

Data Fusion in Wireless Sensor Networks

Hitesh Prajapati
Dept of Computer Science
Lakehead University
Thunderbay, Ontario
Student id : 1114577
hpajap1@lakeheadu.ca

Darshil Prajapati
Dept of Computer Science
Lakehead University
Thunderbay, Ontario
Student id : 1105933
dpajap1@lakeheadu.ca

Abstract—Wireless sensor networks (WSNs) emerge as an active research area in which challenging topics involve energy consumption, routing algorithms, selection of sensors location according to a given premise, robustness, efficiency, and so forth. The network layer deals with routing issues in sensor networks. Since radio transmission and reception consumes large amount of energy, power and data loss are an important factor to be investigated on. If a data packet received is corrupted or not received, which results in packet loss, the main goal of the network is defeated. If we can transmit and receive the information without any loss with optimal power consumption, that system would be considered feasible for studying many aspects in real life by deploying sensor nodes and gathering data. Data loss is thus a key issue in wireless sensor networks. Ongoing research involves designing routing protocols that requires less energy during communication thereby extending the networks lifetime as well as reliable network for efficient communication. Our research has come up with DSDV (Destination Sequenced Distance Vector) routing protocols that can enhance network's capabilities in different aspects.

I. INTRODUCTION

In wireless sensor networks (WSNs), there are already a high number of current problems in which these networks can be applied. Some application fields include tracking, monitoring, surveillance, building automation, military applications, and agriculture, among others. In all cases for the design of any application, one of the main objectives is to keep the WSN alive and functional as long as possible. A key factor in this is the way the network is formed. In fact, the topology is mostly defined based on the application environment and context. The sensor information is usually collected through the available gateways in a given topology. This information is then forwarded to a leader node or to a base station known as sink. A Wireless Sensor Network (WSN) is a distributed network and it comprises a large number of distributed, self-directed, tiny, low powered devices called sensor nodes alias motes. Motes are the small computers, which work collectively to form the networks. Motes are energy efficient, multi-functional wireless device. The necessities for motes in industrial applications are widespread. A group of motes collects the information from the environment to accomplish particular

application objectives. They make links with each other in different configurations to get the maximum performance. Motes communicate with each other using transceivers. In WSN the number of sensor nodes can be in the order of hundreds or even thousands.

These are similar to wireless ad hoc networks in the sense that they rely on wireless connectivity and spontaneous formation of networks so that sensor data can be transported wirelessly. In comparison with sensor networks, Ad Hoc networks will have less number of nodes without any infrastructure. WSNs measure environmental conditions like temperature, sound, pollution levels, humidity, wind, and so on.

II. LITERATURE REVIEW AND BACKGROUND

Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet

A. Authors

(Philo Juang, Hidekazu Oki, Yong Wang , Margaret Martonosi, Li-Shiuan Peh, and Daniel Rubenstein.)

B. Overview

This paper discusses the design trade offs and early experiences in building a low-power wireless system for position tracking of wildlife. By using peer-to-peer net-working techniques, the system can forward data to a researcher's mobile base station without assuming the presence of any cellular phone service or widely-available telecommunications support. They present initial design ideas, measurements, and weight estimates, and discuss how battery and weight limits translate into energy and storage limits for the system and its protocols. Although the protocol development is still very much underway, the early protocol data in the paper provides may be generally useful to the ad-hoc networking and systems communities. It represents new steps in protocols for mobile sensor networks, and offers in-sights into how storage and energy limits may impact protocol design. In particular, by having protocols that well-support nodes of disparate speeds. Finally, their history-based approach currently is stateless (it transfers the information as part of the peer discovery process); they are considering state-based approaches that might

decrease peer discovery time. Overall, ad hoc networking is presently a very active research area. Their work on ZebraNet makes a significant contribution to that domain by offering detailed systems-level perspectives on how to build low-power peer-to-peer systems that operate effectively and are optimized to the characteristics of a particular application domain.

