



SOFTWARE AGENT BASED FOREST FIRE DETECTION (SAFFD): A NOVEL APPROACH

Surbhi Gupta¹, Preeti Sethi¹, Dimple Juneja², Naresh Chauhan¹

¹Department of Computer Engineering

¹YMCAUST, Sector 6, Faridabad, INDIA

²Department of MCA

²Maharishi Markendeshwar University, Mullana, Ambala, INDIA

Abstract: The foundation of current research work is to make use of software agents for an event driven application like forest fire detection as deployment of software agents in non deterministic environment of WSN has proved to be better than the conventional client server technology. In this work the task of data dissemination and the decision on the aggregated results is achieved through software agents. The implementation of the same has been done in aglets and the results obtained has indicated significant increase in the lifetime of network which itself is a unique contribution in the community of WSN.

Keywords: Software agents, WSN, Cluster, Energy Efficiency, Life Time of Network.

I. Introduction

Wireless Sensor Network (WSN)[11][12] has attracted researchers over the years because of the plethora of applications it caters with, for instance, forest fire detection. Various sensor nodes forming WSN are deployed in a forest to gather data such as temperature and transfer the same in raw or processed form to a center via base station (or sink node), where incoming data can be analyzed and can be used by scientists for further visual information and predict future states. As a result, fires and some other related events can not only be detected at the center without requiring manual or human-centric operations but also can be avoided in future.

Various Approaches used for forest fire detection have been making use of the conventional client-server methodology in wireless sensor networks. The main drawbacks in this methodology is the delay incurred in transmission of data packets from source to destination and the wastage of network bandwidth during the whole transmission process. The very idea of taking advantage of significant properties such as reactivity and proactivity of software agents has proved to be a solution to the above stated snags. In fact, software agent is an object of the environment which acts proactively and is capable of carrying both code and data to the processing end. Unlike RPC (Remote Procedure Calls), software agents do not engage the caller node/ source node during the processing. Software agents have various features like autonomous nature, mobility, intelligence, cooperative behavior etc which makes them different from conventional objects.

This work primarily makes use of mobile agents and presents an innovative approach for an event based application like forest fire detection. The paper is divided into five sections. Section 1 gives the introduction to the work. Section 2 describes the challenges in the given field and discusses the problem. Section 3 presents the proposed work and working algorithms. Section 4 is about the implementation of the work and discusses results and discussions. Section 5 finally concludes by presenting pros and cons of the current work.

II. Related Work

This section focuses on the work of distinguished researchers in the allied fields. Literature explored highlights that a big volume of scientific population has been actively working towards the best usage of WSN in non-deterministic environment. Further, the community of agent-based researchers has also joined hands so as to improve the results. The section first presents the research related to using WSN in forest fire detection and later highlights the contribution of agents in WSN [14][15].

Buratti et al. [2] has described the wireless sensor technology and gives the overview of forest fire detection as an application of WSN. It provides a general study of various technologies and systems that are currently used towards this goal are system employing Charge Coupled Devices (CCD), cameras and Infrared (IR) detectors, satellite systems and images and wireless sensor networks. **Aslan and Korpeoglu [1]** presented a framework for the design of a WSN for forest monitoring and forest fire detection with random sensor deployment scheme and clustered network architecture. It considers a denser deployment strategy where the distance between neighboring sensors are quite short in order to detect forest fires in a faster way at base station. It proposes an environment aware intra cluster and inter-cluster communication protocol. **Hefeeda and Bagheri [3]** developed a wireless sensor network for forest fire detection based on Fire Weather Index (FWI) system which is one of the most comprehensive forest fire danger rating systems in USA. The system determines the spread risk of a

fire according to several index parameters. It collects weather data via the sensor nodes, and the data collected is analyzed at a center according to FWI. A distributed algorithm is used to minimize the error estimation for spread direction of a forest fire.

