A CROSS-LAYER APPROACH FOR OPTIMIZING THE EFFICIENCY OF WIRELESS SENSOR AND ACTOR NETWORKS

by

Eckhard Bernhard Kohlmeyer

Submitted in partial fulfilment of the requirements for the degree

Master of Engineering (Computer Engineering)

in the

Faculty of Engineering, the Built Environment and Information Technology

UNIVERSITY OF PRETORIA

August 2008

ACKNOWLEDGEMENTS

Thank you to my family and friends, for their support and continued encouragement throughout the duration of my research. With a particular emphasis on the special person in my life, Elke, who has been my biggest daily encouragement and supported me with prayer and persistent motivation.

A special word of thanks also to my supervisor, Prof. Gerhard Hancke from the Department of Department Electrical, Electronic and Computer Engineering, as well as my co-supervisor, Prof. Derrick Kourie of the Department of Computer Science at the University of Pretoria for their continued support.

Most importantly, I am thankful to God, who has kept me afloat throughout the duration of my studies and with whom anything is possible.

SUMMARY OF DISSERTATION

A CROSS-LAYER APPROACH FOR OPTIMIZING THE EFFICIENCY OF WIRELESS SENSOR AND ACTOR NETWORKS

by

Eckhard Bernhard Kohlmeyer

Supervisor: Prof. G. P. Hancke

Department: Electrical, Electronic and Computer Engineering

University of Pretoria

Degree: Master of Engineering (Computer Engineering)

Keywords: wireless sensor actor network, cross-layering, cross-layer framework, latency, energy efficient, network centric, distributed control, clustering, hierarchical routing, data prioritization, aggregation, medium access control, preamble sampling.

Recent development has lead to the emergence of distributed Wireless Sensor and Actor Networks (WSAN), which are capable of observing the physical environment, processing the data, making decisions based on the observations and performing appropriate actions. WSANs represent an important extension of Wireless Sensor Networks (WSNs) and may comprise a large number of sensor nodes and a smaller number of actor nodes. The sensor nodes are low-cost, low energy, battery powered devices with restricted sensing, computational and wireless communication capabilities. Actor nodes are resource richer with superior processing capabilities, higher transmission powers and a longer battery life.

A basic operational scenario of a typical WSAN application follows the following sequence of events. The physical environment is periodically sensed and evaluated by the sensor nodes. The sensed data is then routed towards an actor node. Upon receiving sensed data, an actor node performs an action upon the physical environment if necessary, i.e. if the occurrence of a

disturbance or critical event has been detected. The specific characteristics of sensor and actor nodes combined with some stringent application constraints impose unique requirements for WSANs. The fundamental challenges for WSANs are to achieve low latency, high energy efficiency and high reliability. The latency and energy efficiency requirements are in a trade-off relationship.

The communication and coordination inside WSANs is managed via a Communication Protocol Stack (CPS) situated on every node. The requirements of low latency and energy efficiency have to be addressed at every layer of the CPS to ensure overall feasibility of the WSAN. Therefore, careful design of protocol layers in the CPS is crucial in attempting to meet the unique requirements and handle the abovementioned trade-off relationship in WSANs.

The traditional CPS, comprising the application, network, medium access control and physical layer, is a layered protocol stack with every layer, a predefined functional entity. However, it has been found that for similar types of networks with similar stringent network requirements, the strictly layered protocol stack approach performs at a sub-optimal level with regards to network efficiency. A modern cross-layer paradigm, which proposes the employment of interactions between layers in the CPS, has recently attracted a lot of attention. The cross-layer approach promotes network efficiency optimization and promises considerable performance gains. It is found that in literature, the adoption of this cross-layer paradigm has not yet been considered for WSANs. In this dissertation, a complete cross-layer enabled WSAN CPS is developed that features the adoption of the cross-layer paradigm towards promoting optimization of the network efficiency. The newly proposed cross-layer enabled CPS entails protocols that incorporate information from other layers into their local decisions. Every protocol layer provides information identified as beneficial to another layer(s) in the CPS via a newly proposed Simple Cross-Layer Framework (SCLF) for WSANs.

The proposed complete cross-layer enabled WSAN CPS comprises a Cross-Layer enabled Network-Centric Actuation Control with Data Prioritization (CL-NCAC-DP) application layer (APPL) protocol, a Cross-Layer enabled Cluster-based Hierarchical Energy/Latency-Aware Geographic Routing (CL-CHELAGR) network layer (NETL) protocol and a Cross-

Layer enabled Carrier Sense Multiple Access with Minimum Preamble Sampling and Duty Cycle Doubling (CL-CSMA-MPS-DCD) medium access control layer (MACL) protocol. Each of these protocols builds on an existing simple layered protocol that was chosen as a basis for development of the cross-layer enabled protocols. It was found that existing protocols focus primarily on energy efficiency to ensure maximum network lifetime. However, most WSAN applications require latency minimization to be considered with the same importance. The cross-layer paradigm provides means of facilitating the optimization of both latency and energy efficiency. Specifically, a solution to the latency versus energy trade-off is given in this dissertation. The data generated by sensor nodes is prioritised by the APPL and depending on the delay-sensitivity, handled in a specialised manor by every layer of the CPS. Delay-sensitive data packets are handled in order to achieve minimum latency. On the other hand, delay-insensitive non critical data packets are handled in such a way as to achieve the highest energy efficiency. In effect, either latency minimization or energy efficiency receives an elevated precedence according to the type of data that is to be handled. Specifically, the cross-layer enabled APPL protocol provides information pertaining to the delay-sensitivity of sensed data packets to the other layers. Consequently, when a data packet is detected as highly delay-sensitive, the cross-layer enabled NETL protocol changes its approach from energy efficient routing along the maximum residual energy path to routing along the fastest path towards the cluster-head actor node for latency minimizing of the specific packet. This is done by considering information (contained in the SCLF neighbourhood table) from the MACL that entails wakeup schedules and channel utilization at neighbour nodes. Among the added criteria, the next-hop node is primarily chosen based on the shortest time to wakeup. The cross-layer enabled MACL in turn employs a priority queue and a temporary duty cycle doubling feature to enable rapid relaying of delay-sensitive data. Duty cycle doubling is employed whenever a sensor node's APPL state indicates that it is part of a critical event reporting route. When the APPL protocol state (found in the SCLF information pool) indicates that the node is not part of the critical event reporting route anymore, the MACL reverts back to promoting energy efficiency by disengaging duty cycle doubling and re-employing a combination of a very low duty cycle and preamble sampling. The APPL protocol conversely considers the current queue size of the MACL and temporarily halts the creation of data packets (only if the sensed value is non critical) to prevent a queue overflow and ease congestion at the MACL

By simulation it was shown that the cross-layer enabled WSAN CPS consistently outperforms the layered CPS for various network conditions. The average end-to-end latency of delay-sensitive critical data packets is decreased substantially. Furthermore, the average end-to-end latency of delay-insensitive data packets is also decreased. Finally, the energy efficiency performance is decreased by a tolerable insignificant minor margin as expected. The trivial increase in energy consumption is overshadowed by the high margin of increase in latency performance for delay-sensitive critical data packets. The newly proposed cross-layer CPS achieves an immense latency performance increase for WSANs, while maintaining excellent energy efficiency. It has hence been shown that the adoption of the cross-layer paradigm by the WSAN CPS proves hugely beneficial with regards to the network efficiency performance. This increases the feasibility of WSANs and promotes its application in more areas.

OPSOMMING VAN VERHANDELING

‘N KRUISLAAG-BENADERING VIR OPTIMISERING VAN DIE DOELTREFFENDHEID VAN DRAADLOSE SENSOR- EN AKTUEERDER-NETWERKE

deur

Eckhard Bernhard Kohlmeyer

Toesighouer: Prof. G. P. Hancke

Departement: Elektriese, Elektroniese en Rekenaar Ingenieurswese

Universiteit van Pretoria

Graad: Meester van Ingenieurswese (Rekenaar Ingenieurswese)

Sleutelwoorde: draadlose sensor-aktueerder-netwerk, kruislaag-raamwerk, energie-doeltreffendheid, netwerk-sentries, verspreide beheer, hierargiese roetering, data prioritisering, aggregasie, mediumtoegang-beheer.

Onlangse ontwikkeling het gelei tot die verskynsel van verspreide Draadlose Sensor en Aktueerder-netwerke (DSAN’e), wat in staat is om die fisiese omgewing te observeer, die data te prosesseer, besluite te neem (gebaseer op die observasies) en geskikte aksies uit te voer. DSAN’e verteenwoordig ‘n belangrike uitbreiding van Draadlose Sensornetwerke (DSN) en bestaan uit ‘n groot aantal sensor-nodes en ‘n kleiner aantal aktuator-nodes. Die sensor-nodes is goedkoop, lae-energie, battery-aangedrewe toestelle met beperkte draadlose kommunikasie, meet- en bereken-vermoë. Aktuator-nodes het beter prosessering-vermoë, hoër transmissie-krag en langer batterylewe.

‘n Tipiese DSAN-toepassing volg die volgende stappe. Die fisiese omgewing word periodies gemeet en die data ge-evaluëer deur die sensor-nodes. Die data word dan na die sentraalgeleë aktueerder-node gestuur. Sodra die aktueerder-node die data ontvang, word aksie geneem indien nodig. Die spesifieke eienskappe van sensor- en aktueerder-nodes gekombineer met

streng toepassingsvereistes stel unieke uitdagings aan DSAN'e. Die belangrikste hiervan is lae vertraging, hoë energie-doeltreffendheid en betroubaarheid. Vertraging-minimering and energie-doeltreffendheid word teen mekaar afgespeel.

Die kommunikasie en koordineering binne DSAN'e word bewerkstellig deur middel van 'n kommunikasie-protokol-stapel (KPS). Die vereistes van lae vertraging en energie-doeltreffendheid moet deur elke laag van die KPS adresseer word om die DSAN'e se uitvuitbaarheid te verseker. Die KPS moet dus versigtig ontwerp word om aan al die vereistes te voldoen en die verhouding tussen pakket-vertraging en energie doeltreffendheid so goed moontlik te hanteer.

Die tradisionele KPS is 'n gelaagte protokol-stapel, met elke laag 'n funktionele eenheid. Daar is egter befind dat die streng-gelaagte KPS-raamwerk sub-optimale netwerk-doeltreffendheid toon vir soortgelyke netwerke met soortgelyke streng vereistes. 'n Moderne kruislaagbenadering, wat interaksies tussen KPS-lae voorstel, het onlangs baie aandag begin trek. Die kruislaag-konsep bied optimisering van netwerk-doeltreffendheid en beloof aansienlike verbetering in werkverrigting. Navorsings-literatuur toon dat die kruislaag-konsep nog nie oorweeg is vir aanwending in DSAN'e nie. In hierdie verhandeling word 'n volledige kruislaag-DSAN-KPS ontwikkel, om die verbetering van netwerk-doeltreffendheid te bewerkstellig. Die nuwe kruislaag-DSAN-KPS behels protokol-lae, wat inligting van ander lae in hulle plaaslike besluite insluit. Elke protokol-laag verskaf inligting, geidentifiseer as voordelig vir ander KPS-lae, met die hulp van 'n nuwe Eenvoudige Kruislaag-Raamwerk (EKLR), aan al die ander lae in die DSAN-KPS.

Die nuut-ontwikkelde, volledige kruislaag-DSAN-KPS behels 'n Kruislaag Netwerk-Sentriese Aktueerder-Beheer met Data-Prioritisering (KL-NSAB-DP) toepassingslaag-protokol, 'n Kruislaag-versameling-gebaseeerde Hierargiese Energie/Vertragingsbewuste Geografiese Roeterings- (KL-KHEVBGR) netwerklaag-protokol en 'n Kruislaag-Draer-Waarneming-Veelvuldige-Toegang met Minimum Vooraf Monstering en Dienssiklus-Verdubbeling (KL-DWVT-MVM-DSV) Medium-Toegansbeheer- (MTB) protokollaag. Elke protokol verbeter op 'n bestaande protokol wat gekies was as 'n basis vir ontwikkeling van die nuwe kruislaagbemagtigde protokolle. Daar is bevind dat die bestaande protokolle in die eerste plek net op energie-doeltreffendheid fokus om netwerk-leeftyd te verleng. Maar, die meeste DSAN-toepassings heg dieselfde belangrikheid aan die verkorting van pakket-vertraging. Die kruislaag-opvatting voorsien ‘n manier om die optimisering van vertragings en energie teweeg te bring. ‘n Spesifieke voorstel word in hierdie verhandeling gelewer vir ’n balans tussen vertraging en energie. Die data, wat deur die sensornodes gegenereer word, word deur die toepasingslaag geprioritiseer, en afhangende van die vertragings-sensitiwiteit, op ‘n spesifieke manier deur elke laag in die KPS hanteer. Vertragings-sensitiewe data-pakkette word op ‘n manier hanteer om vertragings te verminder. Die data pakette wat nie vertragings-sensitief is nie word op ‘n spesifieke manier hanteer om die hoogste energie-doeltreffendheid te bereik. Dus word daar meer aandag geskenk aan vertraging-minimering van energie-doeltreffendheid afhangende van die pakket-tipe. Die kruislaag-toepassings-protokollaag verskaf vertragings-gevoelige inligting van ‘n data-paket aan ander lae. As die netwerklaag-protokol dus opmerk dat ‘n datapakket vertragings-gevoelig is, word die roeterings-benadering van ‘n energie-doeltreffende roete, wat bestaan uit nodes met die meeste oorblywende energie, verander na ‘n roeterings-benadering wat die vinnigste roete na die aktueerder-node kies. Dit word bewerkstellig deur die wakker-word-skedule en kanaal-gebruik inligting te oorweeg, wat spesifiek deur die MTB-laag-protokol binne die EKLR gestoor word. Met die addisionele informasie word die ‘volgende-hop’ node keuse primêr op die kortste tyd-tot-wakker-word gebaseer. Die kruislaag-MTB-laag-protokol dra by tot die verkorting van die vertraging, deur middel van ‘n prioritiserings-toustelsel en ‘n tydelike dienssiklus-verdubbelings-eienskap. Dienssiklus-verdubbeling word aangewend as ‘n sensornode se toepassingslaag-protokol aandui dat die node deel is van die rapporterings-roete van ‘n kritieke gebeurtenis. As die toepassings-protokollaag aandui dat die node nie meer deel is van die rapporterings-roete nie, word die dienssiklus-verdubbeling gestaak en teruggekeer na energie-doeltreffendheid met ‘n lae dienssiklus en vooraf monstering. Die toepassings-protokollaag oorweeg die huidige MTB-laag tougrote en voer ‘n tydelike staking van enige pakketgenerasie uit om oorloping van die toustelsel en kongestei in die MTB-laag te voorkom.

Deur simulasies is bewys dat die kruislaag-DSAN-KPS die streng-gelaagde KPS oortref vir verskillende netwerktoestande. Die gemiddelde vertraging van vertragings-sensitiewe datapakette word drasties verminder. Verder word die vertraging van vertragings-onsensitiewe datapakette ook verminder. Oplaas word die energie-doeltreffendheid net effens verminder, soos verwag. Die triviale styging in energieverbruik word oorheers deur die aansienlike vertraging-verbetering van vertragings-sensitiewe datapakette. Die nuwe kruislaag bereik dus ‘n aansienlike vertragings-verbetering vir DSAN’e terwyl ‘n goeie energie-doeltreffendheid behou word. Daar is dus bewys dat die aanwending van die kruislaag-benadering deur die DSAN-KPS hoogs voordelig is in terme van netwerk doeltreffendheid. Dit verhoog die toepassingsmoontlikhede van DSAN’e.

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
Aggr	Aggregation
APPL	Application Layer
B-MAC	Berkeley MAC
CCA	Clear Channel Assessment
CF-PMS	Contention Free Periodic Message Scheduler
CHEAGR	Cluster-based Hierarchical Energy-Aware Geographic Routing
CL	Cross-Layer
CL-CHELAGR	Cross-Layer enabled Cluster-based Hierarchical Energy/Latency-Aware Geographic Routing
CL-CSMA-MPS-DCD	Cross-layer enabled CSMA-MPS with Duty Cycle Doubling
CL-NCAC-DP	Cross-layer enabled NCAC-DP
CLO	Cross-Layer Optimizer
CLOI	Cross-Layer Optimization Interface
CoLaNet	Cross-Layer approach of an energy efficient sensor Network
CPS	CPS
CRC	Cyclic Redundancy Check
CS	Carrier Sense
CSMA	Carrier Sense Multiple Access
CSMA-MPS	CSMA with Minimum Preamble Sampling
CTS	Clear To Send
DAC	Digital to Analog Converter
da-tree	data aggregation tree
DCD	Duty Cycle Doubling
DEAP	Delay-Energy Aware Routing Protocol
DMA-CLD	Dynamic Multi-Attribute Cross-Layer Design
DSP	Digital Signal Processing
GPS	Global Positioning System
GBP	Geometric Broadcast Protocol
GAF	Geographical Adaptive Fidelity

HGC	Hierarchical Geographic Clustering
ISM band	Industrial, Scientific and Medical band
ISO	International Organization for Standardization
LEACH	Low-Energy Adaptive Clustering Hierarchy
LPL	Low Power Listening
MAC	Medium Access Control
MACL	MAC Layer
MANETs	Mobile Ad-hoc Networks
MEMS	Micro Electro-Mechanical Systems
MS-MAC	Mobility-aware MAC protocol for Sensor networks
MST	Minimum Spanning Tree
NCAC-DP	Network-Centric Actuation Control with Data Prioritization
NETL	Network Layer
Non-CL	Non Cross-Layer
OSI	Open Systems Interconnection
PAMR	Power Aware Many-to-Many Routing
PHYL	Physical Layer
RF	Radio frequency
RRDSE	Reliable Reporting of Delay-Sensitive Events
RTS	Request To Send
RX	Receive
SCLF	Simple Cross-Layer Framework
S-MAC	Sensor-Medium Access Control
SMACS	Self-organizing MAC for Sensor networks
SNR	Signal-to-Noise Ratio
SSR	Scalable Source Routing
STEM	Sparse Topology and Energy Management
SYNC	Synchronization
TDMA	Time Division Multiple Access
T-MAC	Timeout-Medium Access Control
TRAMA	Traffic-Adaptive Medium Access
TX	Transmit

WIDENS	Wireless Deployable Network System
WSANs	Wireless Sensor Actor Networks
WSNs	Wireless Sensor Networks
WLAN	Wireless Local Area Network

TABLE OF CONTENTS

PAGE

Chapter 1

1 RESEARCH OVERVIEW

1.1 INTRODUCTION

In the industrialized world, nearly all consumers and businesses make use of embedded sensing and control systems on a daily basis. These systems are critical in manufacturing, automobiles, aviation, building heating and air-conditioning, power distribution, and a huge array of other domains [1]. Modern technological advances have lead to the development of low-power, wireless embedded control systems that have the potential to significantly alter and expand the applicability of control technology.

Recent development has lead to the emergence of distributed Wireless Sensor and Actor Networks (WSANs) which are capable of observing the physical environment, processing the data, making decisions based on the observations and performing appropriate actions [2]. This type of network satisfies an ever-increasing need for a collection of devices that constitute an autonomous network capable of gathering information on the physical world, assess its meaning, communicate it and adapt or control the physical processes or environment accordingly.

WSANs represent an important extension of Wireless Sensor Networks (WSNs). WSNs comprise of a number of small independent devices called sensor nodes that are randomly deployed in a target area to perform a collaborative sensing task. The sensed data is usually collected by a single base station that solely serves as a data gathering and storage facility. In WSANs however, the network comprises heterogeneous sensor nodes and actor nodes. The sensor nodes gather information by observing the physical environment and then relay this information to the actor nodes that take decisions and then perform suitable actions upon the physical environment. The sensor nodes are low-cost, low energy, battery powered devices with restricted sensing, computational and wireless communication capabilities. The role of actor nodes is to collect and process sensor data and then perform appropriate actions if necessary. These actors are resource richer with superior processing capabilities, higher transmission powers and a longer battery life. Moreover, the number of sensor nodes deployed in a target area may be in the order of hundreds or thousands. Such a dense deployment is usually not necessary for sophisticated actor nodes, since actors have higher capabilities and can act on large areas.

The WSANs ability to make autonomous decisions and perform actions in response to sensor measurements provides a widespread potential for applications such as agricultural maintenance, precision farming and automatic irrigation, as well as localized delivery of medication [3]. Furthermore, WSANs can be an integral part of systems such as battlefield surveillance and microclimate control in buildings, nuclear, biological and chemical attack detection, home automation and environmental monitoring [2]. For example, in fire detection applications, the sensor nodes relay the information about the exact origin and fire intensity to water sprinkler actor nodes so that the fire can be extinguished before spreading uncontrollably. Similarly, motion and light sensor nodes in a room can detect the presence of people and then direct the appropriate actor nodes to execute actions based on user pre-specified preferences.

The specific characteristics of sensor and actor nodes combined with some stringent application constraints impose unique requirements for WSANs. The fundamental challenges for WSANs are to achieve low latency, energy efficiency and high reliability. Most WSAN applications demand low end-to-end latency due to a real-time requirement imposed by specific scenarios. For example a fire needs to be monitored in real-time to ensure effective timely counter measures to be employed. Furthermore, sensor nodes are typically battery-driven and are often deployed in very remote or dangerous environment, which implies that these nodes are inaccessible for a battery replacement or recharge. Hence, the energy efficiency of network communications is also crucial, since the resource-constrained sensors nodes feature limited battery lifetime and communication capabilities. Without a viable network lifetime the WSAN becomes infeasible.

The communication inside WSANs is managed via a traditional layered protocol stack situated on every node. The construction of the physical channel is done by the physical layer, the access of the node into the medium is done by the medium access layer, the selection of the routing path through which the node transmits its data and the transport of packets from one node to another is done by the network layer and finally the handling of the actual sensed data is done by the application layer. Careful design of the communication management is crucial in attempting to meet the unique requirements of WSANs. WSANs have only recently come into existence and little research has been conducted in this field that is showing increasing interest in a lot of different application areas [2].

1.2 SCOPE

The research in this dissertation deals with the comprehensive design of a complete WSANs communication protocol stack (CPS) that adopts a cross-layering approach to optimize the network efficiency with regards to latency, energy and reliability. Existing WSANs and where relevant, existing WSN protocols are used as a basis for development of the cross-layer enabled WSAN CPS design that uses a generic framework design to enable cross-layer interaction throughout the CPS. The cross-layer approach is implemented towards facilitating the efficiency optimization of latency, energy (network lifetime) and reliability of WSANs.

1.3 PROBLEM STATEMENT

In WSANs, which characteristically have limited resources, it is required to deliver event data in an energy-efficient way while respecting real-time delay constraints and maintaining good reliability. These requirements must be met by the network’s CPS that typically consists of the physical layer (PHYL), the MAC layer (MACL), the network layer (NETL) and the application layer (APPL). These layers deal with the construction of the physical channel, the access of the node into the medium, the selection of the routing path through which the node transmits its data and the handling of the actual application data. In this traditional layered model each layer in the CPS is designed independently and operates independent of other protocol layers.

The traditional CPS interfaces between protocol layers are static and each layer reacts independently to the network requirements. This paradigm has greatly simplified network design. However, the inflexibility and sub-optimal nature of the independent layers paradigm results in poor or limited performance for WSANs, which strive towards low latency, energy-efficiency and high reliability [2]. To meet these stringent requirements, research and development of a cross-layer CPS design that aims at network efficiency optimization of latency, energy and reliability, across multiple layers of the CPS, is needed for WSANs. A design is needed that suggests a more integrated CPS that allows for network efficiency optimization on a global stack-wide perspective rather than on a traditionally local scale, i.e. within separate layers only.

Taking into consideration that WSAN is a new type of network that evolved from WSNs, a fair amount of literature only exists for WSN protocols. A few WSN literatures have touched on some cross-layer interaction issues. However, the proposed cross-layer interactions are very limited and are not the focal point, but rather another requirement on the side. Little research has been conducted on WSANs specific protocols and none on using the promising cross-layer approach to achieve performance gains [1]. Research is needed in the field of WSANs CPS design and network efficiency optimization to ensure the feasibility of WSANs, characterized by stringent requirements.

1.4 OBJECTIVES

The main objective of this research is to develop a new cross-layer enabled CPS design that features protocols that adopt a cross-layer interaction approach to optimize network latency, energy efficiency (network lifetime) and reliability.

The CPS design should include the following characteristics:

- The CPS should contain protocols suitable for WSANs.
- A cross-layer approach should be implemented by adopting a new framework that allows cross-layer interactions among relevant layers while incurring minimally added overhead.
- The CPS design should contain enhancements, modifications and adaptations of relevant layered physical, MAC, network and application layer protocols to accept, interpret and integrate information from other layers into their local layer specific decisions. This should be done to achieve a stack-wide optimization of the WSAN efficiency with regards to latency, energy and reliability.
- The design should show improved performance compared to a strictly layered WSAN CPS comprising only independent layers that exhibit no interaction.

1.5 RESEARCH APPROACH

The following research approach was taken in fulfilling the objectives of the proposed research.

- To investigate and attain a comprehensive understanding of requirements, challenges and application areas of WSANs.

- To conduct a thorough literature study of existing WSAN and relevant WSN protocols, cross-layer frameworks and existing cross-layer implementations in WSNs to
  - attain an understanding of the operation, difficulties, challenges and problems of physical, MAC, network and application layer protocols,
  - identify which specific protocols are most suitable for use in WSANs that cater for a wide variety of applications,
  - determine which particular protocols could be suitable for adopting the cross-layer paradigm in WSANs and
  - gain insight into different cross-layer frameworks and methodologies.

- To design a new cross-layer framework for WSANs and a new cross-layer enabled WSAN CPS that takes a cross-layer interaction approach to optimize latency, energy efficiency and reliability by
    - investigating and considering existing cross-layer frameworks of similar network types,
    - investigating and determining which layers are relevant to particular cross-layer interaction in WSANs,
    - investigating and determining parameters within every layer in the CPS that may be advantageous to other layers towards optimizing the overall network efficiency of WSANs,
    - modifying and adapting specific single layer WSAN or relevant WSN protocols to accept, interpret and integrate information, that comprises operation or status parameters from other layers into their local layer specific decisions, to ensure overall optimization of the WSAN efficiency with regards to latency, energy, and reliability, and
    - considering trade-offs, challenges and limitations of network efficiency optimization in WSANs.

- To evaluate the performance gains of the cross-layer enabled CPS design with cross-layer interactions as compared to a strictly layered CPS design with no cross-layer interactions.

1.6 DISSERTATION OVERVIEW

The rest of this dissertation is organized in the following manner.

Chapter 2 investigates the fundamentals of WSANs. A description of WSANs, their application areas, challenges and characteristics is given. Chapter 3 focuses more specifically on the communication architecture and protocols for WSANs. It identifies and discusses tasks, requirements and challenges for every layer in the CPS. Furthermore, the existing literature on WSAN protocols and where relevant, literature on WSN protocols is dealt with in greater detail. Chapter 4 presents the cross-layer paradigm, with literature on existing frameworks used for similar types of networks. Existing literature on cross-layering in WSNs is also discussed. On the basis of the existing literature, in Chapter 5, a new cross-layer framework for the WSAN CPS is proposed. Furthermore, on the basis of the existing literature a complete WSAN CPS design is proposed. This WSAN CPS design is then further developed into a new cross-layer enabled WSAN CPS design that implements the proposed cross-layer framework and proposes specific cross-layer interactions. Results of the verification of the new cross-layer enabled CPS design in comparison to the layered CPS will be presented in Chapter 6. A discussion of these results is also given. Finally, the research documented in this dissertation is concluded in Chapter 7.

Chapter

2 WIRELESS SENSOR ACTOR NETWORKS

The Wireless Sensor Actor Network is a new type of network that has emerged in the rapidly expanding field of wireless networks. This chapter serves as a fundamental basis for the understanding of the WSANs and puts the research in this dissertation into context by giving important background regarding the network composition, requirements, challenges and application areas.

2.1 THE WIRELESS SENSOR ACTOR NETWORK CONCEPT

Since the beginning of this decade, a concept has been envisaged that promises a platform for pervasive sensing, surveillance, and computing which may revolutionize information gathering and processing [4]. This visionary concept has lead to the Smart Dust project [5] that finished in 2001 and explored whether an autonomous sensing, computing, and communication system can be packed into a cubic-millimetre mote to form the basis of integrated and massively distributed sensor networks. As miniaturization of electronics proceeds further, a previously unthinkable realization of ‘smart dust’ becomes feasible. Kristofer Pister, one of the Smart Dust investigators, predicts: “In 2010 micro electro-mechanical systems (MEMS) sensors will be everywhere, and sensing virtually everything. Scavenging power from sunlight, vibration, thermal gradients, and background RF, sensor motes will be immortal, completely self contained, single chip computers with sensing, communication, and power supply built in. Entirely solid state, and with no natural decay process, they may well survive the human race” [4]. Pister’s optimistic forecast has increasingly gathered realization due to continuous technological advances and progress in the growing research field of sensor networks.

In the light of the visionary concept and with the quest to observe and control every aspect of the physical environment, recent technological progress has lead to the emergence of WSANs. WSANs can be an integral part of systems such as battlefield surveillance and microclimate control in buildings, nuclear, biological and chemical attack detection, home automation and environmental monitoring [2].

WSANs comprise a collection of heterogeneous sensor nodes and actor nodes, and are not only capable of observing and monitoring the physical world, but also processing the data, making decisions based on the observations and performing appropriate actions. Essentially, the sensor nodes gather information by observing the physical environment and then relay this information to the actor nodes that take decisions and then perform suitable actions upon the physical environment. Sensor nodes are low-cost, low power devices with limited sensing, computation, and wireless communication capabilities. Actor nodes distinguish themselves from sensor nodes in that they are typically resource richer nodes equipped with better processing capabilities, higher transmission powers, and longer battery life.

WSANs can be regarded as a hybrid of the much researched Mobile Ad-hoc NETworks (MANETs) and Wireless Sensor Networks (WSNs). MANETs are self-configuring network of mobile nodes connected by wireless links - the union of which forms an arbitrary topology. The nodes are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably [6]. WSNs comprise distributed small low-cost sensor nodes that collaboratively send sensed data via short wireless transmissions to a base station. WSNs serve as monitoring or observation networks that solely gather sensed data from the physical environment they are deployed in.

WSANs cannot be simply regarded as WSNs due to the existence of mobile actor nodes, nor as general MANETs due to the existence of large amount of “static” tiny sensor nodes [7]. It is to be noted that the two elements in a WSAN, i.e., the WSN and the MANET, have the following important differences despite some similarities between them.

- The number of the nodes in a WSN is significantly larger than that in a MANET.
- Sensor nodes are usually low-cost devices with severe constraints with respect to energy source and computation capabilities; while MANET nodes (similar to actor nodes in WSAN) generally have relatively stronger computation capability and higher energy storage, which allows longer wireless transmission distance.
- Unlike nodes in a MANET, sensor nodes in a WSN are usually stationary or with limited mobility.
- The mode of communication in WSNs typically is many-to-one (from sensor nodes to sink), while it is peer-to-peer in MANETs.

The fusion of WSN and MANETs into WSANs necessitates communication and coordination features of both to be jointly effectively addressed. However, WSAN can be considered to have evolved from WSNs, rather than MANETs, as an extension to a monitoring only concept. WSANs have similar application areas to WSNs. While WSNs have been studied for more than a decade, it can be considered that the extension with actuators is a more recent thrust of research that greatly enhances their capabilities and range of applications, at the cost of addition stringent constraints [4]. This WSAN concept has recently attracted a lot of interest since its evolvement.

For the vast range of WSANs applications (discussed in section 2.5) a basic physical architecture is illustrated in Figure 2.1.

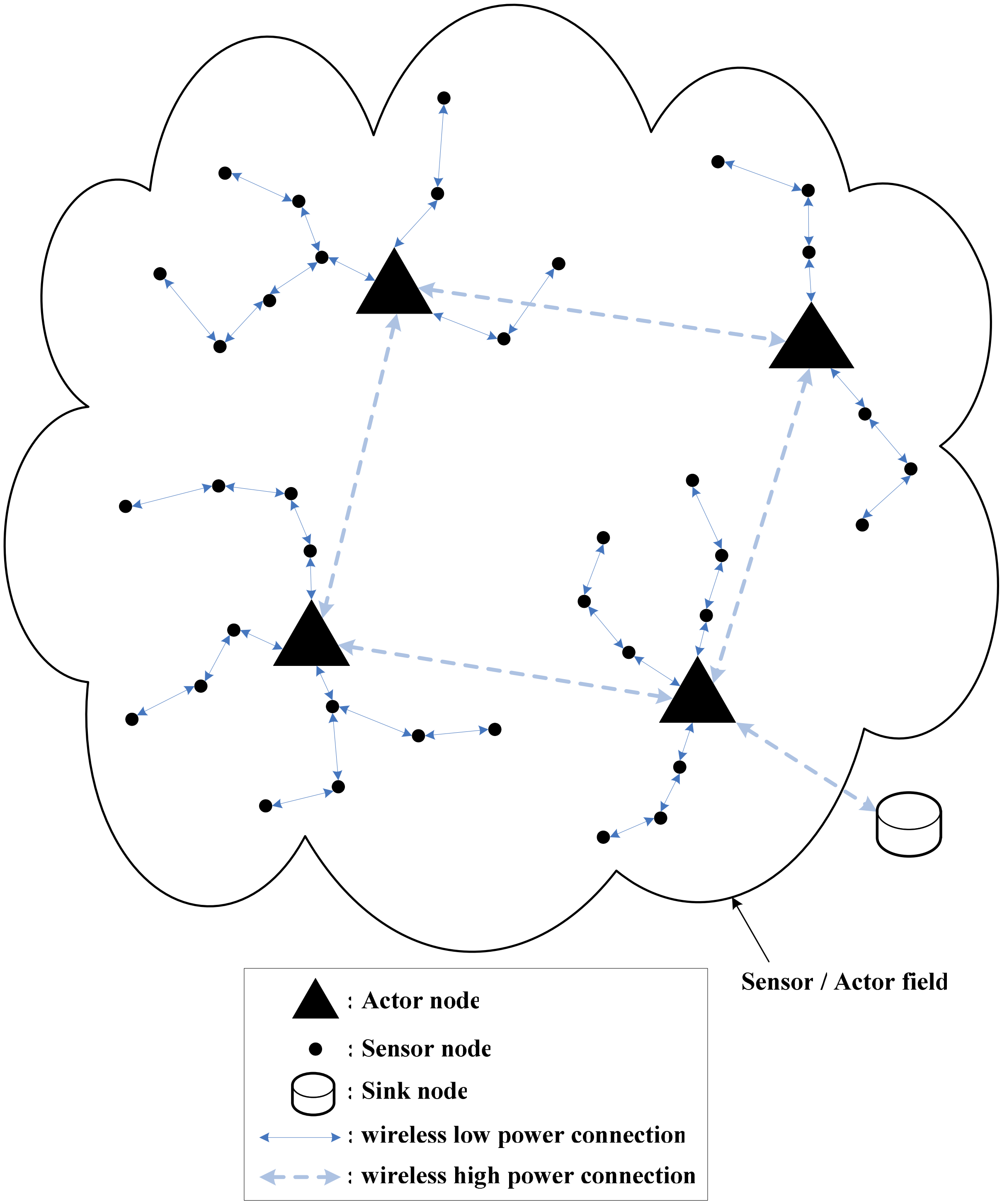

Figure 2.1 The physical architecture of a WSAN.

