid systems. Each super-peer represents an autonomic grid system in our G2G system. To achieve a decentralized G2G framework, the super-peer network adopts an overlay network for communication among super-peers in different grid systems through a federation of wide-area grids.

This paper introduces a meta-grid framework of the G2G system based on a super-peer network. We also present a preliminary implementation of the proposed G2G framework and develop a Grid-to-Grid network based on the overlay network in which each grid system communicates and negotiates with other grid systems. The remainder of this paper is organized as follows. Section 2 discusses related works. In Section 3, we present the overview of the G2G framework and the implementation of a G2G prototype. The experimental results of the G2G system are shown in Section 4. We conclude this paper with the future work in Section 5.

RELATED WORKS

Scalability in large-scale grid systems has posed research challenges in recent years. There are some studies (Ranjan, Harwood, & Buyya, 2008; Trunfio et al., 2007) that discuss the adoption of the P2P technique to improve the scalability of grid systems. Some similarities and differences between P2P computing and grid computing were presented in the literature (Foster & Iamnitchi, 2003; Talia & Trunfio, 2003). Several previous studies aimed to improve the centralized-based grid infrastructure by using the P2P technique.

A decentralized event-based object middleware, DERMI (Pairot, Garcia, & Skarmeta, 2004), is proposed to favor the scalability problem of a centralized model. A DHT-based DERMI prototype uses Pastry (Rowstron & Druschel, 2001) as its underlying network topology and adopts Scribe (Castro, Druschel, Kermarrec, & Rowstron, 2002) as its publish/subscribe notification service which is built on top of Pastry. However, from a view point of the middleware layer, DERMI can facilitate the wide-area grid. From another view point, our study aims to coexist with the existence of grid systems without modifying the middleware.

The integration of a distributed event brokering system with the JXTA technology has been proposed in the literature (Fox, Pallickara, & Rao, 2005) to enable Peer-to-Peer Grids. In a previous study (Pallickara & Fox, 2003), the authors utilize the NaradaBrokering based on the hierarchical structure in the broker network. By integrating the NaradaBrokering with JXTA, services are mediated by the broker middleware or by the P2P interactions between machines on the edge of Internet. NaradaBrokering aims to present a unified environment for grid computing with a P2P interaction. In addition, the overhead will be costly for NaradaBrokering to maintain the broker network by the hierarchical topology in a dynamically changed network.

A P2PGrid platform based on a two-layer model for integrating P2P computing into the grid is presented in a previous study (Cao, Liu, & Xu, 2007). All grid services are provided in the grid layer while the grid entities or common PCs become peers to negotiate with each other in the P2P layer. The P2PGrid tries to provide a solution for integrating the grid computing environment with the P2P computing ones. In this study, JXTA is adopted for developing JXTA Agents to create peers, deal with dynamics of peer groups, and communicate with peers. Jobs are submitted and dispatched to workers which are organized by the above-mentioned peers on the underlying P2P computing network. By the implementation of the P2PGrid platform, peers on the edge of Internet are able to consume grid services without maintaining grid middleware packages. Adopting a separate layer from an existing grid system is a great benefit because that the original behaves of the grid layer can be preserved without modifications, and that the modification of the P2P manner will not affect the efficiency of the grid layer.

In our study, we present a decentralized meta-grid framework on top of existing autonomic grid systems. The autonomic grids are coordinated based on the super-peer network to form a Grid-to-Grid collaborative computing environment. A super-peer is able to provide/consume the grid services to/from other super-peers in remote grid systems. Super-peers are organized based on an unstructured super-peer overlay to negotiate with each other. By adopting a separate layer, the G2G framework can integrate with existing grid systems without modifying the original mechanisms and policies. In addition to the support of computation services and data services, we also propose a solution for the verification of accessing remote resources by considering the security issues in the Grid-to-Grid environment.

G2G FRAMEWORK AND PROTOTYPE

Currently, most grid systems are deployed according to centralized or hierarchical management approaches. However, these approaches have poor performance in terms of scalability, resiliency, and load-balancing for managing distributed resources (Mastroianni et al., 2007). Centralization and hierarchy are the weaknesses of deploying large multi-institutional grid systems, let alone in the widely internetworking system. In general, the performance by adopting the super-peer model is more efficient and convenient than that without adopting the super-peer model in large-scale computing environments.

In our G2G framework, we utilize the super-peer network to coordinate existing grid systems. The P2P technique is applied for the communication between super-peers. In this section, we describe the concept of the G2G system before introducing the meta-grid framework. At the end of this section, we present a preliminary prototype of the G2G system.

Super-Peer Based G2G System

The super-peer network is proposed to combine the efficiency of the centralized search as well as the features in terms of autonomy, load balance, and robustness of the distributed search. A super-peer is a node that acts both as a centralized server to a set of ordinary nodes and as a coequality to negotiate with other super-peers. In order to achieve the seamless integration of the grids in the G2G system, this study adopts the super-peer network on top of the existing grid systems, and harmonizes existing autonomous grids with each other without rebuilding/modifying any grid system.

Each super-peer in our G2G system acts as a coordinator which is responsible for coordinating a local autonomic grid system and negotiating with other super-peers in remote grid systems. For example, after obtaining a request for task execution, the super-peer first checks whether the request can be processed locally; otherwise, the request will be forwarded to other grid systems by cooperating with other super-peers. For the sake of simplicity, we currently consider only one super-peer deployed on top of the local grid systems. Similar examples have been proposed in past studies (Mastroianni, Talia, & Verta, 2005). Since there are multiple autonomic grid systems in the G2G system, we utilize an unstructured overlay to facilitate the federation of super-peers. In this way, the Grid-to-Grid interactions among distinct grid systems are based on the super-peer network by way of the P2P overlay. Based on a P2P overlay, each grid can easily join the G2G system and supply its resources and services to other grid systems. The resource utilization can be improved after applying the G2G system.

The basic concept and architecture of the G2G system are shown in Figure 1 and Figure 2. The G2G system mainly consists of the Cross-Grid layer and the Local-Grid layer. The Local-Grid layer consists of some autonomic grid systems which are built by the grid middleware to collaborate distributed resources. In the Cross-Grid layer, the G2G layer is responsible for coordinating the super-peers in autonomic grids. These super-peers not only take charge of integrating the autonomic grids by the developed common interfaces but also deal with the negotiation between grid systems in the G2G layer. The Meta-Grid interface is responsible to bridge the Cross-Grid layer and the Local-Grid layer. Using these common interfaces, the Cross-Grid layer can acquire the resources and services from the Local-Grid layer without knowing the policies, mechanisms, or algorithms in the Local-Grid layer. Since the Cross-Grid layer and the Local-Grid layer are independent, the Cross-Grid layer does not need to be modified when the mechanisms in the Local-Grid layer are modified or replaced.

G2G Framework

In this study, a meta-grid framework of the G2G system is proposed for a federation of multiple autonomic grid systems as shown in Figure 3. The G2G framework aims to support the seamless integration of the computing services and the data accessing services in the autonomous grid systems. Therefore, the super-peer in the Grid-to-Grid layer consists of seven components to bridge the Cross-Grid layer and the Local-Grid layer: *the interactive interface*, *the security management*, *the network management*, *the task management*, *the data management*, *the resource management* and *the information service*.

The *task management* component handles job computation, and the *data management* component is responsible for integrating the distributed storage systems. The *network management* component handles the network topology and the G2G interaction between distinct grids. The *resource management* manages the distributed resources in grid systems according to the resource status supported by the *information service* component. The *interactive interface* component deals with the login process for users, and the *security management* component is in charge of the authorization of using grids. When each component would like to negotiate with local grid system, the Meta-Grid Interface exchanges the information. By cooperating

Figure 1. Basic concept of the meta-grid framework

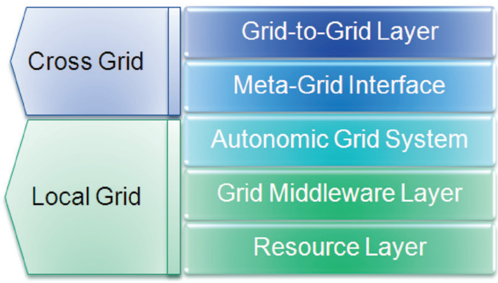
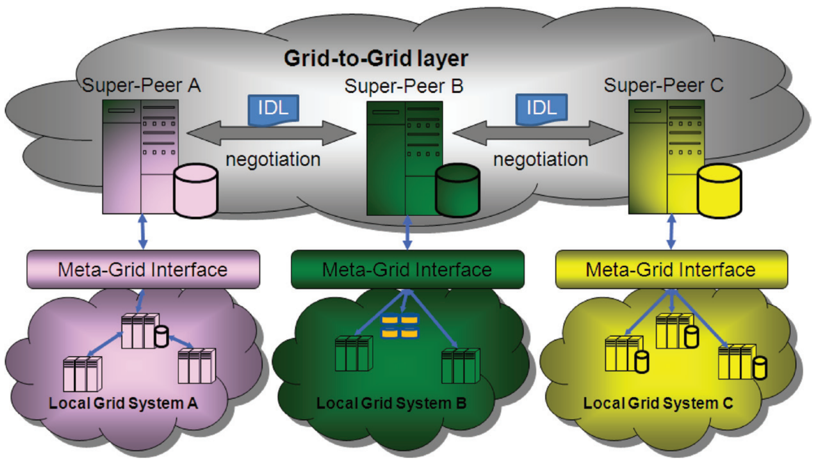


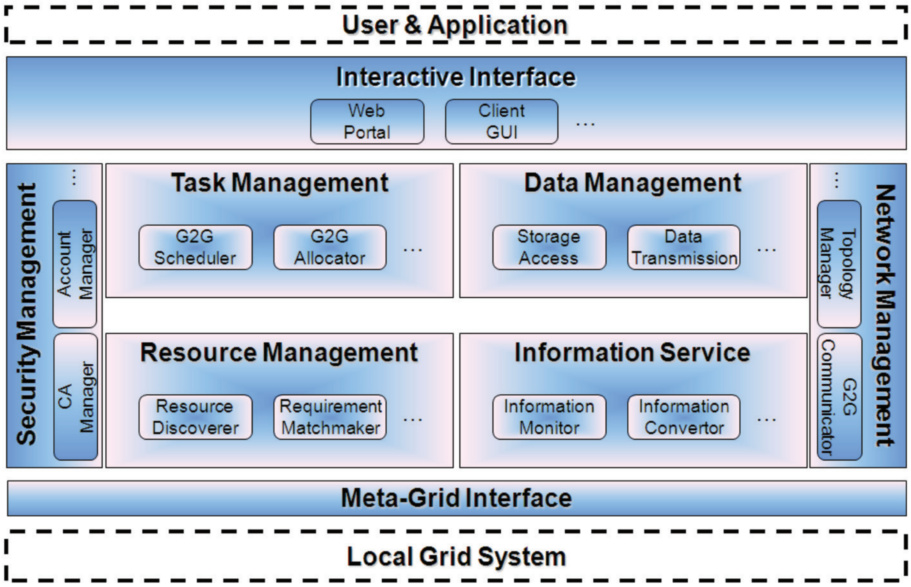
Figure 2. Architectural overview of the G2G system based on a super-peer network



these components in the G2G framework, we can apply grid applications on this framework.

We use a simple example to describe the cooperating procedure. When a user wants to login into the G2G system, the web portal will call the SecurityMgmt.loginManager() to handle the login process. This function will deliver the user's login information to the local security service through the Meta-Grid Interface. If the login process is successful, the user can edit jobs on the web portal or upload the data related to the job computation. When a user wants to submit a job for execution, the TaskMgmt.g2gScheduler() will receive the request from the web portal and call the ResourceMgmt.resourceDiscoverer() to locate the desired resources. The discovery process will firstly call the localResourceDiscovery() to check whether local resources are available by using the InformationService.informationMonitor(). If there are available resources in the local grid system, the TaskMgmt.g2gAllocator() will submit the job for local execution through the Meta-Grid Interface. In other cases, if no resources in the local grid system are available, the discovery process will call the remoteResourceDiscovery() to exploit remotely available resources by using the NetworkMgmt.g2gCommunicator(). If the remote login process is successful by using the SecurityMgmt.caManager() and the SecurityMgmt.accountManager(), the discovery process of the remote grid system is similar as above. The detail notions of developing a G2G prototype are shown in next subsection.

Figure 3. Framework overview of components in the Grid-to-Grid layer



G2G Prototype

This study uses JAVA to develop the proposed G2G framework. The developed components of the super-peer are deployed on top of each autonomic grid system to form the Grid-to-Grid federation environment. The super-peers communicate with each other by using a P2P overlay. In this subsection, we describe the implementation and the cooperation of all components in the G2G computing system.

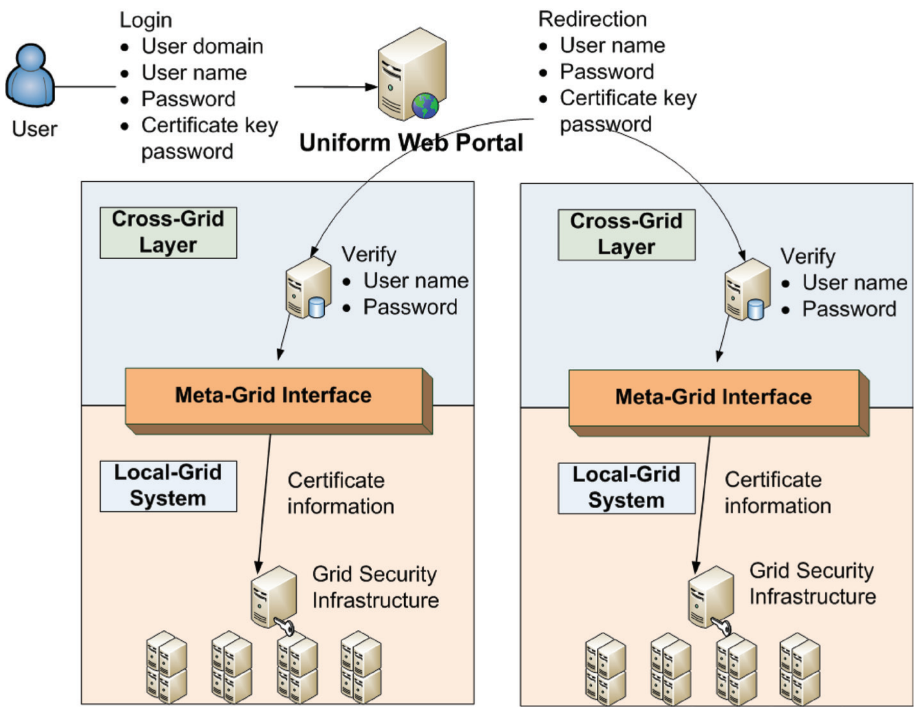
Portal and single sign-on. In general, a friendly interactive interface is important for users while using the grids. Therefore, this study develops a uniform web portal for users to easily enter a grid system and to utilize the authorized resources and services. There are two important functions for developing a uniform web portal: Single Sign-On (SSO), and workflow operation.

Single Sign-On (SSO) is adopted for users to access the grids with only-once login. Each user can utilize grid resources/services after the successful verification through the proxy server and the security management. This study proposes a uniform web portal on top of each autonomous grid system. A redirection mechanism is also developed in the uniform web portal as shown in Figure 4.

When a user logins the G2G system from this uniform web portal, the portal will determine which grid system the user should be entered according to the user's login information. The candidate grid system will verify whether the user's login information is valid or not. If the login is successful, the portal will deliver the user's login information to the local security service through the Meta-Grid interface. If the certificate of the user is also valid, the login process is successful and complete. Otherwise, it will be a failure. Since the login process is accomplished through the integration of original grid systems, if the local grid supports the SSO mechanism, the G2G layer can also sustain the SSO mechanism.

On the other hand, the workflow operation in the G2G system supports the task submission. The workflow structure in the G2G system is similar to the M-Task structure (Rauber & Runger, 2005). Each workflow is composed of

Figure 4. Redirection mechanism for users to login on the uniform web portal



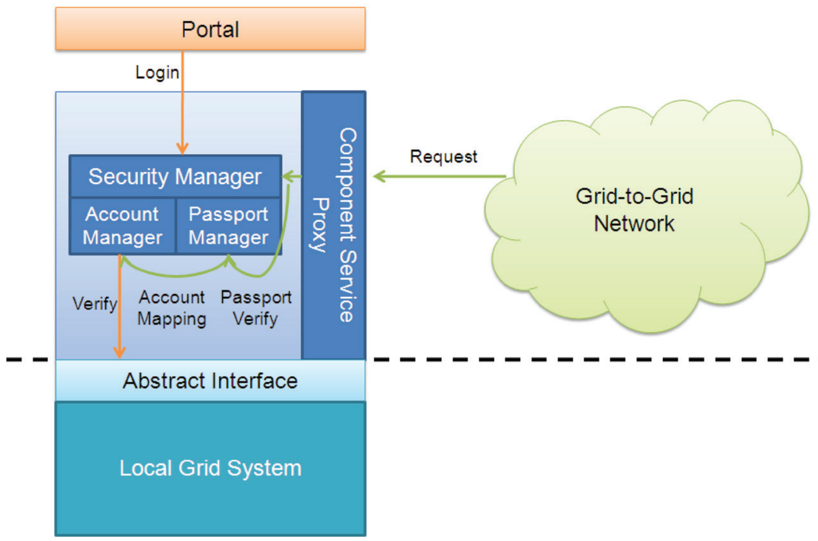
multiple stages and each stage is composed of multiple jobs. Jobs between distinct stages may be dependent; however, jobs in the same stage are all independent, that is, all jobs within the same stage can be scheduled and allocated for simultaneous execution.

Our study also develops a workflow editor in the uniform web portal. According to the resource status obtained from the information service component, users can edit the tasks on the portal and also specify desired resource requirements. We adopt an XML-based structure language to describe the task information and support the resource discovery by multiple attributes with the range query. After the task submission, the edited workflow can be transformed into the XML-based form and can be stored in the database for users to lookup, cancel, or refine their tasks at anytime.

Security service. Grid authentication and authorization are key services in grid security management. A previous study (Foster, Kesselman, Tsudik, & Tuecke, 1998) has defined the Grid Security Infrastructure (GSI) for the legal utilization of grid services. In the G2G system, the security management component deals with not only the certificate authorization locally, but also the admission request from remote grid systems. The secrecy and privacy mechanisms in the G2G system have to guarantee the original legal services in the local grid systems and accept the permission of utilizing local resources/services for other remote grid systems. The security management component includes two primary functions as shown in Figure 5: the passport manager for the authorized privilege and the account manager for the legal account management.

The passport manager takes care of the passport registration and the verification in the

Figure 5. Certificate authentication and authorization for our G2G system



G2G system. A passport represents the admission or verification of the request from remote grid systems. If one grid system tries to access resources in another grid system, it must get a visaed passport from the target grid system before accessing the resources. This study develops a distributed passport-interchange-mechanism in the G2G system. According to the maintenance of neighborhood relationship, each grid system can request a remote resource/service from its neighbors or neighbors' neighbors by forwarding the resource/service request along the overlay network. After discovering the available resource/service in remote grids, the requester will receive the visaed passports from the granted grid system; and then, the requester can submit tasks to the granted remote grids with legal permission.

On the other hand, the account manager is responsible for the management of legal accounts. In this study, the function of the account authentication is adopted for a "local account" to login the grids. A local account indicates an original user account in the local grid system. Once an account requests for a login from the portal, the portal asks the account manager to verify its identification. Another important aspect of the account manager is the account mapping mechanism. Account mapping is used to handle requests issued by foreign users from remote grid systems. Every grid system which wants to use the resources in other grids must register to the granted grid system before accessing those resources. The register process acquires a passport and gets a temporary account. Once the register process is completed, every request with the visaed passport from remote grid systems will be treated as a local user account through the account mapping mechanism.

Data service. Data management is responsible for data maintenance and high performance transmission in the data grid (Chervenak, Foster, Kesselman, Salisbury, & Tuecke, 2000). In this study, the data service of our G2G system supports specific APIs for the transparent accesses of existent data storage in each local grid system and for the data transmission among different autonomic grid systems. The abstract APIs is responsible to contact with a storage system in a local grid system or a general file system. Data accesses between the Cross-Grid layer and the Local-Grid layer adopt the well-defined data operations; otherwise, the data accesses from one grid to another grid adopt the G2G communication mechanism through super-peers.

When a communication between grids is necessary for data transmission, the super-peer takes care of the negotiation and communication with other super-peers in the G2G layer. We use the account manager to manage the foreign data files in this case. When the data files are accessed from remote grid systems, these data files can be stored in the local storage system and then be mapped to local owners. After data mapping and account handling, the foreign data file can be accessed by local users.

In this study, we focus on collaborating with the computing grid system and the data grid system, and omit developing efficient policy for data replication. Those issues about data coherence and parallel downloading between distinct grid systems remain as a future work.

Information service. The main responsibilities of the information service include the resource indexing and monitoring for capturing the resource status in a grid system. Traditional Grid Information Service (Czajkowski, Fitzgerald, Foster, & Kesselman, 2001; Fitzgerald et al., 1997) generally adopts the centralized or hierarchical architecture. Such architectures for the information service are hard to directly apply to the G2G system because of the single point of failure problem. To alleviate the failure problem, this study develops an information service for crossing the inter-grid systems on top of the existent information monitoring system.

Our information service consists of two mechanisms: the *information monitor* and the *information convertor*. Each grid system in the Local-Grid layer is responsible for monitoring the local information. The super-peer queries the local information by using the proposed information monitor to negotiate with the local grid system. On the other hand, we propose an Information Description Language (IDL) to negotiate the information with the local-grid system and exchange the information between cross-grid systems. The IDL is a XML-based structure, which is shown as Figure 6 and Figure 7, for describing the grid information such as the workflow submission, the job requirement, the resource information, and so on. We also develop the information convertor to transform diverse XML-based information into our IDL format, and vice versa. By using the information convertor, each grid system of our G2G system can extract the information from the local-grid or remote-grid description.

By the IDL, information can be negotiated from Cross-Grid layer to Local-Grid layer and messages can be exchanged between distinct grid systems in the G2G layer. A workflow or a complicate job requirement with multi-attribute range query can be supported in the distributed resource discovery.

Network management. This study proposes a Grid-to-Grid overlay based on the super-peer network. In the G2G system, the super-peer in each autonomic grid system takes responsible for the negotiation and communication with other super-peers over the G2G network. The decentralized overlay network is adopted to construct the neighborhood relationship and to forward a request between super-peers.

There are two main management functions in the G2G network management: the *topology manager* and the *G2G communicator*. The topology manager maintains the overlay network for the neighborhood relationship or routing information between super-peers. As one super-peer tries to join the G2G network, the topology manger guides the new super-peer to join this G2G system, and then the super-peer will forward a request to build its routing table after successfully joining into the G2G network.

