Context-Aware Sensors and Data Muling

Suan Khai Chong [1], Ian McCauley [2], Seng Wai Loke [3], and Shonali Krishnaswamy [1]

[1]School of Computer Science and Software Engineering
900 Dandenong Rd, Monash University, Caulfield Campus, Australia.
{Suan.Chong,Shonali.Krishnaswamy}@infotech.monash.edu.au,
[2]Analytical Services Development Biology PIRVic,
Attwood, 475 Mickleham Rd, Attwood 3049, Australia.
{Ian.McCauley}@dpi.vic.gov.au,
[3]215 Franklin Street, Melbourne, VIC 3000,
Latrobe University, Australia.
{S.Loke}@latrobe.edu.au,

Abstract. Efficient collection of data represents one of the key challenges for sparsely deployed wireless sensor networks, due to issues that include heavily imposing particular nodes to relay sampled data and also draining sensor energy due to possible transmissions over long distances. A possible solution is to use existing data carriers, also known as data mules, in the environment. Though the application is often that of delay-tolerant networks, the technique is cost-efficient as it maximizes the use of existing network nodes. The domain of our interest is random mobility, where certain sensor nodes move with a random velocity as observed in real-life application scenarios. We believe that the use of data mules for real-life sensor networks require a unique solution. Thus, in this paper, we propose the use of a context-aware framework for data muling involving randomly mobile data mules. As a comparison, we have implemented a neighbour detection protocol for data muling and contrast it with an implementation that uses a context-aware approach with RFID sensors. The experimental results show that the use of contextual information derived from RFID sensors allows coordination of more reliable transmissions to be achieved in minimal time.

1 Introduction

Ongoing efforts in the development of radio transceivers and integrated circuits have enabled the production of small and affordable sensor devices that can exchange data in wireless environments. Past and present applications of sensor networks have included deployments that monitor for interesting phenomena in the environment including habitat monitoring on remote islands [17], monitoring weather conditions in vineyards [20], environmental monitoring in Antarctica [6], etc. Primarily, the aim of these applications is to relay sampled readings from the sensors to nearby base-station(s) through ad-hoc multihopping of data between sensors. However, for sparsely deployed networks, this implies costly data transmissions over long instances and load imbalance on sensor nodes responsible for routing data packets.

As a solution, various researchers have proposed mobility when data is not immediately required in real time. The notion of using mobility is that mobile entities, or

`Data Mules`, can be used to gather data from static nodes when they come into range and deliver the data gathered to a central station. This also allows us to recharge data mules when they return to the central station and eliminate the need to use intermediate nodes to relay data. In retrospect, research in the area of mobility has mainly focused on controlled mobile entities with only some work on using mobile entities with random motion.

On the other hand, real life applications manipulate existing carriers in the environment as data mules such as animals and cars. As an example, in [20], data mule motes are mounted onto spades so that when workers use spades in the vineyard, data would be transmitted from static motes in the vineyard to the mules on spades when both are in physical proximity. Another application is a heterogeneous sensor network deployed underwater [27] where an AUV is used as a data mule to collect readings from underwater sensors. The AUV then uses a camera mounted on it to detect the location of nodes and upon detection, stops and collects the data. In these scenarios, external environmental information affects how data mules are to be implemented in a sensor network application. Here, we believe that such information, either formed by external environmental conditions or later gathered internally by sensors, create contextual information that could trigger sensors themselves to conserve energy in their sensing operations.

In this paper, we examine the use of contextual information to form useful triggers for data mules by utilising a context-aware framework. As experimentation, we initially propose a basic muling approach for data muling in an envisaged pig shed environment and compare the method to using our context-aware framework for muling when additional context information is available. An instance of the additional context that we have used is location context by RFID sensors that detect incoming data mules and provide information when mules are in range. We discover through our observations that, for a data muling sensor network application, several areas of the application would be enhanced through the use of context-aware monitoring sensors that will eliminate detection costs and associated transmission costs between sensors. Further investigations of our context-aware approach in data muling also show that we can achieve energy savings by eliminating repeated acknowledgements sent between the sender and receiver. The rest of the paper is organised in the following way. In section 2, we present an overview of past and current research in relation to our work. The model of our framework is presented in section 3. In section 4, we describe the basic muling and context-aware implementations for a data muling application and compare the costs involved in the two implementations. Finally, we conclude our studies in section 5.