The distribution pattern and density of sensor nodes inside the target field depends entirely on the application. The number of sensor nodes deployed in a sensor and actor field may be in the order of tens or hundreds where such a dense deployment is usually not necessary for sophisticated actor nodes, since actors have higher capabilities and can act on large areas.

As illustrated in Figure 2.1, the sensor nodes in a specific region connect to the nearest actor node either directly or via other sensor nodes. The sensor node forwards the sensed data towards the actor node in a hop-by-hop manner. On reception of sensed data, the actor node in turn determine if any necessary action is to be taken and if necessary collaborates and coordinates with other actor nodes via the wireless high power connection, to act on a target area. An actor node may interface to an external sink solely to store sensed and network data. The sink in turn may allow read access of the data to external parties interested in viewing network activity and event logs.

Decision taking and control is done by the actor nodes in the network and hence no central controller is required. The non-existence of a central controller classifies the architecture as an automated architecture.

2.2 FACTORS INFLUENCING THE WSAN DESIGN

In the wide variety of applications for which WSANs can be implemented, common important design factors exist that ensure the overall feasibility of WSANs. Various common requirements and challenges that are presented by WSANs are discussed below.

2.2.1 Network Lifetime

A very important requirement posed by applications is for the WSAN to remain functional for as long as possible. The wireless sensor node, being a micro-electronic device, can only be equipped with a limited power source. In most application scenarios, replenishment of power resources might be very impractical or even impossible [8]. Network lifetime, therefore, depends strongly on the sensor nodes ability to use its power supply sparsely and remain functional as long as possible. If sensor nodes fail, network partitioning may occur and cause the network to fail its purpose.

2.2.2 Real-time requirement

A characteristic pertaining to WSAN applications is the presence of critical delay-sensitive data in the network at some point of time. With the occurrence of critical events for example a fire breakout, the sensed data must still be valid at the time of acting sprinklers. Sensor data arriving late at an actor node leads to late response and subsequent futile acting. Responding late to a fire for example, may have caused the fire to spread uncontrollably. The time it takes to send sensor measurements towards an actuator through the network depends on many network characteristics such as network topology, routing scheme, etc [9]. Many WSAN applications require high delay sensitivity because late critical data may have catastrophic consequences. Therefore, the communication protocols used in WSANs must meet stringent real-time requirements that imply minimum latency.

2.2.3 Reliability

The reliability of the WSAN determines the effectiveness of such network. This objective necessitates efficient and application specific communication protocols to assure the reliable communication of the sensed event features and hence enable the required actions to be taken by the actor nodes in the smart environment [4]. The network should guarantee high reliability by maintaining adequate connectivity of network nodes. The network should be able to function reliably without any assistance from outside sources or controlling entities, by staying true to the automated architecture, which signifies the non-existence of a central controller.

2.2.4 Node Heterogeneity

The node heterogeneity of WSANs distinguishes it from other types of networks [10]. New challenges are posed by the coexistence of two different types of nodes, namely the sensor and actor nodes. Sensor nodes are very resource constraint and have sensing capabilities for sensing the physical environment. The number of sensor nodes in WSAN is much greater that the number of actor nodes. Actor nodes have actuation capabilities and are involved in performing actions upon the physical environment. Actor nodes typically are less resource constraint and have a large acting region. Together these two node types give the network its unique characteristics.

2.2.5 Hardware constraints

Mainly sensor nodes are subjected to harsh hardware constraints. Due to the sometimes high volumetric densities of sensor nodes in WSANs, the production cost of such devices must be kept to minimum to justify the overall cost of the network [8]. Moreover, WSAN applications generally require disposable small sized sensor nodes that maintain an autonomous behaviour and are able to operate unattended.

Sensor nodes are produced with very limited hardware resources to satisfy the size and production cost requirements while still sustaining sufficient operability. Therefore, sensor nodes have limited processing capabilities, a small amount of memory, a short range transceiver and a *very limited energy source*. The energy source determines the lifetime of the node. The lifetime has to be maximized for the sensor nodes, which usually have no chance of energy replenishment, to remain functional for as long as possible.

2.2.6 Coordination and Communication

Coordination and communication challenges that are posed by WSANs are largely attributed to the coexistence of sensor and actor nodes. As previously mentioned, in an automated architecture no central controlling entity exists in the network. The controlling in WSANs is done by the actor nodes, which have to perform the functions of data collection and coordination. In WSANs, new networking phenomena called sensor-actor and actor-actor coordination occurs. In particular, sensor-actor coordination provides the transmission of event features from sensors to actors. After receiving event information, actors need to coordinate with each other in order to make decisions on the most appropriate way to perform the necessary action [2].

The effective communication of data inside the WSANs is vital for overall success and feasibility. This implies that the optimization of the efficiency of communication protocols is very important, since the network efficiency is based on the effectiveness of the communication protocols and how well the protocols incorporate requirements and cater for network constraints.

2.2.7 Mobility

Some applications require WSANs to cater for the possibility of mobile actor nodes. This means that while static sensor nodes gather information about the environment, actor nodes may move to specific target areas to perform necessary action taking. The mobility of nodes inside a network introduces complex connectivity issues and hence mostly limited mobility of actor nodes is considered.

2.2.8 Topology

The network topology of WSANs may change from time to time as new sensor nodes are added or energy depleted nodes 'die'. This implies a quite dynamic nature of WSAN topologies. The network should be able to react to this dynamic nature and maintain connectivity for as long as possible.

2.2.9 Scalability

The number of sensor and actor nodes present in a network depends entirely on the application. Some applications may involve very expensive actor nodes and consequently the number of actor nodes in the network will be kept to a minimum. In another application that may aim to monitor and control any fire outbreaks in a building for instance, an actor node in every sizable room together with several smoke sensor nodes may be required, resulting in a network with hundreds of actor nodes and thousand of sensor nodes. WSAN protocols should scale well to different application's size demands.

2.3 THE SENSOR NODE

The main tasks for sensor nodes in WSANs includes information gathering, by observing or sensing the physical environment, and relaying sensed information to the actor nodes. The architecture and characteristics of sensor nodes will be discussed in this section.

2.3.1 Architecture

The basic architecture with the main components that make up a sensor node is illustrated in Figure 2.1.

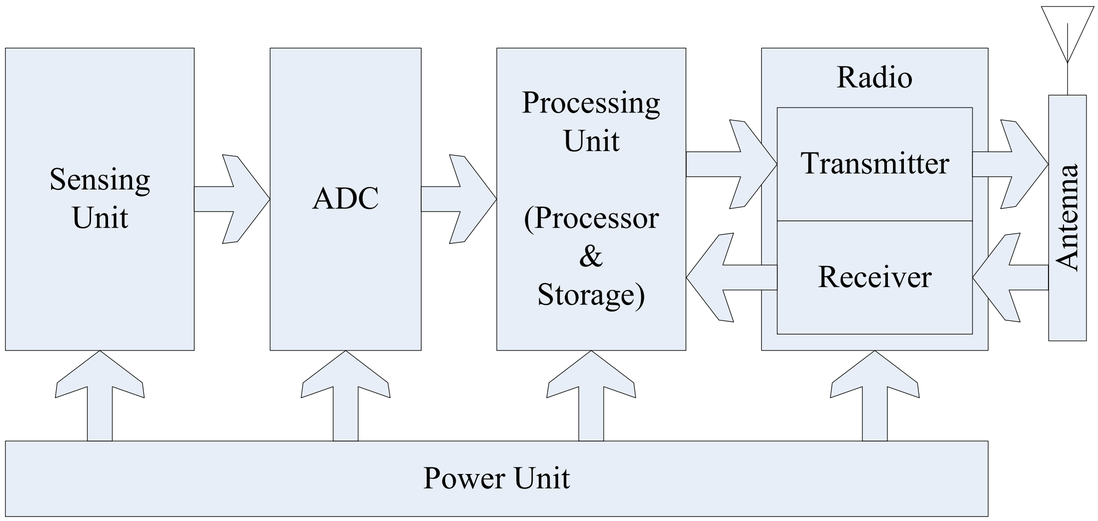

Figure 2.2 The basic components of a sensor node in WSANs.

The sensor node contains five main components: the power unit, a radio transceiver, a processing unit, an Analog to Digital Converter (ADC) and a sensing unit. The power unit supplies power to all sensor node components. Depending on the application, the sensing unit can contain one or more sensors that can sense ambient environmental conditions like temperature, humidity, movement, pressure or sound. After the sensing unit has attained some analog data from the sensor, this data is forwarded to the ADC, which serves as an interpreter by converting the analog sensed data to digital data that can be processed by the processor in the processing unit. The sensor node's processing unit contains all coordination and communication protocols and algorithms. After constructing a data packet containing the sensed digital data, the processing unit forwards the data packet to the radio transmitter component. The radio is responsible for transmitting and receiving data over a wireless channel for communication with other sensor or actor nodes. After receiving data the radio receiver component forwards it to the processing unit for processing.

2.3.2 Characteristics

Together, the basic components form a fully functional node that can sense data, process it and send it to other nodes. However, due to the stringent WSANs requirements, sensor nodes are designed to exhibit certain hardware characteristics which are discussed as follows.

Amongst others, the requirements posed by WSANs applications for sensor nodes (as adopted from [8]) include:

- small in size,
- remain functional for as long as possible,
- consume as little power as possible,
- communicate wirelessly,
- able to operate in high volumetric densities,
- have low production cost and be dispensable,
- operate autonomously and unattended, and
- adaptive to the environment.

The most important components of a sensor node are the power unit, the processing unit and the radio. These components will be discussed further since they play important roles in satisfying the stringent WSANs requirements.

The *power unit* is responsible for supplying power to all the other components of the sensor node. Power consumption can be divided into three domains: sensing, data processing and communication [8]. The aim for sensor nodes is to stay functional for as long as possible to ensure the feasibility of WSANs. The main design factors for the power unit include supplying power to the node for as long as possible, remain small in size and have low production cost. An example of a sensor node is the popular MICA2 Mote from Crossbow shown in Figure 2.3. The MICA2 Mote has a power unit in the form of two AA sized batteries that should keep the node functional for more than a year when using sleep modes [11].

Figure 2.3 The MICA2 Mote developed by Crossbow Technology Inc. [11].

The *processing unit*, which consists of a processor and some storage, is the sensor node's intelligent component which contains all coordination and communication protocols and algorithms. A CPS that governs all aspects of communication and coordination of a sensor node inside the WSAN is implemented on the processor. Specifications and challenges for the CPS are discussed in detail in section 3. The processor used on sensor nodes are typically microprocessors for example the Atmel's ATmega128L 8-bit microcontroller with 128 Kbytes memory used in the MICA2 Mote from Crossbow [11] or an 8-bit ARM processor used by the Intel Mote shown in Figure 2.4, which was developed by Intel Research Berkeley [12]. The 3cm x 3cm physical size of the Intel Mote makes it one of the smallest sensor nodes available.

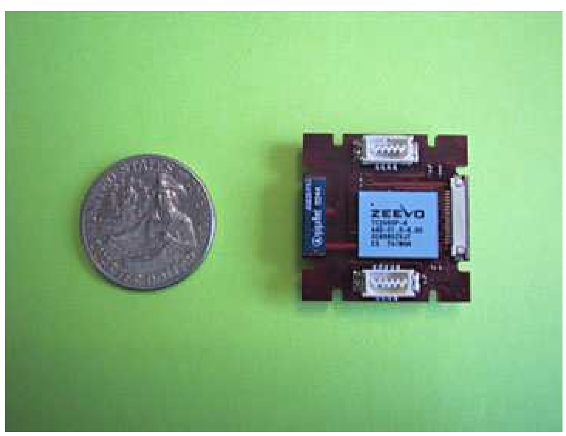

Figure 2.4 The Next-Generation Intel Mote developed by Intel Research Berkeley [12].

The *radio* is used to communicate with other nodes in the network. For WSANs, radio frequency (RF) transmissions are used to establish short-range wireless links between nodes by making use of the industrial, scientific and medical (ISM) bands, which offer license-free communication in most countries. WSANs sensor nodes are required to use small-sized, low-cost and ultra low power transceivers. Certain hardware constraints and the trade-off between antenna efficiency and power consumption limit the choice of a carrier frequency for such transceivers to the ultrahigh frequency range [8]. Popular radios make use of bands at 433 MHz, 868 MHz, 915 MHz and 2.4 GHz. Transmitting at higher frequency bands requires more power for the same distance but allows higher bandwidth. The higher bandwidth permits shorter transmitting time and smaller antennas. Several other radio specifications like turn-on time and transmit-receive switching time also need to be considered when searching for a suitable radio for an applications specific WSAN sensor node.

The importance of choosing a suitable radio for power constrained sensor nodes is marked by the fact that the radio is the primary power consumer. The radio can be either in transmit, receive, idle or sleep mode, with the sleep mode being the least power consuming. The less time the radio is busy with transmitting, receiving and idling, the longer the node's lifetime. Therefore, the effective controlling of the radio plays a great part in ensuring the feasibility of sensor nodes in WSANs. An example of low power radio used for sensor nodes is Chipcon's CC2400, which can transmit at bit rates of 10 kbps, 250 kbps or 1 Mbps and uses the 2.4 GHz band [13].

Dependent on the application, a location finding system or algorithm may require additional hardware to be fitted to the node. A small, low power global positioning system (GPS) unit for example, may assist a sensor node to relay its position together with the sensed data towards the actor node. If a GPS unit cannot operate satisfactory due to specific environmental circumstances, a low powered radio, for example the CC2400 from Chipcon, which features a Received Signal Strength Indicator (RSSI), may be used to assist the node in determining its relative position to an actor node [13].

2.4 THE ACTOR NODE

The main tasks of an actor node, also referred to as an actor node, are to collect and process sensor data from sensor nodes and then perform appropriate actions. In WSANs, if necessary, actor nodes collaborate and coordinate with other actor nodes to decide on and perform suitable action taking to achieve the overall application objective. Sometimes an actor node may interface to an external sink to store sensed and network data and allow read access to external parties interested in viewing network activity and event logs. Compared to sensor nodes, actor nodes in WSANs are resource richer with superior processing capabilities, higher transmission powers and a longer battery life [2]. The basic architectural components and characteristics of actor nodes will be discussed in this section.

2.4.1 Architecture

The basic architecture, which includes the main components of an actor node, is illustrated in Figure 2.5.

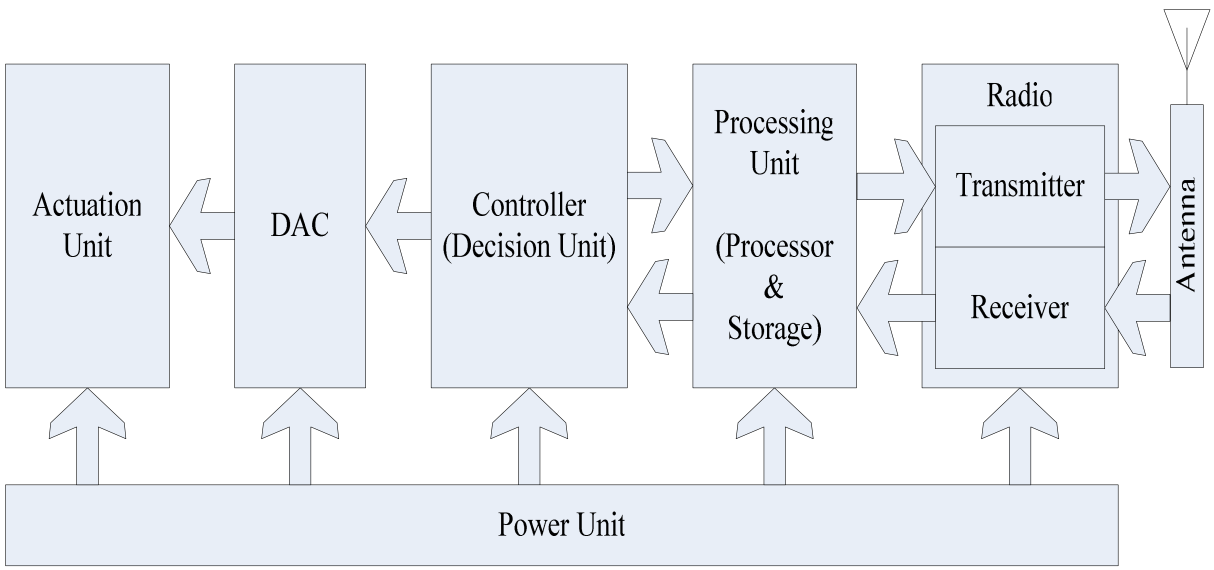

Figure 2.5 The basic components of an actor node in WSANs.

Similar to the sensor nodes, the power unit supplies power to all other components of the actor node. Data is received via the radio receiver component and sent to the processing unit for analysis. The processing unit contains the coordination and communication protocols and algorithms responsible for all operational aspects of the node inside the WSAN. Once the received data is processed, appropriate parameters are then forwarded to the controller. The controller is a decision unit which functions as an entity that takes sensor readings or parameters as input and generates action commands as output. These action commands are then converted to analog signals by the Digital to Analog Converter (DAC) and are transformed into actions via the actuation unit as shown in Figure 2.5. The function of the actuation unit is to change or influence the physical environment of a specific target area. Regular feedback from the sensor nodes located in the specific target area is necessary for the controller to incur effective acting via the actuating unit. Any collaborative efforts with other actor nodes concerning the acting process is decided on by the controller, processed by the processing unit and then sent via the transmitter component of the radio.

In some applications, integrated sensor/actor nodes may replace actor nodes in a WSAN [2]. Since an integrated sensor/actor node is capable of both sensing and acting, it has a sensing unit and an ADC in addition to all components of an actor node shown in Figure 2.5.

2.4.2 Characteristics

The wide range of applications gives rise to unlimited number of different variations of actor nodes. Compared to sensor nodes, actor nodes in WSANs are generically characterized by being resource richer with superior processing capabilities, higher transmission powers and a longer battery life [2]. Essentially, in WSANs, the actor node has to collect sensed data from sensor nodes and then perform necessary actions by either acting alone or collaboratively acting with other actors.

Some applications may require small, cost effective, power efficient actor nodes to satisfy the overall cost of the network. Others have less strict requirements due to the fact that the number of deployed actor nodes is generally much fewer than sensor nodes because they have a greater communication and acting ranges. The smaller number of actor nodes in a WSAN, prompts most applications to focus more on the ability of actor nodes to conduct effective acting, resulting in relaxed size, cost and energy constraints. Hence, the selection of suitable power unit, radio and processor for the actor nodes rests on the application-specific requirements. Some general guidelines for these components are as follows.

The *power unit* of an actor node determines the lifetime of the node. The type of power unit selected for a specific actor node depends on the required lifetime imposed by the application. The node should remain functional until replenishment is available. A particular power unit is mostly required for actor nodes that are either mobile or remotely deployed. Power unit for these nodes typically contain batteries or solar panels. Other immobile actor nodes, which are situated on air conditioners or heaters for example, may connect to the electricity grid and do not require a separate power unit. These nodes are indicated as resource unconstrained

The *processing and decision unit* can be handled by a single microcontroller. Depending on the application, typically actor nodes may employ more powerful Digital Signal Processors (DSP) to conform to the higher processing capability demands.

The *radio* that is used by actor nodes uses RF transmissions to communicate wirelessly with close proximity sensor nodes and other actor nodes which are usually situated further away. The radio should be similar to the sensor node radio (see section 2.3.2) which operates in the ISM band but should allow output power programming to cater for long and short range communication.

As with sensor nodes, actor nodes contain some kind of location finding system. A small, low power global positioning system (GPS) unit for example, would provide a crucial coordination reference in ensuring an effective and successful acting process.

2.5 APPLICATION EXAMPLES

Having gained some understanding of the architecture and characteristics of WSANs and its underlying entities, examples of some real applications are discussed below to give insight into the application areas of WSANs.

WSANs can be implemented in a variety of different application domains that include commercial, industrial, scientific, agricultural and military applications [14]. WSANs can be an integral part of systems such as battlefield surveillance and microclimate control in buildings, nuclear, biological and chemical attack detection, target tracking, home automation, environmental monitoring [2] and sensing and maintenance in large industrial plants [14]. Furthermore, WSANs can be used for agricultural maintenance, precision farming and automatic irrigation [3].

Some example applications are as follows.

- *Fire control*: Fires can have a detrimental effect on nature reserves, farms, and buildings. The damage incurred by a fire outbreak depends mostly on the time it took to respond to the outbreak. A timely response to fire outbreaks is the most crucial requirement of fire control. When not contained in the shortest time possible, fires can spread at horrific rates when non-favourable conditions prevail. WSANs can be implemented to detect fires and take immediate action to prevent it from spreading. A tree plantation, shown in Figure 2.6, would be a deployment area for hundreds of sensor nodes (yellow octagons) scattered throughout the region to measuring temperature and moisture for example. The actor nodes (blue triangles) analyze the received sensor data and respond to critical flagged information. When the static sprinkler actor nodes are unable to reach a target area, collaborate efforts would engage mobile actor nodes equipped with a water tank and cannon to respond to fire outbreaks by moving towards the target area.

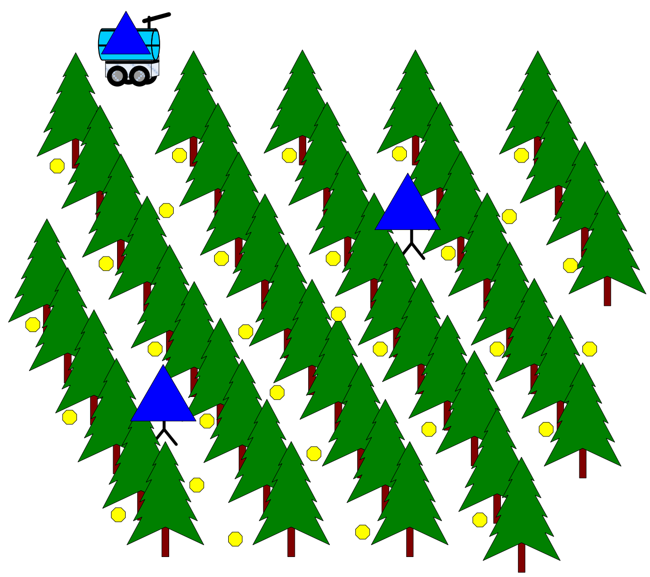

Figure 2.6 A tree plantation with a WSAN for fire control.

- *Target tracking and attack detection*: In many real applications, integrated sensor/actor nodes, especially robots, are used instead of actor nodes [2]. Sensor nodes are distributed on a battlefield to sense movement and sound for example. They relay monitoring and critical event information to robots, which are actors that carry sensors in addition to the primary actor components. An example robot, shown in Figure 2.7, is the mobile Robotic Mule which is an autonomous battlefield robot designed for the Army. These types of developed battlefield robots can detect and mark mines, carry weapons, function as tanks or maybe in the future totally replace soldiers in the battlefield [2].

Figure 2.7 A Robotic Mule used in the battlefield [2].

When actively tracking a target, the WSAN provides real-time data to the actor nodes that may want to engage in target annihilation or simply video surveillance. Similarly for attack detection, sensor nodes relay critical event data, which can include the location, severity and extent of an attack, to the actor nodes in the WSAN for analysis. The actor nodes then take decisions concerning damage control and alerting relevant parties.

- *Microclimate and lighting control in buildings*: Modern buildings typically feature highly sophisticated heating, ventilation and air conditioning systems that enable extensive monitoring and control over the specific settings [1]. For microclimate control, a WSAN would be employed with sensor nodes that monitor the temperature, humidity, etc. in various parts of the building and actor nodes that control the cooling and heating devices such as air conditioning vents and furnaces or floor heating systems respectively. The microclimate would be controlled to achieve user specified preferences in an open floor plan office, illustrated in Figure 2.8, for example.

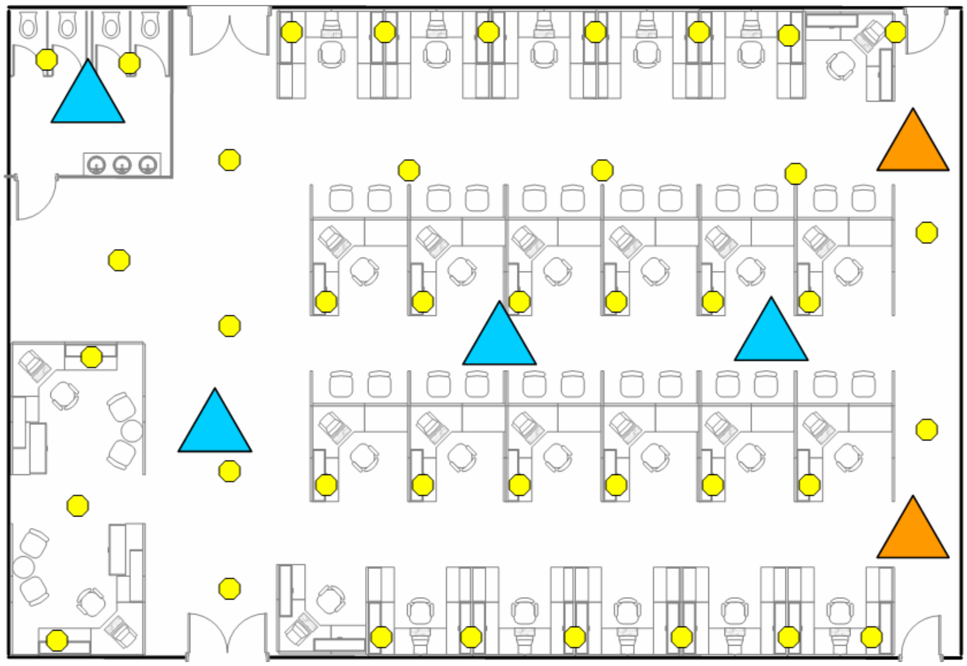

Figure 2.8 An open floor plan office with a WSAN for microclimate control.

In Figure 2.8, sensor nodes (yellow octagons) are scattered over an open floor plan office and connect wirelessly (not shown) to the nearest actor node (blue and red triangles). The blue actor nodes are air conditioners for cooling, while the red actor nodes represent floor heating control stations.

In addition to controlling the climate in buildings, WSANs can be used to monitor occupancy in the building and adjust heating, cooling and lighting settings to the needs of the current occupants [1]. For example, parts of a building with no occupants would not need light or heating and only an active ventilation system would suffice. Considering the ever increasing cost of energy, employment of such monitoring and control by WSANs could result in building’s operation expenses being significantly reduced.

- *Home automation*: A modern home contains a large number of appliances and equipment that can be both remotely monitored and controlled. Besides the heating and cooling system, such systems include lights, digital media equipment, and major appliances like refrigerators and ovens. The possibilities for automation using sensing devices are numerous, for example, a user might express a preference such as "keep the floor clean", and "do not run the noisy robotic vacuum cleaner when I am listening to music". This would involve a simple control system that uses sensor nodes that detect dirt and the presence of a user, and an actor node that actuates an automatic vacuum cleaner as needed [1].

- *Planet exploration*: The WSAN can be applied to aerospace industries as well. For example, a number of sensor nodes and actor nodes can be deployed on a planet for exploration. The sensor nodes collect data on the planet and report interesting data to the actor nodes. Then, some mobile actor nodes can go to particular locations identified by the multitude of sensor nodes, for more detailed observations. They may collect some stone samples for bringing back to the space ship, capture high-resolution pictures, or record videos for deeper investigations [14].

- *Structural damage detection*: Systems that detect and locate damage in large structures such as buildings, ships, bridges and aircraft can improve safety and reduce maintenance costs. Structural health monitoring techniques rely on measuring structural response to ambient vibrations or forced excitation. Ambient vibrations can be caused by earthquakes, wind, or passing vehicles, and forced vibrations can be delivered by hydraulic or piezoelectric shakers. These techniques infer the existence and location of damage by detecting differences in local or global structural characteristics before and after damage. Structural health monitoring is an obvious application for wireless sensor actuator networks. Unconstraint sensor nodes can reduce cabling costs and increase the flexibility of instrumentation placement [15].

- *Precision farming*: In the quest to make agricultural processes more automated and efficient, WSANs concentrate on providing the means for observing, assessing and controlling agricultural practices. A facet of precision agriculture concentrates on sitespecific crop management. This encompasses different aspects, such as monitoring soil, crop and climate in a field; providing a decision support system for delivering insight into possible treatments, field-wide or for specific parts of a field; and the means for taking differential action, for example, varying in real-time an operation such as fertilizer, lime and pesticide application, tillage, or sowing rate [17]. Suitable sensor nodes can be implemented to measure air temperature, relative humidity, light, soil moisture, and soil temperature. Actor nodes take suitable actions that can include turning on a fan, a light, a heating element or a sprinkler. These actor nodes are positioned so that they can influence measurements taken by the sensor nodes [16].

2.6 CHAPTER SUMMARY

This chapter served as a general overview of what the recently developed Wireless Sensor Actor Networks concept involves. A typical physical architecture was illustrated and significant factors that relate to the design of WSANs were discussed. The architectural structure, operation and characteristics of sensor and actor nodes were given and compared, to give insight into the network entities that make up a WSAN. Finally some application examples were listed and show the versatility of WSANs.

The discussed WSAN requirements, challenges and characteristics have to be considered extensively when designing an efficient communication and coordination protocol stack, which is the central subject of this dissertation. The CPS design needs to be efficient towards energy, latency and reliability for the overall network to be feasible.

Chapter

3 COMMUNICATION ARCHITECTURE AND PROTOCOLS

Having gained an overview of WSAN requirements, challenges and characteristics in chapter 2, the underlying communication and coordination protocol stack is discussed in this chapter. This is the primary subject of interest of this dissertation. The protocol stack is responsible for regulating and controlling all operations within the WSAN. The effective communication and coordination of data inside the WSANs is vital for overall application success and feasibility. Therefore, the protocol stack has to incorporate WSAN requirements and cater for network constraints in order to achieve a stable communication and coordination environment. In WSANs four types of coordination exist: sensor-to-sensor (to use multi-hop communication mode to relay sensing data), actuator-to-sensor (downlink transmission to instruct sensors to execute or change a certain sensing tasks), sensor-to-actuator (uplink transmission to report new events) and actuator-to-actuator (in determining which actuators should respond to which sensing area) [22].

In this chapter, the WSAN communication architecture is given, followed by a layer-by-layer investigation into the operational tasks and challenges of every CPS layer. Furthermore, existing literature of WSAN protocols are given and for some layers, relevant existing literature of WSN protocols are examined and evaluated.

3.1 COMMUNICATION ARCHITECTURE FOR WSANS

3.1.1 The OSI Basic Reference Model

The communication architecture used for WSANs is based on the Open Systems Interconnection (OSI) reference model, which is the basis for most networks' communication design. In 1977, the International Organization for Standardization (ISO) began to develop its OSI networking suite. OSI has two major components: an abstract model of networking (the Basic Reference Model, or seven-layer model), and a set of concrete protocols [18]. In the abstract model, a networking system is divided into layers. Within each layer, one or more entities implement its functionality. Protocols enable an entity in one host to interact with a corresponding entity at the same layer in a remote host.

Layering is a structuring technique which permits the network of Open Systems to be viewed as logically composed of a succession of layers, each wrapping the lower layers and isolating them from the higher layers [18]. Traditionally, the idea of layering is that each layer adds value to services provided by the set of lower layers in such way that the highest layer is offered the set of services needed to run distributed applications. Layering thus divides the total problem into smaller pieces. Another basic principle of layering is to ensure independence of each layer by defining services provided by a layer to the next higher layer, independent of how these services are performed. This permits changes to be made in the way a layer or a set of layers operate, provided they still offer the same service to the next higher layer.

In effect this architecture enables a plug-and-play approach for single layer protocols. The advantage of such architecture is that the design and development of single layer protocols can be conducted independently and afterwards simply inserted into the corresponding layer position. For this reason, current literatures deal with the research and development of specific single layer protocols. To arrive at a complete CPS for a network, relevant single layer protocols are implemented and combined focus to achieve overall network specific objectives.

The OSI reference model consists of 7 functional specific layers that include: application, presentation, session, transport, network, data link and physical layer [18].

3.1.2 The Communication Protocol Stack for WSANs

The main aim of the WSAN CPS is to provide the collection of network nodes with a means of communicating and coordinating in such a way as to satisfy the overall WSAN objectives. The protocol stack should take all WSAN constraints into account and ensure a stable communication environment. Protocols inside the CPS need to be very efficient with regards to energy, latency and reliability to ensure the feasibility of WSANs.

There exists no standardized CPS for WSANs to date. However, with respect to the OSI reference model, relevant layers for WSANs are: *application*, *transport*, *network*, *data link* and *physical* layer [2]. Due to the network characteristics and node constraints, layers 5 and 6 of the OSI reference model are not relevant for WSANs. The transport layer (layer 4) is also less relevant since the reliability and real-time requirements which have been proposed to be handled by the transport layer by Akyldiz et al. [2] should generally be integrated by all layers of the protocol stack which consequently renders the transport layer superfluous. Furthermore, the data link layer of the WSANs protocol stack mostly consists of a MAC protocol only and is therefore referred to as the MACL.

