Effective routing with utilisation of cross-layer information 
for industrial wireless sensor networks

A THESIS SUBMITTED TO
THE SCIENCE AND ENGINEERING FACULTY
OF QUEENSLAND UNIVERSITY OF TECHNOLOGY
IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Lapas Pradittasnee
Science and Engineering Faculty
Queensland University of Technology

July 2014
Copyright in Relation to This Thesis

© Copyright 2014 by Lapas Pradittasnee. All rights reserved.

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signature: Signature

Date: 25 July 2015
To my family
Abstract

In recent years, wireless sensor networks have become an attractive solution for various types of applications. The significant improvements in micro-electro-mechanical systems (MEMS) technology enables manufacturers to create low-cost but powerful wireless sensing devices. Wireless sensor systems can be deployed in industrial application environments without the infrastructure, which wired systems require, making them an attractive solution for industrial applications. However, providing reliable data transmission in a wireless environment is a complicated task, especially in harsh conditions in industrial areas. One major issue is that the condition of the wireless channel can suddenly change from good to bad within a small period of time. Another major issue is that deploying a large number of sensor devices increases the complexity of the wireless networks. Therefore, network protocols such as routing, which is the main focus of this thesis, for industrial wireless sensor networks (WSNs) must be designed to provide reliable data transmission under such harsh conditions and also support large scale deployment effectively.

This thesis addresses the issues of scalability and reliable data transmission through effective routing with utilisation of cross-layer information in industrial WSNs. For the scalability issue, depending on the application scenarios, either proactive or reactive routing mechanisms could be employed. However, both of them have some drawbacks when implemented in large scale networks. Reactive routing will likely entail a large amount of delay due to its route discovery process; while proactive routing can create a large amount of routing overhead in the presence of a large number of sensor nodes in the network. This thesis makes use of the proactive routing mechanism with our own developments. Our design combines the proactive routing mechanism with the multipath routing mechanism, leading to a routing solution that can effectively provide prompt responses to sudden changes in the network condition. For the reliable data transmission issue in industrial WSNs, existing solutions mainly focus on supporting a single type of network
traffic. However, in many industrial WSN applications, multiple types of traffic are present in the same network, and each type of network traffic requires a different level of service. A crucial part of this thesis is to provide specific mechanisms to maintain the reliability of data transmissions for each type of traffic through effective routing in the network.

Focusing on effective routing with utilisation of cross-layer information, the major contributions of this thesis include the following three aspects, which all have been tested and validated through simulation experiments:

1. A routing framework with a two-tier routing structure and efficient route update and maintenance processes is designed. Both the two-tier structure and the route update and maintenance processes aim to support large-scale network implementation and also provide a quick response to changes in network conditions.

2. A theoretical framework is presented to mathematically analyse and evaluate the performance of the presented route update and maintenance processes. With this mathematical analysis, we are able to determine a trade-off between the amount of routing overhead and the accuracy of detecting the condition of the routing paths.

3. Four specific routing mechanisms are designed to support both periodic and sporadic traffic through effective routing in the same network for industrial WSNs. Reducing the transmission delay is the main requirement of periodic traffic, while sporadic traffic focuses on both delay and the reliability of the data transmission process.
Keywords

Wireless sensor networks, routing, cross-layer, route update, route maintenance, multipath routing
Acknowledgments

Firstly, I would like to express my deep gratitude to my principal supervisor, Professor Glen Tian, for his kindness, patience and much helpful advice during my studies. Without his guidance, it would have been impossible to complete this research work. I would like to thank my associate supervisor, Dr Dhammika Jayalath, who has also supported me through my study. Dr Dhammika not only provided much helpful advice but also helped me to improve my writing in all my conference and journal papers. I also wish to thank all my fellow graduate students in The Big Data and Networking lab for their friendship and support.

I would like to thank Dr Christina Houen for editing this thesis in compliance with the standards of the Institute of Professional Editors.

Next, I would like to thank my parents, my sister and my family back in Thailand for believing in me and always having time for me when I need their help. I also want to thank all my friends in Thailand and all friends that I met during my study in Brisbane. With their friendship, I have never felt lonely and have been able to fully concentrate on my research.

Lastly, I would like to thank all my students. All of you are one of the main reasons that I want to become a better teacher.
# Table of Contents

Abstract v  
Keywords vii  
Acknowledgments ix  
Nomenclature xv  
List of Figures xxi  
List of Tables xxiii  

## 1 Introduction  
1.1 Research Background ................................. 1  
1.2 Technical Gaps and Motivations ............................ 3  
1.3 Statement of Research Problems ............................ 5  
1.4 Significance of Research .................................. 5  
1.5 Main Contributions ..................................... 7  
1.6 Thesis Outline ......................................... 9  
1.7 Publications During my PhD Study ........................ 9  

## 2 Previous and Related Works  
2.1 Routing Mechanisms for Large-scale Industrial WSNs .......... 11  
2.1.1 Proactive and reactive routing mechanisms ............ 11
# 2.1 Routing Path Conditions

## 2.1.1 Single path routing

## 2.1.2 Multipath routing

## 2.1.3 Reliable data transmissions for industrial WSNs

## 2.1.4 Solving scalability issue in proactive multipath routing

## 2.2 Routing Path Conditions

### 2.2.1 Using single routing metric

### 2.2.2 Using a combination of routing metrics

## 2.3 Supporting Time-sensitive Traffic

### 2.3.1 Reducing interference level

### 2.3.2 Reducing congestion level

### 2.3.3 Load balancing in multipath routing

## 2.4 Chapter Summary

## 3 Two-tier Structure of Routing Framework for Large-scale Industrial WSNs

### 3.1 The Objectives of Routing Framework

### 3.2 Routing Overhead Problem in Proactive Routing

### 3.3 The Structure of the Framework

#### 3.3.1 Two-tier routing structure concept

#### 3.3.2 Combined use of periodic update and local update

### 3.4 Routing Mechanisms in the Two-Tier Structure

#### 3.4.1 The routing process in the core routing level

#### 3.4.2 The routing process in the local routing level

### 3.5 Verification Through Simulations

#### 3.5.1 Experimental design

#### 3.5.2 End-to-end delay performance

#### 3.5.3 Routing overhead performance

### 3.6 Chapter Summary
### 4 Route Update and Maintenance Processes of Routing Framework for Large-scale Industrial WSNs

<table>
<thead>
<tr>
<th>4.1 Combination of Periodic and Local Update Processes</th>
<th>48</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.1 Periodic update process based on overall quality of routing path</td>
<td>50</td>
</tr>
<tr>
<td>4.1.2 Local update process based on link quality</td>
<td>54</td>
</tr>
</tbody>
</table>

| 4.2 Determine Values of Control Parameters | 56 |

<table>
<thead>
<tr>
<th>4.3 Verification Through Simulations</th>
<th>61</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.1 Experimental design</td>
<td>62</td>
</tr>
<tr>
<td>4.3.2 Evaluation criteria</td>
<td>63</td>
</tr>
<tr>
<td>4.3.3 End-to-end delay performance</td>
<td>64</td>
</tr>
<tr>
<td>4.3.4 Packet reception ratio (PRR) performance</td>
<td>65</td>
</tr>
<tr>
<td>4.3.5 Routing overhead performance</td>
<td>69</td>
</tr>
</tbody>
</table>

| 4.4 Chapter Summary | 70 |

### 5 Theoretical Framework for Modelling the Routing Update and Maintenance Processes

| 5.1 Notations | 74 |

<table>
<thead>
<tr>
<th>5.2 Conceptual Design of the Modelling Framework</th>
<th>74</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.1 Estimate the performance of the main routing path</td>
<td>75</td>
</tr>
<tr>
<td>5.2.2 Estimate the performance of the backup routing path</td>
<td>76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5.3 Mathematical Modelling in a Routing Path</th>
<th>77</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.1 Modelling the RTS/CTS Process in Link ‘i’</td>
<td>80</td>
</tr>
<tr>
<td>5.3.2 Modelling the jth Attempt of Data Transmissions</td>
<td>81</td>
</tr>
<tr>
<td>5.3.3 Modelling the Transmission in Link ‘i’</td>
<td>82</td>
</tr>
<tr>
<td>5.3.4 Modelling the Whole Routing Path</td>
<td>82</td>
</tr>
</tbody>
</table>

| 5.4 Mathematical Modelling Backup Routing Path | 83 |

| 5.4.1 Estimate the operation time of main routing path | 83 |
| 5.4.2 Estimate the operation time of backup routing path | 88 |
# Nomenclature

## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIFS</td>
<td>Adaptive Interframe Space</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad hoc On Demand Distance Vector</td>
</tr>
<tr>
<td>AOMDV</td>
<td>Ad hoc On Demand Multipath Distance Vector</td>
</tr>
<tr>
<td>ARP</td>
<td>Address Resolution Protocol</td>
</tr>
<tr>
<td>CBR</td>
<td>Constant Bit Rate</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>DCF</td>
<td>Distributed Coordination Function</td>
</tr>
<tr>
<td>DIFS</td>
<td>Distributed Interframe Space</td>
</tr>
<tr>
<td>DSDV</td>
<td>Destination-Sequenced Distance Vector</td>
</tr>
<tr>
<td>ETX</td>
<td>Expected Transmission Count</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>IETF ROLL</td>
<td>Internet Engineering Task Force Routing Over Low power and Lossy networks</td>
</tr>
<tr>
<td>LQI</td>
<td>Link Quality Indicator</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro-Electro-Mechanical Systems</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>PRR</td>
<td>Packet Reception Ratio</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received Signal Strength Indicator</td>
</tr>
<tr>
<td>RTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>TCP</td>
<td>Transport Control Protocol</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
</tbody>
</table>

**Symbols**

- $B_a$: Average value of backoff period
- $CW$: Contention window
- $C_t$: Current time
- $D_i$: Average delay at node ‘i’
- $d(i)$: Congestion level of node ‘i’
- $d_f$: Probability that a data packet will successfully arrive at a destination node
- $d_r$: Probability that an ACK packet will successfully arrive at a source node
- $E(D)$: Average contention time at each node
- $E(ST)$: Average length of a time slot
- $E(X)$: Average number of time slots that a data packet must wait before it can be transmitted
- $e_T$: Total amount of remaining energy of all nodes in the routing path
- $e_c(i)$: Current remaining energy of node ‘i’
- $e_0(i)$: Initial energy of node ‘i’
- $H(\nu)$: Cost of traffic load for the $\nu^{th}$ routing path
- $I_j$: Number of transmitted data packets for $j^{th}$ attempts
- $k_s$: Number of received data packets at sink node
- $M_p$: Maximum number of probe packets that can be lost in a $T_e$ period
- $M_d$: Maximum number of packet drop in a $T_p$ period
- $N_s$: Expected $T_p$ period that poor path condition will be occurred for 2 consecutive periods
- $n_c$: Maximum number of retransmission for RTS packets
\( n_d \)  Maximum number of retransmission for data packets
\( o_\nu \)  Traffic ratio for the \( \nu^{\text{th}} \) routing path
\( P_{db} \)  Probability that poor routing path condition is detected in the 1\(^{\text{st}}\) evaluation period
\( P_{dm} \)  Maximum number of packet drop in a \( T_p \) period
\( P_{dx} \)  Probability that poor routing path condition is detected at the last evaluation period
\( P_{ec} \)  Probability that the event of poor link quality is detected in the current \( T_e \) period
\( P_{ex} \)  Probability that poor routing path condition is detected at the \( x^{\text{th}} \) evaluation period
\( P_{eN} \)  Probability that the event of poor link quality is detected in the \( N^{\text{th}} \) of \( T_e \) period
\( P_{nc} \)  Probability that the main routing path remains in good condition for the whole \( T_t \) period
\( P_{vt} \)  Probability that poor routing path condition is detected
\( PRR_c \)  Current value of PRR
\( PRR_p \)  PRR of the routing path
\( PRR_t \)  Threshold value of PRR
\( pb_r \)  Total number of received probe packets in a \( T_e \) period
\( pb_s \)  Total number of transmitted probe packets in a \( T_e \) period
\( Q_i \)  Quality of link ‘\( i \’ \’
\( Q_p \)  Quality of routing path
\( Q_t \)  Threshold value of link quality
\( R \)  Data transmission rate
\( r_{ij} \)  Number of packets that successfully complete RTS/CTS process of link ‘\( i \’ \’ in \( j^{\text{th}} \) attempts
\( S_i \)  Success rate of the RTS/CTS process of link ‘\( i \’ \’
\( T_b \)  Estimated time that the backup routing path is used
\( T_c \)  Contention delay
\( T_d \)  Estimated time that the source node detects the event of poor routing path condition
\( T_e \)  Local evaluation period
\( T_{ed} \)  Expected time that the number of packet drop can reach threshold value
\( T_{eN} \)  Expired time of path ‘\( N \’ \’ in the routing table
$T_{e2e}$  Average end-to-end delay of the routing path
$T_m$  Estimated time that only the main routing path is used
$T_{mt}$  Monitoring period
$T_p$  Periodic update period
$T_{rc}$  Average contention time in the case that the data packet must be retransmitted
$t_a$  Average packet inter-arrival rate
$t_s$  Average packet service time of node ‘i’ at MAC layer
$u_{ij}$  Number of unsuccessfully received data packets of link ‘i’ in $j^{th}$ attempts
$x_\nu$  Total amount of external traffic in path ‘$\nu$’
$z_\nu$  Total number of node in path ‘$\nu$’
$s_{ij}$  Number of successfully received data packets of link ‘i’ in $j^{th}$ attempts

**Greek Letter**

$\sigma$  The length of a slot time
# List of Figures

1. Research contributions ........................................... 7

2. OSI model ................................................................. 19

3. A two-tier routing structure ......................................... 37

4. A network architecture of 14 node topology .................... 40

5. Average end-to-end delay for scenario 1 with 14 nodes .......... 42

6. Average end-to-end delay for scenario 2 with 40 nodes .......... 43

7. Routing overhead for scenario 1 with 14 nodes .................. 44

8. Routing overhead for scenario 2 with 40 nodes .................. 44

9. Time diagram for periodic and local update ...................... 49

10. Concept of the combination of periodic and local update processes .... 50

11. Route update in periodic update process .......................... 51

12. Simple periodic update process ..................................... 53

13. Proposed periodic update process ................................. 54

14. Normal period and alert period .................................... 55

15. Probability of detecting poor link quality $(pb_s = 10)$ .......... 59

16. Probability of detecting poor link quality $(pb_s = 15)$ .......... 60

17. Probability of detecting poor link quality $(pb_s = 20)$ .......... 60

18. Probability of detecting poor link quality $(pb_s = 25)$ .......... 61

19. A network architecture of the simulations in NS-2 ............... 62

20. Two-state error model implementation ........................... 62
4.13 Average end-to-end delay in 50 node topology ........................................... 66
4.14 Average end-to-end delay in 100 node topology ........................................ 66
4.15 Average end-to-end delay in 200 node topology ........................................ 67
4.16 PRR of 50 node topology ........................................................................... 67
4.17 PRR of 100 node topology ........................................................................... 68
4.18 PRR of 200 node topology ........................................................................... 68

5.1 Data transmission diagram: (a) With ARP mechanism: and (b) Without ARP mechanism ............................................................. 78
5.2 A routing path for mathematical analysis ..................................................... 79
5.3 A Markov chain of the data transmission process for link ‘i’ ....................... 80
5.4 Time diagram of $T_i$ and $T_p$ ..................................................................... 83
5.5 Best case scenario ......................................................................................... 85
5.6 Worst case scenarios .................................................................................... 85
5.7 Detecting poor routing path quality at the evaluation period number 2 .......... 85
5.8 Detecting poor routing path quality at the evaluation period number 3 .......... 86
5.9 A time diagram of the operation times of the main path and the backup path .. 87
5.10 An example of routing path for calculating $T_{paw}$ ....................................... 88

6.1 Load balancing decision when using node-disjoint paths .............................. 100
6.2 Load balancing decision based on external traffic sources .......................... 101
6.3 Example of DCF operations ......................................................................... 102
6.4 Adaptive contention period mechanism ....................................................... 104
6.5 Markov chain model for normal CSMA/CA ................................................. 107
6.6 Markov chain model for adaptive contention period mechanism ............... 108
6.7 Priority queues in each sensor node ............................................................ 110
6.8 Reliable data transmission with multipath ................................................. 110
6.9 Network topology for verifying the load balancing techniques .................... 112

xx
6.10 Comparisons of network lifetime for the two load balancing techniques . . . 113
6.11 Network topology for verifying the adaptive contention period mechanism . . 114
6.12 Comparisons of end-to-end delays with CSMA/CA . . . . . . . . . . . . . 115
6.13 End-to-end delay of 100 node topology . . . . . . . . . . . . . . . . . . 116
6.14 End-to-end delay of 200 node topology . . . . . . . . . . . . . . . . . . . 117
6.15 PRR of 100 node topology . . . . . . . . . . . . . . . . . . . . . . . . . . 119
6.16 PRR of 200 node topology . . . . . . . . . . . . . . . . . . . . . . . . . . 119
List of Tables

4.1 T-test of PRR values from 50 node topology ........................................ 64
4.2 End-to-end delay in milliseconds ......................................................... 64
4.3 Routing overhead in Kilobytes ............................................................ 69
4.4 Improvement of our work in term of routing overhead ......................... 69

5.1 Notations used in this chapter ............................................................. 74
5.2 Comparisons of PRR between 100 nodes topology and 150 nodes topology . 91
5.3 Comparisons of PRR between 100 nodes topology and 200 nodes topology . 91
5.4 Comparisons of PRR between 150 nodes topology and 200 nodes topology . 92
5.5 Configuration parameters ................................................................. 92
5.6 Comparisons of PRRs between simulation and mathematical modelling .... 94

6.1 Notations used in this chapter ............................................................. 96
6.2 Parameters for verifying the adaptive contention period mechanism ........ 113
6.3 End-to-end delay in milliseconds from 100 node topology .................... 116
6.4 End-to-end delay in milliseconds from 200 node topology .................... 116
6.5 PRR from 100 node topology ............................................................. 118
6.6 PRR from 200 node topology ............................................................. 118
Chapter 1

Introduction

This thesis aims to address both the scalability issue and the reliable data transmission issue in industrial wireless sensor networks (WSNs) through effective routing with utilisation of cross-layer information. The overall objective is to improve network layer protocols in order to provide reliable data transmissions for large-scale industrial WSNs. To achieve this objective, a routing framework is developed, consisting of a two-tier routing structure and efficient route update and maintenance processes. This routing framework maintains the up-to-date routing information and also supports large-scale implementation. Moreover, specific routing mechanisms, which are based on cross-layer information, are developed to support both periodic and sporadic traffic in the same network.

This chapter starts with an introduction to the background of the research project, followed by discussion of the technical gaps and the motivation of the research, the statement of the research problems, the significance of the study, claims of the research contributions, and an outline of the thesis.

1.1 Research Background

In recent years, WSNs have become an attractive solution to various types of applications. This is due to the significant improvements in micro-electro-mechanical systems (MEMS) technology, which enables manufacturers to create low-cost but powerful wireless sensing devices [Akyildiz et al., 2002, Yick et al., 2008]. The implementation cost for large-scale deployment of application systems, such as an industrial automation system, can be considerably reduced
The total number of sensor nodes in large-scale industrial WSNs ranges from hundreds to thousands [Gungor and Hancke, 2009; Kumar et al., 2014]. However, it is more important to develop a mechanism with high capability to adapt to the changes in network size than to develop a mechanism that only works well in a specific network size [Akerberg et al., 2011]. A mechanism that effectively works in both 50-node topology and 200-node topology is better than a mechanism that only works effectively in 1000-node topology. This is mainly because the requirements of the industrial applications are likely to change with time. Some requirements may require a smaller set of sensor nodes in the area and some requirements may request a large number of sensor nodes in order to achieve their objectives [Kumar et al., 2014].

In large-scale network implementations, reactive routing protocols are likely to experience a significant amount of delay from the route discovery process [Asadinia et al., 2010; Xu et al., 2010]. The route discovery process requires each source node to transmit a route request packet to its destination node. After the destination node receives the route request packet, it will transmit a route reply packet back to the source node. After the source node successfully receives the route reply packet, it can begin to transmit a data packet. The amount of delay from the route discovery process can be notably increased in large-scale networks because the route request packet and the route reply packet must propagate through a large number of nodes before they can reach their destination.

Conversely, proactive routing protocols are unlikely to experience a long delay from the route discovery process, as in the case of reactive routing protocols [Farooq and Tapus, 2013; Trivino et al., 2013]. This is because each node that implements a proactive routing mechanism must establish the routing paths to all destination nodes at the beginning of network operation period. All of the routing paths can be used immediately without any delay when the source node has a packet to transmit. However, the route maintenance process of proactive routing protocols also creates a large amount of routing overhead, especially in large-scale network implementation.

Using proactive routing protocols with multipath routing has a high potential to effectively support the implementation of large-scale industrial WSNs. Multipath routing can provide a high level of data transmission reliability in harsh environments of industrial WSNs [Pister et al., 2009]. A proactive routing mechanism also helps to eliminate the delay from the route
discovery process. However, multipath routing also exaggerates the routing overhead problem of proactive routing mechanism. Therefore, the routing overhead problem must be solved before the combination of proactive routing protocol and multipath routing can be used.

Identifying a routing path that can satisfy the requirements of the application is a crucial process to provide reliable data transmissions in WSNs. These requirements can be satisfied by using cross-layer information. Multiple metrics from different layers of the Open Systems Interconnection (OSI) model are used to create the combination of routing metrics based on the specific requirements of each application in the network. This combination of routing metrics is used by network protocols to create the routing paths. However, most previous research studies in this area do not focus on how to effectively distribute multiple routing metrics to all participating nodes in the network, and this approach may provide an inaccurate routing path information.

In some WSN applications, multiple types of data traffic can coexist in the same network [Felemban et al., 2006]. In an industrial monitoring process, periodic traffic is used for maintaining the performance of the machine processes. Emergency traffic, which occurs sporadically, will be created when a sensor node detects a potential problem. It is used for reporting the major problem to the main controller. As a result, the emergency traffic is likely to require higher level of services from the network than the periodic traffic. Therefore, specific mechanisms must be developed to support specific requirements for each type of data traffic.

### 1.2 Technical Gaps and Motivations

In order to provide reliable data transmissions in harsh environments, the Internet Engineering Task Force Routing Over Low Power and Lossy networks (IETF ROLL) working group suggests to using multipath routing [Pister et al., 2009]. With multipath routing, each source node in the network maintains multiple routing path to a destination. If the harsh conditions cause a high error rate in one of the routing paths, the source node can switch to other available paths and continues the data transmission process without any disruption. However, multipath routing is likely to have a major implementation issue when it is used with reactive routing protocols. Implementing multipath routing with reactive routing protocols is likely to experience a long propagation delay, especially in large-scale networks. This is mainly because the route
discovery process requires a significant period of time to complete. There are two main sources of the propagation delay: time to discover multiple routing paths between each pair of a source node and a sink node; and time to complete the exchanges of a route discovery packet and a route reply packet between the source node and the sink node. This large amount of propagation delay makes the combination of multipath routing and reactive routing mechanisms unsuitable for the time-sensitive applications, such as industrial systems.

The combination of multipath routing and proactive routing mechanisms has high potential to provide good performance in term of reliable data transmissions in large-scale industrial WSNs. The proactive routing protocol is unlikely to experience the same propagation delay problems as reactive routing protocols do [Farooq and Tapus, 2013; Trivino et al., 2013]. With multipath routing, the source node can quickly adapt to the event of a broken routing path. Both the large-scale network implementation issue and the reliable data transmission issue can be addressed with this approach. However, the routing overhead problem must be solved before this combination can be implemented. This is one of the main objectives that our research aims to accomplish in this thesis.

Establishing a routing path that can satisfy the specific requirements for the applications in the networks is crucial for providing reliable data transmissions in WSNs. A cross-layer approach is one of the well-known techniques to achieve this objective [Foukalas et al., 2008]. Many studies propose many combinations of routing metrics based on cross-layer information [Daabaj et al., 2011; Djenouri and Balasingham, 2011; Wang et al., 2012; Yang et al., 2009]. For example, a combination of the success rate of data transmissions metric and interference metric is used to establish the routing path that can support a high number of received data packets at the destination node and the packets are quickly delivered to the destination node due to a low amount of interferences from other routing paths [Ashraf et al., 2008]. This type of routing path is suitable for time-sensitive applications. However, most previous works in this area focus only on what is the best combination that can accurately establish the routing path. They do not address the issue of how to effectively propagate the new routing information to other nodes in the network. Without an effective method to distribute the current routing information to all participating nodes, the value of the routing information may be outdated and it cannot be used to establish the routing path.