2 Related Work

A key challenge in wireless sensor networks is to conserve energy due to limited battery energy in sensors and the difficulty of a battery replacement in hard-to-reach terrains where they might be deployed in. As indicated by performance measurements in [10], the main resource overhead lies in radio communication of sensor readings between sensors or from sensors to a central base station. In the case of sparsely deployed sensors in WSN applications, this further stresses some sensor nodes to relay data to neighbouring nodes that are distant. As shown in [9], mobility can be an important primitive to optimise network communications. In the same way, the lifespan of a sensor network would also be significantly improved if we can use mobility to reduce the amount of data relaying necessary.

One form of mobility is controlled mobility with mobile entities such as robots, granting us the benefit of control over the reliability of data transfer through motion control and the ability to establish shorter data routes. Several studies have evidenced the advantages of using mobile entities in the sensor network domain. For instance, in a study by [24], the authors discussed a network infrastructure based on controllably mobile nodes. The implementation involves the use of a mobile base station that moves in a near fixed path, collecting data from cluster heads formed from embedded static nodes. The implemented system prototype utilises the infrastructure to improve the energy performance for battery constrained sensors. Prior to this work, [25] has suggested scheduling of mobile entities for efficient data collection in WSNs to visit sensor nodes before their buffers are full. Controlled mobility has also been examined in the area of ad-hoc networks, such as [29] and [16]. In [29], they addressed the issue of efficient data delivery in sparse mobile ad hoc networks. The technique is termed by the authors as Message Ferrying, in which special mobile nodes moving in a non-random fashion are used to carry data for nodes in the network. It exploits the non-randomness to provide physical connectivity among nodes. Also, in [16] we note the use of mobile hosts that are able to modify their trajectories actively to transmit messages in order to transmit messages in disconnected ad-hoc networks.

In contrast, we are interested in the applicability of mobile entities for real-life sensor applications where data mules would be carriers that already exist in the environment. These entities can be medical wearable sensors for humans, sensors mounted on pigs or mounted on vehicles. In this environment, several additional challenges are present due to issues such as changing speeds of muling entities and unforeseeable mule arrival times. Predictability of mule path is also another issue and has been studied in [3].

The concept of using random mobile entities has been studied by [10], [12], [16] with applications as shown in [23], [14] in the domain of ad-hoc networks. In [10], the authors demonstrated the use of intermediate relay nodes to carry data between source and destination so as to maximise the throughput in the network. Moreover, programmed/unrealistic movement models as stated above may affect the real performance of protocols, for instance, the reliability of data transfer. Through simulations, [12] has shown that the mobility model has a significant effect on the routing protocol employed. The concept of random mobility have been explored in several ad-hoc networks systems, as examples, [23] that uses whales as the mobile nodes in the network where data is replicated and spread as whales surface; and in [14], the use of tracking collars on zebras and the use of peer-to-peer network techniques to forward data to mobile base stations.

Extending existing ad-hoc routing protocols to sensor networks, however, presents further challenges due to different requirements in terms of the allowable bandwidth and network scalability [2]. Sensor networks also share a different energy optimisation goal because for sensor networks, sensor network lifetime optimisation is also concerned with duty-cycling sensor nodes [22]. Recent researchers that have studied random mobility for sensor networks include [11], [13] and [1]. In [11], the authors proposed DataMULE, a three-tier architecture for collecting data in sparse sensor networks. The DataMULE architecture uses mobile entities present in the environment to transport data from sensor nodes to access points. The primary aim of the architecture is to achieve energy savings in sensor nodes by using mobile entities with short range radios as low power transport medium for sensor data. The concept of using multiple data mules as transport is detailed in [13] with simulation results. In another study, [1] explored data muling with an experimental test beds using mica2 motes and reported results on the effect of moving speed of mobile entities in relation to muling performance.

Our initial implementation of a data muling application follows closely the study in [1]. Nevertheless, while the aim of [1] is to evaluate the performance of a data MULE model based on the architecture in [11], we aim to further study and detail possible experimental challenges of data muling in an envisaged real-life scenario. The observations from the experiments prompted the idea of harnessing *context-awareness* into a data-muling application. Particularly, we are motivated by existing work and interesting uses of contextual information in other applications. Discussed in [19], different data management issues are involved in this context management process, from collection to processing of data, these issues include storing abundant transient sensor data, processing data in real time and obtaining useful knowledge from the data processed. Analysis of the data would then yield contextual information where *context*[7] in our work generalises sensor data derived from sensors. Context could then be used to provide different forms of services to the user, as relevant to the current task [8]. Some examples of the context-aware applications can be found in [28, 15, 21, 4].

Our approach involves obtaining relevant context and using it to assist a mule in data collection. In the next section, we describe our context-awareness methodology and our context-aware framework for muling.

