Context Aware Resource Allocation in Distributed Sensor Networks

Lokesh. B. Bhajantri[1], Nalini. N[2], Gangadharaiah. S[3]

[1]Department of Information Science and Engineering,
Basaveshwar Engineering College, Bagalkot, Karnataka, India.
lokeshcse@yahoo.co.in

[2]Department of Computer Science and Engineering,
Nitte Meenakashi Institute of Technological, Yelahanka, Bangalore. Karnataka, India.
nalinaniranjan@hotmail.com

[3]Department of Information Science and Engineering,
Acharya Institute of Technology, Soladevanahalli, Bangalore.
gangadhar.s@gmail.com

ABSTRACT

Distributed Sensor Networks (DS'.s) have been an area for active research over the past few years, due to potentially widespread applications in creating smart computing environment. In this work, we have proposed new smart computing environment for super/hyper market. The proposed system uses Distributed Sensor Networks (DSN's) for real-time tracking of the products. While new technologies are making sensors nodes smarter, smaller, and cheaper, it is very challenging to find an optimal to allocate sensors and other network resources.

Context awareness is one of the methods to improve resource utilization in Distributed Sensor Networks (DSN's). We have studied Indian retail industry as case study and we listed out the drawbacks of Indian retail industry. But proposed system can be applied to most of the retail industries across the globe to overcome similar problems. As a part of proposed system, we have derived many contexts in retail industry based on human perspective. At a basic level of design, sensors are deployed in each rack which provides the real time information about available products in each rack. This information can be used by warehouse manager, store manager for day to day activity in super/hyper market. R&D team can also use this information to understand consumer behaviour. The context information derived from sensor node is used in allocation sensor node or resources. We have tested our proposed system using Castalia simulation tool, a derivative of OMNeT++. The results obtained from our experiment shows advantages of context aware system. As DSN's are data centric, analyzing and implementing system in user context prospective will greatly contribute in terms of cost, performance. We have simulated the context aware resource allocation in distributed sensor networks to test the operation scheme in terms of performance parameters. The DSN's promise to revolutionize sensing in a wide range of application domains. This is because of their reliability, accuracy, flexibility, cost effectiveness and ease of deployment, Smart sensors can offer vigilant surveillance and can detect and collect data.

KEYWORDS

Distributed Sensor Networks (DSN's), Context, Retail Industry, Store Manager, Floor Manager, Cluster Head, and OMNET++.

DOI : 10.5121/ijwmn.2012.4203

1. INTRODUCTION

Advances in hardware and wireless network technologies have created low-cost, low-power, multifunctional miniature sensor devices. These devices make up hundreds or thousands of ad hoc tiny sensor nodes spread across a geographical area. These sensor nodes collaborate among themselves to establish a sensing network. A Distributed Sensor Network (DSN) that can provide access to information anytime, anywhere by collecting, processing, analyzing and disseminating data. Thus, the network actively participates in creating a smart environment. Distributed Sensor Networks (DSN's) have been one of the choices for building smart environments. Smart environments can be designed for building, utilities, military, shopping mall, industrial, home, shipboard, and transportation systems automation. Sensory data generated from multiple sensors in the network is used for creating smarter environment. The main goal of DSN's is to make decisions or gain knowledge based on the information fused from distributed sensor inputs. At the lowest level, individual sensor node collects data from different sensing modalities on-board. An initial data processing can be carried out at the local node to generate local event detection result. These intermediate results will then be integrated/fused at an upper processing center to derive knowledge and help making decisions. A general DSN consists of a set of sensor nodes, a set of processing elements (PE's), and a communication network interconnecting the various PE's [1, 2].

While this new kind of networks has wide range of applications, it also poses serious challenges like frequent network topology change, limited computational, memory and power supply [3, 4]. Sensors are more prone to failures. With all these constraints an efficient and effective method to extract data from the network is challenging task. To address these challenges DSN have flexible design both in hardware, software. Hardware for sensor node is designed with dynamic voltage scaling, different level of sleep modes, hardware level data fusion & aggression, intelligent signal processing, small battery, remote configuration etc. In software, the operating system is designed to work under constrained memory, computation power, battery power, with limited bandwidth etc. The communication system is also designed to achieve greater energy efficiency. Different protocols like SMAC, LEACH, PEGASIS, LSU and TEE are designed specially for energy efficiency [4, 5]. The benefits and limitation of DSN are: DSN provide smart method of computation. The data gathered from different sensors can be utilized to derive more meaningful context. But it also posses many challenges like node should be cost effective, energy efficient, smaller size, stable and standard middleware. The network developed from this sensor should be self configuring and fault tolerant, application specific and data centric.

Context awareness has great potential for creating new service modes, resource allocation and improving service quality in distributed sensor networks. The various types of contexts in distributed sensor networks are: computing context, user context, physical context, location context and time context. Context could be active or passive. In active context awareness, an application automatically adapts to discovered context, by changing the application's behaviour and in passive context awareness, an application presents the new or updated context to an interested user or makes the context persistent for the user to retrieve later [6]. Context aware computing extends the idea of contexts to refer the physical and social situations, in which computational devices are located.

In this paper, we have considered 3 types of contexts are: Time, seasonal and hybrid context in distributed sensor network environments for resource allocation. Context means situational information or can be stated as: "Context is any information that can be used to characterize the situation of an entity". Context aware computing aims to provide maximal flexibility of a computational service based on real time sensing of any forms of contexts.

Context awareness has great potential for creating new service modes, resource allocation and improving service quality in distributed sensor networks. The various types of contexts considered in distributed sensor networks are: computing context, user context, physical context, location context and time context. Context could be active or passive. In active context awareness, an application automatically adapts to discovered context, by changing the application's behaviour and in passive context awareness, an application presents the new or updated context to an interested user or makes the context persistent for the user to retrieve later. Context aware computing extends the idea of contexts to refer the physical and social situations, in which computational devices are located [7].

