

Building a case for FIPA Compliant Multiagent Approaches for Wireless Sensor Networks*

Khalid Nawaz
Databases and Distributed Systems Group
Department of Computer Science
TU Darmstadt, Germany
khalid[at]dvs.informatik.tu-darmstadt.de

Alejandro P. Buchmann
Databases and Distributed Systems Group
Department of Computer Science
TU Darmstadt, Germany
buchmann[at]dvs.informatik.tu-darmstadt.de

ABSTRACT

There have been some efforts to apply agent based approaches in Wireless Sensor Networks (WSNs) for different application scenarios in recent past. These efforts have met with varying levels of success so far. However, the efforts to assess the suitability of *Foundation for Intelligent Physical Agents* (FIPA) compliant multiagent solutions for WSNs are almost non-existent. In this paper we present a scenario from an industrial domain and assess the suitability of a FIPA compliant multiagent based solution for it. The scenario is based on a WSN deployment in an underground mine where it could be put to use for several purposes like hazardous gas plume detection and tracking, mine lighting control, Proximate Environment Monitoring (PEM), structural health monitoring of mine structure, to name a few. However, we only consider the PEM usage scenario in this paper.

Categories and Subject Descriptors

C.2.1 [Computer Communication Network]: Network Architecture and Design; C.3 [Computer Systems Organization]: Special-Purpose and Application-Based Systems

General Terms

FIPA Compliance

Keywords

Multiagent Systems, Wireless Sensor Networks, Middleware, FIPA

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are composed of tiny nodes that have sensing, processing, and communication ca-

*This research is supported in part by HEC Pakistan, DAAD and DFG Graduiertenkolleg GRK 1362 at TU Darmstadt.

pabilities. Their use is increasing in many spheres of human activity including, but not limited to, environmental monitoring, health-care, natural habitat monitoring, agriculture, military, structural health monitoring, pervasive computing, ambient intelligence etc. One very useful application of WSNs could be for monitoring mine environments, where they could be used as a warning system for the prevention of mining accidents. Each year human lives and machinery are lost or damaged as a result of accidents in mines around the world. Many of these lives could be saved if we make use of modern technology like WSNs as a warning system for the detection of hazardous and explosive gases like carbon monoxide and methane in underground mines.

Having said that, it is also very important to choose an appropriate methodology for WSN programming that meets all the requirements posed by such a scenario. Aspects like, how to efficiently distribute commands in the network and how to collect the data of interest back at the base station, which could be fixed or mobile, must be considered. Therefore, first of all, we try to establish what sort of requirements are posed by the underground mining scenario and then we look at the possible approaches to deal with these requirements.

In most industrial setups, it is one of the basic requirements, imposed by the industrial safety regulations, that the environment in close proximity to the workers should be made completely safe for them. Normally underground mining environments have many potential dangers for the miners due to the presence of hazardous gases like methane, carbon monoxide, carbon dioxide etc. Therefore, mine safety regimes have to devise ways to ensure miners' safety by providing systems for countering all potential threats.

The threats in underground mines could be effectively countered if there is an effective mechanism to identify them first. Additionally, there is the need for monitoring other environmental parameters, such as temperature. The deployed infrastructure (WSNs) should be able to provide that information on a just-in-time (JIT) basis. We call this requirement Proximate Environment Monitoring (PEM) and depict it in Figure 1. It is evident from the Figure 1 that only the nodes in close proximity to the miners take part in processing their request for environmental monitoring around them. Therefore, PEM tasks should only be disseminated to the proximate nodes of the miners. This proximate set of nodes would keep on changing as the team of miners (workers) moves through a mine. Therefore, any strategy that we use for providing for this PEM requirement should be flexi-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SOMITAS 2008, February 14, Quebec, Canada
Copyright © 2008 ICST 978-963-9799-16-5
DOI 10.4108/ICST.AMBISYS2008.2906

ble enough to tackle this important constraint.

