

This PDF consists of two research projects.

- **Thesis Project** - Analysis of wildCENSE for Habitat Monitoring Applications in Mobile Wireless Sensor Network
- **Course Project** – Topology formation in Sparse and Dense Mobile Ad-hoc networks

Thesis Report

Analysis of wildCENSE for Habitat Monitoring Applications
in Mobile Wireless Sensor Network

Mentor: Professor Sanjay Srivastava
Student: Pratik Sajnani (ID: 200801059)

Introduction

Wireless sensor networks are very specific kind of networks are mostly used in monitoring and tracking activities. A very important part of sensing is to gather data and store it for analysis. Usually nodes in this networks are sensors placed on or around the object of interest. So, the mobility of these sensors is determined by the object being tracked or monitored. In the case of mobile objects of interest, the sensors are also mobile, and so the energy consumed in various activities carried out by sensors is supposed to be efficiently utilized and the same is true for memory that is limited in size.

What are sensors made of?

Every sensor has the following parts with specific purposes -

1. Sensing device - To track the object of interest
2. Micro-controller - overall integration
3. Memory (usually non-volatile) - For temporary storage of gathered data
4. Transceivers - To transmit the collected data and to relay data of other nodes
5. Interfaces (like GPS and serial ports) - To get positional data

What sensors do?

The sensors carry out two main activities in a constrained scenario:

- A - To collect data about the object of interest (sensing)
- B - To send it to the appropriate entity for analysis (sending)

Who do they work for?

The sensors carry out the data collection activity and through communication in the network of other sensors relay their data to a final destination where it is stored for analysis at a later point in time. The basic purpose of the communication activity is to ensure that the data reaches the final destination.

General Concepts

Sensing

Sensing usually depends on the hardware that is used in the sensor and can do with a limited range of signals whose primary purpose is to generate data by probing the object of interest continuously or at regular intervals. Mostly as far as sensing is concerned, all the rates of flow, be it data accumulation or dissipation of energy are nearly constant.

Sending

In cases where sensors are mobile the entity which is the final destination is stationary and may not lie in the trajectory of the sensor node. Hence, sending data is rarely done directly. Each node relays data via other nodes and so the network of

nodes becomes a vital communication network for every node involved in the sensing activity. Thus the primary concern is the efficiency and reliability of the sending activity. Here, the energy, the bandwidth and the computing power become constraints.

Efficiency

The efficiency can be intuitively defined as the probability of successful sending operation and the reliability as the probability that the sent data will reach the final destination. This problem of data extraction (or sensing and sending together) varies depending on the context, i.e. what purpose is the data gathered for? And what is the data? These two important questions are responsible for the differences that arise in data extraction techniques.

Problem Statement

Our problem is concerned with habitat monitoring, where sensors are placed on animals and kept for a very long time, usually a year or more, to track their movement patterns and send this data to a base station which is the centralized entity for dumping data. Many sensors are placed on animals whose movement is restricted to a fixed but large area. These sensors become a network for each other and cooperation between networks makes the process more efficient and reliable. Thus every sensor works not only for communicating its own data but also for communicating the data of the other nodes and in return can relay its own data to these nodes. Thus every node needs to communicate and take decisions based on some inquiry about its neighbors and its current state. Thus, there is a need for a protocol of communication which would act as a uniform course of action based on events that can possibly occur in the lifetime of these nodes.

Subsections of the protocol

1. Synchronization

Synchronization is a very important part of this protocol. The nodes function based on what they know about their state and about their neighbors. But every communication event, including the event which requires communication just for the sake of neighbor information has to happen in sync, otherwise there is a significant impact in the communication. Even in a simple two node interaction the act of receiving messages from one node helps clock the events, but a simple problem like who sends first may become a very big issue in a scenario of nodes more than two. Such is the case of our present communication scenario. A lot of nodes need to relay information and all of them cannot do it at once! The bandwidth utilized is very limited and should be used carefully. Thus events here need to be clocked at the cost of some energy and some processing power for a larger benefit or regularity and efficiency in communication between nodes. Thus this protocol needs to address the issue of synchronization.

Each node has an external clock and this clock times the functions that the node performs while interacting with other nodes. Nodes sleep and wake up and schedule

their tasks which are repeated periodically based on this clock. Thus they work in units of time called duty cycles, a concept basic to the underlying hardware. These duty cycles need to be brought in sync, so that the nodes can perform activities in a synchronized manner. The current protocols use GPS data for this purpose. The nodes communicate with a satellite which returns their position data along with a timestamp. The nodes use this timestamp to stay in sync with every other node.