Turning our attention to exploring the role of agents in WSN, efforts of Chen and his co authors [6] is appreciable. Authors have proposed mobile agents for efficient data dissemination for reducing and aggregating data in sensor networks. In the traditional client/server-based computing architecture, data at multiple sources are transferred to a destination; whereas in the mobile-agent based computing paradigm, a task specific executable code traverses the relevant sources to gather data. Mobile agents can be used to greatly reduce the communication cost, especially over low bandwidth links, by moving the processing function to the data rather than bringing the data to a central processor.

Sethi et al. [4] has proposed an agent based framework for an event driven application. The work describes the high level view of hierarchical WSN and the working of software agents once an event is detected. **Sethi along with her co-authors [5]** has discussed in detail the packet format of software agents and use of mobile agents for hybrid data fusion.

To the best of our knowledge and available literature reveals that none of the work till date has made use of software agents in forest fire detection especially for clustering [10]. Therefore, a novel approach for data dissemination and aggregation in this application is being proposed in the upcoming section.

III. Proposed Work

The current work is basically an extension as well as an application of AERDP[4] which is an agent based event driven route discovery protocol. The prime functionality of AERDP is to initially detect an event and later discover the best possible route for forwarding the information. Although it makes use of mobile agents for route discovery but it lacks focus on data dissemination, clustering and aggregation. Also, the protocol has failed to focus on suggesting some application, in particular. The proposed work continues to use mobile agent to carry processing code from sensors to actors. The mobile agent then propagates over the network and performs data processing incrementally while proceeding from node to node. Let us consider the case of detecting fire in a forest.

The SAFFD

As it is already mentioned that the foundation of this work is AERDP[4], hence the network which is designed to detect fire is an agent-based network. Now, when fire is detected by such a network, sensors relay this information through the means of mobile agents to the appropriate actors in their respective cluster where actors are cluster heads having decision making capabilities along with a hardware unit attached to them called water sprinklers. These sprinklers then, immediately come into action and extinguish the fire in that particular cluster where the collected and sensed data is still be valid at time of taking actions by actor nodes. The algorithms used for clustering and for computing agent's itinerary are beyond the scope of this work and are left as future works. In this protocol, there are two classes of components- Sensors and Actor nodes. Sensors are regular, low cost, low power devices with limited sensing, computation and wireless communication capabilities. They are intended for sensing the environment and local processing can be done on it. On the other hand, actor nodes are special resource rich nodes which have been made intelligent by embedding mobile agents in them. Multiple actor nodes can coordinate to decide on appropriate actions based on information received from multiple sensors deployed all over the network in random fashion. These have attached water sprinklers to put off fire as and when required. In traditional scenarios, a central entity called sink which performs the function of data collection and coordination which poses a possibility of failure to sink will affect all tasks in network. But in our proposed work, this possibility of single point of failure is being eliminated with the help of software agents.

The proposed approach works in two phases namely data collection phase and risk analysis phase (see figure 1). The description of each phase is given below:

A. Data collection phase

In this phase, MA is dispatched to first node to be visited. When MA arrives at a sensor node, it first looks at id of current node to decide whether it has arrived on destination or not. If not, MA continues migrating towards and specific sources. Otherwise, it operates as follows:

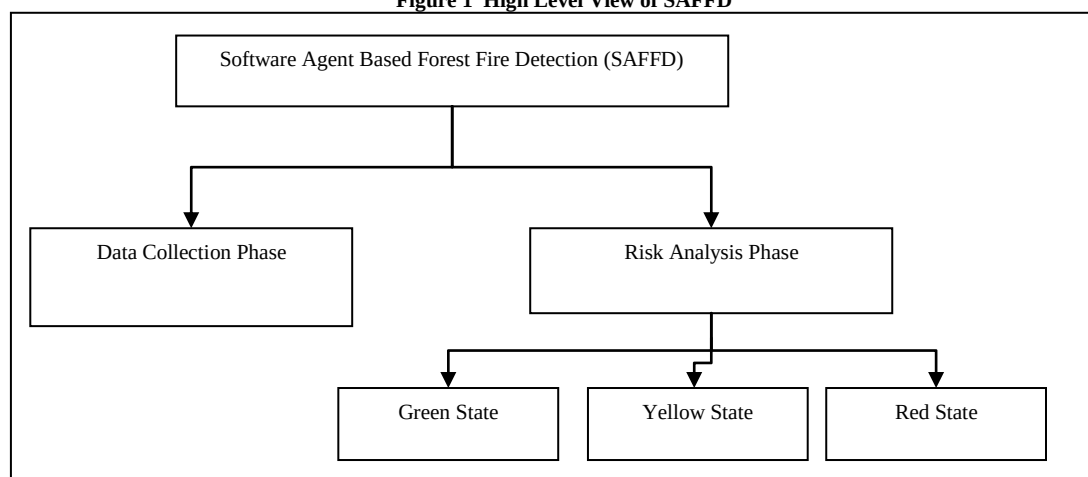
- (i) Collect locally processed sensed data
- (ii) Delete id of current target from *source_list*
- (iii) Choose next source to move on.
- (iv) If current source is *last_source*, the MA will return to actor.

B. Risk Analysis Phase

This phase deals with analyzing the severity and risks associated because of intensity of fire. The severity of fire being detected is categorized in 3 levels i.e. green state, yellow state and red state. Here, *Green State* represents the lowest risk state for possible fire danger. In this state, sensor's sensing period is longer as compared to other states' period. Further, in contrast to *Red State* which is the highest risk state and water sprinklers get activated

in this state to put off the fire, *Yellow State* represents the medium risk state. This state may have transition to red or green state.

Figure 1 High Level View of SAFFD



Working of SAFFD

As mentioned earlier, SAFFD works in two phases: data collection phase and risk analysis phase. In data collection phase, actor dispatches MA to first_source to the last_source according to the itinerary chosen in source_list. At each sensor node, it collects sensed data and process it locally. At last, the aggregated and processed temperature value is passed to actor. In risk analysis phase, actor decides whether the risk state is green, yellow or red using some predefined thresholds to specify the severity level of fire. The green state is least severe while yellow state is moderate risk state. But red state means fire sprinklers need to be initiated to put off the fire before it spreads in the whole forest area. The working algorithms are being listed in figure 2(a) and figure 2(b) respectively.

Figure 2(a) MA Data Collection Phase

Input:
source_list //sequence of all sensor nodes in the cluster
first_source //first source where MA is dispatched initially
last_source //destination node in cluster.
Output: T_agg_data // Aggregated and processed temperature value
Steps:
Step 1: Increment MA_SEQ_NUM and timestamp the Event
Step 2: Dispatch MA to first_source from actor for a round in cluster
2.1 if current_source=first_source then, MA migrates towards first_source node
2.2 else if current_source=next_source and next_source!=last_source then,
(i)MA collects locally processed sensed data
(ii)delete the id of current_source from source_list in MA
(iii) Among the sources in source_list, select the one with Maximum gradient as next_source
(iv)Set next_source in MA packet
(v)MA migrates towards next_source
(vi)Make status of current_source sleep=true
2.3 else if current_source=last_source then, MA collects locally processed sensed data.
MA migrates back to actor node.
2.4 else if next_source=first_source then, MA migrates between source nodes.
Step 3: Receive T_agg_data i.e. aggregated processed data from sensor nodes at Actor
Step 4: Analyze risk state and take appropriate action accordingly. Dispatch MA to Base Station.

