Virtual Sensor Networks: An Embedded Agent Approach

Richard Tynan, G.M.P. O'Hare, Michael J. O'Grady & Conor Muldoon
CLARITY: The Centre for Sensor Web Technologies
School of Computer Science and Informatics
University College Dublin, Belfield, Dublin 4, Ireland
Email: {richard.tynan, gregory.ohare, michael.j.ogrady, conor.muldoon}ucd.ie

***Abstract*—Many documented instances of existing research on Wireless Sensor Networks (WSN) use deployments that either fall short of, or barely meet, the resource requirements of the application. In this paper, it is envisaged that future WSN deployments will far exceed the resource requirements of any one single application. In a similar fashion to the use of virtual machines on a mainframe, sub-networks of adequate resources will be carved out of the entire deployment to fulfil the requirements of multiple applications. These will be hosted simultaneously on the network, and in many cases, certain WSN nodes will form a component in a number of these Virtual Sensor Networks (VSN). Such VSNs will also be dynamic in nature, adapting resources as nodes go offline. An additional requirement of such networks will be to engage in opportunistic power management, such as node hibernation, while the networks are adapting. In this paper, a solution for both of these issues is proposed, underpinned by a Multi-Agent System (MAS) resident on individual nodes. This solution facilitates both the practical operation of adaptive VSNs, while ensuring aggregate energy consumption can be minimised.**

I. INTRODUCTION

A number of trends are emerging in the development of Wireless Sensor Network (WSN) technology - namely reduced power consumption, form factor and cost. Additionally, nodes are being equipped with ever increasing resources such as memory, CPU and battery capacity. This can be seen from the evolution between the Mica2 Mote [5] to the latest Sun SPOT running the Squawk JVM [15]. The implication of this is that over-deployment of resources will become easier should those trends continue. For instance, more nodes than necessary to perform a given task may be deployed in the environment. Only critical nodes are used to deliver the level of service required, with the possibility that redundant nodes engage in opportunistic power conservation, such as through hibernation.

When this is the case, there are two distinct types of deployment - *hard* and *soft*. The hard deployment is the physical distribution of nodes, while the soft deployment is a software maintained virtual deployment of nodes. Each soft deployment is a sensor network in its own right, albeit a virtual one [6]. The concept of a Virtual Sensor Network (VSN) can be seen as an analogy to the development of Virtual Machines (VM) for mainframe computers. A mainframe will have a vast array of resources e.g. computation and memory. A VM is a partition of these resources dedicated to one user or one application. From a holistic viewpoint, the mainframe is a patchwork of delimited VMs and they can operate both in isolation and co-operatively. For comparison purposes, the mainframe is equivalent to the hard deployment and the VM is analogous to the software maintained VSN.

In this paper, a similar trend is envisaged for WSNs in the future. Resource rich deployments will provide capabilities far in excess of any one application's requirements. This will facilitate the hosting of multiple applications or services on the network collectively. Each of these applications will have their own resource requirements, which must be maintained in order to ensure a level of service. In this work, we demonstrate how a Multi-Agent System (MAS) can be used as the paradigm through which such multitasking of the deployed resources can be delivered. Crucially, we also show how this can be performed with cognisance of the inherent power limitations of the nodes in mind. Therefore, the WSN can engage in opportunistic power management, while providing individual VSN requirements.

In the next section, VSNs are explored in more detail, including some specific examples of these virtual networks and how they must coexist on a node. In Section III, a layered architecture for delivering power aware VSNs is presented along with experimental evidence of its poor performance in Section IV. As a MAS approach is advocated here, some existing agent systems for WSNs are presented in Section V and this is followed by the proposed embedded agent solution to power aware VSNs in section VI. Using a multiple agents on the node, is in contrast to the traditional one-agent-per-node approach [14]. Finally, conclusions and future work resulting from this experimentation end the paper.

II. VIRTUAL SENSOR NETWORKS

The term Virtual Sensor Network (VSN) has been used previously in the field of WSNs to describe the production of a single sensor network from the fusion of underlying distinct collections of heterogeneous nodes [8]. For example, a unified addressing scheme may be used and individual nodes may act as bridges between the individual networks, partitioned according to hardware constraints such as transmission frequencies. While this is useful, it does not address the issue of logically partitioning the sensor network according to the requirements of an application or service. In this work, the unified network resources are partitioned for an applications requirements. This solution also offers the potential to conduct sophisticated power management, such as node hibernation, which is not considered by other approaches [6].

