BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Traffic Prioritization Using Packet Aggregation in Saturated WSN

Authors

Mehedi Hassan	ID: 124420
Zubair Ibne Alam	ID: 124448

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)
Organisation of Islamic Cooperation (OIC)
Gazipur-1704, Bangladesh

November, 2016

Islamic University of Technology

Department of Computer Science and Engineering (CSE)

Traffic Prioritization Using Packet Aggregation in Saturated WSN

Authors

Mehedi Hassan	ID: 124420
Zubair Ibne Alam	ID: 124448

Supervisor

Dr. Abu Raihan Mostofa Kamal
Associate Professor
Department of CSE
IUT

A thesis submitted to the Department of CSE

in partial fulfillment of the requirements for the degree of B.Sc.

Engineering in CSE

Academic Year: 2015-16

November - 2016

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Mehedi Hassan and Zubair Ibne Alam under the supervision of Dr. Abu Raihan Mostofa Kamal, Associate Professor of Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Mehedi Hassan

Student ID - 124420

Zubair Ibne Alam

Student ID - 124448

Supervisor:

Dr. Abu Raihan Mostofa Kamal

Associate Professor

Department of Computer Science and Engineering

Islamic University of Technology (IUT)

Acknowledgment

We would like to express our grateful appreciation to **Dr. Abu Raihan Mostofa Kamal**, Associate Professor, Department of Computer Science & Technology, IUT for being our advisor and mentor. His motivation, suggestions and insights for this thesis have been invaluable. Without his support and proper guidance this research would not have been possible. His valuable opinion, time and input provided throughout the thesis work, from first phase of thesis topics introduction, subject selection, proposing algorithm, modification till the project implementation and finalization which helped us to do our thesis work in proper way. We are really grateful to him.

It was our pleasure to get the cooperation and coordination from Professor Dr. M.A. Mottalib, Head of CSE Department, IUT during various phases of the work. We are grateful to him for his constant and energetic guidance and valuable advice.

Our deep appreciation extends to all the respected jury member of our thesis committee for their insightful comments and constructive criticism of our research work. Surely they have helped us to improve this research work greatly.

Lastly we would like to thank the faculty members of CSE Department, IUT who have helped to make our working environment a pleasant one, by providing a helpful set of eyes and ears when problems arose.

Abstract

In the sensor networks, network traffic prioritization is gaining attention in the WSN community, as more and more features are being integrated these networks. Real-world deployment experience suggests that WSN brings new challenges to existing problems, such as resource constraints, low data-rate radios, and diverse application scenarios. But most of traffic prioritization problems deals with how to handle the HP (high-priority) data packet ignoring the LP (Low-Priority) data packet. Whenever any HP data packet arrives, LP data packet transmissions are suspended. This results in a loss of LP packets due to network congestion in a saturated network, weak radio signal, multi-path fading or cache overflow. We propose a framework that that will not suspend the LP data packet completely when any HP data packet has arrived. The framework will cache and aggregate the LP data packets and transmits the LP data packet after certain interval while transmitting the HP data packets. We differentiate between the HP and LP data by leveraging transmission power difference and radio-capture effect. We classify the LP data traffic by using a hierarchical aggregation algorithm for data reduction of similar data.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Prioritization of network traffic is simple in concept: give important network traffic precedence over unimportant network traffic. That leads to some interesting questions. What traffic should be prioritized? Who defines priorities? Do people pay for priority or do they get it based on traffic type (e.g., delay-sensitive traffic such as real-time voice)? For Internet traffic, where are priorities set (at the ingress based on customer preassigned tags in packets, or by service provider policies that are defined by service-level agreements).

Prioritization is also called CoS (class of service) since traffic is classed into categories such as high, medium, and low (gold, silver, and bronze), and the lower the priority, the more "drop eligible" is a packet. E-mail and Web traffic is often placed in the lowest categories. When the network gets busy, packets from the lowest categories are dropped first.

Traditional way of solving traffic prioritization problems is to suspend the LP data to provide the HP data the full network capacity after the control overhead. Since the HP data are sporadic and delay-intolerant, this kind of solution seems feasible. Whenever a HP data packet is arrived, LP data packet transmission is suspended. Some LP packets which are already in transmission may be corrupted or lost. Now LP data could be lost due to cache overflow, weak signal strength etc. LP packets can be saved in a cache to transmit later when HP data packets have finished transmit. They classify the similar kind of LP data and aggregate them together to reduce data and send or transmit after a certain interval whenever HP data packets are transmitting. Adopting such a solution will increase the network throughput and reduce the amount of redundant data eventually relaxing the performance constraints.

1.2 Motivation

- Categorization of data traffic based on priority.
- Network utilization.

1.3 Problem Statement

The problem statement of our thesis work is as follows:

> Different traffic prioritization scheme halts LP data traffic completely to transmit HP data. In these schemes, LP data packets loss is frequent because of weak signal strength, multi-path fading or cache overflow. LP packets are large in number and delay-tolerant. Many redundant LP packets are sent over the network from time to time which consume the whole capacity of network. Redundancy decreases the throughput of network.

1.4 Our approach

In this work, we propose an online disease identification framework integrating case based reasoning and machine learning methods. Using numerous natural language processing techniques, relevant attributes are extracted from the unstructured user input. Then these attributes are used to generate a ranking of probable diseases from a symptom based clustered disease database.

