



# **Quality Aware Data Gathering**

**and Disseminating in Chain-based  
Wireless Sensor Networks**

Zahra Taghikhaki

**Quality-aware Data Gathering and  
Disseminating in Chain-based Wireless  
Sensor Networks**

Zahra Taghikhaki



Graduation Committee:

Chairman: Prof.dr.ir. P.M.G Apers  
Promoter: Prof.dr.ing. P.J.M. Havinga  
Assistant Promoter: Dr.ir. N. Meratnia

Members:  
Prof.dr.ir. G.J.M. Smit University of Twente  
Prof.dr.ir T.Tinga University of Twente  
Prof.dr. C. Petrioli University of Rome 'La Sapienza'  
Prof.dr. S.Baydere Yeditepe University  
Dr. P. Guo Wuhan University



**CTIT**

This research is supported by the EU FP7-ICT project GENESI (<http://genesi.di.uniroma1.it/>), under the Grant No. 257916.

CTIT Ph.D. Thesis Series No. 14-344  
Centre for Telematics and Information Technology  
P.O. Box 217, 7500 AE Enschede, The Netherlands.

ISBN: : 978-90-365-3829-9  
ISSN: 1381-3617  
DOI: 10.3990/1.9789036538299  
<http://dx.doi.org/10.3990/1.9789036538299>

Printed by Gildeprint, Enschede

Cover design: Hadis Roghangarha & Mahdi Beheshti  
Cover photo: Vita Vilcina

Copyright © 2015 Zahra Taghikhaki, Enschede, The Netherlands.

All rights reserved. No part of this book may be reproduced or transmitted, in any form or by any means, electronic or mechanical, including photocopying, microfilming, and recording, or by any information storage or retrieval system, without the prior written permission of the author.

QUALITY-AWARE DATA GATHERING  
AND DISSEMINATING IN CHAIN-BASED  
WIRELESS SENSOR NETWORKS

DISSERATION

to obtain  
the degree of doctor at the University of Twente,  
on the authority of the rector magnificus,  
prof.dr.H.Brinksma,  
on account of the decision of the graduation committee,  
to be publicly defended  
on Wednesday 28 January 2015 at 14:45

by

Zahra Taghikhaki

Born on 18 November 1981

In Shahrood, Iran

This dissertation is approved by:

Promoter: Prof.dr.ing. Paul J.M. Havinga

Assistant Promoter: Dr.ir. Nirvana Meratnia

تقدیم به وجود مقدس پدر بزرگوارم، مادر مهربانم و علیرضای عزیزم  
که حضورشان در زندگیم مصداق سخاوت بی ریاست





---

# Abstract

---

Recently wireless sensor network has emerged as a promising technology that could induce an innovation wave in the field of (infra)structures monitoring because of its fast deployment, little interference with the surrounding, self-organization, flexibility and scalability. A key factor for the proliferation of this revolutionary technology is designing effective protocols to meet the quality of service requirements of the application considering deployment properties and characteristics.

Structural condition monitoring using wireless sensor networks can be used for many (infra)structures such as bridge, railways, tunnel, pipelines and highways. These applications exhibit strong similarity in their deployment properties and the way that sensor nodes collect and disseminate their data. Monitoring condition, and operational performance of such large (infra)structures often requires wireless sensor network deployment to long stretch of narrow and elongated spreads which features a linear sensor arrangement and thus its topology resembles a chain. Moreover, ensuring quality of services has been put forward as an essential consideration for wireless sensor networks which are (i) often deployed in unattended and open environments and (ii) characterized by their limited resources and high unreliability. Quality of service in a wireless sensor network can be affected by several constraints out of which (i) the relative position of the node to the base station and other nodes, (ii) the internal reliability state of the network, (iii) the internal reliability state of individual sensor nodes, and (iv) the nodes' available power, are the most important ones. Quality of service support and guarantees in wireless sensor networks especially for linear wireless sensor networks, is an emerging area of research.

In this context, the main focus of this thesis is the design and development of solutions to guarantee combination of four important quality of service parameters, i.e. coverage, long-lifetime, reliability and timeliness for chain-based topology data collection and dissemination. To this end, first we ensure quality of service to some extent at the topology level. However, quality-aware topology control alone is not sufficient to ensure quality of services for disseminating data of many applications

whose packets may convey different types or amount of information. Therefore, we concentrate on using dynamic error control schemes which are allocating the correctional power in an on-demand manner based on both the packet-level constraints and channel state. In this way and for the sake of efficiency, we put the amount of information a packet carries or the time-constrained a sensory data imposes and the state in which the channel is in, into perspective with the amount of effort (in terms of energy expenditure) that is required to reliably transmit the given packets. The main contributions of this thesis can be summarized as follows:

- ✓ **Trust-based probabilistic coverage:** We investigate and address the coverage problem to determine a schedule based on which a selection of the sensor nodes are kept active to efficiently cover the whole monitoring area, using a probabilistic coverage model. By efficient coverage of monitoring area we mean ensuring long network lifetime as well as maintaining sufficient sensing coverage and reliable sensing. Moreover, assuming a probabilistic coverage model we aim to capture the real world sensing and transmitting characteristics of the nodes. In this regard, we propose a trust-based probabilistic coverage algorithm, which leverages the trust concept to tackle the time-varying uncertainties introduced by the sensor nodes and the environment they operate in.
- ✓ **QoS-aware Cluster-head/Chain-leader Selection in a Two-tier Architectural model:** We propose a well-balanced quality of service aware approach to deliver data packets collected by the sensor nodes to the base station, respecting application requirements in addition to coverage. We address three quality of service parameters, i.e., (i) long-lifetime, (ii) reliability, (iii) delay or data freshness. More specifically in this contribution we (i) introduce a two-tier architecture model in order to energy efficiently, reliably and fast aggregate and disseminate sensed data toward the base station, (ii) integrate the three quality of service parameters (long-lifetime, reliability, and delay) with the possibility to adjust their priorities according to the specific application requirements.
- ✓ **QoS-aware Dynamic Chain-Cluster Forming:** In order to relax some assumptions we made before regarding communication capability of the nodes to communicate directly with other nodes or with the base station as well as the fixed-size of the chain-cluster, we propose two solutions which make the size/shape of the clusters adaptive regarding the state of the nodes and links. The proposed solutions well-incorporate energy, delay and transmission reliability together to construct clusters and to select proper cluster heads in each cluster.

## Abstract

---

- ✓ **Reliable Dissemination of Time-Constrained Data:** Meeting the Time-To-Live (TTL) constraint of the sensory data which should reliably be transmitted toward the base station in a low duty-cycle network that suffers from short-term burst errors is the main focus of this contribution. By short-term burst errors we mean the errors which are localized in short-term and occurs in burst forms. In this respect, we propose a runtime adaptive packet-link-local error control scheme that operates based on the links' qualities, packets' TTL, and duty-cycle and is able to counteract periodic short-term burst-errors in a chain topology.
- ✓ **Information-link-aware Data Dissemination:** In the same line of the previous contribution which considered the TTL as one of the packet-level indicator or constrains to ensure quality of service, in this contribution we concentrate on the information-value or amount of information a packet carries as another packet-level indicator. In this way, we propose a Run-time Adaptive FEC-based data dissemination protocol. In the proposed approach, each node decides which error control code to use abiding to the computational constraints of embedded sensors, the information-value of the packet, and the statistical properties of the observed errors for the upward link. This adaptation gives the possibility to vary the code strength and complexity on-demand and on the fly.





---

# Samenvatting

---

De afgelopen jaren hebben draadloze sensornetwerken zich gemanifesteerd als veelbelovende technologie die een golf van innovatie zou kunnen gaan bewerkstelligen, vanwege hun snelle inzetbaarheid, de geringe verstoring van de omgeving, het vermogen tot zelfstructurering, hun flexibiliteit en schaalbaarheid. Van doorslaggevend belang voor het succes van deze revolutionaire technologie is het ontwerpen van effectieve protocollen om aan de 'quality of service' eisen van de toepassing te voldoen en daarbij rekening te houden met de omgeving en eigenschappen van het netwerk.

Draadloze sensornetwerken kunnen worden ingezet voor het bewaken van structurele integriteit, het zogenaamde 'structural condition monitoring', van infrastructuur, zoals bruggen, spoorwegen, tunnels, pijpleidingen en snelwegen. Deze toepassingen lijken veel op elkaar met betrekking tot de manier waarop de netwerken worden ingezet en de wijze waarop de sensor nodes gegevens verzamelen en verspreiden. Voor het monitoren van deze vaak grote bouwwerken is vaak een draadloos sensor netwerk nodig dat bestaat uit een of meerdere lange lineaire rijen van sensoren; de netwerk topologie lijkt op een ketting. In dit soort toepassingen wordt quality of service als zeer belangrijk gezien omdat (i) de netwerken onbeheerd en in de open lucht gebruikt worden en (ii) vanwege de inherente onbetrouwbaarheid en gelimiteerde middelen van de sensor nodes. Quality of service wordt beïnvloed door verschillende factoren waarvan (i) de relatieve positie van een node tot zijn basis station en andere nodes, (ii) de interne betrouwbaarheidstoestand van het netwerk, (iii) de interne betrouwbaarheidstoestand van elke sensor node, en (iv) de beschikbare hoeveelheid energie van elke node, de belangrijkste zijn. De mechanismen en methoden die quality of service in draadloze sensornetwerken garanderen, en dan met name lineaire draadloze sensornetwerken garanderen, zijn onderwerpen van recent opkomend onderzoek.

Het hoofddoel van dit proefschrift is het ontwerpen en ontwikkelen van oplossingen om de vier belangrijke quality of service parameters te kunnen garanderen, namelijk dekkinggraad, netwerk levensduur, betrouwbaarheid, en

tijdigheid van informatieverspreiding in ketting-gebaseerde draadloze sensornetwerken. Om dit te bereiken wordt de quality of service tot op zekere hoogte gegarandeerd op topologieniveau. In veelvoorkomende toepassingen, waarbij de te verspreiden data bestaat uit verschillende types of hoeveelheden informatie, is quality of service regulering op topologieniveau alleen niet voldoende om de kwaliteit te garanderen. Daarom richten wij ons op het gebruik van dynamische foutcorrectie technieken die foutcorrectie toepassen daar waar en wanneer het nodig is, gebaseerd op zowel de packet-niveau eisen als op de kanaal toestand. Gegeven de hoeveelheid informatie van een packet, zijn toegestane verspreidingstijd en de toestand van het kanaal brengen we, via de totale energie die nodig is om de informatie foutloos te versturen, de efficiëntie van deze techniek in kaart. De hoofdbijdragen van dit proefschrift kunnen als volgt worden samengevat:

- ✓ **Vertrouwensgebaseerde probabilistische dekking:** We onderzoeken het netwerkdekkingsprobleem met behulp van een probabilistisch model om tot een tijdsschema te komen waarbij een deel van de sensornodes actief gehouden wordt, met als doel op efficiënte wijze het hele monitoringsgebied te kunnen bewaken. Met efficiëntie wordt een zo lang mogelijke netwerklevensduur bedoeld met tegelijkertijd voldoende en betrouwbare monitoring. Via het probabilistische model leggen we het monitoringsgedrag en de zendeigenschappen van de nodes, zoals deze in de praktijk zijn, zo goed mogelijk vast. Wij stellen een probabilistisch dekkingsalgoritme voor dat vertrouwen exploiteert om zo de tijdafhankelijke onzekerheid van de sensornodes en van hun omgeving aan te pakken.
- ✓ **QoS-bewuste Cluster-/Ketting-hoofd Selectie in een Tweelaags Architectuur-model:** We stellen een uitgebalanceerde quality-of-service-bewuste methode voor voor het afleveren van datapackets die door sensornodes zijn verzameld waarbij we zowel de toepassingseisen als de netwerkdekkingseisen in acht nemen. Hierbij beschouwen we drie quality-of-service parameters: (i) netwerklevensduur, (ii) betrouwbaarheid en (iii) vertraging ofwel dataversheid. In deze bijdrage (i) introduceren we een twee-laags architectuur voor om op energie-efficiënte wijze en met hoge betrouwbaarheid de verzamelde gegevens tijdig door het netwerk te verspreiden richting het basisstation, (ii) integreren we de drie eerdergenoemde quality-of-service parameters met de mogelijkheid hun onderlinge prioriteit aan te passen aan de eisen van de toepassing.

- ✓ **QoS-bewuste Dynamische Ketting-Cluster Vorming:** De communicatie-paden tussen nodes onderling en van node naar basestation kunnen niet altijd voor beschikbaar worden aangenomen. Ook is de grootte van het ketting-cluster niet constant. Om het effect van deze aannames te verminderen stellen we twee oplossingen voor die de grootte en vorm van de clusters adaptief maken, afhankelijk van de verbindingen en de toestand van de nodes. Deze oplossingen beschouwen gelijktijdig energie, vertraging van de informatie en de verbindingsskwaliteit om clusters samen te stellen en voor het kiezen van het juiste clusterhoofd in elk cluster.
- ✓ **Betrouwbare Verspreiding van tijd-kritische Gegevens:** Het voldoen aan de Time-To-Live (TTL) eisen van de sensor gegevens, die betrouwbaar naar de basestation verzonden moeten worden via een netwerk dat met een lage duty-cycle opereert en last heeft van korte termijn burst fouten, is het hoofddoel van deze bijdrage. Met korte termijn burst fouten bedoelen we fouten die zich in een korte tijdsspanne en opeenvolgend voordoen. Om met deze situatie om te gaan stellen een lokaal run-time adaptief packet link fout corrigerend algoritme voor dat werkt op basis van de kwaliteiten van de verbindingen, de packet TTL en de duty-cycle van het netwerk. Het algoritme kan korte periodieke burst fouten in een ketting topologie ongedaan maken.
- ✓ **Informatieverbindingsbewuste Gegevens Verspreiding:** Vergelijkbaar met de vorige bijdrage, waar de TTL als packet-niveau kwaliteitsindicator gebruikt werd, beschouwt deze bijdrage de informatie-waarde of –hoeveelheid van een packet als kwaliteitsindicator. In deze bijdrage stellen we een run-time adaptief FEC dataverspreidingsprotocol voor. Elke node beslist welke fout-corrigerende code er gebruikt wordt waarbij de rekeneisen van de node, de informatiewaarde van het packet, en de statistische eigenschappen van de geobserveerde fouten in de uplink meegenomen worden. Door de adaptieve eigenschap van het protocol is het mogelijk om de sterkte van de fout-corrigerende code en zijn complexiteit al naar gelang de omstandigheden onmiddellijk aan te passen.





---

# Acknowledgements

---

*One attains only what he strives for, and that his efforts will definitely be witnessed.*

Quran, 53:(39-40)

Pursuing a Ph.D. project is a both painful and enjoyable experience. It is just like climbing a high peak, step by step, accompanied with bitterness, hardships, frustration, encouragement and trust and with many people's kind help. The support of friends and colleagues have made this whole endeavor more than worthwhile though, and I am very grateful to them all. Some I would like to mention more specifically.

Above all, I owe it all to Almighty God for granting me the wisdom, health and strength to undertake this research task and enabling me to its completion.

Special thanks are given to Prof. Paul Havinga for giving me the opportunity to work on the Pervasive System group and giving me directions during the PhD career. I would also like to express my sincere gratitude to Dr. Nirvana Meratnia, my daily supervisor, for her supervisions and her kind attitude.

I would like to thank Prof. Gerard Smit, Prof. Tiedo Tinga, Prof. Sebnem Baydere, Prof. Chiara Petrioli, and Dr. Peng Guo for accepting being part of my committee. I feel honored to have such experts in my defense.

I would like to thank one of my best colleague, Niels, who offered me a lot of friendly help and also translated the abstract of my thesis into Dutch.

Everyday working life would not have been so joyful without my colleagues in the Pervasive Systems group, my sincere thanks go to all PS members.

My gratitude also goes to Ellen van Erven, the support-staff from the International Office of the university, for her kind help and invaluable support throughout all these years.

## Acknowledgements

---

Many thanks to our secretaries, Nicole Baveld, Thelma Prenger, Marlous Weghorst, for providing information and assistance in arranging many things in these years.

I am so grateful to my paranymphs, Maryam and Hajar, not just because they agreed to be by my side on the defense day but because they are two of my very dearest friends.

I would like to appreciate two wonderful friends of mine, Hadis and Mahdi for designing the thesis cover.

I would like to extend my gratitude to all my Iranian friends in the Netherlands for the numerous good moments we shared during this time.

I would like to acknowledge the tremendous sacrifices that my lovely parents made to ensure that I had an excellent education. It is through their encouragement and care that I have made it through all the steps to reach this point in life, and I could not have done it without them. For this and much more, I am forever in their debt. It is to them that I dedicate this dissertation.

I would like to convey my heartfelt thanks to my beloved brothers Amin, Ali and my wonderful sister Reyhane for being a source of moral encouragement during times of distress.

I would like to express my sincere gratitude to my father in-law who was always thinking the best of me. You are forever in my heart and you will never be forgotten. May you forever rest in peace. A special thanks to my mother in-law who her prayers were always with me and constantly provides emotional support.

Without hesitation, my greatest thanks are for my love, my best friend and my husband, *Alireza*. Alireza jan, these past several years have not been an easy ride, both academically and personally. I truly thank you for sticking by my side, even when I was irritable and depressed. I feel we both learned a lot about life and strengthened our commitment and determination to each other and to live life to the fullest. I will forever be thankful to God for gifting me with you. I know I can always count on your unconditional love and support ...

Zahra Taghikhaki  
January 2015  
Enschede, The Netherlands

# Contents

---

<b>1 Introduction</b> .....	<b>1</b>
1.1. Application domains of long linear wireless sensor networks .....	2
1.2. Network topology and data dissemination .....	4
1.3. The need of quality of service for data traffic .....	7
1.3.1 QoS in dense or sparse nodes deployment .....	8
1.3.2 QoS guarantees in a linear topology.....	9
1.3.2.1 Long lifetime .....	10
1.3.2.2 Reliability .....	11
1.3.2.3 Timeliness.....	14
1.3.2.4 Coverage.....	15
1.4. Research objectives.....	16
1.5. Thesis contributions .....	17
1.6. Thesis organization .....	19
1.7. Bibliography .....	21
<b>2 Trust-based Probabilistic Coverage</b> .....	<b>25</b>
2.1. Introduction.....	26
2.2. Problem statement and our contributions.....	27
2.3. Related work.....	28
2.3.1 Going beyond the existing solutions.....	32
2.4. Assumptions and models used .....	33
2.4.1 Assumptions .....	33
2.4.2 Models used.....	35
2.4.2.1 Node and link uncertainty models.....	35
2.4.2.2 Reputation and trust models .....	37
2.5. Trust-based probabilistic coverage .....	39
2.5.1 Calculating node's confidence level .....	41
2.6. A trust-based probabilistic ILP-based coverage algorithm (TPC) .....	44
2.6.1 A greedy trust-based probabilistic heuristic algorithm (TPC-Greedy) ...	47
2.6.1.1 Parameter definition for TPC-Greedy .....	47
2.6.1.2 TPC-Greedy algorithm .....	48
2.7. Performance evaluation .....	51
2.7.1 Performance metrics .....	52
2.7.2 Simulation setup and scenario .....	52
2.7.3 Performance evaluation .....	53
2.8. Chapter summary .....	58



---

2.9. Bibliography .....	59
<b>3 Quality of Service Aware Chain-Cluster-Based Data Dissemination .....</b>	<b>63</b>
3.1. Introduction.....	64
3.1.1 The need for aggregation-aware data dissemination.....	66
3.2. Related work.....	68
3.2.1 Data dissemination taxonomy.....	69
3.2.2 Chain-based data dissemination.....	70
3.2.3 Cluster-based data dissemination.....	72
3.3. Problem statement and our contribution .....	74
3.4. Assumptions and models used .....	75
3.4.1 Two-tiers architecture model .....	76
3.5. Quality of service aware cluster head selection .....	77
3.5.1 Transmission reliability .....	78
3.5.1.1 Transmission reliability in dense chain-based networks .....	79
3.5.1.2 Transmission reliability in sparse chain-based networks .....	81
3.5.2 Lifetime .....	82
3.5.3 Delay.....	82
3.6. A reliability-, energy-, and delay-aware data dissemination in linear topology... 83	
3.6.1 First-tier: intra-cluster chain .....	84
3.6.1.1 Phase I: initialization .....	84
3.6.1.2 Phase II: situation-aware data gathering.....	84
3.6.1.3 Phase III: leader election .....	85
3.6.2 Second tier: inter-cluster chain .....	86
3.6.2.1 Phase I: initialization .....	86
3.6.2.2 Phase II: situation-aware data disseminating.....	87
3.6.2.3 Phase III: leader election .....	87
3.7. Performance evaluation of QoS-ACA .....	88
3.7.1 Performance metrics .....	88
3.7.2 Simulation setup and scenarios.....	90
3.7.3 Performance evaluation .....	91
3.8. Enhancing QoS-ACA.....	98
3.8.1 Dynamic cluster formation: REC protocol .....	100
3.8.1.1 Phase I: Cluster-head selection and chain-cluster formation.....	101
3.8.1.2 Phase II: data dissemination .....	105
3.8.2 Distance-aware cluster formation: REC+ protocol.....	105
3.9. Performance evaluation of REC and REC+.....	106

## Contents

---

3.9.1 Performance metrics .....	107
3.9.2 Performance evaluation .....	108
3.10. Chapter summary .....	112
3.11. Bibliography .....	113
<b>4 Reliable Dissemination of Time-Constrained Data .....</b>	<b>117</b>
4.1. Introduction.....	118
4.1.1 The need of adaptive approach .....	119
4.1.2 Burst error vs. Random error .....	120
4.1.3 Delay of error control schemes .....	121
4.2. Assumptions and models used .....	123
4.2.1 Channel model .....	125
4.3. Related work.....	127
4.4. Problem statement and our contribution .....	129
4.4.1 Simultaneous real-timeness and reliability .....	130
4.5. Reliable disseminating time-constrained data (READ) .....	131
4.5.1 Initialization.....	133
4.5.1.1 Calculating fractional portion of nodes .....	133
4.5.1.2 Disseminating initializing information.....	134
4.5.2 Situation-aware data gathering .....	134
4.5.2.1 TTL adjusting.....	135
4.5.2.2 Fairly allocating time slots to nodes .....	137
4.5.2.3 Disseminating data packets .....	138
4.5.3 Updating nodes' fractional portion from time slots .....	139
4.5.3.1 Central updating $Y$ in the base station .....	140
4.5.3.2 Local updating $Y$ in the sensor nodes .....	140
4.6. Performance evaluation .....	141
4.6.1 Performance metrics .....	141
4.6.2 Simulation setup and scenario .....	143
4.6.3 Performance evaluation .....	145
4.7. Chapter summary .....	155
4.8. Bibliography .....	156
<b>5 Information-Link-aware Data Dissemination .....</b>	<b>159</b>
5.1 Introduction.....	160
5.1.1 The need for packet-level FEC .....	162
5.1.1.1 Characteristic of forward error correction codes .....	162
5.2. Assumptions and models used .....	163

---

5.2.1 Channel model .....	164
5.2.2 Reed-Solomon codes .....	164
5.3. Related work .....	165
5.3.1 Link-aware adaptive reliable data dissemination .....	167
5.3.2 Information-aware adaptive reliable data dissemination .....	168
5.4. Problem statement and our contribution .....	172
5.5. An Information-link-aware data dissemination protocol (RAFEC) .....	172
5.5.1 Assigning error control codes to the channel states .....	172
5.5.2 Assessing packet information and link quality .....	173
5.5.2.1 Estimation of packet's information value .....	174
5.5.2.2 Estimation of link quality .....	177
5.5.3 Adaptive packet-link-local error control .....	181
5.5.4 Execution of RAFEC algorithm .....	183
5.5.4.1 Initialization phase .....	183
5.5.4.2 Data dissemination phase .....	184
5.6. Performance evaluation of RAFEC .....	189
5.6.1 Performance metrics .....	189
5.6.2 Simulation setup and scenario .....	191
5.6.3 Performance evaluation .....	192
5.7. Chapter summary .....	198
5.8. Bibliography .....	199
<b>6 Conclusion.....</b>	<b>201</b>
6.1. Contributions revisited.....	201
6.2. Conclusion and lessons learned .....	208
6.3. Future works .....	210
6.4. Bibliography .....	212
<b>About The Author .....</b>	<b>213</b>

## Introduction

---

Wireless sensor networks (WSNs) are a collection of small, low-cost, energy-constrained, easily deployable, and self-organizing sensor nodes that usually collaborate to measure local environmental conditions and events. Generally speaking, wireless sensor nodes perform the following tasks: (i) sense the environment, (ii) communicate with their neighbors, and (iii) in many cases do some (pre)processing on the collected data. Unlike the traditional networks, wireless sensor networks generally depend on (i) a dense deployment of large number of spatially distributed nodes and (ii) collaboration among nodes to carry out their tasks. These unique characteristics make them a powerful platform for pervasive computing.

Applications of wireless sensor networks are diverse ranging from environmental monitoring and control, infrastructure health monitoring and protection, industrial sensing, diagnostics and control, to healthcare [1].

Structural health and condition monitoring using wireless sensor networks are used for many manmade (infra)structures such as bridge [2], railways [3], tunnel [4], pipelines [5] and highways [6] to list just a few. These applications exhibit strong similarity in their deployment and the way that sensor nodes collect and disseminate their data. Monitoring health, condition, and operational performance of such large (infra)structures often requires wireless sensor network deployment to long stretch of narrow and elongated spreads. Generally speaking, almost all of WSN-based monitoring applications whose deployments naturally extend over relatively long distances in an inaccessible area, have two common characteristics as follows:

1. **Linear deployment:** Since these (infra)structures that are monitored have a linear structure, wireless sensor nodes are required to be deployed and aligned in a linear geometry. As the length of these infrastructures is often much larger than their width, their deployment pattern resembles a long linear geometry.
2. **Long operational life requirement:** Typically, all or part of deployment area in long linear applications are inaccessible due to safety reasons or the natural properties of the application whose nodes should be installed inside a structure (e.g. measuring air

temperature inside a civil structure like as bridge). In these cases, changing battery of the sensor nodes is not easily and conveniently feasible. On the other hand, the sensors are expected to remain active and collect data during the service life of the (infra)structure, which in general is more than 50 years for a bridge structure. Therefore, long lifetime requirement and thus carefully power management is usually one of the most critical features of this type of wireless sensor network deployment.

### 1.1. Application domains of long linear wireless sensor networks

Some of the applications of wireless sensor networks for health, condition, and operational performance monitoring of long linear (infra)structures include:

- **Water pipeline monitoring:** One of the applications for wireless sensor networks is constantly monitoring and maintenance of large commercial pipelines which carry water [7, 8], to ensure structural integrity. For example, Saudi Arabia which is a global leader in water desalination heavily depends on over 4000 kilometer of pipeline in order to transport water from several desalination plants scattered throughout the country. Active monitoring and inspection is then required to maintain the pipeline health. Monitoring long-spanned water pipeline is a challenging task because of the difficulties in maintaining the system [9]. There has been increasing awareness and consolidated effort to use a robust and reliable technique to monitor leaks, bursts and other anomalies in the system. WSN-based pipeline monitoring can provide a remote facility to monitor pipeline status by (i) measuring inside (pressure, flow, temperature, and etc.) and outside (leakages) of the pipelines, and (ii) transferring the measurements collected by different sensor nodes being deployed at either randomly or critical points.
- **Bridge monitoring:** Between 1989 and 2000, more than 500 bridges had partially or totally collapsed in the United States due to events such as earthquake or vehicle collision, design and construction error, unreliable visual inspection, aging infrastructure, poor or lack of maintenance, and undetected structural deterioration (e.g. scour or fatigue) [10, 11]. The I-35W Mississippi River bridge located in Minneapolis, Minnesota, USA, is one example which failed on August 1, 2007, collapsed to the river and riverbanks beneath, killed 13 people and injured 145 (Figure 1.1) [12]. This suggests a need for effective, continuous monitoring systems so that problems can be identified at early stages and economic measures can be taken to avoid costly replacement and/or bridge failures. Therefore, there is a need for bridge health monitoring technologies and systems to enable continuous

## 1.1 Application domains of long linear wireless sensor networks

---

monitoring and real time data collection. Wireless sensor network technology can prevent and potentially reduce number of bridge collapses by enabling self-diagnostic structures, which can monitor and predict possible problems. Condition assessment or health monitoring of the bridge structure is usually accomplished through use of only four types of sensors mounted along the supporting structure of a bridge. These four sensors are (i) accelerometer to measure hanger tension and stiffening truss vibration, (ii) strain gauge to measure stiffening truss stress, (iii) thermometer to measure the main cable temperature, and (iv) wind gauge to measure wind load [2].



**Figure 1.1. Bridge over the Mississippi river in Minneapolis, Collapsed Aug 1, 2007 [12]**

- Highway/Urban monitoring: Nowadays, traffic jam and high number of accidents in urban and metropolitan areas become more and more stressful and lead to dramatic consequences on economy, human health, and environment. Statistics show that around six million accidents occur in United States every year [13]. Several factors such as vehicle mechanical problems, bad weather conditions, drivers' behaviors are considered as the main reason behind these accidents [14]. Wireless sensor networks can be utilized for traffic monitoring, urban surveillance, and road surface monitoring because of its large number, high-density and real-time communication features.

## 1.2. Network topology and data dissemination

Regardless of the application, the way that data and information are disseminated is an important aspect in wireless sensor networks. Before elaborating on the network topology and data dissemination mechanisms one should note that we consider a wireless sensor network which consists of following device types:

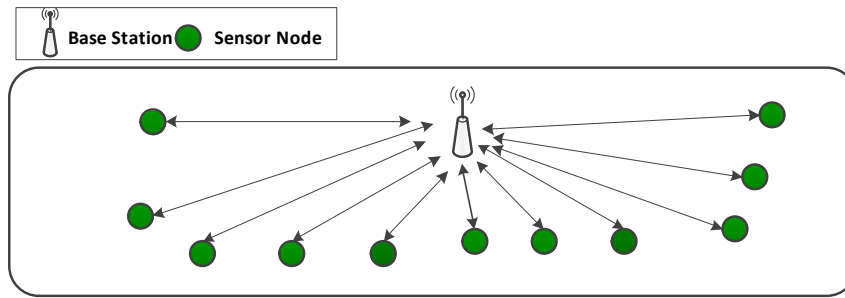
- (i) Sensor node: Depending on the role assigned to a sensor node, each sensor node can act as (a) a typical sensor node or (b) a cluster-head/chain-leader which is a gateway between other sensor nodes and the base station.
- (ii) Base station: It is the sink node for data collected by the sensor nodes in the network. It has enough resources and may be equipped with different interfaces or radio modules. For example a base station may utilizes IEEE802.15.4 to communicate with the sensor nodes or cluster-head/chain-leader and utilizes IEEE802.11g or even fiber to communicate with other base stations.

In general, according to the forwarding mechanism, wireless sensor networks can be classified as single-hop or multi-hop networks.

- Single-hop network: Figure 1.2 shows the topology of a single-hop network, also known as star topology. Start topology is the simplest topology in which every node can communicate directly with the base station. Since the failure of a single node/link does not influence the operation of the rest of the network, a star topology is considered robust. However, this topology is spatially limited by the transmission range of the nodes and thus is not scalable without a great deal of power supply which may help to increase transmission range of the nodes. In this topology, the battery of the nodes far away from the base station may quickly drain as transmission power usually increases as a power function of the distance between sender and receiver. Therefore, this topology is appropriate only for (i) small-scale applications whose coverage area does not extend beyond the transmission range of the nodes and (ii) applications in which both sensor nodes and the base station have enough power to transmit and receive data of nodes far away from the central points.

## 1.2 Network topology and data dissemination

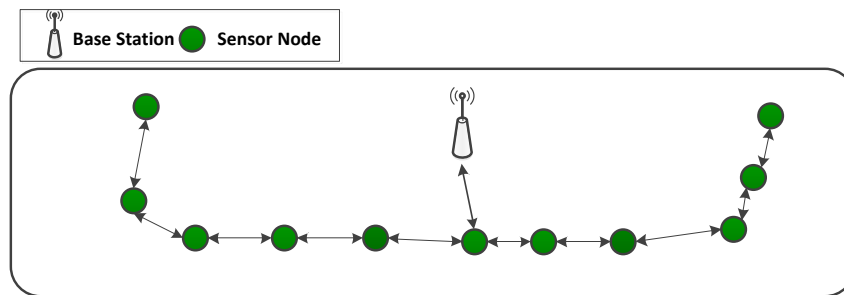
---



**Figure 1.2. Single-hop communication**

- **Multi-hop network:** The limited radio range of wireless sensor nodes makes single-hop packet transmission impractical especially for large-scale or long deployment of wireless sensor networks. In this way, as shown in Figure 1.3, a multi-hop data-forwarding mechanism which transfers data between the wireless sensor nodes and the base station by the help of intermediate nodes in a multi-hop fashion, is an attractive alternative for the communication. The multi-hop data-forwarding can further be classified into: (i) Mesh, (ii) Cluster-based, (iii) Tree-based, and (iv) Chain-based. Among these four categories, chain-based topology which is well-suited to the linear (infra)structures, is shown to perform best in terms of energy efficiency, lifetime and load balancing, in particular for large-scale network [15, 16, 17]. The mesh topology exhibits the least efficiency in terms of energy [15]. Chain-based topology minimizes many constraints that wireless sensor networks suffer from, for example, it can effectively balance the node's energy dissipation. In other words, energy distribution in a chain-oriented topology is even and thus it offers longer lifetime [17, 18, 19]. Moreover, due to logical structure of the sensor nodes in a chain-based topology, it offers substantial advantages for aggregating correlated data on their way to the base station [20, 21, 22, 23]. Finally, since the communication of a sensor node in a chain is restricted only to its neighboring nodes (i.e. successor and predecessor node), chain-based topology can gain the advantage of avoiding wireless communication problems like interference [15, 24]. Even though chain-based topology can be used because of its advantages in any deployment, sometimes the deployment itself enforces using the chain-based topology. For instance, in a linear network where the distance between sensor nodes are such long that each node can only communicate with the one-hop adjacent node, the network usually employs a chain-based topology.





**Figure 1.3. Multi-hop communication**

In a multi-hop wireless sensor network, intermediate nodes may participate in the aggregation process (if needed) and forward (aggregated) data packets from the source nodes toward the base station. Besides passively receiving data packets from sensor nodes, the base station may sometimes actively communicate with all wireless sensor nodes through intermediate nodes in order to send commands and queries. Identifying which set of intermediate sensor nodes should perform forwarding/aggregation task while the required quality of services of the application are satisfied, is the primary task of data disseminating algorithm.

Data disseminating protocols have typically two main phases:

- Initial phase: which is initiated by the base station by sending a query to the sensor nodes. This query includes task-related information.
- Data transmission phase: which indicates the way that data should be reported from the sensor nodes to the base station. This phase includes, but is not limited to, selecting unicast or broadcast mechanisms and data replication.

It is worth mentioning that the network architecture can be (i) single-tier in which all sensor nodes are programmed to perform all possible application tasks, hence, all nodes have identical roles or (ii) multi-tier in which network elements are organized into hierarchical levels according to their functionalities and capabilities. In a multi-tier architecture, from the low hierarchy tiers till top ones, typically an increase on the complexity of the tasks to be performed is expected. A multi-tier network architecture has shown to be more beneficial than a single-tier architecture in terms of energy consumption, reliability and scalability [25]. Regardless of the architectural model, any network topology can be utilized within a tier.

Network topology inherently defines the type of routing and data dissemination paths. Therefore, topology plays a vital role for resource-constraint sensor networks. In this regard,

### **1.3 The need of quality of service for data traffic**

---

choosing the right topology model before presenting any protocol helps to reduce the amount of communication and thus energy consumption needed for a particular problem.

There is not much work on chain-based data dissemination in wireless sensor networks and thus there are some areas to which special attention should be paid. Even though many routing and disseminating protocols have been designed for wireless sensor networks [26], most of them are usually designed for a general topology such as mesh which work well in a multi-dimensional deployment. For applications with linear topology, in which nodes are usually lined up in one-dimensional formation, however, a mesh topology may not be appropriate or simply not feasible due to the physical structure or measuring point distribution, among others. Moreover, it is a good idea to take the advantage of a linear topology over a predetermined linear infrastructure (e.g. bridge, tunnel, etc.), which may be quite different than a randomly deployed network. Therefore, by being aware about the underlying topology (linear in this thesis), more efficient routing and disseminating scheme can be designed.

Disseminating data in a large wireless sensor network is inherently a difficult problem whose solution must meet several challenging design requirements, including correctness, robustness, and optimality with respect to some performance metrics such as throughput, loss rate and delay. The natural properties of wireless sensor networks, combined with severe energy and bandwidth limitations, introduce additional challenges that should be addressed to provide the traffic requirements of the applications, while prolonging the network lifetime. This necessitated the need for considering quality of service.

### **1.3. The need of quality of service for data traffic**

One of the principal barriers which is required to be tackled in order to make wireless sensor networks more pervasive is the lack or low quality of service guarantees.

Wireless sensor network applications have a mixture of periodic and aperiodic traffic types depending on which different levels of quality of services are required for the data traffic. Basically, a number of wireless sensor network applications operate in an event based manner, i.e. nodes only send data or alarm if an abnormal or specific condition occurs. In contrast another class of applications periodically gather data from the sensor network and use this data for a particular purpose such as logging a phenomenon or creating models based on the collected data. However, providing acceptable quality of service for the traffics with the above characteristics is a challenging problem mainly because of time-varying uncertainties and specific nature of wireless sensor networks that include: (i) resource

constraints, (ii) dynamic topology caused by harsh environment, node mobility, node failure and addition, (iii) large scale and high density, (iii) unreliable nature of medium, and (iv) data redundancy. Considerable research has been done on various aspects of wireless sensor networks including protocol and architecture, routing, and power conservation. Quality of service support and guarantees in wireless sensor networks, however, especially for linear wireless sensor networks is an emerging area of research.

### 1.3.1 QoS in dense or sparse nodes deployment

Wireless sensor networks often rely on the collective effort of densely deployed sensor nodes which continuously monitor the phenomena in the environment of interest. The harsh and hostile environment in which the sensor nodes are often demanded to operate, necessitates the redundant or dense deployment of sensor nodes in the territory of the observation. Since employing a large number of sensor nodes reduce the vulnerability of the system to failure, this dense deployment has the potential to increase the reliability and dependability of the system. In a dense deployment, however, the ability to combine collected information becomes an important issue in managing bandwidth and facilitating final decision making. In a wireless sensor network, it is quite likely that sometimes data of some nodes is either completely lost or unreliably received by the base station. In this case, no or less contribution of these given nodes for that time instant will be recorded. However, thanks to spatial data correlation the data of that node is likely to be extrapolated by using other nodes' data.

In a sparse wireless sensor network where the density of the node is low, there is not that much spatial correlation among sensor nodes' data compared to a dense deployment. Therefore contrary to dense deployment the missing/erroneous data of sensor nodes in sparse deployment cannot easily be reconstructed. Due to this reason, in a sparse deployment the base station is usually interested to receive data of all individual nodes whose sensory data cannot be well-estimated by other nodes because of low spatial-correlation between sensor nodes' data.

Considering the scope above, the mechanism utilized to ensure quality of service for data traffic in a dense deployment should be different with the one utilized in a sparse deployment.

It is worth mentioning that one application may utilize both dense and sparse deployment; dense deployment in the critical area and sparse deployment on the uncritical and less-important area.

### **1.3 The need of quality of service for data traffic**

---

#### **1.3.2 QoS guarantees in a linear topology**

Depending on the type of target application, quality of service in wireless sensor networks can be characterized by long lifetime, reliability, timeliness, and coverage, among others. There are many other quality of service parameters worth mentioning, but these four are the most fundamental [6, 27, 28, 29, 30]. Parameters such as throughput, delay, and packet loss rate may be used to measure the degree of satisfaction of these quality of services. Although these quality of services are important, the priority among them can be demanded differently depending on the applications goal. Periodic data monitoring applications whose goal is to monitor and examine evolution of certain parameters require a long lifetime and could better tolerate missing data or delay than event detection applications. For example, structural health monitoring applications need the entire data from all measuring points to build a model and analyze it. In this way, the effectiveness of this type of the applications depends on how reliably the network can deliver the sensory data to the base station(s) and thus it is important to efficiently handle losses and ensure reliable data delivery between the sensor nodes and the base station(s) at a desired level.

Another example is mobile object tracking systems which have been used in many application domains such as urban/highway monitoring, among others. These applications need to meet certain real-time constraints in response to moving mobile objects. If the mobile object moves to a new position it should be reported to the command center so that immediate remedial actions can be taken. The position of the mobile object is however valid within a specific time interval so that timely pursuit actions can be initiated by the command center before the mobile object moves out of the sensing range. The valid time interval mainly depends on the target speed and some other environmental parameters. As the place of the given object is changing over time, if the packets which convey object's position reach the command center after the deadline, they may mislead the command center by their wrong and outdated information about the current position of the object. Therefore, it would be more effective to drop such kind of packets. In addition, if the alarm messages in case of having a fire reach the command center after the specified deadline, the fire may go out of control and so the consequence would be a catastrophic. On the other hand, some applications require to have commands or queries sent by the base station on all sensor nodes within a certain real-time constraints. If one sensor node receive the command or query message after the deadline, it starts reporting the information which may not be interesting for the command center anymore. By reporting such useless data, the bandwidth and energy of the relaying nodes are wasted.

### 1.3.2.1 Long lifetime

Energy consumption and thus energy efficiency has the highest priority in wireless sensor networks to ensure long network life time. In this regard, from the networking point of view, long lifetime can be considered as a quality of service parameter. Combining the duty-cycling and multi-hop transmission is one of the solutions to save more energy and to allow the wireless sensor networks operate for long time.

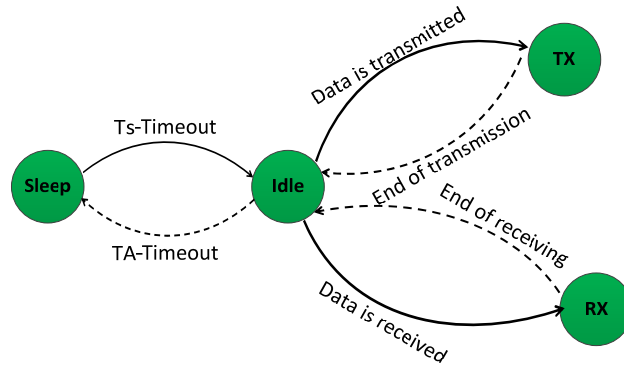
Since one of the most energy-expenditure operations is transmitting and receiving data, each sensor node often turns its radio off and goes to asleep state to obtain significant energy saving. Using duty-cycling, nodes can reduce the idle listening burden, which is typically reported as the biggest energy overhead in wireless sensor networks. A duty-cycle is the proportion of time during which the radio of a sensor node is ON [31]. In a duty-cycle-based power management scheme, each sensor node goes to sleep and wakes up periodically. The time during which each sensor node spends in sleep mode has direct impact on the data delivery delay, packet loss, and throughput. The shorter the duty-cycle, the lower the event detection probability and the longer the detection delay which will be even worse for a long chain-based topology. Therefore, duty-cycle is an important concern which should be carefully managed subject to the application mission. In a scheduling scheme, a sensor node is allowed to switch between four operation modes:

- Sleep state: In this state the radio of a node is turned off.
- Receiving state (RX): In this state the node can receive data from others.
- Transmitting state (TX): In this state the node can transmit data.
- Idle state: In this state the radio is ready to receive and/or transmit data. According to the conditions, the radio is changed to the appropriate active state either RX or TX.

Figure 1.4 presents the state diagram illustrating the main states of the radio and the ways state transitions occur. Once the sleeping time  $T_S$  is over, the radio must undergo a transition to idle state. Specifically, a dormant node transits to the active state when it is scheduled to switch to the active state. On the other hand, the radio of a node must be switched off as soon as the active time  $T_A$  is finished. It is worth noting that these four states have different levels of energy consumption, which differ from one radio model to another.

### 1.3 The need of quality of service for data traffic

---



**Figure 1.4. State diagram of radio states**

Obviously, a node cannot communicate with other nodes if it is not in the active state. Therefore, to deliver data within a specified deadline in a low duty-cycle network, the communication needs to be wisely managed among sensor nodes. In order to minimize the end-to-end delay raised from low duty-cycle, in this thesis we utilize a streamlined wake up schedule [32], which synchronizes duty-cycles of sensor nodes into a streamlined sequence that very well fits into chain-based topology. This idea is similar to turning traffic lights green right before the arrival of vehicles from previous intersections. This approach assigns sensor nodes a label according to the shortest hop count to the base station and then makes a path in such a way that each node is able to transmit the just received packet to the node which is located one hop closer to the base station.

#### 1.3.2.2 Reliability

Reliability and fault tolerance play a vital role in the success of event detection applications or periodic applications which require collecting all data without loss. However, ensuring data delivery is a crucial and challenging task in wireless sensor networks due to (i) node/network resource constraints (ii) failure-prone nature of cheap sensor nodes (iii) dynamic, harsh and hostile environment the sensor nodes are deployed in.

Reliable data dissemination is traditionally guaranteed by applying error control approaches, which could provide an adequate degree of quality even in presence of errors. There are two key strategies in wireless sensor networks for maintaining reliable communication over noisy channels: Forward Error Correction (FEC) [33] and Automatic Repeat Request (ARQ) [34]. Forward error correction or FEC relies on transmission of

redundant data in order to make the receiver node capable of reconstructing the original data. Automatic repeat request or ARQ relies on retransmitting a packet which has been missed or received erroneously. Although error control protocols can greatly increase reliability, however, this achievement usually comes at the expense of high energy consumption and long delay.

An error control protocol can either correct errors at (i) hop-by-hop level, in which the next hop is responsible to ensure the reliable transmission towards the destination (i.e. base station) or (ii) end-to-end level, in which only the end points (i.e. only the source node and destination node) are responsible for ensuring the successful transmission of information.

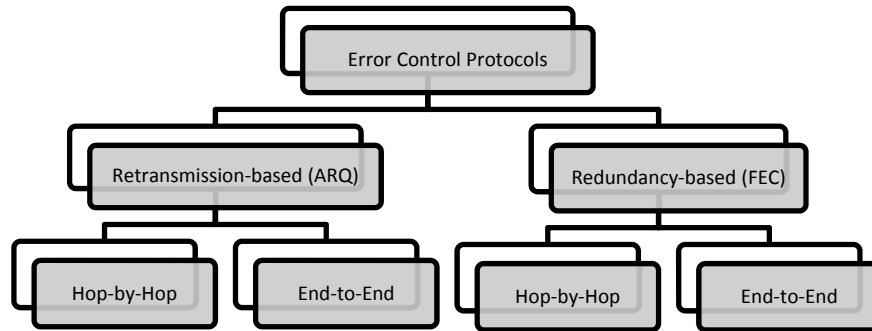


Figure 1.5. A classification of error control protocols

In an end-to-end retransmission, only source node should generate the lost packet to be retransmitted while in the hop-by-hop retransmission any intermediate node which encounters a packet loss should perform retransmission. Likewise, in use of end-to-end and hop-by-hop redundancy-based protocol, encoding and decoding procedures are performed in source/sink and each intermediate node, respectively. Since wireless sensor networks typically rely on the collective effort of several (intermediate) sensor nodes, conventional end-to-end reliability solutions are not always efficient for wireless sensor networks and would often lead to a waste of scarce sensor resources. Hence, the wireless sensor network paradigm often necessitates a collective hop-by-hop reliability notion rather than the end-to-end notion.

In a general way, the number of packets which are error-freely received by the destination, denoted by  $NPkt_R^{ErF}$ , can be obtained through Equation (1.1).

$$NPkt_R^{ErF} = P_D^{Suc} \times NPkt_S \quad (1.1)$$

### 1.3 The need of quality of service for data traffic

---

where  $NPkt_S$  represents the number of sent packets and  $P_D^{Suc}$  represents the probability of successful delivery.

The ultimate goal of reliable protocols is to increase the number of error-freely received packets in such a way that sensory data can be received/reconstructed in an energy-efficient way. One should note that in some applications, it is also important which packets with respect to some additional criteria, are received. These criteria include the packet-level constraints which can be the amount of information a packet carries or the time constraint that data packet exhibits, among others. According to Equation (1.1), to increase the number of error-freely received packets we should increase either the number of sent packets ( $NPkt_S$ ) or the probability of successful delivery ( $P_D^{Suc}$ ). Increasing the number of packets sent can be interpreted as adding redundancy to the information or retransmitting more packets. The side effect of increasing the number of packets sent is dissipating more energy. Increasing the probability of successful delivery or changing the loss distribution can mitigate the issues but cannot easily be tackled by the redundancy alone. For example, if  $P_D^{Suc}$  is not randomly distributed and the available erasure code [33] can correct up to  $r$  losses, erasure code is unable to reconstruct the original information if more than  $r$  packets are lost. In this situation, one possible solution is to select an alternative path with a higher  $P_D^{Suc}$ . Therefore, the knowledge of error nature and error distribution in wireless channels is an essential constituent for designing a reliable data dissemination protocol.

In order to further improve transmission reliability, multi-path data dissemination techniques can simultaneously be utilized along with the error control approaches. In general, each multi-path data dissemination protocol constructs multiple paths and distributes network traffic over the discovered paths. In this way, a multi-path retransmission-based error control approach is based on transmitting multiple copies of an original data over different paths to ensure recovery from several paths failures. Therefore, if data transmission over at least one path is performed successfully, data transmission reliability can be well-guaranteed. Moreover, in a redundancy-based error control approach in which some packets are added to the original data packets, multi-path technique can be utilized in order to transmit the original and redundant packets through different paths. To reconstruct original packets, a certain number of transmitted packets should be received by the destination. Therefore, if a few number of paths failed to transmit packets, the transmission reliability can still be guarantee through reconstructing the original data packets from the successfully received packets.

Although multipath routing approaches have been widely utilized for different network management purposes such as improving data transmission reliability and providing fault-



tolerant routing in wired and wireless sensor networks, the achieved performance gain is significantly affected by the ability of the utilized topology to construct an adequate number of high-quality paths. In a chain-based topology where the communication of a sensor node is often restricted only to its one-hop neighboring nodes (i.e. successor and predecessor node), we cannot well-benefit from the availability of alternative paths to salvage data packets from node/link failures. Moreover, in a chain-based topology when the distance between a sensor node with two-hop neighbors is higher than the transmission range of the nodes, there will not be any alternative path. Therefore, not all existing reliable data dissemination approaches proposed for the general topologies like mesh, can be applied to the chain-based topology.

### 1.3.2.3 Timeliness

An increasing number of wireless sensor network applications [35] (e.g. tracking mobile objects whose location information is valid only for a specific time interval in the highway monitoring applications or reporting accident in order to reroute the other cars' path) require to have the reported phenomena data in the destination within a specific end-to-end communication deadline and therefore impose a real-time bound on communication delay. Time-critical applications highly depend on the availability of real-time data as the data is not valuable if it is not received within the specific deadline. Therefore, if a packet does not reach the destination within the specific deadline, its contribution to the real-time capacity is zero. Outdated data is not only useless but may also be harmful as it may have negative impacts on the decisions made by the command center on the basis of invalid and stale information. Moreover, transmitting expired data depletes the bandwidth and energy of the relaying nodes inappropriately.

Applications can be divided into the following categories based on the notion of time they require and support:

- Time-unrestricted: These type of applications are not time-critical and have *no* dedicated deadline.
- Soft Real Time (SRT): In these type of applications, the usefulness of a packet received after its deadline decreases as it becomes stale, which in turn results in a graceful degradation of application performance. A common approach in these applications is to reduce the deadline miss ratio of the packets. Therefore, it would be more interesting for the SRT-based applications to increase the freshness degree

### 1.3 The need of quality of service for data traffic

---

of the reported data by finding a fast-enough transmission mechanism. The faster transmission or transmission-path approach, the fresher data at the destination.

- Firm Real Time (FRT): In these type of applications, the usefulness of a packet received after its deadline is *zero*. These applications can tolerate *infrequent* deadline misses.
- Hard Real Time (HRT): These type of applications highly rely on receipt of *all* packets before their deadline ends.

While the real-time performance is a major concern in all of the above mentioned applications except the first category, it should be compatible with other important performance measures such as reliability and energy consumption.

In a duty-cycle chain-based data dissemination protocol, sensory data might have to travel through a large number of hops (intermediate nodes) which may have a short period of activity (caused by low duty-cycle) only during which they can relay/transmit data packets. Therefore, the large number of hop counts and low duty-cycle lead to a long delay that may not be appealing for the event-driven applications. This is a challenge faced by majority of the existing real-time data dissemination protocols. In this way, although saving energy is the key primary in wireless sensor networks and is achieved using both chain topology and duty-cycle, shortsighted optimization for energy can lead to wireless sensor networks that cannot fulfill their tasks. Hence, energy saving must be balanced with the task related goals of the applications which may require reliable and real-time data dissemination.

#### 1.3.2.4 Coverage

Coverage is an important factor for the success of many monitoring and surveillance applications and thus can also be considered as a quality of service parameter in order to show how well a wireless sensor network can monitor physical regions [36]. The main objective of coverage is to guarantee that each physical region in the area of interest is within the sensing range of at least one sensor node. Moreover, depending on the type of the applications, some applications may require to have some special regions, called critical regions, in the sensing range of more than one sensor nodes while uncritical regions are sufficient to be usually monitored by only one node.

Providing a full coverage while minimizing the consumed energy has been an active area of research in wireless sensor networks. In this way, the coverage protocols usually aim to prolong network lifetime by distributing the sensor nodes into a number of sets each of

which can solely cover the whole monitored area. In this manner, mutually exclusive sets of sensor nodes are activated successively. This will bring about less spatial density for the active nodes compared with when all nodes are active. As a consequence of having a few nodes active at each time slot, interferences and contention at the MAC layer will be reduced, which in turn leads to prolonging network lifetime and increasing transmission reliability.

### 1.4. Research objectives

The main focus of this thesis is the design and development of solutions to guarantee combination of four important quality of services, i.e. coverage, long-lifetime, reliability and timeliness for chain-based topology data dissemination. In this regard, the main research question of this thesis is:

*How can coverage, long-lifetime, reliability and timeliness be ensured for disseminating different types of data traffic in a chain topology?*

We address this question at topology and error control levels. In both levels, we aim to retain the advantages of the chain-based topology and at the same time overcome the problem of the chain-based topology.

Quality of service in a wireless sensor network can be affected by several constraints out of which (i) the relative position of the node to the base station, chain-leader and other sensor nodes, (ii) the internal reliability state of the network, (iii) the internal reliability state of individual sensor nodes, and (iv) the nodes' available power, are the most important ones.

The aforementioned constraints can be greatly tackled by the means of topology control. In other words, an efficient topology helps wireless sensor networks minimizing different constraints. Each sensor node in a wireless sensor network can potentially change the network topology by (i) turning its radio state, (ii) adjusting its transmission power (range), (iii) selecting specific nodes to forward its message, and (iv) changing its role to be either a cluster-head/chain-leader or a typical sensor node. The goal of topology control is to build and maintain a network structure (or connectivity) that can best tackle the available constraints taking the required quality of service guarantees in combination. To this end, sensor nodes should be selected in such a way that the best shapes/boundaries, with respect to the required quality of service, for the chains and clusters are formed.

We address network topology control by investigating coverage problem and chain-cluster routing problem. More explicitly, we (i) select the most-proper sensor nodes

## 1.5 Thesis contributions

---

according to their contributions' quality to the coverage-related goal, (ii) create efficient clusters/chains and setting their boundaries, and (iii) select the most-suitable and promising node as the cluster-head or chain-leader, subject to the application goal.

Topology control alone is not sufficient to ensure quality of services for disseminating data of many applications whose packets may convey different types or amount of information. For example, end-to-end transmission reliability cannot efficiently be guaranteed without taking the (sensory) data constraints (importance) into account. To handle this issue, after building an efficient network structure with respect to the quality of service parameters, we give emphasize to the error control. More explicitly, we investigate the error control for two different packet types: (i) packets which carry time-constrained data (ii) packets which carry different information-value for the base station. By information-value we mean the amount of information or importance a packet may have for the base station.

### 1.5. Thesis contributions

In order to achieve thesis research objective, we provide the following contributions:

- ✓ **Trust-based probabilistic coverage:** Many wireless sensor network applications require different observation quality for different regions. The more sensitive and critical regions should be monitored in a more reliable way and their reported data need to be received with higher reliability. The underlying reliability of reported data could be achieved by multiple sensor nodes that monitor the same region at the same time. However, this reliability using redundant nodes could come at the expense of shorter network lifetime and so this redundancy is beneficial if the region is critical. In Chapter 2, we investigate the coverage problem based on probabilistic coverage concept and propose a trust-based probabilistic coverage algorithm, to increase reliability depending on the type of monitoring region. The proposed approach leverages the trust concept to tackle the time-varying uncertainties introduced by the sensor nodes and the environment they operate in, which may affect the quality of sensory data. In this way, we (i) explore and evaluate the aforementioned time-varying uncertainty parameters, (ii) formulate the coverage problem as an Integer Linear Programming (ILP), called TPC, based on the explored parameters, (iii) propose a greedy heuristic algorithm called TPC-Greedy to approximate the optimal solution. We consider wireless sensor networks where the transmission range is relatively large compared to the sensing range (e.g. a highly dense wireless sensor network area). In this regard, after finding the

coverage set which consists of the minimum number of sensor nodes needed for full sensory coverage of the regions, the resulting primary sensor nodes have indeed some neighbors within their transmission range to communicate with.

- ✓ **QoS-aware Cluster-head/Chain-leader Selection in a Two-tier Architectural model:** The problem we deal with in the first part of Chapter 3 is to find a well-balanced quality of service aware approach to deliver data packets collected by the sensor nodes (which are selected in previous contribution by TPC or TPC-greedy) to the base station, respecting application requirements. We address three quality of service parameters, i.e., (i) lifetime, (ii) reliability, (iii) delay or data freshness. In this regard, we propose QoS-ACA a reliable, fast, and energy-efficient data dissemination scheme to deliver data packets collected by the sensor nodes to a base station. In a chain-based topology, in which sensor nodes that are not leader can only communicate with their two adjacent left and right neighbors, routing is not very complicated. Therefore, we mostly concentrate on the chain leader election algorithm instead of routing. More specifically the contribution of the first part of this chapters is twofold, i.e., (i) introducing a two-tier architecture model in order to energy efficiently, reliably and fast aggregate and disseminate sensed data toward the base station, (ii) integrating the three quality of service parameters (lifetime, reliability, and delay) with the possibility to adjust their priorities according to the specific application requirements in order to find the most proper nodes as the chain leaders in both tiers. QoS-ACA, dependent on the network density, ensures reliability in two different ways for the sparsely and densely deployed sensor nodes. Moreover, in the interest of conserving both energy and bandwidth along with providing meaningful information to end-users, our protocols in this chapter utilize data aggregation on both chain-leaders or cluster-heads and intermediate nodes along the path toward the destination.
- ✓ **QoS-aware Dynamic Chain-Cluster Forming:** In order to relax some assumptions regarding communication capability of the sensor nodes to communicate directly with other nodes or with the base station as well as the fixed-size of the chain-clusters, which QoS-ACA (discussed in previous contribution) and many data dissemination protocols rely on, in the second part of Chapter 3 we propose REC and REC+ solutions which make the size/shape of the clusters in QoS-ACA adaptive regarding the state of the nodes and links. In this way, the main concern of REC/REC+ is building chain-clusters and setting boundaries of the clusters in an adaptive and dynamic way subject to the application level quality of service constrains.

## 1.6 Thesis organization

---

- ✓ **Reliable Dissemination of Time-Constrained Data:** The main contribution of Chapter 4 is ensuring reliable dissemination of time-constrained data in a low duty-cycle network suffering from short-term burst errors. In this respect, we propose READ a runtime adaptive packet-link-local error control protocol which operates based on the links' qualities, packets' Time-To-Live (TTL), and duty-cycle and is able to counteract periodic short-term burst-errors in a linear topology. The main idea of READ is fairly distributing the available TTL based on the links qualities among the sensor nodes who utilize the allocated times to enhance the transmission reliability.
- ✓ **Information-link-aware Data Dissemination:** In the same line of the previous contribution which considered the TTL as an indicating parameter to ensure quality of service, in Chapter 5 we concentrate on the information-value or the amount of information a packet carries as another packet-level indicator, considering which the reliability performance of a dissemination protocol can greatly be enhanced. To this end, in Chapter 5 we (i) explore, quantify and integrate the factors that may influence the information-value of a packet and (ii) cope with this crucial design problem of choosing an appropriate error control code by adaptively selecting the codes for each *individual links*, which may experience long-term fading and for each *individual packet* at run-time instead of applying network-wide settings prior to network deployment. In this way, we propose RAFEC protocol, which is a Runtime Adaptive FEC-based data dissemination protocol. In RAFEC each node decides which error control code to use abiding to the computational constraints of the embedded sensors, the information-value of the packet, and the statistical properties of the observed errors for the upward link. This adaptation gives the possibility to vary the code strength and complexity on-demand and on the fly.

### 1.6. Thesis organization

The organization of this thesis and relationship among different chapters are illustrated in Figure 1.6. In Chapter 2 we investigate the coverage problem in wireless sensor networks and propose TPC a trust-based probabilistic coverage algorithm. After finding the coverage set in Chapter 2, Chapter 3 builds some chain-clusters among active sensor nodes subject to the application level quality of service constrains for data traffic, either in a static way through QoS-ACA or in a dynamic fashion through REC and REC+. Importantly, Chapter 2 and 3 both give emphasize to the topology. Later in Chapter 4 and 5, we further concentrate on the error control approaches. In this regard, in Chapter 4 we propose READ, i.e., a

reliable and energy efficient data dissemination protocols while adhering to the packet-level constraint TTL. In Chapter 5, we propose RAFEC, i.e., a run-time adaptive FEC-based data dissemination protocol which considers the amount of information a packet carries as another packet-level constraint in order to ensure quality of service for data traffic. Finally, in Chapter 6 we draw the conclusion of the thesis along with some future works.