A. Hard and Soft Deployments

Other stake-holders aside, the identification of a nodes redundancy and subsequent hibernation, depends on how a given approach defines precisely what it means for a node to be redundant. Essentially all techniques adhere to the generic approach depicted in Algorithm 1. Nodes continually evaluate their redundancy and hibernate or remain active accordingly.

```
while true do
    read(Neighbour Information);
    if this.isRedundant then
        this.hibernate(sleepDuration);
    else
        this.doWork(activeDuration);
```

Algorithm 1: Algorithm for redundancy identification.

When operating in isolation, a node can hibernate once it is deemed redundant. This can be seen in figure 1, where the less dense VSN soft deployment emerges to satisfy the resource requirements of the single application. In this figure the two exemplar VSNs of connectivity and sensing coverage are depicted, which will be detailed in the next section. Crucially multiple soft deployments can be carved out of the larger hard deployment to provide the required partitioning of resources for each application. Some nodes may be part of one, many or no VSN and in the latter case it can be hibernated.

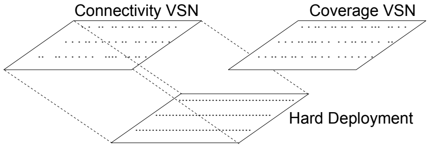

Fig. 1. The hard deployment is the physical deployment of nodes. The soft deployment is a software maintained virtual deployment of nodes. Each soft deployment will be a subset of the resources of the hard deployment.

However, when multiple VSNs are considered, a node cannot blindly hibernate without considering the requirements of the other VSNs. Direct or indirect co-operation between the VSNs is vital to ensure all requirements are met. In addition, they will also have to operate in an adaptive fashion due to node expirations and failures. We will return to this issue in section III, after some sample VSNs are presented.

B. Sensing Coverage VSN

The first exemplar VSN stems from an applications requirement for a defined resolution of sensing data. With an inadequate number of sensor readings, the application may not register an interesting event in the environment because a sensor was not close enough to the event for it to be picked up. When this is the case, a minimum density of deployed nodes is required and any over deployment of nodes will be surplus to the requirement of the VSN. In order to select which nodes are redundant, the template method in Algorithm 1, above, must be instantiated by providing an appropriate redundancy definition. As different techniques can define redundancy differently, they will produce a different topology for the soft deployment with varying properties. The goal for all techniques, however, will be to adaptively ensure that the QoS provided by the network in terms of resolution, does not degrade.

Density maintenance for area coverage is typically achieved through the definition of a sensing radius *r* for each sensor. Within this defined sensing radius, the sensor is capable of sensing the environment, but outside this nothing can be sensed at all. An area to be sensed is covered if all the points within it are inside the sensing radius of at least one sensor. An added degree of redundancy, *k*, can be incorporated into this approach by ensuring that each point is within the radius of *k* sensors - the so-called *k*-Coverage Problem [2]. Node density can be varied to the required degree by increasing or decreasing the radius r.

A number of redundancy definitions have been proposed in the literature and a good survey of these is presented by Wang and Xiao [20]. The techniques outlined in their work essentially create an active VSN of the required resolution. The Coverage Configuration Protocol (CCP) [21] is a spatial redundancy identification technique that considers a node redundant if

1) All intersection points of neighbouring sensing circles are within the sensing range of K_s active sensors.
2) All intersection points of neighbouring sensing circles and the boundary of the sensed are are within the sensing range of K_s active sensors.

Where K_s is the sensing degree of coverage to be maintained by the network. CCP is based on a message beaconing technique, where active nodes periodically beacon to inform neighbouring sensors that they are alive. The neighbours then calculate redundancy based on these beacons, the sensing range and the required coverage degree. It is important to keep in mind that contemporary WSN nodes are equipped with a suite of sensory modalities and as such it may be necessary for multiple VSN densities to be maintained for the various sensors. For example, the sensing density for a temperature application may be greater than that of an acoustic application using the sensors microphones. Simply having enough sensors to sense the environment may not be the sufficient for the effective relaying of the data to the appropriate locations. With this in mind, the next VSN is concerned with adaptively maintaining a connected networking infrastructure.