In this work, we propose an algorithm to aggregate the low priority data packets based on clustering the similar kinds of data packets and sending only one packet instead of dropping the low priority data packets thus increasing reliability, throughput and decreasing packets drop ratio.

Chapter 2

Related Work

Many works have been done traffic prioritization and aggregation for data reduction. Most of them has focused on HP data traffic protocols. Energy consumption in such protocols were also taken sensitively. Aggregation of data has been done from the leaf node to parent node. Classification has been done on data for reduction.

2.1 RushNet: Practical Traffic Prioritization for Saturated Wireless Sensor Networks

RushNet [1] framework that prioritizes two common traffic patterns in multi-hop sensor networks: low-priority (LP) traffic that is large-volume but delay-tolerant, and high-priority (HP) traffic that is sporadic but latency-sensitive. RushNet achieves schedule-free and coordination free delivery differentiates with the following features.

First, RushNet works with most data collection protocols to deliver LP traffic.

Second, RushNet leverages transmission power difference and radio capture effect to implement on-demand HP packet delivery with low overhead.

Third, RushNet proposes a retro-diction technique to help nodes minimize the overhead of recovering LP packet loss due to concurrent HP traffic. In this paper, latency is used as the key traffic class differentiator because it implies both spatial and temporal constraints, more stringently than reliability and energy consumption. For example, a deployment demanding low latency mostly implies the use of delivery reliability mechanisms to bind the data arrival time at the gateway. However, a deployment with reliability mechanisms does not necessarily put constraints on the latency. RushNet has the following properties.

First, RushNet does not put strict assumptions on the implementation of the LP transport service. This allows network administrators to select the ones that fit their application requirements. While the popular Collection Tree Protocol (CTP) is a suitable choice, our current version of RushNet provides a LP transport service implementing token based congestion avoidance with hop-by hop block transfers. As token-based data collection protocols can minimize various delays in packet transmissions, the network can deliver high-volume of LP data at high throughput.

Empirical results show that RushNet can improve the network throughput by a factor of 4.75 over previous sensor network data collection systems. Second, for the HP transport service to minimize the control overhead in sending an additional traffic class, the challenge is to reduce explicit coordination among potential transmitters. Our approach is to give the HP transport service the freedom to inject low-volume of HP packets at higher transmission power at any time, while leveraging the radio capture effect. In addition, RushNet goes one step further by proposing preemptive packet train, a technique that encourages the radio capture effect to happen on of the-shelf IEEE 802.15.4 radio chips. The insight from real testbed measurements is that popular of-the-shelf 802.15.4 radio chips restrict capture effects to certain radio states only, i.e., preamble search.

Preemptive packet train essentially repeats the same HP packet transmission in a certain way to influence the receiver's radio state. Micro-benchmarks show that our techniques improve the packet reception ratio due to capturing by a factor of 10.

2.2 Hierarchical Aggregate Classification with Limited Supervision for Data Reduction in Wireless Sensor Networks

A hierarchical aggregate classification (HAC) [2] protocol is proposed which can reduce the amount of data sent by each node while achieving accurate classification in the face of insufficient label information. In this protocol, each sensor node locally makes cluster analysis and forwards only its decision to the parent node. The decisions are aggregated along the tree, and eventually the global agreement is achieved at the sink node. In addition, to control the trade off between the communication energy and the classification accuracy, they designed an extended version of HAC, called the constrained hierarchical aggregate classification (cHAC) protocol. cHAC can achieve more accurate classification results compared with HAC, at the cost of more energy consumption. It often happens that multiple sensor nodes detect the same events. Different sensor nodes due to their inaccuracy (e.g., noise in the data) and heterogeneity (e.g., different types of sensors), usually observe the events from different but complementary views. Therefore, aggregating the outputs of individual nodes can often cancel out errors and reach a much more accurate result.

However, aggregation of classification results is not an easy task in the absence of sufficient labeled data due to the lack of correspondence among the outputs of different nodes.

Additionally, to control the trade off between the communication energy and the classification accuracy, they designed an extended version of HAC, called the constrained hierarchical aggregate classification (cHAC) protocol. cHAC can achieve more accurate classification results compared with HAC, at the cost of more energy consumption.

2.3 Flash Flooding: Exploiting the Capture Effect for Rapid Flooding in Wireless Sensor Networks

Traditional flooding protocols can be very slow because of neighborhood contention: nodes cannot propagate the flood until neighboring nodes have finished their transmissions. The Flash flooding protocol avoids this problem by allowing concurrent transmissions among neighboring nodes. It relies on the capture effect to ensure that each node receives the flood from at least one of its neighbors, and introduces new techniques to either recover from or prevent too many concurrent transmissions.

Three types of Flash [3] flooding techniques is there:

Flash-I is identical to a standard flooding protocol except that nodes do not use media access control before propagating the flood. Thus, nodes in a Flash-I flood repeat the message as soon as they receive it, even if their neighbors are still transmitting. They do not perform clear channel assessment (CCA) as would a node in a CSMA network, nor do they wait their turn as would a node in a TDMA network. The result is that multiple nodes in the same neighborhood will be transmitting simultaneously. Flash-I can be applied in both high-duty cycle and low-duty cycle networks by transmitting either individual packets or X-MAC packet trains, respectively.