Figure 8 depicts the procedure for a new grid system to join the G2G system. Every

Figure 6. Example for the task submission and the job requirement in the IDL structure

```
<TASK>
    <WORKFLOW NAME="Workflow Name">
        <STAGE NAME="Stage Name" ORDER="1">
            <JOB NAME="Job Name">
                <PROGRAM NAME="Matrix Multiplication" ARG="1024" NP="4" TYPE="MPI" />
                <INPUT_FILES>
                    <INPUT_FILE DATA_ID="Data ID for the Program" />
                </INPUT_FILES>
                <REQUIREMENTS>
                <!-- One of the following requirement tag should be matched for executing the job. -->
                    <REQUIREMENT>
                    <!-- All the required information should be matched with the following conditions. -->
                        <INFORMATION_TYPE NAME="Information Name" VALUE="Information Value" VALUETYPE="Value Type"
                                          UNIT="Value Unit" OPERATOR="AND, NOT" RELATION="EQUAL, GREAT, LESS" />
                    </REQUIREMENT>
                    <REQUIREMENT>
                        <!--
                          -- For example, the job requires the desired resource with CPU is greater than 2GHz,
                          -- and the machine type is not x86_64.
                        -->
                        <INFORMATION_TYPE NAME="2" OPERATOR="1" RELATION="2" UNIT="1" VALUE="2000" VALUETYPE="6" />
                        <INFORMATION_TYPE NAME="21" OPERATOR="2" RELATION="1" UNIT="0" VALUE="x86_64" VALUETYPE="1" />
                    </REQUIREMENT>
```

Figure 7. Example for resource information in the IDL structure

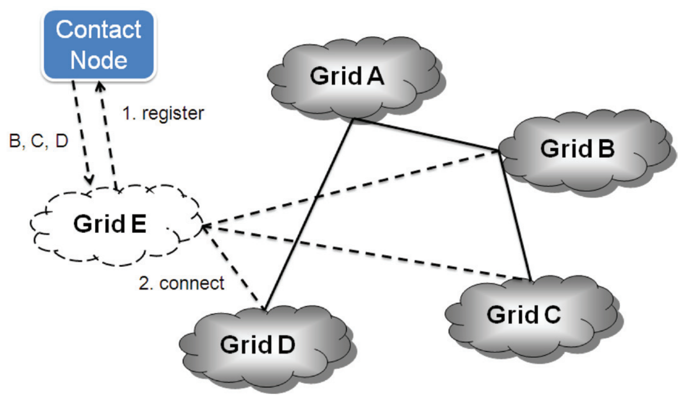
```
<RESOURCE>
    <NODE ADDRESS="Super-peer Address">
        <CLUSTER ADDRESS="Cluster Address">
            <HOST ADDRESS="Host Address 1" REPORTED="Last-record TIMESTAMP">
                <RESOURCE_STATE NAME="Attribute NAME" VALUE="Attribute VALUE" TYPE="Value TYPE" UNIT="Value UNIT" />
            </HOST>
            <HOST ADDRESS="Host Address 2" REPORTED="Last-record TIMESTAMP">
                <!-- For example, a resource with the CPU is 3200 MHz and the machine type is x86. -->
                <RESOURCE_STATE NAME="2" VALUE="3200" TYPE="6" UNIT="1" />
                <RESOURCE_STATE NAME="21" VALUE="x86" TYPE="1" UNIT="0" />
            </HOST>
```

autonomic grid system acts as a super-peer to be a member of the super-peer network. When a grid system wants to join the G2G system, it registers to a contact node and gets a list of grid systems selected by the contact node in random. After obtaining the random list, the new grid system measures the network latency or the bandwidth with all the candidate members. After the measurement process, the new grid system selects members with better performance to register as its neighbors. This join procedure for a new member is similar to the previous work (Mastroianni et al., 2005). We implement this mechanism for a new super-peer to join the G2G network, and enhance the construction of an unstructured overlay network.

We also present the G2G communicator to take care of the network communication and the message negotiation. In order to communicate with different autonomic systems, we not only apply the IDL to describe the exchanged information but also design an application-level request format for message transmission. Every communication is accomplished by using the socket connection. The communication in-between two grid systems can be divided into sender- and receiver-modules. For the sender module, all the requests will be transformed into a predefined request format, and then the requests are sent to remote super-peers in serial. For the receiver module, the remote super-peer de-serializes all the received requests and forwards to the corresponding components.

Task management with resource discovery. The task management is in charge of the task submission through the interactive interface. A task consists of jobs executed in sequential or in parallel. In the G2G system, tasks are not only submitted from local users, but probably are requested from remote grid systems. Depending on available grid resources, the G2G scheduler and G2G allocator need to consider the job

Figure 8. Procedure for a new grid system to join the G2G network



execution among the intra-grid submission or the inter-grid submission.

In the task management module, we adopt a workflow structure to organize jobs in a predefined order for execution. The workflow structure is constructed by stages. The jobs in one stage must wait for execution until all jobs in previous stage are completed because of the stage-by-stage approach to avoid destroying the job dependence in different stages. We also develop the workflow manager and the job manager to handle requested tasks. After a task is submitted to the workflow manager, the manager schedules the order of jobs and decides where to execute these jobs. The decision of migrating the executable jobs to a local grid or a remote grid depends on the system performance or the resource requirement.

On the other hand, each job has its resource requirement for execution. This study also applies a resource discovery mechanism (Mastroianni et al., 2005) to explore the distributed resources status over the Grid-to-Grid overlay network, and supports a matchmaking policy to provide candidate resources satisfied the specified requirements. After a task is submitted to the waiting queue for execution, the G2G scheduler picks one of queuing jobs according to the First-Come-First-Served (FCFS) policy, and then checks whether local resources are sufficient or under loading at first. The decision of where to execute a job depends on not only whether local grid system is over loading, but also whether local resources are satisfied with requirements through information service. If the local grid system is not busy and there are sufficient available resources, the job will be submitted to the local grid system to be executed. Otherwise, the job manager will ask the distributed resource discovery module to search available resources over the overlay network. If there are sufficient resources in other remote grid systems through the super-peer network, the job will be migrated to the remote grid system for execution. Otherwise, this job will be queued in the waiting queue for available resources.

EXPERIMENTAL EVALUATION

In this section, a simple experiment is conducted to evaluate the system performance of the proposed framework. At first, we describe the experimental environment in our evaluation. Then, we show the experimental results in terms of the average turnaround time and the extra overhead occurring from a separate layer by adopting the G2G system.

Experimental Environment

In the experimental environment, two autonomic grid systems based on the framework of

Taiwan UniGrid (Shih et al., 2008) are applied. In order to limit the impact of the heterogeneous environment, we utilize the same cluster in each autonomic grid system for execution. One autonomic grid system contains a cluster with 8 higher computational power CPUs (i.e., the grid with higher computational power.) The other autonomic grid system contains a cluster with 32 lower computing power CPUs (i.e., the grid with rich resources). In each local grid system, the SRB (Baru, Moore, Rajasekar, & Wan, 1998) is adopted to be the local data storage system; and the ganglia information monitoring system (Massie, Chun, & Culler, 2004) is utilized to monitor the local information. We deploy the proposed super-peer network on top of each autonomic grid system to form a Grid-to-Grid federation environment.

We use a matrix multiplication program as the benchmark. Each job in the experiment is a parallel program written by MPI with C. The matrix size is 1024x1024. The number of required processors for each job is set to be 4. The ratio of communication to computation of the test program is about 1 to 100. Each job is submitted per 10 seconds; and the total number of submitted jobs varies from 5 to 20 in two cases to evaluate the system performance. In case 1, the same jobs are submitted to each autonomic grid system for four rounds. The number of jobs in each round is 5, 10, 15, or 20, respectively. For this case, all the jobs are only executed in the local grid system. In case 2, the same jobs of each round are submitted to each autonomic grid system with the G2G federation environment. For this case, each job will be executed in the local grid or the remote grid according to the decision made by the above-mentioned G2G scheduler and G2G allocator.

We estimate the average turnaround time of each job for completing the computation and also measure the average overhead of each job with regard to the time consumed from the Cross-Grid layer to the Local-Grid layer. The job turnaround time is defined as the time period from the time when a job is submitted to the waiting queue for processing in the Cross-Grid layer, to the time when the submitted job is completed in the Local-Grid layer. The overhead of migrating a job is defined as the time period from the time when a job is submitted to the Cross-Grid layer, to the time when the submitted job is decided to which local-grid system for execution. By definition, the job turnaround time includes the execution time and the migrating overhead. If the cost of overhead takes too much time in the Cross-Grid layer, the overall turnaround time for executing a job will be increasing. The G2G system is expected to reduce the average turnaround time of each job with little overhead when there is over-loading or there are no available resources in the local grid system.

Experimental Results

Figure 9 depicts the experimental results in terms of the average turnaround time. The results show that each grid system requires longer turnaround time to complete a job when there are more jobs submitted for execution without the help from remote grid systems. As a result, the grid with higher computational power takes the longer time to complete a job once there is getting more jobs for execution. This is because a job is queued for a long time if local resources are often busy in the grid system with fewer resources. The longer time a job is queued, the more time the turnaround time will be consumed. However, the system performance can be improved by using the proposed meta-grid framework. When there are no available resources in a local grid system, the job will be migrated to a remote grid system with available resources for execution. On the other hand, Figure 10 and Figure 11 depict the ratio of average overhead compared with the average turnaround time. The experimental results show that the average overhead in all cases is light. The reason is that the average overhead will be reduced if a job can be allocated to available resources for execution as soon as possible. Hence, the average turnaround time of completing a job will be reduced as well.

In summary, a grid system with G2G federation can not only reduce the cost of overhead

Figure 9. Average turnaround time for completing jobs

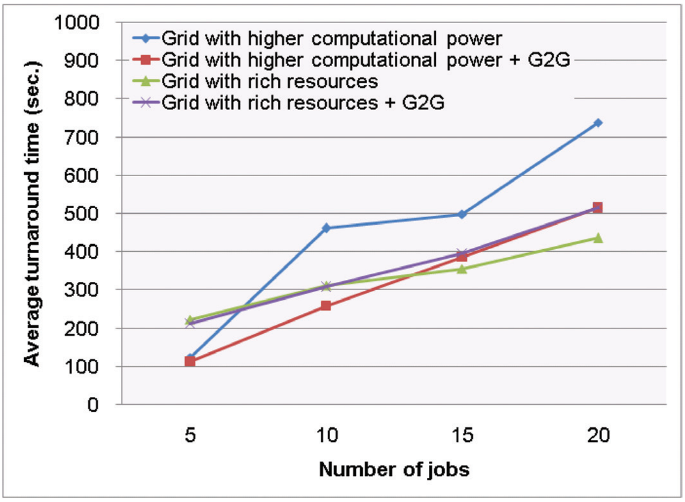
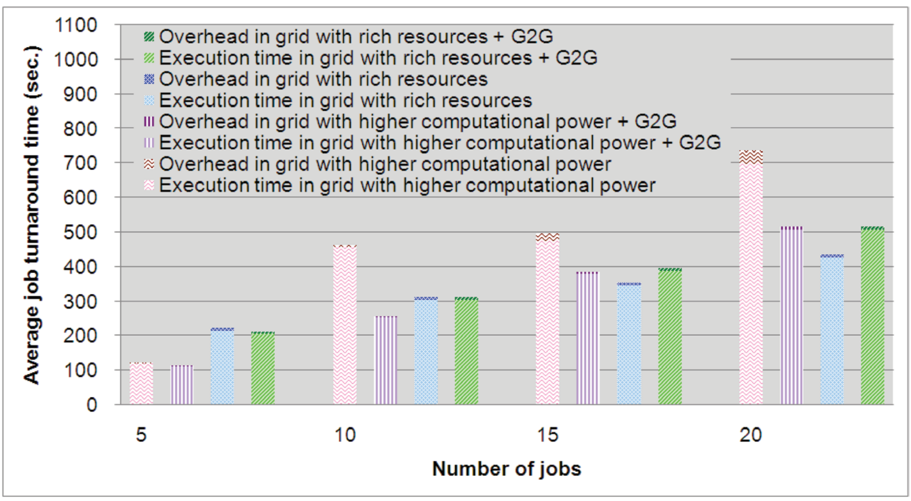


Figure 10. Distribution of average turnaround time for completing jobs

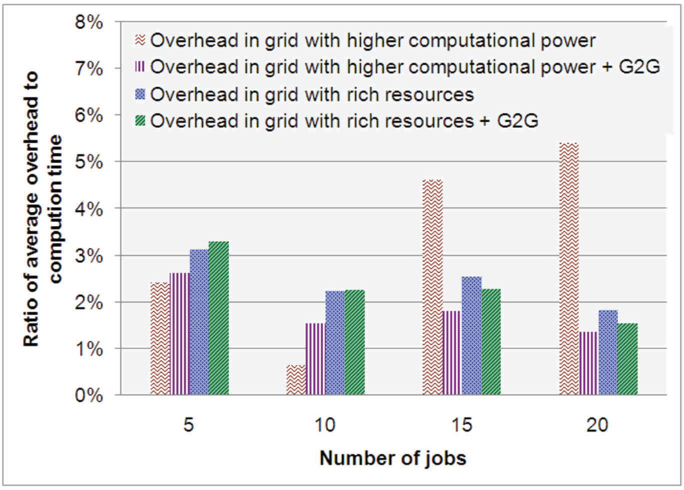


when there are not available resources for execution, but also reduce the average turnaround time with little overhead. The reason is that some jobs submitting to the grid with higher computational power can be migrated to the grid with rich resources for execution. Hence, the grid with high computational power can achieve lower average turnaround time and the grid with rich resources can achieve higher resource utilization. From the experimental results, the average turnaround time can be reduced up to about 45% and the average overhead in all cases is less than 6%.

CONCLUSION AND FUTURE WORKS

Integrating the grid computing with P2P technique can improve the scalability of the large-

Figure 11. Ratio of average overhead to computation time for completing jobs



scale grid system. This study proposes a meta-grid framework, named the G2G framework, for the Grid-to-Grid federation of autonomic grid systems without modifying the original mechanisms and policies. Based on the super-peer network, we adopt a separate layer built on top of existing grid systems to develop the Grid-to-Grid collaborative computing environment. Each super-peer represents a grid system. A super-peer in the G2G system is responsible for coordinating an internally autonomic grid system and for communicating with other super-peers. The overlay network among super-peers is constructed via the unstructured approach.

A grid system is deployed with the capacity of the super-peer for coordinating the G2G system. With well-defined APIs, the G2G system resembles an abstract layer that is separate from the existing grid systems. In our G2G system, an existing grid system can be upgraded its G2G capability without upsetting original mechanisms. We not only take care of the support of computation services and data services, but also consider a solution for the grid security across different grid systems. To evaluate the performance of the G2G system, we implemented a preliminary system to show that the proposed system not only is workable but also improves the system performance.

In the future work, we will refine the IDL format to follow the standard description language, and also study on the efficiency of grid security across diverse grid systems. We also intend to integrate with more autonomic grid systems and enhance the G2G framework to the Service-Oriented Architecture (SOA) in order to develop a service-oriented G2G computing system.

REFERENCES

Baru, C., Moore, R., Rajasekar, A., & Wan, M. (1998). *The SDSC Storage Resource Broker*. Paper presented at the Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research.

Cao, J., Liu, F. B., & Xu, C.-Z. (2007). P2PGrid: Integrating P2P Networks into the Grid Environment. *Concurrency and Computation*, *19*(7), 1023–1046. doi:10.1002/cpe.1096

Castro, M., Druschel, P., Kermarrec, A.-M., & Rowstron, A. I. T. (2002). Scribe: A Large-Scale and Decentralized Application-Level Multicast Infrastructure. *IEEE Journal on Selected Areas in Communications*, *20*(8), 1489–1499. doi:10.1109/JSAC.2002.803069

Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., & Tuecke, S. (2000). The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *Journal of Network and Computer Applications*, *23*(3), 187–200. doi:10.1006/jnca.2000.0110

Czajkowski, K., Fitzgerald, S., Foster, I., & Kesselman, C. (2001). *Grid Information Services for Distributed Resource Sharing*. Paper presented at the Proceedings of the10th IEEE International Symposium on High Performance Distributed Computing, New York.

Fitzgerald, S., Foster, I., Kesselman, C., von Laszewski, G., Smith, W., & Tuecke, S. (1997). *A Directory Service for Conturing High-Performance Distributed Computations*. Paper presented at the Proceedings of The Sixth IEEE International Symposium on High Performance Distributed Computing.

Foster, I., & Iamnitchi, A. (2003). On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. In *Peer-to-Peer Systems II* (pp. 118-128).

Foster, I., Kesselman, C., Nick, J. M., & Tuecke, S. (2002). *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. Retrieved from http://www.globus.org/alliance/publications/papers/ogsa.pdf

Foster, I., Kesselman, C., Tsudik, G., & Tuecke, S. (1998). *A Security Architecture for Computational Grids*. Paper presented at the Proceedings of the 5th ACM conference on Computer and communications security.

Foster, I., Kesselman, C., & Tuecke, S. (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, *15*(3), 200–222. doi:10.1177/109434200101500302

Fox, G., Pallickara, S., & Rao, X. (2005). Towards Enabling Peer-to-Peer Grids. *Concurrency and Computation*, *17*(7-8), 1109–1131. doi:10.1002/cpe.863

Massie, M. L., Chun, B. N., & Culler, D. E. (2004). The Ganglia Distributed Monitoring System: Design, Implementation, and Experience. *Parallel Computing*, *30*(7), 817–840. doi:10.1016/j.parco.2004.04.001

Mastroianni, C., Talia, D., & Verta, O. (2005). A Super-Peer Model for Resource Discovery Services in Large-Scale Grids. *Future Generation Computer Systems*, *21*(8), 1235–1248. doi:10.1016/j.future.2005.06.001

Mastroianni, C., Talia, D., & Verta, O. (2007). *Evaluating Resource Discovery Protocols for Hierarchical and Super-Peer Grid Information Systems*. Paper presented at the Proceedings of the 15th Euromicro International Conference on Parallel, Distributed and Network-Based Processing.

Pairot, C., Garcia, P., & Skarmeta, A. F. G. (2004). *DERMI: A Decentralized Peer-to-Peer Event-Based Object Middleware*. Paper presented at the Proceedings of the 24th International Conference on Distributed Computing Systems.

Pallickara, S., & Fox, G. (2003). NaradaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids. In . *Proceedings of Middleware*, *2003*, (pp.998–999).

Ranjan, R., Harwood, A., & Buyya, R. (2008). Peer-to-Peer-Based Resource Discovery in Global Grids: A Tutorial. *IEEE Communications Surveys & Tutorials*, *10*(2), 6–33. doi:10.1109/COMST.2008.4564477

Rauber, T., & Runger, G. (2005). *M-Task-Programming for Heterogeneous Systems and Grid Environments*. Paper presented at the 19th IEEE International Parallel and Distributed Processing Symposium.

Rowstron, A., & Druschel, P. (2001). Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Proceedings of Middleware*, *2001*, (pp. 329–350).

Shih, P.-C., Chen, H.-M., Chung, Y.-C., Wang, C.-M., Chang, R.-S., Hsu, C.-H., et al. (2008). *Middleware of Taiwan UniGrid*. Paper presented at the 2008 ACM symposium on Applied computing.

Talia, D., & Trunfio, P. (2003). Toward a Synergy Between P2P and Grids. *IEEE Internet Computing*, *7*(4), 96–95. doi:10.1109/MIC.2003.1215667

Trunfio, P., Talia, D., Papadakis, H., Fragopoulou, P., Mordacchini, M., & Pennanen, M. (2007). Peer-to-Peer Resource Discovery in Grids: Models and Systems. *Future Generation Computer Systems*, *23*(7), 864–878. doi:10.1016/j.future.2006.12.003

Yang, B., & Garcia-Molina, H. (2003). *Designing a Super-Peer Network*. Paper presented at the 19th International Conference on Data Engineering.

Wu-Chun Chung received his B.S. degree in computer and information science from Aletheia University in 2004 and his M.S. degree in computer science and information engineering from National Dong Hwa University in 2006. He is currently pursuing his Ph.D. degree in the department of computer science at National Tsing Hua University. His research interests include grid computing, distributed systems, peer-to-peer computing, and wireless networks.

Chin-Jung Hsu received his B.S. degree in computer science and information engineering from Fu-Jen Catholic University in 2007. He is currently pursuing his M.S. degree in the department of institute of information systems and applications at National Tsing Hua University. His research interests include grid computing, distributed systems, peer-to-peer computing, and cloud computing.

Yi-Hsiang Lin received his B.S. degree in information management from National University of Kaohsiung in 2007. He is currently pursuing his M.S. degree in the department of institute of information systems and applications at National Tsing Hua University. His research interests include grid computing, distributed systems, and peer-to-peer computing.

Kuan-Chou Lai received his MS degree in computer science and information engineering from the National Cheng Kung University in 1991, and the PhD degree in computer science and information engineering from the National Chiao Tung University in 1996. Currently, he is an associate professor in the Department of Computer and Information Science at the National Taichung University. His research interests include parallel processing, heterogeneous computing, system architecture, P2P computing, grid computing, and cloud computing. He is a member of the IEEE and the IEEE Computer Society.

Yeh-Ching Chung received a B.S. degree in Information Engineering from Chung Yuan Christian University in 1983, and the M.S. and Ph.D. degrees in Computer and Information Science from Syracuse University in 1988 and 1992, respectively. He joined the Department of Information Engineering at Feng Chia University as an associate professor in 1992 and became a full professor in 1999. From 1998 to 2001, he was the chairman of the department. In 2002, he joined the Department of Computer Science at National Tsing Hua University as a full professor. His research interests include parallel and distributed processing, cluster systems, grid computing, multi-core tool chain design, and multi-core embedded systems. He is a member of the IEEE computer society and ACM.

Personal Storage Grid Architecture:
Consuming Cloud Data Space Resources

Mian-Guan Lim, Cranfield University, UK

Sining Wu, Cranfield University, UK

Tomasz Simon, Cranfield University, UK

Md Rashid, Cranfield University, UK

Na Helian, Hertfordshire University, UK

ABSTRACT

On-demand cloud applications like online email accounts and online virtual disk space are becoming widely available in various forms. In cloud applications, one can see the importance of underlying resources, such as disk space, that is available to the end-user but not easily accessible. In the authors' study, a modern file system developed in linux is proposed, which enables consuming of cloud applications and making the underlying disk space resource available to the end-user. This system is developed as a web service to support cross operation system support. A free online mail account was used to demonstrate this solution, and an IMAP protocol to communicate with remote data spaces was used so that this method can mount onto any email system that supports IMAP. The authors' definition of infinite storage as the user is able to mount file systems as a single logical drive.

Keywords: *Cloud Applications, Cloud Computing, Data Flow Architecture, Data Management, Grid Computing*

INTRODUCTION

Centralizing storage is one primary aspect in grid computing (Foster, 2002). GOS (Wang et al., 2007) emphasizes the importance of not only centralizing data but as well as distributing data. Online storage space such as online email accounts, social network storage and free online drive space are becoming widely available. We can see that such a trend will continue to expand based on Moore's law. These data storage space resources are widely available however they are distributed. A single user may have access to multiple storage resources such as email storage and remote disk drives however these resources are underused due to the diversity of interfaces.

Our motivation is to demonstrate the application of a grid based storage architecture,

DOI: 10.4018/jghpc.2010070102

which emphasizes on centralizing of personal resources and making it available to as a single resource. Grid-storage architecture provides a single transparent interface that binds a range of distributed services from company wide network storage to cloud enabled services such as online email.

We propose a personal storage grid architecture, which is aimed at connecting and managing data space resources from various domains while maintaining transparency and security. We emphasize the importance of “personal” in our storage architecture. In a user’s perspective, various online data space resources are available but not readily accessible for personal usage (Figure 1).

We refer to online data spaces resources as free drive spaces, webhosting spaces, social networking services and online free mail. A personal storage grid architecture, which consumes various services and presents them as data space to the user, allows the user to access an infinite amount of storage. Puffs (Kantee, 2007)(Kantee & Crooks, 2007) also presented bringing file-system to the userspace. The term infinite storage refers to where the user can mount on to n amount of services and merge them as a single logical mount, allowing the end-user to transparently access storage space conforming to a grid-like architecture.

Our demonstration is to use data space resources offered by online services to create a local storage. This architecture allows the user to access online data space through as a local mounted drive resource and be able to perform tasks such as directory listing, copying files between file systems, deleting, creating, and renaming files and directories. CurlFtpFS (Araujo, 2007) and FuseFTP (Thiesen, 2005) both demonstrate that file operations are possible on a FUSE framework.