C. Connectivity VSN

While the previous protocols attempt to maintain spatial coverage from a sensing perspective, the goal of this VSN is network connectivity. In the presence of both hibernating and failing nodes, it is crucially important that network connectivity is not compromised. Due to the potential size of the area to be monitored and the limited transmission range of WSN devices, multi-hopping of messages will be required to deliver them to their destination. Examples of such protocols for WSNs include: Ad-hoc On-demand Distance Vector (AODV) [11] and Greedy Perimeter Stateless Routing (GPSR) [7]. Interestingly, from Xing et al. [21] it has been proved that when $2 * R_{sense} < R_{transmission}$ then sensing coverage automatically implies network coverage. This implies that to maintain network connectivity on its own, a sensing coverage technique can be implemented where the value for $R_{sense} = R_{transmission}/2$.

Some of the connectivity redundancy identification techniques operate on a purely geographical basis, in much the same way as the sensing coverage techniques. They ensure a certain node density, which will guarantee connectivity based on the transmission radius for each node. Geographical Adaptive Fidelity (GAF) [22] is a protocol used to maintain routing connectivity through the use of a virtual grid placed on top of the actual deployment. Alternatively, some approaches to connectivity redundancy identification also factor in a Quality of Service, such as throughput or latency when deciding on a nodes redundancy. SPAN [2] is a protocol that maintains a routing backbone such that all nodes in the network are connected. SPAN also ensures that congestion in the network will not increase due to the introduction of bottlenecks from the hibernation of certain nodes.

Looking at the previous two VSNs, it is clear that the limited sensing range and transmission capabilities of a node require that multiple sensors exist in order to provide an adequate level of service. Another limited resource on a node is it's processing capabilities. In some cases it may be necessary for a larger number of nodes to be active than are required for sensing alone. In which case a VSN of execution entities may be required to adequately process sensing data and this is the VSN we consider next.

D. *Execution VSN*

Suppose that in order to distinguish various events, some processing of sensor readings is required. Now suppose that the number of nodes required to perform this in a reasonable time exceeds the number of nodes required merely to acquire the data. In which case, it may be necessary for additional nodes to remain active to handle the computation. This will happen where there is a dense deployment of nodes but a requirement of a less dense sensing coverage for the application. In which case the sensing redundancy VSN, discussed previously, must co-operate directly or indirectly with this execution maintenance VSN in order to adaptively meet the resource requirements. To deliver the appropriate number of nodes, one of the techniques outlined in Section II-B can be employed. How the tasks are allocated to the sensors is beyond the scope of this paper. Now that we have discussed three possible VSNs that could foreseeably be utilised on a deployed WSN, we turn our attention to how these individual VSNs can be realised in the next section.

III. LAYERED VSN ARCHITECTURE

In order to realise multiple VSNs on a WSN it may at first seem that the most appropriate solution would be based on a layered approach. After all, most WSN applications are built around this concept. The anatomy of a standard WSN application is depicted on the left of figure 2. A MAC layer [12] will usually be located below a routing layer e.g. GPSR, and this in turn will be below the application layer. The MAC layer is responsible for transmission of the messages from the upper layers across the physical wireless channel to a neighbouring node. It will mediate the use of the channel and retransmit failed packets if necessary. While it can attempt to guarantee a perfectly reliable channel, in practice this can be both costly and unnecessary due to the transmission costs and latency.

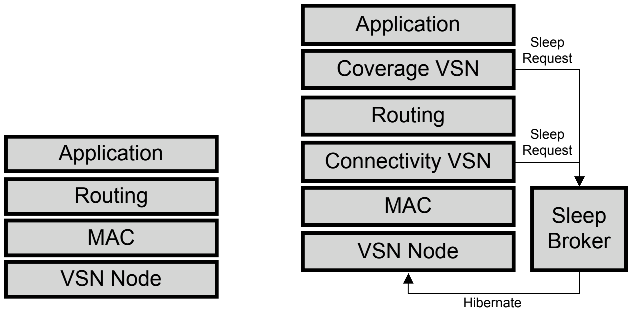

Fig. 2. WSN layers for a basic application. Higher layers relay packets through lower layers for transmission (left). Layered implementation of VSNs on a node with lower layers presenting VSN to upper layer (right).