In Flash-II, each node sends a three-part flooding sequence. First, it immediately sends a packet transmission with no CCA or MAC delay, just like Flash-I. Then, it performs CCA and uses a MAC delay to wait for all neighbors to finish transmitting. Finally, it sends a second message, identical to the first. The purpose of the second packet is to reach any nodes that missed the first wave of packets due to concurrency and collisions. The CCA/MAC delay preceding the second message ensures that it will not interfere with the first wave of packets and that it will not itself be lost in a collision. The net effect is that the second packet is highly likely to reach any node or nodes that missed the first wave of packets, effectively jump starting the flood if it ended prematurely due to collisions. Flash-II can be applied straightforwardly to both high-duty cycle and low-duty cycle networks by using either single-packet transmissions or packet trains, respectively.

Flash-III applies a new technique to sense the amount of transmission concurrency in a network. First, they introduce a small inter-packet spacing (IPS) between packets in the packet train. Second, they introduce a very small CCA before the packet train is sent, where CCA ¡ IPS. The key idea behind Flash-III is that a nodes is more likely to pass the CCA check during low levels of concurrency because its CCA interval is more likely to coincide with the IPS intervals of the transmitting nodes. Therefore, it will also begin transmitting and will increase the level of concurrency. On the other hand, a node is unlikely to pass the CCA check during high levels of concurrency because its CCA interval is more likely to coincide with at least one packet. The flood begins when the source node A transmits a packet train. During this packet train, all of its neighbors B, C and D wake up and hear a flooding message. The CCA of nodes B and C coincide with the IPS intervals and these nodes transmit immediately, but the CCA of node D overlaps with the packets of C and so it will wait until C has finished transmitting before propagating the flood. Node D could have transmitted if the other nodes had larger IPS intervals.

2.4 Congestion Control Protocol for Wireless Sensor Networks Handling Prioritized Heterogeneous Traffic

Heterogeneous applications could be assimilated within the same wireless sensor network with the aid of modern motes that have multiple sensor boards on a single radio board. Different types of data generated from such types of motes might have different transmission characteristics in terms of priority, transmission rate, required bandwidth, tolerable packet loss, delay demands etc. Considering a sensor network consisting of such multi-purpose nodes, in this paper they propose Prioritized Heterogeneous Traffic-oriented Congestion Control Protocol (PHTCCP) [4] which ensures efficient rate control for prioritized heterogeneous traffic. This protocol uses intra-queue and inter-queue priorities for ensuring feasible transmission rates of heterogeneous data. It also guarantees efficient link utilization by using dynamic transmission rate adjustment. Detailed analysis and simulation results are presented along with the description of our protocol to demonstrate its effectiveness in handling prioritized heterogeneous traffic in wireless sensor networks.

There are 3 separate queues for each type of data. The number of queues in a node depends on the application requirements. The packet service ratio reflects the congestion level at each sensor node. When this ratio is equal to 1, the scheduling rate is equal to the forwarding rate (i.e., average packet service rate). When this ratio is greater than 1, the scheduling rate is less than the average packet service rate. Both of these cases indicate the decrease of the level of congestion. When it is less than 1, it causes the queuing up of packets at the queue. This also indicates link level collisions. Thus, the packet service ratio is an effective measure to detect both node level and link level congestion. PHTCCP uses hop-by-hop rate adjustment which ensures that heterogeneous data reach to the base station at their desired rates.

The simulation settings for evaluating PHTCCP were as follows: 100 sensors were randomly deployed in 100 100 m2 sensor field. The transmission range of each sensor was set to 30 m. Maximum communication channel bit rate was 32 kbps. Each packet size was set to 33 bytes. The control packet size (RTS, CTS, and ACK) was set to 3 bytes. The weight used in the exponential weighted moving average calculation of packet service time (equation 2) was set to 0.1. They considered three sensing units (e.g., temperature, seismic, and acoustic) mounted on the single radio board for each node in which temperature flow is set the highest priority valued as 3; seismic reading is given the priority value 2, and the acoustic as 1. Each queue size was set to hold maximum 10 packets. That is, the total queue length for a node was 30 packets (10 packets for each queue). Throughout the simulation, they used a fixed workload that consists of 10 sources and 1 sink. The initial originating rate was 4 pps (packets per second) and maximum originating rate was limited to 16 pps. They have compared our protocol with CCF as it also performs the distributed rate adjustment of the child nodes based on the parent's transmission rate. Here, they have used the term buffer and queue interchangeably.

Chapter 3

Design

This section states the classification model, LP data transport service and HP data transport service which we have taken into account while designing our algorithm.

3.1 Classification Model

We consider a wireless sensor network where many sensors are deployed within an area. Each node is capable of sensing any particular data depending on the application scenario and sending it to the neighbor node. That neighbor node will send data to its neighbor node and so on until it reaches the sink node.