For industrial WSNs, there are two types of traffic that are crucial to the performance of
industrial processes: periodic and sporadic traffic. Both types of traffic are likely to coexist in the same network [Gungor and Hancke, 2009]. In order to maintain the performance of the plant processes, the network system must receive periodic data from the plant processes. The received data will be used for analysing and adjusting the operation of the processes to remain at an expected level of throughput. Sporadic data traffic is used to notify the occurrence of events or the potential failure to the main controller. This information is used to solve the problem before the plant processes are disrupted. However, previous research studies have little focus on providing reliable data transmissions for multiple types of traffic in the same network. Their solutions are based on the assumption that there is only one type of traffic in the network. We aim to provide specific routing mechanisms to provide the reliability of data transmission process for both periodic and sporadic traffic.

1.3 Statement of Research Problems

The main objective of this research is to improve network layer protocols in order to provide reliable data transmissions for large-scale industrial WSNs.

The research problems addressed in this thesis are:

- Current routing protocols for data transmissions in industrial WSNs do not effectively work with large-scale network implementation.

- Information from network layer alone does not capture the full routing dynamics for theoretical evaluation of routing performance.

- There is a lack of techniques to provide reliable data transmission for multiple traffic types in the same network.

1.4 Significance of Research

Reliable data transmission is one of the critical issues in industrial WSNs [Anastasi et al., 2011]. The critical efficiency parameters of plant equipments, such as temperature, vibration and pressure, must arrive at a sink node as soon as possible in order to solve the potential problems before they cause major disruption to the overall industrial system. However, the
condition of the wireless channel is likely to be unstable due to harsh environments in industrial area \cite{Low2005, Lu2009, Tang2009}. It is crucial for each source node to maintain the best routing path that can successfully deliver the packets to the sink node. Therefore, routing mechanisms in industrial WSNs is a significant area that can help to provide reliable data transmissions in the harsh environments of industrial WSNs.

Our research has good potential to make a significant contribution towards improving network protocols based on cross-layer information for large-scale WSNs. Many works have proposed using cross-layer information to improve the routing performance in WSNs \cite{Baoshu2010, Dehghani2008, Huang2005}. However, they only focus on determining the best combination of the routing metrics that meet their specific requirements. There still is a lack of focus on how to effectively evaluate the value of each routing metric. A different type of routing metric requires a different evaluation period in order to provide an accurate evaluation result. Most of the previous works only use a single evaluation period for all routing metrics, which may not be able to provide the best evaluation result. Our research solves this problem through an effective routing framework support.

One of the important issues of network protocols is to theoretically evaluate the condition of the routing paths. Generally, network protocols only evaluate the routing path condition based on the information from network layer. Most of the previous studies are based on the same assumption that each node only transmits one packet, which passes through the network layer, to the next-hop node per single data transmission process. However, this assumption may not provide an accurate evaluation result because each node may transmit more than one packet per single data transmission process. These additional packets come from the requirements from the data link layer, which operates below the network layer. Our research provides a method to accurately evaluate the routing path condition based on the information from both network layer and data link layer.

There are multiple classes of data traffic that coexist in the same network. Each traffic class has different levels of operational requirements \cite{Gungor2009, Kumar2014}. Different classes of data usually require a different set of services from the network. However, most of the routing mechanisms for WSNs are based on the assumption that there is only one type of data traffic in the network. Our research solves this problem by providing the specific routing mechanisms that can support the specific requirements from multiple classes of traffic.
As a result, the overall performance of the industrial systems can be significantly improved.

1.5 Main Contributions

To address the three research problems mentioned in Section 1.3, this thesis makes three main contributions as shown in Figure 1.1 and described below.

![Figure 1.1: Research contributions](image-url)
(1) A new routing framework has been developed in this thesis. It consists of two main components: two-tier routing structure, and route update and maintenance processes.

The two-tier routing structure is designed to significantly reduce the total amount of routing overhead for large-scale network implementation. It is specifically developed to support the combination of multipath routing and proactive routing protocol, which alone is likely to create a large amount of routing overhead in large-scale networks. The two-tier routing structure reduces not only the total amount of routing overhead but also unnecessary traffic, in order to obtain more bandwidth for the data transmission process.

The route update and maintenance processes are proposed to provide effective mechanisms to quickly react with the changes in network conditions due to harsh environments in industrial WSNs. They are designed to replace the simple periodic update process, which is likely to perform poorly in large-scale networks. With our proposed work, the best routing paths based on the requirements from the application can be maintained with a small amount of routing overhead. Moreover, our proposed work also provides an effective method to distribute the new routing information to all participating nodes in the network. Therefore, the major problem from the routing paths can be quickly detected and an appropriate response can be issued to solve the problem before it can cause a significant drop in the overall performance of the routing process.

(2) A theoretical modelling framework has been developed in this thesis to mathematically analyse and evaluate the performance of the proposed route update and maintenance processes. The proposed modelling framework considers the effect from the MAC layer protocols, which can influence the outcome of the network layer protocols. With the proposed theoretical modelling framework, a trade-off between the amount of routing overhead and the accuracy of detecting the condition of the routing paths can be accurately determined.

(3) To support both periodic traffic and sporadic traffic in the same network, specific routing mechanisms are proposed. This is because each type of traffic requires a different level of services. These proposed routing mechanisms are designed to be implemented in the new routing framework developed in Contribution 1. For periodic traffic, the proposed routing mechanisms notably reduce two main types of propagation delay: queuing delay and contention delay. For sporadic traffic, our proposed routing mechanisms ensure a high level of data transmission reliability in order to deliver important information to the sink node because missing the deadline can cause system failure or even disasters to industrial processes.
1.6 Thesis Outline

The thesis is organized as follows. Chapter 2 gives a comprehensive literature review of the existing research related to this project. It identifies the research gaps and describes how our research problems have been derived. Figure [1.1] in this introductory chapter illustrates how all three main contributions of this thesis provide the solutions to the research problems. Chapter 3 develops a two-tier routing structure, which is one of the main components of the proposed routing framework for proactive multipath routing in industrial WSNs. This routing framework is designed to solve the routing overhead problem. The route update and maintenance processes, which are another main component of the proposed routing framework, are presented in chapter 4. In chapter 5, a theoretical modelling framework is proposed for the developed route update and maintenance processes. Chapter 6 proposes routing mechanisms to support both periodic and sporadic traffic in the same network. Finally, chapter 7 concludes this thesis and suggests future work.

1.7 Publications During my PhD Study


• Praditasnee, L., Tian, Y.-C., and Jayalath, D. Efficient route update and maintenance processes based on path quality metric for large-scale industrial wireless sensor networks. Ready for submission to a journal.
Chapter 2

Previous and Related Works

This chapter reviews the literature from the fields of routing protocol and Quality of Service (QoS) support for industrial WSNs. In order to implement improved proactive multipath routing protocols in large-scale networks efficiently, the fundamental concepts of proactive and reactive routing protocols, multipath routing algorithms, network structure and a specific type of routing metrics for various objectives in WSNs are investigated in this chapter. The past and contemporary research studies are analyzed to contextualize the problems in industrial WSNs.

2.1 Routing Mechanisms for Large-scale Industrial WSNs

2.1.1 Proactive and reactive routing mechanisms.

There are two main types of routing mechanisms in wireless networks; proactive and reactive. Proactive mechanisms require all nodes in the network to establish the routing paths to all destination nodes and maintain this routing information in their routing table. Each node must allocate enough memory space for all entries in its routing table [Jacquet et al. 2001; Perkins and Bhagwat 1994]. On the other hand, reactive mechanisms establish the routing path when a node has a packet to transmit [Perkins and Royer 1999]. The routing path will be terminated after the data transmission process is completed.

In reactive routing mechanisms, establishing a new routing path for each data transmission process can ensure that the routing path is established based on the latest routing information. With up-to-date information, the routing path is likely to give successful delivery of the packet.
CHAPTER 2. PREVIOUS AND RELATED WORKS

to its destination [Goh et al., 2006]. Moreover, reactive routing mechanisms do not require a sophisticated route maintenance process, because the routing path will be terminated after the data transmission is completed. As a result, they can be easily implemented in a sensor node.

The major problem of reactive routing mechanisms is a significant amount of delay from the route discovery process [Asadinia et al., 2010; Xu et al., 2010]. The route discovery process requires the source node to flood a route request packet to its destination node. After the destination node receives the route request packet, it will transmit a route reply packet back to the source node. After the source node successfully receives the route reply packet, it can begin to transmit a data packet. The amount of delay from the route discovery process can be notably increased in large-scale networks because the route request packet and the route reply packet must propagate through a large number of nodes before they can reach their destination.

Conversely, proactive routing mechanisms are unlikely to experience such long delays from the route discovery process as reactive routing mechanisms do [Farooq and Tapus, 2013; Trivino et al., 2013]. In proactive routing mechanisms, each node establishes the routing paths to all destination nodes at the beginning of the network operation and then stores this routing information in its routing table. All of the routing paths can be used immediately without any delay when the source node has a packet to transmit.

A major problem of implementing proactive routing mechanisms in large-scale networks is a significant amount of routing overhead [Carrillo A and Ramos R, 2011; Shen et al., 2013]. In order to maintain an up-to-date routing information, a simple periodic update process is implemented in all nodes in the network. Each node must transmit a route update packet to its neighbor nodes for every specific period. This process must be activated even when the information in the routing table is unchanged from the previous periodic update period. The simple periodic update process can lead to a high amount of routing overhead [Shi et al., 2011]. Jiang and Garcia-Luna-Aceves [2001] conducted a performance analysis of proactive routing protocols. The results from this study confirm that the proactive routing protocols perform poorly when the size of the network is larger.

2.1.2 Single path routing.

Single path routing is commonly implemented in many routing protocols for both wired and wireless networks. It is preferable in numerous network systems because of its simplicity and
the small amount of resources required for routing operations. Each node requires to store and maintain only one routing path per destination.

Single path routing has one major drawback. It requires a significant amount of time to recover a broken routing path. After a node detects that the routing path is broken, it must initiate a route discovery process to find the new routing path. The time to complete the route discovery process is directly related to the size of the network. Large-scale networks require more time to establish a new route. No data transmission will occur until the route discovery process is completed. This long transmission delay can significantly reduce the overall performance of WSNs.

Single path routing is not suitable for implementation in industrial WSNs because it reacts slowly with the changes in network conditions. Many industrial applications require each data packet to arrive at the sink node within a specific period. In single path routing, a new route discovery process is required when the current main routing path is broken. The route discovery process may cause a significant amount of delay. As a result, the data packet may arrive at the sink node after the deadline has expired. Therefore, routing protocol for industrial WSNs must be able to quickly adapt to the changes in the routing path condition.

2.1.3 Multipath routing.

In multipath routing, each node establishes and maintains multiple routing paths per destination. There are two main approaches to establish multiple routing paths between two selected nodes: disjoint multipath routing and braided multipath routing.

- Disjoint multipath routing is a classical approach that has been used in many research studies [Alwan and Agarwal, 2010; Marina and Das, 2001, 2006; Radi et al., 2010; Xiao et al., 2010]. The main advantage of the disjoint multipath routing algorithm is its high resilience property. Each sensor node cannot be the member of multiple routing paths at the same time. When one routing path is broken, this fault event will not affect the data transmissions in other routing paths. However, the strict constraint of finding disjoint routing paths can result in a longer path length. The longer path increases both transmission delay and energy consumption level.

- Ganesan et al. [2001] have proposed braided multipath routing. The braided multipath
routing algorithm relaxes the requirement of node disjointedness by allowing the creation of partially disjoint routing paths. The average path length of this method is likely to be shorter than the disjoint multipath routing technique. As a result, braided multipath routing is likely to provide a smaller amount of transmission delay than disjoint multipath routing.

There are two main approaches for each source node to utilise multipath routing: using all paths simultaneously and using only one path at a time.

- Using all routing paths at the same time can help to balance the power dissipation among all nodes in the network. It aims to avoid using only one routing path for a long period of time.

- Using a single routing path at a time is suitable for harsh environments, which the routing path condition can drastically change within a short period of time. If the current routing path is broken, a backup path can be quickly activated to continue the data transmission process. However, it is difficult to determine the condition of the backup paths because they are not used in the data transmission process.

There are multiple types of implementation for using all routing paths at the same time. For example, Zhu et al. [2006] have proposed an additional mechanism for each source node to determine the amount of traffic per each routing path. The flow rate will be assigned to each path based on the amount of energy needed to transmit a packet through the routing path. Min and Shining [2010] have proposed assigning the traffic flow of each routing path based on residual energy, propagation delay and bandwidth.

Teo et al. [2007] have proposed the hybrid method for utilising multipath routing. They propose that each node is required to establish three routing paths to a destination. Two paths will be used concurrently and one remaining path will be reserved for backup in case one of the main routing paths is broken.

Multipath routing can provide many types of improvement to WSNs. For example, multipath routing can help extend an overall network lifetime. The data traffic from a source node can be distributed into multiple routing paths. Each routing path is responsible for a small amount of traffic load. Lower traffic load in each path means a smaller amount of energy is required for the data transmission process. Therefore, the overall network lifetime can be extended.
In addition, multipath routing can also help improve the speed of the data transmission process. It can be achieved by splitting the data packet into multiple segments and transmitting these data segments through multiple routing paths at the same time \cite{Ren_and_Yu_2005}.

There are many variations of multipath routing protocols for WSNs. Most of them are based on reactive routing \cite{He_et_al_2008b,Liang_et_al_2012,Lou_and_Kwon_2006,Tarique_et_al_2009}. Reactive routing mechanisms are likely to experience long delays from the route discovery process, especially in large-scale networks. Proactive routing mechanisms do not suffer from this type of delay as do reactive routing mechanisms. Combining a proactive mechanism with multipath routing, each sensor node maintains multiple routing paths to the sink node. If the main routing path is broken, the other routing path can be quickly activated for transmitting the data packet. No route discovery process is required. However, proactive routing mechanisms also create a large amount of overhead in large-scale network implementation. This routing overhead problem needs to be addressed before the combination of multipath routing and proactive mechanism can be implemented.

2.1.4 Reliable data transmissions for industrial WSNs.

Reliable data transmission is one of the important issues in industrial WSNs \cite{Anastasi_et_al_2011}. The critical efficiency parameters of plant equipments, such as temperature, vibration and pressure, must arrive at a sink node as soon as possible. If these parameters are analyzed in time, potential problems in the plant processes can be detected and resolved before the efficiency of the plant equipment drops or fails completely.

Providing reliable data transmissions in industrial environments is a complicated task. The mechanical structures in industrial plants usually use metallic enclosures that can deteriorate the quality of radio communications \cite{Lu_and_Gungor_2009,Tang_et_al_2009}. Moreover, the movement of the metal objects and the presence of other types of wireless systems, such as wireless LAN and Bluetooth, in industrial areas can severely affect the quality of wireless signals \cite{Low_et_al_2005}. These interferences may not completely terminate the data transmissions, but may occur for a considerable period of time. These short disruptions can result in unsuccessful transmissions and may create a considerable amount of routing overhead, in order to detect this type of event.

The Internet Engineering Task Force Routing Over Low power and Lossy networks (IETF
ROLL) working group suggests that multipath routing can be used to improve communication reliability in circumstances of unpredictable link quality [Pister et al., 2009]. Routing protocols with single path selection do not effectively perform under such conditions. With single path selection, each sensor node is required to maintain only a single path per destination. When the current routing is broken by a high level of interference, the source node must initiate the route discovery period and wait until the new routing path becomes available. No data transmission can be performed during the route discovery period. Moreover, the time for completing the route discovery can be very long when the network size becomes large.

2.1.5 Solving scalability issue in proactive multipath routing.

Reducing routing overhead.

Routing overhead is a major problem in implementing proactive routing in large-scale networks. A simple periodic update process of the proactive routing mechanism is likely to create a large amount of routing overhead. This is mainly because all information in the routing table must be included in a route update packet. Moreover, multipath routing also exaggerates the routing overhead problem because each node is required to maintain multiple routing paths per a destination, which can significantly increase the size of the routing table.

Many works propose methods to reduce the amount of routing overhead. [Chen and Nasser] [2006] and [Ben-Othman and Yahya] [2010] have proposed reducing the amount of routing overhead by piggybacking update information with the data packets. In this way, the periodic update process can be avoided and the amount of routing overhead can be significantly reduced. [Aronsky and Segall] [2010] have proposed reducing the amount of routing overhead by dynamically scheduling the update period. Their proposed method requires the sink node to keep monitoring all traffic from the source nodes. [Dai and Wu] [2005] have proposed to limit the number of nodes that can send periodic update. The node that can transmit the route update packet to its neighbor must be a node that have been sent or received the data packets above the minimum threshold value. In addition, the route update process will be activated only when a node finds the new routing path with better metric. As a result, the amount of routing overhead can be significantly reduced.
Effect of network structure.

Al-Karaki and Kamal [2004] suggest classifying the routing protocols according to network structure, as flat-based routing and hierarchical-based routing. Applications and protocols in WSNs require specific network structure. We must select the right combination of applications, protocols and network structure in order to achieve good performance from the network.

In flat-based routing structure, every sensor node is equal in functionality. Each node can either directly transmit a packet to the sink node or relay the packet to other nodes in the network until the packet reaches the sink node. Many approaches which follow this method have been proposed [Braginsky and Estrin 2002; Intanagonwiwat et al. 2003; Shah and Rabaey 2002].

The main drawback of a flat-based routing structure is its poor scalability. Many protocols using this method do not perform well in terms of latency and energy efficiency. This is because a large number of nodes means a higher amount of routing information exchanges. Each node must finish the exchanges of control information before beginning to transmit data again. Many solutions have been proposed to solve this problem, but those solutions also increase the complexity of routing protocol, making them even more difficult to implement in wireless sensor nodes.

A hierarchical-based routing structure aims to solve the scalability issue of flat-based routing [Chan and Perrig 2004; Heinzelman et al. 2000; Kumarawadu et al. 2008; Palma and Curado 2012; Sevgi and Kocyigit 2008]. This structure divides the sensor nodes into multiple clusters. The routing between different clusters will be handled by cluster heads. There is one cluster head per cluster. Each cluster head is responsible for controlling the data transmission within the cluster and in inter-clusters transmissions. There have been many research studies based on this concept [Lindsey and Raghavendra 2002; Nasser et al. 2011; Selvakennedy and Sinnappan 2007; Subramanian and Katz 2000; Wang and Chen 2013; Ye et al. 2005; Younis and Fahmy 2004].

Separating a network into multiple clusters can help to reduce the size of routing table in each node. Each node in a cluster is required to maintain the routing paths to all nodes in the same cluster, while each node in a flat-based routing structure must maintain the routing paths to all nodes in the network. A smaller size of routing table will result in a smaller size of route update packet, which can partially decrease the total amount of routing overhead [Van Der Werf and Chung 2005].
Using hierarchical-based routing structure can decreases the size of the routing table but it still cannot completely solve the routing overhead problem in large-scale network implementation. This is mainly because many protocols using a hierarchical-based routing structure only focus on how to construct the network topology. There is a lack of focus on how to effectively distribute the routing information to all nodes in the network. Implementing a hierarchical-based routing structure with a simple periodic update process still creates a large amount of routing overhead. The main source of routing overhead comes from the cluster heads, because each cluster head still has to maintains the routing paths to all cluster heads in the network.

Many research studies propose methods to reduce the amount of routing overhead. Reducing the number of control packets is one of the possible solutions to solve the routing overhead problem [Mao et al., 2010]. The implementation of a proactive routing protocol, such as DSDV, provides an option for changing the value of the periodic update period. A longer periodic update period can notably reduce the number of control packets, but it also increases the response time of the routing protocol in order to react with the changes in network condition.

In this section, it can be seen that using multipath routing with a proactive routing mechanism can provide reliable data transmissions in industrial WSNs. However, the routing overhead problem must be solved before this combination can be implemented. This is because the simple periodic process which is used by many proactive routing protocols is likely to create a large amount of routing overhead when the network size increases. This routing overhead problem is one of the major issues in the research problems that we identified in Chapter 1; Current routing protocols for data transmissions in industrial WSNs do not effectively work with large-scale network implementation. We will propose a two-tier routing structure to solve the routing overhead problem, which will be described in Chapter 3.

2.2 Routing Path Conditions

An accurate method for determining routing path condition is crucial for supporting specific requirements from each application in the network. For example, industrial applications usually requires a high success rate of data delivery [Anastasi et al., 2011]. Data retransmission must be avoided as much as possible. This is mainly because the data retransmission process can cause a data packet to miss the deadline at its destination. In order to provide a high success rate for
data transmissions, the routing path established must be based on link quality. Therefore, the routing path consists of the communication links that can provide a high success rate of data transmission, which can significantly reduce the number of data retransmissions for each data packet.

The best known method for determining the routing path conditions is to define the conditions based on a specific type of routing metrics. These routing metrics can be categorized based on the OSI model as shown in Figure 2.1. Generally, routing protocols will a routing metric based on the network layer of the OSI model, such as hopcount, for representing the routing path condition. However, many application’s requirements are too complicated to be represented by the hopcount metric. Many research studies have proposed various types of routing metric for representing the routing path condition, which will be discussed in the following section.

![Figure 2.1: OSI model](image-url)

2.2.1 Using single routing metric.

Using a single routing metric to represent the routing path condition is one of the best-known methods. Some of the routing metrics are similar to the routing metrics in wired networks. Some routing metrics, such as residual energy, are specifically defined for WSNs.
Shortest path metric.
The shortest path metric is commonly used to identify the routing path that can provide the least amount of delay. It is based on the assumption that the shorter routing path can provide a lower propagation delay when compared with the longer routing path.

There are two popular routing methods that can be used as the shortest path metric: using the number of hopcounts and using the distance between source node and sink node.

- Using hopcounts to determine the shortest routing path is the simplest method and can be easily implemented. Hopcounts are easy to determine and calculate at each node. Many routing protocols use hopcounts to select the best routing path [He, 2009; He et al., 2008a; Perkins and Royer, 1999]. The main drawback of using the hopcount method is its inability to always represent the best routing path. There are multiple issues, such as high congestion levels and a low amount of available bandwidths that can increase the total delay in the routing path. Therefore, the routing path with the lowest number of hopcounts may not be able to provide the least delay when compared with a longer routing path with lower congestion level and higher bandwidth.

- Using the distance between the source node and the sink node as the routing metric can provide a more accurate evaluation of routing path conditions than using hopcount information. Each node must determine the distance from itself to its destination. In order to achieve this distance information, a Global Positioning System (GPS) is required. Many real-time routing protocols use the distance information metric to guarantee the arrival time of the data packet at the sink node [He et al., 2003; Heo et al., 2009]. The major disadvantage of using distance information metric is that the GPS device must be installed in all nodes, which can significantly increase the implementation cost. Moreover, a GPS system cannot give very accurate results in in-door environment.

Energy metric.
The main objective of the energy metric is to find a routing path that can deliver a packet to its destination with the smallest amount of energy. This is the most common routing metric for WSNs because the energy problem is the major issue for every type of application for WSNs. The sensor node relies on a limited amount of battery power for its operations. In order to
prolong the network lifetime, choosing the routing path that consumes the least amount of energy is the best solution to the energy problem.

Energy metric in WSNs can be categorized into two categories: energy consumption and residual energy:

- Energy consumption can be calculated from a total amount of energy for transmitting and receiving a data packet [Chen and Nasser, 2006; Khalid et al., 2009]. The work of Yahya and Ben-Othman [2009] also includes the energy of the retransmission process in order to provide a more accurate evaluation outcome.

- Residual energy can be determined based on the remaining energy in each node. The routing path with a low amount of residual energy must be used less frequently than the routing path with the high amount of residual energy in order to prolong the overall network lifetime [Akkaya and Younis, 2003].

However, using a routing metric that only relies on energy consumption or residual energy can mean that the same routing path is repeatedly used. The nodes in the optimal routing path are likely to run out of battery before other nodes in the network. Therefore, the overall network performance can be significantly decreased.

Channel quality metric.
The channel quality metric is used for identifying the routing path that has a high probability of successful data transmissions [Xiao et al., 2010; Yu et al., 2007]. This routing metric can be easily derived because the current radio module in many wireless devices can detect and evaluate the channel quality parameters, such as Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI).