The multi-hop communication protocol resides above the MAC layer, which forwards packets addressed to destinations out of direct transmission range of the node. This layer will typically maintain a neighbour table to route messages to the destination. The final component is the application resident on the nodes. This could perform a number of tasks that will typically involve sampling the sensors, performing some processing and transmitting data to either neighbouring nodes or a data sink. Introducing VSNs into the layered approach produces the architecture depicted on the right of figure 2. The strategy adopted here is that the VSN layer presents a topology to the layers above it. In the diagram therefore, the Connectivity VSN presents a connected sensor network over which the routing layer can forward packets reliably. Above that, the Coverage VSN ensures that the required density of sensors are active so that the application has an adequate data resolution.

A. *Layered VSN Power Management*

There are however a number of problems with this layered approach as the experimentation in the next section will illustrate. Firstly, upper layers can only relay messages through lower layers. This poses a problem when the lower layer deems the node redundant but the upper layer deems it to be critical. Two approaches can be selected here, the redundant layer can remain active and consume more energy than required. Alternatively the lower layer can hibernate, but the layer must remain active in some regard to forward messages from the upper layers for transmission. This renders such an architecture unsuitable when the layers can hibernate independently.

Therefore, if we select the first alternative and only hibernate the node when both layers deem it to be redundant, the redundancy evaluation period must be harmonised between the layers. This is to ensure that both redundancy decisions are reached as close to simultaneously as possible. The reason for this can be seen when one layer deems the node redundant but is in the process of evaluating the subsequent redundancy. If the other layer decides the node is redundant (now both layers agree), then the node will be hibernated. This is ignoring the fact that the redundancy evaluation of the first layer may deem the node critical.

The final issue with this layered approach to VSNs is that the layers must know about the other layers that will relay information through them. This approach reduces flexibility because a layer cannot be altered or replaced without affecting other, unrelated layers. With this in mind we have conducted a number of experiments on some common VSN combinations that highlight some of these disadvantages. The results of these experiments are presented in the next section. Xing et al. [21] claim to integrate SPAN and CCP when the radius constraint is not met, however, no details on how this can be achieved in practice are given. In the next section we provide details of some initial experimentation on the layered approach. The results from this experimentation motivate the use of a MAS to deliver VSNs.

IV. VSN EXPERIMENTATION

In order to illustrate the concept of VSNs and to demonstrate some of the disadvantages of the layered based approach we now detail two sets of VSNs. In the first configuration, CCP is used as the connectivity maintenance layer by adopting a radius value of half the transmission range as discussed in section II-C. Another instance of CCP is used as the coverage layer in figure 2. In the second VSN, we retain CCP for connectivity purposes but use the Interpolation-based Redundancy Identification and Sensor Hibernation (IRISH) protocol as the coverage VSN [18].

A. Experimental Setup

The simulation environment used for our experimentation is J-Sim [17]. The simulated area for this set of experiments is defined as 100 meters x 100 meters with a deployed node density of one node every 5m. The result of this is that a fixed density of 400 nodes are used to cover the region of interest. One of the primary reasons for selecting this setup is to allow the results to be generalised to large areas by concatenation of networks similar to this. For example, a 500m x 500m region could be configured using 25 instances of the setup used here in a 5 x 5 grid formation.

The power management of nodes takes place using a combination of the sensing and connectivity VSNs. Membership of neither may result in node hibernation by the mediator. The 802.11 MAC layer and GPSR that come with J-Sim complete the networking protocol stack for both experiments. The application simply relays the simulated sensor reading at a node every 10 seconds, however, this reading is currently discarded at the base station. The values chosen for the power consumption in the different states of the transceiver are based on those used in the experiments of CCP [21]: Transmit power 1.4W, Receive power 1W, Idle power 0.830W, sleeping power 0.130W. Nodes are initially given 100 Joules of energy, which roughly translates to 1/100 of the potential energy for a node powered using current battery technology.

B. VSN 1: Connectivity (CCP) & Coverage (CCP)

The effect of density on the % of active nodes can be clearly seen in figure 3. For a sensing radius of 2m and 4m, no node is redundant as the deployed inter-node separation is 5m. After this point the increase in radius brings about an ever reducing % of active nodes required to maintain the density. One important characteristic of this curve is the diminishing return from increasing the radius. The reason for this stems from the fact that a second CCP layer is maintaining a density for the connectivity VSN, therefore after a point increasing the sensing radius will have no effect on density as the minimum node activity has been reached to maintain connectivity.