The basic layers of the CPS for WSANs are illustrated in Figure 3.1.

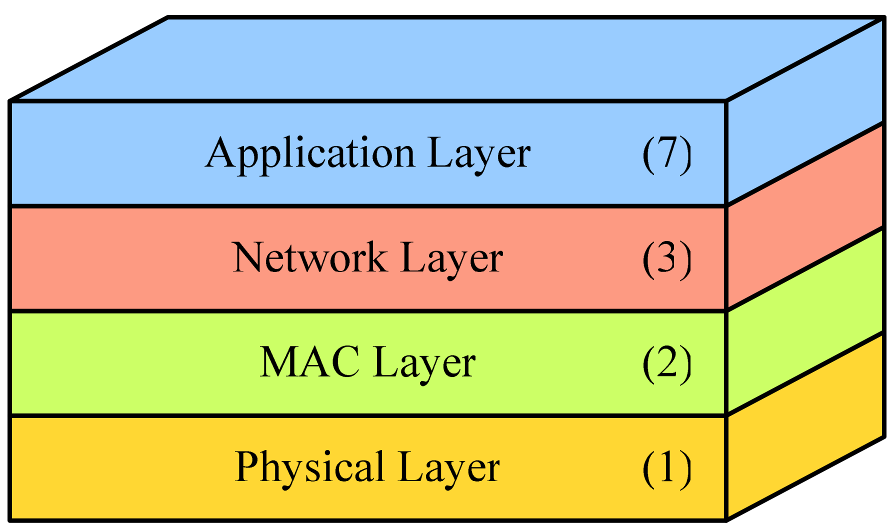

Figure 3.1 The Communication Protocol Stack layers for WSANs.

Each layer of the protocol stack has its own tasks while all layers combined strive to achieve the overall WSAN objectives. In short, the construction of the physical channel is done by the PHYL, the access of the node into the medium is done by the MACL, the selection of the routing path through which the node transmits its data is done by the NETL and finally the handling of the actual sensed data is done by the APPL.

This basic protocol stack architecture is implemented on every node inside the WSAN. However, the content of some layers differs between actor and sensor nodes. The sensor nodes inside the network have to deal with *sensor-sensor* and *sensor-actor* communication and coordination. The actor nodes on the other hand, are concerned with *actor-sensor* and *actor-actor* communication and coordination. Actor nodes encompass additional algorithms necessary for actor-actor coordination, needed to perform effective acting on a target area. In addition to the difference in communication and coordination responsibilities between actor and sensor nodes, the actor nodes are less resource constraint nodes and thus have a less strict policy for the use of their energy resource [2]. The actor nodes focus on the real-time requirement and methods for effective acting which might involve collaboration between several actor nodes in the WSAN.

As mentioned, every layer has specific tasks and plays a different role inside the protocol stack. In the next sections, the 4 basic layers of the WSAN CPS are discussed in detail. Tasks and challenges are examined to acquire an in depth understanding of the responsibilities of a specific layer. Although *only* the sensor node CPS is extensively considered in this dissertation, tasks and challenges of the actor node's protocol layers are given where relevant, to ensure a complete understanding of WSANs and its CPS requirements. An investigation into existing WSAN protocol literature is given and where relevant, additional existing WSN protocol literature is discussed.

3.2 THE APPLICATION LAYER

The APPL is the highest layer of the WSAN CPS that does not provide services to any other layer. The primary concern of the APPL is with the semantics of the application [19]. All application processes reside in the APPL. This layer is very application dependent and the protocol design should closely consider the requirements and objectives defined by a specific application. The main tasks for sensor and actor nodes are investigated and then existing literature that relates to the APPL protocol is discussed.

3.2.1 Primary Tasks and Design Challenges

The APPL protocol tasks for sensor nodes and actor nodes differ in some aspects since the actor nodes involve actuation processes while the sensor nodes entail sensing processes. Due to the fact that the APPL protocol design largely depends on the specific application, a generic view of the primary tasks and design challenges of the APPL for sensor and actor nodes is considered (some are adopted from [2] and [8]).

- *Construction of application layer specific data frame*. The payload of the APPL frame may contain the actual sensed measurement, time, location, severity and extent of a sensed event. Once the payload is ready, the frame is sent to the next layer in the CPS.

- *Providing data security*, by means of encryption or authentication techniques. Some application may deal with very confidential data and require solid security. However, often applications dismiss this requirement due to the added complexity and overhead burden. The sensor nodes have limited memory resources which have to be shared by all protocol layers.

- *Localization or positioning information* may be necessary at sensor and actor nodes to assist coordination schemes. Sensor nodes are randomly deployed into an unplanned infrastructure. The problem of estimating spatial-coordinates of the node is referred to as localization [20]. When an event occurs, the location of such an event is important for the coordination of a suitable response by the actor nodes. This information may also be used by the NETL protocols that base their routing on the geographic position of nodes.

- *Provide adequate time synchronization* among different sensors reporting the same event to multiple or same actor node in order to facilitate a one-time response in the entire region. This requirement depends on the application, since some applications have distributed sensor nodes that consider collaboration less important and unnecessary overhead inducing. However, some form of synchronization is necessary among nodes in the WSAN to ensure that new sensed data is regarded as new by all nodes. Moreover, many network and MACL protocols require synchronization for successful operation.

- *Support a distributed actuation approach* to ensure timely action performing within the WSAN.

In addition to the above two tasks, the ***sensor* nodes** in WSANs have the following tasks.

- *Obtainment and evaluation of sensed data*. The application protocol requests the sensing unit of a sensor node to provide it with a sensor reading or measurement. The actual sensed data is obtained from the sensing unit via the ADC, which converts it to digital format. The sensed digital data is then processed and evaluated. Immediate evaluation of sensed data is important for applications that require data to be rated according to threshold levels and prioritized. The frequency of obtaining sensed data is totally dependent on the application. Most applications require periodic data acquiring, which is necessary when monitoring the environment for irregularities.

- *Ensure ordering* between the different events when they are reported to the actor nodes. Keeping track of the time of event occurrences is necessary to ensure sequential reported of sensed data.

- *Data aggregation* may be considered at the APPL, since sensor nodes may generate significant redundant data, similar packets from multiple nodes can be aggregated so that the number of transmissions is reduced. Sensor readings are combined from different sources according to a certain aggregation function, for example duplicate suppression, minima, maxima and average. This technique may be used to achieve energy efficiency [20] and alleviate congestion. The use of aggregation functions depends on the application since some applications require very accurate measurements to be recorded. Accuracy can be jeopardized by taking an average of several values. Furthermore, some kind of aggregation may alleviate traffic congestion in large networks.

- *Data prioritization* may be considered to ensure reports of different event types to reach the destination within their latency bound. By giving preference to important events information, less important information may temporarily be delayed to allow delay sensitive information to reach its destination sooner. Some applications may require data to be branded with different delay bounds to distinguish between different levels of importance and criticalities of events.

In addition to the first two tasks, the ***actor*** **nodes** in WSANs have the following tasks.

- *Processing of sensor data*, which was received through the network. The sensor data is stored and examined to determine if any acting is necessary. Some applications may require the sensed data to be stored for logging purposes and later sent to a sink, which enables external parties to view the data and event history.

- *Coordination and collaboration towards effective acting*. If it is established by an actor node, that the received sensed data calls for an action to be performed, the actor node has to consider the time, location and severity of the event to determine which kind of coordination and collaboration is necessary to ensure effective acting. The following considerations need to be taken into account.

  • One actor node may not be enough to perform the required action, thus, other nearby actors should be triggered.

  • If multiple actor nodes receive the same event information and there is an action threshold, these actor nodes should communicate with each other in order to decide which one of them should perform the action.

  • In certain applications, if multiple actor nodes are required to cover the entire event region, it may be necessary to ensure that these regions are non-overlapping or mutually exclusive in order to ensure uniform acting behaviour over the entire region.

  • If multiple actor nodes receive information from multiple sensors for the same event, then it may be necessary to ensure that these multiple actors act on the environment at the same time. This synchronization requirement in the execution of the task is required in applications where a partial execution of the task alters the state of the event in the region where it has not been executed.

- *Initiation of acting* once all necessary coordination and sensor data processing has been completed. The application protocol has to initiate relevant acting by activating and controlling the actuation unit via the DAC.

Integrated sensor/actor nodes, which are mentioned in section 2.4.2, encompass all tasks pertaining to sensor and actor nodes.

3.2.2 Existing Literature on WSAN Application Layer Protocols

The APPL is the layer in the protocol stack that links the network and communication protocols with the actual application. The actual application essentially requires sensing and subsequent acting on the physical environment. The APPL protocol is not interested in the details of how the data is communicated, but invokes communication and deals with the effective action taking by actor nodes.

As mentioned in the APPL tasks, the *sensor* node APPL protocols entails creating, prioritizing and aggregating the sensed data. *Actor* node APPL protocols on the other hand, have to deal with collecting and processing sensor node data and performing appropriate actions. The challenge of performing effective acting is not a trivial one. However, advanced coordination and collaboration schemes for acting are beyond the scope of this dissertation. Some existing literature exists that investigate actor-actor coordination, actuation strategies and other APPL relevant functionality. What follows is a discussion of existing literature, to show the direction and progress of research conducted with regards to APPL relevant responsibilities.

3.2.2.1 A Distributed Coordination Framework

The distributed coordination framework proposed by Melodia et al. [21] investigates amongst other the actor-actor coordination that is required for WSANs. This would typically be handled by the APPL of the actor nodes. It is assumed that the coordination problem involves overlapping target areas as illustrated in Figure 3.2 and is formulated as a mixed integer non-linear program. The selection of a subset of the actor nodes and their action powers is considered to optimally divide the action workload, so as to maximize the residual energy of the actor nodes while respecting the action completion bound, in order to extend the lifetime of the actor nodes.

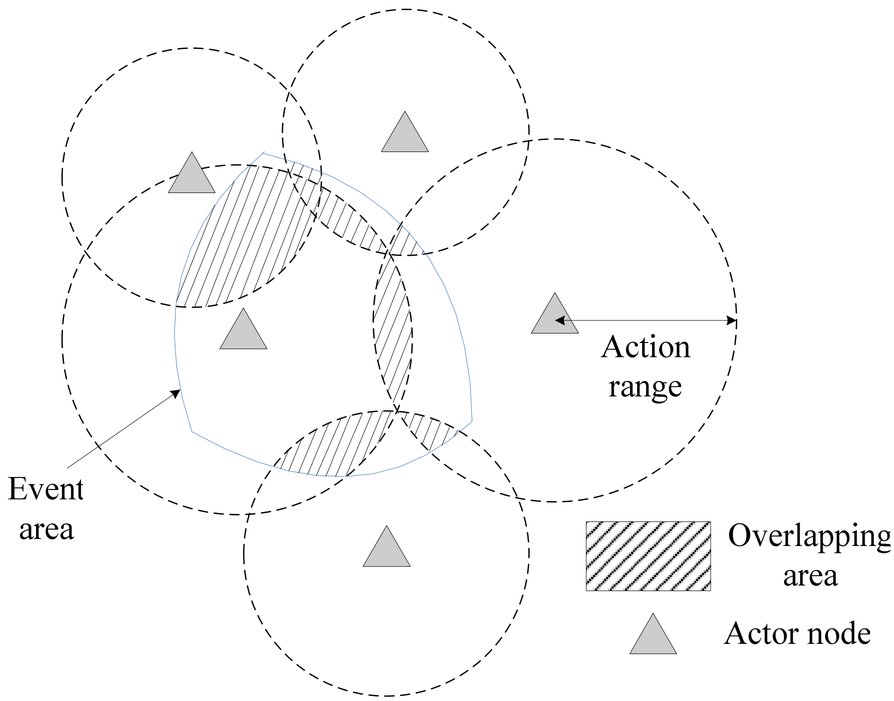

Figure 3.2 Actor nodes with action ranges and overlapping areas.

The proposed approach is based on a real-time auction protocol that describes the behaviour of actor nodes participating in transactions as buyers or sellers. The objective of the auction is to select the best set of actor nodes to perform the action on each overlapping area. Thus, overlapping areas are items that are traded by the actor nodes. The actor nodes can assume the roles of seller, auctioneer or buyer. The seller is the actor node responsible for a portion of the event area from which it received event information, the auctioneer is the actor node in charge of conducting the auction on a particular overlapping area and the buyer is the actor nodes that can act on a particular overlapping. A localized auction takes place in each overlapping area. The bid of each actor participating in the auction consists of a power level and of the corresponding action completion time, as well as the available energy of the actor. Multiple localized auctions take place in parallel under the responsibility of different auctioneers. This is preferred to one central auction since it causes lower signalling overhead and the auction process workload is shared among a higher number of actors. The auction mechanism can be classified as a single-round sealed-bid auction, where each buyer submits its bids in one shot irrespective of the bids from other buyers.

In this specific framework, the lifetime of actors is considered extensively, which might not be representative of various applications, since actor nodes are mostly regarded as energy unconstraint in comparison to sensor nodes. However, the single-round auction based approach presents a powerful coordination solution since overlapping of target areas is a common problem when performing acting tasks. Localized communication between actor nodes is favourable in WSANs to limit overhead and maintain a distributed control inside the network.

3.2.2.2 Grid-based Coordination Mechanism

In another coordination mechanism for WSANs by Yuon et al. [22], one or more actor nodes will be triggered, according to the characteristic of the current event, to perform one or more actions. To solve the action allocation problems, the actor-actor coordination mechanism is divided into two forms namely the action-first scheme and the decision-first scheme.

In the action-first scheme, when actor nodes around an event area receive event information, they perform actions immediately without negotiating with other actor nodes. During the processing procedure, each of these actor nodes transmits the action information to other actor nodes so they can learn the information about the event and processing procedure. Each actor node makes a decision to join or retreat the action independently. To control the number of actor nodes to join the action, a pre-specified action threshold is set. In the decision-first scheme, all actors receiving event information coordinate and collaborate strongly in order to maximize their overall task performance. In order to avoid action competition among the available actors, and achieve the goal of rational action, a central controller is needed. If an action needs to be performed, the action command is transmitted to the appropriate actor nodes by the central controller.

The acting area of each actor node is located among a location-based grid. The action workload is optimally divided among different actor nodes by the characteristics of the event. This estimation of the acting area tries to eliminate the problem of overlapping areas. The proposed coordination mechanism simplifies the problem without considering the consequences. When employing a central controller for decision-first scheme, the scalability of the network is compromised and a huge communication overhead is incurred. However, the action-first scheme provides a suitable solution for WSANs by promoting a distributed control.

3.2.2.3 Network-centric Actuation Control

A network-centric actuation control is proposed by Dressler [23], which is inspired by the information handling in cell biology. A rule-based system is proposed that allows all decision taking within the network itself with no external control required. Using simple rules that are pre-programmed into network nodes, the network becomes able to solve aggregation or other decision problems without having a global view to the behaviour of the entire system. Actuation is handled by self-organization methodologies that provide a network-centric actuation control by processing measurement data within the network and directly interacting with associated, co-located actor nodes. This provides accelerated response, i.e. in-time actuation by employing local behaviour control by means of a state machine. Schemes for actor node collaborations and specific rules for action taking are not mentioned. However, WSANs latency requirements are satisfied by suggesting local reactions to events.

3.2.2.4 Actuation strategies using neural networks

An actuation strategy that employs an approach similar to neural systems is proposed by Rozell and Johnson [24]. Each actor node determines its individual contribution to a behavioural goal through a combination of the sensor measurements. Inter-sensor and inter-actuator communication is purged to eliminate the communication overhead necessary for such communication. To generate the optimal total action in this actuation strategy model, actuator intensity coefficients are required. However, each coefficient is a sum of all sensor measurements, which implies that each individual actuator would require knowledge of every sensor measurement in order to generate optimal actuation intensity. A scenario where every sensor in the network communicates its measurement to every actuator would present an unreasonable communication burden on the network. To mitigate this problem each sensor-actuator link has an associated importance value that can be used to efficiently determine which communication links are least important to the final actuation fidelity.

By eliminating communications regarded as vital by other application algorithms, more problems arise and the complexity of the initial solution is unnecessarily increased and the feasibility of the protocol compromised.

3.2.2.5 A Service-Oriented Query Architecture

In the quest to overcome the disadvantages of application specific and generic algorithms, a service-orientated query architecture is proposed by Rezgui and Eltoweissy [25] as a novel programming and querying paradigm. It is stated that the generic solutions often battle with network inefficiency for the reason that they strive to stay generic for all types of applications. On the other hand, application-specific deployments are criticized of featuring too tight coupling between the application and the network. This often leads to considerable reprogramming efforts that are necessary to make the network able to serve new applications.

The proposed architecture features two layers which include the service layer and the service-oriented query layer. The service layer runs a collection of lightweight services, with each service being a software module that carries out some sensing, actuation, or control function. The service-oriented query layer comprises three modules. The first module is the Event Detection Module, which detects the events that are relevant to the current query load at the local node. The second module is the Service Scheduling Module, which schedules continuous queries according to their frequencies and expiration time. The third module is the Service Invocation Module, which receives requests for service invocation from the other two modules and then invokes the service and, if applicable, returns the results of the invocation to the query issuer. A query engine, which is situated outside the network, generates a query plan that is executed and spread throughout the network. It is not clear what this plan entails, but nodes respond to the query plan while optimization strategies are employed to handle multiple queries and further optimize the query plan.

This architecture is intended for networks that respond to queries rather that generating events based on the physical environment only. It introduces a modular service design that simplifies programming networks for new applications. However, issuing network-wide queries might not scale well to larger networks and introduce an unnecessary traffic load to the network at the cost of scarce energy. Actor-actor coordination for effective acting is not discussed in the design.

3.2.2.6 Adaptive Actuator Allocation, Aggregation and Prioritization Strategy

As part of a reliable reporting of delay-sensitive events framework, Ngai et al. [26] propose an algorithm that features adaptive actuator allocation. This algorithm aims at covering unevenly distributed events, by providing an allocation algorithm that balances the load of the actor nodes and minimizes the distance between significant sensor nodes and actor nodes. The total area is divided into virtual grids and the event occurrence frequency of every grid is considered to determine the placement of the actor node. Such an allocation can be performed in the initial stage based on pre-estimated frequencies, or, with mobile actor nodes, performed periodically to accommodate event dynamics.

Furthermore, the framework introduces data aggregation done by the APPL protocol. In a densely deployed sensor network, multiple sensor nodes may sense the same event with similar readings. Hence, it is preferably to aggregate them before reporting to the actor nodes. The aggregation algorithm is grid-based and for each grid, there is an aggregating node that first collects the event data. The node filters out significant different readings and finally calculates the mean value before sending it to an actor node or the next aggregation node.

Another relevant strategy for the APPL protocol is prioritization of data. The key design objective is to maximize the number of reports reaching the destination within their latency bound, and, for different event types, give preference to important events. A pre-emptive priority queue is adopted in each sensor, where the packets for the event data are placed according to their data importance, and each priority is served in a first-in-first-out discipline.

Data prioritization and aggregation are important tasks of the APPL protocol. However, actuator allocation that involves actor nodes frequently moving to different locations might cause a lot of routing disturbances and network reconfiguration burdens. In most applications, events occur randomly and locations cannot be predetermined which would require the allocation algorithm to constantly revise the position of actor nodes. Limiting the mobility of actor nodes might be the answer to maintain some form of network configuration and curb frequent significant actor nodes position changes.

3.2.2.7 Literature Summary

Table 3.2 contains a summary of the WSAN APPL relevant literature with main features and identified problems pertaining to each protocol.

Protocol	Main features	Problems
Distributed Coordination Framework	▪ A real-time auction protocol to select the best set of actor nodes to perform action on each overlapping area ▪ The bid consists of a power level, completion time and available energy of the actor	▪ Lifetime of actors is considered extensively, which is not representative of most WSANs
Grid-based Coordination Mechanism	▪ The coordination mechanism is divided into two forms: action-first scheme and decision-first scheme ▪ Acting area located among a location-based grid dividing the action workload ▪ Action-first scheme provides distributed control	▪ Vague estimation of acting area tries to eliminate the problem of overlapping areas ▪ Proposed coordination mechanism simplifies the problem without considering the consequence ▪ Central controller for decision-first scheme compromises scalability of the network and incurs a huge communication overhead
Network-Centric Actuation Control	▪ Rule-based system that allows decision taking within the network itself ▪ Actuation handled by self-organization methodologies that provide a network-centric actuation control	▪ Schemes for actor node collaborations and specific rules for action performing are not mentioned
Actuation Strategies using Neural Networks	▪ Inter-sensor and inter-actuator communication is purged to eliminate the communication overhead ▪ Actuator intensity	▪ Each actuator would require knowledge of every sensor measurement to generate optimal actuation intensity ▪ By eliminating

	coefficients used for optimal total action	communications, more problems arise and the complexity of the initial solution is unnecessarily increased
Service-Orientated Query Architecture	▪ Comprises two layers: service layer and service-oriented query layer ▪ Optimization strategies are employed to handle multiple queries and further optimize the query plan	▪ Architecture is intended for networks that respond to queries rather that generating events based on the physical environment ▪ Service layer runs a collection of lightweight services
Adaptive Actuator Allocation, Aggregation and Prioritization Strategy	▪ Balancing of actor node loads by providing an allocation algorithm ▪ Minimizes the distance between significant sensor nodes and actor nodes ▪ Encompasses virtual grids ▪ Employs data aggregation ▪ Prioritization of data	▪ Actor nodes frequently moving to different locations might cause a lot of routing disturbances and network reconfiguration burdens

Table 3.1. WSAN APPL Protocol literature summary.

3.2.3 Discussion

Although many application areas for WSANs are defined and proposed, potential APPL protocols for sensor actor networks remains a largely unexplored region [8]. However, the list of primary tasks serves as a guideline for application protocols that may be developed to include adaptability to different application areas.

The existing literature that is discussed tries to deal with the challenges and tasks of the APPL protocol. With the exception of the last discussed paper, all of the discussed literatures deal with the actual actuation once measurement data is received by the actor nodes. This is only the final step in the process of sensing and acting on an event. It is important to consider the other tasks, like data aggregation and prioritization as proposed by the strategy in section 3.2.2.6, as well. The traffic load amongst low power sensor nodes can be minimized by employing such strategies.

The less discussed challenges of providing localization and positioning, and time synchronization in WSANs is not trivial and needs to be addressed to ensure operational success of WSANs. This information may not only be needed by the APPL, since other layers in the CPS may rely on this information for successful operation. For the issue of providing localization and positioning information to nodes, the distance between neighbouring nodes can be estimated on the basis of incoming signal strengths. Relative coordinates of neighbouring nodes can be obtained by exchanging such information between neighbours [20]. Alternatively, the location of nodes may be available directly by communicating with a satellite, using GPS (Global Positioning System), if nodes are equipped with a small low power GPS receiver. Recent development of small, inexpensive, and low-power GPS receivers make them viable for use in WSANs [22]. These GPS receivers can also provide time synchronization by using the received global clock information of the GPS. In another approach, simple synchronization may also be achieved by periodically broadcasting a synchronization packet throughout the network for example. However, before such an approach can be implemented, the added communication load on the network has to be considered. Further elaborate discussions for the issue of localization and time synchronization in the APPL is however beyond the scope of this dissertation.

As suggested by several literatures, a distributed actuation paradigm is the most suitable for WSANs, by following a fully automated architecture. In semi-automated architectures, sensed data has to travel all the way to a central controller before actuation control can be signalled to the actor node. This increases the time it takes for an actor node to actually perform an action after a sensor node had initially detected an event. By employing an automated architecture, sensed data is directly sent to a single or multiple actor nodes for quick response [2].

3.3 THE NETWORK LAYER

The layer that follows the applications layer in the CPS is the NETL. The NETL of a sensor node receives the APPL packets and is then responsible to choose a suitable path in relaying and routing them towards an actor node. The NETL protocols are hence also referred to as routing protocols. Primary NETL tasks and challenges are considered for sensor and actor nodes, followed by a discussion of WSN routing protocol classification and finally an examination and evaluation of current WSAN NETL protocol literature is given.

3.3.1 Primary Tasks and Design Challenges

The NETL protocols for WSANs have to deal with the occurrence of four types of communication phenomenon which include sensor-sensor, sensor-actor, actor-sensor and actor-actor communication. However, NETL protocols specified for the sensor nodes only involve sensor-sensor and sensor-actor communication, while the actor node NETL protocols involve actor-sensor and actor-actor communication. This leads to some NETL protocol differences for sensor nodes versus actor nodes. The primary tasks and challenges are listed as follows.

- The *initial network connectivity setup* is handled by the NETL. Since the network nodes are mostly unattended, self configurable algorithms are needed [27] for nodes to organize themselves after node deployment to form a functional network. A setup challenge includes the selection of a suitable actor node by the sensor nodes inside the network [2].

- *Construction of NETL specific data packet*. The header of the NETL packet contains the source and destination address amongst others, while the payload may contain different NETL parameters. The APPL frame should be encapsulated by the NETL packet. Once the payload is added and the header constructed, the packet is sent to the next layer in the CPS.

- Some kind of *addressing scheme* is required to distinguish between neighbouring nodes. Global or local addressing of nodes should be considered to eliminate messages being sent in endless loops. A source and destination address should be included in the NETL packet header for destination nodes to recognize where a packet came from and identify its immediate neighbours.

In addition to the above task, the ***sensor* nodes** in WSANs involve sensor-sensor and sensor-actor communication with the following tasks and challenges.

- *Maintaining connectivity* inside WSANs is very important since network dynamics have to be considered and dealt with effectively [28]. One dynamic nature of WSANs involves the mobility of actor nodes. Another dynamic nature of WSANs involves the failure of sensor nodes. High node density in sensor networks precludes them from being completely isolated from each other. Therefore, sensor nodes are expected to be highly connected. This, however, may not prevent the network topology from being variable and the network size from shrinking due to sensor node failures. Some sensor nodes may fail or be blocked due to lack of power, physical damage, or environmental interference. The failure of sensor nodes should not affect the overall task of the WSAN [20]. If sensor nodes fail, routing protocols must be adaptive by accommodating formation of new links and routes towards the data destination actor nodes.

- *Real-time*, *min delay routing* of delay sensitive data from sensor nodes towards the actor nodes is one of the main WSAN requirements. In addition to determining the path selection and data delivery, the routing protocol should support real-time communication to ensure the validity and timely response of actor nodes to a critical event occurrence. This is one of the most important challenges for WSAN routing protocols, since the real-time requirement distinguishes WSANs from other types of networks.

- *Energy efficiency routing* of data from sensor nodes towards an actor node. At the NETL, the main aim is to find ways for energy efficient route setup from the sensor nodes to the actor nodes so that the lifetime of the network is maximized. Since the sensor nodes are extremely resource constraint with limited memory, communication bandwidth, computation capabilities and especially limited energy, the routing protocol is responsible for selecting the most energy efficient route to ensure the maximum preservation of energy [2].

- *Reliable relaying of data* inside the WSAN is crucial for the overall success of the application. If critical data is not received by the actor nodes reliably, effective acting cannot be guaranteed. Since sensor nodes are prone to failure, the routing protocol has to mitigate this occurrence by selecting new paths through regions of the network where there are higher levels of available energy in order to preserve the overall network functionality.

- *Dispersion of node status parameters*, pertaining to the NETL, to neighbouring nodes is needed to keep nodes informed of their neighbour's status. The routing protocol may not only base its route decisions on the distance from the actor node, since multiple same distance routes may exist towards the actor node. Therefore, protocols include other parameters in deciding between equally attractive paths. These might include energy resource information or current traffic load at the next node. The challenge is to broadcast status parameters to neighbours at an interval that will not incur congestion.

- *Data aggregation* may also be considered within the NETL. If a packet queue is employed, packets with equal destination may be aggregated. This technique may be used to achieve energy efficiency [20] and alleviate congestion.

- The *routing protocol has to be scalable*. The number of sensor nodes deployed in the sensing area may vary between different applications. The routing protocols have to function efficiently for smaller and larger number of sensor nodes [20].

In addition to the first task, the ***actor* nodes** in WSANs involve sensor-actor and actor-actor communication with the following tasks and challenges.

- *Communication between actor nodes* is necessary when collaboration is needed in performing actions in response to sensor data. Dependent on the application, actor nodes have a specific range of acting. If an event occurs that needs some acting, actor nodes are involved by either responding alone or in collaboration with other actor nodes. Collaboration may be necessary when attending to a larger target area or as a precautionary measure to prevent an event from spreading. The actor node routing protocols have to include algorithms for effective actor-actor communication, that involves low overhead and low latency [2]. Actor-actor communication usually involves two actor nodes communicating directly with each other in a peer-to-peer manor.

- *Communication to sensor nodes* is necessary when actor nodes are instructed to query sensor nodes in the network by the application. Some applications may want to change sensor node parameters, like for example sensing threshold levels or the frequency of sensing, or instruct sensor nodes in a specific region to send some other particular information. This requirement necessitates the inclusion of actor-sensor communication and coordination by the actor node network protocol.

3.3.2 WSN NETL Protocol Classification

The problem of routing data within WSNs has been researched extensively. Due to the fact that WSANs have additional stringent requirements, WSNs routing protocols cannot be directly used for WSANs. However, most current WSAN NETL literature has taken WSN network protocols as a base or starting point in developing protocols for WSANs. It is therefore relevant to give an overview of WSNs routing classifications to understand the direction of WSANs routing protocol research. Basically, WSNs are networks that comprise of only low powered, randomly deployed sensor nodes that all relay sensed data towards a single base station or sink.

Current research on routing in WSNs routing protocols mostly focused on protocols that are energy aware to maximize the lifetime of the network, are scalable to accommodate a large number of sensor nodes, and are tolerant to sensor damage and battery [29].

In general, routing in WSNs can be classified into three main approaches, as shown in Figure 3.3, that include *flat-based* routing, *hierarchical-based* routing, and *location-based* routing. They can be further divided into *proactive*, *reactive* and *hybrid* protocols depending on how the source finds a route to the destination. In proactive protocols, all routes are computed and maintained before they are really needed, while in reactive protocols, routes are computed on demand. Hybrid protocols use a combination of these two ideas [20].

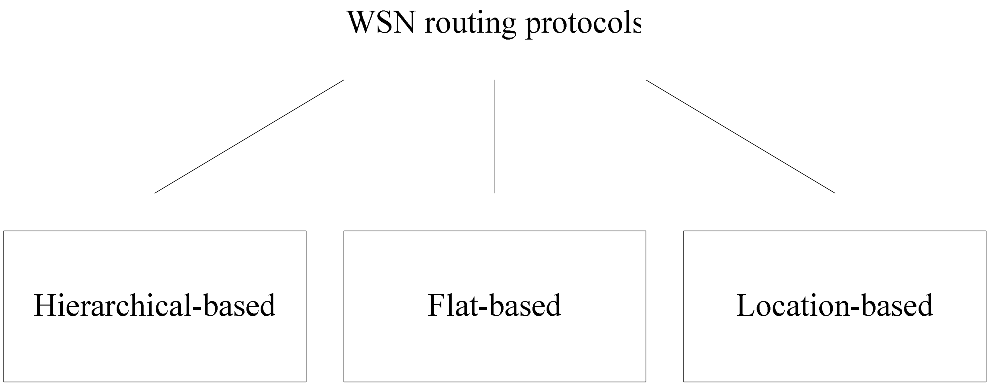

Figure 3.3 High level routing protocol classification.

These three main high level classifications relating to the network structure are discussed as follows. In flat-based routing, all nodes are typically assigned equal roles or functionality. This first category of routing protocols is the multi-hop flat routing protocols. In flat networks, sensor nodes collaborate together to perform the sensing task. Due to the large number of such nodes, it is not feasible to assign a global identifier to each node. This consideration has led to data centric routing, where the base station sends queries to certain regions and waits for data from the sensors located in the selected regions. Popular flat-based routing algorithms include: Flooding and Gossiping, Sensor Protocols for Information via Negotiation (SPIN), Directed Diffusion, Rumor routing and Gradient-Based Routing (GBR) [28].

In hierarchical-based routing, however, nodes play different roles in the network. The hierarchical or cluster-based routing, originally proposed in wire-line networks, are well-known techniques with special advantages related to scalability and efficient communication. As such, the concept of hierarchical routing is also utilized to perform energy- efficient routing in WSNs. In a hierarchical architecture, higher energy nodes can be used to process and send the information while low energy nodes can be used to perform the sensing in the proximity of the target. This means that creation of clusters and assigning special tasks to cluster heads can greatly contribute to overall system scalability, lifetime, and energy efficiency. Hierarchical routing is an efficient way to lower energy consumption within a cluster by performing data aggregation and fusion in order to decrease the number of transmitted messages to the base station. Hierarchical routing is mainly a two-layer routing concept, where one layer is used to select cluster heads and the other layer is used for routing

[20]. The research issue regarding such protocols is how to form the clusters so that the energy consumption and contemporary communication metrics such as latency are optimized [28]. Popular hierarchical-based routing protocols include: Low-Energy Adaptive Clustering Hierarchy (LEACH), Threshold-sensitive Energy Efficient Protocols (TEEN and APTEEN), Power-Efficient Gathering in Sensor Information Systems (PEGASIS), Self-organizing protocol and Hierarchical Power-aware Routing (HPAR) [20].

In location-based routing, sensor nodes' positions are exploited to route data in the network. In this kind of routing, sensor nodes are addressed by means of their location. The distance between neighbouring nodes can be estimated on the basis of incoming signal strengths. Relative coordinates of neighbouring nodes can be obtained by exchanging such information between neighbours. Popular location-based routing algorithms include: Geographic Adaptive Fidelity (GAF), Geographic and Energy Aware Routing (GEAR) [20].

The routing approaches that have received the most attention are the hierarchical and flat routing approaches. As a comparison, Table 3.2 lists the main differences between the two routing approaches for WSNs.