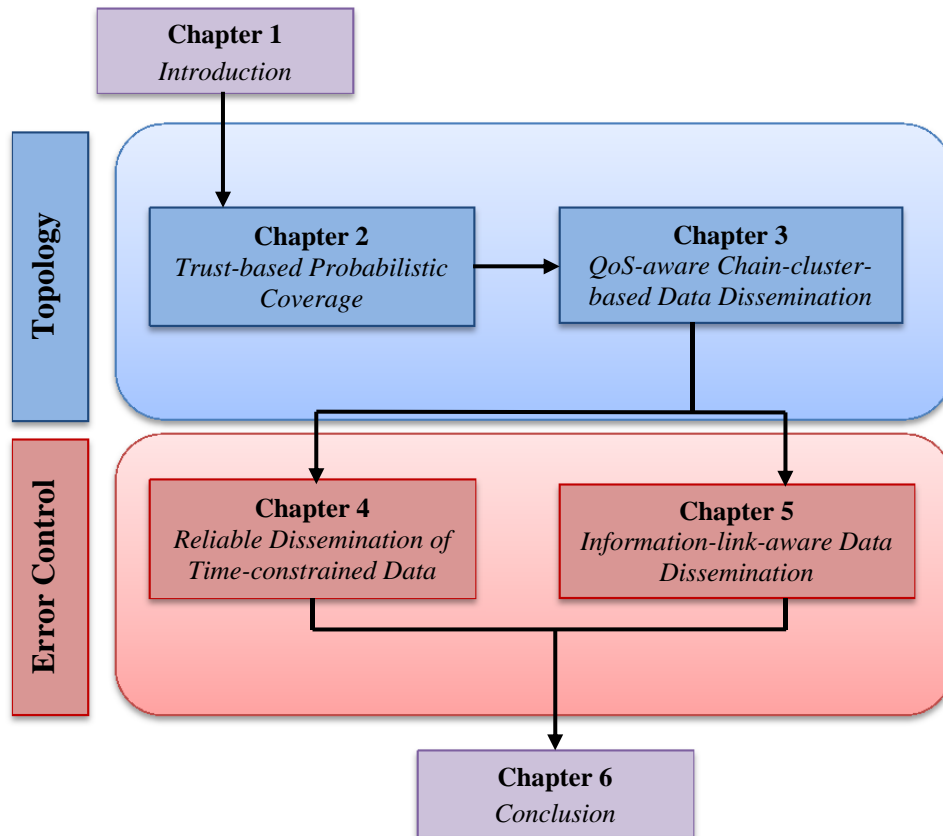


Figure 1.6. Organization of thesis

### 1.7. Bibliography

- [1]. Sohraby, K., D. Minoli, and T. Znati, *Wireless sensor networks: technology, protocols, and applications*. 2007: John Wiley & Sons.
- [2]. Chae, M.J., et al., *Development of a wireless sensor network system for suspension bridge health monitoring*. Automation in Construction, 2012. 21: p. 237-252.
- [3]. Bischoff, R., et al. *Event-based strain monitoring on a railway bridge with a wireless sensor network*. in *4th International Conference on Structural Health Monitoring of Intelligent Infrastructure (SHMII-4)*. 2009.
- [4]. Bennett, P.J., et al., *Wireless sensor network for monitoring transport tunnels*. Proceedings of the ICE-Geotechnical Engineering, 2010. 163(3): p. 147-156.
- [5]. Carrillo, A., et al., *New distributed optical sensor for detection and localization of liquid leaks: Part i. experimental studies*. Sensors and Actuators A: Physical, 2002. 99(3): p. 229-235.
- [6]. Taghikhaki, Z., et al., *QoS-Aware Chain-Based Data Aggregation in Cooperating Vehicular Communication Networks and Wireless Sensor Networks*. 2012.
- [7]. Jawhar, I. and N. Mohamed. *A hierarchical and topological classification of linear sensor networks*. in *Wireless Telecommunications Symposium*. 2009. Piscataway, NJ, USA.
- [8]. Stoianov, I., et al. *PIPENET: A wireless sensor network for pipeline monitoring*. in *6th International Symposium on Information Processing in Sensor Networks*. 2007.
- [9]. Jawhar, I., N. Mohamed, and D.P. Agrawal, *Linear wireless sensor networks: Classification and applications*. Journal of Network and Computer Applications, 2011. 34(5): p. 1671-1682.
- [10]. Spencer Jr, B.F. and S. Cho. *Wireless smart sensor technology for monitoring civil infrastructure: technological developments and full-scale applications*. in *World Congress on Advances in Structural Engineering and Mechanics (ASEM'11)*. 2011.
- [11]. Wardhana, K. and F.C. Hadipriono, *Analysis of recent bridge failures in the United States*. Journal of Performance of Constructed Facilities, 2003. 17(3): p. 144-150.
- [12]. [http://en.wikibooks.org/wiki/Professionalism/URS\\_and\\_the\\_I-35\\_Bridge](http://en.wikibooks.org/wiki/Professionalism/URS_and_the_I-35_Bridge).
- [13]. Gay, D., *American Highway Roulette* 2010: Denton Gay.
- [14]. Zohdy, I. and H. Rakha. *Optimizing driverless vehicles at intersections*. in *10th ITS World Congress*. 2012.
- [15]. Mamun, Q., *A qualitative comparison of different logical topologies for wireless sensor networks*. Sensors, 2012. 12(11): p. 14887-14913.



- [16]. Lindsey, S., C. Raghavendra, and K.M. Sivalingam, *Data gathering algorithms in sensor networks using energy metrics*. Transactions on Parallel and Distributed Systems, 2002. 13(9): p. 924-935.
- [17]. Mamun, Q., *Designing Logical Topology for Wireless Sensor networks: A multi-chain oriented approach*. International Journal of Ad Hoc, Sensor & Ubiquitous Computing, 2013. 4(1).
- [18]. Pham, M.-L., et al., *Power aware chain routing protocol for data gathering in sensor networks*. International Journal of Distributed Sensor Networks, 2005. 1(2): p. 253-267.
- [19]. Shin, J. and C. Suh. *Energy-efficient chain topology in ubiquitous sensor network*. in *10th International Conference on Advanced Communication Technology*. 2008.
- [20]. Cheng, Z., M. Perillo, and W.B. Heinzelman, *General network lifetime and cost models for evaluating sensor network deployment strategies*. Mobile Computing, IEEE Transactions on, 2008. 7(4): p. 484-497.
- [21]. Pandana, C., *Resource and Environment Aware Sensor Communications: Framework, Optimization, and Applications*, 2005.
- [22]. Luo, H., et al., *Data fusion with desired reliability in wireless sensor networks*. Parallel and Distributed Systems, IEEE Transactions on, 2011. 22(3): p. 501-513.
- [23]. Wu, K., Y.-m. Ding, and Z. Zhong. *A chain-based fast data aggregation algorithm based on suppositional cells for wireless sensor networks*. in *2nd International Conference on Power Electronics and Intelligent Transportation System* 2009.
- [24]. Mamun, Q., S. Ramakrishnan, and B. Srinivasan. *An efficient localized chain construction scheme for chain oriented wireless sensor networks*. in *10th International Symposium on Autonomous Decentralized Systems (ISADS)*. 2011.
- [25]. Kulkarni, P., et al. *SensEye: a multi-tier camera sensor network*. in *13th annual ACM international conference on Multimedia*. 2005.
- [26]. Al-Karaki, J.N. and A.E. Kamal, *Routing techniques in wireless sensor networks: a survey*. Wireless communications, IEEE, 2004. 11(6): p. 6-28.
- [27]. Bouyssounouse, B. and J. Sifakis, *Embedded Systems Design: The ARTIST Roadmap for Research and Development* 2005: Springer-Verlag New York, Inc.
- [28]. Chen, D. and P.K. Varshney. *QoS Support in Wireless Sensor Networks: A Survey*. in *International Conference on Wireless Networks*. 2004.
- [29]. El-Gendy, M.A., A. Bose, and K.G. Shin, *Evolution of the Internet QoS and support for soft real-time applications*. Proceedings of the IEEE, 2003. 91(7): p. 1086-1104.
- [30]. Li, Y., et al. *Real-time QoS support in wireless sensor networks: a survey*. in *7th IFAC Int Conf on Fieldbuses & Networks in Industrial & Embedded Systems (FeT'07)*. 2007. Toulouse, France.

## 1.7 Bibliography

---

- [31]. Anastasi, G., M. Conti, and M. Di Francesco, *Extending the lifetime of wireless sensor networks through adaptive sleep*. Industrial Informatics, IEEE Transactions on, 2009. 5(3): p. 351-365.
- [32]. Cao, Q., et al. *Towards optimal sleep scheduling in sensor networks for rare-event detection*. in *4th international symposium on Information processing in sensor networks*. 2005.
- [33]. Halsall, F. and D. Links, *Computer Networks and Open Systems*. Addison-Wesley Publishers, 1995: p. 112-125.
- [34]. Schwartz, M., *Telecommunication networks: protocols, modeling and analysis*. Vol. 7. 1987: Addison-Wesley Reading, MA.
- [35]. Papp, Z., J. Sijs, and M. Lagioia. *Sensor network for real-time vehicle tracking on road networks*. in *5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. 2009.
- [36]. Meguerdichian, S., et al. *Exposure in wireless ad-hoc sensor networks*. in *7th annual international conference on Mobile computing and networking*. 2001.



# Trust-based Probabilistic Coverage<sup>1</sup>

---

Coverage is a fundamental issue in WSNs for environmental monitoring and surveillance purpose. In general, *coverage problems* aim to either deploy nodes to cover the sensing field completely or select active nodes in a densely deployed WSN to cover all the sensing field. The last case is known as an *activity scheduling problem* and many coverage schemes propose to organize nodes into a number of coverage (or active) sets the members of each of which are able to cover the whole area and only one set should be active at any moment of time. This way of allocating nodes prolongs lifetime by avoiding nodes monitor the area which is already covered by other node(s). Although saving energy is an important concern in WSN, shortsighted optimization for energy can lead to a system that cannot fulfill the task-related goal. Most of the study in coverage domain rely on the assumption that if nodes have enough energy, they can cover the monitoring area. However, there is inherent uncertainty for each node, which may prevent them to function well and so the failure of one node may cause the failure of the whole system. Due to nodes' resource limitations, inherent uncertainties associated with their measurements, and the harsh and dynamic environment in which they are deployed, achieving a reliable and energy-efficient coverage scheme is very challenging. In this chapter we propose a reliable coverage scheme which builds the coverage sets for a network whose nodes and links can fail independently during the normal operation. Moreover, most of available coverage schemes are static and find the coverage sets only once at network deployment phase and inform nodes about their activity scheduling for their whole lifetime. However, due to dynamic and time-varying nature of WSN, having a static coverage scheme is inefficient. Therefore to address

---

<sup>1</sup> This chapter is based on the following publication:  
*A trust-based probabilistic coverage algorithm for wireless sensor*. In Proceedings of The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2013).

above considerations at the design stage, a coverage set is needed to be built upon request while taking the most current system situation into account. Specifically the main contributions of this chapter are exploring time-varying uncertainty parameters that may affect the quality of sensed data and is required to build a situation-aware scheme, quantifying those parameters using trust concepts, formulating the coverage problem based on those parameters as an Integer Linear Programming, and proposing a greedy heuristic algorithm to approximate the optimal solution. The proposed coverage scheme can be applied for any kind of deployment including linear. The simulation results prove the superiority of our proposed approaches in terms of reliability and energy-efficiency for covering the critical regions.

### 2.1. Introduction

As energy is a very challenging issue in wireless sensor networks, various techniques have been utilized over years to minimize energy consumption in wireless sensor networks. These techniques include, but are not limited to, (i) scheduling the sensor nodes to alternate between active and dormant modes specially in a dense deployment, (ii) adjusting the transmission/sensing range of the sensor nodes, and (iii) designing energy-efficient processing protocols. Due to the fact that communication has the highest share in energy consumption, as pointed out for example in [1], the best method to save more energy is to turn off the radio of as many sensor nodes as possible and for as long as possible while the system still functions well. Therefore, determining a schedule based on which sensor nodes are kept active to efficiently cover the whole monitoring area is an important challenge. By efficient coverage of monitoring area we mean ensuring long network lifetime as well as maintaining sufficient sensing coverage and reliable sensing. On the other hand, the resource constraints of sensor nodes which are usually asked to operate in a harsh and hostile environment, necessitates the dense deployment of sensor nodes in the territory of observation to improve lifetime and reliability. This dense deployment can facilitate saving nodes' energy by using a time-varying subset of the nodes for fulfilling reliable sensing coverage and connectivity while other redundant nodes are in dormant state.

The activity scheduling problem [2, 3] whose goal is to prolong network lifetime on the premise of preserving the sensing coverage, aims to allocate sensor nodes into a number of sets each of which could solely cover the whole monitored area. In this manner, mutually exclusive sets of sensor nodes are activated successively. This will bring about less spatial density for the active nodes compared with when all nodes are

## 2.2. Problem statement and our contributions

---

active. As a consequence of having a few nodes active in each time slot, interferences and contention at the MAC layer will be reduced, which in turn leads to prolonging network lifetime and increasing transmission reliability. Coverage can also be considered as a QoS parameter for wireless sensor networks in order to show how well a network can monitor the critical regions [4].

The problem of guarantying the coverage while meeting application requirements is usually formulated as a combinatorial optimization problem typically solved using Integer Linear Programming (ILP) [5, 6]. There are many studies on the coverage problem [7, 8, 9, 10, 11, 12, 13, 14, 15], whose main concern is energy conservation while less attention has been paid so far to the reliability support within these algorithms. Wireless sensor nodes are highly vulnerable and may malfunction or fail. They may generate reports regardless of the actual sensor reading or they encounter faults (such as dropping a packet) due to many reasons such as congestion, buffer overflow, nodes movement, nodes sleep and a sheer failure. Non-malicious behaviors- such as the malfunctioning of radios/sensors or even imperfect environments- can result in generation of false data which could adversely impact the correctness of the overall sensor network finding and thereby results in detrimental effects on the performance of the network in terms of reliability and lifetime. Therefore, reliability should also be considered in addition to lifetime, when designing a coverage algorithm in order to minimize the impact of faulty measurements. To evaluate the confidence level and trustworthiness of the sensor nodes, we exploit trust and reputation concepts. One should note that we use the terms reliability and confidence level interchangeably throughout this chapter.

In this chapter, we explore and address the coverage problem, using a probabilistic coverage model, which aims to capture the real world sensing and transmitting characteristics of the nodes. The rest of this chapter is organized as follows. First we describe the problem statement and our contributions and then briefly discuss the state of the art. Thereafter, we present the assumption and models used. Then, a detailed description of our proposed approaches will be provided, which will be followed by performance evaluation. Finally, we will end the chapter with summary.

## 2.2. Problem statement and our contributions

We address the probabilistic coverage problem while adhering to the application requirement exploiting the trust model. Many wireless sensor network applications require different observation quality for different regions. The more sensitive and

critical regions should be monitored in a more reliable way and their reported data need to have higher reliability. The underlying reliability of reported data could be achieved by multiple sensor nodes that monitor the same region at the same time. However, this reliability using redundant nodes could come at the expense of shorter network lifetime.

Given an already deployed wireless sensor network with some known critical and uncritical regions, the problem at hand is to derive a feasible set ( $FsbSet$ ) which comprises of the minimum number of sensor nodes using which:

- (i) Every critical and uncritical region is monitored.
- (ii) The quality of data gathered from each critical/uncritical region meets the application requirement for the respective region.

Given the aforementioned problem, our contributions related to the coverage problem are:

1. Investigating, modeling and evaluating the parameters, which may affect the quality of sensing data.
2. Formulating a situation-aware trust-based probabilistic coverage scheme into an Integer Linear Programming (ILP), which is aware of the network dynamism. Specifically, our scheme is to find such active sensor nodes set which is able to fully and reliably cover the area while prolongs the network lifetime.
3. Proposing a greedy heuristic scheme to achieve almost the same result of ILP without suffering from computational complexity that ILP usually offers to solve large-size instances.

### 2.3. Related work

In the field of wireless sensor networks, some of existing studies [16, 17, 18, 19, 20, 21, 22, 23, 24, 25] aim at maximizing coverage utilizing mobility of sensor nodes. Mobile sensor nodes could be employed to fill in coverage holes. The approaches, which target improving coverage by getting help from mobile nodes, could be classified into three categories: (i) virtual force, (ii) coverage pattern, and (iii) grid quorum based movement [19]. The approach presented in [19] assumes more powerful cluster heads which runs an algorithm called virtual force to identify whether two sensor node are too close to each other. If so, cluster head asks them to push each other away, using a virtual negative force, as far as the optimal distance is obtained.

### 2.3. Related work

---

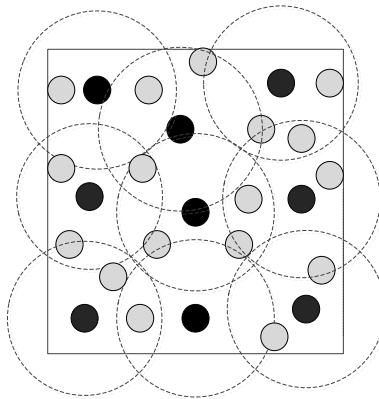
However, this approach suffers from single point of failure related to the cluster heads. Although mobile nodes could significantly improve coverage, using them is constrained due to practical reasons. Mobile nodes are usually more expensive than immobile sensor nodes and in comparison with communication or sensing task, mobility expenditures extensive energy. In this sense, mobile nodes are allowed to traverse short distance to not deplete the whole energy of the node. The work presented in [9] is a computational geometry based approach, while [12, 26] and [13] are potential field based and incremental deployment scheme based approaches, respectively. In [14], authors studied the performance of the network with  $n$  mobile nodes which move randomly over the unit area. They showed that node mobility increases capacity of the network. Although mobility of sensors may increase capacity of the network, these protocols have the same assumption that all nodes should be able to re-adjust their positions in the region. Different from these approaches, we consider static sensor nodes that do not have the privilege to move after deployment.

According to the requirement of sensing task, the coverage problem can be formulated as an area coverage or target/critical point coverage.

In the area coverage problem, all points in the deployment area should be monitored by at least one sensor node. The objective here is to build a sensing backbone by determining a subset of sensor nodes using which every point in the deployment area could be covered (Figure 2.1). In Figure 2.1, the square is fully covered if all intersection points are covered. The black nodes in Figure 2.1, represent the active nodes according to the scheduling algorithm. There are many solutions [1, 12, 13, 14, 15, 26, 27, 28, 29, 30] in area coverage whose objective is to cover the given area in an energy efficient way.

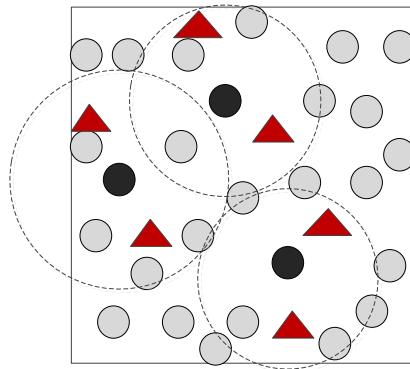
Area coverage has been proposed for many applications such as habitat monitoring, underwater surveillance, volcano monitoring network, forest fire detection, and birds habit monitoring [31, 32, 33, 34, 35, 36].





**Figure 2.1. An example of area coverage problem**

Instead, in the target/critical point coverage problem, each target/critical point which are shown by triangle in Figure 2.2 needs to be monitored by at least one sensor node. The objective here is to select a subset of sensor nodes using which all critical points could be covered. Critical point coverage problem is a special case of area coverage problem. If critical points form a contiguous region, the critical point coverage problem is turned into the area coverage problem. In other words, an area coverage problem could be approximated by a critical point coverage problem using a minute grid. The level of approximation could be increased with finer grid. In this sense, a solution for the point coverage problem could be extended to that of area coverage problem. The target coverage problem has been studied extensively and many solutions [7, 8, 9, 10, 11] have been proposed.



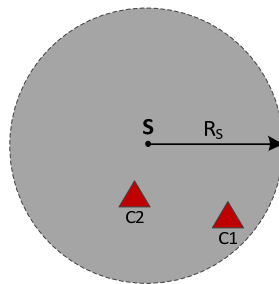
**Figure 2.2. An example of critical point coverage problem**

### 2.3. Related work

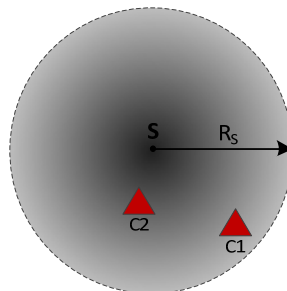
---

The coverage computational models used so far can be classified into two categories: (i) binary and (ii) probabilistic.

In the binary coverage model, a sensor node is usually assumed to be uniform in all directions and is represented by unit disc. This implies that in case of having a critical region coverage problem, a node can monitor a critical region with the highest confidence level if the critical region is located within the sensing radii of the given node. This binary model, which relies on having a perfect environment and quite reliable node, has been used in a number of research works [5, 6, 11, 14, 37, 38]. This model is built based on an unrealistic and strong assumption of perfect coverage in a circular disc for all the nodes. The practical implication of this model is that in Figure 2.3 the quality of reported data by node S for critical point  $C_1$  and  $C_2$  are similar as both of these critical points are within the sensing range of the given node S.



**Figure 2.3. Binary coverage model**



**Figure 2.4. Probabilistic coverage model**

However, in real deployments, it is possible that a critical region cannot be monitored by a sensor node even if it is located within the sensing range of the given node. This non-deterministic behavior is introduced by the uncertainties associated

with the (i) sensor node (such as quality of the sensors), (ii) environmental parameters such as obstacles (which may block or limit the sensing quality), background noise: magnetic field of the earth, weather: temperature, humidity, rainy, windy [29], and (iii) the deployment position.

As it is vital to address practical considerations at the design stage in order to well monitor the sensing behavior, coverage model should be assumed to be probabilistic as suggested in [29, 39, 40, 41, 42, 43, 44]. A direct implication of applying probabilistic model, as shown in Figure 2.4, is that if the distance between a sensor node and a critical region is considered as the only parameter which influences sensing quality, the quality of reported data by node  $S$  for  $C_1$  is less than that for  $C_2$ . In Figure 2.4, coverage is shown as a gradient so that  $C_2$  which is closer to node  $S$  has better coverage compared to  $C_1$  which is further away.

Another line of work on the coverage studies [7, 8, 10] aims to extend lifetime by allocating the sensor nodes into disjoint sets such that every set can independently cover all targets. Then, these disjoint sets are activated successively such that at any moment in time only one set is active. As all targets are monitored by each sensor set, the goal is to determine a maximum number of disjoint sets, so that the time interval between two activations for any given sensor becomes longer.

### 2.3.1 Going beyond the existing solutions

Most of the existing coverage solutions are based on the binary coverage model that are often not realistic. Contrary to these approaches, we use the probabilistic coverage model, because in reality the detection of a target is not deterministic. Although, some studies [1, 29, 39, 40, 41, 42, 43, 44] address the probabilistic coverage model, they mostly focus on the distance between sensor nodes and the critical point/target as the only parameter, which affects sensing quality. Different from these approaches we make our probabilistic coverage more general and consider more parameters, which may affect sensing quality.

Different from the coverage schemes [7, 8, 10] that define all feasible Coverage Sets (CS) at the beginning, we opt to find each set whenever the status of the nodes of the current feasible set changes in terms of energy level and reliability. This is especially beneficial to cope with the dynamicity of the network, where the confidence coefficient of the fault-prone nodes may change during their lifetimes, discovering all such disjoint sets at the beginning is useless. As most of the noises and errors in the

## 2.4. Assumptions and models used

---

network are transient and last for a short while, selecting each node at the right moment when it does not suffer from any fault or imperfect area could bring about better performance in terms of reliability and energy. Selecting one node as an active node irrespective of the real situation it is in, can result in generation of false data which could have detrimental effects on the overall performance of the network. Therefore, before calculating another feasible set whose members should be activated for the next time interval, we calculate and consider the most recent confidence level of the nodes. Doing so may bring quite different coverage set than that of other approaches suggest.

Most of the previous coverage schemes give emphasize to the energy consumptions of the sensor nodes in order to prolong network lifetime. In this chapter, we approach the coverage problem from a different perspective by targeting the confidence level of the sensor nodes, which may fundamentally influence the system performance in terms of reliability and energy efficiency. Erroneous data generated by the sensor nodes must be protected from entering the network for effective bandwidth usage and energy utilization. To this end and different from others, we leverage the trust model in order to quantify the reliability and trustworthiness of each node. The trustworthiness of the nodes could be exploited as a promising means to decide including a node in the coverage set or not, based on the confidence level the given node exhibits. It is worth recalling that for a successful operation, a network is required to provide the sensing coverage which meets the application-specific reliability requirement. Therefore, controlling and enhancing the reliability of the coverage problem in wireless sensor networks is quite worthy.

## 2.4. Assumptions and models used

### 2.4.1 Assumptions

We make the following assumptions regarding the wireless sensor network:

- The wireless sensor network consists of  $n$  nodes uniformly, randomly and redundantly deployed over a region which could represent a linear deployment or other kinds of deployments. In order to better explain the type of scenario considered in this chapter we refer to Figure 2.5. The area is divided into plurality of regions of equal size according to the geographic location information, and some nodes are placed uniformly at random locations within each region. The location of these regions is adjacent to each other but non-overlapping.

- The coverage area of each sensor is modeled as a circle whose center represents the sensor while the radius  $R_S$  represents the sensing range of the sensor.
- At each time instant each region is required to be partially covered by at least one sensor node. By partially coverage we mean only a part of the region, which should include the center of the given region is required to be covered. The number of active nodes in each region is in direct relation with the required confidence level for the reported data of the respective region.
- According to Figure 2.5, There are two types of regions: critical region and uncritical region. The critical regions are those regions, which have critical points inside and should be monitored with a higher reliability that is specified by the application.

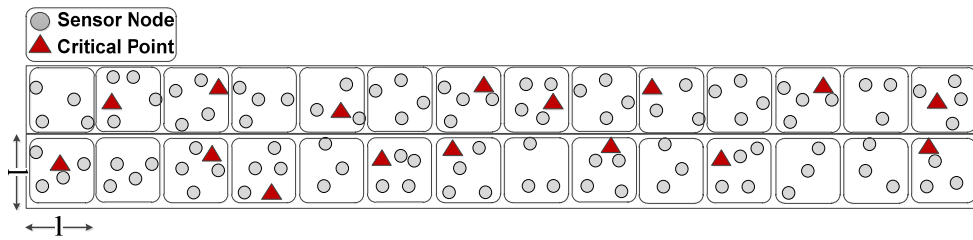


Figure 2.5. An example of nodes deployment

- The transmission range ( $R_T$ ) of sensor node is variable and also relatively large compared to the sensing range. For instance for the scenario depicted in Figure 2.5, the assumption  $R_T = \sqrt{5}l$  is both necessary and sufficient to ensure that coverage of the given regions implies connectivity among the active nodes in an arbitrary network. This transmission range implies that, the sensor nodes of two adjacent regions, regardless of the place the sensor nodes are located, can communicate with each other. From another perspective, assuming the transmission range  $R_T$  and the height of the deployment area  $H$  (in this figure  $H = 2l$ ), the length of each region should be calculated as Equation (2.1) in order to ensure any nodes of two adjacent regions can communicate directly with each other.

$$l = \sqrt{(R_T)^2 - H^2} \quad (2.1)$$

- The sensing range ( $R_S$ ) of sensor node is  $\frac{l}{\sqrt{2}}$ . This assumption is both necessary and sufficient to ensure that the center of each region is covered by the nodes in a given region.

## 2.4. Assumptions and models used

---

- Node sensing and processing quality as well as link quality vary over time. There are some obstacles in the environment, each of which either influences the sensing quality of a number of sensor nodes for a specific critical point or affects the transmission quality between pair of nodes.
- Sensor nodes send their measurements to a base station for central processing, directly or through intermediate nodes.
- Sensor nodes have the same initial energy.

### 2.4.2 Models used

#### 2.4.2.1 Node and link uncertainty models

Each sensor node may suffer from node or area imperfections, which can affect the quality of information received by the base station. The real world characteristics of the nodes and environment can be captured by some models.

We utilize a Three-state Markov model, illustrated in Figure 2.1, in order to model and use in the simulations nodes' confidence level. This model transits among "Good", "Bad" and "Failed" states according to the independent state transition probabilities  $p_m$ ,  $q_m$ ,  $t_m$  and  $s_m$ . State G (Good) implies that the sensor node works properly and is in a good (reliable) condition. State F (Failed) represents the failed state, in which the node stops working due to component failure or energy depletion. In state B (Bad), the node is subject to misbehavior or malfunction (e.g. packet drop, packet modification, erroneous sensing) and hence, node temporarily fails to provide useful data. The probabilities of remaining in the same state, namely, in the good and bad state, are  $1-p_m-s_m$  and  $1-q_m-t_m$ , respectively. Depending on the Node Failure Rate (NFR), these four parameters  $p_m$ ,  $q_m$ ,  $s_m$ ,  $t_m$  need to be changed accordingly. The amount of time staying in each state could be derived based on  $L_b = \frac{1}{q_m}$  and  $L_g = \frac{1}{p_m}$  for bad (which also represents the average error length) and good state, respectively.

Basically, NFR defines how likely the sensor node will report an incorrect value, caused by either an obstacle or the node malfunction like as imperfection sensor node's units. Therefore, we need to distinguish these two different situations as they may bring different consequences. In this sense, according to Figure 2.7, if two critical points are within the sensing range of a node and an obstacle is located between the node and one of the critical points (i.e.  $C_1$ ), the sensing quality of this

node should not be influenced for another critical point (i.e.  $C_2$ ). To address this issue in the node failure model, we should know the reason (obstacle or node malfunction) due to which the node transits to the state B. If this transition is because of an obstacle, which obstructed the node's sensing field, as shown in Figure 2.7, the reported data of this node may not be representative of C1 situation while at the same time could be representative of C2 situation. Instead, if the reason to transit to state B is because of node malfunction and not an obstacle, the reported data of this node for both C1 and C2 would be unreliable. This matter can be implemented by picking a random number from interval  $[0,1]$  upon transiting to state B, and deciding whether it should represent an obstacle or a node malfunction. If the random number falls into  $[0, t_o]$ , the reason of the erroneous data would be obstacle and we should simulate obstacle effects, otherwise if the random number falls in  $(t_o, 1]$ , we need to simulate node misbehavior. Assigning different values to  $t_o$ , the occurrence frequency of having obstacle or node malfunction could be taken in control.

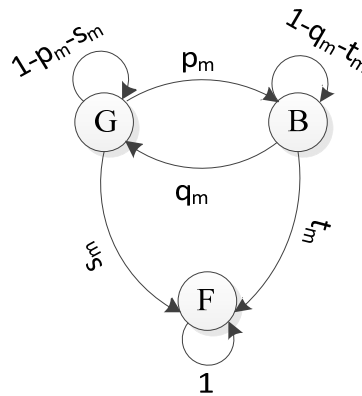


Figure 2.6. Three-state Markov model to model node's confidence level

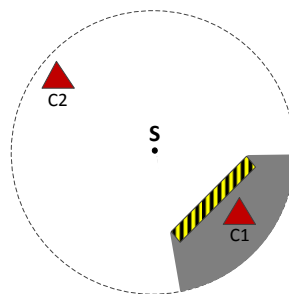


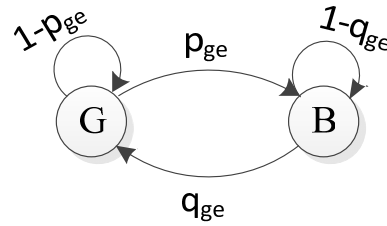
Figure 2.7. Effect of obstacle on sensor coverage

## 2.4. Assumptions and models used

---

As mentioned above, the error of the obstacle is already integrated to the error of nodes, since the error at a single node should not be double counted.

Apart from node failure/malfunction, the area or network conditions may introduce some challenges for quality of data received by the base station. Therefore, it is possible that the sensed data of one node is changed during the packet transmission or even is not received by the base station because of hostile/harsh area or channel imperfections. To avoid this situation, selecting active nodes at the locations nearby those areas experiencing favorable conditions are more effective. We utilize the Gilbert–Elliott model [45] with some probabilities ( $p_{ge}$  and  $q_{ge}$ ) tied to the model (Figure 2.8), on the packet level to model dynamic channel whose reliability can be affected by the environmental factors. Depending on the Channel Failure Rate (CFR), these two parameters  $p_{ge}$  and  $q_{ge}$  need to be changed accordingly.



**Figure 2.8. Gilbert-Elliott model to model network dynamicity**

In addition to the node/area conditions, the distance between a node and a given critical region could also impact the quality of reported data for the coverage task. Therefore, we also consider this parameter so that the coverage quality varies exponentially with the distance between a node and a given critical region [46], as shown in Equation (2.10).

### 2.4.2.2 Reputation and trust models

When the base station is not able to directly monitor nodes functions, it is essential to be able to enquire about the functional reputation of those nodes. In human societies this is usually achieved by soliciting opinions from others that may have had direct experience with a given entity. We utilize a reputation based model, in which the actions of each node are observed by other nodes in an attempt to evaluate their trustworthiness. To this end, the base station will ask other nodes their opinions about performance of a given node. These opinions will be given as real numbers. Reputation



and trust concepts are being recently used in wireless sensor networks to diminish the impact of malicious and faulty nodes and links [47]. Having history of nodes' activities and links' states can give useful information about their situation, based on which the best situation-aware policy can be chosen to improve network performance. A basic assumption of reputation systems is that reputation values can help to better predict the future performance of the respective entities and thereby to reduce uncertainty of relying parties during the decision making processes.

In binomial reputation systems, the ratings would be expressed with two values, as either positive (satisfying) or negative (unsatisfying), which reflect a corresponding performance of an entity. Instead, multinomial reputation systems give the possibility of providing ratings in different discrete levels such as {extremely negative, negative, neutral, positive, extremely positive}.

We select a binomial reputation system, which is applied to the binary state space {positive, negative}. The best way to represent reputation is a statistical probability distribution such as Beta, Poisson, and Gaussian, but to judge the reputation of the nodes and links we must have numerical values, representing the trust values. In order to model reputation, we opt to use beta distribution, which is a continuous distribution functions over a binary state space indexed by the two parameters  $\alpha$  and  $\beta$ . The main reason of choosing beta distribution is because of its simplicity, strong foundation on statistical theory, and its appropriateness in representing the posteriori probability of binary events as explained in [47]. In short, beta distribution represents the probability distribution of seeing a particular combination of satisfying and unsatisfying values from a node/link.

Trust can also be defined as the probability expectation value of the reputation function. The beta Probability Density Function (PDF)  $f(p | \alpha, \beta)$  can be described using the gamma function as:

$$f(p | \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1} (1-p)^{\beta-1} \quad (2.2)$$

$$0 \leq p \leq 1, \alpha > 0, \beta > 0$$

where  $p \neq 0$  if  $\alpha < 1$  and  $p \neq 1$  if  $\beta < 1$ .

The probability expectation value of this beta distribution is given by Equation (2.3), which represents the reputation score or the trust value.

## 2.5. Trust-based probabilistic coverage

---

$$E(p) = \frac{\alpha}{\alpha + \beta} \quad (2.3)$$

In our binomial reputation system, if  $Rt_p$  denotes the number of positive rating and  $Rt_n$  denotes the number of negative ratings, then the PDF of observing positive service in the future can be expressed as a function of past observations by setting:

$$\begin{aligned} \alpha &= Rt_p + 1 \\ \beta &= Rt_n + 1 \\ Rt_p &\geq 0, Rt_n \geq 0 \end{aligned} \quad (2.4)$$

## 2.5. Trust-based probabilistic coverage

Our approach consists of two processes one being performed by the sensor nodes and the other by the base station. In what follows, we elaborate on these processes which are also depicted in Figure 2.9.

- Process on the sensor node:
  - Given an already densely deployed wireless sensor network, sensor nodes regularly send their opinions about their neighbors in terms of availability and sensing/processing to the base station in a multi-hop manner. Even asleep sensor nodes are required to do so, which implies that they have to wake up, scan their neighborhood, and inform the base station about their opinion, if needed. The mechanism of neighborhood scanning and opinion quantifying will be elaborated on in Section 2.5.1. All sensor nodes should also inform the base station if the amount of their energy goes below a predefined threshold  $\theta^E$ . This enables the base station to have a comprehensive view of states, in which all nodes are.
- Process on the base station:
  - Upon receiving this information by the base station, it quantifies the reliability of each node as will be explained in Section 2.5.1, in the form of node's confidence level.
  - The nodes confidence level and available energy enable the base station to find a set of most appropriate sensor nodes which could effectively cover the deployment area. In order to find such set which is called Feasible Set

(FsbSet), the base station runs either TPC or TPC-Greedy algorithm which are elaborated on in Section 2.6.

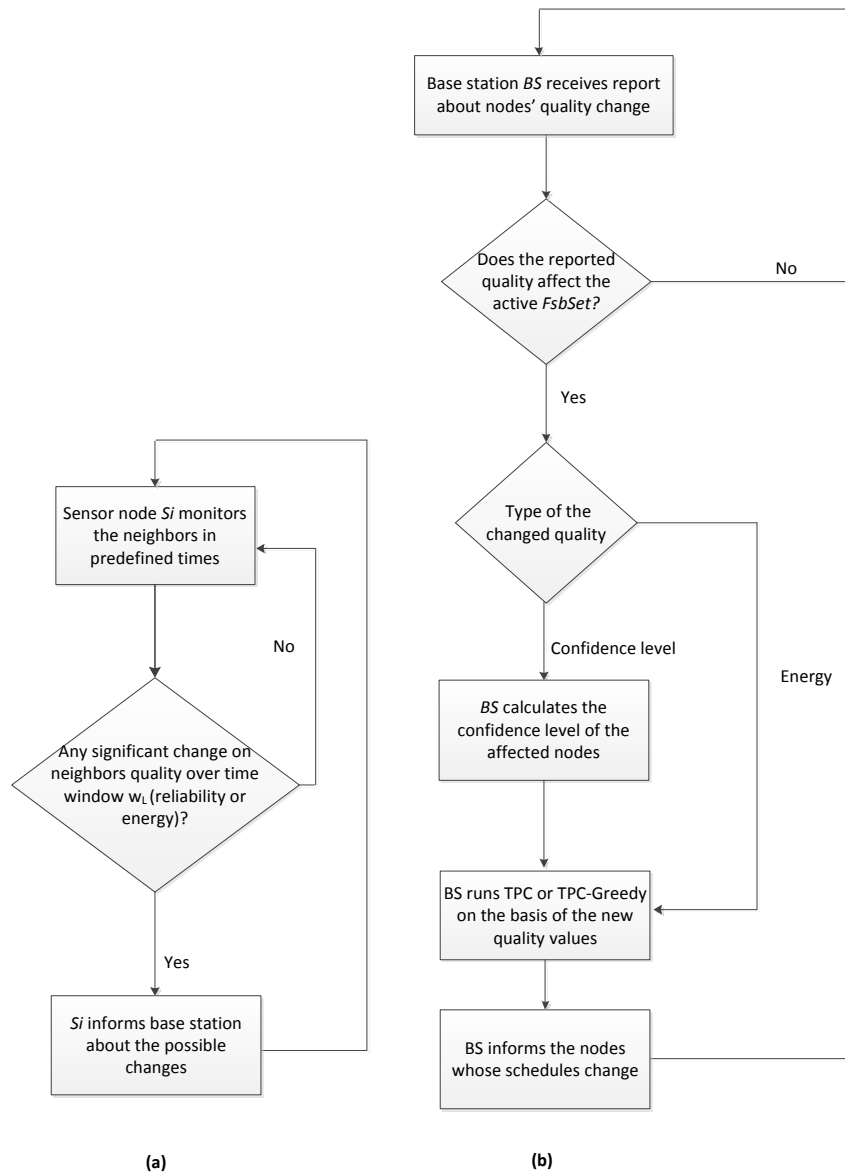


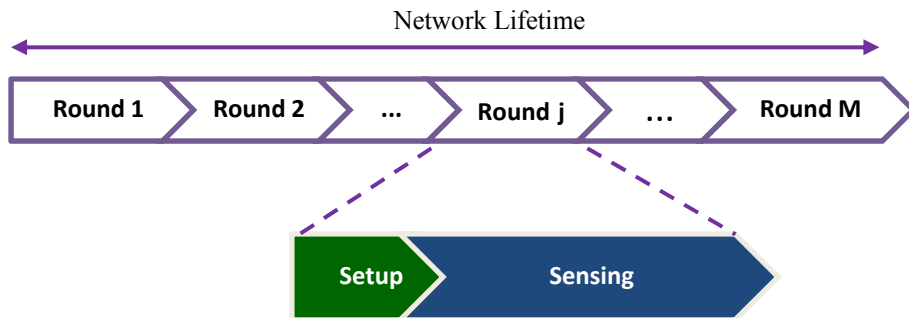
Figure 2.9. Flowchart of processes in (a) the sensor nodes and (b) the base station

## 2.5. Trust-based probabilistic coverage

---

- Finally, the base station informs sensor nodes belonging to the FsbSet to be active (stay ON) and monitor their nearby critical/uncritical regions. The members of FsbSet should stay ON until the energy of any of them goes below a threshold  $\theta^E$ , or any significant change in the confidence level of the active nodes reported. In this situation, the base station should select another feasible set whose members should be informed to become active.

We organize the network activities into several rounds with not necessarily the same length. This means that the base station runs our algorithms at intervals of a round. Each round, as shown in Figure 2.10, starts with a set-up phase followed by a sensing phase. In the set-up phase, the base station has to decide a set of nodes (feasible set), which should be kept active in the sensing phase.



**Figure 2.10. Network activities organization**

### 2.5.1 Calculating node's confidence level

In this section we present how to calculate the confidence level of each node. The confidence level is required by the base station to judge about a node's appropriateness to act as an active node. We address the following three parameters which lead to uncertainty of the nodes measurements:

- (1) The distance between a critical region and a sensor node.
- (2) The internal reliability state of the network.
- (3) The internal reliability state of individual sensor nodes.

Usually, sensing ability of a node is directly dependent on the distance between the given node and a critical region. We assume the sensing ability of a given node is

reduced if its distance to the critical regions increases. As the sensor nodes and the critical regions are stationary, the distance between a sensor node and the critical regions remains always unchanged.

The second and third parameters mentioned above are not fixed and may change depending on the physical conditions that network or nodes experience. In actual deployments the coverage area can be affected by the obstacles, which can absorb or reflect the RF signal put out by the node thereby making the area behind them invisible to the node (Figure 2.7). Therefore, the sensing quality of each node may vary for different critical points due to the environmental conditions surrounding each critical point. In this sense, we should calculate the sensing quality of each node per critical point as each node may introduce different sensing qualities for different critical points.

As described in Section 2.4.2.2, we assess the quality of measurements using functional reputation and trust model. In this way, each sensor node  $a$  should act as a referee node and periodically calculates the sensing/processing reputation  $\rho_{a,b}^{Sen}$  and availability reputation  $\rho_{a,b}^{avl}$ , for the given node  $b$  using Equation (2.3) and (2.4).

All sensor nodes have to wake up after a certain amount of time  $\tau_o$  to monitor their region to detect any misbehavior of their neighbors in terms of sensing or availability. To this end, each sensor node  $a$  should broadcast a hello message or a packet which carries the sensed value of that node at that time to one-hop neighborhood. If any of the neighboring nodes (e.g. node  $b$ ) does not receive this messages due to environmental imperfections or wrong active/sleep schedule executed by the neighboring node  $b$ , or does not respond the given message, the availability of that neighboring node  $b$  should be reduced. If the respond message which contains the measurement of node  $b$  is received by node  $a$ , but that measurement significantly differs from the measurement of the given node  $a$ , the sensing quality of the neighboring node  $b$  should be reduced. One should note that depending on the application and network deployment, different similarity function or distance measurement can be utilized in order to quantify the similarity between sensory data of node  $a$  and sensory data of node  $b$ . We utilize a simple similarity function just by comparing the result of subtracting two sensory data of node  $a$  and  $b$ , with a specific threshold.

After finding the similarities/dissimilarities, the positive or negative opinion (quality) of the referee node  $a$  about reference node  $b$  is recorded in a numerical format: -1 if the opinion is negative and +1 if the opinion is positive.

## 2.5. Trust-based probabilistic coverage

---

After gathering a certain amount of opinions,  $\tau_w = L \times \tau_o$  ( $L > 1$ ) we will have  $L$  opinions about a reference node  $b$  in node  $a$ 's memory. We define a window with size  $L$ , which represents the opinion of a given node about their neighbors during past  $\tau_w$  time unit.

If the reputation value of node  $b$  during the past  $\tau_w$  time unit does not significantly differs from the previous reputation value calculated for this node  $b$ , no update value about  $b$ 's reputation will be sent to the base station. Otherwise, the base station will be updated with the new state of node  $b$  from the viewpoint of node  $a$ . This information could be piggybacked to the reported data of active nodes, in order to reduce the data transmission overhead.

Base station is responsible to determine the best active nodes for the next round according to these reputation values. Therefore, after receiving the information from each sensor node, the base station must obtain a consensus on the neighbors' viewpoint of every node about its observation quality for a given critical region. To this end, the base station first employs Equation (2.5) and Equation (2.6) which are trust-based weighted voting, to calculate global availability reputation  $\gamma_b^{avl}(t)$  for a given node  $b$  as well as global sensing reputation  $\gamma_{b,c}^{sen}$  for a given node  $b$  about the critical region  $c$ , at time instant  $t$ .

$$\gamma_{b,c}^{sen}(t) = \frac{\sum_{a \in Nei(b)} \gamma_{a,c}^{sen}(t-1) \times \rho_{a,b}^{sen}}{\sum_{a \in Nei(b)} \gamma_{a,c}^{sen}(t-1)} \quad (2.5)$$

$$\gamma_b^{avl}(t) = \frac{\sum_{a \in Nei(b)} \gamma_a^{avl}(t-1) \times \rho_{a,b}^{avl}}{\sum_{a \in Nei(b)} \gamma_a^{avl}(t-1)} \quad (2.6)$$

Where  $Nei(b)$  denotes the neighboring nodes of  $b$ . As shown in these Equations, the impact of the opinion of one node  $a$  on the final reputation value of a node  $b$  in time period  $t$ , depends on the global reputation of this node  $a$  for the given functionality (sensing or availability), which is calculated in time period  $t-1$ . This trust-based voting is motivated by the assumption that the opinions of the node with the higher trust value should have more impact on the final decision about global reputation value of other node.

In calculating  $\gamma_{b,c}^{sen}$ , the base station could also consider the distance-based weighted voting which is motivated by the assumption that the sensor nodes nearest to a critical

point usually has the most accurate opinion about the given critical point, as the signal propagation from a critical region to sensor node usually follows a probabilistic model. Therefore, opinion of the node close to those points exposes more weight. Using distance-based weighted voting scheme, Equation (2.5) will turn into Equation (2.7). According to Equation (2.7), the base station considers the reverse distance of  $a \in Nei(b)$ , to the given critical point  $c$  as a weight of  $a$ 's vote about the  $b$ 's sensing quality for critical point  $c$ . This implies that the nodes closer to a critical point have better insight about that critical point and hence, their opinion could be considered with a higher weight.

$$\gamma_{b,c}^{sen}(t) = \frac{\sum_{a \in Nei(b)} dist(a, c)^{-1} \times \gamma_{a,c}^{sen}(t-1) \times \rho_{a,b}^{sen}}{\sum_{a \in Nei(b)} dist(a, c)^{-1} \times \gamma_{a,c}^{sen}(t-1)} \quad (2.7)$$

where  $dist$  is the Euclidean distance.

Thereafter, the base station combines weighted global sensing/processing reputation and weighted global availability reputation to obtain the total reputation  $\gamma_{b,c}^{total}$  of node  $b$  for critical point  $c$  as Equation (2.8):

$$\gamma_{b,c}^{total}(t) = \omega_{av} \times \gamma_b^{avl}(t) + \omega_{sn} \times \gamma_{b,c}^{sen}(t) \quad (2.8)$$

We introduce two weights  $\omega_{sn}$  and  $\omega_{av}$  for global sensing/processing reputation and global availability reputation in order to prioritize them if needed. We could also address the impact of distance on the sensing reputation value using Equation (2.7), and if this distance parameter is not important for the application we could employ Equation (2.6) instead of (2.7).

After finding the nodes confidence level, the base station could find the  $FsbSet$  whose nodes should be active in the next time interval. The algorithm, i.e. TPC or TPC-greedy, that base station utilizes to find such set is expanded in the Section 2.6.

## 2.6. A trust-based probabilistic ILP-based coverage algorithm (TPC)

We formulate the problem of minimizing the number of sensor nodes in a feasible set while guaranteeing the quality and reliability requirements, as an Integer Linear Programming.

## 2.6. A trust-based probabilistic ILP-based coverage algorithm (TPC)

---

For the sake of simplicity in the formulation and without loss of generality, we assume all cells depicted in Figure 2.5 including both critical and uncritical regions have a critical point, but with different required reliability. In this sense, those cells, which are already specified to have critical points, demand higher reliability than the uncritical regions.

We define a *FsbSet*, which is a set of sensor nodes selected in such a way that all regions are covered at least by one of the nodes of such set.

An appropriate or minimal *FsbSet* is denoted by a decision binary vector  $x$ , where  $Sn_j$  is included in the set if  $x_j = 1$ , otherwise  $x_j = 0$ .

The optimization problem is stated as follows:

Given:

- A set of  $n$  sensor nodes,  $SS = \{Sn_1, Sn_2, \dots, Sn_n\}$
- A set of  $m$  critical regions,  $CRS = \{CR_1, CR_2, \dots, CR_m\}$
- A vector *SRRL* each element of which represents the reliability required by each critical region,  $SRRL = [RRL_1 RRL_2 \dots RRL_m]$
- A matrix  $SCL_{m \times n}$  consists of  $\gamma_{i,j}^{total}$  for a given critical region  $j$  and a given sensor node  $i$ .
- A vector *Eg* each element of which represents the residual energy (*RsEg*) of the nodes,  $Eg = [RsEg_1 RsEg_2 \dots RsEg_n]$ .
- A matrix  $Rm_{m \times n}$  which makes a relation between *SS* and *CRS* as Equation (2.9) where  $R_s$  represents the sensing range and *dist* shows the Euclidean distance.

$$Rm_{i,j} = \begin{cases} 1 & \text{if } dist(Sn_j, CR_i) \leq R_s \\ 0 & \text{otherwise} \end{cases} \quad (2.9)$$

The above binary sensor model assumes that sensor measurements have no associated uncertainty considering distance, while in reality, sensor measurements should be represented in probabilistic form as the signal propagation from a critical region to a sensor node follows a probabilistic model. To this end, the above binary relationship matrix can be replaced with Equation (2.10) to represent the impact of the distance on the sensing quality:



$$Rm_{i,j} = \begin{cases} e^{-\delta \times dist(Sn_j, CR_i)^\lambda} & \text{if } dist(Sn_j, CR_i) \leq R_s \\ 0 & \text{otherwise} \end{cases} \quad (2.10)$$

where  $\delta$  and  $\lambda$  are parameters related to the physical characteristics of the sensing device, which can be obtained from field experiments.

Apart from the impact that the distance between a sensor node and a critical region may have on the sensing quality, malfunctioning of the sensor nodes or changing environmental conditions may influence the sensing and coverage quality as well. By Equation (2.11), we address these parameters where  $SCL_{i,j}$  is already introduced above and  $P_{j,i}^{obs}$  is the probability that  $Sn_j$  could observe  $CR_i$ , which is a combination of  $Rm_{i,j}$  and  $SCL_{i,j}$  as follows:

$$P_{j,i}^{obs} = Rm_{i,j} \times SCL_{i,j} \quad CR_i \in CRS, Sn_j \in SS \quad (2.11)$$

We could also assign weights to  $Rm_{i,j}$  and  $SCL_{i,j}$  in Equation (2.11) in order to prioritize them.

Objective: Minimizing the bellow function

$$\sum_{j=1}^n x_j \quad (2.12)$$

Subject to

$$\sum_{j=1}^n x_j \times P_{j,i}^{obs} > 0 \quad \text{for all } CR_i \in CRS \quad (2.13)$$

$$RsEg_j > MinE \quad \text{for all } Sn_j \in SS \quad (2.14)$$

$$P_{cvr}(i) \geq RRL_i \quad \text{for all } CR_i \in CRS \quad (2.15)$$

$$x_j \in \{0,1\} \quad \text{for all } Sn_j \in SS \quad (2.16)$$

It is worth mentioning as shown in the constraints, the residual energy of the nodes which are selected for the coverage set, should be higher than a minimum requirement  $MinE$ .

## 2.6. A trust-based probabilistic ILP-based coverage algorithm (TPC)

---

As it is quite possible that one critical region is covered by more than one active sensor node, the probability of monitoring that critical region must be calculated precisely. Usually, when probability of two simultaneous events is calculated while these events are not mutually exclusive, additive law of probability [48] is used. This means that the probability of observing a critical region  $C$  with two sensor nodes  $A$  and  $B$  is calculated as follows:

$$P_{Cvr}(C) = P_{A,C}^{obs} + P_{B,C}^{obs} - P_{A,C}^{obs} \times P_{B,C}^{obs} \quad (2.17)$$

where  $P_{Cvr}(C)$  is the probability that critical region  $C$  is covered by any active sensor node.

In case of having more than two nodes observing a specific critical region  $C$  at a time, the additive law probability becomes complex. To overcome this issue, we examine the problem from the unobserved probability perspective and calculate the cumulative coverage probability as:

$$P_{Cvr}(C) = 1 - \prod_{j \in NS_C} (1 - P_{j,C}^{obs}) \quad (2.18)$$

$$NS_i = \{j | Rm_{i,j} > 0 \text{ and } Sn_j \text{ is active}\}$$

### 2.6.1 A greedy trust-based probabilistic heuristic algorithm (TPC-Greedy)

The complexity of finding an optimal solution using ILP grows exponentially as the size of the network increases. Therefore, we put forward a greedy heuristic algorithm, which ideally aims to achieve the same results as of ILP approach using reasonable computational resources without exploring all combinations of sensor nodes. This algorithm finds a sub-optimal solution for the coverage problem in polynomial time and can easily be extended to work in a distributed fashion. The pseudocode of TPC-Greedy is presented in Figure 2.12.

#### 2.6.1.1 Parameter definition for TPC-Greedy

Before elaborating on our greedy algorithm, we need to explain some parameters, which will be used by the algorithm.

First parameter is node's contribution degree  $\psi$ , which states the number of critical/uncritical regions (cells in Figure 2.5) each node could cover. For example as Figure 2.11 illustrates the contribution degree of marked nodes are as follows:  $\psi(S_1) = 2$ ,  $\psi(S_2) = 1$  and  $\psi(S_3) = 2$ .

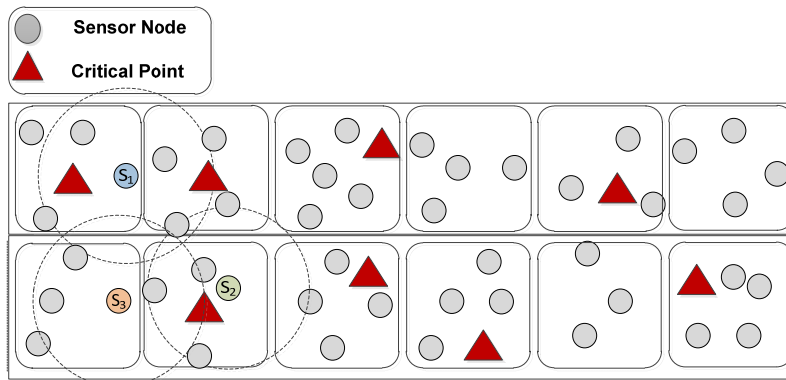


Figure 2.11. An example of node's contribution degree

Moreover, we have two sets of nodes which are referred in our algorithm:

- $Z^{cr}$  : a set of sensor nodes that have at least one critical point within their sensing range.
- $Z^{co}$  : a set of sensor nodes that have the center of at least one uncritical region within their sensing range.

### 2.6.1.2 TPC-Greedy algorithm

TPC-Greedy operates in iterations during which one *FsbSet* from available nodes is built. Basically, TPC-Greedy adds sensor nodes to the *FsbSet* in a greedy way based on  $P^{obs}$  and  $\psi$  that nodes offer for different regions.

This algorithm should be executed twice, once with the input parameter  $Z^{cr}$  (step A.20. in Figure 2.12) and once with the input parameter  $Z^{co}$  (step A.21. in Figure 2.12). What we do in these two steps is basically to first select those nodes using which all critical regions can be observed with the required reliability RRL. Thereafter, we select those nodes using which all uncritical regions can be monitored if they have not yet covered by the already selected nodes for the critical regions. The union of outputs of these two execution rounds of algorithm results in *FsbSet*.

## 2.6. A trust-based probabilistic ILP-based coverage algorithm (TPC)

---

The sensor nodes from the  $FsbSet$  will be put in active mode and the rest will be in sleep mode. Whenever the energy of the current active nodes goes below the threshold  $\theta^E$  or the base station noticed a significant variation in the confidence level of active nodes, the base station has to update the  $FsbSet$  using the up-to-date state of the nodes and network. To this end, the base station runs our greedy algorithm (Figure 2.12) from the beginning (step B.1) with the new values, to discover a new  $FsbSet$  (if any). Finally, the network will die if the base station is unable to define  $FsbSet$  anymore using the available sensor nodes. In what follows, we present and elaborate on our greedy heuristic algorithm.

To be able to find  $FsbSet$ , residual energy of each node is used to probabilistically choose the initial set of active nodes. In this way, first the base station selects the nodes whose energy are higher than  $MinE$  and then sort them according to the  $P^{obs}$  that they offer for different  $m$  critical regions in the descending order. It is possible that some nodes provide almost similar  $P^{obs}$  and hence we sort those nodes according to their  $\psi$  parameter in the descending order. This implies that if there are two nodes whose  $P^{obs}$  are almost similar, we select the node with higher contribution degree  $\psi$ . By doing so, in the sorted matrix  $CC$  we will have  $(Sn_s, P_{s,h}^{obs})$  pair as:

$$CC = \begin{matrix} & 1 & \dots & k & \\ \begin{matrix} 1 \\ \vdots \\ m \end{matrix} & \begin{bmatrix} (Sn_l, P_{l,1}^{obs}) & \dots & (Sn_k, P_{k,1}^{obs}) \\ \vdots & \ddots & \vdots \\ (Sn_r, P_{r,m}^{obs}) & \dots & (Sn_q, P_{q,m}^{obs}) \end{bmatrix} & & \end{matrix} \quad (2.19)$$

Then we select the node, which provides the highest  $P_{x,y}^{obs}$  and  $\psi$ , and check which critical/uncritical region is covered by that node (step B.10). If the given region has not already been covered by any node or its  $RRL$  has not been satisfied yet, we add that node to the  $FsbSet$ , otherwise we repeat this step while replacing the node with the one which has the next highest  $P_{x,y}^{obs}$ . As each sensor node may be able to observe more than one critical/uncritical region, we check whether there is other region(s), which can be covered by this node. If so, we mark those region(s) as being covered by that specific node with the probability extracted from  $C$  matrix for each region (step B.14). After this node is added to the  $FsbSet$ , it is removed from  $C$  matrix (step B.18). We repeat step B.9 till step B.23 as long as no region is remained uncovered. We need to calculate the reliability that  $FsbSet$  provides for each region (step B.25). We use the additive law probability represented by Equation (2.18) to calculate reliability for a region if more than one member of  $FsbSet$  is observing the given region. If the obtained reliability by the  $FsbSet$  is less than  $RRL$  for a critical/uncritical region, step B.8 till step B.27 are repeated as long as all critical regions are covered with their  $RRL$ . The base station

informs sensor nodes about their schedule based on which they should switch between active/sleep mode for the next time interval.

**Procedure** TPC-Greedy-Main ()

```

A.1.   For  $CR_i \in CRS$ 
A.2.      $CCov(CR_i) = \{\}$ 
A.3.      $P_{Cvr} = 0$ 
A.4.     If ( $CR_i = CriticalRegion$ )
A.5.        $CrR = CrR \cup \{CR_i\}$ 
A.6.        $Z^{cr} = \{Sn_j | dist(Sn_j, CR_i) \leq R_s, Sn_j \in SS\}$ 
A.7.     End
A.8.     If ( $CR_i = CommonRegion$ )
A.9.        $CoR = CoR \cup \{CR_i\}$ 
A.10.       $Z^{co} = \{Sn_j | dist(Sn_j, CR_i) \leq R_s, Sn_j \in SS\}$ 
A.11.    End
A.12.    End
        /*Contribution degree*/
A.13.    For  $CR_i \in CRS$ 
A.14.      For  $Sn_j \in SS$ 
A.15.        If ( $dist(Sn_j, CR_i) \leq R_s$ )
A.16.           $\psi_{Sn_j} = \psi_{Sn_j} + 1$ 
A.17.        End
A.18.      End
A.19.    End
        /*Main*/
A.20.     $FsetCr = TPC\_Greedy(Z^{cr}, CrR)$ 
A.21.     $FsetCo = TPC\_Greedy(Z^{co}, CoR)$ 
A.22.     $FsbSet = FsetCr \cup FsetCo$ 

```

**Function** TPC-Greedy (Z, CRS)

```

B.1.     $FSet = \{\}$ 
B.2.    For  $Sn_j \in Z$ 
B.3.       $SE = \{Sn_j | RsEg^{Sn_j} > MinE\}$ 
B.4.    End
B.5.    For  $Sn_j \in SE$ 
B.6.       $SCov(Sn_j) = \{CR_i | P_{Sn_j, CR_i}^{obs} > 0, CR_i \in CRS\}$ 
B.7.    End
B.8.    Repeat
B.9.      Repeat
B.10.      $[x, y] = \mathbf{arg\ max}_{x,y} (P_{x,y}^{obs}, \psi_x)$ 

```

## 2.7. Performance evaluation

---

```

B.11.      If ( $CCov(CR_y) = \{\}$  ||  $P_{Cvr}(CR_y) < RRL(CR_y)$ )
B.12.           $FSet = FSet \cup \{Sn_x\}$ 
B.13.      For  $z \in SCov(Sn_x)$ 
B.14.           $CCov(z) = CCov(z) \cup \{Sn_x\}$ 
B.15.      End
B.16.      For  $P_{s,t}^{obs} \in CC$ 
B.17.          If ( $s == x$ )
B.18.               $P_{s,t}^{obs} = 0$ 
B.19.          End
B.20.      Else
                find the next best (x,y) using  $\arg \max_{x,y} (P_{x,y}^{obs}, \psi_x)$  and go to B.11.
B.21.      End
B.22.      Until ( $(CCov(CR_i) \neq \{\}, \forall CR_i \in CRS)$  ||  $(P_{Sn_j,CR_i}^{obs} = 0, \exists CR_i \in CRS, \forall Sn_j \in SE)$ )
B.23.      For  $CR_i \in CRS$ 
B.24.          Use equation (2.18) to calculate reliability
B.25.      End
B.26.      Until ( $(P_{Cvr}(CR_i) \geq RCL_{CR_i})$  ||  $(P_{Sn_j,CR_i}^{obs} = 0, \exists CR_i \in CRS, \forall Sn_j \in SE)$ )
B.27.      Return  $FSet$ 

```

---

**Figure 2.12. Pseudocode of TPC-Greedy**

## 2.7. Performance evaluation

In this section, we evaluate our algorithms (TPC, TPC-Greedy) and compare them with two coverage schemes, so called BNR and DST. BNR [11] is a binary-based coverage algorithm which represents the effect of distance between a sensor node and a given critical point in binary way as Equation (2.9). DST [40] is a probabilistic-based coverage scheme in which the coverage quality varies exponentially with the distance between a node and a given critical region as shown in Equation (2.10). The reason that we select these two approaches is to have a fair comparison about (i) the effect of probabilistic coverage by comparing with BNR and (ii) the effect of other important parameters that may influence the quality of coverage function by comparing with DST approach which only looks into the distance as the determinative parameter for putting a node in a coverage set.