Figure 2(b) Risk Analysis Phase

Inputs:
Tgreen_threshold, Tgreen_MAX, Tgreen_Avg, Tyellow_max, Tyellow_Avg // some predefined thresholds constants
T_agg_data // received processed data in above round.
C // Error value constant for precision.
Output: risk state
Steps:
Step 1: Green State
if T_agg_data >= Tgreen_threshold + C then, enter yellow state
Step 2: Yellow State
If T_agg_data >= Tgreen_MAX + C then, enter RED State
if T_agg_data < Tgreen_Avg + C then, enter Green state
Step 3: Red State
if T_agg_data < Tyellow_Avg + C then, enter Yellow state.
if T_agg_data >= Tyellow_max + C then, initiate water sprinklers.

Packet Format of Mobile Agent

The master agent now maintains a data structure known as MA packet structure. Table 1 delineates the modified packet format of the master agent.

Table 1: Packet Structure of Mobile Agent

Agent Identification(I,j)	Agent Itinerary(a,b,c)	MA_SEQ_NUM	Aggregated and processed data	Processing code

- i. **Agent Identification (I, j):** It is a quadratic tuple (I, j) where, 'I' is the id of dispatcher actor and 'j' is the timestamp assigned to MA.
- ii. **Agent Itinerary(a,b,c):** It is a cubic tuple(a,b,c) where, 'a' is id of first node (first_source) to be visited, 'b' is the id of destination node(last_source) and 'c' is the base address of source_list containing the id's of all the nodes to be visited in order.
- iii. **MA_SEQ_NUM :** It is used to define an MA packet which is migrated incrementally from sensor to sensor within a cluster.
- iv. **Aggregated and processed data:** It is locally processed sensed data stored in MA data buffer.
- v. **Processing code:** It is the execution code carried by MA used to process sensed data.

IV. Implementation And Results

The work has been implemented using Aglets. Aglet API is an agent development kit consisting of a set of Java classes and interfaces that allows creation of mobile Java agents i.e. "write once, go anywhere" as once an aglet is written, it will run on every machine that supports the Aglet API [7]-[9]. Aglet API mirrors the applet model in Java and hosted by an Aglet Server i.e. Tahiti Server. Mobile agent [13] is an agent which supports the idea of autonomous execution and dynamic routing on its itinerary as they are the java programs that can halt execution, travel across the network (with both code and state together) and continue execution at another host. The implementation has been carried out on a single machine with three ports (the work however is scalable to three different machines on LAN). The first port acts as an actor which dispatches agents to other two ports acting as sensors in a predefined sequence. These sensors then, sends aggregated and processed temperature value from all the sensors back to actor. The execution of code is shown in figures 3(a) to 3(d).

Figure 3(a) Command Window with for Actor(port 4434)

```
C:\Windows\system32\cmd.exe - aglet -f C:\Java\aglets\conf\aglets.props
C:\Java\aglets\public>aglet -f C:\Java\aglets\conf\aglets.props
[Loading security policy file: C:\Users\ADMIN\lib\security\java.policy... done.]
[Loading security policy file: C:\Users\WELCOME\aglets\security\aglets.policy... done.]
[Loading shared secrets from file C:\Users\WELCOME\aglets\security\secrets.dat ... file not found.]
No secrets.

Licensed Materials - Property of IBM
    (C) Copyright IBM Corp.
    1996, 1998 All rights reserved.
    US Government Users Restricted Rights -
    Use, duplication or disclosure restricted
    by GSR RDP Schedule Contract with IBM Corp.

[IBM Aglets Class Library 2.1.0]
reading aglets property from C:\Users\WELCOME\aglets\users\aglet_key\aglets.properties
reading RDP property from C:\Users\WELCOME\aglets\users\aglet_key\aglets.properties
[Warning: The hostname seems not having domain name.]
Please try -resolve option to resolve the fully qualified hostname
or use -domain option to manually specify the domain name.]
reading property for tahiti from C:\Users\WELCOME\aglets\users\aglet_key\tahiti.properties
[Warning: The hostname seems not having domain name.]
AUTHENTICATION MORE OFF.
creating loader
Agent arrived ...MobilityEvent(HRR100L)
Sensor1 doing job
getting ADVERTISEMENT for cluster formation
getting Join Request
Replied by Sensor1: Yes
getting Sense Request
Sensed temperature by Sensor 1- 1000
creating loader
Sensor2 aglet is created
Before Moving ...MobilityEvent(DISPATCHING)
No integrity check because no security domain is authenticated.
Advertisement message sent by Actor to Sensor 1
Joining message sent by Actor to Sensor 1
response from Sensor1: Yes
Advertisement message sent by Actor to Sensor 2
Joining message sent by Actor to Sensor 2
response from Sensor2: Yes
```