There are 2 similar projects we encourage for study: (1) Luo’s IMAP storage (Luo, 2007) based on Fuse-J is close to what we are pro-

Figure 1. Webservice that consumes cloud services

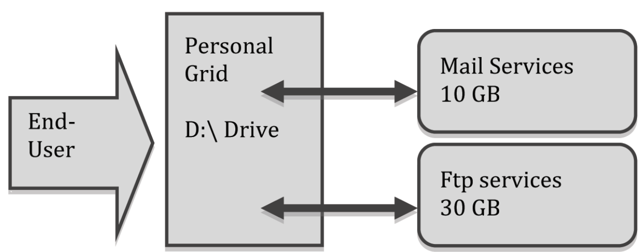
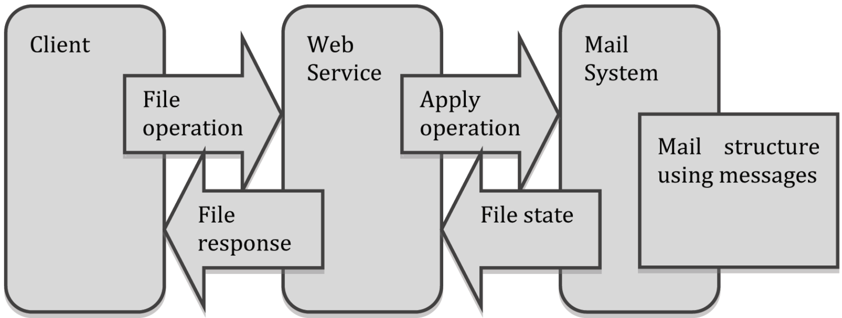


Figure 2. Web architecture



posing: functioning as a real storage solution. (2) GmailFS (Jones, 2007) implemented using libgmail currently supporting only Gmail space.

A PERSONAL STORAGE GRID ARCHITECTURE (PSAG)

We constructed a web service that consumes end-user's file and directory handling operations, and applies it to a designated online resource. Our resource is defined as an online email account and file or directory operations are passed on to our web service, which is applied onto the mail service via IMAP. The file and directory operations our file system needs to perform are: Get Attributes, Open File, Read File Data, Create Empty File, Write File, Remove File, Rename File, Read Directory, Remove Directory, Create Directory (Almeida, 1999; Nguyen, 2004; Alexandrov, Ibel, Schauser, & Scheiman, 1997) (Figure 2).

In our architecture, we considered XDrive (openXdrive, 2007), which uses JavaScript Object Notation (JSON) to allow access to its online space and Openomy using RESTful (Representational State Transfer) API (Pautasso, Zimmermann, & Leymann, 2008) to access the storage and Google Mail. To demonstrate that free email services can be used we tested Google Mail with our architecture. For the mails storage to be exposed as a logical drive locally, we used FUSE framework on a typical linux ubuntu desktop edition. FUSE allows the creation of file systems in the user space.

We used XML that complies with SOAP standard for passing messages between the linux file system (Card, Ts'o, & Tweedie, 1994) and our webservice. The interface between the email accounts can be built using either POP or IMAP. We selected IMAP, as it is able to maintain connection using online mode. POP access does not have the option of online mode (Figure 3).

In our experiment, we used an online mail's free storage. The local storage solution can be applied on any mail account (Dornfest, Bausch, & Calishain, 2006) that provides an IMAP protocol. We also used a linux virtual file system (VFS), IMAP email communication protocol, and Fuse for our implementation. There are 4 main components to our solution: (1) FUSE architecture (Figure 4), which is a kernel module, is registered as a file system and DEV char on the host operating system (OS). The kernel module connects the calls from the user space (Gera, 2006) via the OS, which is linux ubuntu. When the VSF node gets a call from the user space, it will store this request in "Queue & Sync", waking up the DEV char. DEV char passes the call to the FUSE library, which calls our implement function.

(2) gSOAP (Engelen, 2009) is used to automatically map XML data types into C data types for the client side application of our system. (3) GlassFish and Java API for XML Web Services (Java EE platform, 2008) are used to implement the system as a web service. Glassfish is an open source application server, which is a derivative of Apache Tomcat Server. (4) JavaMail API is used for development of Mail User Agent (MUA) for reading and sending email messages. It facilitates work with SMTP, POP and IMAP.

IMPLEMENTING A FILE SYSTEM ON PSGA

A file system is a method of storing and managing data and metadata (Bovet & Cesati, 2005). File systems can use data from various sources and currently hard disk source is the most common. There are other common file systems such as Linux ext (Card et al., 1994; Jones, 2007), Microsoft Windows FAT32, NTFS(Driscoll, Beavers, & Tokuda, 2008). The components of a typical file system include files and directories.

Files are created and accessed through filenames, which are commonly restricted to 255 characters. Regular files can be either in ASCII or Binary.

Directories are logical structures of file groups enabling organization of the data in

Figure 3. Communication between components for personal grid architecture

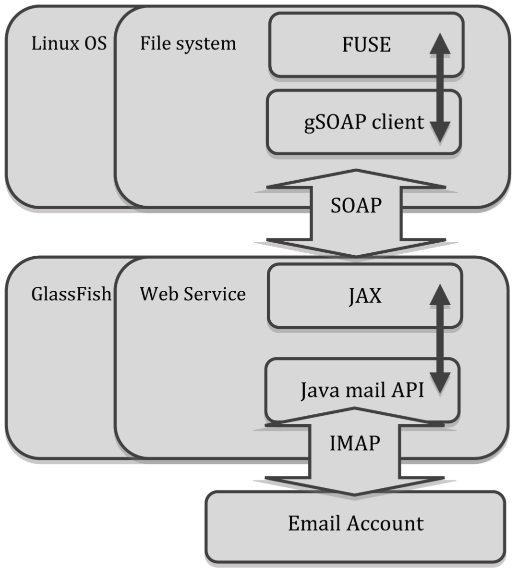
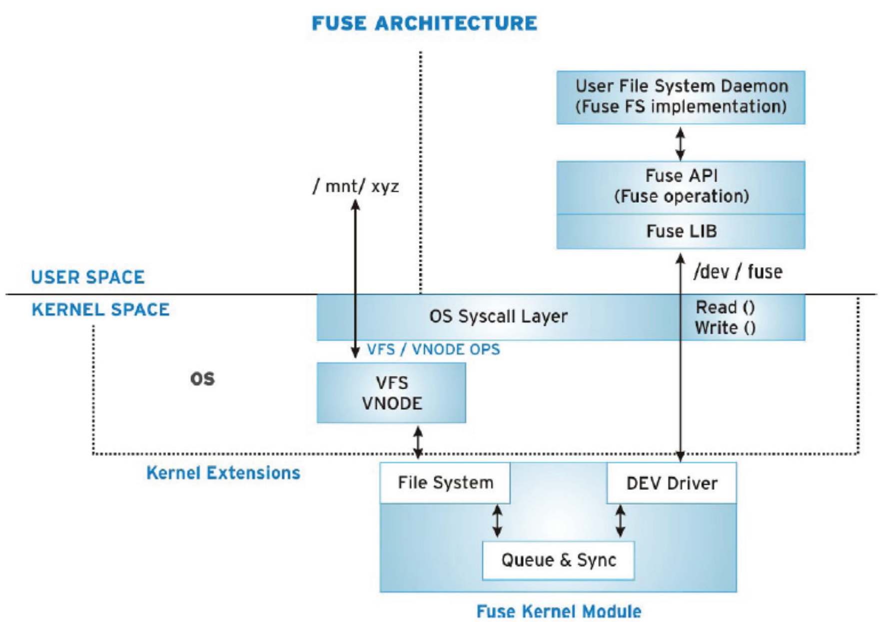


Figure 4. FUSE architecture



a file system. There are 3 forms of directory systems: Single leveled directory, 2-leveled directory and Hierarchical directory. Single leveled directory is having 1 main directory storing files. A 2-leveled directory is where a master directory contains users directory. Finally a hierarchical directory, commonly used by modern file systems, allows users to create multiple sub-directories.

There are 3 main methods of implementation: (1) Contiguous allocation, (2) Linked list allocation, and (3) indexed allocation.

Contiguous allocation implementation is based on file blocks stored continuously one after another. This solution allows fast access, requiring the start data block and the size of the file block. As new data is always written to the last block, it causes data fragmentation and unused free space when data is removed.

Linked list allocation uses a linked list to point to disk blocks forming a file. This solution frees files from physical data orientation since a file can be composed of different data blocks. This method is an improvement over contiguous allocation, as space is not wasted. Linked list suffers from slow random access and space wasted for pointers in each block. MS (FAT) improved on this method by using a table in memory storing all the pointers. The fall back of this method is that it is not scalable; the size of the disk grows with the size of the table in memory.

Indexed allocation uses one fixed block to store a fixed size of array of pointers. The main advantage is this allows fast random data access. The only problem is that space is wasted for unneeded pointers. An improvement of this method is the use of some pointers pointing to blocks that are used as an index rather than a data block. Such method is implemented in ext2 linux file system and commonly known as inode.

In our solution we opted for indexed allocation. We used 3 messages in the online email system for presenting one file in the user space: I-node structure, Dentry and file. These 3 pieces of information are supported by VFS used by UNIX like file structures: (1) I-node messages store the metadata about the file such as i-node number, file type, file permissions and file size. (2) Dentry messages connect the i-node message with the name of the file. A typical Dentry in our system stores the file name and i-node number of the file. (3) File messages stores data in the attachment and i-node number specifying a particular file.

We use the subject of the message as storage to improve performance of the overall system. Each information piece is split by an identifier, asterisk "*".

First file of each message describes type of the information stored (i-node – "i", path – "p", data – "d"). Next filed for each message is a unique file system identification number that with conjunction with previous field allows for easy message identification. GmailFS (Jones, 2007) uses a similar solution with the file system identification number, which ensures that the file system uses only self-created messages.

The last fields for each message are i-node number of the file. This identification number enables linking of all the messages concerning one file. Next fields are different for each message. I-node message beside regular metadata stores total number of data messages what is necessary to read the whole file. Since messages can be split into chunks, each data message stores size and number of its part. Directories are represented in the same way like files, but the only difference is that data message will not be used.

We chose to use 3 messages as one e-mail message with an attachment storing all the information about a file has low read and write the performance. For instance change of the file name or arbitrary change of the file attributes will cause that whole message with attachment has to be downloaded and send again with new name or modified attributes. In the case of this solution just i-node or path message is modi-

Table 1.

I-node	Subject: i*fs_id*i-node number*number of data parts*metadata
Path	Subject: p*fs_id*i-node number*file name*
Data n	Subject: d*fs_id*i-node number*size of the part*part number Attachment: file

fied whereas message with the attachment is untouched.

Getting Attributes: Client Side

In a linux operating system, *getattr* system call returns file attributes of the current file system. To get attributes of a file or a directory, simfs_getattr function is called with two arguments: Absolute path to the file, Structure stat storing file attributes – output.

This function on a successful call returns the stat structure filled in with the metadata concerning a file passed in the path as a first argument. If error occurs, negative value will be returned.

Getting Attributes: Server Side

Firstly from the path passed as an argument extracted are: path to the parent directory and name of the file (Line 1). Parent directory is necessary to identify folder that stores the file on the email server. Before folder will be opened (Line 3) it is required to obtain hash code for the parent directory path (Line 2), because all the folders are represented by its hashed values.

The getInode (Line 4) function will find message containing name of the file passed as argument. It will be only one message, because it is not allowed to store the same file names in one directory. An i-node number will be extracted from the found message and passed to the getFileAttr function (Line 5). This function will prepare following search term: "i*id_fs*inodeNumber*", which will allow to find unique i-node message for the directory and extract from it attributes of the file.

Open File: Client Side

The open file operation determines if the user have sufficient privileges to open the file.

Function "open" defined needs 2 input arguments: Absolute path to file, structure of the file, which the client needs.

Function "open" passes 2 arguments to the remote procedure: Absolution path to file, Mode of file. The mode of file is defined as: read =0, write = 1, read and write = 2. The remote procedure returns 0 to signify file can be opened.

```
static int simfs_open(const char
*path, struct fuse_file_info *fi)
```

Open File: Server Side

"Getattr" method on the server side is invoked to get the mode of the file. Mode stored in the message consists of both permission and type, however in open file operation, we only need the permission.

Function "checkFileMode" defined needs 2 arguments: File flags, Mode of file.

At the beginning mode of the file is translated to the octal mode (Line 1), this operation will facilitate next steps. Assume that we have a file with decimal mode 33188 (in octal equals 100644) or a directory 16877 (in octal – 40755). Three last digits inform about file permissions and the rest describe type of the file: 100 – regular file, 40 – directory.

Due to the difference, it is necessary to retrieve owner file permission in two different ways (Line 3 and 5). Error will be returned if the user does not have permissions to open the file in the mode described by the flag (Line 6).

Table 2.

Line 0	int[] getattr(path)
Line 1	solvePath(path): parentDir, fileName
Line 2	hash(parentDir): hashedDir
Line 3	openFolder(hashedDir)
Line 4	getInode(filename): errorNb, inodeNb
Line 5	getFileAttr(inodeNb): errorNb, stat

Read Data:Client Side

The file read operation performs data transmission of the file contents to the client.

The function "read" passes 5 arguments: Path to file, Buffer, Size of the data to read and size of the buffer, Offset from the beginning of the file, Information about read operation.

```
static int simfs_read(const char
*path, char *buf, size_t size, off_t
offset, struct fuse_file_info *fi)
```

In this function, we have to check if 2 conditions are met: Offset is smaller than length of the file, Offset plus size of the data to read is smaller than length of the file.

Read Data: Server Side

In line 5, the inode number is known. Messages storing data will be found by constructing following search term: "d*id_fs*inodeNumber*". Data message is a multipart message because it stores a file, thus it is necessary to retrieve message marked as an attachment. Then attachment has to be stored in the buffer, which will be returned to the client application.

Create Empty File: Client Side

The make node function allows the client to construct an empty file.

The "mknod" function passes 3 arguments: Path to the file to be created, Mode of the file consisting of both file permission and type of the node, Argument used only for bulk and character type files.

```
static int simfs_mknod(const char
*path, mode_t mode, dev_t rdev)
```

"mknod" passes the first 2 arguments to the remote procedure and waits for the result.

It is necessary to set flag at this point telling getattr function that file was created, as immediately after execution of mknod function, getattr function will be fired to check if creation of the file succeeds.

Create Empty File: Server Side

File creation is possible when permissions of the parent directory allow for writing (Line 3) and there is no file with the same name (Line 2). If two mentioned conditions were met,

Table 3.

Line 0	int checkFileMode(flags, fileMode)
Line 1	toOctal(fileMode): octalMode
Line 2	If regular file
Line 3	retrieveOwnerModeReg(octalMode): mode
Line 4	Else
Line 5	retrieveOwnerModeDir(octalMode): mode
Line 6	compare(flags, mode): errorNo

Table 4.

Line 0	int read(path)
Line 1	solvePath(path): parentDir, fileName
Line 2	hash(parentDir): hashedDir
Line 3	openFolder(hashedDir)
Line 4	getInode(filename): errorNb, inodeNb
Line 5	findMessage(inodeNb): messages
Line 6	getAttachment(messages): data

three messages that represent one file are sent (Line 4 – 6).

As all the messages by default are sent to the "inbox" directory, it is important to move messages to the directory specified in the path. The problem is that messages will not appear immediately on the e-mail account, so the method has to poll the e-mail server to check if the messages arrived and then move them to destination directory (Code 5-9). In order to return control immediately to the user, the MoveMessages() method runs parallel in a new thread.

Write File: Client Side

The write file operation allows the client to add data to the contents of the file.

"Write" function consists of 5 arguments: Path to file where data has to be written,

Buffer storing content of the file, Size of the buffer, Offset from beginning of the file, and information about write operation.

```
static int simfs_write(const char
*path, const char *buf, size_t size,
off_t offset, struct fuse_file_info
*fi)
```

The "Write" function uses copy command, where the data from the local file system is copied to our file system. Each read or write operation is executed by the Linux kernel by using the largest size of permitted blocks (4096B) [28], dividing bigger in to 4KB batches. In order to reduce number of connections with the web service, each batch is added to the buffer of the size 1200 * 4096B = 4915200B. When the buffer is full, data is sent to the remote procedure. The files bigger than 4915200B are stored in n attachments of the size close to 5MB. We used this solution to remove any constraints on the size of the file.

Write File: Server Side

Remote procedure will get following arguments: path to the file, data from the cumulative buffer and permission flag of the file. If the function succeeds, amount of the written bytes will be returned, otherwise negative value will be returned.

The function checks if the parent directory allows for writing in it, then it is necessary to delete file specified by the path if it already exists. The new file is created by sending three messages and is completed by returning the size of the file written.

Remove File: Client Side

The remove file operation is allow client to delete a file.

```
Static int simfs_unlink(const char
*path)
```

When the file has to be deleted, only file path is passed to the unlink function. This argument will be passed on to the remote procedure.

Table 5.

Line 0	int mknod(path, mode)
Line 1	solvePath(path): parentDir, fileName
Line 2	IF FileNotExists(path)
Line 3	IF ParentDirPermission(path, WRITE)
Line 4	createInode()
Line 5	createPath()
Line 6	createData()
Line 7	MoveMessages()

Zero will be returned if the file was successfully removed.

Remove File: Server Side

The "remove" file function checks if parent directory allows for modifications (Line 1), then using function open to check if file exists and if deletion is permitted. If conditions are met, we find i-node number of the file (Line 3) and search custom IMAP folder for rest of the messages with the same i-node number. We can remove the file simply by moving it to a trash or deleted folder of the email.

Rename File/Directory: Client Side

The rename function allows changing the file name or moving the file. This function also applies to directories. Two arguments: origin and destination are passed to remote procedure, which will return zero if operation finished successful.

```
static int simfs_rename(const char
*from, const char *to)
```

Rename File/Directory: Server Side

There are five different scenarios for the rename. (1) The first scenario is moving a directory with the contents to a destination directory, which involves creating a new custom IMAP folder with the same name like origin directory. All messages from the origin directory will be moved to the new one and the origin directory will be removed. (2) The second scenario is similar to scenario 1, however the new directory name is different. (3) The third scenario is overwriting destination file, where all messages describing existing files are deleted. The message storing file name of the origin file is deleted and the two remaining messages are moved to the destination path. Finally message with a new file name is sent to the destination path. (4) The forth scenario is similar to the third, however the destination file does not

Table 6.

Line 0	MoveMessages(from, to, term, numberOfNewMsgs)
Line 1	. . .
Line 2	DO:
Line 3	foundMsgNb = findMessages(term)
Line 4	Wait(5 seconds)
Line 5	WHILE: foundMsgNb != numberOfNewMessages
Line 6	MoveFoundMsg(from,to)

Table 7.

Line 0	int write(path, file, flags)
Line 1	solvePath(path): parentDir, filename
Line 2	IF ParentDirPermission(path, WRITE)
Line 3	IF FileExists(path)
Line 4	deleteInode(), deleteData(), deletePath()
Line 5	createInode(), createPath(), createData()
Line 6	MoveMessages()

have to be deleted. (5) In the final scenario, destination file name is not provided, thus the destination is created with the origin name. If the destination file exists, the third scenario is executed, otherwise fourth is executed.

Read Directory Client

The readdir function reads the content of the directory pointed by the path.

```
Static int simfs_readdir(const char
*path, void *buf,fuse_fill_dir_t
filler, off_t offset, struct fuse_
file_info *fi)
```

Function readdir takes five arguments: Path to the directory, Buffer for the directory content, Pointer to function allowing adding content to the buffer, Offset in directory entries, and Information about the readdir operation.

The path to the directory is passed to remote procedure. When the result will be returned by the web service method, filler function will be used to add directory content to the buffer. On success readdir will return zero.

Read Directory Server

All custom IMAP folders are represented as a hash code of the path. We obtain hash value of the path to the directory (Line 1) and when the folder is opened, it is possible to search it (Line 3). To list all the files within the directory, it is enough to construct following search term: “p*id_fs*”. It will be possible to find all messages storing the names of the files, so the last step (Line 4) is to retrieve file name from each message. The array of the file names in the given directory is consumed by the client application.

Create Directory Client

The mkdir function creates new empty directory path to the new directory and file permissions are passed to the mkdir function and passed on to remote procedure. On success zero will be returned.

```
static int simfs_mkdir(const char
*path, mode_t mode)
```

Table 8.

Line 0	int unlink(path)
Line 1	IF ParentDirPermission(path, MODIFY)
Line 2	IF open(path, WRITE)
Line 3	findInode(path): inodeNb
Line 4	findMessages(inodeNb): messages
Line 5	moveToTrash(messages)

Table 9.

	Origin	Destination
1	Directory	Existing directory
2	Directory	Not existing directory
3	File	Existing file
4	File	Not existing file
5	File	Existing directory

Create Directory Server

Unlike to regular files, directories are described only by two messages that are initially stored in the "inbox" folder (Line 5). In the Line 6 new custom IMAP folder is created. The name of this folder is the hash code of its path. The process is completed by moving two messages describing newly created folder to the parent directory

Remove Directory Client

Directory deletion is performed by the rmdir file operation.

To delete the directory (Code 5-19) it is necessary to know its path, which will be passed to the remote procedure. On success function will return zero, otherwise will be returned negative value.

```
static int simfs_rmdir(const char
*path)
```

Remove Directory Server

First must be checked if directory exists and its permissions allows for deleting (Line 2), next check if parent directory can be accessed. It is possible to remove only empty directory, so if directory will store files, function will return with negative value (Line 4 – 6). Assuming that empty IMAP directory was successfully deleted, it is possible to remove two messages describing directory (Line 7 - 9). These messages are stored in the parent directory of the directory that was deleted in Line 5.

RESULTS

The web services with specified functions were deployed. We invoked remote procedure from the client application simulating file system call and finally implement this into the FUSE framework.

The client application was moved to the FUSE framework using gSOAP. In the development of the read and write file system operations we noticed that Linux kernel split files into 4096B chunks, allow the initial implementation to work fine only for files smaller than size of the one chunk. To avoid sending data every time when write function is called we decided to save data in the buffer and send it when buf-

Table 10.

Line 0	String[] readdir(path)
Line 1	hash(path): hashedPath
Line 2	openFolder(hashedPath)
Line 3	findAllMessages(): messages
Line 4	retrieveFileNames(messages): fileNames

Table 11.

Line 0	int createDir(path, mode)
Line 1	solvePath(path): parentDir, fileName
Line 2	IF FileNotExists(path)
Line 3	IF ParentDirPermission(path, WRITE)
Line 4	addDirFlag(mode)
Line 5	createInode(), createPath()
Line 6	createFolder(path.hash())
Line 7	MoveMessages(parentDir)

fer is full or last chunk of the file was loaded. This has a problem, as there is no information if the current chunk of the file is the last one. We understood that when data is written to the file it must be opened and for every open function there is always one release function. Therefore we made last chunk of the file to be written and sent in the release function.

When the file is copied to different folder, normally file system creates a file in the new destination before data is written to it. In our solution it means that a new file has to be deleted immediately in order to send the messages with data. Instead of creating the new file immediately, a special flag is set and checked in the release function. If there was write function called in the chain of functions for current Linux command (for instance cp) then it means that new file does not have to be created.

The file system was mounted on the new e-mail account, which created a root directory allowing for using file system. We were able to perform the described file operations on the new mount.

CONCLUSION

We are able to realize a personal storage grid architecture on an online email account which allows file and directory operations. Our result shows how file systems can be built into a mail service, which is a common form of data space resource available on the Internet. An implemented personal storage grid architecture allows file and directory operations that can be mounted to any mail that provides IMAP protocols.

We carried out this study to implement a file system that wraps online services to provide

Table 12.

Line 0	int createDir(path)
Line 1	solvePath(path): parentDir, fileName
Line 2	IF open(path, WRITE)
Line 3	IF ParentDirPermission(path, WRITE)
Line 4	IF emptyFolder(path.hash())
Line 5	deleteFolder(path.hash())
Line 6	ELSE return -1
Line 7	findInode(path): inodeNb
Line 8	findMessages(inodeNb): messages
Line 9	moveToTrash(messages)

a single logical drive space for the end-user. This implementation shows that a file system can be built on a grid like architecture, which is scalable and works in conjunction with online resources and services. It is able to consume cloud applications such as online email services and present a file system layer to the end-user.

This realization of consuming various services and transforming these services into disk space resources shows how system resources can be integrated and used by the end-user. Currently speeds of accessing disk space using the above method is not as fast as our local SATA drives which is able to reach 286MB/s. Our future work will move towards improving the compatibility with other storage resources and improving the usability of the system by caching instructions of file operations.