3 Context-Aware Sensors

3.1 Sensor Roles

We manipulate relevant contextual information in a sensor network application to control the operations of sensors in a data muling application. We consider that, for sensors in any heterogeneous sensor network, sensors would have different computational power and radio range, allowing them to perform a multitude of roles within a sensor network. Figure 1 shows the partitioning of the roles a sensor can perform.

In this partitioning, we note that certain sensors can be used solely for the purpose of providing useful contextual input to the system (i.e. β), some sensors for monitoring which can be controlled by context (i.e. α) and others that can provide both functionality (i.e. $\alpha + \beta$). For instance, in a pig shed, monitoring sensors such as cameras that track animals coming in and out of the shed will initiate how often temperature sensors are to relay temperature conditions in the shed, i.e. maybe only when pigs are in shed. Sensors that provide this contextual input is thus,*context-aware*. In the next section, we

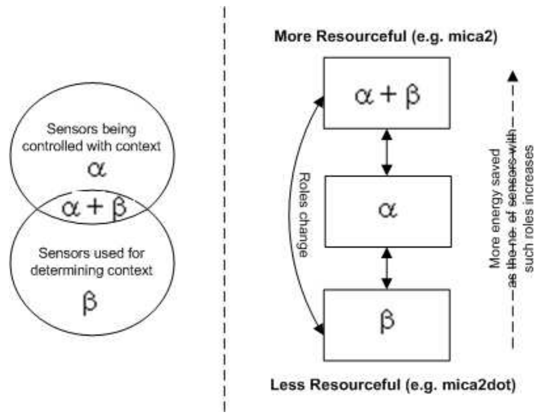

Fig. 1. Sensor Monitoring Roles

describe quantitatively, the different forms of contextual input that can be manipulated in a sensor network.

3.2 Contextual Input

Drawing upon definitions of context in [4] and [21], context in sensor networks can broadly be classified into five categories: (i) **Computing context:** information describing the computing status of sensors such as network connectivity, communication costs and remaining battery power. (ii) **Sensor context:** Relative to sensors, this form of context refers to the sensors' profile, such as the group they belong to, their location in a sensor network and a common situation they face (e.g. weather is hot). (iii) **Physical Context:** Physical Context refers to external conditions that can be measured by sensors, for instance, the lighting of a room, the temperature and the sound levels of surroundings. (iv) **Time Context:** time would refer to the time of day, week or month when sensor readings are obtained and which further describes other sensor context. (v) **Historical Context:** in most cases, a history of sensor readings accumulated over a certain time span could describe additional information about the current situation being sensed. Coupled with information about readings of surrounding sensors, this yields historical context, which can be used to predict future sensor readings. The different types of contexts described above, be it a derivation from single raw sensor readings or collections past sensor readings can be further classified into two main levels of context. In line with the observation by Dey in [7], they are: (a) **Primary context:** Basic information that answers directly what is sensed from the environment (e.g. temperature and date), which can be used to deduce further context information. (b) **Secondary context:** Information derived from primary context (e.g. temperature and date could describe weather information)

3.3 A Context-Aware Framework for Data Muling

We first introduced a context-aware framework to conserve energy in wireless sensor networks in [5]. To apply this framework to data muling, we view the RFID detection of mules as a form of contextual input to the framework, and signal events sent to mules/sensors to initiate transmissions as the contextual triggers in this framework. The diagram below depicts our modified context-aware framework for the data-muling application scenario:

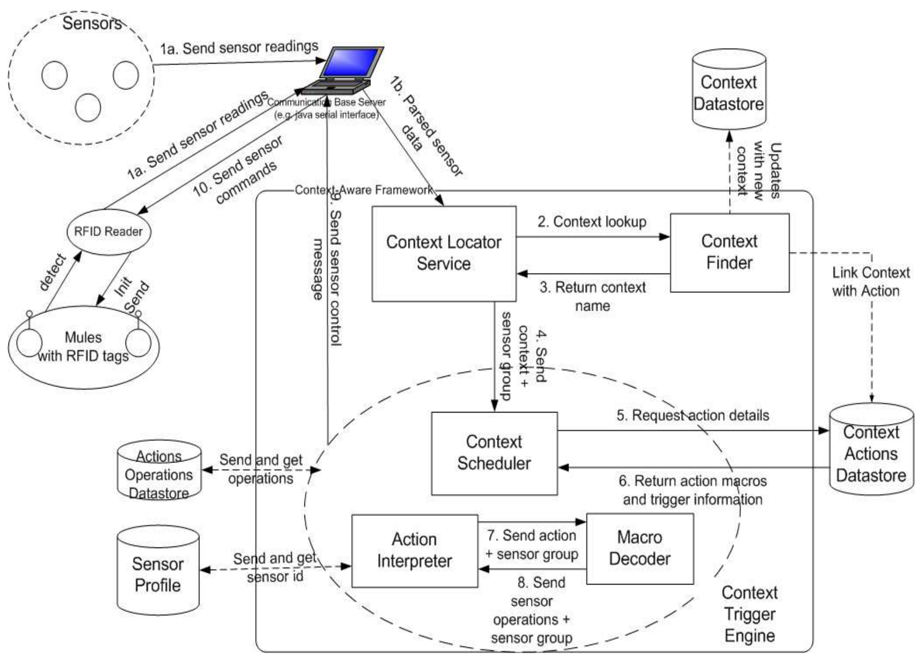

Fig. 2. Context Aware Muling Framework

Summarising our earlier paper, the framework is made up of various components from the collection of contextual information to application of triggers to sensors. Instances of this framework would run on a centralised system whereby data would be processed. As an overview, the **Communication Server** is responsible for the receiving of raw readings from sensors and relaying the messages to the application. Control messages are also sent from this server. In the diagram, this module is isolated from the framework because different sensors require a different communication interface (for instance, SerialForwarder in java under TinyOS). The **Context Locator Service** and **Context Trigger Engine** are the key components in the framework. The function of the Context Locator is to abstract raw sensor readings into context labels that can be internally recognised within the system. A method for discovering the context label is through if-else condition rules. Otherwise, the system can try to learn from data-readings, and discover patterns, which could potentially derive a particular context we are interested in, for instance, historical context from past readings of sensors.

When a valid context is present (when there is at least one context to action mapping), the **Context Trigger Engine** checks the data stores to determine the sensor operations that the system has to trigger based on the context given and the profiles of the sensors in the database. To make this approach generic for controlling sensors, we can employ the use of action macros on sensors, which we define as follows using a form of BNF rules:

```
S -> M
M -> M + M | OP + M | OP
OP -> OP + OP | instruction
```

where M represents the set of macros, conditional that every macro cannot be defined in terms of itself, and where instruction is the actual command sent to sensors. For instance,

```
REDUCE_TEMP_GRP_1 -> CHANGE_TEMP_RATE_100_a
                   + CHANGE_TEMP_RATE_100_b
                   + REDUCE_TEMP_GRP_0
```

where the `REDUCE` macro translates to `CHANGE_TEMP` operations and another `REDUCE` macro (but on a different group of sensors). A sensor can receive such a macro command and then perform operations as described by the macro.

The representation can be done in an XML language. For example, for the context `mule_in_range`:

```
<context label="mule_in_range"
 sensor_group="mule" detected_time="3mins">
  <title>Signal</title>
  <macro name="signal_sensors">
    <operation> SEND_MULE_ID</operations>
    <macro name="signal_mules">
    </macro>
  </macro>
</context>
```

Thie above rule states that when the mule is in range, use the `signal_sensors` macro, which in turn initiates the `signal_mules` macro.

4 Implementation and Issues

To evaluate our context-aware approach to muling, we envisage a pig farm scenario and outline the process of muling in this scenario with flow diagrams. We first perform a nearest neighbour basic muling approach on this scenario and later, compare our results with a context-aware approach.

4.1 Application Scenario

Consider the AWSN test system comprising a group of pigs housed individually in an experimental pig facility [18] to measure the effect of external stressors in a shed(e.g. temperature) on the core body temperature of pigs. The setting consists of Mica2Dot wireless sensors mounted on four pigs to measure each pig's core temperature below the skin and a combination of Mica2/Mica2dot motes in the shed to measure internal shed temperatures. Environment sensors are placed statically in the shed and supports a multi-hop network. Nevertheless, as stated by the authors, in such an environment, the network quality is often quite poor. In deployment, though a base station was placed centrally in the shed, the range of reception at high power was only a few meters and often suffered packet loss due to noise in the shed and mobility of the pigs.

In this scenario, we consider the use of data mules in this sensor network to improve network performance by maximising the use of existing mica2dots mounted on pigs. In the diagram below, the scenario depicts mules A, B, C that are able to move randomly within the shed to collect data from static sensors when they are in close proximity and offload data when they are outside the shed (Refer Figure 3).