Figure 3(b) Command Window for Sensor1(port5000)

```
aglet -f C:\Java\aglets\conf\aglets.props -port 5000
C:\Windows\system32\cmd.exe - aglet -f C:\Java\aglets\conf\aglets.props
C:\Java\aglets\public>aglet -f C:\Java\aglets\conf\aglets.props
[Loading security policy file: C:\Users\ADMIN\lib\security\java.policy... done.]
[Loading security policy file: C:\Users\WELCOME\aglets\security\aglets.policy... done.]
[Loading shared secrets from file C:\Users\WELCOME\aglets\security\secrets.dat ... file not found.]
No secrets.

Licensed Materials - Property of IBM
    (C) Copyright IBM Corp.
    1996, 1998 All rights reserved.
    US Government Users Restricted Rights -
    Use, duplication or disclosure restricted
    by GSR RDP Schedule Contract with IBM Corp.

[IBM Aglets Class Library 2.1.0]
reading aglets property from C:\Users\WELCOME\aglets\users\aglet_key\aglets.properties
reading RDP property from C:\Users\WELCOME\aglets\users\aglet_key\aglets.properties
[Warning: The hostname seems not having domain name.]
Please try -resolve option to resolve the fully qualified hostname
or use -domain option to manually specify the domain name.]
reading property for tahiti from C:\Users\WELCOME\aglets\users\aglet_key\tahiti.properties
[Warning: The hostname seems not having domain name.]
AUTHENTICATION MORE OFF.
creating loader
Agent arrived ...MobilityEvent(HRR100L)
Sensor1 doing job
getting ADVERTISEMENT for cluster formation
getting Join Request
Replied by Sensor1: Yes
getting Sense Request
Sensed temperature by Sensor 1- 1000
creating loader
Sensor2 aglet is created
Before Moving ...MobilityEvent(DISPATCHING)
No integrity check because no security domain is authenticated.
Dispatched to Sensor2
```

Figure 3(c) Command Window with for Sensor2(port 6000)

```
aglet -f C:\Java\aglets\conf\aglets.props -port 6000
C:\Windows\system32\cmd.exe - aglet -f C:\Java\aglets\conf\aglets.props
C:\Java\aglets\public>aglet -f C:\Java\aglets\conf\aglets.props
[Loading security policy file: C:\Users\ADMIN\lib\security\java.policy... done.]
[Loading security policy file: C:\Users\WELCOME\aglets\security\aglets.policy... done.]
[Loading shared secrets from file C:\Users\WELCOME\aglets\security\secrets.dat ... file not found.]
No secrets.

Licensed Materials - Property of IBM
    (C) Copyright IBM Corp.
    1996, 1998 All rights reserved.
    US Government Users Restricted Rights -
    Use, duplication or disclosure restricted
    by GSR RDP Schedule Contract with IBM Corp.

[IBM Aglets Class Library 2.1.0]
reading aglets property from C:\Users\WELCOME\aglets\users\aglet_key\aglets.properties
reading RDP property from C:\Users\WELCOME\aglets\users\aglet_key\aglets.properties
[Warning: The hostname seems not having domain name.]
Please try -resolve option to resolve the fully qualified hostname
or use -domain option to manually specify the domain name.]
reading property for tahiti from C:\Users\WELCOME\aglets\users\aglet_key\tahiti.properties
[Warning: The hostname seems not having domain name.]
AUTHENTICATION MORE OFF.
creating loader
Agent arrived ...MobilityEvent(HRR100L)
Sensor2 doing job
getting ADVERTISEMENT for cluster formation
getting Join Request
Replied by Sensor2: Yes
creating loader
Agent arrived ...MobilityEvent(HRR100L)
Sensor2 doing job
getting Sense Request
Sensed temperature by Sensor 2- 1500
Aggregated processed Temperature in Cluster-1500
Dispatched to ActorAglet
Enter yellow state
```