FUTURE WORK

Currently our personal grid architecture consumes only a single e-mail account, thus next step will be to develop file system that allows for joining different free online storage resource by consuming other popular protocols.

The second direction is to increase speed of the file system and file operations by decreasing number of connections with the server. Further investigation can be made on caching metadata about the files for the current directory; such as the "getattribute" function is used most often. Structuring of the file system can be stored in the XML file on the local disk. Every time when will be needed information about the files from directory that was previously used they will be retrieved from the local disk instead of the server.

Less important improvement is to add service of the multi user file system, since currently it is possible to use file system only by one user.

REFERENCES

Alexandrov, A. D., Ibel, M., Schauser, K. E., & Scheiman, C. J. (1997). Extending the operating system at the user level: the Ufo global file system. In *Proceedings of the annual conference on USENIX Annual Technical Conference* (pp. 6-6).

Almeida, D. (1999). FIFS: a framework for implementing user-mode file systems in windows NT. In *Proceedings of the 3rd conference on USENIX Windows NT Symposium* (Vol. 3, pp. 13).

Araujo, R. B. (2007). *CurlFtpFS - A FTP filesystem based in cURL and FUSE*. Retrieved from http://curlftpfs.sourceforge.net/

Bovet, D. P., & Cesati, M. (2005). *Understanding the Linux Kernel*. Sebastopol, CA: O'Reilly Media.

Card, R., Ts'o, T., & Tweedie, S. (1994). Design and Implementation of the Second Extended Filesystem. In *Proceedings of the First Dutch International Symposium on Linux*.

Dornfest, R., Bausch, P., & Calishain, T. (2006). *Google Hacks* (3rd ed.). Sebastopol, CA: O'Reilly Media.

Driscoll, E., Beavers, J., & Tokuda, H. (2008). *FUSE-NT: Userspace File Systems for Windows NT. University of Wisconsin-Madison*. Retrieved from pages.cs.wisc.edu/~driscoll/fuse-nt.pdf

Engelen, R. V. (2009). *gSOAP 2.7.15 User Guide. FSU Computer Science*. Retrieved from http://www.cs.fsu.edu/~engelen/soapdoc2.html

Foster, I. (2002, July 22). What is the Grid? - a three point checklist. *GRIDtoday, 1*(6).

Gera, V. (2006, September 22). *Filesystems in User Space*. Retrieved from twiki.dsi.uniroma1.it/pub/Sistemioperativi3/OnLine/Fuse.pdf

Java EE platform. (2008). JAX-WS Reference Documentation. *Sun Microsystems*. Retrieved from https://jax-ws.dev.java.net/

Jones, M. T. (2007, October 30). *Anatomy of the Linux file system. IBM*. Retrieved from http://www.ibm.com/developerworks/linux/library/l-linux-filesystem/

Jones, R. (2007). *Gmail Filesystem - GmailFS*. Retrieved from http://richard.jones.name/google-hacks/gmail-filesystem/gmail-filesystem.html

Kantee, A. (2007). Pass-to-Userspace Framework File System. In *Proceedings of the 2nd Asia BSD Conference* (pp. 29-42).

Kantee, A., & Crooks, A. (2007). ReFUSE: Userspace FUSE Reimplementation Using puffs. In *Proceedings of the 6th European BSD Conference*. http://freshmeat.net/projects/fuseftp/

Luo, X. (2007). *IMAP Storage Filesystem*. Retrieved from http://imapfs.sourceforge.net/

Nguyen, B. (2004, July 30). *Linux Filesystem Hierarchy. The Linux Documentation Project*. Retrieved from http://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/

openXdrive. (2007). *The Xdrive Data Services Platform (XDSP) JSON API. Xdrive LLC*. Retrieved from dev.aol.com/xdrive_resources/json_apidocs/api-index.html

Pautasso, C., Zimmermann, O., & Leymann, F. (2008). Restful web services vs. "big"' web services: making the right architectural decision. In *Proceeding of the 17th international conference on World Wide Web* (pp. 805-814).

Thiesen, M. (2005, December). *FuseFTP*. Retrieved from

Wang, F. Z., Wu, S., Helian, N., Parker, A., Guo, Y., Deng, Y., & Khare, V. (2007). Grid-oriented Storage: A Single-Image, Cross-Domain, High-Bandwidth Architecture. *IEEE Transactions on Computers*, *56*(4), 474–487. doi:10.1109/TC.2007.1005

One Anchor Distance and Angle Based Multi-Hop Adaptive Iterative Localization Algorithm for Wireless Sensor Networks

S. B. Kotwal, SMVD University, India

Shekhar Verma, Indian Institute of Information Technology, India

G. S. Tomar, Malwa Institute of Technology, India

R. K. Abrol, SMVD University, India

Suryansh Nigam, SMVD University, India

ABSTRACT

This paper presents distance and angle measurements based Multi-Hop Adaptive and Iterative Localization algorithm for localization of unknown nodes in wireless sensor networks (WSNs). The present work determines uncertainty region of unknown nodes with respect to known (anchor) nodes using noisy distance and angle measurements. This node transmits its uncertainty region to other unknown nodes to help them determine their uncertainty region. Because of noisy distance and angle measurements, the error propagation increases the size of regions of nodes in subsequent hops. Using only one anchor node as reference, the proposed iterative localization algorithm reduces the error propagation of this noisy distance and angle measurements and the uncertainty region of all unknown nodes within a given communication range. The results clearly indicate the improved efficiency of the proposed algorithm in comparison with existing algorithms.

Keywords: *Angle of Arrival (AOA), Localization, Received Signal Strength (RSS), Time Difference of Arrival (TDOA), Time of Arrival (TOA), Wireless Sensor Networks (WSNs)*

I. INTRODUCTION

Developments in the field of electronic devices, components and in modern communication technologies has lead to the development of small, cheap, and smart sensor nodes (Stojmenovic, 2005; Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002). Hundreds or thousands of such nodes, able to sense the environment, compute simple tasks and communicate with each other, form a huge wireless sensor network (WSN) (Chong & Kumar, 2003). Collected information (e.g., temperature, humidity etc.) from relevant node then transmitted in a multi

DOI: 10.4018/jghpc.2010070103

hop fashion over direct neighbors to a data sink, where the data interpreted and action taken accordingly.

Localization in Wireless Sensor Networks (WSNs) refers to creation of a map of a WSN by determining the geographical coordinates of each and every node. A number of applications of WSNs like target tracking (Li, Wong, Hu, & Sayeed, 2002), forest fire surveillance, fluid quality monitoring in industry, intrusion detection, traffic management etc require information about physical location of sensor nodes in the network. Localization also helps in geographical data packet routing (De, Qiao, & Wu, 2003) and collaborative information and signal processing (Heidemann & Bulusu, 2000). More over once location of a node in network is known, the coordinates that eventually will save the size of data packet to be sent by it can simply replace its node ID. Many limiting aspects of a node like computation intensity, power consumption and memory location imposed by the small devices is to be considered in location identification in a WSN. Because of the random deployment nature of WSN, it is not feasible to place nodes while recording their locations one by one. A network consisting of 1000 nodes will require around 17 hours to localize whole network assuming 1 minute required for placement of each node while recording its location.

One possible solution is to equip nodes with GPS. But limiting factors like small size, limited computation power and energy source, the possible solution excludes use of GPS. In almost all localization techniques some percentage of nodes are assumed to know their location a priori. Nodes which known their location a priori are called anchors. These anchor nodes help in absolute localization of nodes in a WSN. Without any anchor, nodes create a local map of their own which may me translated, rotated or mirror image of the actual map. These anchor nodes may be the nodes placed manually while recording their locations or nodes with additional capability like GPS as in (Meguerdichian, Slijepcevic, Karayan, & Potkonjak, 2001). The percentage of anchors required for localization of a WSN depends upon the technique of localization adopted. The proposed algorithm uses only one anchor node with RF beam steering capability to localize a full WSN. Simulations show that only one anchor node is capable for localization of a WSN within an acceptable level of error in localization.

Rest of the paper is organized as follows. Section II describes the problem associated with localization of sensor nodes followed by the related work in section III with proposed solution in section IV. The simulation and results are given in section V. Conclusion in is Section VI.

II. PROBLEM STATEMENT

We define the localization problem as estimating the smallest region which has the highest probability of having a node. With the available hardware and software support, any node within one hop from anchor can determine its uncertainty region w.r.t. anchor node from data sent by anchor with received signal strength index (RSSI) and angle of arrival (AOA) measurements. Before working on an algorithm based on above measurements, the uncertainties and their effects need to be taken into consideration. Therefore we discuss these uncertainties as under:

A. ***Uncertainty due to noisy distance measurement:*** With available RSSI, the distance between two nodes can be approximated by:

$$U_{j2} = U_{ki} \oplus U_{k1} \quad (1)$$

Where $\widetilde{d_{ij}}$ is the distance between node i and node j, P_{tx}, power transmitted by ith node, P_{rx} power received by jth node and ∞, the propagation constant. Because of the inherent irregularity in RSSI measurement especially indoors due to multipath effects, it becomes practically impossible to predict exact distance between two nodes (Hel'en, Latvala, Ikonen, & Niittylahti, 2001). But it is practical to assume distance error e_0 for 1 meter to be 0.2 meters as in (Savvides, Han, & Strivastava, 2001). That means the distance range is,

$$d_{ij} = \widetilde{d_{ij}} \pm e \tag{2}$$

Where, $e = \widetilde{d_{ij}} * e_0$.

If $d_{ij1} = \widetilde{d_{ij}} - e$ & $d_{ij2} = \widetilde{d_{ij}} + e$, then node j is within $d_{ij1} \leq d_{ij} \leq d_{ij2}$ from node i as shown in Figure 1.

B. ***Uncertainty due to noisy angle measurement:*** We assume instead of the unknown nodes determining their angle of arrival, the anchor node is beam forming signal in all directions w.r.t north and the data packet sent in each direction contains the angle of beam formed also. This way, unknown nodes at one hop from anchor can determine their inclination w.r.t north also. The localized nodes at first hop can also steer beam in all possible direction w.r.t north and 2nd and so on hops can determine their inclinations also. If a node j is within communication range of node i, then to account for a beam width of transmitting antenna and inaccurate angle of transmission $\tilde{\theta}_{angle(ij)}$ at node j from node i, w.r.t some reference axis say north, has a variance of σ_{ij}. This way the node j may find itself within an annular region as specified by,

$$\theta_{ij1} \leq \theta_{ij} \leq \theta_{ij2} \tag{3}$$

Where, $\theta_{ij1} = \tilde{\theta}_{ij} - \sigma_{angle(ij)}$ and $\theta_{ij2} = \tilde{\theta}_{ij} + \sigma_{angle(ij)}$ as shown in Figure 2.

Experiments in [b] show that an assumption of $\sigma_{angle(ij)} \cong 5^0$ is practical.

Considering node i as an anchor node and inaccuracy in distance and angle measurements, the uncertainty region of a node j within communication within one hop of another node can be constructed as suggested in (Doherty, Pister, & El Ghaoui, 2001) a convex region as shown in Figure 3. We find that the uncertainty region created by anchor for all nodes in hop-1 will be trapezoidal in shape.

C. ***Localization error propagation:*** Any node in second hop cannot communicate with anchor directly; it determines its region from first hop node. Its uncertainty region is determined from all vertices of hop 1 node. The resultant region is much bigger in size as compared to the uncertainty region of node in hop 1, which means that uncertainty is increased as shown in Figure 4. This way error is propagated from hop-1 to hop-2 and so on. Due to this localization error propagation from 1st to 2nd and other hops, there arises a need of localization algorithm that can reduce error propagation and achieve high accuracy.

Figure 1. Distance constrain by noisy RSSI measurement

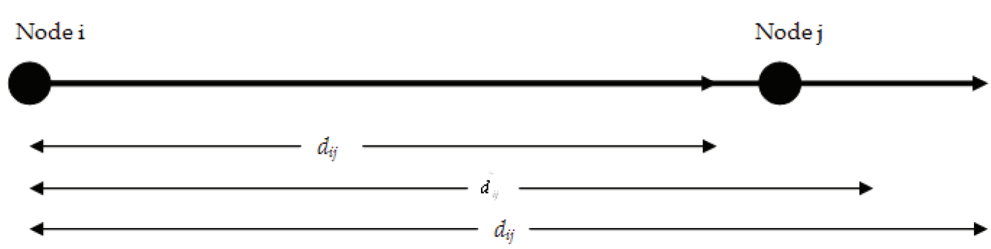
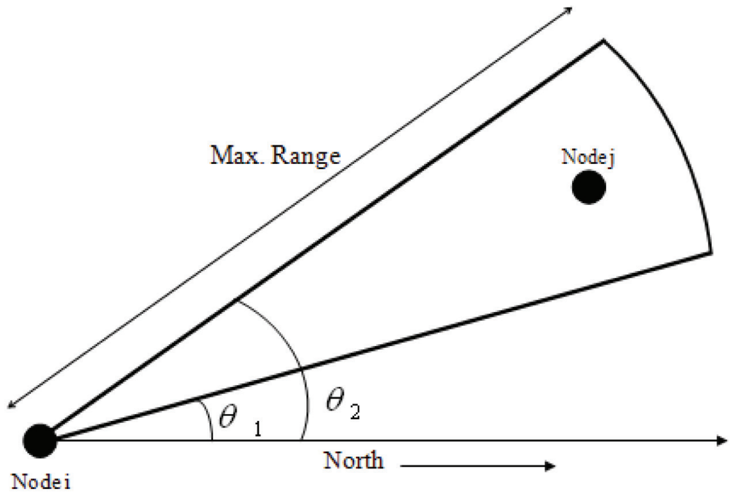


Figure 2. Uncertainty constrain by noisy angle of arrival



The problem of localization for a WSN can be formulated as a scheme to determine the uncertainty region for each node that approximates its location, such that:

$$\frac{1}{M}\sum_{j=1}^{M}\sqrt{(x_j - x_j^{'})^2 + (y_j - y_j^{'})^2} \sim 0 \tag{4}$$

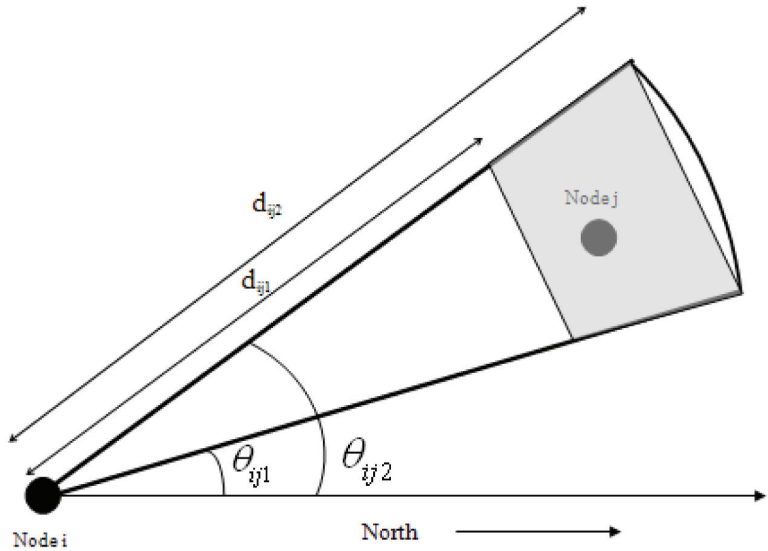
Where (x_j, y_j) are the actual (x, y) coordinates of a j^{th} node, in an M node WSN and (x_j, y_j) is the center of the uncertainty region of that node.

III. RELATED WORK

Form the measurement's perspective, localization algorithms are broadly based into two methods, i.e. range based and range free methods. The range based methods use measurements like distance and/or angle between nodes using received signal strength index (RSSI) (Elnahrawy, Li, & Martin, 2004), time of arrival(TOA), time difference of arrival (TDOA) or angle of arrival (AOA) (Peng & Sichitiu, 2006). Whereas the range free methods do not use the distance or angle measurements with respect to their neighbors. They use connectivity information to identify the nodes and in their radio range, and then estimate their position. The range free methods are simple to implement but lack in accuracy. A number of range based methods have been proposed in recent years. A very practical model for localization with hardware design is given in (Denegri, Zappatore, & Davoli, 2008). A novel localization scheme based on AoA measurements with noisy angle measurements is proposed in (Peng & Sichitiu, 2006). The practical model in (Denegri, Zappatore, & Davoli, 2008) uses three types of nodes (anchor, mobile and general purpose). The model in (Denegri, Zappatore, & Davoli, 2008) exploits advantages for single hop WSN. . In (Elnahrawy, Austin-Francisco, & Martin, 2007) a curve fitting model based on AoA and RSSI measurements is proposed. The scheme proposed needs previous data for training purpose.

The need of multi-hop, single anchor and less complex localization scheme can be estimated from related work. In contrast to all these approaches authors in this paper present a simple region intersecting method of localization without any data required for learning purpose.

Figure 3. Combining constrains of distance and angle to approximate. The uncertainty region of node-j is approximated as convex hull shown in gray.



IV. PROPOSED SOLUTION

Generally nodes in a WSN have fixed transmission power because of that nodes have limited communication range therefore it becomes necessary to divide nodes in groups to avoid data packet loss and save power.

The proposed model works under the assumptions that:

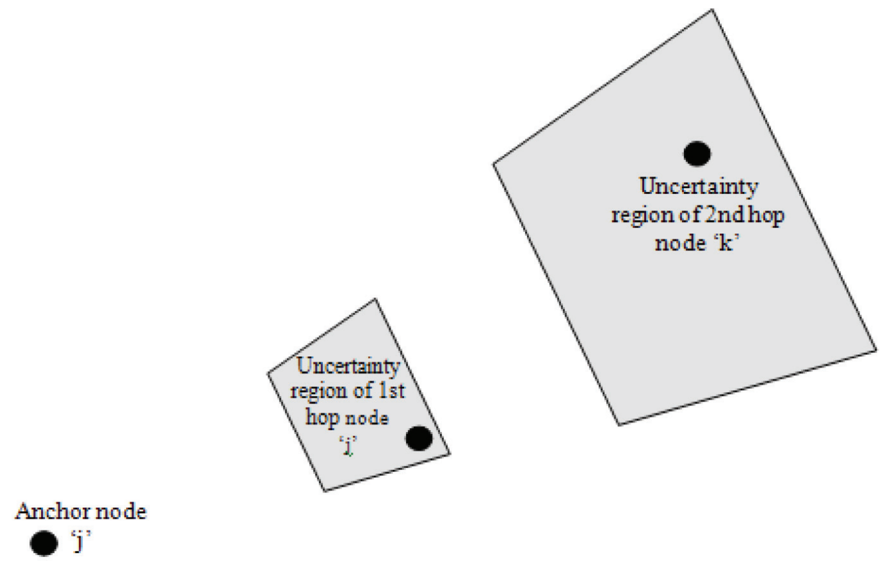
1. All nodes are deployed randomly.
2. At least one (anchor) node has its location known a priori.
3. Nodes are equipped with compass.
4. All nodes are capable of transmitting or receiving data packets in isotropic or anisotropic half duplex mode.

After deployment of nodes in a field, anchor sets itself to transmission mode and all other nodes in receiving mode. Anchor transmits data packet containing id, coordinates and angle of transmission information in all 360^0 directions while rotating its beam in anti clockwise direction from a reference line (North). The pseudo code (c-1) for anchor is shown below.

The hop value of anchor is set to zero. Our algorithm divides all nodes in N hops in terms of a distance from anchor node. The localization starts with determining the nodes in first hop. The algorithm is fully distributed; in fact it is requirement of the algorithm to be distributed as a node after running one iteration of proposed algorithm helps in reducing uncertainty region of other nodes. The distributed algorithm runs pseudo code (c-2) on each node except anchor node.

Localization by first hop nodes: A node-j in first hop runs pseudo code (c-2) given below to localize itself. It receives data packet from anchor and determines its hop count, its north direction and uncertainty region. The id field and hop field are used to determine its hop value. Based of AoA technique it determines the angle from which data packet of anchor is coming. From x, y coordinate field, it determines

Figure 4. Localization error propagation. Node-j determines its region from anchor node-i, and node–k at 2nd hop from anchor determines its uncertainty region from node-j. Note that there is a significant increase in the region of node-k



the distance range $(d_{ij1} \; \& \; d_{ij2})$. Angle field and AoA technique combine to give $(\theta_{ij1} \; \& \; \theta_{ij2})$. Combining all above information, node-j determines its x and y coordinates as follows:

$$x_{j1} = x_i + \begin{bmatrix} d_{ij1} \times \cos(\theta_{ij1}) \\ d_{ij1} \times \cos(\theta_{ij2}) \\ d_{ij2} \times \cos(\theta_{ij1}) \\ d_{ij2} \times \cos(\theta_{ij2}) \end{bmatrix},$$

$$y_{j1} = y_i + \begin{bmatrix} d_{ij1} \times \sin(\theta_{ij1}) \\ d_{ij1} \times \sin(\theta_{ij2}) \\ d_{ij2} \times \sin(\theta_{ij1}) \\ d_{ij2} \times \sin(\theta_{ij2}) \end{bmatrix} \quad (5)$$

Suppose nodes j & k are in the first hop. After determining their respective uncertainty regions as $U_{j1} = [x_{j1}, y_{j1}]$ and $U_{k1} = [x_{k1}, y_{k1}]$, nodes j & k share their regions using data packet as mentioned already. Node-k's x, y coordinates are non-singular therefore region of node-j w.r.t it will be large. Therefore node-j determines its uncertainty region w.r.t. the farthest vertex of node-k as U_{ki} The uncertainty region of node-j w.r.t all possible points inside U_{k1} is computed as:

$$U_{j2} = U_{ki} \oplus U_{k1} \quad (6)$$

This uncertainty region is intersected with U_{j1} and becomes U_{j1}

Same code is run by all nodes in first hop. After running its fixed number of iterations in n^th hop nodes, (n+1)^th hop nodes determine their hop count and location similarly till all N hop nodes are localized. The termination of algorithm depends upon two conditions, one compares the predefined number of iteration and second condition given in (c) to check if all vertices of U_{j1} are equal to a singular point. If any of the above condition is satisfied execution

Algorithim 1.

Pseudo code for anchor(c-1)
1. For 1m sec *Turn on transmitter & Transmit data packet, rotating beam in all 360 angles.* *2. Turn off transmitter and turn on receiver.* *3. Is there any request for re transmission* *a. If yes; got to step-1* *b. If not; repeat step 3*

stops. The further execution of algorithm will increase cost of communication without any advantage in localization error

Termination condition for pseudo code (c-2) is:

$$\sum_{k=1}^{L}(x_k - x_j^{'})^2 + (y_k - y_j^{'})^2 = 0 \qquad (7)$$

Here (x_k, y_k) is the kth vertex of an L-vertex uncertainty region of jth node and $(x_j^{'}, y_j^{'})$ is the centroid of the uncertainty region. Localization by subsequent hop nodes is done in a similar manner and termination The fixed number of iterations of same code (c-2) is run by all nodes in subsequent hops to determine and update their feasibility regions. The data packet format is shown in Figure 5. Authors propose same data packet format for all nodes.

V. SIMULATION AND RESULTS

A. Simulation Setup

The simulations for proposed algorithm are done in Matlab. The simulation setup consists of 100 unknown nodes deployed randomly in a field of 100m x 100 m field. Figure 6 shows 100 nodes divided into hops according to the distance from anchor node at $(x_i, y_i) = (50, 50)$. Every sensor node can communicate with other sensors if distance between them is smaller than sensor range. Transmission range for each node is fixed at 18 m. For all possibly communicable nodes, the algorithm for localization comprising of pseudo code -1 & 2 is implemented. The maximum number of iterations run for each node is only 5, practically the number of iteration may be < 5 for some nodes because of the termination condition given in (c). The localization error for each hop

Algorithim 2.