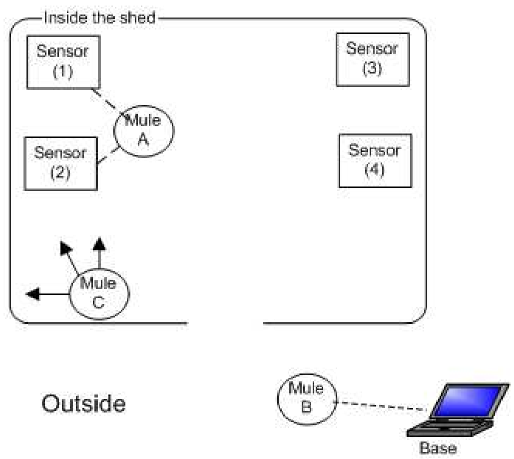

Fig. 3. Muling Scenario

Theoretically, a data muling approach in this scenario allows us to minimise in-shed communication by having pigs save data into internal flash memory in motes and offloading the data outside the farm where communication is more reliable. Additionally, by using the pigs to transport data from environmental sensors at the same time as they move out of the shed, we hope to save communication costs (i.e., use the mules to physically transport data over the distance). The trade-off of this technique is the delay in receiving the data but we foresee the possibility of using data muling at least to some extent. For instance, pigs might carry only some history data from static environment sensors.

4.2 A Basic Muling Approach - Sensors “Polling” for the Mule

Modelling the scenario we proposed above, we configured a base mote connected to a laptop running MacOSX on a/c power source, mica2dots as the mules with limited battery power, and static mica2s as the sensors that regularly sample temperature and light readings in the environment, running on two AA batteries. The motes are programmed in nesC under the tinyOS[26] operating system, with SerialForwarder running at the base station to provide the serial interface. We apply a basic nearest neighbour detection technique. The decision flows for the mule, sensor and base (station) for this technique are in Figures 4, 5 and 6.

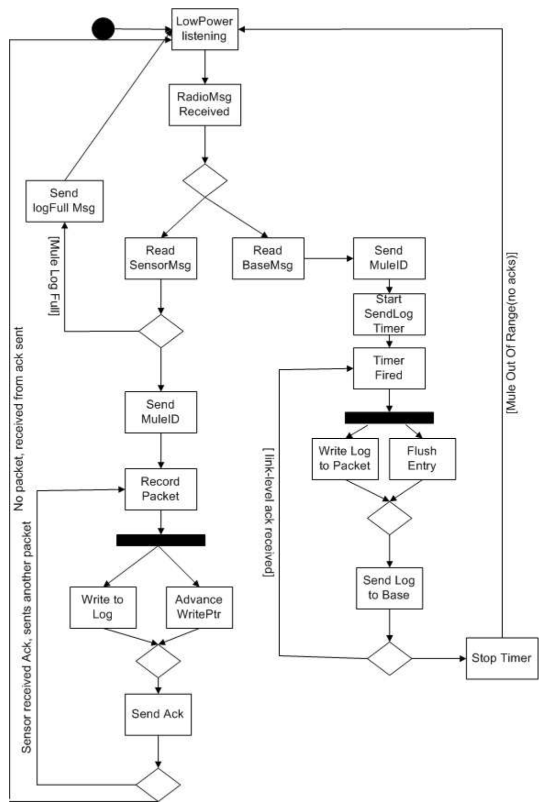

Fig. 4. Mule Decision Flow Diagram

In the flow diagrams, packets are sent by the sensor once mules are in range, i.e. when acks can be received from the mule by the sensor. Note that even when this starts to happen, the communication between the mule and sensor might not be stable enough for reliable transmission of logged readings (i.e., the application data). The mule, in fact, has to come within a “safe distance” of the sensor for data transmission to be reliable.

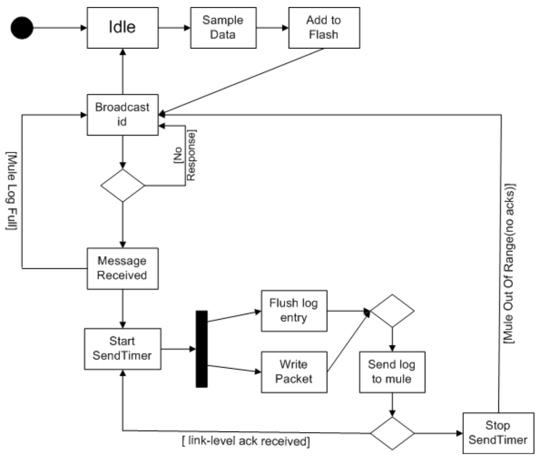

Fig. 5. Sensor Decision Flow Diagram

Due to the isotropic nature of the radio range, we use link-level acks provided by tinyOS to determine if a sensor/base is still at range during packet transmission. Although not used due to radio communication costs, one way to establish a reliable communication is for a sensor/basestation to send repeated acks when a mule is detected so that we reduce the amount of packets loss when a mule falls out of range. For instance, if we can receive 8 acks out of 10 that have been sent, then we have a higher confidence that the mule is within range. The sensors (with the data or logged readings) are, in effect, polling to determine if the mule is there and whether the mule is near enough. This can be resource wasting.