In the next section we present three different strategies

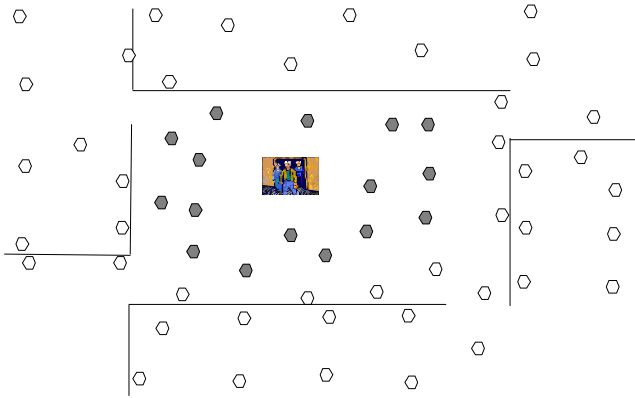


Figure 1: Proximate Environment Monitoring using a WSN deployment in an underground mine. Only the shaded nodes take part in PEM process.

to provide for PEM requirement in our considered scenario. Our goal is to determine the effectiveness of Foundation for Intelligent Physical Agents (FIPA) [4] compliant multiagent approach with mobile base stations in relation to the other more traditional approaches.

2. CASE FOR FIPA AGENTS IN WSNS

In our example scenario, presented in the last section, we introduced the requirement of PEM in the mining scenario. This requirement dictates that any system, deployed inside a mine, should provide for a way for the miners to know environmental conditions around them. This is a very basic worker safety issue that is relevant in many other industrial setups, such as chemical processing plants where chemical spills might result in the release of toxic fumes that might prove dangerous for the workers present in the vicinity. There could be different approaches to provide for the PEM requirement using WSNs. Here, we present three approaches and compare them for their efficiency by constructing a mathematical model in Section 3.

2.1 Fixed Base Station Approach

In this traditional approach, there is normally one (sometimes more) fixed base station(s) that is used to interact with a WSN. In order to extract data from the network, commands are issued using the base station. These commands are either flooded in the whole network or routed to appropriate nodes in a multihop fashion depending upon the network topology and routing protocol used. After receiving a command, nodes in the WSN act accordingly. The results generated are routed back to the base station in a multihop fashion.

There are some important issues to consider, if this approach were to be used in the mining scenario. First, the base station should be somehow made aware of the exact whereabouts of the team of miners. Similarly, there should be a way, for the team of miners, to issue requests to the WSN

through the base station. Therefore, existence of a direct communication link between the team of miners and the base station is required in this approach. Only then there could be a meaningful implementation of the PEM requirement. However, existence of such a communication link in an underground mine is quite impractical. On top of that, even if there is such a communication link, the delay experienced in the two way communication would be intolerable. Additionally, as the team of miners moves through a mine, set of proximate sensors would keep on changing. Each time new proximate set of sensors is encountered, the team of miners would have to issue a new request to the base station that would in turn be injected into the WSN by it. The results will be routed back to the base station through the network that will in turn communicate them to the team of miners through some wireless link other than the deployed WSN.

Another important aspect of this communication between proximate sensors and the base station is that it is multihop and involves intermediate nodes that simply act as relay. Therefore, the nodes closer to the base station process more messages on average than the ones farther from it. This results in unequal energy consumption in the network, with nodes closer to the base station depleting their battery power and dying earlier than the nodes farther from it, thus shortening the life of the network. Additionally, this technique is more prone to communication delays, since proximate sensors report their data to the base station which in turn informs the team of miners about the proximate environmental conditions. Due to these problems, providing proximate environmental information to the miners on just-in-time basis is very hard to achieve using fixed base station approach.

2.2 FIPA Compliant Multiagent Approach

As opposed to the fixed base station approach, a multiagent approach with mobile base stations offers a much more flexible solution to the problems mentioned in the PEM scenario. Agent based systems are similar to mobile code systems in some respects, but there are some differences too. For instance, when agents migrate from one node to the other in a WSN, they take their state, data and code during migrations. Agents are also autonomous in their decision making and thus don't require to consult the user node (base station or sink) while making a migration or any other decision. This makes them a very useful choice for scenarios where we want WSN applications to do in-network decision making.

There are a number of other advantages of the agent paradigm cited in literature that it offers to the resource constrained environment of WSNs:

Energy Conservation: With features like in-network data aggregation multiagent approach makes intelligent use of the limited battery power of nodes.

Suits well to the disconnected Nature of WSNs: Communication in WSNs is very unreliable, to say the least. Ensuring perfect connectivity each time a node wants to send a message to some other node in the network is not possible. Network partitioning in WSNs is also a commonly occurring problem, since some of the nodes could die after running low on battery, thus making the network disconnected. Multiagent approach with mobile base stations suits well to this disconnected nature of WSNs, since the user device (i.e. a

mobile base station) used to interact with the network is only required to remain connected while either injecting or receiving agents. Additionally, there is no such requirement that each node in the network should be transitively connected to the base station, as is the case in fixed base station approach.