Pseudo code for all nth hop nodes (c-2):
1. For i=1 to n(iterations) *i) Update own hop count to n+1* *ii) While determining AoA, receive a data packet with highest RSSI* *a) If own hop count=1;* *determine own uncertainty region* *b) Else;* *determine own uncertainty region using eqn. (7)* *iii) Intersect with old region if present and update as old region* *iv) Transmit own data packet by steering rf beam anticlockwise w.r.t north* *v) Is termination condition given by (6) satisfied* *a) If yes;* *Go to step 2* *b) If not;* *Go to step 1(ii)* *2. exit*

Figure 5. Data packet format of a node

node id	hop count	coordinates		angle w.r.t north
		x	y	

nodes is determined for different values of variance in angle $\sigma_{angle(ij)}$.

B. Results and Discussion

Figure 7 shows the uncertainty regions after 1st and 5th iteration for all 100 nodes in 4 hops after running 5 iterations of algorithm with $e_0 = 0.1m$ and $\sigma_{angle(i,j)} = 5^0$. Figure 8 shows percentage of mean area reduction per iteration for all nodes in each hop from a reference area.

Reference area is of the uncertainty regions formed for a node at first hop from anchor as was shown in Figure 3. Coordinates (x_j, y_j) of an unknown node are approximated by taking centroid of the remaining region.

Figure 9 shows actual vs. estimated coordinates for all 100 nodes inside the field. The estimated locations and actual coordinates are almost overlapping with each other. Figure 10 below shows close view of some nodes from Figure 9 for better visualization of actual vs estimated locations for 4 nodes. We calculated error of localization for all nodes in different hops as a function of number of iterations to reflect the accuracy of scheme.

Figure 11 shows a graph of localization error in meters vs. number of iterations for all 4- hop nodes. Graph shows that for all hop nodes

*Figure 6. 100 nodes deployed in a field of 100m x 100m. Red,green, blue and orange dots indicate nodes at 1st, 2nd, 3rd and 4th hop from anchor respectively. Note that anchor is at (50,50) indicated by red *.*

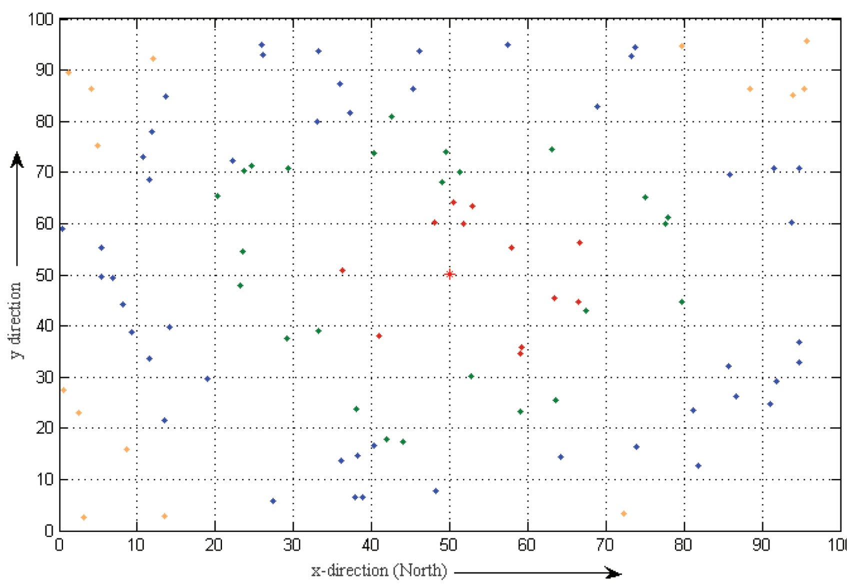


Figure 7. Regions with red, green, blue and orange after 1st iteration of algorithm for 1st, 2nd, 3rd and 4th hop nodes respectively. Regions reduced for each node after 5th iteration are shown by black patches.

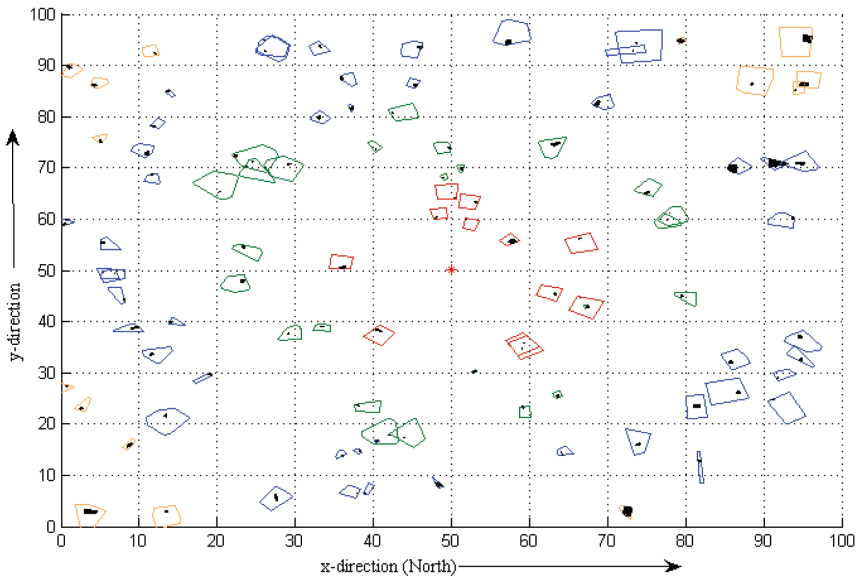
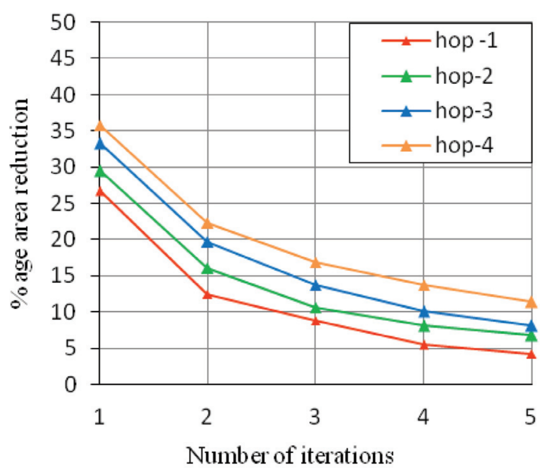


Figure 8. Percentage of area remained of nodes after each iteration



localization error decreases with increasing number of iterations of algorithm. Localization error is a decreasing exponential function of iterations and remains almost constant after 4^{th} iteration.

Running algorithm for number of iterations beyond 5 will not be practical as the cost of communication compared to accuracy required will be very high. There is a small increase in the error for 4^{th} hop node from 2^{nd} to third iteration,

Figure 9. Actual vs estimated locations. Actual and estimated locations are indicated by red 'x' and black 'o' respectively.

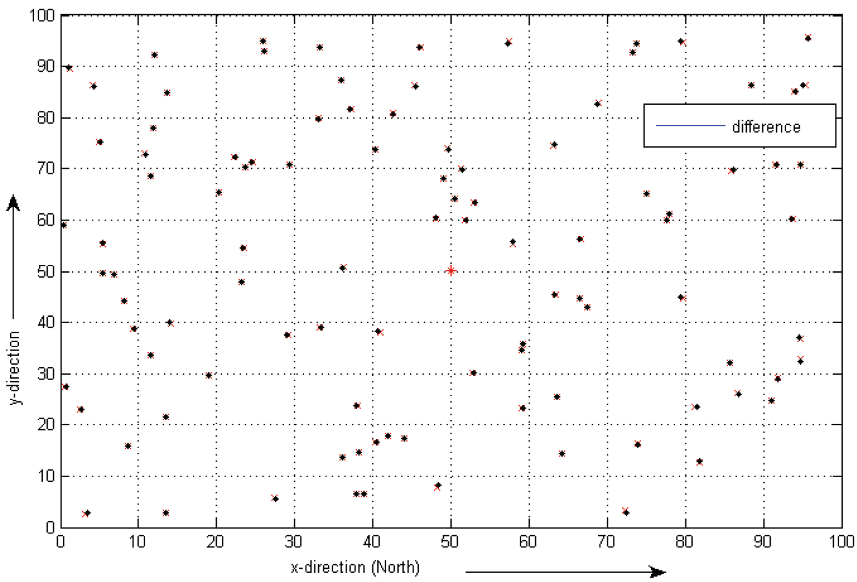


Figure 10. Actual vs estimated location for 4-nodes

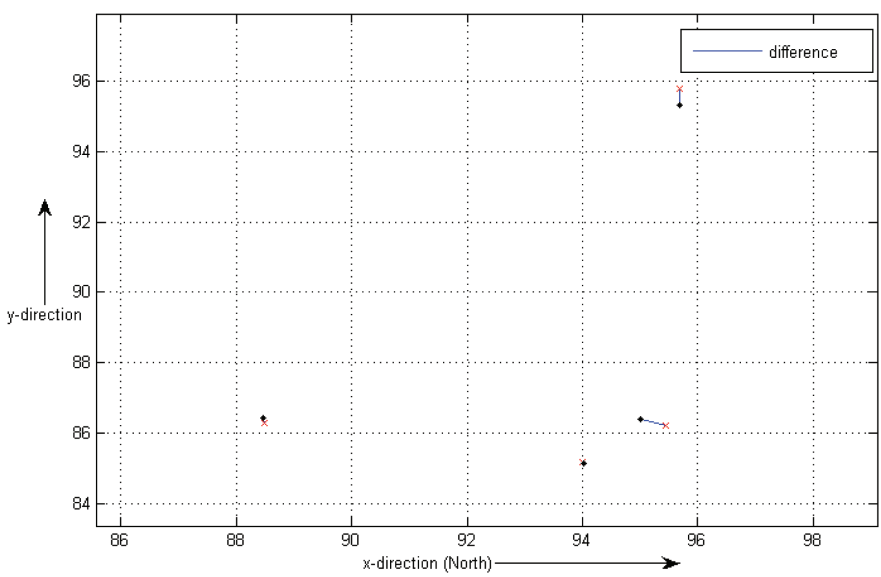


Figure 11. Localization error vs. number of iterations with $c_v = 0.1m$ *and* $\sigma_{angle(ij)} = 5^0$

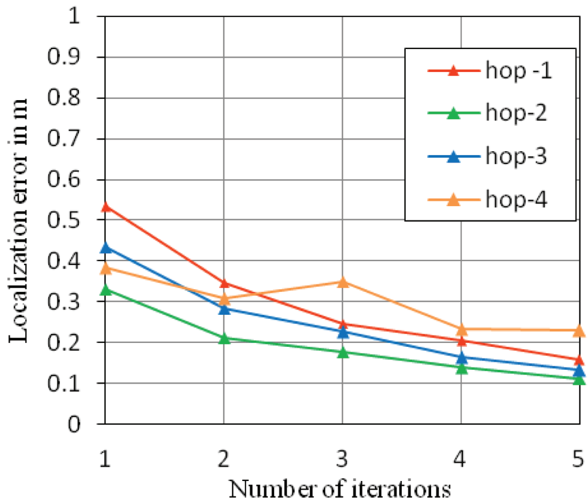
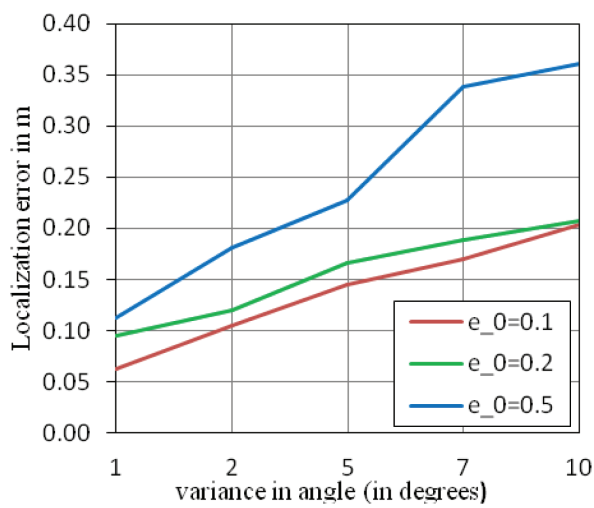


Figure 12. Performance analysis for different values of e_0 *and* $\sigma_{angle(ij)}$



this is because of the fact that estimated location after each iteration is assumed to be the centroid of convex hull (uncertainty region) and since the shape of convex hull changes with each iteration, the centroid may not always remain close to the actual position.

The average localization error of 142.7cm is obtained for all 100 nodes with $e_0 = 0.1m$ and $\sigma_{angle(ij)} = 5^0$. The simulation is repeated for different values of $\sigma_{angle(ij)}$ and e_0, plot is drawn in Figure 12. Comparing with the localization error of algorithm proposed in (Montillet, Braysy, & Oppermann, 2005) based on DFP and DM show minimum error nearly equal to 1m and 0.05m with DM and DFP respectively keeping $\sigma_{angle(ij)} = 10^0$. The proposed algorithm performs better in the sense that for $\sigma_{angle(ij)} = 10^0$ minimum error equals to 0.6m. Note that the in (Montillet, Braysy, & Oppermann, 2005) error is introduced for ToA for distance and AoA for angle calculations, and for minimum error case of nearly equal to 0.05m, error due to ToA is assumed to be 0, which is not practical. Our algorithm takes a minimum ranging error of 0.1m for 1m i.e. at least 10% and maximum of 50% of the distance.

VI. CONCLUSION

A new angle and distance based algorithm is proposed to localize nodes in wireless sensor networks. To the best knowledge of authors this algorithm is the only one that has used one anchor node to localize a full WSN of 100 nodes with high accuracy. Proposed algorithm exploits the advantages of multi-hop WSN in which the localized nodes in $(n)^{th}$ hop further reduce the size of uncertainty region of $(n-1)^{th}$ hop nodes. The proposed algorithm is robust to ranging error due to irregular RSSI as it can be seen from Figure 12 that increasing ranging error from 5 times increases localization error less than two folds. The advantage of cheap RSSI without increasing cost of hardware as in ToA or TDoA motivated authors to Take RSSI errors into consideration. In future authors wish to implement algorithm on hardware.

REFERENCES

Akyildiz, I., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002, August). A survey on sensor networks. *IEEE Communications Magazine*, *40*(8), 102–116. doi:10.1109/MCOM.2002.1024422

Cheng, K.-Y., Lui, K.-S., & Tam, V. (2009). HyBloc: Localization in Sensor Networks with Adverse Anchor Placement. *Sensors (Basel, Switzerland)*, *9*(1), 253–280. doi:10.3390/s90100253

Chong, C.-Y., & Kumar, S. (2003, August). Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, *91*, 1247–1256. doi:10.1109/JPROC.2003.814918

De, S., Qiao, C., & Wu, H. (2003). *Meshed Multipath Routing: An Efficient Strategy in Wireless Sensor Networks*. Computer Networks.

Denegri, L., Zappatore, S., & Davoli, F. (2008). Sensor Network-Based Localization for Continuous Tracking Applications: Implementation and Performance Evaluation. *Advances in Multimedia* (Vol. 2008, Article ID 569848).

Doherty, L., Pister, K. S. J., & El Ghaoui, L. (2001, April). Convex position estimation in wireless sensor networks. In *Proceedings of the IEEE Infocom 2001*, Anchorage, AK (Vol. 3, pp. 1655-1663).

Elnahrawy, E., Austin-Francisco, J., & Martin, R. P. (2007, February). Adding Angle of Arrival Modality to Basic RSS Location Management Techniques. In *Proceedings of the IEEE International Symposium on Wireless Pervasive Computing (ISWPC'07)*.

Elnahrawy, E., Li, X., & Martin, R. (2004). The limits of localization using signal strength: a comparative study. In *Proceedings of the First Annual IEEE Conference on Sensor and Ad-hoc Communications and Networks* (pp. 406-414).

Heidemann, J., & Bulusu, N. (2000). Using Geospatial Information in Sensor Networks. In *Proceedings of MOBICOM*.

Hel'en, M., Latvala, J., Ikonen, H., & Niittylahti, J. (2001). Using calibration in RSSI-based location tracking system. In *Proceedings of the 5th World Multiconference on Circuits, Systems, Communications & Computers (CSCC20001)*.

Li, D., Wong, K. D., Hu, Y. W., & Sayeed, A. M. (2002, March). Detection, Classification, and Tracking of Targets. *IEEE Signal Processing Magazine*.

Meguerdichian, S., Slijepcevic, S., Karayan, V., & Potkonjak, M. (2001). Localized Algorithms In Wireless Ad-Hoc Networks: Location Discovery and Sensor Exposure. In *Proceedings of the MobiHOC 2001*, UCLA, Los Angeles, CA.

Montillet, J. P., Braysy, T., & Oppermann, I. (2005). *Algorithm for Nodes Localization in Wireless Ad-Hoc Networks Based on Cost Function*. Paper presented at the International Workshop on Wireless Ad Hoc Networks (IWWAN 2005), London.

Peng, R., & Sichitiu, M. L. (2006, September). Angle of Arrival Localization for Wireless Sensor Networks. In *Proceedings of the Third Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, Reston, VA.

Savvides, A., Han, C. C., & Strivastava, M. B. (2001). Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking*.

Sichitiu, M. L., Ramadurai, V., et al. (2003). *Simple algorithm for outdoor localization of wireless sensor networks with inaccurate range measurements*.

Stojmenovic, I. (2005). *Handbook of Sensor Networks*. New York: Wiley. doi:10.1002/047174414X

Intelligent Industrial Data Acquisition and Energy Monitoring using Wireless Sensor Networks

Sumeet Gupta, SMVD University, India

Shekhar Verma, Indian Institute of Information Technology, India

G.S. Tomar, Malwa Institute of Technology & Management, India

Raj Kumar Abrol, SMVD University, India

ABSTRACT

Most of the application-oriented research in the field of Wireless Sensor Networks has been in remote monitoring, including environmental, building automation, and security. However, this paper presents the methodology followed for implementation of a Wireless Sensor Network based solution in a process plant for energy management and leak detection. The sensor network acquires data pertaining to detection of leakage in a plant. The network further serves effectively as a maintenance and diagnostic system that is used to manage the plant and conserve energy in a process plant. The critical design issues, testing methodologies and implementation problems pertaining to the system are also presented. Additionally, special focus has been placed on the calculations pertaining to the network life time.

Keywords: Data Acquisition Systems, Diagnostic Systems, Energy Conservation, Energy Management Systems, Industrial Plant, Wireless Sensor and Actuator Networks, Wireless Sensor Networks

INTRODUCTION

The rising cost of energy has made it absolutely essential for industries to ensure that energy wastage is minimized during every phase of the production process. Therefore the last two decades have seen a renewed focus on improving the efficiency of the process besides overall plant efficiency to reduce the production cost. Overall production cost includes the cost of various utilities like steam, compressed air, electricity, water etc. besides the raw material and labor cost. Any effort in improving the plant efficiency necessarily involves collecting the data pertaining to all these parameters, besides the critical process parameters and then acting upon it.

DOI: 10.4018/jghpc.2010070104

The collection of information needs to be done in a manner ensuring the accuracy and integrity of the data collected since this has a direct impact on process control.

The reliability of measurement, communication and the control action is critical for the system to work effectively. Various methods of instrumentation and processing of the data have continuously evolved over period of time. Fully integrated modern electronic SCADA systems have evolved which manage plant wide collection of information accurately and reliably.

A critical requirement of this integrated system is the transfer of the parameter measured by the sensor in the field to the centralized location. This transfer of information over the wired medium was traditionally done in the analog form, either current (the standard 4-20ma loop) or voltage (1-5V). With advancement in technology and reduction in cost, more intelligence could be placed at the sensor end itself and the transmission of data became digital. Protocols like RS232 for short-distance and RS 485 & HART™ for long-distance have become more popular using which it has become possible to collect plant-wide information.

Using the wired medium is laborious in terms of installation and maintenance. This also severely reduces re-location flexibility besides increasing the system cost. The obvious solution is to use wireless communication technology.

Wireless Sensor Networks have emerged as a possible solution for acquisition of data which is being sensed by multiple sensors over various locations in a large geographic area (Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002). These sensor nodes are inexpensive, have low computational capability and are energy constrained. They are however equipped with a radio for wireless communication and have the network and applications software capabilities to form a self-organized network, over which they pass the sensed data from each point to another. Since the energy constraint could severely reduce their working life, these nodes are generally used to sense some event rather than measure continuously in real-time.

The authors have proposed and implemented a Wireless Sensor Network based solution in an industrial environment and presented the same in details in the following sections. This paper is organized as follows- Section II presents the problem statement, Section III surveys the Existing Solutions, Section IV presents the proposed solution, Section V contains the design process. Section VI presents the deployment issues, Section VII presents the results, Section VIII presents the conclusions (Figure 1)

STATEMENT OF THE PROBLEM

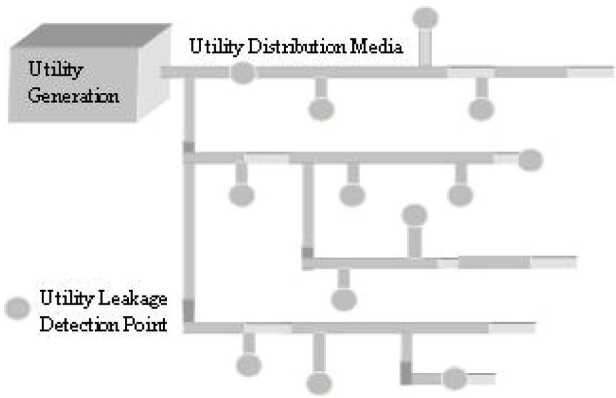
Utilities like water, electricity, compressed air, steam etc. are used commonly in a process plant. The utility is either generated within the plant itself (steam, compressed air) or is obtained from an external supplier (electricity, water). In both cases, the utility is further distributed in an efficient manner across the various sections of the plant.

Depending upon the size of the plant this could require kilometers of pipes (for steam, water, compressed air) laid down in an ordered manner to ensure that sufficient amount of utility is available at the right place in the plant to adequately supplement the production process.

In case the media carrying the utility springs a leak then the loss of the utility could have a sizeable impact on not only the cost of production but also on the quality of the manufactured product. There could also be an adverse impact on the manufacturing equipment which could get damaged because of the poor quality of the utility. E.g. leakage of compressed air would not only produce loss in terms of the energy spent to compress the air but also reduce the pressure of compressed air thus providing insufficient power to the equipment driven by the compressed air, making it work in-efficiently.

Detection of a leakage is therefore essential for maintaining the quality of the production process and for maintaining excellent plant efficiency. It is therefore desirable that leak detection equipment should be attached at multiple locations alongside the distribution media

Figure 1. Block schematic of the utility distribution network illustrating the utility leakage detection points in a process plant



to detect loss of the utility. Depending upon the size of the plant, there could be anywhere between 50-5000 such leak detection points.

Further since the distributing pipes criss-cross across the plant at different levels and locations, most of them being extremely hazardous in nature, these leak detection points are generally located at hard to reach places. The objective therefore was to design a system that would do the following:

a) Monitor the Leakage points to detect if any leakage is taking place.
b) In case any leakage is detected then convey the information regarding the type of fault to the utilities manager at a centralized location
c) Raise appropriate visual and audio alarm to indicate the fault situation
d) Maintain a record of the fault condition encountered
e) The system itself should require minimum maintenance and must have self- diagnostics built-in.
f) The system should be rugged and robust enough to withstand the harsh industrial environment over long period of time.
g) The system should conform to the relevant industrial standards and norms for operations & safety

EXISTING METHODS

A detailed survey of the existing available solution and their comparison shows the following:

a) *SCADA Systems:* These are highly reliable, powerful, expensive, centralized process monitoring and control systems with plant-wide data collection capability, generally used for monitoring and controlling the process parameters. Information is passed over the wired medium since the process parameters are critical and need high reliability communication. This solution was discarded because of the following considerations:
    - The collection of data from the large network of detection points will become an added task for the system which could significantly dampen the response of the system.
    - Another mitigating factor was the use of wires as the transmission media since the cost and installation of the wires would be prohibitively expensive and laborious.
b) *Telemetry based Systems*: This solution consists of using powerful transmitters, working over specific frequency band, located close to the detection points and having the ability to transfer the data to

a centralized location over long distance. This solution would also automatically circumvent the wired media problem. However this was also discarded based on the following considerations:

- ◦ High power transmitters may require large electrical power which can only be supplied by line power.
- ◦ Since large power is being transmitted over a long distance in an open media, a license is required to be obtained from the regulatory body.