In a sensor there can be two types of data: Low-priority (LP) and High-priority (HP). We assume any node in the network can be source. So whenever any data is generated by the source node, it will determine whether it is LP data or HP data. Here the question arises how it will determine whether a data packet is LP or HP data? It will simply check the threshold value of the data. If the data exceeds the threshold value then it will be classified as HP data packet and a higher transmission power will be assigned otherwise normal transmission power will be assigned. The receiving node will check the transmission power of the data packet and determine whether it is LP or HP data packet. Data packets

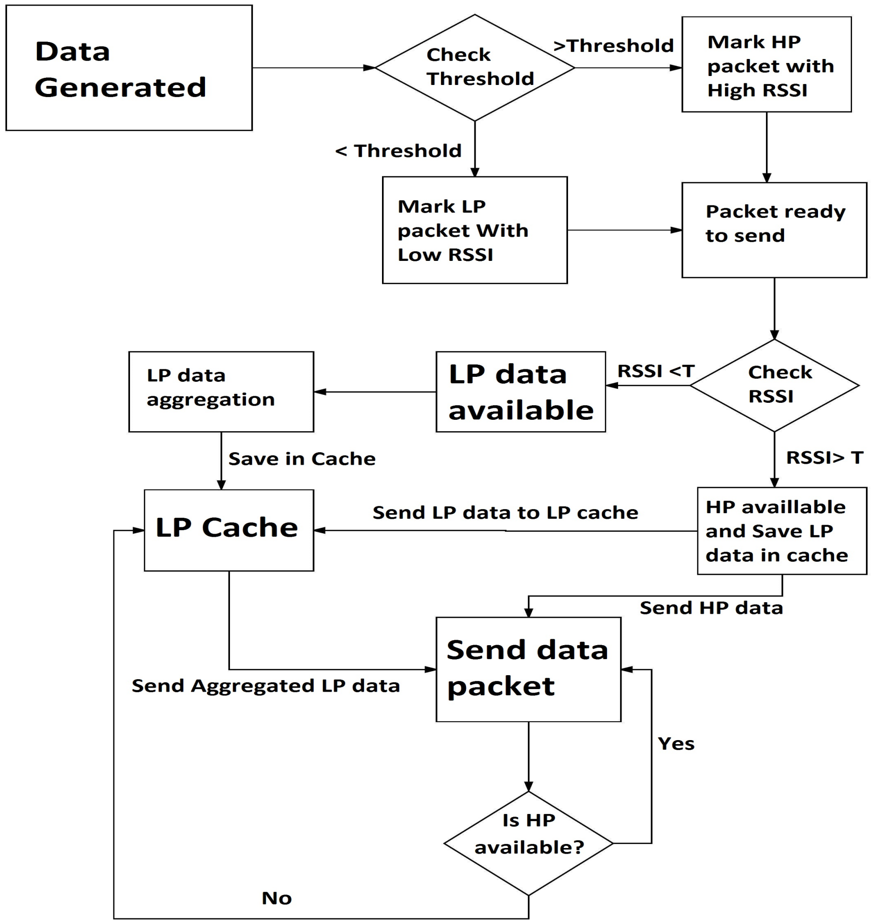


Figure 3.1: Overall Flow chart of our proposed system.

that are classified as LP data are most of the time redundant. If this redundant data is reduced then the network traffic will be improved along with the channel occupancy. One way to do that is to aggregate the data packets and send only one or two packets. So the source node will apply the aggregation process on LP data. We will apply Linear Regression technique to aggregate those data. After aggregation, LP data will be stored in the LP cache. At the time of sending any data packet every node will check if there is any HP data available or not. If the node has any HP data it will send that data using preemptive packet train without any delay. This will halt any other LP data transmission. Now here comes the significance of our proposed system. Halted LP data will be lost in this process.

But since we are storing the LP data in the LP cache we can retransmit those data after the HP data transmission is finished. By this way we are providing reliability.

3.2 LP Data Transport Service

Low-priority traffic has the characteristics of being delay tolerant and high-volume. It suggests that network throughput is the most important metric. We aim to improve network throughput with two mechanisms: token-passing mechanism and a hop-by-hop transmission mechanism to reduce inter packet transmission delays.

In token-passing mechanism, a node is only allowed to transmit if it holds the token. This mechanism allocates the full radio-medium access to one transmitter. Token-passing mechanism does not required the knowledge of the network. It simply determines the next node that should hold the token. So the overhead of the knowledge of the whole network is reduced.

In Hop-by-Hop Block transport, when a node holds the token it sends all of its data since its last transmission to its parent node.

3.3 HP Data Transport Service

Usually the transmission power difference between LP and HP data packets is 3 dBm. But we maintain about 5 dBm to increase the success rate of differentiating between LP and HP data packets. We minimize the gap between data packets while using preemptive packet train by short-cutting the transmission process. This is done by strobing the radio chip multiple times for more data packets from one node instead of strobing once. This mechanism is supported by most 802.15.4 radio chips.