OMNeT++ is a public-source, discrete event simulation, component-based, modular and open-architecture simulation environment with strong GUI support and an embeddable simulation kernel. Its primary application area is the simulation of communication networks. Because of its generic and flexible architecture it has been successfully used in other areas like the simulation of IT systems, queuing networks, hardware architectures, and business processes. OMNeT++ is rapidly becoming a popular simulation platform in the scientific community as well as in industrial settings. Several open source simulation models have been published, in the field of Internet simulations (IP, IPv6, MPLS, etc), mobility and ad-hoc simulations and other areas. However, such a growing community faces also growing challenges and problems: Integration of different simulation tools, porting of simulation models between different platforms, testing and comparison of applications [8].

The rest of the paper is organized as follows. Section 2 describes the problem overview. Section 3 discusses related work in this area. Section 4 describes the proposed work. Section 5 presents the simulation procedure, performance parameters and the results of the proposed work. Finally, we conclude in section 6.

2. PROBLEM OVERVIEW

The Indian retail industry is moving steadily from unorganized sectors to organized sector. With this huge growth rate, organized retail industries are facing many challenges, mainly proper IT infrastructure to manage the day to day activities in all its outlets. So there is an acute need for smarter computing devices to reduce the burden on retail chains. Other challenges faced by Indian retail industry are absence of developed supply chain and integrated IT management, rising rental & labour values, mall management, shortage of trained staff, manual resource allocations, lack of quality locations, regulations restricting real estate purchases, and cumbersome local laws, and taxation, which favours small retail businesses.

2.1 Present Working Model of Indian Retail Industry

The Figure1 explains the present working model of Indian Retail Industry. At the end of the day, store manager get the transaction summary about sold items. This transaction summary is system generated and will be sent to warehouse before they close the outlet shop. Next day the warehouse division will analyze the system generated file and plan for the further processing. All these process will take at least one day before product is replaced in each outlet. In some retail shops, we have observed that this process is taking more than two days. So there is always a chance that outlet may not have item what consumer demands for which result in loss of business.

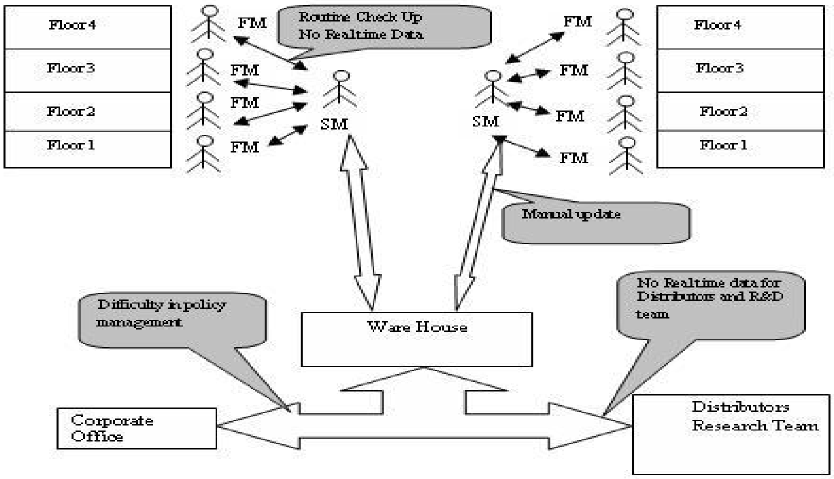

Figure1. Present working model of Retail Stores.

This process model has certain setbacks: 1) Floor manager has to check manually each part of the floor to track the item sold 2) During major festivals or weekend products may sold earlier than expected 3) Always Store Manager (SM) should be in contact with each Floor Manager (FM) for product availability in each floor 4) Some unexpected events like marriage, meeting or some other type of events may happen in around the outlet shop. In this case product may sell earlier than expected by Store manager 5) It increase the labour cost as each floor manager will has to check by himself or with help of other 6) Since all the decision are taken manually, there always chance for human error factor, 7) No visibility to warehouse division about present transactions, 8) Outlet stores have limited space as most of them inside the city. Especially in case of India where real estate cost is considerably high.

In our proposed system, DSN's are used to track the products on each rack of Retail shop. We have developed model by considering only the weight sensors. But depending on type of the products to be tracked, DSN's may be designed to have one or more different type of sensors. We also derived many context related to retail industry. These contexts are used for smarter resource allocation and management. Main contexts derived in our system are: Seasonal Context, Time Context and Hybrid Contexts.

In the proposed system, each sensor node is pre-configured to fixed weight and Context (Either Seasonal or Time context base). After initial setup, each sensor nodes will sample data in a regular interval. Depending on the context and available information, each node will derive an emergency value which signifies the attention to be given to that particular tray. Derived value is forwarded to the Cluster Head (CH), which summarize further with other sensor nodes. Sensor nodes will also modify the event sensing rate or sleep time depending on available information, to reduce the power consumption. Catering the service to every floor by Store manager (SM) solely depends on emergency value forwarded by DSN. The cluster head (CH) acts as intermediate node to forward the data to SINK node. It collects the data from all the node belongs to its cluster. Then it summarizes the data, mainly the emergency value of particular product. Then it forward the data to next cluster which connect to SINK or another cluster.

3. RELATED WORK

We have not found any work related to real time product tracking in retail industry with context aware DSN's for resource allocation. We derived many contexts, related to retail industry. These contexts are used in product tracking and resource allocation. In this section discuss about related work in context aware systems and resource allocation. Creating meaningful knowledge from raw data gathered from sensor challenging task.

The work given in [9] presents a context aware approach to conserving energy in wireless sensor networks. As sensor hardware technology proliferates, research in prolonging sensor battery life gains more interest. Conservation of sensor energy, therefore, becomes a practical approach to prolonging sensor life and eventually reducing the frequency of battery replacement. In retrospect, hardware driven approaches have enabled significant power savings but on the expense of genericity as they are typical to certain sensor hardware. In this paper, they proposed a software based approach that is not constrained by sensor hardware and which is independent of any specific application domain. An implementation is done utilizing the underlying framework.

The work given in [10] discusses many applications of context-aware system including the context aware pill bottle for elderly persons. The system remind elderly users when it is time to take their medication, and a medication monitor situated in a caregivers home that displays awareness information for elder users medication compliance. Context-aware system provides countless opportunities.