Liang et al. [2010] and Shuaib and Aghvami [2009] have proposed using LQI information as their routing metric. The results from the experiments show that LQI information provides a routing path with better channel quality than using only hopcounts as a routing metric. Consequently, unnecessary power consumption and wasted transmission bandwidth can be saved.

The channel quality metric can also be used to determine the distance to a specific destination. Mohajerzadeh and Yaghmaee [2009] use RSSI information to determine the distance to the sink node. This distance information is used as a part of the proposed routing metric to identify
the end-to-end delay of the possible routing path. This method is simpler than other typical methods that require information from GPS devices. However, the value of the channel quality metric can be frequently changed from time to time. This unpredictable variation may lead to instability of the routing information [Alippi and Vanini, 2004]. Moreover, the estimation process requires the sensor nodes to collect the RSSI information for a long period of time in order to provide good estimation results. This estimation process requires a lot of resources at each node and can reduce the throughput of each sensor node.

Fault incidents.

For industrial WSNs, the ability of the network to detect fault incidents in the routing path is one of the key requirements. A quick response to fault incidents enables immediate action to solve the problems before they can cause any damage to the overall system.

The type of fault incidents can be categorized based on the OSI model. For example, the main types of fault incidents, which are based on the application layer, cause node malfunction and problems with sensor measurements [Cardei et al., 2008; Jiang, 2009; Krishnamachari and Iyengar, 2003]. The main fault incident of the transport layer is the congestion event that occurs between the source node and the sink node [Akan and Akyildiz, 2005; Rajendran et al., 2004; Wan et al., 2005]. For the network layer, identifying any blocked nodes in the routing path can effectively prevent the fault event [Zhao et al., 2007]. A blocked node is a sensor node that cannot forward the packet to its next-hop node.

However, the majority of the efforts in this area focus on how to create an alternative routing path when the fault incidents in the current routing path are detected. In order to effectively respond to the fault incidents, not only the ability to establish an alternative routing path is required but the capability to distinguish the differences between the routing paths that are clearly unable to support the requirements of the application, and the routing paths that only perform poorly for a short period of time is also important for solving fault incidents in WSNs.

Success rate of packet transmission.

The success rate of packet transmission is another routing metric that can be used to identify the routing path with as high a success rate of data transmissions as the channel quality metric. Many research studies in WSNs use this routing metric in their protocols [Jafarian and Jaseemuddin, 2008; Jiuqiang et al., 2008; Zhang and Zhang, 2009].
Packet reception ratio (PRR) is one of the popular routing metrics for representing the success rate of packet transmission. The value of PRR can be calculated from the ratio of the number of received packets at the sink node and the number of transmitted packets at the source node. There are many works that use this routing metric in their routing protocols [Baoshu and Hui, 2010; Kim and Ngo, 2011].

The Expected Transmission Count (ETX) metric is another well-known routing metric for determining the success rate of packet transmission [De Couto et al., 2005]. It is based on the success rate of packet transmission in both forward and backward directions. The ETX value of each link can be calculated as shown in Equation (2.1).

\[
ETX = \frac{1}{d_f \times d_r}
\]  

(2.1)

where \(d_f\) is the probability that a data packet will successfully arrive at a receiver and \(d_r\) is the probability that an ACK packet will successfully receive at a sender.

In order to evaluate both forward and backward probability, each node in both ends of a communication link must periodically exchange the special control packets that contain the number of successfully received packets. The longer period for the estimation process will provide a better estimated result when compared with a shorter estimation time.

There are many variations of ETX metric [Huang et al., 2012; Jiuqiang et al., 2008; Radi et al., 2010; Ren and Yu, 2005; Sang et al., 2007]. The work of Jiuqiang et al. [2008] modifies the ETX metric by adding the residual energy of each sensor node. The value of residual energy helps break the tie between two routing paths that have the same ETX value. Results from the experiments show that this solution can help distribute the traffic load and increase the energy efficiency of the overall network. Sang et al. [2007] have setup the experiments to show that the backward probability does not have a high impact on the success rate. They propose to use only forward probability. This decision helps reduce the complexity of information exchanges and also reduce the amount of overhead traffic.

Using the success transmission rate as a routing metric requires a high level of processing resources at each sensor node. A sensor node must periodically exchange information with the sensor node in the other end of the link. It is required to be active longer and must allocate memory space to store the exchanged information. In addition, the correctness of the routing
metric normally depends on the amount of collected information. The result will be more accurate with a larger set of statistical information. However, it also creates more overhead traffic in the network, and may lead to a high level of energy consumption.

Many types of routing metrics have been proposed to determine the conditions of the routing path. Each routing metric is used to represent the routing path based on the specific requirements from the applications in the network. However, some application’s requirements are more complicated to represent with a single type of routing metric. Gungor et al. [2007] have proposed using both energy consumption and residual energy as a combination of routing metrics for representing the routing path condition. This combination gives more weight to energy consumption than residual energy at the beginning of the network operation. The routing path with lower energy consumption will be selected. The residual energy of each sensor node does not have significant impact on the routing path selection process because all sensor nodes still have a high level of battery power. When the residual energy falls below a threshold value, the routing protocol will give more weight to the value of residual energy information. The best routing path in this case will be the routing path with the highest summation of residual energy of the sensor nodes along the path, As a result, the combination of routing metrics can provide better performance than using only single routing information. Next, we will discuss in detail how to use the combination of routing metrics to determine the routing path conditions.

2.2.2 Using a combination of routing metrics.

Using a single type of routing metric may not be suitable for a system in a dynamic environment [Loh et al., 2009]. Generally, a single routing metric is based on information from each layer of the OSI model. The information from each layer of OSI can provide only one perspective of the network condition. For example, the information from the data link layer only provides the information within a communication link, while the information from the transport layer can provide information about a routing path that consists of many communication links. Moreover, some application’s requirements are more complicated and require multiple types of routing information from different layers of the OSI model in order to identify the routing path that can satisfy all of the requirements.

Using multiple information from different layers of the OSI model can be accomplished by applying the cross-layer concept. This concept allows direct communications between different
layers of the OSI model [Foukalas et al., 2008; Smith et al., 2009]. The shared information between multiple layers of OSI model can provide a better perspective of the routing path conditions.

Many research studies have proposed the combination of routing metrics in the cross-layer concept [Daabaj et al., 2011; Djenouri and Balasingham, 2011; Huang and Fang, 2005; Meng et al., 2010; Wang et al., 2012; Yang et al., 2009]. For example, Baoshu and Hui [2010] have proposed using a combination of routing metrics based on ETX and network load. ETX focuses on the rate of packet loss of the communication link, but lacks information about the network load. High traffic load can cause unsuccessful data transmission because some sensor nodes may drop data packets from their buffer queue due to a buffer overflow. Combining ETX and network load information may result in a better quality of the routing path. Ashraf et al. [2008] have proposed a combination of routing metrics that consist of ETX and link interference. The major objective of this combination is to identify a routing path that can provide the highest success rate of data transmissions. Moreover, this routing path delivers the data packet to the destination node with the least delay.

Congestion level can be used to discover the routing path for applications that require real-time delivery [Fonoage et al., 2010; Gungor et al., 2008]. Dehghani et al. [2008] propose a the combination of routing metrics based on congestion level and residual energy. This combination aims to establish a routing path that can satisfy both the delay requirement and energy requirement. The best routing path is the path that has the lowest level of network congestion. Moreover, the information about residual energy can be used to effectively distribute the traffic load in order to prolong the overall network lifetime.

Many research studies have proposed various combinations of routing metrics based on cross-layer information that can accurately represent the routing path conditions. However, they mainly focus on how to identify the best routing metrics that can precisely represent the routing path based on the requirements from each application in the network. There is a lack of investigations into effective distribution of the routing information to all participating nodes in the network. The distribution process of the routing information is crucial for the routing process because even if the routing information can accurately represent the routing path condition, this information can become useless when it cannot arrive at all the participating nodes on time. As the network becomes larger, this route distribution problem becomes severe, and will
cause a deterioration of the overall performance of the network. Our research overcomes this route distribution problem in large-scale networks with efficient route update and maintenance processes, which will be described in Chapters 4 and 5.

2.3 Supporting Time-sensitive Traffic

Many industrial applications usually generate time-sensitive traffic. For example, industrial monitoring applications create two types of data traffic: periodic traffic and sporadic traffic. Periodic traffic is used for maintaining the performance of the machine processes. Sporadic traffic is used for reporting the major problem to the main controller. Both periodic traffic and sporadic traffic require the data packet to arrive at the sink node as soon as possible. Therefore, the routing protocol must be able to deliver the data packets within the time threshold from both types of traffic.

Decreasing delay in the data transmission process is a critical component for supporting time-sensitive traffic. There are many types of problem that can cause delay in the data transmission process, which will be discussed in the next section.

2.3.1 Reducing interference level.

In industrial environments, interference usually occurs and may increase the overall amount of transmission delay. Moreover, interference escalates the failure rate of the data transmission process. When the data transmission process fails to deliver the data packet to its destination, the source node must retransmit the same data packet again. This retransmission process can significantly increases the overall transmission delay. Therefore, a routing path with a low level of interference is required for supporting industrial applications [Lu and Wang, 2012].

One of the well-known routing metrics to determine interference level is a total number of communication links that can interfere with others when they are active at the same time. Both of [Teo et al., 2008] and [Park and Kasera, 2005] use this interference information as their routing metric. In order to acquire this interference information, each node will count the number of links that can interfere with others. However, only the communication links that belong to the same routing path will be considered. Counting only these links may not provide enough information, because it is possible that the communication links from other routing paths can
2.3. SUPPORTING TIME-SENSITIVE TRAFFIC

also interfere with the data transmission process.

Radi et al. [2011] have proposed another type of routing metric to represent the interference level. It is based on the number of packets that each node can overhear. This interference information can provide a better perspective of the interference level. It considers all possible communication links from all routing paths. Li et al. [2013] use this type of interference metric for balancing the traffic load. However, the major disadvantages of this interference metric is that all communication links within the receiving range of each node will be counted even some of the links which may not be active at the same time. Moreover, each node is required to continuously monitor the wireless channel to gather interference information. As a result, the battery inside each node is likely to run out of power sooner and the overall network lifetime is decreased.

Interference information can be used not only to determine the routing path conditions but also to establish the new routing path. Both Nikseresht et al. [2012] and Wang and Zhang [2010] have proposed a method to establish multiple routing paths between a pair of source nodes and sink nodes. In this method, all nodes in these routing paths must be located in the positions that are out of the interference range of the nodes from other paths. They assume that the interference range is equal to two times the transmission range of each node.

Most research studies focus on using the interference metric in the route discovery process. The interference information is used to establish a routing path that does not interfere with other routing paths in the same area. However, it still lacks focus on how to effectively maintain the best routing path in the routing table of each node. In industrial environments, the condition of the wireless channel can drastically change within a short time. The most effective mechanism to maintain the best routing path is just as important as the route discovery mechanism.

2.3.2 Reducing congestion level.

The congestion level is one of the major factors that can cause a significant delay in wireless networks. When the level of congestion is high, the data packet will likely wait in the buffer queue for a long period of time. In the worst case scenario, the data packet can be dropped.

The main objective of the routing information based on the congestion level is to identify the congestion areas in the network.
There are various types of routing information that can be used to represent the level of congestion. A popular parameter is the forwarding capability. Wang et al. [2007] define the congestion level of node \( i \) \((d(i))\) as

\[
d(i) = \frac{t^s_i}{t^a_i}
\]

where \( t^s_i \) is the average packet service time at MAC layer of node \( i \) and \( t^a_i \) is the average packet inter-arrival rate. Many previous studies also use a similar type of parameter to evaluate the level of congestion [Li et al., 2013; Monowar et al., 2008; Sheu and Hu, 2008].

Teo et al. [2008] use the value of the remaining buffer of each node in the routing path to identify a routing path that will be unlikely to experience a high level of congestion. Thus, the source node can assign more traffic into this routing path in order to balance the energy consumption between all available routing paths.

The congestion metric can also be used to determine the transmission rate for each routing path [Monowar et al., 2008]. However, a careful consideration must be made because multiple routing paths will be used simultaneously. A trade-off between the energy cost of spreading traffic and the spatial balance of energy in the network must be made [Baek and de Veciana, 2007].

Detecting the congestion level in the network is only the first step for solving the congestion problem. After this problem is detected, the congestion control mechanism must be activated to decrease the congestion level back to the normal condition.

The basic congestion control mechanism is to decrease the data transmission rate at the source node and clear the buffer queue in all nodes along the routing path [Monowar et al., 2008; Teo et al., 2008].

Li et al. [2013] have proposed an interesting congestion control mechanism. Three levels of threshold values are defined at each node \((i)\) for different levels of congestion. Normally, there are only two possible congestion levels: normal and congest.

- Light level of congestion \((0.8 < d(i) < 1)\)
- Medium level of congestion \((0.4 < d(i) < 0.8)\)
- High level of congestion \((d(i) < 0.4)\)
A light level of congestion is acceptable. For a medium congestion level, the detected node will clear the buffer and report to its parent node to reduce the data rate. For a high level of congestion, the detected node will report to the source node to modify the transmission schedule or discover the alternative routing path. Sheu and Hu [2008] have proposed a method for predicting the level of congestion by using the value of congestion metric from neighbour nodes. They aim to prevent the congestion before it occurs. However, this prediction method only works in a hop-by-hop approach, which may not be suitable for large scale networks.

2.3.3 Load balancing in multipath routing.

The load balancing technique is one of the solutions that helps decrease an overall delay when multipath routing is implemented in the network. A source node will divide a total amount of data traffic into multiple portions. Each portion of data traffic will be transmitted to the sink node via a selected routing path. Transmitting a small chunk of data packets simultaneously through multiple routing paths will likely deliver all data packets to the sink node more quickly than transmitting all data packets through a single routing path.

There are various types of load balancing implementation in WSNs. Yang and Heinzelman [2011] have proposed a method to improve both transmission reliability and network lifetime. A subset of all possible routing paths is selecting for simultaneously transmitting the data packets to the sink node. Moreover, the Forward Error Correction (FEC) technique is used for correcting errors at the sink node. The routing paths that are not active in the data transmission process will operate in sleep mode to conserve energy.

Using multiple routing paths at the same time also increases the level of interference between the routing paths. Li et al. [2013] have proposed adding the interference level information into their load-balancing criteria parameters. With interference information, the source node can allocate more traffic to the routing path that has a low level of interference.

Radi et al. [2011] have proposed a combination of routing metrics that helps to distribute the data traffic based on multiple objectives. This combination of routing metrics consist of interference level, remaining battery and ETX. More data traffic will be allocated to the routing path that has a high battery level, a high level of success rate of data transmissions and a low level of interference. Sheyibani et al. [2012] and Alwan and Agarwal [2013] have proposed useful routing information that can be useful for determining the amount of traffic per each
routing path, such as signal to noise ratio ($SNR$) and the available buffer at each node.

The discussions in this section have shown that most of the previous studies are based on the assumption that there is only one type of data traffic in the network, which we identified in Chapter 1 as one of the research problems for this thesis. For industrial WSNs, multiple types of data traffic coexist in the same network. Therefore, our research provides a specific level of support for each type of traffic. We propose specific routing mechanisms for both periodic and sporadic traffic in Chapter 6.

### 2.4 Chapter Summary

Through the literature review, many important routing mechanisms for WSNs are studied. A combination of multipath routing and proactive routing mechanisms has a high potential to provide reliable data transmissions in large-scale industrial WSNs. However, multipath routing also exaggerates the routing overhead problem, especially when a network becomes large. This routing overhead problem must be solved before a combination of multipath routing and proactive routing mechanisms can be implemented.

In this chapter, research to determine the routing path conditions based on the specific requirements of the applications has been reviewed. A single type of routing information or cross-layer routing information can be used to represent the current routing path condition. However, a technical gap remains to be filled. Most of the research studies only focus on how to find the routing metrics that correctly represent the routing path conditions. There is a lack of focus on how to distribute the routing information to all nodes in the network. If the routing information cannot transmit to all participant nodes in a timely way, the routing information can be outdated and cannot accurately represent the routing path condition.

The research studies described in Section 2.3 present different types of techniques to support time-sensitive traffic. However, most previous studies are based on an assumption that there is only one type of data traffic in the network. For industrial WSNs, there are multiple types of data traffic in the same network. Thus, this technical gap needs to be solved.

This thesis fills the technical gaps that the literature cited above has not yet filled. In order to maintain reliable data transmissions in large-scale industrial WSNs, three main components are developed in this thesis:
(1) A routing framework is designed to effectively reduce the total amount of routing overhead and provide a quick response to the changes in network conditions. It consists of a two-tier routing structure and efficient route update and maintenance processes.

(2) A theoretical modelling framework is developed for the proposed route update and maintenance processes. It provides mathematical models to evaluate the routing performance in large-scale WSNs.

(3) Specific routing mechanisms are presented to provide specific levels of service for both periodic and sporadic traffic. This is to fulfill the specific requirements for each type of traffic.
Chapter 3

Two-tier Structure of Routing Framework for Large-scale Industrial WSNs

For large-scale deployment, an exclusive focus on reducing the amount of routing overhead may not be the best solution. Filtering the routing control traffic can significantly improve the overall performance of the network. The clustering concept can be extremely helpful to accomplish this task. Use of the clustering algorithm can reduce the size of the routing table stored at the sensor nodes that are assigned as cluster members. These sensors are required to maintain a connection with their cluster head. Conservation of the communication bandwidth can also be improved, since inter-cluster communication is limited to the cluster heads. Finally, applying the clustering algorithm can increase the stability of the network topology. The cluster members focus on the routing path to their cluster heads, and are not affected by any changes in the inter-cluster network.

A proactive routing protocol is likely to perform better than a reactive routing protocol in a large-scale network implementation. A reactive routing protocol experiences long transmission delays because the source node must wait until the route discovery process is completed before the data packet can be transmitted. To complete the route discovery process, the source node must transmit a route request packet to the sink node and wait for a route reply packet. In large networks, both the route request packet and the route reply packet require a significant amount of time to propagate to their respective destinations. By contrast, a proactive routing protocol does not experience a long delay in the route discovery process because all routing paths are established at the beginning of the network operation. As a result, the source node can begin to transmit data packets immediately after packets become available.
However, the major problem in implementing a proactive routing protocol in a large-scale network is a high amount of routing overhead caused by the route update and maintenance processes. Jiang and Garcia-Luna-Aceves [2001] have compared the performance between proactive and reactive routing protocols in terms of routing overhead. The results from their study show that the total amount of routing overhead from a proactive routing protocol is significantly increased when the network size becomes larger. This is due to the simple periodic update mechanism that many proactive routing protocols use. In the simple periodic update mechanism, each sensor node must send periodic route update packets to all of its neighbour nodes, even if the routing information is similar to the previous update.

To make use of the advantages of proactive routing while overcoming its disadvantages in large-scale implementation, a new routing framework is presented in this chapter based on a two-tier routing structure and an efficient route update and maintenance processes. The main objective of this routing framework is to reduce the amount of routing overhead from each sensor node in order to support large-scale implementation. Another objective is to eliminate the simple periodic update process implemented in many proactive routing protocols because this process is not suitable for large-scale deployment.

In this chapter, the main concepts of a two-tier routing structure and an efficient route update and maintenance process are explained to provide a big picture of how our proposed routing framework can solve the first research problem that we defined in Chapter 1: current routing protocols in industrial WSNs do not work effectively in large-scale network implementation. This chapter will focus on how our proposed routing framework solves the routing overhead problem with a two-tier routing structure. Effective route update and maintenance mechanisms will be discussed in detail in Chapter 4.

### 3.1 The Objectives of Routing Framework

The proposed routing framework has two main objectives: (1) reducing the total amount of routing overhead; and (2) providing an effective route update and maintenance mechanisms.

In this chapter, we focus on the first objective by developing a two-tier routing structure for supporting large-scale network implementation.
3.2 Routing Overhead Problem in Proactive Routing

A simple route update process in the proactive routing protocol is the root cause of the routing overhead problem. In the simple periodic update process, each node in the network must transmit the route update packets to its neighbours every specific period of time. This periodic update must be activated even when the information in the routing table is unchanged from the previous period. As a result, many unnecessary route update packets are likely to be generated, which increases the total amount of routing overhead.

Moreover, each node that uses the simple periodic update process must include all information within the routing table in the route update packet. As each node must store the routing paths to all other nodes in the network in its routing table, the size of the route update packet is increased when the size of network become larger.

With the unnecessary route update packets and the massive size of each route update packet, it can be concluded that the simple route update process can create a substantial amount of routing overhead in large-scale network implementations.

Many research studies have proposed methods to reduce either the size of the routing table or unnecessary route update packets. Aronsky and Segall [2010] have proposed to reduce the amount of routing overhead by dynamically scheduling the update period, but their proposed method requires the sink node to keep monitoring all data traffic, which can significantly decrease the battery power of each node. Dai and Wu [2005] have proposed limiting the number of nodes that can send periodic updates. The node that can transmit the route update packet to its neighbour must be a node that has been sent or has received data packets above the minimum threshold value.

However, previous studies in this area do not provide a solution that is suitable for industrial WSNs. Most previous studies only aim to reduce the total amount of routing overhead by decreasing the number of times that each node transmits the route update packets. There are two major drawbacks of this approach: (1) the response time of the network is likely to increase; (2) the routing path conditions may not be accurately evaluated.

For industrial WSNs, it is crucial to have effective routing mechanisms that quickly react with the changes in the network because the wireless channel in industrial area is unstable due to the harsh environments. Therefore, the routing framework proposed in this chapter is designed
to solve the routing overhead problem, and can also provide effective methods to accurately detect and quickly react to the changes in network conditions.

3.3 The Structure of the Framework

3.3.1 Two-tier routing structure concept.

The proactive multipath routing mechanism is selected to provide reliable data transmissions in industrial WSNs as mentioned in Chapter 1. Multipath routing requires each node to store multiple routing paths per each destination, which significantly increases the size of the routing table and the overall amount of routing overhead.

A two-tier proactive multipath routing structure is designed to solve the routing overhead problem. Separating a network into two levels can reduce both the average size of the routing table in each node and the average size of the route update packet.

In this structure, the sensor nodes in the upper-tier level are assigned the role of core routing operations as shown in Figure 3.1. The sensor nodes in the core routing level are responsible for establishing and maintaining multiple routing paths to the sink node. The sensor nodes in the lower-tier or local routing level only focus on maintaining a connection to the closest sensor node in the core routing level.

In order to reduce the size of the routing table in each sensor node, the number of nodes in the core routing level must be smaller than the number of nodes in the local routing level. This is mainly because the node in the core routing level must maintain multiple routing paths per each destination, while the node in the local routing level only maintains a single routing path to a sensor node in the core routing level.

The two-tier routing structure also helps to limit the areas that the route update packets can be propagated into. The route update packets from the core routing level will not be relayed to the local routing level or vice versa. Limiting the number of route update packets in each routing level can increase the overall bandwidth for the data transmission process.
3.3. Combined use of periodic update and local update.

A combination of periodic update and local update is developed to replace the simple periodic update process, which is a root cause of the routing overhead problem in the proactive routing protocol. It aims to provide an effective method to update and maintain the routing information in the core routing level of the two-tier routing structure. Moreover, the combination of periodic update and local update also creates a small amount of routing overhead, which is suitable for large-scale network implementations.

The combination of periodic update and local update aims to support the use of multiple types of routing metrics to determine the routing path condition. The periodic update process is designed for a routing metric that requires a long period of time to obtain the value of the metric. The local update process is designed for a routing metric that requires a short period of...
time to evaluate the value of the metric.

The combination of periodic update and local update is based on an assumption that each type of routing metric requires a different amount of time to obtain the accurate value of the metric. For example, the value of packet reception ratio ($PRR$) requires a significant period of time to evaluate, while the event of packet drop can be obtained within a short period of time. Most previous studies evaluate the values of all routing metrics within the same specific period of time. As a result, some values of the routing metrics may not accurately represent the conditions of the routing path.

3.4 Routing Mechanisms in the Two-Tier Structure

In a two-tier routing structure, the sensor nodes in the core routing level will have a higher workload than the sensor nodes in the local routing level.

In order to conserve the energy of the sensor node, the routing process is divided into multiple rounds. At the beginning of each round, each sensor node determines that it will become the core routing node or the local routing node. All sensor nodes maintain their respective roles until the end of each round and repeat the same process when the next round begins.

Each sensor node will determine that it can become the core routing node based on its distance from the sink node, current residual energy and its role in the previous round. If this sensor node has been acting as a core routing node in the previous round, it will have a lower possibility to become the core routing node again in the next round.