C. Methodology

ZebraNet explores certain issues for sensors that are more coarse-grained than many prior sensor proposals. The larger weight limits and storage budgets allowed the authors to consider different protocols with improved leverage for sparsely-connected, physically-widespread sensors.

1) *Design Goals:* The ZebraNet project is a direct and ongoing collaboration between researchers in experimental computer systems and in wildlife biology. The wildlife biologists have articulated the tracker's overall design goals as:

- GPS position samples taken every three minutes.
- Detailed activity logs taken for 3 minutes every hour
- 1 year of operation without direct human intervention. (That is, we should not count on tranquilizing and recollaring an animal more than once per year.)
- Operation over a wide range (hundreds or thousands of square kilometers) of open lands.
- The authors planned to deploy our system at the Mpala Research Centre in central Kenya
- No fixed base stations, antennas, or cellular service.
- While latency is not critical, a high success rate for eventually delivering all logged data is important.
- For a zebra collar, a weight limit of 3-5 lbs is recommended. Smaller animals may need even lower weight limits.

2) *Tracking Movement Patterns:* Zebra movement can be characterized in terms of three main states: grazing, graze-walking, and fast-moving. Zebras spend most of their time grazing, both day and night. Zebras prefer to graze in areas of short but rapidly-growing grasses. These areas offer high energetic gains and low risks of predation. While grazing on short grass swards, zebras typically exhibit low movement rates and high turning angles. At other times, zebras walk deliberately, with heads lowered, clipping vegetation as they move.

These latter movements are referred to as "graze walking" and are characterized by higher step rates and smaller turning angles than those for focused bouts of grazing. Finally, either due to predators or because an area's vegetation has been exhausted, zebras will occasionally move much more quickly, for longer distances, with their heads raised because they are not grazing. They categorize this as the fast-moving state.

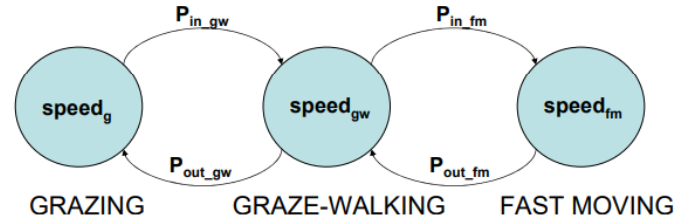


Figure. 1

3) *Collar Design:* This section gives an overview of the tracking collar node design used by the authors. The evaluation board for the GPS - MS 1E (containing a GPS, Flash RAM, and CPU), a short range radio, and a long range radio with its packet modem. The block diagram illustrates the different components and their interactions with one another. To minimize the part count and overall size and weight of the system, they use a single-chip miniature GPS solution from micro Blox : GPS - MS 1E. The GPS - MS 1E is a 12-channel GPS receiver capable of getting a position update every second (though we get them less frequently). It has an integrated 20Mhz Hitachi SH 132-bit microprocessor as well as I/O support. They use the SH1 for data capture and protocol control ; it is the only programmable CPU in the Zebra Net node. The GPS - MS 1E also has a built in 1MB Flash RAM module ; 640K B is available for user data while the rest is used to store the firmware. Using the GPS - MS 1E's microprocessor, they periodically obtain the position coordinates and store them in its on-board flash RAM. The processor also coordinates the communications over the two radios. They chose to use two radios so they can have broad control over trade offs in energy vs. communication range. First, the Linx Technologies SC-PA series is a data radio with a range of only 100 meters but very low power consumption. Second, they use a slow but higher-power data radio and packet modem for longer-range (8km) transfers. The short-range radio is power-efficient for peer transfers when zebras are congregating by water sources, while the longer-range radio is necessary for communicating to the base station over the large area studied with relatively few tracking collars.