The work given in [11] discusses the Context-aware Distributed Sensor Network (DSN) for dementia patients. Incontinence in Patients with Dementia (PWD) due to a decline in their physical and mental abilities. Those PWD may lie in soiled diaper for prolonged periods if timely diaper change is not in place. Current manual care practices may not be able to immediately detect soiled diaper, although costly and labour intensive scheduled checks are performed. Delays in diaper change can cause serious social and medical issues. It uses different sensors like wet sensor, pressure sensor, and accelerometer to design DSN's which help in managing the PWD. Inputs from these sensors are also used to conserve battery power, either by reducing sampling rate of sensor or by switching off the sensor itself.

The work given in [12] presents combinatorial auctions for resource allocation in a distributed sensor network. This work discusses a solution to the problems posed by sensor resource allocation in an adaptive, distributed radar array. They have formulated a variant of the classic resource allocation problem, called the setting-based resource allocation problem, which reflects the challenges posed in domains in which sensors have multiple settings, each of which could be useful to multiple tasks. Further, they have implemented a solution to this problem that takes advantage of the locality of resources and tasks that is common to such domains. This solution involves translating tasks and possible resource configurations into bids that can be solved by a modified combinatorial auction, thus allowing us to make use of recent developments in the solution of such auctions. Also developed an information-theoretic procedure for accomplishing this translation, which models the effect various sensor settings would have on the network's output.

The work given in [13] describes the resource allocation for a distributed sensor network. In this work, they describe a project undertaken for the Office of Force Transformation (OFT) to investigate alternative resource allocation strategies for America's armed forces. In particular, OFT is interested in understanding how resource allocation strategies can be used in the context of distributed, network-centric units. To address this problem they have developed a simulation tool using agent-based modelling to explore the emergent properties of a distributed sensor network. To focuses on the task of using distributed sensors with varying characteristics and capabilities trying to detect and track the movement of enemy units in an urban environment. The goal of the project is to identify the impact of different resource allocation strategies on the performance of the sensor network.

The work given in [14] presents a resource allocation and congestion control in distributed sensor networks–a network calculus approach. The establishment of the overall objectives of a distributed sensor network is a dynamic task so that it may sufficiently well ‘track’ its environment. Both resource allocations to each input data flow and congestion control at each decision node of such a network must be performed in an integrated framework such that they are sensitive to this dynamically established overall objectives. In this paper, the effectiveness of a ‘per-flow’ virtual queuing framework that decouples the input data flows to each decision node is demonstrated. Under this framework, the buffer set point level of a decision node is established via the control of set point levels of individual virtual buffers assigned to each source node. Network calculus notions are utilized to model the end-to-end flow and design a simple yet effective feedback control law for each input data flow. The control strategy, while enabling satisfactory tracking of a dynamically allocated buffer queue set point, is also robust against the time varying nature of network delays and buffer depletion rate. Some of the works given in [15, 16, 17, 18]

DSN’s always work under constrained resources. The resources can be any type like computational power, network bandwidth, battery power, memory or sensor node itself. Main motto in resource allocation is to increase the overall network life time of DSN’s. Data caching, data aggression, context summarization, mobile agent based data collection are few important technique to improve network life time of DSN’s by conserving resources.

4. PROPOSED WORK

Distributed Sensor Networks (DSN’s) are mainly used for monitoring, tracking and remote controlling the system. We propose new context aware resource allocation of DSN’s in retail chain management. We have taken Indian Retail industry as Case. DSN’s are used in our proposed system to track the products in real time. Field Survey is done for major retail companies like More, Food World, Big Bazaar, Reliance, and Spencer. In the survey, we have found no real time information management system is available in market. And there is acute need for this kind of systems. We have received positive feedback from most of the store manager for our proposed system. Considering current retail industry, this section discusses about proposed system model, and functioning schema (proposed algorithm).

4.1 System Model.

In our proposed system, DSN’s are deployed to obtain the real time information about availability of products in retail shop. We have developed model by considering only the weight sensors. But depending on type of the products to be tracked, DSN’s may be designed to have one or more different types of sensors.

Figure2 shows the design of each rack. Each tray contains sensor which will keep track of how many items are available. This information is send to the cluster head (CH) in regular interval. The cluster head (CH) will also collect similar data from other racks. Data is aggregated in each cluster head before it is forwarded to another cluster head (CH).

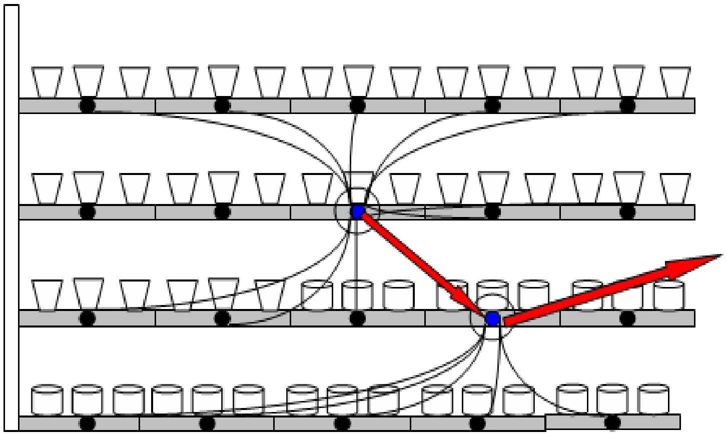

Figure2. Design of each rack.