### 2.7.1 Performance metrics

We consider lifetime, reliability, and energy-efficiency as three performance metrics. There are different definitions for lifetime of a wireless sensor network. Indeed, lifetime is an application-dependent concept. For the coverage problems, lifetime is the time duration that all targets/critical points or the entire monitoring area is continuously covered. The lifetime in the coverage problem is usually considered to be in direct relation with the number of obtained feasible sets [7, 8, 10], which are activated successively whenever the energy of the current active nodes is depleted. This way of evaluating lifetime is valuable when the only important parameter is energy. However, since we also look into reliability parameter along with energy, the number of feasible sets (or rounds) could not accurately represent the lifetime of the networks for our approaches. Therefore, we could consider the number of sensed data as a metric to judge about the lifetime of different approaches.

Energy efficiency is defined as the ratio of the energy consumed for the packets which satisfy required reliability RRL to the total energy spent in the network.

### 2.7.2 Simulation setup and scenario

We simulate a stationary network with sensor nodes, which are uniformly randomly distributed in an area of  $200m \times 25m$  and a base station is located at (100,0). We consider 4 critical regions, sensing range of nodes is set to 35m and  $l = 25$ .

Unless otherwise stated, the simulation parameters are as described here. At any moment in time, 70% of all nodes and links work almost properly with failure rate of 0.09. The failure rate of other 30% of the nodes/links is set to 0.85. The selection of failing nodes/links occur randomly after every 1000 time units in order to simulate temporal correlation among failures of those 30% nodes/links. All these faulty nodes/links misbehave according to the three-state Markov model (shown in Figure 2.1) for a duration of 1000 time units. Failure rates and the parameters related to the three-state Markov model and Gilbert-Elliot model for the normal nodes and links, which work almost properly and those 30% faulty node/link are presented in Table 2.1.

The output power of our radio model (TICC2420) is programmable in eight levels (from approximately  $-25$  to  $0$  dBm). All dormant nodes use the minimum power level to be able to reach the closest neighbor in order to build reputation values. The initial energy of the nodes  $IniE$  are  $5J$  and the  $MinE=0.3 \times IniE$  and  $\theta^E = 0.1 \times IniE$ .

## 2.7. Performance evaluation

Required reliability (RRL) for all critical regions is 0.98 and for the uncritical regions is 0.60.

**Table 2.1. Simulation parameters**

Normal Node	NFR= 0.09, $p_m=0.01$ , $q_m=0.1$ , $s_m=0.00001$ , $t_m=0.0001$
Faulty Node	NFR=0.85, $p_m=0.9$ , $q_m=0.16$ , $s_m=0.0001$ , $t_m=0.001$
Normal Link	CFR= 0.09, $p_{ge}=0.01$ , $q_{ge}=0.1$
Faulty Link	CFR=0.85, $p_{ge}=0.9$ , $q_{ge}=0.16$

In our assumption we stated that the transmission range ( $R_T$ ) of sensor node is variable and also relatively large compared to sensing range. Therefore, we send the sensory data to the base station in a multi-hop way through the intermediate nodes. The radio model is assumed to be CC2420.

The reputation weights are  $\omega_{sn} = 1$  and  $\omega_{av}=1$ . Moreover,  $\tau_w=25$  samples and  $\tau_o = 5$  samples,  $\delta = 0.1$  and  $\lambda = 2$ .

We use the optimization toolbox in Matlab to solve the ILP.

### 2.7.3 Performance evaluation

We carry 50 experimental trials and the average results of these trials are presented in Figures illustrated in this Section.

In the first experiment, we vary the number of sensor nodes between 40 to 110 with an increment of 10. By varying this parameter, we analyze the impact of node density on the performance. From Figure 2.13 we observe that with an increase in the network density, the network lifetime shows a linear increase because a given cell or region can be monitored by more sensor nodes. This offers a higher opportunity for a region to be in the sensing range of multiple sensors. One can see that the network lifetime when applying TPC algorithm is longer than when TPC-Greedy is applied. This happens because the TPC explores almost all combinations of sensor nodes that satisfy optimization constraints and can provide the optimal solution, however, TPC pays for this by either imposing a longer execution time or requiring a powerful processing unit to explore all combinations. As TPC and TPC-Greedy should satisfy the *RRL* of the critical regions which in our scenario is 0.98, possibly sometimes more than one node is asked to observe a critical region. Moreover, building trust values requires extra energy. This is the main reason that these two approaches usually present shorter lifetime.



We illustrate the average reliability of uncritical and critical regions separately in two separate graphs, as the required reliability RRL for uncritical and critical regions are different. The average reliability provided for the uncritical regions where  $RRL=0.6$  is shown in Figure 2.14, while the average reliability provided for the critical regions where  $RRL=0.98$  is shown in Figure 2.15. One can see that although the RRL is satisfied for uncritical regions by all four coverage schemes, in case of critical regions, only TPC and TPC-Greedy guarantee the required reliability RRL.

Figure 2.16 presents the energy efficiency of these four coverage schemes according to the type of the region being monitored. In the case of critical regions, TPC algorithm is the most promising one as it almost always guarantees the RRL requirement which is 0.98 for the critical region and 0.6 for the uncritical region. However, almost 30% of energy spent in DST and 20% energy utilized in BNR are not quite wasted. The main reason is that the provided reliability by these approaches is lower than RRL (i.e. 0.98) and thus that sensory data is not valuable for the base station. In this regards, the energy spent for such unreliable data is considered to be just wasted. In case of uncritical regions whose RRL is 0.6, even a straightforward algorithm like BNR could often find such a coverage set that satisfies RRL. Therefore, energy-efficiency of TPC and TPC-Greedy for uncritical regions are slightly lower than others because of updating trust values which may be unnecessary when RRL is small. To cope with this, we could lower down the trust variation threshold of the uncritical regions to avoid having unnecessary updates over trust values. In other words, updating trust values of uncritical regions should be less frequent than that of critical regions. According to Figure 2.13 although the lifetime of BNR and DST is slightly longer than TPC and TPC-Greedy but Figure 2.16 reveals that most of that lifetime spent for relaying quite unreliable data which does not meet the application requirement. Therefore, the energy of BNR and DST is just wasted for such unreliable data.

In the next experiment we vary the failure rate of the 30% faulty nodes/links between 5% and 95%, in order to see the impact of failure rate over the reliability of uncritical/critical regions. We also change the RRL for uncritical regions to 0.8. According to Figure 2.17 and Figure 2.18, TPC and TPC-Greedy well adapt to the failure rate and always guarantee the required reliability RRL. However, BNR and DST show reliability reduction according to the failure rate that the network is suffering from. According to Figure 2.17 although reliability for the uncritical region decrease with TPC and TPC-Greedy, but the required reliability which is 0.8 for the uncritical region is always guaranteed.

## 2.7. Performance evaluation

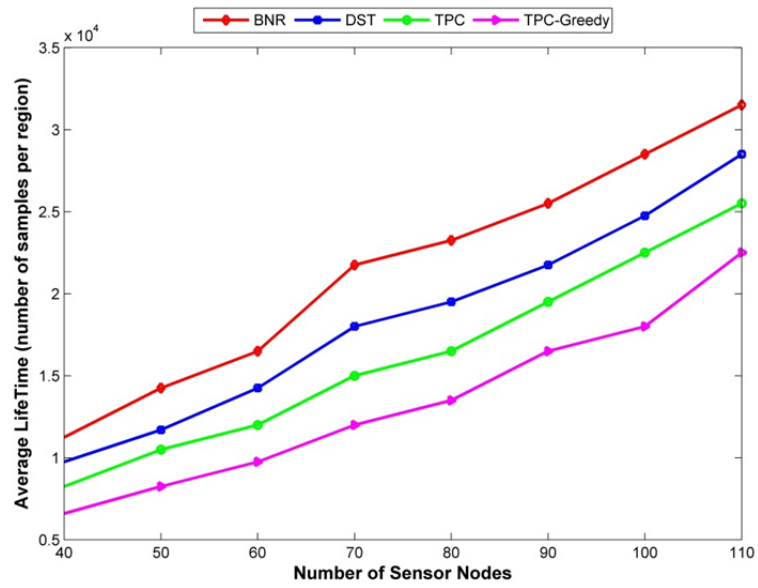


Figure 2.13. Network Lifetime in terms of number of samples per region

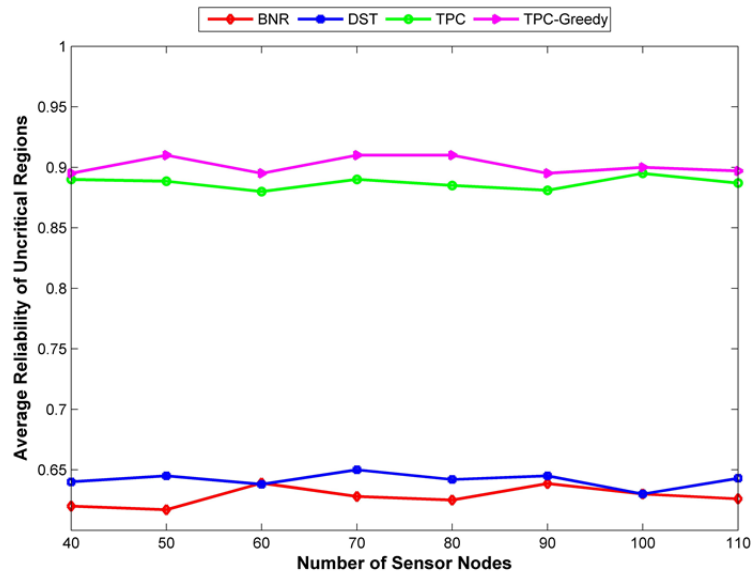


Figure 2.14. Average reliability of the uncritical regions vs. number of nodes

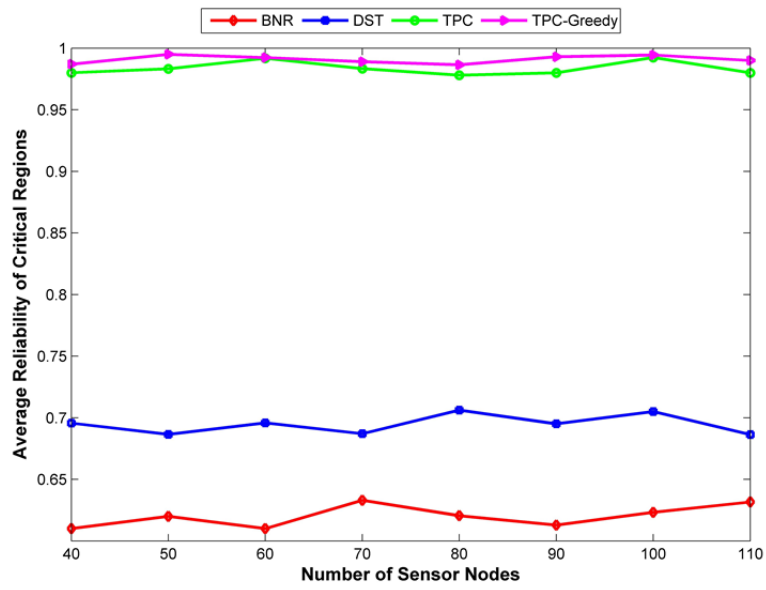


Figure 2.15. Average reliability of the critical regions vs. number of nodes

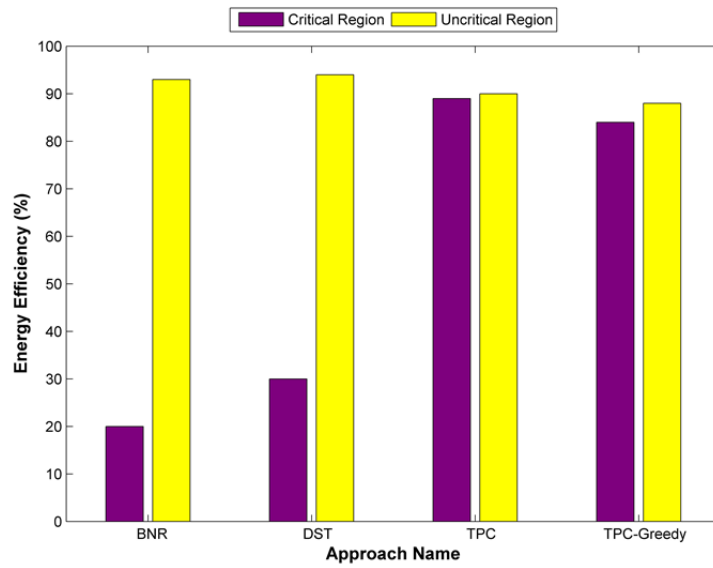


Figure 2.16. Energy efficiency

## 2.7. Performance evaluation

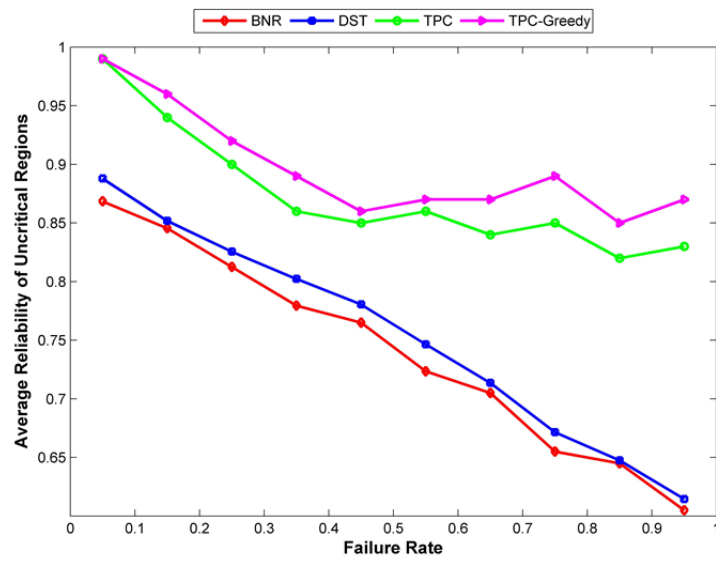


Figure 2.17. Average reliability of uncritical regions vs. failure rate

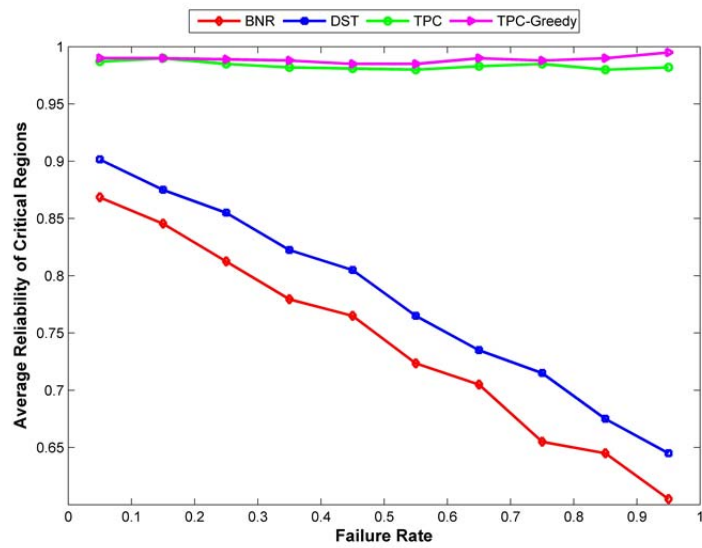
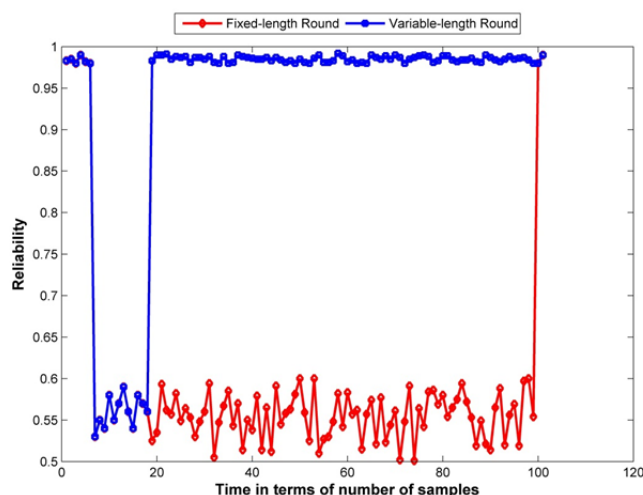


Figure 2.18. Average reliability of critical regions vs. failure rate



**Figure 2.19. Impact of being situation-aware on reliability**

In the last experiment, we evaluate the situation-awareness property of TCP and TCP-Greedy algorithms. To have a fair judgment, we assume a network, in which all nodes and links are almost error-free. Therefore, having only one node is enough to guarantee RRL for a given region. After a while, confidence level of one of the active nodes sharply drops, which significantly influences the reliability of the reported data. In this case, if we have a fixed-length round composed of 100 samples, the effect of this change would last until the end of the current round. However, if that variation is reported to the base station as soon as it happens even if it is in the middle of a round, the average reliability of the reported data would not be influenced that much. Figure 2.19 represents the effect of having a fixed-length round and variable-length round on the achieved reliability over time. If the confidence level variation occurs at the beginning of the interval, almost the whole data gathered for that round is unreliable and probably useless. This becomes even worse and really inefficient if the round length is too large. This matter motivates the need for having variable-length rounds, length of which depends on the node and network situation as in TCP and TCP-Greedy.

## 2.8. Chapter summary

In this chapter, we investigate the coverage issue based on probabilistic coverage and propose a trust-based probabilistic coverage algorithm, which leverages the trust concept to tackle the time-varying uncertainties introduced by the sensor nodes and the

## 2.9. Bibliography

---

environment they operate in. The uncertainty parameters target in this chapter arise from the hostile environment, faulty sensor/processor units and the distance of each node to the critical points. Considering these issues we formulate this problem as an Integer Linear Programming problem, which is able to guarantee the required reliability despite the error-prone nature of wireless sensor networks. We address variable-length rounds for our proposed algorithms TPC and TPC-Greedy in order to cope with the network dynamism which is inevitable for many wireless sensor networks. We also put forward a greedy heuristic algorithm, which achieves almost the same results as ILP without suffering from complexities imposed by ILP. The simulation results show the superiority of our proposed approaches by as much as 60% increase in energy efficiency and up to 40% increase in reliability for critical regions in a dynamic network.

## 2.9. Bibliography

- [1]. Carle, J. and D. Simplot-Ryl, *Energy-efficient area monitoring for sensor networks*. Computer, 2004. 37(2): p. 40-46.
- [2]. Wang, B., *Coverage control in sensor networks* 2010: Springer.
- [3]. Wang, B., et al., *Scheduling sensor activity for information coverage of discrete targets in sensor networks*. Wireless Communications and Mobile Computing, 2009. 9(6): p. 745-757.
- [4]. Meguerdichian, S., et al. *Exposure in wireless ad-hoc sensor networks*. in *7th annual international conference on Mobile computing and networking*. 2001.
- [5]. Chakrabarty, K., et al. *Coding theory framework for target location in distributed sensor networks*. in *International Conference on Information Technology: Coding and Computing*. 2001.
- [6]. Chakrabarty, K., et al., *Grid coverage for surveillance and target location in distributed sensor networks*. IEEE Transactions on Computers, 2002. 51(12): p. 1448-1453.
- [7]. Cardei, M., et al. *Energy-efficient target coverage in wireless sensor networks*. in *24th Annual Joint Conference of the IEEE Computer and Communications Societies*. . 2005.
- [8]. Cardei, M., et al. *Maximum network lifetime in wireless sensor networks with adjustable sensing ranges*. in *International Conference on Wireless And Mobile Computing, Networking And Communications*. 2005.

- [9]. Zhang, H. *Energy-balance heuristic distributed algorithm for target coverage in wireless sensor networks with adjustable sensing ranges*. in *Asia-Pacific Conference on Information Processing*. 2009.
- [10]. Zhang, H., H. Wang, and H. Feng. *A distributed optimum algorithm for target coverage in wireless sensor networks*. in *Asia-Pacific Conference on Information Processing*. 2009.
- [11]. Cardei, M. and D.-Z. Du, *Improving wireless sensor network lifetime through power aware organization*. *Wireless Networks*, 2005. 11(3): p. 333-340.
- [12]. Torkestani, J.A., *An adaptive energy-efficient area coverage algorithm for wireless sensor networks*. *Ad hoc networks*, 2013. 11(6): p. 1655-1666.
- [13]. Cardei, M., et al., *Wireless sensor networks with energy efficient organization*. *Journal of Interconnection Networks*, 2002. 3(03n04): p. 213-229.
- [14]. Slijepcevic, S. and M. Potkonjak. *Power efficient organization of wireless sensor networks*. in *IEEE International Conference on Communications*. 2001.
- [15]. Tian, D. and N.D. Georganas. *A coverage-preserving node scheduling scheme for large wireless sensor networks*. in *1st ACM international workshop on Wireless sensor networks and applications*. 2002.
- [16]. Srinivasan, W.W.V. and K.-C. Chua. *Trade-offs between mobility and density for coverage in wireless sensor networks*. in *13th annual ACM international conference on Mobile computing and networking*. 2007.
- [17]. Wang, G., et al. *Sensor relocation in mobile sensor networks*. in *24th Annual Joint Conference of the IEEE Computer and Communications Societies*. 2005.
- [18]. Sriram Chellappan, B.E., *On deployment and security in mobile wireless sensor networks*. Thesis, The Ohio State University, 2007.
- [19]. Wang, G., G. Cao, and T. La Porta, *Movement-assisted sensor deployment*. *IEEE Transactions on Mobile Computing*, 2006. 5(6): p. 640-652.
- [20]. Mathur, P., et al. *Coverage improvement for wireless sensor networks using grid quorum based node mobility*. in *Research Conference on Networking and Electronic Commerce 2012*.
- [21]. Saipulla, A., et al. *Barrier coverage with sensors of limited mobility*. in *11th ACM international symposium on Mobile ad hoc networking and computing*. 2010.
- [22]. Howard, A., M.J. Matarić, and G.S. Sukhatme, *Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem*, in *Distributed autonomous robotic systems 52002*, Springer. p. 299-308.

## 2.9. Bibliography

---

- [23]. Poduri, S. and G. Sukhatme. *Constrained coverage for mobile sensor networks*. in *International Conference on Robotics and Automation*. 2004.
- [24]. Howard, A., M.J. Matarić, and G.S. Sukhatme, *An incremental self-deployment algorithm for mobile sensor networks*. *Autonomous Robots*, 2002. 13(2): p. 113-126.
- [25]. Grossglauser, M. and D. Tse. *Mobility increases the capacity of ad-hoc wireless networks*. in *20th Annual Joint Conference of the IEEE Computer and Communications Societies*. 2001.
- [26]. Carle, J., A. Gallais, and D. Simplot-Ryl. *Preserving area coverage in wireless sensor networks by using surface coverage relay dominating sets*. in *10th IEEE Symposium on Computers and Communications*. 2005.
- [27]. Wang, X., et al. *Integrated coverage and connectivity configuration in wireless sensor networks*. in *1st international conference on Embedded networked sensor systems*. 2003.
- [28]. Wu, J. and S. Yang. *Coverage issue in sensor networks with adjustable ranges*. in *International Conference on Parallel Processing Workshops, 2004. ICPP 2004 Workshops*. 2004.
- [29]. Ye, F., et al., *Energy efficient robust sensing coverage in large sensor networks*. Computer Science Dept-UCLA, Tech. Rep, 2002.
- [30]. Zhang, H. and J.C. Hou, *Maintaining sensing coverage and connectivity in large sensor networks*. *Ad Hoc & Sensor Wireless Networks*, 2005. 1(1-2): p. 89-124.
- [31]. Kumagai, J., *Life of birds [wireless sensor network for bird study]*. *IEEE Spectrum*, 2004. 41(4): p. 42-49.
- [32]. Szewczyk, R., et al., *Habitat monitoring with sensor networks*. *Communications of the ACM*, 2004. 47(6): p. 34-40.
- [33]. Cayirci, E., et al., *Wireless sensor networks for underwater surveillance systems*. *Ad hoc networks*, 2006. 4(4): p. 431-446.
- [34]. Son, B., Y.-s. Her, and J. Kim, *A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains*. *International Journal of Computer Science and Network Security (IJCSNS)*, 2006. 6(9): p. 124-130.
- [35]. Werner-Allen, G., et al., *Deploying a wireless sensor network on an active volcano*. *Internet Computing*, 2006. 10(2): p. 18-25.
- [36]. Hefeeda, M. and M. Bagheri, *Forest fire modeling and early detection using wireless sensor networks*. *Ad Hoc & Sensor Wireless Networks*, 2009. 7(3-4): p. 169-224.



- [37]. Heo, N. and P.K. Varshney, *Energy-efficient deployment of intelligent mobile sensor networks*. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2005. 35(1): p. 78-92.
- [38]. Huang, C.-F. and Y.-C. Tseng, *The coverage problem in a wireless sensor network*. Mobile Networks and Applications, 2005. 10(4): p. 519-528.
- [39]. Ahmed, N., S.S. Kanhere, and S. Jha. *Probabilistic coverage in wireless sensor networks*. in *The 30th Conference on Local Computer Networks*. 2005.
- [40]. Hefeeda, M. and H. Ahmadi. *A probabilistic coverage protocol for wireless sensor networks*. in *IEEE International Conference on Network Protocols*. 2007.
- [41]. Liu, B. and D. Towsley. *A study of the coverage of large-scale sensor networks*. in *International Conference on Mobile Ad-hoc and Sensor Systems*. 2004.
- [42]. Ying, T., Z. Shu-Fang, and W. Ying. *A distributed protocol for ensuring both probabilistic coverage and connectivity of high density wireless sensor networks*. in *Wireless Communications and Networking Conference*. 2008.
- [43]. Zou, Y. and K. Chakrabarty, *Sensor deployment and target localization in distributed sensor networks*. ACM Transactions on Embedded Computing Systems (TECS), 2004. 3(1): p. 61-91.
- [44]. Zou, Y. and K. Chakrabarty, *A distributed coverage-and connectivity-centric technique for selecting active nodes in wireless sensor networks*. IEEE Transactions on Computers, 2005. 54(8): p. 978-991.
- [45]. Ebert, J.-P. and A. Willig, *A Gilbert-Elliot bit error model and the efficient use in packet level simulation*. Report, TKN-99-002, Technical University of Berlin, 1999.
- [46]. Zhang, J., T. Yan, and S.H. Son, *Deployment strategies for differentiated detection in wireless sensor networks*. Vol. 1. 2006: IEEE.
- [47]. Román, R., et al., *Trust and reputation systems for wireless sensor networks*. Security and Privacy in Mobile and Wireless Networking, 2009: p. 105-128.
- [48]. Maisel, L. and L.S. Maisel, *Probability, statistics and random processes*1971: Simon and Schuster.

# Quality of Service Aware Chain-Cluster-Based Data Dissemination<sup>1</sup>

---

Data dissemination is a basic building block for wireless sensor networks in order to collect data from the environment and to deliver it to a destination. For some applications, dissemination data could be more effective if a number of quality of services which are specified by the application, are guaranteed. Basically, ensuring quality of services has been put forward as an essential consideration for wireless sensor networks which are often deployed in unattended and open environments and are characterized by their limited resources. In this chapter, we target the contribution of three quality of service parameters namely lifetime, reliability, and delay. Although various data dissemination protocols have been proposed over the years, they mostly put emphasis on the efficient energy consumption and rely on an over-optimistic assumption that wireless links are reliable. Less studies so far worry about the end-to-end transmission reliability especially for a chain-based network architecture. Reliable data dissemination is usually performed by using error control protocols which are expensive in communication and computation costs. Different from traditional data dissemination approaches, in this chapter we concentrate on the relative positions of the sensor nodes in a chain with regards to the leader and other nodes in order to ensure the quality of services. In this regard, our proposed QoS-ACA targets finding such leader node in a chain, which results in **(i)** the sufficient end-to-end transmission reliability without the need of using any error control protocol, **(ii)** a well-balanced energy

---

<sup>1</sup> This chapter is based on the following publications:

- (i) *A reliable and energy-efficient chain-cluster based routing protocol for wireless sensor networks*. In Proceeding of The Eighth IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2013).
- (ii) *QoS-aware chain-based data aggregation in cooperating vehicular communication networks and wireless sensor networks*. In the Book of Roadside Networks for Vehicular Communications: Architectures, Applications, and Test Fields.

consumption, and (iii) the shortest delay. Furthermore, dependent on the network density, QoS-ACA ensures reliability in different ways. Moreover, we propose REC and REC+ solutions which make the size/shape of the clusters in QoS-ACA more adaptive regarding the state of the nodes and links. The main concern of REC/REC+ is building clusters and setting the boundaries of the clusters in an adaptive and dynamic way subject to the application level quality of service constrains. Our proposed approaches are validated through a set of simulations over a number of performance metrics for different scenarios.

### 3.1. Introduction

Data dissemination in wireless sensor networks is the process using which monitored information or queries are distributed in the wireless sensor networks [1, 2]. Therefore, designing an effective data dissemination to transmit periodic or on-demand data from all or a set of sensor nodes is of prime importance for wireless sensor networks.

Data dissemination protocols for wireless sensor networks need to fulfill a number of requirements.

1. First, they should meet the lifetime related goal by well-balancing the workload throughout the network. Lifetime is the most crucial requirement of data dissemination in particular for the applications whose sensor nodes are deployed in an environments that replacing or charging battery is not easily possible once the battery is drained out. On the other hand, having uneven workload in a wireless sensor network could lead to hotspots and premature energy exhausting on some nodes. The challenge of unbalanced energy consumption for sensor nodes has to be considered while designing the protocol [3]. In this regard, it is imperative to effectively and evenly consume the energy of sensor nodes especially for the applications that require to have all nodes alive simultaneously.
2. Although saving energy is a significant goal in wireless sensor networks, saving energy alone does not lead to a system that could fulfill application goals. The wireless data dissemination protocol should meet a minimum reliability requirement. Dependent on the application mission, different quality levels for the provided reliability may be demanded. Although reliable transmission is of importance for almost all applications, decision making applications such as event detections need high degree of transmission reliability, as events detected locally in the network or critical data should certainly be received by the base station(s). Missing data can in this type of applications trigger expensive alarm conditions. On the other hand,

### 3.1 Introduction

---

periodic monitoring applications can better tolerate missing data which can e.g. be extrapolated by using either past data or other nodes' data. In this sense, they carry less important data than event messages and hence the required reliability for disseminated data would be lower. Reliable data dissemination is traditionally guaranteed by applying error control approaches, which could provide an adequate degree of quality even in the presence of errors. There are two key strategies in wireless sensor networks for maintaining reliable communication: Forward Error Correction (FEC) [4] and Automatic Repeat Request (ARQ) [5]. Forward error correction relies on transmission of redundant data in order to allow the receiver node to reconstruct the original data. Automatic repeat request relies on retransmitting a packet which has been missed or received erroneously. The main shortcoming of error control protocols is that they have higher energy consumption and longer delay. It is therefore energy inefficient to equip the uncritical data of the periodic monitoring applications with the error control schemes. According to above, using existing data dissemination protocols some of which will be discussed later, achieving reliability and energy efficiency at the same time is by no means trivial. Due to end-to-end transmission reliability requirement, it is thus necessary to investigate how reliability can be provided in an energy efficient way.

3. Third, in many applications it is desirable to have fresh-enough data updates and hence stale data may be less useful or even useless. Although multi-hop short-range data transmission in the chain-based data dissemination protocols could bring about saving energy, this benefit usually comes at the expense of a longer delay which influences the freshness of the reported data. In this sense, by favoring energy efficiency using multi-hop communication, the freshness of received data at the destination may be affected.

In a chain-based network topology, chain leader should usually handle a huge amount of traffic received from two sides of the chain. Therefore, the location of the leader in the chain and thereby the situation of the nearby area of the leader is very important for the quality of the received traffic in the center. To this end and to address the three above requirements, we concentrate on the relative positions of the nodes in a chain with regards to the leader and other nodes. We hence target finding such leader node in a chain, which could bring about (i) the sufficient end-to-end transmission reliability in a multi-hop network without the need of using any error control technique, (ii) a well-balanced energy consumption to avoid creating holes in the network and (iii) a short delay to keep reported data as fresh as possible. These requirements are typically in trade-off.

### 3.1.1 The need for aggregation-aware data dissemination

In wireless sensor networks, it is very expensive in terms of communication for the sensor nodes to send all their raw data to the base station very frequently. The raw data transmission from every sensor node to the base station will deplete wireless sensor nodes' limited energy, which consequently may influence the quality and quantity of their measurements. On the other hand, as sensor nodes usually monitor common phenomena, spatial correlation in sensor data bring about significant data redundancy. Therefore, in wireless sensor networks data generated by different sensors can be jointly processed while being forwarded toward the sink to contribute to energy saving. In this way, use of data aggregation techniques within the network, out of which any pattern can be detected locally, not only helps save energy by communicating less data but also provides meaningful information to the end-users and prevents them from being flooded with huge amounts of data.

Data aggregation is a simple type of in-network processing which combines data from different sources or nodes into a single entity. According to [6], "in-network aggregation is the global process of gathering and routing information through a multi-hop network, processing data at intermediate nodes with the objective of reducing energy consumption, thereby increasing network lifetime". Although data aggregation has significant impact on energy consumption and overall network efficiency, data size reduction through in-network processing should not diminish required granularity of information about the monitored area.

Data aggregation techniques are closely related to the way data is gathered at the sensor nodes as well as how packets are disseminated through the network. Therefore, in wireless sensor networks aggregation techniques and dissemination protocols are not independent, rather interdependent. Disseminating protocol design takes into consideration the targeted data aggregation at some nodes and accordingly decides packet routing mechanism. Similarly, while designing aggregation techniques the routing or disseminating protocol used underneath plays a vital role.

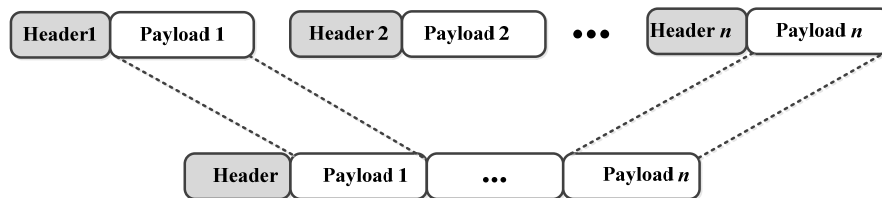
Generally speaking, use of data aggregation approaches in wireless sensor networks offers (i) reducing energy consumption, (ii) eliminating overheads of the redundant packets, (iii) decreasing total load of the network, and (iv) making a meaningful data for the end user exploiting smart filtering of data [7]. The main shortcomings of the data aggregation techniques may include: (i) imposing extra data delay, (ii) decreasing accuracy and data integrity in case of inappropriate aggregation, (iii) the need to handle duplicated data in case of having a complex aggregation function, and (iv) the need for proper coding [7].

### 3.1 Introduction

---

Data aggregation is performed using one of the following approaches:

- Concatenating the payloads of the packets: In this case, as illustrated in Figure 3.1, only the headers of packets are merged and the payload parts are concatenated. This type of aggregation results in a bigger packet size compared with the several original small packets. Combining multiple packets into one eliminates the number of communications and the cost associated with packet overhead. This type of aggregation is especially effective in sensor networks where individual sensor readings are small in size, leaving much room for concatenation. It can also be employed in case of having different type of measurements (i.e. having heterogeneous sensors).
- Combining the payloads of packets: In this case, both headers and payload are merged individually so the size of the resultant packet remains the same as the original packets. Basically, in this approach a number of data packets are passed through an aggregation function that generates a single packet which may have not any information about the original data. Thus, at each intermediate node, the amount of outgoing data is typically much lower than the amount inputted.



**Figure 3.1. Data aggregation (concatenation type)**

While routing the data from sensor nodes to a base station, aggregation can be done either on some specific sensor nodes or on every sensor node. In case of not opting for aggregation on every node, having fixed aggregators should be avoided. This is due to the fact that these fixed aggregators may become single points of failures. Moreover, load balancing and thus prolonging lifetime cannot well be provided. Therefore, instead of having fixed aggregators, it is better to select nodes that satisfy best the necessary criteria as aggregator. By doing so, the role of aggregator can be assigned to different nodes in different time intervals.

Our protocols in this chapter tries best to utilize data aggregation on both cluster heads and intermediate nodes along the path toward the destination, in order to exploit the benefits of aggregation.

This chapter consists of two major parts:

- The first part is related to QoS-ACA which is a quality-aware chain-based data dissemination protocol for multi-hop wireless sensor networks whose sensor nodes are deployed in a (i) sparse or (ii) dense manner. Basically, dependent on the deployment density we may require to employ different techniques to ensure quality of service. The main concern in this part is selecting the most appropriate sensor node as the cluster head.
- The second part is related to two enhanced versions of QoS-ACA which are called REC and REC+. The main concern in this part is quality of service aware dynamic cluster building and setting cluster boundaries in an effective and adaptive manner subject to application level performance constraints. By adaptive we mean that the shape/size of the clusters should be changed according to the conditions which may influence the quality of performance targets for the specific application.

The rest of this chapter is organized as follows. First we describe some related works in Section 3.2, which is followed by the problem statement and our contribution in Section 3.3. Then in Section 3.4, we describe the assumptions and models used. We elaborate on evaluating three quality of service parameters which are of importance for data dissemination in wireless sensor networks in Section 3.5. We expand on the proposed QoS-ACA protocol in Section 3.6. Then in Section 3.7 we present our simulation results for QoS-ACA. We explain REC and REC+ in Section 3.8 which is followed by evaluating the performance of REC/REC+ in Section 3.9. Finally we present a summary in Section 3.10.

### 3.2. Related work

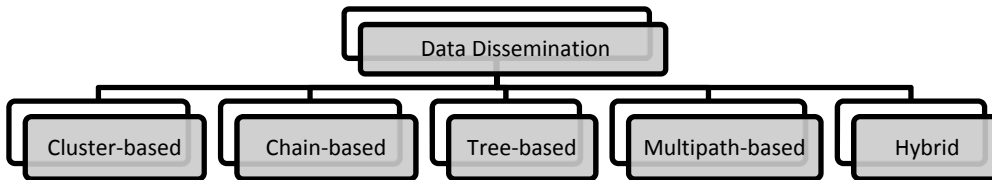
The design of effective data dissemination techniques for wireless sensor networks is a challenging task, on which extensive research has been performed in recent years. The functioning of data dissemination and routing protocols depends heavily on the network topology, which may vary for different applications. Since topology of the network plays a vital role in the performance of data dissemination techniques, we categorize existing data dissemination techniques based on their supported topology and present an architecture-based taxonomy. In the following sections, we first briefly discuss each class of data disseminating protocols separately by reviewing the main concepts and briefly commenting main advantages and limitations of each scheme. Later, we survey a number of cluster-based and chain-based data aggregation techniques, as these are the most relevant to the topic of this chapter.

## 3.2 Related work

---

### 3.2.1 Data dissemination taxonomy

Regarding the supported topology, existing data dissemination techniques for wireless sensor networks can be categorized as shown in Figure 3.2 into cluster-based, chain-based, tree-based, multi-path based, and hybrid methods.



**Figure 3.2. Data dissemination taxonomy**

Cluster-based data dissemination techniques [8, 9, 10, 11] organize nodes into a number of clusters and some local cluster heads. The cluster head may aggregate data from sensor nodes of its cluster and transmits the aggregated data to a base station. The cluster heads can communicate with the base station directly via long range communication or via multi-hops through other cluster heads.

Chain-based data dissemination techniques [12, 13, 14, 15] build a linear chain connecting all nodes of a network and enable each node to transmit the data only to its closest neighboring node along the chain. Eventually, the leader node whose role is similar to the cluster-head in cluster-based approaches, transmits the (aggregated) data to a base station. Effectiveness of chain-based data dissemination techniques depends heavily on the construction of an effective chain. Compared to cluster-based techniques in which all nodes should usually communicate directly with the cluster-head or the base station, chain-based techniques reduce excessive energy consumption for communicating with the base station if the base station is located far from the cluster heads. They typically aim to choose a path with a large number of small-range hops, since they consume less power than an alternative route that has a smaller number of hops, but consider a larger distance for individual hops. In short, chain-based schemes better balance the energy consumption among different nodes.

Tree-based data dissemination techniques [16, 17] organize nodes into a tree structure, where data aggregation could be performed in intermediate nodes along the tree. Finally, the root node whose role is similar to the cluster-head in cluster-based approaches, transmits the aggregated data to a base station. Compared to chain-based techniques, tree-based techniques



allow parallel aggregation and shorten the routing delay. However, the maintenance of the tree needs many control messages, which lead to a high volume traffic.

Multi-path based data dissemination techniques [18, 19, 20] build multiple paths among sensor nodes for data transmission and data aggregation. In this technique, multiple copies of the packet are sent along the multiple paths from source to destination to make possible data delivery with the desired reliability. In this way, it is quite possible that multiple copies of one packet is received by the destination, which influences the duplicate-sensitive aggregation functions such as SUM, COUNT, AVERAGE. Therefore, sensor nodes should have this capability to detect and discard the duplicate values in order to provide correct values. Multi-path based techniques improve the robustness, and gain maximum data accuracy and reliability in presence of link or node failure. However, they have higher energy consumption and generate more data traffic to maintain the multi-paths.

Hybrid data dissemination techniques [21, 22, 23, 24] utilize a combination of different dissemination techniques and architectures in order to take the benefit of different data dissemination in various conditions.

#### 3.2.2 Chain-based data dissemination

There is not much work on data dissemination in the chain-based network architecture. A very first chain-based data aggregation technique is PEGASIS [23]. In PEGASIS, nodes are organized into a linear chain using a greedy algorithm. The greedy chain formation assumes that all nodes have global knowledge about the network. The furthest node from the base station initiates chain formation and at each step, the closest neighbor of a node is selected as its successor in the chain. In other words, in the greedy chain construction algorithm proposed in PEGASIS, the process starts with the furthest node from the sink. This node is the head of the chain. At each step, a non-chain node which is the closest to the chain head is selected and appended to the chain as the new head. The procedure is repeated until all nodes are in the chain. In each data gathering round, a node receives data from one of its neighbors, aggregates the data with its own, and transmits the aggregated data to its other direct neighbors along the chain. Finally, the leader node transmits the aggregated data to the base station. PEGASIS effectively reduces the energy consumption in communication since the distances through which they should transmit their data are shorter compared to cluster-based techniques, especially when the cluster heads are far away from the sensor nodes. However, the main disadvantage of PEGASIS is the strong assumption that all nodes have global knowledge about the entire network to pick suitable neighbors and minimize the neighbor distance. Additionally, due to the fact that only one leader exists in the entire network,

### 3.2 Related work

---

PEGASIS causes an excessive delay for the nodes at the end of the chain, which are far away from the base station. To alleviate the excessive delay in PEGASIS, two chain construction techniques, namely COSEN and MSC, were presented in [14, 15]. COSEN [15] is a hierarchical chain-based technique, in which sensor nodes are organized into one higher level chain and several lower level chains. In every lower level chain, nodes transmit the data to a chain leader, which is elected based on the residual energy. Then all lower level leaders send the aggregated data to a higher level leader, which has the highest energy among all lower level leaders. Eventually, the higher level leader performs final data aggregation and forwards the result to the base station. Compared to PEGASIS, COSEN has much shorter delay because of using multiple chains and hierarchical structure to aggregate and route data,

Authors in [14] propose an algorithm to build a chain by leveraging Minimal Spanning Chain (MSC). This technique aims to enhance PEGASIS by reducing the chain length in order to shorten the delay. This algorithm has two main steps, i.e., (i) configuring an initial chain, and (ii) link exchange. The first step uses Kruskal Minimal Spanning Tree [25] algorithm with maximum degree of two in order to make an initial chain. They verified that the Kruskal algorithm performs better than Prime algorithm, which PEGASIS employs. The rationale behind this claim is that the Kruskal algorithm selects the minimum link among more links. The second step involves reducing the total chain length considering links cost and by avoiding link crossing which makes a chain longer and influences delay, energy consumption and lifetime. The proposed technique does not exhibit any crossing links which PEGASIS has many.

The two above chain construction techniques [14, 15] effectively alleviate the excessive delay in PEGASIS, however, they have the same energy consumption for chain construction. To develop an energy efficient chain construction algorithm, [12] proposes a multiple-chain technique that uses a sequence of insertions to add the least amount of energy consumption to the whole chain. The multiple chain technique divides the whole network into four regions centered at the node that is the closest to the center of the sensing region. For each region, a linear chain is constructed which ends at the center node. The multiple chain technique aims to decrease the total transmission distance for all-to-all broadcasting.

Almost all aforementioned chain-based data dissemination algorithms do not take the reliability issue of the wireless links into account and assume wireless links are reliable or error-free. Due to end-to-end reliability requirement of many applications, it is essential to study how such reliability can be achieved. Reliable data dissemination in wireless sensor networks is traditionally performed by applying error control approaches which could provide an adequate degree of quality even in the presence of errors. There are two key error

control strategies in wireless sensor network for maintaining reliable communication over unreliable channels. The first one is Forward Error Correction (FEC) [1], which relies on transmission of redundant data to make the receiver node capable of reconstructing the original messages. The second strategy is Automatic Repeat Request (ARQ) [2], in which high rate detection codes are usually used and a retransmission is requested if the received data is found to be erroneous. Error control approaches introduce redundancy into an information sequence by adding extra parity bits/packets. Typically, using error control protocols would be more advantageous if the application requires high degree of reliability in the presence of unreliable links. However, there are some monitoring applications which prefer to have high transmission reliability, but not at the expense of using error control schemes which are based upon data redundancy. For this kind of applications, we should provide the highest reliability that could be achieved for the given deployment if no error control means are utilized. Moreover, due to the fact that a leader may become a single point of failure, a robust technique should not have just one special leader all the time. To overcome this issue, we could assign the role of leader to such nodes which could well meet the required criteria that one leader should have. In order to address other mentioned issues, we could express the required criteria for selecting a leader which ensures the highest reliability that could be obtained without using any error control approach, increases lifetime and shortens the end-to-end delay. This is the very contribution of our technique.

#### 3.2.3 Cluster-based data dissemination

LEACH [10] is one of the most well-known distributed clustering algorithms for WSNs. The operation of LEACH is organized into rounds each of which starts with a set-up phase followed by a steady-state phase. Each node has a certain probability of becoming a cluster head per round and the task of being cluster head is almost rotated among nodes. The cluster head then creates a TDMA schedule to avoid intra-cluster collisions and asks subordinated nodes to send their data within their assigned time slots. In the steady state, each cluster head sends its one hop neighbor base station, an aggregated packet including the related sensor nodes' data. Data collection in LEACH is centralized and periodically. The drawback of LEACH is that the dynamic cluster construction results in a huge overhead that increases the network energy consumption. Moreover, it is based on 1-hop communication and assumes each node can communicate directly with the base station. LEACH offers no guarantee about the placement and number of cluster heads due to lack of global coordination among nodes. However, using a central control algorithm to form the clusters may produce better clusters through well distribution of the cluster heads throughout the network. This is the key idea of LEACH-C [26], which uses a centralized clustering algorithm and has the same steady-state

### 3.2 Related work

---

phase as LEACH. In the set-up phase, each node sends information about its current location and energy level to the base station. Base station utilizes this information to introduce better clusters that require less energy for data transmission per round and then broadcasts the clustering information to the network. By doing so, it allows each node to know its related cluster head. Then, each node determines its TDMA slot for data transmission and goes to sleep until when is scheduled to transmit its data. Compared with LEACH, LEACH-C is more robust since it selects clusters and cluster heads based on location information and residual energy of the nodes. However, it still has high communication cost because of frequent communication between the base station and the nodes to exchange data.

Multi-hop-LEACH protocol [27] is a cluster-based protocol similar to LEACH. In order to transmit the data from one cluster head to the base station, it selects the optimal path between the cluster head and the base station through other cluster heads.

In LEACH-CC [28] which is a centralized low energy-consumption chain-based routing protocol, every node should send information about its current location and residual energy to the base station. Base station runs the simulated annealing algorithm to determine the cluster and cluster heads at the beginning of each round. A chain routing is established between clusters in order to reduce the amount of nodes directly communicate with the base station. LEACH-CC also use only one cluster head to transmit the data of all cluster heads to the base station.

CCM [29] is mainly a hybrid of LEACH and PEGASIS. CCM divides the networks into several chains and selects one node in each chain as the leader and then forms a cluster from all chain leaders and the most powerful leader is selected as the cluster head to transmit data directly to the base station. The consumed energy of CCM increases as network size grows. CCBRP [30] is a two layers chain-based protocol, which divides a wireless sensor networks into a number of chains exploiting greedy algorithm of PEGASIS and selects a random leader for each chain. Then, it makes a chain of all leaders and selects one of them randomly as the global leader being responsible to transmit the aggregated data to the base station. CCBRP outperforms LEACH, PEGASIS and CCM with respect to the product of the energy consumption and the experienced delay.

CCRP [31] is another LEACH based protocol that adopts a more balanced cluster head selection algorithm and an improved data transmission mechanism from the cluster head to the base station. The cluster head selection of CCRP is done similar to LEACH but the residual energy of the node and the number of neighbors are also considered for calculating the threshold of a given node. CCRP also forms a chain of cluster heads, which is constructed in such a way that each cluster head receives from and transmits to an adjacent

cluster head and the leader of this chain is the closest cluster head to the base station. The aggregated data of each cluster head is transmitted along this chain until it is received by the base station. CCRP supports reliability by dynamically adjusting the transmission power according to the receiver noise condition and the distance between transmitter and receiver to ensure the required packet delivery rate.

Most of the aforementioned approaches assume all nodes depending on the need can reach the base station with enough power. Moreover, all of them rather aim to achieve an energy-efficient clustering using some criteria for cluster formation or cluster head selection and maintenance. They also assume that nodes and data transmission are quite reliable and packets are not lost due to low links' quality among other factors. Lack of good connectivity among cluster heads or their instability force them to accomplish retransmission which may introduce significant energy consumption, delay, and overall network performance degradation. Some approaches enhance the existing cluster-based data collection protocols and address packet loss caused by the cluster head failures via introducing back up cluster heads [32, 33]. Generally, they try to select one or an optimal set of back up cluster heads, which takes over the current cluster head responsibility when needed.

Gupta et al [34] suggest using a few powerful gateway nodes to organize sensor nodes into clusters. However, because these gateway nodes take a critical role in the network, the system is more sensitive to their failure. This method assumes a heterogeneous sensor network that employs specialized cluster head gateways and cannot be applicable to those cluster-based protocols, in which the role of cluster heads is rotated.

BARC [35] aims to provide reliability and prolong lifetime together. It addresses lifetime and load balancing by assigning the cluster head role to the proper node according to a battery recovery scheme. BARC exploits the trust concept to increase reliability by letting each node joins the cluster whose cluster head is trustworthy. BARC assumes the sensor nodes are equipped with some monitoring mechanisms or intrusion detection units, which makes them capable of observing the behavior of its d-hop neighbors to update their trust parameters. Moreover, it cannot handle packet loss caused by unreliable links.

### **3.3. Problem statement and our contribution**

The problem we deal with in this chapter is to find a well-balanced quality of service aware approach to deliver data packets gathered by the sensor nodes to the base station respecting application requirements. We address three quality of service parameters, i.e., (i) lifetime, (ii) reliability, and (iii) data freshness or delay. In other words, we investigate an

### 3.4 Assumptions and models used

---

energy-balanced, reliable and fast data dissemination scheme to deliver data packets gathered by the sensor nodes to a base station. By energy-balanced, we mean to take the residual energy of the sensor nodes into account when we are assigning a task to a specific node. This could be more beneficial if the application requires to have all sensor nodes alive simultaneously. In this case, death of any node creates a hole in the network that may hinder the network operations. By data freshness we mean to deliver data packet with short delay.

In a chain-based topology, in which sensor nodes that are not leader can only communicate with their adjacent left and right neighbors, routing is not very complicated. Therefore, we mostly concentrate on the chain leader election algorithm instead of routing.

We summarize our contribution related to this chapter as follows:

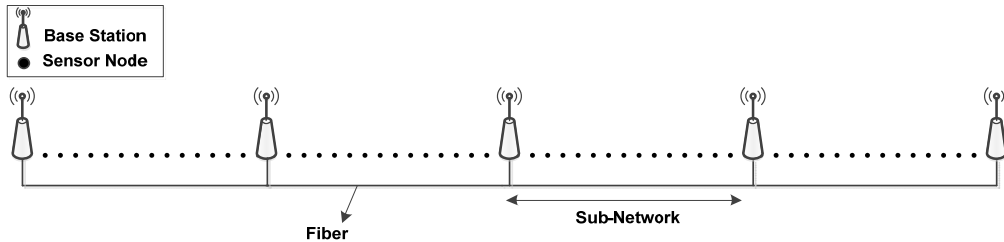
1. Introducing a two-tier architecture model in order to reliably and fast aggregate and disseminate sensed data toward the base station while the lifetime is prolonged.
2. Integrating three quality of service parameters (i.e., lifetime, reliability, delay) with the possibility to adjust their priorities according to the specific application requirements, in order to find the most proper nodes as the chain leaders in both tiers.
3. Reliably collecting sensor nodes' data without using any error control approach. Two algorithms will be proposed in order to guarantee maximum transmission reliability for both densely-deployed and sparsely-deployed networks.
4. Proposing a quality of service aware dynamic cluster formation and cluster boundaries setting in an effective and adaptive manner subject to application level performance constraints. Most importantly, our proposed approach relaxes some strong assumptions (such as the ability of nodes to communicate directly with other nodes or with the base station) that other existing techniques rely on.

### 3.4. Assumptions and models used

We make the following assumptions regarding the wireless sensor network:

- As illustrated in Figure 3.3, the wireless sensor network consists of  $N$  sensor nodes uniformly and randomly deployed in a linear topology.
- The network is divided into some sub-networks each of which is located between two base stations.

- The base stations are intelligent gateway nodes and are equipped with high computation power. They also have a high speed, e.g. fiber, communication link in order to talk with each other.



**Figure 3.3. Network deployment**

- Each sub-network is divided into some equal-size chains/clusters each of which has  $N_C$  sensor nodes (Figure 3.3).
- The nodes in each chain is able to directly talk with the nodes of adjacent chain(s).
- Chain leaders could adjust their power level in order to reach one of two base stations.
- In case of not being a chain leader, sensor nodes can only communicate with their adjacent (i.e. one-hop) upstream/downstream neighbors.
- The location of sensor nodes and the base stations are fixed and are known *a priori*.
- Aggregation is performed along the path from each node towards the base station.

As each cluster contains one chain, from now on we use the terms chain and cluster interchangeably through this chapter. Our approach can be applied for each sub-network independently. Therefore, without loss of generality, from now on we refer to a sub-network as our given network.

#### 3.4.1 Two-tiers architecture model

We consider a static cluster-based wireless sensor network and a two-tiers architecture model, as illustrated in Figure 3.4. Every sensor node in a cluster must send its data to its upstream neighbor which has been selected in the chain construction phase. Within each cluster, intermediate nodes along the path to the cluster head aggregate the data received from the downstream nodes with their own data and forward the local aggregated value towards to the cluster head or chain leader. The leader should perform final aggregation on

### 3.5 Quality of service aware cluster head selection

the data received from two sides of the chain and send the results to the base station through the overlay networks of chain leaders which construct the second-tier.

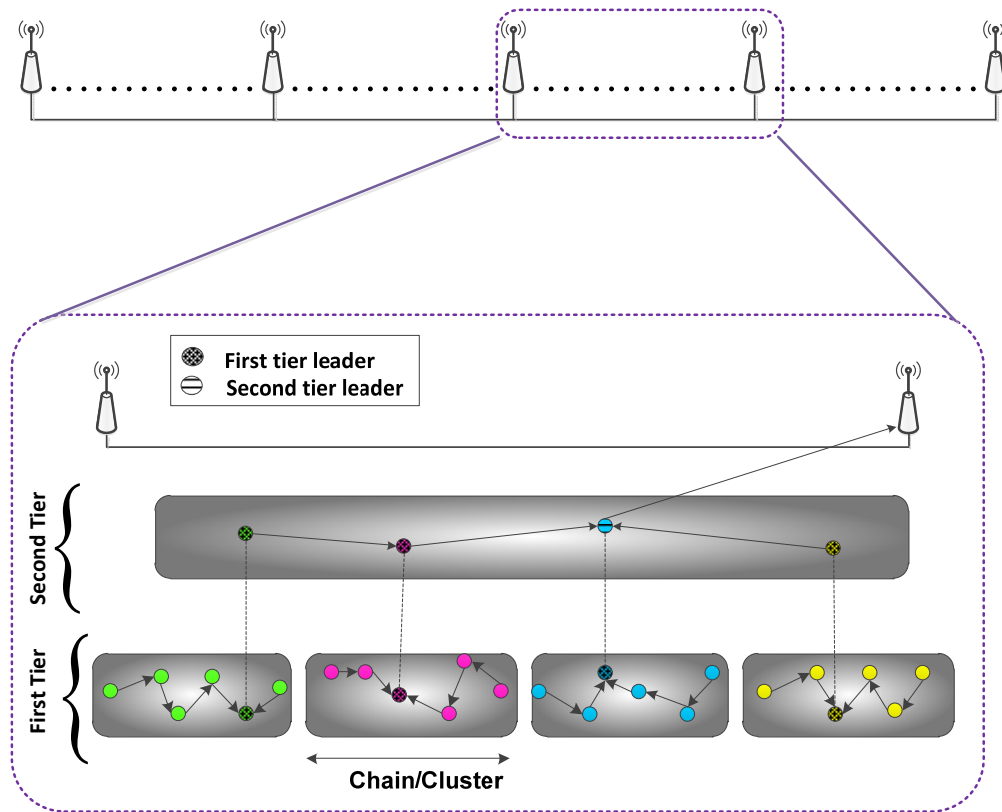


Figure 3.4. Two-tier architecture model

### 3.5. Quality of service aware cluster head selection

In a linear topology routing is almost straightforward. In our data dissemination approach we have to select a leader which can best guarantee the quality of services requested by the application. To this end, we need to assess the utility of each node in terms of different quality of service parameters to determine how its selection will impact the requested quality of service parameters.



We consider three quality of service parameters, i.e., lifetime, reliability and delay/freshness. It is worth mentioning that dependent on the application, different priorities among these three quality of service parameters may be demanded.

Single-hop or direct long-range communication usually demands substantial transmission power. In wireless communication, the amount of energy used to transmit a signal usually grows exponentially proportional to the traversed distance of the signal. Although the single-hop long range communication have a tendency to be energy-exhaustive, but it imposes shorter end-to-end delay. In contrary, multi-hop short-range communications are helpful in reducing transmission power but they would increase the end-to-end delay because of an increase in the number of hops. Therefore, short-range hop-by-hop communication is preferred over direct long-range communication to the destination, in terms of energy consumption.

There are different definitions for lifetime of a wireless sensor network. In fact, lifetime is an application-dependent concept. There are some applications consider lifetime to be the time at which the first node dies, while others consider lifetime to be the time at which the last node dies. We consider our lifetime with a system-wide impact and thus opt for load-balancing and evenly assign tasks to the sensor nodes by considering their residual energy.

End-to-end delay is an important quality of service parameter in multi-hop wireless sensor networks and refers to the total amount of time a packet takes to reach the destination. Although multi-hop short-range data transmission could bring about saving energy specially for the nodes located far from the base station, this benefit usually comes at the expense of a longer delay.

Basically, end-to-end delay is a major metric for quality of service, which is a joint effect of both data rate and end-to-end latency. In case of sending a small data packet, the dominant factor is end-to-end latency, however, in transferring a large data or even a large file the dominant factor is data rate [36]. Therefore, in a typical wireless sensor network that sensor nodes usually generate small packets, the end-to-end latency plays a more significant role.

#### **3.5.1 Transmission reliability**

Depending on the network density, the position of the cluster-head/chain-leader and typical sensor nodes may have different effect on transmission reliability.

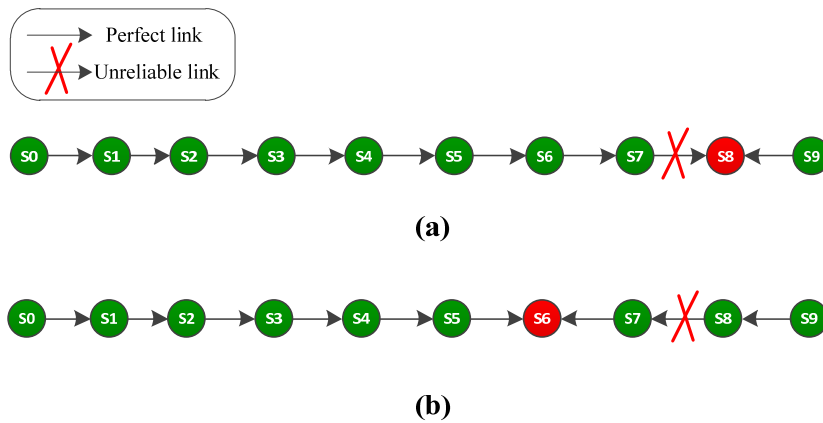
### 3.5 Quality of service aware cluster head selection

#### 3.5.1.1 Transmission reliability in dense chain-based networks

Generally speaking, the main goal of wireless sensor network applications is detecting or estimating phenomenon features from the captured data provided by the sensor nodes with the highest reliability. In a densely deployed network where data of the close nodes are usually correlated, the application is interested in estimating the condition of an area according to the observation of the sensor nodes located in the given area. The more reported data and higher reliability for the given data captured from densely deployed sensor nodes, the more accurate observation of the physical phenomenon in the base station.

Having a chain whose sensor nodes are densely deployed, in this section we put forward a mechanism which aims to collect as many data as possible from that chain while guarantees the highest transmission reliability. To this end, we emphasize on directing as much network traffic as possible toward the high reliable links of the chain to provide better reliability.

To motivate the need of our proposed approach let us consider the chain illustrated in Figure 3.5 where all links experience a perfect situation in terms of reliability, except the link between  $S_7$  and  $S_8$  which is unreliable.



**Figure 3.5. One chain with different chain leaders (a)  $S_8$  is leader (b)  $S_6$  is leader**

According to Figure 3.5, if  $S_8$  is selected as the chain leader, all the traffic received from the left side of the chain will be affected by this unreliable link even though they all reliably received till one hop before in  $S_7$ . The only reliable data will be reported by  $S_9$ . Instead, if  $S_6$  is selected as the chain leader, all data reported by the node on the left side of the chain as well as  $S_7$  will be reliably received and only data reported by  $S_8$  and  $S_9$  will be influenced by

the unreliable link. This Scenario would be more challenging if all links suffer from different amount of unreliability. In this case, we should select such node to undertake leader role that could provide the highest amount of cumulative reliability for the gathered data from the chain.

