PULSE SWITCHING: AN ULTRA-LIGHT NETWORK PARADIGM WITHOUT PACKET ABSTRACTION

By

Qiong Huo

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering–Doctor of Philosophy

2013

ABSTRACT

PULSE SWITCHING: AN ULTRA-LIGHT NETWORK PARADIGM WITHOUT PACKET ABSTRACTION

By

Qiong Huo

In this dissertation we develop a novel pulse switching framework for ultra light-weight networking applications involving severely resource-constrained embedded devices. The key idea is to abstract a single pulse, as opposed to multi-bit packets, as the network switching granularity. Pulse switching can be shown to be sufficient for on-off style event monitoring applications for which a monitored parameter can be modeled using just a binary variable. Monitoring such events with conventional packet paradigm can be prohibitively inefficient due to the communication, processing, and buffering overheads of the large number of bits within packet data, header, and preambles for synchronization. In the proposed paradigm, an event can be coded as a single pulse, which is then transported multi-hop while preserving sufficient information so that a destination (i.e., sink or actuator nodes) can derive certain spatio-temporal context information about the event in question. The resulting operational lightness, leveraged via zero buffering, no addressing, no packet processing, and an ultra-low communication energy budget makes the framework applicable for embedded devices such as Radio Frequency Identifiers operating with ultra-tight energy budgets, such as from harvested energy.

Specific contributions includes: 1) Developing a joint MAC-Routing abstraction for pulse switching, 2) Mapping the proposed pulse switching architecture on Ultra Wideband Band (UWB) impulse radio technology, 3) Developing a hop-angular spatial context abstraction and

the related estimation algorithms for event localization, 4) Designing a cellular alternative to the hop-angular abstraction, 5) Designing a peer-to-peer pulse routing protocol, finally 6) System characterization through modeling and large scale simulation.

To My Parents
For All Their
Love and Support

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Subir Biswas for his support and instruction through my whole Ph.D. career. His passion and rigor about research guided me in the past four years. I would also like to thank my committee Dr. Sandeep Kulkarni, Dr. Edward Rothwell, and Dr. Jian Ren for their time and support.

I would like to give many thanks to Bo Dong for all of the brainstorming and collaboration. I would also like to thank Anthony Plummer Jr. for his collaboration and support in my first conference paper during my Ph.D. career. Additionally, I would like to thank Mahmoud Taghizadeh for his support in implementation discussions. Furthermore, thanks must be given to my other lab mates Debasmit Banerjee, Dezhi Feng, Faeze Hajiaghajani, Stephan Lorenz, Muhannad Quwaider, Jayanthi Rao, Yan Shi, William Tomlinson, Clifton Watson, and Fan Yu.

Last but not least, I would like to give my thanks to my parents and my friends for their support and encouragements during my Ph.D. period.

TABLE OF CONTENTS

LIST OF TABLES xiii

LIST OF FIGURES xiv

LIST OF ALGORITHMS xvii

CHAPTER 1:INTRODUCTION 1
1.1 MOTIVATION 1
1.1.1 Wireless Sensor Networks 1
1.1.2 Wireless Sensor and Actuator Networks 1
1.2 OBJECTIVE 2
1.3 CHALLENGES 3
1.4 CONTRIBUTIONS 3
1.5 PROTOCOL REQUIREMENTS 4
1.6 DISSERTATION FLOWCHART 4

CHAPTER 2:RELATED WORK 8
2.1 INTRODUCTION 8
2.2 ENERGY-BASED TRANSMISSIONS 8
2.2.1 MAC Protocols 8
2.2.2 Routing Protocols 10
2.2.3 Joint MAC-Routing Protocols 11
2.3 NON-ENERGY-BASED TRANSMISSIONS 12

CHAPTER 3:TECHNOLOGY FOR PULSE REALIZATION 15
3.1 INTRODUCTION 15
3.2 ULTRA WIDE BAND IMPULSE RADIO TECHNOLOGY 15
3.2.1 Ultra Wide Band Impulse Radio 15
3.2.2 Ultra Wide Band Impulse Radio Slotting 16
3.2.3 Modulation and Synchronization 16
3.2.4 Energy Budget 17
3.2.5 Sources of Errors 17
3.3 SUMMARY AND CONCLUSION 18

CHAPTER 4:HOP-ANGULAR PULSE SWITCHING PROTOCOL 19
4.1 INTRODUCTION 19
4.2 NETWORK MODEL 19

4.2.1 Network Topology 19
4.2.2 Pulse as Protocol Data Unit 20
4.3 HOP-ANGULAR PULSE SWITCHING ARCHITECTURE 23
4.3.1 Joint MAC-Routing Frame Structure 23
4.3.2 Hop-Distance Self Discovery 24
4.3.3 Pulse Forwarding using Wave-front Routing 25
4.3.4 Exploiting Route Diversity 27
4.3.5 Pulse Merging 28
4.4 MITIGATING IMPACTS OF PHYSICAL LAYER COOPERATION 23
4.4.1 Motivation 29
4.4.2 Mitigating Cooperation during Hop-distance Discovery 29
4.4.3 Immunity to Node Cooperation during Pulse Forwarding 31
4.4 SUMMARY AND CONCLUSION 31

CHAPTER 5:CELLULAR PULSE SWITCHING PROTOCOL 33
5.1 INTRODUCTION 33
5.2 NETWORK MODEL 33
5.2.1 Network Topology 33
5.2.2 Pulse as Protocol Data Unit 34
5.2.3 Cellular Event Localization 35
5.3 CELLULAR PULSE SWITCHING ARCHITECTURE 35
5.3.1 Joint MAC-Routing Frame Structure 35
5.3.1.1 Frame Components 35
5.3.1.2 Accommodation for propagation delay 37
5.3.2 Global Frame Synchronization 37
5.3.3 Route Discovery 38
5.3.3.1 Route Discovery Logic 38
5.3.3.2 Routing Table Formation 39
5.3.4 Pulse Forwarding 40
5.3.4.1 Pulses in Localization Area 40
5.3.4.2 Pulse Forwarding Logic 41
5.4 FAULT-TOLERANCE 42
5.4.1 Response Mechanism 42
5.4.2 Prioritized Routing Table Lookup 43
5.4.3 Routing Loops 44
5.4.3.1 Example Scenario 44
5.4.3.2 Loop Resolution 45
5.5 SUMMARY AND CONCLUSION 47

CHAPTER 6:POINT-TO-POINT PULSE SWITCHING PROTOCOL 49
6.1 INTRODUCTION 49
6.2 NETWORK MODEL 49
6.2.1 Network Topology 49
6.2.2 Pulse as Protocol Data Unit 50

6.2.3 Cellular Event Localization 51
6.3 POINT-TO-POINT PULSE SWITCHING ARCHITECTURE 52
6.3.1 Joint MAC-routing Frame Structure 52
6.3.2 Route Discovery 54
6.3.2.1 Initiation and Termination 54
6.3.2.2 Discovery Area Details 54
6.3.2.3 Route Discovery Logic 55
6.3.2.4 Routing Table Formation 56
6.3.3 Pulse Forwarding 57
6.3.3.1 Pulses in Localization Area 57
6.3.3.2 Pulse Forwarding Logic 59
6.4 RELIABILITY ENHANCEMENT AND COLLISION HANDLING 61
6.4.1 Motivation 61
6.4.2 Response Mechanism 62
6.4.2.1 Response without Collision 63
6.4.2.2 Response with Collision 65
6.5 SUMMARY AND CONCLUSION 67

CHAPTER 7:ENERGY SAVING MEASURES 69
7.1 INTRODUCTION 69
7.2 PROTOCOL STATE MACHINE 69
7.2.1 Cellular Pulse Switching 69
7.2.2 Point-to-point Pulse Switching 70
7.3 TRANSMISSION ENERGY SAVING MEASURES 71
7.3.1 Parameterized Routing Control Method 71
7.3.1.1 Sector-Constrained Routing in Hop-angular Pulse Switching 71
7.3.1.2 Reducing Route Diversity in Cellular Pulse Switching and Point-to-point Pulse Switching 72
7.3.2 Pulse Compression Algorithm 73
7.3.2.1 Spatial Compression 73
7.3.2.1.1.Spatial Compression Logic 73
7.3.2.1.2.Spatial Compression Model 74
7.3.2.1.3.Implementation of Spatial Compression in Pulse Switching Protocols 74
7.3.2.2 Temporal Compression 76
7.3.2.2.1.Temporal Compression Logic 76
7.3.2.2.2.Temporal Compression Model 76
7.4 IDLING ENERGY SAVING MEASURES 79
7.4.1 Hop-angular Pulse Switching 79
7.4.1.1 Inter-frame Sleep 79
7.4.1.2 Intra-frame Sleep 79
7.4.1.3 Delay-traded Sleep 80
7.4.2 Cellular Pulse Switching 81
7.4.3 Point-to-point Pulse Switching 82
7.5 SUMMARY AND CONCLUSION 83

CHAPTER 8:ERROR ANALYSIS 84
8.1 INTRODUCTION 84
8.2 FALSE POSITIVE EVEN GENERATION 84
8.2.1 Hop-angular Pulse Switching 84
8.2.2 Cellular Pulse Switching 86
8.2.3 Point-to-point Pulse Switching 88
8.3 PROTECTION FROM FALSE POSITIVE PULSES 89
8.3.1 Hop-angular Pulse Switching 90
8.3.1.1 Single Pulse Code (SPC) Algorithm 91
8.3.1.2 Parity Pulse Code (PPC) Algorithm 92
8.3.1.3 Multi Pulse Code (MPC) Algorithm 93
8.3.1.4 Summary 93
8.3.2 Cellular Pulse Switching 94
8.3.3 Point-to-point Pulse Switching 95
8.4 IMMUNITY FROM PULSE LOSS 96
8.4.1 Hop-angular Pulse Switching 97
8.4.2 Cellular Pulse Switching and Point-to-point Pulse Switching 99
8.5 IMPACTS OF PROTECTION ON PULSE LOSS 101
8.5.1 Hop-angular Pulse Switching 101
8.5.2 Cellular Pulse Switching and Point-to-point Pulse Switching 102
8.6 SUMMARY AND CONCLUSION 103

CHAPTER 9:COMPARISON BETWEEN PULSE AND PACKET SWITCHING 105
9.1 INTRODUCTION 105
9.2 COMPARABLE PACKET BASED SOLUTION 107
9.3 HOP-ANGULAR PULSE SWITCHING 108
9.3.1 Event Reporting Delay 108
9.3.2 Power Consumption 110
9.3.2.1 Pulse Switching 110
9.3.2.2 Packet Switching 112
9.3.2.3 Numerical Model Results 112
9.3.2.3.1.Communication Power Consumption 112
9.3.2.3.2.Idling Power Consumption 114
9.3.2.3.3.Total Consumption 115
9.3.2.4 Simulation Results 116
9.4 CELLULAR PULSE SWITCHING 116
9.4.1 Event Reporting Delay 116
9.4.2 Power Consumption 118
9.4.2.1 Pulse Switching 118
9.4.2.2 Packet Switching 120
9.4.2.3 Numerical Results 121
9.4.2.3.1.Communication Power Consumption 121
9.4.2.3.2.Idling Power Consumption 122
9.4.2.3.3.Total Consumption 123
9.5 POINT-TO-POINT PULSE SWITCHING 124

9.5.1 Collision Modeling 124
9.5.1.1 Motivation 124
9.5.1.2 Model Description 124
9.5.1.3 Probabilistic Modeling 126
9.5.2 Event Reporting Delay 127
9.5.2.1 Pulse Switching 127
9.5.2.2 Packet Switching 128
9.5.3 Power Consumption 129
9.5.3.1 Pulse Switching 129
9.5.3.2 Packet Switching 131
9.5.4 Numerical Results 132
9.5.4.1 Event Reporting Delay 132
9.5.4.2 Power Consumption 134
9.5.4.2.1.Communication Power Consumption 134
9.5.4.2.2.Idling Power Consumption 135
9.5.4.2.3.Total Power Consumption 136
9.6 SUMMARY AND CONCLUSION 136

CHAPTER 10:PERFORMANCE OF HOP-ANGULAR PULSE SWITCHING 138
10.1 INTRODUCTION 138
10.2 STATIC EVENT SCENARIO 138
10.2.1 Impacts of Sector-constraint on Wave-front Routing 139
10.2.2 Impacts of Spatial Compression 141
10.2.3 Communication Energy Performance 144
10.2.4 Impacts of Delay-traded Sleep 147
10.3 MOVING TARGET SCENARIO 147
10.3.1 Simulation Settings 147
10.3.2 Different Target Trajectories 148
10.3.3 Impacts of Target Speed and Temporal Compression Factor 150
10.4 ERROR ANALYSIS 151
10.4.1 Impacts of False Positive Errors and Protection 151
10.4.2 Impacts of Pulse Loss 153
10.4.3 False Positive Event Rate at Sink 154
10.4.4 Impacts of Compression 155
10.5 SUMMARY AND CONCLUSION 157

CHAPTER 11:PERFORMANCE OF CELLULAR PULSE SWITCHING 158
11.1 INTRODUCTION 158
11.2 STATIC EVENT SCENARIO 158
11.2.1 Pulse Transmission Count along the Route 158
11.2.2 Cumulative Pulse Transmission Count 160
11.2.3 Different Source Cells 160
11.2.4 Impacts of Route Diversity 161
11.3 MOVING TARGET SCENARIOS 161

11.3.1 Different Target Trajectories ... 163
11.3.2 Impacts of Target Speed and Detection Rate ... 164
11.4 COMPRESSION ANALYSIS ... 165
11.4.1 Impacts of Spatial Compression ... 165
11.4.2 Impacts of Temporal Compression ... 166
11.5 PULSE ERROR ANALYSIS ... 167
11.5.1 Impacts of False Positive Errors and Protection ... 167
11.5.2 Impacts of Pulse Loss Errors ... 168
11.6 NETWORK FAULTS ... 170
11.6.1 Single Fault Analysis ... 170
11.6.2 Impacts of Fault Locations ... 173
11.7 IMPACTS OF NODE DENSITY ... 176
11.7.1 Transmission Reduction ... 176
11.7.2 Event Loss Rate ... 177
11.8 SUMMARY AND CONCLUSION ... 178

CHAPTER 12:PERFORMANCE OF POINT-TO-POINT PULSE SWITCHING ... 180
12.1 INTRODUCTION ... 180
12.2 STATIC EVENT SCENARIO ... 180
12.2.1 Impact of Distance on Pulse Transmission Count ... 180
12.2.2 Impact of Route Diversity on Pulse Transmission Count ... 181
12.2.3 Impact of Compression on Pulse Transmission Count and Delay ... 181
12.3 ERROR ANALYSIS ... 182
12.3.1 Impacts of False Positive Errors and Protection ... 182
12.3.2 Impacts of Pulse Loss Errors ... 184
12.4 COLLISION HANDLING ... 187
12.4.1 Impacts of Random Back-off Range ... 187
12.4.2 Impacts of Frame Extension Factor ... 189
12.5 SUMMARY AND CONCLUSION ... 190

CHAPTER 13:SUMMARY AND FUTURE WORK ... 191
13.1 SUMMARY ... 191
13.2 FUTURE WORK ... 192
13.2.1 Pulse-Packet Coexistence ... 192
13.2.2 Multi-Sink Pulse Transport ... 193

BIBLIOGRAPHY ... 195

LIST OF TABLES

Table 9-1: System parameters..... 105

LIST OF FIGURES

Figure 1-1: Dissertation flowchart (For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation)..... 5

Figure 3-1: Pulse switching with un-modulated UWB impulses..... 16

Figure 4-1: Network model with hop-angular event localization..... 20

Figure 4-2: Hop-distance event localization..... 21

Figure 4-3: Joint MAC-routing frame structure for hop-angular pulse switching..... 24

Figure 4-4: Wave-front routing using synchronized transitions..... 26

Figure 4-5: Route diversity and pulse merging/aggregation in HPS..... 28

Figure 5-1: Network model with arbitrarily defined sensor-cells..... 34

Figure 5-2: MAC-routing frame for cellular pulse switching..... 36

Figure 5-3: Example pulse routing loop..... 45

Figure 6-1: Network model with arbitrarily defined sensor and actuator cells..... 50

Figure 6-2: MAC-routing frame for point-to-point pulse switching..... 52

Figure 6-3: *Discovery Area* of a discovery frame for point-to-point pulse switching..... 55

Figure 6-4: *Event Report Area* of a frame for point-to-point pulse switching..... 58

Figure 6-5: Example of pulse collisions..... 62

Figure 7-1: Wave-front routing envelope due to pulse fan-out in HPS..... 72

Figure 7-2: Target within a sensor's detection range..... 77

Figure 7-3: Delay-traded sleep for idling energy conservation for HPS..... 81

Figure 9-1: Packet-to-pulse delay ratio for HPS..... 109

Figure 9-2: Communication and idling power consumption for pulse and packet in HPS..... 113

Figure 9-3: Total power consumption for pulse and packet switching in HPS 115
Figure 9-4: Packet-to-pulse delay ratio in CPS..... 117
Figure 9-5: Communication power consumption for pulse and packet switching in CPS 121
Figure 9-6: Idling power consumption for pulse and packet switching in CPS 122
Figure 9-7: Total power consumption for pulse and packet switching in CPS 123
Figure 9-8: Collision modeling diagram for PTP 125
Figure 9-9: Delay and communication power consumption between pulse and packet in PTP. 132
Figure 9-10: Idling and total power consumption between pulse and packet switching in PTP 135
Figure 10-1: Wave-front routing envelope due to pulse fan-out in HPS..... 139
Figure 10-2: Pulse transmission count, route diversity and reception diversity in HPS..... 140
Figure 10-3: Pulse Bounds of pulse transmission complexity in HPS 142
Figure 10-4: Effects of event scope on route and reception diversity in HPS..... 143
Figure 10-5: Communication power consumption per node in HPS..... 144
Figure 10-6: Communication energy performance with various δ, α and S in HPS..... 145
Figure 10-7: Effects of delay-traded sleep in HPS 146
Figure 10-8: Experimental trajectories in HPS..... 148
Figure 10-9: Pulse transmission count for HPS with and without compression..... 149
Figure 10-10: Impacts of target speed and deactivation time in HPS..... 150
Figure 10-11: Impacts of FPPR and hop-count on FPEGR in HPS 152
Figure 10-12: Performance impacts of PLR and FPPR in HPS..... 154
Figure 10-13: Impacts of pulse loss errors and false positive errors in HPS..... 156
Figure 11-1: Pulse transmission count in CPS..... 159
Figure 11-2: Target trajectories in CPS 162

Figure 11-3: Pulse transmission count for various trajectories, speeds and detection rates in CPS 163

Figure 11-4: Impacts of spatial and temporal compression factors in CPS 165

Figure 11-5: Impacts of FPPR on FPEGR with and without protection in CPS 167

Figure 11-6: Impacts of PLR on ELR without response in CPS 168

Figure 11-7: Impacts of PLR on ELR with response in CPS 169

Figure 11-8: Impacts of faults on pulse transmissions in CPS 171

Figure 11-9: Impacts of network faults with radius 1 in CPS..... 173

Figure 11-10: Impacts of network faults on ELR with radius 1 in CPS 174

Figure 11-11: Impacts of network faults with radius 2 in CPS..... 175

Figure 11-12: Impacts of network faults on ELR with radius 2 in CPS 176

Figure 11-13: Impacts of node density on reduction percentage in CPS..... 177

Figure 11-14: Impacts of node density on ELR in CPS..... 178

Figure 12-1: Impacts of distance and route diversity on pulse transmission count in PTP..... 181

Figure 12-2: Impacts of compression on pulse transmission count and delay in PTP 182

Figure 12-3: Impacts of FPPR on FPEGR in PTP..... 183

Figure 12-4: Impacts of PLR on ELR without protection in PTP 184

Figure 12-5: Impacts of PLR on ELR with protection in PTP 185

Figure 12-6: Impacts of random back-off range of collision handling in PTP..... 188

Figure 12-7: Impacts of frame extension factor in PTP..... 189

LIST OF ALGORITHMS

Algorithm 6-1: Pulse forwarding in PTP 60

Algorithm 6-2: Response mechanism without collisions in PTP 64

Algorithm 6-3: Response mechanism with collisions in PTP 66

Chapter 1: Introduction

1.1 Motivation

1.1.1 Wireless Sensor Networks

Energy efficiency is considered to be one of main concerns of wireless sensor networks (WSNs). In WSNs, sensors are responsible for detecting, monitoring and tracking events with as low energy consumption as possible. A specific application of sensor networks is binary event sensing. An example application for such sensing is *Structural Health Monitoring* (SHM) [1]. For example, while monitoring a bridge, a sensor only needs to send a fatigue event to notify a sink of a crack's occurrence, and keeps silent if no crack detected. Another example application is intrusion detection [2] in which while surveying a building, it is often sufficient for a sensor to generate an event to indicate an intrusion in its vicinity. Sending an event to a sink ideally requires a single bit information (i.e., binary information) transport for which the traditional mode of packet communication can be highly energy inefficient. Such inefficiency stems from communication, processing, and buffering overheads of a large number of bits within the payload, header, and the synchronization preambles [3] in each packet. Furthermore, by correlating the time and approximate locations of multiple binary failure or intrusion events, it is possible to study the dynamics of structural failure or the trajectory of an intruding object.

1.1.2 Wireless Sensor and Actuator Networks

WSANs are widely applied to medical, environmental, agricultural and industrial fields. In WSANs [4], sensors monitor the physical environment, while actuators make decisions and perform appropriate actions based on the readings from sensors. An example for binary event

sensing and actuation is as follows. Sensors in an office may monitor the temperature and send the information to air-conditioning actuators so that an air-conditioner can cool or heat the room accordingly. Another application is a fire extinguishing system, in which sensors detect smoke or abnormally high temperature and notify water sprinkler actuators of the fire location to extinguish the fire in time.

In WSANs, sensors that detect an event could either transmit their readings to actuators or through the sink which would process the data and send commands to actuators. For the case through the sink, nodes nearer to the sink may drain their energy faster due to heavy transmission loads. Besides, it is a waste of energy to transmit the binary data through sink. Even though sensors can directly communicate with actuators, it might still be energy-inefficient to use packets when the transported information is binary. Such energy-inefficiency stems from a large number of bits within each packet. For the fire extinguishing system, it is often sufficient for a sensor to generate a binary event to indicate a fire occurrence in its vicinity. Sending an event to a water sprinkler actuator ideally requires a single bit information transport.

1.2 Objective

The objective of this dissertation is to develop a family of transformative energy-efficient pulse switching protocols that are able to implement packet-less event communications for wireless networking applications. Pulse switching can be designed using different localization structures and Physical Layer (PHY) technologies for binary event monitoring and binary event sensing and actuation applications. Unlike multi-bit packet communication, the proposed pulse switching paradigm is based on pulse communications where a node either transmits a pulse or keeps silent to send binary information. A pulse is realized as a short burst of baseband signal. An event is coded as a single pulse which is transported multi-hop to the destination (i.e., sink in

WSNs and actuators in WSANs) while preserving the event's localization information in the form of the pulse's temporal position within a synchronized frame. The resulting operational lightness, leveraged via no addressing, no packet processing, and ultra-low communication and energy burdens make the protocol applicable for severely resource-constrained sensor devices such as Radio Frequency Identifiers (RFIDs) operating with tight energy budgets, possibly from harvested energy [5].

1.3 Challenges

The primary challenges of developing pulse networking are: 1) how to transport localization information using a single pulse, and 2) how to route a pulse multi-hop without being able to explicitly code any information within the pulse, and 3) how to deal with errors and faults.

These problems are architecturally solved by integrating a pulse's (i.e., event's) location of origin within the medium access control (MAC) and routing protocol syntaxes. More specifically, by observing the time of arrival of a pulse with respect to the MAC-routing frame, the destination (i.e., sink in WSNs or an actuator node in WSANs) can resolve the location of the corresponding event. The problem of multi-hop pulse routing is solved by introducing a certain localization method combined with *synchronized pulse frames*. In addition, the problems of errors and faults are solved by adding specific MAC-routing protocol syntaxes.

1.4 Contributions

The contributions of this dissertation are: 1) Developing a joint MAC-Routing abstraction for pulse switching, 2) Mapping the proposed pulse switching architecture on Ultra Wideband Band (UWB) impulse radio technology, 3) Developing a hop-angular spatial context abstraction

and the related estimation algorithms for event localization, 4) Designing a cellular alternative to the hop-angular abstraction, 5) Designing a peer-to-peer pulse routing protocol, finally 6) System characterization through modeling and large scale simulation.

1.5 Protocol Requirements

The applicability of the proposed event networking architectures is scoped as follows. First, they are targeted mainly to small networks with few tens of event areas distributed within a restricted geographical area. Although they may not scale well for very large networks, the protocols can enable low-frequency binary event sensing or event sensing and actuation applications. Second, the energy benefits of the architectures are predicated on the assumption of low event frequency, which is often true for structural fault monitoring type applications and intrusion detection. Finally, the proposed mechanisms in this dissertation are meant for single-sink scenarios. Using multi-band operation and syntax extensions, multiple simultaneous sinks can also be supported.

1.6 Dissertation Flowchart

Chapter 1 introduces the concept of *pulse switching* in sensor networks, and briefly discusses its motivation, challenges, contributions and protocol requirements. Figure 1-1 shows a comprehensive flowchart of the dissertation.

Chapter 2 first investigates other traditional packet-based protocols of MAC and routing layers for the sake of energy efficiency. Then it reviews two non-traditional mechanisms that use silence to transport information and thus save energy.

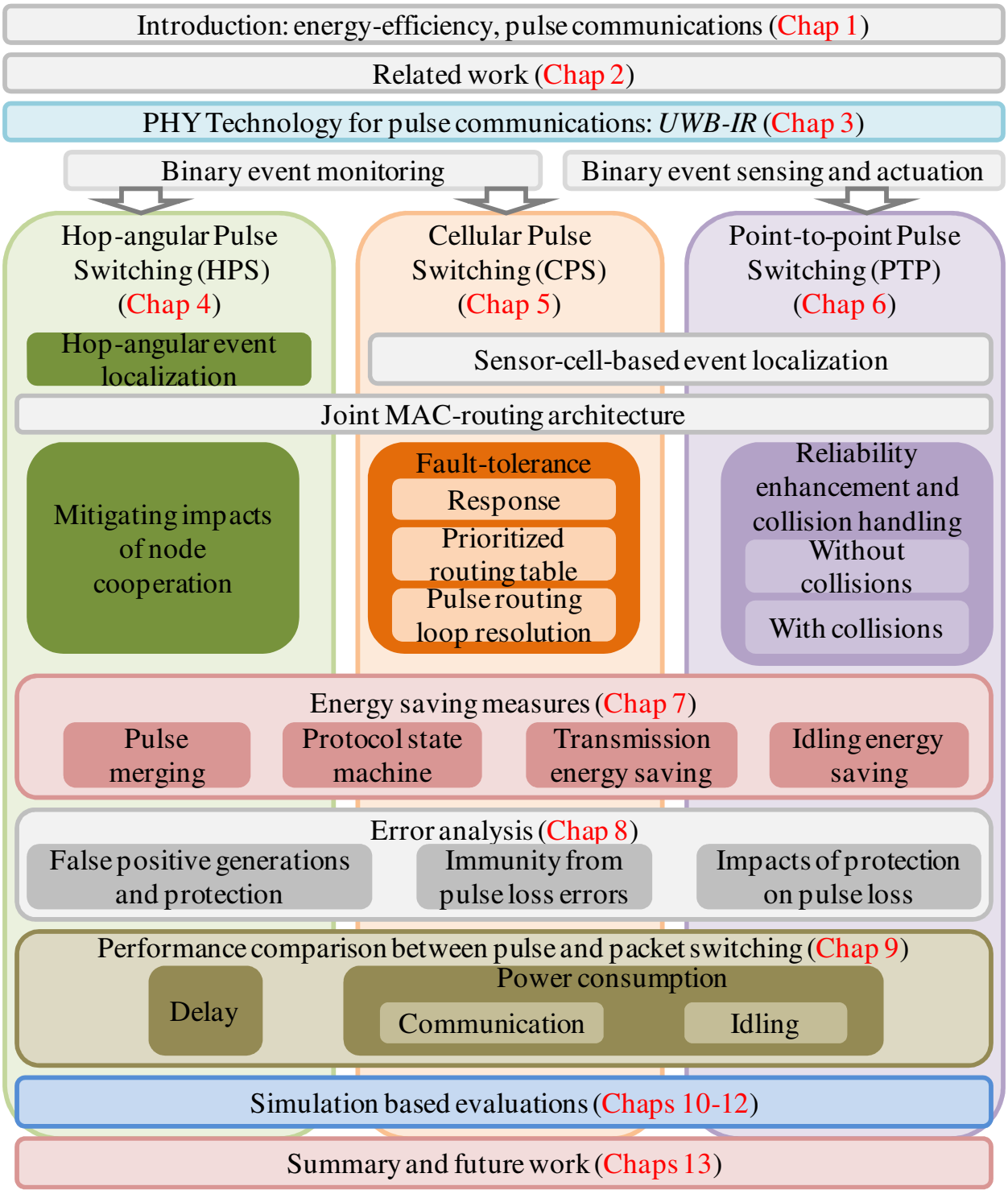


Figure 1-1: Dissertation flowchart (For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation)

Chapter 3 provides an Ultra Wide Band Impulse Radio based PHY layer technology for implementing pulse communications. The key requirement of the PHY layer is to be able to transmit and receive a single pulse without per-pulse synchronization overhead.

In Chapter 4, we propose an ultra light-weight pulse switching protocol with a novel hop-angular localization method for binary event monitoring and target tracking applications in WSNs, which is termed as Hop-angular Pulse Switching (HPS). The architectural details and their interworking for the proposed pulse switching architecture are presented.

Chapter 5 proposes an energy-efficient and fault-tolerant pulse switching protocol using a new sensor-cell based localization for binary event monitoring and target tracking applications in WSNs, which is termed as Cellular Pulse Switching (CPS). Chapter 5 provides details of the architectural design and corresponding protocol processes for CPS.

Unlike multi-point-to-point communications in Chapters 4 and 5, Chapter 6 presents an energy-efficient and fault-tolerant point-to-point pulse switching (PTP) protocol with the cell based event localization for binary event sensing and actuation applications in WSANs. The detailed architectural design and corresponding processes are provided in this chapter.

In addition to protocol state machines for CPS and PTP, Chapter 7 presents several architectural measures to improve the energy-efficiency of the HPS, CPS and PTP protocols.

Chapter 8 analyzes the functional and performance impacts of false positive errors and pulse loss errors on the proposed HPS, CPS and PTP protocols. Several protection methods from false positive errors in these three switching protocols are also proposed and then formulated in this chapter.

In Chapter 9, we introduce the comparable packet solutions for the proposed three pulse switching protocols. Then we build models for comparing the performance of pulse and packet switching in terms of the event reporting delay and power consumption.

The following Chapters 10-12 provide the simulation based evaluations of HPS, CPS and PTP protocols.

Finally, Chapter 13 summarizes this dissertation and briefly discusses the possible future work.

Chapter 2: Related Work

2.1 Introduction

There are very few efforts existing in the literature on packet-less networking using pulse communications. Traditionally, nodes in wireless networks transport data in the form of bit streams, which is referred to as Energy-based Transmissions (EbT) [6]. Nodes can convey information by silence, which is termed as Non-energy-based Transmissions (NEbT). This chapter first investigates traditional EbT protocols of MAC and routing layers from an energy-efficiency standpoint. Then it reviews three NEbT mechanisms that use "silence" to transport information in order to minimize energy consumption.

2.2 Energy-based Transmissions

2.2.1 MAC Protocols

Packet aggregation [7] is a natural and effective approach to reduce synchronization preamble and header overheads by aggregating the payloads from multiple *short* packets into a single *large* packet that is routed to a monitoring sink node. While being able to cut the energy costs, aggregation still requires the inherent packet overheads at the originating nodes for the *short* packets, and then end-to-end for fewer numbers of the *large* packets. The paper in [7] reduces preamble and header overheads of packet communication by aggregating payloads from multiple short packets into a single large packet that is routed to a sink. The objective of this dissertation is to develop protocols for fully eliminating such overheads required due to packet abstraction. While reducing the energy cost, aggregation still requires the inherent packet overheads.

The paper in [8] proposes a wireless MAC protocol that utilizes out-of-band contention pulses for packet collision detection. Unlike the solution proposed in this dissertation, pulses in [8] are of varying length, rendering technologies such as Ultra Wide Band Impulse Radio (UWB-IR) unusable. Additionally, although pulses are used for handling collisions, traditional packets are still used for sending information. Therefore, the Protocol Data Unit (PDU) related overheads of packet switching are present in [8].

Idling energy reduction in synchronous packet-based MAC protocols such as the Timeout-MAC protocol (T-MAC) [9] is accomplished via interface sleeping in appropriately scheduled packet slots. Idling in asynchronous protocols such as the Berkeley MAC protocol (B-MAC) [10] is reduced by relying on low-power listening, also called preamble sampling, to link together a sender to a receiver that is duty cycling. Hybrid protocols also exist that combines a synchronized protocol like T-MAC with asynchronous low power listening [11]. Distributed time division multiple access (TDMA) protocols [12] avoid idling consumption by turning interface off in all packet slots except when needed for transmissions and receptions. The paper in [13] proposes a TDMA based scheduling scheme that balances energy-saving and end-to-end delay. Such a scheme achieves the reduction of the end-to-end delay caused by the sleep mode operation while at the same time it maximizes the energy savings. Although the above MAC solutions can improve idling energy expenditure in low duty-cycle networks, they still use packet switching, thus suffering from the overheads that pulse switching can avoid.

The authors in [14] present a low-energy and delay-sensitive TDMA-based MAC protocol for WSANs. A network is constructed by multiple clusters, each of which is managed by a single actuator. To avoid inter-cluster downlink interference, Code Division Multiple Access (CDMA) codes are introduced. The paper in [14] identifies the advantages of the distributed MAC

protocol in [14] over existing TDMA-MAC schemes in cluster-based sensor networks. Additionally, the simulation results demonstrate that the protocol in [14] greatly improves the WSANs lifetime and achieves a good trade-off between the packet delay and sensor energy consumption. However, the paper in [14] does not consider the design of the routing layer protocol and still uses the packet abstraction.

2.2.2 Routing Protocols

Instead of finding the lowest energy paths, the energy-aware routing in [15] uses probabilistic forwarding to send data on different routes for low energy and low bit rate networks. Compared to directed diffusion routing [16], the network lifetime is increased up to 40% by the energy-aware routing.

The first cluster-based forwarding service for reducing the number of retransmissions is presented in [17], which is designed as an architectural extension to existing routing protocols in WSNs. In such a cluster-based forwarding, each node forms a cluster so that any node in the next-hop's cluster is able to participate in the forwarding process.

In order to maximize the network lifetime and minimize the total consumed energy, the authors in [18] propose an adaptive routing protocol with energy efficiency and event clustering (ARPEES) in WSNs.

The paper in [19] presents an Energy-Balanced Routing Protocol (EBRP), which builds a mixed virtual potential field in physics. It significantly improves the performances of energy balance, network lifetime, coverage ratio and throughput compared to the commonly used energy-efficient routing algorithms.

The authors in [20] present a Quality of Service (QoS) mechanism based routing protocol for WSANs, which meets application requirements for low latency, high delivery reliability,

uniform energy consumption and fault tolerance. Besides, it utilizes the interactions among sensors, actuators and the sink to provide a better QoS solution for WASNs. In details, optimal routes are constructed in actuator nodes under the above application requirements. Moreover, the paper in [20] uses publish/subscribe mechanism to promote the interactions among sensor, actuator, and sink nodes.

In [21], the Delay-Energy Aware Routing Protocol (DEAP) for heterogeneous sensor and actor networks is proposed to enable a wide range of tradeoffs between delay and energy consumption. DEAP includes an adaptive energy management scheme and a loose geographic routing protocol. It is a distributed and randomized algorithm where nodes are able to locally decide whether to sleep or not depending on the delay level experienced by packets.

A localized Delay-bounded and Energy-efficient Data Aggregation (DEDA) scheme is proposed in [22] for low-traffic and request-driven WSANs. The concept of Desired Hop Progress (DHP) is used in DEDA. With edges from a Local Minimal Spanning Tree (LMST), DEDA is able to build the shortest path tree rooted at an actor for data aggregation. DEDA is shown to be able to save 25-75% energy per node, and can extend network lifetime up to 150% compared with the existing localized solution [23].

Although the above routing solutions attempt to improve energy efficiency in WSNs and WSANs, they still use packets to transport information. This dissertation will establish that the overheads resulting from packets can be avoided by pulse switching, in order to significantly reduce the energy consumption.

2.2.3 Joint MAC-Routing Protocols

The paper in [24] proposes a centralized energy-efficient routing protocol, which can distribute the energy dissipation evenly among all sensor nodes to improve network lifetime and

average energy savings. A high-energy base station is utilized to perform most of the energy-intensive tasks, such as cluster setup, cluster head selection, routing path formation, and TDMA schedule creation.

Joint MAC scheduling and route computation is proposed in [25] for delay and energy optimization for event monitoring applications. In the cross layer approach in [25], it is shown how reporting delay can be optimized in the presence of pre-defined sleep-wake MAC cycles.

The authors in [26] consider a geographical-location-based forwarding technique combined with a collision-avoidance MAC scheme, in which the relaying node is randomly selected via contention among receivers. In the analysis and simulation results, it is shown that some relevant trade-offs are highlighted and parameter optimization is pursued.

The paper in [27] proposes an integrated MAC and routing protocol (AIMRP) for rare event detections in WSNs. A completely asynchronous and random duty-cycling scheme is designed in AIMRP for power saving. Compared to the Sensor-MAC protocol (S-MAC) [28], AIMRP performs better in terms of power consumption for event detection applications with the same event reporting latency constraints.

Although the above cross-layer solutions can improve energy expenditure in WSNs and WSANs, they still use packet switching, thus suffering from the overheads that a pulse based system can avoid. It will be shown in this dissertation that by sending a single pulse, instead of a packet, the energy consumption can be significantly reduced. In addition, the processing and buffering costs of packets can be avoided using pulses.

2.3 Non-energy-based Transmissions

The mechanism in [6] partially mitigates the packet switching overhead by sending two very short packets for a sensed event, so that the arrival delay between those short packets

represent a value corresponding to the event. The short packets serve only the purpose of start/stop delimitation and do not carry any data. Although there are a number of practical challenges as outlined in [6], it is an innovative approach for partially eliminating the need for packet PDU related overheads. The short control packets, including *start*, *stop*, and *intermediate*, however, require substantial bit overheads contributed by packet headers with node addresses and per-packet preambles. The objective of this dissertation is to develop pulse-based protocols for fully eliminating such overheads.

The authors in [29] develop models for energy and delay bounds for bit (i.e., packet based) and pulse communications in single hop networks. The main results in [29] are that the worst case energy performance of pulse communication can be substantially better than that of packet based communication, although with a possibly worse delay performance. A notable limitation is that the paper does not provide mechanisms for scaling these results for multi-hop networks. Also, no MAC and routing protocol details are provided. Routing a pulse multi-hop can be particularly challenging given that no explicitly coded information can be carried in a single pulse. The objective of this dissertation is to design a MAC-routing framework that can be used for practical implementations of a pulse based communication paradigm working in multi-hop environments.

For target tracking applications, [30] proposes a binary sensing model in which each sensor returns only one-bit information regarding a target's presence or absence within its sensing range. Although this binary sensing saves energy to some degree, the approach in [30] too uses the packet abstraction. The objective of this dissertation is to fully eliminate packets via replacing them by pulse switching.

The limitations of the above three NEbT mechanisms [6, 29, 30] are addressed in this dissertation through the design of a MAC-routing framework that can be used for practical implementation of multi-hop pulse switching. Communication energy cost for pulse switching can be significantly smaller than those for packets due to the difference in the number of bits to be transported. In addition, the processing and buffering costs of packets can be avoided using pulses.

Chapter 3: Technology for Pulse Realization

3.1 Introduction

In order to realize pulse communications, the physical layer technology should be able to transmit and receive a single pulse without per-pulse synchronization overhead. This chapter discusses Ultra Wide Band (UWB) Impulse Radio (IR) [31-35] as an option for the physical layer technology for pulse communications.

3.2 Ultra Wide Band Impulse Radio Technology

3.2.1 Ultra Wide Band Impulse Radio

UWB transmission can be defined as a transmission where the radio frequency (RF) bandwidth exceeds 1 GHz [31]. UWB systems are ideally targeted and designed to be of low power, low cost, high data rates, precise positioning, and extremely low interference [35]. The traditional narrowband wireless technology broadcasts on separate frequencies. UWB transmissions spread across a very wide range of frequencies. Accordingly, a pulse or a series of pulses can replace the sinusoidal radio wave per time unit.

UWB Impulse Radio is based on very short pulses. *Gaussian monocycle* can be a UWB-IR pulse type, as shown in Figure 3-1a. The waveform of a *Gaussian monocycle* is expressed as [35]:

$$v(t) = A\frac{t}{\tau}\exp\left\{-6\pi\left(\frac{t}{\tau}\right)^2\right\} \qquad (3\text{-}1)$$

Where A is an amplitude scale factor and τ is the time constant of the pulse. The total pulse width is approximately τ , which is typically between 0.2 and 1.5 *ns*.

3.2.2 Ultra Wide Band Impulse Radio Slotting

Figure 3-1 depicts a UWB implementation of the proposed pulse structures in this dissertation. The UWB pulse width is chosen to be 1 *ns*, and the pulse repetition period T_b is 1000 *ns* [31], which determines the slot size in this case. This large difference between pulse width and slot size minimizes the overlapping probability between pulses in adjacent slots in the presence of multi-path delay, as shown in Figure 3-1b.

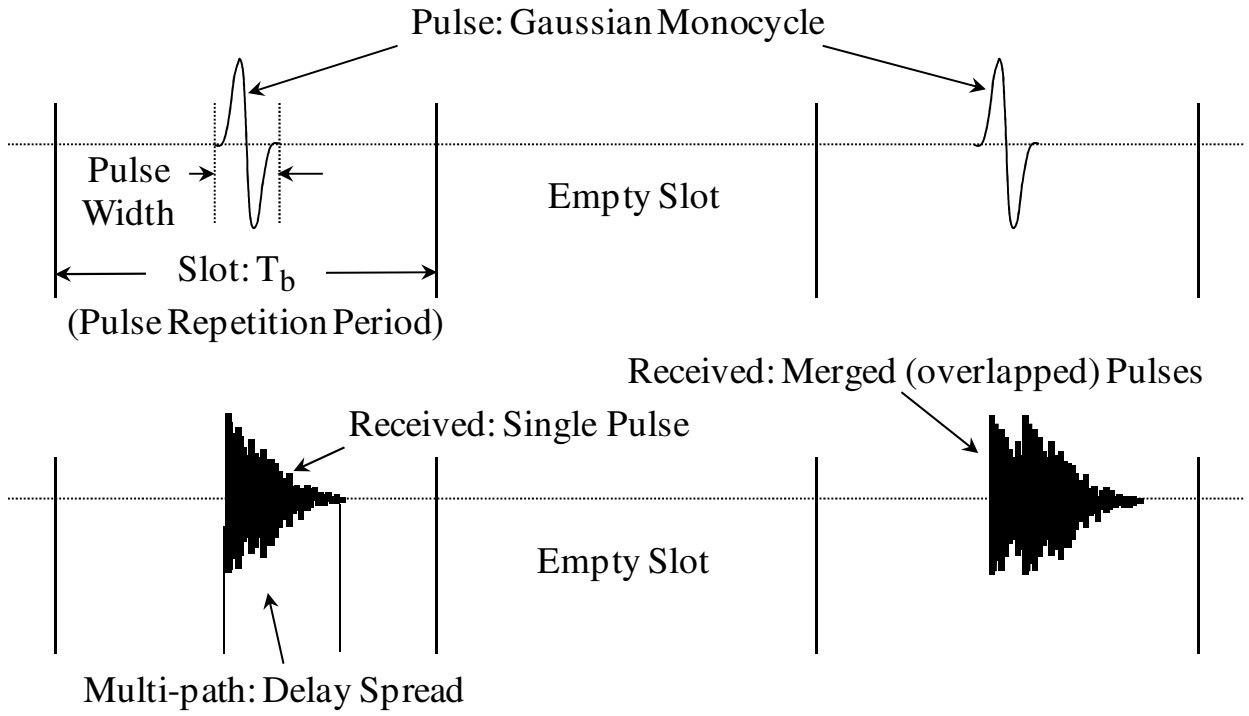


Figure 3-1: Pulse switching with un-modulated UWB impulses

3.2.3 Modulation and Synchronization

Since each individual pulse carries information about an event by itself, unlike in packet based UWB-IR, no Time Hopping Sequence (THS) [32] based Pulse Position Modulation (PPM) is needed in this architecture. Additionally, no per-pulse synchronization preambles are needed.

This helps avoiding huge synchronization overheads of up to 600 pulse repletion periods [36], as typically needed for packet preambles.

3.2.4 Energy Budget

The all-digital baseband operation of the UWB-IR enables it to be implemented in low-cost Complementary Metal-Oxide-Semiconductor (CMOS) logic [33]. For example, with 0.18 μm CMOS based UWB, power consumptions of 4 nJ for each pulse transmission (average 4 mW with 1000 ns pulse repetition period), 8 nJ for each pulse reception (average 8 mW) and idling consumption of 8 mW can be typical [33].

3.2.5 Sources of Errors

Primary sources of errors for UWB-IR are large multi-path delay spread (see the right graph in Figure 3-1), noise and interference. Large multi-path delay spreads can cause overlapping pulses, leading to errors in pulse detection. As shown in Figure 3-1, very large slot size (i.e., *1000 ns*) can be chosen in comparison to the pulse width (i.e., *1 ns*) for minimizing the overlapping probability by constraining the multi-path delay spread within a slot. Pulse losses due to the misdetection of overlapped pulses need to be dealt within the proposed pulse switching architectures.

The other primary contributors to errors are interferences including Multi-User Interference (MUI) and Narrow-Band Interference (NBI). The approach in [37] proposes a receiver to mitigate MUI by a combination of statistical interference modeling and thresholding. The above factors can result in one of two types of detection errors at a UWB receiver [38]. They are *false positive pulse errors* and *pulse loss errors*.

3.3 Summary and Conclusion

This chapter mainly investigated the feasibility of UWB-IR as a physical layer technology for pulse switching. There are other physical layer technologies that can be used for pulse communications. Such technologies include ultrasonic Lamb waves in a metal [39] or composite, audio chirps over air or liquid and infra-red pulses.

Chapter 4: Hop-angular Pulse Switching Protocol

4.1 Introduction

This chapter proposes an ultra light-weight pulse switching protocol with a novel hop-angular localization method for binary event monitoring and target tracking applications in WSNs. The mechanism is termed as Hop-angular Pulse Switching (HPS). In the HPS architecture, synchronized pulse waves are created in the network so that a pulse can simply "ride" *synchronized phase waves* across different hop-distance nodes from a sink in order to get delivered to the sink.

4.2 Network Model

4.2.1 Network Topology

As shown in Figure 4-1, a network contains arbitrarily distributed sensors that send pulses to a sink. Depending on the node locations and the transmission range (assumed to be non-uniform), each node resides at a certain hop-distance from the sink. In Figure 4-1, the hop-distance for each node is marked by the node.

The sink is assumed to be capable of making high-power transmissions with full network coverage for frame-synchronizing the sensors. For small networks with a few tens of sensors distributed in a restricted geographic area, such reach-ability is achievable. This is especially true with high power pulse generation techniques described in [40]. Since the proposed architecture is targeted to small networks, full network coverage by the sink is a valid assumption.

The sensors in this architecture are never individually addressed, and therefore no per-sensor addressing is necessary at the MAC or routing layers. An event is always identified by its location of origin, and not the sensor of origin.

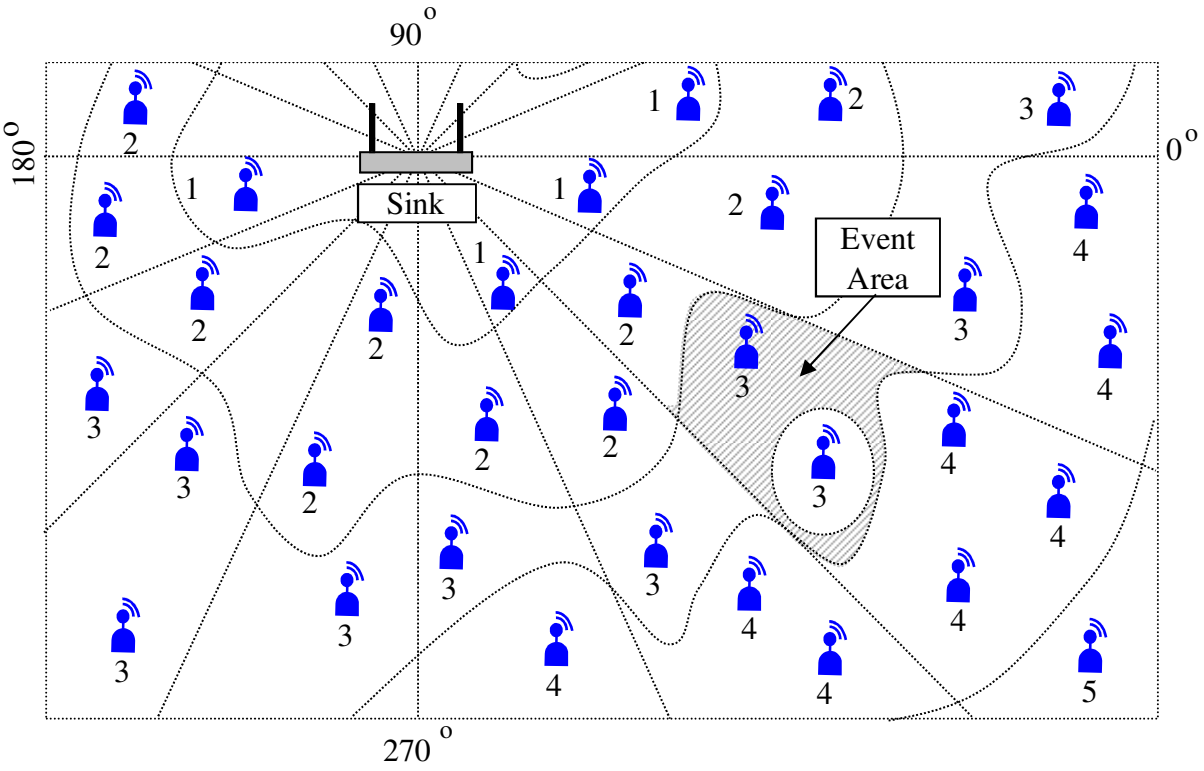


Figure 4-1: Network model with hop-angular event localization

4.2.2 Pulse as Protocol Data Unit

Pulse networking can cater to on-off style monitoring for structural health, intrusions, and disasters – all for generating events when specific parameters of interest cross predefined thresholds. An event results in a pulse which is transported multi-hop to a sink. A pulse is able to represent: a) the very occurrence of the event, and b) its location of origin. As investigated in [6], even with such limited information, several application level conclusions can be derived at the sink by correlating multiple event pulses. For example, while monitoring a bridge, by correlating the time and approximate location of a fatigue event it is possible to study the structural failure

dynamics. Similarly, by fusing pulses from different intrusion events, the trajectory and speed of an intrusion can be inferred. The success of pulse switching crucially hinges upon its ability to do event localization using a single pulse transported to a sink. Hop-angular Event Localization

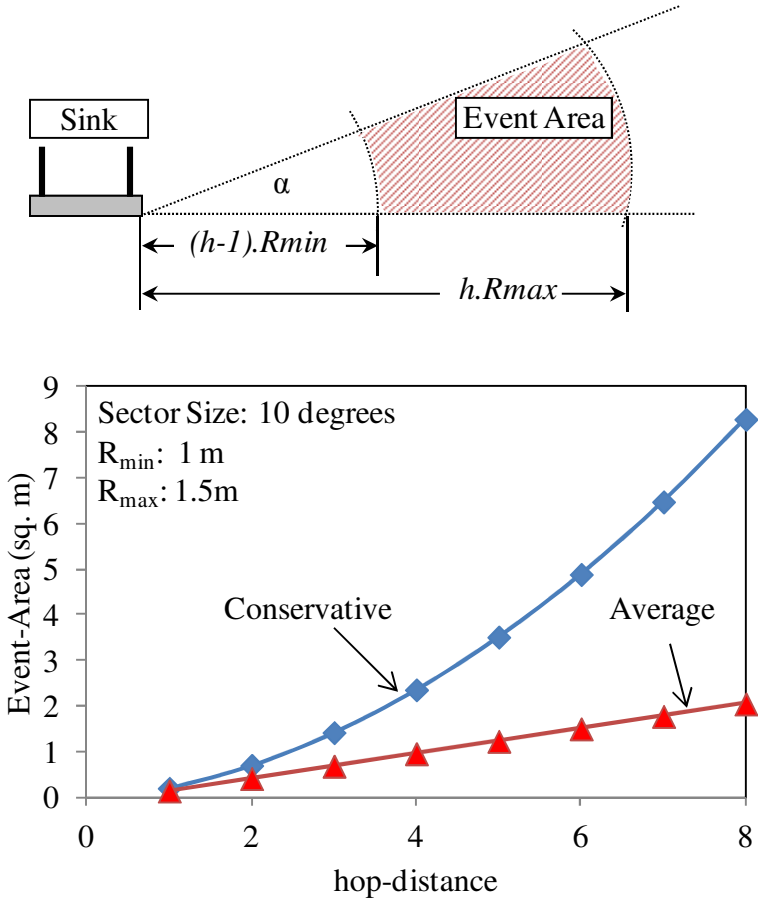


Figure 4-2: Hop-distance event localization

A concept of hop-angular event area is introduced for event localization. The network is logically divided into a fixed number of angular sectors. In Figure 4-2, for example, there are 16 $22.5°$ wide sectors. With a pre-defined sector-width ($\alpha°$), the location of a sensor can be represented by the tuple {sector-id, hop-distance}. For example, the location of the encircled sensor in Figure 4-1 can be represented as {15, 3}, meaning the node is located in the 15^{th} sector, with a hop-distance 3 from the sink. While the angle for a node is pre-programmed at the

deployment time, its hop-distance can be dynamically discovered using the process outlined in Section 4.3.2. The {*sector-id*, *hop-distance*} tuple indicates an event-area, whose size determines the event localization resolution. This tuple for an event's origin is carried to the sink by the corresponding pulse.