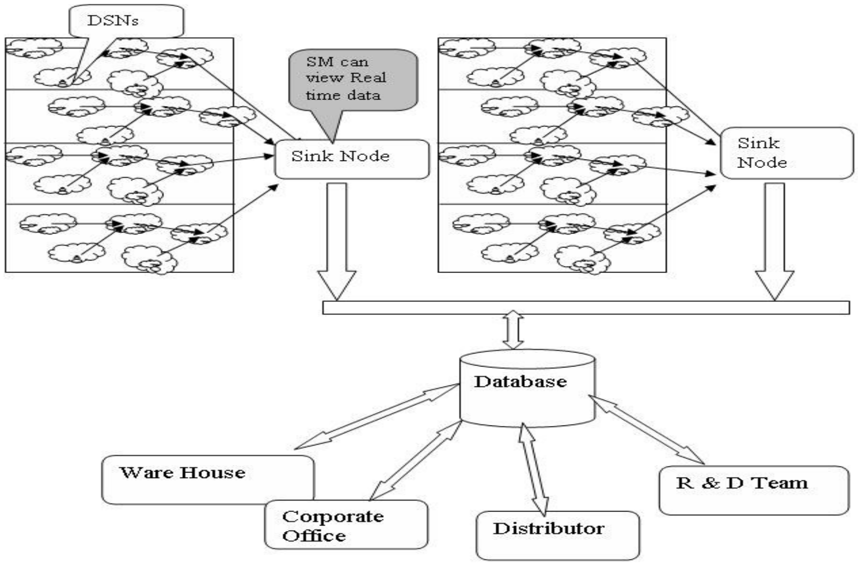

Figure3. Detail view of proposed DSN's system model for Retail Chain Management.

The Figure3 will explain proposed model. Data from each sensor in rack is forwarded to its cluster head. Each cluster head will aggregate the information sent by its sensor group and forward to its parent cluster. Forwarded information from all cluster head can be stored into a single system in each outlet shop. If there are many retail shops, then similar data from all retail shop can be stored in single centralized database. The store manager, corporate manager, distributor or R & D team can view this data for further analysis. Proposed system has following advantages compare to existing model 1) Store manager (SM) will be less depends on Floor Manager (FM) for any kind of updates 2) Product replacement within outlet for any floor can be done efficiently 3) No need of routine check up for each floor or rack 4) Labor cost can be reduced dramatically as fewer members are required for maintenance 5) Ware house manager can view the product availability at any time in each outlet. By this sold products can be replaced to each outlet depend on emergency. This process will to take at least one day in present system 6)Corporate head office can keep track of each and every outlet shop 7) Product Distributors will also get enough time to prepare for supply 8) R &D team can analyze the data obtained by this network for understanding consumer behavior, and 9) Reduces manual or paper work.

In our proposed system i.e. retail shop; we have derived many contexts in distributed sensor network environment. These contexts are used for efficient resource allocation. Main contexts are 1) Seasonal Context 2) Time Context and 3) Hybrid context.

In present working model of retail industry, Floor Manager (FM) has to do regular routine check up for availability of products in each rack. Figure5 discuss the advantage of DSN's over present system. The algorithm discussed below will explain how the burden on Store Manager(SM) is reduced by deriving meaningful data from context awareness of the system. Algorithm also explains about how to allocate resources depending on the context.

4.2 Steps for each Sensor Node in Proposed Work

Begin

1. Each sensor node is pre-configured to fixed weight and context. (Either Seasonal or Everyday context base).
2. After initial setup, each sensor nodes will sample data in a regular interval.
3. Depending on the context and available information, each node will derive an emergency value which signifies the attention to be given to that particular tray.
4. Derived value is forwarded to the Cluster Head (CH), which summarize further with other sensor nodes.
5. Sensor nodes can modify the event sensing rate or sleep time depending on available information, to reduce the power consumption.
6. Catering the service to every floor by Store manager (SM) solely depends on emergency value forwarded by DSN.
7. Steps 2 to 7 continue till the End of the Day (EOD).

End

4.3 Steps for each Cluster Node in proposed work

Cluster head forward the summarized emergency level to parent cluster head. Each Cluster Head will maintain data table related emergency level of its group. Update to table is done by each sensor node independently.

Begin

1. Initialize the cluster head to receiving mode.
2. Store the data in vector table.
3. Summarize the emergency value for Product information.
4. Forward the data to parent Cluster head in case of higher emergency.
5. Wait for next update from Sensor nodes in the cluster.
6. Repeat steps from 2 to 6.

End

4.4 Functioning Schema: Algorithm for each Sensor Node

We have used following data types and data structure for the algorithm. The algorithm has been developed using Castalia an OMNeT++ derivative. We have defined three type of context like time, seasonal, hybrid context. Three different scale and emergency level has been used. Structure *clusterRecord* is defined to store productId, rackId, emergency level to summarize the emergency information. *struct nodeRecord* store the static data for initializing the each node. This contains the table of information related to context, productID, rackID for each sensor.

Class *Connectivity Map* is designed according to Castalia requirement. This class is derived from VirtualApplication of Castalia publicly. VirtualApplication is generic class designed for development of application in Castalia. We also defined virtual functions like startup(), finishSpecific() and Application packet to exchange the data between the nodes. Following steps explain Implementation of Algorithm (using Castalia).

Step1. Once the basic initialization is done by Castalia, startup() is called for each node

```
Startup().
{
    Initialize variables specific to each node
        if(cluster node == true)
            Clear the cluster info.
        if(!Cluster Node)                              // Configure each normal node
             Configure_each_node()
        if ( nodeType = TIME_CONTEXT_SENSOR )
             setTimer(TIME_CONTEXT_TIMER, startTxTime);
        elseif(nodeType == HYBRID_CONTEXT_SENSOR )
            setTimer(HYBRID_CONTEXT_TIMER,txInterval_total*3);
            sendNodeToSleepState();
        else if ( nodeType == SEASONAL_CONTEXT_SENSOR )
            setTimer(SEASONAL_CONTEXT_TIMER, txInterval_total*9);
            sendNodeToSleepState();
}
configure_each_node()
{
        for each node i=1 to n
            intialise context type, productid, rackid, emergencyLevel.
}
```

Step2. After setting timeout in startup(), we wait for timeout. Check for sensor reading. If it is emergency send the packet to sink/cluster node.

```
timerOutCallback()
{
 If(  timeContextTimer) {
        emgLavel = getSensorValue();
        if(emgLevel == EMERGENCY_HIGH or EMERGENCY_MEDIUM )
             toNetworkLayer(packet2Net, SINK_NETWORK_ADDRESS);
        setTimer(TIME_CONTEXT_TIMER, txInterval_total);
}

If(  hybridContextTimer)
{
        emgLavel = getSensorValue();
        if(emgLevel == EMERGENCY_HIGH or EMERGENCY_MEDIUM )
          toNetworkLayer(packet2Net, SINK_NETWORK_ADDRESS);
        setTimer(HYBRID_CONTEXT_TIMER, txInterval_total);
}

If (seasonalContextTimer)
{
        emgLavel = getSensorValue();
```

```
            if(emgLevel == EMERGENCY_HIGH or EMERGENCY_MEDIUM )
                        toNetworkLayer(packet2Net, SINK_NETWORK_ADDRESS);
            setTimer(SEASONAL_CONTEXT_TIMER, txInterval_total);
}
```