Figure 3(d) Integrated results for all ports

```
C:\Windows\system32\cmd.exe - aglet -f C:\Java\aglets\conf\aglets.props
C:\Java\aglets\public>aglet -f C:\Java\aglets\conf\aglets.props
[Loading security policy file: C:\Users\ADMIN\lib\security\java.policy... done.]
[Loading security policy file: C:\Users\WELCOME\aglets\security\aglets.policy... done.]
[Loading shared secrets from file C:\Users\WELCOME\aglets\security\secrets.dat ... file not found.]
No secrets.

Licensed Materials - Property of IBM
    (C) Copyright IBM Corp.
    1996, 1998 All rights reserved.
    US Government Users Restricted Rights -
    Use, duplication or disclosure restricted
    by GSR RDP Schedule Contract with IBM Corp.

[IBM Aglets Class Library 2.1.0]
reading aglets property from C:\Users\WELCOME\aglets\users\aglet_key\aglets.properties
reading RDP property from C:\Users\WELCOME\aglets\users\aglet_key\aglets.properties
[Warning: The hostname seems not having domain name.]
Please try -resolve option to resolve the fully qualified hostname
or use -domain option to manually specify the domain name.]
reading property for tahiti from C:\Users\WELCOME\aglets\users\aglet_key\tahiti.properties
[Warning: The hostname seems not having domain name.]
AUTHENTICATION MORE OFF.
creating loader
Agent arrived ...MobilityEvent(HRR100L)
Sensor1 doing job
getting ADVERTISEMENT for cluster formation
getting Join Request
Replied by Sensor1: Yes
getting Sense Request
Sensed temperature by Sensor 1- 1000
creating loader
Sensor2 aglet is created
Before Moving ...MobilityEvent(DISPATCHING)
No integrity check because no security domain is authenticated.
Dispatched to Sensor2

C:\Windows\system32\cmd.exe - aglet -f C:\Java\aglets\conf\aglets.props
C:\Java\aglets\public>aglet -f C:\Java\aglets\conf\aglets.props
[Loading security policy file: C:\Users\ADMIN\lib\security\java.policy... done.]
[Loading security policy file: C:\Users\WELCOME\aglets\security\aglets.policy... done.]
[Loading shared secrets from file C:\Users\WELCOME\aglets\security\secrets.dat ... file not found.]
No secrets.

Licensed Materials - Property of IBM
    (C) Copyright IBM Corp.
    1996, 1998 All rights reserved.
    US Government Users Restricted Rights -
    Use, duplication or disclosure restricted
    by GSR RDP Schedule Contract with IBM Corp.

[IBM Aglets Class Library 2.1.0]
reading aglets property from C:\Users\WELCOME\aglets\users\aglet_key\aglets.properties
reading RDP property from C:\Users\WELCOME\aglets\users\aglet_key\aglets.properties
[Warning: The hostname seems not having domain name.]
Please try -resolve option to resolve the fully qualified hostname
or use -domain option to manually specify the domain name.]
reading property for tahiti from C:\Users\WELCOME\aglets\users\aglet_key\tahiti.properties
[Warning: The hostname seems not having domain name.]
AUTHENTICATION MORE OFF.
creating loader
Agent arrived ...MobilityEvent(HRR100L)
Sensor2 doing job
getting ADVERTISEMENT for cluster formation
getting Join Request
Replied by Sensor2: Yes
creating loader
Agent arrived ...MobilityEvent(HRR100L)
Sensor2 doing job
getting Sense Request
Sensed temperature by Sensor 2- 1500
Aggregated processed Temperature in Cluster-1500
Dispatched to ActorAglet
```

The implementation results above convey that the use of mobile agents drastically increase the lifetime of network as number of packets transmitted over the network have been reduced to a great extent. It thus overcomes the shortcomings of Client/Server based methodology.