Fault Tolerance: While navigating a WSN, agents can avoid visiting those nodes that are running low on battery. This feature of multiagent approach ensures fault tolerance, since agents visiting low battery nodes can possibly die with the nodes.

Retasking WSNs: The process of retasking an existing WSN with approaches like Maté and deluge[6] that claim to address retasking issues in WSNs is not that efficient and flexible. Whereas multiagent approach offers a more flexible, cost effective and easy way of retasking. These approaches retask an existing network statically, i.e. user defines the set of nodes that needs to be retasked. Conversely, in multiagent approach retasking could be done in a more proactive way. One simply needs to inject relevant agents into an existing WSN and they can proactively migrate to the target nodes. Therefore, in a way the agents themselves define the way they get distributed in a WSN.

Sharing WSNs for Multiple Applications: Since each node in a WSN can host multiple agents depending upon its memory and processing capabilities, the same WSN could be used for multiple applications at the same time [5]. This feature could be very significant in the mining scenario in general, since it would be possible to support multiple applications like controlling mine lighting system, monitoring structural stability and monitoring the mine environment for the presence of any hazardous substances at the same time with the same set of nodes.

In-Network Decision Making: Autonomy is another important feature of agents. This enables them to make decisions on their own without consulting the base station each time. This feature becomes more significant in situations where in-network actuation is desired. Agents can inform the nearest actuator of the change in environment (event of interest) without going through the base station.

2.2.1 What is FIPA Compliance?

For agent based systems, there is a standards defining body known as *FIPA*. It has defined standards that agent based systems should follow in order to get the status of full compliance. These standards are mostly targeted at agent systems for resource rich environments, but, to our understanding, some of them can be applied to the resource constrained environments like WSNs too. We, therefore, consider a system that complies to the basic set of standards that are suitable for resource constrained environments. This basic set of FIPA standards mandates defining an Agent Management System (AMS), Directory Facilitator (DF) and Agent Communication Language (ACL) for agent based systems. The major task of an AMS is to provide services that control the life cycle of agents in an agent based system. The major task of a DF is to provide a central or distributed service for agents to find out information about the services offered by other agents in an agent based system. Using this standard in agent based approaches in WSNs, we believe, communication between agents could be made more reliable. Agents can query DF to find out information about the services offered by other agents in an

agent based system. Then if some agent is interested in communicating with some other agent, it can query the AMS to find out the whereabouts of other agent in the system. The third standard (ACL) defines specifications for agents to communicate with each other in an agent based system. This standard promotes interoperability between different agent based systems belonging to the same domain. Therefore, in the long run, following this standard would prove very useful, since different agent based systems, in the domain of WSNs, could be made inter-operable.

2.2.2 A Representative Example of a FIPA Compliant Multiagent Approach

There have been a number of efforts to provide agent based frameworks for WSNs but so far none of them is FIPA compliant. We are working towards building a system that complies to the aforementioned three basic standards defined by FIPA. Describing the complete architecture of our system is beyond the scope of this paper, but we provide a brief overview of it, to illustrate its effectiveness in the scenario of PEM in a mine.

We assume a cluster based approach where the WSN is

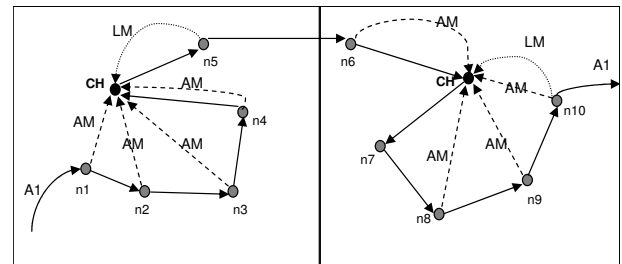


Figure 2: Itinerary of a single agent through a WSN performing Proximate Environment Monitoring.

divided into clusters each managed by a cluster head. AMS and DF agents run on the cluster heads. On each node, there runs a controller agent that reports the arrival of application agents (the agents that carry out user’s tasks like average temperature determination) on that node to the cluster head’s AMS and/or DF.