Hierarchical routing	**Flat routing**
Data aggregation by cluster-head	Node on multi-hop path aggregates incoming data from neighbours
Simple but not always optimal routing	Routing can be made optimal but with an added complexity
Requires some form of synchronization	Links formed on the fly without synchronization
Overhead of cluster formation throughout the network	Routes formed only in regions that have data for transmission
Lower latency as multiple hops network formed by cluster-heads always available	Latency in waking up intermediate nodes and setting up the multi-path
Energy dissipation is uniform	Energy dissipation depends on traffic patterns

Table 3.2. Hierarchical vs. flat topologies routing [20].

The second level of classification, mentioned above, includes proactive, reactive and hybrid routing as illustrated in Figure 3.4.

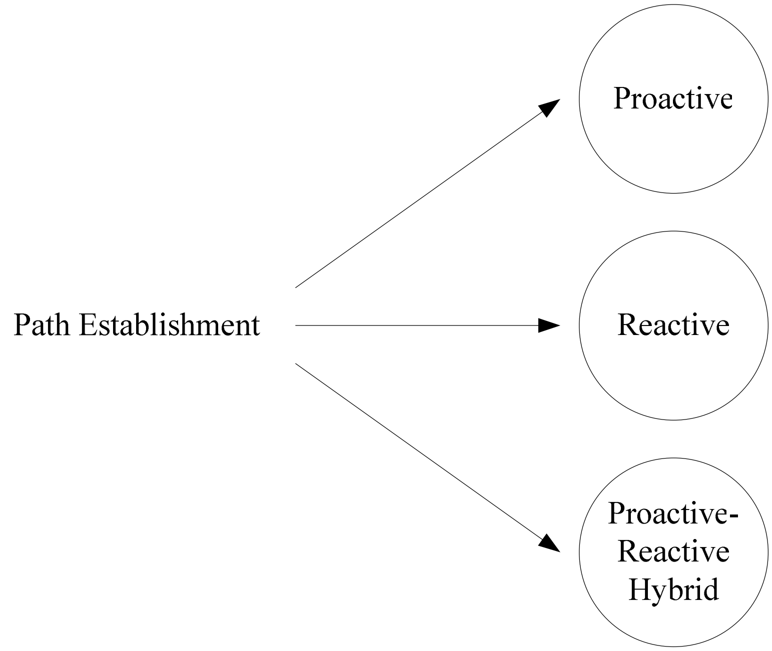

Figure 3.4 Path Establishment routing protocol classification.

Proactive algorithms maintain routing information about the available paths in the network even if these paths are not currently used. The main drawback of these approaches is that the maintenance of unused paths may occupy a significant part of the available bandwidth if the topology of the network changes frequently. *Reactive* routing protocols maintain only the routes that are currently in use, thereby reducing the burden on the network when only a small subset of all available routes is in use at any time. However, they still have some inherent limitations. First, since routes are only maintained while in use, it is typically required to perform a route discovery before packets can be exchanged between communication peers. This leads to a delay for the first packet to be transmitted. Second, even though route maintenance for reactive algorithms is restricted to the routes currently in use, it may still generate a significant amount of network traffic when the topology of the network changes frequently. Finally, packets en route to the destination are likely to be lost if the route to the destination changes. *Hybrid* routing protocols combine local proactive routing and global reactive routing in order to achieve a higher level of efficiency and scalability [20].

3.3.3 Existing Literature on WSAN Network Layer Protocols

The real-time and energy efficient requirements are the main design goals of a WSANs routing protocol, while maintaining a considerable reliability. The NETL is responsible for routing the data from the sensor nodes towards the actor nodes. Sensed data travels in a hop-by-hop manor towards the actor nodes. When receiving data, the routing protocol on a sensor node determines which the next-hop sensor node is on route to the actor, with special considerations for latency, energy, and reliability efficiency. In this section, a literature overview is given to give insight into the WSANs NETL literature, followed by existing cluster-based and non cluster-based WSAN NETL protocols and finally a tabulated summary with features and problems of each protocol.

3.3.3.1 Literature Overview

The research pertaining to the NETL protocols for WSANs has grown over the past few years and several protocol proposals have been made to address the routing requirements inside WSANs. As a starting point or basis for development, WSNs routing protocols have been considered since WSNs have similar underlying routing requirements. Current research on routing in WSNs mostly focused on protocols that are energy aware for the maximization of the network lifetime and are scalable to accommodate a large number of sensor nodes. Since most of the research in WSNs is dominated by the issue of energy efficiency, the delay minimization requirements was never a primary concern in most of the published WSN work on the routing layer. However, in WSANs, depending on the application, there may be a need to rapidly respond to sensor input. Moreover, to provide right actions, sensor data must still be valid at the time of acting. Therefore, the issue of real-time communication is very important in WSANs since timely actions need to be performed on the environment in response to the sensing [29]. While being able to control the use of energy remains a paramount design goal for WSANs, it is becoming increasingly important how to better trade it for lower delays.

When compared to WSNs, WSANs have much more stringent requirements and comprise heterogeneous nodes. Most applications require distributed control and impose special requirements of latency, energy and reliability efficiency. These requirements render WSN routing protocols inappropriate for WSANs. However, the trend of current emerging WSAN NETL protocols research shows that a cluster-based hierarchical routing approach is most favourable for WSANs. It is the most suitable approach when dealing with the existence of resource richer actor nodes spread among numerous sensor nodes throughout the network. Very suitably, well-sized clusters can be formed around actor nodes that take the role of cluster head. In WSN protocols, the role of a cluster head is commonly assigned and rotated among sensor nodes, which takes its toll on the network lifetime. The problem may be mitigated in WSAN by making the resource richer actor nodes the cluster heads, ensuring longer network lifetime. The division of the WSAN into clusters consisting of cooperating nodes, has several advantages such as: increased robustness and security; simplified addressing, routing, and localization; lower memory requirements; and it enables a much more focused optimization approach towards lower delays and lower energy consumption. Since sensor and close-by actor nodes are in the same cluster, the delay in the control loop is reduced, ensuring quick responses to events [30].

The WSN routing protocol classification discussed in section 3.3.2 also applies to WSAN routing protocols. Moreover, routing schemes may employ a unicast, multicast, broadcast or anycast approach. In *unicast*, there is a one-to-one association between source and receiver, where each source uniquely identifies a single receiver. In *broadcast* and *multicast*, there is a one-to-many association between source and receivers where each source identifies a set of receiver node, to which all information is replicated. In *anycast*, there is also a one-to-many association between source and receivers, where each source identifies a set of receivers, but only one of them is chosen at any given time to receive information from any given sender. Data is routed to the "nearest" or "best" destination as viewed by the routing topology.

In the following subsections, existing WSANs literature is discussed in detail. Although the cluster-based approaches seem most suitable and viable for WSANs, existing protocols that do not employ a cluster-based approach are also examined and evaluated to form a complete survey of existing WSAN NETL protocols.

3.3.3.2 Cluster-based Protocols

Existing protocols that use a clustering approach are evaluated in this section.

3.3.3.2.1 Hierarchical Geographic Clustering Protocol

A novel three-level coordination model proposed by Yuan et al. [22], is based on a Hierarchical Geographic Clustering (HGC) paradigm in which cluster formation is done by dividing the action area into fixed zones to form a virtual grid. The virtual grid is introduced to optimally split the working area and workload between different actor nodes. It is assumed that sensor nodes are stationary and location-aware, whereas actor nodes may change their locations dynamically.

For the network setup, an R-based clustering algorithm that is based on the Geographical Adaptive Fidelity (GAF) [31] routing protocol, which is also considered as a hierarchical protocol, is used. As shown in Figure 3.5, the virtual grid size is based on the nominal radio range R of stationary sensor nodes. The distance between two possible farthest sensor nodes in any two adjacent grids, such as grid C2 and C3, must not be larger than R. The distance R is the furthest distance the node radio transmitter can successfully transmit data.

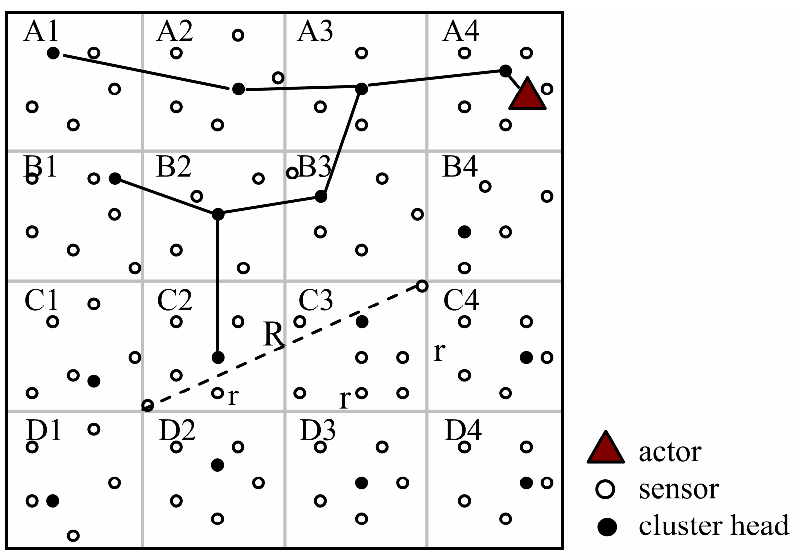

Figure 3.5 R-based clustering and routing. From [22].

As illustrated in Figure 3.6, inside each grid, sensor nodes will elect one sensor node as cluster-head that aggregates the sensed data received from each sensor node associated with it. This presents the first level of coordination and communication. The cluster-head is responsible for reporting gathered data to an appropriate actor node. Actor nodes broadcast their information, which includes the current location and an accurate time, periodically or throughout the duration of a position change. After receiving this information, each cluster-head in the WSAN achieves clock synchronization, and maintains a routing table that includes nearby actor node(s) to deal with the mobility of actor nodes. When an event occurs each cluster head transmits or relays event information immediately to the closest actor node with one hop or multi-hop, as illustrated in Figure 3.6, to ensure minimal energy or timing consumption. This presents the second level of coordination and communication.

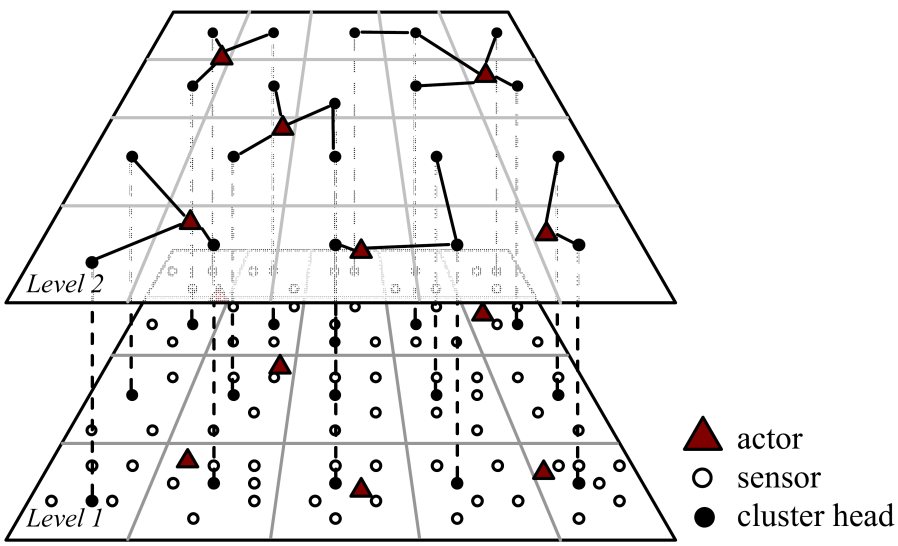

Figure 3.6 Coordination levels 1 and 2. From [22].

All cluster heads associating with one actor node construct a data aggregation tree towards the selected actor node. All actor nodes triggered by the same event, construct a second level aggregation tree in the actor-actor coordination level, towards the actor node in the center of an event area as illustrated in Figure 3.7. This actor-actor coordination and communication is the third level in the communication model.

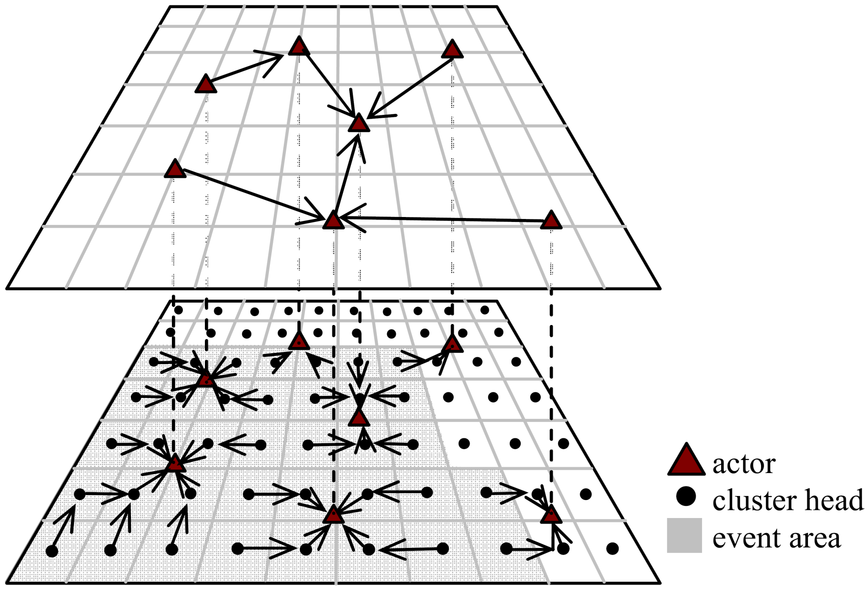

Figure 3.7 Two-level aggregation tree. From [22].

Since there are a smaller number of actor nodes and the power capacities of these actors are higher than the sensor nodes', the authors propose the use of an existing routing protocol that has been specifically designed for wireless ad-hoc networks, for the actor-actor coordination level. No specific protocol is mentioned. Finally, the addressing scheme employed by the communication model uses the serial number of each grid as the cluster head id. The id is calculated on demand and shared by each sensor within the same cluster to decrease the overhead of id maintenance.

This sensor-actor coordination paradigm may be well suited for application scenarios where multiple events occur simultaneously or the occurrence of events is frequent. Furthermore, it is also well suited for applications scenarios where query requests are made on a regular basis. This model envisages utilizing actor nodes to perform the most of energy-consuming tasks such as routing computation, data aggregation and large-scale wireless communication by exploiting their superior processing capabilities, stronger transmission powers and longer battery life.

One problem that could have detrimental effects is with the selection of the cluster-head. Some form of rotation of the cluster-head role is necessary to prevent rapid energy depletion of single sensor nodes that remain the cluster head for a longer duration of time. Furthermore, a very dense deployment of sensor nodes is assumed with radios that have a considerably high transmitter power. The sensor node radios need to be more powerful since it is assumed that adjacent cluster heads (selected sensor nodes) can communicate with each other. With the use of powerful radios a lot of channel noise and interference may be incurred inside the network. This places daunting problems on the underlying MACL protocol. Furthermore, more power radios deplete the energy resource more rapidly and are less cost effective.

3.3.3.2.2 Event-driven Clustering Protocol

Another cluster-based paradigm is presented by the distributed coordination framework proposed by Melodia et al. [21]. A new event-driven clustering paradigm is proposed, where cluster formation is triggered by an event so that clusters are created on-the-fly to optimally react to the event. Sensor nodes detecting an event coordinate and communicate with each other so as to optimally associate each sensor node with an actor node. Sensor nodes do not elect additional sensor cluster-head nodes. Only the event area is clustered, and each cluster consists of those sensor nodes that send their data to the same actor node. This ensures that the event information is collected at the optimal actor nodes while existing energy resources are better utilized, since clusters are formed only when necessary, based on the event features and on the position of the actors. The resulting architecture is shown in Figure 3.8. The event-driven clustering approach tries to eliminate the communication overhead of maintaining clusters before an event occurs. It is assumed that a node knows its position, the position of its neighbours and the position of the actors, while the network is always synchronized.

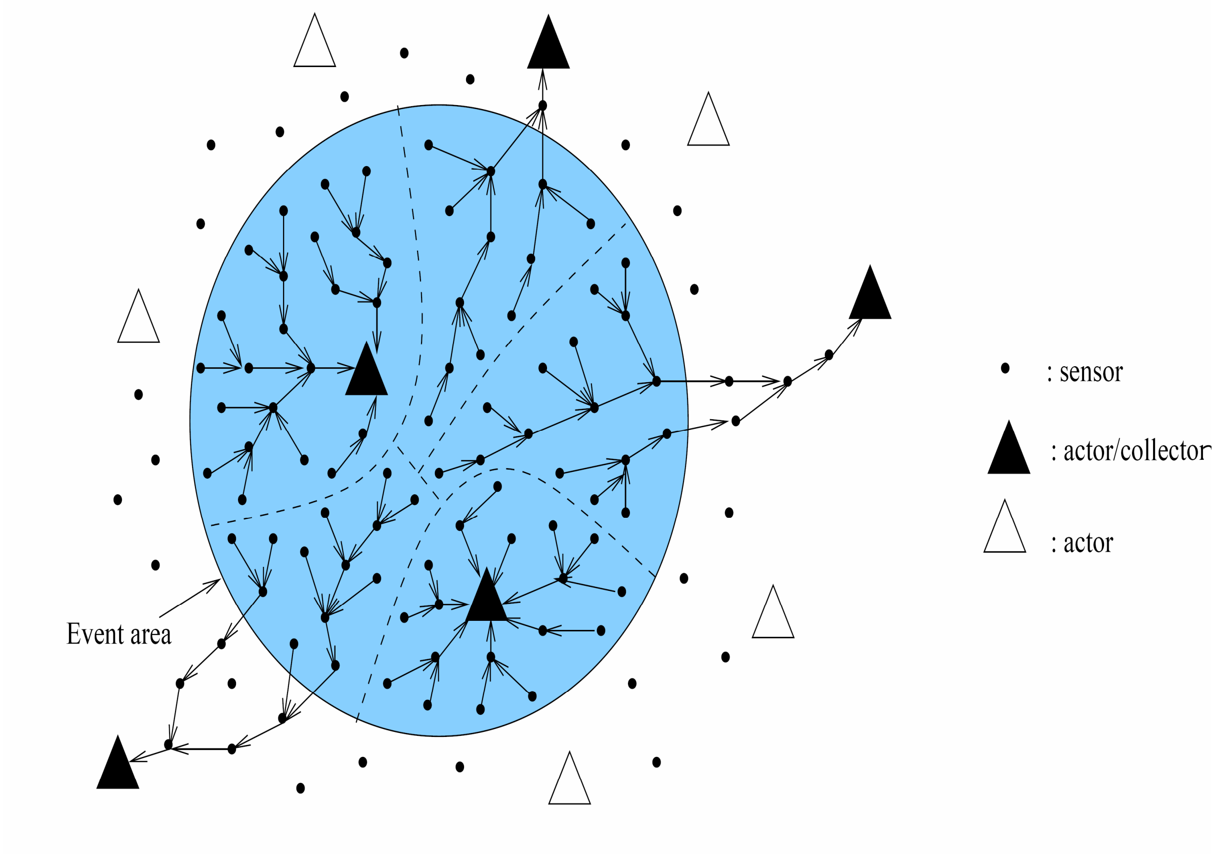

Figure 3.8 Event-driven clustering with multiple actor nodes [21].

The optimal solution to the event-driven clustering problem is determined by Integer Linear Programming. A multi-state distributed algorithm that achieves an energy-efficient solution for sensor-actor coordination and communication is proposed, and includes an adaptive mechanism that trades-off energy consumption for delay when event data must be delivered to the actor nodes within predetermined latency bounds. The proposed protocol strives to minimize the energy consumption by relying on localized information and on greedy routing decisions. A data aggregation tree (da-tree) is thus created between each actor node, called collector, and the sensor node sources associated with that actor node. This way, the set of sources is implicitly clustered, each cluster being composed of the sources associated with a single collector.

The protocol encompasses four states which include the *idle*, *start-up*, *speed-up* and *aggregation* state. The main objective of these state transitions is to reduce the number of hops when the real-time requirement is violated and to save energy when the real-time requirement is met. A sensor node enters the start-up state from the idle state when it detects an event, or when it receives a packet to be relayed to an actor node. It then selects its next-hop based on the so-called two-hop rule. According to the two-hop rule, a node i selects as next hop among its neighbours the node j that minimizes the sum of the energy consumption from i to j and the energy consumption from j to the actor node closest to j. In the speedup state, the objective of the collaborative operation of nodes is to minimize the number of hops between sensor node sources and actor nodes. This is achieved by applying the Greedy Routing Scheme, where each sensor node sends the packet to the node closest to the destination within the transmission range. Finally, the objective of the aggregation state is to reduce the overall energy consumption, by employing a data fusion algorithm implemented on each sensor node. Since data packets can be aggregated by any node in the network, the sensor node routes data to the closest node in its neighbourhood that is part of the da-tree. This way, the incremental energy consumption to collect the information from the considered sensor node is minimized. In summary the different states cater for different energy and latency demands posed by an event.

When all sensor nodes coordinate with each other to divide themselves into clusters and each node selects a different actor node to transmit the data to, there may be a need for many data transmissions among these sensor nodes every time an event occurs. This may cause high energy consumption, network overload as well as high latency. The conclusion is that an event-driven clustering paradigm may not be well suited for application scenarios where multiple events occur simultaneously or the occurrence of events is frequent or periodic. However, in WSANs a trade-off does exist between energy savings and latency minimization and this protocol introduces a sensible way of incorporating both by analyzing the current network status and event situation in deciding which one to promote.

3.3.3.2.3 Hierarchical, Energy-Efficient Routing with Ripple-Zones

A hierarchical routing scheme with special consideration for energy efficiency is proposed by Hu et al. [7]. A huge WSAN is assumed, with a very low ratio of actor nodes (medical phones) to sensor nodes. As illustrated in Figure 3.9, the network is divided into domains and then within a domain, into zones each with a sensor node acting as root and aggregation master. A Voronoi-Tessellation-based algorithm is used to determine the number of zones and probability of each sensor becoming a spanning tree-root. The number of zones equals the root-selection probability multiplied by the total number of sensors. Each root broadcasts a 'hello' packet to its neighbouring sensor nodes. The sensor nodes then compare the received signal strengths from all in-range roots. Finally each sensor node chooses the closest root, i.e. with the strongest received signal strength, and claims itself to be a member of its zone. An actor node and its zones form a domain and each node in the domain maintains an id array. After the self-organization into domains, all the members that belong to the same zone exchange packets to form a minimum spanning tree (MST) [32]. A MST routing scheme is chosen since it can provide a loop-free and multi-hop relay-based communication mode. This scheme chooses the transmission power between any two nodes as the weight of an edge to quickly converge to a stable MST. An important feature of the MST algorithm is the determination of the minimum transmission power of each sensor node to maintain connectivity. This is in contrast to traditional topology control algorithms in which each node transmits using its maximum transmission power. Furthermore, the transmission power has to be set at the PHYL and cannot be done within a strictly layered NETL. Therefore, this protocol cannot function without interaction with the PHYL via some sort of cross-layer framework.

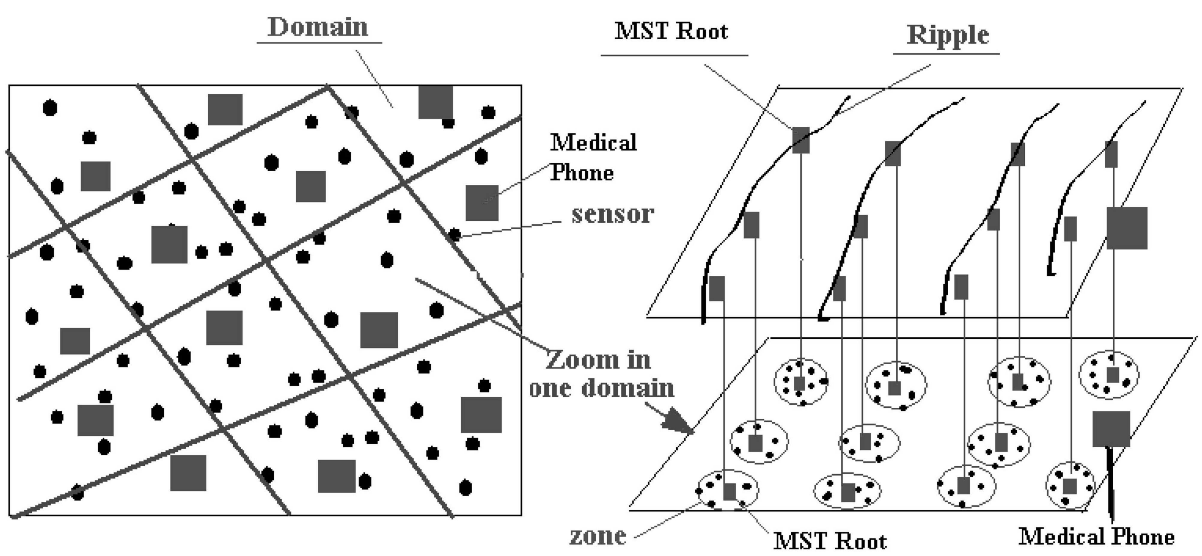

Figure 3.9 Network clustering with domains and zones. From [32].

As illustrated in Figure 3.10, the concept of a Ripple-Zone around each actor node within the domain of each actor node is proposed. The MST-roots, acting as aggregation masters, are assigned to different "ripples" based on their distances, in number of hops, from their actor node. The MST-root aggregates data from the sensor nodes in its zone before it transmits data to a master in next ripple that is closer to the actor node or directly to the actor node. An efficient path balancing and traffic splitting algorithms for balancing the energy consumption among masters towards the actor node is proposed. It employs two approaches of direct transmission and multi-hop transmission proportionally during data transmission in order to optimize the lifetime of the entire sensor network as an integral entity.

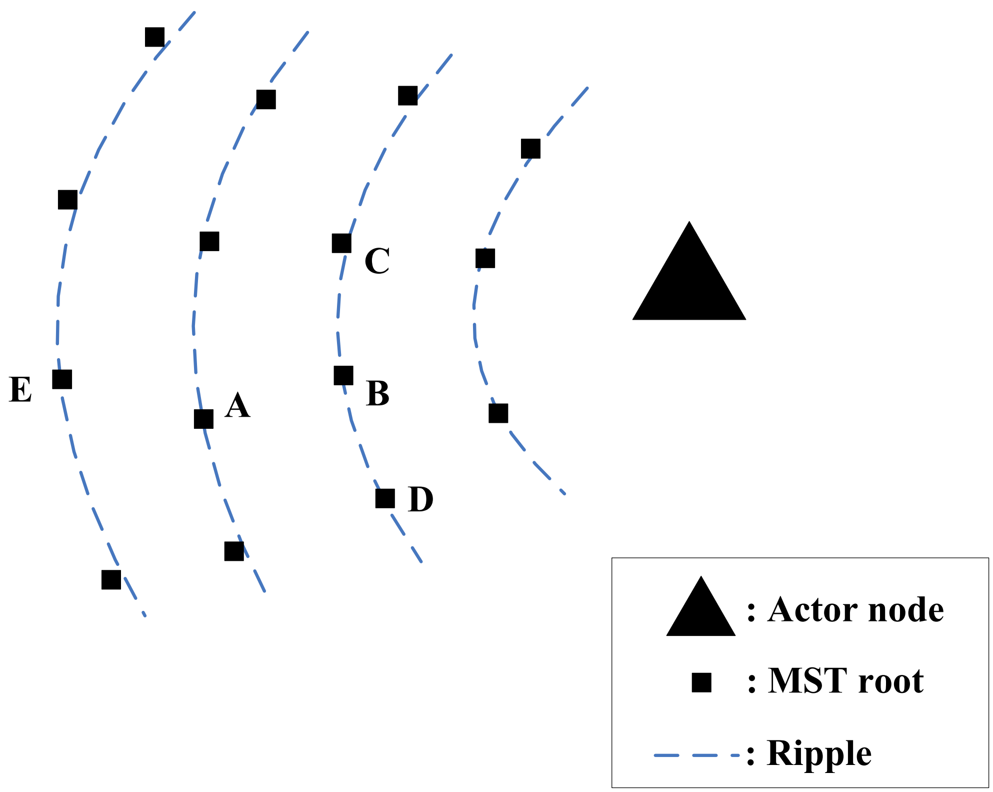

Figure 3.10 Ripples as a function of distance from the actor node.

Reliable packet delivery is achieved by employing single-path unicast routing with local recovery. Unicast is communication between one sender and one receiver. The single-path unicast routing scheme is explained according to Figure 3.10. Suppose a root B in the next ripple is not reachable by A due to its battery depletion or an unreliable wireless link. Root A then checks if the A's neighbouring masters C and D are reachable. If any one of them is reachable, it is used. If C and D both fail, the packet is sent backwards to the next ripple further away from the actor node to root E. Root E then tries to delivery the packet via A's neighbouring roots.

The proposed hierarchical routing scheme makes use of domains and zones around an actor node, which is similar to the grid system with cluster-heads of the HGC protocol in section 3.3.3.2. The authors compared the single path approach with a multi-path routing with no packet loss recovery approach and established that multiple paths from any root to an actor node needs higher maintenance overhead and generates multiple copies of the same packet, which consumes more energy than single-path approach.

However, with regards to the data transmissions from aggregation masters/MST-roots to the actor node, i.e. inter-zone communication, direct transmissions to the actor node are sometimes required, which is only possible if the master nodes/MST-roots have higher powered radios that can transmit with enough power to reach the actor node. Even for multi-hop inter-zone routing with transmissions between masters of adjacent ripples, radios are required to transmit packets over comparatively long distances. High powered radios are much less cost-effective and require much more energy. This may compromise the feasibility of WSANs. Furthermore, proposing the use of aggregation masters/MST-roots may prove to be a huge disadvantage since if nodes within a zone first have to send their packets towards their aggregation master rather that directly to the next zone, the packets travel further, consuming more energy and adding more latency. This approach may only be viable for excessively large WSANs that entail a very dense sensor node deployment. Moreover, the issue of satisfying the real-time requirement has not explicitly been addressed. A TDMA-based MAC protocol is used for the evaluation of this protocol, which as explained in the next section (3.4), may be less suitable for WSANs.

3.3.3.2.4 Clustering by Weighted Bi-partite Matching

A low cost distributed actuation strategy, that involves the formation of clusters for a distributed control approach is proposed by Lin and Megerian [33]. The main idea in the distributed approach is to group the sensor nodes into smaller clusters and solve a problem locally. Distributed control is favoured for WSANs since increased power consumption and latency occurs when transmitting and routing the necessary information all the way to a centralized controller and back to the actor nodes in a centralized control approach.

The authors propose to solve the underlying problem of assigning sensor nodes to clusters by introducing an optimization technique of weighted bi-partite matching. The actor and sensor nodes can be thought of as a bi-partite graph with edges connecting each sensor node (or set of sensor nodes) to each actor node. Each edge is weighted according to a cost function that for example could be based on the distance from the sensor to the actor node. Optimal matching is done to form clusters. When a new sensor node S is added to the network and activated, it listens to collect current cluster information in its neighbourhood. If there are already formed clusters, S chooses the cluster whose corresponding actor node has the shortest distance with itself by sending a join request to that cluster. Each cluster has a capacity limitation imposed to limit the traffic within the cluster. If the cluster is full, S will choose the next best cluster in a greedy fashion. This repeats until S successfully joins one cluster.

Both overlapping and non-overlapping cluster scenarios are considered. Non-overlapping clusters present clusters that are formed as disjoint sets of sensors, while overlapping clusters are allowed to have overlaps where the shared sensors serve as a means of data exchange between clusters.

Specifics on the method of routing data within a cluster are not mentioned by the literature. However, the challenge of creating a communication structure within a network by forming clusters has been dealt with sensibly. The approach is scalable and caters for the random addition of new nodes.

The following NETL protocols are all non cluster-based approaches.

3.3.3.3 Non Cluster-based Protocols

Existing protocols that do not use a clustering approach are evaluated in this section. These protocols do not specify the formation of clusters and mostly do not structure the physical network.

3.3.3.3.1 Anycast Tree-based Communication Paradigm

A non cluster-based anycast service design that deals with the extension of network lifetime, reduction of end-to-end latency and improvement of network scalability is proposed by Hu et al. [34]. With anycast, there is a one-to-many association between source and destination nodes: each destination address identifies a set of receiver endpoints, but only one of them is chosen at any given time to receive information from any given sender. The term exists in contradiction to multicast, communication between a single sender and a group of selected receivers, and unicast, communication between one sender and one receiver in a network. The anycast routing design targets five primary goals that include: simple, energy-efficient, self-organizing, adaptive and distributed. A simple anycast design accommodates small sensor nodes, by being computational and memory efficient. Energy-efficiency is attained by incurring minimal energy overhead for control as well as data communications. Self-organization and adaptability is achieved by being responsive to sinks joining and leaving dynamically and robust to node failures. Finally the anycast design employs a distributed approach to scale to arbitrarily large sensor networks.

In addition to the sensor and actor nodes, multiple micro-servers (sinks) interested in the same data are assumed to be present in the network. Sinks may be mobile and data needs to only reach one sink, thus motivating an anycast service. The routing protocol adopts a shared tree approach where, corresponding to each event source, a shortest-path tree rooted at the source is constructed. Sinks form the leaves of the tree and can dynamically join or leave the anycast tree. To handle sink mobility, paths to all sinks are simultaneously maintained by every source, by receiving periodic packets from sinks and refreshing the anycast table entries. Upon receiving the periodic packets, each node updates its anycast table by setting the cost metric of the branch to the sink. This approach is similar to the frequently cited WSN's Direct Diffusion routing protocol. The anycast table size may be controlled by limiting the number of destination sink entries, since the protocol is only interested in delivering packets to the nearest sink and therefore, it is not necessary to maintain paths to all sinks in every anycast table. When a sensor node receives no update from a specific sink contained in the table within a particular time, the entry is deleted. Once a data packet arrives or is generated by a specific sensor node, the node looks up its anycast table for the sink with minimum cost before it forwards the packet along the corresponding branch.