In this implementation, data mules are only responsible for collecting data individually. We avoid duplicate copies of the log to be sent by only allowing exclusive send/receive transmissions between a single sensor and mule. In other words, a sensor will only process one mule at any time. This implementation saves us from the complexities of manipulating multiple read/write pointers to the log data in the case of servicing multiple motes. Another possibility is manipulate mule to mule communication so that data can be multi-hopped between mules and allowing, packets to reach the base-station sooner.

To conserve network radio energy, static sensors can also alternate between packet broadcasting and sleep mode, while mules are in constant low-power listening mode. The basestation, on the other hand, will be constantly broadcasting signals. We can also operate the static sensors and the mules at different sleep cycles to more efficiently conserve energy, for instance, sleep sensors at 5 seconds intervals and mules at every 2 seconds intervals, but the adjustment of the sleep cycles would be application-specific and dependent on the mobility of the data mules, and so, hard to manually fine-tune.

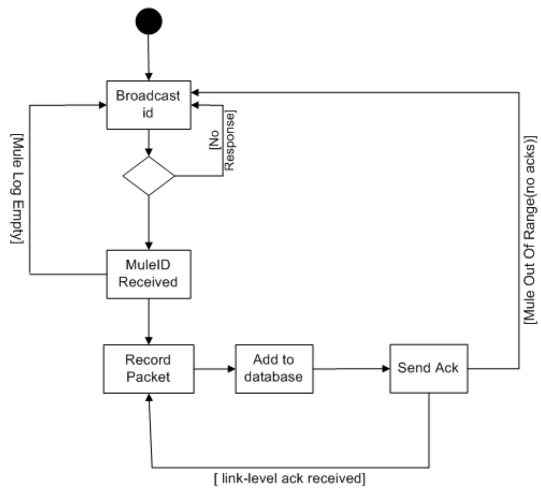

Fig. 6. BaseStation Decision Flow Diagram

4.3 Context Aware Muling Approach

With reference to the pig farm scenario above, we now model the sensors (to be data muled) as "the sensors that are to be controlled by context triggers," additional RFID sensors as "sensors that can provide the contextual information for control" (about the whereabouts or proximity of the mule in this case) whereas other sensors in the farm can provide both contextual information and be controlled. We illustrate this arrangement for the context-aware application in the farm environment as below (Refer Figure 7):

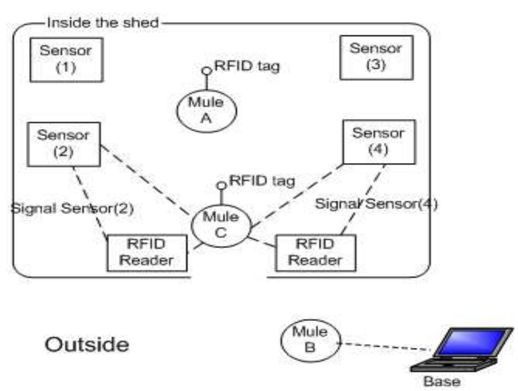

Fig. 7. Context Aware Muling Scenario

Our context-aware approach differs from the earlier approach as contextual information provided by the RFID readers is used to trigger mule detection. The RFID readers are also connected to a PC that launches the context-aware application (or CAP, for short) that, based on its preprogrammed rules, can send appropriate macro commands to sensors (designated to be controlled) depending on contextual input received (from another group of sensors designated as those providing such contextual information). In the given scenario, with RFID tags attached to the data mules, once mule C enters the shed, the RFID readers located at the entrance of the shed will detect mule C on entry and send the detection information to the context-aware application. Signals will then be sent from the PC to sensor 2, to initiate and establish a data transfer connection with mule C. Basically, sensor 2 will only send logged readings to mule C upon a trigger sent from the base node. After mule C is in safe distance of sensor 2, mule C will receive data from sensor 2 and send acknowledgements to sensor 2. Mule C remains in listening mode if there are no more packets from sensor 2. When mule C leaves the shed again, the context-aware application sends a trigger to sensor 2, stopping the communication. Note that, in this approach, the sensors don't need to "poll" for the mule but is told when they are near enough by the CAP.