To be able to find the best leader in terms of reliability in a chain, first we should calculate the end-to-end transmission reliability  $\rho_{i,l}^e$  that each designated leader  $l$  provides for each sensor node  $i$  of the chain using Equation (3.1).

$$\rho_{i,l}^e = \begin{cases} \prod_{j=i}^{l-1} \rho_{j,j+1}^h & l > i \\ \prod_{j=l-1}^i \rho_{j+1,j}^h & l < i \end{cases} \quad (3.1)$$

According to Equation (3.1), the end-to-end transmission reliability could be derived by the product of the hop-by-hop reliability  $\rho^h$  of the sensor nodes along the path to the given candidate leader.

After finding the end-to-end transmission reliability  $\rho^e$  of all nodes to the designated leader, we should evaluate the utility  $U$  of the given designated leader  $l$  for the chain  $c$  in terms of end-to-end reliability. Therefore, we employ Equation (3.2) in order to find the cumulative end-to-end reliabilities that node  $l$  provides for the chain  $c$ .

$$U_c^{\rho^e}(l|x = dense) = \sum_{i=1, i \neq l}^{n_c} \rho_{i,l}^e \quad (3.2)$$

where  $n_c$  represents the total number of nodes in the chain  $c$ .

The best leader node  $L^c$  for chain  $c$  finally will be chosen based on Equation (3.3) which implies that  $L^c$  is such node that introduces the maximum utility in terms of end-to-end transmission reliability for the chain.

$$L^c = \{j \in S^c | U_c^{\rho^e}(j | x = dense) \geq U_c^{\rho^e}(i, x = dense), \forall i \in S^c\} \quad (3.3)$$

where  $S^c$  is the set of nodes in the chain  $c$ .

After finding all the utility values in a chain, base station selects the sensor node which provides the maximum benefit in the role of the leader for a given chain. This selection ensures the maximum reliability that this chain can provide.

### 3.5 Quality of service aware cluster head selection

---

#### 3.5.1.2 Transmission reliability in sparse chain-based networks

In the previous subsection, we assume a dense network for which we maximize transmission reliability. In that type of deployment, due to high spatial correlation among reported data of sensor nodes we maximize the sum of the end-to-end reliabilities irrespective of the contribution of *each sensor node* to the aggregated value received by the cluster heads or base station. In this sense, it will be likely that sometimes data is either completely lost or is unreliably received from the nodes located nearby unreliable links. In this case, no or less contribution of these nodes for that time instant will be recorded. However, thanks to temporal and spatial data correlation the data of that node is likely to be extrapolated by using either past data or other nodes' data.

There exist many applications in wireless sensor networks, especially those with sparse deployed nodes, requiring contribution of all or as many sensor nodes as possible. There is not that much spatial correlation among data of sparsely deployed nodes compared to that of in dense deployment. Therefore, in order to have a good overview of the monitored area, we should reliably receive data of *all* sensor nodes. While the applications of a dense network are typically interested to receive *as many data as possible* from the monitored area irrespective of contribution of each sensor node to the aggregated value, in a sparse network the applications are mostly interested to receive data from *as many sensor node as possible*. In this regard, we put forward an approach which aims to collect as many data as possible from every node in a chain while provides the maximum reliability for the nodes suffering the most from unreliable links. To this end, we evaluate all candidate nodes in the chain against the minimum end-to-end transmission reliability they offer for the specific chain  $c$ .

The utility  $U_c^{\rho^e}(l | x = \textit{sparse})$  of each designated leader  $l$  would be obtained as Equation (3.4).

$$U_c^{\rho^e}(l | x = \textit{sparse}) = \min(\rho_{i,l}^e) \quad \forall i \in S^c \quad (3.4)$$

The sensor nodes with higher end-to-end transmission reliability have higher contribution to the final aggregation value sent from a given chain to the base station. In this way, this minimum value  $U_c^{\rho^e}(l | x = \textit{sparse})$  reflects the contribution of the node with the least end-to-end reliability, to the final aggregated value provided node  $l$  is selected as leader. Therefore, the higher this value, the higher the contribution of the least reliable node of the given chain  $c$ .

After finding the utility of all candidate nodes, we select the nodes set  $L_S$  whose members maximize the minimum end-to-end reliability existed in the chain as (3.5).

$$L_s = \{j \in S^c | U_c^{\rho^e}(j|x = sparse) \geq U_c^{\rho^e}(i|x = sparse), \forall i \in S^c\} \quad (3.5)$$

If  $L_s$  consists of one node, that node will be chosen as the chain leader  $L^c$ . Otherwise, the node which belongs to  $L_s$  and provides higher utility according to Equation (3.2) will be chosen as the chain leader  $L^c$ . Doing so, the node which offers (i) the highest reliability for the data may come from the proximity of unreliable area and (ii) higher cumulative reliability is chosen as the chain leader as shown in Equation (3.6).

$$L^c = \{j \in L_s | U_c^{\rho^e}(j|x = dense) \geq U_c^{\rho^e}(i|x = dense), \forall i \in L_s\} \quad (3.6)$$

Although this approach may bring uncertainty to some extent for the data of nodes nearby perfect links at the same time it lowers down the uncertainty that the nodes nearby unreliable links are suffering from. In other words and with respect to the Equation (3.5), this approach tends to provide fairness in terms of contribution of sensor nodes to the aggregated value and hence, tries to ensure that the aggregated value received by the base station from a chain contains the data of *as many sensor nodes as possible* in that given chain.

### 3.5.2 Lifetime

Data dissemination protocols have significant impacts on lifetime and delay performances. Therefore, choosing an efficient dissemination approach is a multi-objective optimization problem.

To ensure this quality of service metric in selecting a leader, we employ Equation (3.7) to calculate the utility  $U_c^e(l)$  that each designated leader  $l$  provides for chain  $c$  in terms of lifetime.

$$U_c^e(l) = r\varepsilon_l \quad (3.7)$$

where  $r\varepsilon_l$  denotes the residual energy of node  $l$ .

The best leader node  $L^c$  for chain  $c$  finally can be chosen according to (3.8) which implies that  $L^c$  is such node that introduces the maximum utility in terms of lifetime.

$$L^c = \{j \in S^c | U_c^e(j) \geq U_c^e(i), \forall i \in S^c\} \quad (3.8)$$

### 3.5.3 Delay

In a linear topology, the end-to-end delay is heavily dependent on the length of the chain and the relative position of the leader in the given chain. Therefore, the utility associated

### 3.6 A reliability-, energy-, and delay-aware data dissemination in linear topology

---

with a leader should thus reflect not merely the transmission reliability and lifetime, but rather the amount of delay imposed by the designated leader.

According to Figure 3.5, if  $S_8$  is selected as the chain leader then the end-to-end delay (in terms of number of hops) imposed by this leader for this chain would be 8. In this case, even though the data of the node located in the right-side of the leader, i.e.  $S_9$ , is received after one time unit but because of aggregation,  $S_7$ 's data has to wait until  $S_0$ 's data is received. Therefore, the longest delay is related to either the leftmost or rightmost node in a chain. In the same chain, if  $S_4$  or  $S_5$  is selected as the chain leader the imposed delay would be 5 time unit. Therefore, selecting leader as close as possible to the middle of the chain results in shorter delay and thereby fresher data.

The utility  $U_c^\delta(l)$  of each designated leader  $l$  in terms of delay would be expressed as the maximum delay introduced by two sides of the chain as Equation (3.9).

$$U_c^\delta(l) = \frac{1}{\text{Max}(d(l, mr_c), d(l, ml_c))} \quad (3.9)$$

where  $ml_c$  and  $mr_c$  represent the leftmost node and rightmost node in the chain  $c$ , respectively. Moreover,  $d(i, j)$  denotes the distance between node  $i$  and  $j$  in terms of number of hops.

Equation (3.9) implies that the utility of node  $l$  is in reverse proportion to the longest delay imposed by  $l$  for chain  $c$ .

The best leader node  $L^c$  for chain  $c$  finally could be chosen according to (3.10) which indicates that  $L^c$  is such node that introduces the maximum utility in terms of data freshness.

$$L^c = \{j \in S^c | U_c^\delta(j) \geq U_c^\delta(i), \forall i \in S^c\} \quad (3.10)$$

### 3.6. A reliability-, energy-, and delay-aware data dissemination in linear topology

In this section, we present our quality-aware chain-based data dissemination protocol for multi-hop wireless sensor networks.

As we employ a two-tiers architecture model illustrated in Figure 3.4, in what follows we expand on the responsibility of each tier in ensuring the required quality of service for the reported data. The first-tier, so called intra-cluster chain, is composed of several chains each of which is built among sensor nodes inside a cluster. The second-tier is the inter-cluster chain which builds the overlay network and consists of the first-tier chains' leaders.

### 3.6.1 First-tier: intra-cluster chain

Our algorithm to guarantee required quality of service consists of three phases, i.e., (i) initialization, (ii) situation-aware data gathering, and (iii) leader election.

#### 3.6.1.1 Phase I: initialization

As previously mentioned, we consider a wireless sensor network which consists of  $N$  sensor nodes deployed in a linear topology. We divide the deployment area into some equal-size clusters/chains in order to increase the degree of parallelism and to shorten the delay imposed by having a single long chain. Each sensor node should first discover its adjacent neighboring nodes and then, using PEGASIS [23] which is a very first chain construction algorithm, one chain per cluster is formed. The base station selects the most appropriate node in terms of required quality of services using the Equations presented in Section 3.5 to undertake the function of chain leader. As upon deployment the base station does not have enough knowledge about the quality of the links, the criteria based on which a leader is selected is only the delay and energy and not links quality.

The base station thereafter informs all sensor nodes in each chain about their roles (being first-tier leader, second-tier leader, regular sensor node) as well as the cluster they belong to and the direction towards which they should send their sensed data.

#### 3.6.1.2 Phase II: situation-aware data gathering

After initialization phase, sensor nodes in each chain start monitoring the environment and reporting periodically their data towards the chain leader through the intermediate nodes. The intermediate node should receive data from downstream node (if any), aggregate that data with its own sensed one and transmit the aggregated value to the upstream node in the chain. Chain leader upon receiving data from two sides of the chain, should do aggregation on them and send the aggregated value either directly or through the leader of adjacent chains to the base station.

Beside sensing, aggregating and relaying data, each sensor node should also evaluate the transmission quality of its adjacent links. To evaluate link quality we may utilize the communication statistics related to either data packets or injected beacon/control packets which bring extra overhead. In this phase to save more energy, we evaluate the link quality by exploiting the statistics related to transmitted data packets which contain information related to the sensed phenomenon. Therefore, the quality of the links along the data

### 3.6 A reliability-, energy-, and delay-aware data dissemination in linear topology

---

transmission path towards the leader is evaluated using data packets traffic. Any significant change in the transmission quality of a link should be reported to the chain leader. To this end, the adjacent upstream node of a given influenced link sends the statistics about that link quality toward the chain leader. Basically, the statistics about links quality are piggybacked with the data sent from the adjacent nodes. By doing so, the chain leader is aware of the latest states of all links in terms of transmission reliability.

#### 3.6.1.3 Phase III: leader election

Due to resource constraints of the sensor nodes and variable link quality which may affect the transmission reliability, the role of chain leader can be undertaken by different chain members in different time intervals. This strategy of assigning the chain leader role to the chain members can (i) distribute the energy load evenly among the sensor nodes in the network and (ii) improve transmission reliability.

According to the system state if the current leader cannot anymore best satisfy the necessary criteria in terms of quality of services, another leader should be selected. To this end, three quality of service parameters which were discussed in details in the Section 3.5 and are the main criteria based on which we select a leader, will be combined. In this way, by getting help from Equations (3.2), (3.4),(3.7), (3.9) we introduce Equation (3.11) in order to derive the total utility  $U_c^\Delta$  for each node  $l$ , respecting necessary criteria.

$$U_c^\Delta(l) = U_c^{\rho^e}(l|x) \times U_c^\varepsilon(l) \times U_c^\delta(l) \quad (3.11)$$

where  $x$  shows the deployment density and could be set to  $x = dense$  or  $x = sparse$ . The higher the utility value  $U_c^\Delta(l)$ , the higher the probability for node  $l$  to be selected as leader.

As different applications may have different requirements in terms of quality services, we can assign weight to each term of Equation (3.11) and modify it as Equation (3.12). Doing so, the priority among different quality of services could be adjusted.

$$U_c^\Delta(l) = (U_c^{\rho^e}(l|x))^{\omega_\rho} \times (U_c^\varepsilon(l))^{\omega_\varepsilon} \times (U_c^\delta(l))^{\omega_\delta} \quad (3.12)$$

where  $\omega_\rho$ ,  $\omega_\varepsilon$  and  $\omega_\delta$  represent the assigned weights.

Using Equation (3.12) we make different criteria each of which may fit the requirements of a given application. For instance, if lifetime is the major concern for an application and the communication links are quite reliable, we assign higher value to  $\omega_\varepsilon$  compared to  $\omega_\rho$  and  $\omega_\delta$ . For such application we can set  $\omega_\rho = \omega_\delta = 0$  and  $\omega_\varepsilon = 1$ , which reflects that



transmission reliability and delay (or freshness) are not important at all. Instead, if reliability is more important than both lifetime and delay we can set  $\omega_\rho = 2$  and  $\omega_\varepsilon = \omega_\delta = 1$ . Although application-dependent, we can assume that these weights e.g. vary between (0, 2). If a given parameter is not important, we set its weight to 0. If a given parameter is vital, we set its weight to 2. If a given parameter is important but not much, we set its weight to 1. Therefore, these weights ( $\omega_\rho, \omega_\varepsilon, \omega_\delta$ ) can be adjusted according to the application specific knowledge and set to any positive number.

After finding the most appropriate leader according to the required criteria, the previous leader should inform the base station and the chain leaders of adjacent chains about the ID of new leader. The new leader should also inform the sensor nodes whose data flow direction are influenced by this change. For instance, according to Figure 3.5 if the previous leader was  $S_8$  and new leader is  $S_6$ , only the data flow direction of node  $S_7$  is influenced. In this case, the upstream node of  $S_7$  should change from  $S_8$  to  $S_6$  and the downstream node of  $S_7$  should change from  $S_6$  to  $S_8$ . In other words, the role of upstream and downstream nodes of  $S_7$  would be exchanged.

### 3.6.2 Second tier: inter-cluster chain

According to the two-tier architecture model presented in Section 3.4, the second tier is dealing with the chain which is formed among the leaders of the first tier chains. The nodes in the second tier should send their data to the base station in a multi hop fashion. Therefore, in the second tier we should also opt for a leader which is responsible to deliver the final aggregated data of the network to the base station subject to application level performance constraints. Our algorithm to guarantee required quality of services for the second tier also consists of three phases, i.e., (i) initialization, (ii) situation-aware data gathering and disseminating, and (iii) leader election.

#### 3.6.2.1 Phase I: initialization

The base station should inform all the sensor nodes in the second tier about their adjacent upstream and downstream neighbor nodes in order to form the second-tier chain. Then, sensor nodes of this tier should adjust their transmission power in an ideal level to reach the adjacent (i.e. one-hop) upstream neighbor which is specified by the base station. Similar to the first-tier chain, base station selects the most proper node in terms of delay and lifetime as the leader of the second-tier chain. The leader should adjust its transmission power level in such a way that it is capable of communicating directly with the base station while no energy

### 3.6 A reliability-, energy-, and delay-aware data dissemination in linear topology

---

is wasted. The chosen transmission power level depends on the distance between the transmitter and receiver nodes.

#### 3.6.2.2 Phase II: situation-aware data disseminating

This phase of the second-tier is almost similar to the corresponding phase of the first-tier except that sensor nodes do not deal with the raw sensed data. In this tier, sensor nodes which are first-tier chains leaders and have an aggregation of all nodes' data of their chains, should progressively route the aggregated messages to the adjacent upstream node. The intermediate nodes of this tier also do aggregation to effectively reduce the communication traffic and frequency at the tradeoff of computational cost.

Similar to the first tier, any change in the concerned quality of service parameters that may influence the leader position and thereby requires running leader election procedure, should be reported to the base station.

#### 3.6.2.3 Phase III: leader election

In electing a leader for the second-tier chain, the same three quality of service parameters which are used for the first tier chains, are employed. In the second tier, although finding delay and lifetime utilities are similar to those in the first tier, the end-to-end transmission reliability should be evaluated in a different way. As previously mentioned, we exploit the statistics related to the data packets traffic in order to evaluate the hop-by-hop transmission reliabilities between any two adjacent nodes in a single chain. In the second tier, however, these statistics, i.e. transmissions quality, are available only for the nodes involved in the previous second-tier chain. Therefore, hop-by-hop reliabilities and consequently end-to-end reliabilities for any pair of nodes which involve in the up-to-date second tier chain may not be available. In this sense, finding the best second tier node in terms of transmission reliability, could not be obtained from the available statistics. As an alternative, we could evaluate the links quality of the second tier chain by getting help from beacon packets. In this way, before calculating the total utility of each node, all sensor nodes in the second tier chain send out a number of beacon packets using which the transmission reliability parameter could be derived. By doing so and using Equation (3.2) and Equation (3.4), the utility of each second tier nodes in terms of transmission reliability could be easily obtained.

Having lifetime, delay and end-to-end transmission reliability benefits of the nodes in the second tier chain  $c'$ , the total utility of a node  $l'$  could be obtained by Equation (3.13).

$$U_{c'}^{\Delta}(l') = (U_{c'}^{\rho^e}(l'|x))^{\omega_{\rho}} \times (U_{c'}^{\varepsilon}(l'))^{\omega_{\varepsilon}} \times (U_{c'}^{\delta}(l'))^{\omega_{\delta}} \quad (3.13)$$

where all variables and parameters have the same description as mentioned for Equation (3.12).

Similar to the first tier chain, the node who provides the highest total utility will be elected as the second tier leader which is responsible to deliver final aggregated data of the network to the base station. The leader ID should be announced in the second tier chain in order to make all nodes aware of the direction toward which they should send their data.

Upon experiencing any significant change in the links quality or residual energy of the second tier nodes, the leader election may happen again using the up-to-date values of the targeted quality of service parameters.

### 3.7. Performance evaluation of QoS-ACA

In this section we elaborate on performance metrics, simulation setup and evaluation results.

#### 3.7.1 Performance metrics

We consider a number of metrics which are listed below to evaluate the performance of our approaches under different circumstances.

- **Average Node's Contribution:** Generally speaking, this metric represents the amount of contribution each node has in the received packet(s). As we already mentioned in Section 3.5.1, depending on the deployment density, applications may utilize different data dissemination schemes in which different amount of nodes' contribution may be exhibited. Generally speaking, to have a good overview of the monitoring area, applications are mostly interested to have the contribution of every individual node in the aggregated data received by the base station in an aggregation-aware data dissemination scheme. To evaluate this metric we count the number of received packets that each node contributed to over the simulated duration and dividing that by the total number of received aggregated packets. It is worth mentioning that if a given node contributes to a packet, it means that the reported data of the given node is included in the aggregated data of the given packet. Ideally, all received packets should contain the data of all nodes, however, due to link quality issue sometimes the received packets does not include the data of some nodes whose end-to-end reliability are relatively low.

### 3.7 Performance evaluation of QoS-ACA

---

The definition of this metric is slightly different when the data dissemination scheme does not perform aggregation along the path toward the base station. In this case, this metric represents the average ratio of the received packets from a specific node to the total packets sent from that node. The difference between the above two definitions is that, in the aggregation-aware scenario each packet could be counted for several nodes which are contributing to make the aggregated data of the given packet. However, for the latter scenario, each packet should be counted for only one node whose raw data are carrying by the given packet.

- **Average Alive Nodes Ratio:** This metric represents the ratio of the nodes which are alive. It is evaluated at any time by counting the number of alive sensor nodes and dividing that by the total number of sensor nodes. By the means of this metric over simulated duration, we observe how well the energy is balanced among the sensor nodes. As we previously mentioned, balancing energy is very important for some applications which need to have all nodes alive at a time. The less difference between the time when the first death is reported and the time when the last death is reported, leads to a system with a more balanced energy consumption.
- **End-to-end Delay:** This metric well-reflects the freshness quality of the received data and thereby it has control over data freshness. In order to evaluate this metric, we should count the number of hops through which the data of the farthest node(s) in the chains is received by the base station. To this end, first we calculate  $\delta^F$  and  $\delta^S$ , which state the longest delay imposed by the first-tier chain leaders and second-tier chain leader, respectively. Moreover, the delay between chain leader of the second-tier and base station should be accounted. Since we assume that the second-tier leader should send the aggregated data directly to the base station then the later delay would be 1.

Finally, we sum up these three delay in order derive the final end-to-end delay  $\delta^T$  as Equation (3.14) .

$$\delta^T = \delta^F + \delta^S + 1 \quad (3.14)$$

- **Aggregation Gain:** This metric represents the benefit of employing aggregation over the sensed and reported data in terms of communication traffic reduction [37]. This metric could also reflects the energy efficiency in case of using data aggregation. To evaluate this metric, we first calculate  $\varphi_{ag}$  and  $\varphi_{Nag}$  which are the total number of the required transmissions in the network in order to fulfill the given task with and without data aggregation respectively. Thereafter, using Equation (3.15) the aggregation gain  $\Lambda$

could be expressed as the ratio of traffic reduction due to data aggregation, to the total traffic without aggregation.

$$\Lambda = 1 - \frac{\varphi_{ag}}{\varphi_{Nag}} \quad (3.15)$$

#### 3.7.2 Simulation setup and scenarios

We simulate a stationary network with sensor nodes which are uniformly and randomly distributed in an area of  $200m \times 25m$ . Two base stations are located at the leftmost and the rightmost of the deployment area. The sensor nodes are evenly distributed into four equal-size chains each of them consists of the same number of sensor nodes. Doing so, each chain has 25% of the sensor nodes in the network.

At any moment in time, the 30% of the links experience transmission quality between 0.99 and 0.8, and the remaining 70% of the links experience a proper link quality as 1. By assuming this scenario, we aim to consider a network most of its links (i.e. 70%) are reliable and the rest of the links (i.e. 30%) experience different qualities.

The initial energy for each sensor node is 0.5 J. Number of sensor nodes in the network is 40 unless otherwise stated. Our radio model is TI CC2420.

We carry 50 experimental trials, the average results of which are presented in the following graphs.

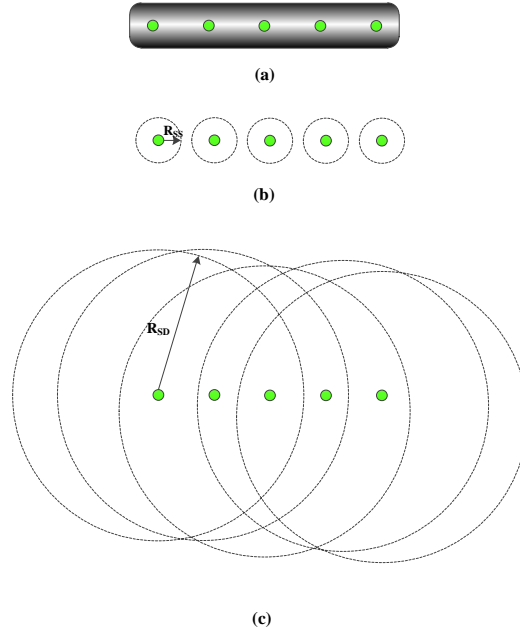
We consider two different Scenarios which are important if the transmission reliability is one of the criteria based on which the leader election should be done.

1. **Scenario I:** When intermediate nodes along the path from a node to the chain leader aggregate the receive data from downstream node with their own and transmits the aggregated data toward the chain leader.
2. **Scenario II:** When intermediate nodes along the path from a node to the chain leader do not perform any aggregation over the received data. In this case, intermediate sensor nodes only relay the received data toward the chain leader.

In order to simulate a dense and sparse deployment in a specific area with the same number of sensor nodes, we utilize density control concept [38]. Density control in a wireless sensor network with sensor nodes whose sensing ranges are fixed refers to the process of deciding which node is eligible to sleep after deployment in order to conserve energy while keeping network coverage [38]. Density control can also be expressed by varying the sensing

### 3.7 Performance evaluation of QoS-ACA

range of the sensor nodes as illustrated in Figure 3.6. In this way, if the sensing range of the nodes are set in such a way that two adjacent nodes do not cross the sensing region of each other, that deployment could be interpreted as sparse. It is obvious that in the sparse deployment the spatial correlation among nodes are quite low. Therefore according to Figure 3.6, by varying the sensing range of the sensor nodes we could make one dense deployment to sparse and vice versa. In our simulation, we consider the second definition of density control in order to simulate dense and sparse deployment. So, we consider the similar number of nodes for both sparse and dense deployment with this assumption that sensor node should adjust their sensing power in order to simulate a dense or sparse deployment.



**Figure 3.6. The impact of sensing range on density ( $R_{SD} \gg R_{SS}$ ) (a) the nodes deployment (b) Representing the given deployment as sparse ( $R_{SS}$  is small) (c) Representing the given deployment as dense ( $R_{SD}$  is large)**

#### 3.7.3 Performance evaluation

In the first experiment, we address the influence of energy parameter in electing a leader. In doing so, we vary the weight of energy parameter in Equation (3.12) and (3.13). First, we set the weights as  $\omega_p=1$ ,  $\omega_\varepsilon = 1$ ,  $\omega_\delta = 1$  which implies that all quality of service parameters are equally important. Second, we give higher priority to the energy parameter by setting

weights as  $\omega_\rho=1$ ,  $\omega_\epsilon = 2$ ,  $\omega_\delta = 1$  in order to state the energy as the most critical quality of service parameter. Finally, we assign weights as  $\omega_\rho=1$ ,  $\omega_\epsilon = 0$ ,  $\omega_\delta = 1$  in order to ignore energy parameter at all.

We run the simulation with respect to the three aforementioned setups for both Scenario I and Scenario II. Moreover, we draw the results for both dense and sparse deployment per each Scenario.

In Figure 3.7, Figure 3.8, Figure 3.9 and Figure 3.10, when  $\omega_\epsilon = 2$  the energy usage is targeted to be balanced in such a way that all nodes are alive almost simultaneously for as long as possible. According to these graphs, the difference between the death time of the first node and the death time of the last node is comparably smaller than that of when  $\omega_\epsilon = 1$  or  $\omega_\epsilon = 0$ , particularly for Scenario I in which data aggregation is accomplished along the path to the leader. For instance regarding Figure 3.7, when  $\omega_\epsilon = 2$  the difference between death times of the first and last node is about 500 time unit while that time difference when  $\omega_\epsilon = 1$  is around 3000 time unit and for  $\omega_\epsilon = 0$  is 4500 time unit. Therefore, by stating the energy parameter as the most critical one we well-balanced the energy consumption.

Since in the sparse deployment sensor nodes should employ the higher power level in order to cover the area and transmit the sensed data toward the base station, the death time of the last nodes are earlier than in the dense deployment. However, the relation among different setups of weights for both deployments are similar.

In Scenario I that aggregation is performed along the path on the intermediate nodes, more energy is conserved compared with when each intermediate node should forward every received packets from other nodes (Scenario II). Therefore, the death time of the last nodes in Scenario II is comparably earlier than that of Scenario I.

When  $\omega_\epsilon = 2$  the energy consumption is more uniform and balanced among sensor nodes. Therefore, all sensor nodes work together and die approximately at the same time. When chain leaders are chosen only based on delay and reliability, energy consumption is unbalanced which leads to creating holes in the network.

One should note that in the Figure 3.7, Figure 3.8, Figure 3.9 and Figure 3.10 the x-axis represents the simulation time up to when the average alive node ratio is higher than or equal to zero. This is the main reason that different graphs depict different ending simulation time in the x-axis.

### 3.7 Performance evaluation of QoS-ACA

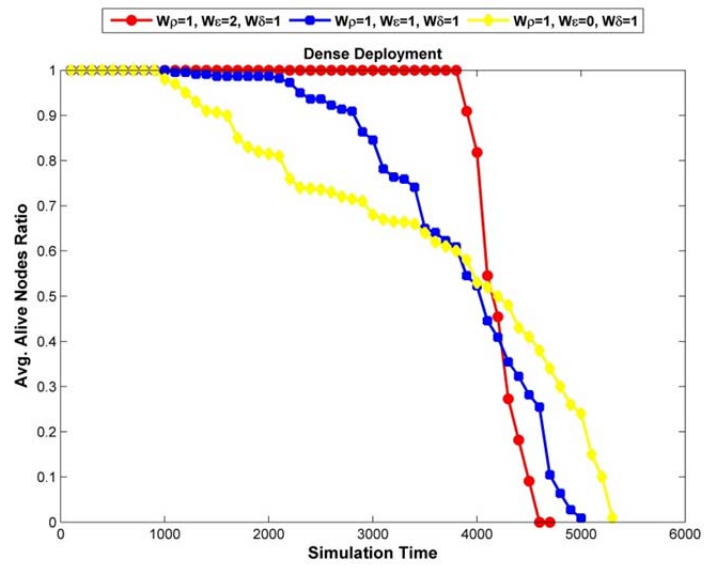


Figure 3.7. Average alive nodes ratio for dense deployment in Scenario I

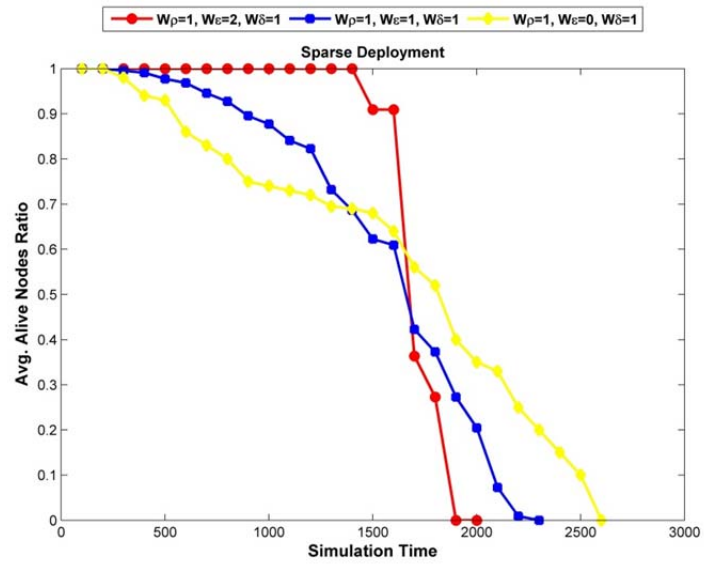


Figure 3.8. Average alive nodes ratio for sparse deployment in Scenario I



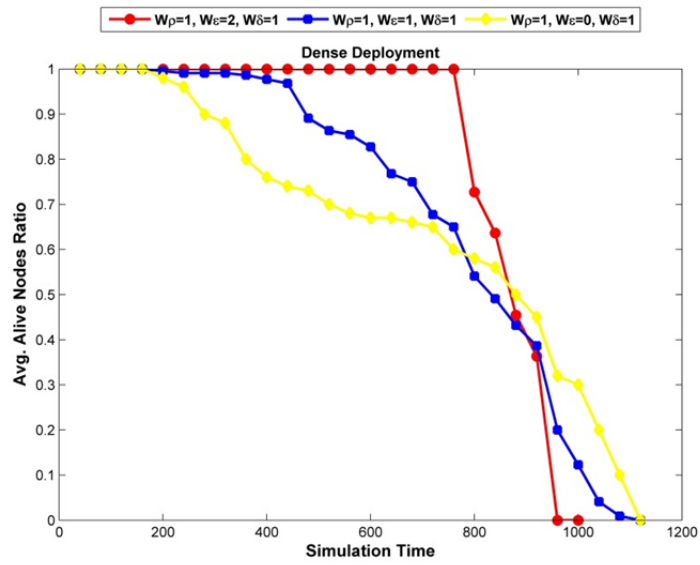


Figure 3.9. Average alive nodes ratio for dense deployment in Scenario II

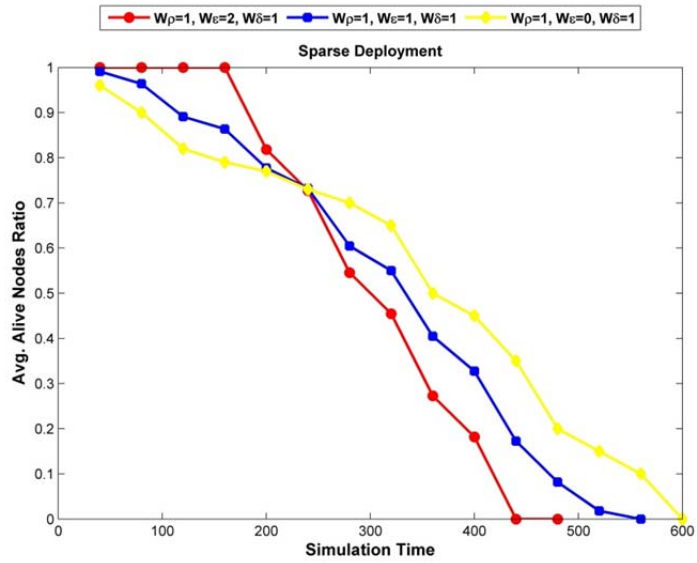


Figure 3.10. Average alive nodes ratio for sparse deployment in Scenario II

### 3.7 Performance evaluation of QoS-ACA

---

In the second experiment we want to concentrate on reliability parameter in selecting a leader. In doing so, we keep the weights of energy and delay parameters fixed as  $\omega_\epsilon = \omega_\delta = 1$  and vary the weight of reliability parameter  $\omega_\rho$  from 0 to 2. We vary the number of sensor nodes in each chain/cluster between 4 to 20 with an increment of 2.

The average contribution of each node on the packets which are collected during simulation time is illustrated in the Figure 3.11 and Figure 3.12 for densely and sparsely deployment, respectively. These graphs contain three different sets of weights while emphasizing on the reliability parameter.

From these Figure 3.11 and Figure 3.12, we observe that with an increase in the number of nodes in each chain, the contribution of each node on the received packets shows a linear decrease. The reason behind this is that the end-to-end reliabilities become low when the chain-length increases. As the end-to-end reliability is the main parameter based on which the contribution of each node is recognized, then having low end-to-end reliability leads to have less-contributed nodes.

Regarding Figure 3.11 and Figure 3.12, although the average node's contribution in sparse deployment is more or less similar to that of dense deployment, but the standard deviation of dense is higher than sparse deployment. This small standard deviation reveals that, the contribution degree of the sensor nodes in the sparse deployment are more close to each other rather than that of dense deployment. In other words, QoS-ACA in sparse networks tends to provide fairness in terms of contribution of sensor nodes to the aggregated value and hence, ensures that the aggregated value received by the base station from a chain contains data of as many sensor nodes as possible in that given chain. According to this graph, the requirement of a sparse deployment as described in Section 3.5.1.2 which is as much contributing as possible of every node, is satisfied. Moreover, these graphs also reveal that the requirement of dense deployment which is receiving as many data as possible from a region (chain) is fulfilled as well. According to these graphs and referring to Equation (3.2) and Equation (3.3), one can see that this way of ensuring reliability for period monitoring application in dense network tends to select the sensor nodes close to the high reliable links as a leader.

One can see that when  $\omega_\rho = 2$ , the average contribution of the nodes is comparably higher than when  $\omega_\rho = 0$ . According to Figure 3.11, the difference between the average contribution of the nodes for  $\omega_\rho = 2$  and  $\omega_\rho = 0$  could increase up to 0.2 which is a high amount and cannot be easily ignored for the applications which demand the maximum contribution of the nodes.

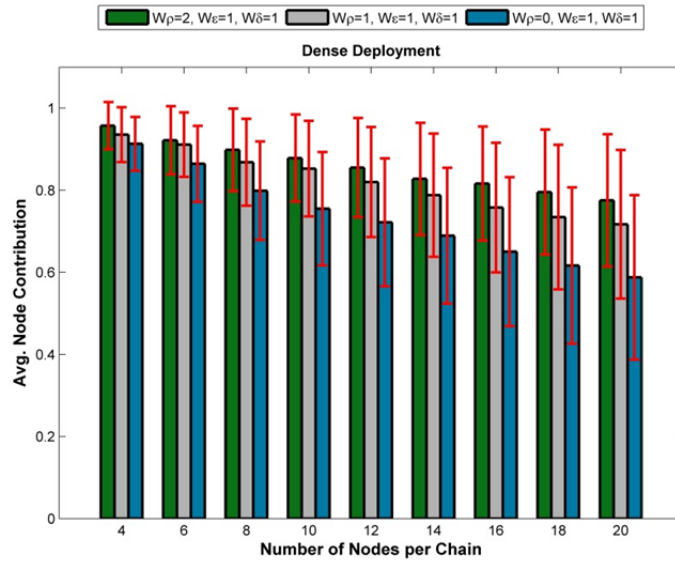


Figure 3.11. Average contribution of each node in dense deployment

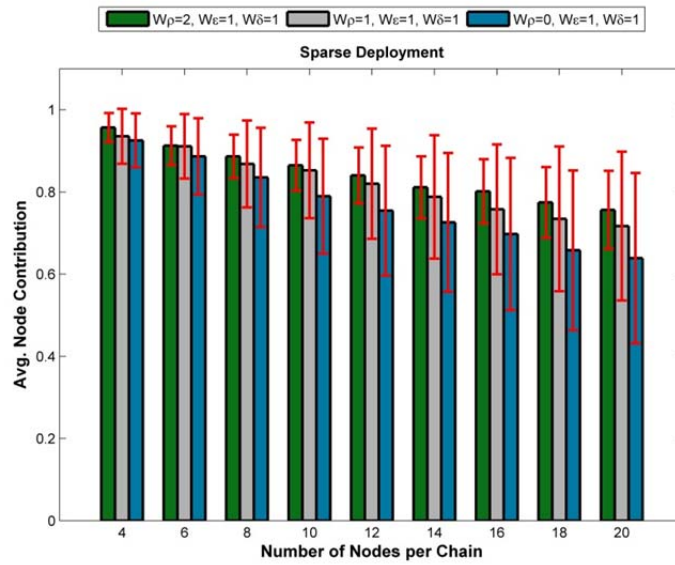


Figure 3.12. Average contribution of each node in sparse deployment

### 3.7 Performance evaluation of QoS-ACA

In the third experiments, we target the delay which represents the temporal accuracy of the monitoring process. Therefore, we investigate the performance of the network in terms of delay. To this end, similar to two previous experiments, we concentrate on three different priorities among quality of services with an emphasize on the delay parameter. In this way, we fix the weights of energy and reliability parameter as  $\omega_\varepsilon = \omega_\rho = 1$  and vary the weight of delay parameter  $\omega_\delta$  from 0 to 2. The effect of these weights assigning is presented in Figure 3.13. When the delay parameter is expressed with the highest priority  $\omega_\delta = 2$  among all quality of service parameters, the resultant end-to-end delay is much shorter. On the contrary, when the delay is not important for the application and then its weight set to  $\omega_\delta = 0$ , the application undergoes a longer delay. The difference between the shortest delay which belongs to  $\omega_\delta = 2$ , and the longest delay which imposed by  $\omega_\delta = 0$  could reach to 6 when the number of nodes is 20. Therefore, if one application requires to receive the packet as fresh as possible, we should state the delay parameter in Equation (3.12) and Equation (3.13) with the highest priority against two other parameters by setting  $\omega_\delta = 2$ .

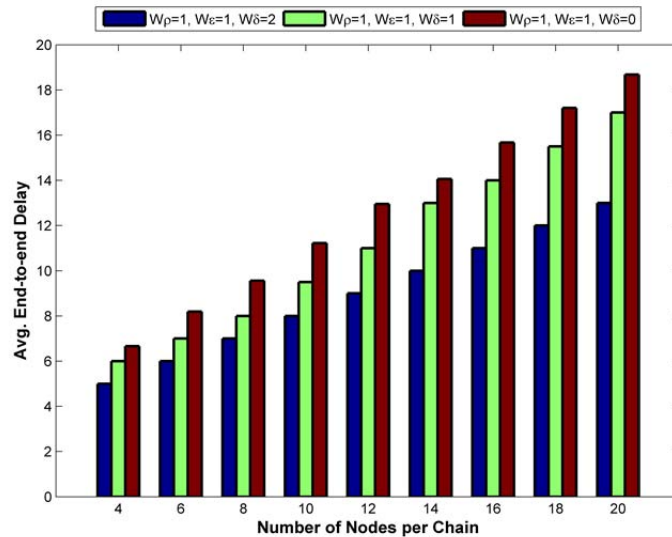
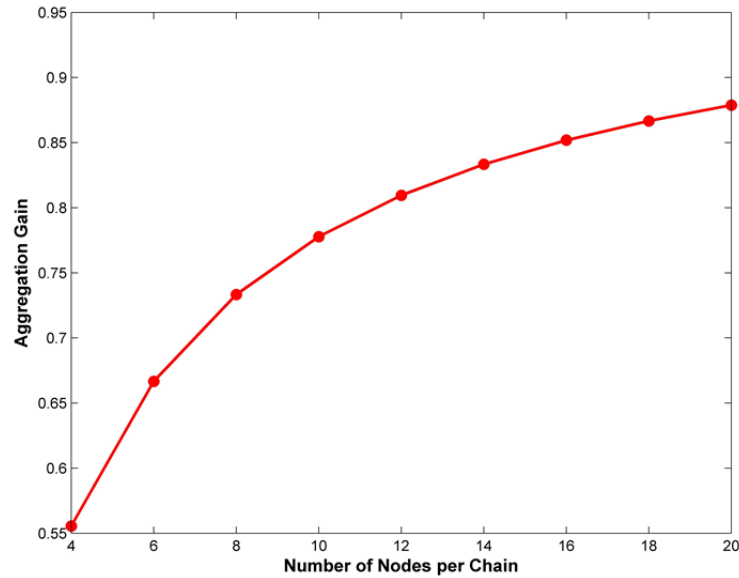


Figure 3.13. Average end-to-end delay

In the fourth experiment, we investigate the benefit of applying data aggregation on the system in terms of communication traffic reduction. Without data aggregation, all sensor

nodes should forward all the packets received from downstream nodes toward the cluster heads.

According to Figure 3.14, the aggregation gain is convexly increasing with the number of nodes per chain. When the number of nodes is around 20, the aggregation gain is about 0.87 which is pretty high. In this regards, data aggregation is highly preferable for densely deployed sensor nodes to avoid redundant transmission of correlated data. The effect of data aggregation would be more noticeable when the communication costs are significantly expensive than computation costs.



**Figure 3.14. Aggregation gain for QoS-ACA**

### 3.8. Enhancing QoS-ACA

So far we put forward QoS-ACA which is a quality of service aware chain-based data dissemination approach for a wireless sensor network whose sensor nodes are deployed in a sparse or dense manner. QoS-ACA concentrates on electing chain leader in such a way that brings the highest possible quality of service requested by the application.

QoS-ACA relies on fixed-size chain/cluster assumption, which may lead to different throughput under different conditions. For example, Figure 3.15 and Figure 3.16 show the average end-to-end reliability of nodes in a given chain for three different Maximum Error

### 3.8 Enhancing QoS-ACA

Rates (MER) on the links, when the number of nodes per chain is  $N_C=20$  and  $N_C=10$ , respectively.

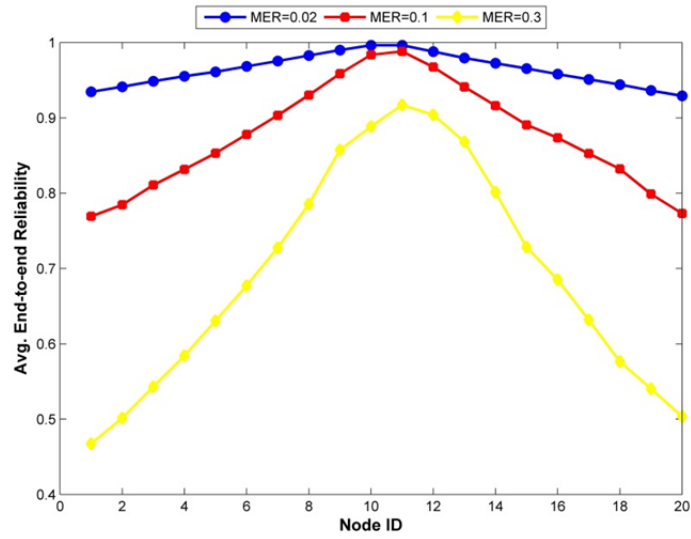


Figure 3.15. Average end-to-end reliability per node when  $N_C=20$

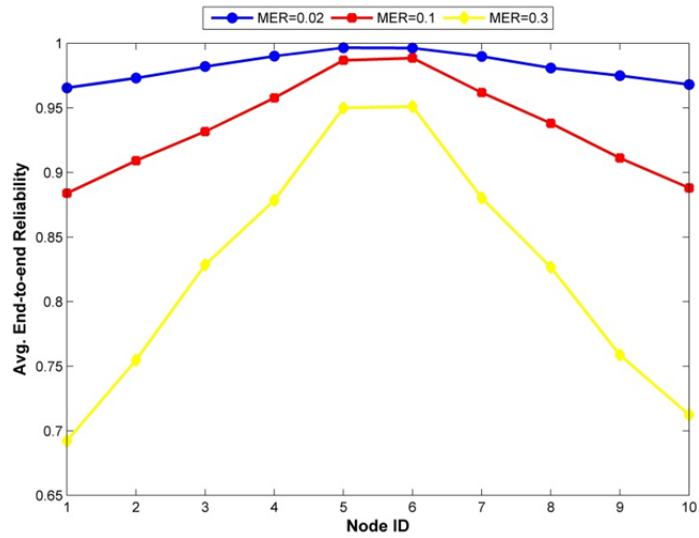


Figure 3.16. Average end-to-end reliability per node when  $N_C=10$

Based on these figures one can see that the end-to-end transmission reliability, which affects the contribution of the nodes, varies proportional to the MER. Therefore, depending on the link quality MER one cluster can represent different amount of contribution. If the minimum required end-to-end reliability is set to 0.85 by the application, then the size of a single chain can increase either up to 20 nodes when MER=0.02 or up to 10 nodes when MER=0.1, while the reliability requirement is still satisfied. Therefore, to be more efficient from the viewpoint of energy and reliability, chains with varying lengths should be formed on the basis of the link quality. To do so, in this section we aim to enhance QoS-ACA by relaxing the fixed-size cluster assumption and bringing the possibility to have adaptive and thereby variable clusters' size on the basis of experience conditions.

To the best of our knowledge, the approaches presented in this section, i.e., the so called REC and REC+, are the first chain-clustering algorithms that well incorporate energy, transmission reliability and delay together to construct clusters and to select proper cluster heads in wireless sensor networks. Most importantly, our proposed approach relaxes some strong assumptions that other existing techniques rely on such as the ability of nodes to communicate directly with other nodes or with the base station, having nodes with the same energy level, and having a fixed transmission range for intra-cluster communication. Moreover, our algorithm uses a d-hop clustering scheme, which consumes less energy than single-hop.

#### 3.8.1 Dynamic cluster formation: REC protocol

Compared to cluster-based techniques, chain-based techniques reduce excessive energy consumption for communicating with the base station, when base station is located far from the cluster heads. Generally speaking, the larger the number of hops between the source node and the base station, the lower the probability of packet contribution or packet delivery. This is due to the fact that, end-to-end reliability is obtained by the product of hop-by-hop reliabilities and having more hops usually lowers the end-to-end reliability. This is the main reason that the number of hops needs to be carefully calculated for a clustering-based approach.

The operation of REC which is a Reliable and Energy-balanced Chain-cluster based data dissemination protocol is divided into two main phases, i.e., (i) cluster head selection followed by chain-cluster forming, and (ii) data dissemination. The cluster head selection and chain-cluster forming phases are two separated phases but are tightly coupled as the corresponding subordinated cluster must be immediately constructed after finding each cluster head and the next cluster head must be chosen among nodes not being assigned to any

### 3.8 Enhancing QoS-ACA

---

cluster yet. The main advantage of our technique over existing clustering algorithms is that it considers not only local properties of a designated cluster head such as residual energy but also cares about the amount of transmission reliability and end-to-end delay that a given cluster head provides for its subordinated nodes.

#### 3.8.1.1 Phase I: Cluster-head selection and chain-cluster formation

Different from almost all clustering algorithms that first construct the chain-clusters and then select the cluster heads based on some criteria, REC first finds the best cluster head by looking at the residual energy of the node and then forms the clusters. The base station first finds the node whose energy utility  $U^e$  which is calculated in Equation (3.7) is higher than others and selects that node as the first cluster head. Then having hop-by-hop reliabilities reported by the nodes, the base station assigns nodes to the first cluster if end-to-end reliability of the given node to the designated cluster head is above the predefined threshold  $\theta^p$ . In the initialization step, in which no statistical information about links quality is available, hop-by-hop reliabilities are set to 1.

Assigning nodes to each cluster only by looking at the end-to-end reliability may result in long chains, which consequently leads to excessive data delivery delay, especially if either the links are almost reliable or the end-to-end reliability threshold  $\theta^p$  is small. To overcome this raising issue, REC uses another threshold  $\theta^\delta$  which denotes the maximum intra-cluster delay and is used to limit the chain size in each cluster. Therefore, end-to-end reliability and imposed delay are two criteria based on which the base station decides whether to add a given node to a specific cluster.

It is worth mentioning that the end-to-end reliability threshold  $\theta^p$  represents the minimum amount of contribution that one node in a given cluster is allowed to have in the aggregated data of its cluster. This definition of  $\theta^p$  is more or less similar to the utility definition of a node in terms of transmission reliability in a sparse deployment as expressed by Equation (3.4). Therefore, by using this threshold  $\theta^p$ , REC and REC+ aim to take control of the minimum contribution that one node could bring for a cluster's final data.

After forming the first cluster, base station finds the next powerful node which does not belong to the first cluster and regards it as the second cluster head. Then again chain-cluster forming phase starts and nodes are assigned to the second cluster as long as thresholds of intra-cluster delay  $\theta^\delta$  and the threshold of the end-to-end reliability  $\theta^p$  are still satisfied.

The node which belongs to the first cluster can leave that cluster and join the second cluster if the end-to-end reliability provided by the second cluster head is higher than the



end-to-end reliability provided by the first cluster head. In case of a tie, the closer cluster head is selected for the given node.

The base station repeats these two phases as long as all nodes are assigned to one of the clusters. After that the base station creates the second tier chain but the length of each cluster may be quite long so that two adjacent cluster head cannot communicate directly with each other as shown in Figure 3.17.b. To overcome this problem, each cluster head employs one or more intermediate nodes along the path towards the next cluster head to relay cluster head's data. Identification of these Relay Nodes (RNs) in each cluster is the responsibility of the base station. To select relay nodes, base station looks at the Maximum Transmission Range (MTR) each cluster head can provide while using the maximum Power Level (PL). The upstream node which is located at a distance of MTR from the given cluster head is selected as the relay node if it has enough energy otherwise one powerful node near the suggested relay node is chosen as the best relay node for that cluster head. It is still possible a relay node cannot relay its cluster head's data to the next cluster head, if the distance between the two is large. In this case, another relay node which is located closer to the upstream cluster head and satisfies the aforementioned criterion is selected to relay the data received from the previous relay node. Selection of relay nodes is repeated as long as one relay node falls in the coverage area of the upstream cluster head and thereby data of the previous cluster head can reach the upstream cluster head (Figure 3.17.b.). The base station also selects the most powerful node, which can directly communicate with the base station as the second-tier leader. This node is responsible to directly send an aggregated value containing data of almost all nodes in the network to the base station.

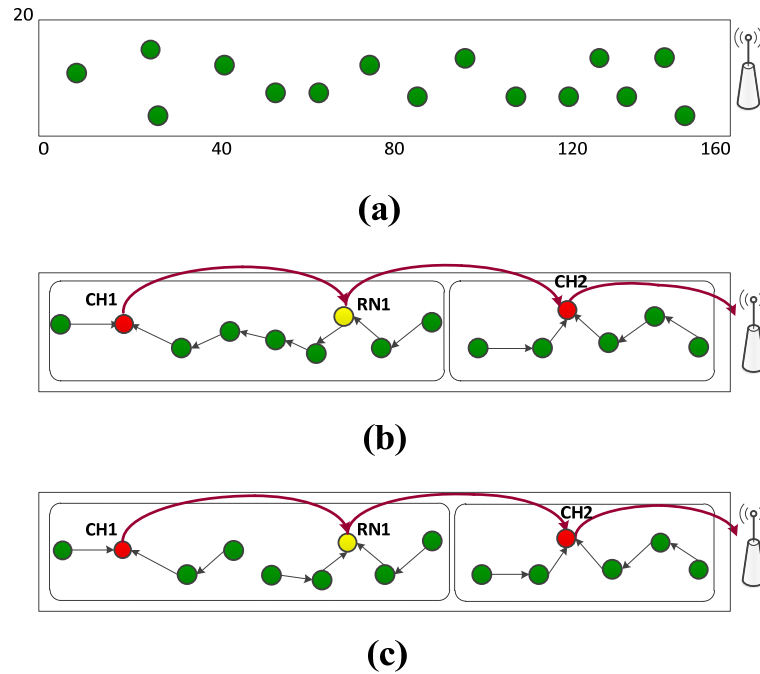
Once the relay nodes, cluster heads, and associated clusters are identified, the base station transmits this information back to all nodes in the network. To this end, the base station broadcasts a message that contains chain-cluster information, in the network by the help of the second tier nodes which are either cluster head or relay node. By doing so, each node becomes aware of its role (being a regular sensor node, cluster head or relay node) as well as the cluster it belongs to and the direction towards which it should send its data.

Moreover, based on the number of nodes in a given cluster, the cluster head of that specific cluster creates a TDMA schedule to inform every subordinated nodes about the time slots through which they should transmit their data.

As idle-listening is the major source of inefficiency in the communication, slotted transmission schemes such as TDMA allow the radio components of each non-cluster head node to be turned off at all times except during their transmission time. By doing so, it minimizes the energy dissipated by the individual sensors. Moreover, TDMA schedule

### 3.8 Enhancing QoS-ACA

ensures that there are no collisions among data messages. Therefore, each cluster head can schedule activities in its own cluster by setting up a TDMA schedule and inform the belonging nodes about it. After the TDMA schedule is known by all nodes in the cluster, the operation of protocol comes into steady-state data dissemination.



**Figure 3.17.** A schematic of the formation of chain-clusters in REC and REC+

In order to reduce the interference from nodes belonging to other clusters, we could utilize CDMA or FDMA. In this regard, each cluster can communicate using different CDMA codes or different frequency of FDMA. In order to overcome the limited bandwidth in FDMA, we could employ CDMA which provides the full bandwidth to the nodes.

The flowchart diagram of phase I is demonstrated in Figure 3.18.

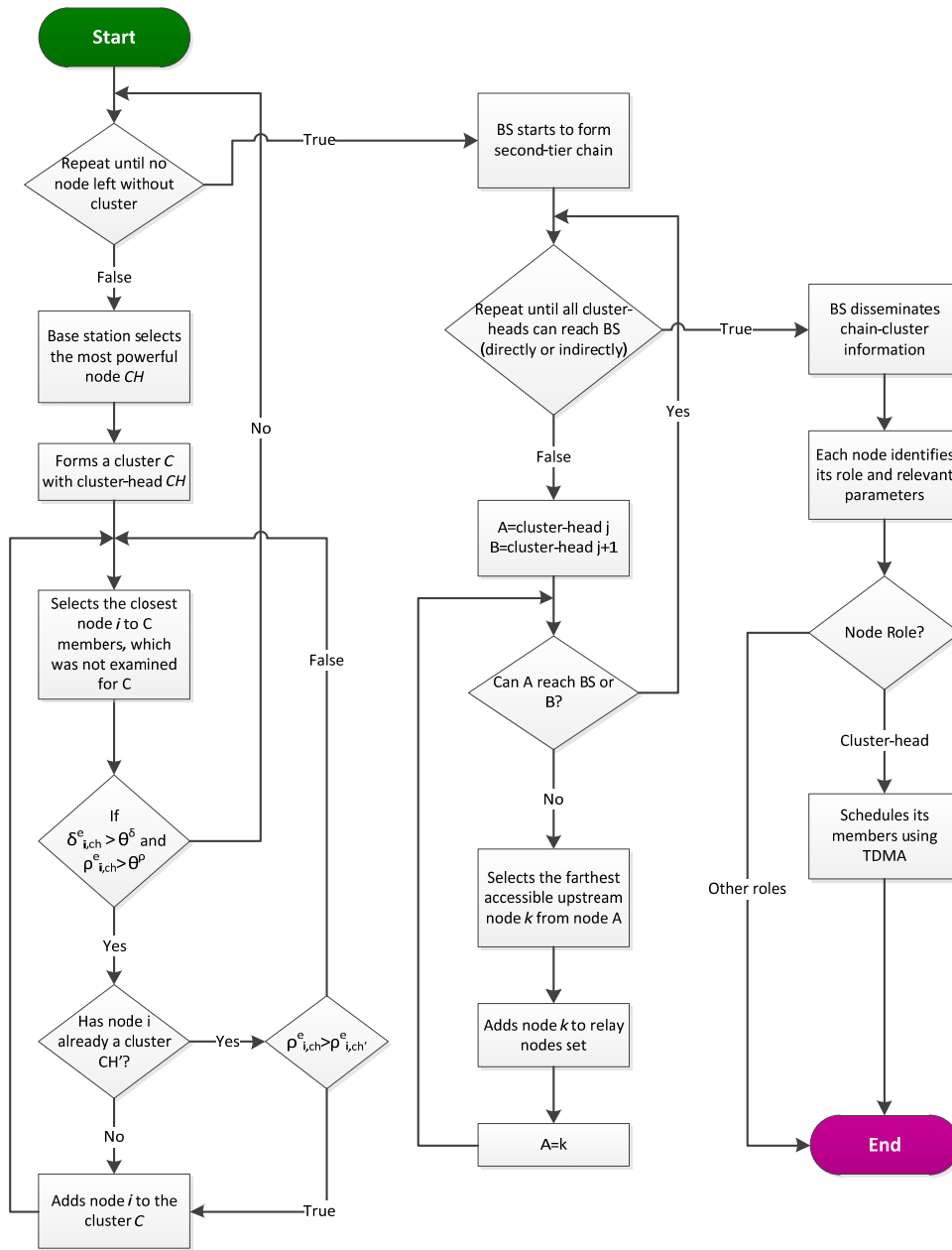


Figure 3.18. Flowchart diagram of phase I of REC

### 3.8 Enhancing QoS-ACA

---

#### 3.8.1.2 Phase II: data dissemination

The second step of REC is the steady state communication or data dissemination. Every node in a chain must send its data in its allocated time slot to its upstream neighbor which is selected in the chain-cluster formation phase. Intermediate nodes along the path to the cluster head aggregate the data received from the downstream nodes with their own (if any) and forward the local aggregated value toward the cluster head. Cluster head gathers the received data, aggregates and finally transmits that through the second tier nodes to the base station. As REC also addresses transmission reliability, any significant change in the links quality must be reported to the base station, which should in turn modify clustering size/shape if needed. For doing so, each sensor node by comparing the sequence number of the packet received from its downstream node with the one expects to receive (by looking into its history) can easily calculate the packet loss of its adjacent link. If this change in the link quality is significant and remained for some successive intervals, that node puts the link reliability it calculates for its adjacent link along with the data it must relay in a packet and forwards it.

After receiving these statistical information about link quality, the base station reforms the clusters around the given link (if needed). If compared to the current information, link quality has decreased, the base station may either divides the cluster whose link quality has decreased into two clusters or assigns some of the nodes belonging to that cluster to the adjacent clusters. In contrary, if the quality of the link has improved, the base station may merge two adjacent clusters or modify them so that they provide better performance.

Finally, the base station informs the sensor nodes whose clustering information has changed to update their clustering parameters.

Each sensor node also sends its residual energy along with the data it must relay to the base station either in some predefined time intervals or when its energy drops below a threshold  $\theta^e$ . Based on the received energy values, the base station may modify some clustering parameters to prolong lifetime.

#### 3.8.2 Distance-aware cluster formation: REC+ protocol

REC takes energy, transmission reliability and delay into account while making clusters but it has no assumption about the distance between two cluster heads, which most of other approaches have. Relaxing the mentioned assumption does not come free and REC pays for it with the increasing delay. As it can be seen from Figure 3.17.b., data of the nodes located between RN1 and CH1 is transmitted and aggregated along the path till it reaches the CH1.

Then CH1 has to send the aggregated value back to the RN1. This imposes an extra delay which could have been shorter if nodes closer to relay nodes send their data towards relay nodes and the nodes closer to the CH1 send their data towards CH1 as depicted in Figure 3.17.c. In this case, delay may be shortened to half of the previous delay as two separated chain can almost send their data in parallel. Also, one intra-cluster communication related to the middle link (bridge link) is omitted. Doing so is similar to having three individual clusters while every relay node acts as a new cluster head. Therefore, to achieve a satisfactory performance, the base station can create a cluster around each relay node while fairly (in terms of delay) assigns sensor nodes between the given cluster head and relay node to the most appropriate cluster. We introduce this modification to REC by making a new protocol called REC+.

### 3.9. Performance evaluation of REC and REC+

In this section we present a comparison among five approaches; REC+, REC, QoS-ACA, PEGASIS, LEACH-CC in cluster formation and routing within clusters in the stable state.

Figure 3.19 illustrates an abstract view of how these approaches operate when transmission range includes five-hop. The cluster size in QoS-ACA is smaller than others and it is because of its assumption that all nodes in two adjacent clusters must be in communication range of each other. QoS-ACA first constructs a cluster and then selects the best cluster head whose role assigned to different cluster members in different time intervals. It implies that all nodes in two adjacent clusters must be able to communicate with each other as it is possible that two farthest nodes to be chosen as two adjacent cluster heads. In contrast, REC and REC+ first find one cluster head and then form a cluster around it. Therefore, they do not require that all nodes in two adjacent clusters be in communication range of each other. Taking the benefit of that, the size of the clusters in REC and REC+ could be bigger than in QoS-ACA. While QoS-ACA undergoes many inter-cluster communications due to small size of its clusters, other approaches suffer from more intra-cluster communications.

One should note that the simulation scenario is similar to the one explained in Section 3.7.2 except that (i) REC and REC+ do not need to know the number of clusters in advance and (ii) the quality of unreliable links varies after 100 time intervals.

Moreover, the thresholds for energy, delay and reliability are set as  $\theta^\epsilon = 0.05$ ,  $\theta^\delta = 8$  and  $\theta^\rho = 0.8$ . In the simulation set up, the weights related to QoS-ACA are set as  $\omega_\rho = \omega_\delta = \omega_\epsilon = 1$ .

