Chapter 4

A SCALABLE AGENT PEDESTAL FOR WSNs

This chapter presents a Scalable Agent Pedestal (SAP) for WSNs. In SAP mainly MAs are used to manage the WSNs. It provides common solution to WSNs fault tolerance, load balancing, energy & end-to-end network delay problems. SAP gives true distributed computing and communication environment with the help of MAs. It supports code mobility over the mobile/fixed BS and SNs.

Rest of the chapter is organized as follows. Issues are explored in Section 4.1. Section 4.2 highlights on system model. Section 4.3 gives the architecture of Scalable Agent Pedestal (SAP) and advantages of it are discussed in Section 4.4. Section 4.5 presents results and discussion and finally chapter is summarized in Section 4.6.

4.1 ISSUES
Distributed WSN (DWSN) has become a very popular research topic due to its wide application spanning across civilian and military domain, including environmental monitoring (e.g. temperature sensing), generic object tracking (e.g. people or object locator), and surveillance in a large building, disaster area or battlefield. The advances in sensor technology and ad hoc wireless networking have brought the study of DWSN to a new stage the emergence and spurs of WSNs. It is economically feasible today to implement WSNs, but there are several technical challenges that must be overcome before they can be used for the increasingly complex information gathering tasks. These tasks, such as battlefield surveillance, remote sensing, global awareness, etc., are usually time critical, cover a large geographical area, and require reliable delivery of accurate information for their completion. The new challenges brought to the study of WSN include:
• data volumes being integrated are much larger due to the increasing amount of sensors being deployed;
• the communication bandwidth for wireless network is much lower;
• the environment is more unreliable, causing unreliable network connection and increasing the likelihood of input data to be in faulty; and
• fixed routing is impossible.

In addition of balancing the load across networks for the above defined issues, the system also provides Fault Tolerance, Optimal Resource Utilization, Route and Resource Discovery, Path Maintenance, Service Discovery, And Security. The current available solutions for WSNs have advantages and severe limitations with regard to performance issues. A better WSNs performance can be achieved with a more flexible and intuitive architecture which should be well-researched. Thus, we need to design an adequate WSNs management system which should meet requirements/challenges for fulfilling need of an application.

4.2 System Model
A communication area is divided into different regions. A high energy node will be the member of a region. This high energy node in a region will work like backbone node (BN) and maintains information about other members of the region in the form of database. Each region would have a backbone node (BN). All the routing is carried out through BNs only. Thus, a BN acts as a centralized control for a region. BN is a node with maximum energy and maximum node degree in a particular region. All the nodes (SNs/BNs) are distributed randomly in communication area. In the model, distance table is assign to each node with an entry in terms of hop distance from every other node in the network. The distance is calculated using Euclidian formula.

\[
\text{MAXEN} = 32\% \text{ of } \text{Etotal} \\
\text{MINEN} = 30\% \text{ of } \text{Etotal}
\]
Euclidian formula to calculate distance table is:
\[ D_{ij} = \sqrt{(x_i-x_j)^2 + (y_i-y_j)^2} \]

Each node sends request to every other node in their region containing information of their node degree and energy. The node with maximum number of node degree and maximum energy is selected as a BN of that region. Once the BN is selected, all the nodes in that region are notified. All BNs of different regions should be in transmission range of each other.

We are required to develop a computing/communication system for WSNs that fulfills most of the above the challenges. The developed system should enable the fast and cost-efficient deployment of self-managed computing/communication sensing devices with high overall management cost, but with low management cost at each SNs. With developed system one should be able to deploy large scale computing/communication systems without the need of cost-intensive distributed sensing infrastructure to monitor sensitive area with long life network. This system should facilitate to improve the performance and incorporate new ideas.

### 4.3 System Architecture

When a BN/BS wants to search some information it requests to BN for members information (viz. ID, energy level, node degree, etc.). If the BN is not aware about availability of the type of services a BN/BS is interested and presence of the same in the region then it guides the same to the BN/BS. Then BN/BS uses a scalable agent pedestal (SAP) for fault tolerance, load balancing, energy efficient and efficient end-to-end packet delivery across the network for WSNs and creates a MA to perform its desired task in the present region.