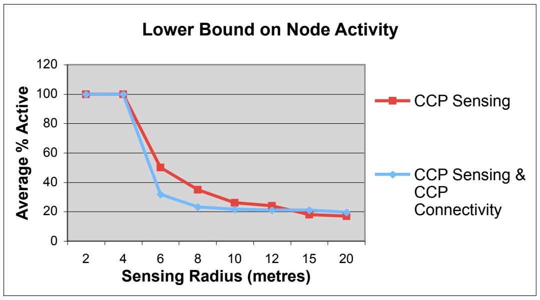

Fig. 3. Node activity of a node as the sensing radius increases. Two approaches are depicted here, the mediated approach and a single CCP instance maintaining sensing redundancy alone.

A similar trend is observed in terms of the average node lifetime. For the smallest two radii, no node can hibernate and therefore an average node lifetime of 120 seconds is observed, which is the maximum lifetime of a node when active all the time. After this a steady increase in node lifetime is observed, however, this curve also levels off due to the connectivity maintenance VSN. An interesting correlation can be seen between node activity, redundancy and node lifetime in figures 3 and 4. Taking for example a radius value of 6m, on average 30% of the 400 nodes are active. This would imply that each node could last three times longer than the base line performance where all nodes are active all the time. However, this is not the case, each node only lasts twice as long, 240 seconds, meaning nodes are expending additional energy in this configuration in order for both CCP layers to operate.

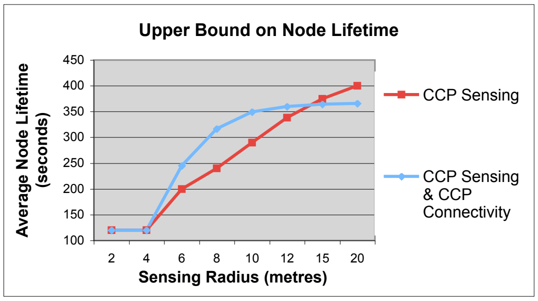

Fig. 4. Average lifetime of a node as the sensing radius increases. Two approaches are depicted here, the mediated approach and a single CCP instance maintaining sensing redundancy alone.

In figures 3 and 4 an additional trend is depicted, where only one CCP instance is in operation to maintain sensing coverage. A number of interesting comparisons can be made between these two trends. Firstly, as the curves representing the system with VSN mediation level off due to the connectivity maintenance VSN operating in the background, the other graphs continue to reduce the number of active nodes and increase the node lifetime. This will be at the expense of the connectivity of the network.

Another interesting point is the fact that both trends in each of the graphs are quite different. It might be expected that identical performance would be observed until the point where the connectivity maintenance creates an asymptote on activity and lifetime. However, such performance is not observed and this can be attributed to a number of factors. Firstly, lost packets due to collisions will increase with two VSNs in operation on the same channel. This will lead to nodes making decisions without a complete picture of their neighbourhood. Secondly, even with homogeneous timing the mediation process may introduce the possibility of a node hibernating based on an old decision currently being revised by one of the layers. If the node had been allowed to stay active for an extra period of time, the decision registered with the mediator would have been updated, section III-A.

C. VSN 2: Connectivity (CCP) & Coverage (IRISH)

In contrast to a density maintenance approach, such as CCP, node activity under IRISH is governed by varying the tolerable interpolation error threshold. The effect of the choice of error value on node activity is depicted in figure 5. A number of interesting points can be seen from this graph, firstly, node activity decreases as the interpolation error increases. A second point of interest is that this decrease is bounded as can be seem from the small reduction in node activity given the relatively large increase in interpolation error from 5 to 10. The diminishing node activity from figure 5 is mirrored in an increasing average node lifetime as the threshold increases, figure 6. Once again, the trends level off at around 160 seconds per node, indicating that a similar, albeit lower, asymptotic value exists for the average node lifetime as in the previous set of experiments.

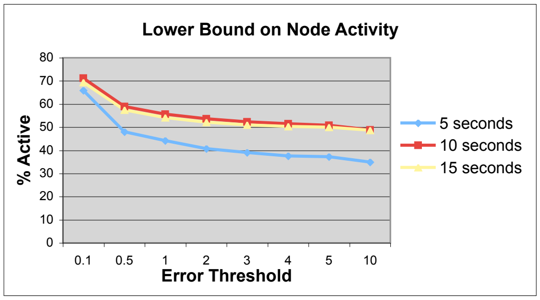

Fig. 5. Node activity as the IRISH error threshold increases. Three different timing regimes are depicted here, where the sleep time is set to 5, 10 and 15 seconds.