3.4 Preemptive Packet Train

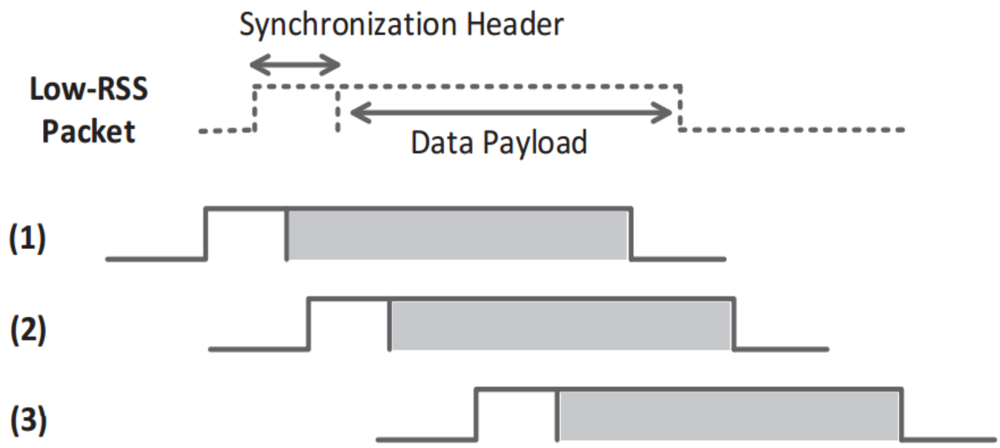


Figure 3.2: Preemptive packet train.

There are two factors involved in measuring the success of the reception of packet in a wireless sensor network. First, the power difference between packet with high RSS (Received Signal Strength) and packet with low RSS should be at least a certain threshold. Second, when the receiver is searching for frame synchronization headers in the air the transmitter should send a packet with high RSS. In the preemptive packet train, instead of sending HP packets with higher output power, same HP packet is sent multiple times in small intervals. This is done to make sure that at least one of these packets in the train is successfully captured by the receiver. These packets can still be successfully received even in the presence of the LP packets. To describe the idea we consider three different ways that two packets can collide.

In figure 2, case 1 and case 2 happens even before the receiver enters the data payload reception mode. If any packet appears with high RSS between two consecutive low RSS packets then the channel should be sufficiently quiet

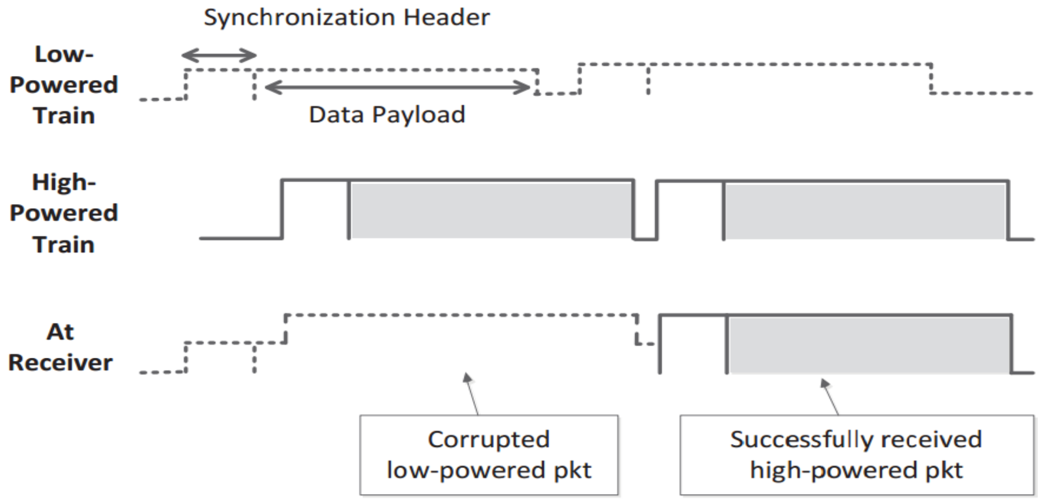


Figure 3.3: Preemptive packet train.

for the receiver to successfully capture the synchronization header. Even if the receiver is processing the preambles of a packet with weaker signal, a packet with stronger signal can force the receiver to resynchronize. But if the receiver already starts to process the data payload of a packet with low RSS, then the receiver will not resynchronize to the packet with high RSS which is depicted in figure 3. In fact the data packet of low RSS gets corrupted in the presence of the higher RSS. It is considered as a noise. So the receiver synchronize to the high RSS packet when the corrupted low RSS packet is finished processing. In this way preemptive packet train forces the medium to send the HP packets with high RSS before the LP packets with low RSS. As soon as the receiver receives the HP packet payload in the preemptive packet train, it relays the payload upstream with another preemptive packet train. Receiver checks the sequence number to eliminate any probability of duplicate data that could occur due to repeated transmission. Link-level reliability is ensured and implemented with overhearing and by taking advantage of the fact that the receiver would immediately relay HP packets upstream.

Chapter 4

Model

4.1 Decision Aggregation Model

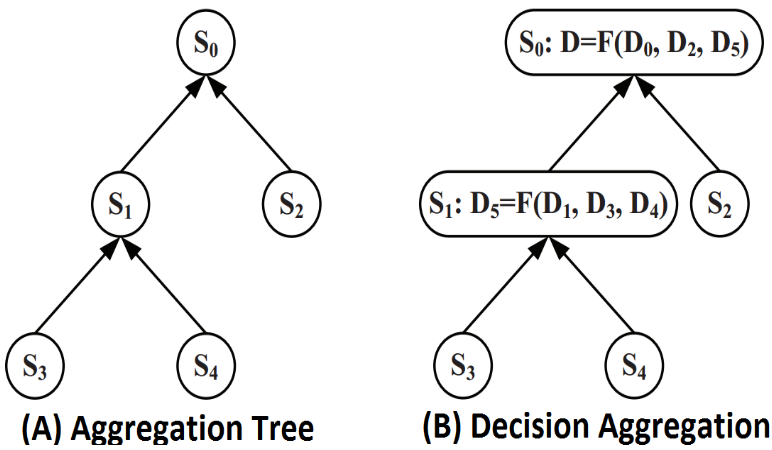


Figure 4.1: Clustered Trees.