With known transmission range (between the parameters $R_{\min}$ and $R_{\max}$) and the sector-width α, the sink can estimate the event-area using the {*sector-id*, *hop-distance*} tuple. Higher angular resolution (i.e., smaller α) and smaller transmission range result in smaller event areas, leading to better localization resolution. Note that multiple simultaneous events from the same event-area will be resolved using the same {*sector-id*, *hop-distance*} tuple.

Consider the example event-area identified by the tuple $\{\alpha, h\}$ in the top portion of Figure 4-2. With a sector-width of α, and $R_{\min}$, $R_{\max}$ representing the known minimum and maximum wireless transmission range, the most conservative (coarse) localization resolution can be expressed as the largest possible event area: $A_{consevative} = \left\{h^2 {R_{\max}}^2 - (h-1)^2 {R_{\min}}^2\right\}\alpha\pi/360$. The average resolution is $A_{average} = \left\{h^2 R^2 - (h-1)^2 R^2\right\}\alpha\pi/360$, where $R = (R_{\min} + R_{\max})/2$. For example, with transmission range spanning between 1 *m* to 1.5 *m*, in a network with sector-width (i.e., α) of $10°$, the size of an event-area that is 5 hop-distance away is approximately 3.5 square meters. For the intrusion detection application, this means that an intrusion can be localized within approximately 3.5 square meter area. For a given α and transmission range, since this resolution reduces with higher hop-distances, the maximum network size will have an upper bound for a desired target resolution for event localization.

4.3 Hop-angular Pulse Switching Architecture

This section presents architectural details and their interworking for the proposed hop-angular pulse switching architecture.

4.3.1 Joint MAC-Routing Frame Structure

Nodes in the proposed system are frame-by-frame time synchronized by the sink, and they maintain MAC-Routing frames (see Figure 4-3) in which each slot is used for sending a single pulse. The slot includes a guard time to accommodate the cumulative clock-drift during a frame, which can be very small for RF technology such as UWB-IR, as the frame size itself can be ultra short (μs) for UWB.

The downlink sub-frame contains a synchronization slot in which the sink transmits a *full power* pulse to make all nodes frame-synchronized. The two following downlink slots and the reconfiguration part of the uplink control sub-frame are used for hop-distance discovery. The *Reconfiguration Area* has $(H+1)$ slots, where H is the maximum hop-distance. The *Forwarding Flag Area* is designed for pulse compression. The H-slot wide *Routing Area* of the control sub-frame is used for energy management.

The event sub-frame contains H slot clusters, each containing $360/\alpha$ slots, where α corresponds to the sector-width. Each slot within a cluster corresponds to a specific {*sector-id, hop-distance*} tuple. Meaning, for each event-area, represented by {*sector-id, hop-distance*}, there is a dedicated slot in the event sub-frame. An event originating node transmits a pulse during the dedicated event sub-frame slot that corresponds to the {*sector-id, hop-distance*} of the node's event area. While routing the pulse towards the sink, at all intermediate nodes it is transmitted at the same event sub-frame slot that corresponds to the {*sector-id, hop-distance*} of its event-area of origin. In other words, while being forwarded, the transmission slot for the pulse

at all intermediate nodes does not change with respect to the frame. This is how information about the location of origin of an event is preserved during routing. Upon reception, the sink can infer the event-area of origin from the {*sector-id, hop-distance*} value corresponding to the slot at which the pulse is received.

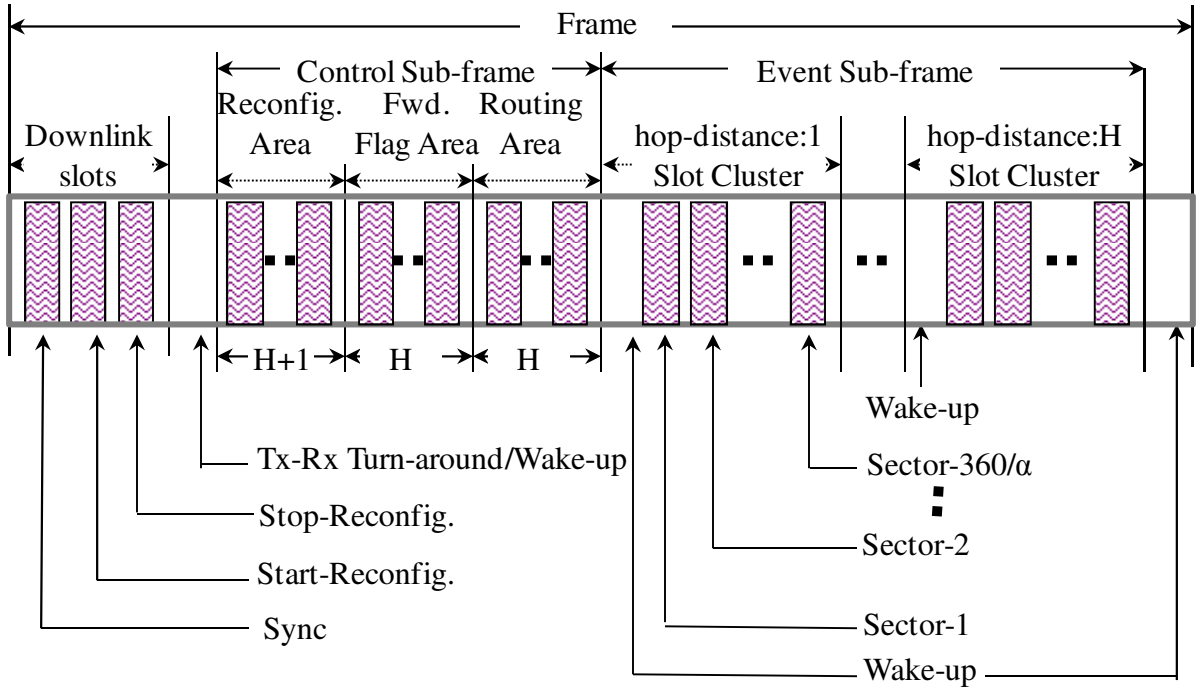


Figure 4-3: Joint MAC-routing frame structure for hop-angular pulse switching

4.3.2 Hop-Distance Self Discovery

The sink initiates a reconfiguration phase by sending a *full power start-reconfig* pulse. It then transmits a *regular power* (i.e., as used by other nodes) pulse at the first slot in the *Reconfiguration Area* in the frame. Nodes that receive this pulse conclude that they are *1* hop-distance away from the sink. All hop-distance *1* nodes send a pulse in the second slot of the *Reconfiguration Area* during the next few frames. Nodes receiving these pulses conclude that they are in hop-distance *2*. This process continues for Hc frames ($c > 1$) after which all nodes are done discovering their individual hop-distances. The general logic is that if a node receives

reconfiguration pulses from nodes with different hop-distances (inferred from the slots of receptions), then its own hop-distance is one more than the smallest hop-distance node from which a pulse was received. After Hc frames, the sink ends the reconfiguration process by sending a *full power* pulse in the *stop-reconfig* slot. Although H frames can be sufficient for the hop-discovery, the factor c is introduced for redundancy to cope with pulse losses.

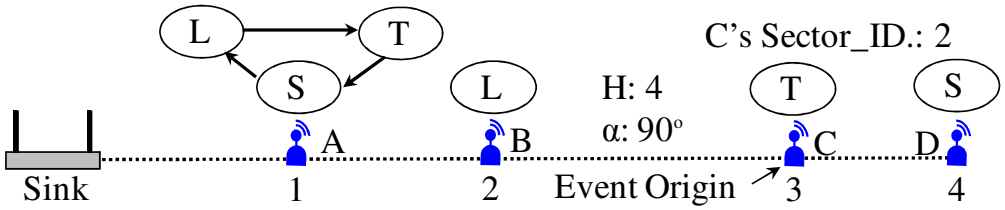
Note that the above discovery process and the proposed architecture in general, do not assume any specific shape (i.e., circular or otherwise) of a node's transmission coverage area. It could be of any arbitrary shape as shown in Figure 4-1. Due to fading and changing network conditions, the transmission coverage of the nodes and the resulting hop-distances are expected to change over time. To accommodate such changes, the above hop-distance discovery process needs to be periodically executed. Although each such discovery process for Hc frames will incur certain additional energy overhead, such long term overhead is expected to be marginal for stable wireless environments.

4.3.3 Pulse Forwarding using Wave-front Routing

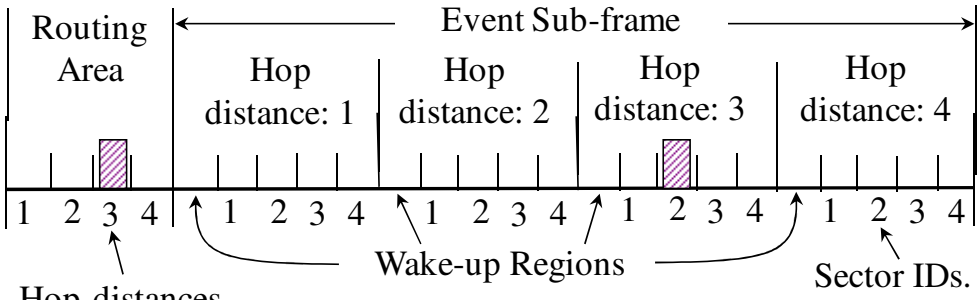
When a pulse is transmitted by a node at hop-distance h, only its neighboring nodes at hop-distance $(h-1)$ need to forward it towards the sink. In the absence of MAC addressing, a node cannot determine the hop-distance of a pulse's transmitter node. We introduce a *wave-front* routing in which nodes synchronously transition in a frame by frame *Sleep (S)–Listen (L)–Transmit (T)* state cycle that enables pulse forwarding towards the sink. Nodes with the same hop-distance cycle in-phase, but those with different hop-distances remain synchronized but out-of-phase so that when the hop-distance h nodes transmit, the hop-distance $(h-1)$ nodes listen, and the hop-distance $(h+1)$ nodes sleep. This synchronized cycling ensures that pulses

transmitted by nodes in hop-distance h are received by those at hop-distance $(h-1)$, but are ignored by nodes at hop-distance $(h+1)$. This creates a *wave-front* that carries pulses closer to the sink on a frame-by-frame basis.

Immediately after the reconfiguration process is terminated, a node at hop-distance h decides its state phase (one of *S*, *L*, or *T*) by simply computing *h modulo 3*. The outcomes *0*, *1*, or *2* cause the node's state to be initialized as *L*, *T*, or *S* respectively. During the subsequent frames, the state machine indefinitely cycles in the sequence *S-L-T*. Irrespective of its current state phase, a node wakes up at the end of a frame (see Figure 4-3) for receiving the frame synchronization pulse from the sink.



a: Sensor hop-distance and alternating state phases



b: Frame level activities

	Node-A	Node-B	Node-C	Node-D
Frame 1: C to B	*S*	*L* ←	*T*	*S*
Frame 2: B to A	*L* ←	*T*	*S*	*L*
Frame 3: A to Sink ←	*T*	*S*	*L*	*T*

c: State transitions sequence

Figure 4-4: Wave-front routing using synchronized transitions

The concept is explained in Figure 4-4 using the example wave-front routing of a pulse generated at node *C* {*sector-id: 2, hop-distance: 3*}. The maximum hop-distance *H* is *4*, and with an α of $90°$, the maximum number of sectors is $360/90 = 4$. The routing area of the control sub-frame, and the event sub-frame are shown in Figure 4-4b. When *C* is in transmit phase and has an event to send, it sends a pulse in the corresponding slot {*sector-id: 2, hop-distance: 3*} in the event sub-frame. As shown in Figure 4-4c, transfer of the pulse from *C* to *B* occurs during Frame-1. In Frame-2, *B* forwards it to *A*, and finally during Frame-3, the pulse is delivered to the sink during the same {*sector-id: 2, hop-distance: 3*} slot. All three frames used by *C*, *B* and *A* look the same as that shown in Figure 4-4b.

This pulse forwarding is termed as *wave-front routing*, because the pulses simply "ride" the *synchronized phase waves* across different hop-distance nodes, and get delivered to the sink. No address based forwarding is needed. The buffering need is drastically smaller than that of the packet-based systems with variable queuing. Also, the routing depends only on a node's knowledge of its own hop-distance, and not on the underlying event localization mechanism (e.g., *hop-angular*). Therefore, as long as the hop-distance information is known, the pulse routing can be implemented with other event-localization mechanisms.

4.3.4 Exploiting Route Diversity

Wave-front routing ensures that a pulse is forwarded only across nodes with reducing hop-distances. As shown in Figure 4-5a, a pulse originated from node *E* is not forwarded by node *D* which has the same hop-distance 3, but both nodes *B* and *C* forward it to node *A*, which in turn delivers it to the sink. Since all transmissions take place at the same slot in the event sub-frame, the transmissions from *B* and *C* get merged while being delivered to node *A*. This phenomenon of multiple intermediate route segments gives rise to *route-diversity*, which provides tolerance

from errors. For example, the pulse can be delivered to sink in spite of a failure of node *B* or *C*, or a transmission error across *E*-*B* or *E*-*C*.

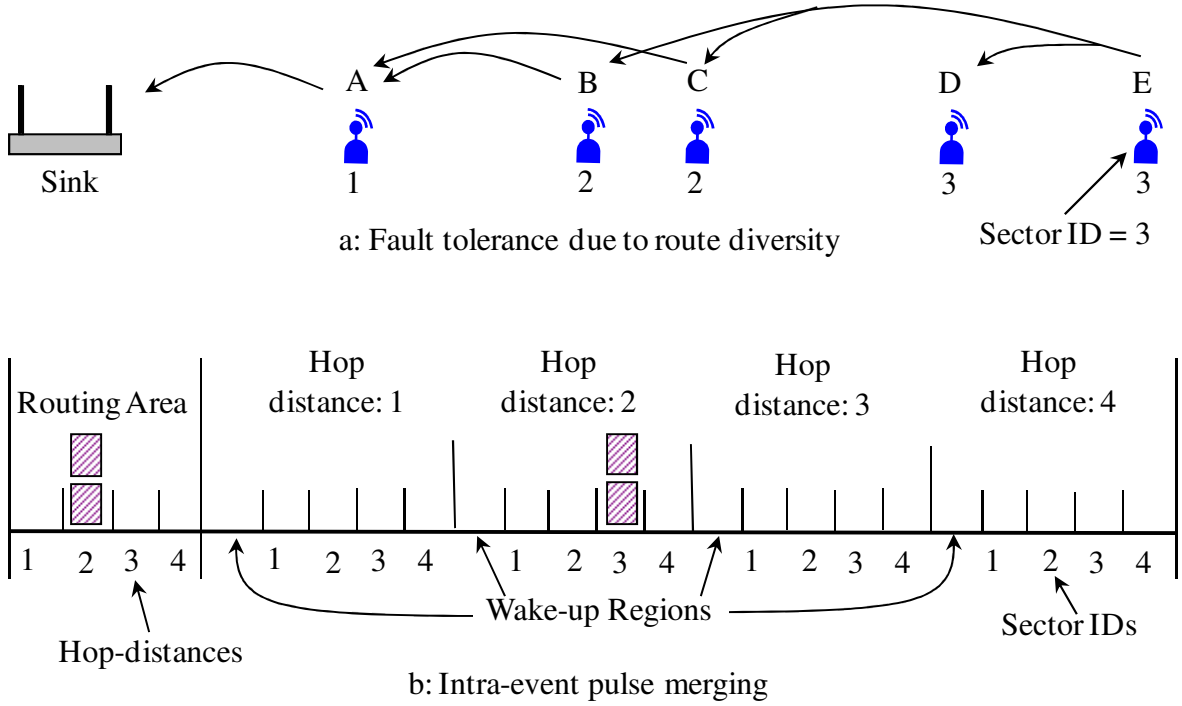


Figure 4-5: Route diversity and pulse merging/aggregation in HPS

4.3.5 Pulse Merging

Pulse merging may happen when multiple nodes transmit pulses during the same slot. In Figure 4-5, since the pulse is transmitted by nodes *B* and *C* on the same slot in the event sub-frame, the receiver *A* receives 2 collided pulses in the same slot.

It is interesting to note that such a pulse collision does not impact the protocol operations. As long as the RF hardware can detect the presence of any of these collided pulses, the routing continues. In other words, the receiver node regards the detected collided pulses within the same time slot as a single pulse. Thus, the pulse collision in a time slot functions as pulse merging. Such merging could also take place when multiple pulses from the same event area are originated in the same frame. Also note that pulse merging can occur at any stage of an end-to-

end route, including at the sink node. Instead of preventing pulse routings, pulse merging work as inherent in-network aggregation for events originated from the same event area.

4.4 Mitigating Impacts of Physical Layer Cooperation

4.4.1 Motivation

Pulse merging, as explained in Section 4.3.5, can give rise to an undesired effect of *node cooperation* [41] when multiple nodes simultaneously transmit pulses in a slot. Such simultaneous transmissions can increase the effective transmission power, thus extending RF transmission range to more than one hop distance. Node cooperation can affect both hop-distance discovery and pulse forwarding as described in Sections 4.3.2 and 4.3.3.

4.4.2 Mitigating Cooperation during Hop-distance Discovery

During hop-distance discovery (see Section 4.3.2), after the nodes at hop-distance *1* have discovered their own hop-distance, they send simultaneous discovery pulses in the same slot in the *Reconfiguration Area*. Unintended node cooperation due to the energy aggregation of all such pulses can cause them to reach nodes that are beyond hop-distance *2*. This can give rise to faulty hop-distance recovery, leading to possible pulse forwarding failures. Such problems in hop-distance discovery due to *node cooperation* can happen at all hop-distances except hop-distance *1*. The following mechanisms are proposed for minimizing impacts of node cooperation by reducing the chances of overlapping pulses during hop-distance discovery.

The number of slots in the *Reconfiguration Area* (see Figure 4-3) is increased from $(H+1)$ to $(HM+1)$. The first slot is still allocated to the sink, and the nodes in each hop-distance are allocated a cluster of M slots instead of a single slot. Additionally, a slot is functionally divided

into N_p pulse positions or mini-slots. The hop-discovery process in Section 4.3.2 is augmented with the following new rule. A node in the hop distance *h* generates a pulse after: 1) randomly selecting one of *M* slots allocated for the h^{th} hop-distance, and 2) randomly selecting one of N_p pulse positions within the slot selected in step 1.

With this rule, the probability of non overlapping pulses generated by N_h nodes in hop distance *h* can be estimated as $P_h^{dif} = 1 - \sum_{k=2}^{N_h} \binom{N_h}{k} (\hat{p})^k (1-\hat{p})^{N_h - k}$, in which the second term represents the probability of having at least two overlapping pulses among N_h nodes, and $\hat{p} = 1/MN_p$ representing the probability of having one pulse in the slot cluster (i.e., *M* slots) allocated for the nodes in hop-distance *h*. For a typical UWB pulse width of *1 ns* and pulse repetition period of *1000 ns* [31], the typical value of N_p is *1000*, with which the quantity P_h^{dif} approaches to *1*. It means that with the augmented hop-distance discovery rule, the probability of non-overlapping pulses generated by the nodes in a hop-distance is near *100%*. Therefore, the effects of physical layer node cooperation on hop-discovery can be mostly mitigated.

In very rare occasions when a node may still receive overlapping pulses (a receiver node cannot detect overlapping pulses), the following additional rule can further reduce the possibility of faulty hop-distance discovery. If a node receives multiple pulses in different hop-distance slots (i.e., hop-distances $1,...,h-1$) in the *Reconfiguration Area*, the node decides its own hop distance as $\max\{1,...,h-1\}+1=h$. For example, consider the rare occasion in which two nodes at hop-distance *2* send out overlapping pulses, which can reach a node with hop-distance *4*. This happens due to node cooperation caused by energy aggregation of the overlapping pulses. Without such cooperation, the receiver node in question should have received discovery pulses

only from nodes in hop-distance *3* and not hop-distance *2*. With cooperation, however, the receiver node ends up receiving pulses from nodes in hop-area *2* as well as correct discovery pulses from hop-area *3*. According to the max-rule, the node interprets its own hop-distance as *4* which is one more than the maximum of *2* and *3*. This way, impacts of node cooperation is fully mitigated during the hop-distance discovery.

4.4.3 Immunity to Node Cooperation during Pulse Forwarding

Wave-front pulse forwarding, as proposed in Section 4.3.3, is inherently immune to node cooperation due to its hop-distance synchronized state transitions. Consider a cooperation situation in which multiple nodes in hop-distance $(h+1)$ (at state *T*) simultaneously forward a pulse for an even that was generated at a higher hop-distance event area. During this transmission, nodes in *h*, $(h-1)$, and $(h-2)$ hop-distances are in states *L*, *S*, and *T* respectively. This means, due to node cooperation even if the pulse reaches areas with hop-distances $(h-1)$ and $(h-2)$, it will be ignored simply because the nodes in those areas are not in a listen state. Nodes only in hop-distance *h* will receive it, which is intended.

4.5 Summary and Conclusion

The key contribution of this chapter is to combine a hop-angular event localization mechanism with a pulse switching protocol in a manner that allows a receiver to localize an event by observing the temporal position of a received pulse with respect to a synchronized frame structure. The combined framework is termed as Hop-angular Pulse Switching or HPS. In addition to network topology and hop-angular event localization, this chapter proposed a joint MAC-routing architecture for HPS, and then demonstrated the corresponding hop-distance self discovery and pulse forwarding processes. The phenomenon of pulse merging was discussed and

regarded as an inherent in-network aggregation. Pulse merging can result in an undesired effect of *node cooperation* [41], which may might impact both the processes of hop-distance self discovery and pulse forwarding. A method of mitigating the impacts of node cooperation on hop-distance self discovery was proposed. Moreover, HPS has been shown to be immune to node cooperation during pulse forwarding.

Chapter 5: Cellular Pulse Switching Protocol

5.1 Introduction

This chapter proposes an energy-efficient and fault-tolerant pulse switching protocol using a cell based localization method.

The concept of *multi-hop pulse switching* was first introduced in Chapter 4 for binary event detection and tracking applications. Although it offered a fundamental conceptual framework for pulse switching, its hop-angular localization abstraction relied on an assumption of stationary nature of the wireless transmission and propagation properties. This assumption may not hold in certain applications due to various kinds of time-varying shadowing and fading phenomena. This chapter adopts a novel sensor-cell based architecture that relies on a preset sensor cell structure for localization, and is not affected by non-stationary wireless transmission properties. In doing so, it achieves a network-wide uniform spatial resolution of event detection which was not possible in the approach used in HPS as described in Chapter 4. Furthermore, a series of new fault tolerance mechanisms are introduced in this chapter for dealing with situations in which event monitoring can continue after clusters of sensors are simultaneously damaged.

5.2 Network Model

5.2.1 Network Topology

As shown in Figure 5-1, a network may contain arbitrarily distributed sensors and a sink. The figure also shows geographically neighbor pre-defined sensor-cells (i.e., arbitrarily shaped

and placed) that individually represent event areas, and have unique *Cell-IDs*. Each sensor is pre-programmed with the *Cell-ID* of its own cell and with a list of *Cell-IDs* of all the geographically neighbor sensor cells. Each cell includes at least one sensor.

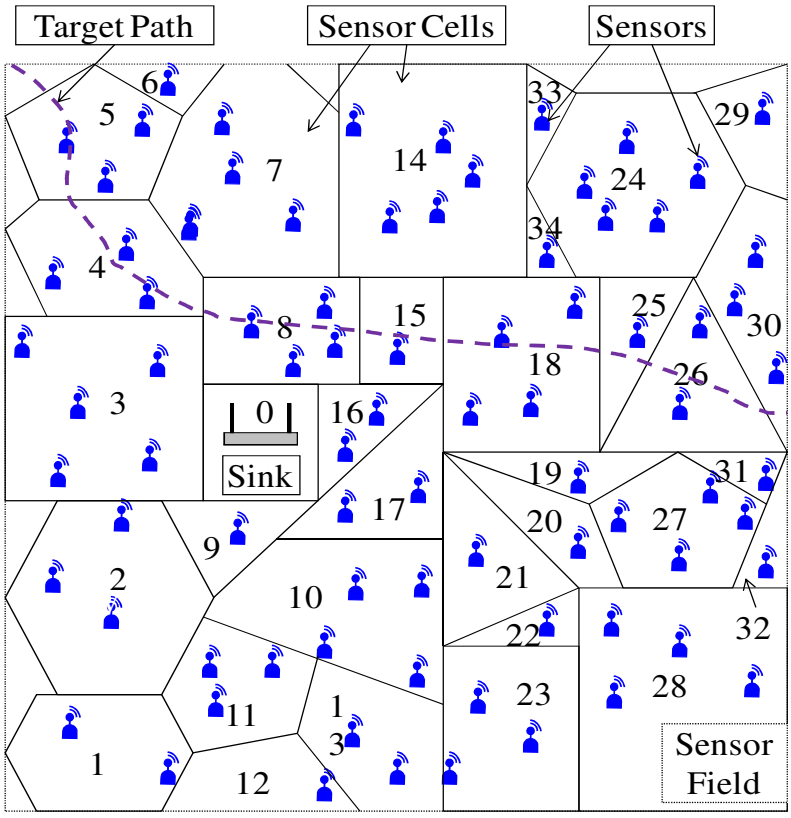


Figure 5-1: Network model with arbitrarily defined sensor-cells

5.2.2 Pulse as Protocol Data Unit

Upon detection of an event, a sensor node generates a single (RF) pulse (e.g., UWB-IR [33]) which needs to be transported multi-hop to a sink. A pulse is able to represent: a) the very occurrence of the event, and b) its location of origin. An event can result in multiple pulses generated by all sensors detecting the event, and all such pulses need to be transported to the sink. With the localization information for each such pulse, several application level conclusions can be derived at the sink by correlating multiple event pulses.

5.2.3 Cellular Event Localization

An event is localized at the spatial resolution of a sensor-cell. In a bridge monitoring scenario, for example, when one or multiple sensors within a cell detect a structural fatigue or failure, the sink node will eventually be able to localize the event in terms of the cell that those detecting sensors belong to. If sensors from multiple cells detect such an event, the sink can resolve the event spanning multiple such cells. The sensors are not individually addressed, and therefore no per-sensor addressing is necessary at the MAC or routing layers. An event is always identified by the *Cell-ID* of its cell of origin. The energy non-constrained sink is assumed to be capable of making high-power transmissions with full network coverage for frame-synchronizing the sensors.

Event localization resolution depends on the size and shape of the sensor cells. For example, with regular hexagonal cells with side length of l, the localization resolution is approximately $2.6l^2$ square units. By shrinking l, it is possible to improve the spatial resolution, which is independent of the wireless transmission range of the sensors. For arbitrarily shaped and size cells, the resolution will vary cell by cell.

5.3 Cellular Pulse Switching Architecture

5.3.1 Joint MAC-Routing Frame Structure

5.3.1.1 Frame Components

As in the hop-angular pulse switching, nodes in the proposed cellular pulse switching system are frame-by-frame time synchronized by the sink as well, and they maintain MAC-routing frames (see Figure 5-2), in which each slot is used for sending a single pulse. The slot

includes a guard time to accommodate the cumulative clock-drift during a frame and the maximum cell-to-cell propagation delay. Because of UWB-IR's very short frame size (*ms*), the clock-drift can be very small.

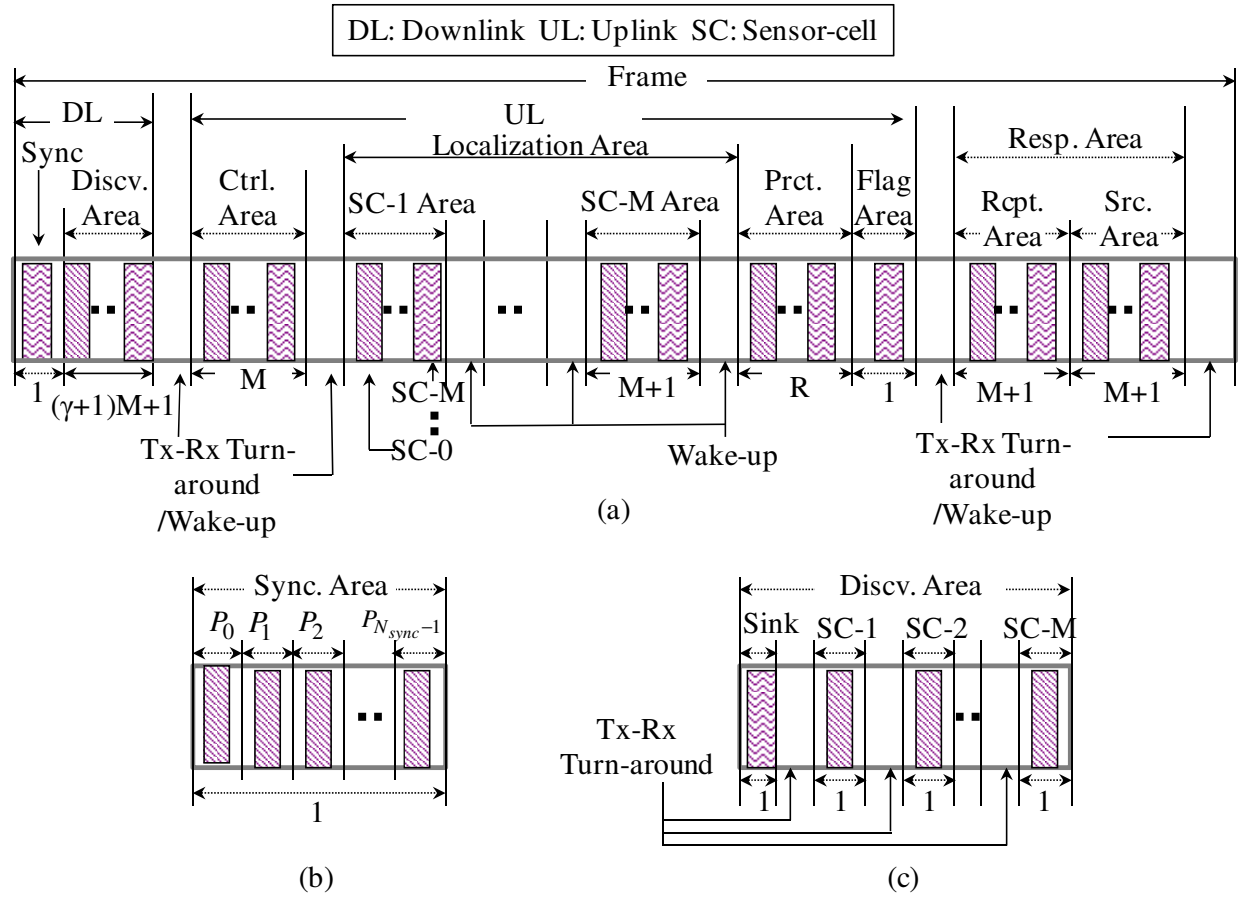


Figure 5-2: MAC-routing frame for cellular pulse switching

As shown in Figure 5-2, each frame contains a downlink part, an uplink part, and a response area in which pulses can be sent either downlink or uplink. The downlink part of the frame includes the *Sync Area* where the sink periodically transmits a full power pulse to frame-synchronize all nodes in the network. The following *Discovery Area* in the frame is designed for adaptive route discovery, as described in Section 5.3.3. In the uplink part, there are three components: a) *Control Area* for energy management, b) *Localization Area* for representing an

event's location of origin and the routing table information, and c) *Protection Area* for false positive error protection. The *Response Area* is used for implementing pulse transmission reliability. Operational details for each of these areas are presented in the next few sections.

5.3.1.2 Accommodation for propagation delay

Ideally, as shown in Figure 3-1 with zero propagation delay, a pulse arrives right in the middle of a time slot T_b. With propagation delay, as long as the deviation is within $T_b/2$ (half a time-slot), the protocol allows unambiguous interpretation of the received pulses. Due to the half-time-slot deviation corresponding to the round-trip propagation delay, the allowed one-way deviation is within $T_b/4$. With T_b set to 1000 *ns* and propagation speed 3×10^8 *m/s*, $T_b/4$ turns out to be approximately 75 *m*. This poses a restriction that the distance between any two nodes in two adjacent sensor cells needs to be less than 75 *m*, which is consistent with our small-network scope as stated in Section 1.5.

5.3.2 Global Frame Synchronization

The first area of a frame is termed as the *Sync Area* which is allocated to the sink for network synchronization. As shown in Figure 5-2b, the *Sync Area* is functionally divided into N_{sync} pulse positions. The sink sends a pulse pattern $\{P_0, P_1, ..., P_{N_{sync}-1}\}$ in these N_{sync} positions within the *Sync Area* ($N_{sync} \geq 3$). Let P_m ($m \in N_{sync}$) be a bit (i.e., *1* or *0*), representing a pulse transmission or not in the m^{th} pulse position. A pulse is always sent in the middle of a pulse position.

A pulse pattern is used in which P_m is set to be 1 for all *m*. When $N_{sync}=3$, the synchronization pulse pattern is '*111*'. The sink sends those three consecutive pulses in the *Sync Area* of every frame. Every node in the network keeps listening for F_{sync} times. If a node detects the specific pulse pattern within the same slot duration for at least $(F_{sync}-1)$ times, the node can obtain the starting time of each frame and the frame duration by observing the reception time of those F_{sync} pulse patterns. The starting time of the current frame is interpreted as $T_b/2N_{sync}$ duration earlier than the reception time of the first pulse within the synchronization pulse pattern, where T_b is the duration of a time slot. While adding certain protocol complexity, such synchronization enables each node finding the start time of the current frame without ambiguity.

5.3.3 Route Discovery

5.3.3.1 Route Discovery Logic

As shown in Figure 5-2c, the *Discovery Area* of a frame contains $(M+1)$ slots, where the transceiver turnaround time period (i.e., γT_b) is inserted between any two adjacent slots. In the *Discovery Area*, the first slot is allocated to the sink, and the rest of the slots are allocated to all *M* sensor-cells.

The discovery process is designed to be a low-frequency event which is periodically initiated by the sink node once in every T_{discv} frames. The sink initiates a routing discovery phase by sending a regular power (as opposed to a full power synchronization pulse) discovery pulse in the first slot of the *Discovery Area*. Upon receiving that pulse, all sensors within the

sink's neighbor sensor cells register the reception time and the *Cell-ID* of the sensor cell from which the discovery pulse was received. The *Cell-ID* is retrieved from the temporal location of the pulse, and the sensor cell is the sink in this case. The sensors then forward the pulse to their neighbors in the corresponding slots in the *Discovery Area* of the next frame. This time-stamp and forwarding process continues for T_{flood} frames till the discovery pulse is flooded throughout the entire network ($T_{flood} << T_{discv}$).

Formally stated, when a sensor in *cell m* receives a discovery pulse in the *SC-n* slot of the *Discovery Area*, the sensor checks if *cell n* is one of its neighbor cells. If not, it simply ignores the pulse. Otherwise, it registers the time of reception and the sender *cell n*, and forwards the pulse in the *SC-m* slot of the *Discovery Area* of the next frame.

5.3.3.2 Routing Table Formation

At the end of a discovery phase, each node develops a routing table which is a list of time-stamps and the corresponding *Cell-ID*s of its neighbor cells, indicating which sensor cells forwarded the discovery pulse at what time. The entry with the earliest time-stamp would indicate the shortest hop next-hop sensor cell to reach the sink. If multiple entries contain the same time-stamp then all the corresponding *Cell-ID*s would indicate shortest hop next-hops to the sink. A node may have one or multiple shortest hop next-hop sensor cells depending on the shape, size, and relative locations of the cells with respect to the sink's location.

Note that the above discovery process and the proposed architecture do not assume any specific shape of a sensor cell or node's transmission coverage area. Since sensors in the same cell all transmit the discovery pulses at the same time, neighbor cells would regard them as a single discovery pulse. The inherent discovery pulse aggregation helps to reduce the amount of

discovery pulse transmissions. Additionally, each sensor needs to execute the above discovery protocol for the following reason. In the absence of pulse errors (i.e., loss and false positives) all sensors in a cell will have the same routing table entry. In the presence of errors and faults, however, sensors within a cell may develop individually different routing table entries for optimal pulse forwarding to the sink. Therefore, each sensor needs to independently participate in the discovery process. Additional optimizations may be feasible with a traditional leader based mechanism. However, the protocol complexity and energy overhead of the leader management (i.e., election, maintenance, rotation etc.), especially without a packet abstraction, are not in line with the simplistic and energy-light operation of pulse switching.

Finally, like in packet based traditional sensor networks, the route discovery interval (i.e., T_{discv} frames) is dimensioned based mainly on the network dynamics, and can be kept quite high for relatively static networks. This will keep the energy overhead for route discovery relatively low when compared to that due to the pulse transportation process.

5.3.4 Pulse Forwarding

5.3.4.1 Pulses in Localization Area

The *Localization Area* of the frame is the key to pulse forwarding. In a network with *M* sensor-cells, this area contains *M Sensor Cell Areas*, where each *Sensor Cell Area* includes $(M+1)$ individual slots. Each *Sensor Cell Area* corresponds to a specific *Cell-ID* which represents an event's cell of origin. For example, when a node in *cell i* senses an event, it transmits a pulse in the *Sensor Cell Area i*. Within a given *Sensor Cell Area*, each slot corresponds to a specific *Cell-ID* which represents the next-hop cell for a transmitted pulse

within that area. The first slot in each *Sensor Cell Area* of the *Localization Area* represents a sink. For example, when a pulse is transmitted at the $(j+1)^{th}$ slot of the *Sensor Cell Area i*, it means that the pulse is originated within *cell i*, and is forwarded to the next hop *cell j*. *Cells i* and *j* in this case are assumed to be geographically adjacent. Also, if *cell j* is the best next-hop cell from *cell i* then *cell j* is expected to be geographically closer to the sink than *cell i*.

5.3.4.2 Pulse Forwarding Logic

It is important to note that the *Sensor Cell Area* for a pulse remains unchanged during the entire forwarding of the pulse from its sensor of origin all the way to the sink node. The localization information of the pulse (i.e., of the corresponding event) is preserved till the pulse arrives at the sink. What changes in a hop-by-hop basis is the specific slot within the *Sensor Cell Area* depending on the specific next hop sensor-cell. This way, when the pulse arrives at the sink, it is still a part of the unchanged *Sensor Cell Area*, which indicates the pulse's cell of origin.

For example, a pulse that is originated in *cell i* is always transmitted within the *Sensor Cell Area i* of successive frames during all the hops till the sink. If a pulse from *cell i* is forwarded first to *cell m* then the sensor of origin transmits the pulse in the $(m+1)^{th}$ slot of the *Sensor Cell Area i*. Now, if a sensor in *cell m* finds the next hop cell to be *cell n*, then it forwards the pulse in the $(n+1)^{th}$ slot of the *Sensor Cell Area i* in the following frame. Thus, when the pulse arrives at the sink, it is still a part of the *Sensor Cell Area i*. It informs the sink that the pulse originates from *cell i*.

Pulse forwarding decisions are made based on a sensor's routing table, which maintains a sorted list of next-hop cells based on the hop-counts of the corresponding resulting routes (see Section 5.3.3). For routing with no diversity, a node chooses the best next-hop sensor cell from the routing table and forwards the pulse. With route diversity (parameterized as δ) enabled, a pulse is forwarded to the top δ next hop sensor cells from the routing table for improving transport reliability.

5.4 Fault-tolerance

There are three possible causes for pulse losses: *1) node state*: during *Pulse Forwarding* (see Section 5.3.4), a node may fail to receive the pulse if it is erroneously in *transmission* or *sleeping* state in the *Localization Area* of a frame; *2) pulse loss*: even when in *listening* state, a node may lose a pulse due to pulse loss errors; and *3) faults*: when nodes in a certain area are energy depleted or destroyed due to external causes. The following *Response Mechanism* and *Prioritized Routing Table Lookup* are added to the baseline *Cellular Pulse Switching* (see Section 5.3) for handling errors and system faults.

5.4.1 Response Mechanism

As shown in Figure 5-2a, the *Response Area* of a frame contains *Reception a*nd *Source sub-areas*, where each sub-area includes $(M+1)$ slots. The *Reception sub-area* indicates the *Cell-ID* of a sensor cell which receives a pulse. The *Source sub-area* indicates the *Cell-ID* of the cell of origin of the event corresponding to the received pulse. The first slot in both *sub-areas* is allocated to the sink, and the rest of the slots are allocated to the *M* individual cells.

The response mechanism is abstracted as a single-pulse acknowledgement for implementing one-hop transmission reliability. After receiving a pulse in the *Localization Area*

of a frame, a receiver sends a *response pulse* to the sender during the *Response Area* of the frame. If the sender does not receive a *response pulse*, it retransmits the pulse in the following frames until a successful response pulse is received.

5.4.2 Prioritized Routing Table Lookup

In case of multiple node failures or faults within a sensor cell, the above response mechanism is unlikely to work. Since such faults can last for very long time (e.g., *hours*) compared to the frame duration (i.e., *ms*), a failed node in the fault area cannot send *response pulses* even after repeated retransmissions from its sender node. Using the *route discovery* process in Section 5.3.3, the sender node eventually discovers any existing alternate route that bypasses the faulty/failed neighborhood sensor cells. This discovery process, however, is slow and may take up to T_{discv} frames in the worst case.

The following *Prioritized Routing Table Lookup* is proposed for reducing rerouting latency after such cell-wide sensor faults as described above. As presented in Section 5.3.3, the routing table for a node is a sorted list of time-stamps and the corresponding *Cell-ID*s of its neighbor cells. Using the prioritized routing table lookup, a sender node first sends a pulse to top δ next hop sensor cells from the routing table, where δ is the chosen route diversity. If no response pulses are received in the current frame, the node will send pulses to the following δ next hop cells from the routing table in a next frame. Such a process of sending pulses to the next δ sensor cells continues till the routing table is exhausted, in which case the sender node wraps around the table and starts with the first δ entries.

5.4.3 Routing Loops

5.4.3.1 Example Scenario

Consider an example situation with the route diversity δ set to 1 (see Figure 5-3), in which the nodes in *cell m* send pulses towards the best next-hop *cell* n_1. Now consider that all the neighbor cells of *cell m* are under fault condition except *cell* n_6, which is farther from the sink than from *cell m* itself.

A sender node in *cell m* first transmits a pulse to its best next-hop *cell* n_1 and then waits for the response pulse. Since *cell* n_1 is in the fault area, the sender does not receive any response pulses. According to the prioritized routing table lookup in Section 5.4.2, the sender node in such situation retransmits the pulse to the following best next-hop *cell* n_i ($i \in [2,6]$) until it receives a response pulse. In this case, only nodes in *cell* n_6 are able to receive and respond to the event pulse from *cell m*. Now, it is possible that *cell m* is the best next-hop of *cell* n_6 since the routing table of *cell* n_6 is not updated yet after the mentioned faults. The node in *cell* n_6 sends the pulse back to *cell m* in the next frame, thus causing a routing loop. As long as the routing table of *cell* n_6 is not updated or the neighbor cells of *cell m* are not recovered from fault, the routing loop between *cell m* and *cell* n_6 will continue.

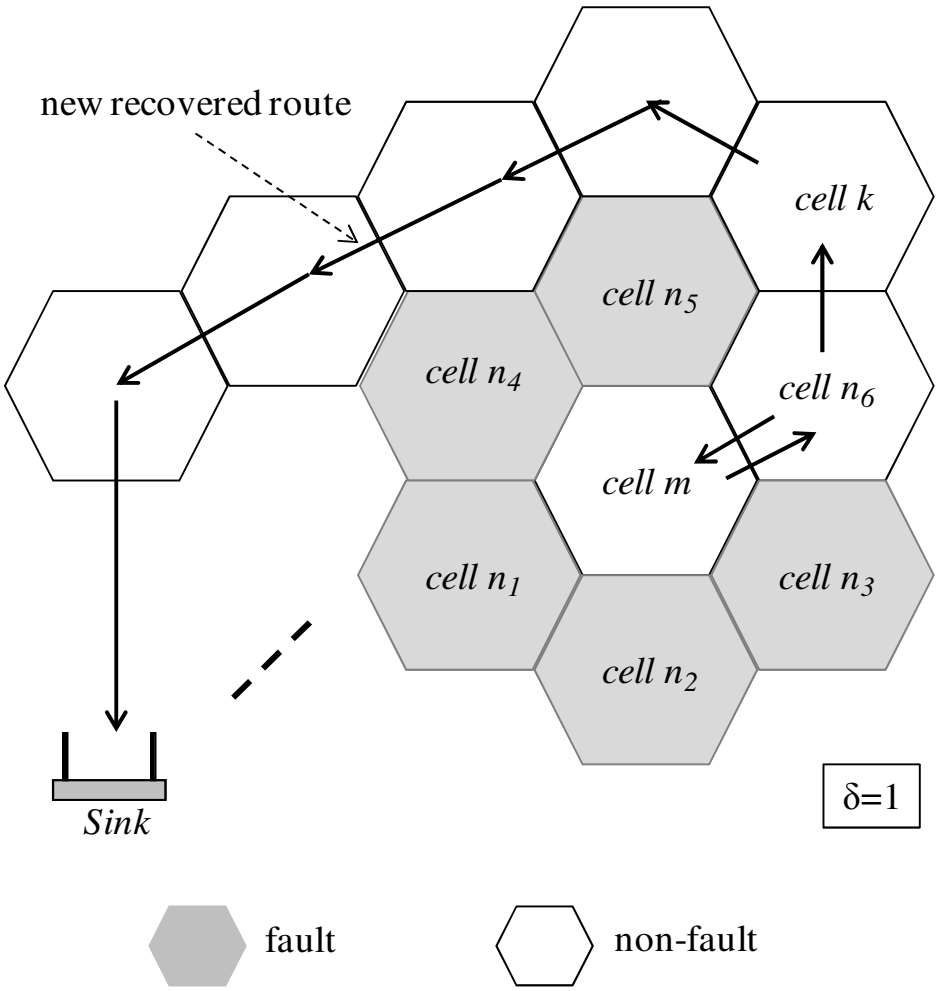


Figure 5-3: Example pulse routing loop

5.4.3.2 Loop Resolution

To resolve such loops, we add a one-slot *Flag Area* (see Figure 5-2a) in the end of the uplink sub-frame. Additionally, in the routing table of a node, the neighbor cells are divided into two groups, one nearer to the sink and the other farther from the sink with respect to the current cell. Before a node sends out an event pulse, it first checks if there is any next-hop cell belonging to the farther-from-the-sink group. If at least one of the δ next-hop cells belongs to that group, the sender transmits an additional pulse in the *Flag Area* of the frame along with the event pulse.

In other words, the flag should be added to a transmitted event pulse if one of the selected δ next-hop cells is farther away from the sink. For example, in Figure 5-3, the sender node in *cell m* transmits a flagged event pulse to *cell* n_6.

The receiver node in a next-hop cell receives the event pulse with flag and replies to the sender with a response pulse. Once the sender node receives a response pulse from any of the δ next-hop cells, it stops the retransmission process. For example, upon reception of a response pulse from *cell* n_6, the sender node in *cell m* stops retransmission processes. As for forwarding, after a node receives a flagged event pulse, it looks into its previous pulse transmission records within a pre-specified period. If the node already sent the same event pulse to certain δ next-hop cells, it retransmits this event pulse to the following δ next-hop cells from its routing table. Otherwise, the receiver sends the pulse to its best δ next-hop cells. In this way, the pulse routing loop can be resolved by leveraging all possible routing table entries.

For example, as shown in Figure 5-3, after a node in *cell* n_6 receives a flagged event pulse from *cell m*, by scanning its own transmission records it finds that most recently it had forwarded pulses to *cell m* as the best next-hop cell. So the node in *cell* n_6 chooses the second best next-hop cell n_3 for forwarding the event pulse in a next frame. According to the prioritized routing table lookup, the node in *cell* n_6 finally retransmits the event pulse to *cell k* in one of the future frames. This is because except *cell m* and *cell k,* all the neighbor cells of *cell* n_6 are in-fault condition. Similarly, nodes in *cell* n_6 and *cell k* also need to send flagged event pulses to their corresponding next-hop cells along a newly recovered route. Eventually, the event pulse is successfully delivered to the sink.

If the node in *cell* n_6 had not transmitted the same event pulse to the best next-hop *cell m* in recent past, it sends the event pulse back to *cell m*. Then the retransmission process from *cell m* to its neighbor cells restarts. Nodes in *cell* n_6 receive the same event pulse from *cell m*. Afterwards, nodes in *cell* n_6 would choose *cell k* as the next-hop cell following the same process as described above.

While the prioritized routing table lookup with loop resolution is utilized as above, the route discovery process continues in parallel in order to find alternate routes bypassing faulty sensor cells. Transmissions using the prioritized table provide an interim mechanism for pulse transmissions – thus avoiding long network routing downtime during large scale sensor faults.

5.5 Summary and Conclusion

For ultra light-weight networking applications in WSNs, a novel cellular pulse switching (CPS) protocol with high energy efficiency and strong reliability has been developed in this chapter. The key contribution of this chapter is to combine a cellular event localization mechanism with a pulse switching protocol in a manner that allows a receiver to localize an event by observing the temporal position of a received pulse with respect to a synchronized frame structure. Unlike HPS in chapter 4, the proposed cellular event localization mechanism is naturally immune to the side effects of node cooperation caused by pulse merging. This is due to its complete insensitivity to the wireless transmission range. In addition to the detailed demonstrations of the joint MAC-routing frame structure and the processes of global frame synchronization, route discovery and pulse forwarding, this chapter highlighted a set of fault-tolerance mechanisms. Such mechanisms include a response syntax, prioritized routing table

lookup method, and loop resolution technique. They all work together to ensure the strong reliability of CPS against errors and faults.