3.4.1 The routing process in the core routing level.

Each node in the core routing level is required to establish and maintain multiple routing paths from itself to the sink node. There are two possible types of routing paths that can be created: node disjoint path and link disjoint path. In this chapter, we describe node-disjoint routing paths.

After the multiple routing paths are created and stored in the routing table, the core routing
node will broadcast the special route control packet to all of its neighbor nodes. When a local routing node that still does not establish any connection to a node in the core routing level receives the special route control packet, this local routing node will send an ACK packet back to the core routing node to establish the connection to the core routing level.

There are two possible approaches to implementing multipath routing. One approach is to select only one routing path as a main routing path to the sink node. Other routing paths will become backup paths. The other approach is to use all of the routing paths at the same time. In this chapter, the first approach is implemented at the core routing level.

### 3.4.2 The routing process in the local routing level.

After the local routing node establishes the connection with the closest core routing node, the connection with the core routing node will be updated via a “hello” message. The “hello” message will be exchanged in every specific period of time. If the hello messages from the core routing node are missing for a specific period, the local routing node will terminate the connection with that core routing node and then broadcast the special route control packet to create a new connection with other core routing nodes in the same area.

By using a hello message, the route update and maintenance processes in the local routing level will not create many routing overheads in the core routing level. The hello message from the local routing node will be processed and terminated at the core routing node that it established the connection with.

### 3.5 Verification Through Simulations

#### 3.5.1 Experimental design.

Two simulation scenarios are designed to evaluate the performance of the proposed route update and maintenance processes:

1. The first simulation scenario has 14 sensor nodes. The sensor nodes are deployed in a square area of 150×150 m. There are 6 sensor nodes in the core routing level and 8 sensor nodes in the local routing level as shown in Figure 3.2.
2. The second simulation scenario uses 40 sensors deployed in a square area of 200x200 m. There are 15 sensor nodes in the core routing layer and 25 nodes in the local routing layer. This scenario is based on the same network architecture of the 14-nodes topology, but the number of hops between the source node and the sink node is increased.

![Network Architecture of 14 Node Topology](image)

**Figure 3.2:** A network architecture of 14 node topology

All simulations are carried out using the NS-2.34 simulator.

Both scenarios share the following configuration parameters. IEEE 802.11 is used as the MAC layer protocol. The network communication model is Two-Ray Ground. There are two CBR traffic sources to generate data packets into the network. CBR1 will generate a data packet with the size of 100 bytes every 15 seconds, while CBR2 will generate a data packet with the same size every 10 seconds. Both traffic sources begin to transmit packets for 30 seconds after the simulation start. The transmission range of each sensor node is 40m. There is one sink node for both simulation scenarios, and the sink node is located at the centre top of the simulated sensor area.
3.5. **VERIFICATION THROUGH SIMULATIONS**

Furthermore, a two-state error model is implemented in both simulation scenarios. One wireless channel remains in good condition for 1000 seconds and then changes to a poor condition (error rate = 0.9) for the next 1000 seconds. This process repeats in the same manner for the whole timespan of each of the two simulation scenarios. The simulation timespan has been set to 4000 seconds for all scenarios.

In order to evaluate the performance of our proposed work, we need to compare the outcomes of our proposed work with other available routing protocols. Most of the routing protocols for industrial WSNs do not work well in large-scale networks, which is one of our defined research problems. This is mainly because they do not focus on how to effectively distribute the routing information to all nodes in the network. Therefore, we choose the three best known routing protocols for WSNs as benchmarks. DSDV is selected to represent proactive routing protocols. AODV and AOMDV are selected to represent reactive routing protocols.

In each of the two simulation scenarios, a proposed routing framework is implemented in the NS-2 simulator. The performance to be evaluated includes: end-to-end delay and routing overhead.

### 3.5.2 End-to-end delay performance.

**Scenario 1 with 14 nodes.**

The first simulation scenario with the topology of 14 sensor nodes is investigated. Figure 3.3 shows average end-to-end delay and its variation range at 95% confidence level for the proposed hierarchical routing structure. The end-to-end delay is the average delay of the mean delay values from 8 independent simulation runs with the same configuration parameters. For comparison, Figure 3.3 shows the performance and its variation ranges at the same confidence level for DSDV, AODV and AOMDV protocols as benchmarks.

Figure 3.3 shows that the routing protocol based on our proposed routing mechanisms and the proactive routing protocol DSDV perform better in average end-to-end delay than the reactive routing protocols AODV and AOMDV. The delay variation ranges of the two reactive routing protocols AODV and AOMDV are much larger than those of the proposed model and DSDV. This is because reactive routing protocols must perform the route discovery process every time the source node has a packet to transmit, and the routing path from a source to a sink node can change dynamically. Some routing paths may have more hops to reach the sink node,
and a longer routing path will likely require a longer period of time to complete a data packet delivery.

**Scenario 2 with 40 nodes.**

Figure 3.4 shows the average end-to-end delay for the second simulation scenario with a topology of 40 nodes. The routing protocol based on the proposed routing mechanisms and the proactive routing protocol DSDV still outperform the reactive routing protocols AODV and AOMDV in terms of the delay performance.

The performance of the DSDV protocol drops significantly in comparison with that from the first simulation scenario with the topology of 14 nodes. This is because when the network size becomes larger, the information about a broken link requires a longer period of time to propagate back to the source node.

The proposed work still provides the best performance in terms of the end-to-end delay when compared with the other three benchmark routing protocols in both simulation scenarios. It performs better than DSDV in a larger size of network because it uses multipath routing. When the main routing path is broken, an alternative path can be activated without delay to replace the broken path.
3.5. VERIFICATION THROUGH SIMULATIONS

3.5.3 Routing overhead performance.

Scenario 1 with 14 nodes.

Figure 3.5 shows the overall routing overhead for the first simulation scenario with the topology of 14 nodes. It shows that DSDV creates the minimum amount of routing overhead when compared with other three routing protocols. The main reason is that AODV, AOMDV and the proposed model implement a regular hello packet. The hello packet will be periodically transmitted to adjacent nodes every 5 seconds, while the DSDV protocol does not implement such a hello packet.

Scenario 2 with 40 nodes.

However, the results from the second simulation scenario with the topology of 40 nodes, as depicted in Figure 3.6, show that the DSDV protocol becomes the routing protocol that generates the highest amount of routing overhead among all four routing protocols under consideration. The sensor node in DSDV implementation must include all entries in its routing table in its route update packet. When there is a larger number of sensor nodes in the network, the overall routing overhead will significantly increase. With the routing protocol based on our proposed routing mechanisms, only the sensor nodes in the core routing level participate in the route
Figure 3.5: Routing overhead for scenario 1 with 14 nodes

update process. Therefore, it can perform much better than the other three routing protocols and can give the lowest amount of routing overhead.

Figure 3.6: Routing overhead for scenario 2 with 40 nodes

3.6 Chapter Summary

Different applications of WSNs require different levels of services. The routing protocols used in WSNs must work effectively with unpredictable wireless channel conditions in harsh environments. For reliable wireless communications in large-scale industrial WSNs, a new
routing framework has been presented for proactive multipath routing. It consists of a two-tier routing structure and efficient route update and maintenance processes. The results from the simulation studies have demonstrated the effectiveness of the two-tier routing structure. It has been shown that our proposed mechanism outperforms all other three popular routing protocols DSDV, AODV and AOMDV as benchmarks in the sense that it provides the lowest amount of routing overhead particularly for large-scale networks. The performance improvement becomes significant in large-scale networks. As a result, the scalability issue has been well addressed in the proposed two-tier routing structure.
Chapter 4

Route Update and Maintenance Processes of Routing Framework for Large-scale Industrial WSNs

For industrial WSNs, maintaining the routing path for a high level ratio of packet delivery is one of the key objectives. In order to maintain the performance of a sensor network application such as an industrial system, the main controller of the system requires that critical information from the sensors arrives at the controller within a specific period of time. If many data packets are lost during the transmissions due to unreliable transmission routes, the recovery mechanism of the system must be activated to transmit the lost packets from the source node again. As a result, the total amount of transmission delay will increase significantly, resulting in degradation of the overall performance of the sensor network system because many data packets cannot arrive in a timely way at the sink node.

Providing all nodes in the network with the current information about the routing paths is a complicated task, especially in industrial environments [Hu et al., 2004]. Harsh conditions in an industrial area can create a sudden change in the quality of the communication links. The link conditions can change from good to poor in a short period of time. This type of event means there is additional complexity in the task of maintaining the freshness of all routing paths in the network.

Both reactive routing protocols and proactive routing protocols use specific approaches to establish and maintain the freshness of the routing paths. Each of these approaches has its advantages and disadvantages.
For reactive routing protocols, a routing path is established only when the source node has the data packets to transmit. Establishing a new routing path every time to transmit a data packet can ensure that the routing path is established based on the latest routing information. However, it also causes a considerable delay before the source node can begin to transmit a data packet.

In proactive routing protocols, each node establishes all routing paths at the beginning and then maintains the routing information periodically. As a result, proactive routing protocols do not experience the same type of delay as reactive routing protocols do. This is because the source node can begin the data transmission immediately when the source node has a data packet to transmit. However, this is achieved with the cost of routing overhead, which may be significant for large-scale sensor networks.

In order to maintain the freshness of the routing path, a well designed routing metric is important. The routing metric should be able to accurately evaluate the condition of the communication links in wireless environments.

Many types of routing metrics have been proposed and employed in various applications. However, none of these routing metrics address how to propagate the new routing metric to other nodes to determine the current condition of the routing path. Without an effective method for transmitting the current routing information to all participating nodes, the current value of the routing information may be outdated.

In this chapter, new route update and maintenance processes based on the combination of periodic update and local update are proposed. The main objective of the proposed work is to provide accurate information about the condition of all routing paths. Further, it also effectively distributes the new routing information to all participant nodes with a small amount of routing overhead.

4.1 Combination of Periodic and Local Update Processes

The main concept of the combination of periodic and local update processes is to provide both a mechanism for detecting sudden changes in network conditions and a mechanism for determining the accurate condition of the routing path. The periodic update process is responsible for evaluating the routing path condition and the local update process handles the process of detecting changes in the network condition.
The periodic update process evaluates the condition of the routing path every $T_p$ seconds. $T_p$ is the value of periodic update period. $T_p$ is set to a large value in order to provide the proper amount of time for determining the routing path condition.

The local update process is designed for supporting the periodic update process in case sudden changes in the network condition occur. The periodic update process cannot quickly react to the change due to its long evaluation period. The local update process uses a small evaluation period ($T_e$) for determining the link condition. There are multiple $T_e$ periods inside a $T_p$ period as shown in Figure 4.1. This setup aims to ensure that if there is a significant drop in the link quality, the local update period will be able to successfully detect this event before the end of the $T_p$ period.

The combination of periodic and local update processes is based on cross-layer information. The periodic update process uses the value of packet reception ratio (PRR), which is the information from the application layer of the OSI model. The local update process is based on the value of link quality, which can be derived from the data link layer of the OSI model.

![Figure 4.1: Time diagram for periodic and local update](image)

In the combination of periodic and local update processes, each routing path has two states: normal or alert. Initially, all routing paths are set to the normal state by default. The state of the routing path will change from normal to alert when the evaluation result from the periodic update process is below the threshold value. The routing path that is set to alert state will be terminated if the routing path condition from the next periodic update period is still lower than the threshold value, as shown in Figure 4.2.

When the current main routing path is in the alert state, the source node will use both the current main routing path and the backup routing path to transmit the data packets to the sink node for the next $T_p$ period. The data packet is transmitted into both routing paths at the same time in order to provide high reliability data transmissions. If the status of the main routing path is changed from alert to normal, the source node will use only the main routing path for the data transmissions.
The route update and maintenance processes based on the combination of periodic update and local update will be implemented in the core routing level of a routing framework, as described in Chapter 3.

4.1.1 Periodic update process based on overall quality of routing path.

The main objective of the periodic update process is to accurately evaluate the overall quality of the routing path. Packet reception ratio (PRR) is selected as the routing metric for the periodic update process. The value of PRR is calculated based on the number of successfully received data packets at the sink node. Therefore, it accurately represents the performance of the routing
4.1. COMBINATION OF PERIODIC AND LOCAL UPDATE PROCESSES

path in terms of reliable data delivery.

PRR is a routing metric that is suitable for industrial applications. This is mainly because most industrial applications require the data packet to arrive in a timely manner at the sink node. A routing path with a high value of PRR is unlikely to experience a long delay in the data retransmission process. As a result, such a path is likely to deliver the data packets to the sink node within the specific deadline.

In each periodic update period($T_p$), each source node creates a route update packet and transmits it through all routing paths, as shown in Figure 4.3. When the route update packet arrives at each node in the routing path, each node will update its current routing information and relay the route update packet to its next-hop node until the route update packet reaches the sink node. Then, the sink node will process the route update packet and create an ACK packet to transmit back to the source node. The estimated value of the routing metric is included in the ACK packet.

![Route update in periodic update process](image)

**Figure 4.3**: Route update in periodic update process

The main responsibility of the periodic update period is to determine that the current routing path still provides a PRR value above the threshold value. If the routing path provides a PRR value lower than the threshold value for two consecutive periodic update periods, it will be terminated.

The algorithm of the periodic update process is presented in Algorithm 1.

At the end of each periodic update period ($T_p$), the sink node evaluates the value of PRR from the total number of received data packets in the current periodic update period. Next, it compares the current value of PRR ($PRR_c$) with the threshold value from the application layer’s requirement ($PRR_t$). If $PRR_c$ is lower than $PRR_t$, the sink node starts to consider that the routing path is likely in poor condition and sets the state of the routing path to the alert state.
If the $PRR_c$ of the next periodic update period is still lower than $PRR_t$, the sink node will set the value of the ‘terminatePath’ field in the header of the route update packet as ‘TRUE’ and send the route control packet to the source node to terminate the current routing path.

**Algorithm 1** Periodic update based on PRR

1: **At sink node every** $T_p$ **seconds**
2: Evaluate the value of PRR for each routing path
3: **if** $PRR_c >= PRR_t$ **then**
4: Send update packet to source node
5: **if** $vCount >= 1$ **then**
6: $vCount := 0$
7: **end if**
8: **if** $alertState == TRUE$ **then**
9: $alertState := FALSE$
10: **end if**
11: **else**
12: $alertState := TRUE$
13: **if** $vCount == 2$ **then**
14: $terminatePath := TRUE$
15: Send update packet to terminate this routing path
16: **else**
17: $T_p := T_{ed}$
18: increment $vCount$
19: **end if**
20: **end if**

21: **At intermediate node** When a node receives an update packet
22: **if** $terminatePath == FALSE$ **then**
23: $PathExpire := Current\_time + (1.5 \times T_p)$
24: Forward the update packet to source node
25: **else**
26: $PathExpire := Current\_time$
27: Forward the update packet to source node
28: **end if**

29: **At source node**
30: Update packet is received
31: Update value of PRR in the routing table
32: **if** $terminatePath == TRUE$ **then**
33: Begin the route terminate process
34: **end if**

If the routing path can provides a value of PRR higher than the threshold value, it means that the current routing path can be used in the next periodic update period. The sink node will create the route update packet with the value of ‘terminatePath’ field sets to ‘FALSE’ and transmit this route control packet back to the source node. When an intermediate node receives
4.1. COMBINATION OF PERIODIC AND LOCAL UPDATE PROCESSES

The role of the source node in our periodic update process is simple. It creates an action based on the information in the route control packet. If the value of ‘terminatePath’ is ‘FALSE’, the source node updates the new value of PRR and extends the expired time for the routing path. If the value of ‘terminatePath’ is ‘TRUE’, the source node begins the route termination process.

In addition, the proposed periodic update process also helps reduce the total amount of routing overhead. The proposed periodic update process creates a route update packet per each routing path in a single periodic update period. Conversely, the simple periodic update process, which most previous proactive routing protocols use, creates a route update packet per each node in the single update period. As a result, the proposed periodic update process will create a smaller amount of routing overhead than the simple periodic update process. This is mainly because the total number of routing paths in the network is much less than the total number of nodes in the network as shown in Figures 4.4 and 4.5.

![Simple periodic update process](image)

**Figure 4.4:** Simple periodic update process
4.1.2 Local update process based on link quality.

The main objective of the local update process is to detect the sudden changes in network condition and notify this potential problem to the sink node. The periodic update process is unlikely to detect this type of problem because it has to wait for a long period of time before it can evaluate the routing path condition.

Each node in the routing paths is required to implement the local update process. It determines the quality of its communication links every $T_e$ period. In order to evaluate the link quality, all nodes in the routing paths must transmit the probe packets to its neighbours every specific period. Each node counts the number of received probe packets per each link for determining the value of the link quality ($Q_l$) when $T_e$ is expired. The value of $Q_l$ will be compared with the threshold value ($Q_t$). If $Q_l$ is lower than $Q_t$, an alert packet will be transmitted to the sink node to notify the link’s problem.

The sink node uses the information in the alert packet to estimate the new value of $T_p$. When the sink node receives the alert packet, it uses the value of $Q_l$ from the alert packet to estimate a period in which the number of packet drops can reach the threshold value ($T_{ed}$). If $T_{ed} < T_p$, the sink node changes the state of the routing path from normal to alert and begins to evaluate the value of $PRR$ when $T_{ed}$ is expired. Otherwise, the sink node waits until the current periodic
Algorithm 2 Local update with link quality evaluation

1: At intermediate node every $T_e$ seconds
2: Evaluate the value of $Q_l$ for each link
3: if $Q_l < Q_t$ then
4: Send alert packet to sink node
5: if alertReport == FALSE then
6: alertReport := TRUE
7: end if
8: if receive a periodic update packet then
9: alertReport := False
10: end if
11: end if

update period expires to evaluate the value of $PRR$.

If the value of $PRR$ from the current periodic update period is lower than the threshold value ($PRR_t$), the state of the routing path is changed to alert state. Then, a new value of $T_p$ is calculated based on the value of $T_{ed}$. At the end of the next periodic update, the routing path will be terminated if the value of $PRR$ is still lower than the threshold value. Otherwise, the state of the routing path is changed from alert to normal as shown in Algorithms 2 and 3 and Figure 4.6.

![Figure 4.6: Normal period and alert period](image)

The local update process is likely to create a small amount of routing overhead. This is because the alert packet will be created only if the value of the routing metric is higher than the threshold value.
Algorithm 3 The combination of periodic and local update

1: At sink node
2: After $T_p$ is expired
3: Evaluate the value of PRR of routing path
4: if $PRR_c < PRR_t$ then
5: \hspace{1em} if $\text{alertState} == \text{FALSE}$ then
6: \hspace{2em} $\text{alertState} := \text{TRUE}$
7: \hspace{2em} $T_p := T_{ed}$
8: \hspace{2em} $vCount := vCount + 1$
9: \hspace{1em} Send update packet to source node to start using the backup path
10: \hspace{1em} end if
11: \hspace{1em} if $vCount == 2$ then
12: \hspace{2em} $\text{terminatePath} := \text{TRUE}$
13: \hspace{2em} Send update packet to source node to terminate the routing path
14: \hspace{1em} end if
15: \hspace{1em} else
16: \hspace{2em} if $\text{alertState} == \text{TRUE}$ then
17: \hspace{3em} $\text{alertState} := \text{FALSE}$
18: \hspace{3em} Send update packet to source node to stop using the backup path
19: \hspace{2em} end if
20: \hspace{1em} end if
21: When receive an alert packet
22: Calculate $T_{ed}$ from $Q_l$
23: $\text{dropestimate} := T_p(begin) + T_{ed}$
24: $T_p(end) := T_p(begin) + T_p$
25: if $\text{dropestimate} <= T_p(end)$ then
26: \hspace{1em} $T_p(end) := \text{dropestimate}$
27: \hspace{1em} if $\text{alertReport} == \text{FALSE}$ then
28: \hspace{2em} $\text{alertReport} := \text{TRUE}$
29: \hspace{1em} end if
30: \hspace{1em} end if
31: At source node
32: if $\text{terminatePath} == \text{TRUE}$ then
33: \hspace{1em} Begin the route terminate process
34: \hspace{1em} end if

4.2 Determine Values of Control Parameters

There are three main control parameters in the combination of periodic and local update processes that require careful consideration:

- The number of transmitted probe packets in a $T_c$ period ($p_{b_s}$)
- The number of $T_c$ periods in a $T_p$ period ($N$)
4.2. DETERMINE VALUES OF CONTROL PARAMETERS

- The expected time that the number of packets dropped is higher than the threshold value ($T_{ed}$)

A high number of probe packets in a single $T_e$ period can provide a more accurate estimation value of link quality based on the number of probe packets. However, a greater number of probe packets also results in a higher amount of routing overhead. As a result, a proper value of the number of probe packets in a $T_e$ period is required in order to provide both an accurate estimation result and a small amount of routing overhead.

As the main objective of the local update process is to notify of potential problems based on the link quality, it is important to ensure that the local update process can detect poor link quality in the routing path before $T_p$ is expired. This is mainly because the number of packets dropped can exceed the threshold value before $T_p$ is expired when the link quality is low. If the local update process can quickly detect poor link quality, it can transmit an alert packet to inform the sink node.

After the sink node receives the alert packet, it uses the information about poor link quality to determine the value of $T_{ed}$. The value of $T_{ed}$ will be used as the new value of $T_p$ for the current periodic update period. The new value of $T_p$ helps the sink node to quickly react to the poor link quality problem by completing the periodic update evaluation before the number of packets dropped is higher than the threshold value.

For the local update process, each node periodically transmits a probe packet every $TP$ second. The probe packets are transmitted to all adjacent nodes that are the members of the same routing path. After $T_e$ is expired, each node evaluates the value of the link quality ($Q_l$) with Equation (4.1):

$$Q_l = \frac{pb_r}{pb_s}$$

(4.1)

where $pb_r$ is the total numbers of received probe packets in a single evaluation period.

The maximum number of probe packets $M_p$ that can be lost in a single evaluation period is calculated as:

$$M_p = (1 - Q_l) \times pb_s$$

(4.2)

where $Q_l$ is the threshold value of the link quality.
A communication link is considered to be in a poor condition when the number of lost probe packets is higher than $M_p$.

Multiple local evaluation periods per single periodic update period are required in the combination of periodic update and local update. For a very poor link condition, a single local update period may successfully detect a poor link condition. It is noted, however, that the probability of detecting the poor link condition in one local update period will notably decrease when the link condition remains poor for a short period of time, or when the condition of the link is improved. In this case, a longer period for determining the link condition is required to improve the successful detection rate for poor link quality.

In order to determine the appropriate number of local update periods in a single periodic update period, an analysis of probability is conducted to detect poor link conditions in each local update period. Let $X = \{1, 2, \ldots, N\}$ represents the identification numbers of the local update periods in which the event of poor link quality can be detected.

The probability that the event of poor link quality is detected in the current local update period, $P_{ec}$, is calculated as follows:

$$P_{ec} = 1 - P_{lp \geq M_p}$$ (4.3)

$$P_{lp \geq M_p} = \sum_{i=0}^{M_p} \binom{pb_s - i}{i} (1 - Q_l)^i (Q_l)^{pb_s - i}$$ (4.4)

With Equation (4.3) and (4.4), the probabilities that the event of poor link condition can be detected in each local evaluation periods are:

$$P_{e1} = P_{ec}$$
$$P_{e2} = (1 - P_{e1}) \times P_{ec}$$
$$\vdots$$
$$P_{eN} = \left[ \prod_{i=1}^{N-1} (1 - \sum_{j=1}^{i} P_{ej}) \right] \times P_{ec}$$ (4.5)

The proper value of $pb_s$ requires careful consideration. A high number of $pb_s$ packets in a single local update period provides a better link evaluation result because it provides more samples.
4.2. **DETERMINE VALUES OF CONTROL PARAMETERS**

To calculate the value of $Q_l$. However, more probe packets within a single update period also create a higher amount of routing overhead.

Similarly, the appropriate value of $N$ is an important factor that can impact the performance of the combination between periodic update and local update. More local update periods per one periodic update period can ensure that poor link conditions can be successfully detected before the periodic update timer expires. However, this also increases the response time of each node in reacting to the changes in the link condition.

Careful consideration is needed for determining the values of both $N$ and $pb_s$. With Equations (4.4) and (4.5), the probability of detecting the poor link quality of each link evaluation period with different values of $Q_l$ and $pb_s$ can be estimated.