We consider a tree T rooted at the sink node and denote the rest of the set of the sensors on this tree by S_t=s_i, i=1,2,....,n_T. When an event occurs, all the nodes collect sensory readings about it. Let E=e_i, i=1,2,....,t denote the sequence of events detected by the sensor nodes. We take a small portion of the events that are labeled. Our objective is to find out he labels of the rest of the events. The general idea is follows. Based on its sensory readings, each node, say s_i, divides the events into different clusters through its local clustering algorithm. After that, s_i forwards the clustering result (referred to as s_i decision, denoted by D_i) to its parent node. At each non leaf node (including the sink node), the decisions of its children nodes, together with its own decision if it has one, are further aggregated. Figure 1 gives an illustrative example of decision aggregation. As can be seen, the non leaf node s_1 aggregates the decisions of s_3 and s_4, together with its own decision on F to represent the operation of decision aggregation. Then, s_1 forwards the aggregated decision D_5 to the sink node $s0$. At s_0, D_5 is further combined with s_0 and s_2 's decisions. Finally, the global decision D is obtained.

To cluster the multiple sensor nodes data we need to label those data. According to the label we need to find data that are similar to the rest of the data or a set of data. This process needs to be least of cost as sensor nodes have limited computational power to prolong its durability. So we propose Linear Regression Model which is very lightweight calculation, can easily be implemented with C and control over its linearism. Linear regression model is a decision aggregation function that takes multiple sensor node data and classifies or clusters into a category based on their linearity.

4.2 Algorithm

Algorithm 1 Linear Regression Algorithm

```
procedure Algorithm(Y(y_1, y_2, ..., y_m))
    Calculate the average of the values of Array Y and save it in yAvg
    if the values of array Y are same then
        directly send the yAvg instead of calculating further
    else
        for each X_i ∈ X do
            temp = X[i] -xAvg
            xxavg[i] = pow(temp,2);
            sumX = sumX + xxavg[i];                          ▷ Calculate sumX
            yyavg[i] = y[i] - yAvg;
            sumY = sumY + yyavg[i];                          ▷ sumY
            yyavgSqr[i] = pow(y[i] - yAvg,2);
            sumYsqr = sumYsqr + yyavgSqr[i];                 ▷ sumXY
            XY[i] = temp * yyavg[i];
            sumXY = sumXY + XY[i];                           ▷ sumYsqr
        end for
        Calculate b1 and b0
        b1 = sumXY/sumX;
        b0 = yAvg - b1*xAvg;
        Calculate sum of Ybar - yAvg
        for each Y_i ∈ Y do
            yBar[i]= b0 + b1 * (i + 1);
            yBar_yAvg[i] = pow(yBar[i]-yAvg,2);
            sumYbar_yAvg = sumYbar_yAvg + yBar_yAvg[i];
        end for
        Rsqr = sumYbar_yAvg/sumYsqr;
        if Rsqr > 0.50 then
            send yAvg instead of 5 values
        else
            send the 5 values individually
        end if
    end if
end procedure
```

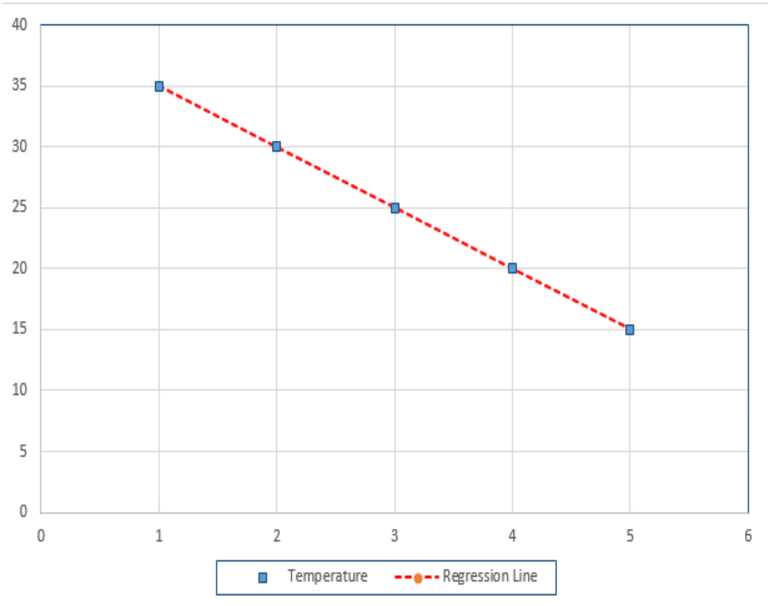
Algorithm 2 Clustering Algorithm

```
procedure Algorithm(Y(y_1, y_2, ..., y_m))
    Calculate the average of the values of Array Y and save it in yAvg

    if the values of array Y are same then
        directly send the yAvg instead of calculating further
    else

        for each y_i ∈ Y do
            temp = temp + pow((yAvg-y_i),2);

        end for
        SD = sqrt(temp);

        if SD > 3.00 then
            for each y_i ∈ Y do
                if (yAvg-y_i)> SD then
                    send y_i

                end if
            end for
            Calculate New avg of array Y in newAvg
            Send newAvg
        else
            send yAvg
        end if
    end if
end procedure
```