Chapter 6: Point-to-point Pulse Switching Protocol

6.1 Introduction

Unlike multi-point-to-point communications in Chapters 4 and 5, Chapter 6 presents an energy-efficient and fault-tolerant point-to-point pulse switching (PTP) protocol with a cell based event localization.

PTP adopts the same cellular event localization method as used in the CPS protocol. Compared to only one sink node in CPS, the destination in PTP is extended to every cell in the network. So the transported localization information should contain both the *Cell-IDs* of an event's origin and destination. Accordingly, the complexity of the joint MAC-routing frame structure for PTP is higher compared to that for CPS. In this chapter, we propose a new MAC-routing structure to handle such complexity.

6.2 Network Model

6.2.1 Network Topology

As shown in Figure 6-1, a network may contain arbitrarily distributed sensors and actuators. The figure also shows geographically neighboring cells (i.e., arbitrarily shaped and placed) that are pre-defined and individually represent event areas with unique *Cell-IDs*. Each cell contains either sensor nodes or actuator nodes, but not both. A node should have wireless reachability to at least another node in one of its neighboring cells. A cell containing sensors is called *Sensor Cell*, whereas a cell containing actuators is named *Actuator Cell*. Additionally, each sensor (*OR* actuator) is pre-programmed with the *Cell-ID* of its own cell and with a list of

Cell-IDs of all the geographically neighboring cells. Moreover, each sensor (*OR* actuator) is pre-programmed with the knowledge of the *Cell-IDs* of its corresponding dedicated actuator (*OR* sensor) cells. Note that, the design of the network size and network clustering is related to the detection and communication ranges of the sensors and actuators, and the specific application requirements.

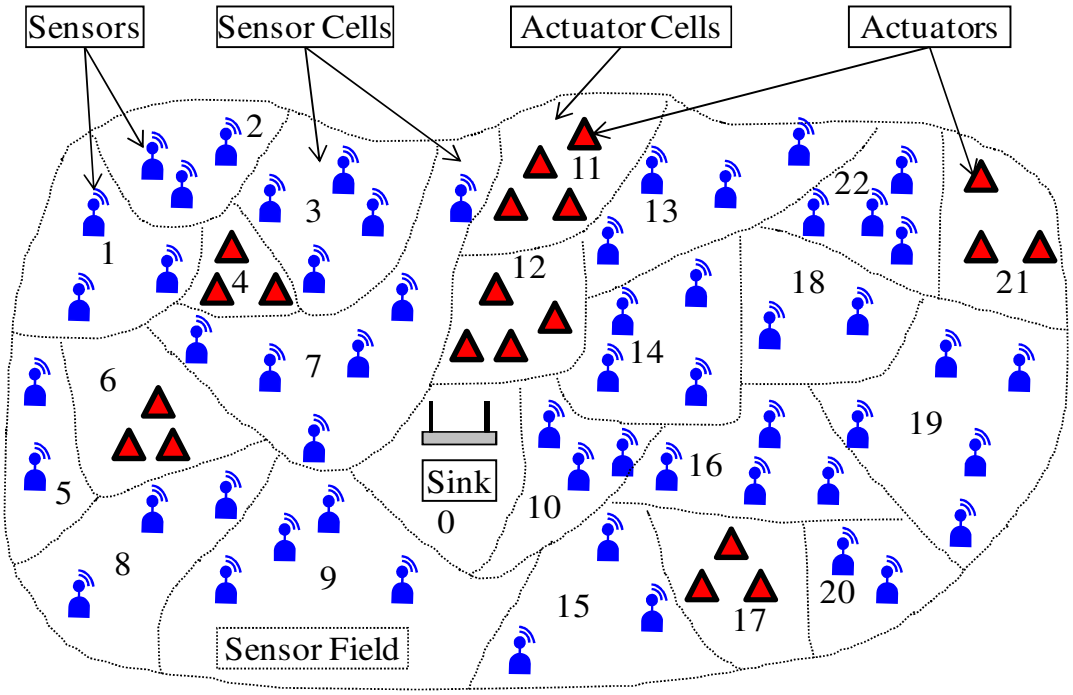


Figure 6-1: Network model with arbitrarily defined sensor and actuator cells

6.2.2 Pulse as Protocol Data Unit

Upon detection of an event, a sensor node generates a single (RF) pulse (e.g., UWB-IR [33]) which needs to be transported multi-hop to its destined actuator cell. An event pulse is able to represent: a) the very occurrence of the event, b) the *Cell-ID* of the event's source sensor cell, c) the *Cell-ID* of the present next-hop cell, and d) the *Cell-ID* of the event's destination actuator cell.

6.2.3 Cellular Event Localization

An event is localized at the spatial resolution of a cell. In a fire extinguishing scenario, for example, when one or multiple sensors within a cell detect a fire occurrence, the nearest water sprinkler actuators in another cell is able to timely localize the fire event in terms of the cell that those detecting sensors belong to, and then extinguish the fire accordingly. If sensors from multiple cells detect such a fire event, the actuators can resolve the event spanning multiple such cells.

Same as the cellular localization method in CPS as shown in Section 5.2.3, no per-node addressing is necessary at the MAC or routing layers. In particular, the sensor or actuator nodes are not individually addressed. An event is always identified by the *Cell-ID* of its cell of origin, and not the node of origin. Note that the proposed architecture in this chapter does not assume geometrically regular cells. As shown in the Figure 6-1, irregular cells of different shapes and sizes can co-exist.

Event localization resolution depends on the size and shape of the event area cells. Same as in CPS, with regular hexagonal cells with side length of l, the localization resolution is approximately $2.6l^2$ square units. By shrinking the side length l, it is possible to improve the spatial resolution, which is independent of the wireless transmission range of the sensors and that of the actuators. For arbitrarily shaped and size cells, the resolution will vary on a cell by cell basis.

6.3 Point-to-point Pulse Switching Architecture

6.3.1 Joint MAC-routing Frame Structure

Sensor and actuator nodes in the proposed point-to-point system are frame-by-frame time synchronized, and they maintain MAC-routing frames (see Figure 6-2), in which each slot is used for sending a single pulse. The slot includes the guard time to accommodate the cumulative clock-drift within a frame, as shown in Figure 3-1. Because of UWB-IR's very short frame size, any hardware clock-drift is expected to be very small.

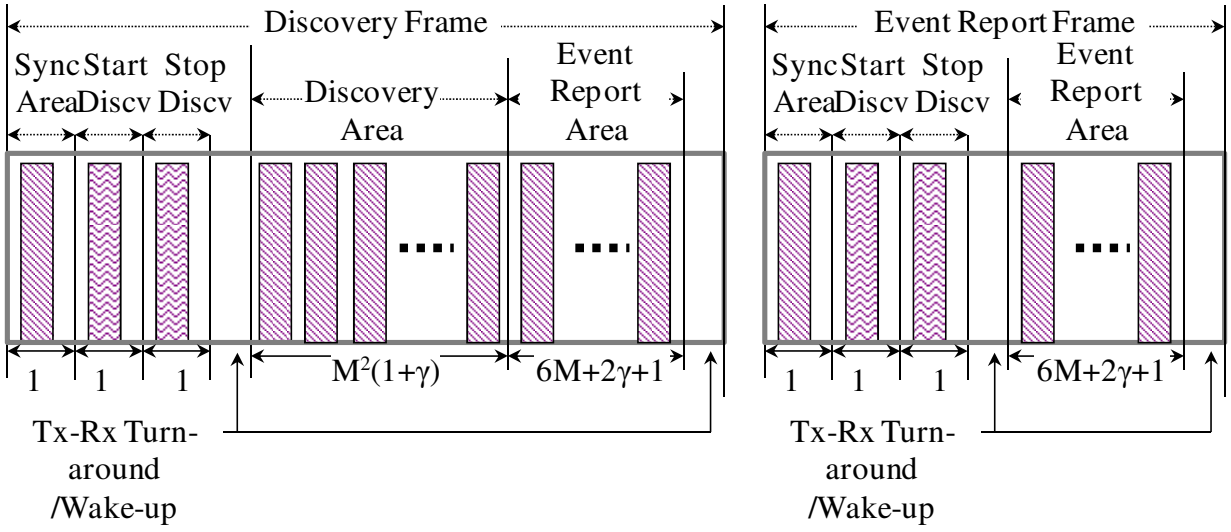


Figure 6-2: MAC-routing frame for point-to-point pulse switching

As shown in Figure 6-2, there are two types of frame structures in terms of functionality: *Discovery Frame* and *Event Report Frame*. The *Discovery Frame* is designed for adaptive route discovery. Considering the occasional changes of network connectivity and the resulting topology, the route discovery phase only needs to be executed periodically with a very low frequency. In case of any event occurrence during the route discovery phase, we also add the

Event Report Area in the end of *Discovery Frame*, which has the same functionalities as that in the regular *Event Report Frame*. The *Event Report Frame* is responsible for reliably transporting the localization information of an event from its location of origin to a specified destination. Usually, the nodes execute the *event report phase* after the *route discovery phase*.

Both types of frames start with the *Sync Area*, *Start Discv Area*, and *Stop Discv Area*, which are all reserved for the sink. Specifically, *Sync Area* is reserved for the sink to transmit a full power pulse pattern to frame-synchronize all nodes in the network. The following *Start Discv Area* and *Stop Discv Area* are reserved for the sink to send a full power pulse to activate or terminate the route discovery phase in the network (see Section 6.3.2). In other words, the above two areas are used for notifying nodes of the type of the current frame. A node begins using the *Discovery Frame* type upon the reception of a pulse in the *Start Discv Area*. After receiving a pulse in the *Stop Discv Area*, the node switches to the *Event Report Frame* type. Operational details for each area in both types of frames are presented in the following sections.

There are two ways to execute the synchronization for the proposed point-to-point pulse switching system: 1) Global Positioning System (GPS) synchronization, or 2) Global Frame Pattern (GFP) synchronization, which is same as the synchronization in CPS (see Section 5.3.2). For the GPS synchronization, each node needs to be equipped with a GPS receiver, where the accuracy can be 14 *ns* [42]. As discussed in Chapter 3, the time slot duration in the proposed system is 1000 *ns*.

6.3.2 Route Discovery

To accommodate the occasional changes in network connectivity and the resulting topology, the route discovery protocol needs to be periodically executed as a background process with a very low frequency, thus causing very low energy overheads.

6.3.2.1 Initiation and Termination

The sink initiates a route discovery phase by sending a full power pulse to all the nodes in the *Start Discv Area* of a frame. Then each node in the network selects the *Discovery Frame* and begins to discover the routing tables for each cell as a destination. This discovery process continues for $M.c$ $(0 < c < 1)$ frames which are needed for all the network nodes to complete the discovery of routing tables towards all the network cells. After $M.c$ frames, the sink ends the route discovery phase by transmitting another full power pulse in the *Stop Discv Area* of the frame. Then each node stops using the *Discovery Frame* and switches to the *Event Report Frame*. The sink usually waits for a relatively long time (F_{discv} frames) to restart the route discovery phase.

In this route discovery phase, the routing table in each node is created or updated in terms of the next-hop *Cell-ID*s destined towards every cell. In order for routing to work, each node needs to be aware of its own *Cell-ID* and those of all the geographically neighboring cells. The detailed route discovery phase is in the following text.

6.3.2.2 Discovery Area Details

As shown in Figure 6-3, the *Discovery Area* of the *Discovery Frame* includes *M Destination Cell Areas*, where each *Destination Cell Area* is responsible for a specific discovery

process rooted from the corresponding cell. Formally stated, the *Destination Cell Area i* of *Discovery Area* is assigned to the route discovery process in which all the nodes discover the specific next-hop cells destined towards *cell i* ($i \in [1, M]$). Furthermore, each *Destination Cell Area* contains *M Forwarding Cell Areas*. Every *Forwarding Cell Area* of a *Destination Cell Area* represents a specific *Cell-ID* of the cell which forwards the discovery pulse for this specific route discovery process. A *Forwarding Cell Area* is a slot during which discovery related transmission happens. The immediate following time after each *Forwarding Cell Area* is reserved for the transceiver turn-around time (between transmission and reception modes).

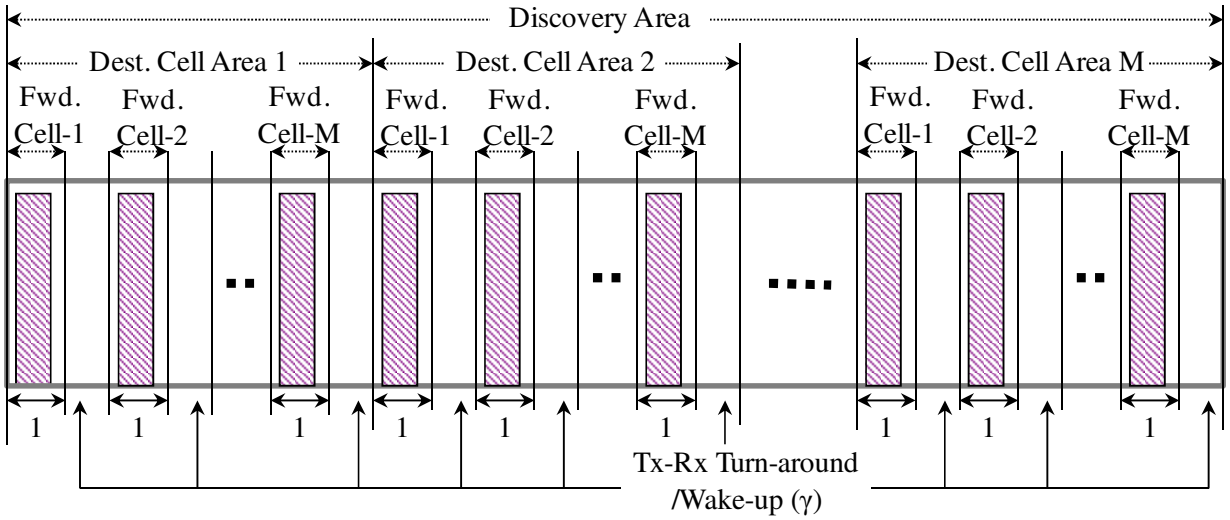


Figure 6-3: *Discovery Area* of a discovery frame for point-to-point pulse switching

6.3.2.3 Route Discovery Logic

Each node in a cell initiates a route discovery process in every discovery phase (F_{discv} frames). Specifically, in order to initiate a route discovery phase, a node within *cell i* sends a regular power discovery pulse in the *Forwarding Cell Area i* of the *Destination Cell Area i* in the *Discovery Area*. Upon receiving that pulse, nodes within the neighboring cells of *cell i* register

the reception time and the *Cell-ID i* for the route discovery process initiated by *cell i*. Then each of these receiving nodes forwards the pulse in the corresponding *Forwarding Cell Area* of the *Destination Cell Area i* of the *Discovery Area* in the next frame.

Formally stated, a node in *cell m* receives a discovery pulse in the *Forwarding Cell Area n* of the *Destination Cell Area i* in the *Discovery Area*. The receiving node checks if the cell of *Cell-ID n* is one of *cell m*'s neighboring cells. If not, the node simply ignores the pulse. Otherwise, the receiving node registers the time of reception and the *Cell-ID n* of the forwarding cell for the route discovery process initiated by *cell i*. Afterwards, this node in *cell m* forwards the pulse in the *Forwarding Cell Area m* of the *Destination Cell Area i* in the *Discovery Area* in the next frame.

Such a time-stamp and forwarding process rooted from *cell i* continues till the discovery pulse is flooded throughout the entire network. Since the above discovery process is executed on a per-*Destination-Cell-Area* manner, multiple discovery processes initiated by different cells can be simultaneously and independently routed in different *Destination Cell Areas* of the *Discovery Area*.

6.3.2.4 Routing Table Formation

After completing the discovery processes rooted from all *M* cells in the network, each node develops a routing table of *M* lists corresponding to *M* destination cells. Each list in a node for a specific destination cell includes time-stamps and the corresponding *Cell-ID*s of the node's neighboring cells, indicating which cells forwarded the discovery pulse at what time. The entry with the earliest time-stamp would indicate the best next-hop cell to reach the destination. If multiple entries contain the same time-stamp then all the corresponding *Cell-ID*s would indicate

best next-hops to the destination cell. For a specific destination, a node may have one or multiple best next-hop cells. It depends on the shape, size, and relative locations of the cells with respect to the destination. The fact that each node has wireless reachability to at least another node in one of its neighboring cells ensures at least one routing table entry to reach the destination.

Same as in CPS, the above discovery processes and the proposed point-to-point architecture do not assume any specific shape of a cell or node's transmission coverage area. Since all nodes in the same cell transmit the discovery pulses at the same time, neighbor cells would regard them as a single discovery pulse. The inherent discovery pulse aggregation helps to reduce the number of discovery pulse transmissions. Additionally, each node needs to execute the above discovery process for the same reasons as in CPS.

6.3.3 Pulse Forwarding

6.3.3.1 Pulses in Localization Area

The *Localization Area* of an *Event Report* frame is the key to pulse forwarding. In a network with *M* cells, this area (see Figure 6-4) contains three parts named *Source Area*, *Destination Area*, and *Forwarding Area* respectively. Each area contains *M* slots. Each slot in *Source Area* corresponds to a specific *Cell-ID* which represents an event's cell of origin. In *Destination Area*, every slot represents a specific *Cell-ID* of the event's destination. Each slot in *Forwarding Area* corresponds to a specific *Cell-ID* of a next-hop cell for a transmitted pulse.

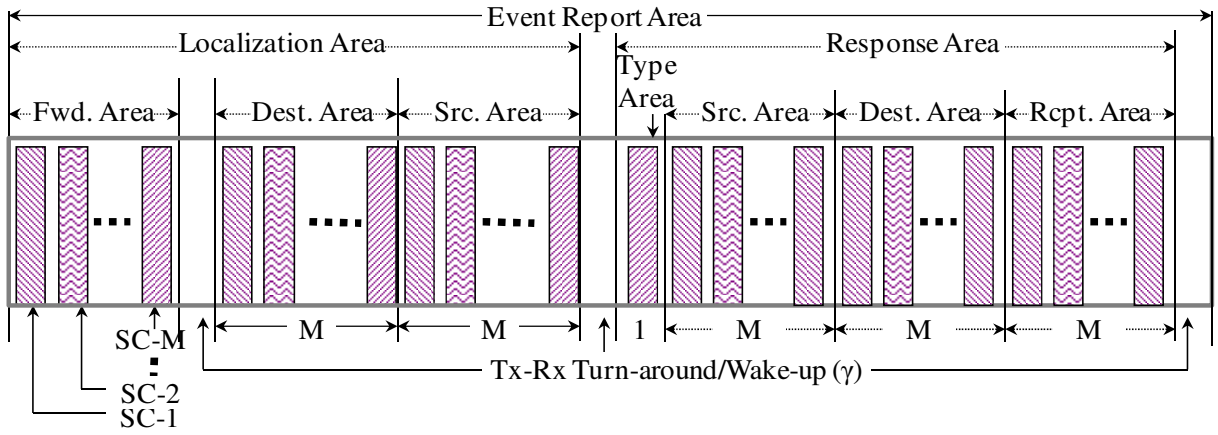


Figure 6-4: *Event Report Area* of a frame for point-to-point pulse switching

For example, a sensor node *A* in a cell with *Cell-ID i* senses an event which needs to be transported to actuator nodes in the dedicated actuator cell with *Cell-ID j* ($i \neq j, i, j \in [1, M]$). According to the routing table ($m \neq i, j \quad m \in [1, M]$), node *A* knows its next-hop is a cell with *Cell-ID m*. So node *A* transmits an event pulse by sending a pulse in the i^{th} slot of the *Source Area* corresponding to the *Cell-ID i*, a pulse in the j^{th} slot of the *Destination Area* corresponding to the *Cell-ID j,* and another pulse in the m^{th} slot of the *Forwarding Area* corresponding to the *Cell-ID m*. *Cells i* and *m* in this case are assumed to be geographically adjacent. Also, if *cell m* is the best next-hop cell from *cell i* then *cell m* is expected to be geographically closer to *cell j* than *cell i*. After pulse forwarding, a sender node records its transmitted pulse into its forwarding history in the form of (*source cell id, destination cell id, next-hop cell id*) for a time period of T_{record}. For instance, node *A* records the above transmission by (*i,j,m*). Note that, sensor and actuator nodes are both able to forward pulses as described above.

6.3.3.2 Pulse Forwarding Logic

Pulse forwarding decisions are made based on a node's routing table and an event's destination. The routing table maintains *M* sorted lists respectively for *M* cells as destinations. Among these *M* lists, a node selects one specific sorted list using the *Cell-ID* of an event's destination cell. Such a sorted list contains next-hop cells based on the hop-counts of the corresponding resulting routes with the dedicated destination. For routing diversity (parameterized as δ) equal to 1, a node chooses the best next-hop cell from the routing table and forwards the pulse. With $\delta > 1$, a pulse is forwarded to top δ next hop cells from the specific sorted list of the routing table.

Consider an example in which a node in *cell r* receives a pulse that was originated in *cell i* and destined towards *cell j* ($i \neq j, r \neq i, j \quad i, j, r \in [1, M]$). Upon receiving the pulse, the node checks the *Forwarding Area* to derive the *Cell-IDs* of cells which are allowed to forward the pulse at this frame. If these forwarding *Cell-IDs* include *Cell-ID r*, then the receiving node will forward the pulse in a next frame. Otherwise, the receiving node in *cell r* simply discards the pulse. With route diversity δ>1, the node finds the top δ next hop cells to be {*cell* n_k } ($n_k \neq j, \quad k \in [1, \delta]$) in the sorted list destined to *cell j* in the routing table. Thus, the node forwards the event pulse by sending a pulse in the i^{th} slot of the *Source Area*, a pulse in the j^{th} slot of the *Destination Area*, and additional δ pulses – respectively in the $\{n_k\}^{th}$ slots of the *Forwarding Area* ($k \in [1, \delta]$). Details are provided in Algorithm 6-1.

$Next(j,r,\delta)$: A node in *cell r* gets the top δ next-hop *Cell-IDs* $\{n_k\}, k \in [1,\delta]$ towards the destination *cell j* from the routing table $RT(r)$, where δ is the route diversity, $j \neq r, \quad n_k \neq j,r$.

$EVENT(i,j,n_1,...,n_\delta)$: An event pulse originated from *cell i* is forwarded to $\{cell \quad n_k\}, k \in [1,\delta]$, which is eventually destined to *cell j*, $i \neq j, \quad n_k \neq i,j$.

IF (*a node in cell i senses an event which needs to be sent to its destination cell j at frame t* **)**
    Sends an $EVENT(i,j,Next(j,i,\delta))$ at frame t;
END

IF (*a node in cell r receives an* $EVENT(i,j,n_1,...,n_\delta)$ *at frame t* **)**
    // cell r is one of the next-hops for this received event pulse
    IF ($r \neq i \;\&\&\; r \in \{n_1,...,n_\delta\}$ **)**
        IF ($r \neq j$ **)**
            //cell r updates the next hops
            Will forward the $EVENT(i,j,Next(j,r,\delta))$ at frame t+1;
        ELSE
            Successful event transmission at destination;
        END
    ELSE
        Discard the received $EVENT(i,j,n_1,...,n_\delta)$;
    END
END

Algorithm 6-1: Pulse forwarding in PTP

It is important to note that pulse locations within the *Source Area* and *Destination Area* for an event pulse remain unchanged during the entire forwarding of the event pulse. For example, an event pulse is originated in *cell i* and destined to *cell j* ($i \neq j \quad i,j \in [1,M]$). Such an event pulse is always transmitted by sending a pulse in the slot of the *Source Area* corresponding to the *Cell-ID i* and another pulse in the slot of the *Destination Area* corresponding to the *Cell-ID j* of successive frames during all the hops till the destination. This way, the localization information

of the pulse (i.e., of the corresponding event) is preserved till the pulse arrives at the destination. What changes in a hop-by-hop basis is the specific slot within the *Forwarding Area* depending on the specific next hop cell. For example, if the pulse from *cell i* is forwarded first to *cell m* (based on its routing table entry) then the node of origin transmits the pulse in the m^{th} slot of the *Forwarding Area* ($m \neq i, j \quad m \in [1, M]$). Now, if a node in the *cell m* finds the next hop cell to be of *Cell-ID n*, then it forwards the pulse in the n^{th} slot of the *Forwarding Area* in a next *Event Report* frame ($n \neq m, i, j \quad n \in [1, M]$).

6.4 Reliability Enhancement and Collision Handling

6.4.1 Motivation

A node would fail to receive the pulse due to pulse loss errors. It might make the destination cell unreported eventually. Besides pulse losses, there might be collision occurrences at the presence of multiple event transportations. That is to say, a node might receive different event pulses during the same *Localization Area* of a frame. As a result, the node is unable to differentiate the source-destination pairs corresponding to those events. For example, there are two different events happening at the same time as shown in Figure 6-5. One event is originated from *cell 3* destined towards *cell 5*, the other is originated from *cell 4* destined towards *cell 1*. So node *A* in *cell 3* sends a pulse to cell 5, which is forwarded by node *D* in *cell 2*, as shown in Figure 6-5a. Similarly, node *B* in *cell 4* transmits a pulse to *cell 1*, which is also routed by node *D* in *cell 2*, as shown in Figure 6-5b. Node *D*, as a node located in the intersection of two event routes, receives both event pulses merged in the same *Localization Area* of the frame (see Figure

6-5c). In this case, node *D* may derive the incorrect (*source*, *destination*) pairs for both events, for example (*cell 3*, *cell 1*) and (*cell 4*, *cell 5*).

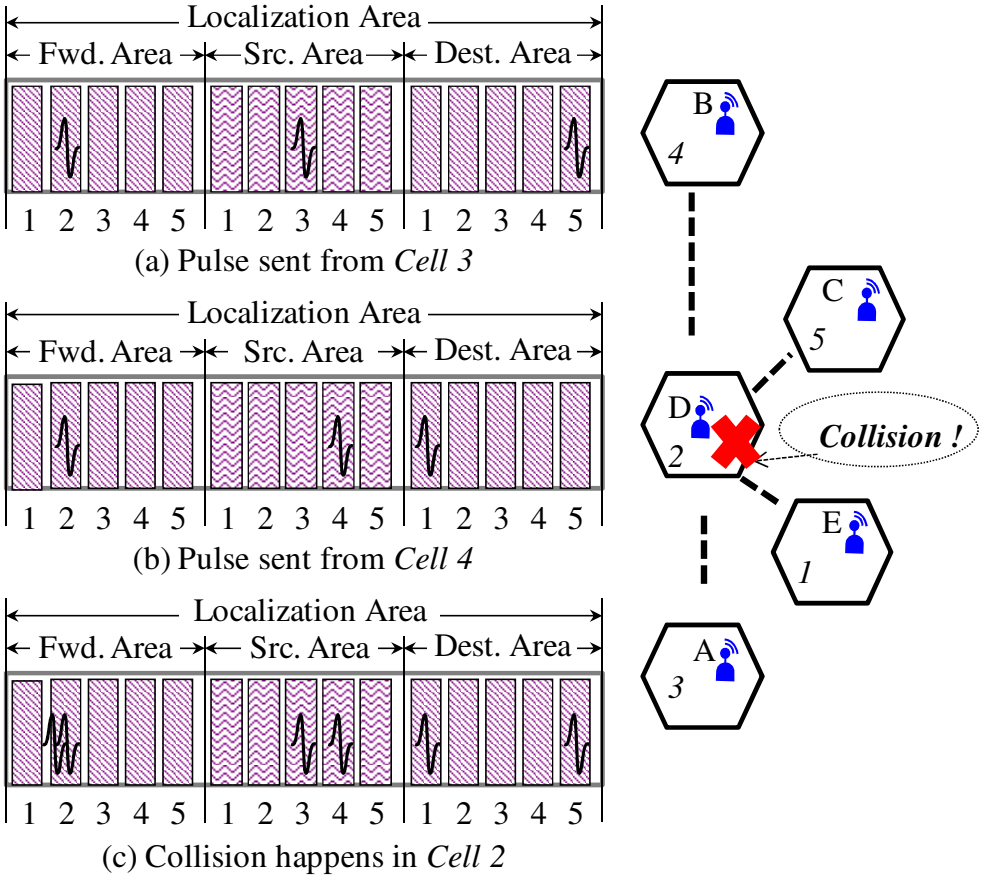


Figure 6-5: Example of pulse collisions

6.4.2 Response Mechanism

In order to enhance the system reliability and to resolve the multi-event collisions, we propose a *Response Mechanism* for PTP, in which a node notifies its sender node about pulse reception status with the specified *response pulse*.

As shown in Figure 6-4, the *Response Area* contains (*3M+1*) slots, where *M* is the total number of cells in the network. This area includes four parts: 1) *Type Area*, 2) *Source Area*, 3) *Destination Area*, and 4) *Reception Area*. *Type Area* notifies the response pulse type, which is

allocated in the first slot. There are two types of response pulses: 1) successful response pulse, and 2) collided response pulse. Specifically, no pulse transmission in *Type Area* corresponds to a successful response pulse, and a pulse transmission in this slot represents a collided response pulse. Additionally, the *Source Area*, *Destination Area* and *Reception Area* of the *Response Area* all contain *M* slots. Same as the *Source Area* and *Destination Area* of the *Localization Area*, each slot in the *Source Area* or *Destination Area* of the *Response Area* respectively corresponds to a specific *Cell-ID* of an event's original cell or a destination cell. Each slot in the *Reception Area* corresponds to a specific *Cell-ID* of the cell which receives a corresponding event pulse and then transmits a response pulse to the sender in the same frame.

Therefore, the *Response Mechanism* is abstracted as a single-pulse acknowledgement for implementing one-hop transmission reliability. After receiving one event pulse or multiple event pulses in an *Event Report Frame*, a receiver sends a specified response pulse to the sender during the *Response Area* of the frame (see Figure 6-4). If the sender does not receive a response pulse, it periodically retransmits the pulse in the following frames until receiving the corresponding *successful response pulse*. Otherwise, the sender checks whether a collision happened in the same frame by looking into the received response pulse. As a result, the sender handles the received response pulse differently depending on the collision occurrence or not. The details of handlings in both scenarios are provided in the following subsections.

6.4.2.1 Response without Collision

Upon receiving a response pulse, the sender node regards it as a *successful response pulse* when the response pulse satisfies the three conditions at the same time: 1) the *Type Area* has no pulse, 2) the *Source Area* only contains one pulse, and 3) the *Destination Area* only contains one

pulse during the *Response Area* of the current frame. Then the pulse transmission between the sender node and the receiver node is identified as a successful transmission.

$SUCC\_RESP(i, j, n)$: A successful response pulse sent from a node in *cell n* for an event originated from *cell i* and destined to *cell j*, $n \neq i, j$.
$Tx(m) = \{(src, dest, nexthop)\}$: A transmitted event pulse list at a node in *cell m*, where each item includes *Cell-IDs* of a source cell, a destination cell and at least one next-hop cells.

IF ($t == kT_{record}, k \geq 0$) *//every* T_{record} *frames*
    Resets $Tx(m), m \in [1, M]$;
ELSE IF (*a node in cell m forwards an* $EVENT(i, j, n_1, ..., n_\delta)$ *at frame t*)
    Inserts an item $(i, j, n_1, ..., n_\delta)$ into $Tx(m)$ at frame t;
END

IF (*a node in cell n receives an* $EVENT(i, j, n_1, ..., n_\delta)$ *at frame t*)
    // cell n is one of the next-hops for this received event pulse
    IF ($n \neq i$ && $n \in \{n_1, ..., n_\delta\}$)
        Sends a $SUCC\_RESP(i, j, n)$ at frame t;
    ELSE
        Discard the received $EVENT(i, j, n_1, ..., n_\delta)$;
    END
END

IF (*a node in cell m receives a* $SUCC\_RESP(i, j, n)$ *at frame t*)
    // cell m used to send the $EVENT(i, j, n)$
    IF ($Tx(m) \neq NULL$ && $(i, j, n) \in Tx(m)$)
        Transmission of the $EVENT(i, j, n_1, ..., n_\delta)$ between
            cell m and *cell n* is successful;
    ELSE
        Will retransmit the $EVENT(i, j, n_1, ..., n_\delta)$ at frame t+1;
    END
END

Algorithm 6-2: Response mechanism without collisions in PTP

For example, a node in *cell n* receives a pulse (*EVENT*) forwarded by a node in *cell m*, which is originated from *cell i* and destined towards *cell j* (*i,j,m,n* are not equal with each other). Besides, the sender node in *cell m* keeps the record of its transmitted pulse in the forwarding history in the form of (*i,j,n*) for T_{record} frames. In the same frame, the receiving node in *cell n* replies the forwarding *cell m* with a *successful response pulse* named *SUCC_RESP*. It includes a pulse in the i^{th} slot of the *Source Area* and a pulse in the j^{th} slot of the *Destination Area* and another pulse in the n^{th} slot of the *Reception Area* of the *Response Area*. The sender node would make a non-retransmission decision if two conditions are both satisfied: 1) the sender node in *cell m* receives this *SUCC_RESP* and no other response pulse merged into the same frame, and 2) the sender used to transmit the event pulse corresponding to this *SUCC_RESP* by checking its forwarding history. See the pseudo-code in Algorithm 6-2.

6.4.2.2 Response with Collision

Upon receiving a response pulse, the sender node regards it as a *collided response pulse* when the response pulse accords with at least one of three conditions: 1) the *Type Area* has a pulse, 2) the *Source Area* contains multiple pulses in different slots, or 3) the *Destination Area* contains multiple pulses in different slots during the *Response Area* of the current frame. Then the sender node would randomly select a number $r_{collision}$ and wait for $r_{collision}$ frames to retransmit the same event pulse. The random back-off range is constrained between 1 and R_{rnd}. Such retransmissions would continue as long as collisions are not resolved. The pseudo-code is provided in Algorithm 6-3.

$COLL\_RESP(flag,\{i_k\},\{j_k\},n)$: A collided response pulse sent from a node in *cell n* for events originated from $\{cells \quad i_k\}$ and destined to $\{cells \quad j_k\}$, $n \neq i_k, j_k \quad k \in [0,M]$.

IF (*a node in cell n receives multiple events* $\{EVENT_k\}$ *merged together at frame t,* $k > 1$ **)** *// a collision occurs*

flag = true;

Sends a $COLL\_RESP(flag,\{i_k\},\{j_k\},n)$ at frame t, $k \in (1,M]$;

END

IF (*a node in cell m receives a* $COLL\_RESP(flag,\{i_k\},\{j_k\},n)$ *at frame t* **)**

// cell m used to send the $Event(i_1, j_1, n)$

IF ($Tx(m) \neq NULL \&\& \ (i_1, j_1, n) \in Tx(m)$ **)**

//select a random number between 1 and R_{rnd}

$t^1_{backoff} = rand()\%R_{rnd} + 1$;

Retransmit the $Event(i_1, j_1, n1_1,...,n1_\delta)$ in $t^1_{backoff}$ frames;

ELSE

Discard the received $COLL\_RESP(flag,\{i_k\},\{j_k\},n)$;

END

END

Algorithm 6-3: Response mechanism with collisions in PTP

For example, a node in *cell n* receives two different event pulses, named *Eevent1* and *EVENT2*, in a frame. In details, *EVENT1* is forwarded by a node in *cell m* which is originated from *cell* i_1 and destined towards *cell* j_1 (i_1, j_1,*m,n* are not equal to each other). Additionally, *EVENT2* is forwarded by a node in *cell r* which is originated from *cell* i_2 and destined towards *cell* j_2 (i_2, j_2,*m,n* are not equal to each other). It causes a collision at *cell n* according to the above collision condition. In the same frame, the receiving node in *cell n* replies the forwarding *cell m* and *cell r* with a *collided response pulse* named *COLL_RESP*. It includes a pulse in the

Type Area, two pulses in the $(i_1)^{th}$ and $(i_2)^{th}$ slot of the *Source Area*, two pulses in the $(j_1)^{th}$ and $(j_2)^{th}$ slot of the *Destination Area* and another pulse in the n^{th} slot of the *Reception Area* of the *Response Area* ($i_1 \neq i_2, j_1 \neq j_2$). If a sender node in *cell m* receives this *COLL_RESP* in the same frame, it would retransmit *EVENT1* pulse after a randomly-selected number of frames $r^1_{collision}$. Similarly, if a sender node in *cell r* receives the *COLL_RESP* in the same frame, it would retransmit *EVENT2* pulse after a randomly-selected number of frames $r^2_{collision}$. If $r^1_{collision} \neq r^2_{collision}$, the sender node in *cell m* would receive *SUCC_RESP1* in $r^1_{collision}$ frames, and the sender node in *cell r* would receive *SUCC_RESP2* in $r^2_{collision}$ frames. Otherwise, the back-off retransmission process above continues until $r^1_{collision} \neq r^2_{collision}$ or timeout. In other words, the collision at *cell n* is resolved by letting sender nodes in *cell m* and *cell r* retransmit event pulses in different time within the time-out period.

The above response mechanism with collision also works for multiple event pulses. Upon receiving its dedicated collided response pulse, each sender node retransmits its specific event pulse after a randomly selected back-off time period.

6.5 Summary and Conclusion

Unlike multi-point-to-point pulse switching protocols in Chapters 4 and 5, this chapter developed an energy-efficient and fault-tolerant point-to-point pulse switching (PTP) protocol with a cell based event localization. The key contribution of the presented architecture in this chapter is to combine the cellular event localization with a distributed pulse switching protocol in a manner that allows a receiver to localize an event by observing the temporal position of a

received pulse with respect to a synchronized frame structure. In addition to the network model of PTP, detailed demonstrations of the joint MAC-routing frame structure, route discovery and pulse forwarding were given. This chapter also emphasized a reliability enhancement and collision handling mechanism. Such mechanisms help nodes successfully deliver pulses to a destination actuator from a source sensor at the presence of errors or multi-event collisions.

Chapter 7: Energy Saving Measures

7.1 Introduction

This chapter presents several architectural measures to improve the energy-efficiency of the HPS, CPS and PTP protocols.

7.2 Protocol State Machine

In both CPS and PTP, each node maintains three separate state machines for *Route Discovery*, *Pulse Forwarding* and *Response* processes respectively. A state is defined in a per-frame manner and can be one of *Transmission* (*T*), *Listening* (*L*), or *Sleeping* (*S*).

7.2.1 Cellular Pulse Switching

During *Route Discovery* and *Pulse Forwarding* processes, the initial state of each node is *L* and a state transition is triggered by pulse reception. During *Route Discovery* process, a node switches from *L* to *T* in the corresponding *Discovery Area* slot which is assigned to the node's *Cell-ID*. After forwarding the received discovery pulse in that slot, the node switches back to *L* during the upcoming *Discovery Area* slots of the same frame. This way, the node can receive discovery pulses from all its neighbor cells. During the *Pulse Forwarding* process a node maintains the *T* state in the *Localization Area* of the current frame. Once pulse forwarding is done, the node switches back to *L* in the *Localization Area* of the next frame. The state of a node in the *Response Area* is complementary to that in the *Localization Area* in a frame. The usage of state *S* for energy saving is described in the intra-frame interface shut-down mechanism in Section 7.4.2.

7.2.2 Point-to-point Pulse Switching

As in CPS, during the *Pulse Forwarding* process, a node maintains the *T* state in the *Localization Area* of the current frame. Once pulse forwarding is done, the node switches back to *L* in the *Localization Area* of the next frame. The state of a node in the *Response Area* is complementary to that in the *Localization Area* in a frame. The usage of state *S* for energy saving is described in the intra-frame interface shut-down mechanism in Section 7.4.3.

In route discovery processes, each node has three types of slot-sets within a *Discovery Frame* in terms of state selection: 1) *L*-set where a node always keeps *L* state, 2) *T*-set where a node might be in *T* state, and 3) *S*-set where a node might be in *S* state. Specifically, the *L*-set of a node includes the *Forwarding Cell Areas*, corresponding to the node's neighboring cells, of each *Destination Cell Area* in the *Discovery Area*. The *T*-set of a node contains *Forwarding Cell Areas*, corresponding to the node's own cell, of each *Destination Cell Area* in the *Discovery Area*. The *S*-set of a node includes the *Forwarding Cell Areas* within the *T*-set where the node does not need to forward pulses, and slots not belonging to either the *L*-set or the *T*-set.

When a node receives a discovery pulse in one of the *Forwarding Cell Areas* in its *L*-set (corresponding to a specific *Destination Cell Area*), it switches to *T* state in the *Forwarding Cell Area* in its *T*-set (corresponding to a specific *Destination Cell Area*) in a next frame to forward the discovery pulse. After the forwarding complete, the node switches back to *L* (OR *S*) state in the upcoming *Forwarding Cell Area* of the same frame if this slot falls into its *L*-set (OR *S*-set). The *L*-set ensures that a node is able to receive route discovery pulses forwarded by nodes only in its neighboring cells. Additionally, the *S*-set helps a node to save much energy during the route discovery phase.

For example, during the discovery process rooted from *cell i*, when a node in *cell m* sends out a discovery pulse ($i \neq m, \quad i,m \in [1,M]$), the node transits from *L* state to *T* state in the *Forwarding Cell Area m* of the *Destination Cell Area i* of the *Discovery Area* of a frame. This node changes its state to *L* (OR *S*) state in the upcoming *Forwarding Cell Area* of the same frame if this slot falls into the *L*-set (OR *S*-set). It is similar for any other discovery processes initiated by cell *j* ($j \neq i,m, \quad j \in [1,M]$).

7.3 Transmission Energy Saving Measures

Theoretically, only one pulse is necessary to notify an event's occurrence for binary sensing applications. However, since each node in a route participates in pulse forwarding (as discussed in HPS, CPS and PTP), there is a certain amount of pulse transmission redundancy during event reporting. This section investigates the methods to reduce such pulse transmission redundancies.

7.3.1 Parameterized Routing Control Method

7.3.1.1 Sector-Constrained Routing in Hop-angular Pulse Switching

Angle based filtering in HPS can be activated so that the forwarding of a pulse remains constrained within a pre-defined number of sectors around that of its origin. While higher *sector-constraints* (i.e., stricter filtering) curtail route diversity and subsequent pulse duplications leading to better energy economy, they also reduce the error tolerance due to lack of pulse duplications.

The extent of *sector-constraints* during wave-front routing can be parameterized using δ, which represents the ratio of the angular resolution α and an angle γ. The quantity γ is the

sector-width beyond which a pulse may not be flooded while routing. For a given α, the minimum and the maximum values of γ are α and 180° respectively. The corresponding δ values are *1* and $180°/\alpha$. When δ is *1*, routing is maximally constrained, indicating the minimum communication energy consumption, and the maximum susceptibility to errors due to the minimum route diversity β.

Figure 7-1 shows an instance of routing in which a pulse is routed from the source to the sink when α is 30°. The dark dots represents nodes that have actively received (was in the *L* state and received) the pulse. The spatial extent of flooding is shown with the *sector constraint* δ set to 0.2, 0.33 and 1.

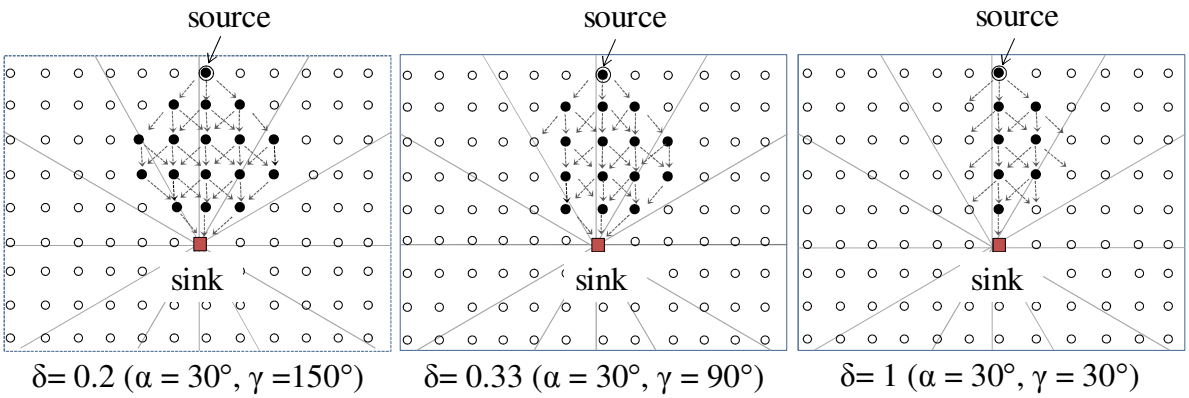


Figure 7-1: Wave-front routing envelope due to pulse fan-out in HPS

7.3.1.2 Reducing Route Diversity in Cellular Pulse Switching and Point-to-point Pulse Switching

In CPS and PTP, the route diversity δ is defined to be the number of next hop cells where a sender node needs to forward a pulse. When $\delta = 1$, a node chooses the best next-hop cell from the routing table and forwards the pulse. With $\delta > 1$, a pulse is forwarded to top δ next hop cells

from the specific sorted list of the routing table. In both CPS and PTP, we can save the transmission energy by choosing a smaller route diversity δ. The minimum value of δ is 1.

7.3.2 Pulse Compression Algorithm

Pulse compression can be applied to all three architectures, namely HPS, CPS and PTP. It includes spatial compression and temporal compression. In this section, an event area in HPS, a sensor cell in CPS, and a sensor or actuator cell in PTP are all named "an event area".

7.3.2.1 Spatial Compression

In order to describe the spatial compression algorithm generally, we define a compression area to be an area where spatial compression can be executed. It can be a hop area in HPS, and an area of sensor cells sharing a next-hop cell in CPS and PTP. In addition, such a compression area can also be an event area in all three protocols.

7.3.2.1.1. Spatial Compression Logic

Although multiple sensors can generate (or forward) pulses for the same event within a compression area, ideally only one such pulse from that area should be forwarded to the destination (i.e., sink in HPS and CPS and actuators in PTP) to inform about the event. Spatial pulse compression accomplishes this as follows. Upon detecting an event (or receiving a pulse), a sensor node-*j* defers its pulse generation (or forwarding) for a back-off period that is randomly distributed between 0 to R_{rnd} frames. If another node within the same compression area generates (or forwards) a pulse before its deferring period is over, then node-*j* cancels its own transmission after hearing that pulse. Typically, this would lead to very few pulses (often a single pulse) generation (or forwarding) per event per compression area.

7.3.2.1.2. Spatial Compression Model

Let k ($1 \le k \le N^{node}$) be the number of nodes in a compression area that select the same smallest number and the rest $\left(N^{node} - k\right)$ nodes choose greater numbers, where N^{node} is the number of nodes in the compression area. Thus, k forwarding transmissions will occur within this compression area. The rest $\left(N^{node} - k\right)$ nodes receive those k pulses and stop their own back-off processes. The spatial compression factor f_{comp}^{sp} is defined to be equal to the quantity R_{rnd}, which is the maximum value of a random back-off period. The probability that k nodes select the same smallest number can be expressed as:

$$p\left(k, N^{node}, f_{comp}^{sp}\right) = \begin{pmatrix} N^{node} \\ k \end{pmatrix} \sum_{m=1}^{f_{comp}^{sp}-1} \left(f_{comp}^{sp} - m\right)^{N^{node}-k} \Big/ \left(f_{comp}^{sp}\right)^{N^{node}} \qquad (7\text{-}1)$$

So the expected number of forwarding transmissions in *event area i* is:

$$N^{\exp} = \sum_{k=1}^{N^{node}} kp\left(k, N^{node}, f_{comp}^{sp}\right) \qquad (7\text{-}2)$$

7.3.2.1.3. Implementation of Spatial Compression in Pulse Switching Protocols

In order to implement spatial pulse compression, a node should be able to receive pulses from the other nodes in the same compression area after a randomly selected back-off period. Additionally, a node should also be able to recognize whether a received pulse is sent from another node in the same compression area or not. In CPS and PTP, a node can extract the *Cell-ID* of the next-hop cell of a received pulse according to the joint MAC-routing frame structures of CPS and PTP respectively in Chapters 5 and 6. If the extracted next-hop *Cell-ID* is same as its

own next-hop *Cell-ID* for the same event, the node decides this received pulse is forwarded by another node in the same compression area.

In order to implement spatial pulse compression in HPS, a node should be able to receive pulses from the other nodes in the same hop area after a randomly selected back-off period. As described in Section 4.3.3, pulse routing in HPS is realized by the frame-synchronized pulses motivated by the protocol state machine in HPS. Nodes in the same hop area cannot communicate with each other due to the fact that they are always in the same state. Thus, we need to make modifications of pulse forwarding in HPS. That is to say, a node in a hop area needs to obtain the knowledge of the hop-distance of the sender node of the received pulse besides the origin location of the received pulse. So we add the *Forwarding Flag* into the *Control sub-frame* as shown in Figure 4-3. Additionally, the *S-L-T* state machine driving the pulse routing in Section 4.3.3 is accordingly adjusted to *Interface-Sleep (IS) – Listen-Only (LO) – Transmit-Listen (TL)* state cycle. In this way, since all nodes within a hop area are in the *TL* (Transmit/Listen) phase, a node is able to listen to pulse transmissions of other neighboring nodes in the same hop area.

The modified pulse routing in HPS is presented as follows. When a pulse is transmitted by a node at hop-distance *h*, only its neighboring nodes at hop-area $(h-1)$ forward it towards the sink. Meaning, the nodes at hop-area *h* and $(h+1)$ should ignore the pulse. This logic ensures that a pulse is eventually delivered to the sink. While transmitting a pulse in the event sub-frame (see Section 4.3.3), its transmitter also sends a pulse in the corresponding slot of the *Forwarding Flag* Area of the *Control sub-frame*. That is, while forwarding a pulse by a hop area *h* node, it sends a pulse in the h^{th} time slot of the *Forwarding Flag Area*. By looking at the received pulse

in the *Forwarding Flag Area*, all the receivers of the pulse can decide if it should be discarded or forwarded towards the sink. This can ensure that a pulse from hop-area *h* should be forwarded only by nodes in hop-area $(h-1)$.

7.3.2.2 Temporal Compression

This subsection discusses the temporal compression algorithm which is applicable for both HPS and CPS.

7.3.2.2.1. Temporal Compression Logic

When a node detects multiple events in quick succession (e.g., detecting a moving target multiple times while it is within the detection range), it generates multiple pulses, each of which are independently forwarded to the sink. Temporal pulse compression can be used to remove this redundancy by ensuring that a node generates at most one pulse within certain time duration, which is referred to as *Deactivation Period*. Temporal compression is also applicable while pulse forwarding so that a node forwards at most one pulse originated from the same event area within a *Deactivation Period*.

7.3.2.2.2. Temporal Compression Model

We define the temporal compression factor to be the length of *Deactivation Period*, termed as f_{comp}^{t}. It needs to be dimensioned based on four factors for the target tracking application: 1) speed of a target *v*, 2) sensor detection range *r*, 3) sensor detection rate f_d, and 4) target trajectory pattern. For simplification, we mainly investigate straight-line, circular and square-

shape trajectories. We model the detection range of a sensor as a circle with radius *r* centering at the sensor.