In order to determine the value of $N$ and $pb_s$, we vary the value of $Q_l$ between 0.1 and 0.8 and $Q_l$ is set to 0.8. The value of $pb_s$ is set in the range between 10 and 25 probe packets in a local update period. The results of this study are shown in Figures 4.7 to 4.10.

![Figure 4.7: Probability of detecting poor link quality ($pb_s = 10$)](image)

Figures 4.7 to 4.10 show that the higher the value of $pb_s$ in a single local update period the higher the probability of detect the poor link quality within the first few evaluation periods.

From Figures 4.7 to 4.10, the appropriate number of local update periods per periodic update period that can successfully detect poor link conditions is 6, because the probability of detecting
the poor link condition is close to zero after the 6th local update period.

For the value of \( pb_s \), we use \( pb_s = 10 \) for good performance after comparing with \( pb_s = 15 \), \( pb_s = 20 \) and \( pb_s = 25 \). With low link quality (between 0.1 and 0.5), each node is likely to successfully detect the poor link event within the third local update period. When 25 probe packets per single local update period are used, the poor link quality event can be detected more
quickly in the second local update period. However, such a slight improvement is only achieved with the cost of more routing overhead, which results in degradation of the overall performance of the network.

The main objective of $T_{ed}$ is to ensure that the sink node will be able to identify a routing path with poor condition before the number of packets dropped exceeds the threshold value. Firstly, the maximum number of packet drops ($P_{dm}$) will be determined. $P_{dm}$ can be evaluated based on the data transmission rate of the source node ($R$) and the current value of $T_p$.

$$P_{dm} = (1 - PRR_t) \times R \times T_p$$  \hspace{1cm} (4.6)

Finally, the value of $T_{ed}$ is estimated as:

$$T_{ed} = \frac{P_{dm}}{R \times (1 - Q_l)}$$  \hspace{1cm} (4.7)

### 4.3 Verification Through Simulations

In this section, we evaluate the performance of the proposed route update and maintenance processes in terms of packet reception ratio, end-to-end delay and total amount of routing overhead. Three simulation scenarios with different network sizes (50, 100 and 200 nodes).
are created to illustrate the benefits of our proposed work in large-scale networks.

### 4.3.1 Experimental design.

In each simulation scenario, we will focus on a pair of source nodes and sink nodes. There are two possible routing paths between these two nodes: routing path 'A' and routing path 'B', as shown in Figure 4.11. A two-state error model with error rate $P_{err}$ is implemented in a member of routing path 'B'. This error model will cause the condition of routing path 'B' to remain in good condition ($P_{err} = 0$) for 300 seconds and then change to poor condition with an error rate equal to $P_{err}$ for the next 1200 seconds. The value of $P_{err}$ varies from 0.2 to 0.9 for each simulation scenario. This process repeats in the same manner for the whole network operation period as shown in Figure 4.12. The simulation timespan has been set to 6000 seconds for all scenarios.

![Network Architecture](image1)

**Figure 4.11:** A network architecture of the simulations in NS-2

![Two-State Error Model](image2)

**Figure 4.12:** Two-state error model implementation
In each scenario, the sensor nodes are deployed in a square area of 1000x1000 m. The application layer’s requirement for each routing path is the value of packet reception ratio (PRR) at the sink node equal to 0.8 or better. At the beginning of the simulation, the source node uses the routing path 'B' to transmit data packets to the sink node.

The simulation environment is based on the following configuration parameters. The data link layer is IEEE 802.11. The network communication model is Two-Ray Ground. There is a CBR traffic source to generate data packets of the size of 100 bytes every 15 seconds. The source node begins to transmit the data packets after the simulation start for 100 seconds. The transmission range of each sensor node is 40 m.

The results from the proposed route update and maintenance processes are compared with a proactive routing protocol DSDV and 2 popular reactive routing protocols (AODV and AOMDV) as benchmarks. The selection of routing protocols has been justified in Chapter 3.

All simulation scenarios are carried out using the NS-2.34 simulator.

4.3.2 Evaluation criteria.

Three performance metrics are used to quantify the performance of the proposed route maintenance processes.:.

(1) PRR, which is the ratio of the number of successfully received packets at the sink node to the total number of transmitted packets at the source node.

(2) Routing overhead is used. Only the routing overhead packets that belong to the specified route maintenance process are considered as the routing overhead. For DSDV, both periodic update packets and trigger update packets in the route maintenance process contribute to the routing overhead. For AODV and AOMDV, all control packets in the route discovery process and the hello packets from each node are considered as routing overhead. For the proposed route update and maintenance processes, both probe packets that are exchanged between the sensor nodes in the routing paths and the periodic update from the sink node are considered as routing overhead.

(3) End-to-End delay of the data transmission between source node and sink node will be used.
All values of these three performance metrics are the average values of the results from five independent simulation runs with the same configuration parameters. All the average values of three performance metrics in this simulation studies can pass the t-test analysis with 95% confidence intervals. Table 4.1 presents the average values of PRR from 50 nodes topology and their 95% confidence intervals.

<table>
<thead>
<tr>
<th>Error rate</th>
<th>Mean</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.9975</td>
<td>[0.9944 1.0006]</td>
</tr>
<tr>
<td>0.3</td>
<td>0.9905</td>
<td>[0.9838 0.9972]</td>
</tr>
<tr>
<td>0.4</td>
<td>0.9705</td>
<td>[0.9645 0.9765]</td>
</tr>
<tr>
<td>0.5</td>
<td>0.9070</td>
<td>[0.8895 0.9245]</td>
</tr>
<tr>
<td>0.6</td>
<td>0.9675</td>
<td>[0.9590 0.9760]</td>
</tr>
<tr>
<td>0.7</td>
<td>0.9710</td>
<td>[0.9613 0.9807]</td>
</tr>
<tr>
<td>0.8</td>
<td>0.9705</td>
<td>[0.9654 0.9756]</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9775</td>
<td>[0.9775 0.9775]</td>
</tr>
</tbody>
</table>

4.3.3 End-to-end delay performance.

Table 4.2: End-to-end delay in milliseconds

<table>
<thead>
<tr>
<th>Protocols</th>
<th>50 nodes</th>
<th>100 nodes</th>
<th>200 nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>130.58</td>
<td>184.65</td>
<td>323.62</td>
</tr>
<tr>
<td>AOMDV</td>
<td>167.8</td>
<td>210.49</td>
<td>352.66</td>
</tr>
<tr>
<td>DSDV</td>
<td>349.72</td>
<td>1,797.25</td>
<td>4,534.3</td>
</tr>
<tr>
<td>Our work</td>
<td>23.66</td>
<td>34.55</td>
<td>64</td>
</tr>
</tbody>
</table>

Table 4.2 provides the overall perspective in terms of end-to-end delay from AODV, AOMDV, DSDV and the route update and maintenance processes proposed in this chapter. All values in Table 4.2 are the average values from these four protocols under different values of error rate ($P_{err}$).

The results from Table 4.2 clearly indicate that DSDV produces the worst end-to-end delay performance in all network sizes (50, 100 and 200 nodes). The differences between the end-to-end delays of DSDV and the other three protocols significantly increase when the size of the
network becomes larger. For a 200 node topology, the end-to-end delay of DSDV is more than
ten times larger than the delay values from the other three protocols. In addition, Figures 4.13
to 4.15 show the end-to-end delay performance of AODV, AOMDV and the proposed model.

The main reason that DSDV performs poorly in terms of end-to-end delay is that DSDV
is based on conventional proactive routing. The simple periodic update process requires a
considerable amount of time to deliver information about poor link conditions to the source
node. As the value of $P_{err}$ increases, the required period to deliver the new routing information
to the source is drastically increased. As a result, the source node is likely to use a poor routing
path for a considerable period until it can change to an other available path.

AODV and AOMDV are based on a reactive routing approach that only creates the routing
path when the source node has a data packet to transmit. The routing paths of AODV and
AOMDV are likely to be created based on the current routing information. Therefore, both
AODV and AOMDV will have lower probability of experiencing the same problem as DSDV
and can provide better end-to-end delay performance.

The proposed route update and maintenance processes are based on the proactive routing
approach as DSDV, but the proactive routing is significantly improved through a combination
of periodic update and local update at each node, as the new route update and maintenance
mechanisms. Table 4.2 shows that the proposed model can provide the best results when
compared with other three protocols.

4.3.4 Packet reception ratio (PRR) performance.

The results from Figure 4.16 to Figure 4.18 show that DSDV provides the lowest level of
performance in term of PRR when compared with AODV, AOMDV and the proposed model.
As the network size becomes larger, the PRR of DSDV becomes much lower than those of other
three protocols. The main reason is the simple periodic update process, which is implemented
in DSDV. The simple periodic update process requires a significant period of time to distribute
the new route information to the source node. During this route distribution period, the source
node is likely to use a routing path with poor conditions before it can change to a new path. As
a result, a high number of data packets can be dropped, especially when $P_{err}$ is high.

In comparison, the reactive routing protocols (AODV and AOMDV) only create the routing
path when there is a data packet to transmit and this routing path will be terminated when the data transmission is finished. The routing paths that AODV and AOMDV create are based
4.3. VERIFICATION THROUGH SIMULATIONS

Figure 4.15: Average end-to-end delay in 200 node topology

Figure 4.16: PRR of 50 node topology
on the current network condition. Therefore, both AODV and AOMDV can provide better performance in terms of PRR when compared with DSDV.
Although the proposed route update and maintenance processes are also based on a proactive routing approach, they can provide a performance that is comparable with AODV and AOMDV as shown in Figures 4.16 to 4.18.

### 4.3.5 Routing overhead performance.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>50 nodes</th>
<th>100 nodes</th>
<th>200 nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>1,725.8</td>
<td>3,331.126</td>
<td>6,691.857</td>
</tr>
<tr>
<td>AOMDV</td>
<td>837.82</td>
<td>1,413.189</td>
<td>2,862.883</td>
</tr>
<tr>
<td>DSDV</td>
<td>4,349.2</td>
<td>14,841.91</td>
<td>57,456.87</td>
</tr>
<tr>
<td>Our work</td>
<td>288.97</td>
<td>481.084</td>
<td>961.555</td>
</tr>
</tbody>
</table>

### Table 4.4: Improvement of our work in term of routing overhead

<table>
<thead>
<tr>
<th>Protocol</th>
<th>50 nodes</th>
<th>100 nodes</th>
<th>200 nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>83.25%</td>
<td>85.57%</td>
<td>85.63%</td>
</tr>
<tr>
<td>AOMDV</td>
<td>65.51%</td>
<td>65.95%</td>
<td>66.41%</td>
</tr>
<tr>
<td>DSDV</td>
<td>93.35%</td>
<td>96.76%</td>
<td>98.32%</td>
</tr>
</tbody>
</table>

The results from Table 4.3 show that DSDV creates the highest amount of routing overhead when compared with the other three protocols. This is mainly because each node that implements DSDV must periodically transmit a routing update packet that contains all of routing information in its routing table to all neighbours in every specific period. As the number of nodes in the network increases, the total amount of routing overhead of DSDV increases drastically.

Reactive routing protocols (AODV and AOMDV) are likely to create a small amount of routing overhead because the routing path will be created only when there is a data packet to be transmitted. There is no extra routing overhead for any route update and maintenance process. Only the control packets in the route discovery process and the hello packets are counted as the overall amount of routing overhead. The results from Table 4.3 show that AODV and AOMDV produce a total amount of routing overhead that is at least 60% lower than DSDV.

Table 4.4 shows that the proposed route update and maintenance processes perform the best in terms of routing overhead when compared with the other three protocols. This is because the
proposed route update and maintenance processes are based on the number of available routing paths. It is not based on each individual node as in DSDV. As the number of available routing paths is likely to be much lower than the total number of nodes in the network, the model proposed in this thesis can create a lower amount of routing overhead. Similarly, the exchange of hello packets of the proposed model are based on the number of available routing paths. In both AODV and AOMDV, all nodes need to use the information about the reception rate of hello packets from all of the neighbour nodes to select the best next-hop node in the route discovery process. As a result, both AODV and AOMDV are likely to generate more routing overhead than the proposed model.

### 4.4 Chapter Summary

To address unpredictable wireless channel conditions in harsh environments in the industrial area, this chapter has developed new route update and maintenance processes. The basic assumption of the design is that all nodes in the network are static. The design uses a combination of a periodic update process and a local update process. The periodic update process aims to provide the accurate condition of the routing paths, while the local update process is responsible for detecting sudden changes in the routing path condition, which the periodic update process is unable to detect due to its long evaluation period.

To evaluate the effectiveness of the proposed route update and maintenance processes, simulation studies have been conducted in three well-designed and typical scenarios. The proposed route update and maintenance processes are compared with three widely used protocols DSDV, AODV and AOMDV as benchmarks. The performance metrics used for the evaluation are packet reception ratio (PRR), average end-to-end delay and total amount of routing overhead.

Simulation studies show that the proposed route update and maintenance processes give improved performance in all three performance metrics. They outperform all three popular routing protocols DSDV, AODV and AOMDV in terms of the total amount of routing overhead and the end-to-end delay. Moreover, they provide comparable performance in terms of PRR when compared with AODV and AOMDV. Therefore, both reliability and scalability issues have been well addressed in the proposed route update and maintenance processes.

However, a trade-off has to be made in order to achieve better performance. The proposed
route update and maintenance processes increase the computation load in each sensor node, especially the sink node.

In the next chapter, the mathematical analysis for the proposed route update and maintenance process will be discussed in detail.
In this chapter, we present a theoretical framework for the mathematical analysis and evaluation of the performance of the route update and maintenance processes developed in Chapter 4. The framework seeks to verify whether or not the current main routing path can provide a value of packet reception ratio (PRR) higher than the threshold value for the whole network operation period. If the answer is positive, we then evaluate the performance of the route update and maintenance processes based only on the performance of the main routing path. If the answer is negative, the source node will switch to the backup path to transmit the data packets to the sink node. Therefore, the source node only uses the main routing path to transmit data packets to the sink node until it detects that the main path is in poor condition, then, the backup path will be activated for transmitting the data packet to the sink node. For this reason, we need to evaluate the performance of the route update and maintenance processes for two periods: (1) The first period is based on the performance of the main routing path; and (2) The second period starts from the time that the backup path is activated for delivering the data packets to the sink node. The total value of PRR for both periods can be evaluated to represent the performance of the route update and maintenance processes.

The main difference between the proposed theoretical modelling framework and previous works is that the proposed work considers the effects of the MAC layer protocols that can influence the outcome of the data transmission process between the source node and the sink node. With this approach, the proposed theoretical framework can provide more accurate mathematical results to evaluate the performance of the route update and maintenance processes.
5.1 Notations

Some notations in this chapter are listed in Table 5.1 below.

**Table 5.1**: Notations used in this chapter

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{ij}$</td>
<td>Number of transmitted data packets through link ‘i’ in $j^{th}$ attempt</td>
</tr>
<tr>
<td>$M_d$</td>
<td>Maximum number of packet drops in a $T_p$ period</td>
</tr>
<tr>
<td>$N_s$</td>
<td>Expected $T_p$ period if poor path condition occurs for 2 consecutive periods</td>
</tr>
<tr>
<td>$n_c$</td>
<td>Maximum number of retransmissions for RTS packets</td>
</tr>
<tr>
<td>$n_d$</td>
<td>Maximum number of retransmissions for data packets</td>
</tr>
<tr>
<td>$P_{db}$</td>
<td>Probability that poor routing path condition is detected in the $1^{st}$ evaluation period</td>
</tr>
<tr>
<td>$P_{dw}$</td>
<td>Probability that poor routing path condition is detected in the last evaluation period</td>
</tr>
<tr>
<td>$P_{ex}$</td>
<td>Probability that poor routing path condition is detected in the $x^{th}$ evaluation period</td>
</tr>
<tr>
<td>$P_{nc}$</td>
<td>Probability that the main routing path remains in good condition for the whole $T_t$ period</td>
</tr>
<tr>
<td>$P_{vt}$</td>
<td>Probability that poor routing path condition is detected</td>
</tr>
<tr>
<td>$PRR_p$</td>
<td>PRR of the routing path</td>
</tr>
<tr>
<td>$PRR_t$</td>
<td>Threshold value of PRR</td>
</tr>
<tr>
<td>$k_s$</td>
<td>Number of received data packets at sink node</td>
</tr>
<tr>
<td>$Q_p$</td>
<td>Quality of routing path</td>
</tr>
<tr>
<td>$R$</td>
<td>Data transmission rate</td>
</tr>
<tr>
<td>$r_{ij}$</td>
<td>Number of packets that successfully complete RTS/CTS process of link ‘i’ in $j^{th}$ attempts</td>
</tr>
<tr>
<td>$S_i$</td>
<td>Success rate of the RTS/CTS process of link ‘i’</td>
</tr>
<tr>
<td>$T_b$</td>
<td>Estimated time that the backup routing path is used</td>
</tr>
<tr>
<td>$T_d$</td>
<td>Estimated time that the source node detects the event of poor routing path condition</td>
</tr>
<tr>
<td>$T_m$</td>
<td>Estimated time that only the main routing path is used</td>
</tr>
<tr>
<td>$T_p$</td>
<td>Periodic update period</td>
</tr>
<tr>
<td>$T_t$</td>
<td>Total network operation time</td>
</tr>
<tr>
<td>$u_{ij}$</td>
<td>Number of unsuccessfully received data packets of link ‘i’ in $j^{th}$ attempts</td>
</tr>
<tr>
<td>$S_{ij}$</td>
<td>Number of successfully received data packets of link ‘i’ in $j^{th}$ attempts</td>
</tr>
</tbody>
</table>

5.2 Conceptual Design of the Modelling Framework

The proposed mathematical modelling framework is based on an assumption that the route update and maintenance processes designed in Chapter 4 can provide similar levels of performance for different sizes of network. The verification of this assumption will be provided in Section 5.5.
In this study, we assume that a source node has two routing paths for transmitting data packets to a sink node. One routing path is assigned as the main routing path. Another routing path is used as the backup path. The backup path will be used only when the source node detects that the main routing path cannot provide a level of performance in terms of PRR that is equal to or higher than the threshold value.

The proposed modeling framework can be divided into two parts:

- Estimate the performance of the main routing path in terms of PRR
- Estimate the performance in terms of PRR in the case that the backup routing path is active and able to replace the main routing path.

5.2.1 Estimate the performance of the main routing path.

The value of PRR is calculated based on the total number, \( k_s \), of the data packets received by the sink node before their respective deadlines. Therefore, the main objective of the proposed modeling framework is to accurately estimate the value of \( k_s \) from each routing path in the network.

Most existing studies on system modelling estimate the value of \( k_s \) based only on the information from the network layer. From network layer’s perspective, each node transmits only one packet to the next-hop node per single data transmission process. As a result, the value of \( k_s \) is calculated from three parameters; quality of routing path \( (Q_p) \), data transmission rate \( (R) \), and total network operation time \( (T_t) \).

\[
k_s = Q_p \times R \times T_t \tag{5.1}
\]

where \( Q_p \) is calculated from the product of the quality of each link \( (Q_i) \) in the same routing path, as shown in Equation (5.2):

\[
Q_p = \prod_{i=1}^{\ell} (Q_i) \tag{5.2}
\]

where \( \ell \) is the index number of each communication link in the routing path.

However, the value of \( k_s \) from Equation (5.1) may not be accurate because the source node is likely to transmit more than one packet per single data transmission attempt. This is because
of the requirements of the data link layer. For example, the source node is required to transmit an RTS request packet to the next-hop node and wait for a CTS packet back from the next-hop node before it can begin to transmit a data packet to the next-hop node. If the RTS/CTS process fails, the data packet will be dropped and the data transmission is considered a failure.

The proposed modelling framework considers two main processes in the data link layer of WSNs: the RTS/CTS process and the retransmission process. Both these two data link processes operate on a hop-to-hop basis. As a result, the proposed modelling framework must estimate the performance of each link in the routing path in order to calculate the value of \( k_s \).

To ascertain the value of \( k_s \) in each routing path, the following steps must be completed:

1. Estimate the number of successfully received data packets of the first link in the routing path based on the value of transmitted data packets at the source node.

2. Use the number of successfully received data packets from the previous link as the number of transmitted data packets for the current link.

3. Estimate the number of successfully received data packets of the current link based on the value of transmitted data packets from Step 2.

4. Repeat Step 2 and Step 3 until the value of successfully received data packets of the last link in the routing path is obtained.

5.2.2 Estimate the performance of the backup routing path.

In this case, the data transmission process uses two routing paths for transmitting the data packets to the sink node. The source node uses the main routing path until it detects that the main routing path cannot provide a value of PRR higher than the threshold value. Then, the backup routing path is activated to transmit the data packet to the sink node.

In order to estimate the performance of both routing paths, we need to estimate the time that the source node uses the main routing path \( (T_m) \) and the time that the source node uses the backup routing path \( (T_b) \).

The value of \( T_m \) can be estimated based on the event that the main routing path will be terminated. In Chapter 4, we define that the routing path will be terminated when the values
of PRR from two consecutive periodic update period are lower than the threshold value. The proposed modelling framework develops a probability model to estimate the time in which two consecutive periodic update periods produce a value of PRR lower than the threshold value.

After the value of $T_m$ is derived, the value of $T_b$ can be obtained as shown in Equation 5.3.

$$T_b = T_t - T_m$$

where $T_t$ is the total network operation time.

### 5.3 Mathematical Modelling of Data Transmissions in a Routing Path

In practical implementation of WSNs, there are multiple packets that the sender node must transmit and successfully receive and acknowledge from the next-hop node before an actual data packet can be transmitted to the next-hop node. The additional packets are based on the data link layer protocol that is implemented in each sensor node. In this study, the widely adopted IEEE802.11 with a request to send / clear to send (RTS/CTS) mechanism is used in all sensor nodes.

The diagram in Figure 5.1 a shows the data transmission process between two nodes in the routing path. Node A must transmit an ARP request packet to Node B and waits for an ARP reply packet from Node B before it can begin the RTS/CTS process. The ARP mechanism only occurs once, when Node A starts to transmit the first data packet. After both pair of nodes learn about the MAC addresses of each other from the ARP mechanism, they can begin the next transmission without exchanging ARP packets again. In the case that there is a significant amount of data packets, the effect of the ARP mechanism is very small. Therefore, the ARP mechanism can be excluded from the proposed mathematical framework.

In this study, the proposed modelling framework is based on the data transmission diagram in Figure 5.1 b. A data packet will be considered as successfully received at the next-hop node when the following events are successful.

- Node A must successfully transmit a RTS packet to the next-hop node and receive a CTS packet back from Node B. If Node A transmits the RTS packet to Node B but it does not receive any CTS packet back within a specific period, it will attempt to retransmit the
RTS packet again. The maximum number of the RTS packet retransmissions is equal to $n_c$. If the sender node reaches the maximum value of $n_c$ but still does not receive the CTS packet back, it will drop the current data packet and reports this error to the upper layers.

- If the RTS/CTS process is successful, Node A will attempt to transmit a data packet. The data packet will be considered as successfully received by Node A when it receives an ACK packet back from Node B. If Node A does not receive any ACK packet, it will attempt to transmit the data packet again, but it must start the RTS/CTS process again before it can transmit the data packet. The maximum number of the data packet retransmission at each node is equal to $n_d$. If Node A tries to retransmit the data packet for $n_d$ times and still does not receive any ACK packet back, it will drop the current data packet, as in the case of the RTS packet transmission.
In order to estimate the value of $k_s$, the RTS/CTS process and the data packet transmission process are considered to be two separate events. The outcome of the data transmission process is independent from the result of the RTS/CTS process.

To estimate the performance of the routing path in terms of PRR, the main objective of the proposed mathematical modelling framework is to derive the total number, $k_s$, of data packets that successfully arrived at the sink node before their deadlines. For the routing path with $N$ nodes as shown in Figure 5.2, $k_s = k_{N-1}$. Thus, the objective becomes to estimate the value of $k_{N-1}$.

![Figure 5.2: A routing path for mathematical analysis](image)

In order to quantify the effect of the data link layer in the data transmission process, a mathematical modelling framework must estimate the success rate of the data transmission process in each of the communication links along the routing path. All data packets that can be successfully transmitted through the previous communication link will be transmitted into the next communication link. For example, for link ‘$i$’ in the routing path shown in Figure 5.2, the total number, $k_i$, of data packets that can be successfully transmitted through the link is estimated based on $k_{i-1}$, the total number of data packets that can be successfully transmitted through link ‘$i-1$’.