In Figure 2, we show the path of a mobile application agent (A1) injected in the network in order to perform PEM task. It visits different nodes and aggregates the sensor readings finally coming back to the user node (which is a handheld gadget acting as a mobile base station). When agent arrives at a node, an Arrival Message (AM) is sent to the cluster head’s AMS notifying it of the arrival of the agent. If the agent visits a node which happens to be a Cluster Head (CH) itself, then the next destination node on the itinerary of the agent, after cluster head, doesn’t need to send an AM to the cluster head. It rather sends a Leave Message (LM) when the agent leaves it and goes to a node that is part of a neighboring cluster. These messages that are sent to the cluster head help track the movement of an agent through

the network. If some agent in a cluster needs to communicate with some other agent in the cluster, it can query the cluster head's DF agent. This makes agent communication more reliable than in some of the other agent based approaches like Agilla [5] that rely on a tuplespace paradigm [9] for agent communication. There are some other agent based approaches like TeenyLime [2] and TinyLime [3] that also make use of tuplespace paradigm. TinyLime is also proposed to work in a mobile data collection setting similar to the one we are considering. However, it doesn't support multihop communication nor does it consider a cluster based approach. Additionally, none of these agent based middleware is FIPA compliant.

In traditional distributed systems, there are several successful implementations of FIPA compliant multiagent middleware like JADE [1], AGLETS [7]. JADE (Java Agent Development Framework) is the most well known FIPA compliant open source middleware platform in traditional distributed systems. In the domain of WSNs, so far no agent based middleware is FIPA compliant though.

2.3 Database Oriented Approaches

Approaches like Cougar [10] and TinyDB [8] view a WSN as a distributed database system. These approaches are user-centric and provide similar abstractions as a relational database system provides. Users can issue SQL like queries to extract their data of interest from a WSN in much the same way they would do in a traditional relational database system. The database oriented approaches normally build a routing tree to distribute the user's query in the sensor network. For aggregating data and efficient distribution of queries in the network, these approaches build different routing trees for different attributes. For instance, in the mining scenario, one such tree could be built using the attribute concentration of a gas. Another could be built using the attribute temperature. Routing tree formation and maintenance does involve overhead of communication costs that we evaluate in Section 3.

After building a routing tree, user's query is disseminated in the network. Once the query reaches all the nodes or all those nodes that might have data of interest requested in the query, the result is routed back towards the root of the tree which normally is a base station. One such routing tree is shown in Figure 3 with bold lines representing edges of the tree and small circles the nodes. Bold-line arrows represent the Tree Formation Requests (TFRs) that each node in the network broadcasts in order to form a routing tree starting from the root node. The dotted-line arrows represent the Parent Selection Message (PSM) that each child node sends to its selected parent. This parent selection can be done based upon different criteria. One such criterion could be physical distance between the nodes with priority given to the closer nodes. In order to form a routing tree, each node in the network first broadcasts a TFR and while selecting a parent, each node, except for the root, sends a PSM.

3. MATHEMATICAL ANALYSIS OF PEM REQUIREMENT

In the mathematical analysis part, we use a metric *average number of messages transferred per node* to compare the approaches discussed in the previous section. This metric provides a good idea of the average message processing

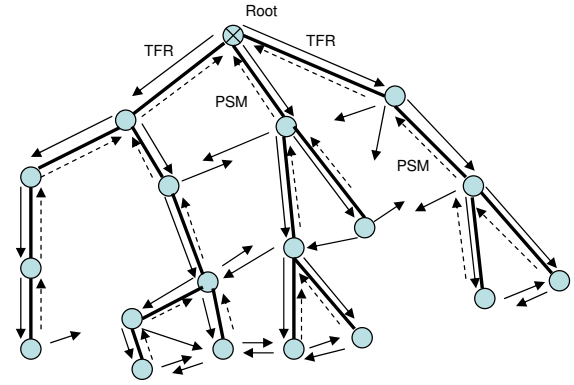


Figure 3: Routing tree formation in a typical database oriented approach in WSNs.

load on each of the nodes in the network. We don't calculate average message processing load on nodes present in different parts of the network though. Nonetheless, the same metric could be easily applied to do that.

We consider a very simple example to compare the approaches discussed in the last section. The benchmark example stays the same, i.e. PEM in an underground mine, with only a slight modification. In order to simplify the mathematical analysis, we consider a grid shaped network as shown in Figure 4 rather than a random distribution of nodes. We can assume that this grid shaped network is around the team of miners that is passing through a passageway in a mine and is interested in knowing proximate environmental conditions. We also make a slight modification in the fixed base station approach to make it more competitive with the multiagent and database oriented approaches with mobile base stations. We assume that the fixed base station is at the center of the grid-shaped network and the team of miners wants to know the proximate environmental conditions around the base station. This would be seldom true in a real WSN deployment though. Nonetheless, this assumption makes the fixed base station approach effectively a mobile base station one.