It is explicitly stated that the protocol does not consider the issue of reliable data delivery since the applications it is intended for can handle small amount of data loss. This narrows the application range down quite a bit and may prove detrimental for WSANs that encompass critical information. However, this anycast approach, which builds on the WSN's Direct Diffusion protocol, suggests a reasonably good way of dealing with mobility, as long as the periodic updates flooded from sinks are not too frequent. When too frequent updates are sent, heavy traffic loads may be incurred on the network. Finally, another problem that needs to be mentioned is that the authors have designed and evaluated the routing protocol in conjunction with the 802.11 standard MAC protocol originally designed for WLANs. This MAC protocol is not at all suitable for WSANs which strive towards high network lifetime. The 802.11 MAC protocol is not designed for low power sensor nodes but rather for devices such as laptop computers or cellular phones to join a wireless LAN widely used in the home, office and some commercial establishments [43]. As mentioned later in section 3.4.5, this protocol assumes the nodes to be always switched-on. This means that the sensor nodes will suffer from energy depletion very quickly, fragmenting the network and leading to total network failure. Hence, the use of the 802.11 MAC protocol is totally inappropriate for WSANs since it will excessively lower the network lifetime and totally undermine the feasibility of WSANs. Therefore, the performance evaluation by the authors is discounted since the routing protocol was designed and evaluated in conjunction with the highly energy-inefficient 802.11 WLAN MAC protocol.

3.3.3.3.2 Reliable Reporting of Delay-Sensitive Events Protocol

Similar to the HGC paradigm discussed in section 3.3.3.2, a Reliable Reporting of Delay-Sensitive Events protocol (RRDSE) [26], uses virtual grids with an aggregation node for each grid. However, this routing and transmission protocol does not specify clustering of sensor nodes around an actor node. A general reliability-centric framework for event reporting is proposed, where the reliability depends not only on the accuracy, but also the importance and freshness of the reported data. A reliability index is used, which gives a measure of the probability that the event data is aggregated and received accurately within pre-defined latency bounds. The communications from the sensor to the actor nodes follow an anycast paradigm, that is, an event reporting is successful if any of the actor nodes receives the report.

The key design objective of the protocol is to maximize the number of event reports reaching the destination within their latency bound, and, for different event types, give preference to important events. A priority queue is adopted in each sensor node to play two important roles of: prioritized scheduling to speed up important event data transmission; and queue utilization as an index for route selection to meet the latency bounds. In the pre-emptive priority queue, the packets for the event data are placed according to their data importance, and each priority is served in a first-in-first-out discipline. In a network with frequent event occurrences, queuing delay can be the dominating factor over the processing and propagation delays. It is not clear how the data prioritization is done since no APPL protocol is proposed and it is simply assumed that the data is already prioritized. However, without some kind of cross-layer framework, the NETL will not be able to view or attain APPL priority information about the data packets that are generated by the APPL.

After a sensor node receives a control packet from its neighbours, it calculates packet arrival rates for low and high priority packets. A sensor node needs to ensure that the end-to-end latency for a data packet is no more than a specific latency bound. When searching for the next hop, it first estimates the advancement gained when forwarding the packet via a neighbour towards the actor node. Secondly, the maximum data hop-to-hop delay that the packet will experience is calculated. For each candidate, the sensor node calculates the maximum data rate (with the hop-to-hop delay) that it can forward the data packet while satisfying the latency bound. Finally, the event data packets are forwarded to the neighbour with the highest advancement and data rate which is closest to the destination. Each intermediate node updates the latency bound before forwarding the packet to the next hop. After the transmission starts, the sensor will update its expected service time and the routes regularly, to make sure the transmission can be completed within the latency bound. If the latency bound is not met, the sensor has to forward the packets to another route. In the worst case, if no alternative can be found, the sensor may inform the previous node to select another route in the future.

The successful routing of data packets depends on the control packets that are sent by the neighbour nodes. It is not stated at which instance the control packets are sent, but special care has to be taken to minimize the traffic load incurred by the control packets. No explicit considerations for energy minimization have been mentioned, which is almost equally important as the latency minimization. Furthermore, it is not clear how the latency bounds of data packets are calculated. Nodes with a specific hop distances will need latency bounds assigned to them according to their specific hop distance. This may impose a lot of complex calculations and seems to add to an already complex routing algorithm. As with the data packet prioritization, any latency bounds are usually determined within the APPL and not the NETL.

Finally, the data rate used as part of the packet forwarding criterion, is calculated without taking into consideration the timing complexity of a feasible underlying MACL protocol. The packet data rate cannot be accurately calculated without considering timing or contention issues of the MACL. The data rate is calculated by simply measuring the delay experienced by a NETL control packet and then defined as the delay that will be experienced by all packets. This is a very inaccurate way of measuring the delay and does not take into consideration important delay influencing parameters that include the current channel conditions and utilization. Detrimental delay calculation inaccuracies may occur if the channel conditions fluctuate a lot, as is commonly the case in WSANs with frequent random events. Furthermore, as with the Anycast Tree-based Communication Paradigm, RRDSE's authors have also designed and evaluated their protocol by using the 802.11 standard MAC protocol designed for WLANs. The data rate calculations within the RRDSE routing algorithm are done with network nodes that are always switched on. This means that if the RRDSE routing protocol was to be implemented in conjunction with a MAC protocol that has been designed for low powered sensor nodes (i.e. employing duty cycles), the data rate calculated will be incorrect since no MAC timing properties are considered. Consequently the delay results obtained by the authors cannot be evaluated against any other WSAN's simulations results obtained with simulations that make use of feasible MACL protocols that have been designed for low powered sensor nodes. Therefore, the performance evaluation by the authors is discounted since the routing protocol was designed and evaluated in conjunction with the highly energy-inefficient 802.11 WLAN MAC protocol.

3.3.3.3.3 Delay-energy aware routing

Another routing protocol that focuses particularly on the combined minimization of delay and energy consumption of packets is the Delay-Energy Aware Routing Protocol (DEAP) [29]. The non cluster-based DEAP mainly consists of two components which include routing based on forwarding sets and the random wakeup scheme. Only the routing is discussed in this section and the wakeup scheme in the MACL section 3.4.4.

DEAP handles the routing by employing greedy geographic routing that forwards packets to an active neighbour that is closest to the destination at each hop. The routing methodology in DEAP is designed to take advantage of the applications where WSANs are densely deployed. A high node density results in the existence of several paths between two given nodes, whose path lengths are very close to the length of the shortest path. A forwarding candidate set is constructed that contains a list of neighbours that meet a forwarding criterion. The forwarding criterion of a sensor node includes sensor nodes within the radio range and closer to the destination than itself with a certain Th threshold distance. This forwarding selection criterion guarantees that there would be no loops in the path. At the same time, this simple criterion cannot guarantee the delivery of a packet to the destination in the presence of holes (failed sensor nodes). Again, at high network densities, it can be assumed that holes would not exist.

Routing based on forwarding sets increases the path length. The Th value limits the maximum path length, as with each transmission a packet traverses at least a distance of Th towards the destination. Intuitively, because of increased path lengths, it might seem that Forwarding Set based routing ads additional overhead in terms of energy consumption. However, when combined with the random wake up scheme the total energy consumed by a sensor in DEAP is lower.

DEAP does not mention a method of selecting an actor to forward data packets to. It simply forwards the data packets towards the closest actor node. This might pose problems for border sensor nodes that have several ‘close’ actor nodes. Furthermore, the mobility of actor nodes has not been considered.

3.3.3.3.4 POWER-SPEED

A non cluster-based protocol that features power and delay minimizations is proposed by Zhou et al. [35]. It is assumed that all nodes are location-aware, and actor nodes can be mobile, suggesting frequent topology changes. This protocol proposes the forwarding of packets to be conducted in a stateless manner, which implies that in-network nodes do not maintain a routing table to the actor nodes. The approach taken by many routing protocols of employing global shortest-path routing is dismissed by the POWER-SPEED protocol authors, since the approach requires frequent reestablishment of shortest paths inevitably causing high overhead for exchanging of control packets. Furthermore, POWER-SPEED introduces the notion of a transmitter power control scheme where each sensor node sets its power level to a minimum value according to the requirement that the packet transmitted by a sensor node should just reach its intended neighbour. This is may prove to be problematic if POWER-SPEED is considered as a strictly layered NETL protocol, since any changes in the transmitter power is done within the PHYL. Therefore, like the Hierarchical Energy-Efficient Routing with Ripple-Zones protocol, this protocol cannot function without interaction with the PHYL via some sort of cross-layer framework.

The routing of an actual data packet by the sensor nodes involves the expiry time of a packet and the minimization of the energy consumption in forwarding it towards an actor node. Since every data packet has a deadline, the protocol sets out to ensure that the packet arrives at an actor node before it expires. This is done by estimating the future hop-to-hop delay a packet will experience, i.e. the delay between each pair of adjacent nodes in the selected traffic path, based on only the hop-to-hop delay estimation of the upstream hops the packet has travelled. This means that POWER-SPEED does not require feedback from downstream nodes to estimate the future hop-to-hop delay and in this way avoids control packets. The estimation approach employs a 2-dimensional (spatio-temporal) exponentially weighted moving average estimation that considers the impact of both time and space historical data.

By doing the delay calculations, a sensor node can estimate the speed of a route via one of its neighbours towards the actor node. The speed is defined as the average number of hops a packet can go through in one second. If a route complies with the data packet expiry demands, the neighbour node is selected as a candidate for the final route selection. Once all candidate neighbours have been established, POWER-SPEED calculates the total energy that will be consumed (influenced by transmitter power control) by each route to the actor node. Finally, among all packet forwarding candidates, the sensor node selects the next-hop node that achieves the minimum energy consumption. It then sets its transmitter power and forwards the packet to this candidate. The authors state that in this next-hop selection scheme, a packet that will expire in a longer period of time will adaptively be sent with lower transmitter power level to save energy. On the contrary, a packet that will expire sooner will be sent adaptively with higher transmitter power level, which results in a fewer number of hops from the sender sensor node to a destination actor node. In this way the packet is thus guaranteed to reach its destination in a shorter period of time. However, in order to skip nodes, knowledge of 2 or 3-hop neighbours is necessary, which means that nodes must have a detailed view of nodes within quite a large range around them. This requires very complex network setup procedures and is only viable for static networks. Furthermore, keeping the neighbourhood tables up-to-date requires high overhead. This leads to a high traffic load incurred by the control messaging and protocol overhead alone. With high traffic loads, latency and energy efficiency is compromised and the feasibility of the protocol in WSAN environments consequently considered doubtful.

POWER-SPEED tries to incorporate both requirements of delay and energy minimization. A method of eliminating control packets is suggested by doing away with complete routing tables towards the actor nodes. This reactive approach enables nodes to be involved in local control packet trafficking only. However, if every node needs info of 2 to 3-hop neighbour nodes (as is necessary with transmission power adjusting to skip nodes) a hefty memory demand is placed on scarcely resourced nodes. Compared to reactive protocols that are only involved with 1-hop neighbours, POWER-SPEED introduces a much higher amount of control packets, which may most likely be even higher than overhead produces by a proactive protocol.

Furthermore, as with the Anycast Tree-based Communication Paradigm and RRDSE, POWER-SPEED’s authors have also designed and evaluated the routing protocol by using the 802.11 standard MAC protocol designed for WLANs. The delay estimations within the POWER-SPEED routing algorithm are done with network nodes that are always switched on.

This means that if the POWER-SPEED routing protocol was to be implemented in conjunction with a MAC protocol that has been designed for low powered sensor nodes (i.e. employing duty cycles), the delay estimations will be incorrect since no MAC timing properties are considered. Consequently, as with the RRDSE, the delay results obtained by the POWER-SPEED authors cannot be evaluated against any other WSAN's simulations results obtained with simulations that make use of feasible MACL protocols that have been designed for low powered sensor nodes. Therefore, the performance evaluation by the authors is discounted since the routing protocol was designed and evaluated in conjunction with the highly energy-inefficient 802.11 WLAN MAC protocol.

There are however, several more issues that are not addressed by the authors. If the estimations of the delay experience by future nodes in a route towards the actor node are based on data from upstream nodes and not on recent feedback from downstream nodes, the estimation can be seriously incorrect when network traffic loads are variable and change due to unexpected events for example. The approach is favoured for networks that handle only periodic data gathering, where traffic loads stay relatively constant and can be better estimated. However, characteristically of WSANs, unanticipated critical events occur and unexpected data packets can be generated from sensor nodes in the event region, incurring additional traffic loads to the network. When the traffic load is increased in certain regions of the network, POWER-SPEED protocol will have incorrect estimation of delays for specific routes, which could result in packets not reaching the destination before its expiry time. Moreover, as with the RRDSE protocol, it is not clear how the data packet expiry demands are obtained. Nodes with a specific hop distances will need expiry times assigned to them according to their specific hop distance. This may impose a lot of complex calculations. Finally, any expiry times are usually determined within the APPL and not the NETL, which is responsible for the routing of packets.

Finally, the biggest problem with transmitter power control is that neighbouring nodes with MAC protocols that employ carrier sense operations might not be able to sense an occupied channel, resulting in high number of packet collisions throughout the network. A sensor node X that sends a packet to a geographically nearby next-hop node will use a small transmitter power that might not be sensed by other neighbours further away who might want to transmit a packet towards sensor node X at the same time. This scenario is known as the hidden terminal problem. Furthermore, when employing different transmitter powers, nodes that can be considered as direct neighbours differ. The initial network setup which seems crucial for the success of this routing approach is not explained by the authors. The setup fee and complexity might overshadow the performance gains for POWER-SPEED.

3.3.3.3.5 Scalable Source Routing

The non cluster-based Scalable Source Routing (SSR) protocol is proposed by Fuhrmann [36] for WSANs with churn. An overlay-like routing in a virtual network structure is combined with source routing in the physical network structure. In contrast to the hierarchical routing mechanisms, SSR does not group the nodes into clusters or subsets. SSR is mainly a reactive routing protocol, but it contains a proactive mechanism too, namely the construction of the virtual ring. The virtual structure is derived from the nodes' location independent addresses only. Thus, the virtual structure does not change when the physical network's topology changes. Each node bears a location independent address and has a given set of physical neighbours. Concerning the virtual structure, all node addresses can be viewed as forming a ring, as illustrated on the right hand side of Figure 3.11. In this virtual ring, two nodes with addresses A and B are direct virtual neighbours if there is no node with address C in the network so that A < C < B. This must take into account that the address space wraps around.

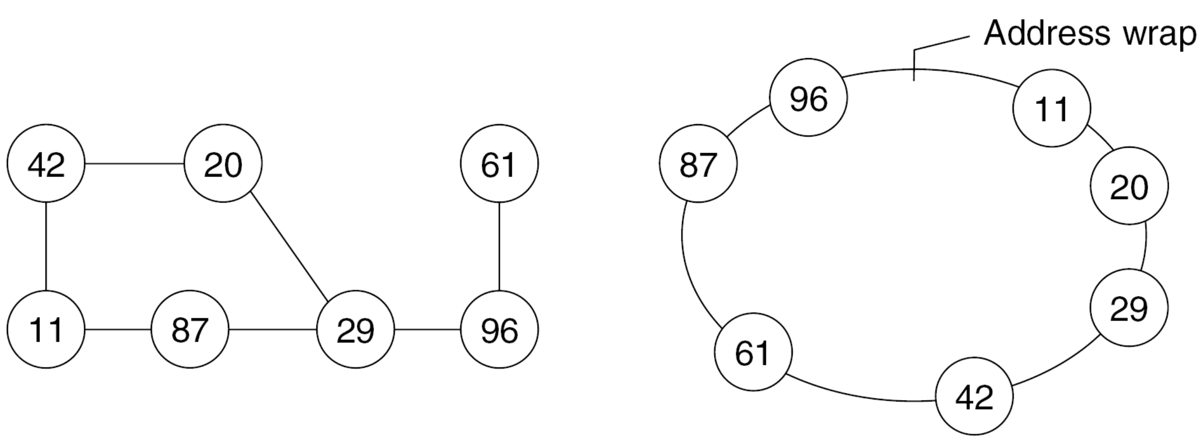

Figure 3.11 Example of a physical network topology (left) and the according virtual ring (right). From [36].

Both, the physical network and the virtual ring are equipped with a distance metric. In the physical network, distances are measured via the hop count. In the virtual ring, distances are calculated from the absolute address differences. The actual routing of packets by the SSP protocol combines source routes and intermediate look-ups. If a node cannot retrieve the entire source route to the destination from its own cache, it constructs a source route to some other, so-called intermediate node. The packet is then forwarded to that intermediate node, which in turn takes care of the packet and appends another source route to it. The selection of such intermediate nodes is governed by a three step process. 1. The next intermediate node must be virtually closer to the final destination than the current node. 2. Of all nodes satisfying the first condition, those nodes are chosen that are physically closest to the current node. 3. From these, the one is chosen that is virtually closest to the final destination.

It is noted that although the routing rule can produce source routes to any destination, these source routes will not necessarily be the shortest paths. SSR avoids the overhead of flooding at the expense of using potentially sub-optimal routes. This trade-off favours SSR when the amount of traffic that is exchanged for a source-destination pair is low. This is a disadvantage that might prove fatal for applications that have medium to high traffic loads. Furthermore, the protocol does not consider the very important WSAN real-time requirement. This protocol may have limited application relevance by not including any delay minimization.

This scheme ideally requires knowledge of all nodes in the network since ideally, the source routing should be able to compute the entire route from source to destination. Furthermore, it is not stated which actor node is chosen as the destination. This might lead to a sensor node choosing the wrong actor node which may be further away or not responsible for a certain event area.

3.3.3.3.6 Power-aware Many to Many Routing

The power aware many-to-many routing (PAMR) non cluster-based protocol, proposed by Cayirci et al. [37], is designed for WSANs that comprise nodes that have the same constant transmitting power level. By assuming the same constant transmitter power level, the authors state that routes with the same hop count will have the same energy consumption and end-to-end delay. The energy consumption might be the same if the distances between sensor nodes in a route are the same however, the delay will most probably not be the same. The delay depends on many factors such as traffic load and channel availability and is not directly related to the hop count only.

Actor nodes register their interest for data to the sensor nodes in the network by broadcasting a registration packet. While the registration packet is being disseminated through the network, the sensor nodes relaying the registration packet build up their registration table by inserting a registration record with the fields extracted from the registration packet. Sensor nodes derive a routing table from the registration table where there is a single record for every unique downstream node and sensing task pairs. Hence a many-to-many multicast tree is obtained for each sensing task. When a sensed data packet is received, it is forwarded to the downstream nodes contained in the routing table. The routing table contains entries for all routes to all interested actor nodes. No excessive calculations are performed in selecting a sensor node as the next hop. The packet is simply sent to the next nearest sensor node towards an actor node. Data packets are sent to all actor nodes registered for the specific sensing task. Actor nodes can deregister from a sensing task by broadcasting a deregistration packet resulting in sensor nodes updating their registration and routing tables accordingly.

A modified version of PAMR is also proposed by the same authors for WSANs that have nodes that can adjust their transmitting power level. An additional parameter that defines the energy consumption of a specific link is simply added to the routing table. The route with the least amount of energy is then chosen for the sending of data packets. However, the power control as with some abovementioned protocols is accompanied by problems concerning MAC carrier sense ineffectiveness and high overhead for mobile nodes.

The PAMR protocol forwards the sensed data generated by any sensor node to every actor node that is interested in that type of data, which may produce a lot of network traffic. Furthermore, the scalability of the protocol is questionable where larger networks may pose heavy strain on the routing table which is required to maintain all routes to several actor nodes. The limited amount memory space of sensor nodes might pose a serious problem when large routing tables are required for larger networks.

3.3.3.3.7 Geometric Broadcast Protocol

The Geometric Broadcast Protocol (GBP), proposed by Durresi and Paruchuri [38], contains a distributed algorithm where nodes make local decisions on whether to transmit based on a geometric approach. GBP does not need any neighbourhood information, implying low communication overhead. It handles seamlessly the presence of actor nodes, by using their richer resources to the advantage of resource constrained sensor nodes. The protocol employs separate sensor and actor node algorithms that handle the broadcasts. In a WSAN configuration as illustrated in Figure 3.12, the radio ranges of nodes determine the coverage area.

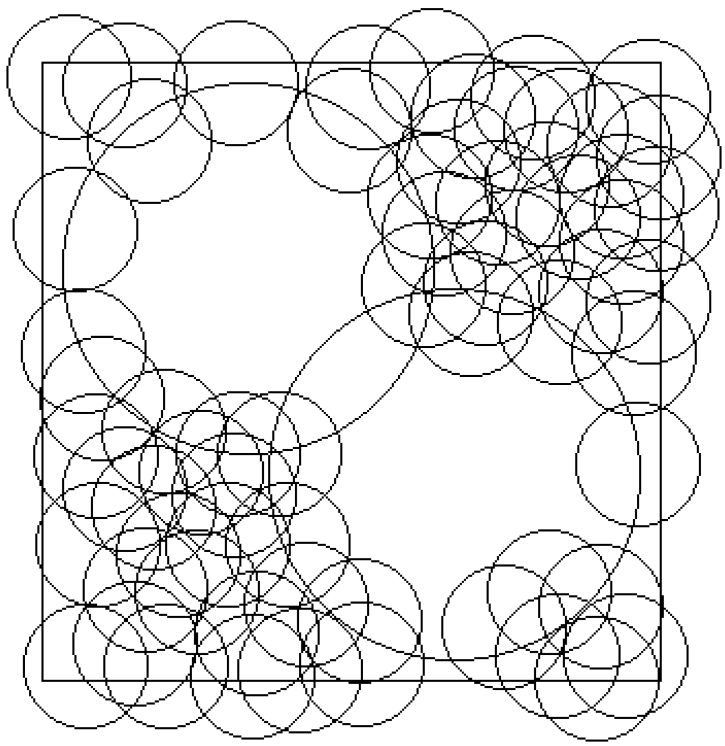

Figure 3.12 Broadcast in a WSAN with two actor and several sensor nodes. From [38].

It can be seen (Figure 3.12) that actor nodes have a large coverage area. After receiving a data packet the actor nodes broadcast the packet further to achieve maximum coverage area in reaching other actor and sensor nodes. On the other hand, the sensor node algorithm states that after receiving a broadcast packet, it first determines if it may be discarded. A broadcast packet may be discarded if: the packet has already been received, the sensor node is in an overlapping area of two actor nodes or a sensor node that is very close has already transmitted the packet. If none of these criteria is met, the sensor node waits for a while, after which it again checks if it has received the same broadcast packet. Finally, if it didn't receive the same packet it in turn broadcasts it further to cover the two-dimensional network area.

The authors state that GBP features a low number of packet retransmissions, signifying lower energy consumption by sensor nodes and faster transmission coverage of a given area. The protocol algorithms seem to only provide rules for broadcasting information in order to cover the whole network area rather than aiming for a specific region. This may render the use of BGP less feasible for WSAN applications. The network wide broadcasting of information may only sometimes be necessary for example when a query needs to be sent to all nodes in the network. Therefore, for most real WSAN applications the approach of GBP is less favoured, since WSANs generally favour a distributed approach in which small localized broadcasts may occur. Total network coverage is unnecessary. The frequent network-wide broadcasting of data packets may lead to traffic congestions and consequently reduced network lifetime.

3.3.3.3.8 Separate Communication Routing Network

A final paper aims to address the different resource constraints, quality of service and communication patterns of sensor and actor nodes, sensing and actuation are implemented as separate networks with distinct routing protocols and addressing schemes. This approach is proposed by Li [39] as part of a light monitoring and control application. The actor nodes do not forward any sensing traffic and vice versa. These two networks are joined at a central server that provides coordination and handles user requests as illustrated in Figure 3.13.

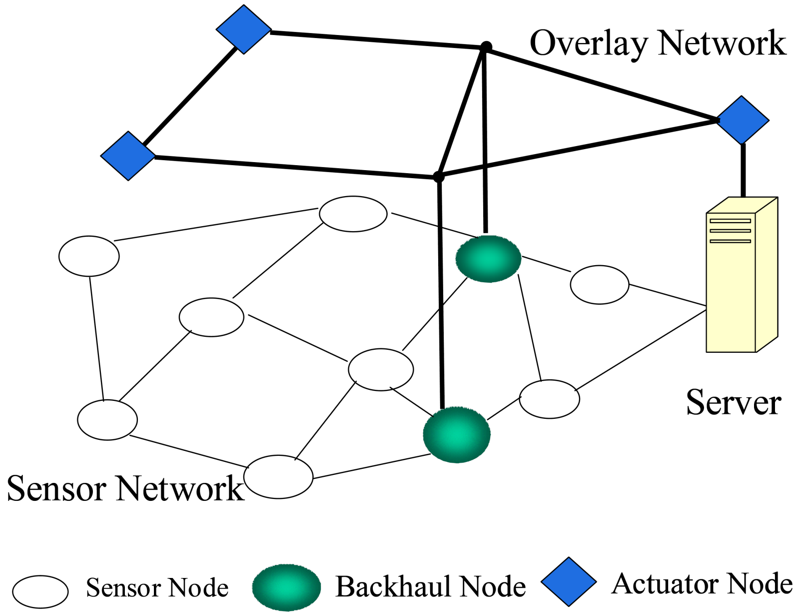

Figure 3.13 Heterogeneous communication architecture for WSANs. From [39].

This paradigm is introduced to curb the effects of connecting sensor and actor nodes in one homogeneous network. According to the author, the consequence of direct connection is interference between the sensing and actuation. When sensing and actuation packets are routed in the same network, actor nodes constantly need to filter out irrelevant periodic broadcast and data packets from the sensor node network. This filtering process consumes extra energy and leads to an increase in overall response time of the actor nodes. Therefore, communication related to sensor nodes is data-centric, low cost, low energy and unreliable. On the other hand, to guarantee packet delivery and minimize delay, actor nodes adopt more reliable, higher bandwidth, higher energy and more computational intensive communication.

The sensor network contains data centric routing architecture where broadcast is frequently used while the overlay network entails individual addressing with point-to-point communication for efficiency. The overlay network employs Ethernet or WLAN routing mechanisms and the two ‘networks’ are integrated and coordinated at the central server, where data collection and actuation decisions take place.

The overlay network is not restricted to actuation traffic. It is also used to deliver sensing packets that are critical or delay sensitive. For example, the detection of abnormal and dangerous conditions needs to be communicated to the server reliably and within a certain delay. To route this critical sensor traffic through the overlay network, the sensor network contains so-called backhaul nodes. These are resource enhanced nodes that can interface with both the sensor and overlay networks.

With this Semi-Automated approach, in which a central server gathers all the sensor data and makes actuation decisions, data has to travel a long way from sensor nodes far away from the server. This approach introduces a high traffic burden, resulting in rapid energy depletion, on nodes close to the server. For this reason most WSANs solutions predominantly employ an automated approach with distributed control to maximize the network lifetime. Furthermore, with the occurrence of backhaul nodes in addition to sensor and actor nodes, these backhaul nodes will have new specifications and requirements leading to an unnecessarily increased network design complexity. Strategic locations have to be selected for these special sensor nodes, which might not be as trivial as illustrated.

3.3.3.4 Literature Summary

Table 3.3 contains a summary of existing WSAN NETL protocol literature, encompassing the main features and identified problems pertaining to each NETL protocol.

	NETL Protocol	**Main features**	**Problems**
Cluster-based Hierarchical Approaches	**Hierarchical Geographic Clustering (HGC)**	▪ **Geographic routing** based on GAF protocol, towards cluster-head or actor node ▪ Novel three-level coordination model ▪ Fixed zones to form a virtual grid each with a cluster head ▪ Data aggregation tree towards actor node ▪ Actor location and time broadcasts	▪ The role of cluster-head should be rotated between sensor nodes to prevent rapid energy depletion of certain nodes ▪ Assumes very dense network ▪ The second level of coordination requires higher powered sensor node radios with high transmitter ranges
	Event-Driven Clustering	▪ **Greedy geographic routing** scheme used ▪ Involves several states of operation ▪ Clusters are created on-	▪ Not well suited for applications with multiple and simultaneous events occurrences ▪ Heavy overhead for applications with frequent or periodic events

Cluster-based Hierarchical Approaches		the-fly to optimally react to the event ▪ Only the event area is clustered ▪ Energy and real-time trade-off control	▪ Relatively complex
	Hierarchical, Energy-Efficient Routing with Ripple-Zones	▪ **Proactive single-path routing** along transmitter power weighted edges with local recovery for intra-zone communication ▪ Employs two approaches of direct transmission and multi-hop transmission, proportionally for inter-zone communication ▪ Self-organization of actor and sensor nodes into separate domains and zones ▪ Ripple-zones around each actor node with aggregation masters ▪ Formation of minimum spanning trees within a zone ▪ Employs transmitter power control	▪ Inter-zone communication requires higher powered sensor node radios with high transmitter ranges ▪ Aggregation masters (roots) may increase path length leading to increased power consumption and latency ▪ Suitable only for excessively large WSANs ▪ Real-time requirement and mobility of actor nodes have not explicitly been addressed ▪ MAC carrier sense operation, which may be used to alleviate the hidden terminal problem, ineffective with transmitter power control within zones ▪ Power control incurs high overhead for mobile network nodes ▪ Designed in conjunction with less suitable TDMA-based MAC protocol
	Clustering by Weighted Bi-partite Matching	▪ Formation of clusters for distributed control to solve problems locally ▪ Assignment of sensor nodes to clusters by weighted bi-partite matching	▪ Focuses on clustering only ▪ Method of routing data within a cluster not clearly stated ▪ Large network may induce heavy calculation overhead
Non Cluster-based Approaches	**Anycast Tree-based Communication**	▪ **Anycast proactive routing** with paths to several sinks/actor nodes maintained simultaneously ▪ Shared anycast tree approach with construction of shortest-path tree rooted at the source ▪ Sinks/actor nodes can dynamically join or leave	▪ Heavy traffic loads might be incurred if periodic updates flooded from sinks are too frequent ▪ Complex routing table updating ▪ Routing towards more than one actor node may lead to high congestion ▪ Designed and evaluated in conjunction with the inapt power inefficient *802.11* MAC

Non Cluster-based Approaches		the anycast tree ▪ Similar to Direct Diffusion WSN routing protocol	protocol designed for WLANs ▪ No consideration for the issue of reliable data delivery
	Reliable Reporting of Delay-Sensitive Events (RRDSE)	▪ **Anycast reactive routing** paradigm ▪ Next-hop delay and advancement calculation ▪ Makes use of data prioritization and delay bounds ▪ Pre-emptive priority queue ▪ Virtual grid with aggregation nodes for data aggregation	▪ High overhead with control packets adding to network congestion ▪ Little considerations for energy minimization ▪ Unclear how data prioritization is done within the NETL ▪ Unclear how latency bounds are obtained ▪ Delay calculations do not consider underlying MAC timing properties ▪ Designed and evaluated in conjunction with the inapt power inefficient *802.11* MAC protocol designed for WLANs
	Delay-Energy Aware Routing Protocol (DEAP)	▪ **Greedy geographic routing**	▪ Requires MAC protocol to cater for delay requirements ▪ Designed in conjunction with inefficient random wakeup MAC with variable wakeup duration dependent on load (queue size) ▪ No method for actor selection by border sensors ▪ No mobility of actor nodes considered
	POWER-SPEED	▪ **Reactive routing** by means of spatio-temporal estimation of forward route delay ▪ Delay estimation of downstream nodes by using historic upstream delay values ▪ Routing also considers total energy consumption ▪ Transmitter power adjusting to skip nodes towards an actor node in order to decrease latency ▪ Employs transmitter power control ▪ Requires only local	▪ Unexpected events might incur traffic load and compromise delay estimations calculated with historic data ▪ High complexity ▪ In order to skip nodes, routing tables containing info of 2 to 3-hop neighbours are needed, incurring high memory demands ▪ Upstream and downstream delays do not necessarily relate (different traffic loads, channel conditions, etc.) leading to false delay estimations ▪ Not clear how delay bounds are obtained

Non Cluster-based Approaches		control packet trafficking	▪ Designed and evaluated in conjunction with the inapt power inefficient *802.11* MAC protocol designed for WLANs ▪ Power control incurs high overhead for mobile network nodes ▪ MAC carrier sense operation, which may be used to alleviate the hidden terminal problem, ineffective with transmitter power control
	Scalable Source Routing (SSR)	▪ **Reactive routing** protocol ▪ Combined overlay-like routing in a virtual network structure with source routing in the physical network structure.	▪ A lot of overhead needed for additional virtual address ring ▪ No consideration for latency minimization ▪ Complex ▪ Works better for low traffic ▪ No explicit actor node selection algorithm
	Power Aware Many-to-Many Routing (PAMR)	▪ **Proactive multicast routing** of data packets to all interested actor nodes ▪ Many-to-many multicast routing tree formation ▪ Actor interest broadcasting ▪ Sensor nodes enable dynamic registration of interested actor nodes ▪ Employs transmitter power control	▪ Heavy traffic load introduced by sending packets to all actor nodes ▪ High overhead in updating routes and actor node interests ▪ Little consideration for latency minimization ▪ Poor scalability ▪ Power control incurs high overhead for mobile network nodes ▪ MAC carrier sense operation, which may be used to alleviate the hidden terminal problem, ineffective with transmitter power control
	Geometric Broadcast Protocol (GBP)	▪ **Geometric broadcast routing** to cover the entire network	▪ Less relevant for most WSAN applications ▪ Traffic overload may occur
	Separate communication routing networks	▪ Totally **separate** routing networks for sensor and actor nodes ▪ Backhaul nodes as mediators between the two networks ▪ Semi-automated approach	▪ Data has to travel a long way from sensor nodes far away from the server ▪ Backhaul nodes introduce unnecessary increased network design complexity ▪ No strategy for determining locations of backhaul nodes

Table 3.3. WSAN Network Layer Protocol literature summary.

3.3.4 Discussion

The CPS NETL has to deal with all aspects of the network topology and communication management. Some discussed protocols are very complex and require a lot of overhead related to communication and coordination between nodes inside the network. This might not be feasible for WSANs, which rely on the NETL protocol to relay critical data as fast as possible towards an actor node for rapid response. Frequent updating and calculating of route information may introduce intolerable delays.