Step3. Cluster node will update the table, which should be forwarded to next cluster

```
fromNetworkLayer( )
{
       If (cluster Node)
            Update Emergency Level of cluster and forward to next cluster or sink
           If (sink node)
               If (emergencyLevel is high)
                        Update to Store Manger and reduce Emergency Level of a node by
supplying products
}
```

Step4. While sending packet to network Layer, fill the product id, rackid, floor id and emergency level.

```
toNetworkLayer()
{
           Create ContextReportingPacket; Fill the productID, rackID, emgLevel, FloorID in
ContextReportingPacket, Send to Cluster Head
}
```

Step5. Collect the output of each module for further process using finishSpecific().

5. SIMULATION

We conducted simulation of the proposed scheme by using Castalia simulator (OMNeT++). The proposed model has been simulated in various distributed sensor network scenarios. For experimental purpose, we have conducted the simulation for 5 to 20 numbers of Nodes. We have considered M=1 to 5 Cluster and in each cluster we have considered N = 5 to 10 nodes. This section presents the simulation model, simulation procedure, results and discussions.

5.1. Simulation Model.

As discussed earlier, Castalia implements different modules of sensor node. Our simulations are conducted by developing separate application with required modification in sensor manager, radio, and power management module. In Castalia, node 0 is considered as SINK node and it also support bypass of routing and MAC layer to ease the testing in application layer. Since Context awareness is implemented in Application Layer, our application will bypass the routing and MAC layer and sent all packets to Node 0. The simulation is conducted with different number of sensor nodes. For simulation purpose, we have considered 5 to 10 nodes in each cluster. The simulation parameters are considered in this work ad shown in table1.

SINK Node	Node 0
Simulation Area	30 X 30 meters
Simulation Time	7000 seconds
Energy required for transmitting single packet [Tx mode Power]	46.2 mWatt
Energy required for receiving & processing single packet [Rx mode Power]	62 mWatt
Initial Power [Power of two AA batteries]	18720 Joules
Baseline Node Power [Energy utilized by node, even if node is not transmitting or receiving]	6 mWatt
Date Rate	250 kbps
Modulation Type	PSK
Bandwidth	20 MHz

Table1. Simulation Parameters.

5.2. Simulation Procedure.

To illustrate the results of simulations, we have setup the configuration files of OMNeT++ i.e. .ini and .ned file. In all the scenarios, Node 0 is considered as SINK node with one hop reaching. This is because Castalia simulation tool except all nodes to be one hop count reach from the sending node. We have conducted simulation of the proposed scheme by using Castalia simulator (OMNeT++). For experimental purpose, we have conducted the simulation for 5 to 20 numbers of Nodes. We have considered M=1 to 5 Cluster. And in each cluster we have considered N = 5 to 20 nodes. Node 0 is considered as SINK node as in Castalia. This section presents the simulation model, simulation procedure, results and discussions. To illustrate the results of simulations, we have setup the configuration files of OMNeT++ i.e. .ini and .ned file. Following are the major default values used in our simulation. Consider the following steps involved in the simulation.

Begin

1. Initialize .ini and .ned files with parameters along with the required default value.
2. Configure each sensor node according to application requirement. Each node has to be configured for Context Type, Product ID, Rack ID and default emergency Level.
3. Set Context Timer for each node, compile and run the simulation with Castalia script. In our simulation we have set 200s, 600s, 1800s for time context, Hybrid context and seasonal context sensor.
4. After timeout, each node sends the data to SINK node. Required data is each layer is collected by collectOutput() in finishSpecific().
5. Use CastaliaResult utility to analyze the collected information.
6. Compute the performance parameters.
7. Generate the Graphs according to results obtained.

End

5.3. Performance Parameters.

We have used the following parameters to measure the performance of the proposed scheme.

1. **Number of Packets Transmitted (Tx):** The packet transmission is defined as the total number packets transmitted by each nodes in different contexts.
2. **Energy Utilization:** It measures the energy utilization of each node in different context as measured in joules.
3. **Time taken to reduce the Emergency Level:** It measures the time required to reduce the emergency level from high to low by nodes in distributed sensor network environment.
4. **Battery Energy Consumption:** It is the battery energy consumption of all nodes in sending, receiving and forward operations. A small amount of energy also lost during state transition and listening mode.
5. **Throughput:** It is the ratio of number of packets sent to the number of packets received by all nodes. Throughput is c calculated for different type of sensor.
6. **Resource Allocation:** It measures the percentage of resources allocated successfully to all the nodes in different contexts. It is measured in percentage.

5.4. Results and Discussions

5.4.1. Number of Packets Transmitted (TX) Vs Number of Nodes.

Figure4 shows that total number of packets sent by all nodes. Number of packets sent for time context is much higher compare to seasonal context and hybrid context. Figure5 shows the graph about total number of packets which are not received due to sleep mode of nodes. Since more packets are sent in time context, it shows maximum packets. Processing each packet will consume 62mw of energy. To avoid this we are sending each node to sleep mode till its timer expires.