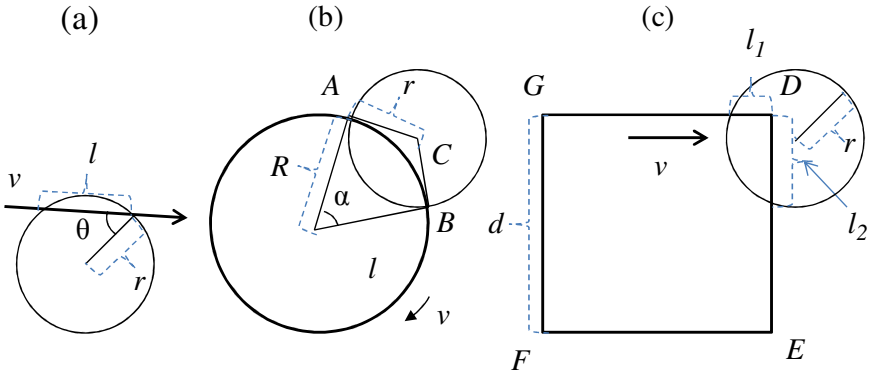


Figure 7-2: Target within a sensor's detection range

For any target trajectory, the sensor should be able to detect a target at least once within the time period t_{stayin} when the target stays in the sensor's detection range. Then f_d is larger than or equal to $1/min\{t_{stayin}\}$. For a given detection rate f_d, in order to ensure reducing pulse transmissions, the temporal compression factor f_{comp}^t should be larger than $max\{t_{stayin}\}$ and $1/f_d$. In other words, the lower bound of f_{comp}^t can be expressed as the maximum value of $1/f_d$ and $max\{t_{stayin}\}$. Furthermore, a target moves back to the location which was visited previously for a closed-loop trajectory. f_{comp}^t should be less than the minimum value of the time duration $t_{consecutive}$ between any two consecutive visits on the same location. Therefore, the upper bound of f_{comp}^t for a closed-loop trajectory is $min\{t_{consecutive}\}$. Note that, the upper bound of f_{comp}^t does not exist for an open-loop trajectory, for example a straight-line trajectory.

Specifically, for a straight-line target trajectory, suppose that l is the distance the target travels within a sensor's detection range, see Figure 7-2a. t_{stayin} can be calculated as l/v, where $l = 2r\cos\theta, 0 \le \theta \le \pi$. Then $max\{t_{stayin}\}$ is equal to $2r/v$. So the lower bound of f_{comp}^{t} for a straight-line target trajectory can be expressed as

$$f_{comp}^{t} > \max\{2r/v, 1/f_d\} \tag{7-3}$$

As shown in Figure 7-2b, for a circular target trajectory, a target enters a sensor's detection range at the *location A* and leaves at the *location B*. Suppose that l is the distance the target travels within a sensor's detection range, and α is the angle corresponding to the *sector ACB*. t_{stayin} is l/v, where $l = \alpha r$, $0 \le \alpha \le 2\arcsin(r/R)$. So the lower bound of f_{comp}^{t} for a circular trajectory can be expressed as $\max\{2r\arcsin(r/R)/v, 1/f_d\}$. Additionally, the target goes back to the *location A* following the circular trajectory. Considering $min\{t_{consecutive}\}$ equal to $2\pi R/v$. the upper bound of f_{comp}^{t} for a circular trajectory can be expressed as $2\pi R/v$. Therefore, f_{comp}^{t} for a circular trajectory is bounded as follows:

$$\max\{2r\arcsin(r/R)/v, 1/f_d\} < f_{comp}^{t} < 2\pi R/v \tag{7-4}$$

Same as circular trajectory, the lower and upper bounds of f_{comp}^{t} also exist in a square-shape trajectory. Suppose that d is the edge length of the square ($d >> r$), and l_1 and l_2 are the distances the target travels in two edges within a sensor's detection range (see Figure 7-2c). t_{stayin} is $(l_1 + l_2)/v$, where $(l_1 + l_2) \le 2r, l_1, l_2 \ge 0$. So the lower bound of f_{comp}^{t} for a circular trajectory can be expressed as $\max\{2r/v, 1/f_d\}$. Considering $min\{t_{consecutive}\}$ equal to $4d/v$.

the upper bound of f_{comp}^{t} for a square-shape trajectory can be expressed as $4d/v$. Thus, f_{comp}^{t} for a square-shape trajectory is bounded as follows:

$$\max\{2r/v, 1/f_d\} < f_{comp}^{t} < 4d/v \qquad (7\text{-}5)$$

7.4 Idling Energy Saving Measures

In this subsection, we propose interface shutdown (sleep) strategies to reduce idling energy consumptions for HPS, CPS and PTP.

7.4.1 Hop-angular Pulse Switching

We discussed the protocol state machine of HPS in Section 4.3.3. Following architectural measures are taken for improving the idling energy savings of HPS.

7.4.1.1 Inter-frame Sleep

Due to the cycling through the *sleep-listen-transmit* states, a node is forced to sleep one in every three frames, leading to a default *33%* conservation of idling energy. Note that because of a pipeline effect, this synchronized sleep does not in fact introduce any additional event delivery delay. Events from an *h* hop-distance node take exactly *h* frames to be transported to the sink.

7.4.1.2 Intra-frame Sleep

During a *listen* or a *transmit* frame, in addition to the downlink slots, a node remains awake during the uplink control sub-frame slots. However, during the event sub-frame, a node can sleep except during the slots it transmits or expects to receive pulses. During a *transmit* frame, since a node knows the {*sector-id, hop-distance*} information about a pulse that need to be transmitted, it can simply wake up before the appropriate slot-cluster and sleep after the

transmission. In a *listen* frame, without the knowledge of the expected pulses such sleeps are not possible. We incorporate the following scheme to address this.

Whenever a pulse (primary) is transmitted in the *Event sub-frame*, an associated pulse (secondary) is also transmitted in the corresponding slot (with the same hop-distance) within the routing area of the *Control sub-frame*. As shown in Figures 4-4b and 4-5b, since the primary pulse is transmitted in the *Event sub-frame* slot {*sector-id: 2, hop-distance: 3*}, the secondary pulse is transmitted in the 3^{rd} hop-distance slot within the *Routing Area* of the *Control sub-frame*. This ensures that a *listen* state node is informed about an impending primary pulse arrival in this frame by listening to the secondary pulse during the slots in the *Routing Area*. This way, the node can now wake up before the appropriate slot-clusters and sleep after the corresponding primary pulse is received. Such sleep reduces idling consumption significantly.

7.4.1.3 Delay-traded Sleep

A third type of sleep mechanism can be used for reducing idling energy consumption by inserting additional sleep frames in between the regular *sleep*, *listen, and transmit* frames. As shown in Figure 7-3, two additional sleep frames, marked as *D*, are inserted after each *S, L* and *T* frame. These *D* frames, which behave same as the *S* frames, do not affect the nodes' ability for synchronized state cycling, thus allowing the wave-front routing to continue. *D* frames provide a tunable mechanism for reducing idling consumption at the expense of additional pulse transportation delay. Using the *D* frames, delay can be scaled up by a constant factor *k*, which is one more than the number of inserted *D* frames. In Figure 7-3, *k* is 3, meaning that the delay is scaled up by 3 times and the idling energy is scaled down by 3. With this arrangement, a pulse may now have to remain buffered at the origin or at an intermediate node before it can be routed.

Frames	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hop-distance 3	S	D	D	L	D	D	T	D	D	S	D	D	L	D	D
Hop-distance 2	T	D	D	S	D	D	L	D	D	T	D	D	S	D	D
Hop-distance 1	L	D	D	T	D	D	S	D	D	L	D	D	T	D	D

Figure 7-3: Delay-traded sleep for idling energy conservation for HPS

7.4.2 Cellular Pulse Switching

Same as the intra-frame sleep strategy in HPS, protocol syntaxes are added in CPS for nodes to be able to selectively turn their RF interfaces off during appropriate parts of the frame. As for transmissions in *Pulse Forwarding* process, a node needs to be awake only during the slot clusters (of the *Localization Area*) at which it needs to transmit. During the other slot clusters, the node can simply keep the interface in sleep mode in order to save energy.

However, considering the asynchronous nature of pulse receptions, a node cannot sleep during all the non-transmission *Sensor Cell Areas* in the *Localization Area*. To address this, a *Control Area* (see Figure 5-2a) is added in the beginning of the uplink part of the frame. The slots in the *Control Area* of a frame are used for notifying about the impending receptions that are expected during the *Sensor Cell Areas* of the *Localization Area* of the frame. When a node plans to send a pulse originally from *cell m* during the *Sensor Cell Area m* of the *Localization Area* of a frame, it also sends a pulse in the m^{th} slot in the *Control Area* of the same frame. All nodes remain awake during the *Control Area*, thus ensuring the reception of notification about impending transmissions in the *Sensor Cell Area m* of the *Localization Area* of the frame. Based on this information, the node can remain awake during the *Sensor Cell Area m* of the *Localization Area*.

Such intra-frame interface sleep strategy can reduce the idling energy consumption. Additionally, the node's state in the *Control Area* of a frame is same as that in the *Sensor Cell Area m* of the *Localization Area* of the same frame.

7.4.3 Point-to-point Pulse Switching

We adopt the concept of the intra-frame sleep into the PTP architecture as well. As shown in Figure 6-4, the *Localization Area* of an *Event Report Frame* includes the *Forwarding Area*, *Source Area* and *Destination Area* respectively. The *Forwarding Area* can be used for notifying about the impending receptions that are expected during the *Source Area* and *Destination Area* of the frame. During the *Pulse Forwarding* process, a node always keeps *L* state in the *Forwarding Area* of a frame if no pulses transmissions are needed. The node may stay in *S* state in the *Source Area* and *Destination Area* of the frame. In details, if a node receives a pulse in the *Forwarding Area* of a frame, which corresponds to the *Cell-ID* of the node's own cell, it means that the node needs to forward this event pulse in a next frame. Then the node switches to be in *L* state both in the *Source Area* and *Destination Area* of the frame from *S* state. Otherwise, the node keeps *S* state both in the *Source Area* and *Destination Area* of the frame.

When a node is transmitting a pulse, it keeps *T* state in the *Forwarding Area*, *Source Area* and *Destination Area* of the current frame. Once pulse transmission is done, the node state will switch back to *L* state in the *Forwarding Area* of the next frame. The states in the *Source Area* and *Destination Area* of the next frame depend on the pulse reception in the *Forwarding Area* of the next frame.

In *Response* process, the state of a node in the *Response Area* is complementary to that in the *Localization Area* in the *Event Report Frame*. If a node is in *L* state only in the *Forwarding*

Area but in *S* state in the rest of the *Localization Area* of a frame, then the node is also in *S* state in the *Response Area* of the same frame.

7.5 Summary and Conclusion

This chapter proposed a number of measures to further improve the energy-efficiency of the proposed pulse switching architectures. These measures are divided into two types: 1) transmission energy saving measures, and 2) idling energy saving measures. The first-type includes the parameterized routing control methods and pulse compression algorithms. Interface sleeping strategies were mainly investigated in the second-type. They utilize the designed protocol state machines to schedule nodes into the sleep mode at specific slots or frames. Performance of these measures is presented in Chapters 9-12.

Chapter 8: Error Analysis

8.1 Introduction

This chapter analyzes the functional and performance impacts of false positive pulse errors and pulse loss errors on the proposed HPS, CPS and PTP protocols. To begin with, the impacts of false positive pulse errors are provided and modeled in this chapter. Then a series of protection methods from false positive pulse errors in those three switching protocols are formulated. Moreover, this chapter analytically proves the immunity of pulse switching from pulse loss errors.

8.2 False Positive Even Generation

When a false positive pulse generated during a frame happens to construct a seemingly valid event according to the frame structure, a false positive event is generated. Once such a false positive event is generated, it is forwarded all the way to the destination (i.e., sink in HPS and CPS, actuators in PTP) as a regular event, leading to a false positive event reporting. Let False Positive Pulse Rate (FPPR) be the probability that a false positive pulse is generated due to faulty detection in a given time-slot. We intend to compute the quantity False Positive Event Generation Rate (FPEGR) which corresponds to the probability of at least one false positive event generation per frame per node. This subsection constructs the models of the FPEGR computation for a given FPPR in the HPS, CPS and PTP architectures.

8.2.1 Hop-angular Pulse Switching

If pulses are erroneously detected [38] by a node in the listening state (L) such that a false

positive pulse in the *Control sub-frame* corresponds to another false positive pulse in the *Event sub-frame* (see Figure 4-3), then a false positive event is detected at that node. The generation of such a false positive event is constrained by the fact that the false positive pulses in the *Control* and the *Event sub-frames* have to exactly correspond. For example, a node (in listening state) in hop-area h will generate a false positive event when paired false positive pulses are simultaneously detected in the i^{th} slot in the routing area of the *Control sub-frame* ($i=(h+1),\ldots,H$) and in the i^{th} slot-cluster in the *Event sub-frame* within the same frame. If such paired false positive pulses are not simultaneously detected, an un-paired pulse is identified as false positive and discarded by the receiver node.

According to the wave-front routing logic, a node in hop-area h is able to receive pulses corresponding to events that are generated at nodes in hop areas $(h+1)$ to H. As a result, the vulnerable area for false positive events in the *Control sub-frame* is from the $(h+1)^{th}$ time slot to the H^{th} time slot (i.e., $(H-h)$ time slots), and the corresponding vulnerable area in the *Event sub-frame* is from the $(h+1)^{th}$ slot cluster to the H^{th} slot cluster (i.e., $(H-h)$ slot clusters, with each slot cluster containing $360°/\alpha$ slots). Let k_h $(1 \le k_h \le H-h)$ be the number of valid false positive events generated in a frame at a node in hop-area h. In order to generate k_h valid false positive events in a frame, false positive pulses should occur in the vulnerable k_h time slots in the *Control sub-frame* and in the corresponding vulnerable $k_h(360°/\alpha)$ time slots in the *Event sub-frame*. Therefore, $P_h^{k_h}$ ($k_h \in [1, H-h]$), the probability of k_h valid false positive events generated in a frame, can be written as:

$$P_h^{k_h} = \binom{H-h}{k_h} p^{k_h} (1-p)^{H-h-k_h} \left(1-(1-p)^{360°/\alpha}\right)^{k_h} \quad (8\text{-}1)$$

where p is the quantity FPPR.

Therefore, $FPEGR_{HPS}$ for a node in the hop-area h ($h \in [1, H]$) can be expressed as:

$$FPEGR_{HPS} = \sum_{k_h=1}^{H-h} P_h^{k_h}$$
$$= \sum_{k_h=1}^{H-h} \binom{H-h}{k_h} p^{k_h} (1-p)^{H-h-k_h} \left(1-(1-p)^{360°/\alpha}\right)^{k_h} \quad (8\text{-}2)$$

8.2.2 Cellular Pulse Switching

If pulses are erroneously detected [38] by a node in the listening state (L) such that a false positive pulse in the *Control Area* corresponds to another false positive pulse in the *Localization Area* (see Figure 5-2), then a false positive event is produced. Generation of such a false positive event is constrained by the fact that the false positive pulses in the *Control Area* and *Localization Area* have to occur in a corresponding pair of slots. In other words, a node in state L in *cell* i will generate a false positive event seemingly original from *cell* j ($i, j \in [1, M], i \neq j$), only when paired false positive pulses are simultaneously detected in the j^{th} slot of the *Control Area* and in the $(i+1)^{th}$ slot of the j^{th} slot-cluster of the *Localization Area* within the same frame. If such paired false positive pulses are not simultaneously detected, then no false positive event is detected and each such individual pulse is discarded by the receiver node.

According to the cellular pulse switching in Section 5.3, the specific slot of the *Control Area* and corresponding slot-cluster of the *Localization Area* notify the event's origin *Cell-ID*, and the slot within that slot-cluster of the *Localization Area* represents the next-hop *Cell-ID*. A node in *cell* i is able to receive pulses originally from any other *cell* j ($i, j \in [1, M], i \neq j$). As a

result, for false positive events at the node in *cell i*, the vulnerable area in the *Control Area* are all slots except the i^{th} time slot, and the corresponding vulnerable area in the *Localization Area* is the $(i+1)^{th}$ slot of any slot-cluster except the i^{th} slot cluster. In other words, there are $(M-1)$ vulnerable matching pairs of slots in the *Control Area* and *Localization Area* respectively. For a node in *cell i*, let E_k^i $(1 \le k \le M-1)$ be the k^{th} valid false positive event seemingly originally from *cell* id_k, which corresponds to a false positive pulse generated in the $(id_k)^{th}$ slot of the *Control Area* and another false positive pulse in the $(i+1)^{th}$ slot of the $(id_k)^{th}$ slot-cluster of the *Localization Area* in a frame. The probability of at least one valid false positive event generated in a frame is equal to the probability of the union of all E_k^i events. Therefore, $FPEGR_{CPS}$ at a node in *cell i* can be expressed as:

$$
\begin{aligned}
FPEGR_{CPS} &= P\left(\bigcup_{k\in[1,M-1]} E_k^i\right) \\
&= \sum_{k=1}^{M-1} P(E_k^i) \quad - \sum_{\substack{k_1,k_2\in[1,M-1] \\ k_1\neq k_2}} P(E_{k_1}^i \cap E_{k_2}^i) \\
&+ \sum_{\substack{k_1,k_2,k_3\in[1,M-1] \\ k_1\neq k_2,k_2\neq k_3,k_3\neq k_1}} P(E_{k_1}^i \cap E_{k_2}^i \cap E_{k_3}^i) + \cdots \\
&+ \sum_{\substack{k_1,\cdots,k_\varsigma\in[1,M-1] \\ \varsigma\in[4,M-2],k_\tau\neq k_\psi,\tau,\psi\in[1,\varsigma]}} (-1)^{\varsigma-1} P(E_{k_1}^i \cap \cdots \cap E_{k_\varsigma}^i) \\
&+ \cdots + (-1)^{M-2} P(E_{k_1}^i \cap \cdots \cap E_{k_{M-1}}^i)
\end{aligned}
$$

$$= (M-1)p^2 - \binom{M-1}{2}p^4 + \binom{M-1}{3}p^6 + \cdots + (-1)^{\varsigma-1}\binom{M-1}{\varsigma}p^{2\varsigma}$$
$$+ \cdots + (-1)^{M-2}p^{2(M-1)} \qquad (8\text{-}3)$$
$$= \sum_{k=1}^{M-1}(-1)^{k-1}\binom{M-1}{k}p^{2k}$$

where p is the FPPR, M is the number of sensor cells in the network, and *2* notifies the number of pulses in a frame. Observe Eqn. 8-3, the quantity $FPEGR_{CPS}$ only depends on the values of M and p for the given number of pulses in a frame (i.e., 2). Thus, $FPEGR_{CPS}$ for any sensor cell is same for the given FPPR.

8.2.3 Point-to-point Pulse Switching

In the *Localization Area* of a frame (see Figure 6-4), a specific slot of the *Source Area* notifies an event's origin *Cell-ID*. In the *Destination Area*, a specific slot corresponds to an event's destination *Cell-ID*. A specific slot of the *Forwarding Area* notifies an event's next-hop *Cell-ID*. If pulses are erroneously detected [38] by a node in *cell i* in the listening state (*L*) such that a false positive pulse in the i^{th} ($i \in [1, M]$) slot of the *Forwarding Area* of the *Localization Area* corresponds to a false positive pulse in the j^{th} ($i, j \in [1, M], i \neq j$) slot of the *Source Area* of the *Localization Area*, and another false positive pulse in the k^{th} ($k \in [1, M]$) slot of the *Destination Areas* of the *Localization Area* (see Figure 6-4), then a false positive event which is seemingly originated from *cell j* and destined to *cell k* is produced. Generation of such a false positive event is constrained by the following four facts: 1) only one false positive pulse occur in the *Source Area* and also in the *Destination Area*, 2) one of seeming next-hop *Cell-IDs* should be the *Cell-ID* of the receiver node, 3) the seemingly original *Cell-ID* should be different from the

seemingly destination *Cell-ID*, and 4) the seemingly original *Cell-ID* should be different from the *Cell-ID* of the receiver node. If a node in *cell i* in the *listening* state detects a false positive pulse in the i^{th} slot of the *Forwarding Area*, and more than one false positive pulse either in the *Source Area* or the *Destination Area*, the node would regard it as a collision. According to the collision handling in Section 6.4, the receiver node sends a collided response pulse to the seeming sender nodes. We assume the probability that the seeming sender nodes happened to transmit the event pulses matching with such false positive events to be very low. As a result, the receiver node would not receive any event retransmission corresponding to these false positive events. Eventually, these false positive pulses would be discarded by the receiver node. Therefore, $FPEGR_{PTP}$ at a node in *cell i* can be expressed as:

$$FPEGR_{PTP} = \binom{M-1}{1}^2 p^3 (1-p)^{2(M-1)} \qquad (8\text{-}4)$$

where p is the quantity FPPR, M is the number of cells in the network. Observe Eqn. 8-4, the quantity $FPEGR_{PTP}$ only depends on the values of M and p. Thus, $FPEGR_{PTP}$ for any cell is same for the given FPPR.

8.3 Protection from False Positive Pulses

In order to limit false positive events, we develop a series of novel *Frame Protection Code* (FPC) mechanisms for HPS, CPS and PTP as follows. An FPC is an R-slot long protection code which is appended at the end of each frame. For each transmitted event, the transmitter node sends up to R pulses as a protection code computed based on a pre-determined algorithm. In HPS and CPS (both allow multiple event transmissions in a frame), the receiver uses the same algorithm to compute the individual FPC codes for each of the received events in that frame. If

the *R*-bit long sequence constructed via slot-by-slot logical OR of those individual FPCs does not match with the received *R*-slot long FPC pulse sequence at the end of the frame, then an error is declared. Subsequently, all the events received in that frame are discarded. The logical OR operation is done since overlapping pulses in a slot result into a single detectable pulse as discussed in Section 4.3.5. In PTP that declares collisions at the presence of multiple event transmissions in a frame, the receiver only applies the same algorithm to compute the FPC code for a single received event in that frame.

Note that in HPS, each transmitted event includes a pulse in the *Control sub-frame* and a corresponding pulse in the *Event sub-frame.* In CPS, each transmitted event contains a pulse in the *Control Area* and a corresponding pulse in the *Localization Area*. In PTP, each transmitted event comprises of a pulse in the *Source Area*, a pulse in the *Destination Area* and another pulse in the *Forwarding Area* of the *Localization Area*.

8.3.1 Hop-angular Pulse Switching

Consider a node in the hop area h ($h \in [1, H]$) which receives *m* events $\{\zeta_1, \zeta_2, \ldots, \zeta_m\}$ in a frame. An event ζ_i ($i \in [1, m]$) is originated at hop-area h_i ($h+1 \le h_i \le H$) and is received as two pulses, one in the *Control sub-frame* and another in the *Event sub-frame*. Since within a frame, a node can receive multiple events originated from the same hop-area, the h_i values are all not necessarily different. Note that these events are received at the node in hop area h immediately from nodes in hop area $(h+1)$. For each event ζ_i, an *R*-slot long FPC code $[\gamma_i]$ is computed and transmitted at the end of the frame by the corresponding immediate transmitter. Transmissions of all $[\gamma_i]$ produce a pulse-merged FPC $[\hat{\gamma}]$ at the receiver. The receiver

computes individual $[\gamma_i]$ by applying the FPC computation algorithm on the received ζ_i, and then compares the quantity $[\gamma_1 OR \gamma_2 \ldots OR \gamma_m]$ with the received $[\hat{\gamma}]$ for detection of errors. Three FPC computation algorithms, each offering different quality of protection, are presented below.

8.3.1.1 Single Pulse Code (SPC) Algorithm

The FPC code $[\gamma_i]$ for event ζ_i can be written as $[\phi_0, \phi_1, \ldots, \phi_{R-1}]$, where ϕ_j ($j \in [0, R-1]$) is a bit representing *0* or *1,* corresponding to no pulse transmission or a pulse transmission in the corresponding FPC slot. The rule for setting ϕ_j is: $if\,(j == h_i \quad MODULO \quad R) then \quad \phi_j = 1, else \quad \phi_j = 0.$ Meaning, for any event ζ_i , the corresponding code $[\gamma_i]$ will contain exactly one bit being *1*. The quantities $[\hat{\gamma}]$ and $[\gamma_1 OR \gamma_2 \ldots OR \gamma_m]$, however, may contain multiple bits as *1*. The code size *R* in SPC determines the strength of protection, and it can range from *2* to *H*, the maximum hop-area.

Let $k_h (1 \le k_h \le H - h)$ be the number of valid false positive events generated in a frame at a node in the *listening* mode (L) in hop-area *h* with SPC based protection turned on. This would require false positive pulses to occur in k_h different time slots in the *Control sub-frame*, $k_h (360°/\alpha)$ vulnerable time slots in the corresponding *Event sub-frame*, and the corresponding $n_{SPC}(k_h)$ slots in the R-bit long protection area. The quantity $n_{SPC}(k_h)$ represents the number of '1's in $[\gamma_1 OR \gamma_2 .. OR \gamma_m]$, and $1 \le n_{SPC}(k_h) \le \min(k_h, R)$. The probability that there are $n_{SPC}(k_h)$ false positive pulses generated in the *Protection Area* and the rest $(R - n_{SPC}(k_h))$ slots

in this area have no pulses can be written as $p^{n_{SPC}(k_h)}(1-p)^{R-n_{SPC}(k_h)}$. Multiplying this term to $P_h^{k_h}$ for the without protection case in Eqn. 8-1, we can write the new expression for $P_h^{k_h}$ with SPC as:

$$P_h^{k_h} = \binom{H-h}{k_h} p^{k_h+n_{SPC}(k_h)}(1-p)^{H+R-h-k_h-n_{SPC}(k_h)}\left(1-(1-p)^{360°/\alpha}\right)^{k_h} \quad (8\text{-}5)$$

Now P_h^{SPC}, the $FPEGR_{HPS}^{SPC}$ under the SPC protection for a node in the hop-area h ($h \in [1, H]$), can be expressed as:

$$\begin{aligned} FPEGR_{HPS}^{SPC} &= \sum_{k_h=1}^{H-h} P_h^{k_h} \\ &= \sum_{k_h=1}^{H-h} \binom{H-h}{k_h} p^{k_h+n_{SPC}(k_h)}(1-p)^{H+R-h-k_h-n_{SPC}(k_h)}\left(1-(1-p)^{360°/\alpha}\right)^{k_h} \end{aligned} \quad (8\text{-}6)$$

8.3.1.2 Parity Pulse Code (PPC) Algorithm

In PPC, the code size *R* is set to *1*, meaning the protection code has only one pulse slot. This scheme behaves like a *1-bit parity protection*, with the rule for setting ϕ_0 ($j = 0$) as: $if(h_i \quad MODULO \ \ 2 == 1) then \ \ \phi_0 = 1, else \ \ \phi_0 = 0.$ With PPC, the only bit in the FPC code can be either *1* or *0*.

The computation for $FPEGR_{HPS}^{PPC}$ with PPC is very similar to that with SPC, except that the quantity $n_{PPC}(k_h)$ is either *1* or *0* and the value of *R* is *1*. The quantity $n_{PPC}(k_h)$ represents the number of '*1*'s in $[\gamma_1 OR \gamma_2 ... OR \gamma_m]$. $FPEGR_{HPS}^{PPC}$ with PPC can be written:

$$FPEGR_{HPS}^{PPC} = \sum_{k_h=1}^{H-h} P_h^{k_h}$$
$$= \sum_{k_h=1}^{H-h} \binom{H-h}{k_h} p^{k_h + n_{PPC}(k_h)} (1-p)^{H+1-h-k_h-n_{PPC}(k_h)} \left(1-(1-p)^{360°/\alpha}\right)^{k_h} \quad (8\text{-}7)$$

8.3.1.3 Multi Pulse Code (MPC) Algorithm

With MPC, the code $[\phi_0, \phi_1, \ldots, \phi_{R-1}]$ is the binary representation of the quantity $(h_i \quad MODULO \quad V)+1$, where $R = \lfloor \log_2 V \rfloor + 1$. The largest value of the parameter *V*, which satisfies the above relationship, is chosen for the maximum possible protection. Note that unlike in SPC, in an MPC-computed code, multiple bits can be simultaneously set to *1*, and it can range from 1 to *R*. For example, with a target code size of *3* pulse slots (i.e., $R = 3$), the largest *V* would be *7*. Now for an event originated at hop-area *5* (i.e., $h_i = 5$), the MPC code would be the binary representation of $(5 \quad MODULO \quad 7)+1$, which is *110*. This means that two pulses are transmitted in the first two slots in the *Protections Area*, and no pulse is transmitted in the last slot. The $FPEGR_{HPS}^{MPC}$ under MPC for a node in the hop-area h ($h \in [1, H]$) can be written as:

$$FPEGR_{HPS}^{MPC} = \sum_{k_h=1}^{H-h} P_h^{k_h}$$
$$= \sum_{k_h=1}^{H-h} \binom{H-h}{k_h} p^{k_h + n_{MPC}(k_h)} (1-p)^{H+R-h-k_h-n_{MPC}(k_h)} \left(1-(1-p)^{360°/\alpha}\right)^{k_h} \quad (8\text{-}8)$$

in which $n_{MPC}(k_h)$ represents the number of '*1*'s in $[\gamma_1 OR \gamma_2 ... OR \gamma_m]$.

8.3.1.4 Summary

Since with MPC multiple bits can be simultaneously set to *1*, the condition $n_{MPC}(k_h) \geq n_{SPC}(k_h)$ is generally true. Based on numerically computed (from Eqns. 8-6 through 8-8) and simulation obtained FPEGR in HPS in Section 10.4.1, it is shown that MPC

offers stronger protection compared to both PPC and SPC. Although SPC shows less protection than MPC, SPC is more energy efficient than MPC. It is due to the fact that SPC has one bit to be 1 which involves only one additional pulse in the *Protection Area*. It is also true for the cellular pulse switching and the point-to-point pulse switching. So we only consider the SPC protection algorithm in the following CPS and PTP architectures.

8.3.2 Cellular Pulse Switching

Consider a node in *cell i* ($i \in [1, M]$) which receives k ($k > 0$) events $\{\zeta_1, \zeta_2, \ldots, \zeta_k\}$ in a frame. An event ζ_j is originated at *cell* id_j ($j \in [1, k]$) which is received as two pulses, one in the *Control Area* and another in the *Localization Area*. Since within a frame, a node can receive multiple events originated from the same sensor cell, the id_j values are all not necessarily different. For each event ζ_j, an R-slot long FPC code $\lfloor \gamma_j \rfloor$ is computed and transmitted at the end of the frame by the corresponding immediate transmitter. Transmissions of all $\lfloor \gamma_j \rfloor$ produce a pulse-merged FPC $[\hat{\gamma}]$ at the receiver. The receiver computes individual $\lfloor \gamma_j \rfloor$ by applying the FPC computation algorithm on the received ζ_j, and then compares the quantity $[\gamma_1 OR \gamma_2 \ldots OR \gamma_k]$ with the received $[\hat{\gamma}]$ for detection of errors.

We use Single-Pulse-Code (SPC) Algorithm to protect from false positive pulses. The FPC code $\lfloor \gamma_j \rfloor$ for event ζ_j can be written as $[\phi_0, \phi_1, \ldots, \phi_{R-1}]$, where ϕ_m ($m \in [0, R-1]$) is a bit representing *0* or *1,* notifying no pulse transmission or a pulse transmission in the corresponding FPC slot. The rule for setting ϕ_m is: $if(m == id_i \quad MODULO \quad R) then \quad \phi_m = 1, else \quad \phi_m = 0$. It means, for any event ζ_j, the corresponding code $\lfloor \gamma_j \rfloor$ will contain exactly one bit as *1*. The

quantities $[\hat{\gamma}]$ and $[\gamma_1 OR \gamma_2 \ldots OR \gamma_k]$, however, may contain multiple bits as *1*. The code size *R* in SPC determines the strength of protection, and it can range from *2* to *M*, the maximum cell number. In the rest of the dissertation, the size of protection code for CPS is chosen to be *M*.

For a node in *cell i*, let E_k^i $(1 \le k \le M-1)$ be the k^{th} valid false positive event seemingly originally from *cell* id_k, which corresponds to a false positive pulse generated in the $(id_k)^{th}$ slot of the *Control Area*, a false positive pulse in the $(i+1)^{th}$ slot of the $(id_k)^{th}$ slot-cluster of the *Localization Area*, and another false positive pulse in the $(id_k)^{th}$ slot of the *Protection Area* in a frame. The probability of at least one valid false positive event generated in a frame with SPC is equal to the probability of the union of all E_k^i events. Thus, the $FPEGR_{CPS}^{PRCT}$ under SPC with $R = M$ for a node in *cell i* can be written as:

$$FPEGR_{CPS}^{PRCT} = \sum_{k \in [1, M-1]} \binom{M-1}{k} p^{3k} \qquad (8\text{-}9)$$

Similarly as $FPEGR_{CPS}$ in Eqn. 8-3, the quantity $FPEGR_{CPS}^{PRCT}$ with SPC only depends on the values of *M* and *p* for the given number of pulses in a frame (i.e., 3). Based on FPEGR without and with protection in CPS obtained from numerical computation (from Eqns. 8-3 and 8-9) and simulation in Section 11.5.1, it is shown that SPC offers good protection from false positive pulses by reducing the effective FPEGR in CPS.

8.3.3 Point-to-point Pulse Switching

As in CPS in the above section, we use Single-Pulse-Code (SPC) Algorithm in PTP to protect from false positive pulses. Consider a node in *cell n* ($n \in [1, M]$) which receives an event

ζ in a frame. An event ζ is originated at *cell i* ($i \in [1, M], i \neq n$) and destined to *cell j* ($j \in [1, M], j \neq i$) which is sent to *cell n*. Such an event is received as three pulses, one in the *Source Area*, one in the *Destination Area* and another in the *Forwarding Area*. The FPC code $[\gamma]$ for event ζ can be written as $[\phi_0, \phi_1, \ldots, \phi_{R-1}]$, where ϕ_m ($m \in [0, R-1]$) is a bit representing *0* or *1,* notifying no pulse transmission or a pulse transmission in the corresponding FPC slot. The rule for setting ϕ_m is: $if (m == i \quad MODULO \quad R) then \quad \phi_m = 1, else \quad \phi_m = 0.$ It means, for an event ζ, the corresponding code $[\gamma]$ depends on the *Cell-ID* of the event's origin. The parameter $[\gamma]$ will contain exactly one bit as *1*. The size of protection code for PTP is chosen to be *M*. Based on the FPEGR without protection formulated in Eqn. 8-4, the $FPEGR_{PTP}^{PRCT}$ under SPC with $R = M$ for a node in *cell n* can be written as:

$$FPEGR_{PTP}^{PRCT} = \binom{M-1}{1}^2 p^4 (1-p)^{3(M-1)} \tag{8-10}$$

Similarly as $FPEGR_{PTP}$ in Eqn. 8-4, the quantity $FPEGR_{PTP}^{PRCT}$ with SPC only depends on the values of *M* and *p* for the given number of pulses in a frame (i.e., 4). Based on FPEGR without and with protection in PTP obtained from numerical computation (from Eqns. 8-4 and 8-10) and simulation in Section 12.3.1, it is shown that SPC offers good protection from false positive pulses by reducing the effective FPEGR in PTP.

8.4 Immunity from Pulse Loss

Pulse losses can manifest in the form of un-reported events. Such effects, however, can be alleviated by exploiting the pulse transmission redundancy inherent to pulse routing, e.g., turning off the functionality of pulse compression or increasing the route diversity. The *Pulse Loss Rate*

(PLR) is defined as the probability that a pulse is lost due to multi-path, channel noise, or various types of interferences. The *Event Loss Rate* (ELR) represents the probability that the destination (i.e., sink in HPS and CPS and actuators in PTP) fails to receive the event due to pulse losses along the route (including diversity) from the event-source to the destination. This subsection respectively analyzes the relationships between PLR and ELR in the HPS, CPS and PTP architectures.

8.4.1 Hop-angular Pulse Switching

As shown in Figure 4-5 for a linear network and in Figure 7-1 for a lattice network, a pulse initially fans out due to the duplicative nature of the wave-front routing, and then converges as it nears the sink. The degree of fan-out and the resulting route diversity depend on the network topology and the routing sector constraint δ. This route diversity due to pulse fan-out can make the architecture partially tolerant to pulse losses within the fan-out areas. For example in Figure 4-5, the event can be reported to the sink in spite of a pulse loss across hops *E-to-B* or *E-to-C*. Since with lower sector-constraints (i.e., lower δ) the fan-out is higher, the resulting higher route diversity is expected to make the system more tolerant to pulse losses. An event is more likely to remain un-reported when pulse losses take place near the origin or the sink, since the route diversity is the minimum in those locations. For example in Figure 4-5, the event will not be reported to the sink if a pulse loss occurs on the *A-to-Sink* hop.

According to the frame structure in Figure 4-3, in each hop an event is represented by one pulse in the *Control sub-frame* and a corresponding pulse in the *Event sub-frame*. Loss of any of these two pulses will lead to the loss of the corresponding event, which will remain un-reported to the sink. Therefore, the probability (termed as e) of losing an event on any hop on the route

of the event is the same as the probability of losing at least one of the following two pulses; one at the *Control sub-frame* and the other is at the *Event sub-frame*. This probability can be expressed as: $e = 1-(1-PLR)^2$. Note that an event routing may consist of multiple pulses routings due to the inherent route diversity as explained in Figure 4-5.

The following model expresses the relationship between *Pulse Loss Rate* (PLR) and the corresponding *Event Loss Rate* (ELR). Let n_i represent a node on the route of an event and p_i represent the probability that the node n_i fails to receive the event due to pulse losses along the route (including diversity) from the event-source to node n_i. Let $\Re_i$ represent the set of *parent* nodes of n_i, such that: 1) each node in $\Re_i$ belongs to a hop-area that is 1-hop higher than that of n_i, and 2) when a node in $\Re_i$ forwards a pulse, node n_i is able to receive it. The probability that node n_i fails to receive an event from a parent node n_j ($j \in \Re_i$) can be written as $p_j + (1-p_j)e$, in which the first term represents the probability that node n_j itself did not receive the event due to a lost pulse on the way, and the second term represents the probability that n_j has received the event, but the pulse from n_j to n_i has been lost. Therefore, the probability that node n_i fails to receive the event can be written as:

$$\begin{aligned} p_i &= \prod_{j \in \Re_i} \{p_j + (1-p_j)e\} \\ &= \prod_{j \in \Re_i} \{p_j + (1-p_j)(2PLR - PLR^2)\} \end{aligned} \qquad (8\text{-}11)$$

For a given topology and *Pulse Loss Rate PLR*, p_i for a node n_i can be iteratively computed starting from a pulse's source node to gradually going down to the nodes in lower hop

areas along its route. When n_i corresponds to the sink node, the quantity p_i represents the *Event Loss Rate (ELR)* for a given *PLR*.

8.4.2 Cellular Pulse Switching and Point-to-point Pulse Switching

In both CPS and PTP, the response mechanisms respectively proposed in Section 5.4 and Section 6.4 can mitigate the effects of pulse loss errors in that a node would retransmit a pulse for as many times as possible until the successful pulse reception in a next-hop cell. We analyze the case when the protection (see Section 8.3) is disabled and the route diversity δ is 1.

According to the frame structure of CPS in Figure 5-2, an event pulse in any sensor-cell is represented by one pulse in the *Control Area* and another corresponding pulse in the *Localization Area*. Loss of any of the two pulses in a frame will lead to the loss of the corresponding event. According to the frame structure of PTP in Figure 6-2, an event pulse in any cell is represented by one pulse in the *Source Area*, one pulse in the *Destination Area*, and another pulse in the *Forwarding Area*. The number of pulses in a frame either in CPS or PTP is termed as n. $n = 2$ in CPS, and $n = 3$ in PTP. Loss of any of these n pulses in a frame will lead to the loss of the corresponding event. Therefore, the probability of losing an event on any transmission hop on the routing path (termed as e) is the same as the probability of losing at least one of these n pulses in a frame. As in HPS, this probability can be expressed as: $e = 1 - (1 - PLR)^n$. An event routing may consist of multiple pulse routings. The following model expresses the relation between *Pulse Loss Rate* (PLR) and the corresponding *Event Loss Rate* (ELR) for both CPS and PTP. Similarly, as in HPS (Section 8.4.1), let n_i represent a node on the route of an event and p_i represent the probability that the node n_i fails to receive the event due to pulse losses along

the route from the event-source to node n_i. Let $\Re_i$ represent the sub-set of *parent* nodes of n_i, such that: 1) each node in $\Re_i$ belongs to a neighbor cell which forwards pulses to n_i, and 2) when a node in $\Re_i$ forwards a pulse, node n_i is able to receive it. The probability that node n_i fails to receive an event from a parent node n_j ($j \in \Re_i$) once can be written as $p_j + (1 - p_j)e$, in which the first term represents the probability that node n_j itself did not receive the event due to pulse losses on the way, and the second term represents the probability that n_j has received the event, but the pulse from n_j to n_i has been lost. Since the node n_i does not receive a pulse from the parent node n_j, it would not respond to n_j. In this case, n_j would keep retransmitting the pulse until it receives the response pulse from n_i. We assume the quantity of the times of such retransmissions is θ_k which can be very large. So the probability that node n_i fails to receive an event from a parent node n_j ($j \in \Re_i$) for the θ_k times can be written as $p_j + \left[(1 - p_j)e\right]^{\theta_k}$. Therefore, the probability that node n_i fails to receive the event can be written as:

$$\begin{aligned} p_i &= \prod_{j \in \Re_i} \left\{ p_j + \left[(1 - p_j)e\right]^{\theta_k} \right\} \\ &= \prod_{j \in \Re_i} \left\{ p_j + \left[(1 - p_j)\left(1 - (1 - PLR)^n\right)\right]^{\theta_k} \right\} \end{aligned} \tag{8-12}$$

For a given topology and PLR, p_i for a node n_i can be iteratively computed starting from a pulse's source node to the nodes in cells closer to the destination (i.e., sink in CPS, actuators in PTP) along its route. When n_i corresponds to the destination node, the quantity p_i represents the *Event Loss Rate* (ELR) for a given PLR. The results from the simulation, as shown in Section

11.5.2 and Section 12.3.2, and the calculation according to Eqn. 8-12 ($n = 2$ in CPS, $n = 3$ in PTP) prove that the introduction of *Response Mechanism* can mitigate the effects of pulse loss errors on ELR.

8.5 Impacts of Protection on Pulse Loss

Although the FPC mechanism alleviates the effects of false positive pulse detection, it may sometime aggravate the impacts of pulse losses due to the following reason. Compared to the case without protection, a successful event transmission in a frame has more pulse transmissions in a frame with FPC protection turned on. We use the parameter n_{prct} to represent the number of pulses in a frame with protection. A pulse loss in any one of these n_{prct} pulses can lead to an event loss. Without FPC, however, an event loss can be caused due to pulse losses in any of $(n_{prct} - n_{FPC})$ pulses, where n_{FPC} represents the number of *1's* in the FPC code according to the rules in Section 8.3. As a result, for a given *Pulse Loss Rate* (*PLR*), FPC protection can increase the corresponding Event Loss Rate (*ELR*). Following is an analysis of *ELR* in the presence of FPC protection in HPS, CPS and PTP architectures.

8.5.1 Hop-angular Pulse Switching

In HPS, a successful event transmission in a frame depends on correct pulse transmissions in the control sub-frame, the event sub-frame and the FPC protection area at the end of the frame with FPC protection turned on. For false positive protection in HPS, there are n_{prct} pulses (i.e., one in *Control sub-frame*, one in *Event sub-frame*, and at least one in the *Protection Area*) in a frame corresponding to a single event generated in the hop area *h*. $n_{prct} = 2 + n_{FPC}$, where

n_{FPC} represents the number of *1's* in the FPC code according to the rules in Section 8.3.1. For example, n_{FPC} represents the number of *1's* after converting the decimal quantity $(h \quad MODULO \quad V)+1$ to binary code in the MPC protection algorithm ($n_{FPC} \in [1, R]$, R is the protection size). Additionally, n_{FPC} is equal to 1 both in the SPC and PPC protection algorithms.

Therefore, the probability of losing an event on any hop on the route of the event is the same as the probability of losing at least one of those n_{prct} pulses in HPS. Such probability e can be expressed as: $e = 1-(1-PLR)^{n\,prct}$. Incorporating this new value of e in Eqn. 8-11, yields the probability that node n_i fails to receive the event with protection:

$$p_i = \prod_{j \in \Re_i} \left\{ p_j + (1-p_j)\left(1-(1-PLR)^{n\,prct}\right) \right\} \qquad (8\text{-}13)$$

Similar to the without protection scenario, when n_i corresponds to the sink node, the quantity p_i represents the *Event Loss Rate* (ELR) for a given PLR. Since the value of e is higher with protection compared to that without protection, the quantity ELR is higher as a result of incorporating false positive protection as detailed in Section 10.4.2.

8.5.2 Cellular Pulse Switching and Point-to-point Pulse Switching

As summarized in Section 8.3.1.4, we only consider the SPC protection algorithm in the CPS and PTP architectures. In the CPS protocol, with SPC protection turned on, a successful event transmission in a frame depends on correct pulse transmissions in the *Control Area*, *Localization Area*, and *Protection Area* of each frame. In other words, n_{prct} in CPS is equal to

3. In the PTP protocol, a successful event transmission in a frame depends on correct pulse transmissions in the *Source Area*, *Destination Area*, *Forwarding Area* and *Protection Area* of each frame with SPC protection turned on. In other words, n_{prct} in PTP is equal to 4.Therefore, the probability of losing an event on one hop on the route of the event is the same as the probability of losing at least one of these n_{prct} pulses in a frame, when the route diversity δ is 1. Such probability *e* can be expressed as: $e = 1-(1-PLR)^{n_{prct}}$ ($n_{prct}=3$ in CPS, $n_{prct}=4$ in PTP). Incorporating this new value of *e* in Eqn. 8-12, yields:

$$p_i = \prod_{j \in \Re_i} \left\{ p_j + \left[(1-p_j)\left(1-(1-PLR)^{n_{prct}}\right) \right]^{\theta_k} \right\} \qquad (8\text{-}14)$$

Similar to the no protection scenario, when n_i corresponds to the sink node, the quantity p_i represents the ELR for a given PLR. Although the value of *e* is higher with SPC compared to that without protection, the large exponential number θ_k introduced by *Response Mechanism* contributes to a significant reduction of the probability. Thus, the negative impact of SPC on ELR can be negligible in CPS and PTP. The results, derived from both model (i.e., Eqn. 8-14) and simulation, are shown in Sections 11.5.2 and 12.3.2.

8.6 Summary and Conclusion

This chapter summarized the functional and performance impacts of false positive pulse errors and pulse loss errors into analytical models for all the three pulse switching architectures (i.e., HPS, CPS and PTP).

A set of the protection methods against false positive errors (i.e., MPC, SPC and PPC) for HPS, CPS and PTP protocols was proposed in this chapter. MPC offers stronger protection

compared to both PPC and SPC. However, SPC is more energy efficient and also impacts pulse loss less than MPC. Therefore, SPC is chosen to be the protection algorithm in all three pulse switching architectures.

This chapter also presented the models of computing the Event Loss Rate (ELR) for a given Pulse Loss Rate (PLR) in HPS, CPS and PTP. In addition, this chapter analyzed the impacts of protection on the relationship between ELR and PLR. The results derived from these models and the simulations as shown in Chapters 10-12 prove the immunity of pulse switching from pulse loss errors.

Chapter 9: Comparison between Pulse and Packet Switching

9.1 Introduction

In order to show the superiorities of pulse switching to packet switching, this chapter analyzes and evaluates the delay and energy performance of pulse switching in comparison to a comparable packet-based TDMA solution. For all subsequent analysis and simulation in this chapter, we will use the baseline system parameters as summarized in Table 9-1. As for the network topology, we choose to distribute nodes in a lattice manner. Note that, since the discovery frequency is very low in all three pulse switching architectures as discussed in Chapters 4-6, we do not consider the discovery power consumption in this chapter. Additionally, the protection and compression algorithms are disabled.

Table 9-1: System parameters

Symbol	Representation	Baseline Value
N	Nodes in the network	*441 (21 x 21) in HPS* *575 (5 x 115) in CPS and PTP*
d	Degree of network graph	*24*
H	Maximum hop count in HPS	*5*
M	Number of cells in CPS and PTP	*115*
α	Sector width (degree) in HPS	*No baseline*
γ	Wave-front routing envelope sector width (degree) in HPS	*No baseline*

Table 9-1 (cont'd)

δ	Sector-constraint in HPS / Route diversity in CPS and PTP	*No baseline*
k	Delay-traded sleep factor	*1*
$R_{\max}$, $R_{\min}$	Maximum and minimum transmission range	*1.5m, 1m*
T_b	Pulse repetition period (sec)	*1 μs*
T_w	Wake-up latency (sec)	*6 μs*
p	Packet preamble size (bit)	*32 bits (ZigBee)*
E_{tx}	Bit transmission cost (joule)	*4 nJ*
E_{rx}	Bit reception cost (joule)	*8 nJ*
P_{idle}	Idle consumption rate (watt)	*8 mW*
λ	Event generation rate (event/node/sec in HPS and CPS, event/cell/sec in PTP)	*No baseline*
l	Network load in PTP (event/ sec)	*No baseline*
φ	Network load factor in PTP (cell)	*No baseline*
R_{rnd}	Random number range in PTP	*10*

9.2 Comparable Packet Based Solution

For comparison, a TDMA packet-based event monitoring architecture is used. TDMA is chosen because of its high energy efficiency compared to the carrier sense based random access mechanisms such as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). TDMA is accomplished by avoiding packet collisions through pre-scheduled transmission slots. A TDMA [12] protocol that uses sink-rooted minimum spanning tree for packet routing has been used as the representative protocol to compare the proposed pulse switching protocol with. This representative protocol is expected to provide the best energy performance among all the TDMA based protocols [9-12, 25] discussed in Chapter 2.

An event detected by a sensor is reported to the sink using min-hop packet routing along the minimum spanning tree. As for MAC, TDMA slots are allocated [12] such that no two nodes that are within up to two transmission hops share the same time slot. This ensures that no direct and hidden collisions can occur. It is experimentally determined that for such slot allocations, the TDMA frame size needs to be at least approximately $(1.75d - 1.5)$ slots, where d is the maximum degree of the network graph. The maximum degree of the network graph is defined to be the maximum number of two-hop neighbors of a node in the network. This expression was found by fitting a curve on a large set of experimental data points representing the minimum TDMA frame size needed for collision-free allocation in networks with different d values.

As for event localization in HPS, each packet contains the minimum amount of information needed to represent a {*sector-id, hop-distance*} event-area. In CPS, each packet contains the minimum amount of information needed to represent {*source cell-id, next-hop cell-id*}. In PTP, each packet contains the minimum amount of information needed to represent {*source cell-id, destination cell-id, next-hop cell-id*}. Additionally, a per-packet preamble [3] is

needed. In this chapter, we do not consider the *Forwarding Area* in the frame structure of HPS.

9.3 Hop-angular Pulse Switching

9.3.1 Event Reporting Delay

For both pulse and packet modes, the reporting delay for an event can be computed by multiplying *h*, the hop-distance of origin from the sink, by the respective frame durations.

For the pulse mode, the frame duration (see Figure 4-3) is:

$$T_f^{HPSpulse} = T_b\{H(2+360°/\alpha)+4\}+T_w(H+2) \qquad (9\text{-}1)$$

For the TDMA allocation [12], described in Section 9.2, with maximum hop-distance *H*, sector-width α , and preamble size *p*, the required packet duration in HPS is $(p+\lceil\log_2(H)\rceil+\lceil\log_2(360°/\alpha)\rceil)$ bits. So the frame duration for the packet mode can be expressed as:

$$T_f^{HPSpkt} = (1.75d-1.5)\{T_w+T_b(p+\lceil\log_2(H)\rceil+\lceil\log_2(360°/\alpha)\rceil)\} \qquad (9\text{-}2)$$

Assuming no packet queuing, the worst case end-to-end event delays D_h^{pulse} and D_h^{pkt} can be computed by multiplying the hop-distance *h* with the frame durations in Eqns. 9-1 and 9-2. The worst case delays are D_h^{pulse} and D_h^{pkt} when *h* is equal to *H*. For pulse, with delay-traded sleep (see Section 7.4.1.3), the delay is scaled up by a constant factor *k*, representing the number of inserted *D* frames after each regular S, L and T frame.