### 3.9 Performance evaluation of REC and REC+

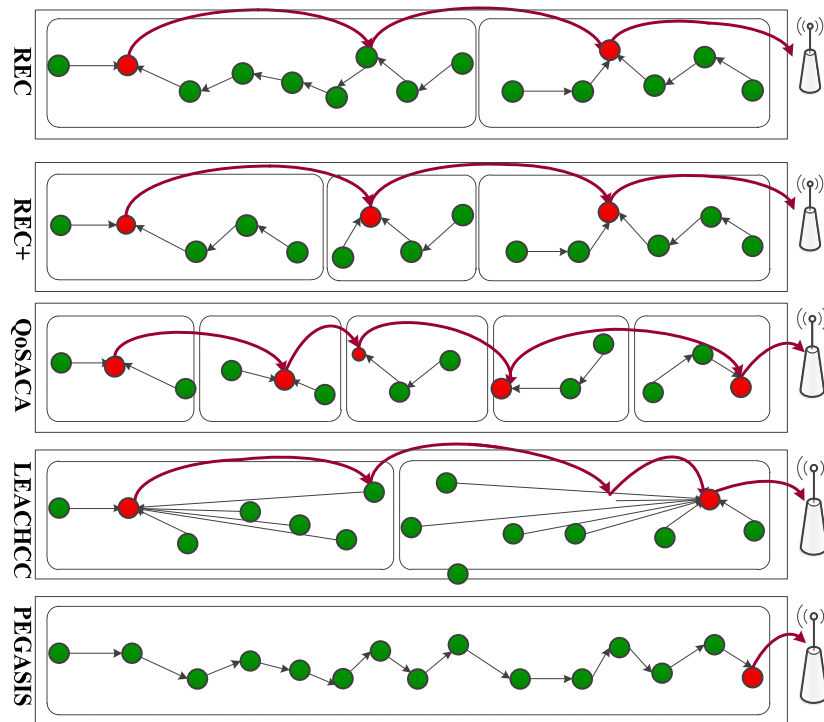


Figure 3.19. An abstract view of how REC+, REC, QoS-ACA, PEGASIS, LEACH-CC operate when transmission range includes 5-hop nodes

#### 3.9.1 Performance metrics

For the performance evaluation, we utilize the same performance metrics as introduced in Section 3.7.1 and we add the following two:

- **Average Energy Consumption:** This metric shows the average energy expended by each node to fulfill its task if it is selected as a chain-leader, relay node, or a regular sensor node for a specific time period.
- **Energy  $\times$  Delay:** This metric is usually used [39] to show how the energy and delay cost are balanced. In general, chain-based approaches imply lower energy consuming and longer delay. The cluster-based approaches, in contrary, usually impose higher energy consumption and shorter delay. Therefore, Energy  $\times$  Delay metric could come in handy to draw the trade-off between delay and energy costs.

### 3.9.2 Performance evaluation

Figure 3.20 compares the ratio of average alive nodes as the simulation progresses. It can be seen that REC+ prolongs network lifetime much more effectively than other approaches. The average alive nodes ratio in PEGASIS sharply drops from almost 0.6 at time instant 3400 to 0 at time instant 3600. So in less than 200 time units 60% of the nodes die. This is due to the fact that the leaders or cluster heads which are located near the base station are unwisely overused irrespective of their energy level. As soon as nodes close to the base station die, the other cluster heads should utilize their highest power level to be able to reach the base station. However, it is quite possible that even by using the highest power level the leader nodes cannot communicate with the base station and therefore the network get disconnected. Since PEGASIS does not consider residual energy as a criterion to select chain leader, it cannot well balance the energy consumptions. In this sense, even though some sensor nodes are still alive but because they cannot communicate with the base station they are considered as dead. Moreover, we find in the simulation that our protocols takes higher number of rounds than that of PEGASIS and LEACH-CC before the first sensor node dies.

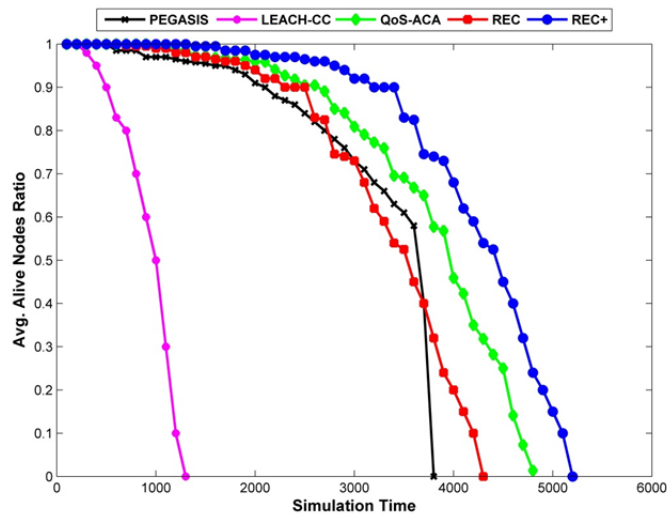


Figure 3.20. Average alive nodes ratio over time

In LEACH-CC, as sensor nodes in each cluster communicate directly with the cluster head, the nodes' energy are depleted much sooner than chain-based approaches. One can see that the direct long range communication of sensor nodes in LEACH-CC severely limits the lifetime of the network.

### 3.9 Performance evaluation of REC and REC+

---

The average contribution of nodes in each cluster is plotted in Figure 3.21. Due to the fact that PEGASIS forms a single chain among the sensor nodes while leader is the nearest node to the base station, the contribution of the farther nodes is much lesser than the closer nodes. The contribution of the farther nodes becomes even lesser as the chain length increases. The contribution of LEACH-CC whose sensor nodes send their reported data directly to the cluster head outperforms other approaches at the expense of lifetime. Although the contribution of nodes in REC and REC+ is slightly smaller than that of LEACH-CC, but it still satisfies the minimum contribution which are set by the application.

It is worth recalling that the lifetime of REC and REC+ is significantly longer than LEACH-CC. The reason that REC and REC+ show lower contribution at the beginning, when the number of nodes is less than 20, compared with QoS-ACA is in relation to the fixed-size clusters in QoS-ACA. In this regards, since QoS-ACA always assumes a fixed number of clusters where the sensor nodes are evenly distributed in those clusters, the number of nodes in each cluster is obtained through dividing total number of nodes in the network by the number of clusters. Therefore, at the beginning, when the number of nodes is smaller than 20, the chain length in each cluster of QoS-ACA is also short which consequently results in the higher nodes' contribution. However, REC and REC+ relax this assumption by giving the opportunity to have different chain lengths according to the nodes and links situation. Therefore, at the beginning REC and REC+ form longer chains, which leads to smaller contribution but still acceptable by the application. This is the main reason of this difference between QoS-ACA and REC/REC+ at the beginning of Figure 3.21. Having the same reason, when the number of nodes increases to more than 20, REC and REC+ outperform QoS-ACA.

The average energy consumption of each node for different approaches are illustrated in Figure 3.22. LEACH-CC introduces the highest energy consumption because of direct communication of the sensor nodes with the cluster heads inside a cluster. In contrary, PEGASIS imposes the lowest energy consumption and longest delay, as shown in Figure 3.23, because of its short-range multi-hop communications. The main source of higher energy consumption of REC compared with REC+ and QoS-ACA is related to the relay nodes, which have to transmit data twice per round, i.e., once when they are sending their data toward the cluster head as an ordinary sensor node and later when they have to undertake the relay node responsibility to transmit the cluster head's data to the upstream cluster head (or upstream relay node). Moreover one can see in Figure 3.19 that REC+ has one communication link (or one transmission) less per each relay node compared with REC. This omitted link is the bridge link which divides one cluster in REC into two clusters for



REC+. The more the number of relay nodes, the more bridge links are omitted that results in less average energy consumption.

By summing up inter-cluster and intra-cluster delay, the total end-to-end delay is shown in Figure 3.23. The most promising approach in terms of end-to-end delay is REC+. Despite using direct communication within a cluster in LEACH-CC, the average-end-to-end delay of LEACH-CC is still longer than REC, REC+ and QoS-ACA. The reason behind this is the TDMA scheduling of the nodes, which prevents having interference inside a cluster. In this respect, even though the direct communication in LEACH-CC provides the shortest delay between sensor nodes and cluster head, but because of scheduling nodes to send their data in the specific time instants the intra-cluster delay of LEACH-CC is high. It is worth mentioning that although other approaches also utilize TDMA scheduling inside a cluster, thanks to the adjusting power level their end-to-end intra-cluster delay is shorter than that of LEACH-CC. Basically, in the chain-based approaches, i.e. PEGASIS, QoS-ACA, REC, REC+, sensor nodes adjust their power level in such a way that could communicate with the closest neighbors. This helps to schedule more than one node in the chain to send its data towards the leader. Therefore, simultaneous transmissions are possible only among spatially separated sensor nodes. For instance, the leftmost and the rightmost sensor nodes could transmit their data simultaneously as they are not in the communication range of each other and moreover their destinations (upstream node) are not similar. In this regard, two sensor nodes, one from left side and another from right side, can be scheduled at the same time instant to transmit their data. This is the main reason that end-to-end delay of LEACH-CC is longer than QoS-ACA, REC and REC+.

Since the trade-off between delay and energy cost is important for data gathering applications, we illustrate in Figure 3.24 the Energy  $\times$  Delay cost to better evaluate the trade-off between energy and delay for these five approaches. As it can be seen from Figure 3.24, the most promising approach in terms of Energy  $\times$  Delay cost is REC+. Although the performance of QoS-ACA is more or less similar to REC and REC+, one should notice that QoS-ACA suffers from the rigid assumption that all nodes in two adjacent clusters must be in transmission range of each other, which is relaxed by REC and REC+.

### 3.9 Performance evaluation of REC and REC+

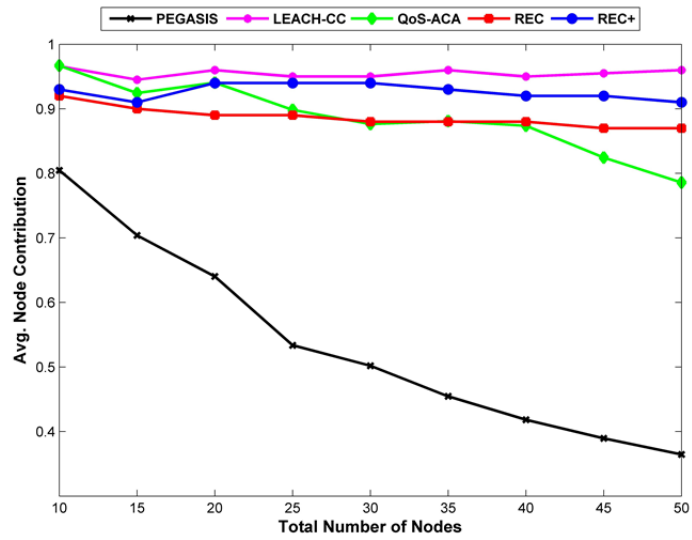


Figure 3.21. Average node's contribution in each cluster

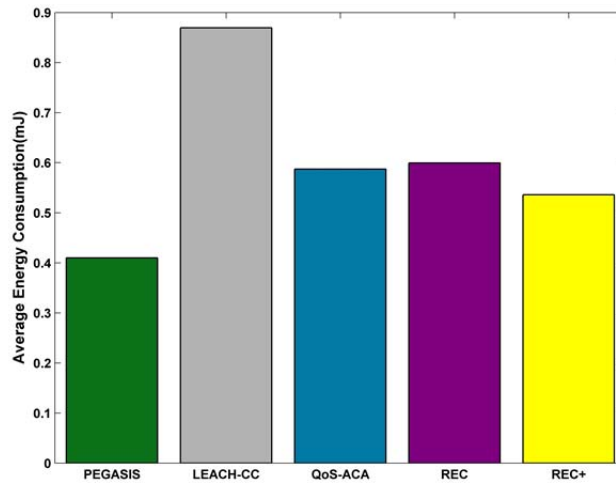


Figure 3.22. Average energy consumption

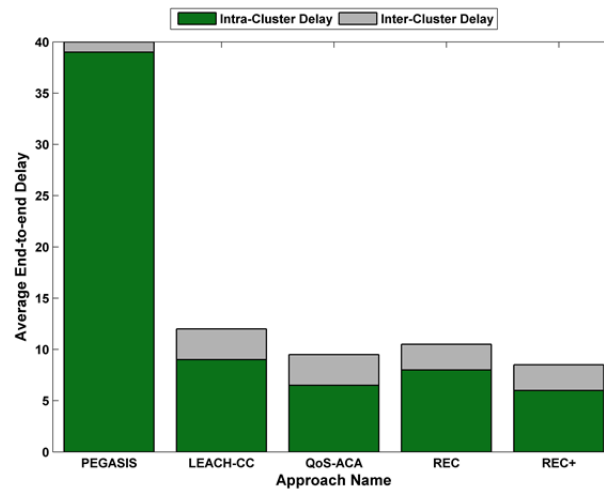


Figure 3.23. Average end-to-end delay

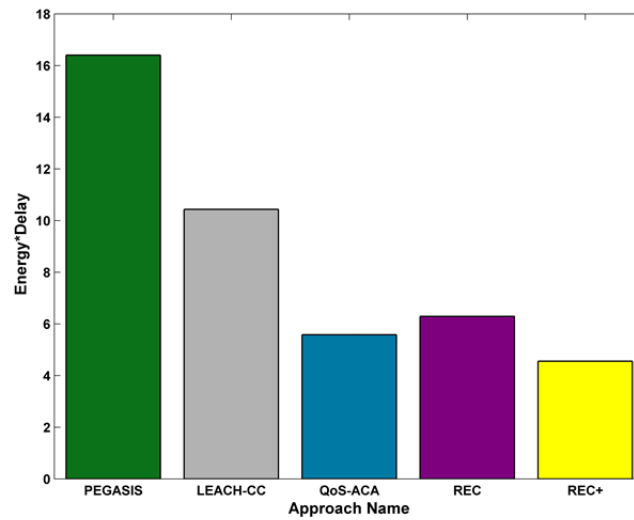


Figure 3.24. Energy × Delay

### 3.10. Chapter summary

In this chapter, we concentrate on the quality of service aware data disseminating for the chain-based wireless sensor networks. Lifetime, end-to-end transmission reliability and delay

### 3.11 Bibliography

---

are the quality of service parameters that our proposed data disseminations aim to guarantee them.

Compared to the existing chain-cluster based data dissemination techniques our protocols combine these three namely quality of service parameters according to the application preferences in order to make a trade-off in ensuring them. Furthermore dependent on the network density, QoS-ACA ensures reliability in two different ways for the sparsely and densely deployed nodes.

The general strategy we adopts to ensure quality of services is either selecting the most appropriate sensor nodes regarding the required quality of services in QoS-ACA or forming adaptive clusters and setting the boundaries of the clusters with respect to the application level quality of service constrains in REC and REC+. In other words, by REC and REC+ we devise an adaptive cluster's shape/size adjustment according to the application requirements and nodes/links conditions.

Moreover, in the interest of conserving both energy and bandwidth along with providing meaningful information to end-users, our protocols in this chapter utilize data aggregation on both chain leader or cluster heads and intermediate nodes along the path toward the destination.

In the simulations we illustrate various performance metrics for the applications whose requirements in terms of the given quality of services are different. The simulation results reveal that compare with other approaches QoS-ACA, REC and REC+ well-disperse the cluster heads throughout the network by using a central quality of service aware algorithm to either select the cluster head or form the chain-clusters.

### 3.11. Bibliography

- [1]. Heinzelman, W.R., J. Kulik, and H. Balakrishnan. *Adaptive protocols for information dissemination in wireless sensor networks*. in *5th annual ACM/IEEE international conference on Mobile computing and networking*. 1999.
- [2]. Mahajan, A., et al., *Comparative Analysis of Different Data Dissemination Strategies in Wireless Sensor Networks*. International Journal of Advanced Research in Computer and Communication Engineering, 2014. 3(4).
- [3]. Zhao, L., et al. *LEGR: A Load-Balanced and Energy-Efficient Geographic Routing for Lossy Wireless Sensor Network*. in *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. 2007.

- [4]. Halsall, F. and D. Links, *Computer Networks and Open Systems*. Addison-Wesley Publishers, 1995: p. 112-125.
- [5]. Schwartz, M., *Telecommunication networks: protocols, modeling and analysis*. Vol. 7. 1987: Addison-Wesley Reading, MA.
- [6]. El Emary, I.M.M. and S. Ramakrishnan, *Wireless Sensor Networks: From Theory to Applications*2013: CRC Press.
- [7]. Palazzo, S., F. Cuomo, and L. Galluccio, *Data Aggregation in Wireless Sensor Networks: A Multifaceted Perspective*, in *Sensor Networks*2009, Springer. p. 103-143.
- [8]. Busse, M., T. Haenselmann, and W. Effelsberg. *TECA: A topology and energy control algorithm for wireless sensor networks*. in *9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*. 2006.
- [9]. Chen, Y. and S.H. Son. *A fault tolerant topology control in wireless sensor networks*. in *3rd ACS/IEEE International Conference on Computer Systems and Applications*. 2005.
- [10]. Heinzelman, W.R., A. Chandrakasan, and H. Balakrishnan. *Energy-efficient communication protocol for wireless microsensor networks*. in *33rd Annual Hawaii International Conference on System Sciences*. 2000.
- [11]. Jung, W.-S., et al. *A hybrid approach for clustering-based data aggregation in wireless sensor networks*. in *Third International Conference on Digital Society*.
- [12]. Du, K., J. Wu, and D. Zhou. *Chain-based protocols for data broadcasting and gathering in the sensor networks*. in *International Parallel and Distributed Processing Symposium*. 2003.
- [13]. Lindsey, S., C. Raghavendra, and K.M. Sivalingam, *Data gathering algorithms in sensor networks using energy metrics*. *IEEE Transactions on Parallel and Distributed Systems*, 2002. 13(9): p. 924-935.
- [14]. Shin, J. and C. Suh. *Energy-efficient chain topology in ubiquitous sensor network*. in *10th International Conference on Advanced Communication Technology*. 2008.
- [15]. Tabassum, N., Q. Mamun, and Y. Urano. *COSEN: a chain oriented sensor network for efficient data collection*. in *Third International Conference on Information Technology: New Generations*. 2006.
- [16]. Kim, H.-s. and K.-j. Han. *A power efficient routing protocol based on balanced tree in wireless sensor networks*. in *First International Conference on Distributed Frameworks for Multimedia Applications*. 2005.
- [17]. Tan, H.Ö. and I. Körpeoğlu, *Power efficient data gathering and aggregation in wireless sensor networks*. *ACM Sigmod Record*, 2003. 32(4): p. 66-71.

### 3.11 Bibliography

---

- [18]. Ramadan, R.A. *Agent based multipath routing in wireless sensor networks*. in *International Symposium on Intelligent Agents*. 2009.
- [19]. Vidhyapriya, R. and P.T. Vanathi, *Energy efficient adaptive multipath routing for wireless sensor networks*. IAENG International Journal of Computer Science, 2007. 34(1): p. 56-64.
- [20]. Dulman, S., et al., *Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks*, in *Wireless Communications and Networking2003*.
- [21]. Fan, Z. and H. Zhou, *A scalable power-efficient data gathering protocol with delay guaranty for wireless sensor networks*, in *Mobile Ad-Hoc and Sensor Networks2007*, Springer. p. 221-232.
- [22]. Lai, K.K. and J.W.M. Chan, *Developing a simulated annealing algorithm for the cutting stock problem*. Computers & industrial engineering, 1997. 32(1): p. 115-127.
- [23]. Lindsey, S. and C.S. Raghavendra. *PEGASIS: Power-efficient gathering in sensor information systems*. in *International Conference on Aerospace*. 2002.
- [24]. Taghikhaki, Z., N. Meratnia, and P.J.M. Havinga. *Energy-efficient trust-based aggregation in wireless sensor networks*. in *International Conference on Computer Communications Workshops*. 2011.
- [25]. Kruskal, J.B., *On the shortest spanning subtree of a graph and the traveling salesman problem*. Proceedings of the American Mathematical society, 1956. 7(1): p. 48-50.
- [26]. Heinzelman, W.B., *Application-Specific Protocol Architectures for Wireless Networks*. PhD Thesis, MIT, 2000.
- [27]. Xiangning, F. and S. Yulin. *Improvement on LEACH protocol of wireless sensor network*. in *International Conference on Sensor Technologies and Applications*. 2007.
- [28]. Gupta, A., M. Goyal, and S. Malik, *Clustering Approach for Enhancing Network Energy using LEACH Protocol in WSN*. International Journal of Wired and Wireless Communications, 2013. 2(1): p. 20-25.
- [29]. Tang, F., et al., *A chain-cluster based routing algorithm for wireless sensor networks*. journal of intelligent manufacturing, 2012. 23(4): p. 1305-1313.
- [30]. Ali, S.A. and S.K. Refaay, *Chain-Chain Based Routing Protocol*. International Journal of Computer Science Issues (IJCSI), 2011. 8(3).
- [31]. Bian, X., X. Liu, and H. Cho. *Study on a cluster-chain routing protocol in wireless sensor networks*. in *3rd International Conference on Communications and Networking 2008*.

- [32]. Hashmi, S.U., H.T. Mouftah, and N.D. Georganas. *Achieving reliability over cluster-based wireless sensor networks using backup cluster heads*. in *International Conference on Global Telecommunications 2007*.
- [33]. Sadat, A., et al. *Reliable and energy efficient backup clustering scheme for wireless sensor networks*. in *International Conference on Information Networking (ICOIN)*. 2010.
- [34]. Gupta, G. and M. Younis. *Fault-tolerant clustering of wireless sensor networks*. in *International Conference on Wireless Communications and Networking*. 2003.
- [35]. Watfa, M.K., O. Mirza, and J. Kawtharani, *BARC: A Battery Aware Reliable Clustering algorithm for sensor networks*. *Journal of Network and Computer Applications*, 2009. 32(6): p. 1183-1193.
- [36]. Cheng, M.X., X. Gong, and P.-J. Wan, *Minimum delay routing in multihop wireless networks*, in *Wireless Algorithms, Systems, and Applications 2011*, Springer. p. 146-156.
- [37]. Roedig, U., A. Barroso, and C.J. Sreenan. *Determination of aggregation points in wireless sensor networks*. in *30th Conference on Euromicro*. 2004.
- [38]. Cheng, Y.-M. and L.-H. Yen. *Range-based density control for wireless sensor networks*. in *4th Annual Communication Networks and Services Research Conference*. 2006.
- [39]. Lindsey, S., C. Raghavendra, and K. Sivalingam. *Data gathering in sensor networks using the energy\* delay metric*. in *International Parallel and Distributed Processing Symposium*. 2001.

# Reliable Dissemination of Time-Constrained Data<sup>1</sup>

---

Wireless sensor networks can be used for many mission-critical applications in which reliable and timely delivery of the sensory data plays a crucial role in the success of the missions. Time-critical applications of wireless sensor networks demand timely data delivery for fast identification of out-of-ordinary situations and fast and reliable delivery of notification and warning messages. Reliable data dissemination is traditionally performed by applying error control mechanisms, which usually suffer from the delay arisen from transmitting redundant data. Moreover, having nodes operated in low duty-cycle in order to save energy and prolong lifetime leads to longer communication delays. Both transmission delay arisen from error control approaches and the delay caused by duty-cycle, may not be tolerable for the time-critical applications. Therefore, providing real-time guarantee and reliable data delivery in wireless sensor networks which needs to operate for long time is quite challenging. In this chapter, we investigate a disseminating strategy using which different packet- and

---

<sup>1</sup> This chapter is based on the following publications:

- (i) *On QoS guarantees of error control schemes for data dissemination in chain-based wireless sensor networks*. Sensors & Transducers Journal, 18. pp. 188-202.
- (ii) *A reliable and real-time aggregation aware data dissemination in a chain-based wireless sensor network*. In Proceeding of The Sixth International Conference on Sensor Technologies and Applications (SENSORCOMM 2012).
- (iii) *An error control scheme for delay constrained data communication in a chain-based wireless sensor network*. In Proceeding of The Seventh IEEE International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA-2012).



link-local error control mechanisms are adaptively employed according to both the Time-To-Live (TTL) constraint of the packets and the reliability of en-route links. In this respect, we propose a runtime adaptive error control approach, called READ, which allocates the available packet TTL proportionately to the packet loss probability of the links along the forwarding path for duty-cycle-based wireless sensor networks. Simulation results reveal the superiority of our approach considering both hit ratio and energy efficiency specially in a low duty-cycle network and when either the TTL is short or the packet loss is high.

### 4.1. Introduction

Time-critical applications highly depend on the availability of real-time data as in these applications data is not valuable if it is not received within the specific deadline. Outdated data is not only useless, but may also be harmful as it may have negative impacts on the decisions on the basis of invalid and stale information. Moreover, transmitting expired data depletes the bandwidth and energy of relaying nodes inappropriately. Therefore, it would be more effective to drop expired or being expected to be expired before being delivered packets. Basically, time-critical applications may require delivery of various types of sensory data with different levels of real-time requirements depending on the dynamics of the sensed environment. For example, location sensory data for a fast moving target exhibits shorter TTL than that for a slow moving target.

Due to transmission environment wireless sensor networks are deployed in, providing real-time guarantees and reliable data delivery in wireless sensor networks is quite challenging. Most of existing real-time algorithms used for networks other than wireless sensor networks assume (i) availability of a reliable network (ii) no limitation on energy resource. However, these real-time algorithms cannot be directly applied to wireless sensor networks, which (i) usually suffer from unreliable nodes and links (ii) need to work on a low duty-cycle in order to save energy and operate for a long time. Low duty-cycle operation leads to orders of magnitude longer communication delays in comparison with traditional always-active networks, imposing a new challenge in many time-constrained applications. On the other hand, reliable data dissemination is traditionally guaranteed by applying error control approaches, which state additional communication delay. Basically, there are two key strategies in wireless sensor networks for maintaining reliable communication over noisy channels: (i) Forward Error Correction and (ii) Automatic Repeat Request. Forward error correction (FEC) relies on transmission of redundant data in order to make the receiver node capable of reconstructing the original data. Automatic repeat request (ARQ) relies on retransmitting

## 4.1 Introduction

---

a packet which has been missed or received erroneously. The main shortcomings of error control protocols are that they cost energy and lead to long delay which is not appropriate for the time-constrained applications. In this regards, both the transmission delay arisen from error control approaches and the delay caused by duty-cycle may be tolerable in some applications which do not demand timely data delivery.

In addition to all above mentioned challenges, another issue which is addressed in this chapter is burst-errors. Different from many dissemination approaches which assume such wireless channel whose errors are independent and random, in this chapter we put emphasize on the errors which are localized in short-term and occurs in burst forms.

The above considerations suggest that several factors including energy (duty-cycle), delay and reliability (error-burst) have to be taken into account while designing a protocol to disseminate time-constrained data. These factors usually interact heavily with each other, thus making the study of an efficient system configuration a challenging task.

In this chapter, we put a special emphasize on TTL as one of the packet-level constraint for data dissemination. In this regard, we propose a runtime adaptive packet-link-local error control for low duty-cycle time-critical wireless sensor networks, which can counteract the channel errors that may appear in burst for short-term.

One should note that by real-time, in this chapter we mean having a packet received by the base station within its TTL.

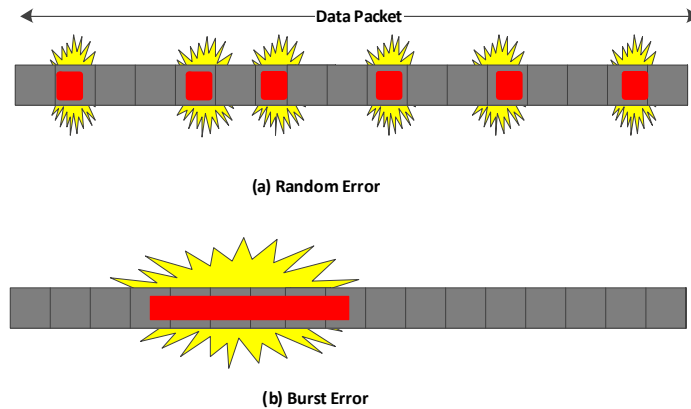
### 4.1.1 The need of adaptive approach

Due to dynamic nature of wireless sensor networks, it is not efficient to equip sensor nodes with fixed error control methods. Fixed parameters setting is only suitable and efficient when the environment conditions do not deviate considerably from the expectation prior to network deployment. Since the conditions of wireless sensor networks change often, the error control in use should be adjusted according to the links quality or channel condition while adhering to the packet-level requirements in terms of delay (TTL) and expected reliability. Basically, by inclusion in each packet some information about the application constraint TTL and providing for each node some information about the channel condition, each node is able to dynamically decide on the error control change in an on-demand manner.

### 4.1.2 Burst error vs. Random error

The knowledge of error nature in wireless channels and so ensuring reliability is an essential constituent of efficient communication protocol design especially for mission-critical applications. Wireless communication experiences both short-term fading and long-term fading. To have an effective error control approach, the characteristics of the communication channel should be taken into consideration. In this regard, in this section we briefly describe different type of errors which could occur in a wireless sensor network .

Common channel models are memory-less [1], in which the errors occur randomly and with a certain probability. On the other hand, dynamic channels may introduce another type of errors, which are localized in a short interval and occur in burst form. In case of having dynamic channels, errors happen in many consecutive bits rather than occurring in bits independently of each other. Figure 4.1 illustrates occurrence of random and burst errors in a packet.



**Figure 4.1. Random errors (a) vs. Burst errors (b) in a packet**

Bit and byte-level FEC work well against bit errors that are uniformly distributed over the transmission. They are, however, not efficient enough to deal with channels which experience varying loss rate over time. In this case, use of the mean loss rate to control FEC redundancy may not result in an effective error control. This is due to the fact that redundancy in FEC allows the receiver to detect only a limited number of errors that are predetermined with estimation of mean loss probability. Basically, the mean loss rate would be more beneficial if the length of the FEC code-word is large enough so that it could well-capture the effect of long-term loss process. However, the mean loss rate

## 4.1 Introduction

---

often fails to represent the short-term loss process fluctuations. Bit and byte-level FEC thus are less suited to handle recovery of burst errors caused by long-term fading and may expand over several packets. In this regard, they are unable to recover a completely lost or delayed packet.

To overcome the unreliability caused by long-term fluctuations, a more advanced loss characterization model should be used to capture the correlation between bit and packet losses [2]. Packet-level FEC and ARQ algorithms are stated to be competitive with bit and byte-level FEC in the presence of lengthy bursts of errors, possibly extending over several packets. This is due to the fact that this type of distortion typically occurs in a small percentage of packets and therefore could be covered by well-adjusting either the maximum number of retransmission in ARQ or the number of redundant packets in packet-level FEC.

Using byte-level FEC, a corrupted or erroneous packet will be detected and discarded either (i) at the link layer with cyclic redundancy check (CRC) or (ii) at the transport layer with checksum. In any case, it will not be available at the higher level to be recovered by the packet-level FEC. Moreover, single parity packet can be used by packet-level FEC to correct different single-packet losses in a group of packets.

Generally speaking, ARQ algorithms can more likely cope with the long-term fading if Equation (4.1) is true.

$$\theta^{Bst} < \theta^{ARQ} \quad (4.1)$$

where  $\theta^{Bst}$  represents the burst length in terms of time and  $\theta^{ARQ}$  represents the time required to have maximum number of retransmission which ARQ is allowed to perform. One should note that  $\theta^{ARQ}$  includes the time required for sending both data packets and acknowledgement packets (or timeout).

The same observation can apply for the packet-level FEC. Common FEC-based on Reed-Solomon codes or XOR functions can generally correct as many losses as the half of the redundancy packets [2]. In this regard, the number of redundancy packets can be adjusted according to the burst length.

### 4.1.3 Delay of error control schemes

As stated in Section 4.1.2, both packet-level FEC and ARQ can be adopted in such a way to combat long-term fading. However, for time-constrained data, the imposed delay due to using these error control protocols should be taken into consideration. Basically, the encoding unit of some error control approaches (e.g. most of packet-level FEC-based

schemes) requires a sequence of sampled data before starts the encoding procedure. Therefore, the small sampling rate may significantly affect the imposed delay of these approaches. In this regard, to not be affected by the delay of the small sampling rate, the error control schemes which are independent of sampling rate should be utilized. In what follows we elaborate on this issue.

The packet-level FEC-based approaches which are known as  $(n,k)$  encode  $k$  data packets into  $n$  code packets by adding  $n-k$  parity packets. If one packet is lost, the decoding and recovery procedure can begin by FEC as soon as  $k$  packets (either data or parity packets) are received. Therefore, the erroneous packet may require to wait as long as (i) the sensory data of all  $k$  data packets are generated by the sensing unit respecting the sampling rate (ii)  $k$  of  $n$  packets being received by the destination.

In this way, the delay arisen from sensing  $k$  sensory data is in reverse relationship with sampling rate  $SR$  and can be bounded by the value shown in Equation (4.2).

$$\frac{k-1}{SR} \quad (4.2)$$

On the other hand, the delay of receiving  $k$  packets of  $n$  packet is in direct relationship with both the number of packets in a code-word (i.e.  $|code\_word|=n$ ) and duty-cycle  $DC$ .

In short, the aforementioned relationships can be represented by Equation (4.3).

$$\delta_p^{FEC} \propto \left( \frac{k-1}{SR}, |code\_word|, \frac{1}{DC} \right) \quad (4.3)$$

Where  $\delta_p^{FEC}$  denotes the imposed delay using a FEC-based error control protocol.

Based on Equation (4.3), if sampling rate is high so that after transmitting one packet the next packet is available to transmit, the code-word length would be the determinative factor in delay. If duty-cycle is considerably low, the delay arisen from both sampling rate and code-word size would be dominated by the duty-cycle delay. Moreover, if TTL of the packet is in order of millisecond, second or minute and the inverse of sampling rate is in order of minute or hour and  $k>1$ , the erroneous or lost packets may not be recovered within their TTL.

One should note that according to Equation (4.3), achieving reliable transmission for time-constrained data by FEC become more challenging if the sampling rate  $SR$  is small and parity packets require more than one data packets (i.e.  $k>1$ ) in order to be built. However, if in Equation (4.3)  $k=1$  then  $\frac{k-1}{SR} = 0$  and thus the FEC-based approach is not affected by the sampling rate. The FEC approach with  $k=1$  is called Repetition code [1].

## 4.2 Assumptions and models used

---

The idea of the repetition code which is known as a (n,1) is to transmit each data packet  $n$  times. In other words, we should send  $n$  copies of a data packet, consecutively.

On the other hand, the delay of ARQ is independent from sampling rate and just bounded by the duty-cycle and the maximum number of retransmissions which ARQ is allowed to perform, as shown in Equation (4.4).

$$\delta_p^{ARQ} \propto \text{Max}(\zeta, \frac{1}{DC}) \quad (4.4)$$

Where  $\zeta$  represents the maximum number of retransmissions.

Basically, the packet-level FEC with  $k>1$  is less suited to be utilized in the real-time application specially if  $SR$  is too small with respect to the TTL time unit. The main reason is that  $k$  packets should be buffered before the encoding and decoding process in the application layer get started. Generally speaking, packet-level FEC with  $k>1$  should be employed for the applications which better tolerate the imposed delay of sampling task. Therefore, even though both packet-level FEC and ARQ are suitable to combat long-term fading, ARQ and Repetition code (i.e. FEC with  $k=1$ ) are the most promising one to provide reliability for time-constrained packets.

The rest of this chapter is organized as follows. First we explain the assumptions and models used in Section 4.2, which is followed by the related work in Section 4.3. Then in Section 4.4, we describe the problem statement and our contribution. We elaborate on our propose READ protocol in Section 4.5. Then in section 4.6 we present the simulation setup and performance evaluation results and finally in Section 4.7 we present the chapter summery.

### 4.2. Assumptions and models used

The assumptions and network model that our proposed approaches are built upon are to large extent similar to the model presented in Section 3.4 of Chapter 3. However, here we add some extra considerations as follows:

Our proposed protocol functions over the chain-clusters that are constructed by REC+, which was elaborated on in Chapter 3. By doing so and without loss of generality (as our approach could be applied for each chain/cluster independently), in this chapter we only concentrate on the chain inside the dashed red rectangle of Figure 4.2. It is worth recalling that in REC+, without having any prior assumption, the cluster-heads of two adjacent clusters can directly communicate with each other.

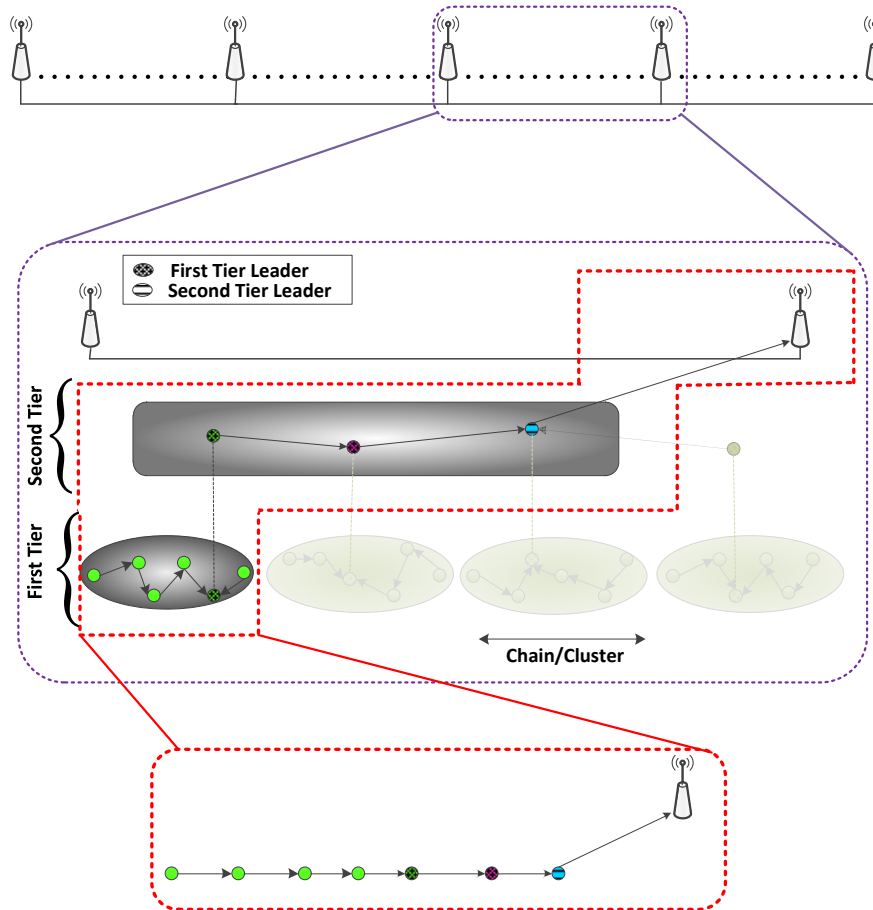


Figure 4.2. Two-tier architecture model

- Data packets are stamped with an auto-incremental sequence number and the copies of a given packet are also stamped with an auto-incremental copy sequence number.
- The sampling rate is assumed to be too small with respect to packet's TTL. Therefore, when one data about the phenomena is sampled, the TTL of the previous data has already expired. This assumption induces that each sensory data should be sent as soon as it is sampled and thus should not wait for the next data to be sampled and sent all together.
- TTL is one or more orders greater than the toggle period. Toggle period is the time interval between subsequent transition of the radio state into sleep (or active) state and is obtained by summing up one sleep period and one active period of the radio.

## 4.2 Assumptions and models used

---

- The interference model studied in this chapter includes the two-hop interference model [3].
- The quality of communication links are assumed to be symmetric.
- Errors are assumed to appear in burst forms for short intervals so that they may influence one or more consecutive packets. In this way, the dwell times of the noise (or the burst length) can be one or a few order greater than the packet transmission time.
- The channel is deemed to vary slowly with respect to the data transmission rate, and thereby the channels state transitions occur infrequently. In other words, the conditions which cause the channel noises is supposed to change slowly in comparison to the rate at which data are transmitted.
- Since the average loss ratio fails to capture long-term loss process fluctuations, we use Gilbert–Elliott model [4] which captures the correlation between losses and could well represent the burst losses. It is noteworthy that this model uses the probability of loss state instead of average loss rate to model communication channel.

### 4.2.1 Channel model

In wireless networks, the cause of packet loss could become more complex and dynamic so that the frequency of the error bursts varies over time.

Gilbert-Elliott model [4] is the most widely known burst model. The Gilbert-Elliott model is basically a hidden Markov chain in nature and in each state it acts exactly like a Binary Symmetric Channel (BSC) [5]. Each state has a probability distribution over the possible output tokens. Therefore, each state has an associated independent PER probability, i.e.,  $P_G$  for the Good state and  $P_B$  for the Bad state, and state transition probabilities could be derived from the experimental channel data as  $P(B|G) = p$  and  $P(G|B) = q$ . Hence,  $1 - p$  and  $1 - q$  are the probabilities of remaining in the same state, namely, in the good and bad state, respectively.

We use a Quasi-Stationary Gilbert-Elliott (QSGE) model, as shown in Figure 4.3, in order to model channel states. Each state  $S_v$ , which corresponds to a specific packet error rate  $PER_p$ , follows a Gilbert-Elliott model with some probabilities ( $p$  and  $q$ ) associated to it. The B (Bad) and G (Good) states are also a series of Bernoulli trials.



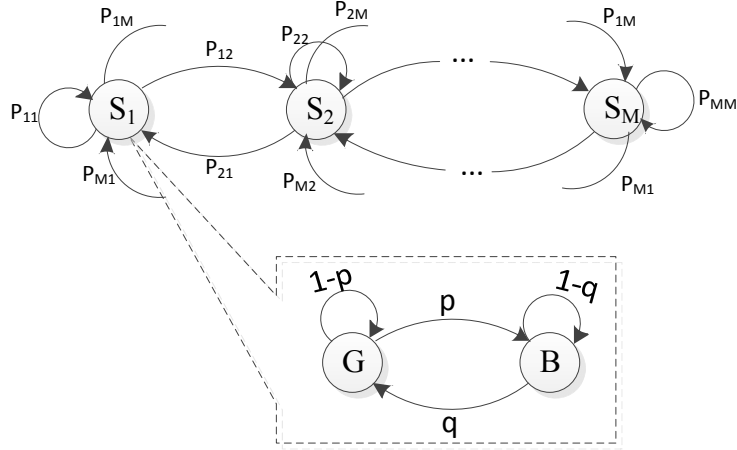


Figure 4.3. Quasi-stationary Gilbert-Elliot model

Each state  $S_i$  represents the expected PER  $r_p^i$  so that  $r_p^1 < r_p^2 < \dots < r_p^M$ , and the conditional one step probabilities of going from channel state  $S_i$  to channel state  $S_j$  is given by  $P_{ij}$ . The channel could be described in form of a transition matrix with entries as cross over probability over all combination of states. If  $S = \cup_{i=1}^M S_i$ , for any pair  $(i, j) \in S \times S$ , let  $P_{ij} = \Pr(j|i)$  denotes the probability that the channel would be in state  $S_j$  in the next interval providing the current state is  $S_i$ .

The corresponding state transition matrix ( $\Gamma = \{P_{ij}\}$ ) of the Figure 4.3, that governs the process of how the channel introduces different error rates, is expressed as:

$$\Gamma = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1(M-1)} & P_{1M} \\ P_{21} & P_{22} & \dots & P_{2(M-1)} & P_{2M} \\ P_{31} & P_{32} & \dots & P_{3(M-1)} & P_{3M} \\ \vdots & \vdots & & \vdots & \vdots \\ P_{M1} & P_{M2} & \dots & P_{M(M-1)} & P_{MM} \end{bmatrix} \quad (4.5)$$

As previously mentioned we assume a slowly varying channel. In order to model this issue, the channel state transitions in our M-state channel model need to occur infrequently. Furthermore, this slowly varying progression corresponds to the movement from one channel state to the adjacent channel states. Although each state could still communicate with any other states, however the farther the current and next states, the less likely that the next state can be reached from the current state. This could be modeled by assigning gradually regression transition probabilities from the current state

### 4.3 Related work

---

to the farther states. Therefore, we define the following relationship among transition probabilities:

$$P_{i,j} \gg P_{i,j+1} \gg P_{i,j+2} \gg \dots \gg P_{i,M} \quad i \leq j, \forall i, j \in [1, M] \quad (4.6)$$

$$P_{i,j} \gg P_{i,j-1} \gg P_{i,j-2} \gg \dots \gg P_{i,1} \quad i > j, \forall i, j \in [1, M]$$

### 4.3. Related work

Several data dissemination protocols have been proposed for wireless sensor networks in the past. However, only a very few of them consider both reliability and timeliness to be ensured simultaneously. Real-time guarantees are usually provided through either real-time scheduling or real-time routing. Real-time scheduling is typically fulfilled by the real-time MAC protocols which bound the one hop delay. Basically, duty-cycle in the MAC layer may cause longer than one hop delay compared with when no duty-cycle is considered. Besides a real-time MAC, real-time routing is also required to achieve a real-time system. Real-time routing schemes normally select the path which could guarantee the packet deadline.

SPEED [6] is a well-known protocol addressing soft real-time guarantee in wireless sensor networks in such a way that packet deadline is mapped to a packet speed requirement. The node with a speed higher than a specified requirement is more likely to be chosen as the upstream node. The SPEED protocol actually provides soft real-time guarantees by keeping a uniform delivery speed in the entire network using feedback control. One should note that SPEED just reduces the deadline miss ratio for time-constrained data which are supposed to be relayed through high reliable links of an always-active sensor nodes.

MMSPEED [7], which is an enhanced version of SPEED, aims to meet reliability and timeliness requirements together. In order to provide reliability, it utilizes multi-path routing in such a way that the number of paths is in direct relation with the required reliability. Moreover, real-timeness is supported by combining the SPEED idea with packet prioritization, which is done on the basis of the required speed for each packet.

R2TP [8] uses a reliable and real-time data dissemination, in which reliability is satisfied by sending several copies of one packet through multiple paths such that sum of the reliability of the considered paths is equal or higher than the requested reliability. The packet is dropped by the intermediate nodes if the elapsed time of a given node is

greater than the delivery time requirement. Otherwise, it forwards that packet through multi paths using the given node's table, which stores the delay of different paths.

Soyturk et al. [9] present a reliable data acquisition approach for time-critical application of wireless sensor networks. Reliability is provided similarly to techniques of [7, 8] leveraging multipath approach, while real-time concern is supported by prioritization of the packets. This technique, therefore, deals with the priority scheduling in order to handle queuing delay, which is the main cause of end-to-end delay.

Even though the impact of unreliable links is addressed in a number of abovementioned real-time routing algorithms, most of them ignore the duty-cycle matter and thereby assume always-active sensor nodes, which is not energy efficient at all. On the other hand, even though utilizing low duty-cycle MAC protocols will result in energy saving, it will have negative influence on the delay performance of routing approaches. Therefore, the delay imposed by such MAC protocols should be considered in the design stage of real-time routing schemes.

Moreover, most of abovementioned solutions assume no interference model and disregard the collision that may happen because of simultaneous packet sending/receiving of the adjacent sensor nodes being located in communication range of each other. This collision could considerably impact the deadline of the packet.

There are some studies [10, 11] that address the impact of duty-cycle on disseminating time-constrained data but they do not take the low quality links into consideration. Even though [12] takes both duty-cycle and links quality into account which leads to high delivery ratio and short delay, it does not consider the expected deadline of the packets and so the assumed delay is unbounded.

WSEDR [13] is a synchronous forwarding scheme for ultra-low duty-cycle systems, which relies on a distributed wakeup scheduling algorithm. It schedules the wakeup time of each sensor node according to both the hop count between the sensor node and the base station, and the expected end-to-end delivery ratio depending on the link probability. RTCF [13] is an asynchronous version of WSEDR, which adopts the preamble sampling mechanism to communicate and dynamically adjust the number of potential relay nodes based on the packet deadline. Basically, preamble sampling technique is employed by RTCF, since a sender does not know the exact wakeup time of its neighbors. Both WSEDR and RTCF guarantee the end-to-end delivery ratio for delay-constrained packets by well scheduling sensor nodes according to the amount of delivery ratio they could provide and then transmitting packets through multi-path. The sensor nodes in the more reliable path are scheduled to wake up earlier in order to relay the time-constrained-packets. Both these cross-layers approaches rely on multipath

#### 4.4 Problem statement and our contribution

---

routing in such a way that sum of the delivery ratio of all paths could meet the required delivery ratio. However, these approaches cannot be so promising for a chain-based topology in which most likely no more than one path is available between sender and receiver, and therefore only one way of scheduling is possible.

Different from all aforementioned protocols, RPAR [14] adjusts the transmission power according to the packet deadline. Basically, RPAR takes the benefit of CC2420 [15] whose the off-the-shelf sensor radio could support several power level. The shorter deadline, the larger transmission range and thereby the higher power level. MCRT [16] exploits both multi-power level and multi-channels to better reduce the packets' deadline miss ratio.

To the best of our knowledge, there is no well-explored work to address reliability and timeliness together in a duty-cycled chain-based wireless sensor network, in which most of the times only one path can be established between source and destination nodes. Moreover, as almost all of the aforementioned approaches support reliability by sending several copies of a packet through different paths, they should employ some other methods to filter out redundant data in case of having duplicate sensitive aggregation functions like as SUM or AVERAGE. To address above challenges, an error control is needed which is able to (i) shorten the delay of the ARQ, (ii) alleviate the impact of lost acknowledgements, (iii) maintain the reliability of ARQ, (iv) ensure energy efficiency, and (v) guarantee the packet delivery before their TTL expires.

#### 4.4. Problem statement and our contribution

Even though as discussed in Section 4.1.3, ARQ is more suitable than FEC (when  $k > 1$  and  $SR$  is too small with respect to TTL) to ensure reliable delivery of the time-constrained data over bursty channels, it suffers from losing acknowledgement packets which may bring inefficiency. In case of losing acknowledgement packets, source node may continue sending copies of the received data, which leads to high energy dissipation and bandwidth wasting. Similarly, if NACK packets are lost, sender will never be informed about erroneous or loss packets and thereby the reliability cannot be ensured. Moreover, the idle time spent waiting for the acknowledgement packets is a waste of time especially for delay-constrained data.

To eliminate the delay and transmission overheads introduced by acknowledgements in ARQ, we can ensure reliability via advance transmission of packet copies by the sender without sending any acknowledgement. In a sense, this can be viewed as a rudimentary form of FEC using so-called Repetition codes.

Having the requested reliability  $\theta^\rho$  by the application for each link and the packet loss rate  $\varphi_{ij}$  for the link  $i,j$  which connects sensor node  $i$  to sensor node  $j$  and vice versa, we could estimate the optimal number of packet copies  $\zeta_{ij}$  that should be transmitted through the link  $i,j$  as stated in Equation (4.7).

$$\zeta_{i,j} = \log_{\varphi_{ij}}^{1-\theta^\rho} \quad (4.7)$$

Even though this approach reduces the acknowledgement overhead and delay, it still may not ensure the TTL requirement of the packet and thereby we cannot utilize this way of estimating number of copies. Ideally, the number of packet copies should be adopted in such a way that beside providing required reliability, the TTL requirement of the packet is also satisfied.

A key question here is how to assign the remaining TTL of a given packet to the intermediate nodes so that a high reliability gain and on-time end-to-end delivery ratio for a duty-cycled network can be achieved. In other words, for how long can a packet be delayed on each intermediate node to improve transmission reliability, so that the application related goals could still be obtained. To this end, in this chapter we are aiming at providing a reliable scheme to guarantee real-time delay bound for the low duty-cycle wireless sensor networks.

We summarize our contribution related to this chapter in three folds as:

1. Proposing a fair and effective heuristic which allocates the available packet TTL to the sensor nodes, proportionally to the packet loss probability of the links along the forwarding path for the low duty-cycle wireless sensor networks.
2. Proposing READ, a runtime adaptive packet-link-local error control protocol which operates based on the links' qualities, packets' TTL, and duty-cycle and is able to counteract periodic short-term burst-errors in a linear topology
3. Investigating the relatively unexplored topic of impact analysis of TTL and link reliability parameters on network performance in terms of attained hit ratio, for three approaches, i.e., READ, ARQ and Simple, to assess the appropriateness of each method facing different conditions.

#### 4.4.1 Simultaneous real-timeness and reliability

To motivate the need to simultaneously address data reliability and real-timeness in our protocol, let us consider the network illustrated in Figure 4.4, which consists of six sensor nodes one of which is selected as the chain-leader ( $S_2$ ). A packet whose TTL is 6

#### 4.5 Reliable disseminating time-constrained data (READ)

second, should be forwarded from  $S_0$  towards the base station. Let us assume that the time required to deliver a packet from  $S_0$  to the leader is  $2s$  and from the leader to the base station is  $1s$ . Clearly, this packet will be received by the base station after  $3s$ . This implies that  $3s$  from its TTL is remained, which can be exploited to achieve higher network performance. We can spend this amount of time for either (i) increasing aggregation degree of the leader or (ii) improving transmission reliability of the network.

If the network has high reliable links and it is almost guaranteed that the packet will be received by its destination through the first transmission, it is better to spend this remaining time for the aggregation process and to increase the aggregation degree of the leader. In this case, the leader can put the received packet on hold and wait for the en-route packets which will arrive to the leader within the limited remaining time of the given waiting packet. Doing so, more packets could contribute in the aggregation process and thereby the aggregation degree increases.

The remaining TTL time can also be utilized by the error control approaches in order to improve the transmission reliability. In this way, dependent on the remaining TTL, several copies of a packet could be transmitted from the source node toward the destination. This is particularly useful and effective if the network suffers from unreliable links and the application requires a certain amount of reliability per hop.

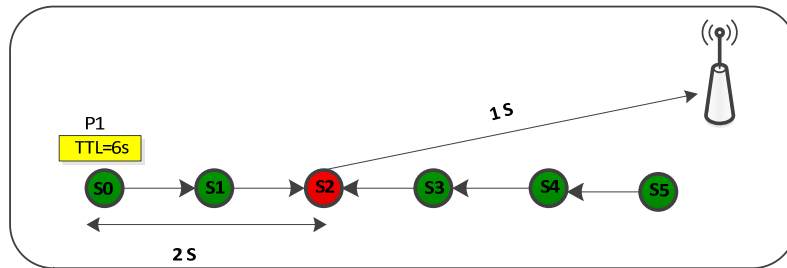


Figure 4.4. An example of a chain based network

Since our assumption is that the network is unreliable, we spend the remaining TTL time for the purpose of error controlling to increase the end-to-end delivery ratio.

#### 4.5. Reliable disseminating time-constrained data (READ)

In this section, we present READ which is a FEC-based or Repetition-based error control protocol to ensure reliability for delay-constrained data over a bursty channel in a chain-based wireless sensor network. To this end, we estimate the optimal number of

packet copies for each link taking both link reliability and packet's TTL into account. Since receiving a packet after its deadline may not only be useless but also depletes energy, it is preferable to drop such packets to prevent wasting energy of the intermediate nodes relaying them. In this regard, we fairly adjust the number of packet copies proportionately to the packet loss probability of the links, in a duty-cycle-based network as far as the packet's deadline is not expired.

We already mentioned that READ operates over the chain-clusters which are constructed by REC+ which is elaborated on in Chapter 3. Therefore, we do not repeat the details of how clusters and chains are constructed in the first and second tier, and how the leaders of the first and second tier are elected.

Given a chain (dashed red rectangle of Figure 4.2) with a leader who is able to communicate directly with the base station, our proposed READ consists of three phases for disseminating data; (i) initialization, (ii) situation-aware data gathering and (iii) updating nodes' fractional portion from time slots. The flowchart diagram of READ is presented in Figure 4.5.

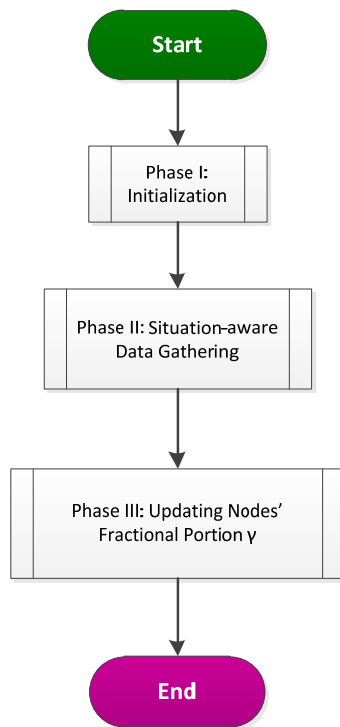


Figure 4.5. Flowchart diagram of READ

## 4.5 Reliable disseminating time-constrained data (READ)

---

Each of these three phases consists of some processes which should be performed either in the base station or in the sensor nodes. In what follows we elaborate on each phase. The flowchart diagram of these three phases are shown in Figure 4.6, Figure 4.7 and Figure 4.8, respectively.

### 4.5.1 Initialization

Upon deployment and after making the two-tiers architecture for data dissemination, the base station asks sensor nodes to evaluate their one-hop adjacent links' quality in terms of transmission reliability. Sensor nodes determine their link quality by exchanging control/beacon packets. They then send out the quality of their links to the base station through first tier and second tier nodes. The base station after receiving nodes information (i) first makes TDMA and activity schedule of each node by utilizing the streamlined wake up schedule [17] (ii) then calculates the fractional portion of each node, as will be explained in Section 4.5.1.1., from TTL.

#### 4.5.1.1 Calculating fractional portion of nodes

We propose a straightforward and fair heuristic, which allocates the available packet TTL proportionally to the packet loss probability of the links along the forwarding path. This way of distributing TTL makes READ capable of fairly using the packet TTL on intermediate nodes so that high reliability gain and on-time end-to-end delivery ratio are achieved.

In order to calculate the portion of each node  $i$  from the available TTL we introduce Equation (4.8).

$$Y_i = \frac{\varphi_{i,i+1}}{\varphi_{l,BS} + \sum_{k=i}^{l-1} (\varphi_{k,k+1})} \quad (4.8)$$

where  $Y_i$  denotes the portion fraction of each node  $i$  from the remaining timeslots of any received packet and  $\varphi_{i,j}$  which was introduced in Equation (4.7) denotes the packet loss rate of link  $i,j$ . One should notice that  $\varphi_{l,BS}$  denotes the packet loss ratio between leader  $l$  and base station BS.

According to Equation (4.8), the higher link's quality brings the smaller portion of timeslots for the given link to be utilized in order to improve transmission reliability. Therefore, using Equation (4.8) the timeslots are fairly distributed among sensor nodes proportional to their links' quality.



4.5.1.2 Disseminating initializing information

The base station, in the next step, informs the sensor nodes about (i) their associated fractional portion (which is calculated in Equation (4.8)) to utilize for the upcoming packets (ii) their associated activity schedule.

The flowchart diagram of initialization phase is shown in Figure 4.6.

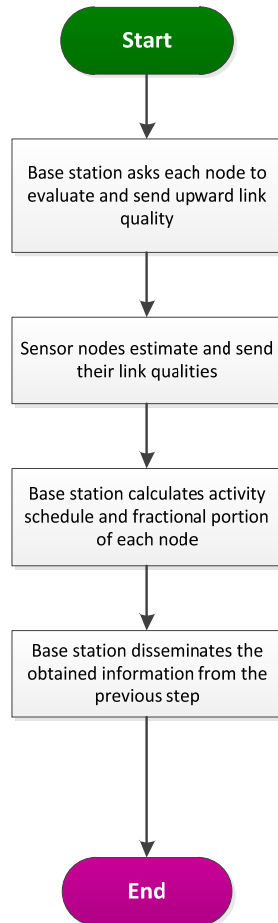


Figure 4.6. Flowchart diagram of phase I: initialization

4.5.2 Situation-aware data gathering

In the data gathering phase, first READ involves the process being performed by the sensor nodes, to either map TTL into time slots or update TTL. In this sense, depending

## 4.5 Reliable disseminating time-constrained data (READ)

---

on the role of the nodes which could be either source node or relay node, they are asked to (i) represent TTL in terms of time slots as will be explained in Section 4.5.2.1.1. or (ii) update TTL as will be described on Section 4.5.2.1.2., respectively. Thereafter, sensor nodes should opt for an appropriate strategy based on which the available TTL is effectively distributed among different sensor nodes. This strategy will be expanded in Section 4.5.2.2.

### 4.5.2.1 TTL adjusting

#### 4.5.2.1.1 Mapping TTL to the time slots by the source node

For the sake of simplicity, READ represents packet's TTL in terms of number of time slots during each of which only one packet or one packet copy could be transmitted. Therefore, in this section we elaborate on representing TTL in the form of number of time slots for a duty-cycle based approach.

Every source node which initiates sending a data packet  $p$  with a specific  $TTL_p$  should first turn the TTL into a number of time slots. Leveraging Equation (4.9) source node could calculate the maximum number of time slots  $\tau_p$  during which the TTL of the packet is still satisfied.

$$\tau_p = \frac{TTL_p}{TT} \quad (4.9)$$

where  $TTL_p$  denotes the TTL of packet  $p$  and  $TT$  denotes the amount of time required to transmit one packet one hop and is called propagation delay. However, since READ considers duty-cycling for the nodes it should therefore address the nodes' sleeping-time which may greatly influence remaining TTL of the packet. As Equation (4.9) is obtained irrespective of duty-cycle, it is not useful for a duty-cycle based protocol. Since one node cannot communicate upward if its upstream neighbor is not active yet, therefore to deliver time-constrained data in a duty-cycle network, communication should carefully be managed among sensor nodes. Due to the streamlined wake-up schedule that our protocols rely on, we are sure that once one node sends the first copy of the packet to its upstream node, that node is awake at that time. However, it is likely that the upstream node goes to sleep mode before transferring all copies of a given packet. The sender therefore should wait for the next wakeup interval when the sensor nodes are active again and so are able to receive the data from downstream neighbors. In this case, the sleep time should be reduced from the remaining TTL. Therefore, in order to address duty-cycle in calculating  $\tau_p$ , the Equation (4.9) should be replaced with Equation (4.10):

$$\tau_p = \frac{Rt}{TT} + n^\Gamma \times n^{Tr} \quad (4.10)$$

$$Rt = TTLp - (\Gamma \times n^\Gamma)$$

$$n^\Gamma = \left\lfloor \frac{TTLp}{\Gamma} \right\rfloor$$

$$n^{Tr} = \frac{AwT}{TT}$$

$$AwT = DC \times \Gamma$$

where  $\Gamma$  represents the toggle period (i.e. the time interval between subsequent transition into sleep (or active) state),  $TT$  denotes the amount of time required to transmit one packet and is correspond to one time slot,  $DC$  shows the duty-cycle and  $AwT$  signifies the awake (active) time period. Moreover,  $n^\Gamma$  denotes the number of toggle periods during which the TTL of the given packet  $p$  has not expired yet, and  $n^{Tr}$  represents the number of transmissions that could be accomplished during one active period. As according to our assumptions we are allowed to send (or receive) each copy of one packet in one time slot, the number of time slots corresponds to the number of packet copies throughout the network.

In other words,  $\tau_p$  represents the maximum number of time slots that each packet  $p$  with a specific TTL requires to transmit all its copies along the path towards the base station while TTL requirement is met and reliability is improved.