- This large power transmission could have an impact on the working of other wireless and wired equipment with-in the plant.
- Directional Antenna needs to be installed at appropriate height for line of sight communication to work properly.
- The network formed is essentially a Star topology network.
- Jamming of specific frequency could adversely affect the performance of this network.

c) *Other Wireless Solutions:* Thompson et al. (n.d.) have demonstrated the use of a Bluetooth based peer-to-peer network for monitoring the engine performance in a marine vessel. This solution demonstrated excellent packet reliability but very poor battery life. Bonnet et al. (2003) have also presented pros and cons of Bluetooth based system. This solution was therefore discarded. Efforts have been made by various researchers to use the Wireless Sensor Networks in industrial applications (Shizhuang, Jingyu, & Yanjun, 2007; Jiang, Ren, Zhang, Wang, & Xue, 2006; Salvadori, de Campos, de Figueiredo, Gehrke, Rech, Sausen, Spohn, & Oliveira, 2007; Jeong & Nof, 2008; Lin, Liu, & Fang, 2007) where simulation based architecture has been presented for use in industrial process data monitoring have been provided. Zhou et al. (2007) have proposed architecture for using Zigbee based monitoring system for Greenhouse (Figure 2).

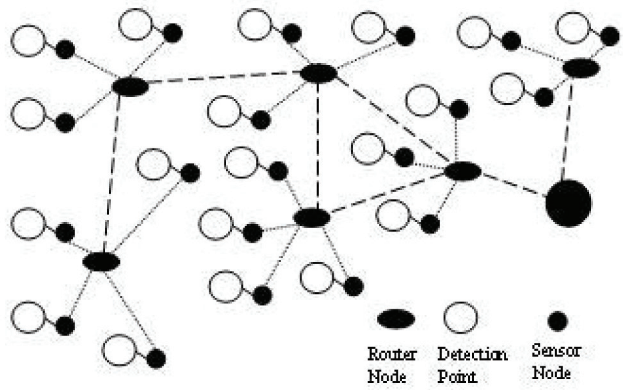
PROPOSED SOLUTION

After carefully studying the requirements of the system it was proposed to design & implement a wireless sensor network based solution for this problem. The factors which weighed heavily in favor of a WSN based solution were as follows:

a) Continuous monitoring of the Leakage detection points was not required and this could be done periodically.
b) Visual and audio alarms were required to be raised only when a leak was detected.
c) Real-time control action was not required
d) Since the detection points could be located at hard-to-reach places, a Wireless solution was highly desirable.
e) There were no regulatory or licensing requirements for using the 2.4GHz ISM band for low distance, low power wireless communication.
f) The sensors used for detecting the fault conditions need not be highly accurate
g) The noisy and harsh industrial environment required a rugged solution
h) Since low power communication is being performed it is extremely suitable for hazardous area applications.
i) The solution did not have to be very accurate but was required to be low cost because of the large number of detection points in a plant.

The solution envisaged having a Sensor Node mounted close to each Leakage Detection point which would monitor to detect occurrence of fault condition. These sensor nodes would self-organize themselves into a network and would pass the data corresponding to sensed values at each detection point to the centralized sink located at the Control panel in the Utility

Figure 2. Schematic of the wireless sensor network based solution



Manager's office. It was further proposed to use the IEEE 802.15.4 defined PHY & MAC layer and Zigbee™ (Zigbee Protocol V1.0, n.d.) defined NWK layer.

The sensor nodes would be powered with a non-rechargeable battery which would give a battery life of approximately 4 -5 years. A portable handheld display unit was also proposed to be designed which could be used by the Utility Manager for communicating directly with each individual Trap and change the configuration settings for the node if required. A customized GUI was also proposed to be designed which would run on a PC connected to the Sink and would store the data collected from each node. The Audio and Visual alarm would be available on the GUI for all the leak detection points. Trend analysis capability was also an objective.

The design of the enclosure was proposed such that the sensor node could withstand the harsh industrial environment having high temperature, high humidity, water, steam, corrosive gases without getting damaged.

The circuit design would also ensure that the sensor nodes could be used in hazardous areas where inflammable gas vapors could be present.

DESIGN CHALLENGES AND ISSUES

As the system consists of multiple elements which need to be designed the following section provides the design challenges and the design process for each element separately.

A. Sensing of Fault Conditions

The principle of resistivity and temperature were used for detecting the leakage of the utility at the various detection points. The sensors used were reliable and had the capability of working for long period of time in the harsh environment.

Under normal circumstances the measured temperature T_M

$$T_L < T_M < T_H \quad (1)$$

where T_L and T_H are the upper and lower limits of temperature being measured when no leakage is taking place. Depending upon the utility, the leakage condition would be indicated if the measured temperature T_M would be

$$T_L > T_M \textit{ for utility like compressed air} \quad (2)$$

$$T_M > T_H \textit{ for utility like steam, hot water} \quad (3)$$

Leakage of hot water or steam will increase the temperature whereas leakage of compressed air will actually reduce the temperature. The temperature reading is supplemented by the resistance reading R_M for detecting the leak condition. Under normal circumstances

$$R_M \to \infty \tag{4}$$

However if there is leakage of utility like Steam or hot water then

$$R_M \to 0 \tag{5}$$

Therefore a combination of the sensed values of temperature and Resistance can be used for detecting if any leakage is taking place. A specially designed sensor chamber is connected to the medium carrying the utility and the Temperature and Resistance sensors are located inside this chamber (Figure 3).

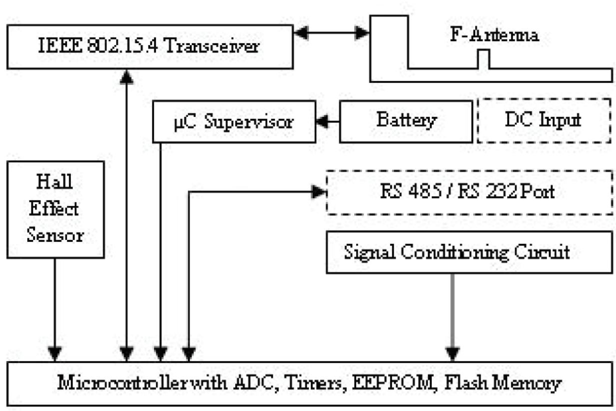
B. Design of Sensor Node

The sensor node is the primary part of the system and has numerous design constraints. Aakvaag et al. (2005) have provided some guidelines for design of sensor nodes for industrial applications. Some of the design requirements & challenges are provided below:

a) The node shall be battery powered and therefore the circuit design should ensure least power consumption.
b) Since sampling will be done periodically the signal conditioning circuitry should be optimized for high speed sampling while consuming less power.
c) It should be possible to put the node into active, idle or extremely low power sleep modes.
d) It should be possible to reset the node remotely as well as re-configure the node remotely since the nodes may be located in hard to reach places.
e) The circuit design and operation should be done to ensure that the battery should last at-least 4-5 years.
f) The node enclosure design should ensure survival of the electronics even in harsh industrial environment.
g) The Sink node should have the capability to transfer data serially to a PC for data logging.
h) IEEE 802.15.4 radio should be implemented to achieve compliance with the IEEE standard. Appropriate antenna design should be incorporated.

There are three distinct types of nodes – Coordinator Node or Sink, Router Node & Sensor

Figure 3. Architecture of Wireless Sensor Node, Components shown in dotted blocks are only available in Sink /Router node. The DC input block is used in router and coordinator nodes while the RS232 block is only available in coordinator/Sink node.



Node. The architecture of the three nodes is essentially similar. There is however a difference in their circuits depending upon the features incorporated depending upon their functions. The categorization of the sensor nodes based on their functional aspects is provided below:

a) *Sensor Node:* Consists of Battery, Signal Conditioning Circuitry, Transceiver, micro-controller supervisory circuit and Hall effect sensor.
b) *Router Node:* Is essentially line powered but has battery backup, Transceiver, micro-controller supervisory ciruit & Hall effect sensor
c) *Sink Node:* Is essentially line powered but has battery backup, Transceiver, micro-controller supervisory circuit, Hall Effect Sensor, RS 485 or RS232 based serial port

A suitable Freescale™ Semiconductors 8 bit micro-controller with 64KB of Flash memory, 4KB of RAM, 8 bit ADC, USART, Timers & RTI capabilities was chosen as the CPU. The microcontroller has multiple modes of operation including 3μA sleep mode and 7 mA active mode. A compatible IEEE 802.15.4 Transceiver (MC 13192, n.d.) with transmit current of 40mA & receive current of 42mA from the same manufacturer was chosen. The choice was also based on the ease of availability of Development tools and the IEEE 802.15.4 implementation of the stack for the micro-controller.

A Hall Effect Sensor is also included in the circuit and is connected to the RESET pin of the micro-controller. This provides the capability of resetting the node remotely from a distance. Special care is taken to protect the memory elements from interference from external sources.

The desired environmental protection rating for the Sensor node is IP65 i.e., protection from splashing water and dust particles of size 1micrometer. This has a direct impact on the antenna design. An external antenna though desirable is not feasible in such a design since this requires another opening in the enclosure making it more susceptible to water or dust ingress into the enclosure.

In this situation, the antenna has to be inside the enclosure, which further puts a constraint on the material to be used for the enclosure. The enclosure can't be metallic as it would adversely affect the transmission and reception of radio signals from antenna housed inside it. The enclosure was designed by using a special fiberglass material which could withstand high temperature up-to 70 degree Celsius, high humidity up-to 100%, rain, direct and long exposure to sunlight.

Since the antenna had to be inside the enclosure and the range was not a critical problem, a PCB embedded antenna design, popularly known as "F antenna" was chosen for the application. Locating the antenna on the PCB itself requires that the dielectric constant of the PCB material should be high and should not vary under severe environmental variations. An FR2 based PCB material was chosen for the PCB laminate, this is basically a paper material with phenolic binder. All components used were of SMD type since a small lead could actually act as an antenna itself putting the whole circuit into a feedback loop making it unstable.

C. Design of Sensing Algorithm

The sensing algorithm is required to be a generic algorithm which can be used and be re-configured depending upon the characteristics of the sensor in use. Sensors used for measurement of parameters like temperature have very low latency whereas the settling-time for parameters like resistivity and pressure is a few milliseconds. Considering this the algorithm has been designed to take multiple ADC readings and use an averaging algorithm to arrive at a more stable reading. This algorithm ensures that the sample value is actually an average of multiple readings to get better accuracy from an ADC

$$avgval = 1 \,/\, num\_samples \sum_{1}^{num\_samples} adc\_val \tag{6}$$

D. Design of Node Algorithm

The operation of the nodes is categorized into two distinct phases namely the *Network Setup phase* and the *Normal Detection phase*. During the *Leak Detection Mode Phase* the node runs the algorithm requesting permission to associate with an existing network. The initiation of this process is dependent upon whether the Router / Coordinator nodes are working in beaconed or non-beaconed mode. In the current application the routers and the coordinator nodes are working in non-beaconed mode and are perpetually in received mode.*Network Setup Phase*: On waking up for the first time the node checks to see if it is associated with some network. If no, then it initiates the process of association by first scanning the media for the nearest neighbors. This is handled by the MLME_SCAN primitive available in IEEE 802.15.4 MAC. Once it is able to detect the presence of the nearest coordinator node or a router node then it generates an association request which is handled by the MLME_ASSOCIATE primitive. Once the association is complete then an acknowledgment is sent by the coordinator to the node confirming the association. The coordinator also generates and sends a unique ID number for the node. The nearest router will update its routing table with information about the new node while the coordinator node will update its association table.

Leak Detection Mode Phase: During this phase the node wakes up every 8 seconds and takes a sample using the sensors. After taking the sample, the node goes back to sleep. This is repeated over a period of 2 minutes. Once 16 samples have been taken then a median value is found from this pool of samples. This median value then undergoes sensor validations tests including boundary value tests. Once it is verified that the sample value is correct then it is checked against the alarm conditions to establish whether the alarm needs to be armed.

The process of transmitting these sensed values is then initiated and a packet consisting of the device ID, Destination ID, Tag No along with the payload of the sensor readings and the alarm conditions is then transmitted. The sensor node then goes into wait mode for receipt of acknowledgment from the nearest router or coordinator node. On receiving the acknowledgment the node goes into sleep mode again and the process above is repeated.

The frequency of sampling, transmission & alarm thresholds are user configurable.

The underlying Zigbee NWK layer takes care of the routing of packets by using a combination of the popular AODV & Cluster Tree algorithms.

E. Estimation of Network Lifetime

The most critical design aspect of the Wireless Sensor Network based solution is the problem of the life-time of the network. This problem is even more critical in an industrial environment where the nodes are located in hard to reach locations in the field and replacement of battery is a difficult proposition.

The life time of the node is dependent on the rate of energy being drawn from the battery. A battery is a finite reservoir of energy and the rate of energy drawn will directly indicate the life time of the node.

The various functional aspects of the network which contribute to consumption of energy are:

a) Sensor Sampling (E_{sample})
b) Wireless Transmission of data ($E_{transmit}$)
c) Wireless Reception of data (E_{recv})
d) Node in sleep mode (E_{sleep})

The energy equation for the battery and the node is represented by:

$$E_{node}=E_{sample} + E_{transmit} + E_{recv} + E_{sleep} \quad (7)$$

The equations governing the calculations of the network energy consumption using which

Algorithm 1. Mean Value Sensing Algorithm

```
For all k ∈ (N,V)

Sense_param (channel_num, num_samples, settling_time)
avgval=0.0
activate (channel_num)
delay (settling_time)
for i= 0 to num_samples
  adc_val = read_adc_val (channel_num)
  avgval =avgval + adc_val;
end for
avgval =avgval / num_samples
```

Algorithm 2. Leak Detection Mode Algorithm

```
For all k ∈ (N,V)

On wake up(k)
activate(batt_life_channel)
  batt_life= read_adc_val(batt_life_channel)
if batt_life > L_Threshold then
  batt_flag=0;
else
  batt_flag=1;
  node(sleep)()
end if
if counter=16 then
  med_val_sensor1= call_func_median(sensor1);
  med_val_sensor2= call_func_median(sensor2)
  counter=0
  if(med_val_sensor1 > sensor1_High or med_val_sensor1< sensor1_Low) or (med_val_sensor2 > sensor2_High or med_val_
senso2< sensor2_Low)
    alarm_status=1;
    radio(wake);
    form_packet(median_val_sensor1, median_val_sensor2, alarm_status)
    init_transmit()
    wait ACK packet from coordinator
    radio(sleep);
    node_sleep(8)
  else
counter=0;
node_sleep(8)
  end if
else
counter=counter+1
  sens_val1[counter]=Sense_param(sensor1, 10, 50)
  sens_val2[counter]=Sense_param(sensor2, 10, 50)
  node_sleep(8)
end if
```

the lifetime of the network can be calculated are available at Appendix I. (Equations (7) –(26)

A sample calculation is shown for a particular case. In this case, it is assumed that 16 samples are taken over a period of 2 minutes. The transmission of data occurs every 2 minutes and a packet of 49 bytes is transmitted.

Each sample is actually an average of 10 ADC readings, as explained in IV (C). Before taking a sample, a sensor stabilizing delay of

50msec is provided. Using Eqn 10, therefore the duration of each sample is T_{sample}=50msec + 10 * 240µsec=52.4 msecs. The energy consumed by the micro-controller and the signal conditioning circuitry for taking one sample can be calculated by using Eqn 12 and is 3.6V * (20/1000)A * (52.4/1000) sec=0.0037728 Joules. The total number of samples taken in a day for this configuration can be calculated using the sample frequency of 8 samples /minute. Using Equation 11, this gives $N_{samples/day}$=24*60*8 = 11520 samples/day. Thus total energy spent in a day on the sampling process can be calculated by using equation 14. For the given case this works out to be 43.46 Joules/day. The next step is to calculate the energy spent by the node in transmitting & receiving packets. This can be calculated by using the equations 15-24

Continuing for the case mentioned earlier $T_{packet\_duration}$=8 * 49 / 250000 = 1.57msec, E_{packet} = 1.57* 3.6 *60 / (1000*1000) =0.000339 Joules / per packet. Further in this case $N_{packets/day}$=24*30= 720 packets / day. This results in total energy spending of $E_{transmit/day}$ = 0.000339 * 720 = 0.244166 Joules/day

A similar set set of calculations will result in $E_{recv/day}$ = 0.9766 Joules/day.

When the node is not performing the task of sampling, transmitting or receiving then it goes into sleep mode where it consumes only 3µA of current. In this case

$T_{sample/day}$ = 52.4 * 11520/1000=603.6 seconds
$T_{transmit/day}$ = *1.57 *720/1000*= 1.1304 seconds
$T_{recv/day}$ = 1.57*720*4/1000 = 4.5216 seconds
$T_{sleep/day}$ = (24*60*60) –(603.6 + 1.1304 + 4.5216) = 85790 seconds
$E_{sleep/day}$=3.6 * 0.003 * 85790 = 0.926532 Joules/day
$E_{total/day}$= 43.46 + 0.244 + 0.976 + 0.926 = 45.606 Joules / day.

A 8500mAH Li -Thionyl -Chloride battery was chosen to serve as the energy source. This battery has the highest power density as well as almost constant voltage characteristics over its lifetime as compared to other standard batteries. As per the calculations, the battery is expected to provide power for duration of 6 ½ years without a change. This is in tune with the desired characteristics. The life time of the node will change with different settings of sampling rate; sleep time ratio & transmission / reception frequency. A detailed simulated analysis of the lifetime of the network based on the various possible settings was done and the results are provided in Table 3

DEPLOYMENT

A total of 55 nodes (sensor nodes & router nodes) were deployed in a process plant for a period of 4 months. The location of the leak detection points was fixed by the utilities manager and the sensor nodes were installed adjacent to these points. Thus the location of the sensor nodes was static and inflexible. A site survey was subsequently done to establish the number and location of router nodes such that all sensor nodes would get radio connectivity. The objective was to identify the location of the routers such that least number of routers would get used whilst providing comprehensive connectivity. An upper limit of 25meters was kept as the radius for nodes having line of sight connectivity with router and 18 meters for those not having direct line of sight connectivity. Extremely low signal zones were also identified.

TESTS AND RESULTS

Table 1 shows the test results obtained after testing the sensor node for the various functional tasks that it is required to perform during its lifetime. Since the sensor node is required to spend most of its lifetime in the sleep mode therefore the extremely low value of sleep current of 3µA is advantageous for the life time of the node.

The microcontroller consumes a total of 8mA in active mode and along with the sensor conditioning circuitry consumes approximately 20mA. The resistivity sensor uses an alternating signal of 10 KHz, thus consuming more current.

This sensor also requires approximately 50msec to stabilize after turning on before it can give a reliable reading.

The total current consumed during transmission and reception at highest transmitting energy of 0dBm nominal power and sensitivity of -92dbm is approximately 60mA. However, as seen from analysis presented in Table 3 it can be seen that this figure is not overtly critical to the performance of the network.

Table 2 shows the power consumed by the sensor node for performing the various tasks during the normal mode of working. As can be seen from the table the maximum amount of energy is consumed by the node in the sampling process while it consumes the least amount of energy in the sleep mode.

Table 3 shows the effect on the lifetime of the network with variations in the sampling rate and the frequency of transmission and reception of data. The results indicate a network life pattern which is quite unique for process industries applications and varies significantly from conventional applications of Wireless

Table 1. Sensor node test results for current consumption

MCU Sleep Mode 3µA
MCU Active with Sensor Conditioning Circuit 20mA
MCU Packet Transmission / Reception 60mA
ADC Conversion time 240µsec
Sensor Settling Time 50msec

Table 2. Task based energy consumption results

MCU Sleep Mode 0.0000108 J/sec
MCU Active with Sensor Circuit ON 0.0037 J/sample
MCU Transmission /Reception 0.00033 J/packet

Table 3. Simulated results for network lifetime

Sampling rate	Communication Frequency	Network Life
8 samples/min	Once every 2 minutes	6.53 Years
8 samples/min	Once every 5 minutes	6.64 Years
8 samples/min	Once every 15 minutes	6.69 Years
4 samples/min	Once every 2 minutes	12.48 Years
4 samples/min	Once every 5 minutes	12.88 Years
4 samples/min	Once every 15 minutes	13.06 Years

Sensor Networks. In conventional applications of Wireless Sensor Networks the transmission of information tends to be event driven. The nodes transmit data only on detecting a specific event. In some cases there is some sporadic periodic transmission also to indicate the status of the node.

However, in industrial environment there is a requirement that the sensing of the phenomenon should occur frequently so that any significant change of the parameter under observation can be reported. In such applications, the energy spent by the network for sending far outweighs the energy spent on communication. Table 3 illustrates this phenomenon quite clearly.

It can be seen from the analysis provided in Table 3 that in case of Industrial applications the frequency of communication does not have a drastic impact on the lifetime of the network as compared to rate of sampling. The rate of sampling is higher in industrial applications since the need for fresh data is higher. It can be safely concluded that for industrial applications involving use of Wireless Sensor Networks, it is desirable to focus more upon the frequency and energy requirements of the sampling process as compared to the communication process. A reduction in the power consumption of the sampling circuit and a reduction in the frequency of the sampling process need to be the focus area for network lifetime enhancement. It can therefore be safely concluded that for industrial applications of wireless Sensor Networks, focus area for improving upon the network lifetime must be the sampling process rather than the communication process.

A test set up was created to study the packet loss rate in an industrial environment at different transmission rates. A specifically noisy part of the process plant was chosen for this test. The location consisted of high speed high rating DC and AC motors which were being switched on & off intermittently. Welding work was also in progress in the vicinity. A Wi-fi computer network was working in the same environment. One node was designated as the Transmitter node and a fixed packet of 49 bytes was transmitted at different rates. Another node was designated as a receiver node and was kept at a distance of 25mtrs from the transmitter node. The two nodes were partially in line of sight. The test results are indicated in Table 4.

Analysis of the test results indicates clearly that the packet loss is very high at high transfer rates and reduces significantly at lower transfer rates. The drop in the packet loss rate however is not linear.

A similar test set up was created to study the radiation pattern of the antenna. This consisted of a sensor node transmitting packets continuously at peak power (4dBm) in an open environment. A similar node connected to a Laptop acted as a receiver (-92dBm sensitivity) and passed on the received signal strength to the Laptop where the data was recorded along with the position. Based on the readings obtained the radiation pattern for the antenna was arrived at. As can be seen from Figure 4, the radiation pattern to an extent is isotropic.