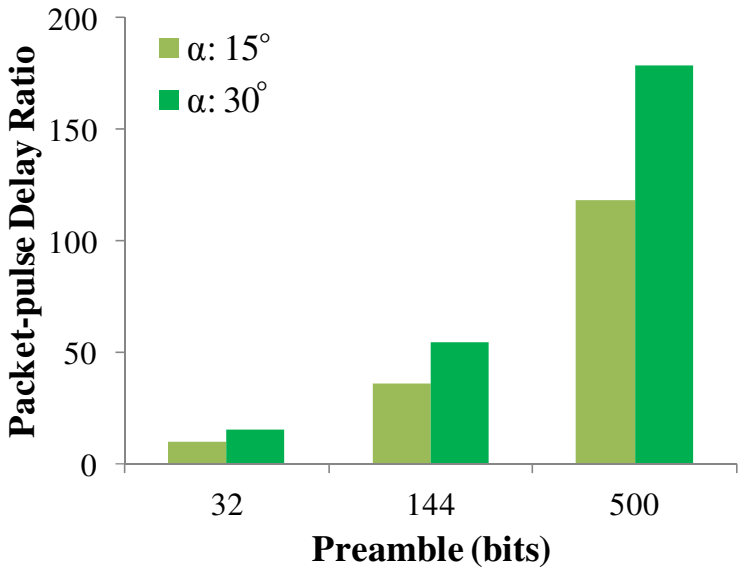


Figure 9-1: Packet-to-pulse delay ratio for HPS

Figure 9-1 plots the packet-to-pulse delay ratio D_h^{pkt} / D_h^{pulse} with k set to 1 for networks with lattice topologies. The pulse repetition period T_b and the transceiver wake-up/turn-around latency T_w are chosen to be *1000 ns* [31] and *6 µs* [22]. As in regular UWB, T_b is considered as the bit duration for the packet case. As expected from Eqns. 9-1 and 9-2, the delay ratio shows an upward trend when the packet preamble size is changed from 32 (for ZigBee), 144 (for 802.11) to 500 (for UWB [43]) bits. From Eqns. 9-1 and 9-2, the sector-width α influences the frame size for both packet and pulse, but as Figure 9-1 demonstrates, the pulse delay is better than the packet delay for all α values due to its smaller frame size. The delay ratio can be significant for large α and packet preambles. For example, in a network with α of $30°$, the packet delay (with 144 bit 802.11 preamble) can be up to approximately 54 times larger than that for pulse. Meaning, for pulse, ample number of D frames can be inserted for reducing the idling consumption, while its delay can still be lower than packet in a comparable network.

The absolute delay values with α of 30° for the packet case are 8.891 *ms*, 31.571 *ms*, 103.661 *ms* for the three preamble scenarios shown in Figure 9-1 and for the pulse case are 0.58 *ms*. With α of 15°, the absolute delay values for the packet case are 9.094 *ms*, 31.774 *ms*, 103.864 *ms* for the three preamble scenarios shown in Figure 9-1 and for the pulse case are 0.88 *ms*.

9.3.2 Power Consumption

9.3.2.1 Pulse Switching

In HPS, the number of transmissions during forwarding a pulse from hop-distance *h* is $2h\beta$, where the factor *2* represents the two pulses, one in *Control sub-frame* and the other in *Event sub-frame*. The route diversity factor β ($\beta \geq 1$) represents the total number of transmitted pulse pairs for routing a pulse from hop-distance *h,* normalized by *h*. Energy required to transmit and receive those pulses is $2h\beta(E_{tx} + \rho E_{rx})$. The parameter ρ ($\rho \geq 1$) represents *reception diversity*, which is the average number of nodes that receives each transmitted pulse. With an event generation rate λ events/node/sec, and the average hop distance $\hat{h}$, the per-node communication power consumption is:

$$\begin{aligned} P_{comm}^{HPSpulse} &= 2\lambda\beta(E_{tx} + \rho E_{rx}) \sum_{i \in [1,N]} h_i \Big/ N \\ &= 2\lambda\hat{h}\beta(E_{tx} + \rho E_{rx}) \end{aligned} \tag{9-3}$$

where *N* is the number of nodes in the network, and h_i is the hop distance for *node i* ($i \in [1, N]$).

As explained in Section 7.4.1.2, while forwarding a pulse from hop-distance *h*, an intermediate node has to remain awake for an entire slot-cluster corresponding to *h*. But only one slot in that cluster is used for the pulse reception. This leads to an idling energy consumption of $[T_w + (360°/\alpha - 1)T_b]P_{idle}$, where P_{idle} is idle consumption rate (*watt*). Therefore, the end-to-end idling consumption along the route is $\beta h[T_w + (360°/\alpha - 1)T_b]P_{idle}$. The average event idling expenditure due to pulse forwarding for each node can be written as:

$$\begin{aligned} P_{fwd\_idling}^{HPSpulse} &= \lambda\beta[T_w + (360°/\alpha - 1)T_b]P_{idle} \sum_{i\in[1,N]} h_i \Big/ N \\ &= \lambda\beta\hat{h}[T_w + (360°/\alpha - 1)T_b]P_{idle} \end{aligned} \quad (9\text{-}4)$$

With no events, the baseline idling at a node is caused due to its awake state during the downlink slots and the *Control sub-frame*, which adds up to the duration $(2H+4)T_b$ per frame. Therefore, the baseline idling consumption is $\left\lfloor (2H+4)T_b \Big/ T_f^{HPSpulse} \right\rfloor P_{idle}$. Due to the *S-L-T* state cycling, the effective consumption is scaled down by a factor 1/3, since a node is required to be up during the *Control sub-frame* of only one (the *L* frame) out of every three frames. Additionally, with delay-traded sleep (see Section 7.4.1.3), the idling energy can be slashed by a constant *k*, representing a factor by which the end-to-end event transport delay is extended by inserting the *D* frames. Considering these factors, the baseline idling consumption per node is:

$$P_{baseline\_idling}^{HPSpulse} = (1/k)(1/3)P_{idle}(2H+4)T_b \Big/ T_f^{HPSpulse} \quad (9\text{-}5)$$

Combining the baseline idling power consumption in Eqn. 9-5 with the forwarding idling power consumption in Eqn. 9-4, the total idling power consumption per node can be written as:

$$P_{idling}^{HPSpulse} = \lambda\beta\hat{h}[T_w + (360°/\alpha - 1)T_b]P_{idle} + (1/k)(1/3)P_{idle}(2H+4)T_b \Big/ T_f^{HPSpulse} \quad (9\text{-}6)$$

9.3.2.2 Packet Switching

Since $(p + \lceil \log_2(H) \rceil + \lceil \log_2(360°/\alpha) \rceil)$ represents the number of bits in a packet, the total number of bits transmitted during an *h*-hop routing is *h* times the number of bits per packet. Considering λ events/node/sec event generation rate, the average event hop distance $\hat{h}$, and the per bit transmission and reception energy costs of E_{tx} and E_{rx}, the per-node communication power consumption (*watt*) can be expressed as:

$$P_{comm}^{HPSpkt} = \hat{h}(p + \lceil \log_2(H) \rceil + \lceil \log_2(360°/\alpha) \rceil)(E_{tx} + E_{rx})\lambda \quad (9\text{-}7)$$

As for idling, the best case for the packet mode is that a node remains awake only for one TDMA slot during frame duration T_f^{HPSpkt}. Considering one wake-up slot T_w before each packet slot, the total idling consumption per node can be written as:

$$P_{idling}^{HPSpkt} = P_{idle}[T_w + T_b(p + \lceil \log_2(H) \rceil + \lceil \log_2(360°/\alpha) \rceil)] \Big/ T_f^{HPSpkt} \quad (9\text{-}8)$$

9.3.2.3 Numerical Model Results

9.3.2.3.1. Communication Power Consumption

The communication power consumption for pulse and packet obtained from Eqns. 9-3 and 9-7 are presented in Figure 9-2a. In the UWB spec. in Chapter 3, the transmission and reception consumptions are set to *4 nJ* and *8 nJ* per pulse [33] respectively, and the baseline idling consumption is taken as *8 mW* [33]. Since the transmission of a pulse using the baseband UWB

has the same energy expenditure for PPM modulated bits in a packet, the same *4 nJ* and *8 nJ* numbers are used for both pulse and bit transmission and reception.

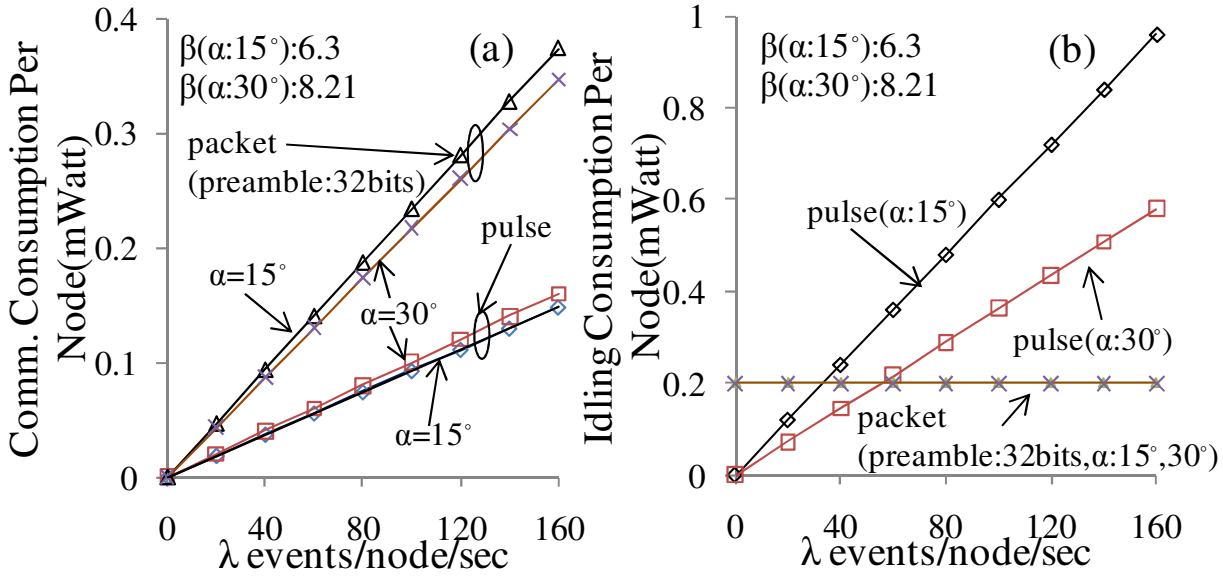


Figure 9-2: Communication and idling power consumption for pulse and packet in HPS

For results in Figures 9-2 and 9-3, the value of *k* is chosen to be 21 and 11 for sector-widths 30° and 15° respectively. At these *k* values, the transport delays for pulse and packet are equalized. From Eqn. 9-3, the communication consumption for pulse seems not to depend on α, but the route diversity β is greater with a higher α. From Eqn. 9-7, the communication consumption for packet decreases with increasing α. The reduction is significantly sub-linear due to the log term.

As expected, the communication consumption is linearly dependent on λ for both pulse and packet. The slope of the packet graph in Figure 9-2a is much higher than that for the pulse, and it is mainly due to the packet preamble *p*. This indicates that irrespective of the event rate, pulse switching is able to accomplish event reporting at a much lower communication cost compared to packets. The figure also shows that the communication consumption for the pulse is

higher for larger sector-width α which gives rise (see Eqn. 9-3) to higher route diversity β, leading to more number of pulse transmissions per event delivery. For packet, the dependency on α for communication consumption is quite marginal (see Eqn. 9-7) which explains its relative insensitivity to sector-width in Figure 9-2a.

9.3.2.3.2. Idling Power Consumption

From Eqn. 9-8, the idling power consumption for packet is not a function of λ, which is why it remains flat in Figure 9-2b. Also, the dependency on α is quite marginal, which is why the line for both $15°$ and $30°$ are virtually overlapping in Figure 9-2b. While the baseline idling component for pulse does not depend on λ (see Eqn. 9-5), the forwarding idling expenditure (see Eqn. 9-4) does depend on λ in a linear fashion. As a result, the total idling power consumption for pulse (Eqn. 9-8) linearly increases with λ; especially at higher λ and smaller α values. This implies that with higher event localization resolutions, the idling power consumption goes up at a higher rate with more frequent events in the network.

In Figure 9-2b, observe that for all α, the baseline idling power consumption for pulse (with zero event rates) is smaller than that of packet. With the increasing λ, depending on specific α, the idling power consumption for pulse crosses the packet line. However, for the practical parameters chosen in this analysis, this crossover occurs at very high event rates (30 events/nodes/sec and higher), indicating that the pulse outperforms packet in networks with low event rates, which the paradigm is primarily designed towards. For example, with an event generation rate of 0.5 events per node per second, which can be considered as a reasonably high event rate for the SHM style event monitoring application introduced in Chapter 1, the idling power consumption for the packet mode is an order of magnitude larger than that of the pulse

mode. For lower rates, the relative energy expenditure for the pulse mode is even lower. The results also show that as the event localization resolution increases (smaller α), the maximum allowable event rate up to which the idling cost for pulse is lower than packet reduces.

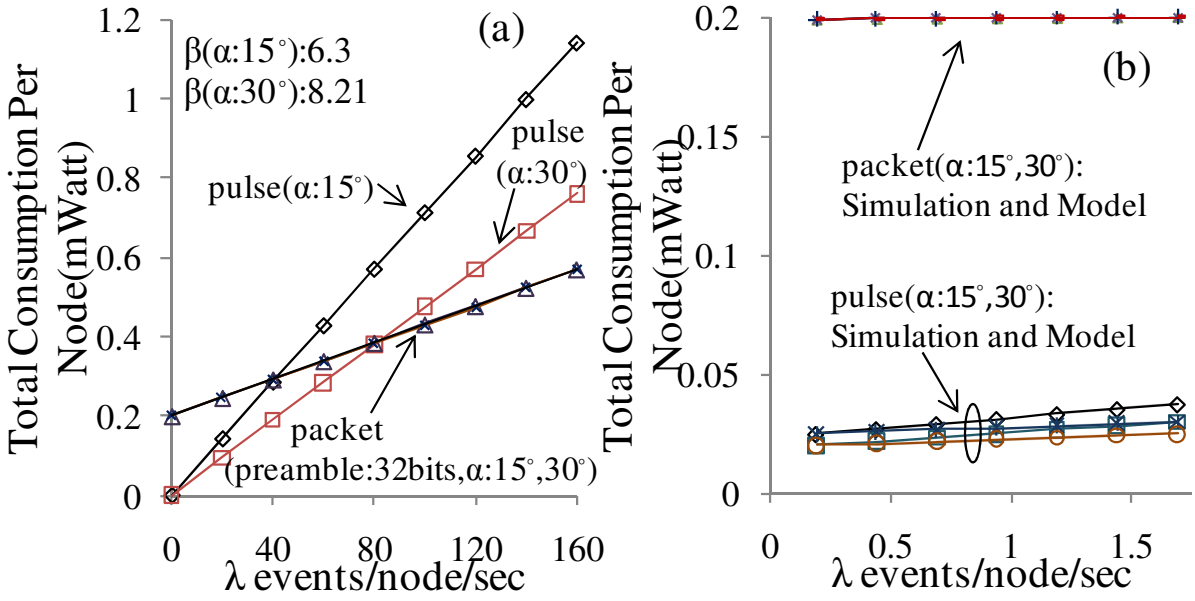


Figure 9-3: Total power consumption for pulse and packet switching in HPS

9.3.2.3.3. Total Consumption

Total power consumption including communication and idling are shown in Figure 9-3a. The relative pattern between packet and pulse for the total consumption is similar to that for idling. However, due to the communication expenditure, the total consumption rate for packet is not flat as in the idling. It goes up with event rate λ. As a result, for a larger α, the crossovers between the packet and the pulse lines occur at higher event rates, rendering a higher energy expenditure compared to the pulse mode. This is an important result indicating that with small angular resolutions of event localization (high α), pulse can significantly outperform packet for a large range of generation rates. From this point onwards, unless noted otherwise, all results of

performance evaluation in the following Chapter 10 correspond to event rates of 39 or 79 events/node/sec with α set as $15°$ or $30°$. These rates correspond to the crossover points between the packet and pulse as shown in Figure 9-3a.

9.3.2.4 Simulation Results

We have developed an event-driven C++ simulator which implements both pulse and packet switching as presented in Chapter 4. It implements MAC framing and routing, along with UWB-IR with the error model from Chapter 8. The model numbers for delay and energy obtained from Eqns. 9-3 through 9-8 in Section 9.3.2.3 were validated using results from the simulator for correctness. Figure 9-3b shows a close match between the energy numbers for both packet and pulse modes for event rates up to 1.5 events/node/sec; this rate can be considered as closed to the practical upper limits for SHM style event monitoring applications [1] that the hop-angular pulse framework is designed for.

9.4 Cellular Pulse Switching

9.4.1 Event Reporting Delay

For both pulse and packet modes, the reporting delay for an event can be computed by multiplying the cell-count/hop-count of origin away from the sink, by the respective frame durations.

For the pulse mode, the frame duration (see Figure 5-2) is:

$$T_f^{CPSpulse} = T_b\left(M^2 + 4M + 5\right) + T_w\left(2M + 2\right) \qquad (9\text{-}9)$$

For the TDMA allocation [12], described in Section 9.2, with an original *Cell-ID* i ($i \in [1, M]$), a next-hop *Cell-ID* j ($j \in [0, M]$), and preamble size p, the required maximum packet

duration in CPS is $\left(p+\lceil\log_2(M)\rceil+\lceil\log_2(M+1)\rceil\right)$ bits. So the frame duration for the packet mode can be expressed as:

$$T_f^{CPSpkt}=(1.75d-1.5)\{T_w+T_b\left(p+\lceil\log_2(M)\rceil+\lceil\log_2(M+1)\rceil\right)\} \qquad (9\text{-}10)$$

The worst case end-to-end event delay is derived from the scenario where the source cell is located farthest away from the sink. For pulse switching, the worst case end-to-end event delay D_{pulse}^{CPS} can be computed by multiplying the cell-count n_{cell} with the frame durations in Eqn. 9-9. Assuming no packet queuing, the worst case end-to-end event delay D_{pkt}^{CPS} for packet can be computed by multiplying the hop-count h with the frame durations in Eqn. 9-10. The hop-count h is the distance between the source and the sink divided by the transmission range. Thus, the packet-to-pulse delay ratio $D_{pkt}^{CPS}/D_{pulse}^{CPS}$ is expressed as $\left(hT_f^{CPSpkt}\right)/\left(n_{cell}T_f^{CPSpulse}\right)$.

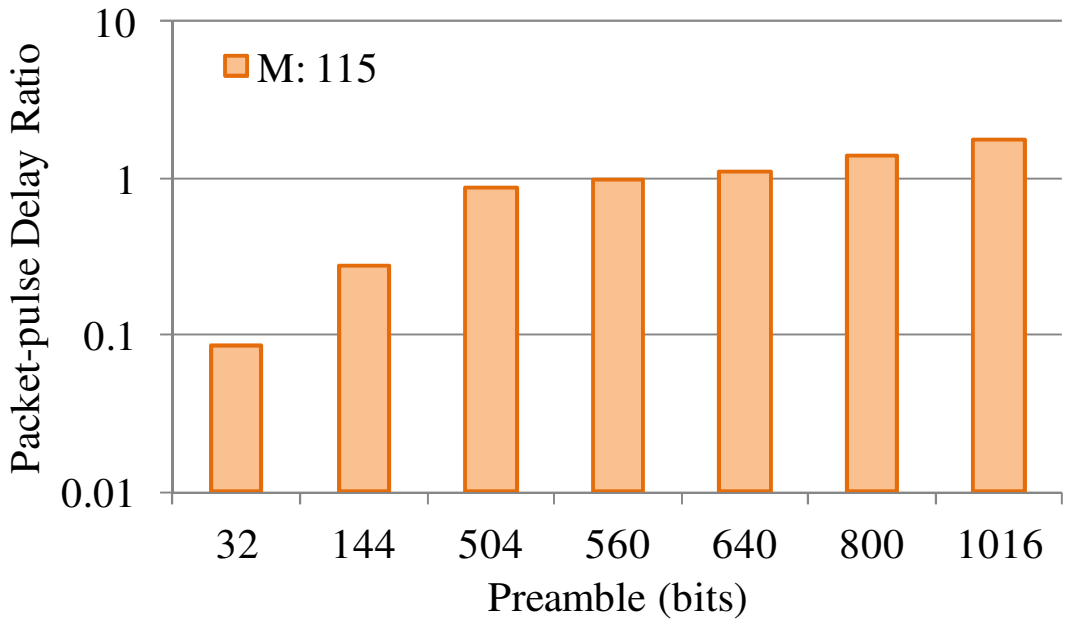


Figure 9-4: Packet-to-pulse delay ratio in CPS

We choose the network size to be 10×10 m^2. The network includes evenly distributed

sensor nodes and a sink node placed at the lower left corner of the terrain. The hop-count h for packet in the worst case is approximately 9.4. As shown in Section 5.2.3, the cellular event localization in CPS uses regular hexagonal cells with side length of 1. It is ensured that each node in a cell is able to reach at least one node in its neighboring cells. So the number of cells M in the network is 115. In the worst case, the cell-count n_{cell} is 15.

Figure 9-4 plots the packet-to-pulse delay ratio for the network described above. As expected from Eqns. 9-9 and 9-10, the delay ratio increases with the increasing packet preamble size (from 32 to 1016 bits [43]). As shown in Figure 9-4, the delay ratio is lower than 1 at the range of p from 32 to 500 bits, and higher than 1 at the range of p from 560 to 1016 bits. The packet solution outperforms pulse in terms of event reporting delay when the preamble size is greater than 560 bits. In addition, the absolute value of the worst-case event reporting delay for pulse D_{pulse}^{CPS} is about 0.226 sec. As concluded in Section 9.2.3.4, the practical upper limit of the event rate for the SHM style event monitoring applications [1] is 1.5 events/node/sec. Then the minimum interval per node between any two consecutive generated events is about 0.667 sec, which is three times of D_{pulse}^{CPS}. Although D_{pulse}^{CPS} is higher than D_{pkt}^{CPS} with preamble size smaller than 70 bytes, it is still small enough to satisfy the requirements of the SHM application.

9.4.2 Power Consumption

9.4.2.1 Pulse Switching

In CPS, for a given route diversity δ, the number of transmissions during forwarding a pulse from a cell of cell-count n away from the sink is $2n_{cell}\hat{N}$, where the factor *2* represents the

two pulses (one in *Control Area* and the other in *Localization Area*), n_{cell} represents the number of cells in an event route, and $\hat{N}$ is the average node count per cell. n_{cell} is related to the cell-count *n*, the route diversity δ, and the relative location of origin to the sink, $n_{cell} \in [n, \delta n)$. Energy required to transmit and receive those pulses is $2n_{cell}\hat{N}(E_{tx} + \rho E_{rx})$. The parameter ρ ($\rho \geq 1$) represents *reception diversity*, which is the average number of nodes that receives each transmitted pulse. With an event generation rate λ events/node/sec, and the average hop distance $\hat{h}$, the per-node communication power consumption:

$$\begin{aligned} P_{comm}^{CPSpulse} &= 2\lambda\hat{N}(E_{tx} + \rho E_{rx}) \sum_{i\in[1,N]} n_{cell_i} \Big/ N \\ &= 2\lambda\hat{n}_{cell}\hat{N}(E_{tx} + \rho E_{rx}) \end{aligned} \tag{9-11}$$

where $\hat{n}_{cell}$ is the average distance (cell-count) between a cell and the sink.

As explained in Section 7.4.2, while forwarding a pulse from *cell i*, an intermediate node has to remain awake for the entire i^{th} *Sensor Cell Area*. But only δ slots in that area are used for the pulse reception. This leads to an idling energy consumption of $[T_w + (M + 1 - \delta)T_b]P_{idle}$, where P_{idle} is idle consumption rate (*watt*). Therefore, the end-to-end idling consumption along the route is $n_{cell}\hat{N}[T_w + (M + 1 - \delta)T_b]P_{idle}$. The average event idling expenditure due to pulse forwarding for each node can be written as:

$$\begin{aligned} P_{fwd\_idling}^{CPSpulse} &= \lambda\hat{N}[T_w + (M + 1 - \delta)T_b]P_{idle} \sum_{i\in[1,N]} n_{cell_i} \Big/ N \\ &= \lambda\hat{n}_{cell}\hat{N}[T_w + (M + 1 - \delta)T_b]P_{idle} \end{aligned} \tag{9-12}$$

where $\hat{n}_{cell}$ is the average distance (cell-count) between a cell and the sink.

With no events, the baseline idling at a node is caused due to its awake state during the *Synchronization Area*, the *Discovery Area* and the *Control Area*, which adds up to the duration $2(M+1)T_b$ per frame. Therefore, the baseline idling consumption per node is:

$$P_{baseline\_idling}^{CPSpulse} = \left(2(M+1)T_b \Big/ T_f^{CPSpulse}\right)P_{idle} \tag{9-13}$$

Combining the baseline idling power consumption in Eqn. 9-13 with the forwarding idling power consumption in Eqn. 9-12, the total idling power consumption per node can be written as:

$$\begin{aligned} P_{idling}^{CPSpulse} &= \lambda \hat{n}_{cell} \hat{N}[T_w + (M+1-\delta)T_b]P_{idle} \\ &+ \left[2(M+1)T_b \Big/ T_f^{CPSpulse}\right]P_{idle} \end{aligned} \tag{9-14}$$

9.4.2.2 Packet Switching

Since $\left(p + \lceil \log_2(M) \rceil + \lceil \log_2(M+1) \rceil\right)$ represents the number of bits in a packet, the total number of bits transmitted during an *h*-hop routing is *h* times the number of bits per packet. Considering λ events/node/sec event generation rate, the average event hop distance $\hat{h}$, and the per bit transmission and reception energy costs of E_{tx} and E_{rx}, the per-node communication power consumption (*watt*) can be expressed as:

$$P_{comm}^{CPSpkt} = \hat{h}\left(p + \lceil \log_2(M) \rceil + \lceil \log_2(M+1) \rceil\right)(E_{tx} + E_{rx})\lambda \tag{9-15}$$

As for idling, the best case for the packet mode is that a node remains awake only for one TDMA slot during frame duration T_f^{CPSpkt}. Considering one wake-up slot T_w before each packet slot, the total idling consumption per node can be written as:

$$P_{idling}^{CPSpkt} = P_{idle}\left[T_w + T_b\left(p + \lceil \log_2(M) \rceil + \lceil \log_2(M+1) \rceil\right)\right] \Big/ T_f^{CPSpkt} \tag{9-16}$$

9.4.2.3 Numerical Results

In the following results of pulse power consumption, we set M to be 115 and route diversity δ to be 1. According to the $10m \times 10m$ network given in Chapter 11, the average cell-count $\hat{n}_{cell}$ is 8.6, the average node count per cell is 5, and the reception diversity factor ρ is estimated to be 10. For packet power consumption computations, we estimate the average hop-count $\hat{h}$ to be 5.5 when the transmission range is 1.5 m in the $10m \times 10m$ network.

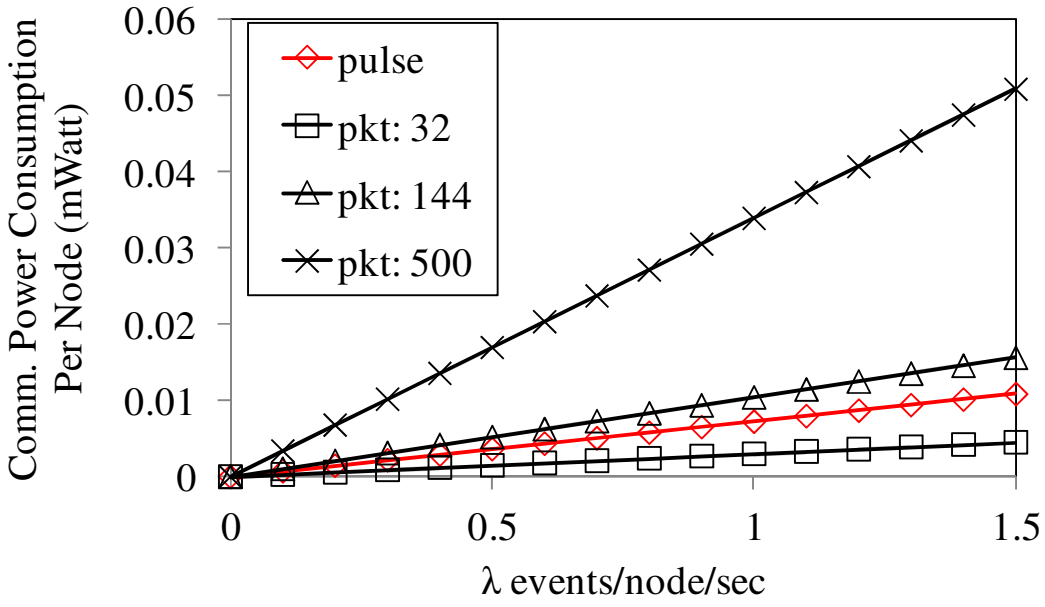


Figure 9-5: Communication power consumption for pulse and packet switching in CPS

9.4.2.3.1. Communication Power Consumption

The communication power consumption for pulse and packet obtained from Eqns. 9-11 and 9-15 are presented in Figure 9-5. Due to the fact that the cellular pulse switching is mainly designed for event monitoring applications, the event rate can be very low. We set the maximum event rate to be 1.5 events/node/sec. As expected, the communication consumption is linearly dependent on λ for both pulse and packet. In addition, the communication consumption of the

packet is higher with a larger preamble size for a given event rate. The slopes of the packet lines corresponding to the packet preamble p to be 144 and 500 bits in Figure 9-5 are higher than that for the pulse, but the slope of the packet line with p to be 32 bits is slightly lower than that for the pulse. It indicates that irrespective of the event rate, pulse switching is able to accomplish event reporting at a much lower communication cost compared to packet switching within the preamble size range $(32,1016]$ bits.

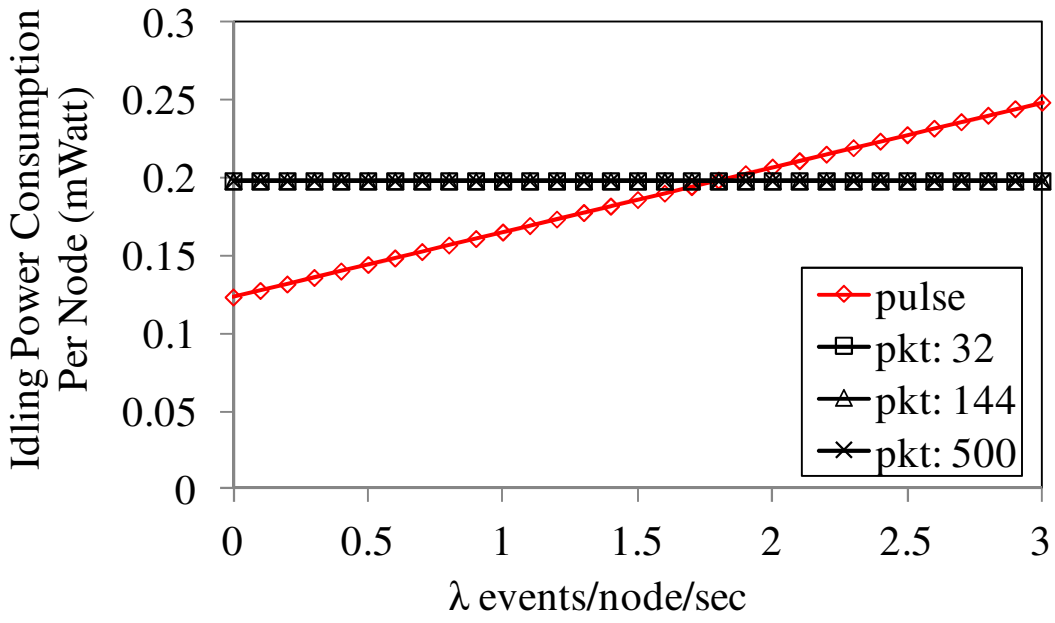


Figure 9-6: Idling power consumption for pulse and packet switching in CPS

9.4.2.3.2. Idling Power Consumption

From Eqn. 9-16, the idling power consumption for packet is not a function of λ, which is why it remains flat in Figure 9-6. While the baseline idling component of the pulse does not depend on λ (see Eqn. 9-13), the forwarding idling expenditure (see Eqn. 9-12) does depend on λ in a linear fashion. As a result, the total idling power consumption for pulse (Eqn. 9-14) linearly increases with λ.

In Figure 9-6, observe that for all preamble size p, the baseline idling power consumption for pulse (with zero event rates) is smaller than that of packet. With the increasing λ, depending on specific p, the idling power consumption for pulse may cross the packet line. However, for the practical parameters chosen in this analysis, this crossover occurs at an event rate higher than 1.5 events/nodes/sec. It shows that the pulse outperforms packet in networks with low event rates, which the paradigm is primarily designed towards. For example, an event generation rate of 0.5 events per node per second can be considered as a reasonably high event rate for the SHM style event monitoring application introduced in Chapter 1.

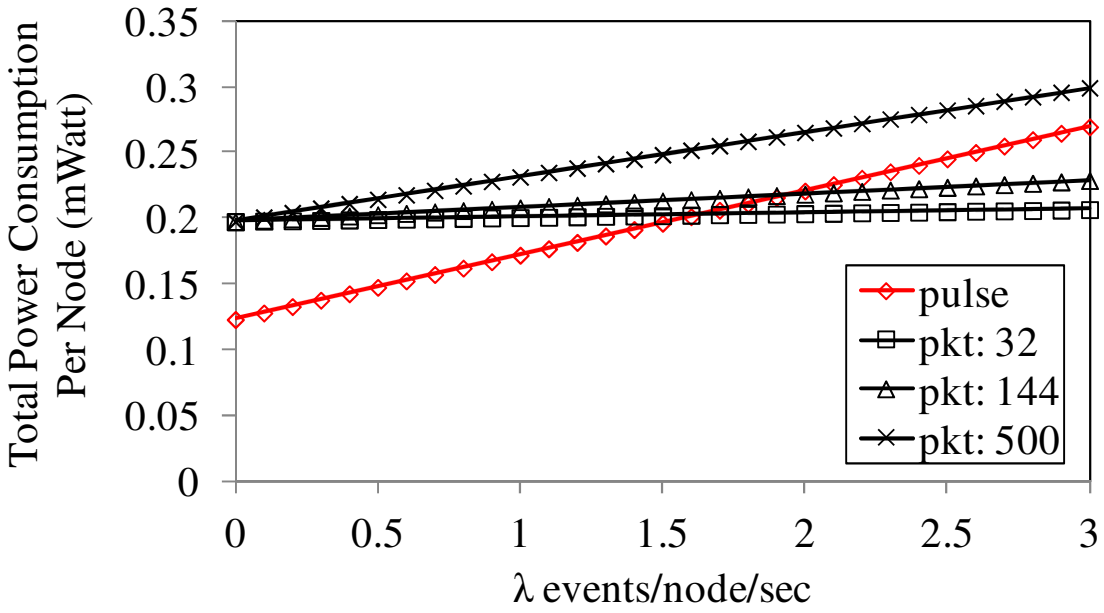


Figure 9-7: Total power consumption for pulse and packet switching in CPS

9.4.2.3.3. Total Consumption

Total power consumption including communication and idling are shown in Figure 9-7. The relative pattern between packet and pulse for the total consumption is similar to that for idling. However, due to the communication expenditure, the total consumption rate for packet is

not flat as in the idling. It goes up slightly with event rate λ, which is not obvious in the figure. As shown in Figure 9-7, pulse outperforms packet in terms of total power consumption with low event rates, which the paradigm is primarily designed towards.

9.5 Point-to-point Pulse Switching

9.5.1 Collision Modeling

9.5.1.1 Motivation

In Chapter 6, we proposed the collision handling measure at presence of multi-event collisions. For a receiver node, after the multi-event collision detection, it responds to each involved sender node with a collided response pulse. Upon the reception of such a response pulse, each sender retransmits its previously transmitted event after a randomly selected back-off period. The above collided response and back-off retransmission process continues until that these randomly selected back-off periods are exclusively different. Intuitively, either higher event generation rates or lower random number ranges of the back-off process lead to the higher chances of multi-event collision occurrences. More collisions result in more energy consumptions and worse event reporting delays. Thus, this subsection investigates the impacts of the event rate and random number range on the delay and power consumption for PTP.

9.5.1.2 Model Description

For simplicity, we assume that each cell has one node. In addition, each cell in the network has the same event generation rate λ (event/cell/sec). Considering that PTP is designed for very low-frequency events, the value of λ can be very low ($\lambda \in [0,1.5]$). If a cell generates an event, all other events generated by the rest of network may collide with this event in the same frame. It

is regarded as the worst case. Such collided events are defined to be the network load l. Formally stated, the network load l for a cell is defined to be the summation of event rates of the rest of network cells. So $l = \varphi\lambda$, where φ is the network load factor ($\varphi >> 1$).

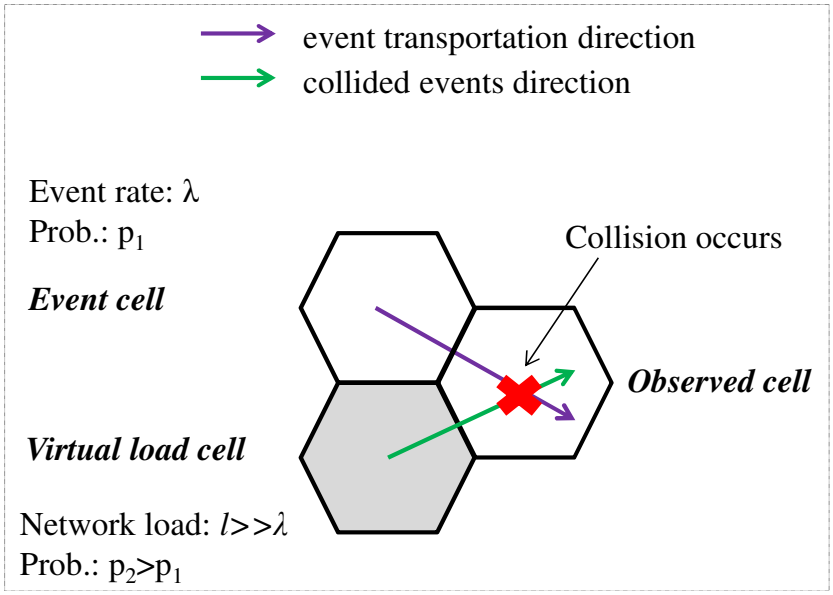


Figure 9-8: Collision modeling diagram for PTP

For any cell with an event rate λ , the impact of the network load l on the delay and power consumption can be derived from the following model. As shown in Figure 9-8, we construct a model containing three special cells: 1) an event cell with an event rate λ , 2) a virtual load cell with an event rate l, and 3) an observed cell which is the next-hop cell of the event cell and virtual load cell. Collisions happen in the observed cell due to events from the event cell and virtual load cell are both forwarded to the observed cell. Then the collided response and back-off retransmission process is executed between the observed cell and the sender cells (i.e., the event cell and virtual load cell). Accordingly, such processes increase the power consumption and delay. The delay is defined to be the duration when events from the event cell are all successfully

forwarded through the observed cell. During the delay, the power consumption is counted within the observed cell.

9.5.1.3 Probabilistic Modeling

For pulses, the models are constructed to compute the expected power consumption and the expected delay of the observed cell for a given network event generation rate λ and a network load *l*.

The probability p_1 of an event generated in the event cell per frame can be derived by $p_1 = \lambda T_f^{PTPpulse}$, where $T_f^{PTPpulse}$ represents the frame duration of PTP. Similarly, the probability p_2 of an event generated in the virtual load cell per frame is expressed as $p_2 = lT_f^{PTPpulse}$. With $l = \varphi\lambda$, p_2 can be expressed as $\varphi\lambda T_f^{PTPpulse}$ as well. Both probabilities are upper-bounded by 1. As a result, the event rate λ and the network load *l* are both lower or equal to $1/T_f^{PTPpulse}$.

With the knowledge of p_1 and p_2, the probability of a collision occurrence in a frame at the observed cell is $p_1 p_2$. According to the collided response and back-off retransmission process, the event cell and virtual load cell retransmit events after randomly selected back-off periods, named as R_1 and R_2. The probability of $R_1 \neq R_2$ is expressed as $(R_{rnd} - 1)/R_{rnd}$, where R_{rnd} is the random number range. Specifically, $(R_{rnd} - 1)/R_{rnd}$ approaches 1 when R_{rnd} is very large. Even though R_{rnd} is not very large, such as 10, $(R_{rnd} - 1)/R_{rnd}$ is still 0.9. It shows that the chances of the event cell and virtual load cell retransmit events in the same frame are very low. That is to say, the probability of the second-time collisions after the first-

time retransmissions is very small. So we assume that the senders (i.e., the event cell and the virtual load cell) retransmit events in different back-off periods. Furthermore, the retransmitted event pulse from the event cell might collide with another newly generated pulse from the virtual load cell with the probability of $p_1(p_2)^2$. Similarly, the event pulse from the event cell collides with a newly generated pulse from the virtual load cell for the k consecutive times has the probability to be $p_1(p_2)^k$, where $k \in [0,\infty]$.

9.5.2 Event Reporting Delay

9.5.2.1 Pulse Switching

Considering route discovery in PTP is executed with a very low frequency, the frame duration can be derived from the *Event Report Frame* (see Figure 6-4). Thus, the frame duration for pulse switching is:

$$T_f^{PTPpulse} = T_b(6M+4) + 4T_w \tag{9-17}$$

If an event pulse from the event cell collides with a pulse from the virtual load cell for the k consecutive times, the cumulative delay during this process can be maximally $(kR_{rnd}+2)T_f^{PTPpulse}$. The summation of the products of the cumulative delay $(kR_{rnd}+2)T_f^{PTPpulse}$ and the corresponding probability $p_1(p_2)^k$ across the range of k ($k \in [0,\infty]$) is equal to the expected delay $D_{\exp}^{PTPpulse}$ of the observed cell for a given λ and φ. In Section 9.5.1.3, we derived $p_1 = \lambda T_f^{PTPpulse}$ and $p_2 = \varphi\lambda T_f^{PTPpulse}$. Therefore, $D_{\exp}^{PTP}$ can be calculated by:

$$
\begin{aligned}
D_{\exp}^{PTP} &= \sum_{k\in[0,\infty]} (kR_{rnd}+2) T_f^{PTPpulse} p_1 (p_2)^k \\
&= \sum_{k\in[0,\infty]} R_{rnd} T_f^{PTPpulse} p_1 k (p_2)^k + \sum_{k\in[0,\infty]} 2T_f^{PTPpulse} p_1 (p_2)^k \\
&= R_{rnd} T_f^{PTPpulse} p_1 \big/ (1-p_2)^2 + 2T_f^{PTPpulse} p_1 \big/ (1-p_2) \\
&= T_f^{PTPpulse} \left[(R_{rnd}+2) p_1 - 2p_1 p_2 \right] \big/ (1-p_2)^2 \\
&= p_1 T_f^{PTPpulse} \left[(R_{rnd}+2) - 2p_2 \right] \big/ (1-p_2)^2 \\
&= \lambda \left(T_f^{PTPpulse} \right)^2 \left[(R_{rnd}+2) - 2\varphi\lambda T_f^{PTPpulse} \right] \big/ \left(1-\varphi\lambda T_f^{PTPpulse} \right)^2
\end{aligned}
\tag{9-18}
$$

9.5.2.2 Packet Switching

As for event localization, each packet contains the minimum amount of information needed to represent {*source cell-id, destination cell-id, next-hop cell-id*}. Additionally, a per-packet preamble is needed. Therefore, with maximum *M*, route diversity δ, and preamble size *p*, the required packet duration is $p + \lceil (2+\delta) \log_2 \lceil M \rceil \rceil$ bits.

Due to no collision occurrences in the packet-based solution, an event generated by the event cell is unrelated to events generated by the virtual load cell, as shown in Figure 9-8. For the same event rate λ as that in pulse switching, the probability of an event generated in the event cell per frame for packets can be computed by $p = \lambda T_f^{PTPpkt}$, where T_f^{PTPpkt} is the frame duration of packet switching. Such a probability is upper-bounded by 1. Due to different frame durations of pulse and packet switching, the upper bounds of the event rates in two cases are different. The frame duration of packet switching is:

$$
T_f^{PTPpkt} = (1.75d - 1.5) \{ T_b (p + \lceil (2+\delta) \log_2 \lceil M \rceil \rceil) + T_w \} \tag{9-19}
$$

The expected delay $D_{\exp}^{PTPpkt}$ of the observed cell for the given λ is expressed as:

$$
\begin{aligned}
D_{\exp}^{PTPpkt} &= pT_f^{PTPpkt} \\
&= \lambda\left(T_f^{PTPpkt}\right)^2
\end{aligned}
\tag{9-20}
$$

9.5.3 Power Consumption

9.5.3.1 Pulse Switching

During the collided response and back-off retransmission process, the observed cell sends response pulses to the event cell and virtual load cell. When the observed cell forwards the event pulses successfully, collisions are resolved. We assume that the event pulse from the event cell collides with the pulse from the virtual load cell for the k consecutive times ($k \in [0,\infty]$). In this case, the probability is $p_1(p_2)^k$. The pulse transmissions of the observed cell include the collided response pulse transmissions (i.e., $(6k+3)$) to the sender cells and the successful pulse transmissions (i.e., $(2+\delta)$) to the next-hop cells. So the total pulse transmission count of the observed cell in this case is $(6k+5+\delta)$, where δ is the route diversity. Thus the expected pulse transmission count of the observed cell for an event pulse with a given λ and φ as follows:

$$
\begin{aligned}
N_{tx}^{\exp} &= \sum_{k\in[0,\infty]}(6k+5+\delta)p_1(p_2)^k \\
&= \sum_{k\in[0,\infty]}6p_1k(p_2)^k + \sum_{k\in[0,\infty]}(5+\delta)p_1(p_2)^k \\
&= 6p_1/(1-p_2)^2 + p_1(5+\delta)/(1-p_2) \\
&= [(11+\delta)p_1 - (5+\delta)p_1p_2]/(1-p_2)^2 \\
&= \left[(11+\delta)\lambda T_f^{PTPpulse} - (5+\delta)\varphi\lambda^2\left(T_f^{PTPpulse}\right)^2\right] \Big/ \left(1-\varphi\lambda T_f^{PTPpulse}\right)^2
\end{aligned}
\tag{9-21}
$$

Similarly, pulse receptions of the observed cell are that from the sender cells. In this case, the pulse reception count is $((4+2\delta-\alpha)k+2+\delta)$, where α is the number of next-hops that the

event cell and virtual load cell share with each other ($\alpha \in [0, \delta]$). Therefore, the expected pulse reception count of the observed cell for an event pulse with a given λ and φ is:

$$\begin{aligned} N_{rx}^{\exp} &= \sum_{k\in[0,\infty]}[(4+2\delta-\alpha)k+2+\delta]p_1(p_2)^k \\ &= \sum_{k\in[0,\infty]}(4+2\delta-\alpha)p_1k(p_2)^k + \sum_{k\in[0,\infty]}(2+\delta)p_1(p_2)^k \\ &= (4+2\delta-\alpha)p_1/(1-p_2)^2 + p_1(2+\delta)/(1-p_2) \\ &= [(6+3\delta-\alpha)p_1-(2+\delta)p_1p_2]/(1-p_2)^2 \\ &= \left[(6+3\delta-\alpha)\lambda T_f^{PTPpulse}-(2+\delta)\varphi\lambda^2\left(T_f^{PTPpulse}\right)^2\right]/\left(1-\varphi\lambda T_f^{PTPpulse}\right)^2 \end{aligned} \tag{9-22}$$

The energy required to transmit and receive those pulses is $N_{tx}^{\exp}E_{tx} + N_{rx}^{\exp}E_{rx}$, where E_{tx} is the pulse transmission cost and E_{rx} is the pulse reception cost. For the given event rate λ and network load factor φ, the expected communication power consumption of the observed cell can be computed by:

$$P_{comm}^{PTPpulse} = \lambda(N_{tx}^{\exp}E_{tx} + N_{rx}^{\exp}E_{rx}) \tag{9-23}$$

Combining Eqns. 9-21, 9-22 and 9-23, the communication power consumption per cell can be computed.

During the event report phase, there are unoccupied slots in a frame, which contribute to the idling power consumption. Thus, the expected idling duration of the observed cell for a given λ and φ is:

$$\begin{aligned} T_{idling}^{\exp} &= T_b\sum_{k\in[0,\infty]}[(3M-3+2\gamma)k+3M-2]p_1(p_2)^k \\ &= T_b\sum_{k\in[0,\infty]}(3M-3+2\gamma)p_1k(p_2)^k + T_b\sum_{k\in[0,\infty]}(3M-2)p_1(p_2)^k \\ &= T_b(3M-3+2\gamma)p_1/(1-p_2)^2 + T_b(3M-2)p_1/(1-p_2) \\ &= T_b[(6M-5+2\gamma)p_1-(3M-2)p_1p_2]/(1-p_2)^2 \\ &= T_b\left[\lambda T_f^{PTPpulse}(6M-5+2\gamma)-\varphi\lambda^2\left(T_f^{PTPpulse}\right)^2(3M-2)\right]/(1-\varphi\lambda T_f^{PTPpulse})^2 \end{aligned} \tag{9-24}$$

Where γ is the transmission turnaround time factor (see Table 1).

Thus, the expected idling power consumption of the observed cell for the given event rate λ and network load factor φ is expressed as:

$$P_{idling}^{PTPpulse} = \lambda P_{idle} T_{idling}^{\exp} \tag{9-25}$$

Combining Eqns. 9-24 and 9-25, we can derive the expected idling power consumption of the observed cell.

9.5.3.2 Packet Switching

For an event from the event cell, the transmitted and received packet bits of the observed cell are equally $\left[p + \left\lceil (2+\delta)\log_2 M \right\rceil\right]$. Considering the event rate λ, and the bit transmission and reception energy costs of E_{tx} and E_{rx}, the communication power consumption of the observed cell can be expressed as:

$$P_{comm}^{PTPpkt} = \lambda^2 T_f^{PTPpkt} \left[p + \left\lceil (2+\delta)\log_2 M \right\rceil\right](E_{tx} + E_{rx}) \tag{9-26}$$

As for idling, the best case for the packet mode is that a node remains awake only for one TDMA slot during frame duration T_f^{PTPpkt}. Considering one wake-up slot T_w before each packet slot, the total idling consumption of the observed cell can be written as:

$$P_{idling}^{PTPpkt} = \lambda P_{idle} \left\{ T_w + T_b \left[p + \left\lceil (2+\delta)\log_2 M \right\rceil\right] \right\} \tag{9-27}$$

9.5.4 Numerical Results

This subsection shows the results derived from the models (i.e., Eqns. 9-17 to 9-27) summarized in Sections 9.5.2 and 9.5.3. In the model calculation, we set λ to be 0.5 events/cell/sec and l to be minimally 1.5 events/sec. M is chosen to be 115 and δ is set to be 1.