4.3 Linear Regression

$$y = \alpha + \beta x \qquad (1) \qquad \bar{y} = \hat{\alpha} + \hat{\beta}\bar{x} \qquad (2)$$

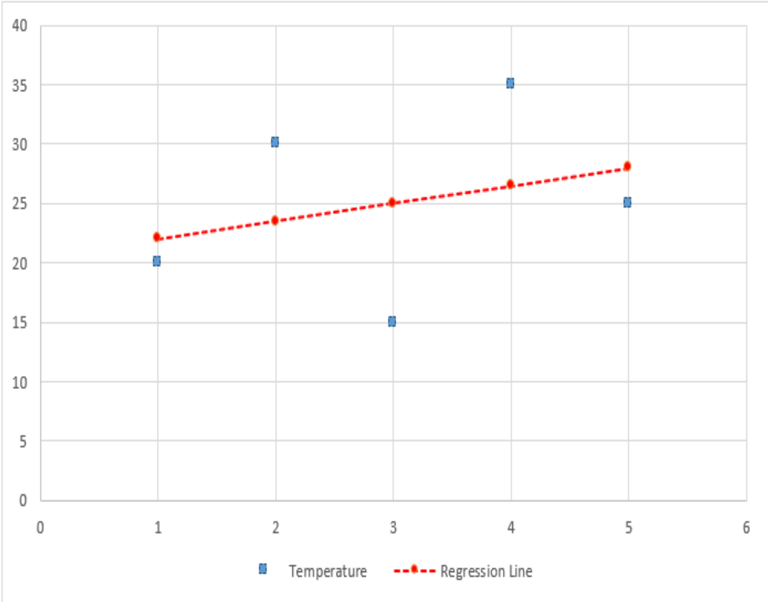
$$\hat{\beta} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \qquad (3) \qquad R^2 = \frac{\sum(y - \bar{y})^2}{\sum(y_i - \bar{y})^2} \qquad (4)$$



Here $R^2 = 1$

Figure 4.2: Linear Equation Graph.

Linear regression [5] is an approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. The case of one explanatory variable is called simple linear regression. Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine. Linear



Here $R^2 = 0.09$

Figure 4.3: Linear Equation Graph.

regression has many practical uses. Most applications fall into one of the following two broad categories:

• If the goal is prediction, or forecasting, or error reduction, linear regression can be used to fit a predictive model to an observed data set of y and X values. After developing such a model, if an additional value of X is then given without its accompanying value of y, the fitted model can be used to make a prediction of the value of y.

• Given a variable y and a number of variables X1, ..., Xp that may be related to y, linear regression analysis can be applied to quantify the strength of the relationship between y and the Xj, to assess which Xj may have no relationship with y at all, and to identify which subsets of the Xj contain redundant information about y.

We primarily used linear regression for packet aggregation considering its lightweight calculation and control over linearism.

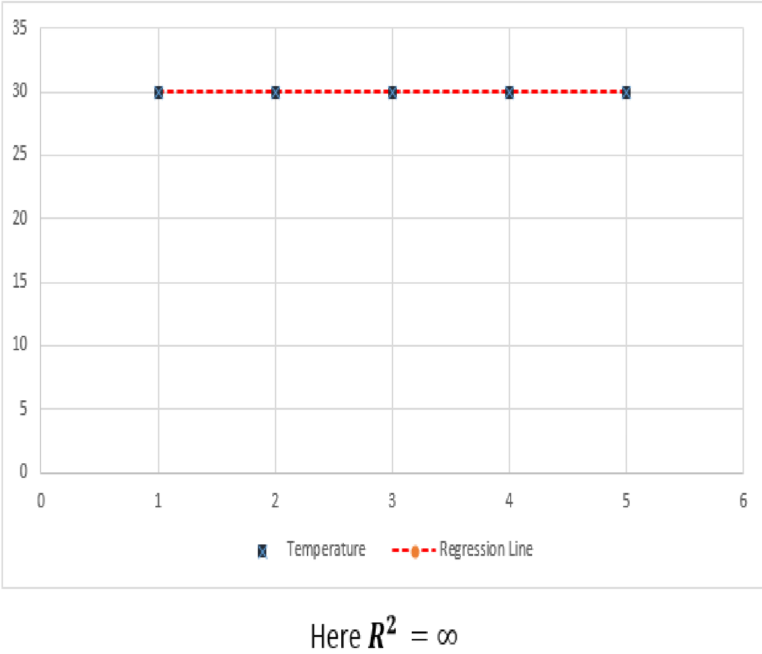


Figure 4.4: Linear Equation Graph.