2. Peer Discovery

In the absence of a specific centralized structure aiding the communication of nodes, the nodes need to collect this information on their own. Every node that cooperates in a network needs to have information about its neighbors before taking any action that requires their assistance. The other advantage is that peer discovery becomes the first probing message into the medium of communication and allows us to determine channel properties at regular intervals, or at least before the data transfer event actually happens.

Peer discovery is done in a synchronized manner, in the sense that it has a fixed time slot in the communication cycle of nodes, but can be done in two different ways

Contention free scheme

Each node is allowed to broadcast a peer-discovery message at a fixed or random time within a range depending on the exact protocol.

Contention based scheme

Each node competes for a share of the communication medium, and sends peer discovery messages.

3. Pre-processing

Pre-processing is an optional phase of the protocol because its significance varies in different contexts. In this part, before the data is actually exchanged, control packets that help improve the performance by reducing transmissions and hence reducing the energy required.

The nodes, prior to exchanging data send control packets that inform them specifically what the other node does not have. So now the nodes exactly know what data is to be sent and this in some cases reduces the amount of transmissions done and hence the time slot for data exchange phase can be utilized more efficiently.

4. Data Exchange

Data exchange phase is the most basic and the largest phase in terms of time. In this phase the nodes exchange useful data and ensure that they are up-to-date with each other in terms of this data. Data exchange is done to improve chances of data reaching the base station and it allows the memory to be freed without having to dump the data directly at the base station, in short it allows the system to function even when some nodes visit the base station less frequently or not at all.

Data exchange can be done in two ways. The first approach is to broadcast data to every node in the neighborhood which may not always be efficient, but in some scenarios may give more reliability and keep the protocol simple. On the other hand, the second approach, a restricted forwarding mechanism has the advantage of utilizing the channel more efficiently and saving node energy.

5. Post-processing

It is a process done at every node after it has successfully communicated its data. It allows the nodes to decide what to do with the data and the information about the data it received from its neighbor.

The data that is exchanged is checked for redundancy by comparing each entry with the data that the node had prior to the exchange. The node also dumps the data that has reached the base station and does not require to be stored anymore, but it retains the information that the data has reached the base station until it informs other nodes about it.

Phase-wise Modifications to the protocol

SYNCHRONIZATION

Packet Structure

GPS Packet

1. Node ID
2. Location Data
3. Timestamp

Constants

GPS_interval: time between two subsequent GPS intervals

GPS_err: This bit indicates if last GPS data retrieval failed

Phase Description

These packets are sent at regular intervals of GPS_interval to a satellite which returns Location Data and Timestamp. This query is not done regularly, its duration is decided by a statistical analysis of how frequently the GPS query should be sent to stay in sync and to prevent redundancy.

Exceptions

In the event of GPS failure this bit will label the node as an out of sync node for this cycle of data exchange.

Peer discovery (done after every time T, for a period of T(pd))

Packet Structure

Broadcast Packet

1. Sender node ID

Constants:

Chan_lock: In contention based scheme this ensure that the channel is available to only those nodes who obtain a lock for the channel.

Phase Description

K: constant to limit the no. of nodes that enter transmitting mode together.

A - Contention Free Scheme

In contention free scheme, the nodes are all in peer discovery phase for time $T(pd)$ and this is further divided into time slots of size $T(n)$. Every node chooses a random time in this time slot $T(n)$ to become a transmitter, at this time, the other nodes listen. If a node receives a packet it adds the node id of the sender node to its neighbor list. This is repeated by every node in the next $T(n)$ time slot. After $T(pd)$ time this phase gets over and is repeated $T(pd)/T(n)$ no. of times. Given that every node chooses a random slot, the probability of collision is less, but if collisions do occur the retransmissions ensure that the discovery packets are not affected.

B - Contention Based Scheme (Time slot - $T(s)$)

In a contention based scenario, nodes start in the receiving mode and has a number of channels for its communication but uses a channel default to a particular time slot. This default channel assignment is done by a formula .Now, each node generates a random number and this random number is compared to a constant K , if, $R \leq K$ for a node, it becomes a transmitter and broadcasts a peer discovery packet. Nodes receive this packet and reply with an acknowledgement. If there are collision, the packet will not reach or the acknowledgement will not reach. In that case, the node is not added to the neighbor list. If the transmission is successful, and a neighbor node receives a peer discovery packet, it replies randomly in the latter half of the timeslot. (Some time in $T(s)/2$). If the node that broadcasted the peer discovery packet gets this acknowledgement, it makes an entry in the neighbor list and sends another acknowledgement. Both these nodes now are each other's neighbors and to indicate that they are ready for the data exchange phase, they assume chan_lock status after this acknowledgement. This prevents them from continuing peer discovery. Thus the two nodes agree to communicate, and if multiple nodes reply, the channels available are assigned in a FIFO manner till they are exhausted. Thus nodes can now enter data exchange phase. The default channel for the next slot is utilized in the next time slot. So, in every slot some nodes assume chan_lock and some wait for the next slot and communicate on the default channel for that timeslot.