The reason for the levelling off of both the average % active and average node lifetime graphs is the underlying connectivity maintenance VSN, which maintains the core set of nodes for the network to route data to the base station. Interestingly, the correlation between activity and lifetime is not present here. Lower node activity translates to reduced node lifetime due to the short sleep period. This illustrates the point that an appropriate timing regime for each VSN is crucial to ensure optimal performance. In order to address the drawbacks of the layered VSN approach we have developed an architecture based on a MAS which, enhances flexibility, removes the possibility of transmission through hibernating layers and facilitates intelligent power management on the nodes. As agents underpin our solution, we examine existing approaches to MAS deployment on WSNs in the next section.

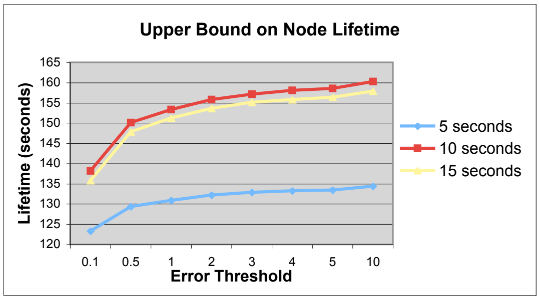

Fig. 6. Average node lifetime as the IRISH error threshold increases. Three different timing regimes are depicted here, where the sleep time is set to 5, 10 and 15 seconds.

V. MASs AND WSNs

Agents have been deployed previously on WSNs primarily to process the raw data in an intelligent fashion with a view to reducing transmissions - the single biggest factor in determining the longevity of the network [13]. Other applications have used agents for the routing of packets around the network [1]. Agents have also been used in lighting control for intelligent energy conservation [14]. The application of agent technologies to the field of WSNs has been increasing steadily over the last few years, in particular, the underlying agent frameworks for WSNs.

Agilla [4] is a middleware platform for deploying mobile agents, essentially mobile code. The agent architecture is tailored toward the computational constraints typical of WSN nodes. It allows for multiple agents to exist on a single sensing node, and provides methods for the reliable movement of agents between nodes. Mate [9] allows WSN programs to be written in TinyScript, a scripting language which is compiled into executable bytecodes for an application-specific virtual machine. Allowing the virtual machine to be application-specific means that the agents can be clear and concise and thus less prone to failure, but this approach reduces compatibility of agents across different platforms.

The ongoing improvement in WSN node technology has lead to the emergence of JavaME enabled devices such as the SunSpot running the SQUAWK JVM [15]. Such developments pave the way for the porting of existing Java based agent environments to the field of WSNs. For example, AgentFactory already has a version, AgentFactory MicroEdition (AFME) [10], capable of running on the SunSpot WSN node. This means multiple BDI style agents, along with agent mobility across heterogeneous devices within the network, can be deployed.

AFME delivers support for the creation of BDI agents that follow a *sense-deliberate-act* cycle. The control algorithm performs four functions. Preceptors are fired and beliefs are resolved within the belief resolution function. The beliefs are used within the deliberation process to identify the agent's desired states. WSN agents are resource bounded and will be unable to achieve all of their desires even if their desires are consistent. Therefore, a subset is chosen within the *intention selection process*, that maximises their self-interest with respect to their finite resources. The problem of determining the subset of commitments that maximises utility with respect to resource limitations is a classic 0-1 knapsack problem [3]. The final function of the control algorithm concerns commitment management. Depending on the nature of the commitments adopted, various actuators are fired. The AFME wireless migration service uses both the Sun SPOT radiogram protocol for topology discovery and the Radiostream protocol for reliable migration. Quantifying the exact energy consumption of such reasoners is part of our ongoing work.

Tynan et al. [19] propose a deployment methodology for a MAS on a WSN. Three distinct phases can transform a central base station implementation into a MAS of AFME agents distributed on the nodes of the network. A case study and some tool support for the methodology are also presented. In the next section we demonstrate how VSNs can be realised using a Multi-Agent System. This alternative solution mitigates some of the disadvantages of the layered based approach also allows intelligent power management to be conducted in order to conserve the limited energy resources of the network.

VI. AGENT BASED VSN ARCHITECTURE

The three primary disadvantages of the previous layered VSN approach can be summarised as: individual layer hibernation, homogenous decision periods and flexibility. Our MAS approach attempts to address all three and the architecture of our solution is depicted in figure 7. The WSN node acts as a host for the agents that will maintain the VSNs with one or more agents performing the maintenance. Membership of a VSN will be designated by the agent and the clients of the agents will interact with them to determine how best to act. For instance, the routing component will now communicate with the Connectivity VSN agent to determine if it should beacon for example.