#### 4.5.2.1.2 Updating TTL by Relay Node

Each sensor node  $i$  upon receiving the packet  $p$  should first update the remaining TTL (or time slots) of the given packet as in Equation (4.11). In this regards, the receiver node  $i$  needs to find out the sequence copy number  $C_{p,i-1}$  of the packet  $p$  which is sent by node  $i-1$ . Basically, the sequence copy number states the number of copies which have already been sent by node  $i$  for the packet  $p$ . Therefore, the available timeslots of the packets should be updated by subtracting the amount of time slots requires to send  $C_{p,i-1}$  packet copies from the available TTL  $\tau_{p,i}$ . The main reason of using sequence copy number  $C_{p,i-1}$  is that we do not know which copy is received by the node  $i$ . Basically, considering the copy sequence number of the received packet, the available timeslots should be changed accordingly. Obviously, the copy sequence number varies between 1 and  $Y_{i-1}$ .

$$\tau_{p,i} = \begin{cases} \tau_p & \text{if } i = \text{source node} \\ \tau_{p,i-1} - C_{p,i-1} & \text{otherwise} \end{cases} \quad (4.11)$$

$$1 \leq C_{p,i-1} \leq Y_{i-1}$$

##### 4.5.2.2 Fairly allocating time slots to nodes

After finding/updating  $\tau_p$  which signifies the TTL of the packet  $p$  in terms of number of time slots, we need an appropriate strategy based on which the time slots are effectively distributed among different sensor nodes. For this purpose, we discuss in detail the policy that READ adopts to fairly utilize the packet TTL on intermediate nodes so that a high reliability gain and on-time end-to-end delivery ratio are achieved.

Every sensor node  $i$  which initiates/receives the packet  $p$ , should find out its portion from the available timeslots of packet  $p$  by employing Equation (4.12). The variable  $Y\tau_{p,i}$  states the maximum number of time slots which are allocated to node  $i$  in order to send as many copy as possible of packet  $p$ . In other words,  $Y\tau_{p,i}$  is the maximum number of copies that node  $i$  is allowed to transmit for packet  $p$  in order to improve reliability.

$$Y\tau_{p,i} = \min(Y\tau'_{p,i}, \zeta_{i,i+1}) \quad (4.12)$$

$$Y\tau'_{p,i} = Y_i \times \tau_{p,i}$$

where  $Y_i$  is obtained by Equation (4.8). Basically, since data packets may have different TTL requirement  $\tau_p$ , sensor node  $i$  multiplies  $Y_i$  by the TTL  $\tau_{p,i}$  of any received packet in order to find out its own portion from the available timeslots of the given packet .

In order to put an upper bound for the number of packet copies  $Y\tau_{p,i}$ , we use  $\zeta_{i,i+1}$  which is obtained using Equation (4.7). It is worth recalling that  $\zeta_{i,i+1}$  shows the maximum number of copies which need to be sent by node  $i$  in order to ensure the reliability requested by the application for the upward link  $(i,i+1)$ . Even if the TTL of the packet is extremely high but sending copies more than  $\zeta$  that is stated to be enough for achieving the requested reliability of the application, is just wasting the network resources. Therefore, to avoid allocating unnecessary time slots and to be more resource efficient we put an upper bound over the number of packet copies, by involving  $\zeta_{i,i+1}$  in Equation (4.12).

4.5.2.3 Disseminating data packets

The flowchart of situation-aware data gathering phase is shown in Figure 4.7.

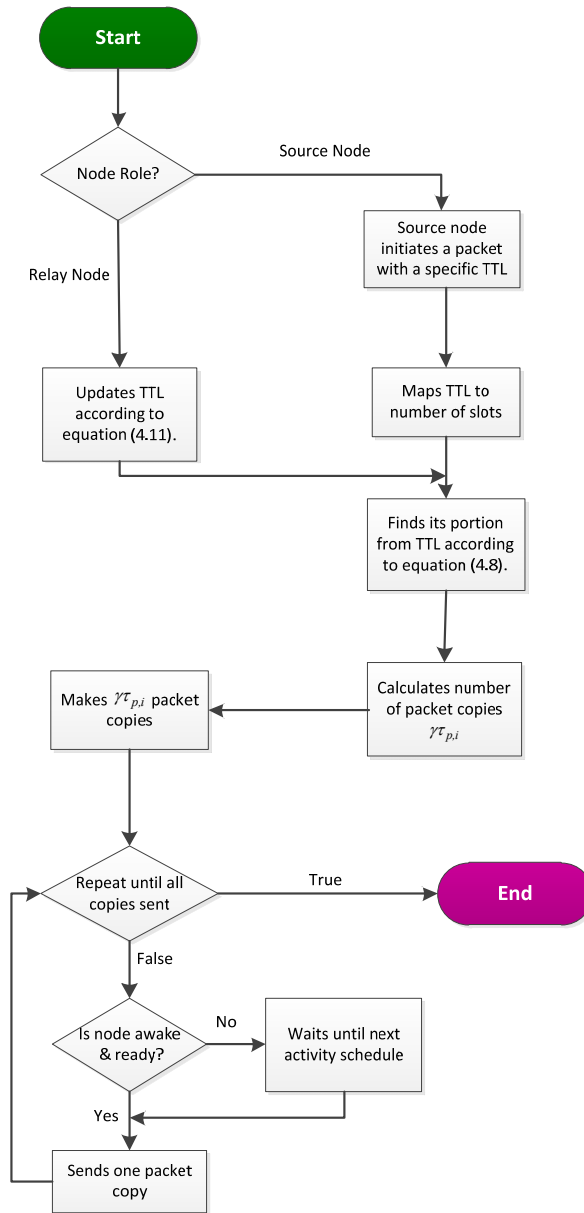


Figure 4.7. Flowchart diagram of phase II: situation-aware data gathering

## 4.5 Reliable disseminating time-constrained data (READ)

---

Basically, after finding  $\Upsilon\tau_{p,i}$  which states the maximum number of copies that node  $i$  is allowed to transmit for packet  $p$  in order to improve reliability, sensor node  $i$  makes  $\Upsilon\tau_{p,i}$  copies of the packet  $p$  each of which consists of sensory data along with the available TTL  $\tau_p$ . Then, sensor node  $i$  according to its activity schedule transmits all packet copies upward and toward the base station. If the application asks for the aggregation, each sensor node  $i$  should also perform aggregation over the received data with its own.

### 4.5.3 Updating nodes' fractional portion from time slots

To deal with inherently non-deterministic quality of wireless links while adhering to the delay requirements of the packets, packet loss rate of the links need to be continuously updated. Any significant change on the links' quality may influence the fractional portion of each node from the upcoming packets' timeslots. Therefore, adjusting the nodes' fractional portion  $\Upsilon$  according to the quality of links through which the time-constrained packets are transmitting would be essential to have an effective approach.

Each sensor node can be aware of its links' quality based on packet sequence number, copy sequence number, timestamp fields of the packets over a window  $\omega$ . Link quality is calculated by averaging the number of received packets (or copies) over a window of  $\omega$  packets at time  $t$ . The window size can be obtained using Equation (4.13).

$$\omega_{i+1}^t = \zeta_{i,i+1}(\varphi^{wrst}) \quad (4.13)$$

where  $\zeta_{i,i+1}$  represents the optimal number of packet copies calculated by Equation (4.7) for link  $(i, i+1)$  based on the worst loss rate  $\varphi^{wrst}$  that can happen. Since we assume the channel state transients among some specific states with the specific loss rates, finding the worse loss rate is not difficult. We use a weighted average in order to calculate loss rate in the time window. In this way, we give more weights on more recent observations and thus the estimation can be more adaptive to temporal changes.

Updating links reliability which has direct impact on the nodes' fractional portion can be accomplished either at the base station which has a global view of the whole network or at the sensor nodes which have a local view of their neighboring links.

In what follows we expand on these two different ways of updating while the main advantages and limitations of them are briefly discussed.

### 4.5.3.1 Central updating $Y$ in the base station

Since sensor nodes only have local information about the links quality, the base station may seem to be the best place to undertake the responsibility of finding the optimal fractional portion  $Y$  of a given packet for each node. In this way, the base station needs to have the most recent packet loss statistics of all links. Therefore, sensor nodes should send the packet loss rate of their links to the base station either in some predefined time instants or as soon as a significant change takes place. For the sake of communication-efficiency, only the nodes whose links' qualities undergo a significant change, report the state of their links in terms of reliability. By doing so, the chain leader is aware of the up-to-date quality of all links in terms of reliability. It is worth mentioning that to conserve energy we could piggyback the statistics regarding links' quality with the reported data which are periodically sent towards the base station.

Thereafter, the base station calculates the new fractional portion of each node by using Equation (4.8) while taking the latest reported state of all links into account and disseminates the new fractional portions to sensor nodes whose this parameter has been changed.

Central updating at the base station would be inefficient if the transmission links frequently experience quality variation. In this case while sensor nodes are waiting to receive their new fractional portions which are adjusted according to their latest links quality, they may undergo another change in their links quality. Moreover, if the packets which carry the statistical information about links are not received or partially received by the base station, no new fractional portions or even wrong fractional portions would be disseminated. This becomes more challenging if the size of wireless sensor network grows. To cope with these raised issues, one promising alternative would be locally updating packet loss rate in each sensor node whose links experience quality variation.

### 4.5.3.2 Local updating $Y$ in the sensor nodes

In the local updating scheme, each individual sensor node is in charge of adjusting its fractional portion from the available time slots of the received packets as soon as the quality of its upward link change.

After finding the packet loss rate, the given sensor node  $i+1$  informs the downstream node  $i$  about the new link's quality if needed.

## 4.6 Performance evaluation

---

As soon as sensor node  $i$  becomes aware of the new packet loss rate of its upward link, it changes its fractional portion from  $\tau_p$  for the upcoming packet(s)  $p$  by employing Equation (4.14) :

$$Y_i^{new} = Y_i^{old} \times \frac{1 - \rho_{i,i+1}^{new}}{1 - \rho_{i,i+1}^{old}} \quad (4.14)$$

where  $Y_i^{old}$  and  $Y_i^{new}$ , according to Equation (4.8), represent the portion of node  $i$  from available timeslots before and after updating link's quality, respectively. Moreover,  $\rho_{i,i+1}^{old}$  and  $\rho_{i,i+1}^{new}$  also denote the amount of reliability of the link  $i,i+1$  before and after updating.

Locally updating packet loss rate may increase the ratio of the number of received packets to the total packets. However due to lack of a global view, it is possible when the cumulative error in the network increases as shown in Equation (4.15), the TTL of some of the received packets expire.

$$(1 - \rho_{l,BS}^{new}) + \sum_{j=1}^{l-1} (1 - \rho_{j,j+1}^{old}) < (1 - \rho_{l,BS}^{new}) + \sum_{j=1}^{l-1} (1 - \rho_{j,j+1}^{new}) \quad (4.15)$$

The flowchart diagram of updating nodes' fractional portion phase is shown in Figure 4.8.

### 4.6. Performance evaluation

In this section, we evaluate our algorithm and compare it with ARQ and a simple transmission approach. The ARQ we simulate is a hop-by-hop error control mechanism, which provides reliability by sending acknowledgements. For the sake of completeness, we also compare our protocol with a so called Simple protocol that does not utilize any error control mechanism. Therefore, in the Simple protocol only the sensory data without any parity or redundancy is aggregated and forwarded along the path toward the destination. One should note that we do not utilize the acknowledgement feature of CC2420 and implement it ourselves. In the simulation the size of the data packet and the size of the acknowledgement packet is considered the same.

#### 4.6.1 Performance metrics

We consider a number of metrics which are listed below to evaluate the performance of our approach under different circumstances.



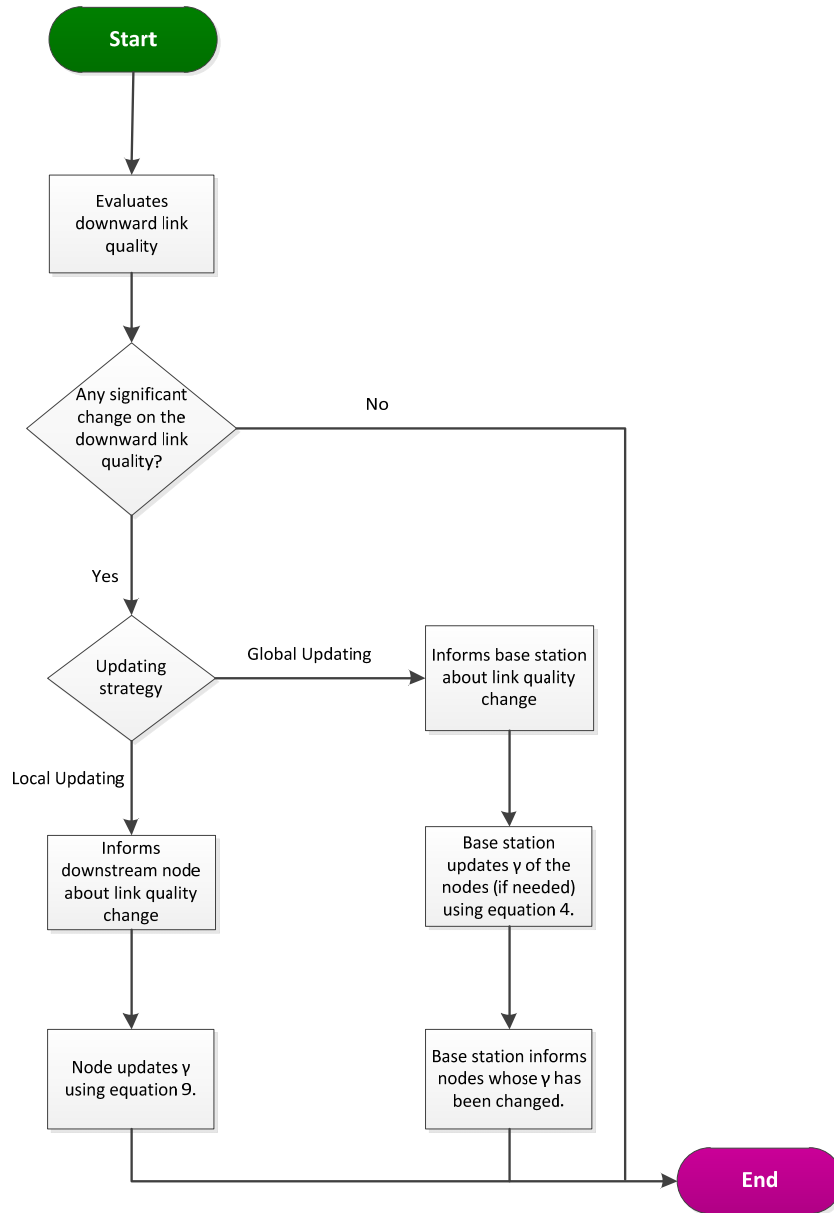


Figure 4.8. Flowchart diagram of phase III: updating nodes' fractional portion

**Deadline Hit Ratio:** This metric indicates the efficiency of a real-time protocol and is defined as the percentage of the source packets reach the base station within their

## 4.6 Performance evaluation

---

deadline to the total number of source packets which are destined toward the base station.

**Unwasted Energy:** This metric  $E_r$  shows the total amount of energy spent to disseminate the source packets that are received in-time by the base station. This metric does not include the energy of the redundant packets (packet copies in READ or acknowledgement and retransmitted packets in ARQ).

**Total Energy:** This metric shows the total amount of energy spent to send all packets; original data, copies or acknowledgement as Equation (4.16).

$$E_T = E_r + E_{ur} + E_{rnit} + E_{or} \quad (4.16)$$

where  $E_r$  denotes the energy spent for the source packets received on-time,  $E_{ur}$  states the energy consumed for the source packets which are not received by the base station and are dropped/lost en-route. Moreover,  $E_{rnit}$  shows the energy dissipated for the source packets which are received but expired and  $E_{or}$  denotes the energy dissipated for either the packets copies in READ or retransmitted packets or acknowledgement in ARQ.

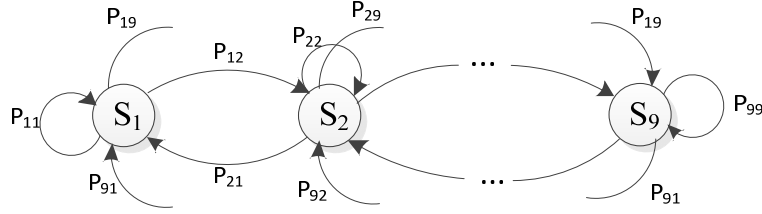
**Energy Efficiency:** This metric is the ratio of unwasted energy consumption to the total energy consumption as shown in Equation (4.17).

$$EE = \frac{E_r}{E_T} \quad (4.17)$$

### 4.6.2 Simulation setup and scenario

As we already mentioned, our protocol READ functions over the chain-clusters that are constructed by REC+, which was elaborated on in Chapter 3. As READ can be applied for each chain/cluster independently, therefore, without loss of generality the simulation results are shown for the chain inside the dashed red rectangle of Figure 4.2 which we assumed consists of 10 nodes. Moreover, the total deployment area shown in Figure 4.2 is  $400m \times 25m$ . In all simulations, the source or initiative node is the leftmost node and data rate is one sample per five seconds.

Unless otherwise stated in the performance evaluation Section, the simulation parameters are as described here. Updating packet loss rates is done locally by the upstream nodes. At any moment in time, 70% of all links have a failure rate of 0.09. We use a nine-states quasi-stationary channel model which described in Section 4.2.1 for the other 30% of the links (Figure 4.9).



**Figure 4.9. A nine-state quasi-stationary channel model**

Each state  $S_s$  of this model corresponds to one specific packet error rate  $PER_s$  as:

$PER_1=0.1$ ,  $PER_2=0.2$ ,  $PER_3=0.3$ ,  $PER_4=0.4$ ,  $PER_5=0.5$ ,  $PER_6=0.6$ ,  $PER_7=0.7$ ,  $PER_8=0.8$ ,  $PER_9=0.9$ .

In order to model slowly varying channel, the channel state transitions in our nine-state channel model need to occur infrequently. In this regard, we utilize following transition probabilities (shown in matrix  $\Gamma$ ) to simulate a slowly varying channel. The probability of staying in one state, i.e.  $P_{i,i}$ , is extracted from [18] and the remainder, i.e.  $1 - P_{i,i}$ , is evenly allocated to transitions from node  $i$  to all other nodes  $j$  ( $i \neq j$ ) so that  $\sum_{j=1}^9 P_{i,j} = 1 - P_{i,i} \quad \forall i \in [1,9], (i \neq j)$ . Basically, the states transition rate typically depends on the degree of temporal correlation. Each entry  $\Gamma_{i,j}$  corresponds to a  $P_{i,j}$ .

$$\Gamma = \begin{bmatrix} 0.995 & 0.0035 & 0.0015 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0033 & 0.992 & 0.0033 & 0.0014 & 0 & 0 & 0 & 0 & 0 \\ 0.0015 & 0.0025 & 0.992 & 0.0025 & 0.0015 & 0 & 0 & 0 & 0 \\ 0 & 0.0015 & 0.0025 & 0.992 & 0.0025 & 0.0015 & 0 & 0 & 0 \\ 0 & 0 & 0.0015 & 0.0025 & 0.992 & 0.0025 & 0.0015 & 0 & 0 \\ 0 & 0 & 0 & 0.0015 & 0.0025 & 0.992 & 0.0025 & 0.0015 & 0 \\ 0 & 0 & 0 & 0 & 0.0015 & 0.0025 & 0.992 & 0.0025 & 0.0015 \\ 0 & 0 & 0 & 0 & 0 & 0.0014 & 0.0033 & 0.992 & 0.0033 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.003 & 0.007 & 0.99 \end{bmatrix}$$

We model sending packets in each state of quasi-stationary channel model first according to a Gilbert-Elliott model and then as a series of Bernoulli trials. The Gilbert-Elliott channel model is defined by  $q$  and  $p$  which will change according to the above  $PER_s$  of  $S_s$  and so is obtained as:

$$p^1=0.1, p^2=0.13, p^3=0.15, p^4=0.16, p^5=0.17, p^6=0.18, p^7=0.19, p^8=0.2, p^9=0.21.$$

$$q^1=0.95, q^2=0.5, q^3=0.33, q^4=0.25, q^5=0.16, q^6=0.125, q^7=0.08, q^8=0.05, q^9=0.02.$$

Depending on the initialization state of the nine-state channel model, the transmission links can experience different packet error rates with different probabilities

## 4.6 Performance evaluation

---

which are listed above. For instance, according to matrix  $\Gamma$  if the channel initialization state is  $S_1$  the links may experience the packet error rate of 0.1 with the probability of 0.995 but if the initialization state is  $S_2$  the links may experience the same packet error rate 0.1 with the probability of 0.0033. In this way, we run the simulation with different setups of initialization state for the channel.

The toggle period (TP) is assumed to be 10000 ms. Simulation results are given by sending 20 source packets from one source node to the base station.

Other simulation parameters are listed in Table 4.1.

**Table 4.1. Simulation parameters**

Mac layer	IEEE 802.15.14
Transmit bit rate	250 kbps
Operation frequency	2.4 GHZ
Radio model	TI CC2420
Packet size	128 B

### 4.6.3 Performance evaluation

The TTL is chosen between  $\lfloor N \times TT + \frac{ST}{2} \rfloor$  and  $\lfloor N \times TT + \frac{ST}{2} \rfloor \times 10$  where N is number of nodes in the chain. The reason we state  $\frac{ST}{2}$  is because when an event is happening the node can be in sleep mode which means that reporting data will be delayed until the node wakes up. It implies that in average a packet may be delayed by  $\frac{ST}{2}$ . It is worth recalling that ST denotes the time duration that a node is in sleep mode and TT states the transmission time per hop. To have a good judgment about the exact relationship between both the link reliability and packet TTL with the attained hit ratio for three given approaches READ, ARQ and Simple, we presents Figure 4.10, Figure 4.11, Figure 4.12, Figure 4.13 and Figure 4.14. Regarding these figures, one can easily assess the appropriateness of each method facing different conditions. Figure 4.10 and Figure 4.11 plot the achieved hit ratio versus both packet TTL and the link reliability for READ when duty-cycle is 0.01 and 1, respectively. The link reliability indicates the reliability of the initialization state with which the simulation is started.

It can be seen from Figure 4.11 that when TTL becomes larger than 500, the hit ratio of the worst link reliability for READ fluctuates between 0.97 and 0.99. It can be seen from Figure 4.10 that when TTL is around 50000, the hit ratio of the worst link reliability is not even 0.9. The main cause of this difference is related to the duty-cycle and the sleeping time which imposes a long delay for the en-route packets. When duty-cycle is 0.01, an intermediate node is active just for 100 ms, while in the rest of 9900 ms

is in asleep mode. In case of having a packet with 2 copies, if the source node after sending the given packet and one of its copy goes to asleep mode, the second copy should wait till the source node wakes up again. Since the sleeping time is 9900 (which is too long), therefore the packet is incurred a long delay. In this case, to have the given packet still fresh and on time when it reaches the destination, the TTL should be long enough to give the source node this opportunity to send all of the packet copies.

The hit ratio of ARQ approach for two duty-cycle 0.01 and 1 is shown in Figure 4.12 and Figure 4.13, respectively. Moreover, the hit ratio of simple transmission is presented in Figure 4.14.

In order to better compare the performance of these three approaches, we put the hit ratio of these approaches for three different loss level and two duty-cycles in Figure 4.15. According to this figure, in case of  $PLR=0.5$ , when either duty-cycle is 0.01 and TTL is shorter than 35000 or duty-cycle is 1 and TTL is shorter than 500, READ outperforms other approaches. However, when TTL become larger than the aforementioned TTLs for the given duty-cycles, READ exhibits more or less the same hit ratio as ARQ. This could be justified as in case of having packet whose TTL is higher than the aforementioned TTL 35000 and 500, the intermediate nodes have enough time to send all packet copies in READ or packet(s) and acknowledgement packet(s) in ARQ, within the packet TTL. In case of  $PLR=0.9$  and  $DC=0.01$ , one can see that READ considerably outperform others. When  $PLR=0.1$ , even though at the beginning READ outperforms other approaches but when TTL is becoming larger than 160 (when  $DC=1$ ) and larger than 5000 (when  $DC=0.01$ ), the hit ratio of READ and ARQ become almost the same. The main reason is that since packet loss rate is small, so not many copies of a packet are required to ensure the reliability. The smaller number of packet copies, the shorter amount of time to make it reach to the destination.

Generally speaking, lower duty-cycle values imply longer waiting time for the packets (or copies) that are ready to be sent but should wait till upstream nodes wake up again. Therefore, for a time-constrained data packet, lower duty-cycle values result in sending smaller number of copies or less retransmissions. As ARQ wastes almost half of the active time waiting for the acknowledgement, READ shows higher hit ratio in presence of low duty-cycles as it uses all awake time to send the packet copies.

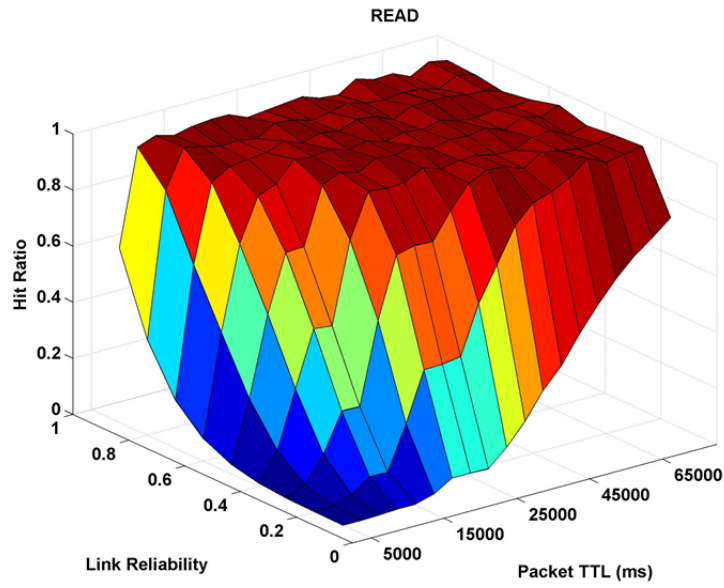


Figure 4.10. Hit ratio of READ with duty-cycle=0.01

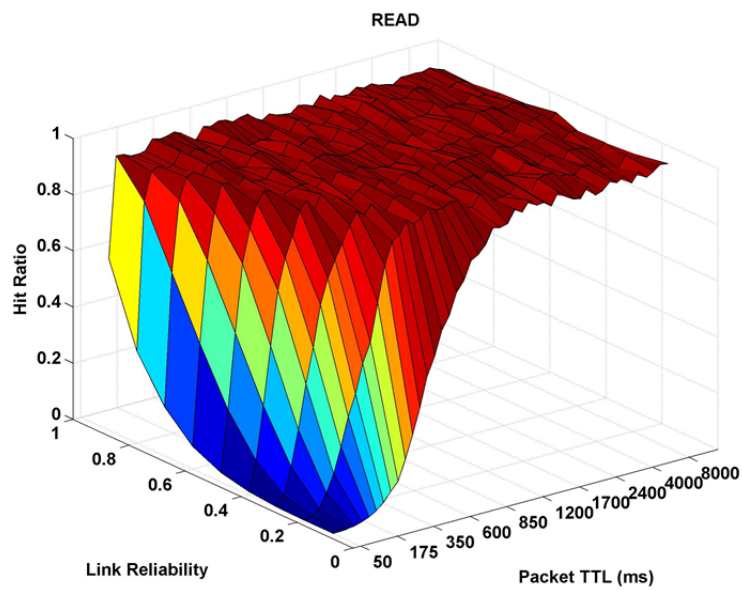


Figure 4.11. Hit ratio of READ with duty-cycle=1

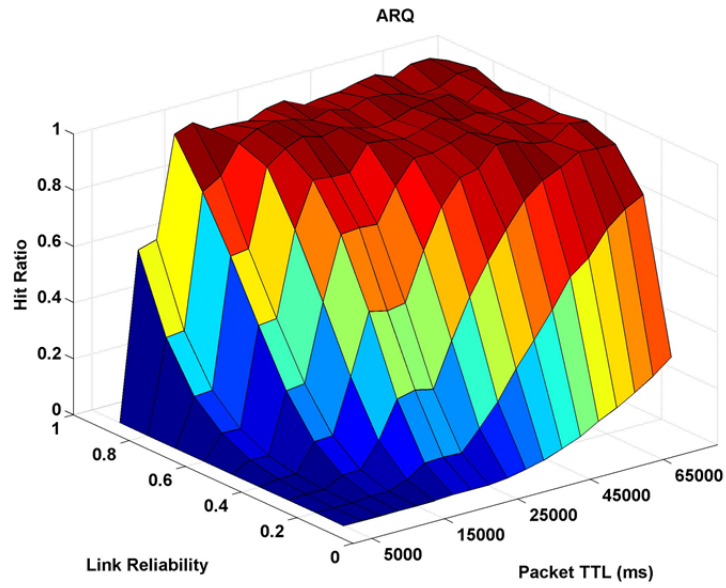


Figure 4.12. Hit ratio of ARQ with duty-cycle=0.01

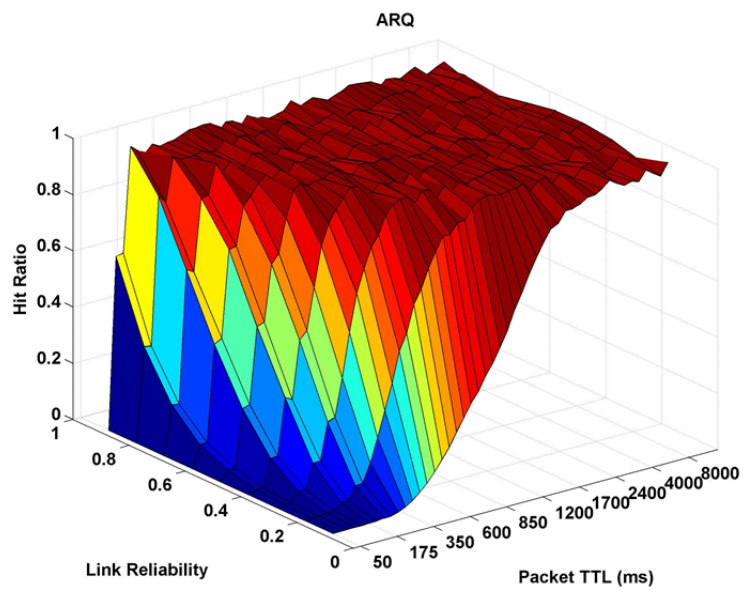
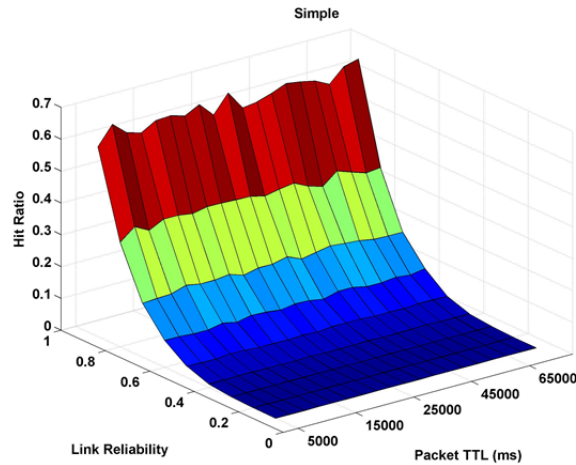


Figure 4.13. Hit ratio of ARQ with duty-cycle=1



**Figure 4.14. Hit ratio of simple**

Both the average total energy and the average unwasted energy for all the source packets, are demonstrated in Figure 4.16. For the sake of clarity, we provide a comparison among the total energy of all approaches in Figure 4.17. Moreover, the energy efficiency of these approaches can be judged in Figure 4.18. With the reference to Figure 4.16 and Figure 4.18, one can see that READ outperforms ARQ in terms of both total and unwasted energy consumption. Even though Simple scheme consumes the least amount of energy but taking a look at the Figure 4.18 reveals that it is not energy efficient specially when the packet loss rate is high. One can see that, total and unwasted energy of Simple approach when links are more reliable is higher than when link reliability is low. This is due to the fact that when link reliability is low, most of the packets are lost in the first links as they are not protected by any error control mechanism, and thus not much energy will be dissipated. In READ and ARQ, however, packets are protected using error control mechanism and thus when link reliability becomes higher, less energy will be dissipated as less redundancy (i.e. packet copies in READ and acknowledgement and retransmission in ARQ) is required.

According to Figure 4.18 READ is the most promising approach in terms of energy efficiency particularly in case of high packet loss rate and short TTL. Energy efficiency of ARQ even in the best condition cannot exceed 0.5 due to acknowledgement overhead. Compared with ARQ, READ is more energy efficient specially when encountering packets with short TTLs. It can be argued that almost all of the very delay-constraint packets received by the base station using ARQ scheme are expired because of the extra delay introduced by the acknowledgement messages.



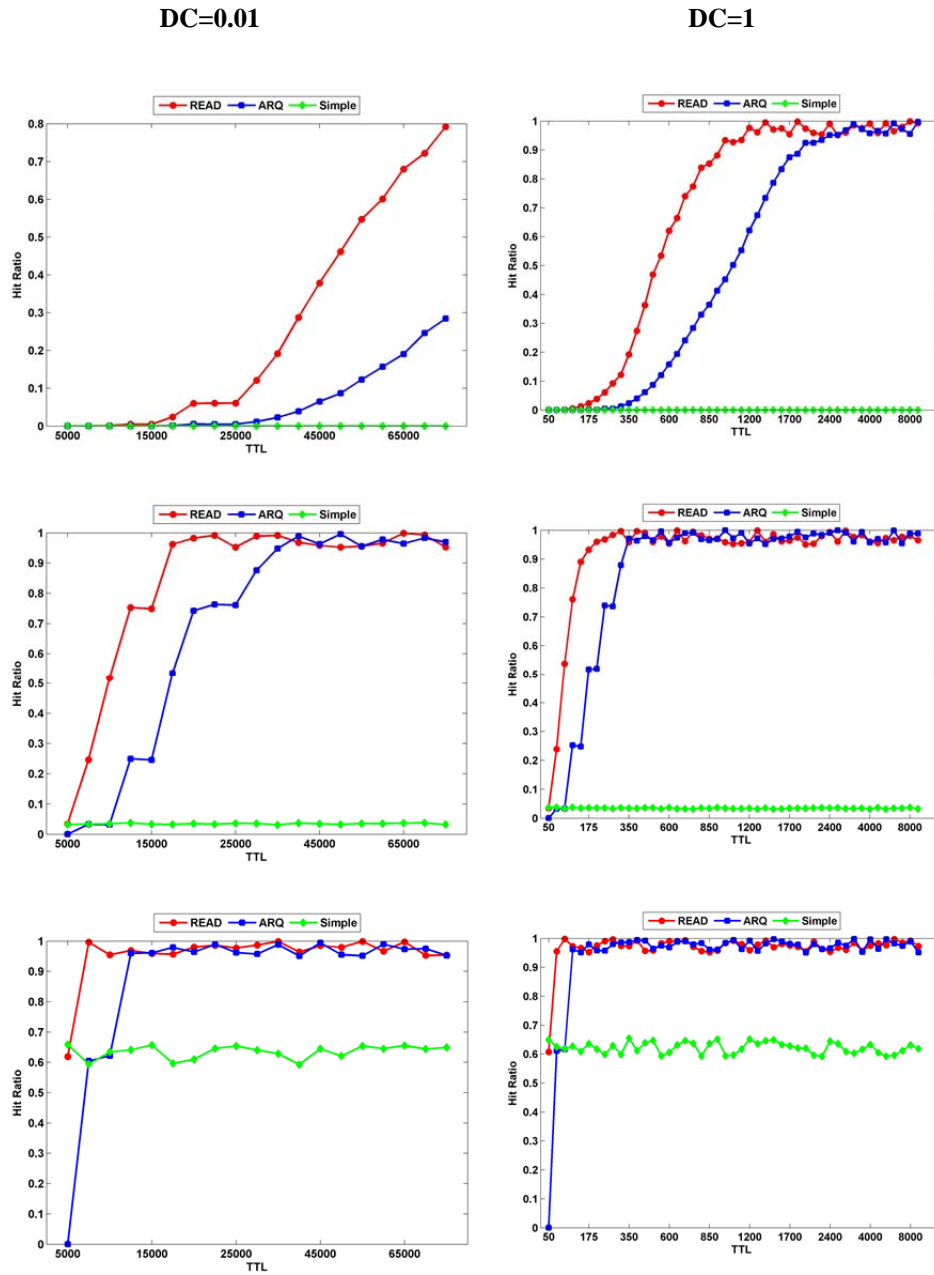


Figure 4.15. Hit ratio when DC=0.01 (left-side graphs) and DC=1 (right-side graphs) for PLR=0.9 (top), PLR=0.5 (middle), PLR=0.1 (bottom)

## 4.6 Performance evaluation

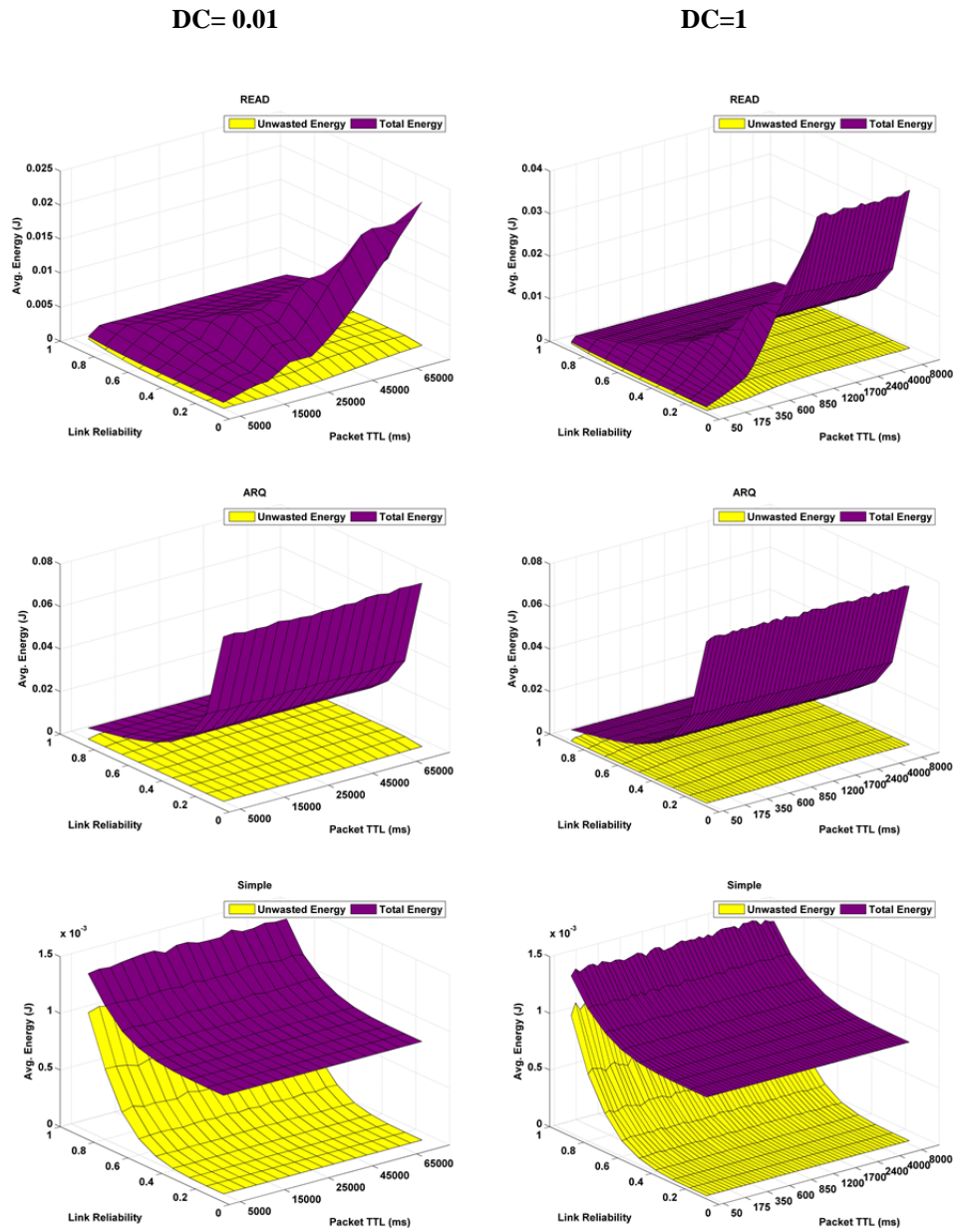


Figure 4.16. Total and unwasted energy when DC=0.01 (left-side graphs) and DC=1 (right-side graphs)

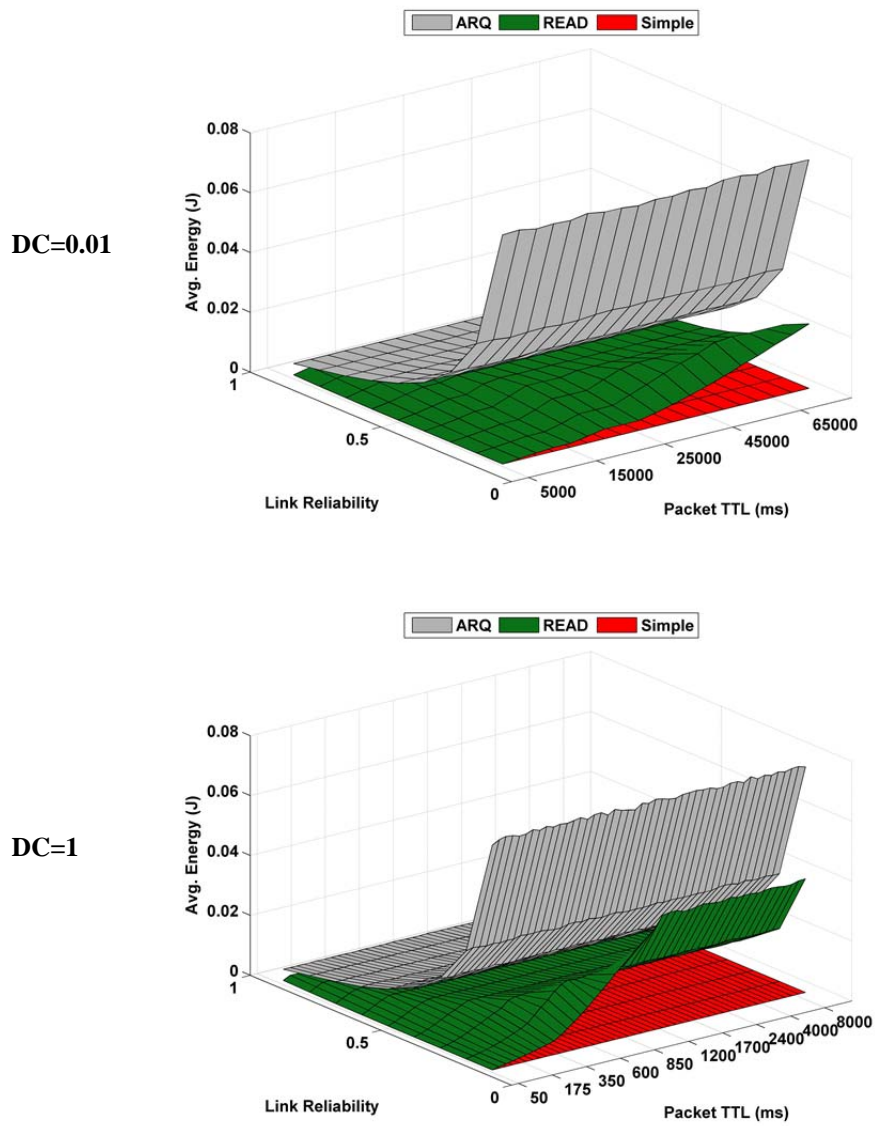


Figure 4.17. Total amount of energy when DC=0.01 (top) and DC=1 (bottom)

## 4.6 Performance evaluation

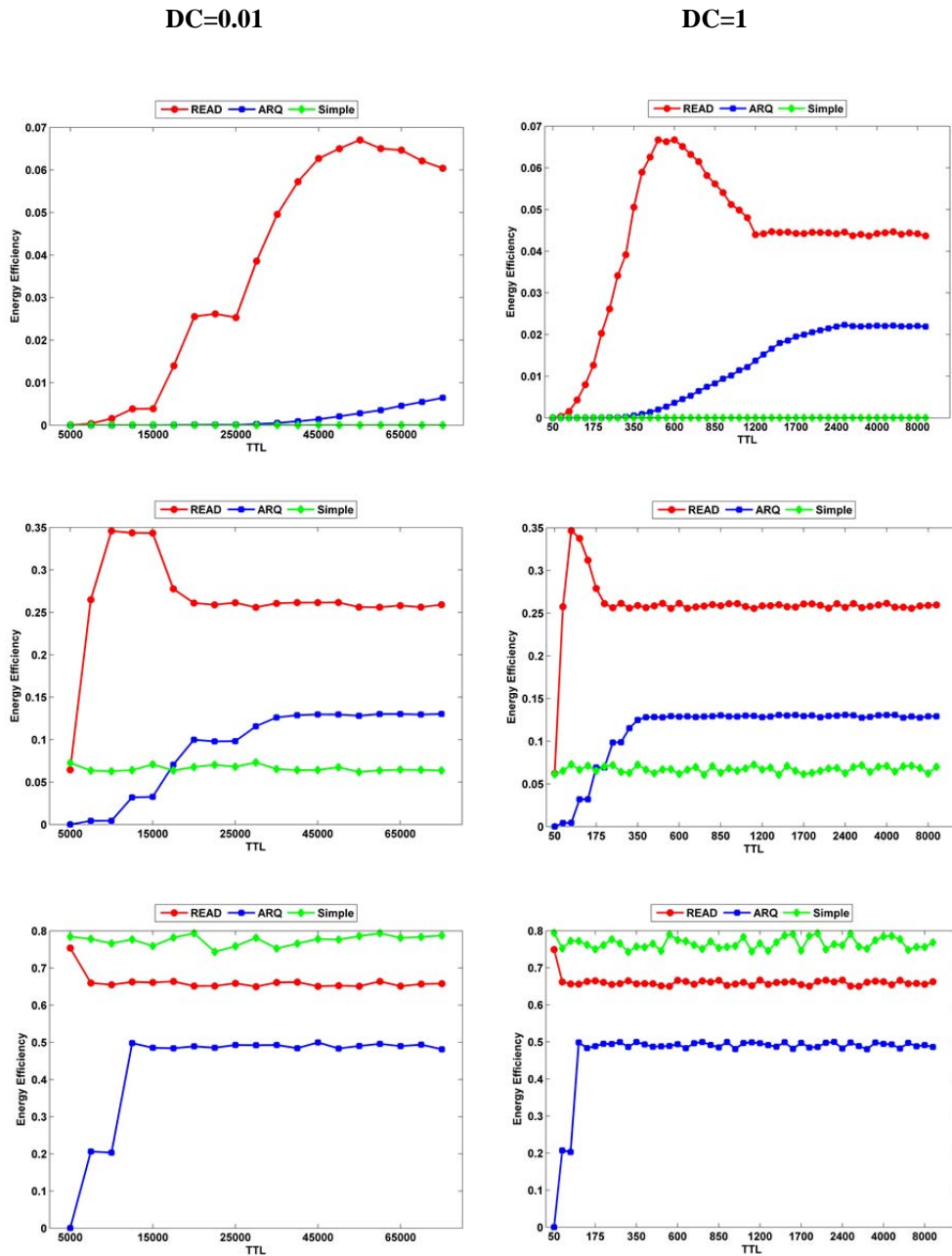


Figure 4.18. Energy efficiency when DC=0.01 (left-side graphs) and DC=1 (right-side graphs) for PLR=0.9 (top), PLR=0.5 (middle), PLR=0.1 (bottom)

In the last experiment we aim to show the effect of different policies regarding updating fractional portion, over the hit ratio. To have a better judgment about updating policies, we assume that the quality of all links is fixed except one whose quality gradually varies. At the beginning the channel condition is in state  $S_3$ , i.e. loss rate is 0.3. As explained in Section 4.5.3., the packet loss rate can be updated either locally by the neighbor nodes or centrally (globally) by the base station.

We define two scenarios as follows:

1. Link quality varies infrequently, thus the probability of staying in the current state (i.e.  $\Gamma_{i,i}$ ) is 0.995 and the probability of transition to other states is 0.005 in total.
2. Link quality varies more frequently, thus the probability of staying in the current state (i.e.  $\Gamma_{i,i}$ ) is 0.85 and the probability of transition to other states is 0.15 in total.

Without loss of generality, the duty-cycle is considered to be 1 and the packet TTL is 100. Figure 4.19 and Figure 4.20 compares the hit ratio of READ based on two different updating mechanisms for the infrequent and frequent loss rate variation, respectively.

According to Figure 4.19, since the channel was assumed to vary quite slowly with respect to the data transmission rate, globally updating packet loss rate will be more efficient in terms of hit ratio. However, one can see that if the loss rate variation is high, locally updating packet loss rate as shown in Figure 4.20 will be more promising. For example, with reference to Figure 4.19, in time interval (80,150), globally updating shows an increase of 0.05 in hit ratio comparing with locally updating. From these two graphs the employed error model can also be justified as any variation on links' quality remains for a while and thus hit ratio varies slowly. According to these graphs, there is up to 0.05 increment in hit ratio in case of using centrally updating instead of locally updating, when loss variation is infrequent. There is also up to 0.15 increment in hit ratio when using locally updating instead of centrally updating, when loss variation is frequent.

Looking the scale of these graphs, one can see that in case of infrequent loss variation the sharp variation in hit ratios, which is two times for 250 samples using local updating, is much less than that of frequent loss variation, which is four times for only 60 samples using local updating.

All in all, although central updating nodes' portion can lead to an optimal solution, it is not fast enough to be utilized for a network whose links' quality suffer from high variation. On the other hand, local updating would be more effective than central

## 4.7 Chapter summary

---

updating in a large and dynamic network, in which cumulative error and thereby the sum of the nodes' fractional portion from TTL is almost fixed.

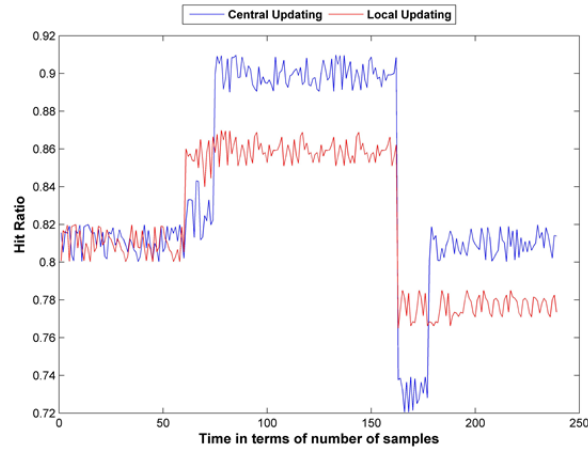


Figure 4.19. Hit ratio of READ for infrequent loss variation

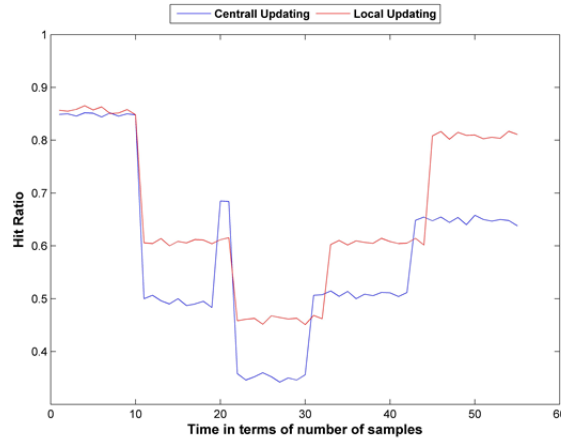


Figure 4.20. Hit ratio of READ for frequent loss variation

## 4.7. Chapter summary

Time-critical applications of wireless sensor networks may require delivery of various types of sensory data with different levels of real-time requirements depending on the dynamics of the sensed environment.

Meeting the TTL constraint of the sensory data which should reliably be transmitted toward the base station in such a low duty-cycle network that suffers from short-term burst errors is the main contribution of this chapter. In this respect, we propose READ which fairly allocates the available TTL based on the links qualities among the sensor nodes who utilize the allocated times to enhance the transmission reliability. Furthermore, READ is able to work under different duty-cycles based on which will tune the fractional portion of the sensor nodes from the TTL.

From another side, READ is an energy-efficient adaptive error-control scheme which improves the reliability to such extent that does not influence the TTL constraint of the packet. In the interest of conserving both energy and bandwidth READ drops the packets which are expected to be expired before reaching the base station. Different from most of the proposed reliable and real-time approaches that are proposed to work for the topologies other than linear and so cannot efficiently work for the linear topology, READ is customized for this poor-explored topology.

In the simulation, we illustrate the relation between both TTL of the packet and link quality and the attained hit ratio for different approaches. The gain of our solution with respect to ARQ is up to 50% improving the hit ratio and up to 30% increasing the energy efficiency when the packet loss rate is high and TTL is short. The simulation results also reveal that if the links suffer from infrequent quality variations, centrally updating the fractional portion of the sensor nodes can enhance the hit ratio of READ in comparison with locally updating scheme.

From our results, we can conclude that READ is a promising approach specially when the packet loss rate is relatively high or when the TTL is not large enough to transmit the optimal number of packet copies.

### 4.8. Bibliography

- [1]. MacKay, D.J.C., *Information theory, inference, and learning algorithms* 2003: Cambridge: Cambridge University Press.
- [2]. Frossard, P. and O. Verscheure, *Joint source/FEC rate selection for quality-optimal MPEG-2 video delivery*. IEEE Transactions on Image Processing, 2001. 10(12): p. 1815-1825.
- [3]. Sharma, G., R.R. Mazumdar, and N.B. Shroff. *On the complexity of scheduling in wireless networks*. in *12th annual international conference on Mobile computing and networking*. 2006.

## 4.8 Bibliography

---

- [4]. Ebert, J.-P. and A. Willig, *A Gilbert-Elliot bit error model and the efficient use in packet level simulation*. Report, TKN-99-002, Technical University of Berlin, 1999.
- [5]. Cover, T.M., *JA Thomas Elements of information theory*, 1991, John Wiley.
- [6]. He, T., et al. *SPEED: A stateless protocol for real-time communication in sensor networks*. in *23rd International Conference on Distributed Computing Systems*. 2003.
- [7]. Felemban, E., et al. *Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks*. in *24th Annual Joint Conference of the IEEE Computer and Communications Societies*. 2005.
- [8]. Kim, K.-I., et al. *Reliable and real-time data dissemination in wireless sensor networks*. in *Military Communications Conference*. 2008.
- [9]. Soyuturk, M. and D.T. Altılar. *Reliable real-time data acquisition for rapidly deployable mission-critical Wireless Sensor Networks*. in *INFOCOM Workshops*. 2008.
- [10]. Lu, G., B. Krishnamachari, and C.S. Raghavendra. *An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks*. in *18th International Symposium on Parallel and Distributed Processing* 2004.
- [11]. Yanagihara, K., et al. *A Sensor Network Protocol for Automatic Meter Reading in an Apartment Building*. in *IFIP International Federation for Information Processing*. 2010.
- [12]. Gu, Y. and T. He. *Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links*. in *5th international conference on Embedded networked sensor systems*. 2007.
- [13]. Fei, Y., *Reliable and time-constrained communication in wireless sensor networks*. PhD Thesis, Zhejiang University, China, 2012.
- [14]. Chipara, O., et al. *Real-time power-aware routing in sensor networks*. in *14th IEEE International Workshop on Quality of Service*. 2006.
- [15]. Instruments, T., *CC2420: 2.4 GHz IEEE 802.15. 4/ZigBee-ready RF Transceiver*. Available at <http://www.ti.com/lit/gpn/cc2420>, 2006.
- [16]. Wang, X., et al., *Flow-based real-time communication in multi-channel wireless sensor networks*, in *Wireless Sensor Networks2009*, Springer. p. 33-52.
- [17]. Cao, Q., et al. *Towards optimal sleep scheduling in sensor networks for rare-event detection*. in *4th international symposium on Information processing in sensor networks*. 2005.
- [18]. Rice, M. and S.B. Wicker, *Adaptive error control for slowly varying channels*. *IEEE Transactions on Communications*, 1994. 42(234): p. 917-926.





# Information-Link-aware Data Dissemination<sup>1</sup>

---

As an application-driven network, wireless sensor networks are generally used to energy-efficiently and reliably sense and disseminate information about the phenomena to maintain detection and response capabilities. Having dynamic of wireless sensor networks in mind, it is almost certain that the channel quality will exhibit high variation over time and even differ across different links. Therefore, using a static error control approach which tolerates either the expected worst-case or the expected best-case error rate, in order to ensure reliability can be either quite energy-inefficient or lead to a great deal of information loss, respectively. Dynamic or adaptive error control schemes which are allocating the correctional power in an on-demand manner based on both the packet information-value and channel state are viable alternatives to static error control schemes, in which the link conditions or packets information-values are not taken into account. In this way and for the sake of efficiency, both the importance of information that a packet carries and the state in which the channel is, can be put into perspective with the amount of effort that is required to reliably transmit the information. This information-aware capability allows a system to deliver critical information with high reliability but potentially at a higher resource cost. From another perspective, it saves energy by delivering less important information at a lower reliability. Since the application of information-aware adaptive error control mechanisms for wireless sensor networks operating under timely and spatially variable channel conditions has generally been less-studied, in this chapter we give emphasize to this type of application. To

---

<sup>1</sup> This chapter is based on the following publications:

- (i) *Protecting informative messages over burst error channels in chain-based wireless sensor networks*. The Forth International Conference on Sensor Networks (SensorNets 2015).
- (ii) *An information-link-aware data dissemination scheme for chain-based wireless sensor networks*. The Sensors Journal (under review).

this end, in this chapter we aim to (i) extract, quantify and integrate the factors that may influence the information-value of a packet and (ii) cope with this crucial design problem of choosing an appropriate error control code by adaptively selecting the codes for each *individual links*, which may experience long-term fading and for each *individual packet* at run-time instead of applying network-wide settings prior to network deployment. In this way, we propose RAFEC protocol, which is a Run-time Adaptive FEC-based data dissemination protocol. In RAFEC each node decides which error control code to use abiding to the computational constraints of embedded sensors, the information-value of the packet, and the statistical properties of the observed errors for the upward link. Simulation results validate the superiority of RAFEC compared with a number of Reed-Solomon-based error control approaches.

### 5.1. Introduction

Adhering to the packet-level or data constraints while designing a data disseminating protocol for wireless sensor networks may improve the system performance.

In the same line of the previous chapter which addressed TTL as one of the packet-level constraints, in this chapter we emphasize another constraint, namely information-value, considering which we can enhance the reliability performance of a dissemination protocol considerably. By information-value we mean the amount of information a packet may have for the base station.

Most telecommunication systems use a fixed channel code to tolerate the expected worst-case error rate, which implies that they fail to operate at all if the error rate is worsened. As we already mentioned in the previous chapters, the wireless channel is time varying and can exhibit high error rate over time. In order to improve the reliability of the data which is transmitted in wireless sensor networks, the error control approaches such as ARQ and FEC can be applied. Putting their advantages aside, existing error control techniques contribute to increase of the energy consumption due to the redundant data to be transmitted. Since energy is a scarce resource in wireless sensor networks, the type and the strength of the error control in use should be dependent on the type of the application. Generally speaking, event detection applications of wireless sensor networks need to execute more efficient and powerful error control techniques compared with periodic monitoring applications. However, the distinction between different packet type as being transmitted in these two classes of applications (periodic data and alarm) is neither general enough nor captures some important cases (e.g. the effect of channel condition or aggregation function) in wireless sensor networks. Therefore, even within a specific class of application, it would not be a proper to use a single error control code for all packets regardless of their different channel conditions

## 5.1. Introduction

---

or importance of information they carry. It is quite likely that even two packets both of which carry periodic monitoring data, not have the same amount of information and importance. For example, in a chain-based wireless sensor network data aggregation mechanisms are often used along the path with the aim of reducing the number of transmitted packets. Therefore, some packets may contain the aggregated readings of many nodes. These packets thus should be sent more reliably as they carry more informational value.

It would be therefore a good idea to classify packets on the basis of their information-value based on which a proper error control scheme can be applied. By doing so, more important packets that have relatively high information-value are transmitted more reliably than packets carrying less important information. This is to well-balance the energy expenditure (caused by data and parity packets) and reliability.

Having dynamics of wireless sensor network into mind, adopting an efficient and accurate network-wide error control approach prior to network deployment is almost impossible. A very weak (in terms of correction capability) error control approach may not be able to correct many errors while a too strong code results in waste of time and energy resource for encoding, decoding and transmitting parity packets. Dynamic error control schemes which are allocating the correctional power in an on-demand manner based on both the information-value and channel state are viable alternatives to static error control schemes, where the link conditions or packets' information-values are not taken into account. In this way and for the sake of efficiency, the amount of information and importance a packet carries and the state in which the channel is in can be put into perspective with the amount of effort (in terms of energy expenditure) that is required to reliably transmit the given packet.

Furthermore, since the wireless channel is inherently lossy and often manifests itself with bursts errors correlated in time, a reliable data dissemination should be capable of counteracting a large number of consecutive or burst errors.

Since the application of run-time information-aware adaptive error control mechanisms for wireless sensor networks operating under timely and spatially variable channel conditions has generally been less-studied, in this chapter we give emphasize to this type of application.

In this chapter, first the factors that may influence the information-value of a packet will be investigated. Then we incorporate all these obtained factors in order to estimate the information-value of the packets. Finally, we exploit the amount of information and importance that a packet carries as a means to properly adjust the parameters of the adaptive error control code in use. In this regards, we propose RAFEC, which is a Run-time Adaptive FEC-based data dissemination protocol to enhance reliability, based on the amount of information the packets carry over a long-term error-bursty channel in a chain-based wireless

sensor network. This adaptation gives the possibility to vary the code strength and complexity on-demand and on the fly.

### 5.1.1 The need for packet-level FEC

As we mentioned in Chapter 4, FEC applied at the bit-level and byte-level is appropriate for short-term errors and additive white Gaussian noise when rapid fluctuation is experienced over a short period of time. This is because in this situation, only some bits or bytes of a packet are influenced. FEC applied at bit- or byte-level is less efficient in recovery from burst bit errors caused by long-term fading and expanded over several packets (due to high concentrations of errors). In this regards, it is unable to recover a completely lost or delayed packet. Therefore, in these cases, as discussed in Chapter 4, either ARQ or a packet-level FEC should be employed. ARQ-based approaches are effective only for a shorter time-scale or short-term burst errors. In this respect, even though ARQ could tolerate long-term fading to some extent, but more persistent fluctuations make this approach as inefficient as bit- and byte-level FEC. To overcome the unreliability caused by more persistent fluctuations or long-term burst errors, application-level or packet-level FEC may be used.