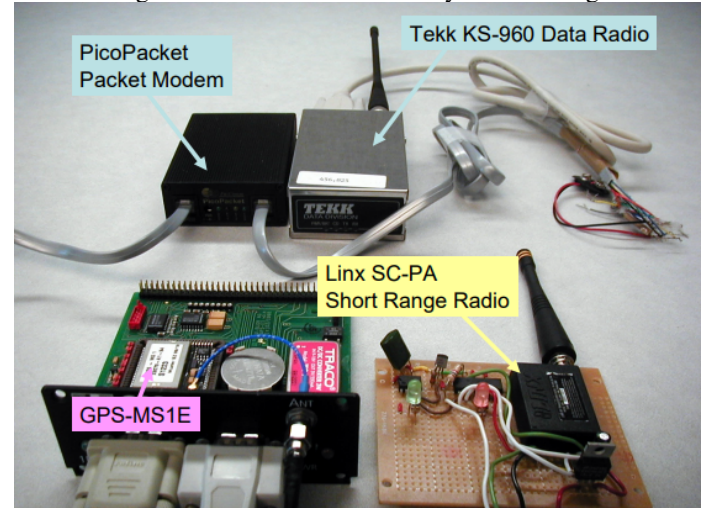


Figure. 2

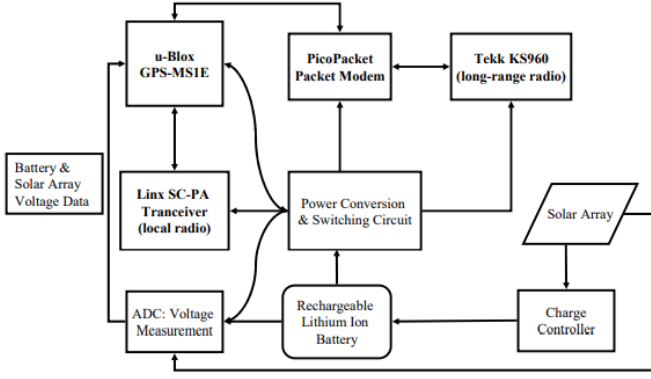


Figure. 3

4) *Flooding Protocol*: The authors have chosen a simple approach to move data back to the base station which floods data to all neighbors whenever they are discovered. If the nodes move extensively and meet a fair number of other nodes, then given enough time, data will eventually migrate back to the base. In this way, a high percentage of the data eventually makes it back to base. The base station does not necessarily have to come into contact with all the nodes in the system; instead, coming into contact with just a few nodes may be enough. Indeed, it can be inferred that by identifying a few highly-interactive nodes, i.e. nodes that meet a large number of other nodes, we can collect a substantial amount of data readily. While flooding can potentially return the highest success rate in a peer-to-peer network, the large amount of data flooded through the network can lead in some situations to exorbitant demands for network bandwidth, storage capacity, and energy.

III. POSSIBLE METHODOLOGIES

A. Data Gathering

Data gathering : It is one of the primary operations carried out in wireless sensor networks. It involves data collection with aggregation and data collection without aggregation, referred to as data aggregation and data collection respectively. In the last decade, many techniques for these two applications are proposed, with different focuses, such as accuracy, reliability, time complexity, and so on. several data gathering techniques have been proposed for WSNs with the main aim of reducing energy consumptions in WSNs by exploiting correlations among sensory data.

We can distinguish them into two broad categories: (i) Compression-oriented (ii) Networking-oriented. The first category, named compression-oriented, is focused on maximizing network lifetime by taking advantage of data compression techniques. In particular, analyze different lossless compression schemes for WSNs exploiting the temporal correlation in the sampled signals. Since radio transmission is the primary source of power consumption in WSNs, a second category of data gathering techniques, named networking-oriented, have dealt with the problem of maximizing network lifetime by

taking into account network protocols and, more specifically, forwarding/routing mechanisms.