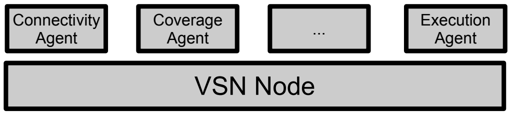

Fig. 7. Agents maintain their own individual VSNs on the node. The agents send/receive messages to and from their counterparts on neighbouring nodes.

The agents can broadcast and receive beacons to and from their counterparts on neighbouring nodes. They can interact with neighbours to deliver the appropriate VSN for the application or service that will utilise their resulting topology. The peer level interaction that maintains the VSNs is depicted in figure 8. Crucially, co-operation between agents across nodes is not the only interaction that can occur, agents can co-ordinate locally in order to optimise performance. Using this architecture, the first two issues with the layer-based approach are alleviated.

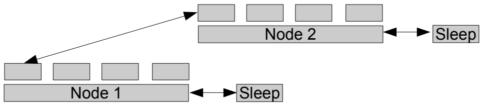

Fig. 8. Agents located on the WSN nodes communicate with their peer agents in order to maintain the required VSN.

Agents can now freely hibernate without the concern of another agent having to relay messages through it. Each agent will have access to the MAC and routing components on the WSN node. While the node may not be part of the connectivity VSN, it may be critical from a sensing coverage perspective and using this architecture the connectivity VSN agent can hibernate while sensing data can still reach its required destination. This strategy also alleviates the problem of homogenous timing criteria. Since the agents can operate autonomously, they can select a timing regime for optimal performance and this can be crucial in determining the level of service provided by the VSN [18].

The final issue previously identified with the layered approach is lack of flexibility, and an architecture built on a MAS offers an elegant solution to this. The WSN can be automatically adapted with a new VSN by placing a new agent on the node that registers it's redundancy state and sleep time with the Sleep Broker, which is discussed next. The location of the new layer in the stack does not have to be selected and surrounding protocols do not have to be interfered with in order to splice in the new layer. The agent can either operate in isolation or in co-operation with other local agents in the latter case some modifications may have to be made to the agents already on the node. Agent mobility also offers a convenient way of deploying the new VSN. The new agent can migrate and clone itself onto the network to produce a new VSN that clients can automatically utilise.

A. Agent Based VSN Power Management

The question of power management within the context of VSNs centres around the issue of when a node is not part of any VSN and for how long will this be the case. When this can be established, then the node could opportunistically hibernate itself until it was required to be active. The strategy adopted here involves the agents informing a Sleep Broker of its redundancy or criticality state and how long this will persist for, figure 8. Consider the case where both the sensing and connectivity VSN agents deem the node to be redundant. In which case all their clients will not require any use of the processing or transmission resources. If each agent informs the broker how long this will last for, then the broker can hibernate the node for the shortest period. This ensures that when the agent subsequently evaluates the node for membership of it's VSN, the hardware will be back online. When any new VSN agents are deployed on the WSN and they inform the local hibernation broker of the hibernation duration, the system will be able to adaptively hibernate the node for the optimal duration to ensure all layers timing regimes are satisfied.

The MAS approach detailed here allows for power management to be conducted at a finer level than simply a binary sleep-active decision. Existing work has shown that it can be beneficial for some components of the node to be hibernated rather than a node directly switching off [16]. A node can go through a phased shutdown procedure, depending on the requirements of the active VSN agents hosted on it. For instance an active Execution VSN may only require the memory and CPU components to remain active while the sensor suite and transceiver can be switched off until they are required again by one of the other agents. In this scenario, agents specify the redundancy state, the timing regime and the resources required. The Sleep Broker on the node can then make decisions about the activity of the components. Therefore, it may not be a clear decision to activate or hibernate a node, but in fact degrees of hibernation and activity can be selected adaptively depending on the redundancy states of its hosted VSN agents.

VII. CONCLUSION AND FUTURE WORK

As WSN resources increase due to developments in node technology, future deployments will have capabilities that far exceed the requirements of any one single application. It will be vital therefore that these resources are delimited so that multiple applications can be hosted on the same WSN. Typically resource partitioning will take the form of a specified subset of nodes, which must remain active in order to adaptively fulfil the requirements of a given application or service. As nodes fail or exhaust their power supplies, additional nodes must be brought online to take over from those nodes. Many techniques have been developed with such adaptivity in mind e.g. CCP and SPAN. In essence these techniques establish a software maintained virtual deployment of nodes termed a VSN.