4.4 Results and Evaluation

In this section, we report our results in using the nearest neighbour approach and our results from applying the context-aware framework in this scenario with RFID sensors. As a performance measure, for both experiments, we note the number of packets sent from the sensor, the number of packets that have been received at the mule and the distance between the two sensor nodes. We measure the packets lost relative to the distance between a single sensor and mule.

In both experiments, we assume a static sensor whereby the mule moves in the direction of the sensor. As the mule approaches the sensor, we then note the distance from the sensor when it is first detected, the number of packets that has been received by the mule since the first detection until the packet loss is minimal (i.e., when readings stabilise and the mule synchronises with the sensor) and the total number of packets received with minimal packet loss. Three sets of readings are collected. For the context-aware approach, the experiment is carried out in the same way but we set a safe distance of 78cm where we place the RFID reader. The logged readings and summarised results are shown in tables 1 and 2.

	Basic Muling Experiments		
Result Sets	A	B	C
Distance in cm, 1st detection	143	157	153
Packets Sent, unstable	46	46	25
Packets Received, unstable	44	42	24
Safe Distance in cm	88	105	99
Packets Sent, stable	41	48	38
Packets Received, stable	40	48	38

Table 1. Table of experimental results

	Context-Aware Experiment
Result	Context-Aware
Distance in cm, mule detected	78
Packets Sent, stable	40
Packets Received, stable	40

Table 2. Table of experimental results

In the basic muling experiments, we observe that, while there are only a few packets lost until a more stable connection has been established, some of the packets that have been recorded are erroneous and arrive at an unstable non-uniform rate at the mule. Our context-aware approach addresses this issue by initiating data communication to occur only when the mule is within range of the sensor. This is evident from our results which show that we can achieve zero to minimal packet loss using location context. Also, energy is conserved, since no packets need to be sent by the sensors to "poll" for the mules or to estimate how close the mules are (or whether the mule is within the safe distance for stable transmission of logged readings). And this can be achieved via the CAP notifying the sensors about the proximity of the mule, with only one message from the CAP (and no further overhead from the sensors with the logged readings).

5 Conclusion

In this paper, we have presented a context-aware framework for efficiently collecting data in wireless sensor networks in the context of random mobility. We enforce modularity and generality in our framework so that the different modules are independent, by using different levels of abstraction for data from using action macros to classifying different levels of context within the system. For example, other sensor devices can substitute RFID sensors being used to provide the location context for detection. Our context-aware approach is evaluated through comparisons of packets lost incurred when using our context-aware model to a basic nearest neighbour detection technique for data muling. The observations from the readings obtained from our experiments show that our framework avoids packet loss as we automate the process of mule detection from the use of context triggers whereas, although the nearest neighbour approach allows us to send more readings since first detection, the connection is observed to be unstable and packets are often corrupted. Initial results also suggest that energy is conserved by eliminating the need to broadcast signals continuously by sensors as noted in the nearest neighbour approach. To summarise, our framework provides a generic data-oriented approach that can apply knowledge in any muling application's environment to enhance its operations. As a consequence, this eliminates network costs incurred for sensors to broadcast signals in detection to achieve considerable energy savings. Also, our context-aware approach works even for mules that move without depending on a pre-specified schedule - the nearness of mules (some animal) are detected via CAP (using RFID) and logged readings transmitted opportunistically. Finally, we note that this paper presents only one application of this notion of context-aware sensors (in this case for energy efficient data muling). Figure 2 shows one application of our context-aware sensors framework as specialized to data muling; we are working on further applications of our framework.