The discussed protocols present the existing literature related to the NETL of WSANs. This area of research is still in its early phase and protocols that have been proposed do not necessarily improve on each other or on a specific base protocol. Several protocols are simply taken from the WSNs literature domain and modified and adapted to include addition WSAN requirements. Therefore, some protocols may seem very dissimilar.

The cluster-based hierarchical approach, originally derived from WSNs, provides a sensible solution for WSANs. It achieves partitioning and division of the network into clusters where latency, energy and reliability efficiency can be optimized locally, inside a cluster, to ensure a much more focused approach. Furthermore, by partitioning the network, the challenges of providing some time synchronization between nodes and data aggregation are also simplified.

Several routing protocols that do not use hierarchical clustering commonly induce an added communication load on the network. This added communication load is a consequence of overhead needed to ensure latency and energy efficiency of large indeterminate regions. With the absence of clearly defined well-sized domains, the task of optimizing the latency and energy efficiency is more complicated and limited. In the non-cluster-based routing protocol literature, it is often not clear how a sensor node will choose an actor node to forward the sensed data to. This results in an increased route discovery and next-hop selection process complexity, since more nodes may be involved in the route towards an actor node. The protocols that divert from using clusters often fail to adhere to the stringent delay requirement primarily posed by WSAN applications.

The NETL literature study revealed that several NETL protocols are designed in conjunction with the 802.11 MACL protocol. This WLAN MAC protocol is not at all suitable for use in the WSAN CPS. The use of the 802.11 MAC protocol will excessively shorten the network lifetime and undermine the feasibility of WSANs. Moreover, the data rate/delay calculations within the routing algorithms are done with nodes that are always switched on. Consequently, the delay results obtained by the authors cannot be compared to any other WSAN's simulations results obtained with simulations that make use of feasible MACL protocols. Therefore, the performance evaluation by the authors may be discounted since the routing protocol was designed and evaluated in conjunction with the highly energy-inefficient inapt 802.11 WLAN MAC protocol.

Another issue that is addressed by the NETL protocol situated on actor nodes in WSANs relates to the actor-actor communication. In addition to information gathering and actor-sensor communication, the NETL protocol of the actor nodes CPS is also responsible for the communication between the actor nodes. Some literatures propose the use of modified ad-hoc network routing protocol, improved for real-time requirements, for the actor-actor communication [2]. Due to the low number of actor nodes present in WSANs, routing data between actor nodes is simple and mostly peer-to-peer. Actor nodes directly communicate with each other via their powerful transmitters and do not need complex routing algorithms for data exchange. Elaborate discussions regarding the use of ad hoc routing protocols for actor-actor communication are beyond the scope of this dissertation.

WSANs NETL protocol research is still in a growing phase and when compared to the existing number of WSNs NETL protocols, WSANs literature still boasts only a small number of protocol designs.

3.4 THE MAC LAYER

The next layer in the CPS is the Medium Access Control layer. After the NETL protocol determines where to send a data packet, the MACL protocol provides access of the node into the medium by handling the connection establishment, maintenance and release of the data link to the destination node [18]. It enables point-to-point or point-to-multipoint connections between nodes inside the WSAN [8]. In the following subsections, primary MACL tasks and challenges are considered for WSANs, followed by a classification of WSN MAC protocols. Furthermore, the existing literature for WSN MAC protocols, current WSAN MACL protocols and MAC related standards are discussed in detail.

3.4.1 Primary Tasks and Design Challenges

The Medium Access Control layer protocols for WSANs have to adhere to several stringent requirements. The primary tasks and challenges for WSANs MAC protocols are listed as follows.

- The *communication channel has to be shared* among network nodes [8]. In WSANs, the communication channel consists of a wireless medium which can be access via the node's radio. The MAC protocol has to allow this wireless communication medium to be shared efficiently between nodes. It should determine when a node may access the communication medium. The use of multiple channels in the wireless medium may alleviate collisions that are typically experienced when using only a single channel. However, it imposes a nontrivial requirement on the hardware of the nodes in the network [2]. Therefore, a single channel is shared to retain the cost-effectiveness of sensor nodes.

- The *establishment of reliable communication links* for data transfer between nodes in the network has to be administered by the MAC protocol [8]. The links between nodes enables data transfer over the wireless medium from one node to another. Control messages should be exchanged between nodes for the reliable link establishment, since the successful data transfer depends on an uninterrupted link. Due to the numerous nodes in the network, a node may have several neighbours that may want to communicate with it. This leads to the possibility of collisions occurring due to multiple nodes trying to access the same node at the same time. These collisions prevent the establishment of a communication link. Therefore, for densely populated networks, the MAC protocol has to implement some kind of control that allocates links at relevant instances.

- The issue of *maintaining connectivity* inside WSANs is very important since network dynamics have to be considered and dealt with effectively [28]. The MAC protocol should feature adaptability to changes. Changes in network size, node density, and topology should be handled rapidly and effectively for successful adaptation. Some of the reasons behind these network property changes are limited node lifetime, addition of new nodes to the network, and varying interference, which may alter the connectivity and hence the network topology. The MAC protocol should gracefully accommodate such network changes [40].

- WSAN applications stress the need for *real-time communication* of data. The sensed data produced by sensor nodes have to be communicated to the actor node with a minimal latency [2]. The MAC protocol should take special consideration for the minimizing of the delay experience by a data packet form source to destination, since this may determine the feasibility of the WSAN. The earlier an actor node receives the critical event information, the sooner it can respond with an appropriate action. Energy efficiency and real-time communication are the main WSANs requirements posed by the applications.

- *Energy efficiency* should be one of the top priorities, in order to prolong the network lifetime [40]. Sensor nodes have a limited energy resource, which has to be used very sparsely to ensure that the nodes stay functional as long as possible to maintain network connectivity and feasibility of the network. The major energy consumer unit of a node is the radio. It can operate in four different states, which include transmit, receive, idle and sleep, where transmitting and receiving consumes the most energy followed by idling. The MACL protocol should provide fine-grained control of the transceiver unit, and allow switching of the wireless radio on and off. How frequent and when such switching have to be performed is the major goal of an energy saving mechanism of the MACL [42]. It should maximize sleep period of the node to conserve the most energy. The main energy related problems faced by the MAC protocol in a network environment with numerous nodes are identified by Karl et al.

[43] as *collisions*, *overhearing*, *protocol overhead* and *idle listening*. These problems have to be addressed and mitigated by the MAC protocol. In each of these problems the radio is used unnecessarily, which causes unwanted energy consumption.

When a collision occurs, the energy used to transmit and receive, by the source and destination node respectively, is wasted. Collisions are followed by retransmissions, incurring even more energy consumption. Overhearing occurs often in a wireless medium, where nodes for which data is not destined for overhear a data packet and subsequently drop it again. This wastes energy as with excessively induced protocol overhead where the MAC protocol uses a lot of control packets for its operation. Finally, idle listening takes place when a node is ready for incoming data packets, but is not currently receiving anything. In this idle state, most radios still consume significant amounts of energy [43].

- The *maximization of bandwidth utilization and throughput* should be considered by the MACL protocols. Vital energy and time may be wasted due to ineffective use of the available bandwidth.

- *Construction of MACL specific data packet*. The MACL packet contains different MACL parameters. The NETL frame should be encapsulated by the MACL packet. Once the payload is added and the header constructed, the packet is sent to the next layer in the CPS.

- The *protocol has to be simple* to allow easy implementation on computationally less powerful and memory constrained sensor nodes.

Due to the fact that the actor nodes in the WSAN are much resource richer than sensor nodes, MAC protocols design for actor nodes have a lighter energy consumption minimization task. The actor node MAC protocols implemented for actor-actor communication rather focus on achieving real-time communication than being concerned with energy-efficiency.

3.4.2 WSN MACL Protocol Classification

Unlike NETL protocols, very few MAC protocols have been specifically researched and proposed for WSANs. The reason for this is that MACL protocols have similar requirements for WSNs and WSANs. This signifies that WSN MAC protocols can be conditionally adopted by WSANs since the network as experienced by the MAC protocol are equal in many regards. The significant difference is that WSN MAC protocols do not consider the latency minimization as one of its top priorities. They are mostly predominantly focused on minimizing the energy consumption.

The categories of WSN MAC protocols as defined by the literature are the *schedule-based* protocols versus *contention-based* protocols [44]. Some literature also refers to these categories as random access vs. fixed-allocation protocols [43].

Although traditional MAC *schedule-based* protocols would employ multiple medium accesses through time, frequency or code division, only time division is applicable to WSN. Frequency division require too complex radio systems and with code division, large memory is required to support all the nodes' codes sequences, in addition to the complexity and the high cost of necessary radio circuitry [44]. This renders these two approaches non-viable for WSNs. In the Time Division Multiple Access (TDMA) based protocols for WSNs, nodes are often required to form clusters. The system time is divided into time slots and each of the nodes in a cluster has assigned its own time slot, and may access the shared medium only in this time slot. A TDMA-based protocol allows avoiding collisions, idle listening, and schedules sleep of the transceiver, without additional overhead. However, a TDMA-based approach also provides a number of drawbacks. If a node doesn't have any useful data to send during its allocated time, the time-slot is wasted. Latency will also be introduced because of waiting for that slot. Furthermore, the difficulty for the cluster to dynamically change its frame length and time slot assignments, in the event of node changes or node inclusions, contributes to poor scalability and poor mobility support. In addition, effective slot assignment in multi-hop networks is also challenging. Finally, the TDMA-based protocol requires a central element for managing schedules with high quality time synchronization since some clock drift may lead to disastrous consequences [42].

The opposing category of WSN MAC protocols are the *contention-based* protocols. These Carrier Sense Multiple Access (CSMA) type protocols allow nodes to independently access to the shared wireless medium. When a node wants to transmit some data, it will continuously sense the medium looking for a free channel for its transmission [44]. Only once the channel is free, i.e. no other node is currently transmitting, the node will commence transmitting its data. In this way, at any given time, only one node will transmit its packets. The contention-based protocols enable node independency that requires no external or central management. These protocols inherit good scalability, and support node changes and new node inclusions [42]. However, collision may occur due to contention for the single channel, leading to retransmissions incurring increased delays. In addition, nodes may be in idle listening mode for long periods, consuming scarce energy. A problem especially experienced by CSMA type protocols is the *hidden-terminal* problem [43]. In a problematic scenario illustrated in Figure 3.14, node A and B are within each other's radio range and node B and C are within each other's radio range. The problem occurs when node A starts transmitting a data packet to B, and node C, which cannot sense node A's transmission, senses a clear channel and also starts transmitting a data packet to node B. A collision occurs and both data packets are useless.

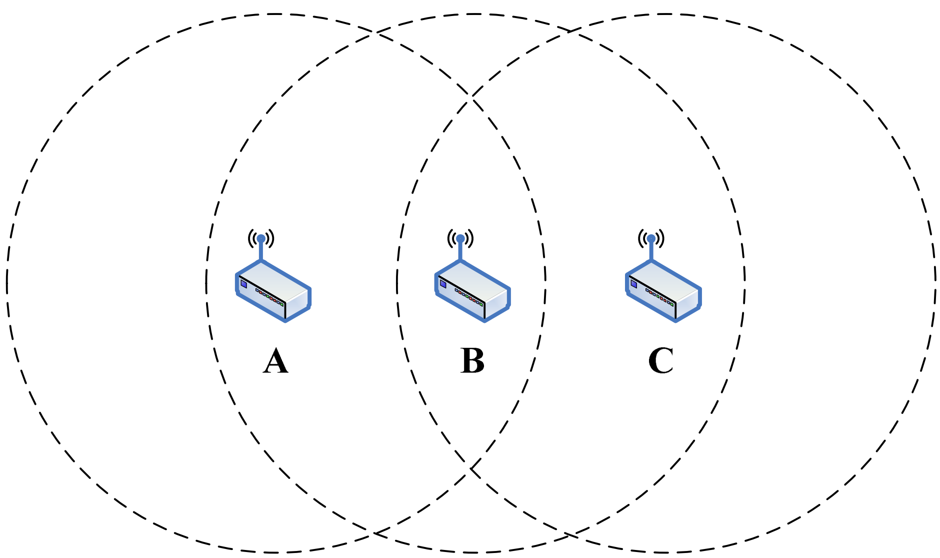

Figure 3.14 Hidden-terminal scenario (circles indicate interference and transmission range of network sensor nodes).

Both schedule-based and contention-based approaches have certain drawbacks and have been extensively researched. Literature is still largely divided over the selection of an appropriate approach for WSNs. The selection of a suitable approach is largely application dependent, due to the fact that they feature different expected communication patterns, bandwidth requirements, network sizes, mobility etc. [46].

To acquire a better understanding of the inner workings of existing protocols, some of most popular schedule-based and contention-based protocols are discussed in the following section.

3.4.3 Existing Literature on WSN MAC Layer Protocols

Many contention-based and schedule-based protocols have been proposed for WSNs. It is important to note that the requirement for a good MAC protocol (for WSNs) is not only energy efficiency, but also scalability and robustness against frequent topology changes [45]. In this section some popular proposed WSN MAC protocols are presented.

3.4.3.1 Contention-based Protocols

Based on CSMA, several schemes have been proposed as energy efficient MAC protocols for wireless sensor networks. They focus on improving CSMA in order to reduce the energy loss and sometimes minimize the transmission delays. Predominantly the focus is set on energy savings. To accomplish energy savings, they need to utilize low duty cycles to reduce idle listening, incorporate effective collision and overhearing avoidance, and minimize the required control overhead. The duty cycle of a node is the ratio of the listen period length to the wakeup period (the listen period plus the sleep period).

3.4.3.1.1 Sensor-MAC

Sensor-MAC (S-MAC), proposed by Ye et al. [47], sets out to achieve good scalability and collision avoidance by utilizing a combined scheduling and contention scheme while keeping energy efficiency as a primary goal. It provides mechanisms to circumvent idle listening, and overhearing as well. The basic idea of S-MAC is to use locally managed synchronizations and periodic sleep-listen schedules based on these synchronizations. The scheme of periodic listen and sleep is used to reduce energy consumption by avoiding idle listening.

Compared with TDMA schemes with very short time slots, the S-MAC scheme requires much looser synchronization among neighbouring nodes. All nodes are free to choose their own listen-sleep schedules. However, to reduce control overhead, neighbouring nodes are preferred to synchronize together. A node selects a schedule based on neighbour schedules resulting in synchronized nodes forming virtual clusters. Neighbouring nodes need to periodically send schedule updates among each other to prevent long-time clock drift. Updating schedules is accomplished by sending a short Synchronization (SYNC) packet that includes the address of the sender and the time of its next sleep.

To provide for both SYNC and data packets, the listen interval of a node is divided into three parts, as shown in Figure 3.15.

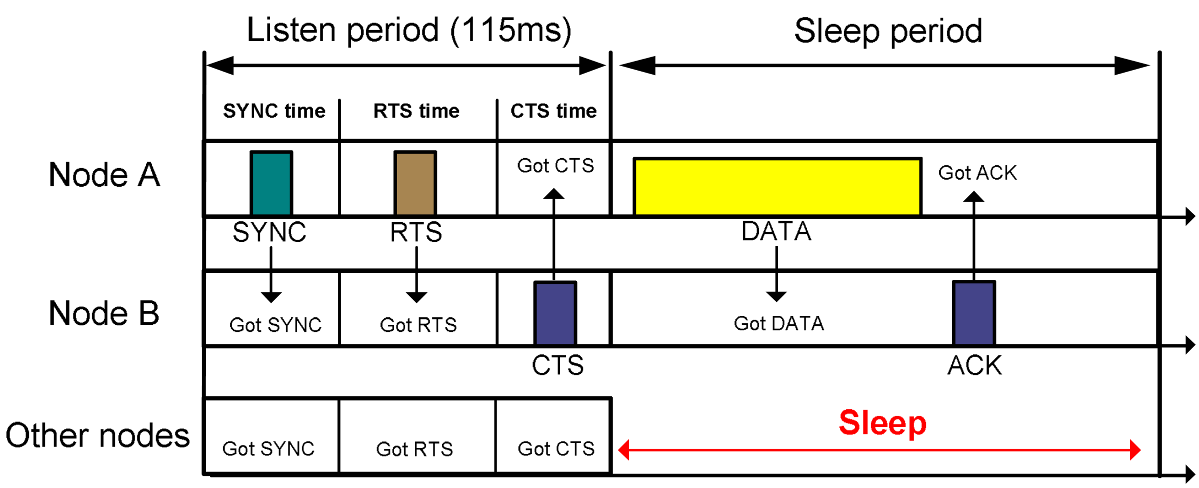

Figure 3.15 The basic operation of the S-MAC protocol. From [48].

The first part is for receiving SYNC packets. If node A wants to send a SYNC packet, it starts carrier sense when the receiver begins listening. It randomly selects a time slot to finish its carrier sense. If it has not detected any transmission by the end of the time slot, it wins the medium and starts sending its SYNC packet at that time. The second part is used for sending of data packets, which follow a Request to Send (RTS) / Clear to Send (CTS) approach. A RTS/CTS mechanism is adopted to address the hidden terminal problem. When sending a data packet from node A to B, S-MAC operates as follows. After node A sent its SYNC packet, it again senses the medium for activity. If it is clear, it sends a RTS packet. If node B receives the RTS packet successfully, it responds with a CTS packet. This indicates to node A that it can commence with sending the data. After the data has been received by node B, it transmits an acknowledgement (ACK) message to confirm successful reception of the data.

S-MAC also includes the concept of message passing, in which long messages are divided into fragments and sent in a burst. Only one RTS packet and one CTS packet are used and reserve the medium for transmitting all the fragments. Every time a data fragment is transmitted, the sender waits for an ACK from the receiver. If it fails to receive the ACK, it will extend the reserved transmission time for one more fragment, and re-transmit the current fragment immediately. Furthermore, the problem of overhearing is avoided by letting interfering nodes go to sleep after they hear an RTS or CTS packet. Since DATA packets are normally much longer than control packets, this approach prevents neighbouring nodes from overhearing long DATA packets and the following ACKs.

S-MAC provides a good scheme of introducing radio sleep periods to reduce energy consumption. Many subsequent protocols have taken S-MAC as a basis for development. A drawback however, is that when each node in the cluster follows its sleep schedule, there would be a delay experienced by the data packets at each hop. In this way, per-hop latency is introduced. Sleep and listen periods are predefined and fixed, which decreases the efficiency of the algorithm under variable traffic loads.

3.4.3.1.2 Timeout-MAC

The static sleep-listen periods of S-MAC result in high latency and lower throughput. To enhance these poor results under variable traffic loads, Timeout-MAC (T-MAC) is proposed by van Dam and Langendoen [49] to make use of an adaptive duty cycle. T-MAC uses the same SYNC, RTS/CTS and ACK control packets as S-MAC, but the duty cycle does not remain static. This is accomplished by allowing a node to remain listening and potentially transmitting, as long as it is in an active period. An active period ends when no activation event has occurred for a specific time TA. An activation event could be

- the firing of a periodic frame timer,
- the reception of any data on the radio,
- the sensing of communication on the radio, e.g. during a collision,
- the end-of-transmission of a node's own data packet or acknowledgement, or

- the knowledge, through overhearing prior RTS and CTS packets that a data exchange of a neighbour has ended.

Consequently, the time TA determines the minimal amount of idle listening per frame. The described timeout scheme moves all communication to a burst at the beginning of the frame.

The active period is adjusted dynamically with variable traffic load. Variable loads in WSNs are expected, since the nodes that are closer to the sink must relay more traffic and traffic may change over time. For homogeneous traffic loads however, T-MAC and S-MAC perform similar, however when traffic loads change a lot, T-MAC would provide much higher energy savings.

3.4.3.1.3 MS-MAC

In another enhancement to S-MAC, a Mobility-aware MAC protocol for Sensor networks (MS-MAC) is proposed by Pham and Jha [50]. This protocol takes the approach of S-MAC and enhances it for the use in WSNs that entail mobile sensor nodes. It is stated that S-MAC provides synchronization of neighbour nodes only every 2 minutes. If a node wants to setup a new connection with a new node in a different cluster, it has to wait for a new synchronization period to be able to detect the SYNC message from the new node. The authors point out that this waiting period of up to 2 minutes could be far too long for some time critical applications.

MS-MAC proposes a new scheme, where each node discovers the presence of mobility within its neighbourhood based on the received signal levels of periodical SYNC messages from its neighbours. The level of change in the received signals also predicts the level of the mobile's speed. Instead of storing only information on the schedule of the sender node as for S-MAC, the SYNC message in MS-MAC also includes information on the estimated speed of its mobile neighbour or mobility information. This mobility information is used by neighbours to create an active zone around a mobile node when it moves from one cluster to another cluster. In the active zone, nodes run the synchronization periods more often resulting in higher energy consumption, but the time it takes to create new connections is lower. Coordinating the active zone may require a lot of overhead resulting in higher energy consumption. However, mobility is managed very well by ensuring quick adoption of mobile sensor nodes to new locations with new neighbours.

3.4.3.1.4 DSMAC

The Dynamic Sensor MAC (DSMAC) protocol is proposed by Lin et al. [51], to use a dynamic duty cycle to better manage the trade-off between energy consumption and latency. The authors state that S-MAC does not perform well for delay-sensitive applications due to its static duty cycle. As with T-MAC, DSMAC makes use of dynamic duty cycles. DSMAC is able to dynamically change the sleeping interval with fixed listen interval length and therefore the duty cycle of sensors is adjusted to adapt to the current traffic condition. DSMAC alleviates the high latency problem presented in S-MAC when the traffic load is high, while still keeping the energy efficiency when the traffic load is low. This is done by only introduces insignificant overhead.

The DSMAC protocol attempts to adjust its sleep-wakeup cycle time dynamically according to the current energy utilization efficiency and average latency experienced by the sensor node. One-hop latency is defined as the difference time between a packet when it gets into the queue and it is successfully sent out. The value is recorded in the packet header by the sending sensor node and retrieved by the receiving node. The average latency of the receiving sensor is the average value of all one-hop latency values collected in the current SYNC period. This average latency value, serves as an approximate estimation of the current traffic condition. Now, when nodes' estimates suggest a high traffic condition, the duty cycle is doubled when the current energy consumption level is acceptable. By exactly doubling the duty cycle of the initially synchronized schedule, as shown in Figure 3.16, new schedule synchronization is unnecessary since the node is able to receive data double as often while the original schedule is still followed.

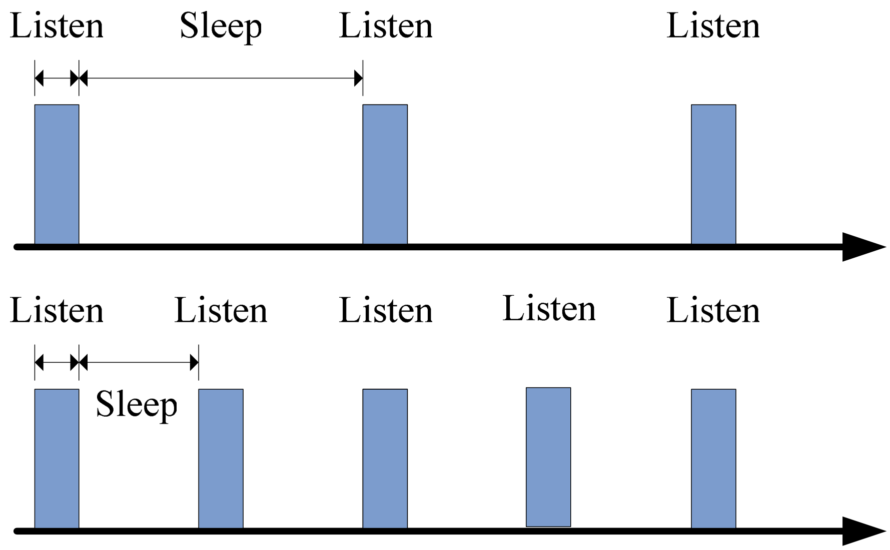

Figure 3.16 DSMAC duty cycle doubling.

The updated duty cycle is communicated to the upstream neighbours in the SYNC packet. The neighbours can send data more often, reducing the latency. Once the traffic condition subsides, the duty cycle is returned to its original value.

The latency observed with DSMAC is better than that observed with S-MAC. Moreover, it is also shown to have better average power consumption per packet.

3.4.3.1.5 WiseMAC

El-Hoiydi and Decotignie [52] propose WiseMAC, a single-channel contention protocol based on non-persistent CSMA. It combines non-persistent CSMA with preamble sampling to mitigate idle listening. Initially, all nodes in a network sample the medium (called the listen slot) with a common listen period T_W, but their relative schedule offsets are independent. If a node finds the medium busy after it wakes up (due to interference or incoming data packet), it continues to listen until it receives a data packet or the medium becomes idle again. The size of the preamble T_P is initially set to be equal to the listen period T_W to ensure that the destination listen slot falls within the source's preamble as shown in Figure 3.17. This is done only initially when the network is initialized or a new node is powered up and does not know its neighbour's wakeup schedules. When an event occurs, the source performs a carrier sense (CS) on the medium and starts the preamble immediately if the medium is free.

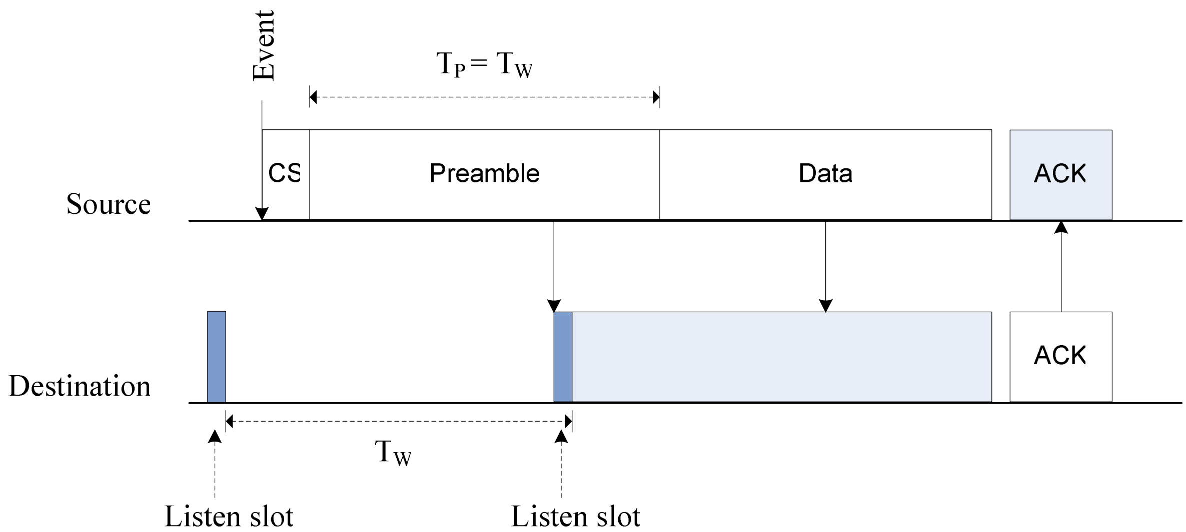

Figure 3.17 Initial operation of WiseMAC with preamble period (T_P) equal to the listen period (T_W).

However, to reduce the power consumption incurred by the predetermined fixed-length preamble (as in Figure 3.17) WiseMAC offers a method to dynamically determine the length of the preamble. The method uses the knowledge of the sleep schedules of the transmitter node's neighbours. The nodes learn and refresh their neighbour's sleep schedule during every data exchange as part of the ACK message. In that way, every node keeps a table of the sleep schedules of its neighbours. Since a node has only a few direct destinations, it can manage such a table even with limited memory resources. Based on the neighbours' sleep schedule tables, WiseMAC schedules transmissions so that the destination listen slot corresponds to the middle of the sender's preamble as shown in Figure 3.18. To decrease the possibility of collisions caused by that specific start time of a preamble, a short random time is added to the estimated preamble start time.

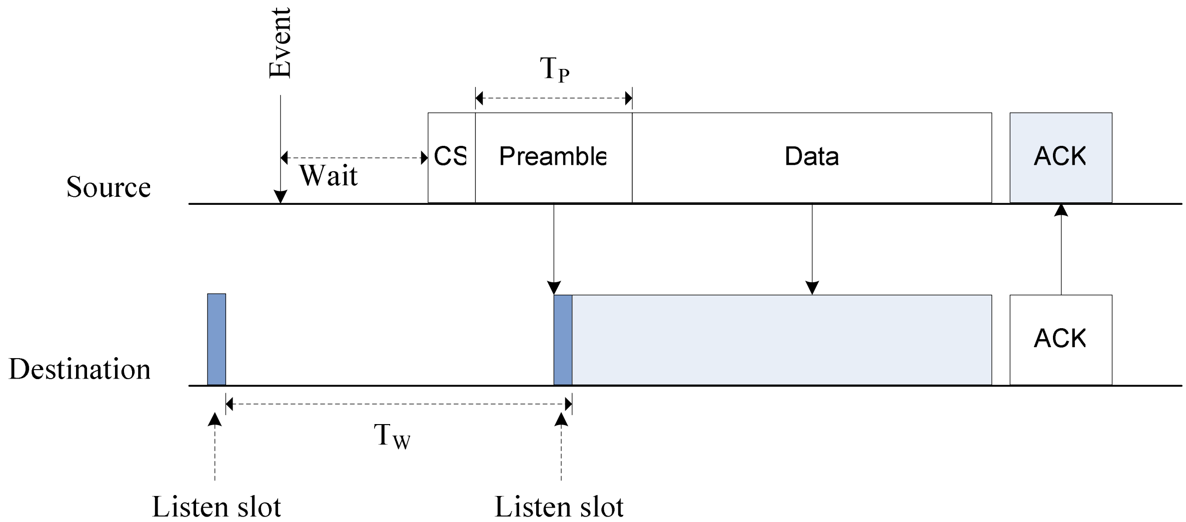

Figure 3.18 WiseMAC operation with known neighbouring node's listen schedules.

Another parameter determining the choice of the preamble length is the potential clock drift between the source and the destination. A lower bound for the preamble length is calculated with consideration to the clock drift Θ of both the sender and receiver. The sender starts sending a time of 2ΘL before the predicted wakeup time where L is the time since the last communication. The minimum length of T_P is therefore 4ΘL and the maximum is T_W.

According to simulations WiseMAC performs better than the S-MAC variant T-MAC [53]. This "Spatial TDMA and CSMA with Preamble Sampling" protocol entailing dynamic preamble length adjustment results in better performance under variable traffic conditions [40]. In addition, clock drifts are handled in the protocol definition, which mitigates the external time synchronization requirement. However, the problem of the hidden terminal scenario is not addressed by the protocol.

3.4.3.1.6 CSMA-MPS

The CSMA with Minimum Preamble Sampling (CSMA-MPS) protocol, proposed by Mahlknecht and Böck [54], was developed after WiseMAC, as an improvement. CSMA-MPS sets out to achieve increased energy efficiency by exploiting the opportunities found when combining the best features of two protocols, which include WiseMAC and the Sparse Topology and Energy Management (STEM) protocol.

The STEM MAC protocol, proposed by Schurgers et al. [55], uses a preamble that employs an alternating sequence of transmitting a small packet and listening for a reply from the destination. The preamble sequence is concluded once the destination receives one of the transmitted preamble packets and responds with an ACK. The source, which then knows that the destination listen slot was hit and is now available, can then respond with the actual large data packet. However, STEM does not use the knowledge of the listen schedule of the node to be contacted. WiseMAC, on the other hand, tries to hit the receiver's listen slot with the latest clock estimate it has obtained during the last communication. However, WiseMAC always sends out the worst case preamble length based on this estimate before transmitting the data frame.

CSMA-MPS uses the alternating transmits and receives of STEM, as illustrated in Figure 3.19, to ensure that preamble packets are not unnecessarily sent when the destination node is already awake and listening. Once the periodically listening destination node receives a preamble packet, it responds with an ACK indicating to the source that it can stop the sending of preamble packets and immediately commence sending of the data packet.

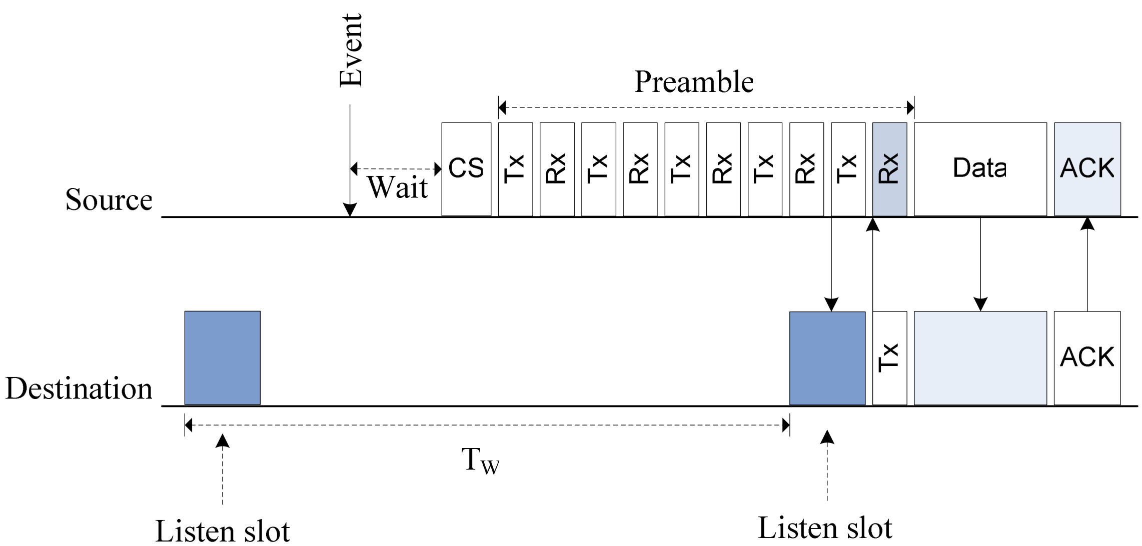

Figure 3.19 CSMA-MPS preamble sampling protocol with known neighbouring node's listen schedules.

This mechanism is combined with WiseMAC's approach of using neighbouring listen schedules to determine the optimum time to start the preamble, once an event has occurred. Moreover, using alternating transmits and receives allows to detect synchronization almost immediately without the need to send out a preamble with worst case length as with WiseMAC. The sender starts to preamble a time of 2ΘL before the predicted listen slot as with WiseMAC. However, the preamble is stopped as soon as the sender responds with a preamble reply and therefore the length of the preamble T_P can be significantly reduced.