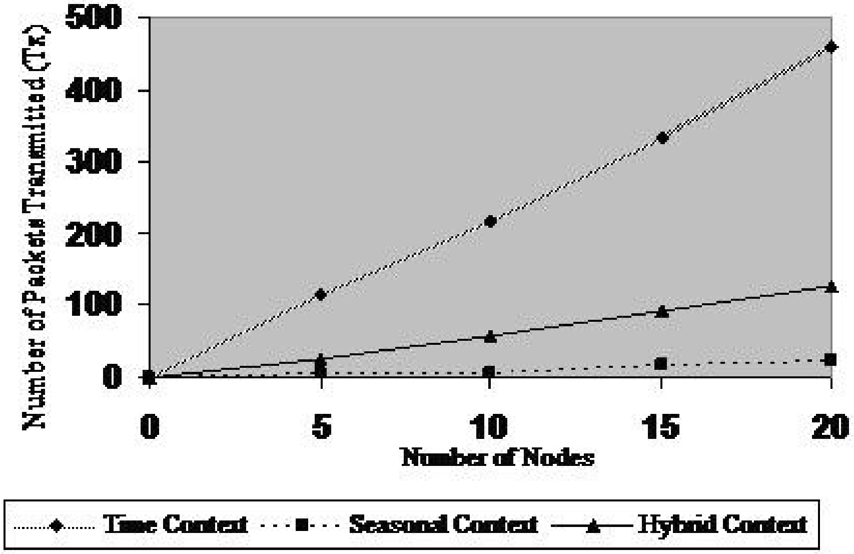

Figure4. Number of Packets transmitted (Tx) Vs Number of Nodes.

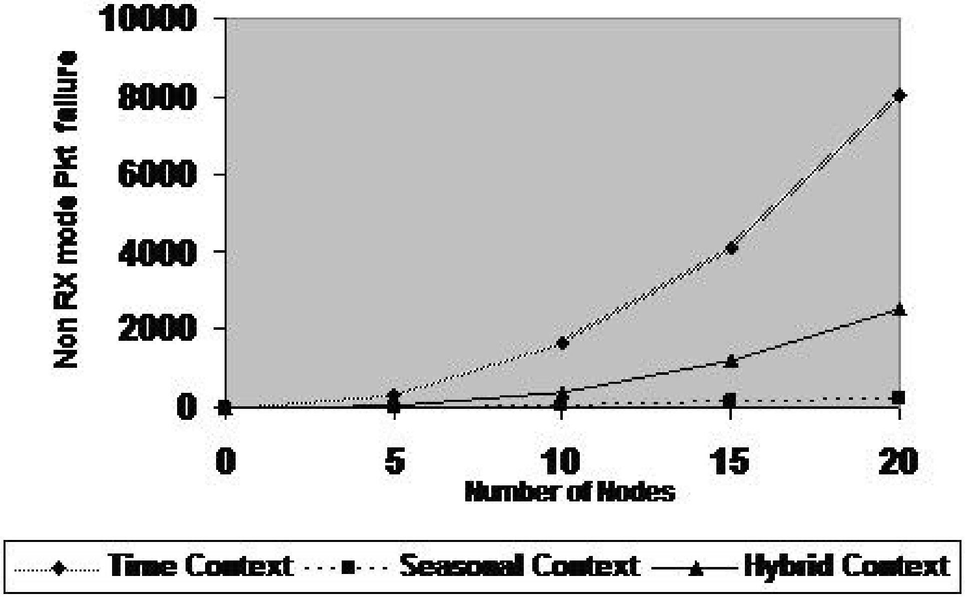

Figure5. Non RX mode Pkt failure Vs Number of Nodes.

5.4.2. Energy Utilization Vs Number of Nodes.

Figure6 shows the graph about average amount of energy spent in each context. As in graph the energy spent in time context is always much high compare to other context. Since overall packets sent in seasonal and hybrid context is less, energy spent is also much lesser compare to time context.

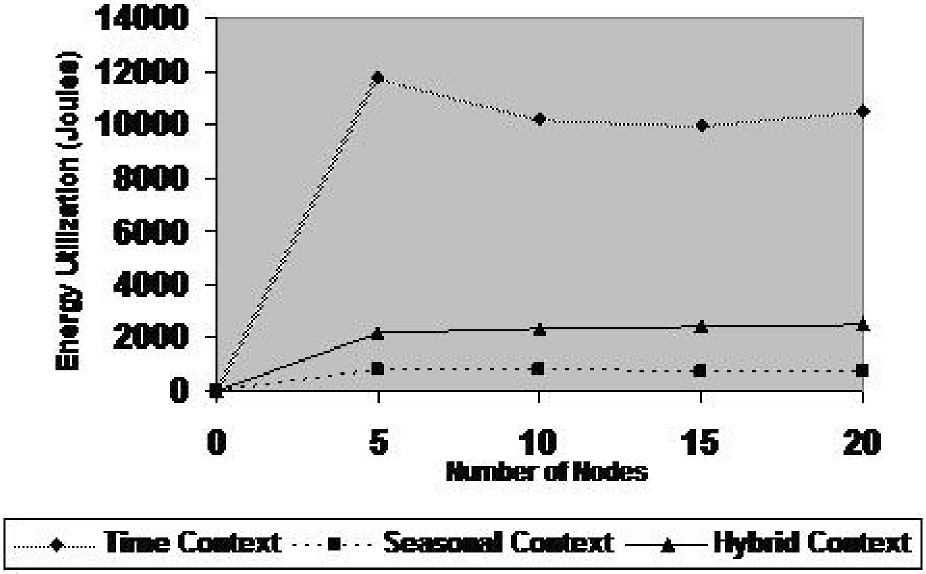

Figure6. Average energy Utilization by all nodes Vs Number of Nodes.

5.4.3. Energy Consumption.

Figure7 shows the battery consumption in different context. In time context sensor nodes, power consumption is almost linear and it is much higher compare other context nodes. The power consumed by hybrid context sensors is less than a 25% that of time context sensor node. Energy consumption by Seasonal sensor node is also much lower than time and hybrid sensor node. (14% of time context sensor node).