Keeping in view of the above defined issues, we have designed and implemented A Platform for mobile agent Distribution & Execution (PMADE) [65-67] based a Scalable Agent Pedestal (SAP) for WSNs {Figure 4.1}. The main components of the system are as follows- Policy Manager, Resource Manager, MAs, Interface, and Agent-Agent communication layer. We have designed SAP for WSN on considering the issues discussed in Section {4.1}. 
This system implements the agents for Load Balancing, Sensing Agents, Application Agents, Interface Agents, etc. depending upon their roles. Details about these agents and other agents will be found in the Chapter 5. These agents execute the predefined policies for network management and share the valuable information with each other through different layers using interface for the
communication. The various components of the system are illustrated in the following sections:

4.3.1 Policy Manager
This manager comprises of different supporting policies and a policy is selected as per requirement of an application.

4.3.1.1 Initiation Selection Policies
It selects one of the predefined policies for executing the MAs. The application selection plays a major role for categorizing and grouping the policies. The policies defined in the development of SAP may be subcategorized into the following:

The load balancing process can be started by under-loaded or the overloaded BN. The overloaded BN is responsible for finding other BNs to share its workload (Base Station Initiative (BS-I)). If a lightly loaded BN is the initiator then it is called receiver initiative (R-I). When above two policies are integrated together, i.e., mixture of first two (BS-I and R-I). It is a hybrid initiative called Symmetrical Initiative (Sy-I).

4.3.1.2 Virtual Server Selection
This policy is further subcategorized into the following: first policy is based on the one-to-one mechanism, where two BNs are picked at random. A virtual server transfer is initiated if one of the BNs is heavily and the other is lightly loaded. Unlike the first scheme, this scheme allows a heavily loaded BN to consider more than one lightly BN at the time of transfer of load. Third scheme is a logical extension of the first two schemes (one-to-one, one-to-many). While in the first scheme we match one heavily loaded BN to a lightly loaded BN and in the second scheme, we match one heavily loaded BN to many lightly loaded BNs, in this scheme we match many heavily loaded BNs to many lightly loaded BNs.
4.3.1.3 Resource Selection
There are various resources available in DWSN. These resources are either available at local site (at the sensing unit, i.e., SN) or at global site (BN/BS). To achieve quick response time and high throughput, proper resource scheduling is always necessary. The resource selection is further subcategorized into the following:

(a) **Route discovery**- This policy is used to select the path from the available SN/BN disjoint paths.

(b) **Topology update**- DWSN is a dynamic network where SNs dead frequently. In this network, topology changes quickly. Hence this policy is used to update the topology of the network.

(c) **Cost computation**- There are various node disjoint paths available from source (SN) to destination (BS). The best path from the available paths is that which is most reliable in terms of data transmission. This policy is used to compute the cost of the available path in terms of reliability.

4.3.1.4 Service Selection
For the sake of simplicity, we have selected three different types of policies for service selection namely multicasting, topology update, and broadcasting. This mechanism is used to avoid delay in transmission of sensitive information from SNs. Topology changes occur frequently in WSN, so to face the challenges due to frequent topological changes each SN sends a hello message to its neighbors.

4.3.1.5 Security Selection
Security policies are based on the PMADE on the security framework [10].

4.3.1.6 Data Aggregation and Correlation Engine (DACE)
It is necessary for the DACE to aggregate and correlate the different detection results before further transmission. The local DACE for a mobile base station (MBS) are capable of operating in a standalone mode and detect attacks against the SN/BN. Since WSNs are constrained by bandwidth, energy consumption, and
process capability, it is desirable to correlate the alert information on the local SNs first, before transmitting every alert across the network.

The functionality of the DACE depends on the node type: if the node is a BN, its DACE utilizes the aggregation and correlation to combine the detection results from the intrazone SNs in the same region and neighboring gateway BNs. If the node is an intrazone SN, the functionality of the local DACE is to distribute the outputs to all the gateway BNs in the same region.

4.3.2 Agent Management System (AMS)
The main task of AMS is management of agents and nodes (SNs/BNs/BSs), i.e., registration, authentication, security and mobility as per PMADE [10, 65-67]. AMS can also be extended to support clustering and teaming

AMS selects the agents as per their requirement. These agents are not fixed but vary according to their role. Agents are divided into different groups and they are picked by AMS depending upon their requirement. Each agent has a role defined and executes the predefined policies. The generalized architecture of the developed SAP consists of the following groups of agents which are used in heterogeneous networks. This list of agent is not only limited to following groups also. Detailed architecture of AMS is given in Chapter 5.

4.3.2.1 Load Balancing- For balancing the load across the network, we have developed a set of agents which are briefly introduced as under.