As the proposed mathematical modelling framework considers the retransmission process, the value of $k_i$ must be estimated based on the number of successfully transmitted data packets from all possible data transmission attempts $s_1, s_2, \ldots, s_{nd}$, where $nd$ is the maximum number of allowed retransmission attempts as mentioned previously.

For the routing path shown in Figure 5.2, the modelling will be established in three steps: 1) For link ‘$i$’, modelling the $j^{th}$ transmission attempt $s_j$, where $j$ ranges from 1 to $nd$; 2) From these results, estimate $k_i$ for link ‘$i$’; and 3) Modelling the whole routing path.
5.3.1 Modelling the RTS/CTS Process in Link ‘i’.

In each data transmission attempt, the sender node must complete two processes: the RTS/CTS process and the data transmission process. The data transmission mechanism will be activated only when the RTS/CTS process is successful. As a result, the success rate of the RTS/CTS process for link ‘i’, denoted by $S_i$, needs to be estimated first.

As shown in Figure 5.3, a Markov chain with an absorbing state is used to estimate the value of $S_i$. It represents the transmission of an RTS packet over a communication link. There are only two states in this Markov chain: 0 for RTS retransmission and 1 for successful RTS transmission. At state 0, the state can either move to state 1 with the transition probability $Q_i$ or stay at the current state 0 with the transition probability 1 - $Q_i$. State 1 is an absorbing state, and there is only 1 outcome at this state. The transition probability of this state is 1. From the

![Figure 5.3: A Markov chain of the data transmission process for link ‘i’](image)

Markov chain in Figure 5.3, the transition matrix $M^{(i,1)}$ of the first attempt of transmission in link ‘i’ can be derived as:

$$M^{(i,1)} = \begin{pmatrix} 1 - Q_i & Q_i \\ 0 & 1 \end{pmatrix} \triangleq \begin{pmatrix} m_{00}^{(i,1)} & m_{01}^{(i,1)} \\ 0 & 1 \end{pmatrix}. \quad (5.4)$$

For $j$ attempts of retransmission, the overall transition matrix $M^{(i,j)}$ is expressed as:

$$M^{(i,j)} = M^{(i,1)} \times \cdots \times M^{(i,1)} \triangleq \begin{pmatrix} m_{00}^{(i,j)} & m_{01}^{(i,j)} \\ 0 & 1 \end{pmatrix},$$

$$m_{00}^{(i,j)} = (1 - Q_i)^j, \quad m_{01}^{(i,j)} = \sum_{\kappa=1}^{j} (1 - Q_i)^{\kappa-1} Q_i. \quad (5.5)$$
Then, $S_i$ can be calculated from as:

$$S_i = m^{(i,n_c)}_0, \quad i = 1, 2, \cdots, N - 1,$$

(5.6)

where $n_c$ is the maximum number of allowed RTS/CTS retransmissions. The data packets to be transmitted will be dropped if the RTS/CTS process still fails after $n_c$ attempts.

### 5.3.2 Modelling the $j^{th}$ Attempt of Data Transmissions.

With $S_i$, the number $r_{ij}$ of data packets with successful preceding RTS/CTS in the $j^{th}$ attempt in link ‘$i$’ is:

$$r_{ij} = S_i I_{ij},$$

(5.7)

where $I_{ij}$ is the total number of data packets that node $i$ transmits to node $i + 1$ through link ‘$i$’ in the $j^{th}$ attempt. The value of $I_{ij}$ is equal to the total number of the data packets that cannot successfully transmitted through link ‘$i$’ in the previous $(j - 1)^{th}$ data transmission attempt.

Let $S_{ij}$ denote the number of data packets successfully transmitted through link ‘$i$’ in the $j^{th}$ attempt. It follows that:

$$S_{ij} = r_{ij} Q_i.$$

(5.8)

Substituting Equation (5.7) into Equations (5.8) yields:

$$S_{ij} = S_i I_{ij} Q_i.$$

(5.9)

After $S_{ij}$ is derived for the $j^{th}$ attempt in link ‘$i$’, the number $u_{ij}$ of data packets that cannot be successfully transmitted to node $i + 1$ in the $j^{th}$ attempt can be derived as:

$$u_{ij} = r_{ij} - S_{ij}.$$

(5.10)

These unsuccessfully transmitted packets in the $j^{th}$ attempt will be retransmitted. Node $i$ will terminate the retransmission process either if node $i$ receives an ACK packet back from node $i + 1$ or when the total number of retransmission attempts becomes higher than the pre-defined threshold $n_d$. 
5.3.3 Modelling the Transmission in Link ‘i’.

After all possible data transmission attempts in link ‘i’ are modelled, the number \( k_i \) of data packets that can be successfully transmitted through link ‘i’ can be calculated as:

\[
k_i = \sum_{j=1}^{n_d} s_{ij}, \quad i = 1, 2, \ldots, N - 1.
\]  

(5.11)

This number of data packets will be further transmitted over link ‘\( i + 1 \)’, i.e.,

\[
I_{i+1,1} = k_i.
\]  

(5.12)

More specifically, the total number of packets to be transmitted from node ‘1’ over the routing path is \( I_{11} \). The value of \( I_{11} \) can be calculated as follow:

\[
I_{11} = T_t \times R
\]  

(5.13)

5.3.4 Modelling the Whole Routing Path.

With \( k_{N-1} \) from Equation (5.11) for the total number of data packets received at the sink node), as well as \( I_{11} \) for the total number of data packets sent from the source node), the success rate of the data transmission process as measured by PRR is derived as:

\[
PRR_p = \frac{k_{N-1}}{I_{11}}.
\]  

(5.14)

The value of \( PRR_p \) can be used to verify the performance of the main routing path. The value of \( PRR_p \) is compared with the threshold value (\( PRR_t \)). If \( PRR_p \) is equal or higher than \( PRR_t \), it means that the main routing path satisfies the level of performance that the application’s layer requires. Therefore, the source node can use the main routing path to deliver all of its data packets to the sink node. The value of \( PRR_p \) can be used to represent the value of PRR for the whole network operation time.

Conversely, when \( PRR_p \) is lower than \( PRR_t \), it indicates that the main routing path cannot provide an acceptable level of performance. A backup routing path must be activated to transmit the data packets to the sink node. In this case, we need to estimate the performance in terms
of PRR from both the main routing path and the backup routing path, which will be discussed next.

5.4 Mathematical Modelling of Data Transmissions in Both Main Path and Backup Path

A backup routing path will be activated to replace the main routing path when the main routing path provides a value of $PRR_p$ lower than the threshold value for two consecutive periodic update periods.

In order to estimate the value of PRR in the case that both the main routing path and the backup routing path is active, the time used by the source node for each routing path to transmit the data packets to the sink node must be estimated.

5.4.1 Estimate the operation time of main routing path.

The main routing path will be terminated when the source node detects that the value of $PRR_p$ from the main routing path is lower than the threshold value for two consecutive periodic update periods. As a result, the operation time of the main routing path will stop when the two consecutive periodic update periods of $PRR_p$ are lower than the threshold value.

Figure 5.4 shows the total network operation time ($T_t$) is divided into multiple periodic evaluation periods ($T_p$).

![Figure 5.4: Time diagram of $T_t$ and $T_p$](image)

The main objective of the mathematical model in this section is to estimate when the event, which two consecutive $T_p$ periods provide a value of $PRR_p$ lower than $PRR_t$, occurs.

The value of $PRR_p$ will be lower than the value of $PRR_t$ when the number of packet drops at the sink node is higher than the maximum threshold value ($M_d$).
The value of $M_d$ can be estimated from the value of $Q_i$ as shown in Equation (5.15).

$$M_d = (1 - Q_i) \times (R \times T_p) \tag{5.15}$$

After the value of $M_d$ is estimated, we can calculate the probability that a $T_p$ period provides the value of $PRR_p$ lower than $PRR_t$ ($P_{vt}$), as shown below:

$$P_{pt} = \sum_{\tau=0}^{M_d} \left( R \times T_p \right)^\tau (1 - Q_i)^\tau (Q_i)^(R\times T_p)^{-\tau} \tag{5.16}$$

$$P_{vt} = 1 - P_{pt} \tag{5.17}$$

where $P_{pt}$ is the probability that the current $T_p$ period provides a value of $PRR_p$ higher than $PRR_t$.

A case study is created to provide a simple example for evaluating the value of $P_{vt}$ at each periodic evaluation period. The important parameters for this case study are listed as follow:

- There are 5 $T_p$ periods in a $T_t$ period;
- Total number of transmitted data packets at the source node is 10;
- $PRR_t = 0.8$ and $M_d = 2$.

Figure 5.5 shows the best case scenario in which the source node can detect the event of a poor routing path condition ($PRR_p$ lower than $PRR_t$ for two consecutive periods). The poor routing path condition will be detected in the first periodic evaluation period.

Let ‘0’ represent the event that the value of $PRR_p$ is equal to or higher than the threshold value and ‘1’ represent the event that the value of $PRR_p$ is lower than the threshold value.

The probability that the event of $PRR_p$ is lower than the threshold value will occur at the first evaluation period, or the best case scenario ($P_{db}$ can be calculated from the product of $P_{vt}$ from the first and second $T_p$ periods.

$$P_{db} = P_{e1} = P_{vt} \times P_{vt} \tag{5.18}$$

For the worst case scenarios, there are multiple possible outcomes as shown in Figure 5.6 but
5.4. MATHEMATICAL MODELLING BACKUP ROUTING PATH

all cases will detect the event that the value of $PRR_p$ is lower than the threshold value for two consecutive periods in the periodic evaluation period number 4.

Based on all of the possibilities for the worst case scenarios, we can calculate the probability that this worst case scenario will occur ($P_{dw}$) with Equation (5.19).

$$P_{dw} = P_{e4} = (P_{pt}^3 \times P_{vt}^2) + 2(p_{pt}^2 \times P_{vt}^3)$$  \hspace{1cm} (5.19)

After the best case scenario and the worst case scenarios are estimated, there are only two
possibilities that a poor routing path condition will be detected: (1) a poor routing path condition is detected at the periodic update period number 2 \(P_e^2\) and (2) a poor routing path condition is detected at the periodic update period number 3 \(P_e^3\).

Figures 5.7 and 5.8 show all of the possible outcomes of each scenario. The calculation of both \(P_e^2\) and \(P_e^3\) can be done in a similar approach to the calculation for the worst case scenario.

\[
P_e^2 = P_{pt} \times P_{vt}^2 \quad (5.20)
\]
\[
P_e^3 = (P_{pt}^2 \times P_{vt}^2) + (p_{pt} \times P_{vt}^3) \quad (5.21)
\]

All the probabilities that the poor routing path condition will occur in each periodic update period are estimated. Then, the expected periodic evaluation period that the event of \(P_{RR_p}\) is lower than \(P_{RR_t}\) for two consecutive periods will be likely to occur \(N_s\) can be evaluated as shown below:

\[
N_s = \frac{\sum_{i=1}^{4} (i \times P_{ei})}{\sum_{i=1}^{4} (P_{ei})} \quad (5.22)
\]

Based on the value of \(T_p\) and \(N_s\), the estimated time that the source node will detect the event of poor routing path condition \(T_d\) can be estimated by Equation (5.23).

\[
T_d = (N_s - 1) \times T_p \quad (5.23)
\]

Figure 5.9 shows the time diagram that represents the operation time of the main routing path \(T_m\). \(T_d\) represents the period from the beginning of network operation to the beginning of the periodic update period during which the event of a poor routing path condition is detected. In
the next periodic update period after $T_d$ is expired, the local update process in the main routing path will detect the poor link condition and will transmit an alert packet to the sink node as described in Chapter 4. After the sink node receives the alert packet, it will re-estimate the new value for the periodic update period ($T_{pa}$).

Algorithm 4 shows the process that the sink node uses for determining the new value of $T_p$. The value of $T_{pa}$ is estimated based on the maximum number of packet drops in a single periodic update period ($M_p$). To calculate $T_{pa}$, we rearrange Equation (5.7) to Equation (5.11) by replacing $k_i$ with $M_d$ and $T_t$ with $T_{pa}$.

For example, we set the value of $M_d$ to 12 and the value of $n_d$ to 4. The routing path of this example is shown in Figure 5.10. Only one link (link ‘i’) in the routing path has a link quality ($Q$) equal to 0.6 and the other links have perfect link quality ($Q = 1$). We can derive the value of $T_{pa}$ as shown below:

0.0115$T_i + 0.0006T_i + 0.001T_i + 0.0016T_i + 0.0028T_i = 12$

$T_i = T_{pa} = 685$

After a new value of $T_p$ is derived, we can estimate the period when only the main routing

---

Figure 5.9: A time diagram of the operation times of the main path and the backup path
path is used for data transmissions \((T_m)\), as shown below:

\[ T_m = T_p + T_d \]  \hspace{1cm} (5.24)

Based on the mathematical model in Section 5.4.1, we can evaluate the total number of data packets that can be successfully transmitted to the sink node via the main routing path \((k_m)\) by replacing the value of \(T_t\) with the value of \(T_m\).

### 5.4.2 Estimate the operation time of backup routing path.

After \(T_m\) is expired, the source node will begin to use the backup routing path in conjunction with the main routing path to deliver the data packets to the sink node as shown in Figure 5.9. In this study, we assume that the backup routing path can provide a 100% successful transmission rate \((Q_{ap} = 1)\). Therefore, the data packets that are transmitted after \(T_m\) is expired, will be successfully received at the sink node via the backup routing path.

The total number of received data packets at the sink node via the backup routing path \((k_a)\) can be calculated as shown below:

\[ k_a = Q_{ap} \times R \times T_b \]  \hspace{1cm} (5.25)

### 5.4.3 Estimate an overall PRR from both main path and backup path.

The total number of received data packets at the sink node for the whole network operation period \((k_t)\) is the summation of the value of \(k_m\) and the value of \(k_a\) as shown in Equation (5.26).

\[ k_t = k_m + k_a \]  \hspace{1cm} (5.26)
5.5 VERIFICATION OF THE MATHEMATICAL MODELLING

With the value of the total number of received data packets at the sink node and the total number of transmitted data packets at the source node, the total value of PRR for the whole network operation period \( PRR_o \) can be evaluated as follow:

\[
PRR_o = \frac{k_i}{T_{11}}
\]  

(5.27)

The value of \( PRR_o \) in Equation (5.27) represents the performance of the proposed route update and maintenance processes in the case that the source node can detect a poor routing path condition in the main routing path, which needs to be replaced with the backup routing path.

However, we also need to consider the case that the source node cannot detect the poor routing path condition in the main routing path, and the source node only uses the main routing path for the whole network operation period. The probability that the main routing path will be used for the whole network operation period \( P_{nc} \) can be simply estimated as shown in Equation (5.28).

\[
P_{nc} = 1 - \sum_{i=1}^{4} P_{ei}
\]  

(5.28)

The value of \( PRR_p \) in the case that only the main routing path is used can be calculated with Equation (5.14).

Finally, the expected value of PRR can be calculated based on both value of \( PRR_o \) and \( PRR_p \) as shown below:

\[
PRR_{avg} = PRR_o \times (1 - P_{nc}) + PRR_p \times P_{nc}
\]  

(5.29)

5.5 Verification of the Mathematical Modelling

For the simulation setup in this section, we will verify the effectiveness of the mathematical modelling. Results from the model computation will be compared with those from simulation studies. There are three network topologies for the simulation studies: 100, 150 and 200 nodes. Different sizes of the network topologies are used to verify the effectiveness of the mathematical modelling in terms of scalability.

All three topologies are based on a similar setup. A pair of source nodes and sink nodes has two possible routing paths; routing path ‘A’ and routing path ‘B’. At the beginning, the source
node uses the routing path ‘B’ as the main routing path to transmit the data packets to the sink node. Routing path ‘A’ is used as a backup routing path.

In order to create poor routing path conditions, one communication link in the routing path ‘B’ will transmit a packet with error rate equal to $P_e$ for the whole network operation period. The value of $P_e$ varies from 0.2 to 0.9.

The simulation environment is based on the following configuration parameters. The data link layer is IEEE 802.11. The network communication model is Two-Ray Ground. There is a CBR traffic source to generate data packets of the size of 100 Bytes every 10 seconds. The source node begins to transmit the data packets after the simulation starts. The transmission range of each sensor node is 40 m.

All simulations are carried out using the NS-2.34 simulator.

5.5.1 Verify the main assumption of the mathematical modelling.

The proposed mathematical modelling framework is based on the assumption that the error rate ($P_e$) in the routing path has a major impact on the value of PRR. This is mainly because the route update and maintenance processes described in Chapter 4 are designed to quickly detect and report the poor link condition event to the source node. The propagation delay of transmitting poor route condition events to the source node is likely to be very small compared with the evaluation time that the route update and maintenance processes use to verify the routing path condition. As a result, the route update and maintenance processes should provide a similar level of PRR for different sizes of network.

To support this claim, we use a paired-samples t-test analysis to verify that the route update and maintenance processes in Chapter 4 can provide a similar level of performance in terms of PRR for different sizes of network. Three comparisons between the results from 100 nodes and 150 nodes topology, 100 nodes and 200 nodes topology, and 150 nodes and 200 nodes topology, are shown in Tables 5.2 to 5.4.
### Table 5.2: Comparisons of PRR between 100 nodes topology and 150 nodes topology

<table>
<thead>
<tr>
<th>Error rate</th>
<th>Topology</th>
<th>Mean</th>
<th>95% Confidence Interval</th>
<th>Std. Deviation</th>
<th>Std. Error Mean</th>
<th>Sig.(2-tailed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>100 nodes</td>
<td>0.9993</td>
<td>[0.9982, 1.0005]</td>
<td>0.0009</td>
<td>0.0004</td>
<td>0.0690</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9973</td>
<td>[0.9950, 0.9997]</td>
<td>0.0018</td>
<td>0.0008</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>100 nodes</td>
<td>0.9906</td>
<td>[0.9837, 0.9976]</td>
<td>0.0056</td>
<td>0.0025</td>
<td>0.6570</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9893</td>
<td>[0.9853, 0.9934]</td>
<td>0.0032</td>
<td>0.0014</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>100 nodes</td>
<td>0.9686</td>
<td>[0.9605, 0.9769]</td>
<td>0.0066</td>
<td>0.0029</td>
<td>0.5730</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9660</td>
<td>[0.9566, 0.9754]</td>
<td>0.0076</td>
<td>0.0034</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>100 nodes</td>
<td>0.9123</td>
<td>[0.9041, 0.9205]</td>
<td>0.0066</td>
<td>0.0029</td>
<td>0.8010</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9099</td>
<td>[0.8867, 0.9333]</td>
<td>0.0187</td>
<td>0.0083</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>100 nodes</td>
<td>0.9410</td>
<td>[0.8809, 1.0012]</td>
<td>0.0484</td>
<td>0.0216</td>
<td>0.9460</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9433</td>
<td>[0.8727, 1.0140]</td>
<td>0.0568</td>
<td>0.0254</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>100 nodes</td>
<td>0.9829</td>
<td>[0.9762, 0.9985]</td>
<td>0.0054</td>
<td>0.0024</td>
<td>0.0640</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9769</td>
<td>[0.9733, 0.9807]</td>
<td>0.0029</td>
<td>0.0013</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>100 nodes</td>
<td>0.9800</td>
<td>[0.9764, 0.9836]</td>
<td>0.0028</td>
<td>0.0012</td>
<td>0.1410</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9773</td>
<td>[0.9746, 0.9801]</td>
<td>0.0023</td>
<td>0.0009</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>100 nodes</td>
<td>0.9816</td>
<td>[0.9816, 0.9816]</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9816</td>
<td>[0.9816, 0.9816]</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

### Table 5.3: Comparisons of PRR between 100 nodes topology and 200 nodes topology

<table>
<thead>
<tr>
<th>Error rate</th>
<th>Topology</th>
<th>Mean</th>
<th>95% Confidence Interval</th>
<th>Std. Deviation</th>
<th>Std. Error Mean</th>
<th>Sig.(2-tailed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>100 nodes</td>
<td>0.9993</td>
<td>[0.9982, 1.0005]</td>
<td>0.0093</td>
<td>0.0004</td>
<td>0.1910</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9980</td>
<td>[0.9958, 1.0003]</td>
<td>0.0018</td>
<td>0.0008</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>100 nodes</td>
<td>0.9906</td>
<td>[0.9837, 0.9976]</td>
<td>0.0056</td>
<td>0.0025</td>
<td>0.860</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9913</td>
<td>[0.9842, 0.9984]</td>
<td>0.0057</td>
<td>0.0025</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>100 nodes</td>
<td>0.9686</td>
<td>[0.9605, 0.9769]</td>
<td>0.0066</td>
<td>0.0029</td>
<td>0.1630</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9733</td>
<td>[0.9713, 0.9754]</td>
<td>0.0016</td>
<td>0.0007</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>100 nodes</td>
<td>0.9123</td>
<td>[0.9041, 0.9205]</td>
<td>0.0066</td>
<td>0.0029</td>
<td>0.5160</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9069</td>
<td>[0.8869, 0.9271]</td>
<td>0.0162</td>
<td>0.0072</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>100 nodes</td>
<td>0.9410</td>
<td>[0.8809, 1.0012]</td>
<td>0.0484</td>
<td>0.0216</td>
<td>0.7210</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9509</td>
<td>[0.9064, 0.9956]</td>
<td>0.0359</td>
<td>0.0161</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>100 nodes</td>
<td>0.9829</td>
<td>[0.9762, 0.9985]</td>
<td>0.0054</td>
<td>0.0024</td>
<td>0.2740</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9796</td>
<td>[0.9757, 0.9836]</td>
<td>0.0027</td>
<td>0.0012</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>100 nodes</td>
<td>0.9800</td>
<td>[0.9764, 0.9836]</td>
<td>0.0028</td>
<td>0.0012</td>
<td>0.1760</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9773</td>
<td>[0.9739, 0.9808]</td>
<td>0.0034</td>
<td>0.0015</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>100 nodes</td>
<td>0.9816</td>
<td>[0.9816, 0.9816]</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9816</td>
<td>[0.9816, 0.9816]</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Table 5.4: Comparisons of PRR between 150 nodes topology and 200 nodes topology

<table>
<thead>
<tr>
<th>Error rate</th>
<th>Topology</th>
<th>Mean</th>
<th>95% Confidence Interval</th>
<th>Std. Deviation</th>
<th>Std. Error Mean</th>
<th>Sig.(2-tailed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>150 nodes</td>
<td>0.9973</td>
<td>[0.9950, 0.9997]</td>
<td>0.0018</td>
<td>0.0008</td>
<td>0.5750</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9980</td>
<td>[0.9958, 1.0003]</td>
<td>0.0018</td>
<td>0.0008</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>150 nodes</td>
<td>0.9893</td>
<td>[0.9853, 0.9934]</td>
<td>0.0032</td>
<td>0.0014</td>
<td>0.5170</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9913</td>
<td>[0.9842, 0.9984]</td>
<td>0.0057</td>
<td>0.0025</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>150 nodes</td>
<td>0.9660</td>
<td>[0.9566, 0.9754]</td>
<td>0.0076</td>
<td>0.0034</td>
<td>0.0690</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9733</td>
<td>[0.9713, 0.9754]</td>
<td>0.0016</td>
<td>0.0007</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>150 nodes</td>
<td>0.9099</td>
<td>[0.8867, 0.9333]</td>
<td>0.0187</td>
<td>0.0083</td>
<td>0.7930</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9069</td>
<td>[0.8869, 0.9271]</td>
<td>0.0162</td>
<td>0.0072</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>150 nodes</td>
<td>0.9433</td>
<td>[0.8727, 1.0140]</td>
<td>0.0568</td>
<td>0.0254</td>
<td>0.8060</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9509</td>
<td>[0.9064, 0.9956]</td>
<td>0.0359</td>
<td>0.0161</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>150 nodes</td>
<td>0.9799</td>
<td>[0.9733, 0.9807]</td>
<td>0.0029</td>
<td>0.0013</td>
<td>0.2100</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9796</td>
<td>[0.9757, 0.9836]</td>
<td>0.0032</td>
<td>0.0014</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>150 nodes</td>
<td>0.9773</td>
<td>[0.9746, 0.9801]</td>
<td>0.0022</td>
<td>0.0009</td>
<td>0.9990</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9773</td>
<td>[0.9739, 0.9808]</td>
<td>0.0027</td>
<td>0.0012</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>150 nodes</td>
<td>0.9816</td>
<td>[0.9816, 0.9816]</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9816</td>
<td>[0.9816, 0.9816]</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

The results from Tables 5.2 to 5.4 show that the values of PRR from all three sizes of network topologies are similar, with 95% confidence level in any level of error rate. As a result, we can conclude that the route update and maintenance processes described in Chapter 4 can provide similar performances in term of PRR for all three network topologies.