There are four main data gathering techniques which are used widely while deploying dense wireless sensor network:

1) *Signal Processing Techniques*: Frequently high correlations among sensor readings exist. In this case, it is inefficient to deliver the entire raw data to the destination and signal processing, in particular Transforms and Encoding Compression techniques, can be exploited in order to reduce the amount of data to send. Here, node collects measurements following the Shannon-Nyquist sampling theorem; these measurements are transformed and properly encoded and the output of such transformation is stored in the payload of one or more packets and sent to the sink. In particular, either lossy or lossless techniques can be used depending on the particular application scenario. With lossy techniques, the original data is compressed discarding some of the original information; this allows achieving higher compression ratios but at the receiver side one can only reconstruct the data with a certain accuracy. However, in some types of monitoring, the accuracy of observations is critical for understanding the underlying physical processes. some application domains (e.g., body area networks BANs in which sensor nodes permanently monitor and log vital signs) demand sensors with high accuracy and cannot tolerate measurements corrupted by lossy compression processes.

2) *Compressive Sensing*: Compressive sensing (CS) is a new paradigm introduced by Candes and Tao and Donoho used to capture and to compress signals in WSNs where compression and sampling are merged and carried out at the same time. CS compresses a signal while acquiring data at its information rate. CS theory states that if a signal is sparse or compressible in a certain basis, then it can be reconstructed from a small number of linear measurements. More precisely, let us define k -sparse signals $x=(x_1, \dots, x_n)^T$ as signals that can be expressed as

$$x = \gamma \alpha, \quad (1)$$

where γ is an orthonormal transform and α is a vector with at most k nonzero entries; CS theory states that x can be recovered from

$$m = O(k \log(n/k)) \quad (2)$$

linear combinations of measurements obtained as $y = \alpha^* x$, where α is an $m \times n$ matrix.

3) *Information theory related techniques*: In order to exploit the correlation of data concurrently acquired by different sensors, DSC (distributed source coding) techniques, inspired by the Slepian-Wolf theorem, can be applied. The DSC techniques imply that each sensor node sends its compressed outputs to the sink for joint decoding. This means that the nodes need to cooperate in groups of two or three so that one node provides the side information and another one can compress its information down to the limit. The most practical and well-known implementation of DSC is DISCUS where sensor nodes are considered divided into clusters. For

each cluster, a node (the cluster head) sends uncompressed data (as side information) while all other nodes transmit encoded (compressed) data. DSC relies on the assumption that statistical characteristics (i.e., correlation function) of the underlying data should be known a priori, which is difficult to obtain in practical scenario. A simple manner to improve reliability is achieved by retransmitting cluster head packets more times but this reduces compression efficiency.

4) *Networking Techniques*: Since radio transmission is the primary source of power consumption at the nodes, the design of energy-efficient routing is another important topic to investigate in the design of data gathering technique. The basic idea is to route the packet through the paths so as to minimize the overall energy consumption for delivering the packet from the source to the destination. The problem focuses on computing the flow and transmission power to maximize the lifetime of the network. Specifically, the energy consumption rate per unit of information transmission for each node depends on the choice of the next hop, that is, the routing decision. This choice can influence the energy required to reach the sink. Data aggregation can be performed on top of the routing algorithm.

B. Data Aggregation

Data aggregation is an energy efficient technique in WSNs. Due to high node density in sensor networks same data is sensed by many nodes, which results in redundancy. This redundancy can be eliminated by using data aggregation approach while routing packets from source nodes to base station. The aggregation function is usually performed by extracting some statistical values (e.g., maximum, minimum, and average) and then by transmitting only these. In such a way, it is possible to reduce the amount of communicating data in the dense sensor networks and reduce the power consumption.