#### 5.1.1.1 Characteristic of forward error correction codes

There are four characteristics of the forward error correction code that affect the functionality of the codes:

- **Linear vs. Non-linear codes:** In linear code, any linear combination of two code-words is also a code-word. Linear codes are important since they have very concise description and easy encoding/decoding process [1]. They are also called “group codes”. Linear codes typically allow for more efficient encoding and decoding algorithms than other codes [1]. However, the above statement is not always true for the Non-linear codes.
- **Systematic vs. Non-systematic codes:** In a systematic code, the data-word (or actual message) appears unaltered in the encoded code-word and just redundant or parity symbols are simply appended to the data-word. The advantage of systematic codes is that the receiver does not need to decode the original data-words if they received correctly. Therefore, those correctly received packets can be used by the receiver directly. This property is more appealing for the applications, which prefer (if possible) not to undergo the incurred delay of the decoding process. However, in non-systematic codes the data-word does not appear in its original form in the encoded code-word. Instead, there exists a mapping between the data-word to the code-word and vice versa.

## 5.2. Assumptions and models used

---

- **Block vs. Convolutional codes:** A block code adds the parity symbols to the input or original symbols (data-word) and transmits the resultant longer symbols, namely code-word, for error correction. The block codes are typically implemented as  $(n,k)$  codes, where  $n$  states the code-word and  $k$  indicates the data-word. Therefore, in block codes one block of messages is transformed into one block of code and no memory is required. The Hamming codes [2], BCH codes [3] and Reed-Solomon codes are the subset of block codes. They also belong to the class of linear codes and hence are called linear block codes. In case of convolutional code, a sequence of message is converted into a sequence of code. Hence encoder requires memory as the encoder outputs at any given time unit depend not only on the present inputs but also on a number of previous inputs. In this way, convolutional encoders do not transform information words into code-words block by block, but transform the whole sequence of information symbols into a sequence of encoded symbols by convolving the information symbols with a set of generator coefficients. Therefore, the difference between block codes and convolutional codes is the encoding principle.
- **Binary vs. Non-binary codes:** In the binary codes, error detection and correction is performed on binary information i.e. bit, and so the error magnitude is one. In binary code just finding the location of error is enough because the correction only means to flip the erroneous bit. On the other hand, in the non-binary codes, error detection and correction is performed on symbols which may be bit, byte or even packet. In this case, both the location and magnitude is required to correct the erroneous symbol. A non-binary code such as Reed-Solomon can repair the symbols affected by long-term bursts [4, 5].

The rest of this chapter is organized as follows. First we explain the assumption and model we used in Section 5.2, which is followed by the related work in Section 5.3. Then in Section 5.4, we describe the problem statement and our contribution. We elaborate on our proposed RAFEC protocol in Section 5.5. Then in section 5.6 we present the simulation setup and performance evaluation results. Finally in Section 5.7 we present the chapter summary.

## 5.2. Assumptions and models used

The assumptions and network model that our proposed approach is built upon are to a large extent similar to the model presented in Section 4.2 of Chapter 4. However, here we add some extra considerations mostly related to the FEC and burst errors as follows:

- In contrary to READ in which the error occurs in short-term-burst, errors are assumed to persist for longer time interval or appears in long-term burst.
- The channel is considered to vary slowly with respect to the data transmission rate, and thereby the channels state transitions occur infrequently.
- A systematic code (vs. non-systematic) is preferred, as it less suffers from delays imposed by the decoding mechanisms.
- Uncertainty parameters of the nodes and links are fixed over transmitting a single code-word.
- The transmission errors are assumed to be local and spatially and temporally variable, which in turn should be tackled on a per-link and not network-wide basis.

### 5.2.1 Channel model

We use a Quasi-Stationary Gilbert-Elliot (QSGE) model, as shown in Figure 4.3. of Chapter 4, in order to model channel states. As discussed in in Chapter 4, each state  $S_v$ , which corresponds to a specific packet error rate  $PER_v$ , follows a Gilbert-Elliot model with some probabilities ( $p$  and  $q$ ) associated to it. The B (Bad) and G (Good) states are also a series of Bernoulli trials.

### 5.2.2 Reed-Solomon codes

In this section we elaborate on Reed-Solomon code a family of which is utilized by RAFEC in order to improve reliability.

Reed Solomon code [6] is a linear systematic non-binary block code. In the encoding, redundant symbols are generated using a generator polynomial and appended to the message symbols. In the decoding, error location and magnitude are calculated using the same generator polynomial. Then the correction is applied on the received code. Reed Solomon code has less coding gain compared to LDPC [7] and turbo codes [8]. However, it has higher coding rate and low complexity. Hence it is suitable for many applications including wireless sensor networks.

Reed-Solomon codes are the non-binary form of the block codes and called maximum distance separable (MDS) codes. This implies that no other FEC coding mechanism can recover lost data symbols from fewer received code symbols. Reed-Solomon code is considered to be an attractive choice for correcting burst errors of wireless sensor networks

### 5.3. Related work

---

[4, 5]. The number and type of errors that can be corrected depends on the parameters and characteristics of the Reed-Solomon code.

A Reed-Solomon code is typically defined as  $RS(n,k)$ , where  $n$  represents the code-word length in terms of symbols and  $k$  denotes the number of original data symbols or data-words.  $n-k$  represents the redundant or parity symbols. If Reed-Solomon code is utilized at link-layer, the symbol is a bit while at application-layer the symbol is a packet. A vector notation is used for data-words and code-words as  $d = (d_1 d_2 \dots d_k)$  and  $c = (c_1 c_2 \dots c_n)$ , respectively, where  $d_i$  represents the data symbol  $i$  and  $c_j$  represents the code symbol  $j$ .

A Reed-Solomon code decoder could correct up to  $t$  symbols:

$$t = \left\lfloor \frac{n - k}{2} \right\rfloor \quad (5.1)$$

The Reed-Solomon code can correct only half of the parity symbols as for every error one parity symbol is used to locate the error and the other is used to correct it. One should note that if erasures are present then the location of error is already known and so only one parity symbol is used to correct the error.

Larger  $t$  implies that more errors can be corrected and consequently more computational power is spent due to encoding and decoding process. Reed Solomon code follows the Galois Field (GF) arithmetic [9] properties for encoding and decoding techniques.

This type of codes may also be shortened by conceptually making some of data symbols zero at the encoder, not transmitting them, and then reinserting them at the decoder side [10]. For example, the RS (15, 11) can be shortened to RS (7, 3). In this case, the encoder takes a block of 3 data packets, adds conceptually 8 null packets to it, creates a (15,11) code-word, and transmits only the 3 data packets and 4 parity packets.

The general procedure of a packet-level Reed-Solomon code is demonstrated in Figure 5.1.

### 5.3. Related work

Although numerous research have been published related to error control in wireless networks, especially in cellular networks, most of these are not directly applicable to wireless sensor networks. The limited energy, low complexity of the sensor node hardware, and harsh/dynamic environment of the deployment area necessitates an energy-efficient and more dynamic or adaptive error control strategy to be used.



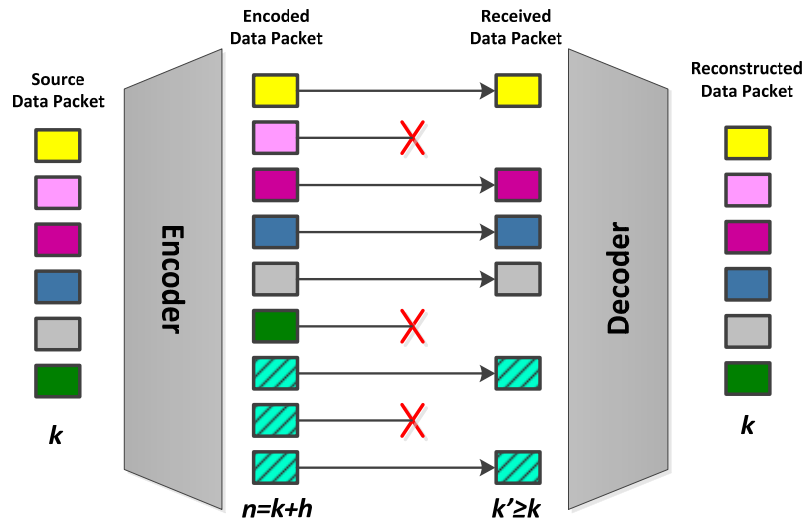


Figure 5.1. Packet-level Reed-Solomon code RS( $n,k$ )

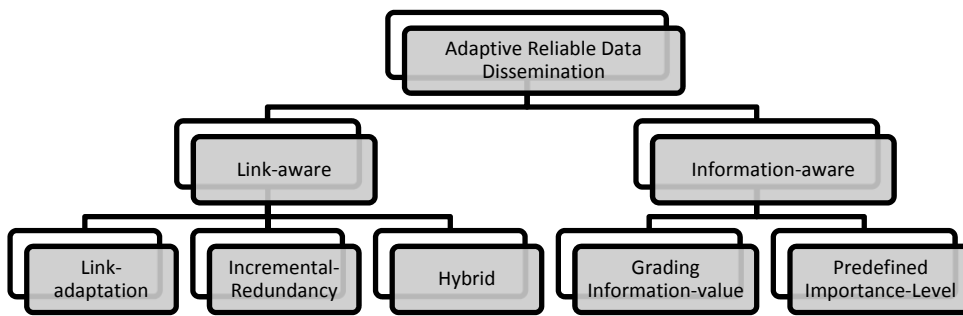
A classification of adaptive reliable data dissemination protocols is shown in Figure 5.2. According to this figure, one can see that the adaptive reliable data dissemination protocols fall in two main categories: (i) Link-aware and (ii) Information-aware.

Link-aware protocols can in turn be divided in three subcategories:

- (i) Link-Adaptation-based, in which the coding scheme is adjusted based on the estimated quality of the links.
- (ii) Incremental-redundancy-based, in which the coding scheme (or code rate) is adjusted by incrementally transmitting redundancy until the decoding is successful.
- (iii) A hybrid of Link-adaptation and Incremental-redundancy which aims to combine the advantage of both to improve reliability.

Information-aware protocols could be divided in two categories:

- (i) Predefined Importance-Level, in which the protocol does not involve in grading the information-value of the packets and they assume the importance level of the sensory data is somehow set by the source nodes. These protocols are usually asked to provide a predefined desired reliability.
- (ii) Grading Information-value, in which the protocol assesses the information-value of the packets at each node individually.



**Figure 5.2. A Classification of adaptive reliable data dissemination protocols**

#### 5.3.1 Link-aware adaptive reliable data dissemination

Some literatures propose error control schemes whose correction capability vary according to the links quality.

Hybrid automatic repeat-request [11] uses a fixed FEC method as long as the FEC can handle the error rate. When the error rate becomes too high it switches to ARQ.

In [12] authors switch among several BCH codes previously implemented, based on the fraction successfully received packets. This approach requires additional feedback channel.

A hybrid ARQ and FEC adaptive scheme is proposed in [13]. This approach evaluates the links quality based on the collected Signal to Noise Ratio (SNR). In environments, in which Received Signal Strength Indication (RSSI) varies frequently and intensively, short-term observation of SNR variations can easily fail in providing the information about the overall trend of change of propagation conditions.

An adaptive forward error correction is proposed in [14] which evaluates the links quality based on receipt of acknowledgements. Utilizing a number of codes with different error correction capabilities, it proposes an adaptive approach to switch among different codes. It first calculates the packet error rate and compares it with a predefined threshold to determine whether to switch to another error control code.

AFECCC [15] is an adaptive FEC-based error control, which adjusts the number of parity bits per packet on the basis of the underlying wireless channel bit error rate. It increases the packet protection by employing a stronger FEC when encountering transmission errors. The strength of error correcting FEC is lowered down after successful transmission in a simple multiplicative-increase additive-decrease (MIAD) manner. Therefore, AFECCC

incrementally derives the proper FEC code using this concept that if number of errors is beyond the correction capability of the code then the node cannot accurately estimate the exact number of corrupted bits. In this way, AFECCC utilizes the most powerful code upon detecting a packet loss and then gradually decreases the code strength if needed. The performance of AFECCC depends on the multiplicative and decay factors, which are rather difficult to determine experimentally. Similar to [14], AFECCC evaluates link quality according to the receipt of acknowledgements.

Three adaptive forward error correction mechanisms, i.e., SA-FEC, SHA-FEC, and SSRA-FEC are proposed in [16]. All these protocols react to various link qualities by deciding the correctional power of the code in an on-demand manner. The only distinguishable fact about these three approaches is related to the strategies they adopt to switch among different codes.

Two adaptive error control approaches to lower down the power consumption are proposed in [17] and [18]. These approaches rely on the multi-path transmission, which are highly dependent on the network topology. Multi-path transmissions are quite weak in some topologies like as chain.

In addition to varying the code strength in proportion to the link quality, some approaches including [19, 20, 21], actively adjust transmission related parameters such as maximum transmission unit size, transmission rate, and modulation scheme based on the average loss rate or signal to noise ratio. For instance, the approach presented in [19] proposes to choose a slow but robust modulation when signal to noise ratio is low and a fast but weak modulation when signal to noise ratio is high.

### 5.3.2 Information-aware adaptive reliable data dissemination

The basic idea of this category of protocols is that not all ‘to be transferred’ packets require 100% reliable delivery. Instead, the reliability is application-specific and reliability requirements depend on the different importance levels of packets or environmental conditions. The advantage offered by this category of protocols is that limited resources, such as bandwidth and energy, will only be spent on important information with high-reliability requirements.

A number of reliable data dissemination approaches including [22, 23, 24, 25, 26, 27, 28] rely on information-awareness and consider diverse priorities among different packets. The novelty of these approaches is that they consider the need for information-awareness and adaptability to the link quality along with allocation of network resources based on the

### 5.3. Related work

---

criticality of data. Each priority level is usually mapped to a desired reliability for data delivery. The design purpose of such protocols is to preserve network resources.

ReInform [23] sends multiple copies of each packet along with multiple paths from source towards base station in such a way that data is delivered with a given reliability. It uses the concept of dynamic packet state to control the number of paths needed to reach the required reliability. ReInform basically combines the importance level of the information along with multiple path routing. In this way, when a source node initiates a packet, it should set the importance level of the packet and then finds the corresponding reliability level. Thereafter, according to this importance level, source node creates multi paths toward the base station and through each path one packet copy is sent.

Deb et al. [27] propose two mechanisms, i.e. HHR and HHRA, to provide the reliable data delivery while using hop-by-hop broadcast. Both of these mechanisms rely on sending multiple copies of the same packet to achieve the required reliability for a given packet. The required number of copies is computed based on a locally estimated packet error rate.

The approaches presented in [23, 27] do not estimate the information-value of the packets and each source node should somehow identify the importance level of its packets by itself. There also exist adaptive error control schemes, including [24, 25, 26, 28], that change the strength of the error control technique according to parameters having direct impact on packet's information-value. These parameters are evaluated on each node. Instead of using sensor data itself to determine information-value of a packet, these approaches typically use number of hop each packet has traversed, number of individual sensor readings contributing to an aggregated messages, or spatial-density of the source node as the criteria based on which the information-value of a packet is determined. These approaches therefore make an assessment of how much effort they should put for a given information-value and then decide on different error corrections at the link layer depending on the amount of information carried out by the data packet.

Authors in [25] present two adaptive techniques that change the error control strategy based on number of hops that packets have traversed and quality of the wireless link over Nakagami-m fading channels. All nodes are required to have BCH, ARQ and Hamming codes programmed on them, which may occupy large portion of their memory. The approach proposed in [28] is another information-aware adaptive dissemination protocol, which adopts an error control code according to the spatial-density of the relay nodes. This approach requires all sensor nodes to keep four versions of BCH error control codes inside.

Since aggregated data is more important and needs better protection through error-control mechanisms, in [26], a packet is protected according to its aggregation degree. A packet that carries the aggregated information from many nodes should be protected in a more

sophisticated way. If such a message, containing the equivalent of many individual messages, is lost due to transmission errors then this has a detrimental effect on the application quality experienced. While these sophisticated mechanisms can increase data accuracy, they can also increase energy consumption of the sensor nodes. Such “aggregation aware link layer” error control techniques are suggested by Karl et al. [24] and Köpke et al. [26]. In [26] various aggregation degrees are mapped to various FEC protection mechanisms, while in [25] four different strategies are utilized to control the errors, among which the strategy that corrects the same number of bit errors as information entities are presented in the packet, is proved to be the most efficient one.

However, most of these approaches including [24, 25, 26, 28] do not take into account the number of remained hops to the base station and rely on only one factor which may influence the information-value of a packet. Moreover, these approaches [24, 25, 26] have limitations in terms of uniform deployment of nodes in the monitoring area.

Despite their advantages, the aforementioned approaches suffer from at least one of the following limitations:

- Most of the aforementioned approaches assume a simple independent loss channel, which is modeled by Bernoulli distribution and therefore they usually fail to be applied in error-bursty channels. Basically, all packet transmissions in these approaches have the same probability to fail and each transmission error is independent from the others. However, wireless channel is inherently lossy and often manifests itself in the form of burst errors correlated in time. Therefore, a reliable data dissemination should be capable of counteracting long-term fading possibly extending over several packets because of high concentrations of errors. To cope with this issue, a packet-level adaptive forward error correction may be a good alternative.
- Some approaches rely on the multiple path transmission, which are highly dependent on the network topology. In a chain-based topology where the communication of a sensor node is often restricted only to its immediate neighboring nodes (i.e. successor and predecessor node), we cannot well-benefit from the availability of multiple paths to salvage data packets from node/link failures. In case of using duplicate-sensitive aggregation functions such as SUM or AVERAGE, these multi-path approaches should employ some more resource demanding methods to filter out the redundant data. Moreover, these approaches require to some extent ensure that only one of the upstream neighbors forward the packet copies through multi-paths, otherwise they will introduce large amount of traffic, which leads to waste of resources in case all upstream neighbors send multiple copies. To strictly enforce that only one of the upstream nodes transmit the packet copies, these approaches may either incur extra overhead in the form of some

### 5.3. Related work

---

control packets or use some probabilistic methods to lower down the probability of transmitting a packet by the upstream nodes [27].

- Majority of the information-aware protocols do not evaluate the information-value of the packets and assume that sensor nodes have a priori knowledge to determine the importance level of the packets. Using these approaches, when a source node initiates a packet, it should set the importance level of the packet. However, asking sensor nodes to determine the importance level of the sensory data introduces new challenges which may require complex algorithms to perform pattern matching or execute artificial intelligence techniques. Moreover, in these approaches the importance level of each packet is set once on the source node and does not change along the path. Therefore, if an important sensory data is modified along the path in such a way that it cannot anymore reflect the phenomena state, transmitting it leads to wasting sensor/network resources. To cope with this issue, the importance level of the packets should vary along the path by considering the factors which may influence the packet importance level.
- Some approaches specially those which consider the aggregation degree to determine the importance level of the packets, poorly perform in case of being applied in non-uniformly distributed deployments. Non-uniform and unevenly distribution of sensor nodes results in some areas to be monitored by many sensors while other areas will be monitored only by a few nodes. Therefore, considering just the aggregation degree of the nodes may not well-reflect the importance level of the data. In this regard, the information-value of the packets should be determined in such a way that could also be applied for non-uniformly distributed deployments.
- The link-adaptation protocols may differ in the way of they exploit packet's arrival history to evaluate the current state of the channel, which is required to decide on the starting FEC level. Since typically each code is associated to one specific channel state, then the behavior and so the performance of these approaches for a certain type of the application could be quite different.

The above discussion highlights the need for an adaptive reliable disseminating protocol based on both packet information-value and link quality. To this end and to address most of above shortcomings, an adaptive energy-efficient reliable disseminating protocol is needed which (i) can be applied to non-uniform deployments with linear topology (ii) tackles the long-term error bursts, (iii) incorporates various factors that may influence the information quality of the packets, and (iv) considers packet delivery ratio as the link quality metric, rather than considering immediate channel quality indicators such as RSSI and SNR which are not appropriate for long-term error burst.

## 5.4. Problem statement and our contribution

Given an already deployed linear wireless sensor network in which TPC (see Chapter 2) and REC+ (see Chapter 3) have already been executed to select the active nodes and form the proper cluster-chains, the problem at hand is to design an adaptive, reliable, energy-efficient, and Information-Link-aware data dissemination protocol.

We summarize our contribution related to this chapter as:

1. Investigating and quantifying different factors which may influence the information-value of packets.
2. Incorporating the above identified factors in evaluation of informational content and importance of packets.
3. Proposing RAFEC, i.e., an adaptive, energy-efficient, reliable, information-link-aware data dissemination approach, which is able to (i) cope with periodic long-term loss process in a linear chain-based wireless sensor networks and (ii) switches among error control codes with different powers to vary the code strength and complexity in on demand.

## 5.5. An Information-link-aware data dissemination protocol (RAFEC)

In this section we elaborate on RAFEC, which is a Run-time Adaptive FEC-based data dissemination protocol that improves reliability of packet delivery based on the amount of information they carry over a bursty channel in a chain-based wireless sensor network.

To this end, (i) the mechanism for associating the error control codes to the states of QSGE model is described (ii) packet information-value and link quality are estimated (iii) the strategy using which an appropriate error control code is assigned to a specific packet is explained and (iv) the execution of RAFEC algorithm in both source nodes and relay nodes is elaborated.

### 5.5.1 Assigning error control codes to the channel states

As we stated in Section 5.2.1, in RAFEC the channel is modeled as a M-states QSGE model with a packet error rate  $PER_s$  assigned to each state  $S_s$ . Therefore, at any moment of time the state of the channel should fit one of the states specified by the channel model.

## 5.5. An Information-link-aware data dissemination protocol (RAFEC)

---

Having the packet error rate  $PER_s$  of each state  $S_s$  of the M-state QSGE model, an error control code which can effectively counteract the available errors may be designed. To this end, first we briefly explain the block codes which we utilize.

The error control codes in RAFEC are selected from a single family of FEC block codes, which exhibit the common characteristics and so reduce the decoder complexity. It is noteworthy that a “family of block” codes is a collection of codes obtained by varying the maximum error correction capability of a particular type of block code. For example all Reed-Solomon codes, whose length are  $n$  can form a “family of block” code  $FoB_{RS}(n)$  represented as:

$$FoB_{RS}(n) = \{RS(n, k) | k = n - 2 \times t, k > 0\} \quad (5.2)$$

where  $k$  represents number of original data and  $t$  represents correction capability of the Reed-Solomon code  $RS(n, k)$ . Each member of family block  $FoB_{RS}(n)$  can correct up to a specific number of error  $t$ .

RAFEC uses  $FoB_{RS}(n)$  for the M-state QSGE model. Therefore, each state  $S_s$  of the M-state channel, which exhibits a specific error rate  $PER_s$ , can adopt one member of  $FoB_{RS}(n)$  based on the below Equation provided that  $|FoB_{RS}(n)| \geq M$ :

$$ECC_s = RS(n, k_s \geq K_s) \quad (5.3)$$

$$RS(n, k_s \geq K_s) \in FoB_{RS}(n)$$

$$K_s = n \times (1 - PER_s)$$

To this end, the most efficient error control code denoted by  $ECC_s$  which exhibits the “just enough” correctional power for the channel state  $S_s$  is  $RS(n, K_s)$ . In this way, each channel state  $S_s$  can be described using two parameters  $PER_s$  and  $K_s$  as  $S_s(PER_s, K_s)$ .

In short, a particular coding strategy  $ECC_s$  is associated with each channel state  $S_s$ . The criteria by which this coding strategy is selected is addressed in above Equation.

Assigning error control codes to the channel states, as will be stated in Section 5.5.4.1, are performed by the base station once at the initialization phase.

### 5.5.2 Assessing packet information and link quality

Since the choice of error control code for each packet in RAFEC is based on the quality of service parameters, the information-value and packet importance as well as properties of



error traces which are captured from transmission history, the following tasks need to be performed by the sensor nodes:

- Estimation of packet's information value
- Estimation of link quality

### 5.5.2.1 Estimation of packet's information value

We recall that the information-value of a packet typically represents how informative a packet is for the application. Different packets typically contain different amount of information and so have difference importance level. Due to resource-constraint nature of wireless sensor networks, we could put packet's information-value into perspective with the amount of effort required to transmit a given packet.

The information-value could be influenced by several factors which may have different priorities in different applications. In what follows we express these factors which we then take into consideration to estimate information-value of a packet.

- Node functionality: Faulty sensor nodes could influence network operation and pose a challenging constraint in the design of a protocol for wireless sensor networks. Most of reliable data dissemination protocols usually concentrate on the link quality and less effort has been put into the node's functionality. Having a reliable dissemination protocol by itself is not useful if relay nodes through which data is disseminated are faulty and malfunctioning. Therefore, it is important that all sensor units relevant to the accomplishing task operate well-enough in order to ensure high reliability. In this regard, the quality of sensing and computing unit of relay nodes should be considered when estimating packet information-value. To estimate the quality of sensor units, we use Equation 2.8 presented in Chapter 2. The node functionality value is basically the confidence level of the node which was already obtained through coverage scheme in Chapter 2.
- Node contribution degree: The relative position of each node in the network may also impact the information-value of a packet being disseminated through the given node. Generally speaking, the higher contribution degree  $\psi$  of a node, the higher information-value. As can be seen from Figure 5.3, contribution degree of node  $S_7$  is higher than  $S_{12}$  as it monitors three critical points ( $\psi(S_7) = 3$ ) while  $S_{12}$  only monitors one critical point ( $\psi(S_{12}) = 1$ ). Node contribution degree is determined by the base station which informs each node about its  $\psi$ .

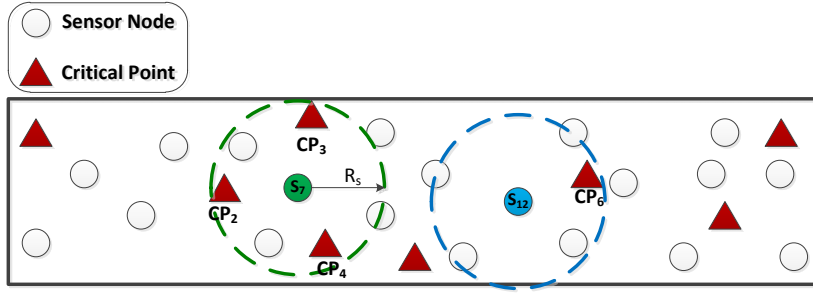


Figure 5.3. Illustrative example of node contribution degree

- Node spatial density: If sensor nodes are not evenly distributed, it is likely that some sensor nodes simultaneously and so redundantly observe a critical point while some nodes only and lonely observe a critical point. In this case, relying only on the coverage degree does not well reflect the amount of information being sent by the sensor nodes. As can be seen from Figure 5.4 although  $\psi(S_2) < \psi(S_{15})$ ,  $S_2$  is the only node which can observe critical point  $CP_1$  while critical point  $CP_6$  is being monitored by other three nodes in addition to node  $S_{15}$ . Therefore, a sensory data coming from a region that is already covered (either fully or partially) by other nodes has less informative content. On the other hand, if a sensor node is located in such a place where it covers one or some critical points which are not been observed by any otherwise node, its sensed data more likely carries quite significant information. Node spatial density can easily be determined by the base station in the initialization phase and then base station informs each node about it.

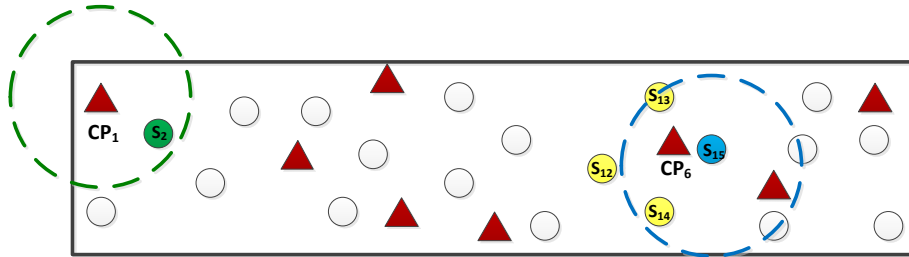


Figure 5.4. Illustrative example of node spatial density

- Strategic Area: The value of data collected from different critical regions may not necessarily be equal. As mentioned in Chapter 2, a given application (either always or sometime) may be more interested in data of some specific cells/regions. Therefore, the information-value of packets carrying this data is higher. As it can be seen from Figure 5.5 although node  $S_6$  monitors two critical points (i.e.,  $CP_2$  and  $CP_4$  both having

importance of 1) and node  $S_{13}$  monitors one critical point  $CP_6$  having importance of 4, information-value of data coming from node  $S_{13}$  is higher as it covers a more strategic area. In the initialization phase, the base station informs each node about the strategic level (or criticalness) of an area in where the given node is located.

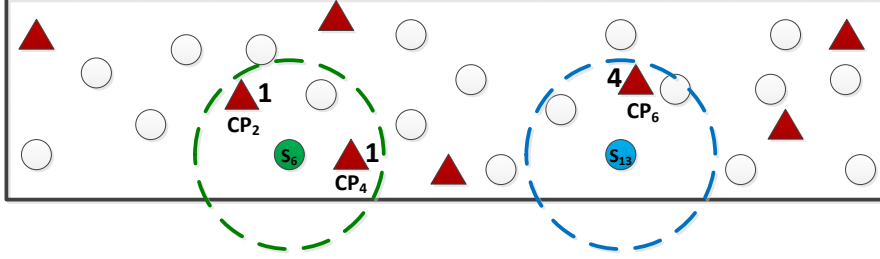


Figure 5.5. Illustrative example of different strategic area

- **Traveled Distance Ratio:** If a packet is lost at the first hops, (i) lesser energy has been consumed for its relay and (ii) lesser information (in case of doing aggregation along the path) are lost, compared to when it is lost at further hops. Therefore, it makes sense to use stronger error control codes for packets being relayed for longer distance. This parameter can be determined by increasing a counter (a packet's field) whose value is zero in the source node.

After identifying the aforementioned factors that may impact the quality of data packets, we here explain how to estimate the information-value and importance of packets per hop.

In Equation (5.4) we combine all aforementioned factors except Traveled Distance Ratio. The reason to leave Traveled Distance Ratio out of this equation is that all other factors are node-dependent while Traveled Distance Ratio is both packet-dependent and node-dependent.

$$\xi(i) = \sum_{k \in SCov_i} \frac{\sigma_k \times \gamma_{i,k}^{total}}{|CCov_k|} \quad (5.4)$$

where  $CCov_k$  represents the set of nodes that cover the common/critical point  $k$ ,  $SCov_i$  states set of common/critical points which have already been covered by sensor node  $i$ ,  $\sigma_k$  denotes how critical and strategic the data of common/critical point  $k$  is, and  $\gamma_{i,k}^{total}$  signifies node  $i$  functionality (estimated by Equation 2.8 of Chapter 2) for common/critical point  $k$ .

To also take Traveled Distance Ratio into account, we utilize Equation (5.5), where  $SIDp$  represents  $ID$  of the source node which initiates the data packet  $p$ . The numerator evaluates

## 5.5. An Information-link-aware data dissemination protocol (RAFEC)

---

the traveled distance while the denominator represents the distance between the base station (BS) and the source node.

$$\vartheta(i, p) = \frac{|i - SID_p|}{|BS - SID_p|} \quad (5.5)$$

Exploiting Equation (5.4) and (5.5) further, information-value denoted by  $\chi(p, i)$  of packet  $p$  being sent by sensor node  $i$  can be calculated using Equation (5.6).

$$\chi(p, i) = \begin{cases} \frac{\chi(p, i-1) + \varpi_1 \times \vartheta(i, p) + \varpi_2 \times \xi(i)}{\hat{\chi}} & i \neq SID_p \\ \frac{\xi(i)}{\hat{\chi}} & i = SID_p \end{cases} \quad (5.6)$$

where  $\hat{\chi}$  represents maximum information-value that a data packet may have.

Weights ( $\varpi_1, \varpi_2$ ) used in Equation (5.6) can be adjusted according to the application specific knowledge. For instance, if the application does not perform aggregation on the intermediate nodes and thus the relay nodes carry only the raw data, we may set  $\varpi_1 = 1$  and  $\varpi_2 = 0$ .

For the sake of simplicity and without loss of generality, we can map packet's information-value denoted by  $\chi$  into  $d_v$  discrete values. Doing so, we will have  $d_v$  different packet types each of which contains a specific amount of information and thus their required reliabilities are different. Therefore,  $d_v$  also shows the number of required error control codes each of which is assigned to a specific information-value. In this thesis by using Equation (5.7), we map packet's information-value into discrete values 1, 2 and 3. By doing so, three different packet types will be defined in terms of information-value they may have. However, depending on the available error control codes which are implemented in the sensor nodes we can have different values for  $d_v$ .

$$\gamma = \begin{cases} 1 & 0 \leq \chi < 0.3 \\ 2 & 0.3 \leq \chi < 0.6 \\ 3 & 0.6 \leq \chi \end{cases} \quad (5.7)$$

### 5.5.2.2 Estimation of link quality

Link quality estimation is a fundamental building block for the higher layer protocols of wireless sensor networks in order to cope with low-quality or unreliable links. RAFEC relies

on channel quality estimation to adopt an appropriate error control code which can effectively enhance the reliability of packet delivery. To estimate the link quality, RAFEC employs a passive link monitoring strategy, which exploits existing traffic without incurring additional communication overhead.

The link quality is normally estimated either by counting the number of hello/data packets in a small period of time or by taking the history of transmissions into account. The first approach is accurate but usually needs many packets to be sent. This may come at the cost of high energy consumption and will fail in capturing long-term fading. The second approach is more energy efficient and more robust in coping with a periodic long-term loss process. Therefore, RAFEC employs the second strategy to estimate the link quality.

The link quality estimation process in RAFEC is performed first over a sequence of packets (say packet-level estimation) and then over a sequence of code-words (say code-word-level estimation). This history-based evaluation of link quality utilized by RAFEC provides a means to cope with longer-term interferences since this strategy does not immediately switch to a less/more powerful code after one single successful/failed transmission.

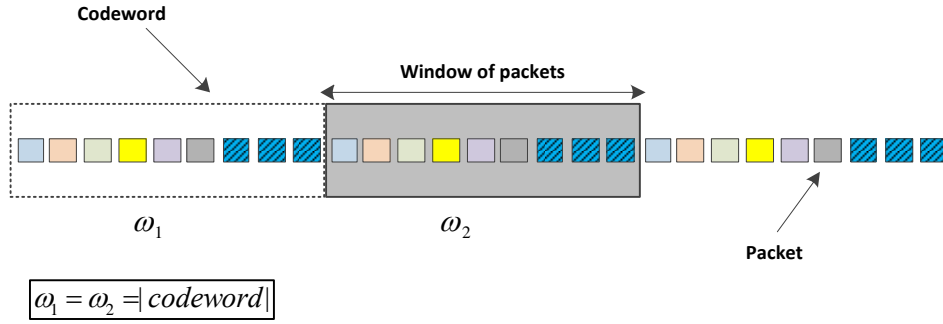
At the packet level, link quality is estimated by counting the number of lost/erroneous and error-free packets during a time window denoted by  $\omega$  whose size is equal to the code-word length as shown in Figure 5.6. Therefore, the number of packets required to estimate the link quality is represented by the window size  $|\omega|$ . One should note that the burst size in the network should be smaller than the window size  $\omega$  in order to well estimate the link quality. Basically, in the initialization phase the code-word length and thereby  $\omega$  is determined based on the biggest burst size that can happen in the network.

During each time window  $\omega_w$  a sensor node  $i$  collects and extracts information from the received packets in order to find out the packet loss ( $\varphi_{ji}^w$ ) of downstream link  $(j,i)$  for that time interval  $w$ , as shown in Equation (5.8).

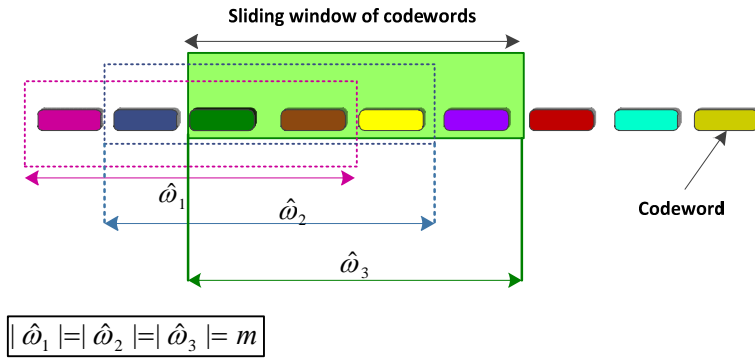
$$\varphi_{ji}^w = \frac{nef}{nt} \tag{5.8}$$

where  $nef$  represents the number of error-free packets received by node  $i$  and  $nt$  represents total number of packets transmitted by node  $j$  over link  $(j,i)$  during last time window.

## 5.5. An Information-link-aware data dissemination protocol (RAFEC)



**Figure 5.6. Packet sequence includes windows of packets**



**Figure 5.7. Code-word sequence includes sliding-windows of code-words**

After gathering a number of  $\varphi_{ji}^w$  for consecutive windows which in turn will form the code-word sequence (as shown in Figure 5.7), the final packet loss rate will be calculated by averaging all  $\varphi_{ji}^w$ . To this end, each sensor node  $i$  stores the calculated packet loss rate  $\varphi_{ji}^w$  for  $m$  consecutive windows. By doing so, the set of packet loss rate  $Set\varphi_i(u)$  estimated by node  $i$  between time interval  $u-m+1$  and  $u$  for downward link  $(j,i)$  can be represented by Equation (5.9).

$$Set\varphi_i(u \geq m) = \bigcup_{w=u-m+1}^u \varphi_{ji}^w \quad (5.9)$$

where  $m = |\hat{\omega}|$  is the size of sliding window in a sequence of code-words as shown in Figure 5.7. Basically,  $m$  represents the number of code-words required to estimate the link quality. Having  $Set\varphi_i(u)$ , which presents statistics about a given link qualities over the last sliding window, different metrics could be used to determine and identify the effective error

rate that should be considered and may occur in the next time window. Dependent on the application at hand, different metrics are typically utilized to measure the effective-error-rate. Among these metrics, the regular average error rate denoted by  $\bar{r}\bar{\varphi}$ , the weighted-average error rate denoted by  $\bar{w}\bar{\varphi}$ , and the maximum error rate denoted by  $\bar{m}\bar{\varphi}$  are three mostly used metrics. As it will be presented in Section 5.5.3, in RAFEC we employ all these three metrics but under different circumstances in order to measure effective-error-rate. In what follows, the way of assessing these three metrics are explained.

The average packet error rate discovered by node  $i$  in each time interval  $u \geq m$  can be calculated using Equation (5.10).

$$\bar{r}\bar{\varphi}_i(u) = \frac{\sum_{w=u-m+1}^u \varphi_{ji}^w}{m} \quad (5.10)$$

As it can be seen from Figure 5.7 and Equation (5.10), the link quality is estimated by averaging the quality estimations obtained during  $m$  consecutive windows. In this way, each sensor node collects and stores the information related to the received packets during last  $m$  time windows  $[\omega_{u-m+1}, \omega_u]$  in order to predict the delivery capacity of the link over the next time window  $\omega_{u+1}$ .

In short, to well-estimate the link quality, we should accurately set two parameters (i) window size  $|\omega|$  and (ii) sliding window size  $|\hat{\omega}|$ , which represent the required number of packets and number of code-words, respectively.

In addition to regular average packet error rate  $\bar{r}\bar{\varphi}_i(u)$ , as will be explained later, RAFEC requires to identify the maximum packet error rate  $\bar{m}\bar{\varphi}_i(u)$  that is captured during last sliding window which includes the packet error rate of  $m$  consecutive time windows  $[\omega_{u-m+1}, \omega_u]$ .

$$\bar{m}\bar{\varphi}_i(u) = \max\left(\bigcup_{w=u-m+1}^u \varphi_{ji}^w\right) \quad (5.11)$$

RAFEC also requires to be aware of the weighted-average of the packet error rate  $\bar{\varphi}_i^{wg}(u)$  which is estimated by Equation (5.12). This weighted-average gives higher weights to the most recent error statistics in the network.

$$\bar{w}\bar{\varphi}_i(u) = \frac{\sum_{w=u-m+1}^u (\alpha_w \times \varphi_{ji}^w)}{\sum_{w=u-m+1}^u (\alpha_w)} \quad (5.12)$$

$$\alpha_w = w - (u - m)$$

## 5.5. An Information-link-aware data dissemination protocol (RAFEC)

---

As will be stated later in Section 5.5.3., dependent on amount of information a packet carries, we will use one of these three metrics to capture the effective-error-rate on the links.

### 5.5.3 Adaptive packet-link-local error control

So far we have estimated information-value of packets and have assigned the proper Reed-Solomon code to each state of our QSGE channel model according to the loss rate of each state. Moreover, we have introduced three metrics denoted by  $\overline{r\varphi}$ ,  $\overline{m\varphi}$  and  $\overline{w\varphi}$ , which can be used to estimate the link quality in three different ways.

Having both information-value of packets and packet error rates captured from transmissions history, strength and complexity of the error control codes can be adapted on demand. Having higher information-value or poorer link quality requires utilization of a more powerful error control code. On the contrary, having lower information-value or higher link quality requires a weaker code.

It is noteworthy that we consider a multi-hop FEC protection mechanism, in which intermediate nodes need to perform encoding and decoding functions individually and locally at each hop. This way of locally protection helps our approach being easily applied to large-scale networks. The main reason of targeting a multi-hop protection mechanism is that it was shown in [29] that isolating and recovering erroneous packets at each hop is more effective than the conventional end-to-end FEC protection mechanism especially in an error-prone environment. The limited radio range of wireless sensor nodes makes single-hop packet transmission to the base station impractical especially for large-scale or long deployment of wireless sensor networks.

In Section 5.5.1, we explained how Reed-Solomon code is assigned to each channel state of QSGE model based on the packet error rate  $PER_s$  of each state  $S_s$ . Basically, effective-error-rate  $\varphi(u)$  of a link at any moment of time  $u$  should correspond to one of the  $PER_s$  specified by the QSGE model as shown in Equation (5.13).

$$\varphi_i(u) \in \bigcup_{s=1}^M PER_s \quad (5.13)$$

where  $\varphi_i(u)$  represents effective packet error rate calculated by node  $i$  for the downward link (i-1,i) during time window  $u$ . In this way, as shown in Equation (5.14), each sensor node upon receiving a packet first finds state  $S_s$  of the QSGE model, which best describes the upward link state in terms of  $\varphi$ .

$$S_s = \{S_x \in QSGE | PER_x \leq \varphi < PER_{x+1}\} \quad (5.14)$$



Then according to Equation (5.15), the error control code  $ECC_s$ , which is associated to the state  $S_s$  could be decided as the code  $EEcc(\varphi)$  that should be utilized for the error rate  $\varphi$ .

$$EEcc(\varphi) = ECC_s \quad (5.15)$$

Depending on amount of information that a packet carries, effective-error rate  $\varphi$  should be initialized with one of the three metrics presented in Section 5.5.2.2, i.e.  $\bar{r}\varphi_i(u)$ ,  $\bar{w}\varphi_i(u)$  and  $\bar{m}\varphi_i(u)$ . It is worth recalling that  $\bar{r}\varphi_i(u)$ ,  $\bar{w}\varphi_i(u)$  and  $\bar{m}\varphi_i(u)$  represent regular average error rate, weighted-average error rate, and maximum error rate captured during last sliding window to be used by node  $i-1$  over link  $(i,i+1)$  for next time window  $\omega_{u+1}$ .

Initializing the effective packet error rate  $\varphi$  with regular average error rate  $\bar{r}\varphi_i(u)$  is not always efficient specially if both packet error rate variance and packet information-value are high. If information-value of a packet is low, it will not be efficient to consider  $\bar{m}\varphi_i(u)$  as the effective error rate  $\varphi$ . For example, if the set of the packet error rates estimated by node  $i$  for time interval  $u-m+1$  and  $u$  for downward link  $(j,i)$  is:

$$Set\varphi_i(u) = \{0.2,0.3,0.4,0.4,0.5\} \quad (5.16)$$

The value of the three aforementioned metrics would be as follows:

$$\bar{r}\varphi_i(u) = 3.6, \bar{w}\varphi_i(u)=0.4, \bar{m}\varphi_i(u)=0.5$$

In this way, the average loss rate is 3.6 but since the variance is not zero it is possible that the packet error rate  $\varphi_{ji}^{u+1}$  for the next time interval  $u+1$  exceeds the regular average loss rate  $\bar{r}\varphi_i(u)$  and so  $\varphi_{ji}^{u+1} > 3.6$ . In this case, the packets which are erroneously received by node  $i$  may not be well-recovered and so this situation should be avoided for the high informative packets. Therefore, it will be wise to consider the worst case error rate during last sliding window, as the effective error rate for such informative packets.

$$\varphi = \begin{cases} \min(\bar{r}\varphi_i(u), \bar{w}\varphi_i(u)) & \gamma = 1 \\ \max(\bar{r}\varphi_i(u), \bar{w}\varphi_i(u)) & \gamma = 2 \\ \bar{m}\varphi_i(u) & \gamma = 3 \end{cases} \quad (5.17)$$

Our strategy to estimate the effective error rate  $\varphi$  as expressed in Equation (5.17) can be summarized as:

- If information-value of the packet is  $\gamma = 1$  then  $\varphi$  will be estimated with the minimum value of  $\bar{r}\varphi_i(u)$ ,  $\bar{w}\varphi_i(u)$ .

## 5.5. An Information-link-aware data dissemination protocol (RAFEC)

---

- If the packet has higher information-value ( $\gamma = 2$ ) then  $\varphi$  will be estimated with maximum value of  $\overline{r\varphi}_i(u), \overline{w\varphi}_i(u)$ .
- If the packet has the highest information-value ( $\gamma = 3$ ), the effective packet error rate  $\varphi$  will be estimated with maximum error rate  $\overline{m\varphi}_i(u)$  occurred in the previous sliding window.

As it can be seen from Equation (5.17), the effective packet error rate  $\varphi$  based on which the error control code is selected, varies according to the information-value  $\gamma$ . In this regard, a packet with high/low information-value should be equipped with a strong/weak error control code  $EECC$  presented in Equation (5.15).

### 5.5.4 Execution of RAFEC algorithm

In this section we elaborate on the execution of RAFEC algorithm.

#### 5.5.4.1 Initialization phase

In the initialization phase, the base station disseminates the information related to the nodes' contribution degree, the nodes' spatial density and the strategic-level of the area in where the nodes are located. It is worth recalling that (as stated in Section 5.5.2.1) these information greatly contribute to obtain the information-value of a packet.

Assigning error control codes to the channel states, which was explained in Section 5.5.1 are performed by the base station once at the initialization phase. The base station disseminates the information related to the error control code assignment and so informs all sensor nodes about it. The sensor node upon receiving this information should save and keep them in order to be capable of selecting an appropriate error control code for a given packet while encountering a specific channel condition. To this end, each sensor node  $i$  maintains a  $M$  rows table each row of which consists of a pair  $(PER_s, ECC_s)$  as shown below:

$PER_1$	$ECC_1$
$PER_2$	$ECC_2$
$\vdots$	$\vdots$
$PER_M$	$ECC_M$

where  $M$  is number of states in the QSGE model,  $PER_s$  represents the packet error rate each state  $S_s$  exhibits and  $ECC_s$  denotes the error control code associated to each state  $S_s$  according to Equation (5.15). Each sensor node  $i$  at any moment of time refers to this table to select which error control code  $ECC_s$  should be utilized for an estimated effective packet error rate  $\varphi$  where  $PER_s = \varphi$ .

Before going to the next step, each sensor node  $i$  must initialize three queues in its buffer:

- $Q_r$  that stores all error-free received packets (either data or parity) which belong to the current and in-progress code-word  $cw_j^i$  on node  $i$ . This queue is used in decoding procedure of the current code-word  $cw_j$  to possibly reconstruct the erroneous packets.
- $Q_s$  that consists of only data packets (either error-free or reconstructed) which should be transmitted. This queue comes in handy in encoding procedure of the already received/reconstructed data packets in order to construct the code-word  $cw_j^{i+1}$  for next hop.
- $Q_{tmp}$  which stores the packets that may arrive but does not belong to the current code-word and should be processed later.

#### 5.5.4.2 Data dissemination phase

In the data dissemination phase, RAFEC involves processes being performed by either source node or relay node.

##### 5.5.4.2.1 Source-node-based dissemination

The source node starts initiating data packets and estimates the information-value  $\gamma_p$  of each data packet  $p$  by employing Equation (5.6) and (5.7) which are described in Section 5.5.1. Then, considering the link quality estimation of the last sliding window, the effective error rate  $\varphi$  for a given information-value  $\gamma_p$  is estimated as explained in Section 5.5.3. Thereafter, the following two tasks will be carried out:

- **Encoding  $cw_j^{snd}$ :** Having information-value of each packet and the effective error rate, using Equation (5.15), the most suitable error control code RS( $n, k^p$ ) is selected by the source node for a given packet  $p$ . Thereafter, source node sends the packet  $p$  upward while a copy of it is stored in a queue called  $Q_s$  to be used later for encoding. The procedure of generating data packets, transmitting them upward and putting a copy of them in  $Q_s$  continues as long as the size of queue becomes equal or greater than  $k^p$  of the strongest code which is decided for the current  $Q_s$  members. Thereafter, the source node starts the encoding procedure over the first  $k^p$  data packets stored in  $Q_s$  in order to generate the  $n - k^p$  parity packets. Afterward, the  $n - k^p$  parity packets are transmitted upward and the source node removes  $k^p$  packets from the head of queue  $Q_s$  and shifts other members (if any) to start the encoding procedure for them. The  $k^p$  data packets and  $n - k^p$  parity packets altogether form the code-word  $cw_j^{snd}$ , where  $snd$  represents the ID of the source node and  $j$

## 5.5. An Information-link-aware data dissemination protocol (RAFEC)

represents the number of code-words already made and sent by the source node. The flowchart of encoding procedure is illustrated in Figure 5.8.

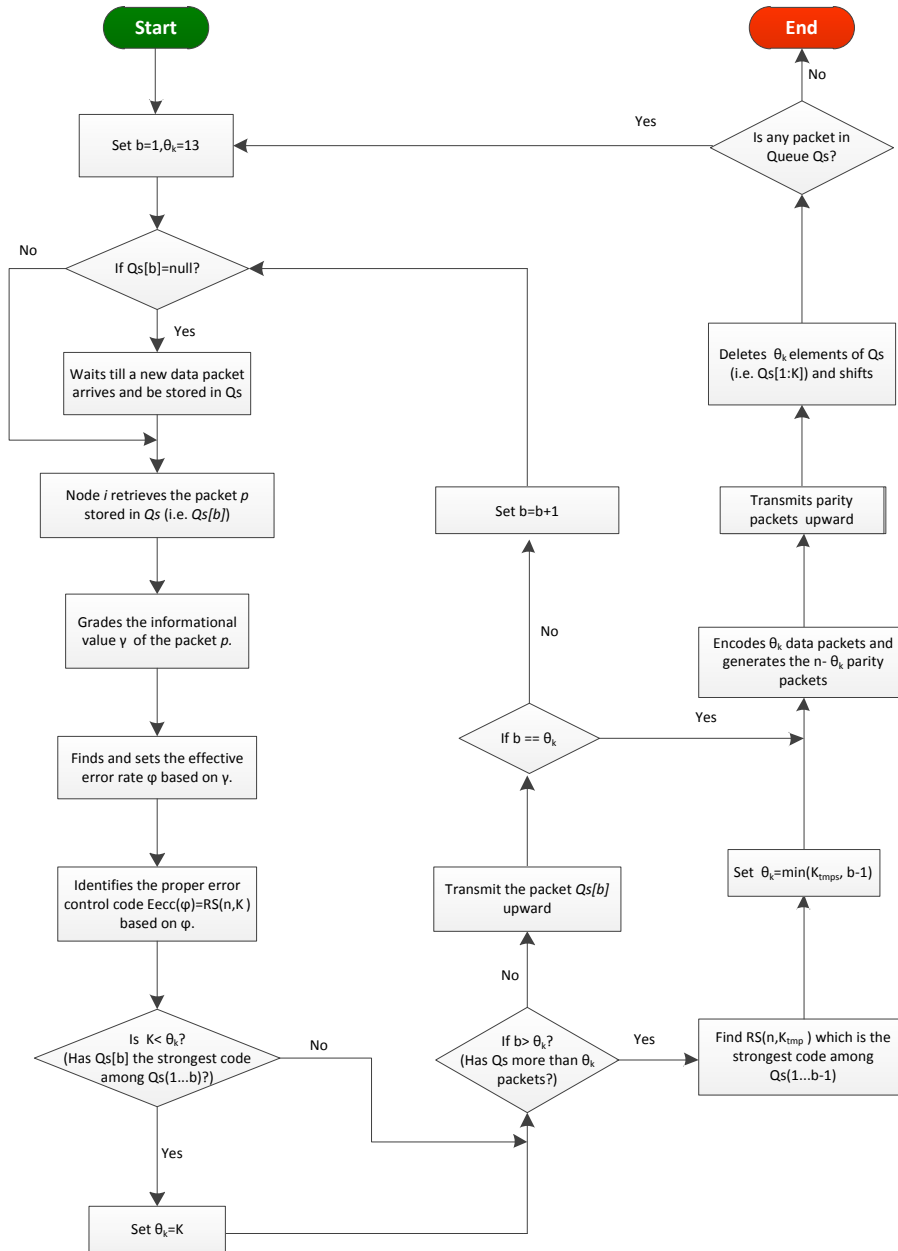


Figure 5.8. Flowchart of encoding process of a code-word

- **Updating Upward Link Quality:** After sending all packets of the code-word  $cw_j^{snd}$ , source node  $snd$  may receive a control packet with an update about the upward link quality  $\varphi_{(snd,snd+1)}^w$ , which is estimated by the upstream node  $snd+1$  based on the statistics collected during the last time window  $\omega_w$ . After receiving this control packet, source node  $snd$  re-estimates  $\bar{r}\varphi$ ,  $\bar{m}\varphi$ , and  $\bar{w}\varphi$  which are important to assess the effective error rate. This will be done using Equations (5.10), Equation (5.11) and Equation (5.12). These updates will later be used by node  $snd$  to select an appropriate error control code for encoding the next code-word  $cw_{j+1}^{snd}$ .

#### 5.5.4.2.2 Relay-node-based dissemination

Upon receiving packet  $p$ , each relay node  $i$  puts it in the appropriate queue(s) which are initiated in initialization phase (Section 5.5.4.1). If  $p$  is an error-free data packet which belongs to the current code-word  $cw_j^i$ , sensor node  $i$  transmits it upward and puts a copy of it in the  $Qs$  to be used later for encoding the code-word  $cw_j^{i+1}$  over the next hop.

Since each packet carries some information indicating the used error control code, sensor node  $i$  can easily figure out the time when the recovery mechanism should be started. This will happen when either the last packet of the current code-word or a packet of the new code-word arrives. In this step, relay node  $i$  performs the following two main tasks:

- **Estimating downward link quality:** Relay node  $i$  estimates the quality of the downward link  $(i-1,i)$  using Equation (5.8) and by exploiting the extracted statistics from the packets which belong to the code-word  $cw_j^i$ . Thereafter, it sends a control packet to inform relay node  $i-1$  about the latest error rate of link  $(i-1,i)$ .
- **Decoding  $cw_j^i$ :** Relay node  $i$  starts the decoding procedure for the code-word  $cw_j^i$  whose packets are stored in queue  $Qr$ . After reconstructing the erroneous/lost packets, sensor node  $i$  empties the queue  $Qr$  and adds the recovered data packets to the  $Qs$  to be sent over the next hop. Moreover, the queue  $Qr$  is possibly filled with packets which were stored in queue  $Qtmp$  and are waiting to be processed as the next code-word  $cw_{j+1}^i$ .

After decoding the code-word  $cw_j^i$ , the error-free and recovered data packets stored in queue  $Qs$  should be transmitted upward. To this end, a new code-word  $cw_j^{i+1}$  will be constructed using all or a number of data packets (dependent on the code requirements) stored in  $Qs$  and the parity packets which need to be generated by the node  $i$ . Thereafter relay node  $i$  performs the following two tasks:

## 5.5. An Information-link-aware data dissemination protocol (RAFEC)

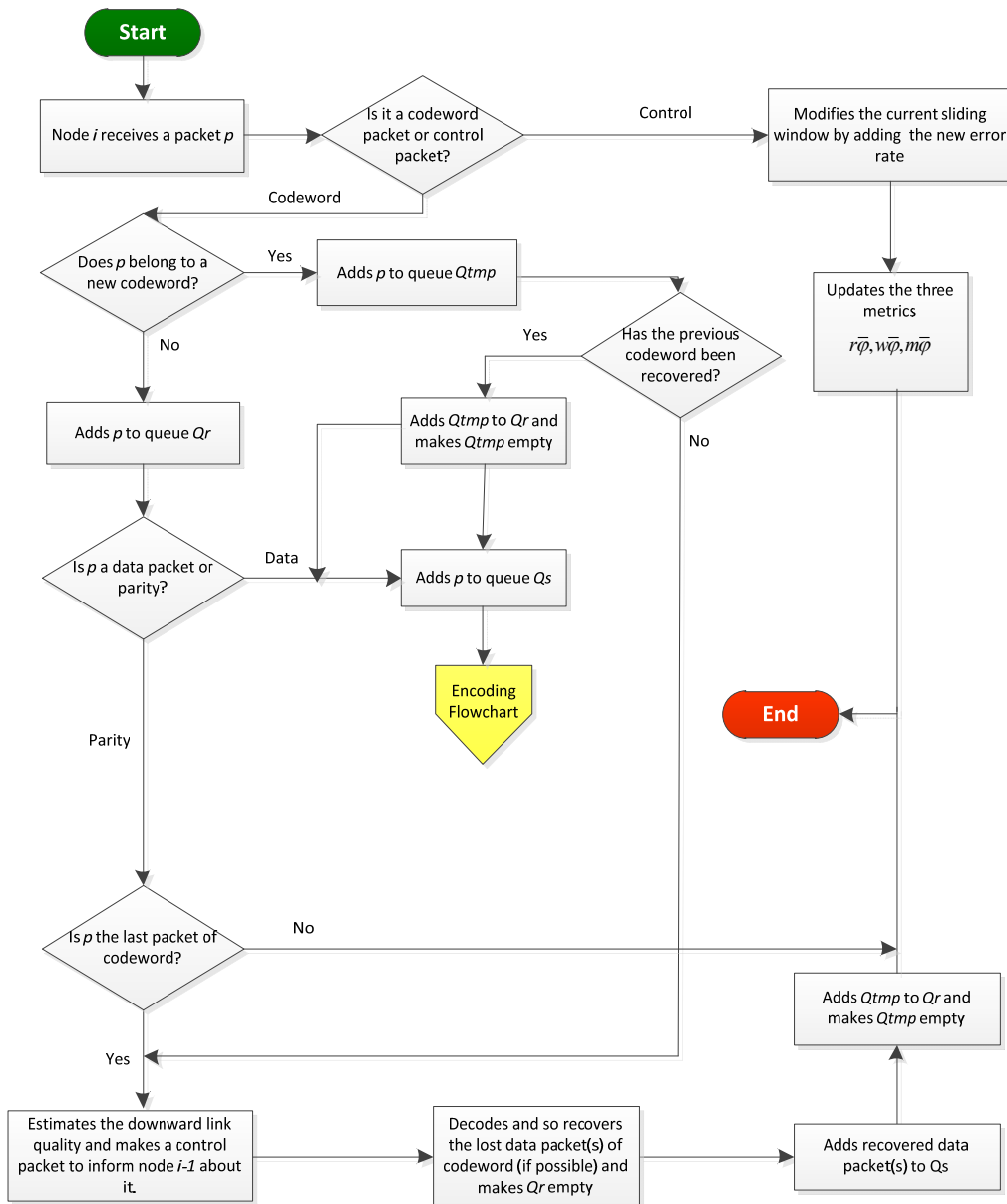


Figure 5.9. Flowchart of process in relay node

- **Encoding  $cw_j^{i+1}$ :** Due to variability of  $\gamma$  and  $\varphi$ , a transmitted code-word  $cw_j^{i+1}$  (by node  $i$  towards node  $i+1$ ) does not necessarily consist of the same packets that the received code-word  $cw_j^i$  by node  $i$  from node  $i-1$  had. Therefore, having information-value  $\gamma$  of the packets in  $Qs$ , which are ready to be sent and the effective error rates, the well-suited error control code RS(K,n) is decided by node  $i$  for the code-word  $cw_j^{i+1}$ , where K represents the size of data-word of the strongest associated Reed-Solomon code to the  $Qs$  members. In this way, first the K data packets which are retrieved from the head of  $Qs$  are transmitted upward to the node  $i+1$ . Then, these K data packets are used by node  $i$  for the encoding process to generate  $n-K$  parity packets to be transmitted upward. The  $K$  data packets along with the generated  $n-K$  parity packets form the code-word  $cw_j^{i+1}$ . If the queue is not empty, relay node  $i$  starts the encoding procedure over the remaining packets.
- **Updating upward link quality:** This task is similar to the corresponding task for the source node explained in previous Section.

The flowchart of relay node-based dissemination is depicted in Figure 5.9.

Activities performed by the relay node  $i$  can be organized into sequences each of which may correspond to processing one code-word as shown in Figure 5.10.

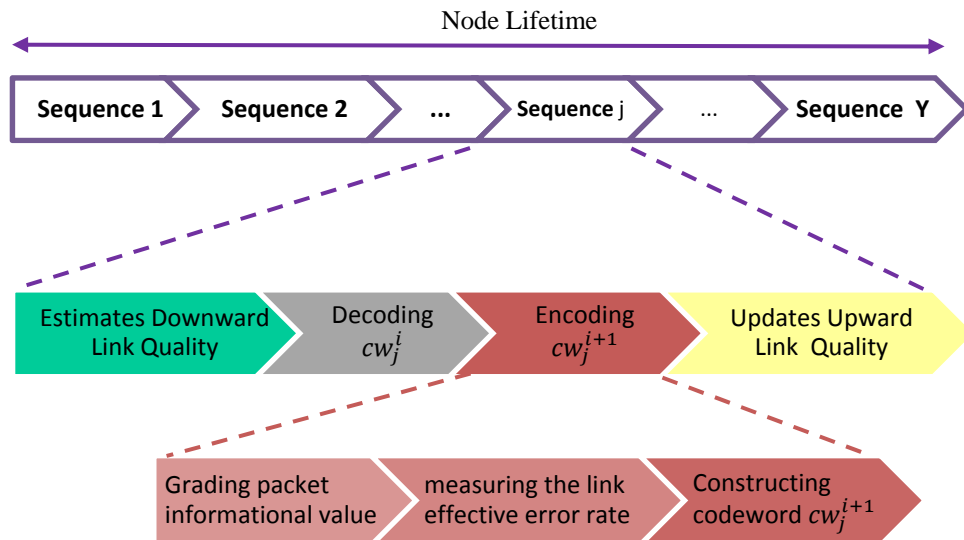


Figure 5.10. Activities performed by relay node  $i$

### 5.6. Performance evaluation of RAFEC

#### 5.6.1 Performance metrics

We consider the following metrics to evaluate the performance of our approaches under different circumstances.