3.4.3.1.7 B-MAC

Berkeley MAC (B-MAC), proposed by Polastre et al. [56], allows MAC specific parameters to be changed based on changing network conditions. The protocol aims at being implemented for use in not only a monitoring application, but also target tracking, localization and triggered event reporting. B-MAC duty cycles the radio through periodic channel sampling which the authors call Low Power Listening (LPL). It makes use of a preamble sampling technique with $T_P = T_W$, where each time the node wakes up it turns on the radio and checks for activity. If activity is detected, the node powers itself up and stays awake for the time required to receive the incoming packet. After reception, the node returns to sleep. If no packet is received, a timeout forces the node back to sleep. The authors clearly state that the sampling technique's success lies with the accurate sensing of the channel. For effective collision avoidance, it sets out to accurately determine if the channel is clear, by employing Clear Channel Assessment (CCA). Since the ambient noise changes depending on the environment, B-MAC employs software automatic gain control for estimating the noise floor. Once the noise floor has been determined from samples taken when the channel is free, future channel sensing is done by taking several samples and comparing it to the noise floor. Commonly, only one sample is taken, and therefore this approach provides a much more accurate channel assessment. Furthermore, B-MAC also provides optional acknowledgment support. If acknowledgments are enabled, B-MAC immediately transfers an acknowledgment code after receiving a packet.

B-MAC boasts a set of interfaces that allow services to tune its operation. Adjustments can be made to mechanisms that include CCA, acknowledgments, preamble length and sampling intervals, and LPL. However, B-MAC does not consider the effective approach of neighbour listen schedules used by CSMA-MPS to optimize preamble starting point and length. A detrimental effect is incurred when preamble lengths and sampling intervals are changed locally by a node. In changing the sampling interval of a node, a neighbour node may attempt to reach it with its original synchronized information. If the new interval is larger, the neighbour node may experience hefty difficulties in reaching the node. This necessitates the need for parameter changes to be communicated and stored by neighbour nodes. It has to be done effectively and incur minimum overhead.

3.4.3.2 Schedule-based Protocols

The second class of WSN MAC protocols is the schedule-based protocols. These employ TDMA schemes, which explicitly assign transmission and reception opportunities to nodes and set nodes to sleep at all other times [43]. A few protocols that are primarily based on TDMA schemes are given in the next sections.

3.4.3.2.1 LEACH

The Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol, proposed by Heinzelman et al. [57], partitions the network into clusters containing a cluster-head that is responsible for the TDMA schedule construction and maintenance. The cluster-head assigns time slots for each member node and broadcasts the schedule information to all member nodes. All members send their sensed values to the cluster-head which aggregates the information. The cluster-head then has to communicate directly with the sink to send the aggregated data. Since the sink may be far away, the cluster-head sensor node has to use a very high transmission power to reach the sink. Due to the high transmission powers required to reach the sink and the cluster-head requirement of having to stay active all the time, the role of cluster-head is rotated to prevent rapid energy depletion of the selected node.

The cluster-head is burdened with many computational and maintenance issues. Since cluster-head rotation is proposed within a cluster, all nodes have to have the capability to handle such role. Furthermore, when the role of cluster-head is shifted to another node, additional overhead is necessary and even more energy is consumed.

3.4.3.2.2 SMACS

The Self-organizing MAC for Sensor networks (SMACS), proposed by Sohrabi et al. [58][59], is a distributed protocol which enables a collection of nodes to discover their neighbours and establish transmission and reception schedules for communicating with them without the need for any local or global master nodes. The goal of SMACS is to detect neighbouring nodes and immediately assign a link with an exclusive channel. At the time of establishment each link will choose a channel at random. A channel could take the form of a frequency or a spreading code. The success of this approach hinges on the assumption that a large number of such channels are physically available.

The TDMA scheme boasts the use of a super-frame which entails time slots for each established link of a node. The super-frame is used for communication of local control packets and data and from time to time for the random access mode where nodes engage to search for new nodes to be absorbed in the network, or rebuild severed links. However, all nodes have the same frame length, which in turn requires strict synchronization. Another issue is the selection of a suitable length for the super-frame [43]. If the length is too short, some of a node’s neighbours may not be visible to it. On the other hand, highly dense networks may require a large frame length to accommodate all links to a node, but not all links may be used with low traffic loads and hence time slots are wasted.

3.4.3.2.3 TRAMA

The Traffic-Adaptive Medium Access (TRAMA) protocol, proposed by Rajendran [60], provides a collision free medium access approach to a single channel. Nodes are assumed to be synchronized and time is divided into random-access (signalling) and scheduled-access (transmission) periods. The random-access period is used to establish two-hop topology information, while the transmission slots are used for collision-free data exchange and also for schedule propagation.

TRAMA employs a traffic adaptive distributed election algorithm that selects receivers based on schedules announced by transmitters. Nodes broadcast their two-hop neighbourhood information and the transmission schedules periodically, specifying which nodes are the intended receivers of their traffic in chronological order, and then select the nodes that should transmit and receive during each time slot. In other words, when a data packet is to be sent, the protocol calculates the number of slots for which it will have the highest priority among two-hop neighbours within the period. The node announces the slots it will use as well as the intended receivers for these slots with a schedule packet. Additionally, the node announces the slots for which it has the highest priority but it will not use, allowing reuse of these slots.

A lot of calculation is needed to determine the data packet specific time slots and receivers. Furthermore, these schedules with calculated slot allocation have to be communicated to the receivers. This means a lot of scheduling information exchanges and for larger network this might incur heavy traffic loads. In addition non-trivial overhead is incurred for the process of gathering two-hop topology information of changing topologies. As opposed to S-MAC, the energy savings of TRAMA depend on the traffic load situation, while in S-MAC it depends on the duty cycle [43]. Furthermore, with TRAMA, delays incurred are found to be higher, as compared to those of contention-based protocols, due to a higher percentage of sleep times [40]. Finally, when considering dense networks, TRAMA may pose a significant node memory requirement to hold the two-hop neighbourhood information.

3.4.3.3 Literature summary and class comparison

Table 3.4 summarizes the features and identified problems of the MACL protocols given above.

	Protocol	Features	Problems
Contention-based	**S-MAC**	▪ Combined scheduling and contention scheme ▪ Virtual clusters ▪ RTS/CTS handshaking ▪ Uses SYNC packet before transmissions ▪ Fixed listen periods	▪ Per-hop latency may occur between clusters ▪ Sleep and listen periods are predefined and constant, which decreases the efficiency of the algorithm under variable traffic loads
	T-MAC	▪ Improves on S-MAC with adaptive duty cycle	▪ Overhearing
	MS-MAC	▪ Adds mobility-awareness to S-MAC ▪ Creates active zone around a	▪ Coordinating active zone not trivial and may introduce a lot of overhead

		mobile node to move between virtual clusters	
	DSMAC	▪ Improves on S-MAC with dynamic duty cycle ▪ Duty cycle doubling when traffic load is high	▪ Unregulated duty cycle changes, calibration needed to estimate the optimal change-over instant
	WiseMAC	▪ Combines non-persistent CSMA with preamble sampling ▪ Constant preamble length	▪ Some overhead is required to exchange schedules ▪ Always sends out the worst case preamble length based on latest clock estimate before transmitting the data frame. ▪ Preamble still continues after destination may already have received a valid preamble packet
	CSMA-MPS	▪ Alternating transmits and receives preamble sampling combined with WiseMAC's neighbouring listen schedules ▪ Preamble length and starting point optimized	▪ Some overhead is required to exchange schedules
	B-MAC	▪ Uses preamble sampling without neighbour listen schedules ▪ Constant preamble length ▪ Uses accurate channel assessment algorithm ▪ Allows parameters to be changed by higher layers, based on changing network conditions	▪ Without neighbour listen schedules, preamble length cannot be optimized ▪ Frequent parameter change incurs high overhead since these must be communicated to neighbours
Schedule-based	**LEACH**	▪ TDMA with complex rotating clusters	▪ High computational and maintenance burden
	SMACS	▪ TDMA super-frame with slots for each link of a node	▪ Determining the frame length not trivial, hefty penalties for unsuitable length
	TRAMA	▪ TDMA single channel with random-access and scheduled-access periods ▪ Per packet transmitter and receiver node determination	▪ Complex two-hop route calculation ▪ Complex election algorithm and data structure ▪ High overhead incurred by explicit schedule propagation ▪ High queuing delay

Table 3.4. MACL protocol summary.

Protocols have been discussed for both contention-based and schedule-based approaches. The schedule-based protocols can be configured to ensure that no collisions occur at the receiver and hence the hidden terminal scenario can be prevented. However, these protocols often require small clusters to be formed which are less adaptive to topology changes and limit the scalability of a network. Schedule-based protocols are more relevant for very high traffic applications. In low traffic networks, wasted time slots, due to no activity, presents a huge problem since scarce energy is wasted. Furthermore, the amount of synchronization and schedule distribution overhead needed is not to be under estimated. Clusters have to be constantly maintained and always kept strictly synchronized. Clusters are also limited in size to limit the time a node has to wait for its next time slot. Finally, the cross-cluster communication needed when data packets travel through multiple cluster towards the sink, presents a non-trivial task for the border nodes. These have to accommodate both schedules which may be totally uncorrelated, incurring delays.

On the other hand, contention-based protocols save energy due to the use of low duty cycles. The nodes have a very short periodic listen slot and sleep for the rest of the time. Duty cycles and listen intervals can be adjusted easily and dynamically by individual nodes to adhere to changing traffic loads. This is in contrast to schedule-based protocols, where schedule changes are very difficult to administer and has to be done by a central entity.

For densely populated multi-hop networks, schedule-based protocols have to either provide many clusters or accommodate more nodes into one cluster. By creating many clusters, increased delay is incurred since packets have to travel through multiple uncorrelated schedules. When accommodating more nodes into one cluster, on the other hand, the total time a node has to wait for its time slot is increased and again results in increased delays. Contention-based protocols are more scalable and handle topology changes much better. However, contention-based protocols have additional problems that have to be managed explicitly. These include the hidden terminal problem, overhearing and the occurrence of collisions at the receiver. To maintain good energy efficiency, some of the contention-based protocols employ the use of schedule exchanges. Virtual low maintenance clusters are formed and direct neighbour nodes exchange their wakeup schedules. In this way preamble sampling protocols can optimize preamble lengths and starting points to ensure minimum energy consumption.

A summary of the advantages and disadvantages of these two classes is given in Table 3.5.

	Contention-based	**Schedule-based**
Advantages	• Lower delays • Only loose synchronization needed • Scalable • Easy handling of topology changes • Simple design • Can be combined with scheduling to promote energy efficiency • Low duty cycles can save a lot of energy • Easy and dynamic changing of individual node duty cycles and listen interval	• Typically collision-free • No hidden terminal problem • Overhearing mitigated • High throughput in high traffic networks
Disadvantages	• Collisions may occur • Hidden terminal problem • High traffic may result in delays • Control packets incur some overhead	• Often requires complex clustering of network • Continuous cluster maintenance required • Do not react well to topology change • Limited scalability • Higher delays due to higher sleep times • Very strict synchronization needed • Low traffic incurs idle listening • Very inefficient and difficult to change schedules to accommodate changing traffic loads

Table 3.5. MACL protocol class comparison.

An investigation by [43] confirmed a well-known property of TDMA protocols stating that these have higher delays but also higher maximum throughput than contention-based protocols. Selection of a suitable class rests on the type of application. Traffic patterns, network density and rate of topology change need to be considered.

3.4.4 Existing Literature on WSAN MACL Protocols

There is a great number of existing MAC protocols proposed for WSNs of which only the most popular ones are discussed above. Many of these WSN protocols are suitable and applicable to WSANs. As mentioned before, very few MAC protocols have been specifically researched and proposed for WSANs. The reason for this is that MACL protocols have similar requirements for WSNs and WSANs. However, WSNs MAC protocols predominantly strive towards energy efficiency, while WSANs have an additional real-time requirement. To ensure the feasibility of WSANs, the delay experience by data packets, travelling from its originating sensor node to an actor node, needs to be minimized while maintaining energy efficiency and reliability.

In the WSANs literature, most proposals primarily involve the APPL and NETL. To date only two MAC specific protocols have been proposed for WSANs. The first MAC protocol is discussed in the following subsection (3.4.4.1). The second protocol (abbreviated as SEC) can, strictly speaking, not be regarded as a MAC only protocol and is mentioned in 3.4.4.2 together with other proposals that contain some relevant MACL considerations for WSANs.

3.4.4.1 CF-PMS

The Contention Free Periodic Message Scheduler (CF-PMS) TDMA MAC protocol for WSANs is proposed by Carley et al. [61]. A very complex technique is proposed to avoid the use of contention. The protocol uses a periodic message model with a contention-free message set. A contention-free message set has the property that messages are sent on release, and packets are transmitted to completion. With the contention-free technique, the schedule specifies one unique message that may access the medium at any time. The protocol determines when a message is to be run, independent of the other messages since they do not contend for the medium. This enables each node to schedule only the messages that it is concerned with. The protocol obtains a contention-free schedule by properly assigning message attributes of phase and period.

For this technique, some synchronization is needed. The synchronization is handled in two parts. First, loose synchronization is achieved with a periodic synchronization broadcast message. Second, assuming loose synchronization, a tight synchronization is achieved by having the receiver of a message begin listening for the message a constant amount of time t_A earlier than it expects the message. In this way nodes are allowed to be out of synchronization by as much as t_A without interfering with the protocol.

CF-PMS states that there is no waiting, queuing, or contention of ready messages within the scheduler. By implementing TDMA with a periodic scheduler, a space complexity linear in the number of messages that concern it is achieved. However, this protocol requires quite complex message schedule calculations with phase and period attributes, which is not necessarily easy to implement on sensor nodes. Furthermore, a substantial amount of memory is needed for these complex calculations if a node is busy with multiple messages. Due to the fact that sensor nodes have less powerful computational abilities and limited memory, this MAC protocol seems to be less suitable for use on hardware constrained sensor nodes.

3.4.4.2 Other MAC related proposals

Among the NETL WSAN literature discussed in section 3.3.3, some outline a MAC approach for WSANs. These are discussed as follows.

In the R-based clustering paradigm of the HGC paradigm [22] (explained in section 3.3.3.2), it is stated that a TDMA/CDMA MAC can be used to reduce inter-cluster and intra-cluster collisions. But TDMA will introduce a delay in reporting the time-critical data. Inside each grid, sensor nodes will elect one sensor node as cluster head to stay awake for a certain period of time, while the other sensor nodes only need to wake up periodically. Until an event or the reception of a query command from actors occurs, most sensors can remain in the sleeping state, with data from the few remaining sensors to provide coarse quality. Vital details to how the nodes are scheduled are not given.

Another routing protocol, DEAP (explained in section 3.3.3.3), bases the underlying MAC on a random asynchronous wakeup scheme [29]. The scheme allows for a node to be active during a randomly chosen fixed interval in each time frame. The idea is to have each node wake up once in every interval, be awake for a predetermined time, and then sleep again. If a node wants to send data, it simply starts transmitting and relies on a probability that at least one neighbour node is currently awake. The scheme is self-adapted to traffic load, where the duration of active period depends on the load level (queue size) of the node. This ensures lower delays. Also, it is observed that although the sensor nodes independently decide their active intervals and durations, the durations of neighbouring sensors are correlated. When a node experiences a high load, once it has contacted the next hop node, that node will receive all of its data packets and consequently will increase its active duration due to incurred higher load. However, this protocol relies on a dense deployment of sensor nodes and will actually incur higher delays for fragmented or less dense networks. Another problem arises if a node A has a high load, it will be awake for a longer duration and transmit all the packets to the next-hop node B. However, all other nodes trying to access the same destination node B will have to wait for node A to finish transmitting all of its packets. This may incur high delays for packets from other sources. Hence the random asynchronous wakeup scheme is only favourable for small low density WSANs with very few events and very low traffic.

Further two papers propose the use of a CSMA approach but handle the occurrence of collisions differently. Vedantham et al. [62] simply state that, when there are losses, retransmission is repeated until successful delivery is achieved. It allows a retransmission time of 1 second. Morita et al. [63], on the other hand, propose a redundant data transmission protocol for efficient data transmission in a ‘lossy’ and resource limited WSAN. This protocol assumes node to be always switched on and listen to the channel for activity. It proposes source nodes to broadcast their sensed values which are redundantly transmitted by multiple messages. When collisions occur, the destination actor node is still guaranteed to receive the message since it has been sent via multiple routes. Reliability is achieved at the cost of very high energy consumption due to always active nodes. This protocol does not seem feasible for use in WSANs.

Finally, a Sleep schedule for the radio of the nodes for fast and Efficient Control of parameters (SEC) MAC protocol is proposed by Vasanthi and Annadurai [64], but cannot be defined as a MAC only protocol. It relies on the APPL information of an event to determine its schedule. The MAC protocol calculates its TDMA scheduling according to the current application state and strives to achieve minimum latency for the reporting of important sensed data. This protocol is discussed in detail in section 4.4.

3.4.5 Related MACL Standards

To conclude this chapter of MAC protocols, relevant existing MAC standards for wireless networks are shortly mentioned. The most relevant standards include the IEEE 802.11, 802.15.1 and 802.15.4 [43].

The IEEE 802.11 commonly known as Wi-Fi, denotes a set of Wireless Local Area Network (WLAN) standards for physical and MACLs that are mainly used for wireless data transmissions between computers. It contains some energy-saving functionality but nodes are required to be always active and the standard is targeted towards high powered transceivers with high bit rates [43] that require orders of magnitude more energy than acceptable in low-bitrate sensor network applications. The 802.11 is an IEEE standard that allows devices such as laptop computers or cellular phones to join a wireless LAN widely used in the home, office and some commercial establishments. This WLAN standard is not suitable for implementation for resource constrained sensor nodes in WSANs.

Another standard presents IEEE 802.15.1, which is based on Bluetooth and includes MACL and PHYL specification. The nodes are grouped into piconets that contain one master and up to seven active slaves. The master polls the active slaves constantly, which requires them to be switched on all the time. The standard is inappropriate for the resource constrained sensor nodes in WSANs. Sensor nodes would be required to accept the role of master and slave incurring considerable complexity [43]. Furthermore, with densely deployed sensor nodes, the standard would be regarded as useless since only a limited number of active nodes are accepted by the master. A large number of master nodes would be required resulting in a decreased network lifetime.

A more applicable standard is the IEEE 802.15.4 standard which is aimed at low-rate wireless personal area networks. This standard also includes a MACL and PHYL but takes a positive approach to achieve energy-efficiency. It is aimed at a variety of applications that include amongst others at WSNs, home automation and security, home networking and connecting devices to a PC [43]. This might suggest a less focused approach. The MAC protocol is a combination of schedule-based and contention-based schemes and proposes the use of different roles taken on by nodes. Reduced function devices are directly connected to a full function device that boasts significant higher resources. This could be applicable to WSANs, with sensor nodes as the reduced function devices and actor nodes as the full function device. However, the number of sensor nodes is much greater that of actor nodes in the network, implying that sensor nodes cannot all be directly connected to an actor node. The full function device acts as a coordinator. In WSANs, the actor node, acting as the coordinator, may not effectively reach sensor nodes situated further away with four or five intermediate sensor nodes between them. However, the standard's combined schedule and contention approach suggests that advantages of both can be successfully reaped when combined correctly.

3.4.6 Discussion

This section provides a detailed overview of the most important MAC protocols that have been proposed for WSNs. In the very young WSANs literature, only one MAC specific protocol has been proposed while only a few other papers, that are more concerned with the NETL, briefly outline an underlying MAC approach. This is due to the fact that WSN MACL protocols could be almost directly adopted by WSANs. However, there is a need for these protocols to be modified and improved to consider the dominant real-time requirement of WSANs [2].

The main MAC protocol classes as discussed are contention-based and schedule-based approaches. As with WSNs, selecting a suitable MAC class for WSANs is not an obvious choice. The choice seems to be application dependent. However, most strictly schedule-based protocols are very complicated and computational intensive. They are complex to maintain in a multi-hop network, due to their strict time synchronization requirement. They are less suited for event-based operation, as they cannot increase the resource utilization due to their reservation schemes [64]. Furthermore, these protocols are not well suited for networks that experience topology changes and mobility of nodes. On the other hand, strictly contention-based protocols may suffer from the added overhead of control packets and have to deal with the occurrence of collisions and the hidden terminal problem.

The advantages and disadvantages of both approaches have to be thoroughly considered when choosing a suitable one for WSANs. The research conducted on existing WSN MAC protocols provides a good background for making an informed decision. There is hence a need for further research on MAC protocol designs, specifically for WSANs.

3.5 THE PHYSICAL LAYER

The lowest layer in the CPS is the PHYL. In WSANs the connection between network entities is a wireless connection and hence the PHYL functionality is performed by a radio. RF transmissions are used to establish short-range wireless links between nodes by making use of the ISM bands. The responsibilities of the PHYL are frequency selection, carrier frequency generation, signal detection, modulation, and encryption [8].

While designing the PHYL for WSANs, *energy and latency minimization* assumes significant importance, over and above the decay, scattering, shadowing, reflection, diffraction, multipath and fading effects [8]. WSANs sensor nodes are required to use small-sized, low-cost and ultra low power transceivers [65]. The importance of choosing suitable transceivers for energy constrained sensor nodes is marked by the fact that the radio is the primary energy consumer. The radio can be either in transmit, receive, idle or sleep mode, with the sleep mode being the least energy consuming. The less time the radio is busy with transmitting, receiving and idling, the longer the sensor node’s lifetime. The main control of the radio is done by the MACL, which issues the sleep and wakeup commands.

Actor nodes, on the other hand, have much richer energy resources and depending on the application, need much more powerful radios to enable a much larger communication range than sensor nodes. Actor nodes may be required to communicate over longer distances with other actor nodes during collaboration efforts towards effective acting.

Several important radio specifications are to be considered when choosing a suitable radio. These are discussed as follows.

3.5.1 Hardware Considerations

Since the PHYL functionality is performed by the radio, certain hardware specifications have to be taken into account when choosing a suitable radio. Specifications that are considered important and relevant for this dissertation include the following.

- Transmission power. The transmission power determines the range of communication that can be achieved by a node. Some radios enable dynamic changing of this parameter to allow nodes to adjust the communication range.

- Data rate. The data rate determines the speed at which data can be sent between nodes. This plays an important role in minimizing latency and is used to determine MACL timing parameters. Chipcon's CC2400 [13] radio uses typical bit rates of 10 kbps, 250 kbps or 1 Mbps.

- Radio turn on time. The quicker the radio can switch on, the sooner data transmissions can occur. This parameter has to be taken into account for MACL timing issues.

- Transmit (TX) / Receive (RX) switching time. The radio can either transmit or receive at any one instance. This parameter determines the time it takes to switch the radio from transmit to receive mode. This parameter is used by some MACL preamble sampling protocols to time the tx/rx preamble sampling. It can operate quicker if this parameter is smaller.

- Sensitivity. The sensitivity defines up to what incoming signal power level data can be successfully received. Another value that may be used is the carrier sense threshold, i.e. the power level by which the channel is deemed busy. This carrier sense threshold is set lower by up to 10dB.

- Carrier frequency. The carrier frequency determines the distance a radio can transmit based on a certain power level. The trade-off between antenna efficiency and power consumption limit the choice of a carrier frequency for the transceivers to the ultrahigh frequency range [8]. Popular radios make use of bands at 433 MHz, 868 MHz, 915 MHz and 2.4 GHz. Transmitting at higher frequency bands requires more power for the same distance but allows higher bandwidth. The higher bandwidth permits shorter transmitting time and smaller antennas.

3.5.2 Discussion

The nature and complexity of the PHYL processing is an important consideration in selecting a PHYL technology for wireless networked nodes. Given the characteristics of WSANs, trade-offs can be made with reliability, range and data rate. The PHYL should have this flexibility to allow latency and energy minimization [65]. Therefore, as part of the PHYL, the selection of a suitable radio plays an important role to enable latency minimization and promote energy efficiency in WSANs. For the scope of this dissertation, only the choice of radio with suitable parameters, as given above in section 3.5.1, is considered relevant.

3.6 CHAPTER SUMMARY

In this chapter, the communication architecture for WSANs is presented. It consists of a CPS with 4 separate layers each with its own tasks and challenges, while all layers strive to achieve the overall WSAN objectives. Primary tasks and challenges have been presented, followed by an examination and evaluation of existing WSAN literature, and where applicable WSN literature, of each layer in the protocol stack. With exception of the PHYL, existing protocols have been discussed to provide insight into the current existing research.

For the Application and NETL, protocols specifically designed for WSANs have been discussed and problems faced by each protocol given. For the MACL, relevant WSN MAC protocols are examined and evaluated since almost no WSAN literature exists on the MACL specifically. It is noted that WSN MACL protocols may be adopted by WSANs as long as the real-time requirement of WSANs is considered a top priority alongside the energy efficiency.

From the literature review, it is observed that the research field of WSANs protocols is still very young. It is noted that since WSANs have only recently come into existence, very little protocols exist in comparison to WSNs. WSAN protocols that do exist, are very application dependent and are therefore very dissimilar. There is hence a need for a more generic CPS that would cater for a variety of application needs. The aim of this chapter has been to provide a background of existing WSANs and WSNs protocol literature in order to facilitate research into a complete WSANs CPS design. The CPS design should strive to achieve the overall WSANs objectives.

Chapter

4 THE CROSS-LAYER PARADIGM

The literature review of chapter 3 revealed that existing protocols proposed for WSANs, and where relevant for WSNs, strive to achieve high efficiency with regards to latency, energy and/or reliability. These WSAN and relevant WSN protocol designs are designed for a specific layer and attempt to achieve high efficiency within that layer. The protocols are stacked according to the layer it is designed for within the strictly layered CPS. However, this strictly layered approach, as proposed by the OSI reference model, has recently been overhauled by a new paradigm. This paradigm has recently attracted increasing attention of researchers and communication system designers that involves using a cross-layer approach in optimizing network efficiency. This wave of attention stems from the large and active community dealing with challenging networking environments such as mobile ad hoc networks, next-generation cellular networks, or sensor networks [66]. The new paradigm promises to achieve higher efficiencies and increased performance compared to the traditional layered communication architecture [2].

In this chapter, the new cross-layer paradigm is introduced and different frameworks are discussed. Furthermore, existing WSNs cross-layer literature is examined and evaluated to give an overview of the existing research that has been conducted.

4.1 THE CROSS-LAYER APPROACH

The strictly layered communication architecture of the OSI reference model was designed to structure the communication within networks by providing a modular design to allow different aspect of a network communication to be developed separately. The layered architecture provides several advantages [66][67] that include,

- a reduced design complexity due to well-defined functional entities,
- improved maintainability due to the modular nature,
- facilitates parallel engineering,
- lowers development cost,
- ensures interoperability, and
- a reasonable degree of flexibility, since layers function independently of each other.

This layered design, in principle, allows for arbitrary combinations of protocol classes [66]. Specific layer protocols can be selected and combined to support a specific type of network or a specific application.

However, this strictly layered approach is considered sub-optimal and inflexible for several types of networks that are faced with stringent constraints [2]. It results in poor performance for networks like mobile ad hoc networks, next-generation cellular networks, WSANs and WSNs that have to deal with challenging networking environments. Poor performance is especially experienced when energy is a constraint or the application has high bandwidth needs and/or stringent delay constraints [68]. To meet these stringent requirements an approach is required that supports adaptability and efficiency optimization of the CPS.

A new approach that has recently attracted increasing attention, introduces a notion that instead of treating each stack layer as a completely independent functional entity, information is shared among the CPS layers. This information can be used to adapt protocol functionality in the presence of changing networking conditions and stringent constraints. Specific decision processes such as route selection and other algorithms use information from other layers as input. This creates a possibility to create new kinds of adaptive applications such as multimedia applications, which are sensitive to changing networking conditions [66]. The ability to share information across layers is the central aspect of the cross-layer design approach. The cross-layer design approach is not a total replacement of the original layered architecture, but instead cross-layering can be seen as an enhancement of the layered approach. The ultimate goal is to preserve the advantages of a layered architecture and in addition to allow for performance improvements and a new form of adaptability with cross-layer interactions.

The authors in [66] note that the cross-layer approach is specifically targeted at networks with dynamic networked environments such as ad hoc networks and sensor networks, where there exist a lot of intrinsic performance bottlenecks and an obvious need to adapt to rapidly changing conditions. These make use a wireless communication medium that is shared and interference-prone, resulting in bit errors, packet collisions, high and varying delays, and an overall lowered throughput. Aside from the difficult communication aspects, devices in sensor networks have limited energy resources and are computationally less powerful. Furthermore, the existence of mobile nodes introduces another significant system dynamic. The mobility can affect the stability of routes and if not handled effectively the re-establishment of routes may consume large amounts of scarce resources such as bandwidth and energy. In such dynamic, resource constraint environments, cross-layer approaches are promising since performance and scalability can significantly be improved.

The cross-layer design benefits are evident for example in the event of network congestion in WSNs. In a strictly layered architecture, the MACL reacts locally by exponential back-off to reduce the traffic, while the APPL reacts by lowering the transmission rates of sensor nodes. When these two layers act independently from each other, inefficiencies are caused due to the duplication of functions. By utilizing a cross-layering approach, each protocol layer may share its data with other protocol layers to avoid inefficiencies [2].

4.2 CROSS-LAYER FRAMEWORKS

To facilitate and manage the sharing of information across multiple layers in the CPS, a framework is needed. This framework is implemented with the protocol stack and determines how the information is shared, and how optimization takes place. For example, some implementations may employ a dedicated optimizing entity, while others simply allow information sharing between key layers. Several relevant framework designs have been proposed and are presented as follows (some are adopted from [66]).

4.2.1 WIDENS

As part of an ad hoc communication systems design for public safety, emergency, and disaster applications, Wireless Deployable Network System (WIDENS) [69] is a framework that provides state information and parameter mapping between *only adjacent layers* in the CPS, to increase protocol re-configurability and adaptability, as illustrated in Figure 4.1. The cross-layer information is utilized in the case that in-layer optimizations cannot prevent performance degradation. The problem of adaptation loops is avoided by only allowing interactions between adjacent layers. Information from nonadjacent layers can only be accessed through the mapping functions of an adjacent layer. This way, unnecessary and unintended cross-layer operations are avoided.

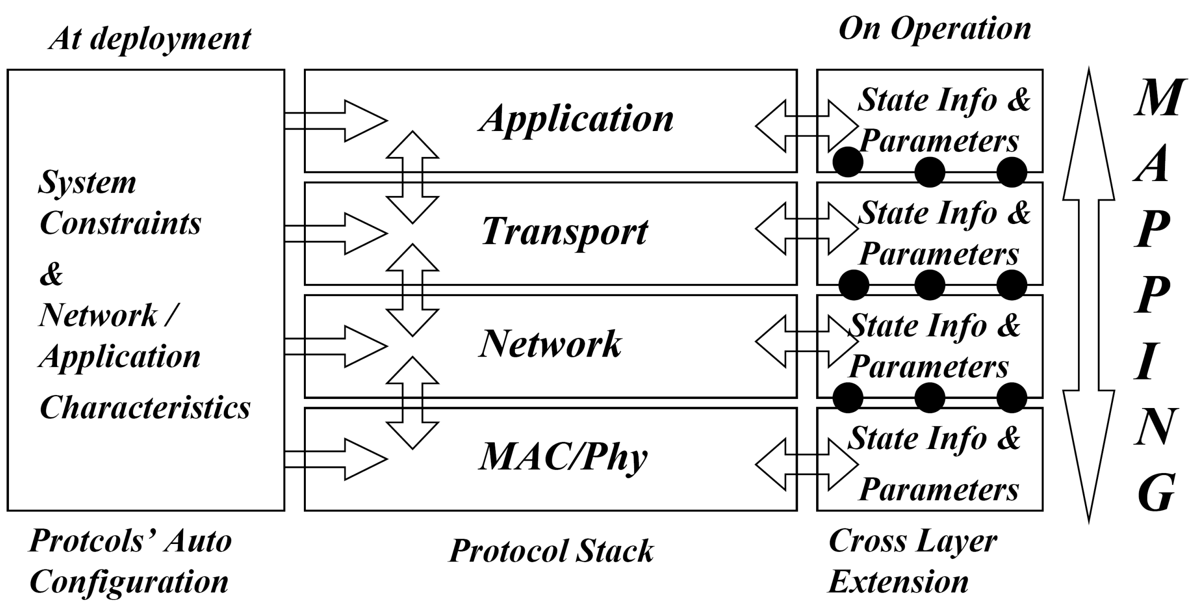

Figure 4.1 WIDENS cross-layer framework. From [69].

4.2.2 Information Management Entity (MobileMan)

In a metropolitan ad hoc network design MobileMan [70], a framework is proposed that entails an *information management entity* called Network Status which is accessible by each layer, as illustrated in Figure 4.2. This management entity is responsible for storing and organizing protocol data. The insertion and the access to that data are controlled by standardized procedures to achieve layer separation. Unlike WIDENS, this framework allows all layers to fully utilize data from all other layers. Protocols that have not been designed to consider data from other layer can still function within the framework. Since those protocols do not add information to the Network Status and do not use the information available, the optimization potential is unutilized but the protocol stack remains fully functional.