IV. PROPOSED METHOD

A. Problem Statement

Method for reconstructing a data packet incorrectly received in a wireless sensor network

B. Cyclic Redundancy Check

In digital data transmission technology there are known error recognition and error correction mechanisms that could also be used in wireless sensor networks. Prior to the transmission, these mechanisms encode available information which is then transmitted in the form of data, as the result of which the information is no longer directly present during the transmission and upon receipt, and instead must first be decoded at the receiver. However, such error recognition and error correction mechanisms include quite complex computations, which in the case of a wireless sensor network must be carried out in the transmitting wireless nodes, which is energy-intensive. In addition, on the receiver side this naturally increases the computing time, since the encoded data must first be decoded according to the error recognition and error correction mechanisms. Such methods are therefore hardly suitable for wireless sensor

networks in the industrial environment. , “Hard decision packet combining methods for industrial wireless relay networks,” Communications and Electronics, 2008, ICCE 2008, Second International Conference on Communication and Electronics, Jun. 4-6, 2008, pp. 104-108, it is described how received data may be corrected in the receiver based on a cyclic redundancy code (CRC). For this purpose, an incorrectly received data packet that is identified based on the CRC is not discarded, but, rather, is stored in a buffer. The incorrectly received data packet is retransmitted by the sender. If the CRC check once again shows an incorrect data transmission, the previously buffered message and the newly received message are analyzed bit by bit, and an attempt is made to reconstruct therefrom the correct data by reiterating the bits in the positions in which the two data packets differ, and in each case checking the CRC. This method is known as combinatorial testing. The data packet cannot be reconstructed, an additional retransmission of the data packet may be requested. This results in a reduction of the data transmission error rate and the number of necessary data transmissions.

C. Destination Sequenced Distance Vector (DSDV)

The DSDV (destination-sequenced distance vector) protocol uses the Bellman-Ford algorithm to calculate paths. DSDV protocol guarantees loop free paths. We can avoid extra traffic with incremental updates instead of full dump updates. Path Selection: DSDV maintains only the best path instead of maintaining multiple paths to every destination. With this, the amount of space in routing table is reduced.

V. TOOLS AND TECHNOLOGIES USED

A. Operating System

Due to the nature of the domain - Wireless Sensor Networks, the compatibility is with Linux systems only. Hence our work is done using Ubuntu 18.04 LTS System.

B. Network Simulator v2 (NS-2)

NS-2 is an object-oriented discrete event simulator targeted at networking research. It is an open source network simulator originally designed for wireless, IP networks. It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, https and DSR. It is primarily Unix based. NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl).

The C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The C++ and the OTcl are linked together using TclCL.

C. NS2 Scenario Generator (NSG)

NSG is a tcl script generator tool used to generate TCL Scripts automatically. It is a Java based tool that runs on any platform and can generate TCL Scripts for Wired as well as Wireless Scenarios for NS-2. The procedure to execute these TCL Scripts on NS-2 is same as those of manually written

TCL Scripts. It can create Wired and Wireless nodes just by drag and drop. Also Supports Ad Hoc routing protocols such as DSDV, AODV, DSR and TORA. It has many setting network parameters such as packet size, start time of simulation, end time of simulation, transmission range, interference, range bandwidth etc for wireless scenarios.

VI. EXPERIMENTAL ANALYSIS & RESULTS

A. Simulation Parameters

```
1 set val(chan) Channel/WirelessChannel;
2 set val(prop) Propagation/TwoRayGround;
3 set val(netif) Phy/WirelessPhy;
4 set val(mac) Mac/802_11;
5 set val(ifq) Queue/DropTail/PriQueue;
6 set val(ll) LL;
7 set val(ant) Antenna/OmniAntenna;
8 set val(ifqlen) 50;
9 set val(nn) 50;
10 set val(rp) DSDV;
11 set val(x) 2043;
12 set val(y) 100;
13 set val(stop) 10.0;
```

Listing 1. Parameters

B. TCP Connection

```
1 set tcp0 [new Agent/TCP]
2 $ns attach-agent $n1 $tcp0
3 set sink1 [new Agent/TCPSink]
4 $ns attach-agent $n0 $sink1
5 $ns connect $tcp0 $sink1
6 $tcp0 set packetSize_ 1500
```

Listing 2. Agents Definition

C. Application Definition

```
1 set cbr1 [new Application/Traffic/CBR]
2 $cbr1 attach-agent $tcp0
3 $cbr1 set packetSize_ 1000
4 $cbr1 set rate_ 1.0Mb
5 $cbr1 set random_
6 $ns at 1.0 "$cbr1 start"
7 $ns at 2.0 "$cbr1 stop"
```

Listing 3. Application Definition

D. Results

After the simulation process is over using the parameters like channel type, number of nodes, protocol definition, antenna type etc, the output of the scenario is saved in a trace file where you can observe the values and estimate the feasibility of the system in real world.