- **Reliability Ratio (RR):** This metric indicates how well the protocol performs in terms of recovering erroneous or lost packets. Reliability ratio is expressed as the ratio of the data packets that were either received error-freely or recovered by the receiver nodes, to the number of sent data packets. Since RAFEC is a multi-hop FEC protection mechanism, in which intermediate nodes need to perform encoding and decoding individually and locally in each hop, we evaluate the average RR using Equation (5.18).

$$RR = \frac{\sum_{i=1}^{\bar{N}_s-1} N^{Ref}(i+1) + \sum_{i=1}^{\bar{N}_s-1} N^{RaR}(i+1)}{\sum_{i=1}^{\bar{N}_s-1} N^{TD}(i)} \quad (5.18)$$

where  $\bar{N}_s$  represents the number of sensor nodes and  $N^{TD}(i)$  is the number of data packets transmitted by node  $i$ .  $N_Y^{Ref}(i+1)$  represents the number of data packets received by node  $i+1$  error-freely while  $N^{RaR}(i+1)$  represents the number of erroneous or lost data packets correctly being recovered by node  $i+1$ .

Clearly, the greater the value of RR means the better the performance of the protocol.

- **Information-aware reliability ratio (IRR):** Since RAFEC is an adaptive error control, which is sensitive to the amount of information a packet carries, it will be even more insightful to evaluate reliability ratio by taking information-value of the packets into account using Equation (5.19):

$$IRR(\gamma) = \frac{\sum_{i=1}^{\bar{N}_s-1} N_Y^{Ref}(i+1) + \sum_{i=1}^{\bar{N}_s-1} N_Y^{RaR}(i+1)}{\sum_{i=1}^{\bar{N}_s-1} N_Y^{TD}(i)} \quad (5.19)$$

where  $N_Y^{Ref}$ ,  $N_Y^{RaR}$  and  $N_Y^{TD}(i)$  have the same description of the corresponding variables in Equation (5.18) but with an emphasize on the information-value  $\gamma$ . According to Equation (5.19), we will have three different IRRs each of which representing the achieved reliability ratio for a specific information-value  $\gamma$ .

Basically, the relationship between RR and IRR is shown in Equation (5.20), which states that average IRR over three information-value will result in RR.



$$RR = \frac{\sum_{\gamma=1}^3 IRR(\gamma)}{3} \quad (5.20)$$

- **Code rate:** This metric represents the proportion of the useful (non-parity) packets in a code-word. By the means of this metric, we express the code's efficiency and the redundancy introduced by the code. If the size of the code-word is  $n$  packets out of which  $k$  packets contain original data (data-word), the code rate is evaluated based on Equation (5.21). Higher code rate results in more data packets and less redundancy at the cost of not being able to recover as many errors.

$$CodeRate = \frac{k}{n} \quad (5.21)$$

- **System Efficiency :** It is generally accepted that additional parity packets (or lowering the code rate) can be tolerated as long as loss-resiliency at the receiver side is increased. Therefore, the system efficiency metric is introduced to express the tradeoff between the energy expenditure and reliability. To precisely evaluate the efficiency of different error control code, the efficiency ratio metric is stated in Equation (5.22):

$$SE = \frac{\sum_{i=1}^{\bar{N}_s-1} N^{Ref}(i+1) + \sum_{i=1}^{\bar{N}_s-1} N^{RaR}(i+1)}{\sum_{i=1}^{\bar{N}_s-1} N^{TD}(i) + \sum_{i=1}^{\bar{N}_s-1} N^{TR}(i)} \quad (5.22)$$

where  $N^{TR}$  represents the number of redundant (parity) packets sent by node  $i$ . The other variables have the same definition as expressed for Equation (5.18).

The system efficiency ideally should be 1 when no redundant packet is transmitted and links are reliable and do not need to recover any error.

Since RAFEC is an information-aware protocol, we address the information gain in the System Efficiency by turning Equation (5.22) into Equation (5.23). To this end we make a relation between information-value arriving at the destination with the amount of redundancy (parity packets) and define Equation (5.23) as:

$$ISE = \sum_{\gamma=1}^3 v(\gamma) \times \frac{\sum_{i=1}^{\bar{N}_s-1} N_{\gamma}^{Ref}(i+1) + \sum_{i=1}^{\bar{N}_s-1} N_{\gamma}^{RaR}(i+1)}{\sum_{i=1}^{\bar{N}_s-1} N_{\gamma}^{TD}(i) + \sum_{i=1}^{\bar{N}_s-1} N_{\gamma}^{TR}(i)} \quad (5.23)$$

where  $v(\gamma)$  represents the amount of gain that an application earns by receiving a packet with an information-value  $\gamma$ . We assume that the gain of a packet with  $\gamma = 3$  is twice of that for  $\gamma = 2$  and four times of that for  $\gamma = 1$ . Therefore,  $v(\gamma = 3) = 2 \times v(\gamma = 2) =$

## 5.6. Performance evaluation of RAFEC

---

$4 \times v(\gamma = 1)$ . By doing so, receiving a packet with information-value  $\gamma = 2$  is worth twice as much as receiving a packet with information-value  $\gamma = 1$ .

### 5.6.2 Simulation setup and scenario

Unless otherwise states, the simulation setup is similar to that of in Chapter 4. The deployment area is divided into some regions ( $l = 25m$ ) half of which are labeled as critical and the rest are labeled as uncritical. It is worth mentioning that since RAFEC is a link-local error control approach, the number of sensor nodes does not much influence the performance of the application. Similar to previous chapters, the sensing range of nodes is set to 35m. We then send 5000 packets from one source node to the base station with frequency of 1 pkt/s. The strategic level (or criticalness) of the critical regions is selected from the interval [2,5] while the strategic-level of the uncritical regions are 1.

At any moment in time, 70% of all nodes and links work almost properly with failure rate of 0.09. The failure rate of other 30% of the nodes is set to 0.85. The failure rate of other 30% of the links vary according to a five-state QSGE model which will be state later. The selection of failing nodes/links occur randomly after every 1000 time unit in order to simulate temporal correlation among failures of those 30% nodes/links. All faulty nodes misbehave according to the three-state Markov model (shown in Chapter 2) for a duration of 1000 time unit in order to simulate temporal correlation among failures of those 30% nodes. One should note that these failure rates (i.e. 0.09 and 0.85) and the parameters related to the three-state Markov model for the normal/faulty nodes are similar to what is stated in Chapter 2.

Five-states quasi-stationary Gilbert-Elliot erasure channel (as explained in Section 4.2.1 of Chapter 4.) is used. In order to simulate a slowly varying channel, similar to Chapter 4, the following specifications are used:

$$\Gamma = \begin{bmatrix} 0.995 & 0.0035 & 0.0015 & 0 & 0 \\ 0.0033 & 0.992 & 0.0033 & 0.0014 & 0 \\ 0.0015 & 0.0025 & 0.992 & 0.0025 & 0.0015 \\ 0 & 0.0014 & 0.0033 & 0.992 & 0.0033 \\ 0 & 0 & 0.003 & 0.007 & 0.99 \end{bmatrix}$$

Each state  $S_s$  of the five-state QSGE model corresponds to one  $PER_s$  as:

$$PER_1=0.1, PER_2=0.3, PER_3=0.4, PER_4=0.5, PER_5=0.7.$$

We model sending packets in each state of QSGE model first according to a Gilbert-Elliot model and then as a series of Bernoulli trials. The Gilbert-Elliott channel model is

defined by  $p$  and  $q$  which change according to the  $N$  and  $N-K$  parameters of the codes assigned to  $S_s$  (Table 5.1.) These two parameters are obtained as:

$$p^1=0.07, p^2=0.09, p^3=0.11, p^4=0.14, p^5=0.2.$$

$$q^1=0.5, q^2=0.25, q^3=0.166, q^4=0.125, q^5=0.1.$$

As stated before, we use the Reed-Solomon code as our FEC with a code-word made up of  $K$  data packets and the FEC adds  $N-K$  redundant parity packets. Among these  $N$  packets, if  $K$  or more packets are received, the original code-word can be successfully reconstructed. In our approach, a length-15 Reed-Solomon (RS) code (i.e.  $N=15$ ) is chosen over a five states channel for packets with three different information-values. The error control code  $ECC_s$  which is assigned to each state  $S_s$  is presented in Table 5.1. The error codes contained in this table are increasing in their correctional power from the left to the right, and similarly with respect to computational and parity overhead.

**Table 5.1. Error control codes of each state**

State	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$
<b><math>ECC_s</math></b>	RS(15,13)	RS(15,11)	RS(15,9)	RS(15,7)	RS(15,5)

The information-value weights are set to  $\varpi_1 = 1, \varpi_2 = 1$  and  $\omega_1 = \omega_2 = \omega_3 = 1$ . Moreover, the channel estimation windows size is  $|\omega| = 15$  while the sliding window size is  $|\hat{\omega}| = 5$ .

### 5.6.3 Performance evaluation

Figure 5.11 represents the average reliability ratio RR calculated using Equation (5.18), irrespective of how informative and useful a delivered packet is. At the beginning when packet error rate is around 0.1, the reliability ratio of all approaches are more or less the same since there are few errors introduced by the channel. When the packet error rate increases, the reliability ratio of all approaches except RAFEC and RS(15,5) sharply reduce. The reason is that they cannot cope with the error rate which is higher than their correction capability. Since this graph shows the overall RR for all three information-values, the adaptive behavior of the RAFEC on the basis of the packets' information-value  $\gamma$  cannot be deduced. Figure 5.11 presents the packet reliability ratio which is averaged over all three information-values. However, in RAFEC we are more interested to see the distribution of RR among different types of the packets considering their information-values. Therefore, the reliability ratio, as given in Equation (5.18), is not a sufficient metric to fully evaluate the

## 5.6. Performance evaluation of RAFEC

performances of the different FEC schemes. An excessive FEC redundancy rate, similar to what RS(15,5) imposes, with high bandwidth inefficiency could indeed go unseen.

To draw a better judgment on the adaptively manner of RAFEC and to understand the impact of information-value of a packet on the gained reliability ratio, we plot the graphs of Figure 5.12 on the basis of Equation (5.19). Each graph in Figure 5.12 belongs to one specific packet error rate (PER) under which a packet that may carry different amount of information is transmitted. One can see that IRR of RAFEC heavily depends on the information-value of the packet. The more informative packet (higher  $\gamma$ ), the less likely the packet will be lost and so the higher contribution in the overall RR. In Figure 5.12 the relationship among the reliability ratio of different information-values in RAFEC is:

$$IRR^{RAFEC}(\gamma = 3) \geq IRR^{RAFEC}(\gamma = 2) \geq IRR^{RAFEC}(\gamma = 1)$$

Following this intuition, the IRR of the most informative packets in RAFEC are always maximum and greater than 90%. Moreover, since packets with  $\gamma = 1$  are less important for the application, the RAFEC does not use robust error control for them and thereby the IRR for them is relatively low. According to Figure 5.12, no fixed relationship among reliability ratio of different information-values for other approaches can be inferred and they just exhibit a very random behavior.

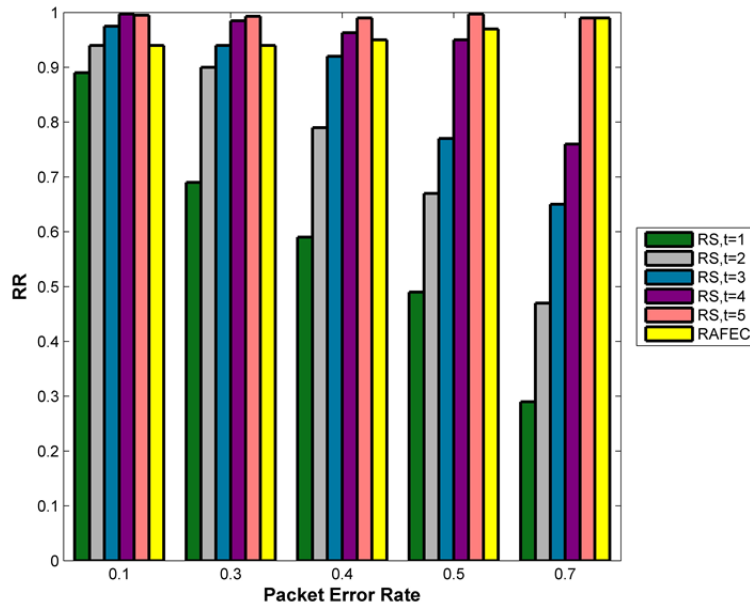


Figure 5.11. Reliability ratio comparison between RAFEC and RSs

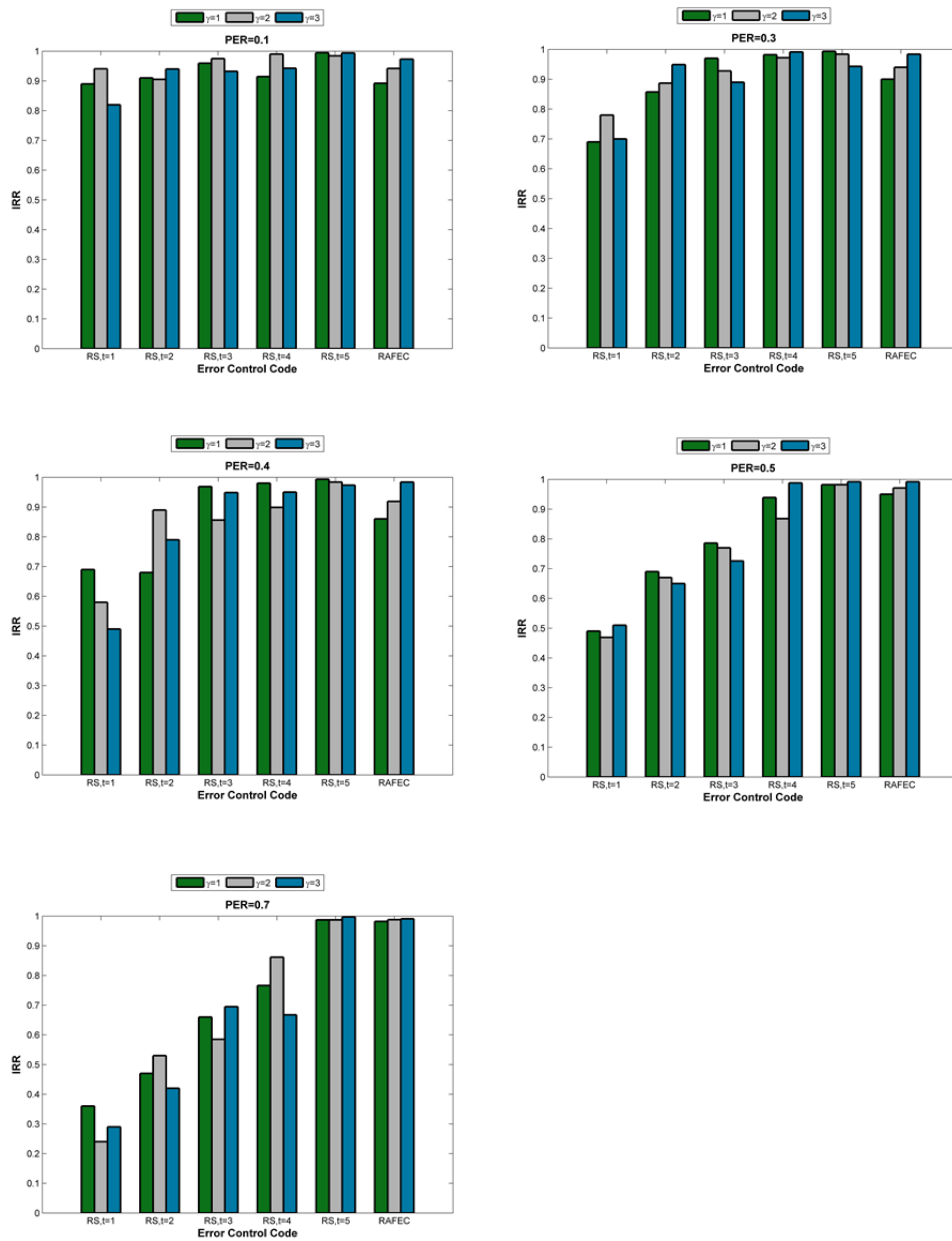


Figure 5.12. Information-aware reliability ratio for different packet error rate for RAFEC and RSs

## 5.6. Performance evaluation of RAFEC

---

The average gained code rate for the received packets which carry different informative content is illustrated in Figure 5.13. The code rate of RAFEC is inversely proportional to the packet error rate as RAFEC needs to dynamically adjust the amount of parity packets to be able to overcome the incurred errors. Generally, the high error rate necessitates the use of more parity packets, which in turn results in a lower code rate. Since other approaches are all static, changing the error rate does not have any effect on the code rate.

The code rates shown in Figure 5.13 are averaged over three information-values. To have a better insight about the obtained code rate per different information-value, Figure 5.14 is presented. The higher packet error rate necessitates to equip data-words with a more powerful code, which results in more parity packets and thereby lower code rate. Obviously, packet with  $\gamma = 3$  produces low code rate which explains its superior performance in terms of reliability.

According to both Figure 5.13 and Figure 5.14, we could generally deduce that the effective error rates that are calculated in Equation (5.17) to be considered for three information-values, correspond to three successive channel states. Therefore, generally speaking, most of transitions among the channel states occur in three successive states. The first state, second state and the third state respectively exhibit the effective error rate for the packet with information-value  $\gamma = 1, \gamma = 2$  and  $\gamma = 3$ .

It is generally believed that the additional parity packets (or lowering the code rate) can be tolerated as long as the loss resiliency at the receiver side is enhanced. Figure 5.15 illustrates the system efficiency which rather reveals how optimal the codes can counteract the channel error. One can see that even though RS(15,5) always provides the highest reliability rate as shown in Figure 5.11, using that code is not always accepted and even energy efficient because of its high redundancy as shown in Figure 5.13. As it can be seen from Figure 5.11, Figure 5.15 and Figure 5.13, when the packet error rate is 0.1, the reliability ratio of all approaches are more or less the same and above 0.85 while the corresponding efficiency ratio and the code rates are quite different. In this situation, RS(15,5) which introduces a high code rate and quite low efficiency ratio will never be an efficient choice. As it can be seen from Figure 5.15, RAFEC presents higher system efficiency than some other codes, however, in some points its performance is equal or less than some other codes. Basically, Figure 5.15 presents the system efficiency which is averaged for three information-value and so the amount of information a received packet has not been considered in the system efficiency. However, it would be more insightful to calculate Information-aware System Efficiency using Equation (5.23). The Figure 5.16

illustrates Information-aware System Efficiency of different codes, from which superiority of RAFEC over all other codes can be seen.

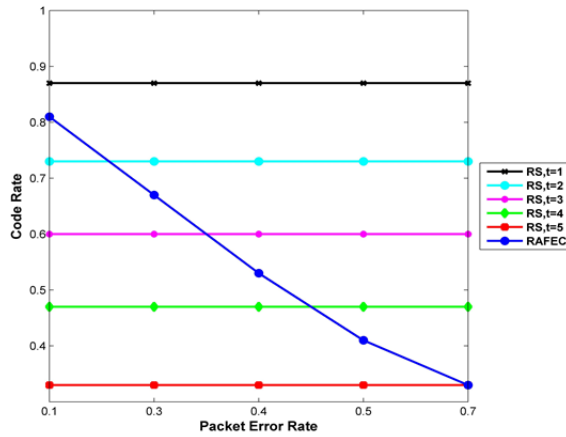


Figure 5.13. Code rate comparison of RAFEC and RSs

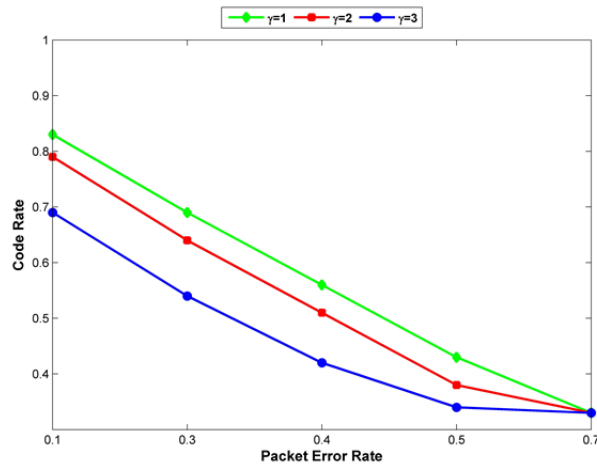


Figure 5.14. Code rate comparison of RAFEC

In the last experiment, we provide a comparison among three approaches, i.e., (i) RAFEC, which is channel- and information-aware, (ii) link-aware: technique proposed in [15] which only addresses the channel condition, and (iii) Information-aware: technique proposed in [28] which only addresses information-value of the packets regardless of the channel state. The assigned codes for the information-values  $\gamma = 1$ ,  $\gamma = 2$  and  $\gamma = 3$  in information-aware approach are RS(15,13), RS(15,11) and RS(15,9), respectively.

## 5.6. Performance evaluation of RAFEC

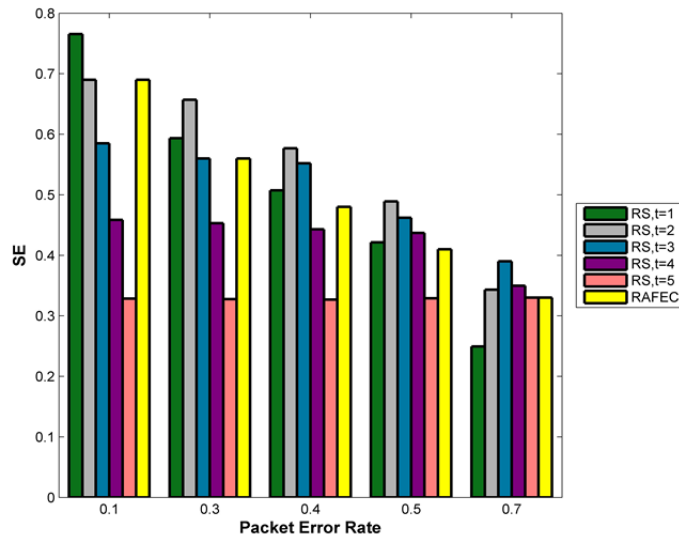


Figure 5.15. System efficiency comparison of RAFEC and RSs

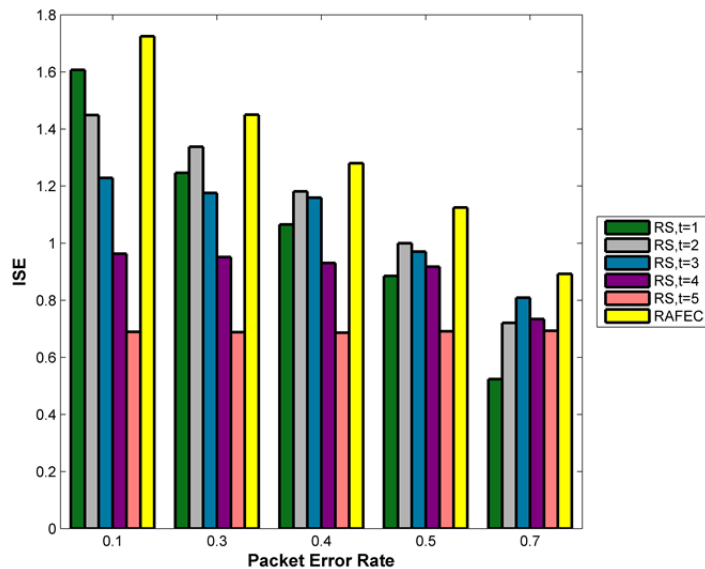


Figure 5.16. Information-aware system efficiency comparison of RAFEC and RSs

As it can be seen from Figure 5.17, link-aware approach provides lower reliability than RAFEC as it always uses the same code for all three information-value  $\gamma = 1$ ,  $\gamma = 2$  and  $\gamma = 3$ . Reliability of information-aware approach also sharply decreases when packet error rate is 0.1 compared with when it is 0.7. This is due to the fact that this approach does not



take the channel condition into consideration and employs the same code for a specific information-value over all the channel states. By doing so, there will be either unnecessary overheads or insufficient redundancy which results in a low reliability.

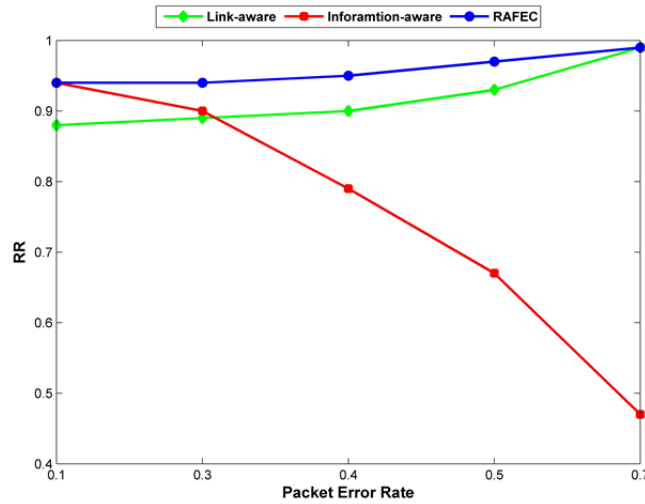


Figure 5.17. RR of link-aware, information-aware, and RAFEC protocols

### 5.7. Chapter summary

The purpose of wireless sensor networks is sensing and disseminating information. Therefore, the loss of important information at the perceived benefit of saving energy, may inhibit the ability of a wireless sensor network to fulfill its primary purpose. In this chapter, we propose RAFEC a packet-level reliable data dissemination protocol to support information-awareness in a chain-based wireless sensor network. Using RAFEC, information can be delivered at desired levels of reliability at proportional cost, in spite of the presence of long-term fading in the channel. RAFEC, basically exploits the concept of dynamic packet state and dynamic link state to control the correction capability of the error control codes exploiting only local knowledge of channel and packets at each hop. Moreover, the history-based evaluating link quality which RAFEC utilizes, provides a means to cope with longer-term interferences, since the mechanism does not immediately switch to a less/more powerful code after one successful/failed transmission. Basically, RAFEC waits until a couple of transmission have succeeded or failed and then change the error control in-use. In the simulation, we illustrate the superiority of RAFEC in terms of several metrics. The gain of RAFEC with respect to other approaches is up to 50% increasing the code rate and up to 150% improvement in ISE when packet error rate is small. The simulation results also reveal

## 5.8. Bibliography

---

that RA-FEC outperforms both information-aware and link-aware approaches in terms of attained reliability.

### 5.8. Bibliography

- [1]. Kuklová, Z., *Coding theory, cryptography and cryptographic protocols—exercises with solutions*. Thesis, Masaryk University, 2007.
- [2]. Hamming, R.W., *Error detecting and error correcting codes*. Bell System technical journal, 1950. 29(2): p. 147-160.
- [3]. Peterson, W., *Error-correcting codes* 1961: Cambridge, MA: MIT.
- [4]. Razavi, R., M. Fleury, and M. Ghanbari, *Adaptive packet-level interleaved FEC for wireless priority-encoded video streaming*. Advances in Multimedia, 2009. 2009: p. 3.
- [5]. Cayirci, E., et al., *Wireless sensor networks for underwater surveillance systems*. Ad hoc networks, 2006. 4(4): p. 431-446.
- [6]. Reed, I.S. and G. Solomon, *Polynomial codes over certain finite fields*. Journal of the Society for Industrial & Applied Mathematics, 1960. 8(2): p. 300-304.
- [7]. Yang, Y., *Information Theory, Inference, and Learning Algorithms*. Journal of the American Statistical Association, 2005. 100(472): p. 1461-1462.
- [8]. Sklar, B., *A primer on turbo code concepts*. Communications Magazine, IEEE, 1997. 35(12): p. 94-102.
- [9]. Massey, J.L. and J.K. Omura, *Computational method and apparatus for finite field arithmetic*, 1986, Google Patents.
- [10]. Khan, M.A., S. Afzal, and R. Manzoor. *Hardware implementation of shortened (48, 38) Reed Solomon forward error correcting code*. in *7th International Conference on Multi Topic* 2003.
- [11]. Comroe, R. and D.J. Costello Jr, *ARQ schemes for data transmission in mobile radio systems*. Journal on Selected Areas in Communications, 1984. 2(4): p. 472-481.
- [12]. Liankuan, Z., et al., *Adaptive error control in wireless sensor networks*, in *IET International Conference on Wireless Sensor Network* 2010.
- [13]. Eriksson, O., et al. *On hybrid ARQ adaptive forward error correction in wireless sensor networks*. in *37th Annual Conference on Industrial Electronics Society*. 2011.
- [14]. Yu, K., et al. *Adaptive forward error correction for best effort Wireless Sensor Networks*. in *International Conference on Communications* 2012.
- [15]. Ahn, J.-S., S.-W. Hong, and J. Heidemann, *An adaptive FEC code control algorithm for mobile wireless sensor networks*. Journal of Communications and Networks, 2005. 7(4): p. 489-498.

- [16]. Hurni, P. and T. Braun, *Link-quality aware run-time adaptive forward error correction strategies in wireless sensor networks*. IAM, University of Bern, IAM-11-003, Tech. Rep, 2011.
- [17]. Charfi, Y., N. Wakamiya, and M. Murata. *Adaptive and reliable multi-path transmission in wireless sensor networks using forward error correction and feedback*. in *Wireless Communications and Networking Conference*. 2007.
- [18]. Yan-ming, C., et al. *An adaptive fault-tolerant scheme for wireless sensor networks*. in *International Conference on Communications and Mobile Computing*. 2009.
- [19]. Holland, G., N. Vaidya, and P. Bahl. *A rate-adaptive MAC protocol for multi-hop wireless networks*. in *7th annual international conference on Mobile computing and networking*. 2001.
- [20]. Lettieri, P. and M.B. Srivastava. *Adaptive frame length control for improving wireless link throughput, range, and energy efficiency*. in *17th Annual Joint Conference of the IEEE Computer and Communications Societies*. 1998.
- [21]. Wu, G., et al. *WINMAC: A novel transmission protocol for Infostations*. in *49th Vehicular Technology Conference*. 1999.
- [22]. Bhatnagar, S., B. Deb, and B. Nath. *Service differentiation in sensor networks*. in *International Conference on Wireless Personal Multimedia Communications*. 2001.
- [23]. Deb, B., S. Bhatnagar, and B. Nath. *ReInForM: Reliable information forwarding using multiple paths in sensor networks*. in *28th Annual IEEE International Conference on Local Computer Networks*. 2003.
- [24]. Karl, H., M. Löbbers, and T. Nieberg. *A data aggregation framework for wireless sensor networks*. in *Dutch Technology Foundation ProRISC Workshop on Circuits, Systems and Signal Processing*. 2003. Citeseer.
- [25]. Kleinschmidt, J.H., W.C. Borelli, and M.E. Pellenz, *An energy efficiency model for adaptive and custom error control schemes in Bluetooth sensor networks*. *AEU-International Journal of Electronics and Communications*, 2009. 63(3): p. 188-199.
- [26]. Kopke, A., H. Karl, and M. Lobbers. *Using energy where it counts: Protecting important messages in the link layer*. in *2nd European Workshop on Wireless Sensor Networks*. 2005.
- [27]. Deb, B., S. Bhatnagar, and B. Nath. *Information assurance in sensor networks*. in *2nd ACM international conference on Wireless sensor networks and applications*. 2003.
- [28]. Kleinschmidt, J.H. and W. da Cunha Borelli. *Adaptive error control using ARQ and BCH codes in sensor networks using coverage area information*. in *20th International Symposium on Personal, Indoor and Mobile Radio Communications*. 2009.
- [29]. Banerjee, S. and A. Misra. *Minimum energy paths for reliable communication in multi-hop wireless networks*. in *3rd ACM international symposium on Mobile ad hoc networking & computing*. 2002.

## Conclusion

---

Wireless sensor networks are growing in popularity for various applications and a key factor for the proliferation of this revolutionary technology is designing effective protocols according to the (i) application requirements (ii) deployment properties and characteristics. In the wireless sensor networks domain, so far, little focus has been given to quality of service aware data dissemination protocols for low-power wireless communications over chain-based topology. On the other hand, there is increasing interest in using wireless sensor networks in monitoring and managing risks of many long-span structural/area which features a linear sensor arrangement and thus its topology resembles a chain. The network topology of these applications has special features, such as multi-hop, long delay, long distance and low reliability. To cope with these issues, the main research focus of this thesis is to study the solutions which can ensure a combination of four important quality of services, i.e. long lifetime, coverage, reliability and timeliness, for data traffic in a chain-based topology.

In Chapter 2, we described our contribution with respect to the coverage issue. In Chapter 3, we elaborated on our schemes whose aim is to reliably and fast aggregate and disseminate sensed data toward the base station while lifetime is prolonged. In Chapter 4 and 5 we focused on using error control schemes while respecting packet-level constraints, in order to disseminate data in a network suffering from short-term and long-term burst errors, respectively.

### 6.1. Contributions revisited

The main focus of this thesis was to study data collection and dissemination solutions for chain-based wireless sensor networks that fulfill the quality of service requirements of data traffic. In particular, the overall objective of the protocols designed in this thesis was to improve the performance and functionality of a chain-based wireless sensor network through the design of efficient data collection and dissemination algorithms. In this regard, the main research question of this thesis was:

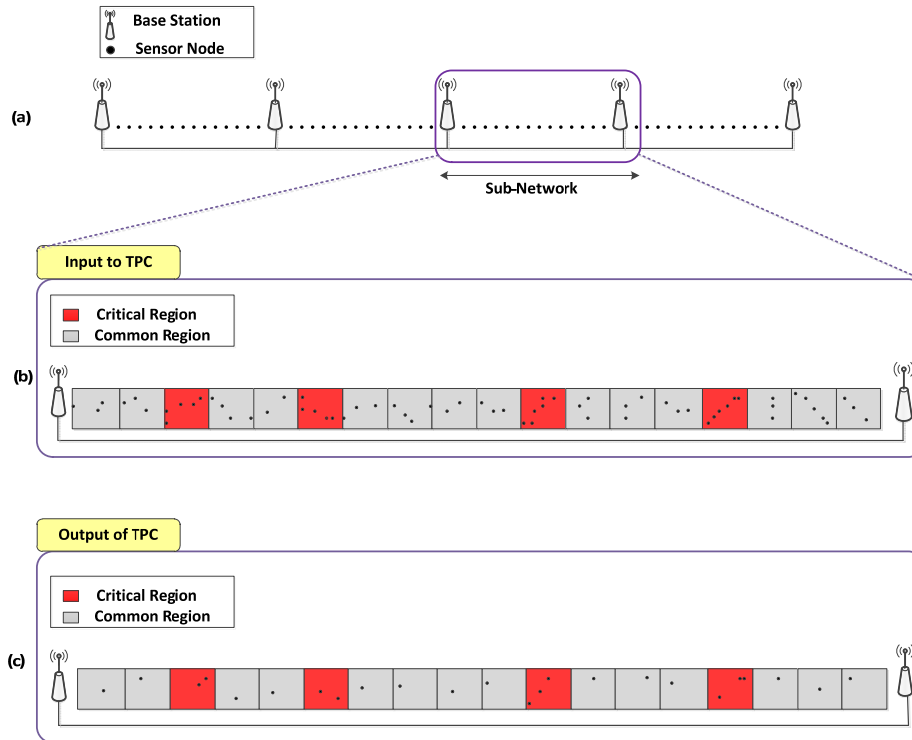
*How can long-lifetime, coverage, reliability and timeliness be ensured for disseminating different types of data traffic in a chain topology?*

We addressed this question (i) at topology level by Contribution 1, Contribution 2 and Contribution 3, and (ii) at error control level by Contribution 4 and Contribution 5. In both levels, we retained the advantages of the chain-based topology and at the same time mitigated some difficulties of the chain-based topology to achieve a quality aware data dissemination scheme. To have a better insight about the achievement, the contributions of this thesis are revisited in the following.

- 1. Trust-based probabilistic coverage:** In Chapter 2, we investigated and addressed the coverage problem to determine a schedule based on which the sensor nodes are kept active to efficiently cover the whole monitoring area, using a probabilistic coverage model. By efficient coverage of monitoring area we mean ensuring long network lifetime as well as maintaining sufficient sensing coverage and reliable sensing. Moreover, assuming a probabilistic coverage model we aimed to capture the real world sensing and transmitting characteristics of the nodes. In this regard, we proposed a trust-based probabilistic coverage algorithm, which leverages the trust concept to tackle the time-varying uncertainties introduced by the sensor nodes and the environment they operate in. To this end, first we explored and evaluated the time-varying uncertainty parameters which may affect the quality of sensory data. Thereafter, having both the explored uncertainty parameters and the application requirements, we formulated a situation-aware trust-based probabilistic coverage problem into an ILP. Specifically, our scheme was to find such active sensor nodes set using which (i) the quality of data gathered from each region meets the application requirement for the respective region and (ii) the network lifetime is maximized. Finally, we proposed a greedy heuristic scheme to approximate the optimal coverage set. It is worth recalling that most of previous coverage schemes give emphasize the energy consumptions of the sensor nodes in order to prolong network lifetime. However, we approached the coverage problem from a different perspective by targeting the confidence level of the sensor nodes, which may fundamentally influence the system performance in terms of reliability and energy efficiency. Basically, erroneous data generated by the sensor nodes must be protected from entering the network for effective bandwidth usage and energy utilization. To this end, we leverage the trust model in order to quantify the reliability and trustworthiness of each node. The trustworthiness of the nodes could be exploited as a promising means to decide including a node in the coverage set or not, according to the confidence level the given node exhibits. Moreover, we addressed variable-

## 6.1. Contributions revisited

length rounds for our proposed algorithms in order to cope with the network dynamism which is inevitable for many wireless sensor networks. The simulation results show the superiority of our proposed approaches in terms of energy efficiency and reliability for the critical regions which require to be monitored in a more reliable way, in a dynamic network. The input and output of our coverage algorithm is illustrated in Figure 6.1.



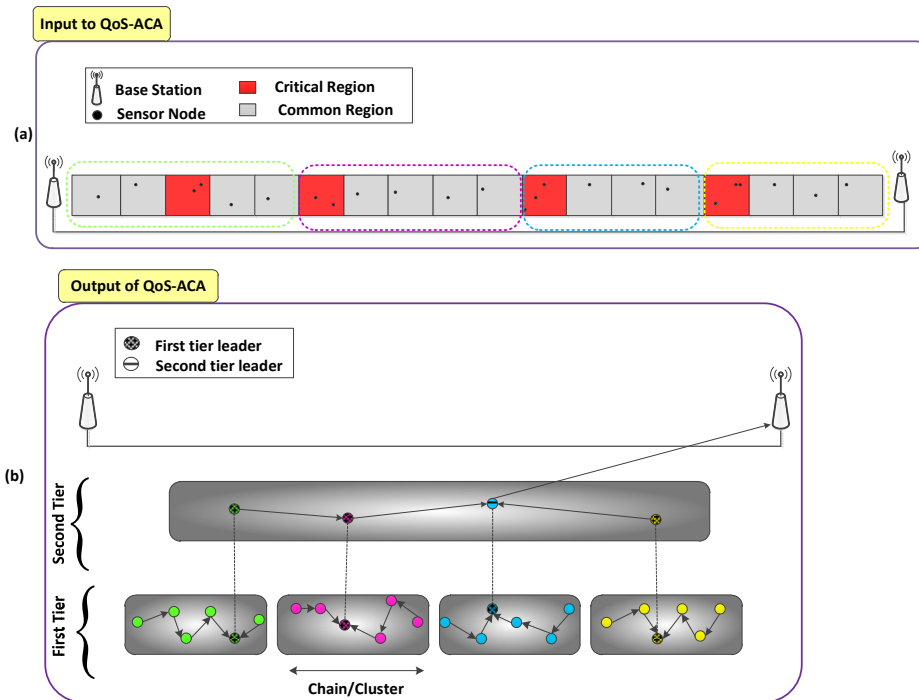
**Figure 6.1.** (a) A long-linear dense deployed wireless sensor network (b) Input to TPC: a sub-network (c) Output of TPC: a coverage set of sensor nodes

After finding the coverage set as indicated in Contribution 1, in Chapter 3 we targeted three quality of service parameters, i.e., (i) lifetime, (ii) reliability, (iii) delay or data freshness and built some chain-clusters among active sensor nodes subject to the application level quality of service constrains for data traffic, either in a static way through QoS-ACA (**Contribution 2**) or in a dynamic fashion through REC and REC+ (**Contribution 3**), which will be discussed in follows.

**2. QoS-aware Cluster-head/Chain-leader Selection in a Two-tier Architectural model:** The problem we deal for this contribution was to find a well-balanced quality

of service aware approach to deliver data packets collected by the sensor nodes (which are selected in previous contribution) to the base station, while respecting application requirements. In this regard, we proposed QoS-ACA a quality of service aware data dissemination scheme to deliver data packets collected by the sensor nodes to a base station. To this end first we introduced a two-tier architecture model in order to quality-aware deliver data packets collected by the sensor nodes to a base station. The first tier consists of some clusters inside each of which a chain of the sensor nodes is formed while the node who provides the highest utility in terms of required quality of services, is selected to undertake the chain-leader role. On the other hand, the second tier consists of a chain whose members are the leaders of the first tier. The criteria based on which the leader of the second-tier is selected is similar to that of the first-tier. Basically, we integrated three quality of service parameters (lifetime, reliability, and delay) with the possibility to adjust their priorities according to the specific application requirements in order to find the most proper nodes as the chain leaders in both tiers. Furthermore, dependent on the network density, reliability is ensured in two different ways for the sparsely and densely deployed sensor nodes. Moreover, in the interest of conserving both energy and bandwidth along with providing meaningful information to end-users, we utilized data aggregation on both chain leaders or cluster heads and intermediate nodes along the path toward the destination. In the simulation we considered three applications (i) which imposed different priority on the three mentioned quality of service parameters (ii) in which the sensor nodes could be deployed either densely or sparsely (iii) in which aggregation could be accomplished or ignored along the path toward base station. We compared these three applications with respect to several performance metrics whose importance-level or priority could be drawn from the application requirements. The flexibility of QoS-ACA allows application developers to assign different weights to the quality of services parameters depending on the requirements of their application. Basically, compared to the existing chain-cluster based data dissemination techniques our protocols combine these three namely quality of service parameters according to the application preferences in order to make a trade-off in ensuring them. The input of QoS-ACA is the output of coverage algorithm (TPC) when the sensor nodes are organized into some equal-size sets in terms of number of nodes (Figure 6.2. ). For example in the Figure 6.2. we have four sets each of which consists of six nodes. It is worth recalling that this rigid assumption regarding fixed-size clusters will be relaxed in the next contribution.

## 6.1. Contributions revisited



**Figure 6.2. (a) Input of QoS-ACA: some equal-size sets of sensor nodes (b) Output of QoS-ACA: a two-tier architectural model consists of some equal-size chain-clusters**

- 3. QoS-aware Dynamic Chain-Cluster Forming:** In order to relax some assumptions regarding communication capability of the sensor nodes to communicate directly with other nodes or with the base station as well as the fixed-size of the chain-clusters, which many data dissemination protocols rely on, in the second part of Chapter 3 we proposed REC and REC+ solutions. The proposed approaches make the size/shape of the clusters in QoS-ACA adaptive regarding the state of the nodes and links. In this way, the main concern of REC/REC+ was building chain-clusters and setting boundaries of the clusters in an adaptive and dynamic way subject to the application level quality of service constrains. The simulation results showed that REC+ outperforms a number of other approaches in terms of delay, delay $\times$ energy and lifetime thanks to well-dispersing the cluster heads throughout the network and efficient setting the clusters boundaries. Compared with existing approaches that reform clusters in each round, REC+ starts to change the cluster shapes when the energy goes below a threshold or end to end reliability changes significantly. To the



best of our knowledge, REC and REC+, are the first chain-clustering algorithms that well incorporate energy, delay and transmission reliability together to construct clusters and to select proper cluster heads in wireless sensor networks. In other words, by REC and REC+ we devise an adaptive cluster's shape/size adjustment according to the application requirements and nodes/links conditions. We showed that compare with other clustering approaches, REC and REC+ well-disperse the cluster heads throughout the network by using a quality of service aware algorithm to form the chain-clusters. As Figure 6.3. illustrates, the input of REC/REC+ is exactly the output of TCP without additional assumptions, which in this figure is a set of sensor nodes whose transmission range includes 8-hop nodes. According to this figure, (i) the output of REC is a two-tier architectural model consists of some variant size chain-clusters and (ii) the output of REC+ is a two-tier architectural model consists of some variant size chain-clusters which are better balanced compared with that of REC.

So far, our contributions were mostly on topological aspects or *node-level constraints*, and we assigned proper role and proper activity scheduling to each *node* based on the amount of utility the given node offers. In the next two contributions we mostly concentrated on the *packet-level constraints* while using error control protocols. In this way, after coping with node-level constraints and constructing a quality aware backbone for data dissemination through Contribution 1, Contribution 2 and Contribution 3, we gave special emphasize in two packet-level constraints, namely TTL (**Contribution 4**) and Information-value (**Contribution 5**), in order to guarantee the quality of service for data traffic taking each *individual* data packet requirements into account.

**4. Reliable Dissemination of Time-Constrained Data:** Meeting the TTL constraint of the sensory data which should reliably be transmitted toward the base station in a low duty-cycle network that suffers from short-term burst errors was the main focus of this contribution which was addressed in chapter 4. By short-term burst errors we mean the errors which are localized in short-term and occurs in burst forms. In this respect, we proposed READ a runtime adaptive packet-link-local error control protocol which operates based on the links' qualities, packets' TTL, and duty-cycle and is able to counteract periodic short-term burst-errors in a chain topology. The main idea of READ is fairly distributing the available TTL based on the links qualities among the sensor nodes who utilize the allocated times to enhance the transmission reliability. Furthermore, READ is able to work under different duty-cycles based on which will tune the fractional portion of the sensor nodes from the TTL. From another side, READ is an energy-efficient adaptive error-control scheme

which improves the reliability to such extent that does not influence the TTL constraint of the packet. In the interest of conserving both energy and bandwidth, READ drops the packets which are expected to be expired before reaching the base station. Different from most of the proposed reliable and real-time approaches that are proposed to work for the topologies other than chain and so cannot efficiently work for the chain topology, READ is customized for this poor-explored topology. In the simulation, we illustrated the relation between both TTL of the packet and link quality and the attained hit ratio for different approaches. From our results, we could conclude that READ is a promising approach specially when the packet loss rate is relatively high or when the TTL is not large enough to transmit the optimal number of packet copies.

- 5. Information-link-aware Data Dissemination:** In the same line of the previous contribution which addressed TTL as one of the packet-level indicator or constraints, in Chapter 5 we gave emphasize to another indicator, namely information-value, considering which the reliability performance of a dissemination protocol can considerably be enhanced. By information-value we meant the amount of information or importance a packet may have for the base station. To this end, in this chapter we (i) explored, quantified and integrated the factors that may influence the information-value of a packet and (ii) coped with this crucial design problem of choosing an appropriate error control code by adaptively selecting the codes for each *individual links*, which may experience long-term fading and for each *individual packet* at run-time instead of applying network-wide settings prior to network deployment. In this way, we proposed RAFEC protocol, which is a run-time adaptive FEC-based data dissemination protocol. In RAFEC each node decides which error control code to use abiding to the computational constraints of embedded sensors, the information-value of the packet, and the statistical properties of the observed errors for the upward link. This adaptation gives the possibility to vary the code strength and complexity on-demand and on the fly. Using RAFEC, information can be delivered at desired levels of reliability at proportional cost, in spite of the presence of long-term fading in the channel. RAFEC, basically exploits the concept of dynamic packet state and dynamic link state to control the correction capability of the error control codes exploiting only local knowledge of channel and packets at each hop. The history-based evaluating link quality which RAFEC utilizes, provides a means to cope with longer-term interferences, since the mechanism does not immediately switch to a less/more powerful code after one successful/failed transmission. Basically, RAFEC waits until a couple of transmission have succeeded

or failed and then changes the error control in-use. In the simulation, we illustrated the superiority of RAFEC in terms of several metrics.

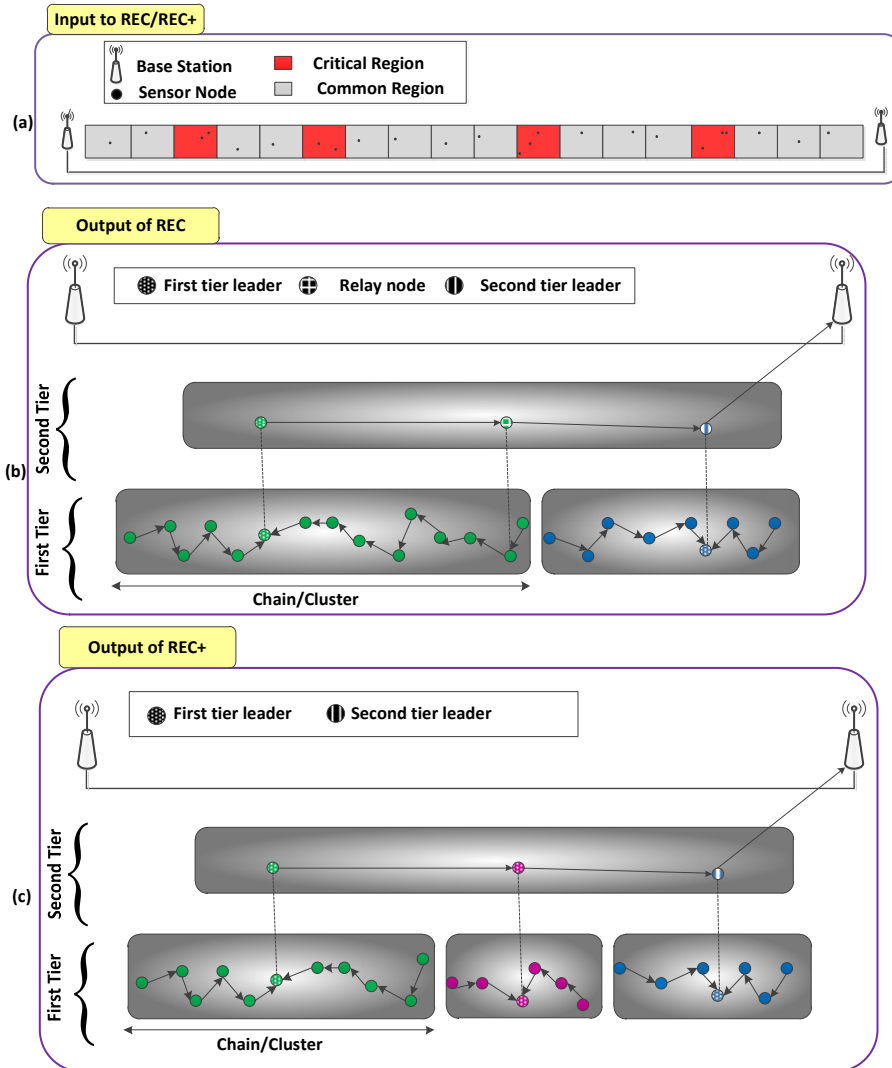


Figure 6.3. (a) Input to REC/REC+ (b) Output of REC (c) Output of REC+

## 6.2. Conclusion and lessons learned

The important lessons that we have learned during this research can be summarized as follows:

## 6.2. Conclusion and lessons learned

---

1. The trustworthiness of the sensor nodes can have great impact on the coverage functionality and network performance. Considering the trustworthiness of the nodes in the coverage decision process in order to include a node in the coverage set or not, can prevent the erroneous data generated by the sensor nodes from entering the network for effective bandwidth usage and energy utilization beside increasing data reliability. A trust-based coverage algorithm is more promising for the region whose reliability requirement is high and thus requires to be monitored by more than one sensor node.
2. Reliable data dissemination can still be achieved to some extent without using error control schemes. Reliable data dissemination is traditionally guaranteed by applying error control approaches, which could provide an adequate degree of quality even in the presence of errors. However in a chain topology, the reliability can be ensured to some extent by selecting the most qualified node which could bring about the sufficient end-to-end transmission reliability for the chain members, as the chain leader. In other words, the quality of data dissemination task in a chain-based topology can significantly be affected by the relative position of the chain-leader(s) with regards to other nodes in the chain(s). Basically, in a chain-based topology, in which sensor nodes that are not leader can only communicate with their adjacent left and right neighbors, routing is not very complicated. On the other hand, in a chain topology, chain leader should usually handle a huge amount of traffic received from two sides of the chain. Therefore, the location of the leader in the chain and thereby the situation of the nearby area of the leader is very important for the quality of the received traffic. Taking these issues in account, the chain leader election algorithm is the most determinative factor in ensuring quality of service for a chain topology.
3. Integrating the quality of service parameters with the possibility to adjust their relative priorities (through some weights) in selecting the chain leader, gives more flexibility to the application developers to set the weights' values with respect to the specific application requirements. Therefore, the chains leaders would be well-dispersed throughout the network according to the assigned weights.
4. Quality-aware topology control alone is not sufficient to ensure quality of services for disseminating data of many applications whose packets may convey different types or amount of information. For example, end-to-end transmission reliability cannot efficiently be guaranteed without taking the (sensory) data constraints (importance) into account. Therefore, ensuring quality of service for data traffic taking each *individual* data packet requirements (e.g. TTL or Information-value) into account is highly promising. On the other hand, the information-value or importance

level of a packet could be influenced by several factors which may have different priorities in different applications. Therefore, according to the application requirements these factors may differently incorporate to estimate the information-value of the packets. The relative priority of each factor can be rated by assigning a weight. The information-value of a packet is a promising means based on which an effective error control code could be assigned to a packet. Dynamic error control schemes which are allocating the correctional power in an on-demand manner based on both the information-value and channel state are viable alternatives to static error control schemes, where the link conditions or packets' information-values are not taken into account. In this way and for the sake of efficiency, the amount of information a packet carries and the state in which the channel is in should be put into perspective with the amount of effort (in terms of energy expenditure) that is required to reliably transmit the given packets.

5. The delay arisen from acknowledgement in ARQ protocol may not always be accepted for the time-constrained data specially in a low duty-cycle and rather unreliable network. To eliminate the delay and transmission overheads introduced by acknowledgements in ARQ, the reliability can be ensured via advance transmission of packet copies by the sender without sending any acknowledgement. In a sense, this can be viewed as a rudimentary form of FEC using so-called Repetition codes. Therefore, a repetition-based code whose parameter is accurately calculated according to the available TTL, links qualities and duty-cycle can be an effective alternative specially if packet loss rate is relatively high or TTL is relatively short.

In conclusion, this thesis has contributed to the existing data dissemination literatures by providing new insights into the process of quality-aware collecting and disseminating data in a poorly-explored chain-based topology which is of interest for many applications especially those with linear deployment. The solutions presented throughout the chapters have the potential to quality-aware collect and disseminate data related to a wide range of WSN applications whose topology resembles a chain such as monitoring operational performance and health of bridge, tunnel, highway, railway and pipeline, thereby assisting better management and increasing safety, efficiency and comfort to the users. One should note that the proposed solutions require no modification in underlying protocol standards.

### **6.3. Future works**

In this section, we provide a list of potential future research directions that are relevant to the topics covered in this thesis.

### 6.3. Future works

---

- *Variable sensing radii*: In our proposed coverage scheme, we assumed a fixed sensing-range for the sensor nodes. However, we can consider a more general model in which each sensor node is able to adjust its sensing range which can be utilized as another parameter in the optimization function. Basically, for larger sensing range, more energy is needed for noise filtering and signal processing in order to improve signal-to-noise ratio [1]. Giving flexibility to the sensing radii, the sensing energy can be optimized by reducing the redundant coverage as long as the desired coverage quality is maintained. The sensing energy consumption becomes more critical for energy-efficient design of coverage protocols since surveillance or monitoring often requires a continuous coverage while data dissemination is only periodic or event-driven.
- *Distributed chain-cluster forming and updating*: Distributed processing and decision making is becoming very popular in wireless sensor networks, especially in case of (i) unavailability of a powerful central point which requires global information of the network or the chain-clusters to well-control them, (ii) requiring an efficient real-time decision making in a large-scale linear network with several hops which may impose long latency. Moreover, in a centralized approach the required bandwidth may increase significantly with the number of nodes which should report any significant change in their (link/energy) qualities to the base station. In this regard, making our proposed centralized chain-clustering schemes in Chapter 3 more autonomous and distributed is another direction of our future works. In the distributed chain-cluster forming/updating approach, a sensor node is able to become a cluster-head/chain-leader or to join/leave a formed cluster on its own initiative without global information of the network or chain-clusters. This way of controlling chain-clusters makes the approach (i) more robust and thus the whole system still goes on smoothly even when a single sensor joins/leaves the networks (ii) scalable, by scalable we mean that each sensor only needs its neighboring node information.
- *Real-time RAFEC*: In our RAFEC algorithm which is discussed in Chapter 5, a Reed-Solomon code is selected based on the burst-error nature in collected sensory readings' traces and does not care about timing constraint of data. Therefore, it would be interesting to enhance RAFEC to provide reliability for time-constraint data as well, by consider timing constraints as another parameter in code selection. In this regard, a particular Reed-Solomon code will be selected based on two criteria: burst-error nature in collected traces and timing constraints of upcoming data.

#### 6.4. Bibliography

- [1]. Wang, B., *Coverage control in sensor networks*, 2010: Springer.

---

## About The Author

---

Zahra Taghikhaki received her B.Sc. degree in Computer Engineering from Ferdowsi University of Mashhad (FUM), Iran, in 2004. She obtained her M.Sc. degree in Software Engineering with emphasis on wireless sensor networks from Iran University of Science and Technology (IUST), in 2008. In summer 2010, she joined the Pervasive Systems group at the University of Twente to pursue her Ph.D. degree in the field of data dissemination in chain-based wireless sensor networks. During this time, she worked on the GENESI research project.

List of publications in which she participated is as follows:

- (1) Taghikhaki, Z. and Meratnia, N. and Havinga, P.J.M. (2015) *Protecting Informative Messages over Burst Error Channels in Chain-based Wireless Sensor Networks*. The Forth International Conference on Sensor Networks (SensorNets), 11-13 February 2015, Angers, France.
- (2) Taghikhaki, Z. and Meratnia, N. and Havinga, P.J.M. (2015) *An Information-link-aware Data Dissemination Scheme for Chain-based Wireless Sensor Networks*. The Sensors Journal (under review).
- (3) Taghikhaki, Z. and Meratnia, N. and Havinga, P.J.M. (2013) *A Reliable and Energy-efficient Chain-cluster Based Routing Protocol for Wireless Sensor Networks*. In: Eighth IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing, IEEE ISSNIP 2013, 2-5 April 2013, Melbourne, Australia. pp. 248-253. IEEE Communications Society.
- (4) Taghikhaki, Z. and Meratnia, N. and Havinga, P.J.M. (2013) *On QoS guarantees of error control schemes for data dissemination in a chain-based wireless sensor networks*. Sensors & Transducers Journal, 18. pp. 188-202. ISSN 2306-8515
- (5) Taghikhaki, Z. and Meratnia, N. and Havinga, P.J.M. (2013) *A trust-based probabilistic coverage algorithm for wireless sensor networks*. In: Proceedings of the The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks, EUSPN 2013, 21-24 Oct 2013, Niagara Falls, Ontario,




- Canada. pp. 455-464. *Procedia Computer Science* 21. Elsevier. ISSN 1877-0509
- (6) Bahrepour, M. and Meratnia, N. and Poel, M. and Taghikhaki, Z. and Havinga, P.J.M. (2012) *Use of wireless sensor networks for distributed event detection in disaster management applications*. *International Journal of Space-Based and Situated Computing*, 2 (1). pp. 58-69. ISSN 2044-4893
  - (7) Taghikhaki, Z. and Meratnia, N. and Havinga, P.J.M. (2012) *A reliable and real-time aggregation aware data dissemination in a chain-based wireless sensor network*. In: *Sixth International Conference on Sensor Technologies and Applications, SENSORCOMM 2012, 19-24 Aug 2012, Rome, Italy*. pp. 260-269. The International Academy, Research and Industry Association (IARIA).
  - (8) Taghikhaki, Z. and Meratnia, N. and Havinga, P.J.M. (2012) *An Error Control Scheme for Delay Constrained Data Communication in a Chain-Based Wireless Sensor Network*. In: *Seventh IEEE International Conference on Broadband and Wireless Computing, Communication and Applications, BWCCA-2012, 12-14 Nov 2012, Victoria, Canada*. pp. 385-390. IEEE Communications Society.
  - (9) Taghikhaki, Z. and Meratnia, N. and Zhang, Yang and Havinga, P.J.M. (2012) *QoS-Aware Chain-based Data Aggregation in Cooperating Vehicular Communication Networks and Wireless Sensor Networks*. In: *Roadside Networks for Vehicular Communications: Architectures, Applications, and Test Fields*. IGI Global, Hershey, PA, USA, pp. 169-190.
  - (10) Bahrepour, M. and Meratnia, N. and Taghikhaki, Z. and Havinga, P.J.M. (2011) *Sensor Fusion-based Activity Recognition for Parkinson Patients*. In: *Sensor Fusion Foundation and Applications*. InTech, pp. 171-190.
  - (11) Masoum, A. and Meratnia, N. and Dilo, A. and Taghikhaki, Z. and Havinga, P.J.M. (2011) *Cross-layer analyses of QoS parameters in wireless sensor networks*. In: *Proceedings of the First International Conference on Computer Science and Information Technology, CCSIT, 2-4 Jan 2011, Bangalore, India*. pp. 595-605. *Communications in Computer and Information Science* 132. Springer Verlag. ISSN 1865-0929
  - (12) Taghikhaki, Z. and Meratnia, N. and Havinga, P.J.M. (2011) *Energy-efficient Trust-based Aggregation in Wireless Sensor Networks*. In: *Proceedings of the 3rd International Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN 2011), in conjunction with IEEE InfoCom, 10 April 2011, Shanghai, China*. pp. 584-589. IEEE Communications Society.

## About The Author

---

- (13) Bahrepour, M. and Meratnia, N. and Poel, M. and Taghikhaki, Z. and Havinga, P.J.M. (2010) *Distributed Event Detection in Wireless Sensor Networks for Disaster Management*. In: International Conference on Intelligent Networking and Collaborative Systems, INCoS 2010, 24-26 Nov 2010, Thessaloniki, Greece. pp. 507-512. IEEE Computer Society.
- (14) Masoum, A. and Meratnia, N. and Taghikhaki, Z. and Havinga, P.J.M. (2010) *Reward and Punishment based Cooperative Adaptive Sampling in Wireless Sensor Networks*. In: Proceedings of the 2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 7-10 December, Brisbane, Australia. pp. 145-150. IEEE Computer Society.



Recently wireless sensor network has emerged as a promising technology that could induce an innovation wave in the field of (infra)structures monitoring because of its fast deployment, little interference with the surrounding, self-organization, flexibility and scalability. A key factor for the proliferation of this revolutionary technology is designing effective protocols to meet the quality of service requirements of the application considering deployment properties and characteristics.



**ISBN: : 978-90-365-3829-9**