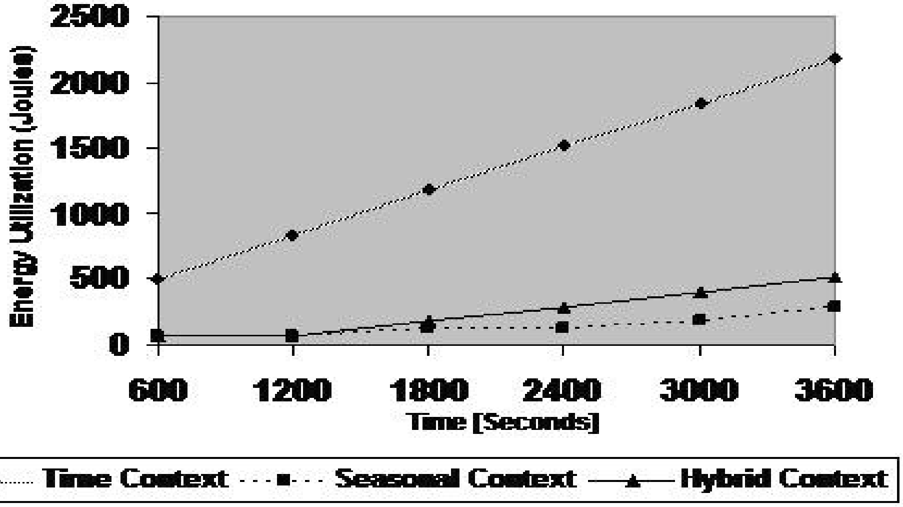

Figure7. Energy Utilization Vs Time.

5.4.4. Time taken to reduce Emergency Level.

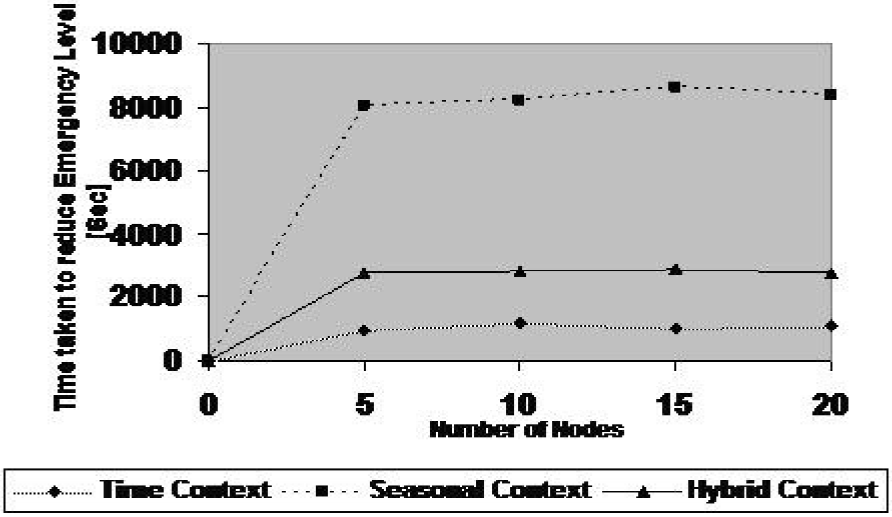

Figure8. Time taken to reduce emergency level Vs Number of Nodes.

Figure8 shows the time taken to reduce the energy level in each context. Generally products related to time context (Products which are daily usage) will be sold quickly. So these products has to be deployed much earlier compare to seasonal and hybrid context.

5.4.5. Throughput.

Figure9 shows the throughput of nodes in different context. Throughput is calculated as ratio of number of packet received to the number of packet sent by all nodes. Sensor nodes in seasonal context outperform the time and seasonal context. This is because number of packets sent in seasonal is less compare to seasonal and hybrid context.

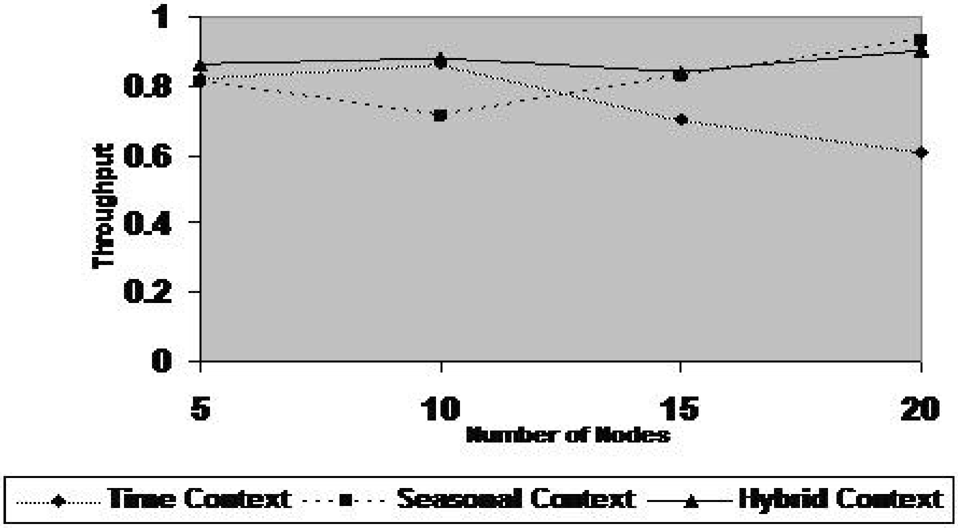

Figure9. Throughput Vs Number of Nodes.

5.4.6. Percentage of Resource Allocation.

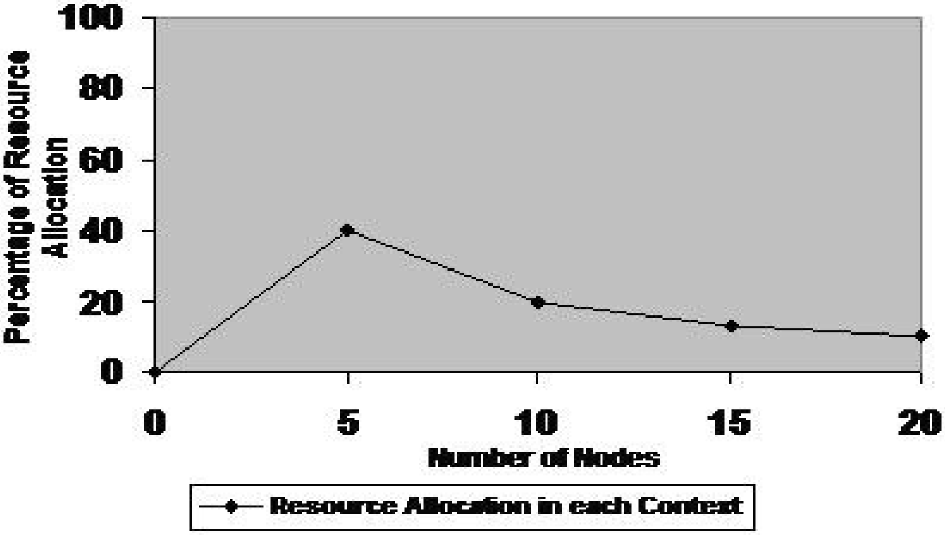

Figure10. Resource allocation Vs Number of nodes.