Over the period of 4 months of deployment a total of 7 nodes malfunctioned due to different reasons. The failure information is presented in

Table 4. Measured packet error rate for the network

Packets Trans-mitted in 1 hour	Packets Re-ceived	Packet Loss
3590	2893	19.4%
2000	1802	9.9%
1000	957	4.3%
500	481	3.8%
100	97	3.0%

Figure 4. Radiation Pattern of "F Antenna". As can be seen the pattern is quite similar to that of an isotropic antenna

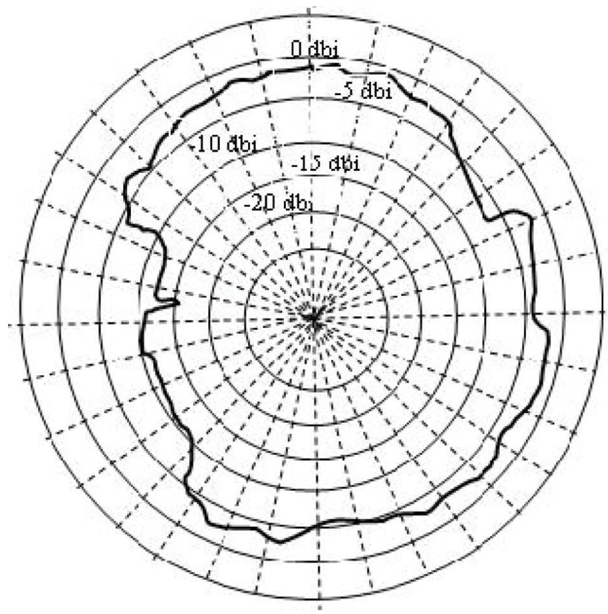


Table 5. As can be seen from the table, three nodes were rendered unusable because of battery failure. In all the three cases, the battery failure occurred within the first 30 days of installation. The same nodes were subsequently refitted with new batteries and working satisfactorily for the duration of the deployment. This battery failure was ascribed to the harsh environmental condition and component failure. The enclosure failed in one case which led to ingress of water into the enclosure resulting in damage to the sensor node. The sensor node was not recoverable in this case. In two cases the sensor nodes malfunctioned due to failure of the electronic components. An analysis required change of Balun in one case and change of capacitor in

Table 5. Sensor node failure data for the network

Total Nodes Deployed	55
Deployment Duration	4 Months
Total Nodes Failed	7
Node failure because of	
Sensor Failure	1
Battery Failure	3
All within first month	
Circuit Component Failure	2
Enclosure Failure	1

radio circuit in another case. Upon the replacement of the components, the nodes behaved satisfactorily.

CONCLUSION

A framework for design and installation of a Wireless Sensor Network for use in Process industry for monitoring of Utilities has been arrived at. All aspects of the design requirements for such an implementation have been taken into consideration. It has been observed that for industrial applications, the critical design parameters for Wireless Sensor Network Design are significantly different from that of a standard environment sensing applications. A customized design for a Wireless Sensor Node has been arrived at and the same has been fabricated and implemented Based on typical industrial application it has also been observed that the conventional approach of focusing more on the communication aspects of the Wireless Sensor Network for increasing the network lifetime is not particularly effective when used in industrial applications. The energy requirement of the sampling process and the frequency of the sampling process have a more pronounced impact on the life time of the network.

Currently the router nodes and the coordinator nodes are required to be powered on perpetually and are therefore line powered. This puts a severe constraint on the number of router nodes as well as their location. Further work can be done to arrive at a scheduling algorithm which can be used to enable these nodes to get some sleep time which could generate the possibility of the router nodes also being battery powered. Some interesting work in this direction has been shown in (Kouba, Cunha, & Alves, 2008; Aakvaag, Mathiesen, & Thone, 2005). Further work also needs to be done to arrive at a more optimized circuit design for low energy sensing with increased time period between the samples without sacrificing the quality of the sensing (Figure 4).

ACKNOWLEDGMENT

The authors would like to acknowledge the support received from M/s Polynol Rubbers for permitting the plant to be used for installation of the Wireless Sensor Network based solution.

REFERENCES

Aakvaag, N., Mathiesen, M., & Thone, G. (2005). Timing and power issues in wireless sensor networks - an industrial test case. In *Proceedings of the Parallel Processing (ICPP 2005)* Workshops (pp. 419-426).

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless Sensor Networks: A Survey. *Computer Networks*, *38*, 393–422. doi:10.1016/S1389-1286(01)00302-4

Bin, Lu., Habetler, T. G., Harley, R. G., & Gutierrez, J. A. (2005, October 30). Applying wireless sensor networks in industrial plant energy management systems. Part II. Design of sensor devices. *Sensors (Basel, Switzerland)*, 6.

Bonnet, B. (2003, December). Bluetooth-based sensor networks. *ACM SIGMOD, 32*(4).

IEEE. 802.15.4. (n.d.). *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)*. Retrieved from http://www.ieee.org

Jeong, W., & Nof, S. (2008, June). Performance evaluation of wireless sensor network protocols for industrial applications. *Journal of Intelligent Manufacturing*, *19*(3), 335–345. doi:10.1007/s10845-008-0086-4

Jiang, P., Ren, H., Zhang, L., Wang, Z., & Xue, A. (2006). Reliable Application of Wireless Sensor Networks in Industrial Process Control. In *Proceedings of the Intelligent Control and Automation (WCICA 2006), the Sixth World Congress on* (pp. 99-103).

Kouba, A., Cunha, A., & Alves, M. (2008, October). TDBS: a time division beacon scheduling mechanism for ZigBee cluster-tree wireless sensor networks. *Real-Time Systems*, 321–354. doi:10.1007/s11241-008-9063-4

Lin, S., Liu, J., & Fang, Y. (2007, August 18-21). “ZigBee Based Wireless Sensor Networks and Its Applications in Industrial. In *Proceedings of the Automation and Logistics, 2007 IEEE International Conference on* (pp. 1979-1983).

Low, K. S., Win, W. N. N., & Er Meng, J. (2005). Wireless Sensor Networks for Industrial Environments. In *Proceedings of the Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on* (pp. 271-276).

MC 13192. (n.d.). *Transceiver Data-sheet*. Retrieved from http://www.freescale.com

Salvadori, F., de Campos, M., de Figueiredo, R., Gehrke, C., Rech, C., Sausen, P. S., et al. (2007, October 3-5). Monitoring and Diagnosis in Industrial Systems Using Wireless Sensor Networks. In *Proceedings of Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on* (pp. 1-6).

Shen, X., Wang, Z., & Sun, Y. (2004). Wireless sensor networks for industrial applications. In *Proceedings of the Intelligent Control and Automation (WCICA 2004), Fifth World Congress on* (Vol. 4, pp. 3636- 3640).

Shizhuang, L., Jingyu, L., & Yanjun, F. (2007, August 18-21). ZigBee Based Wireless Sensor Networks and Its Applications in Industrial. In *Proceedings of Automation and Logistics, 2007 IEEE International Conference on* (pp. 1979-1983).

Thompson, H. (n.d.). *Bluetooth based Monitoring system for Marine Propulsion systems*. Rolls Royce University Technology Centre in Control & Systems Engineering, U. K.

Zhou, Y., Yang, X., Guo, X., Zhou, M., & Wang, L. (2007, September 21-25). A Design of Greenhouse Monitoring & Control System Based on ZigBee Wireless Sensor Network. In *Proceedings of Wireless Communications, Networking and Mobile Computing (WiCom 2007)* (pp. 2563-2567).

Zigbee alliance. (n.d.). *Zigbee Protocol V1.0*. Retrieved from http://www.zigbee.org

APPENDIX I

Equations related to Total energy spent/day

$$E_{node}=E_{sample} + E_{transmit} + E_{recv} + E_{sleep} \quad (7)$$
$$E_{node/day} = E_{sample/day} + E_{transmit/day} + E_{recv/day} + E_{sleep/day} \quad (8)$$
$$E_{total/day}= E_{sample/day} + E_{transmit/day} + E_{recv/day} + E_{sleep/day} \quad (9)$$

Equations Related to Sampling Process

$$T_{sample} = T_{sensor\_stable} + 10 * T_{ADC} \quad (10)$$
$$N_{samples/day} = 24 * 60 * Sample\ frequency \quad (11)$$
$$E_{sample} = V_{battery} * I_{sample} * T_{sample} \quad (12)$$
$$T_{sample/day} =T_{sample} * N_{samples/day} \quad (13)$$
$$E_{sample/day} = \sum_{1}^{N_{samples/day}} E_{sample} \quad (14)$$

Equations Related to Transmit Process

$$E_{packet} = T_{packet\_duration} * V_{battery} * I_{transmit} \quad (15)$$
$$T_{packet\_duration} =8 * packet\ size / Bit\ Rate \quad (16)$$
$$N_{packets/day} = 24*Packet\_frequency / hour \quad (17)$$
$$T_{transmit/day} = T_{packet\_duration} * N_{packets/day} \quad (18)$$
$$E_{transmit/day} = E_{transmit\_packet} * N_{packets/day} \quad (19)$$

Equations Related to Receive Process

$$E_{packet} = T_{packet\_duration} * V_{battery} * I_{treceive} \quad (20)$$
$$T_{packet\_duration} =8 * packet\ size / Bit\ Rate \quad (21)$$
$$N_{packets/day} = 24*Packet\_frequency/hour \quad (22)$$
$$T_{recv/day} = T_{packet\_duration} * N_{packets/day} * 4 \quad (23)$$
$$E_{recv/day} = E_{packet} * N_{packets/day} \quad (24)$$

Equations Related to Sleep Process

$$T_{sleep/day} = (24*60*60) - (T_{sample/day} + T_{transmit/day} + T_{recv/day}) \quad (25)$$
$$E_{sleep/day} =V_{battery} * I_{sleep} * T_{sleep/day} \quad (26)$$

Design of SOA Based Framework for Collaborative Cloud Computing in Wireless Sensor Networks

S. V. Patel, Veer Narmad South Gujarat University, India

Kamlendu Pandey, Veer Narmad South Gujarat University, India

ABSTRACT

WSN deployments are growing at a fast rate; however, current WSN architectures and setup do not promote the sharing of data on an inter-WSN basis. Cloud computing has emerged as a promising area to deal with participatory and collaborative data and services, and is envisaged that collaborative cloud computing WSN could be a viable solution for sharing data and services for WSN applications. In this paper, SOA based architecture has been proposed to support collaborating cloud computing in WSN. The architecture consists of layered service stack that has management, information, presentation and communication layers with all required services and repositories. Interactions between WSN, subscribers and other cloud are also presented as sequence diagrams. The proposed framework serves the cloud subscribers with wide range of queries on the data of multiple WSNs through suitable interface to solve large scale problems.

Keywords: *Cloud Computing, Collaborative Approach, Registry, Service Oriented Architecture, Wireless Sensor Network*

INTRODUCTION

The technology of wireless sensor networks has completed a decade or so. In these years researchers, deployers, application programmers have put tremendous efforts to make these new technologies reach to the beneficiaries. The benefits of sensor networks are well realized and many deployments are taking place to study the serious problems of environmental monitoring, habitat monitoring, healthcare, production system, inventory management etc. However, for the solutions to some gross problems like global warming or pollution etc. we need to collect, understand the data as well as its patterns in totality and wide scale. This will not be possible unless diverse physical WSN collaborate. The cloud computing comes as a possible answer to the problem. To achieve this task there is an urgent need to rethink about the present architectures orientation and seclusion

DOI: 10.4018/jghpc.2010070105

of wireless networks. Further, there is a need to change the perception of WSN organisation.

Our earlier work described in this paper, prepared the background in the direction of this new structure. In this paper the problem pertaining to the current state of WSNs architecture is identified, the need of new architecture supporting cloud computing on WSN is emphasized and a solution comprising of a framework having Service stack divided in four major layers, each layer supporting various services and repositories, is proposed. Functioning of the architecture and interaction sequences between major players like WSN, Subscribers and Other clouds are also described. The new architecture uses the features of cloud like virtualisation and Service oriented architecture to deal with heterogeneity.

The paper is structured as follows: related work on various architectures that focus on data sharing and limited WSN collaboration and limitations of current WSN architectures are presented. The path toward collaborative model is also introduced. The model with service stack for WSN cloud is described and the need of SOA approach for achieving heterogeneity is stated. The functioning of WSN cloud is also shown. Finally the need of WSN cloud to solve the global problem is emphasized.

RELATED WORK

Looking to the scenario of research in the area of cloud computing pertaining to sensor networks we don't find much published work. However there has been an effort depicting architectures which employ grid-based or publish subscribe models to achieve efficiency, availability, scalability and collaboration to some extent for the present and future wireless sensor networks deployments. OGC open Geospatial consortium (Botts, Percivall, Reed, & Davidson, 2006) is engaged in promoting the agile Sensor Web Enablement for the wireless sensor networks. It has developed XML schemas for configuring a WSN ready to serve on web. It also presents connectivity architecture to deliver the sensor data. Chen Khong et al (Tham & Buyya, 2005) present a report on sensor grid network to deal with the processing of the data and application logic by putting them into a grid based architecture. Mark Gaynor et al show how to integrate WSNs with a grid (Gaynor, Moulton, Welsh, LaCombe, Rowan, & Wynne, 2004). Geoffrey et al has worked on the Collaborative Sensor Grid framework (Fox, Ho, Wang, Chu, & Kwan, 2008) which presents the design to increase the efficiency of data delivery and how one grid can collaborate with the other grid. The problem with grid based architecture is that that the designers concentrate on achieving efficiency in the processing by clustering the HPC based machines. The scalability, availability, collaborative ness and homogeneity are not the prime goals. Looking to the various solutions for wireless sensor networks, the most common thing which is observed is, even in the collaborative, distributed and grid based approach, they cater the need of their own organisation. John Douglas et al present an open distributed architecture for environmental risk management (Douglas, Uslander, Schimak, Esteban, & Denzer, 2008) in which they use two architectures ORCHESTRA and SANY. However it doesn't indicate how to search for independent secluded networks, which are the part of other organisation. Mires (Souto et al., 2005) is a publish/subscribe architecture for WSNs in which the sensors publish the sensed data at the cluster heads and finally to the sink node which give the data to the subscriber. The TinySIP (Krishnamurthy, 2006) architecture uses publish/subscribe and instant messaging model in which also it covers the session semantics. It offers support for multiple gateways. The most recent work done on the sensor cloud is by Mohammed Mahedi hassan et al. (2009) which proposes a cloud based framework for the sensor networks. They propose the cloud architecture and its services for WSNs. It also highlights the methodology of establishing a VO within the clouds. It also gives a event matching algorithm for event notification to the subscribers. However, this frame work does not take into account the degree of collaborative-ness of the individual WSN setups and ways to search the sensor networks. It has been fairly assumed that all the WSNs have architectures to support participation. Secondly less is talked about the way to implement security and data representation to the subscribers. In view of the above, the requirement of achieving global WSN is yet to be fulfilled.

LIMITATION OF EXISTING WSN SETUP

The limitations of the conventional setup of WSN are:

i) Inaccessibility of WSNs: Normally the WSNs are designed to serve the purpose of some limited geographical area with accessibility to limited no of users. This creates a bottleneck especially if we seek a solution to the problems concerning global environmental studies.
ii) High Infrastructure Requirement: The data generated by WSNs is quite huge. This requires the deployment of high duty storage processing and communication systems, involving high cost. This does not justify the cost benefit ratio.
iii) Non Collaborative Approach: Most of the WSNs are designed for single organisation without any scope of sharing their data with others. This restricts the participation by other WSNs having common interest.
iv) Heterogeneity: As there is no universal framework or the platform for their setup and implementation, it becomes difficult to make WSNs talk independently with each other.

COLLABORATIVE APPROACH

Present setup of existing WSN pose serious limitations for applications that need to process data on the global scale. The solution is to redesign the existing architectures to support collaborative cloud computing in WSNs.

The design should evolve as follows:

1) As a first step we need a uniform Registration process for all the WSNs, just like Domain Name System for Internet or web servers to facilitate easy searching of WSNs based on various criteria.
2) The Collaborative SOA Architecture to be followed by the WSN Setups at the deployment place.
3) Cloud Setup for Collaborating with WSNs and providing services regarding data representation and decision making to the subscribers.

The above steps in aggregation can be considered as a Sensor Collaboration Model. Our earlier work focused on step 1 and step 2, whereas this paper proposes the final step i.e., collaborative cloud for WSN. For understanding the complete model, Uniform registration process and architecture of collaborative WSN are introduced as follows.

1) *The Uniform Registration Process:* A distributed SOA based registry for sensor networks as designed by us in (Pandey & Patel, 2009) to enable all the sensor networks of the world to register themselves needs to be implemented. The registration of user WSNs requires attributes like sensor domain name, Geo-spatial Attributes, Political Attributes, Data definitions, Security Roles, host or cluster description, Gateway characteristics, mobility status key words, Sensor Network Description services and URLS etc. The corresponding schemas of WSDL and XML involved in this task are also supported. It is expected that end users or service brokers will consult the registry to search sensor networks based upon the attributes given above; the registry setup also presents front end application to sort and search the sensor networks. All sensor networks have been given the URL of their home page in the registry to be consulted for further actions. This effectively solves the problem of seclusion in the universe

of WSN Figure 1 presents the scheme to access the registry and data from WSN. This registry will be used as one of the constituent blocks in collaborative approach described in the next section.

2) *The Collaborative WSN Setup:* It is desired for the WSNs to follow an architecture and service stack which promote the collaboration of their data with other sensor networks or sensor clouds in the future (Fox, Ho, Wang, Chu, & Kwan, 2008). As a next step towards sensor clouds, a service stack was designed which consist of four layers viz Administrative, Data Layer, Application layer and Network layer (Pandey & Patel, 2009) all the layers are using XML based repositories to communicate. The Data pull and push services are working in accordance with administrative, security and pricing policies of WSN. The Service stack for the collaborative WSN Setup is given in Figure 2.

3) *Cloud computing based Services using integrated WSNs:* The final step evolves with the cloud. The cloud computing is the promising technology for the collaborative and participatory approach. The recent developments in cloud computing allows sharing of resources through virtualisation and secure means of transactions through light weight service oriented architecture. Therefore, we have incorporated it in our model to meet the goal of forming global WSN in real sense.

THE SERVICE STACK FOR CLOUD COMPUTING IN WSN

The clouds are owned and operated by cloud service providers. Therefore the main task is to design the service stack for CSP. We propose the service stack for the cloud based services to integrate inter-organisational wireless sensor networks deployed at different geographical locations. The goal of collaborative and participatory approach for WSN data processing can be achieved by this service stack.

Various layers to satisfy groups of services, which will constitute this collaborative service stack, are designed as under:

A. Management Layer
B. Enterprise Information Layer

Figure 1. Working of SOA based sensor web registry

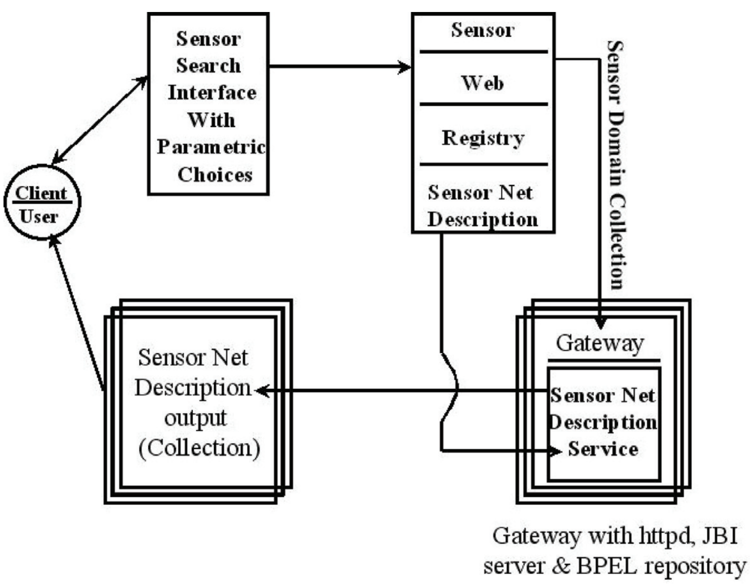
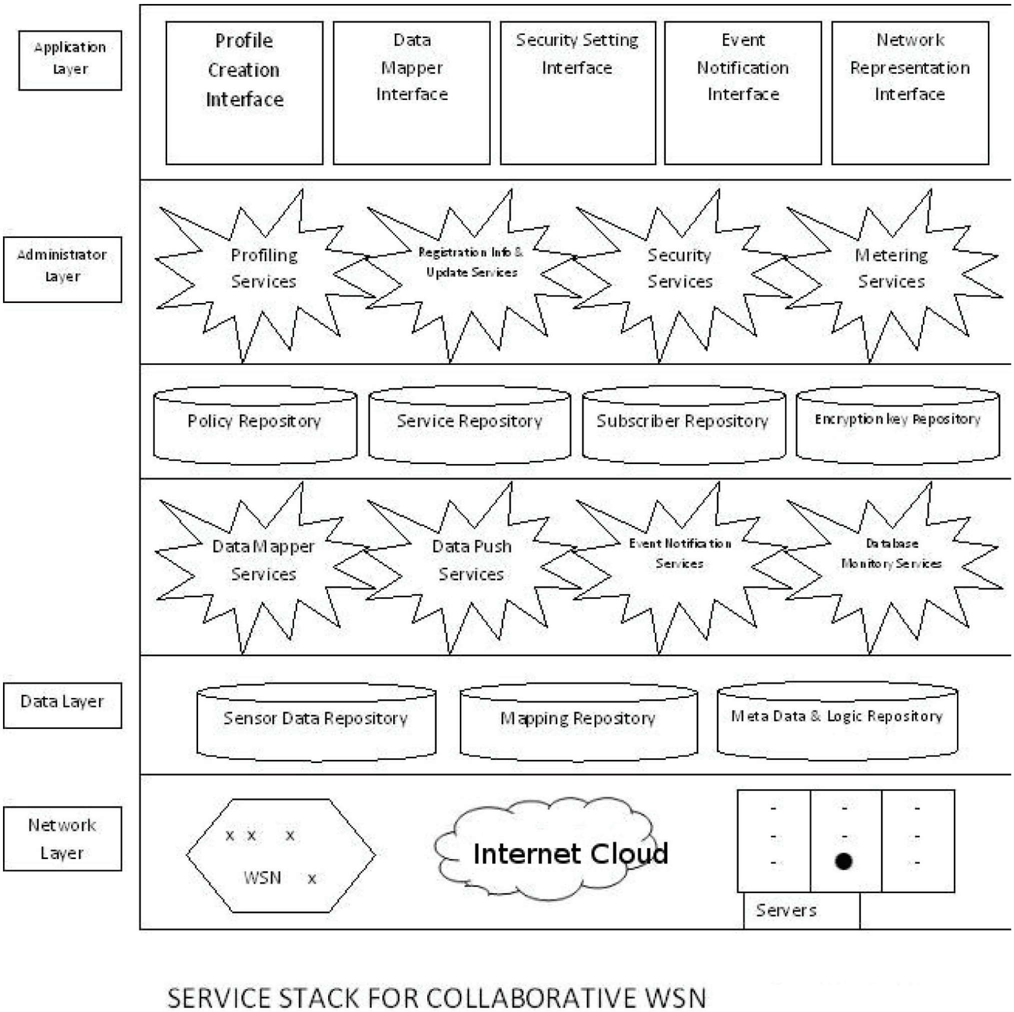


Figure 2. Collaborative WSN service stack



C. Presentation Layer
D. Virtualisation Layer
E. Communication Layer

A. Management Layer

This contains the collection of services and repositories which deal with management part of WSN Cloud setup while registering new WSN Setup of different organisations, Subscribers to the Cloud Services, managing policy repositories, security handling and configuring business value. Following services are categorised under management services

1. CWSN registration Services
2. Subscriber registration service
3. Cloud Security Service
4. Metering Services
5. Inter-cloud Connectivity and Mediator Service

6. Virtualisation Service
7. Collaborator Service

The Repositories which will leverage the above services in the administrative layer are:

i. WSN Repository: stores the info of WSNs
ii. Subscriber Repository: stores the profiles and status of subscribers
iii. Policy Repository: deals with administrative and pricing policy
iv. Web Services repository: deals with web services and their orchesterisation
    iv. Encryption Keys Repository: Stores keys and certificates from subscribers and WSNs
    v. Cloud Repository: stores the details of other linked clouds

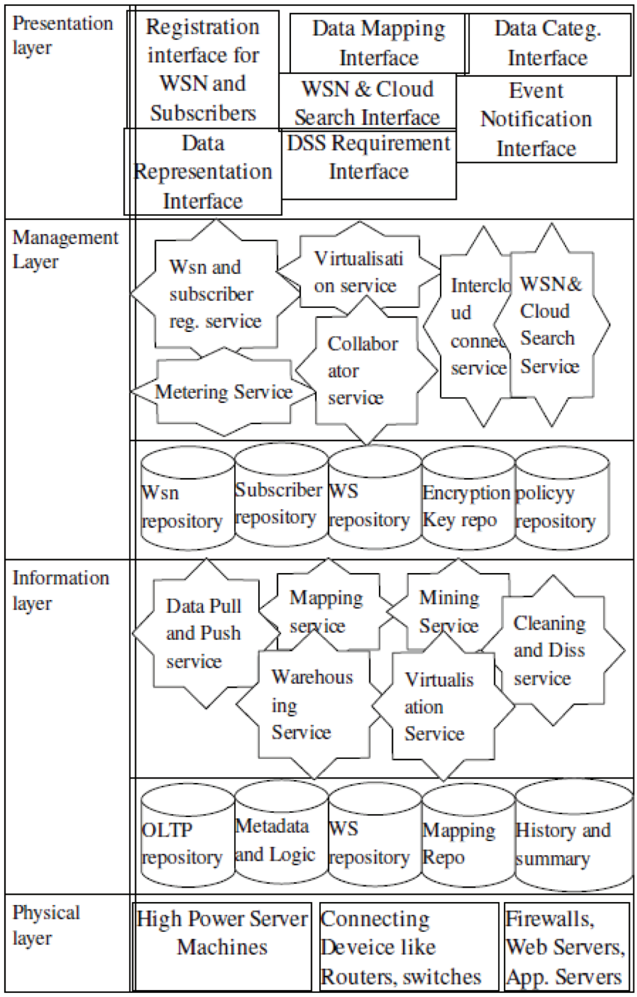
The service Stack for cloud based collaborative framework is given as under in Figure 3.