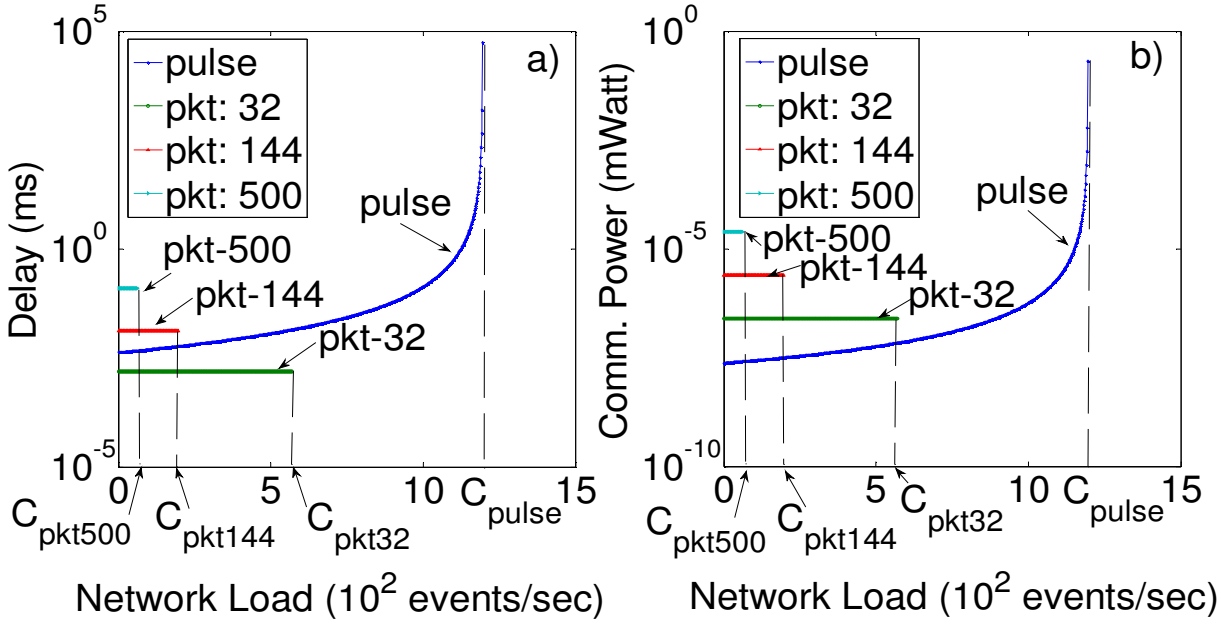


Figure 9-9: Delay and communication power consumption between pulse and packet in PTP

9.5.4.1 Event Reporting Delay

For both pulse and packet modes, the event reporting delay per cell (e.g., the observed cell in Figure 9-8) is defined to the average time duration when an event is generated in the event cell and then is successfully forwarded by the observed cell. For a given event rate λ, the impacts of the network load l on the delay for pulse and packet switching are shown in Figure 9-9a. The results are obtained from Eqns. 9-18 and 9-20. As modeled in Eqn. 9-18, the expected delay for pulse switching is an increasing function of the network load l for the given event rate λ, the pulse frame duration $T_f^{PTPpulse}$ and the back-off random number range R_{rnd}. In other words,

the per-cell event reporting delay increases with the increasing network load factor φ. And the delay would be infinitely increasing when the network load is approaching the network capacity C_{pulse}.

As shown in the figure, the lines representing event reporting delays for packets with different preamble sizes are flat within the corresponding network capacities. In addition, the delay is higher with a larger preamble size for the packet solution. Note that the network capacity increases with the decreasing preamble size for packet switching. As discussed in Section 9.5.1.3, the probability of an event generated in the event cell per frame for packets is upper-bounded by 1. $p = \lambda T_f^{PTPpkt}$, where T_f^{PTPpkt} is the frame duration. So the event rate λ for packets should be lower or equal to $1/T_f^{PTPpkt}$. Since that the preamble size determines the frame duration for the given *M* and δ as expressed in Eqn. 9-19, it also impacts the upper bound of the event rate (i.e., the network capacity). For example, as shown in Figure 9-9, C_{pkt-32}, as the network capacity of the packet solution with the preamble size equal to 32 bits, is higher than $C_{pkt-144}$ as the network capacity of the packet solution with the preamble size of 144 bits.

Due to different network capacities for pulse switching and three packet-based solutions, we select the minimum network capacity $C_{pkt-500}$ to compare the delay performance. The delay for the packet with the 32-bit preamble, as the lowest one among these packet solutions, is lower than the delay for pulse. However, pulse switching performs better in terms of event reporting delay than packet switching with the preamble size larger than or equal to 32 bits.

9.5.4.2 Power Consumption

9.5.4.2.1. Communication Power Consumption

For a given event rate per cell, the impacts of the network load l on the communication power consumption per cell for pulse and packet switching are shown in Figure 9-9b. The results are obtained from Eqns. 9-23 and 9-26.

As expected, the per-cell communication power consumption increases with the increasingly heavy network load l. In other words, the per-cell communication power consumption is an increasing function of the network load factor φ according to Eqn. 9-23, for the given λ and φ. Due to the heavier network load, collisions happen more frequently which accordingly consumes more energy. When the network load l is approaching the network capacity C_{pulse}, the communication power consumption is infinitely increased. It results from the fact that the collided response and back-off retransmission mechanism is unable to resolve such a huge amount of collisions.

Since no packet collision occurrences in the packet-based solution, the network load l does not impact the packet transportations for a given event rate λ. As a result, the per-cell communication power consumption keeps the same value within the network capacity for the packet-based solution with a specific preamble size. Additionally, with a greater preamble size, packet switching consumes more energy for a given event rate λ.

Within the minimum network capacity $C_{pkt-500}$, pulse switching has the lowest per-cell communication power for a given event rate λ per cell. In addition, pulse switching is still more

efficient in terms of communication power within the largest capacity of packet solutions C_{pkt-32}.

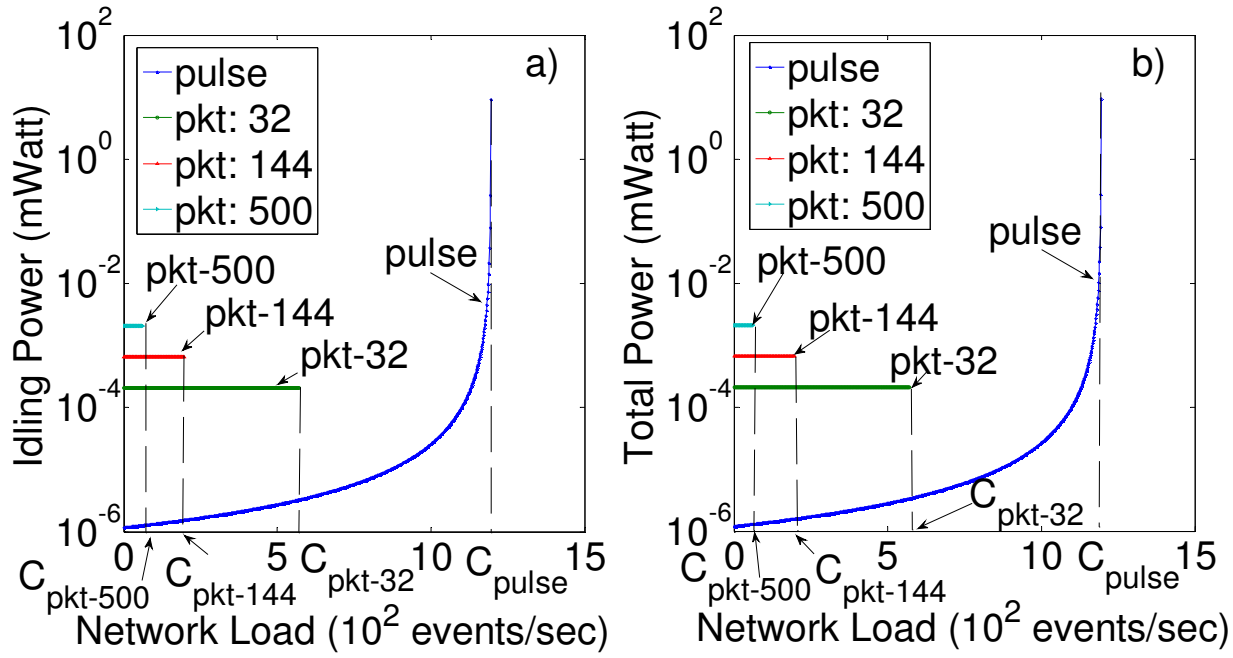


Figure 9-10: Idling and total power consumption between pulse and packet switching in PTP

9.5.4.2.2. *Idling Power Consumption*

For a given event rate λ per cell, the impacts of the network load l on the idling power consumption per cell for pulse and packet switching are shown in Figure 9-10a. The results are obtained from Eqns. 9-25 and 9-27.

With the increasing network load l, the per-cell idling power consumption increases. As shown in Figure 9-8, more events generated from the virtual load cell within a given time period (corresponding to a higher l), collisions occur more frequently. It leads to more pulse retransmissions and receptions between the sender cells (i.e., the event cell and the virtual load cell) and the observed cell. In this situation, the observed cell has to remain awake for more frames. Similar as in the above analysis for communication power consumption, the per-cell

idling power consumption would be infinitely increasing when the network load is approaching C_{pulse}.

Same as in the analysis for the communication power consumption, the per-cell idling power consumption for packet switching is not a function of the network load *l*, which is why it remains flat in Figure 9-10a with a specific preamble size. As modeled in Eqn. 9-27, the idling power consumption is linearly dependent on the preamble size *p* for the given λ, *M* and δ. So the packet-based solution with a larger preamble size has the higher idling power consumption.

Within $C_{pkt-500}$, pulse switching has the lowest per-cell idling power for a given event rate λ per cell. Even within the largest capacity of packet solutions C_{pkt-32}, pulse switching is still more efficient in terms of idling power.

9.5.4.2.3. *Total Power Consumption*

Total power consumption including communication and idling are shown in Figure 9-10b. The relative pattern between packet and pulse for the total consumption is similar to that for communication and idling. Therefore, pulse switching can significantly outperform packet switching for a large range of the network load (between 0 events/s and 150 events/s). Moreover, pulse switching benefits from the highest network capacity compared to these three packet-based solutions.

9.6 Summary and Conclusion

This chapter first introduced the comparable packet-based solution. Then it respectively constructed the models for comparing the proposed three pulse switching architectures and the corresponding packet-based TDMA paradigm in terms of delay and power consumption. In

general, the results derived from these models show that pulse switching outperforms packet switching both in terms of delay and power consumption with a relatively low event rate.

Chapter 10: Performance of Hop-angular Pulse Switching

10.1 Introduction

We developed an event-driven C++ simulator which implements MAC framing and pulse routing with the hop-angular localization mechanism as proposed in Chapter 4. The proposed Hop-angular Pulse Switching (HPS) protocol uses the UWB-IR model as presented in Chapter 3. The network terrain size is set to 10×10 meter^2 with 441 sensor nodes, and a sink node which is placed at the center of the terrain.

Each event has a scope *S*, which represents the number of sensors that detect the event and individually generates a pulse. Scope is introduced for capturing scenarios in which a single physical event (e.g., a crack in a structure) can generate multiple simultaneous pulses from different sensors. Depending on the location of an event and its scope, pulses from multiple nodes, possibly from different event-areas (see Figure 4-1), can be generated. Each transported pulse to the sink provides localization about the event-area of its originating sensor. If at least one pulse from an event is delivered to the sink, the event is considered reported. Unless stated otherwise, an event scope of 1 is used throughout the analysis of this chapter.

10.2 Static Event Scenario

This section presents results when a single pulse is generated for a single static event from within an event area. Unless stated otherwise, all results correspond to without compression scenarios in this section.

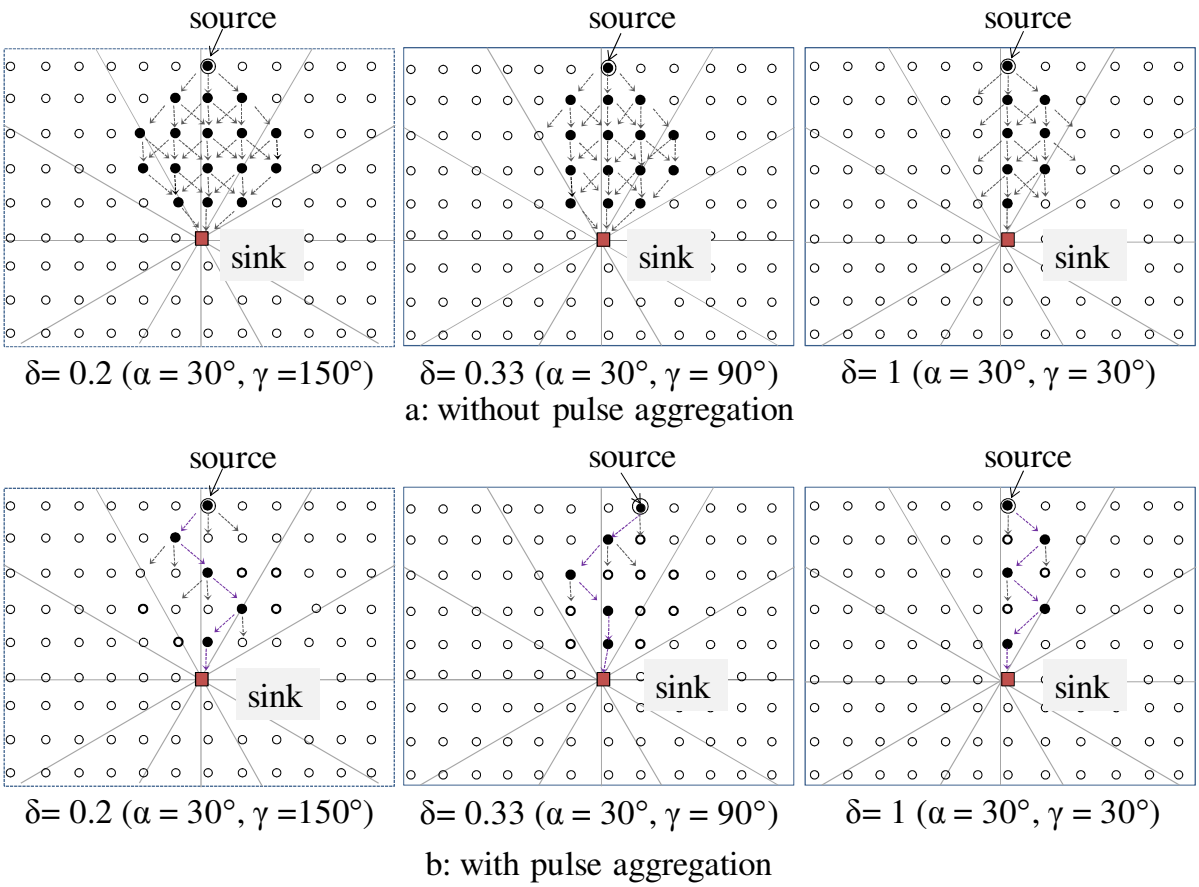


Figure 10-1: Wave-front routing envelope due to pulse fan-out in HPS

10.2.1 Impacts of Sector-constraint on Wave-front Routing

In this subsection, we only consider the case of no compression (NC). Figure 10-1a shows an instance of routing in which a pulse is routed from the source to the sink when α is $30°$. The dark dots represents nodes that have actively received (was in the *L* state and received) the pulse. The spatial extent of flooding is shown with the *sector constraint* δ set to 0.2, 0.33 and 1. Observe how the number of participating nodes in routing (i.e., the route diversity β) reduces with higher δ. Also note the shape of the routing envelope with varying δ. The pulse from the source first fans laterally outwards, but as it nears the sink it converges laterally inwards, thus the energy overhead of routing.

Figure 10-2a reports the number of pulse transmissions across different hop areas for an event generated at hop-distance 5 in a 441 node network. Numbers are reported for two different sector-constraints ($\delta = 0.2$ and $\delta = 1$). For both the cases, observe how the number of pulse transmissions maximizes at an intermediate hop-distance, confirming the lateral fan-out and convergence seen as the routing envelopes in Figure 10-1a.

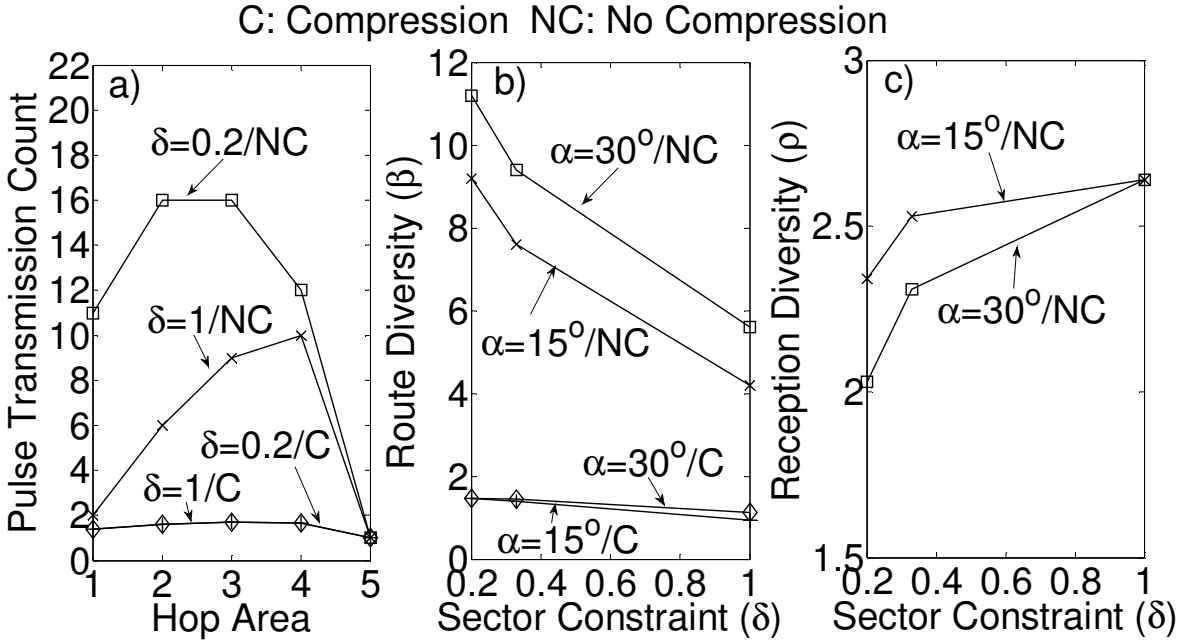


Figure 10-2: Pulse transmission count, route diversity and reception diversity in HPS

Figure 10-2b demonstrates the expected decrease in the route diversity β with the increasing sector-constraint δ. For a given δ, a larger α means more nodes are involved in the wave-front routing (see Figure 10-1a), leading to higher route diversity β.

Figure 10-2c demonstrates the variation of *reception diversity* ρ, which is the average number of nodes that receive each transmitted pulse. It is computed by dividing the number of nodes that receive pulses by the total number of transmitted pulses. For a given α, with a higher δ, ρ increases slightly because with stricter sector-constraints the reduction of the total number

of pulse transmissions for an event is greater than the decrease of the total number of nodes that take part in receiving transmitted pulses.

The rate of increase in ρ, however, is very low when the sector-constraint δ is higher than approximately 0.33. It is because the number of nodes that receive transmitted pulses decreases at a rate that is similar to the decrease rate of the number of transmitted pulses. This can be seen from the example routing in Figure 10-1a and the variation of the number of pulse transmissions in Figure 10-2a. The same phenomenon also happens for smaller α, leading to larger *reception diversity* ρ.

10.2.2 Impacts of Spatial Compression

As shown in Figure 10-2a, with spatial compression enabled, for both δ observe how the number of pulse transmissions maximizes at an intermediate hop-distance, confirming the lateral fan-out and convergence seen as the routing envelopes in Figure 10-1b. With compression, the number of pulse transmissions in all hop areas are reduced to a very small value due to the back-off process as described in Section 7.3.2.1. With such spatial pulse compression, the minimum number of forwarding transmissions per event area is 1, and it stays around 1 for a sufficiently large spatial compression factor f_{comp}^{sp}. Because of this small spread in the forwarding transmission count, unlike the case without compression, the compression lines in Figure 10-2a do not show an obvious maximum.

Because of compression, the spatial compression algorithm demonstrates much lower and δ-insensitive route diversity. As shown in Figure 10-2b, the low diversity is due to the transmission reduction caused by the back-off based compression, and the δ-insensitivity is due to the fact that the number of forwarding nodes in this case is already very low.

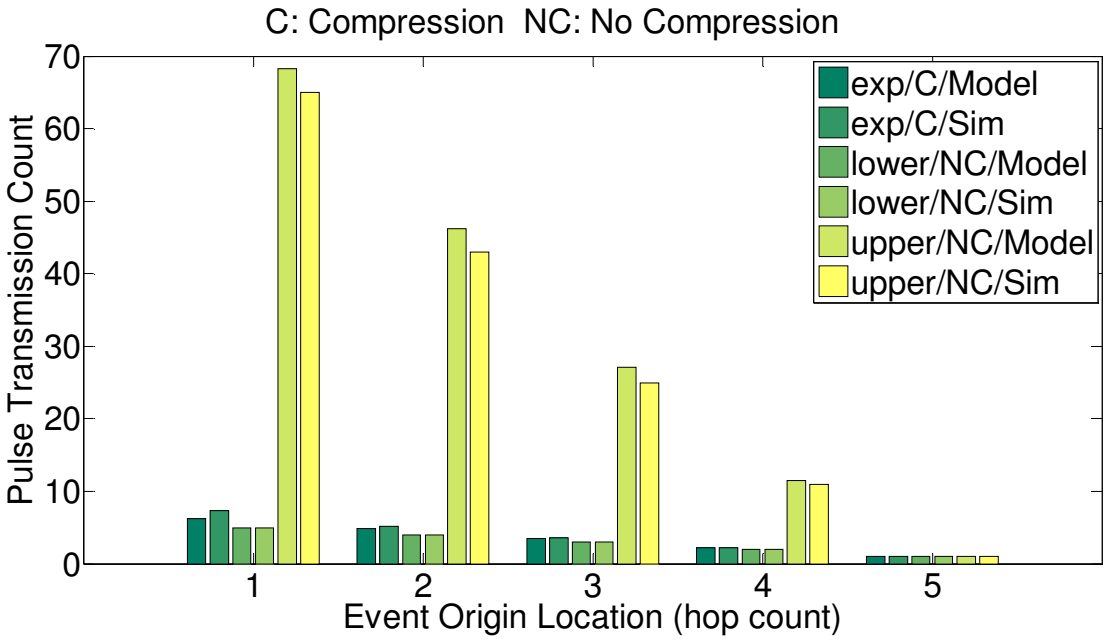


Figure 10-3: Pulse Bounds of pulse transmission complexity in HPS

Figure 10-3 depicts the bounds of forwarding transmission complexity computed from simulation and the analysis in Section 7.3.2.1. The figure shows number of forwarding transmissions for a single event generated in different hop areas. The slight difference between the model and the simulation results, especially for the upper-bound of the no-compression case, was due to the fact that the simulation was run on a practical square grid with local non-uniformity in node placements, whereas the model assumed fully uniform node placements. It should be noted that the compression case delivers the same performance as that of the lower-bound of the no-compression case, further validating the effectiveness of spatial pulse compression as presented in Section 7.3.2.1.Impacts of Event Scope on Pulse Routing

Figure 10-4a reports the impacts of event scope S on route diversity β. As S increases, β generally decreases. The route diversity factor β ($\beta \geq 1$) represents the total number of transmitted pulse pairs for routing a pulse from hop-distance h. With $S > 1$, β is equal to the

total number of transmitted pulse pairs divided by the product of the hop-distance *h* and event scope *S*. This is because with larger *S*, meaning more pulses generated from the same event area, pulse aggregation as explained in Section 4.3.5 becomes more prevalent. Higher aggregation prevents the transmitted pulse count from increasing as much as *S*. Thus, a larger *S* leads to a smaller β. The experiment was repeated with δ varying from 0.2 through 1 for sector width α of $30°$. Figure 10-3a also shows that with higher δ, the route diversity is lower also observed in Figure 10-2b.

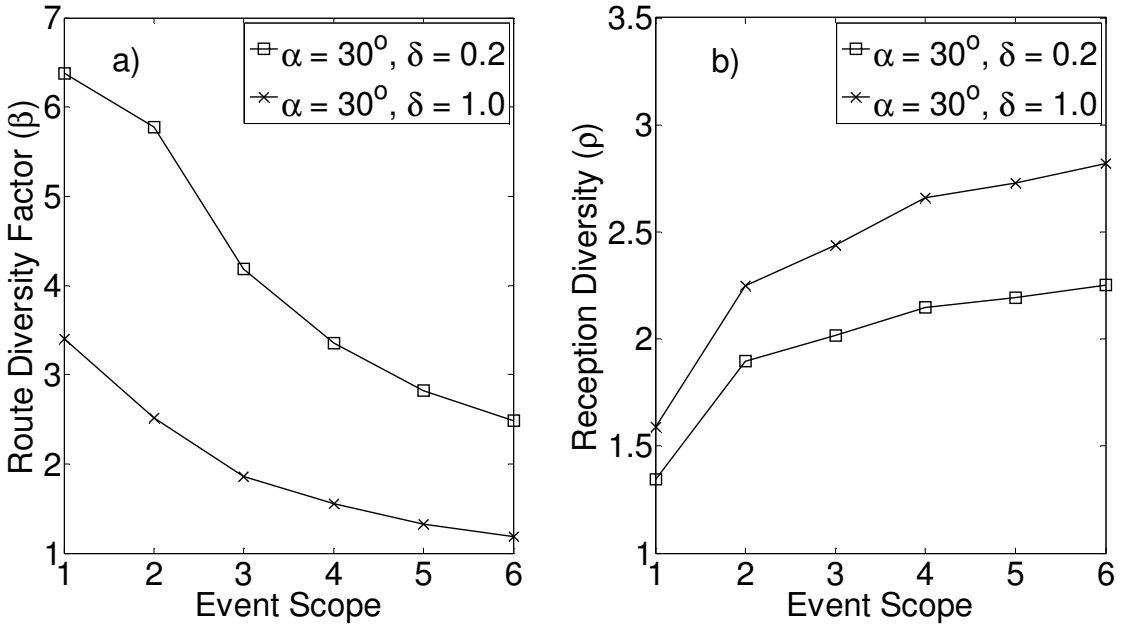


Figure 10-4: Effects of event scope on route and reception diversity in HPS

Since with increasing *S* (event scope) the pulse aggregation becomes more prevalent, the number of nodes that receive pulses increases at a higher rate than the number of transmitted pulses. This leads to increasing *reception diversity* ρ with increasing event scope *S*. The effect is reported in Figure 10-4b. The reason why the *reception diversity* ρ with higher δ is greater than that with lower δ is the same as that described for Figure 10-2c.

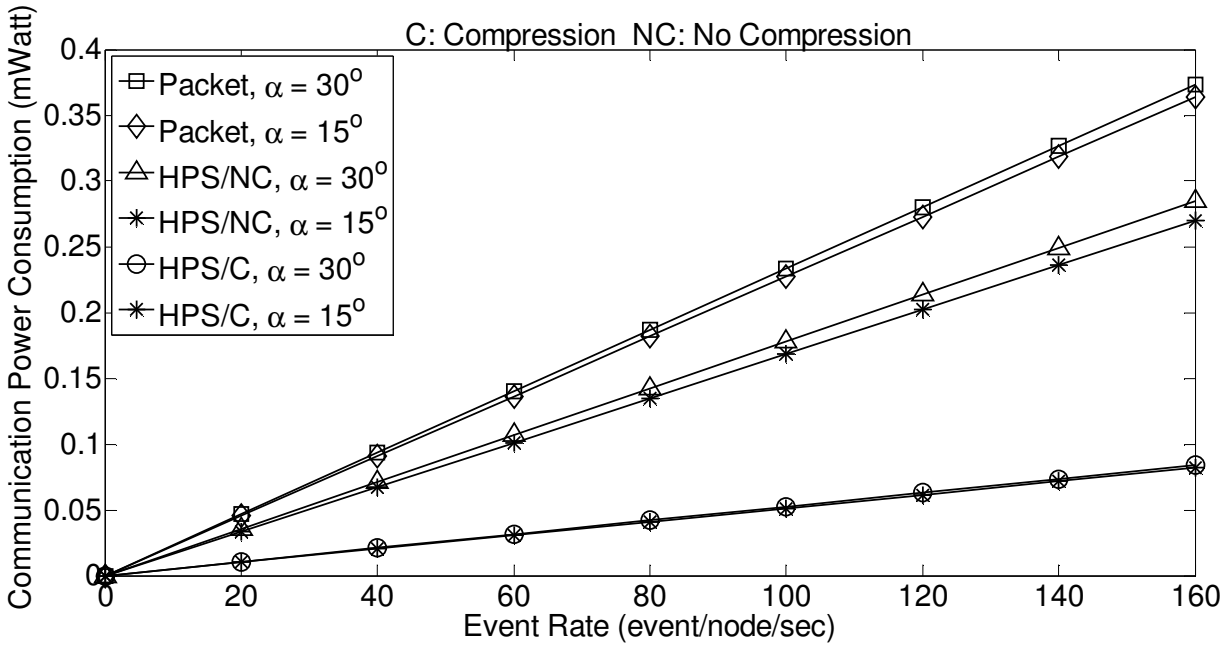


Figure 10-5: Communication power consumption per node in HPS

10.2.3 Communication Energy Performance

A TDMA with sink-rooted minimum spanning tree for packet routing has been used as the representative protocol (see Section 9.2) to compare its energy consumption with that of the proposed pulse switching. Figure 10-5 reports the communication power consumption for both pulse and packet transport with varying event rates. Observe that the consumption is linearly dependent on the event rate λ for both pulse and packet scenarios. As concluded in Section 9.3, the slope of the packet graph in Figure 10-5 is noticeably higher than that for HPS with and without compression, and it is mainly due to the overhead of per-packet preamble and payload overheads. Also observe that the slope of the HPS with compression case is lower than that of the HPS without compression, indicating the reduction of energy expenditure as a result of spatial pulse compression. This result further validates the low *route diversity* of the case with compression in comparison to that with no compression. Overall, Figure 10-5 validates the

primary premise of pulse switching that it can transport multi-point-to-point binary events at a lower energy budget compared to traditional packet switching.

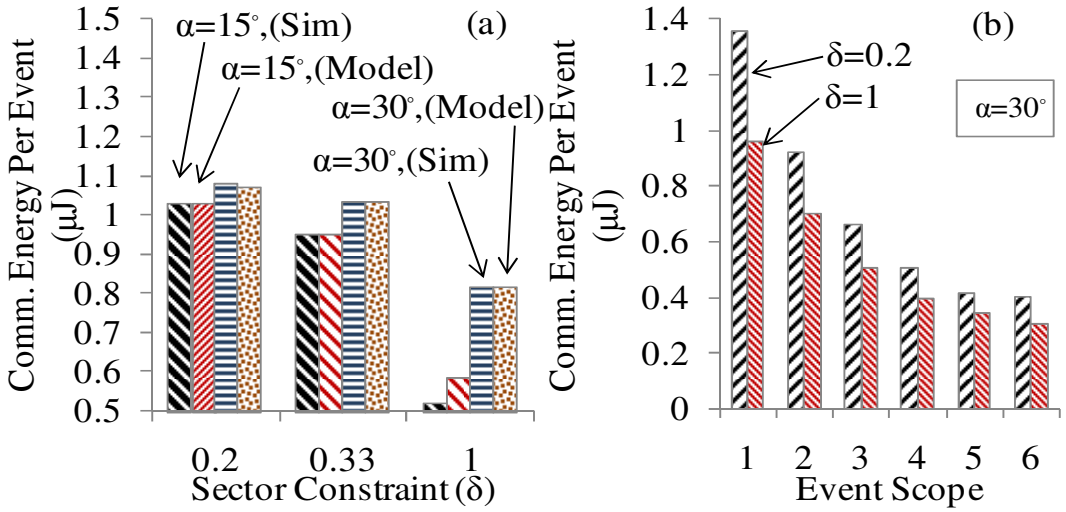


Figure 10-6: Communication energy performance with various δ, α and S in HPS

The change in per-event communication energy consumption with varying δ and α are reported in Figure 10-6a. As expected, with higher δ, since the pulse flooding is spatially constrained, the communication consumption reduces for all values of α. With larger α, the number of events generated in each event-area increases (see Figure 10-1). Since all pulses originated from the same event-area are identified by the same {*sector-id, hop-distance*}, all such pulses are aggregated and delivered as a single pulse to the sink. This aggregation, inherent to the architecture, scales down the communication energy for large α values.

The values of β and ρ obtained from the simulation (see Figures 10-2b and 10-2c) are plugged in the per event communication energy expression $2\hat{h}\beta(E_{tx} + \rho E_{rx})$ for computing the model results presented in Figure 10-6a. Observe that the model results match very well with the simulation results with the only exception when δ is very high for small α. This is due to pulse drops (under severe sector-constraints) which are not modeled by the energy expressions used in

Section 9.3.2. Pulse drops were modeled in Section 8.4 and are further analyzed in Section 10.4. Since the idling consumption does not depend on sector-constraint δ, it is not reported.

Note that as a pulse gets closer to the sink, less number of relay nodes are available since the event-area themselves are smaller for shorter hop-distances. Therefore, for low network densities, high *sector constraints* (i.e high δ) may lead to pulse drops when the constrained routing area does not have any qualified nodes to forward a pulse. This implies that for a given set of network parameters, there exists an upper-bound of the *sector constraints* δ that can be allowed for drop-less pulse communication even in a loss-less channel.

The communication expenditure per event in Figure 10-6b shows the same performance trend as for route diversity β, confirming that the energy in this case is influenced mainly by β and not by ρ.

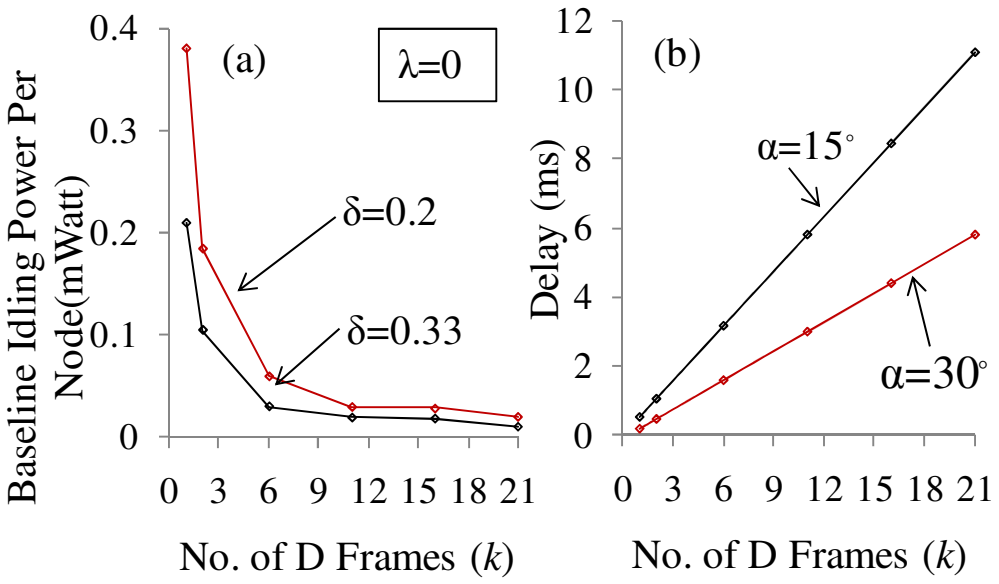


Figure 10-7: Effects of delay-traded sleep in HPS

10.2.4 Impacts of Delay-traded Sleep

As introduced in Section 7.4.1.3, delay-traded sleep frames (*D* frames) can be inserted between the regular *S, L, and T* frames for reducing the idling consumption at the expense of increased event transport delay. Figure 10-7a demonstrates the impacts on the idling consumption for different sector-widths as a result of varying *k* (used in Eqn. 9-5) which represents the number of *D* frames inserted after each *S, L,* and *T* frame. Eqn. 9-5 represents the baseline idling consumption (no event traffic). As expected from Eqn. 9-5, higher *k* causes the baseline consumption to reduce because the nodes can sleep longer during frequent *D* frames. Substituting the frame duration from Eqn. 9-1 into Eqn. 9-5, it can be seen that the baseline idling consumption depends on the sector-width α as a direct variation, which is why higher α in Figure 10-7a corresponds to lower baseline idling. Figure 10-7b shows the expected linear relationship between *k* and the event delay.

10.3 Moving Target Scenario

10.3.1 Simulation Settings

This subsection presents results for scenarios in which a moving target causes the sensor on its path to generate a series of pulses that are transported to the sink in order to estimate the target's trajectory. Results are obtained with the following parameterization. 1) *target speed* in meter per frame duration; 2) *sensor detection range* that represents the radius of detection; 3) *sensor detection rate* in sensing per frame duration; 4) *target trajectory* path. As shown in Figure 10-8, three diagonal (D1-D3), three horizontal (H1-H3), and three circular (C1-C3) trajectories that were used for the presented results. Once a target enters the network and follows one of the above trajectories, the affected nodes (i.e., within the sensing radius) generate pulses at the

specified detection rate. The pulses are then transported to the sink using HPS with or without compression.

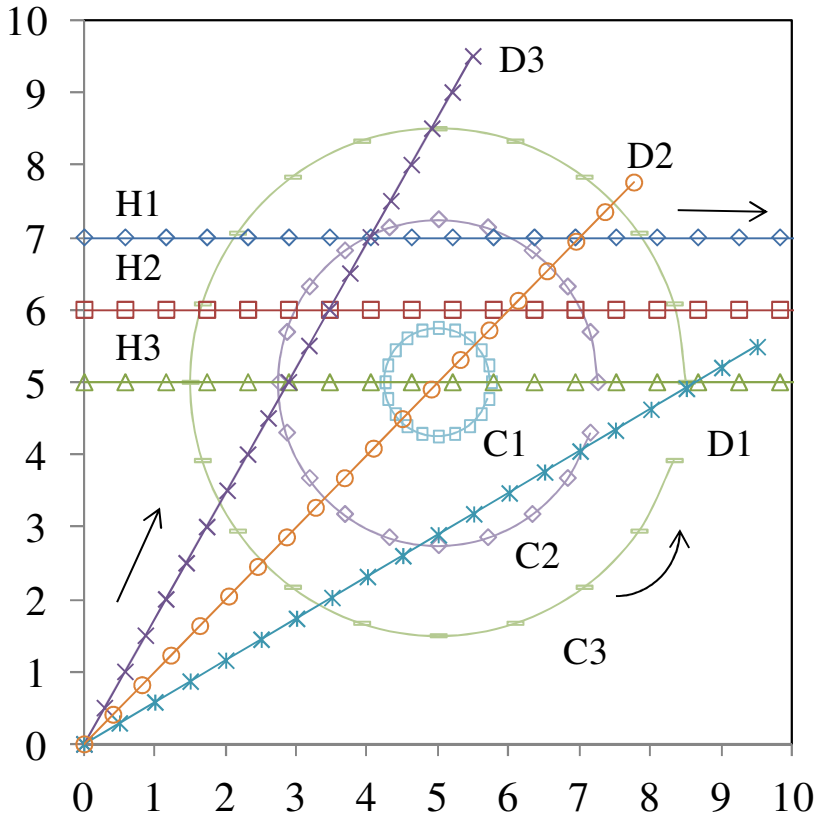


Figure 10-8: Experimental trajectories in HPS

10.3.2 Different Target Trajectories

Figure 10-8 shows the pulse transmission counts for various target trajectories (for both with and without compression) when the detection range and detection rates were chosen as 0.75 m and 5 times/frame respectively. The target speed was set to 0.0116 m/frame for the *diagonal* and *horizontal* trajectories, and 0.314 m/frame for the *circular* ones.

As expected, the compression algorithm significantly reduces the forwarded pulse count compared to the case without compression for HPS. This huge reduction in this mobile target

case is contributed not only by the spatial compression as observed in the static single event case, but also by the temporal compression as described in Section 7.3.2.2.

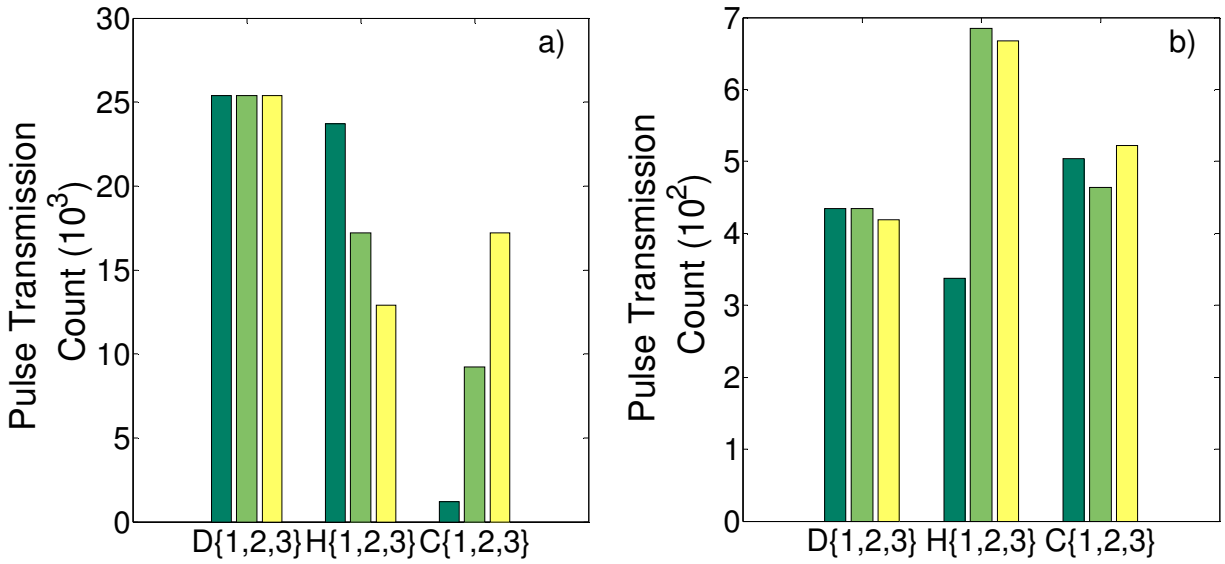


Figure 10-9: Pulse transmission count for HPS with and without compression

Since pulses generated in higher hop areas result in more forwarding transmissions, without compression the horizontal trajectory H1 generates more pulses than H2, and H2 generates more pulses than H3 (see Figure 10-9a). Similar observations can be also made for trajectories C1-C3, among which the trajectories with higher hop radii generates more pulse transmissions. For the diagonal trajectories D1-D3, however, there is no significant different in forward transmission count since all those trajectories have the exact same hop-distance properties. As shown in Figure 10-9b, HPS with compression can report the moving target's locations with approximately 38.59 times (on an average) fewer forwarding transmissions compared to that without compression. Because of the randomness of the back-off process used in spatial compression, the transmission count numbers for HPS with compression do not follow a clear monotonic trend as seen for HPS without compression in Figure 10-9a.

10.3.3 Impacts of Target Speed and Temporal Compression Factor

Figure 10-10a reports the impacts of target speed on the forwarding pulse count. For the shown results, which are for trajectory D3, observe that the pulse count reduces with the increasing target speed for both with and without compression cases in HPS. This is because for a given event detection range and event detection rate, the number of pulses generated by each source sensor (on and around the trajectory) is less for a target moving at higher speed. Also, as expected, irrespective of the speed, the pulse-count for HPS with compression is lower than that of HPS without compression, further validating the effectiveness of temporal compression for moving targets.

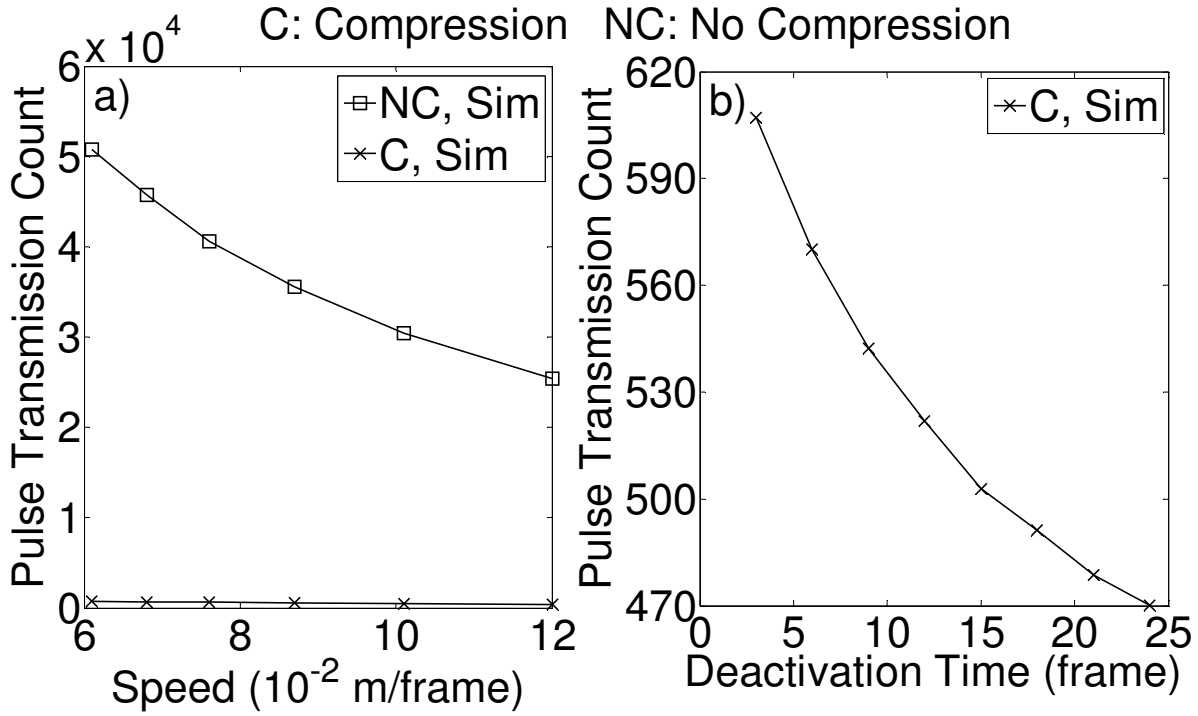


Figure 10-10: Impacts of target speed and deactivation time in HPS

The impacts of the temporal compression factor f_{comp}^{t}, a key parameter used for temporal pulse compression (see Section 7.3.2.2), are shown in Figure 10-10b. These results are for the

trajectory C3. Observe that as f_{comp}^{t} increases, the number of forwarding transmissions decreases, because of higher level of compression. The flip-side of higher f_{comp}^{t} is possible unreported events for high speed targets. In other words, for a given detection range and rate, a sensor may not report successive events when they are generated in quick succession due to high target speed. Therefore, there exists a trade-off between compression performance and the maximum supported target speed. The length of f_{comp}^{t} needs to be dimensioned based on the expected speed of the moving targets in a sensor field.

Compression in HPS is accomplished during pulse generation and forwarding, both spatially and temporally. To track the breakdown of the spatial and temporal compressions we measured the compression factor by turning the deactivation (i.e., spatial compression) on and off for the circular trajectory C3. With no temporal compression, the spatial compression factors in this case were observed to be 3.91 and 23.99 during generation and forwarding respectively, leading to a resulting compression factor of 32.04. With temporal compression activated, the factors were 4.11 and 25.1 during generation and forwarding, leading to an overall compression factor of 34.2.

10.4 Error Analysis

10.4.1 Impacts of False Positive Errors and Protection

The impacts of False Positive Pulse Rate (FPPR) on False Positive Event Generation Rate (FPEGR), both with and without protection, are demonstrated in Figure 10-11a. The figure shows results from both the simulation experiments and analytical expressions developed in Sections 8.2 and 8.3. During this experiment, Pulse Loss Rate (PLR) was set to zero, and the

FPEGR is reported at hop area 3 (i.e., $h = 3$), which is roughly in the middle of the experimental network that has the maximum hop-count of 5 (i.e., $H = 5$). Observe that the simulation results do exactly match with the FPEGR values that are numerically obtained from Eqns. 8-2, 8-6, 8-7, and 8-8, thus mutually validating the simulation and the experiments.

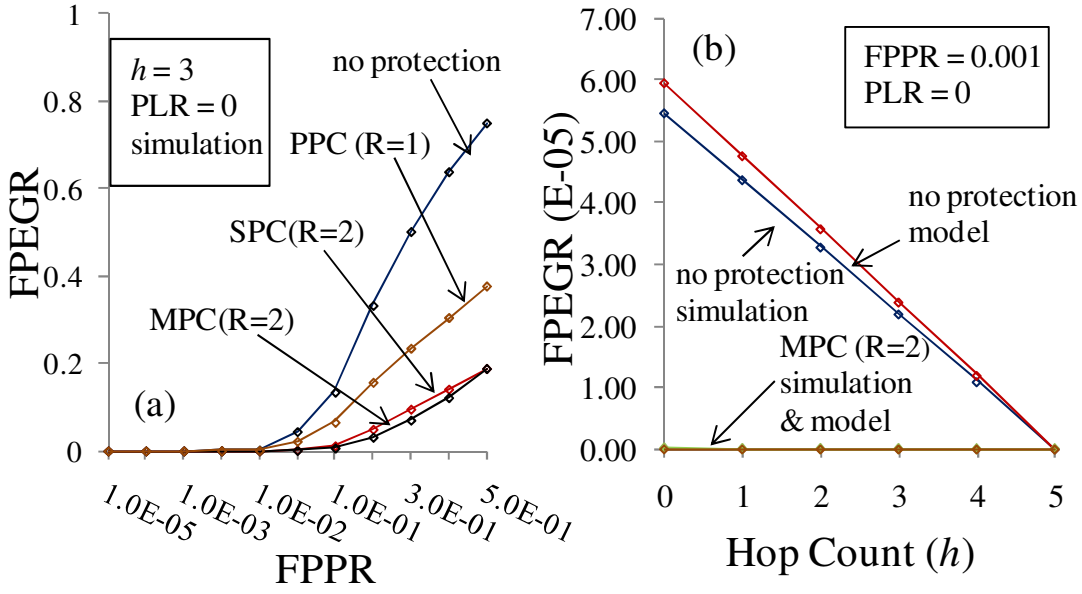


Figure 10-11: Impacts of FPPR and hop-count on FPEGR in HPS

It is shown in Figure 10-11a that although the FPEGR is quite low across the practical range of FPPR [44], it can be unacceptable when the false positive detection probability exceeds the threshold of 10^{-2}, which is unusually large for the practical UWB-IR detectors [45]. However, in unlikely situations with such large FPPR, the proposed protection mechanisms are able to significantly reduce the FPEGR. Observe that with both SPC and MPC mechanisms (see Section 8.3.1), the FPEGR can be kept within an acceptable range for FPPR as high as approximately 0.1. As expected, both SPC and MPC outperform the PPC scheme that relies on only one protection pulse at the end of each frame. Between SPC and MPC, the latter provides lower FPEGR across a wider range of FPPR values. This is because the MPC algorithm assigns

at least one pulse in the protection area for every single event, whereas the SPC algorithm assigns exactly one pulse. This leads to higher $n_{MPC}(k_h)$ compared to $n_{SPC}(k_h)$, as used in Eqns. 8-8 and 8-6, causing comparatively lower FPEGR for MPC. Considering its superior performance compared to SPC and PPC, all protection results from this point onwards will be presented for the MPC algorithm.

Figure 10-11b reports the impacts of FPPR on FPEGR with varying node location (i.e., hop area *h*) in the network for an FPPR of 0.001. Results in this figure are from simulation without pulse lost errors (PLR is set to zero) and the analytical expressions modeled by Eqns. 8-2 and 8-8 in Section 8.3.1. Without FPC protection, observe that the severity of the impacts of false positive detection increases near the sink. This is because as the hop area *h* reduces the vulnerable area for false positive detection within a frame increases, leading to more frequent false positive event generation. However, with MPC based protection turned on (with $R = 2$), the effects of hop area *h* becomes negligible due to the vanishingly small FPEGR values.

10.4.2 Impacts of Pulse Loss

Effects of pulse loss due to multi-path, noise, and interference (see Section 3.2.5) on event loss at sink are reported in Figure 10-12a. These results are with uncorrelated and uniformly distributed pulse losses, and without false positive detection errors (i.e., $FPPR = 0$). Figure 10-12a reports results with and without FPC protection activated from the analytical expressions in Eqns. 8-11 and 8-13.