In the grid shaped network of Figure 4, the size of the network is $p \times q$, where 'p' is the number of rows and 'q' is the number of columns.

The total number of nodes in this grid shaped network is $p \times q$. For the sake of simplicity, let us assume that number of rows and columns is equal. Furthermore, we also assume multihop communication between the nodes and that each node communicates only with its one-hop neighbors.

In Figure 4, the nodes are shown to be part of different square shaped layers around the base station with the inner most layer containing 4 nodes and the outer most layer containing 28 nodes. The nodes that form the inner most layer around the base station can send their messages to the base station directly traveling just one-hop. The messages coming from the nodes in the outer layers will travel more hops to reach the base station. Let us try to compute mathematically and see the actual communication load on the nodes of this network under different techniques described in the previous section.

Number of nodes in the successive layers starting from the

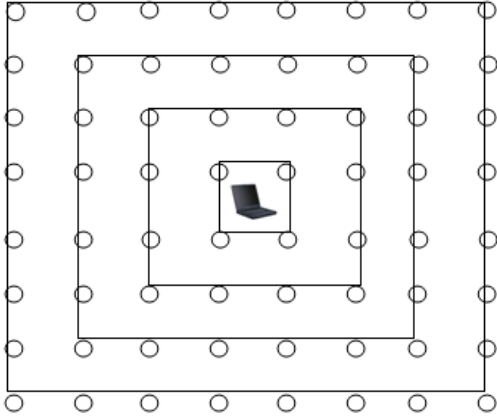


Figure 4: A hypothetical grid-shaped WSN with base station placed at the center in order to make it convenient for all the nodes to communicate with it in a multihop fashion and to make it comparable to PEM requirement.

inner most layer is 4, 12, 20 and 28 respectively. Since we are assuming multihop communication, a message originating at a node in layer ‘n’ travels ‘n’ hops to get to the base station. We count each hop as a message transfer. Therefore, number of messages transferred when each node in the network sends one message to the base station could be calculated as follows.

$$(1 \times 4) + (2 \times 12) + (3 \times 20) + \dots + 4n(2n - 1)$$

In general, the total number of message transfers in the network when each node at layer ‘n’ sends one message to the base station are given by the following equation.

$$= 4n(2n - 1) \quad (1)$$

Number of nodes in the ‘nth’ layer is $4(2n - 1)$. In order to find the total number of nodes in the network, we apply summation to it, i.e. $\sum 4(2n - 1)$.

Simplifying this summation, we get the total number of nodes in the network (N_T).

$$N_T = 4n^2 \quad (2)$$

3.1 Analysis of Fixed Base Station Approach

We assume that the team of miners has sent a request through the base station to know average methane gas concentration around them. This request is disseminated in the network by the base station. In response to this request, all nodes report their readings back to the base station in a multihop fashion. In order to make things simple, we first look at the request dissemination process.

Request Message Dissemination: In order to disseminate the request in the network, a command message is broadcast by the base station to all its one-hop neighbors. The nodes receiving this message broadcast it further until the message gets to all the nodes in the network of Figure 4. It is reasonable to assume that the nodes receiving duplicate command messages don’t broadcast all of them. Each node checks if it has already received a command message before broadcasting it further. Since each node in the network broadcasts a command message once to all its one-hop neighbors, the number of messages broadcast for disseminating the command in the network is equal to the number of nodes in the

network, i.e. $4n^2$.

Result Reporting to the Base Station: Once each node gets the command message, it sends back the result in a multihop fashion to the base station. We assume that each node doesn’t just broadcast result message, rather it communicates the result message to the node from which it had received the command message initially. We further assume that the destination node is closer to the base station than the node sending the result message. Simply stating, we assume that the result messages take the shortest route to get to the base station. We make a similar assumption in the multiagent approach as well in order to make a fair comparison of these approaches. When each node in the network of Figure 4 sends back its result to the base station in a multihop fashion, the total number of messages transferred by each node can be calculated as follows.

Applying summation to Equation 1, we calculate total number of messages transferred by all the nodes in the network while reporting the result back to the base station.