How to handle Exceptional cases

Out of sync nodes (GPS failed)

According to Zebranet the time for GPS synchronization is 25 seconds, and the time allotted to peer discovery is 30 seconds, so the nodes that are out of sync receive Peer discovery packets but do not acknowledge them, instead, they send a GPS signal and resynchronize. This also implies that the nodes will be out of the pre-processing, data exchange and post-processing phase for that cycle. They resume normal behaviour in the subsequent cycle. GPS error bit had discussed previously.

Isolated nodes problem (Base station GPS data assumed)

If a node is isolated for a very long time, say for 5 cycles, and there is no neighbour found in the Peer discovery phase, the node checks if its current GPS position is close to Base station, if not, the node starts dumping data of other nodes, starting with oldest data, at every cycle of Peer discovery till it discovers a node in any subsequent cycle of peer discovery. If it has finished with other nodes data, it dumps its data, but only that data which has been communicated to other nodes. This state is indicated by a special bit, ISOLATED_BIT

ISOLATED_BIT = 1 (The node is isolated and keeps dumping data from other nodes, till it finds a neighbor)

ISOLATED_BIT = 0 (The node is not isolated)

Pre-processing

The Packet Structure

1. Sender Node ID
2. Receiver Node ID
3. Data Representation (data entry list)

a) Data dump bit --> A = Data Absent from base station, P = Data Present in base station, D = to be dumped

b) Data Popularity Number --> This number denotes how many nodes possess this data, it helps when dumping data that could not be transmitted before the memory overflow occurred.

c) Data to Send Bit--> This bit is altered by the node by comparing its data using the dump bit and checking for data that is already there. This denotes data to be exchanged.

Phase Description

In this phase the nodes exchange what is called as a data representation list, which informs each of them what data has to be exactly exchanged and what data can be

dumped. The node that receives this data checks first for the presence of the data in its own memory, if the data is present and Data Dump Bit is P, it keeps the data to be dumped later by changing the Data Dump Bit to D and this data is not sent to the other node. If the data is present, but the Data Dump Bit is A, it does not dump the data, but does not send it to this node. Only the remaining data is made available for data exchange and this is denoted by the Data to Send Bit. After having done this, the pre-processing phase is over and the nodes know exactly what data to send.

Data-exchange

Packet Structure

Sender Node ID
Receiver Node ID
Data Field

Phase Description

In this phase exchange of data takes place between nodes or between a node and a base station.

Contention Free Scheme

In a contention free scheme the data exchange happens using multiple slots. These slots are assigned in a round robin fashion and the data exchange between nodes takes place for the duration for which the phase lasts. If no neighbor is found in peer discovery for a particular node, it remains in sleep mode for this phase. If even one neighbor is present the node enters the data exchange as receiver till it gets its slot for transmission.

If a node has to exchange data with a base station, it can send only in its timeslot. The node gives higher priority to its own data when sending to a base station and the base station acknowledges the data received so that the node can purge the data.

Contention Based Scheme

The nodes that obtain a lock for a channel start exchanging data on the same channel. Every node that was a receiver starts transmitting data to a node based on the acknowledgement number it had received from that node i.e. every node transmits at $\text{time-slot} * (\text{acknowledgement number})$ time. This way each transmitting node gets turns to transmit to a particular node based on the acknowledgement number assigned to it by the receiving node.

A special timeslot for the base station is reserved in the end of this data exchange phase. The peer discovery is done in the same way followed by all nodes listening to the base station to update their data representation. This happens like the data exchange phase in case of a node to node communication. The base station sends the representation to inform the nodes about the received data and the deficient data. The nodes then send the required data to the base station. After sending the

required data nodes purge it and the base station updates its data representation.

Post-processing

Phase Description

In this phase the nodes process the changes in their data representation and purge the data that has already been sent to the base station. This is done by checking the Data Dump bit. All entries with the Data Dump bit set to 'D' are dumped and the memory is freed. The newly received representation is checked for timestamps and if the timestamp is newer then the node updates its data representation and finds out the deficient data by subtracting from the latest timestamp available with it.