Load Index Agent (LIA)
This agent calculates the load index (LI) of each resource (Processing Unit: PU, Memory, I/O) on a particular node (SN/BN). LI of PU is the sum of remaining PU lifetimes of the tasks running on a node. LI of memory is the sum of page fault processing time of tasks on a node (SN/BN). Similarly LI of I/O is the sum of I/O processing time of tasks on a node (SN/BN). The sum of load index of each resource is the total value function for a node (SN/BN) and is used for load transfer by the respective agent. Normally this cared by BNs, because we are balancing the load on BNs only. If model will be used at global network (Internet) level then BS is also included in the list for managing the load.
Load Transfer Agent (LTA)

This agent is used for migration of task from heavily loaded BN to lightly loaded one. It executes two predefined policies namely–local and global. The policy is chosen according to the response time of task submitted for execution. If the response time of a task at the local site is less than the global site then local policy is executed otherwise global policy.

Resource Management Agent (RMA)

This agent is responsible for gathering information about each node’s (SN/BN) resource requirement and passing this information to Resource Manager (RM)[194], which makes its entry in resource database and allocates the appropriate resource to the requesting node.

Routing Agent (RA)

It is a stationary agent responsible for updating the routing table that resides at each node (SN/BN). RA carries a route vector table containing the communication cost from the assigned node (SN/BN) to other nodes (SN/BN) in the network. This table has a lifetime measured by the number of hops. It plays an important role in informing each node (SN/BN) in the network about the addresses of other nodes (SN/BN) and if failure of link for a particular node (SN/BN) is detected, RA spreads route failure information over the network by flooding the updated table.

Load Computation Agent (LCA)

This is a set of MA. It is responsible for information gathering. It travels around the BNs and collects the load information, and propagates this information to other BNs.

Directory Agent (AD)

This agent is activated whenever an overloaded situation arises on a BN. AD finds the suitable receiver partner for the overloaded BN that launched it.
4.3.2.2 Resource Discovery- For route discovery and balancing the load across the network, a set of agents are developed. A brief introduction about these agents is as follows:

**Route Discovery Agent (RDA)**
This agent keeps a record of SN disjoint paths which are not deemed failed yet. As soon as the rating of a path falls below a given threshold, the path is discarded from the Active Path Set (APS) (set of SN disjoint paths) and accordingly a new path is added for future references. This agent executes route discovery policy.

**Topology updates Agent (TUA)**
This agent keeps record of one hop neighbors using the hello message technique. As in WSN, SNs died the network in a random interval of time, so every SN/BN should have knowledge about its neighbors. This agent executes topology update policy. For simplicity BN maintains information about SNs in its region.

**Cost computation Agent (CCA)**
This agent computes the cost of sending the messages to another SN in the network. The cost is measured by the number of hops traveled by the message and bandwidth lost due to the presence of selfish SN in between the paths. As the battery power of SN is a sacred resource in WSN, this agent plays an important role in cost computation in terms of bandwidth loss. This agent executes the cost computation policy.

4.3.2.3 Fault Tolerance: For fault tolerance across different types of networks, a set of agents are developed. They are briefly introduced as under-

**Process monitor agent**
This agent monitors the state and starvation of a process in a task queue. It classifies a process state into a processing state, a stop state, a silent state, and an unknown state.
**Processor monitor agent**
This agent monitors the crash state of a node (SN/BN) (shutdown, power value) and the normal state of a node (SN/BN). During the normal execution of a processor, this collects the used and the available node (SN/BN) processing power utilization.

**Network monitor agent**
This agent monitors communication bandwidth, communication latency time, network disconnection, and partition between its own node (SN/BN) and connected nodes (SN/BN).

**Fault decision agent**
This agent decides the occurrence of a failure by analyzing state information of each resource and identifies a process failure, a node failure or a network failure.

**Rescheduling agent**
This agent evaluates the performance benefits that can be obtained due to task migration and decides whether task migration occurs or not. This agent also decides a new resource allocation for tasks.

**State display agent**
This agent shows the state of each resource and the type of failures occurred. Also it decides whether task migration occurs or not. If this agent receives a rescheduling result for migration from the rescheduling agent, it requests to allocate new selected resources and restarts execution.

**4.3.2.4 Service Discovery**
For selection of service across the network, we have developed a set of agents which are briefly introduced as under.

**Advertising Agent**
This agent actively broadcasts service descriptions already registered. The Policy Manager controls the rate of advertisements. Various policies are employed to adjust the rate of advertisement. For example, if the network is fairly static, then
the advertisement rate can be slowed down. Also policy is event driven (Events represent the availability of the paths from source (SN) to destination (BS)). Advertisements can also be assigned different priorities.