1. CWSN Registration Services: The Cloud Service Provider (CSP) is responsible for registering the Collaborative WSNs with them. This registration service must include the following fields
    i) Wsn_id: The identification of wsn
    ii) Categorisation: The category of WSN as per sensed data (environmental, defense, weather, flood etc)
    iii) Registerd_name: The registered name of the WSN in distributed registry for WSN. The registry will provide all the details regarding the WSN which includes
        a) Sensor Domain Name: The domain name of the WSN in registry
        b) IP address of the Gateway
        c) Ports at which gateway webserver and database servers are running
        d) Mobility status: static or mobile
        e) Geospatial locations: Latitude and Longitude of WSN physical deployment area.
        f) Data Type Sensed, its category, entity, unit and sensing interval
        g) Political Data: country/state/city/region etc.
        h) URL of Sensor Network Description Service
        i) Owners: name, contact detail of organisation and persons
        j) Layout/ deployment diagram
        k) Security policy on data
        l) Pricing policy

In case of WSN with non collaborative approach, they need to explicitly specify their details as mentioned above.

    iv) Authentication Parameters: user name / password etc.
    v) Public Key or Certificate
    vi) Data Push interval
    vii) Mapping their data to Cloud's real time database
2. *Subscriber Registration Service:* The subscribers are the users of the sensed data. In order to access, view, analyse the sensed data the subscribers need to register themselves with the CSP. The registration data may include
        a) Subscriber Info
    i) Subscriber id – provided by the Cloud
    ii) Subscriber Type -User, Service Broker, Cloud Provider, Federal Agency, Administrators, other users etc
    iii) Name of Agency
    iv) Address Info – address, state country PIN, phone, email etc.
    v) Representative person details – name, address, phone, email, responsibility
    vi) Authentication Parameters - like username, password, hint question etc.
        b) Data delivery Parameters
i. Data interested in – list of sensed data types
ii. WSNs of interest – choose the registered CWSN/WSNs for the data of interest
iii. Fields interested in – Show the multiple values of attributes for data collected
iv. Data Refresh Rate– hour, min, sec
v. Event interested in, Threshold value of data less than, more than or equal to threshold value

Figure 3. Service stack for cloud based collaborative framework



vi. Data Delivery Timings – Start Time & End Time

c) Services of Interest: To choose various services like data delivery, geographical mapping, data mapping, analysis, monitoring etc for the subscription

d) Metering parameters: This is the static information given to subscriber

i) Charge on basis services chosen

ii) Final Negotiated Charge

e) Confirmation Status

Role Status: Admin, UA, SB, Federal, Guest, Trial

Online Legal Agreement Document

Subscription Confirmation Status: Rejected, Awaited, Confirmed, Temporary

Public Key Download

3. *The Cloud Security Service*: This service will be required to control the access and maintain the confidentiality and authentication of information. Following tasks are to be performed by security services

a) To assign the Role to the WSNs, subscribers, users, and will decide the access level and rights to the data as decided by the policies stored in policy repository.

b) Authentication parameters for subscribers as mentioned in profile
c) Deciding data encryption policy – with symmetric keys and asymmetric PKI infrastructure
d) Key Repository and Key Distribution Methods and Policy The keys are to be distributed to the subscribers and WSNs on periodic basis

4. *Metering Service*: This service will be activating if the services to the subscriber is the paid one. To every service being used the metering service is activated. The metering service checks the charge policy decided by the cloud
5. *Inter-Cloud Connectivity and Mediator Service:* One Cloud can always connect to the other cloud to have the task of collaboration to the global level. It requires a mediator service to establish the connectivity under environment of mutual trust. This takes account of establishment of secure VPN by using a secure keying to send the ESP and AH. It can be supported by mutual certificate authentication. The Certificate Authorities are going to play a major role in it. The role of mediator service is to ensure the credentials of the participating clouds and maintain the relationship of trust.
6. *Virtualisation Service:* Virtualisation is the one of the main features of cloud computing. The virtualisation service will decide to host the data, processing power, storage to the array of machines employed at distributed locations in the control of the CSP.
7. *Collaborator Service:* This is the central player. While delivering the subscribed services to the subscriber the collaborator service maintains and looks into the repositories for WSNs, subscribers, other sensor clouds, policies, security, web services, orchestrasisation of web services, virtualizations, mediation etc. so that the user may be served, tracked and billed correctly.

B. Enterprise Information Systems Layer

The Information System at the cloud deals with the management, monitoring, storing, disseminating and processing the incoming data from various WSNs. It also deals with the queries produced by the subscribers and deliver the data after query optimization and execution. The services which will do these tasks are

1) Data Pull and Push Service
2) Database Mapping Service
3) Data Cleaning and Dissemination Service
4) Data Warehousing Service
5) Data Mining Service
6) Event Notification Service
7) Database Virtualisation Service
8) Database Monitoring Service

The repositories which are required for the functioning of the services are

i) The Real Time Data Repository: storing incoming data from various WSNs
ii) Meta Data and Logic Repository: specifying metadata structure, procedures, function, triggers etc.
iii) Mapping Repository: The data mapping details for every WSN to Cloud's transactional database
iv) Historical and Summarised Data Repository: deals with the data warehouse maintained by cloud

   1. *Data Pull and Push Service*: Depending upon the terms agreed upon by the by WSNs the cloud setup pulls the database from the WSN organisation. Similarly the current cloud can also be a data provider for other cloud, and then data push services are required.
   2. *Database Mapping Service:* The sensed data is coming from various WSNs and every WSN can have its

own data definition and dictionary. To collaborate we may need a uniform data representation. The mapping service maps the data from WSN to the cloud's transactional database. The mapping strategy is already being agreed upon by the WSN at the time of registration. This mapping involves entity to entity and attribute to attribute mapping. The mapping are stored in mapping repository which contains WSN wise mapping information in a well formatted XML
3. *Data Cleaning and Dissemination Service:* The incoming data from various sources needs to be treated and formatted before it is stored in the cloud database.
4. *Data warehouse Service:* This service helps to maintain a data warehouse for keeping historical and summarised databases depending upon various subjects of interest.
5. *Data Mining Service:* The data stored in the warehouse is useful for extracting the patterns and associations of interest. The mining service can have a insight into the warehouse and extract the information which will be useful for decision making.
6. *Event Notification Service:* This service will be continuously running in the background just to notify the subscribers for the event of interest which is pre-decided at the time of subscriber registration in a true asynchronous manner. This is stored in the subscriber profile repository.
7. *Database Virtualisation Service:* The enormous volume of the incoming data and complexities of the data retrieval from subscriber's query pushes cloud to go for virtualisation of the databases where task is handled by number of remote hosts clustered together. It is possible to use the database host of the WSNs itself and virtualize the cloud database on them. The service will take part in allocation of the data and query processing or real time basis. This way there can be a lot of cost cutting through cloud infrastructure.
8. *Database Monitoring Service:* Looking the huge volume of data generated from the sensor networks, the database monitoring service monitors the health of the database and reports if any unusual condition and triggers the corrective action too.

C. Presentation Layer

The management and the information services are often executed and run in the background. The subscriber and WSN organisations need well defined interfaces to work with. The Services can serve as a starting point to invoke the service of services in the background as orcheasterd by any Business Process Modelling Languages. The front end interfaces are as under

i) Registration Interface for WSN and Subscribers
ii) Data mapper Interface
iii) Sensor categorisation Interface
iv) Data representation Interface
v) Event Notification Interfaces
vi) Geographical Representation Interface for WSNs and other Clouds
vii) DSS Requirements and Analysis Interfaces
viii) WSN and Cloud search interface
1. *Registration interface for Subscribers and WSNs*: All the services and repositories being mentioned in management and information layer requires a visual interface to enter the data required at the time of registration
2. *Data Mapper Interface:* To provide the input to the the mapping repository in the information layer suitable interface is provided where every WSN can map their database with corresponding clouds database
3. *Sensor Categorisation Interface:* There are many number of entities which can be sensed by sensors, may

be from the environment, health, ocean, videos, chemicals etc. The WSNs may sense one or more of such entities for this they need to be classified and categorised which this interface is presenting. This will help the subscribers to search WSNs as per their interest domains. This interface is used by Cloud Administrators.

4. *Data Representation Interface*: This application provides representation of sensed data in the form of graphs, maps, contours etc. The subscribers can access this application to monitor and asses the current data and its pattern over the area. Here tools like Geographical Information Systems (GIS) can be useful for producing query on sensed data and its geographical representations.
5. *Event Notification Interface*: This interface alerts the subscribers for the event of interest by flashing text, dialogs, and alert bells. This notification invokes the web service which does the necessary action and control.
6. Geographical Representation Interface for WSNs and other Clouds

This interface shows the layout of WSNs, inter WSNs and other clouds on the map. It helps the subscribers to see the WSNs in their vicinity or the place of interest.

7. *DSS Requirements Interface:* This interface deals with the decision making process of subscribers and WSN administrators. The interface will allow to define the inputs for pattern recognition, unusual behaviours, associativity with other data etc. They can also specify the graphical representation of historical and summarised data in form of different types of graphs.
8. *WSN and Cloud Search Interface:* An interface must be provided for searching the WSN based on criteria like data sensed, latitude and longitude, region, area, country, state, district, mobility etc.

D. Communication Layer

All the above layers are building upon following network infrastructure

Interconnected Physical Devices: These include Virtualised Network Machines which hosts the file systems, database, web service assemblies, repositories mentioned above, security key rings, certificates, Geographical information system, Images, videos and additional tools.

ii) Network Connectivity from Collaborative cloud Setup for WSNs and Subscribers: This connectivity is based on the current internet infrastructure or a Virtual Private Network with LAN and WAN connectivity
iii) Network Servers: This includes high power machines as Database Servers, Application Server, Web Server, Mail Server etc. All the servers must be suitably connected with the internet and should be equipped with firewall policy to thwart network attacks. If the load is more a cluster of the servers can be formed to take the load. A System Administrator and its staff must carefully plan the scheduling and backup
iv) Inter-connecting devices: High level routers, switches gateways, firewalls and continuous power availability will facilitate the smooth connectivity and speed.

SOA APPROACH TO DEAL WITH HETEROGENITY

All the services mentioned above need to collaborate with invocation from outside and inter-invocation as well. We strongly recommend to use the present state of art Service Oriented architecture (SOA) which can be very well implemented in open source language like Java which in its recent development has provided a complete framework with help of

Java Business Integration, Business process execution language to orchestrate the services and various service engines to connect from outside. It enables to connect through http, soap and Messaging infrastructure. The application interfaces can be suitably developed in presentation layer using servlets or JSP. A Java enabled Application Server will allow to deploy all service based applications (Figure 4).

THE FUNCTIONING OF THE COLLABORATIVE CLOUD WSN SET UP

The major entities in the working of Collaborative cloud setup are the WSN/CWSNs, Subscriber and the Cloud. All the entities deal separately with the sensor cloud. Their interactions are explained as below

i) *Interaction between WSNs and Cloud:* The WSNs which wish to deliver their service on cloud for the purpose of business or community service need to register themselves through the interfaces provided by the cloud which systematically categorise the WSN, map the sensed data, establishes security and trust amongst them. It uses several services and repositories which are arranged in a multi-layered stack as mentioned above. Once the registration is completed the data push and pull services will be activated and thereby other services will become active to clean, format and store in the transactional database and thereby maintain in a data warehouse to keep historical data. The Figure 5 shows the inter-working of WSNs and Cloud.

ii) *Interaction between Subscribers and Cloud:* The cloud service provider may promote its cloud as business model or community service. This will require the subscription from interested users. The adequate interfaces are provided by the cloud to the subscribers and their profile, choices and preferences are activated by the underlying services and stored in repositories. The subscriber may be billed by the time, volume of data consumed etc. The cloud will ensure the confidentiality and integrity of the data being delivered or accessed. The mediator services need to play an important role in establishing the relationship of trust. Figure 6 shows the interaction.

iii) *Interaction among clouds:* This interaction is opening all the avenues of data

Figure 4. A SOA- JBI architecture (adapted from cwiki.apache.org)

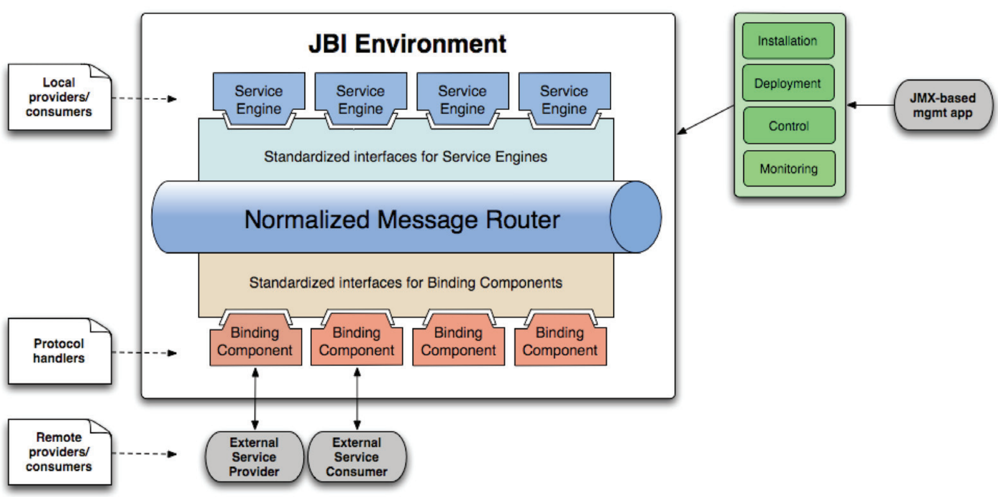


Figure 5. Interaction of WSN and cloud

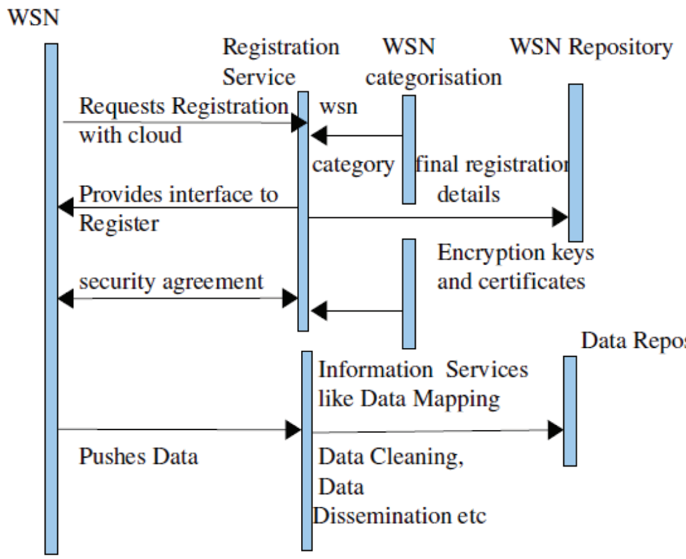
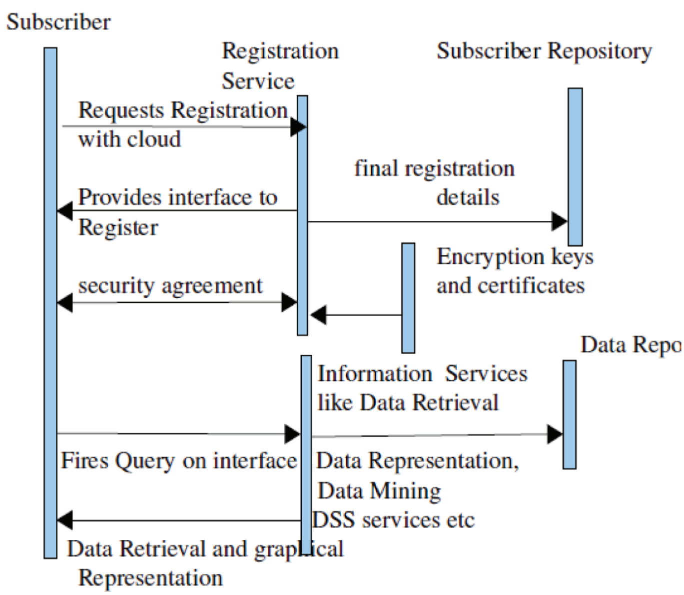
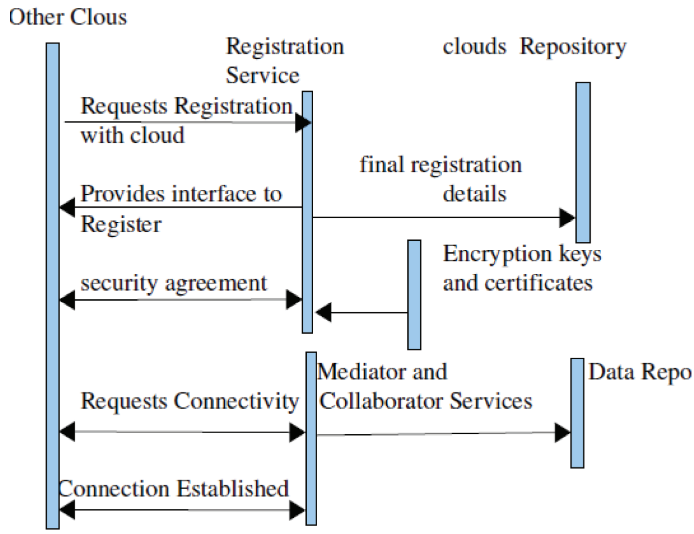


Figure 6. Interaction of subscriber and cloud



sharing and collaboration on global basis. The cloud initiated the interaction with a joining request in a predefined format to the other cloud via a mediator service. The mediator will forward the request to the target cloud in the format understandable by the target cloud. The request must carry the credentials and the profile of the

Figure 7. Interaction of cloud to other clouds



requesting cloud accompanying with the certificate signed by renowned certificate authority. The request must also carry the description of policies. The collaborator agents at both the ends help in negotiating the connection. The collaborator agent at target cloud will respond with its own policies and criteria through mediator services. The initiator collaborator agent will make decision to accept or reject the connection request. Once the connection request is accepted then they can map the database mutually and the pull and push services at both the ends will initiate the data submission and retrieval process this will increase the subscriber base for both the interacting clouds (Figure7).

CONCLUSION

To support the cause of global availability of the sensed data from wireless sensor networks, the c loud computing offers a promising role. A framework with service stack is presented here for the cloud based services which will offer a central point of connectivity with the WSNs deployed anywhere on earth. Whole Stack consists of four layers Viz management, Information, presentation and physical. Every service and repositories in the layers are described. Finally the working of cloud is explained by the interaction of WSN, subscriber and cloud-to-cloud. This architecture can percolate through the barriers of physical and political boundaries and provides a solution to the global problem.

REFERENCES

Botts, M., Percivall, G., Reed, C., & Davidson, J. (2006, July 19). *OGC Sensor Web Enablement: Overview and High Level Architecture*. Paper presented at the Open Geospatial Consortium (OGC 06-050r2 v2).

Douglas, J., Usländer, T., Schimak, G., Esteban, J. F., & Denzer, R. (2008). An Open Distributed Architecture for Sensor Networks for Risk Management. *Sensors (Basel, Switzerland)*, *8*, 1755–1773. doi:10.3390/s8031755

Fox, G., Ho, A., Wang, R., Chu, E., & Kwan, I. (2008). A Collaborative Sensor Grids Framework. In *Proceedings of Collaborative Technologies and Systems*, Irvine, CA (pp. 29-38).

Gaynor, M., Moulton, S. L., Welsh, M., LaCombe, E., Rowan, A., & Wynne, J. (2004, July-August). Integrating Wireless Sensor Networks with the Grid. *IEEE Internet Computing*, *8*(4), 32–39. doi:10.1109/MIC.2004.18

Hassan, M. M., Song, B., & Huh, E. N. (2009, January 15-16). *A Framework of Sensor - Cloud Integration Opportunities and Challenges*. Paper presented in ICUIMC-09Suwon, South Korea.

Krishnamurthy, S. (2006). TinySIP: Providing Seamless Access to Sensorbased Services. In *Proceedings of the 1st International Workshop on Advances in Sensor Networks (IWASN)*.

Pandey, K., & Patel, S. V. (2009, July). Design of SOA based Sensor Web Registry. In *Proceedings of CICSyn IEEE conference*, Indore, India.

Pandey, K., & Patel, S. V. (2009). Design of SOA based Service Stack for Collaborative Wireless Sensor Network Submitted. In *Proceedings of WSCN 2009, IEEE Conference*, Allahabad, India.

Souto, E. (2005). Mires: A publish/subscribe middleware for sensor networks. *ACM Personal and Ubiquitous Computing*, *10*(1), 37–44. doi:10.1007/s00779-005-0038-3

Tham, C.-K., & Buyya, R. (2005, June 24). *SensorGrid: Integrating Sensor Networks and Grid Computing* (Tech. Rep. No. GRIDS-TR-2005-10). Melbourne, Australia: Grid Computing and Distributed Systems Laboratory, University of Melbourne.

International Journal of Grid and High Performance Computing

An official publication of the Information Resources Management Association

Mission

The primary objective of IJGHPC is to provide an international forum for the dissemination and development of theory and practice in grid and high performance computing. IJGHPC publishes refereed and original research papers and welcomes contributions on current trends, new issues, tools, societal impact, and directions for future research in the areas of grid and high performance computing. The journal is targeted at both academic researchers and practicing IT professionals.

Subscription Information

IJGHPC is published quarterly: January-March; April-June; July-September; October-December by IGI Global. Full subscription information may be found at www.igi-global.com/IJGHPC. The journal is available in print and electronic formats.

Institutions may also purchase a site license providing access to the full IGI Global journal collection featuring more than 100 topical journals in information/computer science and technology applied to business & public administration, engineering, education, medical & healthcare, and social science. For information visit www.infosci-journals.com or contact IGI at eresources@igi-global.com.

Copies of Articles

Copies of individual articles are available for purchase at InfoSci-On-Demand (www.infosci-on-demand.com), an aggregated database consisting of more than 31,000 research articles written by prominent experts and scholars in the field of information/computer science and technology applied to business & public administration, engineering, education, medical & healthcare, and social science.

TOC Alerts

Please go to www.igi-global.com/IJGHPC and click on the "TOC Alert" tab on the left menu if you want to be emailed the table of contents of each issue of this journal as it is released.

Copyright

Correspondence and questions:

Editorial:
Emmanuel Udoh, Editor-in-Chief
IJGHPC
Tel: 260 450 3845
Email: udohe123@yahoo.com

Subscriber Info:
IGI Global
Customer Service
701 E Chocolate Avenue
Hershey PA 17033-1240, USA
Tel: 717/533-8845 x100
E-mail: cust@igi-global.com

IJGHPC is currently listed or indexed in: Cabell's; GetCited; INSPEC; Media Finder; Standard Periodical's Directory; Ulrich's Periodical Directory