Total number of messages transferred by all the nodes in the network = $\sum_{k=1}^n 4k(2k - 1)$

After simplifying the above summation,

Total number of messages transferred by all the nodes while sending the result to the base station:

$$= \frac{2n(n + 1)(4n - 1)}{3} \quad (3)$$

Combining the results of request dissemination and result reporting to the base station, we calculate the total number of messages sent in order to carry out the PEM request of miners.

$$= \frac{2n(n + 1)(4n - 1)}{3} + 4n^2 \quad (4)$$

Using results of Equation 2 and Equation 4, we can calculate average number of messages transferred per node while carrying out the PEM request of the miners.

$$= \frac{(2n(n + 1)(4n - 1)/3)}{4n^2} + \frac{4n^2}{4n^2} = \frac{(n + 1)(4n - 1)}{6n} + 1 \quad (5)$$

3.2 Analysis of Database Oriented Approaches

Now let us see how does the database oriented approaches fare in carrying out the task of PEM in the given scenario. As it has been mentioned in Section 2.3 that a routing tree, based on some attributes, is normally built in database approaches first and then users’ queries are disseminated in the network using this tree. Therefore, we will first see how many messages do nodes exchange in order to form such a routing tree. There is one important aspect of PEM requirement that distinguishes the database oriented approaches with mobile base stations from database oriented approaches with fixed base stations in WSNs. In case of database oriented approaches with fixed base station, the routing trees that are built from the fixed base station are reused because queries are injected from the same base station each time. While with mobile base station approach, the queries are injected by the users while on the move using their handheld devices. Therefore, with each query from a new location, a new routing tree needs to be built and only then that query could be serviced by the network.

Let us now calculate the number of messages that needs to be exchanged in order to form a routing tree and to disseminate the user’s query in the network. It should be noted

that all the nodes in the network broadcast the TFR, but the parent selection message is sent by all nodes except for the root node. Here, we assume that while broadcasting TFR in the network, each node broadcasts it just once to its one-hop neighbors and duplicate TFRs are dropped by the nodes.

Messages Exchanged for Routing Tree Formation (M_{RTF}) = Tree Formation Requests (M_{TFR}) + Parent Selection Messages (M_{PS})

$$M_{RTF} = M_{TFR} + M_{PS} = (4n^2 + 4n^2 - 1) = 8n^2 - 1$$

The messages exchanged when the user (team of miners) injects a query in the network can be calculated as follows.

Messages Exchanged for servicing user's query (M_{UQM}) = User Query Propagation in the Network (M_{UQP}) + Routing back the Result (M_{RR})

$$\begin{aligned} M_{UQM} &= M_{UQP} + M_{RR} \\ &= (4n^2 - \text{No. of Leaf Nodes}) + (4n^2 - \text{Root Node}) \\ &= (4n^2 - 4(2n - 1)) + (4n^2 - 1) \\ &= 8n^2 - 8n + 3 \end{aligned}$$

Total number of messages exchanged in the database oriented approach for servicing a PEM request comes out to be:

= Messages Exchanged for Routing Tree Formation + Messages Exchanged for servicing user's query

$$= 8n^2 - 1 + 8n^2 - 8n + 3 = 16n^2 - 8n + 2 = 2(8n^2 - 4n + 1) \quad (6)$$

Average number of messages exchanged in database oriented approach, in order to carry out one PEM request of the user, can be calculated as follows.

$$= \frac{2(8n^2 - 4n + 1)}{4n^2} = \frac{8n^2 - 4n + 1}{2n^2} \quad (7)$$

3.3 Analysis of Multiagent Approach

In Section 2.2.2, we described some details of a FIPA compliant multiagent approach to address the PEM requirement in an underground mining scenario. Since it is a cluster based approach, we consider the cost involved in cluster formation first and then the communication cost for carrying out the PEM task.

Cluster Formation: There could be very complex approaches to forming clusters in a WSN, but we consider a very simple one. We assume that the cluster heads are chosen at the time of deployment. Then they start the process of cluster formation by broadcasting a Cluster Formation Request (CFR) to their one-hop neighbors. Each neighbor, upon receiving the CFR, broadcasts it to its one-hop neighbors. This process continues until the time-to-live (TTL) of the CFR expires. In the multiagent approach, we assume small size clusters to reduce the communication costs. Therefore, TTL for CFR could be restricted to very small values. For the sake of our example, we assume that every sixteenth node is a cluster head and that each node, other than cluster heads, broadcasts the CFR once to its one-hop neighbors. We make a similar assumption that we made in database and fixed base station approaches that each node, before broadcasting a CFR, checks if it has already done so. If it has already broadcast the same request received from some other neighbor, then it drops the CFR message otherwise it broadcasts it further to all its one-hop neighbors. Nodes join a cluster by responding the CFR with a Cluster Joining Request (CJR). CJR is sent to the cluster head

by the nodes interested in becoming cluster members of the node whose CFR they had received. Nodes send CJR only to that neighbor from which they had received CFR earlier. Each onehop neighbor of a cluster head, after broadcasting the CFR further, waits for CJRs of its neighbors (until the expiry of some predefined timeout). Once it receives CJRs from its neighbors (or the timeout expires), it sends them, piggybacking its own CJR request, in a single message to the cluster head. Using this information, we can calculate the number of messages exchanged for cluster formation as follows.