Observe the two lines in Figure 10-12a for no protection, which demonstrate how the Event Loss Rate (ELR) increases with increasing Pulse Loss Rate (*PLR*). This relationship was modeled in Eqn. 8-11. These two lines also depict that for a given sector size (i.e., α), stricter

sector constraints (i.e., larger δ) result in higher rates of event losses at the sink. As shown in Figure 10-2b, as the sector-constraint δ increases, the route diversity β decreases, causing higher event losses triggered by pulse losses.

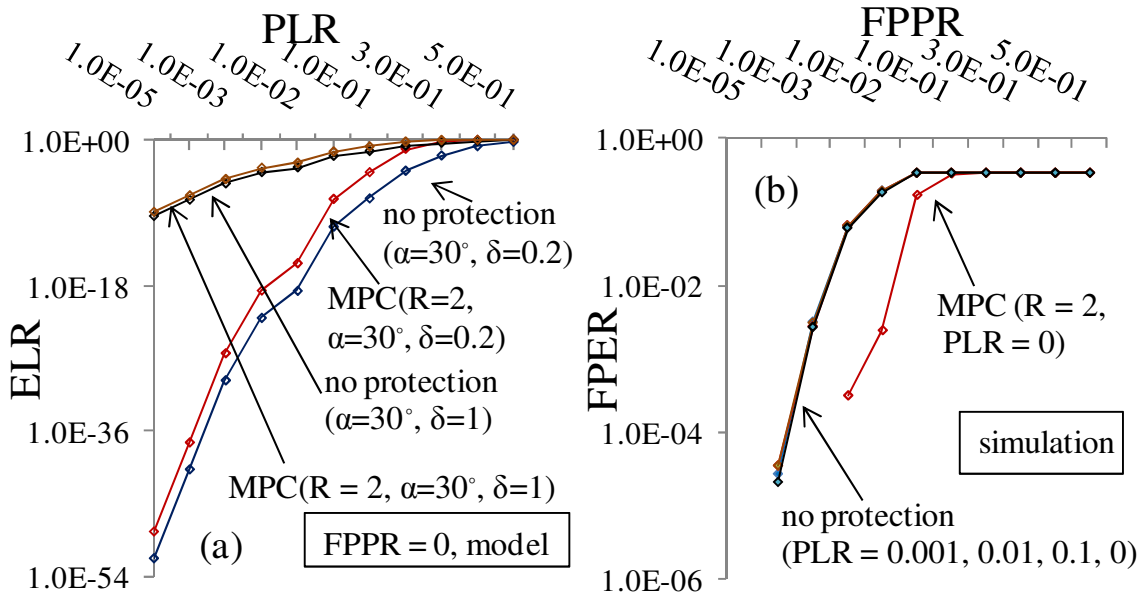


Figure 10-12: Performance impacts of PLR and FPPR in HPS

Figure 10-12a also reports the negative impacts of false positive protection mechanism on event losses. Observe that for both the without-protection cases (with different δ), the application of MPC based protection pushes the ELR values up, although slightly. As modeled in Eqn. 8-13, this effect can be ascribed to the fact that the additional pulses added for the MPC code increases the chance of losing an event due to pulse losses. Note, however, this slight increase in ELR is hugely outweighed by the massive reduction of False Positive Event Generation Rate (FPEGR) caused by the MPC protection as shown in Figure 10-12a.

10.4.3 False Positive Event Rate at Sink

False Positive Event Generation Rate (FPEGR) in the absence of pulse losses (i.e.,

$PLR = 0$) was investigated in Section 10.4.1. In this section, we explore the False Positive Event Rate (FPER) at sink when both false positive detection and pulse loss errors are simultaneously turned on. FPER is defined as the probability of at least one false positive event reported to sink in a given frame. Unlike FPEGR, which captures the local effect of valid false positive event generation, FPER captures the end-to-end effects after such false positive events are routed to the sink.

Figure 10-12b corresponds to results from simulation experiments, and depicts that in the absence of protection, FPER initially increases with FPPR at a high rate and then tapers off after FPPR reaches a threshold value of approximately 0.05. The graph also shows that this pattern is almost insensitive to the pulse loss rate when it is in the range from 0 to 0.1. This indicates that the false positive event errors are reasonably immune to the pulse losses, mainly due to the inherent route diversity of the proposed pulse routing framework. Now with the MPC based false positive protection turned on, the FPER can be significantly reduced when FPPR is less than 0.1, which is almost certain in practical UWB-IR systems. As can be seen in the figure, the MPC based protection algorithm can achieve almost zero FPER when the FPPR is in the vicinity of 0.005, which is a practical range. This further validates the efficacy of the false positive protection mechanism in the presence of both pulse errors and false positive detections at the same time.

10.4.4 Impacts of Compression

Figure 10-13a depicts simulation results of PLR versus ELR for a single event generated in hop area 5 of the 441-node network. Observe that for practical range of PLR [44], the ELR for the no-compression case remains vanishingly small and it is generally insensitive to the value of

PLR. This is mainly because of the huge redundancy in pulse transmissions (i.e., route diversity) for the no-compression case as demonstrated in Figure 10-2. The physical implication of this result is that at the cost of transmission redundancy the no-compression case offers very high immunity from pulse losses.

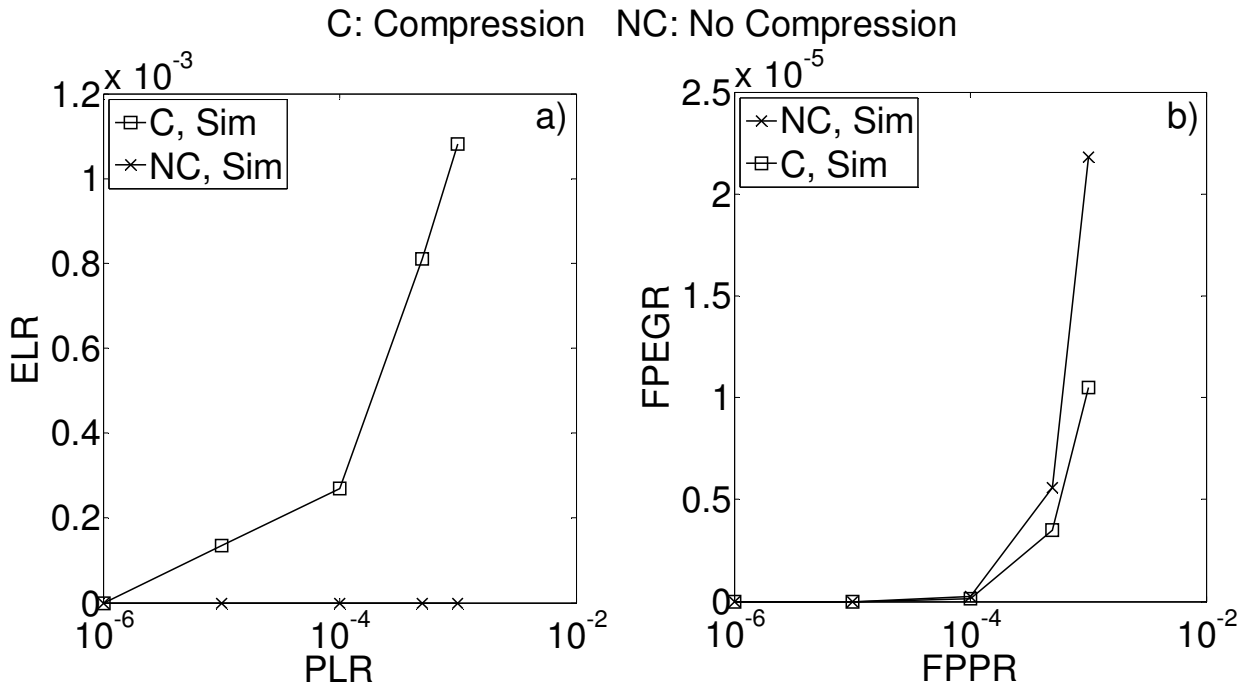


Figure 10-13: Impacts of pulse loss errors and false positive errors in HPS

With spatial compression enabled, the ELR is higher and it shows clear sensitivity to PLR. The higher ELR is caused by almost zero transmission redundancy (i.e., near-one β in Figure 10-13b) which leads to high probability of event losses when pulses are lost. It should be noted that although the compression case is less immune to pulse losses compared to the no-compression case, the resulting ELR is still low for practical PLR values [44].

If pulses are erroneously detected [38] by a node in state *LO* or *TL* such that a false positive pulse in the *Control sub-frame* corresponds to another false positive pulse in the *Event sub-frame* with corresponding forwarding flag (see Figure 4-3), then an event is falsely detected

at that node. Once such a false positive event is generated, it is forwarded all the way to the sink, leading to a false positive event reporting.

The impacts of FPPR on FPEGR in hop area 3 both with and without compression are shown in Figure 10-13b. Hop area 3 is chosen because it represents the middle of the experimental network. Observe that in both cases of compression and no compression, FPEGR is extremely small with practical range of FPPR [44] which is lower than 10^{-4}. This indicates that both with and without compression, the proposed pulse switching protocol is fairly immune to false positive errors. Also observe that for larger *FPPR* (i.e., larger than 10^{-4}) when the compression is turned on, the case with compression shows less sensitivity to false positive errors compared to that without compression.

10.5 Summary and Conclusion

This chapter evaluated the energy-efficiency of the hop-angular pulse switching protocol (HPS) in the static event scenario and the moving target scenario respectively. In both scenarios, the impacts of the significant system parameters were investigated and discussed, such as the sector constraint, the event scope, the spatial and temporal compression factors and the target speed and so on. In addition, this chapter also provided the results about the impacts of false positive errors and pulse loss errors. More importantly, it proved the effectiveness of the proposed protection mechanisms and the immunity of pulse switching from pulse losses. Through analytical modeling and simulation based experiments, it was shown that the proposed hop-angular pulse switching architecture can be an effective means for energy efficiently transporting information that is binary in nature.

Chapter 11: Performance of Cellular Pulse Switching

11.1 Introduction

We developed the event-driven C++ simulator for implementing the MAC framing and pulse routing of the cellular pulse switching (CPS) as proposed in Chapter 5. This chapter evaluates the performance of CPS through the results derived from the simulator. A network with terrain size of 10×10 meter^2 includes evenly distributed sensor nodes and a sink node placed at the lower left corner of the terrain. Sensors are grouped into 115 regular hexagonal cells, each containing 5 nodes in average.

11.2 Static Event Scenario

This subsection presents results for scenarios in which a static single event causes the sensor within its sensing range to generate a single event pulse, which is transported to the sink using the proposed routing and localization protocols. The protection described in Section 8.3 is disabled.

11.2.1 Pulse Transmission Count along the Route

Figure 11-1a reports the number of pulse transmissions (i.e., with and without compression) in sensor cells that are at specific cell-counts away from the sink. An event is generated at a cell of 15 cell-count away from the sink, which is then routed (with route diversity 1) to the sink using the presented pulse switching protocol. The retransmission times θ_k is set to 1. The resulting number (from simulation) of transmitted pulses within the sensor cells along the route is plotted in Figure 11-1a. The x-axis represents the distance of the cells along the route

expressed as cell-count from the sink. Cell-count 1 represents the sensor-cell nearest to the sink and the highest cell-count represents the source cell.

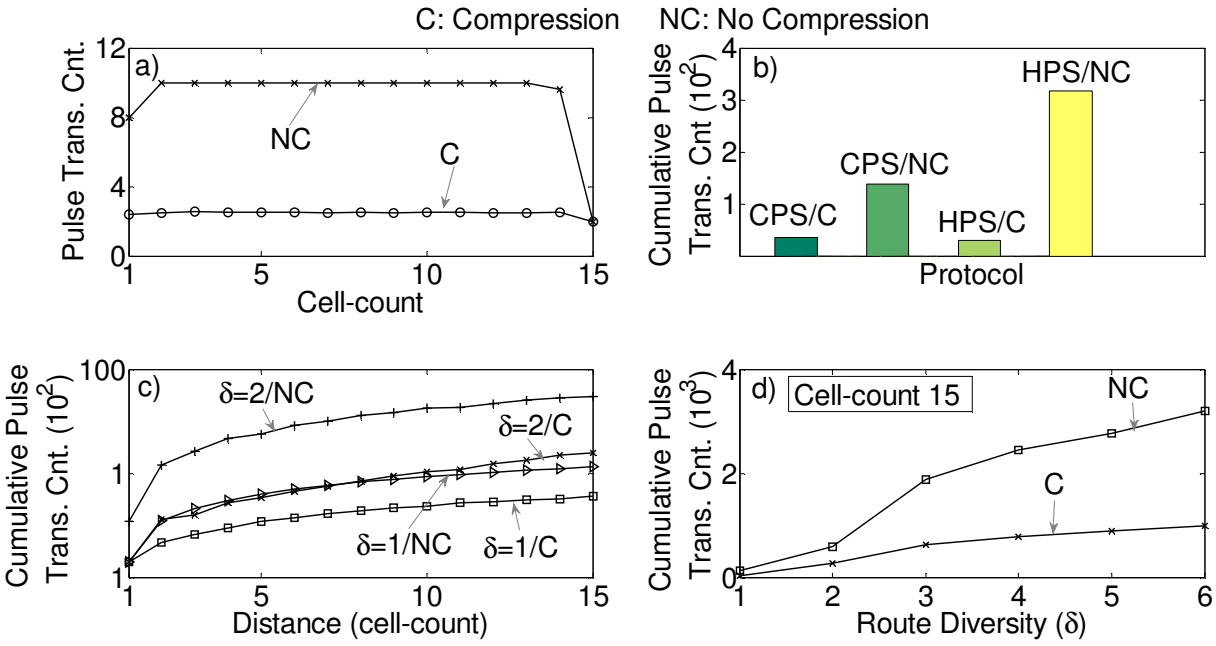


Figure 11-1: Pulse transmission count in CPS

The number of pulse transmissions in a sensor-cell is equal to the forwarding node count in the corresponding sensor-cell multiplied by the number of pulses in a frame. As shown in Figure 11-1a, the line representing the compression case is flat across different cell-counts except for the cell-counts 1 and 15. It is due to the fact that the forwarding node counts are same across different cell-counts except for the cell-counts 1 and 15. With compression applied, the expected pulse transmission count is reduced to a very small value due to the back-off process as modeled in Section 7.3.2.1. With such pulse compression, the minimum number of pulse transmissions per cell-count is 2, and it stays around 2 for a sufficiently large spatial compression factor f_{comp}^{sp}.

11.2.2 Cumulative Pulse Transmission Count

For comparison purpose, we also present pulse transmission count for the *Hop-angular Pulse Switching* protocol (HPS) proposed in Chapter 4 applied to the same network. For HPS, the resolution and the sector-constraint are set to 30° and 1 respectively. Figure 11-1b reports the cumulative pulse transmission count along the entire route in both CPS and HPS scenarios. The simulation results include both compression and no compression cases. The cumulative pulse transmission count of the single event originated from the source node towards the sink in HPS is 2.23 times greater than that in CPS without compression. With compression applied, the cumulative number of pulse transmissions in HPS is only 1.07 times greater than that in CPS, due to the effectiveness of pulse compression on reducing pulse transmission counts.

These results indicate that CPS is able to achieve better energy efficiency than HPS by being able to transport events to the sink with lower number of pulse transmissions. Since the size of a sensor cell in CPS is generally smaller than that of a hop area in HPS, CPS offers better localization resolution compared to HPS.

11.2.3 Different Source Cells

Figure 11-1c shows the cumulative pulse transmission count as a function of the distance between the sink node and the sensor cell of event generation. Such distance, represented as cell-count, is varied from 1 through 15. Pulse transmission count results are collected for two different route diversities (i.e., δ set to 1 and 2) with and without compression. As expected, Figure 11-1c demonstrates the cumulative pulse transmission count increases with the increasing distance (i.e., cell-count) between the source sensor-cell and the sink node. Obviously, the cumulative pulse transmission count with $\delta = 2$ is significantly higher than that with $\delta = 1$ for a

given distance for no compression scenario. With compression, the cumulative pulse transmission count with $\delta = 2$ is only slightly higher than that with $\delta = 1$ for a given distance. Pulse compression described in Section 7.3.2.1 significantly reduces the cumulative pulse transmission count of an event originated in any distance with $\delta = 2$.

11.2.4 Impacts of Route Diversity

Figure 11-1d reports the cumulative pulse transmission count with different route diversities applied to the switching/routing process for a pulse generated at a distance corresponding to cell-count 15. Route diversity δ was varied from 1 to 6. Results contain both cases of with and without compression from the simulation. The figure shows that with increasing route diversity, the pulse transmission overhead increases. Eventually, the CPS process approaches to effective pulse flooding for very high δ values. From Figure 11-1d, it can be observed that the increment in pulse transmission count is generally most prominent when δ is changed from 2 to 3 for the cell-count 15. Besides, Figure 11-1c shows that the cumulative pulse transmission count is higher with a greater δ for any given cell-count. So the observations from Figures 11-1c and 11-1d suggest that the route diversity should be kept below 2 for keeping the energy cost under control.

11.3 Moving Target Scenarios

This subsection presents results for scenarios in which a moving target causes the sensor on its path to generate a series of pulses that are transported to the sink in order to estimate the target's trajectory. The protection in Section 8.3.2 is disabled. And the spatial compression factor was set to 10 for any trajectory in the case of compression. Results are obtained from simulation with the following parameterization: a) target speed in meter per frame duration; b) sensor

detection range that represents the radius of detection; c) sensor detection rate in sensing per frame duration; d) target trajectory path. We choose the route diversity δ to be 1. As shown in Figure 11-2, three diagonal (D1-D3), three horizontal (H1-H3), three vertical (V1-V3), three sector (S1-S3) trajectories and one square (SQ) trajectory were used for the presented results. Once a target enters the network and follows one of the above trajectories, the affected nodes (i.e., within the sensing radius) generate pulses at a specified detection rate. The pulses are then transported to the sink.

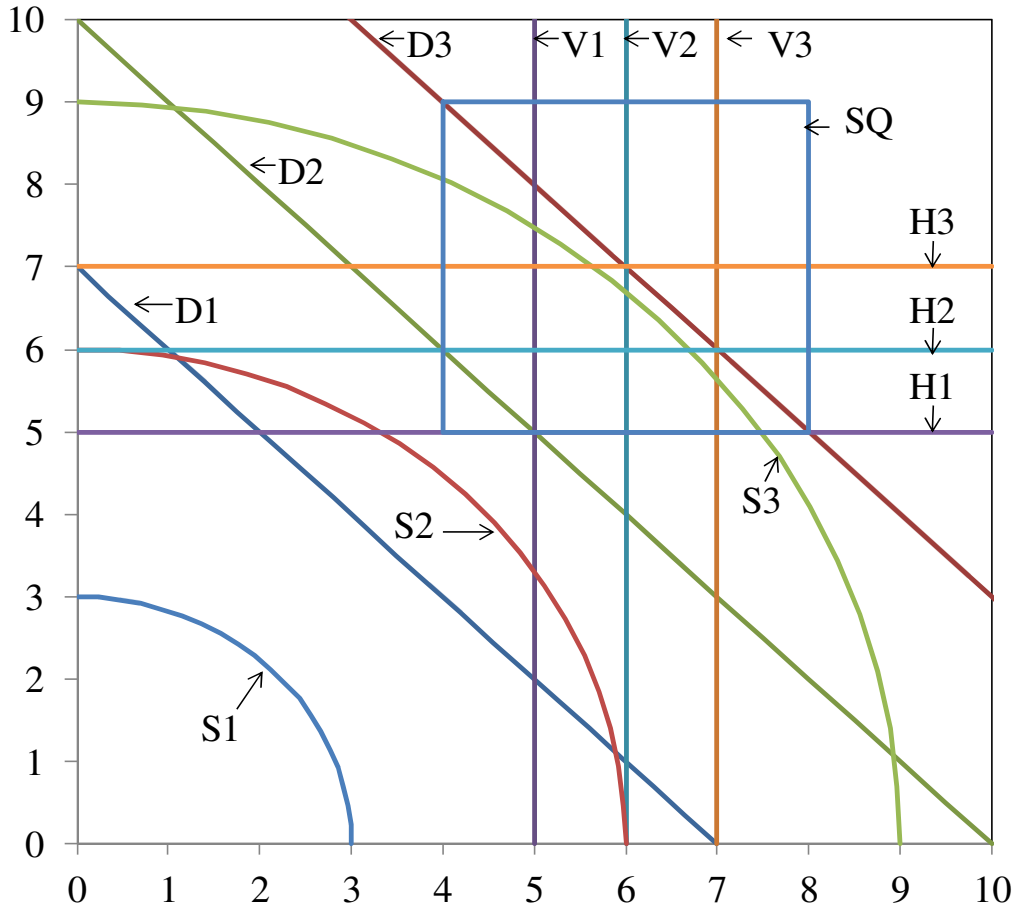


Figure 11-2: Target trajectories in CPS

11.3.1 Different Target Trajectories

Figure 11-3a shows cumulative pulse transmission counts for various target trajectories with and without compression. The detection range and detection rate were chosen as 0.5 *m* and 0.2 *sensing/frame* respectively. The target speed was set to 0.1 *m/frame* for the diagonal, horizontal and vertical trajectories, and 0.016 *rad/frame* for the sector ones. As expected, pulse compression described in Section 7.3.2 significantly reduces the forwarded pulse count compared to that without compression for all trajectories. This huge reduction in this mobile target case is contributed not only by the spatial compression as observed in the static single event case (see Section 11.2), but also by the temporal pulse compression.

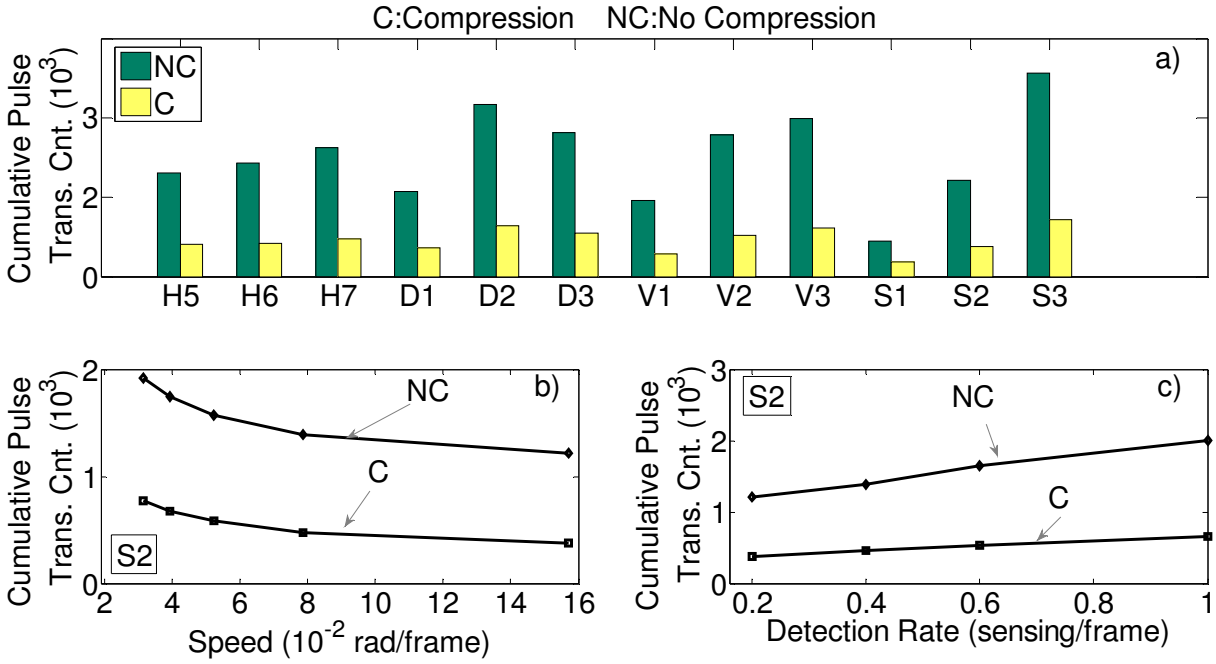


Figure 11-3: Pulse transmission count for various trajectories, speeds and detection rates in CPS

Cumulative pulse transmission count for moving target scenario depends on the number of nodes that participate in transmitting pulses for a given target speed, sensor detection range, and detection rate. Specifically, it increases with increasing number of sensor-cells that a moving

target goes through, and increasing average distance of the moving target from sink. Therefore, as shown in Figure 11-3a, H3 exceeds H2, whereas H2 exceeds H1, in terms of cumulative pulse transmission count with and without compression. Similar observations can be made for V1-V3. As to diagonal trajectories, D2 has more pulse transmissions than D3, whereas D3 exceeds D1. Furthermore, a sector trajectory with a greater radius generates more pulse transmissions among trajectories S1-S3. As shown in Figure 11-3a, the protocol with compression can report the moving target's locations with approximately 3.25 times fewer forwarding transmissions compared to that without compression on average.

11.3.2 Impacts of Target Speed and Detection Rate

Figure 11-3b reports the impacts of target speed on the cumulative pulse transmission count for trajectory S2. Observe that the cumulative pulse transmission count reduces with increasing target speed for both with and without compression. This is because for a given event detection range (0.5 *m*), event detection rate (0.2 *sensing/frame*), spatial compression factor (10) and temporal compression factor (100 *frames*), the number of pulses generated by each source sensor (on and around the trajectory) is less for a target moving at a higher speed. Also, as expected, irrespective of the speed, the pulse count with compression is lower than that without compression, further validating the effectiveness of compression.

Figure 11-3c shows how the detection rate impacts cumulative pulse transmission count with and without compression. Observe that the pulse transmission count increases with increasing detection rate in both with and without compression cases. For the given target speed (0.016 *rad/frame*), detection range (0.5 *m*), spatial compression factor (10) and temporal compression factor (100 *frames*), higher detection rate leads to more transmission redundancy

and also more energy consumption, but acts stronger against the pulse loss errors. Contrarily, a relatively low detection rate might prevent a sensor from detecting the target in time.

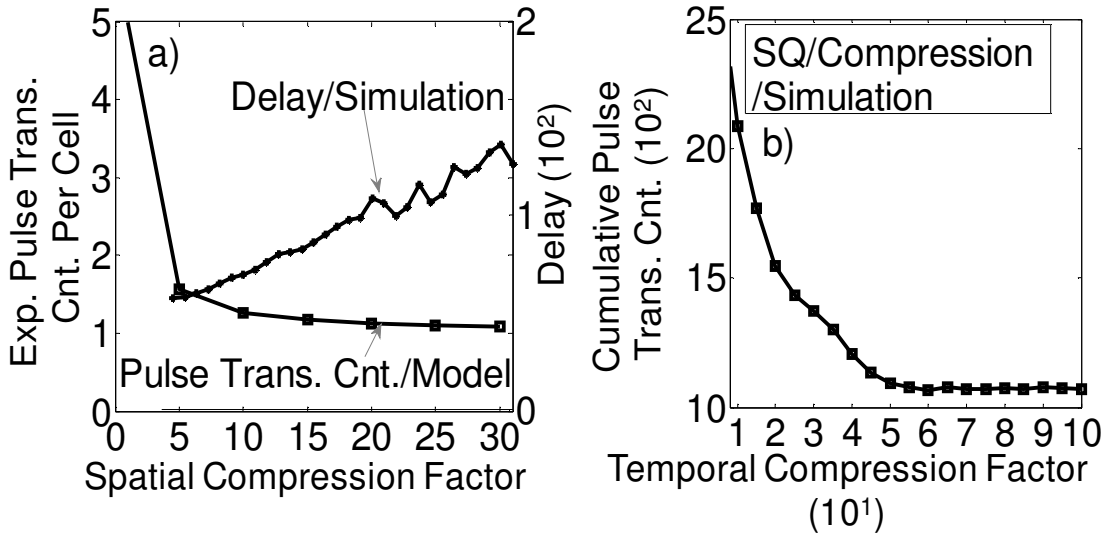


Figure 11-4: Impacts of spatial and temporal compression factors in CPS

11.4 Compression Analysis

11.4.1 Impacts of Spatial Compression

For a single event, the performance of pulse compression is mainly impacted by the spatial compression factor f_{comp}^{sp} in terms of expected pulse transmission count per cell and delay. The delay is defined to be the time duration between the generation time of a single event and the arrival time at the sink of the first pulse representing this event. We set the source cell at a distance corresponding to cell-count 15. Suppose that all nodes in the intermediate sensor-cells participate in forwarding pulses and the retransmission times θ_k is 1. According to Eqn. 7-2, the expected pulse transmission count per cell (containing 5 nodes on average) decreases with the increasing f_{comp}^{sp}, as shown in Figure 11-4a. Additionally, the decreasement approaches

saturation, when f_{comp}^{sp} goes beyond 10. On the contrary, the delay fluctuatingly increases with the increasing f_{comp}^{sp}, as shown in Figure 11-4a. Considering the tradeoff between pulse transmission count and delay, the reasonable range of f_{comp}^{sp} is [5,10].

11.4.2 Impacts of Temporal Compression

As shown in Section 11.3, the performance of pulse compression is impacted not only by the spatial compression factor f_{comp}^{sp} but also by the temporal compression factor f_{comp}^{t} in the moving target scenario. The impacts of f_{comp}^{t} are shown in Figure 11-4b. These results are for the trajectory SQ (the length *d* is 4 m). The detection range (*r*) and detection rate (f_d) were chosen as 0.5 m and 0.2 times/frame respectively. The target speed (*v*) was set to 0.1 m/frame. As analyzed in Section 7.3.2.2, the lower and upper bounds of f_{comp}^{t} can be calculated by incorporating values of *r*, *v*, f_d and *d* into Eqn. 7-5, resulting in [10,160] frame. In Figure 11-4b, observe that within the time period [10,50] frame, as f_{comp}^{t} increases, the number of forwarding transmissions decreases, because of higher level of compression. When f_{comp}^{t} goes beyond 50 frame, cumulative pulse transmission count stablizes at around 1000. Therefore, for a given detection range, detection rate, target speed and target trajectory, the temporal compression performance is gradually improved with increasing f_{comp}^{t} until saturation.

11.5 Pulse Error Analysis

This subsection presents results for effects of error occurrences and the protection mechanism proposed in Section 8.3.

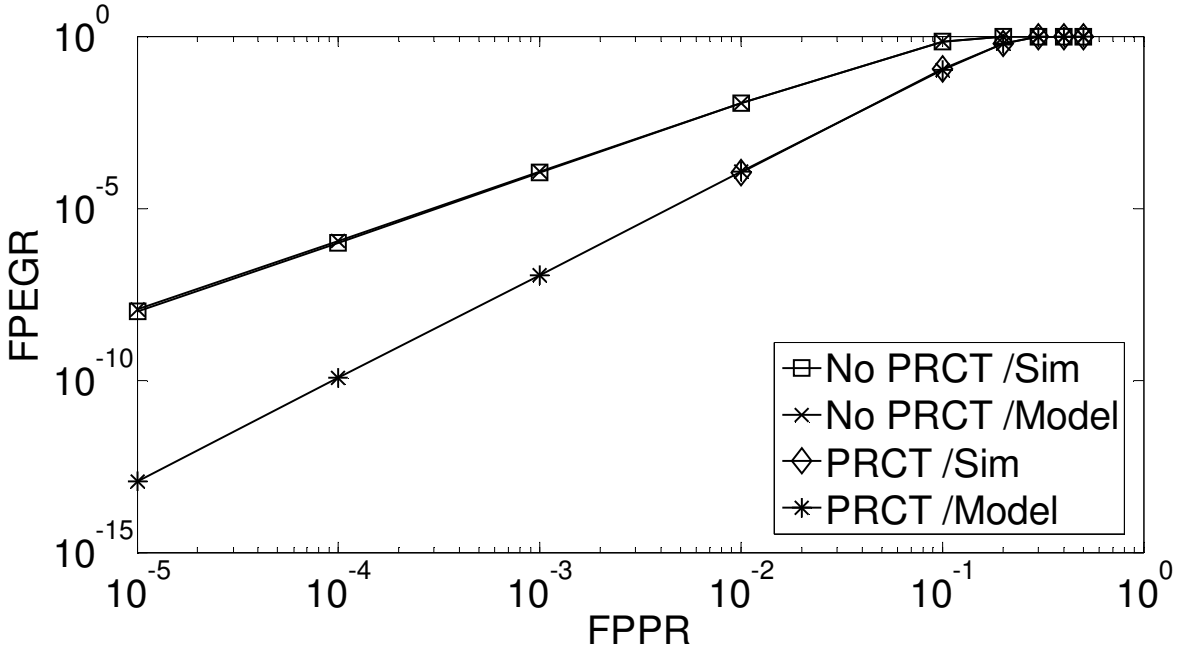


Figure 11-5: Impacts of FPPR on FPEGR with and without protection in CPS

11.5.1 Impacts of False Positive Errors and Protection

The impacts of FPPR (False Positive Pulse Rate) in any sensor-cell, both with and without protection, are shown in Figure 11-5. Simulation results do exactly match the FPEGR (False Positive Event Generation Rate) values that are numerically obtained from Eqn. 8-3. In Figure 11-5, with and without protection, FPEGR is extremely small within the practical range of FPPR which is lower than 10^{-4} [44]. This indicates that both with and without protection, the proposed pulse switching protocol is fairly immune to false positive errors. Also observe that for larger

FPPR (i.e., larger than 10^{-4}), when the protection is turned on, it shows less sensitivity to false positive errors compared to that without protection.

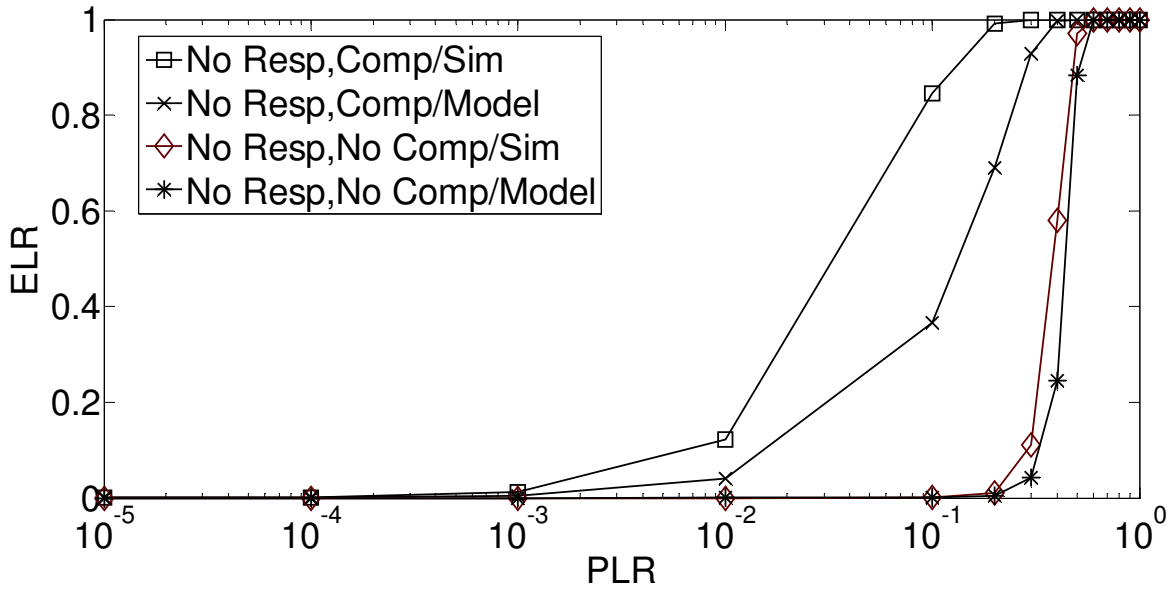


Figure 11-6: Impacts of PLR on ELR without response in CPS

11.5.2 Impacts of Pulse Loss Errors

Figures 11-6 and 11-7 depict impacts of PLR (Pulse Loss Rate) on ELR (Event Loss Rate) for a single event generated at a distance of cell-count 15. The route diversity δ is set to 1. Observe that for practical range of PLR (less than 10^{-4}) [44], the ELR for all cases remains vanishingly small and it is generally insensitive to the value of PLR. Note that when the route diversity δ is larger than 1, the system reliability against pulse loss errors would be stronger than that with $\delta = 1$, due to more pulse transmission redundancies. In the case of no response as shown in Figure 11-6, observe that the results derived from the model (i.e., Eqn. 8-12 and retransmission times θ_k to be 1) and simulations match quite well. The minor differences were found to be stemming from limited simulation time. In the case with response enabled, we set the

maximum value of retransmission times θ_k to be 100 in the model calculation. In this way, Eqn. 8-12 models the lower bound of the ELRs, which are smaller than the ELRs derived from simulations. It is exactly how Figure 11-7 demonstrates.

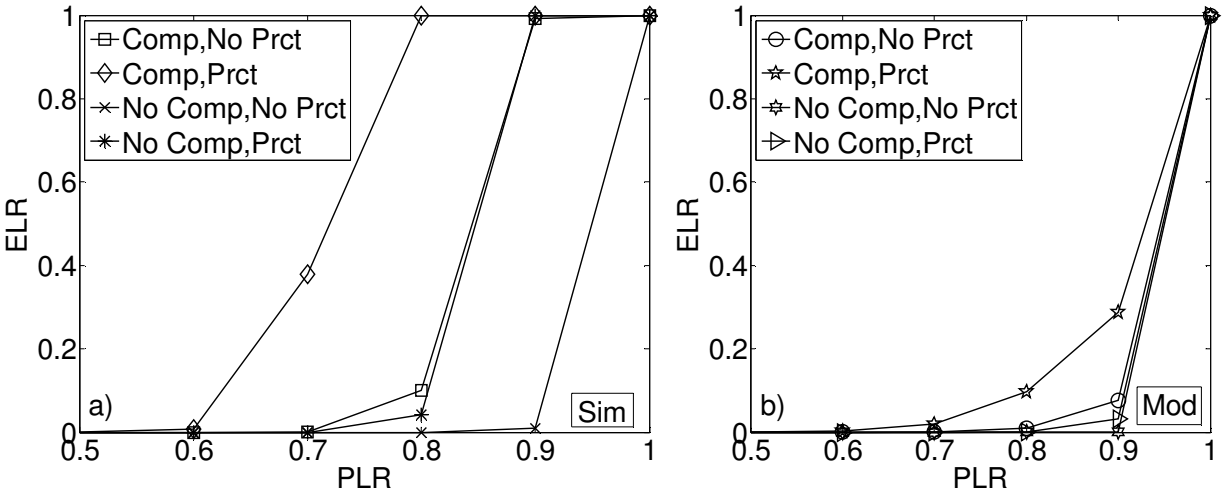


Figure 11-7: Impacts of PLR on ELR with response in CPS

These two figures also show the impacts of the response mechanism, pulse compression method, and protection algorithm on the relationship between PLR and ELR. We define the critical point of PLR to be the maximum value of PLR corresponding to zero ELR.

Figure 11-6, Figure 11-7a (simulation) and Figure 11-7b (model) show that the critical points of PLR are larger than 0.5 when the response mechanism is activated. θ_k introduced by *Response Mechanism* contributes a lot to decrease the ELR as low as zero for high PLRs. With response mechanism disabled, as shown in Figure 11-6, the critical points are respectively 10^{-3} and 10^{-1} which are lower than 0.5 but still higher than the practical range of PLR (less than 10^{-4}) [44]. It shows that the proposed response mechanism in Section 5.4 strengthens the system reliability against pulse loss errors, especially with high PLR.

No matter *Response Mechanism* or *Protection* is enabled or not, the ELR without compression is lower than that with compression for a given PLR. The higher ELR with compression is caused by low transmission redundancy which leads to high probability of event loss. Note that although the compression mechanism is less immune to pulse losses compared to the without compression case, the resulting ELR is still low corresponding to the practical PLR range (less than 10^{-4}) [44].

In Figure 11-7b, compared to the no-protection cases, the ELR with SPC protection is similar to that without SPC for model results obtained from Eqn. 8-12. According to Section 8.4.2, the value of e with SPC, the probability of losing an event once on any transmission hop on the route of the event, is higher comparing to that without protection. However, the exponential number θ_k hugely reduces the probability ELR. With higher θ_k, the offset effect of *Response Mechanism* on a higher e becomes stronger. From the observations above, *Response Mechanism* is able to offset the increase of the ELR with protection for the given PLR.

11.6 Network Faults

Results presented in this subsection demonstrate the protocol's ability to protect from network faults as presented in Section 6.4.

11.6.1 Single Fault Analysis

Figure 11-8 shows the impacts of network faults on a single event route and its corresponding cumulative pulse transmission count without compression. We model a fault as a circular area, which is parameterized by its center and radius. In Figure 11-8a, for example, the centers of fault-1 and fault-2 are located at coordinates (5,6) and (3,4) respectively. An event is

generated at a distance of cell-count 11 and the route diversity δ is set to 1. Observe that both the faults block the route of the event (see the no-fault route in Figure 11-8a).

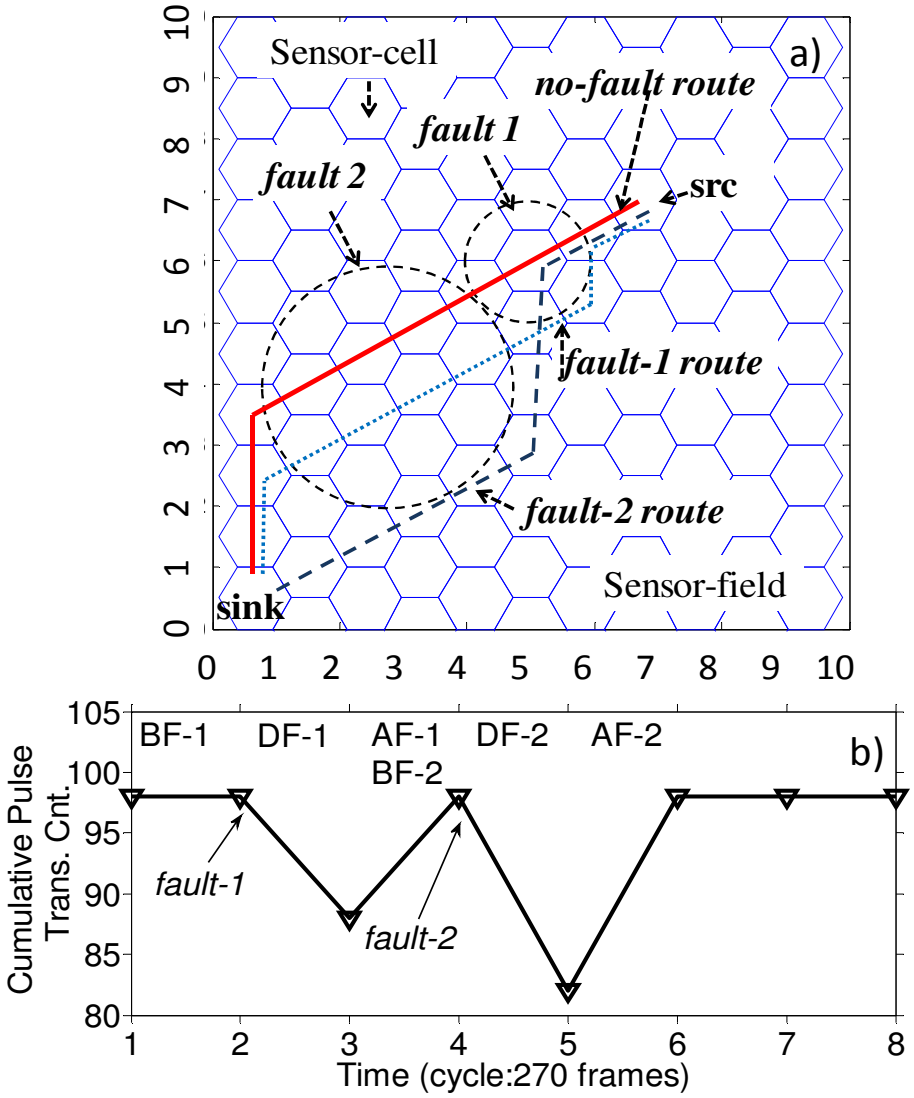


Figure 11-8: Impacts of faults on pulse transmissions in CPS

The time in this experiment is divided into multiple cycles, each containing 270 frames. An event generation happens after the route discovery process completes in each cycle. At the end of every 270-frame cycle, the cumulative pulse transmission count is calculated. The faults are programmed to happen in the 2^{nd} and 4^{th} cycles respectively. A fault is present only during

the specified 270-frame cycle. The *Route Discovery* process helps sensors to re-discover their new next-hops, and the *Response Mechanism* ensures pulse delivery towards the sink at the presence of faults. For a specific fault, there are three cycles: before-fault (BF), during-fault (DF) and after-fault (AF). Figure 11-8b reports as to how the cumulative pulse transmission count varies in these cycles. It is the way to conclude how the fault impacts the cumulative pulse transmission count.

As shown in Figure 11-8b, during BF-1 for fault-1, the cumulative pulse transmission count is 98 for an event generated in this period. After fault-1 occurs, the cumulative pulse transmission count reduces to 88 during DF-1. As expected, the pulse transmission count increases monotonically with the forwarding node count along the route. During DF-1, the newly constructed fault-1 route includes fewer forwarding nodes than the no-fault route as shown in Figure 11-8a. That's why, the cumulative pulse transmission count decreases during DF-1. During AF-1, the cumulative pulse transmission count returns to 98 due to fact that the first faulty area is recovered and the route changes back to the no-fault route as shown in Figure 11-8a. Subsequently, fault-2 occurs at the 4^{th} cycle. The cumulative pulse transmission count variation is similar to that of the fault-1.

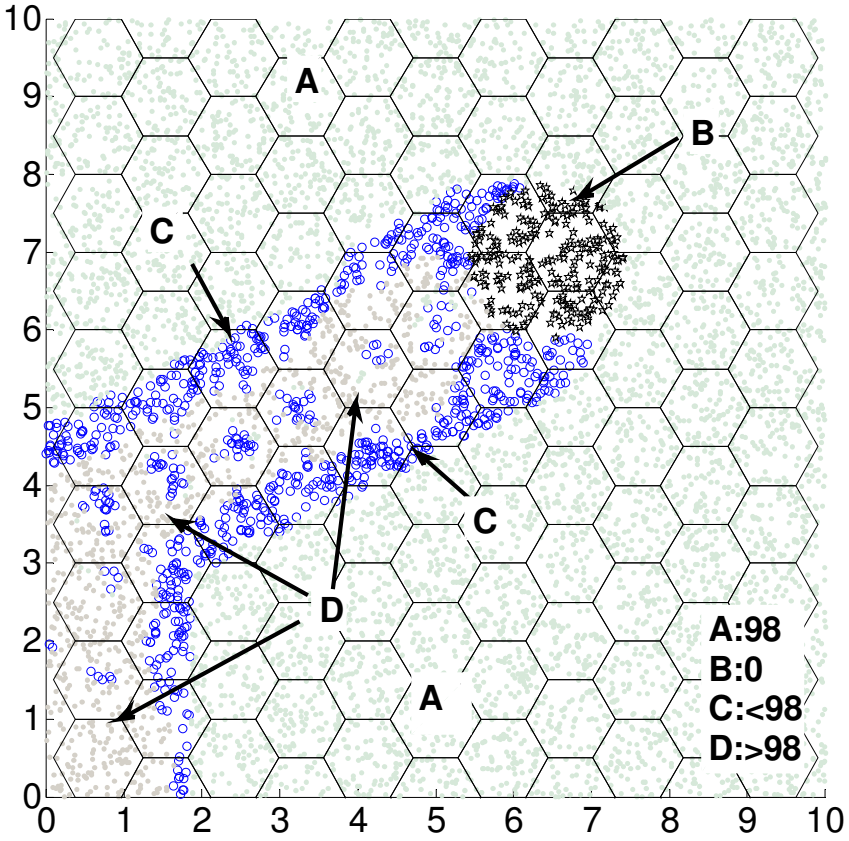


Figure 11-9: Impacts of network faults with radius 1 in CPS

11.6.2 Impacts of Fault Locations

Faults of different sizes (i.e., radii 1 and 2 units) are generated at randomly chosen locations in order to observe their impacts on cumulative pulse transmission count and event loss rate (ELR) during a period of 90-frames. A single event is generated at a distance of cell-count 11. The route diversity δ is set to 1. In the absence of network faults, the cumulative pulse transmission count for this event is 98.

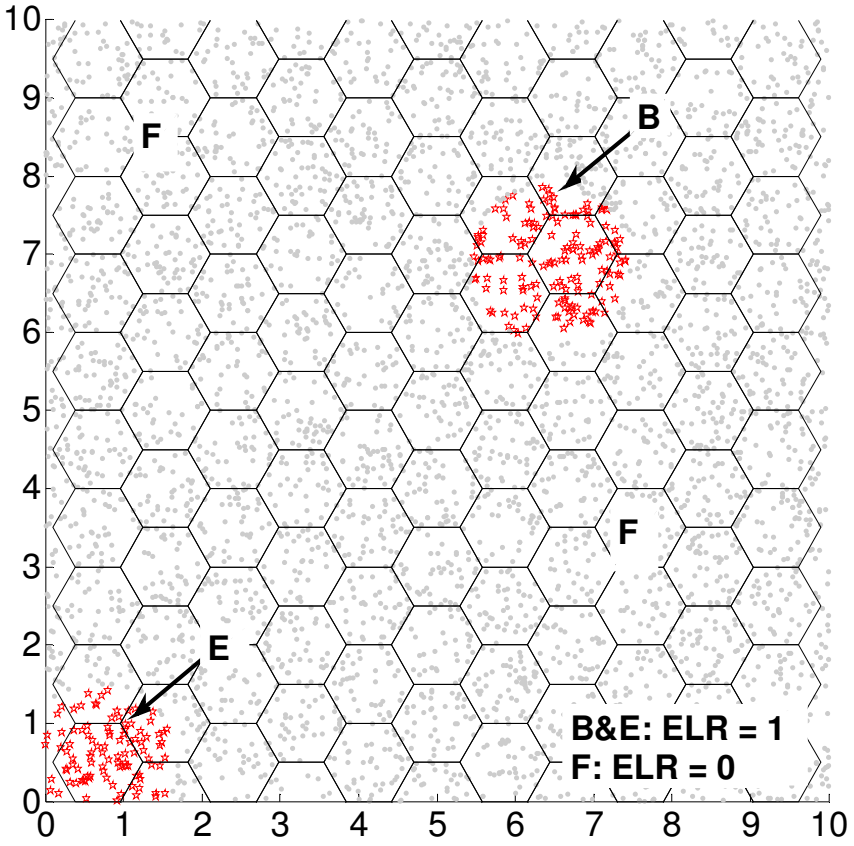


Figure 11-10: Impacts of network faults on ELR with radius 1 in CPS

Figures 11-9 and 11-10 show the effects of faults with radius equal to 1. As shown in Figure 11-9, a fault in *Area A* does not impact the cumulative pulse transmission count and ELR. It is because any fault located in this area does not block the no-fault route (see Figure 11-8a). On the contrary, a fault in *Area B* completely blocks the source node so that no event can be generated. Thus, the cumulative pulse transmission count is zero when a fault falls into *Area B*. Furthermore, a fault in *Area C* reduces the cumulative pulse transmission count compared to that without fault. This is because the number of forwarding nodes in the no-fault route is more than that of the post-fault newly discovered route. A fault in *Area D*, however, increases the cumulative pulse transmission count due to the fact that it involves more forwarding nodes in the

newly discovered route. Figure 11-10 shows the impacts of fault locations on ELR at the sink. ELR is equal to 0 in *Area F* and 1 in *Areas B* and *E*. In other words, event loss happens in *Areas B* and *E* which blocks either the source or the sink node.