Figure10 shows the percentage of resource allocation in any context. Here numbers of active nodes are considered as resource. At any point of time, only cluster node and transmitting nodes are activated in any cluster group. The remaining nodes are pushed to sleep state. Because scheduled activation, energy consumption considerable reduced.

6. CONCLUSION

Context aware systems are designed using human perspective, domain specific information. We have analyzed drawbacks of Indian Retail Industry and then we have proposed new system using DSN's. In our design, sensors are deployed in each rack which provides the real time information of products available in that rack. This information can be used by warehouse manager, Store manager for day to day activity in super market or hyper market. R&D team can also use this information to understand consumer behaviour. Sensor nodes and other network resource are also allocated according to context information. To save the energy further, nodes are programmed to sleep state as soon as they send information to cluster node. Simulation results show the advantages of embedding the context awareness in system for resource allocation. Bandwidth utilization, energy consumption is greatly reduced in each context. We have simulated the proposed work by using Castalia simulation tool which is based on OMNeT++. Castalia doesn't provide the support for cluster based protocol and also it support only one hop routing in default mode. Due to these constraints, we have considered sink node (node 0) as cluster head and simulated our module.

REFERENCES

[1] S.S., Iyengar, Ankit. Tendon, and R.R., Brooks, "A overview of Distributed Sensors Network".

Available From:

http://books.google.com/books/about/Distributed_sensor_networks.html?id=Nff5

[2] Shivakumar Sastry and S.S., Iyengar, "Taxonomy of Distributed Sensors Network".

Available From:

http://books.google.com/books/about/Distributed_sensor_networks.html?id=Nff5

[3] Jamal. N, Al. Karaki, Ahmad. Kamal, "Routing techniques in wireless sensor Networks: A survey, Journal of IEEE Wireless Communications, vol. 11, No.6, pp 6-28, 2004.

[4] K. Akkaya, M.Younis, "A Survey of Routing Protocols in Wireless Sensor Networks". In Elsevier Ad Hoc Network Journal, vol. 3, No. 3, pp 325-349, 2005.

[5] Hairon. Qi, S. Sitharama., Iyengar, Krishnendu. Chakrabarty, "Distributed sensor networks - review of recent research". Journal of Franklin institute, vol.338, pp 665-668, 2001.

[6] Faraz. Rasheed, Young-Koo. Lee, Sungyoung. Lee, "Applying Context Summarization Techniques in Pervasive Computing Systems". The 4th IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and the Second International Workshop on Collaborative Computing, Integration, and Assurance, pp. 107-112, 2006.

[7] Suan. Khai., Chong, Shonali. Krishnaswamy, Seng. Wai., Loke, "A Context-Aware Approach to Conserving Energy in Wireless Sensor Networks". In proceedings on 3rd IEEE Conference on Pervasive Computing and Communications, pp 401 – 405, 2005.

[8] Available From;

http://www.omnetpp.org/doc/omnetpp41/manual/usman.html#sec101

[9] Chong, S. K., Krishnaswamy, S., Loke, S. W., "A context-aware approach to conserving energy in wireless sensor networks". In proceedings of 3rd International Conference on Pervasive Computing and Communications Workshops, pp. 401-405, 2005.

[10] Anand. Agarawala, Saul. Greenberg, Geoffrey. Ho, "The context-aware pill bottle and medication monitor department of computer science". In proceedings of video proceedings and proceedings supplement of the UBICOMP 2004.

[11] Aung. Aung., Phyo.., Wai, Foo. Siang., Fook, Maniyeri. Jayachandran, Jit. Biswas, Jer-En. Lee, Philip. Yap, "Implementation of Context-Aware Distributed Sensor Network System for Managing Incontinence among Patients with Dementia". In proceedings of International conference on Body Sensor Networks, pp 102-105, 2010.

[12] J. Ostwald, V. Lesser, S. Abdallah. "Combinatorial Auctions for Resource Allocation in a Distributed Sensor Network". In proceedings of the 26th IEEE International conferences on Real-Time Systems Symposium, (RTSS'05), IEEE Computer Society Press, pp 274 – 279, 2005.

[13] Martin. C., Martin, Iavor. Trifonov, Eric. Bonabeau, "Resource allocation for a distributed sensor network", 2005.

[14] J. Zhang, K. Premaratne , Peter H. Bauer, "Resource allocation and congestion control in distributed sensor networks–A network calculus approach". In Proceedings of 15th International Symposium on Mathematical Theory of Networks and Systems, 2002.

[15] Zhikui. Chen, Zhe. Wei, "Intelligent Home-Hospital System Based on Context-Aware Technology". In proceedings of international conference on Industrial and Information Systems, pp 23- 26, 2009.

[16] Hristova, A. Bernardos, A.M. Casar, J.R., "Context-aware services for ambient assisted living: A case-study". In proceedings of Applied Sciences on Biomedical and Communication Technologies, pp 1-5, 2008.

[17] HK Pung, T Gu, Wenwei Xue, PP Palmes, J Zhu, WL Ng, CW Tang and NH Chung, "Context-Aware Middleware for Pervasive Elderly Homecare". IEEE Journal on Selected Areas in Communications, vol. 27, no.4, pp 510 -524, 2009.

[18] Tomoya. Kawakami, Bich. Lam., Ngoc.., Ly, Susumu. Takeuchi, Yuuichi. Teranishi, Kaname. Harumoto, Shojiro. Nishio. "Distributed Sensor Information Management Architecture Based on Semantic Analysis of Sensing Data". In Proceedings of SAINT, pp 353 -356, 2008.