4.3.1 Solution

Due to facing these problems we changed the aggregation method. We followed clustering method using Normal Distribution.

4.3.2 Why Normal Distribution

The normal distribution [6] is useful because of the central limit theorem. It states that averages of random variables independently drawn from independent distributions converge in distribution to the normal.

If any data was different from the center of the cluster value more than a threshold, then the data packet was sent without aggregation. Standard deviation (S.D.) was checked against threshold value to determine whether the packets should be aggregated or not. If S.D. is greater than threshold value, then individual packets with higher deviation than S.D. will be sent without aggregation. Rest

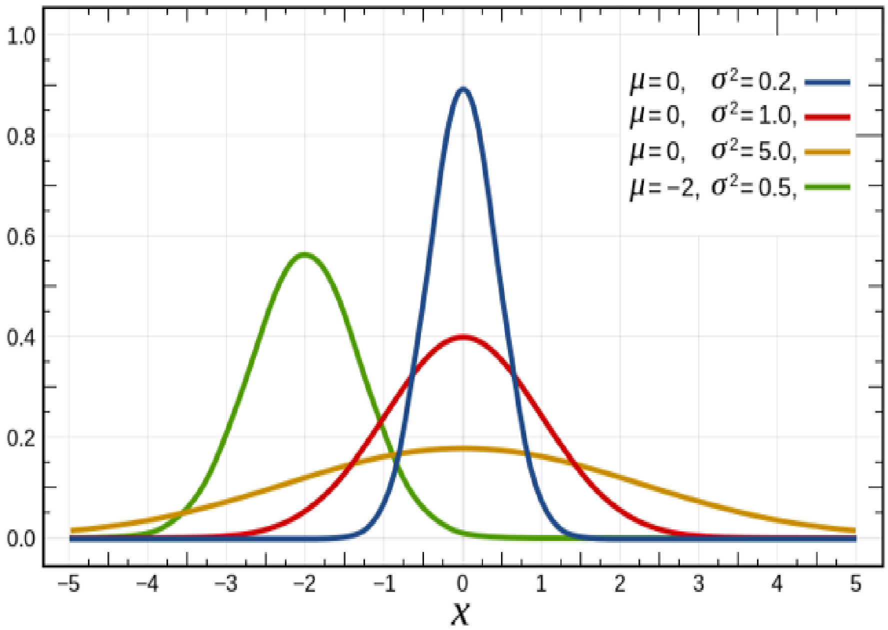


Figure 4.5: Normal Distribution.

of the packets will be sent based on their average value.

Our approach is shown in the following flow chart.

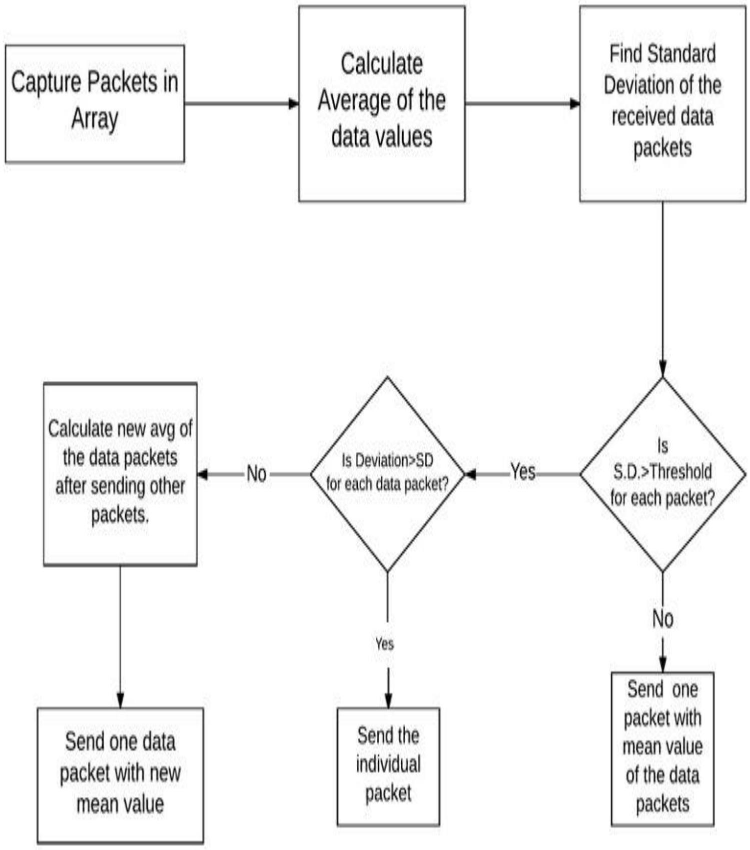


Figure 4.6: Flow Chart of Data aggregation.

Chapter 5

Implementation and Evaluation

We implemented the Aggregation of Low Priority data packets. Simulation process could not be completed due to Hardware Problem. Implementation of Preemptive Packet Train for High Priority data packets was tried but we failed. We hope to finish the simulation and overall implementation as soon as possible. The complete evaluation will be observed then.

Chapter 6

Conclusion and future works

Traffic prioritization increases channel utilization. The evaluation throughput and latency will define how better our system is. Our plan for future work is to complete the implementation and evaluate the whole system.