The impacts of fault locations with radius equal to 2 on the cumulative pulse transmission count and ELR are shown in Figures 11-11 and 11-12. The effects are similar to that of a fault with the radius equal to 1, except the sizes of these areas. Specifically, the sizes of *Areas A* and *F* are smaller but the sizes of *Areas B*, *C*, *D* and *E* are larger.

The general observation is that as long as a fault does not completely block up the routes towards source or sink nodes, the proposed fault-tolerance mechanism, as proposed in Section 6.4, is able to help nodes successfully deliver pulses to the sink.

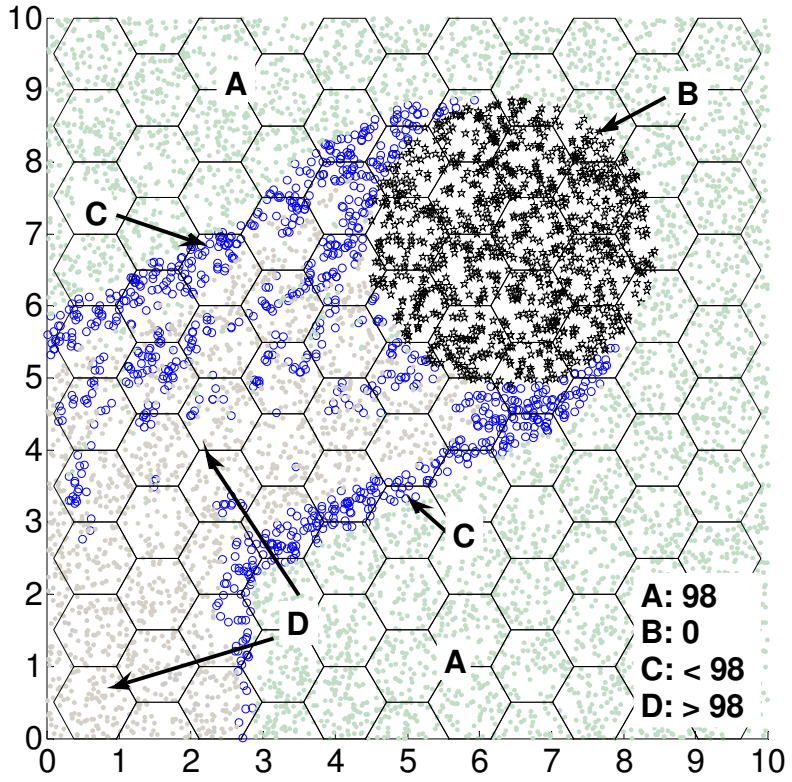


Figure 11-11: Impacts of network faults with radius 2 in CPS

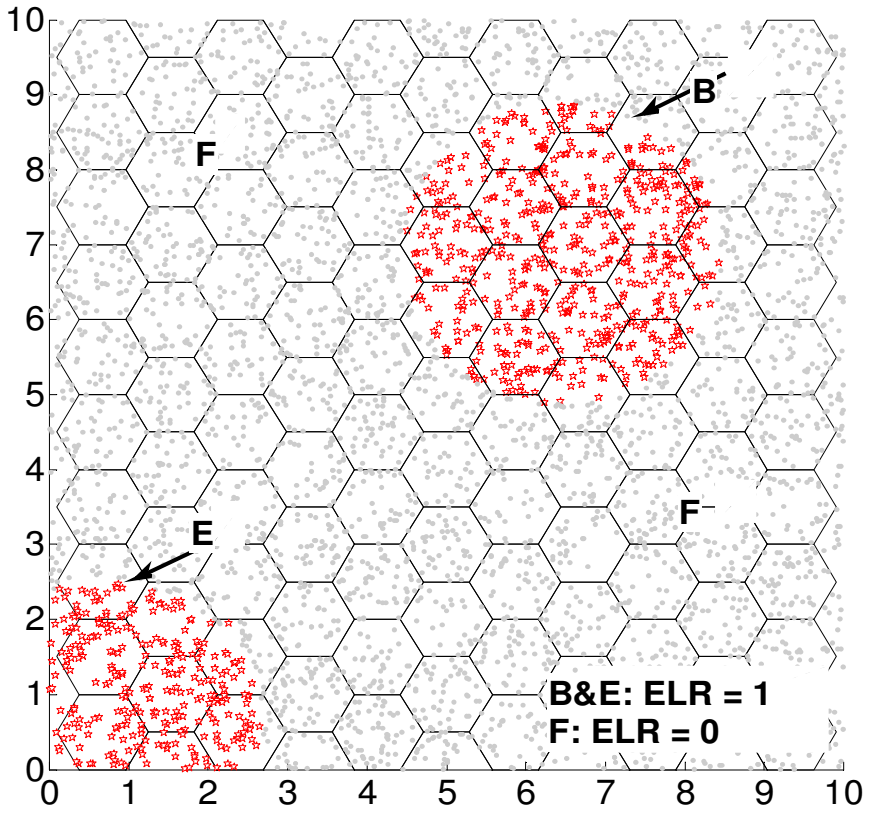


Figure 11-12: Impacts of network faults on ELR with radius 2 in CPS

11.7 Impacts of Node Density

Impacts of node density, defined as the average number of nodes per cell, is evaluated in this section.

11.7.1 Transmission Reduction

Figure 11-13 shows the reduction in the cumulative pulse transmission count due to pulse compression for varying node densities. The results correspond to an event generated at 10 cell-distance from the sink and for two different route diversity (i.e., δ) values 1 and 2. The reduction percentage is expressed as $\left(1-PC_{comp}\right)/PC_{nocomp}$, where PC_{comp} and PC_{nocomp} respectively present the cumulative transmission counts with and without compression.

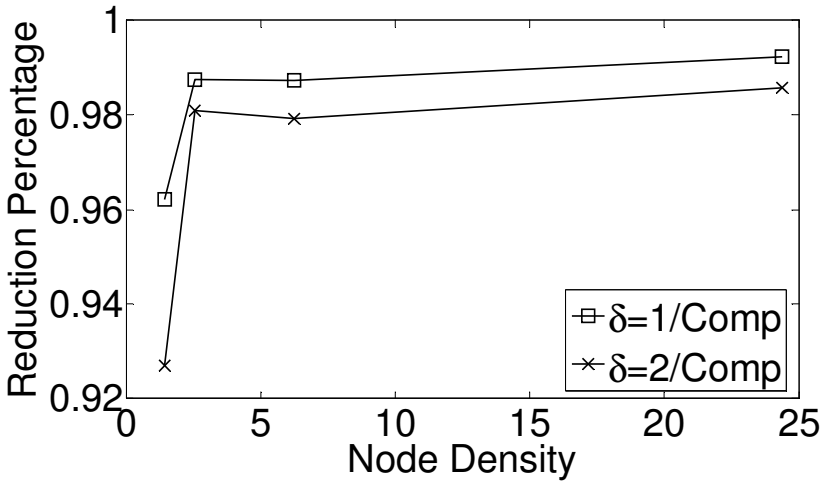


Figure 11-13: Impacts of node density on reduction percentage in CPS

The simulation results in Figure 11-13 demonstrate that for high node density values, the compression algorithm is able to reduce pulse counts by more than 95% for different route diversities. For a given route diversity, the reduction percentage increases with node density. This is because with higher density there is more redundancy in transmissions without compression. The percentage reduction eventually saturates due to the fact that a minimum number of transmissions per cell always happen even under the strictest compression scenarios. Figure 11-13 also demonstrates that the reduction percentage with higher route diversity (i.e., $\delta = 2$) is slightly smaller than that with lower diversity (i.e., $\delta = 1$). It matches with the expression of the reduction percentage $\left(1 - PC_{comp}\right) / PC_{nocomp}$.

11.7.2 Event Loss Rate

Figure 11-14 shows results indicating the impacts of node density on ELR with various PLRs with no response, no compression, and route diversity set to 1. The results derived from models and simulations are in excellent agreement with each other. For a given PLR, the ELR generally decreases with increasing node density due to the increasing pulse transmission

redundancies. This effect, however, becomes less prominent with higher PLR values. When the node density is very high (e.g., greater than 20), the increased pulse transmission redundancy can reduce the ELR to almost zero for very high PLRs. Therefore, without response mechanism we can choose a higher node density and also turn off the compression functionality to reduce the ELR with a high PLR. However, with the response mechanism enabled, the ELR is always vanishingly small with PLR less than or equal to 0.5 for various node densities (see Figure 11-7), and that is irrespective of the compression functionality.

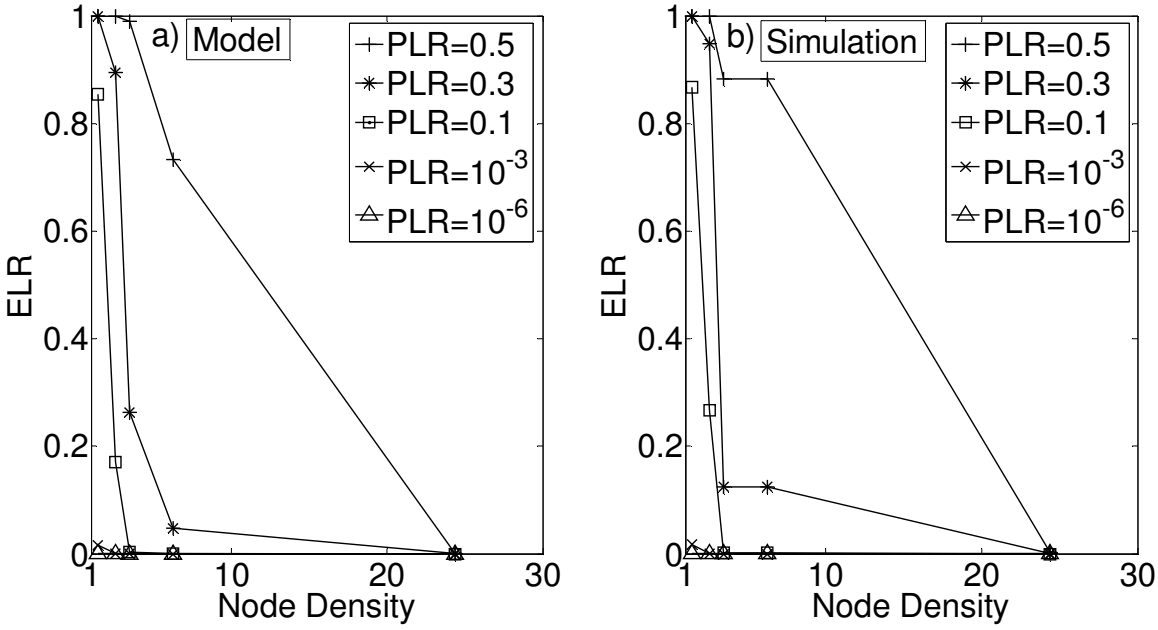


Figure 11-14: Impacts of node density on ELR in CPS

11.8 Summary and Conclusion

This chapter evaluated the cellular pulse switching protocol (CPS) proposed in Chapter 5 in terms of energy-efficiency and reliability. Specifically, this chapter investigated the energy-efficiency in both static event scenario and moving target scenario. In addition, the results indicate that CPS is able to achieve better energy efficiency and offers better localization

resolution than HPS. Furthermore, this chapter provided the results about the impacts of compression to show the effectiveness of compression. Besides the strong reliability against errors, CPS was proven to be robust in presence of network faults. In other words, as long as a fault does not completely block up the routes towards source or sink nodes, the proposed fault-tolerance mechanism in this chapter is able to help nodes successfully deliver pulses to the sink. In the end, this chapter investigates the impacts of node density on the effectiveness of pulse compression and on Event Loss Rate (ELR). To sum up, through simulation based experiments and analytical models, it was shown that CPS can be an effective means for transporting information that is binary in nature with high energy efficiency and strong reliability against errors and faults.

Chapter 12: Performance of Point-to-point Pulse Switching

12.1 Introduction

We developed the event-driven C++ simulator for implementing the MAC framing and pulse routing of the point-to-point pulse switching (PTP) as proposed in Chapter 6. This chapter evaluates the performance of PTP through the results derived from the simulator. The network with terrain size of 10×10 meter^2 includes evenly distributed sensor and actuator nodes. A sink node is placed at the lower left corner of the terrain for synchronization if necessary. Nodes are grouped into 115 regular hexagonal cells, each containing 5 nodes in average.

12.2 Static Event Scenario

This subsection presents results for scenarios in which a static single event causes the sensor within its sensing range to generate a single event pulse, which is transported to the actuator cell using the proposed routing and localization protocols in Chapter 6.

12.2.1 Impact of Distance on Pulse Transmission Count

Figure 12-1 shows the cumulative pulse transmission count as a function of the distance between the destination actuator node and the sensor cell of event generation. Such distance, represented as cell-count, is varied from 1 through 14. Pulse transmission count results are collected for three different route diversities (i.e., δ set to 1, 2 and 3). As expected, Figure 12-1 demonstrates the cumulative pulse transmission count increases with increasing distance (i.e., cell-count) between the source sensor-cell and the destination actuator-cell for a given δ route.

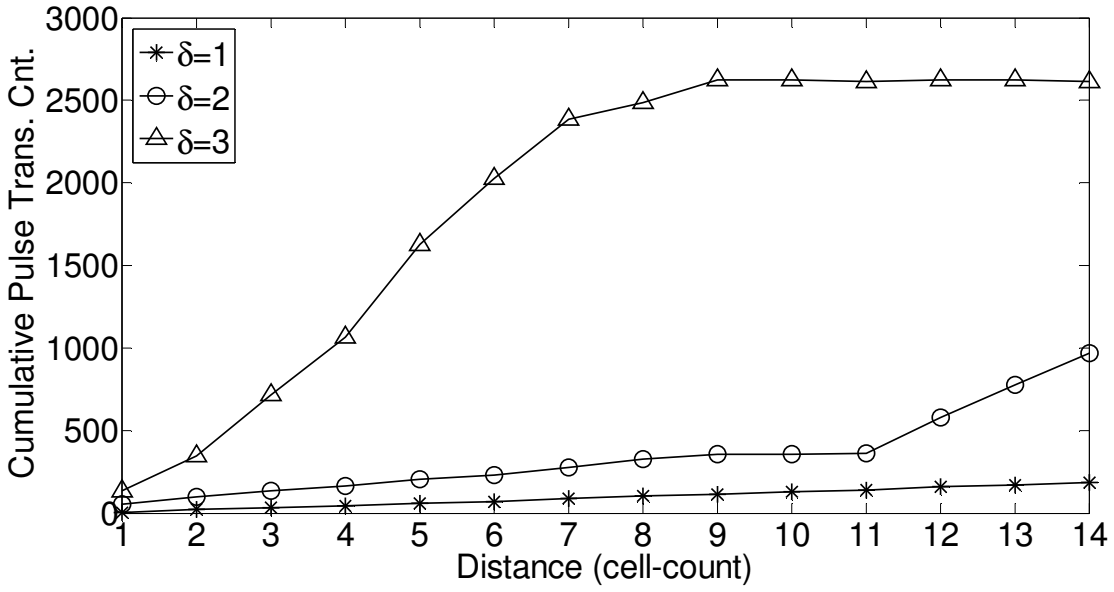


Figure 12-1: Impacts of distance and route diversity on pulse transmission count in PTP

12.2.2 Impact of Route Diversity on Pulse Transmission Count

As shown in Figure 12-1, with increasing route diversity δ, the pulse transmission overhead increases for a specific distance. Specifically, the cumulative pulse transmission count with $\delta = 3$ is significantly higher than that with $\delta = 1$ and $\delta = 2$ for a given distance. Eventually, it approaches to effective pulse flooding for $\delta = 3$ when the distance is greater than 9. So the observations from Figure 12-1 suggest that δ should be kept below 2 for keeping the energy cost under control.

12.2.3 Impact of Compression on Pulse Transmission Count and Delay

Figure 12-2 shows the effects of compression on the cumulative pulse transmission count and event reporting delay with various distances for a given route diversity (i.e., $\delta = 1$) and a distance range (i.e., from 1 to 14). Such distance, represented as cell-count, is varied from 1 through 14. Pulse transmission count results are collected with and without compression. As

expected, Figure 12-2a demonstrates that pulse compression proposed in Section 7.3.2.1 reduces the cumulative pulse transmission count for any distance between 1 cell-count and 14 cell-count. In addition, the event reporting delay is higher with compression than that without compression, as shown in Figure 12-2b. It is due to the back-off process in the spatial pulse compression (see Section 7.3.2.1). There is a trade-off between the energy-efficiency and delay when pulse compression is turned on.

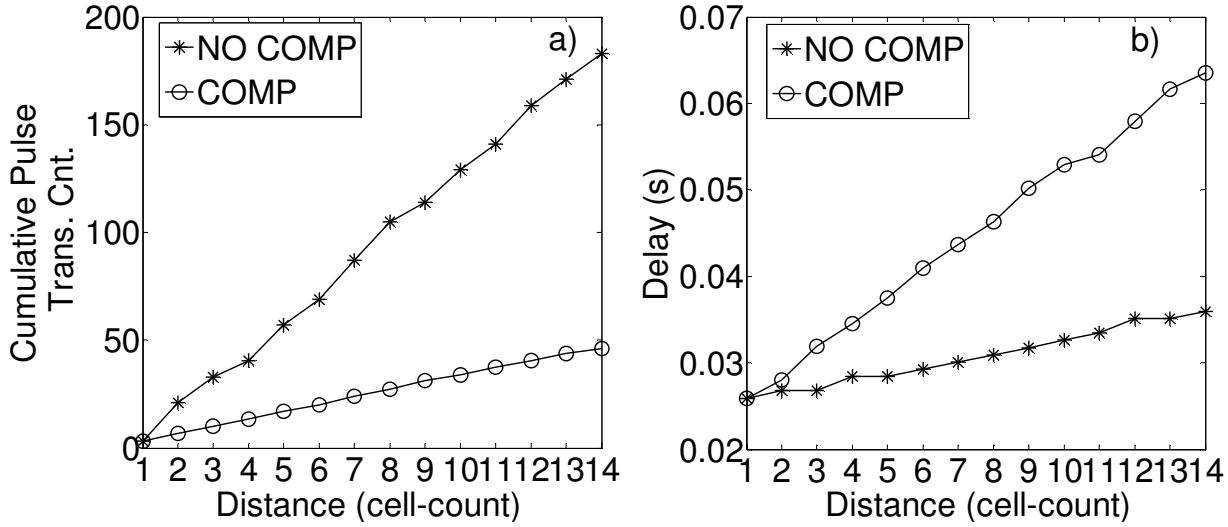


Figure 12-2: Impacts of compression on pulse transmission count and delay in PTP

12.3 Error Analysis

12.3.1 Impacts of False Positive Errors and Protection

The impacts of FPPR on FPEGR for any cell are shown in Figure 12-3. The results derived from the models (Eqns. 8-4 and 8-10) and simulations match very well. FPEGR is extremely small within the practical range of FPPR which is lower than 10^{-4} [44]. This indicates that PTP is fairly immune to false positive errors.

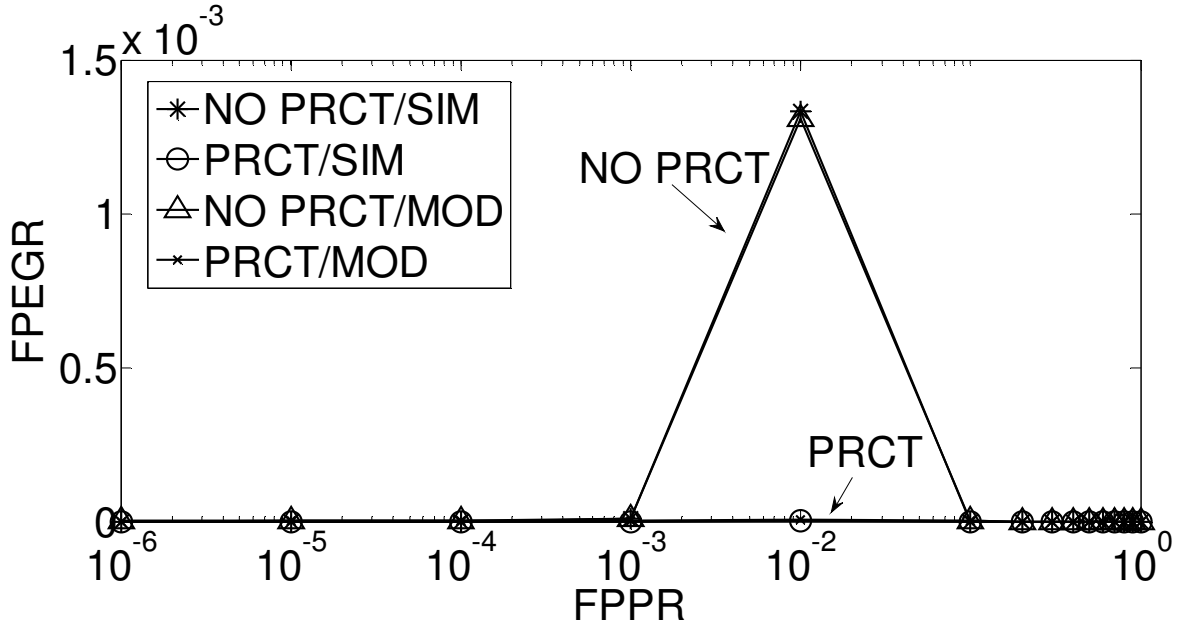


Figure 12-3: Impacts of FPPR on FPEGR in PTP

Also observe that across the range of FPPR from 10^{-6} to 1 in the case of no protection, FPEGR is always extremely small except the point 10^{-2}. In other words, when FPPR is equal to 10^{-2}, FPEGR reaches a peak value. It matches with the equation of FPEGR computation in PTP (see Eqn. 8-10). In addition, with protection enabled, the peak value of FPEGR is reduced a lot. Although FPEGR reaches an optimal value with FPPR equal to 10^{-2}, such a peak is still smaller than 1.4×10^{-3}.

For the FPPR smaller than 10^{-2}, the probability of three matching false positive pulses generated in the *Source Area*, *Destination Area* and *Forwarding Area* of the *Localization Area* for a node is ultra low. When the FPPR greater than 10^{-2}, the chances of more than three false positive pulses generated in these three sub-areas of the *Localization Area* for a node is relatively high. However, the node would deal with these false positive events as a collision and eventually

discard them, according to the collision detection rule in Section 6.4.2.2. That is to say, the node regards these false positive pulses as a collision, and then transmits a collided response pulse to the seeming sender nodes. Since the fact that the seeming senders did not send any pulse to this node, they simply discard the received collided response pulse. Without the reception of the pulse retransmissions from the seeming senders, the node will not forward any false positive event. In this way, the collision handling mechanism in PTP is an inherent protection from high-rate false positive pulses.

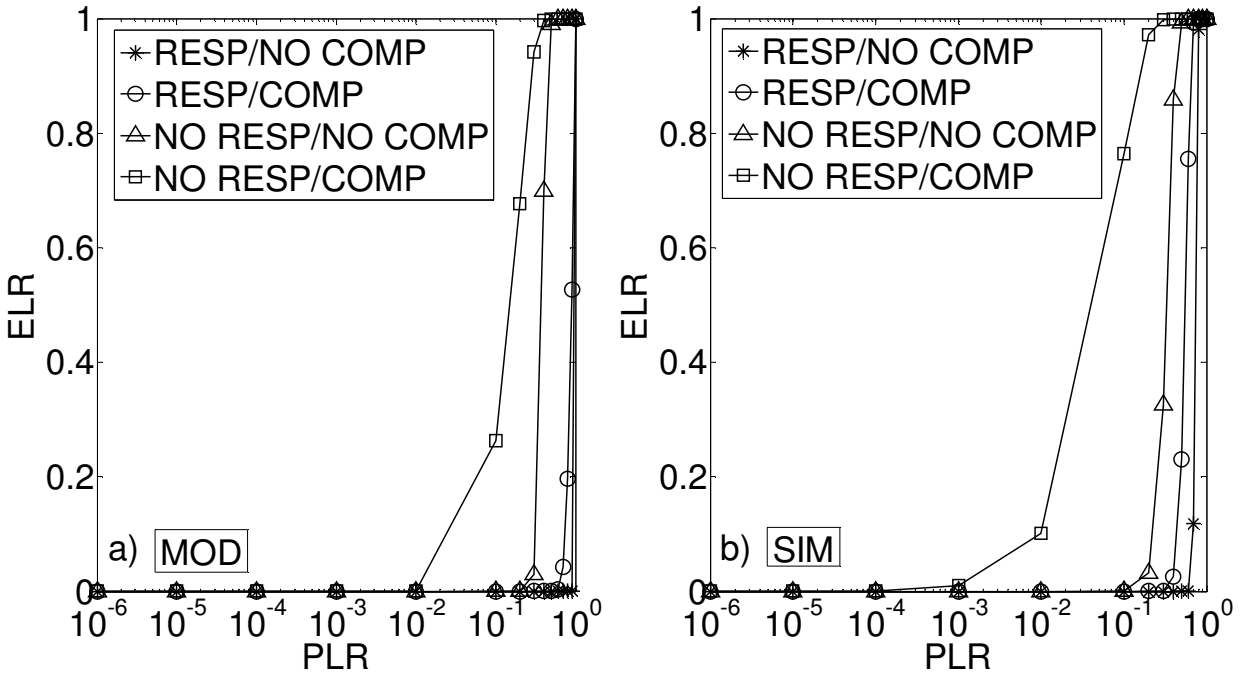


Figure 12-4: Impacts of PLR on ELR without protection in PTP

12.3.2 Impacts of Pulse Loss Errors

Figures 12-4 and 12-5 depict the impacts of PLR on ELR with and without response when protection is turned on and off for a single event generated at a distance of cell-count 8. The route diversity δ is set to 1. Observe that for the practical range of PLR (less than 10^{-4}) [44],

the ELR for both cases remains vanishingly small and it is generally insensitive to the value of PLR. Note that when the route diversity δ is larger than 1, the system reliability against pulse loss errors would be stronger than that with $\delta = 1$ due to more pulse transmission redundancies.

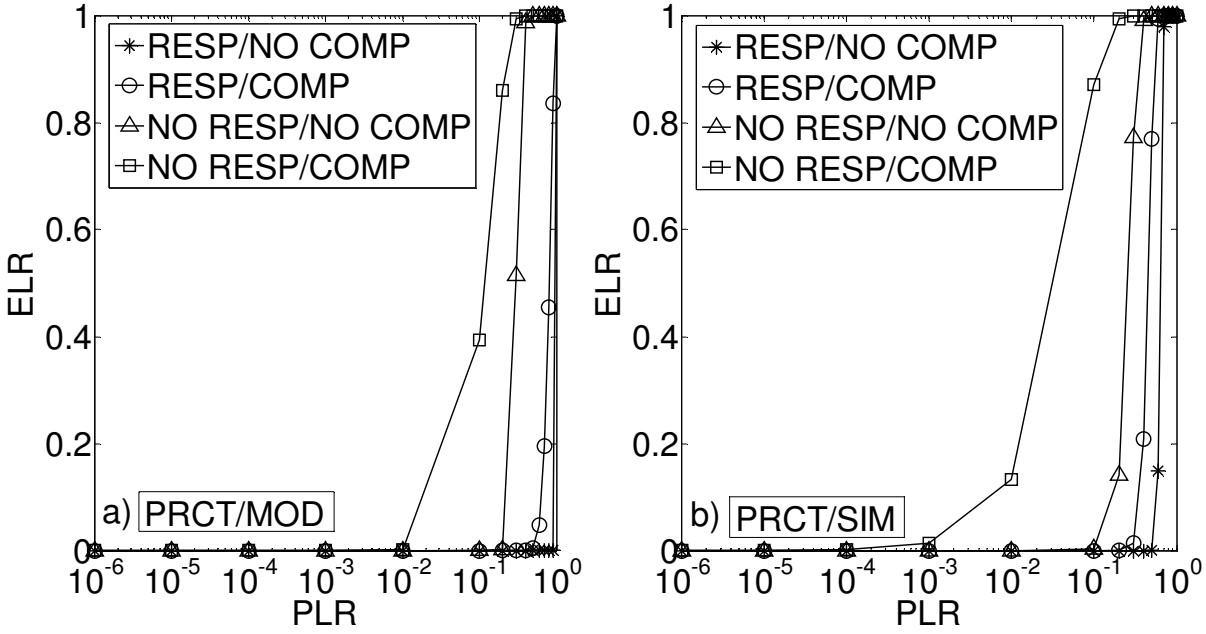


Figure 12-5: Impacts of PLR on ELR with protection in PTP

In the case of no response as shown in Figures 12-4 and 12-5, observe that the results derived from models (i.e., Eqns. 8-12 and 8-14 with retransmission times θ_k to be 1) and simulations match quite well. The minor differences were found to be stemming from limited simulation time. In the case with response enabled, we set the maximum value of retransmission times θ_k to be 100 in the model calculation. In this way, Eqns. 8-12 and 8-14 respectively model the lower bounds of the ELRs without and with protection, which are smaller than the ELRs derived from simulations. It is exactly how Figures 12-4 and 12-5 demonstrate.

These two figures also show the impacts of the response mechanism, pulse compression method, and protection algorithm on the relationship between PLR and ELR. We define the critical point of PLR to be the maximum value of PLR corresponding to zero ELR.

Figures 12-4 and 12-5 show that the critical points of PLR without and with protection are both larger than 0.5 when the response mechanism is activated. θ_k introduced by *Response Mechanism* contributes a lot to decrease the ELR as low as zero for high PLRs. With response mechanism disabled, as shown in Figures 12-4 and 12-5, the critical points with and without protection are both within 10^{-3} and 10^{-1} which are lower than 0.5 but still higher than the practical range of PLR (less than 10^{-4}) [44]. It shows that the proposed response mechanism in Section 6.4 strengthens the system reliability against pulse loss errors, especially with high PLR.

No matter *Response Mechanism* or *Protection* is enabled or not, the ELR without compression is lower than that with compression for a given PLR. The higher ELR with compression is caused by low transmission redundancy which leads to high probability of event loss. Note that although the compression mechanism is less immune to pulse losses compared to the without compression case, the resulting ELR is still low corresponding to the practical PLR range (less than 10^{-4}) [44].

Compared to the no-protection cases, the ELR with SPC protection is similar to that without SPC for model results obtained from Eqns. 8-12 and 8-14. According to Section 8.4.2, the value of e with SPC, the probability of losing an event once on any transmission hop on the route of the event, is higher comparing to that without protection. However, the exponential number θ_k hugely reduces the probability ELR. With higher θ_k, the offset effect of *Response*

Mechanism on a higher *e* becomes stronger. From the observations above, *Response Mechanism* is able to offset the increase of the ELR with protection for the given PLR.

12.4 Collision Handling

This subsection presents results in scenarios where four events collide at a specific cell. With the help of the collision handing mechanism in Section 6.4, collisions are resolved and these four events are respectively transported to their own destination cells. The results are collected with three different event rates (i.e., $\lambda_1 = 120$ events/sec/cell, $\lambda_2 = 12$ events/sec/cell, $\lambda_3 = 6$ events/sec/cell).

12.4.1 Impacts of Random Back-off Range

As shown in Figure 12-6, we tune the collision random back-off range R_{rnd} and observe its impacts on the cumulative pulse transmission count and delay. Results for each event rate include six points corresponding to six specific random back-off ranges. From these points, we can observe the variation trend of pulse transmission counts.

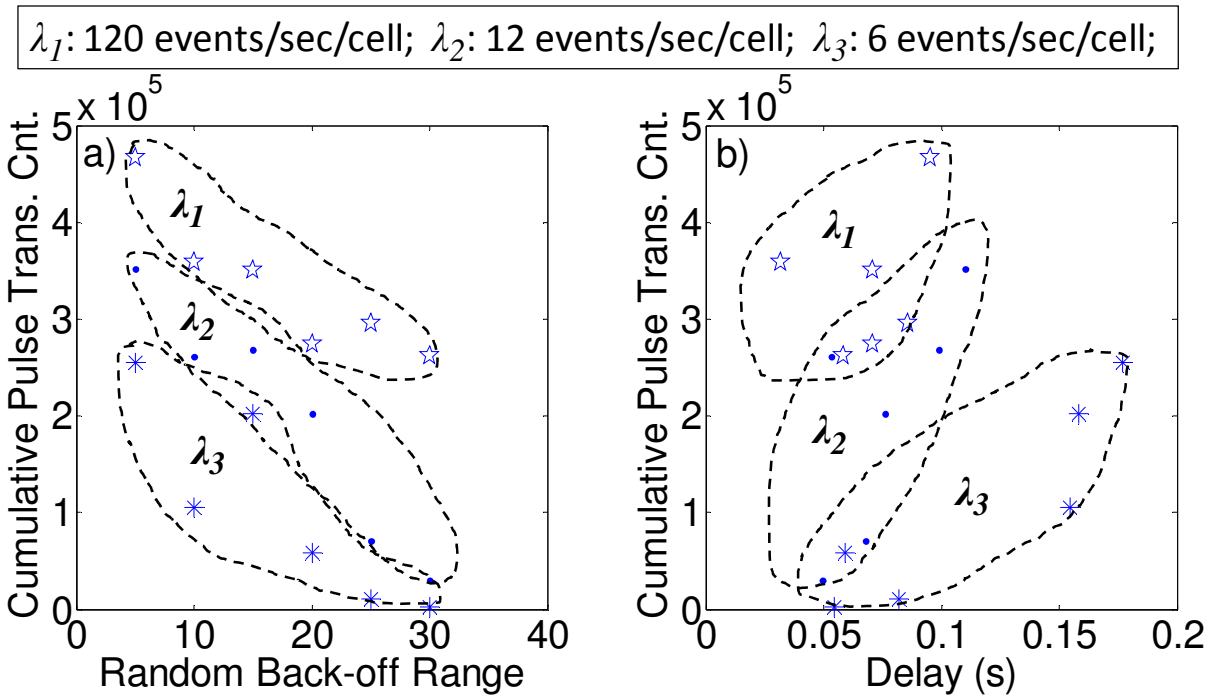


Figure 12-6: Impacts of random back-off range of collision handling in PTP

In Figure 12-6a, for a given event rate, the cumulative pulse transmission count decreases with the increasing random back-off range. It is because that the chances of second collision occurrence after the first-time back-off procedure are lower with a larger random back-off range. It results in less pulse retransmissions due to collisions. In addition, the cumulative pulse transmission count is larger with a higher event rate for a given random back-off range. Specifically, the pulse transmission count for λ_1 is greater than that for λ_2, which is greater than that for λ_3.

Figure 12-6b shows the relationship between the cumulative pulse transmission count and the delay for three different event rates. For a given event rate, the cumulative pulse transmission count increases with the increasing delay. It is due to the fact that a smaller random back-off range leads to more collisions, which brings larger delay and more pulse retransmissions.

Therefore, in such a four-event collision scenario in this subsection, we can choose a relatively large random back-off range in order to reduce the energy consumption and delay.

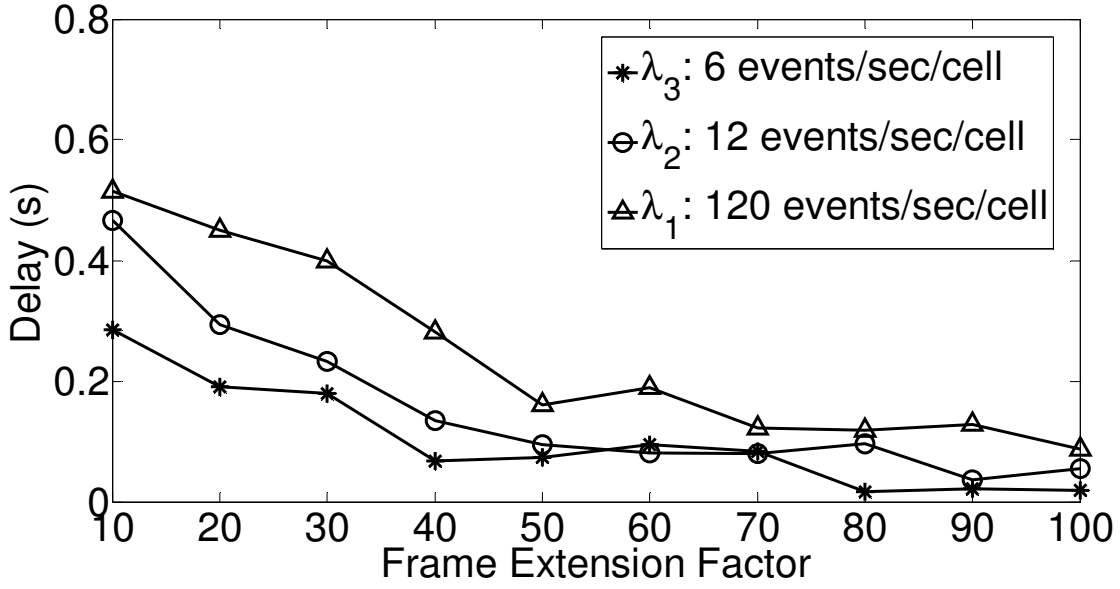


Figure 12-7: Impacts of frame extension factor in PTP

12.4.2 Impacts of Frame Extension Factor

As shown in Figure 6-4, the *Localization Area* of a frame has only one set of *Source Area*, *Destination Area* and *Forwarding Area*. The reason why a collision happens is that more than one pulse are in the *Source Area* or *Destination Area*. We extend the *Localization Area* to include m sets of *Source Areas*, *Destination Areas* and *Forwarding Areas* ($m \geq 1$). m is named as *frame extension factor*. With $m > 1$, we let each node randomly select its own set (among these m sets) to send a pulse. Intuitively, the chances of collisions are smaller and the frame duration is longer with a greater m.

Figure 12-7 shows the impacts of the frame extension factor on the delay. We observe that the delay fluctuatingly decreases with the increasing frame extension factor for a given event rate. The reason is as follows. As discussed in the collision resolution process in Section 6.4.2.2,

the delay is proportionate to the number of back-off processes and also the frame duration. With a larger frame extension factor, the reduced number of back-off processes contributes more than the increased frame duration to the delay. Therefore, we can increase the frame extension factor for the sake of high energy-efficiency and low delay at the presence of intense collisions. Note that in absence of collisions the frame extension factor should be as small as possible, which turns out to be 1.

12.5 Summary and Conclusion

This chapter evaluated the point-to-point pulse switching protocol (PTP) proposed in Chapter 6. This chapter first investigated the impacts of distance, route diversity and compression on energy-efficiency in the static event scenario. Then it provided the results of error analysis in order to show the system reliability of PTP against false positive errors and pulse loss errors. In addition, the collision handling mechanism in PTP was proven to be an inherent protection from high-rate false positive pulses. In the end, this chapter showed the impacts of random back-off range and frame extension factor on energy-efficiency and delay at the presence of collisions. Through simulation and analytical models, it is shown that PTP can be an effective means for point-to-point transporting information that is binary in nature with high energy efficiency, low event reporting delay and strong reliability against errors.

Chapter 13: Summary and Future Work

13.1 Summary

In this dissertation, we proposed a family of transformative pulse switching protocols that are able to implement packet-less event communications for wireless networking applications. This ultra light-weight pulse switching architecture includes: 1) Hop-angular Pulse Switching (HPS), 2) Cellular Pulse Switching (CPS), and 3) Point-to-point Pulse Switching (PTP). HPS and CPS were designed for binary event monitoring and target tracking applications in WSNs, and PTP was developed for binary event sensing and actuation applications in WSANs. All three of them utilize UWB-IR as the PHY layer technology.

The key idea behind pulse switching is to abstract a single pulse, as opposed to multi-bit packets, as the information exchange mechanism. An event is coded as a single pulse which is transported multi-hop to a destination (i.e., a sink or an actuator) while preserving the event's localization information in the form of the pulse's temporal position within a synchronized frame.

A series of energy saving, error handling and fault tolerance measures were presented. Analytical models were also built for comparing the performances of pulse and packet in terms of power consumption and delay. The model results showed that the proposed HPS, CPS and PTP protocols are able to outperform comparable packet solutions under realistic scenarios. Moreover, an event driven C++ simulator was developed to evaluate the performance of all three pulse switching in different scenarios. Through analytical models and simulations, the protocols were proven to be effective for transporting binary information with high energy efficiency, low event reporting delay, and strong reliability against errors and faults.

13.2 Future Work

The following two architectural extensions of pulse switching will be pursued as the immediate future work.

13.2.1 Pulse-Packet Coexistence

In order to design a network supporting multiple application types with events and packets with varying information content, we plan to investigate mechanisms for simultaneously supporting both pulse and packet based transports. We take the hop angular pulse switching architecture as an example.

Two ideas will be explored. The first one is to insert a packet sub-frame as a part of the same frame as proposed in Chapter 4. The second one is to define a new packet based frame and interleave it in between the pulse based frames, along with the *D* frames. Such interleaving can achieve complete functional decoupling between the pulse and packet modes. This way, any conventional packet based frames (e.g., [12]) and the corresponding MAC and routing layer protocols can be deployed without interfering with the proposed hop-angular pulse based operations including the wave-front routing. The performance of the pulse paradigm, however, will be affected by the existence of the packet frames, which can force additional delay between the successive S-L-T cycles.

The researchable items in this topic will include: 1) designing co-existence architectures with packet based paradigm TDMA-W [12] with spanning tree based multi-point to point routing and the integrated MAC-routing mechanism from [27], and 2) evaluating the performance of packet and pulse paradigm individually, as well their mutual interference in terms of energy and delay with varying mixes of pulse and packet traffic in the network.

13.2.2 Multi-Sink Pulse Transport

As discussed in Section 1.5, the proposed pulse switching in this dissertation is mainly for small networks and single-sink scenarios. As the network size grows, the energy consumption per node and worst-case event reporting delay increase due to the larger distance between sensors and the sink. In order to enhance the scalability of pulse switching, we will extend the baseline network and event traffic model by allowing simultaneous pulse switching towards multiple sinks. This subsection uses the cellular pulse switching architecture as an example.

Depending on pre-programmed destinations, a node can report an event to more than one sinks. The network model requires that one sink node should assume the role of synchronizer for all sensors and the other sink nodes. A separate frame is used for each sink in the network. The frame structure as shown in Figure 5-2 remains unchanged except that the *Sync Area* is utilized only in the frame corresponding to the synchronizing sink. Each sensor maintains its sink-specific routing table information for pulse forwarding. To achieve this, all sensors take part in the route discovery process for each individual sink during a *Discovery Area* started by the corresponding sink.

The key researchable issues in this task will be to develop and evaluate models for estimating energy, delay and their tradeoffs with varying number of sinks and heterogeneous event destinations under different network topologies and event generation rates.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] C. R. Farrar, G. Park, D. W. Allen and M.D.Todd, "Sensor network paradigms for structural health monitoring", Journal of Structural Control and Health Monitoring, vol. 13, no. 1, pp. 210-225, 2006.

[2] Ashfaq Hussain Farooqi and Farrukh Aslam Khan, "Intrusion detection systems for wireless sensor networks: A survey", Communications in Computer and Information Science, Vol. 56, pp.234-241, 2009.

[3] Z. Yuanjin, C. Rui, and L. Yong, "A new synchronization algorithm for UWB impulse radio communication systems", In Proceedings of the International Conference on Communication Systems (ICCS), pp. 25-29, 2004.

[4] Ian F. Akyildiz and Ismail H. Kasimoglu, "Wireless sensor and actor networks: research challenges", Elsevier Ad Hoc Networks, vol. 2, no. 4, pp. 351-367, 2004.

[5] J. Gummeson, S. S. Clark, K. Fu and D. Ganesan, "On the limits of effective hybrid micro-energy harvesting on mobile CRFID sensors", In Proceedings of the 8th ACM International Conference on Mobile systems, applications, and services (MobiSys '10), pp. 195-208, New York, NY, USA, 2010.

[6] Y. Zhu, R. Sivakumar, "Challenges: communication through silence in wireless sensor networks", In Proceedings of the 11th ACM International Conference on Mobile Computing and Networking (MobiCom '05), pp. 140-147, New York, NY, USA, 2005.

[7] A. Jain, M. Gruteser, and D. Grunwald, "Benefits of packet aggregation in Ad-Hoc wireless networks", Technical Report CU-CS-960-03, Dept. of Computer Science, Boulder, Colorado, August, 2003.

[8] J. Peng, L. Cheng, B. Sikdar, "A wireless MAC protocol with collision detection", IEEE Transaction on Mobile Computing, vol. 6, no. 12, pp. 1357–1369, Dec. 2007.

[9] T. van Dam, K. Langendoen, "An adaptive energy efficient MAC protocol for wireless sensor networks", In Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys '03), pp. 171-180, New York, NY, USA, 2004.

[10] J. Polastre, J. Hill, D. Culler, "Versatile low power MAC for wireless sensor networks", In Proceedings of the 2nd International ACM Conference on Embedded Networked Sensor Systems (SenSys '04), pp. 95-107, New York, NY, USA, 2004.

[11] G. Halkes, T. V. Dam, K. Langendoen, "Comparing energy-saving MAC protocols for wireless sensor networks", ACM Mobile Networks and Applications 10, pp. 783-791, 2005.

[12] Z. Chen, A. Khokhar, "Self-organization and energy-efficient TDMA MAC protocol by wake up for wireless sensor networks", 1st IEEE Communication Society Conference on Sensor and Ad Hoc Communications and Networks (SECON), pp. 335–341, Oct. 2004.

[13] Nikolaos A. Pantazis, Dimitrios J. Vergados, Dimitrios D. Vergados and Christos Douligeris, "Energy efficiency in wireless sensor networks using sleep mode TDMA scheduling", Elsevier Ad Hoc Networks, vol. 7, pp. 322-343, Mar. 2009.

[14] Munir, M.F. and Filali, F., "Low-energy, adaptive, and distributed MAC protocol for wireless sensor-actuator networks", 18th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp.1-5, Sept. 2007.

[15] R.C. Shah and J.M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks", IEEE Wireless Communications and Networking Conference (WCNC), vol.1, pp.350-355, Mar 2002.

[16] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks", In Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom '00), pp.56-67, New York, NY, USA, 2000.

[17] Qing Cao, T. Abdelzaher, Tian He, and R. Kravets, "Cluster-based forwarding for reliable end-to-end delivery in wireless sensor networks", 26th IEEE International Conference on Computer Communications (INFOCOM '07), pp.1928-1936, May 2007.

[18] Vinh Tran Quang and Takumi Miyoshi, "Adaptive routing protocol with energy efficiency and event clustering for wireless sensor networks", IEICE Transactions on Communications, vol. E91.B, no. 9, pp. 2795-2805, 2010.

[19] Fengyuan Ren, Jiao Zhang, Tao He, Chuang Lin, and S.K.D. Ren, "EBRP: Energy-balanced routing protocol for data gathering in wireless sensor networks", IEEE Transactions on Parallel and Distributed Systems, vol. 22, no.12, pp.2108-2125, Dec. 2011.

[20] A. Boukerche, R.B. Araujo, and L. Villas, "A novel QoS based routing protocol for wireless actor and sensor networks", IEEE Global Telecommunications Conference (GLOBECOM '07), pp.4931-4935, Nov. 2007.

[21] Arjan Durresi, Vamsi Paruchuri, Leonard Barolli, "Delay-energy aware routing protocol for sensor and actor networks", In Proceedings of 11th International Conference on Parallel and Distributed Systems, vol. 1, pp. 292-298, July 2005.

[22] Chendong Xu, Xu Li, A. Nayak, and I. Stojmenovic, "Localized delay-bounded and energy-efficient data aggregation in low-traffic request-driven wireless sensor and actor networks", 7th International Wireless Communications and Mobile Computing Conference (IWCMC), pp.7-12, July 2011.

[23] Tommaso Melodia, Dario Pompili, Vehbi C. Gungor, and Ian F. Akyildiz, "A distributed coordination framework for wireless sensor and actor networks", In Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '05), pp. 99-110, New York, NY, USA, 2005.

[24] S.D. Muruganathan, D.C.F. Ma, R.I. Bhasin, and A. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks", IEEE Communications Magazine, vol.43, no.3, pp.S8-13, March 2005.

[25] G. Lu, B. Krishnamachari, "Minimum latency joint scheduling and routing in wireless sensor networks", Elsevier Ad Hoc Networks, vol. 5, no. 6, pp. 832–843, August 2007.

[26] M. Zorzi, R.R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance", IEEE Transactions on Mobile Computing, vol.2, no.4, pp.349-365, Oct.-Dec. 2003.

[27] S. Kulkarni, A. Iyer, and C. Rosenberg, "An address-light, integrated MAC and routing protocol for wireless sensor networks", IEEE/ACM Transactions on Networking, vol.14, no.4, pp.793-806, Aug. 2006.

[28] Wei Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks", In Proceedings of 21st Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02), vol.3, pp.1567-1576, 2002.

[29] C. Fragouli and A. Orlitsky, "Silence is golden and time is money: power-aware communication for sensor networks", In Proceedings of Allerton Conference on Comm., Control and Computing, 2005.

[30] Zijian Wang, E. Bulut, and B. Szymanski, "A distributed cooperative target tracking with binary sensor networks", IEEE International Conference on Communication Workshops (ICC Workshops '08), pp. 306-310, May 2008.

[31] S. Haykin and M. Moher, "Modern wireless communications", Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2004.

[32] M. Win and R. Scholtz, "Impulse radio: how it works", IEEE Communication Letters, vol. 2, no. 2, pp. 36-38, Feb. 1998.

[33] B. Poucke and B. Gyselinckx, “Ultra-wideband communication for low-power wireless body area networks”, Industrial Embedded Systems Resources Guide, 2005.

[34] T. Nakagawa, R. Fujiwara, G. Ono, M. Miyazaki, “UWB-IR receiver with accurate time-interval-measurement circuit for communication/location system”, IEEE International Symposium on Circuits and Systems (ISCAS '09), pp.397-400, May 2009.

[35] M. Ghavami, L. B. Michael and R. Kohno, “Ultra wideband signals and systems in communication engineering”, John Wiley & Sons, Ltd., 2004.

[36] IEEE 802.15 working group for wireless personal area networks DS-UWB PHY layer submission to 802.15 task group, 2005.

[37] M. Flury and Jean-Yves Le Boudec, “Interference mitigation by statistical interference modeling in an impulse radio UWB receiver”, IEEE International Conference on Ultra-Wideband, pp. 393-398, Sept. 2006.

[38] H. L. Van Trees, "Detection, estimation, and modulation theory. Part I: detection, estimation, and linear modulation theory", John Wiley & Sons, Ltd. England, 2001.

[39] S. Lorenz, B. Dong, Q. Huo, W.J. Tomlinson Jr., and S. Biswas, “Pulse based sensor networking using mechanical waves through metal substrates”, SPIE, 2013.

[40] S.T. Abraha, C.M. Okonkwo, E. Trangdiongga and A.M.J. Koonen, “Experimental demonstration of 2 Gbps IR-UWB transmission over 100m GI-POF using novel pulse generation technique”, In 36th European Conference and Exhibition on Optical Communication (ECOC) , pp. 1-3, Sept. 2010.

[41] A. Nosratinia, T.E. Hunter and A. Hedayat, “Cooperative communication in wireless networks”, IEEE Communication Magazine, vol. 42, no. 10, pp. 74-80, 2004.

[42] David W. Allan, Neil Ashby, and Clifford C. Hodge, “The science of timekeeping”, Hewlett Packard, 1997.

[43] “IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirement part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)” *IEEE Std 802.15.4a-2007 (Amendment to IEEE Std 802.15.4-2006)* , pp.1-203, 2007.

[44] Yu. Andreyev, A. Dmitriev, E. Efremova, A. Khilinsky, and L. Kuzmin, “Qualitative theory of dynamical systems, chaos and contemporary wireless communications”, International Journal of Bifurcation and Chaos, vol.15, no.11, 2005.

[45] F. S. Lee and A. P. Chandrakasan, “A 2.5nJ/b 0.65V 3-to-5GHz subbanded UWB receiver in 90nm CMOS”, In IEEE International Conference on Solid-State Circuits (ISSCC '07), Digest of Technical Papers, pp. 116-590, Feb. 2007.