**Forwarding Agent**
This agent receives service advertisements and requests for service messages. Then it decides whether to drop or to propagate the advertisement based on the policy. To prevent broadcast storms, this agent uses multicast tree for selectively forwarding service advertisements. For example, this uses to forward advertisements to more active or resource rich SNs in the network.

**Cache Agent**
This agent is responsible for handling remote advertisements, storing remote advertisements of services, handling requests to match services present in the cache. The Forwarding Agent, on receiving an advertisement might also decide to forward it to other SNs/BNs or broadcast the advertisement to all other BNs and from it to SNs. Each advertisement contains a lifetime. When a new advertisement is received by a Cache Agent, the agent decides to either accept it or reject it. An advertisement is accepted only when there is sufficient space in the cache to hold this advertisement or when an old advertisement is removed from the cache based on the policy chosen.

**4.3.2.5 Security**
A set of agents are developed for secure data transmission across the network. These agents are briefly introduced as under:

**Multi path Secure Routing Agent**
This agent executes the multipath policy which has multiple paths to combat the frequent topological change and link instability problem in WSN, since the use of multiple paths could diminish the effect of possible link failures.

**Secret Sharing Agent**
This agent executes secret sharing policy. In this policy, the secret message is divided into $N$ pieces such that in order to get message, the adversary must
compromise at least $T$ shares. With fewer than $T$ shares, the enemy cannot learn anything about the message and has no better chance to recover the secret than an outsider who knows nothing about the message. This gives the desirable security properties.

**Share Allocation Agent**

This agent executes share allocation policy with the objective of maximizing the message security. It chooses a SN disjoint path for secret allocation so that adversary can never get the message.

**4.3.3 Resource Manager (RM)**

Resource Manager (RM) manages the resources in the network. Each task (agent) in execution has its own resource requirement, which is provided by RM. It keeps track of which resource is available at which node (SN/BN). Resources may be at local site or at global site. As soon as the demand of resources comes from the tasks(agents) in execution, these are provided by the RM. MAs execute predefined policies to provide the desired resources to the demanding task (agent). In this process they also consume certain resource like memory, processor time, etc. So, to keep track of the all these resources, RM is included in the architecture which fulfills the demand of resources as per the requirement.

**4.3.4 Agent-Agent Communication**

SAP includes P2P type communication between agents. It supports both local & remote communication and uses wireless transmission as the transport mechanism for remote communication. Agent Communication Channel (ACC) is a message routing agent integrated in the SAP, which delivers messages as requested by their senders. It supports both local as well as remote communication. The agent-to-agent communication, also known as the agent-based messaging paradigm, and is uses message queue processor [193] for making the different type of communications viz. federated communication, direct communication, etc[192-193]. Agents can communicate directly or indirectly through the ACC. The agents can be located at the same node (SN/BN/BS) or communicate wirelessly between remote nodes (SN/BN/BS).
For information processing in WSN applications, SAP supports both types of computational models, namely, remote communication between agents located at different nodes (SN/BN/BS), as well as agent mobility with local communication at the same node (SN/BN/BS). In the CSCP, the SAP Sensing Agents (SSAs) {details will be found in Chapter 5} located at the constituent SNs communicate remotely with the agents in BNs. In the MACP, the mobile SAAs move from SN-to-SN and communicate locally with the SSAs, residing on the same SNs. This layer provides the facility for agent-agent communication. There are three layers for agent-to-agent communication.

**Resource and Data Management Layer**
RM which is the key component of the system architecture operates on this layer. As discussed earlier it is responsible for managing the resources consumed by MAs when it executes on a particular site. On each layer the respective MA operates.

**Coordination Layer**
MAs communicate and coordinate using communication and coordination layers. The request an agent receives from the communication layer is submitted to the coordination layer for further processing. Agents communicate by exchanging messages using mobile group approach through reliable communications channels. This is normally happen between BS-BS, BN-BN, BS-BN and SN-BN.

**Mobility Manager Layer**
SAP supports agent mobility. It provides ability to agent to move from one node (SN/BN/BS) to another in the network. Migration of an agent from source (BS) to destination (SN) and vice-versa may trigger updates for the service provider agents.

Among the different components of the MA, the most important is the MA's itinerary. Itinerary can be determined either statically or dynamically {details can be seen n Chapter 5}. That is, it can be calculated either before the agent is dispatched or while the agent is migrating. Dynamic itinerary planning is more flexible, and can adapt to environmental changing (sensor ups and downs) in real time. However, since the itinerary is calculated on the fly, it also consumes more
computation time and more power of the local sensor. In the SAP this job is done by BNs and exchange between BN and BS which saves energy of SNs. Computation-efficiency, power-efficiency, and flexibility are three conflicting objectives that cannot be satisfied at the same time.