5.5.2 Compare between simulation results and mathematical results.

Table 5.5 presents all values of the input variables for the proposed mathematical modelling to estimate the performance in term of PRR of the route update and maintenance processes.

Table 5.5: Configuration parameters

<table>
<thead>
<tr>
<th>Input variables</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_t$</td>
<td>6000 sec.</td>
</tr>
<tr>
<td>$T_p$</td>
<td>600 sec.</td>
</tr>
<tr>
<td>$P R R_t$</td>
<td>0.8</td>
</tr>
<tr>
<td>$R$</td>
<td>0.1 packet/sec.</td>
</tr>
<tr>
<td>$n_c$</td>
<td>4</td>
</tr>
<tr>
<td>$n_d$</td>
<td>4</td>
</tr>
</tbody>
</table>
The PRR values from the proposed mathematical modelling and the simulation results are presented in Table 5.6. The results from Table 5.6 show that most of the PRR values from the proposed mathematical modelling are within the 95% confidence interval of the simulation results. The values of PRR from the proposed mathematical modelling that do not fall in the 95% confidence interval are very few, and show only 1% difference from the confidence interval in the worst case scenario ($P_e = 0.4$ in 200 nodes topology). Overall, we can conclude that the results from the proposed mathematical modelling framework can accurately represent the performance in terms of PRR for the route update and maintenance processes presented in Chapter 4.

5.6 Chapter Summary

For industrial applications, the routing path must be accurately evaluated in order to maintain an acceptable performance level for data transmissions. In this chapter, a theoretical mathematical framework has been developed to analyze and evaluate the performance of the route update and maintenance processes presented in this thesis. Comparison of the outputs from the proposed mathematical modelling and the outputs from the simulation results shows that the proposed mathematical modelling provides similar outcomes to the simulation studies. Moreover, the results from the simulation studies show that the route update and maintenance processes proposed in this thesis can provide similar levels of performance in terms of PRR under different sizes of network. The main difference of the proposed mathematical modelling from previous studies is that our proposed work considers the effect of the data link layer in the data transmission process. Both the retransmission process and the RTS/CTS process, which are widely adopted in WSNs, are included in the mathematical modelling in order to provide more accurate results.

In the next chapter, the specific mechanisms to support multiple types of data traffic in the same network will be discussed.
Table 5.6: Comparisons of PRRs between simulation and mathematical modelling

<table>
<thead>
<tr>
<th>Error rate</th>
<th>Topology</th>
<th>Mean</th>
<th>95% Confidence Interval</th>
<th>Mathematical analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>100 nodes</td>
<td>0.9993</td>
<td>[0.9982, 1.0005]</td>
<td>0.9967</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9973</td>
<td>[0.9950, 0.9997]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9980</td>
<td>[0.9958, 1.0003]</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>100 nodes</td>
<td>0.9906</td>
<td>[0.9837, 0.9976]</td>
<td>0.9867</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9893</td>
<td>[0.9853, 0.9934]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9913</td>
<td>[0.9842, 0.9984]</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>100 nodes</td>
<td>0.9686</td>
<td>[0.9605, 0.9769]</td>
<td>0.9600</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9660</td>
<td>[0.9566, 0.9754]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9733</td>
<td>[0.9713, 0.9754]</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>100 nodes</td>
<td>0.9123</td>
<td>[0.9041, 0.9205]</td>
<td>0.9100</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9099</td>
<td>[0.8867, 0.9333]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9069</td>
<td>[0.8869, 0.9271]</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>100 nodes</td>
<td>0.9410</td>
<td>[0.8809, 1.0012]</td>
<td>0.9517</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9433</td>
<td>[0.8727, 1.0140]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9509</td>
<td>[0.9064, 0.9956]</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>100 nodes</td>
<td>0.9829</td>
<td>[0.9762, 0.9898]</td>
<td>0.9800</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9769</td>
<td>[0.9733, 0.9807]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9796</td>
<td>[0.9757, 0.9836]</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>100 nodes</td>
<td>0.9800</td>
<td>[0.9764, 0.9836]</td>
<td>0.9800</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9773</td>
<td>[0.9746, 0.9801]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9773</td>
<td>[0.9739, 0.9808]</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>100 nodes</td>
<td>0.9816</td>
<td>[0.9816, 0.9816]</td>
<td>0.9800</td>
</tr>
<tr>
<td></td>
<td>150 nodes</td>
<td>0.9816</td>
<td>[0.9816, 0.9816]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200 nodes</td>
<td>0.9816</td>
<td>[0.9816, 0.9816]</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 6

Supporting Both Periodic and Sporadic Traffic in Large Scale Industrial WSNs

In some WSN applications, multiple types of traffic coexist in the same network: periodic and sporadic traffic [Felemban et al., 2006]. In industrial monitoring processes, periodic traffic is generated for maintaining the performance of the machine processes. Emergency traffic, which occurs sporadically, is used for reporting the major problem to the main controller.

Different types of traffic require different levels of service from the network. The main requirement for periodic traffic is to deliver data packets to the sink node within a specific deadline. Generally, a small number of packet losses is acceptable for periodic traffic, while emergency traffic cannot tolerate the event of packet loss. This is mainly because the information in emergency traffic is critical for the machine processes and must be delivered to the destination node as soon as possible. As a result, an immediate response can be issued on time before the problem affects the overall performance of the system.

In this chapter, we propose specific routing mechanisms for supporting the requirements from periodic and sporadic traffic. A load-balancing technique that considers the external traffic from other source nodes in the network and an adaptive contention period mechanism are developed for periodic traffic. For sporadic traffic, a reliable data transmission mechanism based on multipath routing and a traffic priority system are proposed to support reliable data delivery and guarantee that the packets from sporadic traffic are delivered to the sink node as soon as possible.
6.1 Notations

Some notations in this chapter are listed below.

Table 6.1: Notations used in this chapter

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_a$</td>
<td>Average value of backoff period</td>
</tr>
<tr>
<td>$CW$</td>
<td>Contention window</td>
</tr>
<tr>
<td>$D_i$</td>
<td>Average delay at node ‘i’</td>
</tr>
<tr>
<td>$E(D)$</td>
<td>Average contention time at each node</td>
</tr>
<tr>
<td>$E(ST)$</td>
<td>Average length of a time slot</td>
</tr>
<tr>
<td>$E(X)$</td>
<td>Average number of time slots that a data packet must wait until it can be successfully transmitted</td>
</tr>
<tr>
<td>$e_T$</td>
<td>Total amount of remaining energy of all nodes in the routing path</td>
</tr>
<tr>
<td>$e_c(i)$</td>
<td>Current remaining energy of node ‘i’</td>
</tr>
<tr>
<td>$e_0(i)$</td>
<td>Initial energy of node ‘i’</td>
</tr>
<tr>
<td>$H(\nu)$</td>
<td>Cost of traffic load for the $\nu^{th}$ routing path</td>
</tr>
<tr>
<td>$k_s$</td>
<td>Number of received data packets at sink node</td>
</tr>
<tr>
<td>$o_\nu$</td>
<td>Traffic ratio for the $\nu^{th}$ routing path</td>
</tr>
<tr>
<td>$T_c$</td>
<td>Contention delay</td>
</tr>
<tr>
<td>$T_{c2e}$</td>
<td>Average end-to-end delay of the routing path</td>
</tr>
<tr>
<td>$T_p$</td>
<td>Periodic update period</td>
</tr>
<tr>
<td>$T_{re}$</td>
<td>Average contention time in the case that the data packet must be retransmitted</td>
</tr>
<tr>
<td>$x_\nu$</td>
<td>Total amount of external traffic in path ‘$\nu$’</td>
</tr>
<tr>
<td>$z_\nu$</td>
<td>Total number of node in path ‘$\nu$’</td>
</tr>
</tbody>
</table>

6.2 Routing Mechanisms for Periodic Traffic

In industrial WSNs, there are many types of applications with periodic data traffic. Controlling and maintaining the performance of the machine processes requires information about the current conditions from all important processes. This information is transmitted from the sensor nodes to the main controller every specific period.

A small number of packet losses during the data transmission processes is tolerable for periodic traffic. If only one data packet is lost and the next data packet successfully arrives at the main controller, the main controller can still use the information in the next packet to evaluate the current condition of the machine processes [Jonsson and Kunert, 2009; Willig et al. 2005].
However, if the main controller cannot receive any data packets from the source node for a long period of time, it becomes more difficult for the main controller to evaluate the current condition of the machine processes. Therefore, it is important to prevent the event of consecutive packet losses.

Another important requirement of periodic traffic is to prevent a long transmission delay between the source node and the sink node. If the data packet experiences a long delay, it may not arrive at the sink node before the deadline is expired. As a result, the information inside this data packet is outdated and cannot be used by the main controller.

In order to prevent both consecutive packet losses, the routing paths between source node and sink node must be able to guarantee a high success rate of data transmissions. This requirement can be achieved with the route update and maintenance processes described in Chapter 4.

For preventing the long transmission delay, we aim to limit the impact from various types of delays. There are two main types of delays; queuing delay and contention delay. Queuing delay is measured from the time that the data packet arrives at a specific node until the data packet is processed by this node. The contention delay occurs based on the type of MAC layer protocols implemented in the sensor node. If there is a large number of nodes in the same area, each node must contend for the shared wireless channel. Only one node can access the wireless channel at a time and other nodes must wait until the wireless channel becomes available again.

A load balancing technique based on external traffic information is developed to reduce the queuing delay, and an adaptive contention period mechanism is developed to reduce the contention delay.

### 6.2.1 Load balancing technique based on external traffic information.

A load balancing technique is proposed to reduce the queuing delay. It is implemented in the core routing level, which is described in Chapter 3. The main objective of the load balancing technique is to fairly distribute the traffic load to all available routing paths between each pair of source nodes and sink nodes. Each routing path is only responsible for a portion of the overall traffic from the source node. With the load balancing technique, the chance that the data traffic overloads the routing paths is reduced. Therefore, the value of the queuing delay can be notably
reduced.

Most existing load balancing techniques are based on the assumption that each sensor node belongs to only one routing path [Li et al., 2013; Radi et al., 2010]. However, the routing framework that is proposed in Chapter 3 requires a small number of sensor nodes in the core routing level. The sensor nodes in the core routing level are responsible for maintaining multiple routing paths to the sink node. Restricting a sensor node to only one routing path may not be suitable for the proposed routing framework in Chapter 3, because it increases the number of sensor nodes in the core routing level. A high number of nodes in the core routing level creates high routing overhead, which can decrease the overall performance of the network.

The proposed load balancing technique is specifically designed to allow a sensor node to become a member of multiple routing paths in the core routing level. This approach is more suitable to the routing framework designed in Chapter 3.

From the concept of periodic and local update processes designed in Chapter 4, the load balancing decision occurs at the beginning of each periodic update period \( T_p \). Each source node transmits a data packet through its available routing paths based on the information from multiple load balancing parameters. Only one routing path will be used for transmitting the data packet at a time. This constraint is designed to prevent interference between the routing paths that belong to the same source node.

There are three load balancing metrics based on cross-layer information for determining the amount of data traffic for each routing path: (1) Packet Reception Ratio \((PRR)\) from the application layer; (2) remaining energy of each routing path \(e_T\) from the data link layer; and (3) an amount of external traffic from other source nodes \(x\) from the application layer.

The value of \(PRR\) is calculated based on the number of received data packets \(k_s\) and the number of transmitted data packets \(I\) from the previous \(T_p\) period as shown below:

\[
PRR = \frac{k_s}{T} \tag{6.1}
\]

\(e_T\) is calculated from the summation of the remaining energy of all nodes in the same routing path as shown in Equation (6.2):

\[
e_T = \frac{\sum_{j=1}^{z_0} e_{c}(j) - e_{0}(j)}{z_0} \tag{6.2}
\]
where \( z_\nu \) is the total number of nodes in path ‘\( \nu \)’, \( e_c(j) \) is the current level of remaining energy of node ‘\( j \)’ in path ‘\( \nu \)’, and \( e_0(j) \) is the initial energy level of node ‘\( j \)’.

\( x_\nu \) is determined based on the data rate from other source nodes that pass through one or more sensor nodes in the routing path ‘\( \nu \)’.

\[
x_\nu = \sum_{i=1}^{n_x} R_i
\]

(6.3)

where \( n_x \) is the total number of external traffic sources that will pass through the routing path ‘\( \nu \)’ and \( R_i \) is the data rate from external source node ‘\( i \)’.

The values of three load balancing metrics can be retrieved by using the periodic update process in Chapter 4. Each source node requires to transmit a route update packet to the sink node at the end of each \( T_p \) period. The value of \( k_s \) is added to the header of the route update packets at the source node. The route update packet is transmitted to all possible routing paths to the sink node. When the route update packet is processed at each node along the path, the value of \( e_c \) of the current node will be updated in the header of the route update packet.

All load balancing calculations will be performed at the sink node. In this study, we assume that the information of \( x \) for all routing paths is available at the sink node.

The best case scenario for the load balancing calculation is that all nodes in the routing paths do not belong to any external source nodes. For this best case scenario, only \( PRR \) and \( e_T \) are used to determine the proportion of data traffic for each routing path for the next \( T_p \) period. The sink node determines the value of the traffic load cost \( (H(\nu)) \) for the \( \nu^{th} \) routing path as shown below:

\[
H(\nu) = PRR(\nu) \times e_T(\nu)
\]

(6.4)

Let \( y \) represent the total number of routing paths between a pair of source nodes and sink nodes. The traffic ratio for the \( \nu^{th} \) routing path \( (o_\nu) \) can be determined as follows:

\[
\min_{o_\nu} \quad (o_1 \times H(1) = o_2 \times H(2) = \ldots = o_y \times H(y))
\]

subject to \( \sum_{\nu=1}^{y} o_\nu = 1 \)

The total traffic from the source node is equally distributed to all available routing paths as
shown in Figure 6.1.

When some sensor nodes in the routing paths also belong to external source nodes, the value of \( x_\nu \) will be used to adjust the new value of \( o_\nu \). For example, all available routing paths between source node (S1) and sink node (D1) are separated into two groups: node-disjoint paths (\( ng \)) and shared-node paths (\( sg \)) as shown in Figure 6.2. For each routing path in group ‘\( sg \)’, the new value of the traffic ratio for the routing path ‘\( \nu \)’ (\( o_\nu^{sg} \)) can be determined as shown below:

\[
o_\nu^{sg} = o_\nu - x_\nu
\] (6.5)

The data traffic that cannot be allocated to the routing path ‘\( \nu \)’ due to the external traffic (\( x_\nu \)) is distributed evenly to all routing paths in group ‘\( ng \)’. The additional traffic load for each routing path in group ‘\( ng \)’ (\( o_e \)) can be calculated as follows:

\[
o_e = \frac{\sum_{i=1}^{n_{sg}} x_i}{n_{ng}}
\] (6.6)

where \( n_{sg} \) is the total number of the routing paths in group ‘\( sg \)’ and \( n_{ng} \) is the total number of routing path in group ‘\( ng \)’.

The total amount of traffic load for each routing path ‘\( \nu \)’ in group \( ng \) can be derived as follows:

\[
o_\nu^{*sg} = o_\nu + o_e
\] (6.7)
6.2. ROUTING MECHANISMS FOR PERIODIC TRAFFIC

6.2.2 Adaptive contention period mechanism.

A high amount of contention delay can substantially increase the overall propagation delay between source node and sink node. A long propagation delay can cause the data packet to miss the deadline at the sink node. Therefore, it is important to provide a routing mechanism that can significantly reduce the contention delay for supporting periodic traffic.

In this study, an adaptive contention period is developed. It is based on Distributed Coordination Function (DCF), which is a well-known MAC layer process in WSNs.

DCF uses CSMA/CA mechanism to control how each node can access the shared wireless channel.

The main responsibility of DCF is to ensure that only one node can transmit the packet in the wireless channel at a time by using the CSMA/CA mechanism. The CSMA/CA mechanism requires that each node must keep monitoring the wireless channel. The node can transmit a data packet only when it finds that the wireless channel is idle for a specific period of time. The objective of this waiting time is to ensure that none of other nodes use the wireless channel. The waiting time is defined as a distributed interframe space (DIFS).

In addition, an exponential backoff is implemented to reduce the probability of packet collision. This backoff period will begin after the DIFS period ends.
Figure 6.3 shows an example of the DCF mechanism when two stations try to access the wireless channel at the same time. For the first transmission effort, the backoff period will be uniformly chosen in the range of 0 to \( w - 1 \). The value of \( w \) is equal to the value of the minimum contention window (\( CW_{\text{min}} \)). If the transmission is unsuccessful, the value of \( w \) is exponentially increased, as shown in Equation (6.8)

\[
CW_i = 2^i \times CW_{\text{min}}
\]  

(6.8)

where \( i \) is the current value of the retransmission attempts.

The CSMA/CA mechanism creates a long delay when there are a large number of nodes in the same area. If many nodes attempt to transmit the data packets at the same time, some nodes may have a high value of backoff period because only one node can access the wireless channel at a time. The data packets of these nodes are likely to experience a significant amount of contention delay.

The adaptive contention period reduces the contention delay by using the transmission time at the source node, which is the cross-layer information from the application layer of the OSI model. The data packet that is transmitted at its source nodes earlier than other data packets in the same contention area will be assigned with a smaller waiting time. With this approach, the data packets that propagate in the network for a long period of time are likely to have a shorter waiting period. As a result, they do not experience a high amount of contention delay and can
arrive at the sink node within the deadline.

The adaptive contention period mechanism can be divided into two components: passive monitoring process and adaptive period assignment process.

- The passive monitoring process is responsible for creating a list of all nodes that must contend for the shared wireless channel ($clist$). The order in $clist$ is based on the transmission time at the source node. The node with the data packet that is transmitted earlier than others will be placed at the top of the $clist$.

- The adaptive period assignment process uses the information in $clist$ to assign the value of Adaptive Interframe Space (AIFS). The node on the top of $clist$ is assigned the smallest value of AIFS, while the node on the bottom of $clist$ is assigned a large value of AIFS.

AIFS function is similar to DIFS in the CSMA/CA mechanism. It is the time that each node must wait until the backoff period can be active.

**Passive monitoring process.**

We define the monitoring period ($T_{mt}$) as the summation of the AIFS period and the number of backoff time slots as shown in Figure 6.4. Each node that has a data packet to transmit must wait until $T_{mt}$ is expired before it can begin to transmit the data packet into the wireless channel.

During the $T_{mt}$ period, the passive monitoring process in node ‘$j$’ will perform the following tasks if this node detects that node ‘$i$’ is transmitting a data packet to the wireless channel.

- Node ‘$j$’ retrieves the information about the transmission time at the source node ($T_{si}$) in the header of the data packet from node ‘$i$’.

- Node ‘$j$’ compares the value of the transmission time at its source node ($T_{sj}$) with $T_{si}$. The node with a smaller value of the transmission time is promoted as the first item in the $clist_j$.

The passive monitoring process can be implemented based on the periodic update time ($T_{pu}$) as detailed in the route update and maintenance processes in Chapter 4.
Algorithm 5 Passive monitoring process

1: During each $T_p$ period
2: At node ‘i’ in the routing path
3: if the data packet from other nodes can be monitored during $T_{mt}$ period then
4: Read the value of $T_s$ from the header of the data packet
5: if $clist_i = \text{NULL}$ then
6: Compare $T_s$ of other node with $T_s$ of node ‘i’
7: Create $clist_i$
8: Sorting the node’s address with smaller value of $T_s$ and its $T_s$ to the top of the list
9: else
10: Compare $T_s$ of the data packet with the information in $clist_i$
11: Sorting the node’s address with smaller value of $T_s$ and its $T_s$ to the top of the list
12: end if
13: end if

The detailed algorithm of this process is presented in Algorithm 5.

Adaptive period assignment process.

The adaptive period assignment process in each node is based on the information in its $clist$. The node on the top of the $clist$ is assigned the smallest value of AIFS ($AIFS_{min}$). The smallest value of AIFS can guarantee that this node can access the wireless channel before other nodes because its $T_{mt}$ will reach zero before other nodes, as shown in Figure 6.4.

For the first $T_p$ period, all nodes still do not create any $clist$. The value of AIFS for all nodes in the first $T_p$ period will be equal to the default value of AIFS ($AIFS_d$).
In order to ensure that all nodes in the same contention area have the same information in their clist, each node creates a special update packet (pktU). The pktU packet contains the clist of its creator. Then, the pktU packets are exchanged between all nodes in the same contention area. If the clist inside the received pktU packet is similar to the clist of the received node, this node will reply back to the source node of the pktU packet with an ACK packet (ackU). Conversely, if the clist inside the received pktU packet is different from the clist of the node, a notification packet (nofU) will be sent to the source node of the pktU packet.

After the adaptive period assignment process is completed in the current $T_p$ period, all nodes still repeat the passive monitoring process in Algorithm 5 in the next $T_p$ period. However, if there is no change in the clist, the exchanges of pktU at the end of the next $T_p$ period will not occur.

The detailed algorithm of the adaptive period assignment process is presented in Algorithm 6.

**Algorithm 6 Adaptive period assignment process**

1. At the beginning of the first $T_p$ period
2. $AIFS := AIFS_i$ for all nodes in the network
3. At the end of each $T_p$ period for node ‘i’
4. if $clist_i \neq NULL$ then
5. Send special update packets(pktU) to all node in $clist_i$
6. end if
7. if pktU is received then
8. Compare clist in pktU with $clist_i$
9. if Both clist are similar then
10. Send ACK packet (ackU) back to the source node of pktU
11. else
12. Send notification packet (nofU) back to the source node of pktU
13. end if
14. end if
15. if nofU is received then
16. Wait until it received all pktU packets from all node in $clist_i$
17. Update $clist_i$ based on received information
18. Send pktU to all node in $clist_i$
19. end if
20. if ackU is received then
21. if ackU from all nodes in $clist_i$ are received then
22. Find order of node ‘i’ ($a_i$) in $clist_i$
23. $AIFS_i := a_i \times AIFS_{min}$
24. end if
25. end if
6.2.3 Mathematical modelling of the adaptive contention period mechanism.

In order to evaluate the performance of the adaptive contention period mechanism, the mathematical modelling for determining the end-to-end contention delay of each routing path \( T_{e2e} \) is presented in this section.

The value of contention delay at each node \( T_c \) will be calculated based on the average time that the node must wait until it can transmit the data packet \( E(D) \) and the average of contention time in the case that the data packet must be retransmitted \( T_{re} \).

\[
T_c = E(D) + T_{re}
\]  
(6.9)

\( T_{re} \) can be calculated based on the probability that the packet transmission will be successful in \( k \) attempts \( (P_{sk}) \) as shown in Equation (6.10)

\[
T_{re} = \frac{\sum_{k=1}^{m} (P_{sk} \times I \times E(D))}{\sum_{k=1}^{m} (P_{sk} \times I)}
\]  
(6.10)

where \( m \) is the maximum number of retransmissions at each node and \( pk \) is the total number of transmitted packets.

In order to determine the value of \( E(D) \), we use the Markov chain models proposed by Tantra et al. [2005] to represent the original CSMA/CA process. Figure 6.5 shows the Markov chain model for a normal CSMA/CA process. Figure 6.6 presents the Markov chain model for a CSMA/CA process that implements the adaptive contention period mechanism. The value of \( E(D) \) can be calculated with Equation (6.11)

\[
E(D) = E(X) \times E(ST)
\]  
(6.11)

\( E(X) \) is the average number of slots that a data packet must wait until it can be successfully transmitted. It can be calculated as shown below:

\[
E(X) = \sum_{i=0}^{r} \frac{(p^i - p^{r+1}) W_{i+1}}{1 - p^{r+1}}
\]  
(6.12)

where \( p \) is the collision probability, which can be derived from the Markov chain models in Figures 6.5 and 6.6. \( E(ST) \) is the average length of a slot and can be calculated with Equation
6.2. ROUTING MECHANISMS FOR PERIODIC TRAFFIC

\[ E(ST) = P_I \sigma + P_S T_S + P_C T_{co} \] (6.13)

where \( \sigma \) is the length of a slot time, \( P_I \) is the probability that a slot is idle, \( T_S \) is the average length of successful transmission, and \( T_{co} \) is the average length of collision. The value of \( T_S \) and \( T_{co} \) can be calculated as follows:

\[ T_S = H + E(P) + SIFS + \delta + ACK + AIFS_{min} + \delta \] (6.14)

\[ T_{co} = H + E(P) + AIFS_{min} + \delta \] (6.15)

We assume that there are two types of nodes in a routing path: contending nodes \((N_c)\) and contention-free nodes \((N_f)\). For contending nodes, the average delay of this type of nodes \((D_f)\) can be determined as follows:

\[ D_f = DIFS + B_a \] (6.16)

where \( B_a \) is the average value of backoff period selection from the range \((0, CW_{min})\).
Lastly, the average end-to-end contention delay for each routing path \( T_{e2e} \) can be calculated as shown below:

\[
T_{e2e} = (N_f \times D_f) + (N_c \times T_c)
\]  

(6.17)

### 6.3 Routing Mechanisms for Sporadic Traffic

For industrial WSNs, the most important traffic class is the safety class [Shen et al., 2014]. The safety traffic category is the type of traffic that requires emergency action. The data packets in this traffic class are required to be transmitted to the sink node as soon as possible. Moreover, these data packets are critical to industrial applications and they must be guaranteed to arrive
6.3. ROUTING MECHANISMS FOR SPORADIC TRAFFIC

successfully at the sink node. Safety traffic is likely to occur sporadically. Therefore, it is possible to allocate a high amount of network resources to support the strict requirements of this traffic class.