V. Conclusion

In an environment where source nodes are close to each other, and considerable redundancy exists in the sensed data, the source nodes generate a large amount of traffic on the wireless channel, which not only wastes the scarce wireless bandwidth, but also consumes a lot of battery energy. Instead of each source node sending sensed data to the sink node, as typically occurs in client/server-based computing, our paper proposes a Software mobile agent based paradigm for data processing/aggregating/ concatenating in a wireless sensor actor

network architecture. Though the work has been carried out for detecting forest fires but it can be extended to any event driven application. The work is scalable i.e. it can be extended to n number of nodes in the cluster. Though the use of software agents causes extra burden on the network, but the increased cost is compensated when the end results increase the lifetime of the network.

VI. References

- [1] Yunus Emre Aslan, Ibrahim Korpoglu, Özgür Ulusoy, "A framework for use of wireless sensor networks in forest fire detection and monitoring" in the Elsevier Computers, Environment and Urban System, 2012
- [2] Chiara Buratti, Andrea Conti, Davide Dardari and Roberto Verdore; "An Overview on Wireless Sensor Networks Technology and Evolution"; in MDPI journal.
- [3] Hefeeda, M., & Bagheri, M. (2009). Forest fire modeling and early detection using wireless sensor networks. *Ad Hoc Sensor Wireless Networks*, 7, 169–224.
- [4] PreetiSethi, Dr. Dimple Juneja, Dr. Naresh Chauhan, "A Mobile Agent Based Event Driven Route discovery protocol for Wireless Sensor network (AERDP)" *International Journal of Engineering Science and Technology (IJEST)*, December 2011.
- [5] PreetiSethi, Dr. Dimple Juneja, Dr. Naresh Chauhan, "Hybrid Data fusion using Software Agents for an event driven application of Wireless Sensor Network".
- [6] Min Chen†, Taekyoung Kwon, Yong Yuan, and Victor C.M. Leung, "Mobile Agents based Wireless Sensor Networks", *JOURNAL OF COMPUTERS*, VOL. 1, NO. 1, APRIL 2006.
- [7] <http://www.javaworld.com/javaworld/jw-04-1997/jw-04-agents.html>
- [8] <http://www.javaworld.com/javaworld/jw-04-1997/jw-04-hood.html>
- [9] <http://www.javaworld.com/article/2076918/learn-java/solve-real-problems-with-aglets--a-type-of-mobile-agent.html>
- [10] Abbasi, A. A., & Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, 30(14–15), 2826–2841.
- [11] Chee-Yee Chong, Kumar, S.P., "Sensor networks: evolution, opportunities, and challenges," *Proceeding of the IEEE*, Vol.91, NO. 8, pp. 1247- 1256, Aug. 2003.
- [12] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4), 393–422.
- [13] Arkady Zaslavsky, "Mobile Agents: Can They Assist with Context Awareness," *IEEE MDM*, Jan. 2004, Berkeley, California.
- [14] Lloret, J., Garcia, M., Bri, D., & Sendra, S. (2009). A wireless sensor network deployment for rural and forest fire detection and verification. *Sensor Nodes*, 9(11), 8722–8747.
- [15] Doolin, D. M., & Sitar, N. (2006). Wireless sensor nodes for wildfire monitoring. In *Proc. of SPIE symposium on smart structures and materials*, San Diego (pp. 477–484).