A simplified sub-optimal solution that determines the agent itinerary on the fly based on three parameters obtained in real time, namely, the remaining energy on board the SN, the signal energy sensed at the current location, and the geographical distance with possible neighbors. When the packet size is the same, the communication cost is proportional to the distance between the source and the destination, which can be calculated using the longitude and latitude information exchanged between nodes(SN/BN). The SAP Broker Facilitator Agent (SBFA) \{details can be seen in Chapter 5\} on each SN is responsible for exchanging information through ACC when the SN is initially activated and when dramatic changes have occurred to any of these three parameters. Therefore, the SBFA on each SN is able to calculate the cost for migrating to each neighboring SN.

Before the MA migrates to the next destination, the SBF compares the cost and directs the SAP Application Agent (SAA) \{details can be seen Chapter 5\} to migrate to the one with the lowest cost, that is, the SN with the high remaining energy, sensed with high signal energy, and very close to the current SN.

### 4.4 Advantages

1. The SAP saves network bandwidth, because only results travel on the network and not the whole agent. Further, SAP agents regularly reduce their size by removing processed methods which are no longer required during their itineraries. When there are several clones of an agent running, saving in bandwidth is quite substantial. It also saves processing power, secondary and primary storage devices at the remote node.

2. From the security point of view, agents are required to report results to the BS/BN in a secure manner, so that further tampering of the result is not possible by the BN. In SAP, the agent provides the result in encrypted form to the BN.

3. If an agent is running on a machine which is disconnected from the network, its results remain on the machine for a fixed period of time. When the machine is reconnected the results are sent to the BS/BN. However, if this duration is
large, the BS/BN removes the entries of the pending agents and the result is lost. The system administrator can set this duration as per the application requirements.

4. SAP is component oriented, both in system environment and in agent construction. The basic component of the system is a MA from which both the system components and user agents are built. Component orientation allows the application developer to adopt a systematic design which is extensible and in which ideas such as transport protocols or communication mechanisms from different systems can be imported in the form of plug-in components.

5. A useful aspect of the design of SAP is modularization to facilitate experimentation. It is possible to redesign and re-implement a single module within SAP, without affecting any others.

6. Major advantages of remote agent creation of the SAP are that user does not have to install the agent platform on the mobile device. The system also reduces communication over wireless links to overcome low bandwidth and network disconnection. SAP enhances service functionality by operating without constant user input. SAP is platform independent. One of its important features is its flexibility and extendibility.

4.5 Results and Discussion

WSNs have continually proven their usefulness in many different fields ranging from temperature regulation to traffic monitoring. Intelligent agents are allowing such WSNs to be more usable, productive, and available and they will continue to do so as the technology continues to develop. Because of hardware constraints, using intelligent agents on WSNs can be a very difficult task, which is why middleware technologies are needed. Middleware software can reduce the memory footprint, provide mobility to agents, help to maintain agents in the network and significantly reduces development time.

There are different middleware software solutions for WSNs available today, each with their own advantages and disadvantages. When MAs are desired on severely limited hardware such as a Mote, SAP may be the good solution because of its small footprint and great support for agent mobility. Agilla is also good and support mobility to agents. If there is more powerful hardware is available, Jade, along with Jess may be a viable solution because of Jade’s
feature rich API and possible expert system shell. If dynamic updates are needed, SAP and Impala may be considered.

It is essential to evaluate each situation and possible choice carefully to identify the best solution. The final solution depends upon the existing requirements of the implemented WSN coupled with the best possible middleware to create the desired outcomes.

### 4.6 Summary

In this chapter we have presented a Scalable Agent Pedestal (SAP) for WSNs. In this system mainly MAs are used to manage the network. SAP is capable of balancing the load across network. It uses set of MAs which are executing predefined policies for finding the load and resource requirement status at each node (SN/BN/BS). Due to the support for mobility of agents in heterogeneous networks, it is able to locate the resource and service in WSNs. Inter-agent communication is supported by the system to get the updated load information on each SN. It generates less message transfer complexity and overhead compared to other existing systems developed earlier. Also it has a major impact on the efficiency of the network with the following performance measurement metrics—response time, energy, throughput, fault tolerance and end-to-end network delay.

In the next chapter MA based adaptive data dissemination model for WSNs is presented.