There are two objectives that need to be achieved in order to support sporadic traffic. Firstly, the data packets from this traffic class will be guaranteed to successfully arrive at the sink node. In order to accomplish this goal, the core routing nodes, which maintain multiple routing paths to the sink node, use all possible routing paths to transmit the data packet from sporadic traffic class. Secondly, the data packet from the sporadic traffic class is delivered to the sink node as soon as possible. The priority system will be implemented. The data packet from the sporadic traffic class will be assigned a priority that is higher than the data packet from other traffic classes.

6.3.1 Priority system for different class of traffic.

In this study, there are two classes of data traffic.

- Class 1: Sporadic traffic. The packets from this class will be assigned with priority (\(pr\)) equal to 0. The packet from class 1 requires to be transmitted to its destination as soon as possible.

- Class 2: Periodic traffic. The packets from this class will be assigned with \(pr\) equal to 1.

At source node, the value of \(pr\) will be added to the header of every data packet. In order to support both traffic classes, priority queues are implemented at each node as shown in Figure 6.7. Q1 is the high priority queue that will serve the traffic from Class 1. If Q1 is not empty, the packet from Q1 will be served first. If Q1 is empty, the packet from Class 2 will be served.

6.3.2 High transmission reliability with multipath routing.

When a core routing node receives a data packet that belongs to Class 1, it will transmit this data packet into all routing paths that are stored in its routing table, as shown in Figure 6.8. Transmitting the data packets through all possible routing paths may notably increase the amount of data traffic. However, the data packet from Class 1 is unlikely to occur very often.
To support the high level of requirements from emergency traffic, allocation of a high number of network resources is an appropriate solution in this case.
6.4 Verification Through Simulations

In this section, verification of the proposed methods is divided into two parts. The first part focuses on the verification of the proposed methods for supporting periodic traffic. The proposed load balancing technique will be compared with load balancing based on a round robin approach. We will use a mathematical analysis to compare the performance of the proposed adaptive contention period mechanism with a normal CSMA/CA mechanism. In the second part, the simulation studies are setup to verify the performance of the proposed methods for supporting the sporadic traffic.

6.4.1 Performance of routing mechanisms for periodic traffic.

The load balancing at core routing level.

In order to verify the proposed load balancing technique, a 100 node topology is created in the NS-2.34 simulator.

For the simulation, we will focus on a scenario with two pairs of source nodes and sink nodes. Each pair of nodes has two possible routing paths, as shown in Figure 6.9. Node ‘X’ in path ‘C’ is also a member of path ‘B’. There are three different data rates (0.1, 0.2 and 0.5 packets/second) implemented in node S1 and node S2. The configuration for the load balancing technique based on the round robin approach will aim to evenly distribute the data traffic into all routing paths.

The simulation environment is based on the following configuration parameters. The data link layer is IEEE 802.11. The network communication model is Two-Ray Ground. There are two CBR traffic sources to generate data packets of the size of 100 Bytes every 2, 5 and 10 seconds. The transmission range of each sensor node is 40 m. The simulation timespan has been set to 10,000 seconds.

The performance metrics for comparing the load balancing techniques is network lifetime. The value of network lifetime will start from the time that the network begins its operations to the time that it cannot deliver the data packet to the destination.

The values of network lifetime from the simulation scenarios are the average values of the results from five independent simulation runs with the same configuration parameters. All the
average values of this simulation study pass the t-test analysis with 95% of confidence intervals.

Figure 6.9 shows the results in terms of network lifetime from the load balancing technique we developed in this study, and the load balancing technique based on a round robin approach. At low data rate (0.1 packets/second), our proposed work performs slightly better than the round robin load balancing technique. The percentage of improvement is about 1.2%. As the data rate increases, the results from our load balancing technique clearly performs better than the round robin one. For the data rate equals to 0.5 packets/second, the result from our proposed technique is about 5.1% better.

The main reason that our proposed work performs better in terms of network lifetime is that our proposed load balancing technique considers the external traffic from multiple source nodes. The load balancing technique based on a round robin approach only aims to evenly distribute the data traffic to all available routing paths. As a result, it is likely that the routing paths for the round robin approach will run out of energy more quickly than the routing paths based on our proposed technique.
6.4. VERIFICATION THROUGH SIMULATIONS

Table 6.2: Parameters for verifying the adaptive contention period mechanism

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma$</td>
<td>50 $\mu$s</td>
</tr>
<tr>
<td>DIFS</td>
<td>128 $\mu$s</td>
</tr>
<tr>
<td>SIFS</td>
<td>28 $\mu$s</td>
</tr>
<tr>
<td>$\text{AIFS}_{\text{min}}$</td>
<td>128 $\mu$s</td>
</tr>
</tbody>
</table>

The adaptive contention period mechanism.

![Graph showing network lifetime comparisons](image)

Figure 6.10: Comparisons of network lifetime for the two load balancing techniques

Figure 6.11 presents the network topology for this case study. There are two routing paths; path ‘A’ and path ‘B’. Node ‘i’ in path ‘A’ and node ‘j’ in path ‘B’ are located within each other’s transmission range. In order to simplify the mathematical analysis, we consider that both path ‘A’ and path ‘B’ have the same number of hop counts. The source nodes of both path ‘A’ and path ‘B’ always have the data packets to transmit. If the source node of path ‘A’ transmits a data packet at time $t_0$, the source node of path ‘B’ will transmit a data packet at time $t_0+X$. As a result, node ‘i’ will be assigned a smaller value of AIFS. All CSMA/CA parameters and the value of AIFS are presented in Table 6.2.
CHAPTER 6. SUPPORTING BOTH PERIODIC AND SPORADIC TRAFFIC

• Case 1: Normal CSMA/CA is implemented. The Markov chain model in Figure 6.5 is used to compute the average packet delay \( E(D) \) of a node that must contend for the wireless channel. The number of nodes that must contend for the wireless channel is varied from 1 to 4. The total number of nodes in both path ‘A’ and path ‘B’ is equal to 10.

• Case 2: Adaptive contention period mechanism is implemented. For this case, node ‘i’ of path ‘A’ is assigned with \( AIFS_{min} \) and node ‘j’ of path ‘B’ is assigned the value of \( AIFS = AIFS_{min} \times 2 \). The Markov chain model from Figure 6.6 is used to calculate the value of \( E(D) \). The topology configuration is similar to the configuration in Case 1.

Figure 6.12 shows the results from the mathematical analysis that compare the value of \( T_{e2e} \) from Case 1 and Case 2. This clearly shows that when the proportion of contending nodes in the routing path increases, the values of end-to-end delay from the adaptive contention period mechanism get better than the end-to-end delay from normal CSMA/CA protocol. The adaptive contention period mechanism assigns a smaller value of AIFS to the node that requires to transmit the data packet more quickly than the other node. It not only delivers the data packet to the destination as soon as possible but also reduces the probability that each node finds the wireless channel busy. It is likely that when the node is assigned a larger value of AIFS, the node will be ready to transmit the data packet when the other node with a smaller AIFS has finished its data transmission process.

While the adaptive contention period mechanism has an additional computation requirement at each node, we assume that the sensor node does not frequently change its position. After the contending list (clist) is created in all nodes inside the congestion area, each node can maintain the same configuration for a long period of time.
6.4. VERIFICATION THROUGH SIMULATIONS

Performance comparisons for supporting periodic traffic.

With the integration of the proposed load balancing technique and the adaptive contention period mechanism, two simulation scenarios with different network sizes (100 and 200 nodes) are created to compare the performance of our proposed works with the other three well-known routing protocols (AODV, AOMDV and DSDV) as benchmarks. The selection of routing protocols has been justified in Chapter 3.

The simulation setup is almost similar to that for the simulation scenarios for verifying the load balancing technique. In addition, a two-state error model with error rate $P_{err}$ is implemented in a member of Routing path ‘D’ in this simulation. This error model will cause the condition of Routing path ‘D’ to remain in good condition ($P_{err} = 0$) for 300 seconds and then change to poor condition with an error rate equal to $P_{err}$ for the next 1200 seconds. The value of $P_{err}$ varies from 0.2 to 0.8 for each simulation scenario. This process repeats in the same manner for the whole network operation period. The simulation timespan has been set to 6000 seconds for all scenarios.

The average end-to-end delay is used as the performance metric. The value of end-to-end delay will be calculated from the time that a data packet is transmitted from the source node to the time that it reaches the sink node. The values of the average end-to-end delay from the

![Figure 6.12: Comparisons of end-to-end delays with CSMA/CA](image)
Table 6.3: End-to-end delay in milliseconds from 100 node topology

<table>
<thead>
<tr>
<th>Protocols</th>
<th>Error=0.2</th>
<th>Error=0.4</th>
<th>Error=0.6</th>
<th>Error=0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our work</td>
<td>32.5</td>
<td>36.3</td>
<td>32.61</td>
<td>31.9</td>
</tr>
<tr>
<td>DSDV</td>
<td>31.08</td>
<td>61.24</td>
<td>303.44</td>
<td>441.06</td>
</tr>
<tr>
<td>AODV</td>
<td>30.62</td>
<td>38.68</td>
<td>40.28</td>
<td>38.79</td>
</tr>
<tr>
<td>AOMDV</td>
<td>30.85</td>
<td>38.93</td>
<td>38.92</td>
<td>38.78</td>
</tr>
</tbody>
</table>

Table 6.4: End-to-end delay in milliseconds from 200 node topology

<table>
<thead>
<tr>
<th>Protocols</th>
<th>Error=0.2</th>
<th>Error=0.4</th>
<th>Error=0.6</th>
<th>Error=0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our work</td>
<td>64.88</td>
<td>67.92</td>
<td>64.84</td>
<td>65.19</td>
</tr>
<tr>
<td>DSDV</td>
<td>63.8</td>
<td>141.16</td>
<td>387.17</td>
<td>464.17</td>
</tr>
<tr>
<td>AODV</td>
<td>60.13</td>
<td>74.86</td>
<td>75.3</td>
<td>76.98</td>
</tr>
<tr>
<td>AOMDV</td>
<td>75.73</td>
<td>75.79</td>
<td>75.83</td>
<td>75.75</td>
</tr>
</tbody>
</table>

simulation scenarios are the average values of the results from five independent simulation runs with the same configuration parameters.

Figure 6.13: End-to-end delay of 100 node topology

The results from Table 6.3 and Table 6.4 can clearly indicate that DSDV produces the worst end-to-end delay performance for all network sizes (100 and 200 nodes) we simulated. The difference between the end-to-end delay of DSDV and the other three protocols significantly
6.4. VERIFICATION THROUGH SIMULATIONS

Figure 6.14: End-to-end delay of 200 node topology

increases when the size of the network becomes larger. For 200 node topology, the end-to-end delay of DSDV with $P_{err}$ equals to 0.8 is more than 10 times larger than the others. In DSDV, the simple periodic update is used to propagate the routing information to all nodes in the network. This update technique requires a significant amount of time for delivering the update information to all nodes in the network. As a result, the source node is likely to use a poor routing path for a considerable period until it can receive the new update information and switch to another available path.

Figures 6.13 and 6.14 show that the proposed work in this chapter can provide the best result in terms of end-to-end delay when compared with AODV and AOMDV. The proposed work only provides a higher value of end-to-end delay than AODV and AOMDV when $P_{err}$ equals to 0.2. When the value of $P_{err}$ is higher than 0.2, the proposed work can provide a smaller amount of end-to-end delay than other two routing protocols for both 100 node and 200 node topology.

6.4.2 Performance of routing mechanisms for sporadic traffic.

In order to evaluate the performance of the proposed routing mechanisms for sporadic traffic, PRR is used as the performance metric. The value of PRR from our proposed mechanisms will
### Table 6.5: PRR from 100 node topology

<table>
<thead>
<tr>
<th>Protocols</th>
<th>Error=0.2</th>
<th>Error=0.4</th>
<th>Error=0.6</th>
<th>Error=0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our work</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DSDV</td>
<td>0.97</td>
<td>0.94</td>
<td>0.8016</td>
<td>0.6916</td>
</tr>
<tr>
<td>AODV</td>
<td>0.9967</td>
<td>1</td>
<td>0.99167</td>
<td>0.9883</td>
</tr>
<tr>
<td>AOMDV</td>
<td>1</td>
<td>0.9933</td>
<td>0.9967</td>
<td>0.99</td>
</tr>
</tbody>
</table>

### Table 6.6: PRR from 200 node topology

<table>
<thead>
<tr>
<th>Protocols</th>
<th>Error=0.2</th>
<th>Error=0.4</th>
<th>Error=0.6</th>
<th>Error=0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our work</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DSDV</td>
<td>0.9567</td>
<td>0.863</td>
<td>0.6967</td>
<td>0.465</td>
</tr>
<tr>
<td>AODV</td>
<td>0.9967</td>
<td>0.985</td>
<td>0.9967</td>
<td>0.9883</td>
</tr>
<tr>
<td>AOMDV</td>
<td>1</td>
<td>0.9933</td>
<td>0.9933</td>
<td>0.9967</td>
</tr>
</tbody>
</table>

be compared with the results from AODV, AOMDV and DSDV as benchmarks.

Two network topologies are designed to verify the performance of our proposed routing mechanisms (100 node and 200 node topology).

Most of the simulation setup for the two network topologies are similar to the simulation setup for verifying the load balancing techniques. However, there are two additional setups. (1) Firstly, there is a CBR traffic source at node S2 to generate data packets of the size of 100 bytes. The data packet will be randomly generated with a data rate equals to 0.025 packets/second. (2) Secondly, a two-state error model with error rate $P_{err}$ is implemented in a member of routing path ‘D’. This error model causes the condition of routing path ‘D’ to remain in a good condition ($P_{err} = 0$) for 300 seconds and then change to a poor condition with error rate equal to $P_{err}$ for the next 1200 seconds. The value of $P_{err}$ varies from 0.2 to 0.8 for each simulation scenario. This process repeats in the same manner for the whole network operation period. The simulation timespan has been set to 10,000 seconds for all scenarios.

Tables 6.5 and 6.6 show the values of PRR from AODV, AOMDV, DSDV and the proposed mechanisms. They show that DSDV performs the worst in term of PRR when compared with other three protocols. The main reason is that the route update process of DSDV requires a notable period of time to distribute the update information to all nodes in the network. As the value of $P_{err}$ increases, the PRR value of DSDV decreases sharply. For AODV and AOMDV, both protocols are based on a reactive routing approach. The routing path is created when the
source node has a data packet to transmit. Therefore, it is likely to find the best routing path for the current period. However, they still cannot successfully transmit the data packets to the sink node. Our proposed work can provide 100% successful delivery of the data packets for the sporadic traffic at any value of $P_{err}$. Therefore, it can be concluded that our proposed routing
mechanisms meets the high level of requirements for the data delivery process of sporadic traffic.

6.5 Chapter Summary

For industrial applications, there are multiple types of data traffic that coexist in the same network. Each type of data traffic requires different level of services from the network. For reliable wireless communications in industrial WSNs, all types of data traffic need to be supported in the same network. In this chapter, multiple methods have been proposed to ensure that the level of services from periodic data traffic and sporadic or emergency traffic can be achieved. For periodic data traffic, a load balancing technique and an adaptive contention period mechanism are proposed to guarantee that the data packets are delivered to the sink node as soon as possible. The simulation results show that the combination of our proposed work provides the best performance in term of end-to-end delay when compared with three popular routing protocols DSDV, AODV and AOMDV as benchmarks. For sporadic traffic, this type of traffic is critical to industrial applications. Therefore, it requires the high level of reliable data transmissions. The results from the simulation studies show that our proposed work provides 100% successful data transmission for sporadic traffic. Therefore, both types of data traffic can be guaranteed to receive an acceptable level of services with the proposed techniques in this chapter.

However, a trade-off has to be made in order to accomplish the objective. The proposed techniques require additional computation load in each sensor node, especially at the sink node.
Chapter 7

Conclusions and Future Work

7.1 Summary of the Research

In recent years, WSNs have become an attractive solution to providing better performance to various types of applications. The main reason is due to the significant improvements in micro-electro-mechanical systems (MEMS) technology that enable the manufacturers to create low-cost but powerful wireless sensing devices. The ability of WSNs to be self-organizing makes a more attractive solution than wire-based systems.

Providing reliable data transmission for industrial WSNs is a challenging task. The harsh environments in industrial areas cause noise and interferences, which can decrease the transmission range and quality of the wireless signal. As a result, it is difficult for the sensor node to maintain the current best routing path under this hostile environment.

This research aims to address both the scalability issue and the reliable data transmission issue through effective routing, with utilisation of cross-layer information in industrial WSNs. The overall objective is to improve network layer protocols in order to provide reliable data transmissions for large-scale industrial WSNs. To achieve this objective, a routing framework is developed. It consists of a two-tier routing structure and efficient route update and maintenance processes. Moreover, specific routing mechanisms, which are based on cross-layer information, are developed to support both periodic and sporadic traffic in the same network.

Most of both proactive and reactive routing protocols for WSNs do not work effectively in large-scale network implementation. Many proactive routing protocols are based on a simple periodic update mechanism. Each sensor node must update the routing information to its
neighbors every specific period. In large scale deployment, this update technique creates a large amount of routing overhead, which can significantly reduce the overall performance of the network. Reactive routing protocols are unlikely to create a significant amount of routing overhead but they are likely to experience a long delay from the route discovery process.

In order to provide reliable data transmission in the harsh environments of industrial WSNs, we propose a new routing framework, which consists of a two-tier routing structure and efficient route update and maintenance processes. The two-tier routing structure is developed to provide an effective solution for large-scale network implementation. The efficient route update and maintenance processes aim to provide the ability to quickly respond to the changes in the network condition. It is developed to replace the simple periodic update process and become the new route distribution process for the proposed routing framework.

Furthermore, a mathematical modelling framework for evaluating the performance of the proposed route update and maintenance processes is developed in this thesis. This mathematical framework is used to determine the best values for the input parameters of the proposed route update and maintenance processes.

For industrial WSNs, there are two types of traffic that are crucial to the performance of industrial processes; periodic and sporadic traffic. In order to maintain the performance of the plant processes, the control system must receive periodic data from the machines. This periodic information is used for analyzing and adjust the operation of the machine processes to remain at the expected threshold level. Sporadic data is used to notify potential failures to the control system. This type of data is used to solve the problem before the plant processes are disrupted.

Previous researches in the area of WSNs have little focus on providing reliable data transmissions for multiple types of traffic in the same network. Their solutions are based on the assumption that there is only one type of traffic in the network. Our research aims to provide a specific mechanism based on cross-layer information to guarantee reliable data transmission for both periodic and sporadic traffic. The main reason to provide specific algorithms for each type of traffic is because each type of traffic requires a different level of reliability. Sporadic traffic is more sensitive to the event of a missed deadline than periodic traffic. This is mainly because an appropriate response from the main control system must be issued on time in order to protect the disruption of industrial processes.

A summary of the research contributions in provided as follows:
1. A routing framework has been designed. It consists of a two-tier routing structure and efficient route update and maintenance processes.

The two-tier routing structure aims to reduce the amount of routing overhead from the routing process. The main objective of separating the routing structure into two levels is to limit the area that the routing control packets can propagate in the network. This routing framework is implemented in the NS-2 simulator. The simulation results show that the routing protocol based on the two-tier routing structure significantly reduces the overall routing overhead when compared with AODV, AOMDV and DSDV as benchmarks.

Efficient route update and maintenance processes based on path quality have been developed. These route update and maintenance processes are designed to replace a simple periodic update process that is commonly used in most proactive routing protocols for WSNs. By using a combination of long periodic update and short local update, our proposed work effectively distributes the current condition of the routing paths to all nodes in the network. Moreover, our proposed work also creates a small amount of routing overhead, which is suitable for implementation in large scale networks.

2. A mathematical modelling framework has been created to verify the performance of the proposed route update and maintenance processes. These mathematical models can be used for determining the best values of the input variables for the proposed route update and maintenance processes. A trade-off between the amount of routing overhead and the accuracy of detecting a poor condition in the network can be determined.

3. In order to support both periodic and sporadic traffic in industrial WSNs, four routing mechanisms based on cross-layer information are developed. A load-balancing technique based on information about external traffic and an adaptive contention period mechanism are developed to support periodic data traffic. A reliable data transmission technique based on multipath routing and a priority system for each type of data traffic are designed to support sporadic data traffic. The data packets from this traffic class are guaranteed a the high level of reliability for the data transmissions. In addition, the data packets from the sporadic traffic class are assigned the highest priority. These data packets will be delivered to the sink node as soon as possible. Therefore, an appropriate response can be issued to resolve the problem before it causes a major disruption in the overall system.
7.2 Limitation and Future Work

There are some limitations in the current work that can be improved in future research.

Firstly, the number of sink nodes per each type of data traffic can be extended.

In current work, there is only one sink node for each type of traffic (periodic and sporadic). Increasing the number of sink nodes can provide more options to maintain the overall performance of the network. It increases the number of possible routing paths between each pair of source nodes and sink nodes. As a result, the data traffic can be distributed to all available paths in order to increase the network lifetime. However, adding more sink nodes into the network also increases the level of complexity of many proposed mechanisms in this thesis. Some of the proposed mechanisms may need to be updated in order to support a multiple number of sink nodes per each type of traffic.

Secondly, implementation of the proposed mechanisms in the real sensor networks can be improved.

Some aspects of the network environments are difficult to model in the simulation software, such as the condition of the wireless channel. In this study, we created the simulation scenarios based on a two-state Markov error model and a uniform error model. Implementing the proposed mechanisms in a real sensor network can provide much helpful information that can be used for improving the proposed mechanisms.

Thirdly, dynamic role assignment for core routing nodes and local routing nodes can be added to the proposed routing framework.

Currently, the ratio between the core routing nodes and the local routing nodes is predefined and each node is randomly assigned to become a core routing node or a local routing node for a specific period. For future work, our routing framework could be extended to implement the dynamic role assignment in order to extend the network lifetime.

Finally, an efficient route discovery mechanism based on the quality of the routing path can be added to the proposed routing framework.

In this research study, all routing paths in the networks are assumed to be available for each source node. Practically, finding multiple possible routing paths between each pair of source node and sink node is a complicated task. It is likely to require multiple exchanges of
control packets and a long period of time to complete the route discovery process. Nevertheless, creating an efficient route discovery mechanism would be very helpful in providing the best initial set of routing paths for the data transmissions.
Literature Cited


mac unreliability problem in ieee 802.15.4 wireless sensor networks. *IEEE Transactions 

1–6, Budapest, Hungary.

colony in mobile ad hoc networks. In *3rd IEEE International Conference on Ubi-media 
Computing (U-Media)*, pages 77–82, Jinhua, China.

pages 1–5, Calgary, Alberta, Canada.

104.

and Technologies (EDT)*, volume 1, pages 454–457, Shenzhen, China.

Ben-Othman, J. and Yahya, B. (2010). Energy efficient and qos based routing protocol for 


heterogeneous wireless sensor networks. *IEEE Transactions on Parallel and Distributed 

routing schemes for infrastructure wireless mesh networks. In *8th International Conference 
on Wireless and Optical Communications Networks (WOCN)*, pages 1–6, Paris, France.


on Sensor and Ad Hoc Communications and Networks (SECON), pages 218–228, Santa Clara, California, USA.