$$\begin{aligned} \text{Messages exchanged for cluster formation} &= \\ &= \text{Number of cluster heads} \times 1 + \text{Number of non-cluster head nodes} \times 1 + \text{CJRs sent by non-cluster head nodes} \\ &= \left(\frac{1}{16} * 4n^2\right) + \left((4n^2 - \frac{4n^2}{16}) * 1\right) + \left((4n^2 - \frac{4n^2}{16}) * 1\right) \\ &= \frac{31n^2}{4} = 7.75n^2 \end{aligned} \quad (8)$$

Using Equation 2 and Equation 8, we can calculate the average number of messages transferred per node for cluster formation in multiagent approach as follows.

$$= \frac{7.75n^2}{4n^2} = 1.94 \quad (9)$$

It is important to note here that cluster formation is a one time activity. There will be some communication needed for cluster maintenance, but we ignore that for the sake of simplicity. We make similar assumptions in other approaches also. Equation 9 shows an interesting fact about cluster formation process that the average number of messages exchanged per node for cluster formation remains constant immaterial of the size of the network.

Execution of PEM Task: Now let us consider that a team of miners has issued a PEM request using their handheld device that acts as a mobile base station. This agent arrives back at the miners' gadget after carrying out the task. We assume that the mobile agent visits each node just once. We also assume that the agent follows the shortest path to get back to the miners' gadget. These assumptions are not that difficult to ensure in a real deployment. In the multiagent approach that we mentioned earlier, mobile agent, while navigating a cluster, can query the cluster head about efficient routes to visit all nodes in the cluster. Since each node in the cluster is registered with the cluster heads' AMS, positioning data of nodes can be easily reported to the cluster head along with registration messages, thus enabling it to calculate efficient routes to cover its managed cluster.

Total number of hops that agent travels in order to cover the network of Figure 4. These hops are equivalent to messages exchanged between nodes in the multiagent approach.

$$= \text{No. of Nodes} + \text{No. of Layers} = 4n^2 + n$$

Assuming that the agent size in terms of number of messages is S_a . Then, the total number of messages transferred by all the nodes can be calculated as follows.

$$\begin{aligned} &= (4n^2 + n)S_a + M_{ch}^1 \\ &= (4n^2 + n) * S_a + (4n^2 - \frac{4n^2}{16}) \end{aligned}$$

Average number of messages transferred by each node

¹ M_{ch} is the number of messages sent by member nodes to their corresponding cluster head's AMS to inform about the arriving or departing mobile agents at the nodes. These messages could be piggybacked with the positioning data of the nodes also.

$$= \frac{(4n^2 + n) * S_a + \frac{15n^2}{4}}{4n^2}$$

On further simplification we get:

$$= \frac{15}{16} + \frac{(4n^2 + n)S_a}{4n^2} \quad (10)$$

It is important to mention here that we consider a single mobile agent carrying out the PEM task. We could have multiple mobile agents doing this task, but then coordinating their routes and message exchanges among them would make communication costs go higher. In multiagent approach, we could use multiple agents performing different tasks or multiple agents collaborating with each other to perform a single task.

4. DISCUSSION OF RESULTS

After having done mathematical analysis of different approaches in the previous section, now we discuss as to how do they fare vis-à-vis PEM requirement in the mining scenario. In Figure 5, we plot average number of messages transferred per node versus the size of the grid-shaped network in terms of number of square shaped layers (n). It is evident from this graph that fixed base station approach doesn't scale well with increasing size of the network. Average number of messages transferred in the network per node increases linearly with increasing size of the network and reaches very high levels soon.

While database oriented approaches do well when the size of network is very small, i.e. for $1 \leq n \leq 2$, they do incur more communication cost as network size increases beyond $n = 2$. Still they do better than the fixed base station approach. They also give quite a good competition to the multiagent approach. Their only drawback is their lack of flexibility and autonomy. They do facilitate data aggregation in the network, but whatever in-network actuation they provide is always user initiated.