Here as you can see, the message packets transmission is shown in the output file through which we were able to understand the path transmission, packet loss if occurs and the time taken for transmission of packets between a 50 node wireless sensor network scenario.

```
1 r 0.033786592 _16_ RTR --- 1 message 32 [0 ffffffff
  1 800] ----- [1:255 -1:255 32 0]
2 r 0.033786679 _43_ RTR --- 1 message 32 [0 ffffffff
  1 800] ----- [1:255 -1:255 32 0]
3 r 0.033786847 _28_ RTR --- 1 message 32 [0 ffffffff
  1 800] ----- [1:255 -1:255 32 0]
4 s 0.093438986 _12_ RTR --- 2 message 32 [0 0 0 0]
  ----- [12:255 -1:255 32 0]
```

```
s 0.093793986 _12_ MAC --- 2 message 90 [0 ffffffff
  c 800] ----- [12:255 -1:255 32 0]
6 r 0.094514246 _31_ MAC --- 2 message 32 [0 ffffffff
  c 800] ----- [12:255 -1:255 32 0]
7 r 0.094514248 _18_ MAC --- 2 message 32 [0 ffffffff
  c 800] ----- [12:255 -1:255 32 0]
8 r 0.094514279 _15_ MAC --- 2 message 32 [0 ffffffff
  c 800] ----- [12:255 -1:255 32 0]
```

Listing 4. Simulation Output

VII. CONCLUSION

As seen in the simulation results, if there is any packet loss occurs or damage packet transmitted then DSDV routing protocol is able to detect it. By using TCP agent that depends on acknowledgement after every packet delivery, so that creates reliable communication link between nodes. And CRC error correction method can detect and correct data packets by matching bits and sends request of retransmission of particular data packet. In our analysis we got efficient results needed for successful wireless sensor network. We can further take in consideration to deploy the network with a greater number of nodes under extreme environmental conditions where nodes failure may occur and our protocol will still retransmit the corrupted data packets choosing the least cost path.

REFERENCES

- [1] P. Juang, H. Oki, Y. Wang, M. Maronosi, L. Peh, D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet", Proc. ASPLOS, Oct 2002
- [2] B. Namazi, K. Faez, "Energy Efficient Multi SPEED Routing Protocol For Wireless Sensor Networks", International Journal of Electrical and Computer Engineering (IJECE), vol. 3, no. 2, pp. 246-253, 2013.
- [3] O. Krejcar, M. Mahdal, "Optimized Solar Energy Power Supply for Remote Wireless Sensors Based on IEEE802.15.4 Standard", International Journal of Photoenergy, vol. 2012.
- [4] L. Ding, X. Gao, W. Wu, W. Lee, X. Zhu and D. Z. Du, "Distributed Construction of Connected Dominating Sets with Minimum Routing Cost in Wireless Networks," 2010 IEEE 30th International Conference on Distributed Computing Systems, Genova, pp. 448-457, 2010.
- [5] P. Zhang, C. Chen, and M. Liu, "The application of compressed sensing in wireless sensor network" in Wireless Communications Signal Processing, 2009. WCSP 2009. International Conference on, 2009, pp. 1-5
- [6] Q. Zhang, Z. Xie, W. Sun, and B. Shi, "Tree Structure Based Data Gathering for Maximum Lifetime in Wireless Sensor Networks," in Web Technologies Research and Development-APWeb 2005. vol. 3399, Y. Zhang, K. Tanaka, J. Yu, S. Wang, and M. Li, Eds., ed: Springer Berlin Heidelberg, 2005, pp. 513-522