In this work we demonstrated some of the disadvantages of using a layered solution to deliver VSN behaviour with power management, namely inflexibility, homogeneous timing constraints and co-operation with hibernating layers. Motivated by this, we developed a solution underpinned by a Multi Agent System with an agent maintaining the required VSN participation. A hibernation broker examines the sleep times of the agents and can hibernate some or all of the components of a node depending on the requirements of the active VSNs. In order to accurately test the MAS solution, we have integrated J-Sim with AFME and are re-implementing CCP and IRISH as agents. We are also integrating the J-Sim power model with AFME to ensure an accurate model of the cost of agent deliberation can be factored into our future experimentation.

REFERENCES

[1] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *WSNA*, pages 22–31, September 2002.

[2] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 85–96. ACM Press, 2001.

[3] B. Dantzig. Discrete Variable Extremum Problems. *Operations Research*, 5:266–277, 1957.

[4] C.-L. Fok, G.-C. Roman, and C. Lu. Mobile agent middleware for sensor networks: An application case study. In *In Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'05)*, 2005.

[5] J. Hill. *System Architecture for Wireless Sensor Networks*. PhD thesis, UC Berkeley, 2003.

[6] A. P. Jayasumana, Q. Han, and T. H. Illangasekare. Virtual sensor networks - a resource efficient approach for concurrent applications. In *ITNG '07: Proceedings of the International Conference on Information Technology*, pages 111–115, Washington, DC, USA, 2007. IEEE Computer Society.

[7] B. Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000. ACM Press.

[8] S. Lei, H. Xu, W. Xiaoling, Z. Lin, J. Cho, and S. Lee. Vip bridge: Integrating several sensor networks into one virtual sensor network. *icisp*, 0:2, 2006.

[9] P. Levis and D. Culler. Mate : A virtual machine for tiny networked sensors. In *Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2002.

[10] C. Muldoon, G. M. P. O'Hare, and J. F. Bradley. Towards reflective mobile agents for resource-constrained mobile devices. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–3, New York, NY, USA, 2007. ACM.

[11] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*. IEEE Computer Society, 1999.

[12] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107, New York, NY, USA, 2004. ACM Press.

[13] H. Qi, X. Wang, S. S. Iyengar, and K. Chakrabarty. Multisensor data fusion in distributed sensornetworks using mobile agents. In *4th International Conference Information Fusion*, 2001.

[14] J. Sandhu, A. Agogino, and A. Agogino. Wireless sensor networks for commercial lighting control: Decision making with multi-agent systems. In *AAAI Workshop on Sensor Networks*, 2004.

[15] D. Simon and C. Cifuentes. The squawk virtual machine: Java on the bare metal. In *OOPSLA '05: Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 150–151, New York, NY, USA, 2005. ACM Press.

[16] A. Sinha and A. Chandrakasan. Dynamic power management in wireless sensor networks. *IEEE Design and Test of Computers*, 18(2):62–74, March/April 2001.

[17] A. Sobeih, W.-P. Chen, J. C. Hou, L.-C. Kung, N. Li, H. Lim, H.-Y. Tyan, and H. Zhang. J-sim: A simulation environment for wireless sensor networks. *anss*, 00:175–187, 2005.

[18] R. Tynan, G. M. P. O'Hare, and A. Ruzzelli. Signal based node activation in wireless sensor networks. In *AVSS '06: Proceedings of the IEEE International Conference on Video and Signal Based Surveillance*, page 47, Sydney, Australia, 2006. IEEE Computer Society.

[19] R. Tynan, A. G. Ruzzelli, and G. O'Hare. A methodology for the development of multi-agent systems on wireless sensor networks. In *Agent-Oriented Software Development Methodology (AOSDM'2005) Workshop*. 17th International Conference on Software Engineering and Knowledge Engineering (SEKE'05), 2005.

[20] L. Wang and Y. Xiao. A survey of energy-efficient scheduling mechanisms in sensor networks. *Mobile Networks and Applications*, 11(5):723–740, 2006.

[21] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 1(1):36–72, August 2005.

[22] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 70–84, New York, NY, USA, July 16-21 2001. ACM Press.