Looking at the communication costs of multiagent ap-

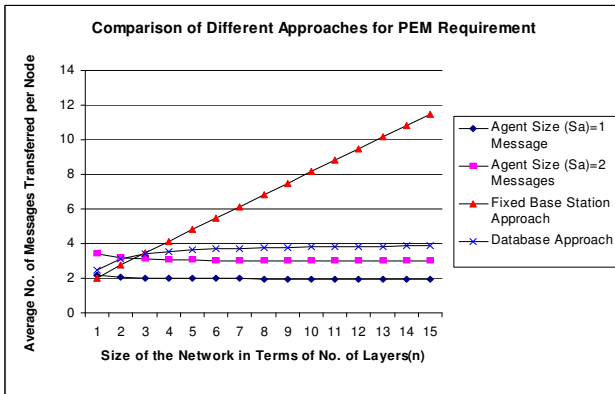


Figure 5: Comparison of different approaches vis-à-vis PEM requirement in the mining scenario

proach, one can easily see that it does scale very well with increasing size of the network. Average number of messages transferred per node remains more or less constant with increasing size of the network. However, one drawback of the multiagent approach is the size of the agents. This approach

competes well with other approaches, but only if we are able to restrict the size of agents to two messages. Otherwise its competitive advantage disappears, as shown in Figure 5.

In this comparison, we don't include cluster formation overhead given in Equation 9, since cluster formation is a one time activity. Including this overhead would make the comparison less fair because cluster formation is not done each time a PEM request is serviced. Dividing the overhead of cluster formation over multiple PEM requests, that the network serves over time, would make its effect negligible.

Though database approaches fare almost as good as the multiagent approach in carrying out the PEM requirement, the other advantages of the multiagent approach should be considered. Ability to make in-network actuation decisions, autonomy, sharing the same network for multiple applications and making intelligent use of node resources are some of the advantages that make multiagent approach a good choice for the kind of scenario discussed in this paper.

5. CONCLUSION

In this paper, we presented a scenario from an industrial domain and discussed the suitability of different WSN approaches for meeting requirements of PEM in it. The approaches discussed are fixed base station approach, FIPA compliant multiagent approach and database oriented approaches with mobile base stations. In the mathematical analysis, we used the metric *average number of messages transferred per node* to compare these approaches. In the mathematical analysis part, we ignored the drawback of fixed base station approach and analyzed it as if it were a simple mobile base station approach. The mathematical analysis showed that the goal of PEM can be more effectively achieved with a FIPA compliant multiagent approach with mobile base stations than with other approaches. However, one potential drawback that could reduce the advantage of multiagent approach over other approaches is the size of the agents. If it increases beyond two messages, then multiagent approach would no longer be as effective as database oriented approaches.

6. REFERENCES

- [1] F. Bellifemine, A. Poggi, and G. Rimassa. Jade: Java agent development framework. In *Proceedings of the 4th International Conference on Practical Applications of Intelligent Agents and Multiagent Technology, London*, pages 97–108, April 1999.
- [2] P. Costa, L. Mottola, A. Murphy, and G. Picco. TeenyLIME: Transiently shared tuple space middleware for wireless sensor networks. In *Proceedings of the International Workshop on Middleware for Sensor Networks, co-located with the 7th International Middleware Conference (Middleware)*, Melbourne, Australia, October 2006.
- [3] C. Curino, M. Giani, M. Giorgetta, A. Giusti, A. Murphy, and G. Picco. TINYLIME: Bridging mobile and sensor networks through middleware. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 61–72, Kauai Island, Hawaii, USA, mar 2005. IEEE Computer Society.
- [4] FIPA. Fipa abstract architecture specification. *at* <http://www.fipa.org>, December 2002.

- [5] C. Fok, G. Roman, and C. Lu. Mobile agent middleware for sensor networks - an application case study. In *Proceedings of the 4th International Conference on Information Processing in Sensor Networks, Los Angeles, California*, pages 382–387. IPSN, April 2005.
- [6] J. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proceedings of the 2nd International Conference on Embedded Network Sensor Systems*, ACM Press, pages 81–94, 2004.
- [7] D. Lange and M. Oshima. *Programming and Deploying Java Mobile Agents with Aglets*. Addison-Wesley Publishing Company, 1998.
- [8] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30(1):122–173, 2005.
- [9] A. Murphy, G. Picco, and G. Roman. Lime: A middleware for physical and logical mobility. In *Proceedings of the 21st IEEE Conference on Distributed Computing Systems*, 2001.
- [10] Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. *SIGMOD Record*, 31:9–18, 2002.