References

1. G. Anastasi, M. Conti, E. Gregori, C. Spagoni, and G. Valente. Motes Sensor Networks in Dynamic Scenarios: an Experimental Study for Pervasive Applications in Urban Environments, Institute for Informatics and Telematics, National Research Council (CNR).
2. I.F. Akyildiz, W. Sankarasubramanian, Y. Su, and Cayirci, A Survey on Sensor Networks, *IEEE Communications Magazine*, 2002.
3. A. Chakrabarti, A. Sabharwal, and B. Aazhang, Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks, IPSN 2003, 2003.
4. G. Chen, and D. Kotz, A Survey of Context-Aware Mobile Computing Research, *Technical Report 2000-381*, Department of Computer Science, Dartmouth College, 2000.
5. S.K. Chong, S. Krishnaswamy, and S.W. Loke, A Context-Aware Approach to Conserving Energy in Wireless Sensor Networks, *In Proceedings of the First International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS 2005)*, USA, 2005.
6. Sensor Web in Antarctica: Developing an Intelligent, Autonomous Platform for Locating Biological Flourishes in Cryogenic Environments, *Presented at the 34th Lunar and Planetary Science Conference*, Houston, 2003.
7. A.K. Dey, and G.D. Abowd, Towards a Better Understanding of Context and Context-Awareness, *Technical Report GITGVU -99-22*, Georgia Institute of Technology, College of Computing, 1999.
8. J. Gwizdka, What's in the Context?, *Position Paper for CHI 2000 Workshop on The What, Who, Where, When, Why and How of Context-Awareness*, 2000.
9. D. Goldenberg, J. Lin, A.S. Morse, B. Rosen, and Y.R. Yang, Towards Mobility as a Network Control Primitive, *Proc. ACM MobiHoc*, 2004.
10. M. Grossglauser and D.N.C. Tse. Mobility increases the capacity of ad-hoc wireless networks. *Proceedings of IEEE INFOCOM '00*, pp. 1360-1369, Israel, 2001.
11. S.Jain, R.C. Shah, W.Brunette, W.Brunette, and S.Roy, Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks, *Mobile Networks and Application*, pp. 327-339, Kluwer Academic Publishers, USA, 2006.
12. A. Jardosh, E. Belding-Royer, K. Almeroth, and S. Suri. Towards realistic mobility models for mobile ad hoc networks. *Procedings of the Ninth International Conference on Mobile Computing and Networking (MobiCom)*, San Diego, 2003.
13. D. Jea, A. Somasundara, and M. Srivastava, Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks, *Proc. IEEE International Conference on Distributed Computing in Sensor Systems*, Marina del Rey, USA, 2005 .
14. P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, and D. Rubenstein, Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. *Tenth Intl. Conference on Architectural Support for Programming Languages and Operating Systems*, San Jose, 2002.
15. M. Lamming, and M. Flynn, Forget-me-not: Intimate Computing in Support of Human Memory, *Proceedings of International Symposium on Next Generation Human interface '94*, pp. 150–158, 1994.
16. Q. Li, and D. Rus, Sending Messages to Mobile Users in Disconnected Ad-Hoc Wireless Networks, *Proc. MobiCom*, 2000.

17. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, Wireless Sensor Networks for Habitat Monitoring, *ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta GA, 2002.
18. I. McCauley, B. Matthews, L. Nugent, A. Mather, and J. Simons, Wired Pigs: Ad-Hoc Wireless Sensor Networks in Studies of Animal Welfare, *Second IEEE Workshop on Embedded Networked Sensors*, Sydney, Australia, 2005.
19. G. Mitchell, Data in a Wireless Sensor Network: A View from the Field, *NSF Workshop on Context-Aware Mobile Database Management*, Providence, Rhode Island, 2002.
20. Vineyard Computing: Sensor Networks in Agricultural Production, *PERVASIVE Computing*, 2004, pp. 38-45.
21. S. Pirttikangas, J. Riekki, J. Kaartinen, J. Miettinen, S. Nissila, and J. Roning, Genie of the Net: A New Approach for a Context-Aware Health Club, *Proceedings ECML/PKDD-2001 Workshop on Ubiquitous Data Mining for Mobile and Distributed Environments*, pp. 1–15, Germany, 2001.
22. S. Singh, M. Woo, and C.S. Raghavendra, Power-aware Routing in Mobile Ad Hoc Networks, *Mobicom*, pp. 181-190, ACM Press.
23. T. Small and Z. Haas, The Shared Wireless Infostation Model-A New Ad Hoc Networking Paradigm (or Where there Is a Whale, there Is a Way), *Proc. ACM MobiHoc*, 2003.
24. , A. Somasundara, A. Kansal, D. Jea, D. Estrin, and M. Srivastava, Controllably Mobile Infrastructure for Low Energy Embedded Networks, *IEEE Transactions on Mobile Computing*, Volume. 5, No. 8, 2006.
25. A. Somasundara, R. Ramamoorty, and M. Srivastava, Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines, *Proc. 25th IEEE International Real-Time Systems Symposium (RTSS)*, 2004.
26. *TinyOS: A Component-based OS for the Networked Sensor Regime*. [online] Available from http://www.tinyos.net [Accessed 1st October 2004], UC Berkeley.
27. I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, Data Collection, Storage, and Retrieval with an Underwater Sensor Network, *SenSys '05*, pp. 154–165, California, 2005.
28. R. Want, A. Hopper, V. Falcao, and J. Gibbons, The Active Location System. *ACM Transactions on Information Systems*, **10**, pp. 91–102, 1992.
29. W. Zhao, M. Ammar, and E. Zegura, A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks, *Proc. ACM MobiHoc*, 2004.