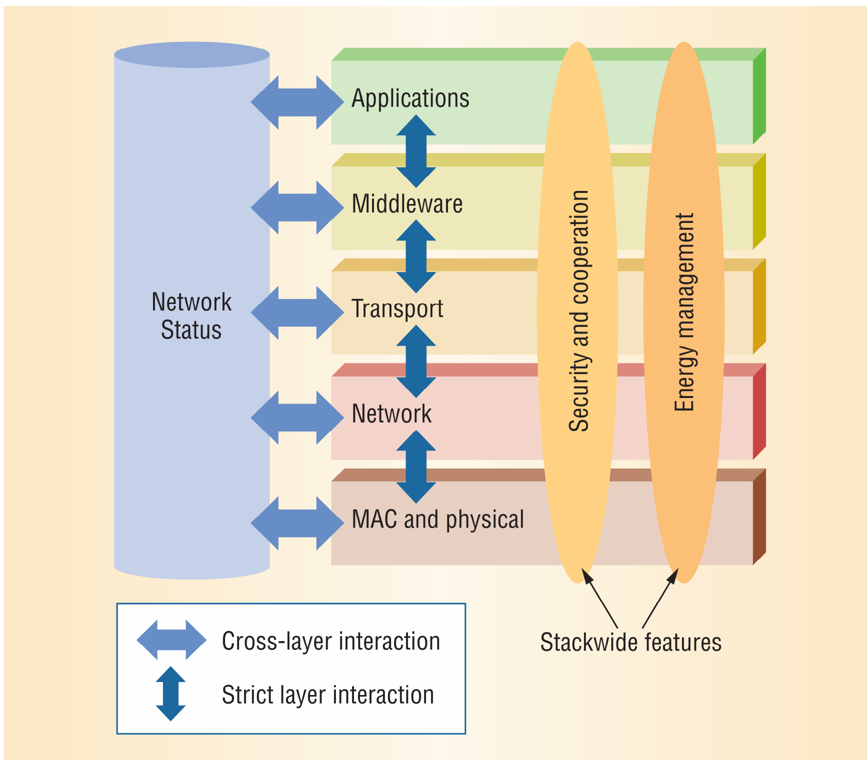

Figure 4.2 MobileMan cross-layer framework with Management entity. From [70].

4.2.3 Global and local adaptations within the protocol stack

Another framework design, specifically for mobile ad hoc systems processing multimedia data, involves a cross-layer interaction which differentiates between two kinds of adaptations or optimization, namely global and local. *Global adaptations* are triggered and handled by a resource manager, which chooses the optimal configuration of each layer at a node to adhere to application needs and utilizes the least resources. *Local adaptations*, on the other hand, only take place within a layer which responds to minor changes in the system as long as the application requirements are not violated and the pre-allocated resource utilization is not exceeded. In this framework however, in contrast to MobileMan, the resource manager chooses the optimal configurations for the layers. The layers provide the information and wait for a response from the resource manager rather than choosing the optimal configuration themselves. This approach requires special calibration of the configurations that are sent to the layers to provide optimal efficiency.

4.2.4 Global and local adaptations within the network (CrossTalk)

In a framework called CrossTalk, proposed by Winter et al. [66], *two data management entities* exist as illustrated in Figure 4.3. The first entity provides a local view of a node and is responsible for the organization of locally available information such as current battery status, load, neighbour count, signal-to-noise ratio (SNR), transmit power, location information, or velocity. This information can be provided and access by each layer of the protocol stack to conduct local or in-layer optimizations. The second entity establishes a network-wide or global view of the same type of information collected in the local view. A data dissemination procedure is provided to gather local information of other nodes in order to construct the global view. Unlike the global view of the framework in section 4.2.3 which is defined as the layer-wide view of a single node, this global view entails information of other nodes in the network. This framework introduces heavy overhead and traffic needed for the construction of the global view, since other nodes' information needs to be gathered consistently. Furthermore, CrossTalk does not provide any structured access of information between layers like in WIDENS. CrossTalk's decision processes are only supported based on the global view of relative state information. Every layer accesses the global view at will and is responsible for efficiency optimization within its layer.

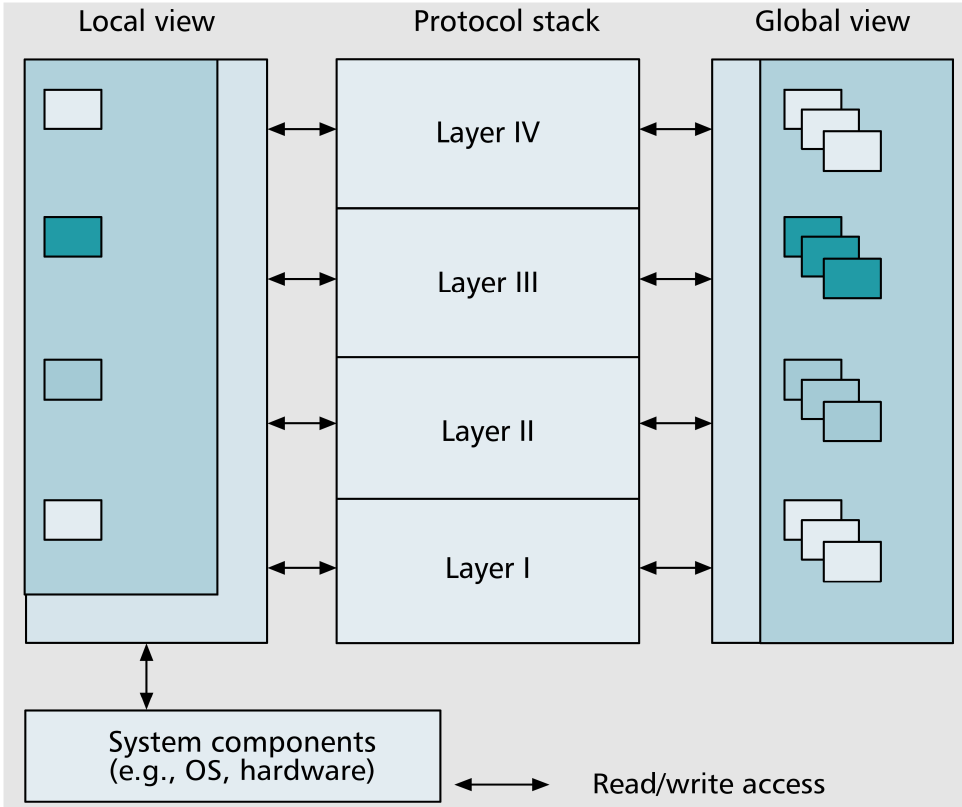

Figure 4.3 CrossTalk cross-layer framework with local and global. From [66].

4.2.5 Dedicated Cross-Layer Optimizer

A further framework, proposed by Khan et al. [71], uses cross-layer optimization for mobile ad hoc multimedia applications as illustrated in Figure 4.4. The cross-layer optimization concept consists of three steps. The first step involves *layer abstraction*, where an abstraction of layer-specific parameters is computed. The number of parameters sent to and used by the *Cross-Layer Optimizer* (CLO) is significantly reduced by the abstraction process. Parameters are divided into three categories: *Descriptive* parameters can be read by the CLO, but cannot be tuned; *directly tunable* parameters can be set directly as a result of the CLO, for example the time slot assignment in a TDMA system; and *indirectly tunable* parameters which cannot be set directly as a result of the CLO but may change as a result of setting directly tunable parameters. Next the CLO finds the values of layer parameters that optimize a specific objective function. Finally, the CLO distributes the optimal values of the abstracted parameters to the corresponding layers. It is the responsibility of the individual layers to translate the selected abstracted parameters back into layer-specific parameters and actual modes of operation. These steps are repeated at a rate that depends on how fast the application requirements and transmission capabilities of the physical medium vary. This CLO approach is similar to the framework in section 4.2.3, but here the parameters are categorized and reduced in number by the abstraction process.

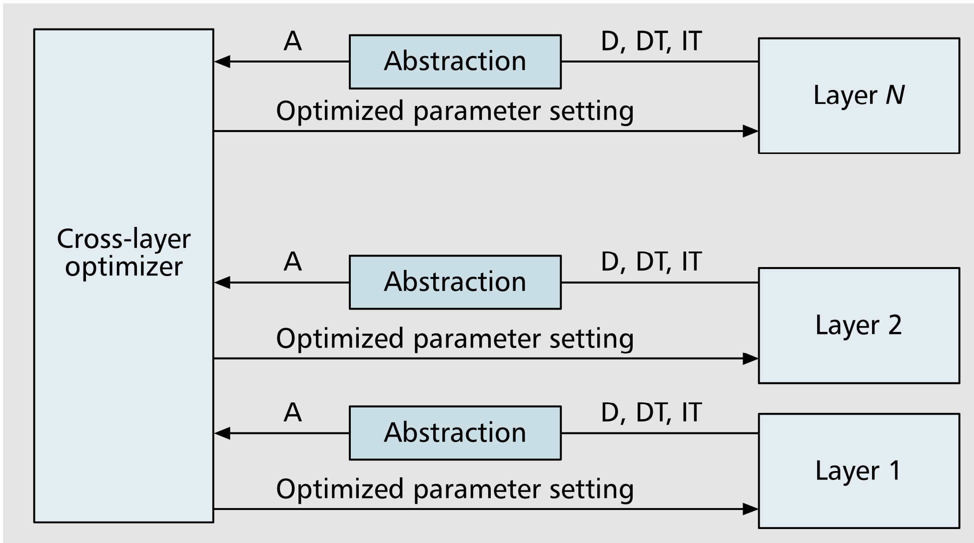

Figure 4.4 Application-driven cross-layer framework for video streaming (parameter types: A-abstracted, D-descriptive, DT-directly tunable, IT-indirectly tunable). From [71].

4.2.6 Optimization towards Common Metric

A simple framework imposing a *less generic approach*, utilizes data from different layers to optimize protocol behaviour towards certain metrics like energy efficiency without going through the trouble of describing an underlying architecture. This may pose problems of adaptability to other protocols who do not strive towards efficiency of the same metric.

4.2.7 A Unified Cross-Layer Module

A unified cross-layer model (XLM), proposed by Akyildiz et al. [72], is developed to achieves efficient and reliable event communication in WSNs with minimum energy expenditure. XLM melts common protocol layer functionalities into one cross-layer module for resource-constrained sensor nodes.

This framework implements the cross-layer paradigm by completely doing away with the layered CPS and replacing it with one single block that contains all communication functionality and features. A major drawback of this approach is the lack of modularity. The CPS traditionally features a layered structure to facilitate and promote separate layer development. If all these layers are melted into one block, the entire block has to be considered comprehensively every time a feature or functionality needs to be developed. Rather than taking a single layer (which are layered according to logical tasks) and developing this layer, the complete protocol block needs to be understood and taken into consideration. This makes the task of developing a single aspect of a network communication very difficult and tedious. By changing something insides the block, another related functionality may be disrupted or even fail.

4.2.8 Cross-Layer Optimization Interface Framework

A WSN cross-layer framework, proposed by Merlin et al. [73], uses a Cross-Layer Optimization Interface (CLOI) as a repository for information, which may be needed by one or more protocol layers. CLOI maintains this information through two structures, a neighbour table and a message pool and it supports services that will fill these data structures either once or continuously, depending on the information. As illustrated in Figure 4.5, CLOI is placed between the routing and MAC layers for two reasons. First, its location allows the interface to retrieve much of the information sent from the node onto the network as well as many incoming packets. The second reason is that it offers potential for abstraction of the link layer. The MACL and PHYL do not have a global vision of the network and cannot provide enough information about its state for automated use with CLOI. Furthermore, CLOI has no authority to make any routing, node activation, or medium access decisions. CLOI simply acts as an interface to the protocols in the stack, allowing them to access common yet important information about the node and its neighbours that can be used to optimize the protocols' performance.

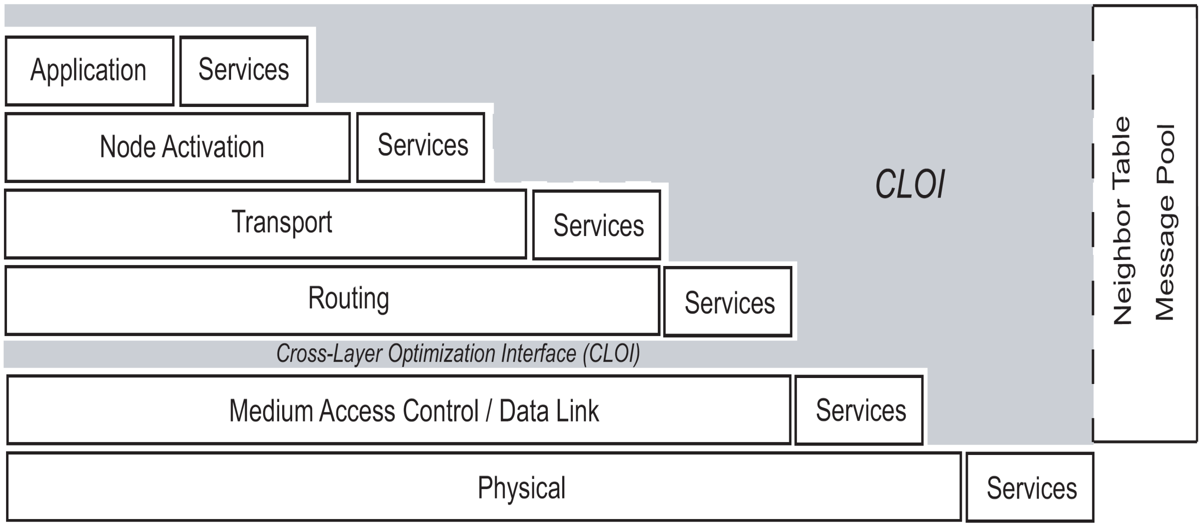

Figure 4.5 Framework with Cross-Layer Optimization Interface. From [73].

In order to make the best decisions, cross-layer schemes need up-to-date information from the node and its neighbours. For this purpose, the authors propose using an information vector piggy-backed to packets by CLOI. Finally, to ensure that various protocols do not modify the data structures in an uncontrolled manner, CLOI provides input and output functions with a fixed syntax.

4.2.9 Dynamic Multi-Attribute Framework

Finally, a Dynamic Multi-Attribute Cross-Layer Design (DMA-CLD) framework, proposed by Safwat [74] also for use in WSNs, uses schemes that are called Energy-Constrained Path Selection and Energy-Efficient Load Assignment. These employ probabilistic dynamic programming techniques and utilize cross-layer interactions between the network and MACLs. As illustrated in Figure 4.6, the DMS-CLD is implemented at the NETL and information is interfaced from the Application, MAC and Physical layers. This framework however, only features unidirectional layer interaction, where all layers feed into the DMA-CLD layer to enable only the NETL to perform optimization. This presents a limited cross-layer approach.

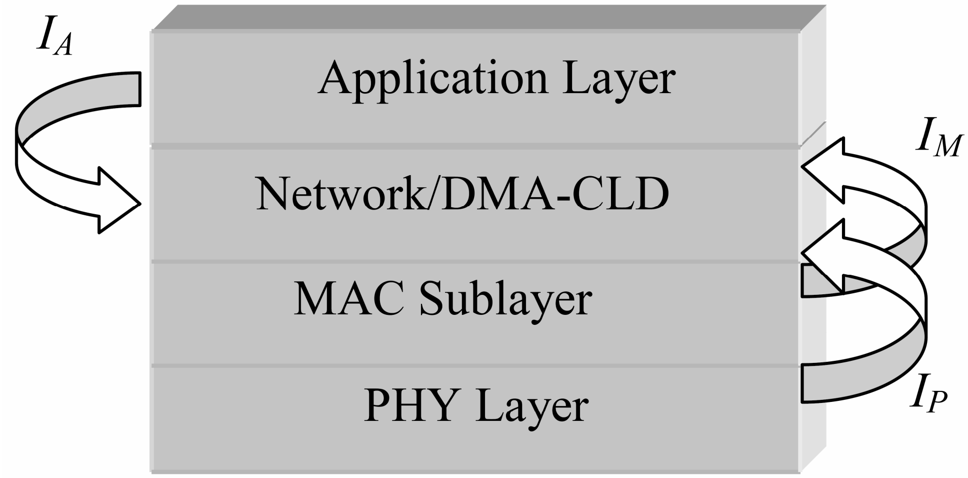

Figure 4.6 Dynamic Multi-Attribute Cross-Layer Design framework (I-interfaced, A, M, P - layers). From [74].

4.2.10 Discussion

These frameworks differ in complexity, memory usage, amount of overhead and often target applications. These aspects are very important and have to be considered thoroughly when applying a cross-layer approach to the CPS of a specific type of network and application. The list of frameworks presents a good overview of some implementation possibilities of the cross-layer paradigm. With exception of the last 3 frameworks, the frameworks are mostly designed for use in mobile ad hoc networks. However, it is noted that ad hoc networks differ from WSANs and WSNs in that they comprise only resource rich nodes. WSANs and WSNs have sensor nodes that are severely resource limited with respect to energy, memory and computational capabilities.

To exploit the benefits of a cross-layer design, resource limited networks such as WSANs and WSNs have to implement the new paradigm in such a way that maximum performance is gained while incurring the minimum strain on the nodes. The framework has to remain simple and incur as little overhead as possible. Moreover, care should be taken that improvements performed in a specific layer will not cause impairments and be counterproductive for other layers [75].

To the best of the author's knowledge, no literature of a cross-layer framework implementation exists for the WSAN network type, which is the network type of interest in this dissertation. Only a single MAC protocol proposed for WSANs exists, which employs a hint of cross-layering by considering the type of data packet created by the APPL. There is hence a need for further research into the benefits of utilizing a cross-layer approach for WSANs, which have very stringent latency, energy and reliability requirements.

For WSNs, which of the existing network types are most similar to WSANs, several proposals that use some kind of cross-layer interactions exist. Existing literature on WSN cross-layering is categorized to give an overview of different implementations and adaptations of cross-layering in WSNs. Finally, the singularly existing WSAN MAC protocol that employs some cross-layer interaction is given and evaluated.

4.3 EXISTING LITERATURE ON WSN CROSS-LAYERING

WSNs are most similar to WSANs and existing literature on cross-layering for WSNs provides some insight into approaches that are followed to implement the paradigm in resource constraint networks. Several different implementation approaches are followed to realize cross-layering within a protocol stack. The predominant goal of employing cross-layer interactions in WSNs CPS is to improve the energy efficiency to maximize the network lifetime. These implementation approaches can be categorized according to interactions or modularity among physical, MAC and NETLs [76]. In the next subsections, the existing WSNs cross-layer literature is categorized according to the layer combination that employs some cross-layer interactions between specific layers. Performance gains of the cross-layer approach of some implementation proposals are given to confirm the benefits of utilizing the cross-layer paradigm.

4.3.1 Physical and MAC Layer Interaction

The first implementation classification, which is used less in the WSN literature, is the use of cross-layer interaction between the PHYL and the MACL. In an implementation proposed by

Wang et al. [77], a cross-layer approach involves a novel MACL access-deferring scheme based on the received signal power. The received signal strength of the PHYL is used by the MACL's most progress forwarding algorithm, in which the node farthest from the sender re-transmits the packet first to reduce the collision possibility and redundancy of retransmissions. This proposed implementation is used for efficient flooding and achieves 20-50% better energy efficiency compared to blind and probabilistic flooding.

In another implementation [78], the IEEE 802.15.4 standard is improved to mitigate the impact of channel impairments. The MAC protocol back-off strategy is regulated by the wireless link quality, by taking into consideration the error performance of the underlying wireless links which is measured by the PHYL. The performance evaluation shows that the cross-layer 802.15.4 design outperforms the basic 802.15.4, achieving up to 69% higher goodput, is more efficient, imposes significantly lower load and delivers up to 154% more bits of information per energy unit. These improvements come at the expense of a 7-18% decrease in the fairness among contending individual nodes, compared to the basic 802.15.4.

4.3.2 MACL and NETL Interaction

The interaction of MACL and NETL is a popular cross-layer approach in WSNs. In an implementation by van Hoesel et al. [75], nodes are assigned an active or passive state by the MACL energy conservation scheme. This MACL state information is in turn used by the NETL to successfully determine a route which only uses active nodes. Without this information, the NETL would try to route data to passive nodes, which would not be able to receive any data packets. The cross-layer design is compared to the combination of the well known Dynamic Source Routing (NETL) and S-MAC (MACL) protocols. The cross-layer design achieve much higher network lifetime especially in WSNs that contain some mobile nodes.

In another proposal by Jurdak et al. [79], a cross-layer framework is presented for optimizing global power consumption and balancing the traffic load in WSNs through greedy local decisions. The framework enables each node to use its local and neighbourhood state information to adapt its routing and MACL behaviour. It employs a flexible cost function at the routing layer and adaptive duty cycles at the MACL, in order to adapt a node's behaviour to its local state. A node's state is a representation of its current number of descendants in the routing tree, its radio duty cycle and its role. The node's role provides information on its functionality within the application. However, the routing scheme employs a proactive approach, which best suits only static network nodes. The implementation framework is compared to the standard B-MAC protocol and achieves significant lower overall energy consumption.

In a further implementation by Ha et al. [80], Sense-Sleep Trees are constructed by combining minimum spanning tree scheme of the NETL with the sleep schedules of the MACL. This is done to overcome overhearing and packet collisions to ensure energy efficiency and reliability. A drawback of this implementation is that every Sense-Sleep tree has its own schedule and only one schedule can be active at any given time, preventing simultaneous data gathering from all nodes within the network.

At the MACL of an implementation proposed by Rossi et al. [81], every node accesses the channel according to its own cost by means of properly defined cost-dependent access probabilities. Costs are used to capture the suitability of a node to act as the relay and may depend on several factors such as residual energies, link conditions and queue state. The cost-aware MAC discriminates nodes in the channel access phase by therefore assisting the forwarding decisions to be made at the routing level. Nodes with high costs are ruled out from channel contention and are not considered when making routing decisions. This provides the routing layer with better relay candidates and, at the same time, decreases the number of in-range devices contending for the channel, thereby reducing interference. The proposed MAC scheme is coupled with routing over hop count coordinates by exploiting first and second order neighbourhood information.

Finally, a Cross-Layer approach to an energy efficient sensor Network (CoLaNet) is proposed by Chou and Chuang [82], which features in addition to the MAC-NETL interaction, the inclusion of the APPL information. CoLaNet takes the characteristics and requirements of the applications, routing in the NETL, and the data scheduling in MACL into consideration. Specifically, the characteristics of the APPL are used to construct the routing tree and then the routing and MAC issues are formulated into a vertex-colouring problem. By solving that problem, CoLaNet is able to determine a data transmission schedule for each sensor node.

This design sets out to achieve energy efficiency and provide good system performance. Compared to the S-MAC protocol with different duty cycles, CoLaNet boasts the best average energy consumption and end-to-end delivery time for small WSNs with 2-15 nodes. However, the design introduces high setup and maintenance cost, since the sink determines the routing trees and MAC schedules of every node inside the entire network. This could incur a lot of traffic in dense WSNs, which may compromise the end-to-end delay.

4.3.3 Physical, MAC and NETL Interaction

A few cross-layer schemes exist that include three protocol stack layers to generally interact towards achieving energy efficiency in WSN and maximize the network lifetime. In a proposal by Madan et al. [83], the optimization of transmission power, transmission rate, and link schedule for TDMA-based WSNs is proposed. Interference between links is measured at the PHYL and the MAC scheduling is adjusted to schedule highly interfering links at different times and the lower interference links together. Furthermore, the energy consumption measured at the PHYL is used by a multi-hop routing scheme that incorporates load balancing.

Another 3 layer joint cross-layer implementation is proposed by Cui et al. [84], in which the slot length of a variable-length TDMA scheme is optimally assigned according to the routing requirement while minimizing the energy consumption across the network. The optimization problem is solved by using convex optimization methods. It considers energy consumption of both transmission energy and circuit processing energy. Based on the analysis by the authors, it is shown that single-hop communication may be optimal in some cases where the circuit energy dominates the energy consumption, instead of transmission energy. However, solving the convex optimization requires a powerful node and may be not be appropriate for resource limited sensor nodes. Furthermore, the single-hop strategy introduces a non-trivial hardware requirement for the radio. High powered radios are necessary which in turn compromises the requirement for low cost sensor nodes.

Finally, an implementation is proposed by Kwon et al. [85], where the sink node determines the PHYL transmission power, the MACL retry limits and the NETL routing path for each sensor node to maximize network lifetime. Once the optimum transmission power is selected, the retry limit can be set to minimize the energy consumption of a node. Furthermore, a link cost function is set according to the ratio of the required transmission energy for a packet transmission to the residual energy, to determine the routing path. The path selection requires an additional knowledge on the residual energy of each sensor node. In order to support this, the sink node estimates the residual energy of all the sensor nodes based on the energy consumption determined by the retry limit allocation algorithm. To ensure realistic estimations each sensor node periodically reports to the sink with the required information. This may incur a lot of network traffic, which for larger networks may be more than the actual sensed data traffic.

4.3.4 Discussion

In the existing WSN literature, cross-layer interactions have proven to increase the performance and efficiency of WSNs. However, many proposals, which have not been included in this literature review, claim the use of cross-layering, but provide a very vague description of the implementation approach and layer interaction. This is due to the fact that cross-layering is often not the focal point but merely another feature on the side. Several propose the joint optimization of layers but do not employ any layer interaction at all. All layers merely strive towards a common metric, which strictly spoken, cannot be regarded as cross-layering.

Most of the investigated WSNs literature does not propose an underlying framework for cross-layer interactions. This can be attributed to the fact that the resource limited nodes have limited memory and cannot accommodate elaborate frameworks which feature separate optimizing entities or managers. Therefore, implementations of tightly coupled layer protocols exists that commonly strive towards energy optimization for maximum network lifetime in WSNs. Typically these do not provide a generic solution but are frequently aimed at a specific application which involves a specific topology, size and traffic pattern.

4.4 A WSAN MAC PROTOCOL USING CROSS-LAYERING

A hint of cross-layer interaction is proposed by a MAC protocol that calculates its TDMA scheduling according to the current application state. The Sleep schedule for the radio of the nodes for fast and Efficient Control of parameters (SEC) MAC protocol is proposed by Vasanthi and Annadurai [64]. The protocol strives to achieve minimum latency for the reporting of important sensed data. Strictly speaking, this protocol cannot be defined as a layered MAC protocol. It relies on APPL information of an event to determine its schedule.

The protocol assumes a semi-automated WSAN architecture. This means that data is firstly sent to the network sink which evaluates the data and if necessary instructs the actor nodes to perform appropriate actions. During normal operation, the sensor nodes sense the environment periodically using a static schedule and send the data towards the sink. However, when the sink evaluates the collected data and finds the sensed value exceeds a set point, the protocol sends a signal to activate the radios of the actor and sensor nodes along the path towards the area of interest i.e., where the event has occurred. The nodes adopt a new schedule as illustrated in Figure 4.7. The green blocks show the time that a sensor node is awake. The sensed data is sent towards the sink as shown by the arrows. Sensed data travels from sensor nodes I to M following a sequential wakeup schedule. Once the sink receives the data, it sends control packets towards the actor node, which is peculiarly situated close to sensor node I, along M to I in the colourless blocks. This kind of temporary activation reduces the end-to-end latency, which is vital for closed loop control operation. These nodes are kept switched on to sense the environment continuously and forward the data towards the sink, until the event is resolved. Once the event is resolved the sleep periods return to the original schedule with shorter awake times as shown on the right hand side of Figure 4.7. Energy consumption is minimized by only changing the sleep schedule of a subset of nodes that are involved in the event area and along the path towards the sink and corresponding actor node.

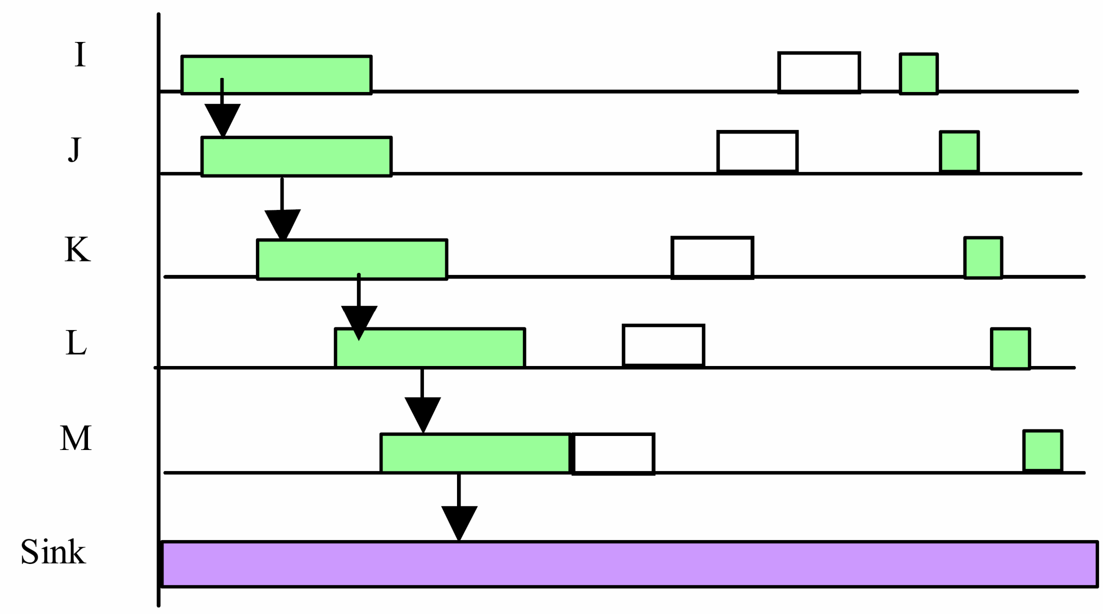

Figure 4.7 Sleep schedule for fast and Efficient Control (SEC). From [64].

It is not clear if the protocol presents a schedule or a contention-based protocol. The schedule change provides a good approach to minimize the latency. However, utilizing a semi-automated architecture, data is required to travel all the way to the sink and then all the way back to an actor node. This consumes a lot more energy when compared to a fully automated architecture, which does not contain a sink. The direct sensor-actor coordination of automated architectures is favoured simply because a much better distributed control can be achieved, ensuring lower delays.

The SEC protocol requires a lot of calculations and has to keep track of the adjustments of the time slots. When multiple events occur close to each other, calculating the optimal schedules that do not clash, may be very complex and a lot of overhead will be needed for management thereof.

4.5 CHAPTER SUMMARY

In this chapter, the cross-layer paradigm was introduced and different frameworks and approaches were discussed. Furthermore, literature that encompasses some form of cross-layer interactions in WSNs has been investigated and categorized according to the combination of layers that interact. The existing WSN cross-layer literature shows that the utilization of a cross-layer approach provides performance gains and adaptability to changing network conditions. To the best of the author's knowledge, with the exception of the discussed WSAN SEC MAC protocol, no cross-layer framework or implementation literature exists for WSANs. Specifically, WSANs involve stringent requirements for latency, energy efficiency and reliability, and depend on these to ensure feasibility and achievement of the overall application objectives. There is therefore a need for further research to investigate how the successful cross-layer approach can be applied to the WSAN's CPS and what performance gains and benefits can be achieved. The cross-layer paradigm promises considerable performance gains and might prove vital for ensuring the feasibility of WSANs.

Chapter

5 CROSS-LAYER INTERACTING CPS AND CROSS-LAYER FRAMEWORK DESIGN FOR WSANS

The research review of chapter 3 provided an overview of existing WSANs and relevant WSNs protocols and revealed that WSANs specific protocol research is still in an early phase. There exists a need for further research on WSANs protocols that specifically satisfy the stringent latency and energy requirements and can be applied to a variety of applications. In the light of this research need, the cross-layer literature review of chapter 4 reveals that the utilization of a cross-layer approach by the CPS provides prominent performance gains and adaptability to changing network conditions. Since the WSANs require latency minimization, high energy efficiency and high reliability, there exists a need for further research to investigate how the successful cross-layer approach can be applied to the CPS of WSANs and what performance gains can be attained by employing such a paradigm.

In this chapter, the design of a new complete WSANs CPS that adopts and implements a cross-layer approach is proposed. The CPS design consists of layered protocols that are modified and enhanced to feature cross-layer interactions to promote latency minimization, energy efficiency and reliability for WSANs. At first, the research focus area and design steps with key challenges are discussed. This is followed by the proposal of a new Simple Cross-Layer Framework (SCLF) design for WSANs. Next, suitable layered protocols are selected and potential cross-layer parameters identified for each layer of the CPS. Finally, cross-layer enabled protocol designs are proposed that feature modifications and enhancement to the layered protocol designs by utilising information from other layers. The new cross-layer enabled protocols aim at providing increased performance with regards to latency and energy efficiency of WSANs.

5.1 FOCUS AREA

Before embarking on a CPS design, it is necessary to define the research focus area. WSANs can be applied to a variety of application areas and several examples are given in section 2.5. It is observed that the WSAN applications have several common characteristics and involve a common operational scenario. To define the focus area of the proposed CPS design, a basic operational scenario of WSAN applications is assumed to involve the following activities.

- The physical environment is periodically sensed and evaluated by the sensor nodes.
- The sensed data is routed towards an actor node.
- Upon reception of sensed data, an actor node performs an action upon the physical environment if the occurrence of a disturbance or critical event has been detected.
- If no acting is required, the sensed data is logged or sent to a log base station for analytical, statistical or system calibration purposes.

This basic operational scenario can be found in some form or another in the majority of the WSANs applications, especially in fire control, precision agriculture (moisture/humidity and temperature control), environmental control (microclimate and lighting control in buildings/homes, home automation, etc.) and target tracking with attack detection and response (military applications). The objective of the CPS design is to ensure fulfilment of the abovementioned WSANs basic operational scenario, by carrying out the operational activities while ensuring maximum efficiency and compliance to the network requirements.

The cross-layer paradigm has shown significant performance improvements in similar types of networks that includes the ad hoc networks and WSNs. This chapter contains a CPS design which implements the cross-layer paradigm for performance gaining purposes in WSANs.

Due to the fact that WSANs comprise heterogeneous nodes, the CPS operational functionality and requirements for actor and sensor nodes differ somewhat in particular the APPL and NETL. The sensor nodes are constantly concerned with communicating sensed data with low latency and energy efficiency to the actor nodes. On the other hand, the actor nodes are concerned with performing effective, appropriate and timely actions and collaborating with each other if necessary. The *focal point* of this dissertation is set on applying and implementing a cross-layer paradigm for the CPS of resource constrained *sensor nodes* inside WSANs. The protocols within the stack need to consider trade-offs and stringent constraints to ensure that viability of the WSAN is sustained. Where relevant, actor node protocol functionality or considerations are given and discussed to highlight specific interoperability with sensor nodes. Since actor nodes are less resource constrained and have higher computational and communicative capabilities, their communication protocols are more concerned with effective functionality and less with efficient resource utilization.

Although only the sensor node CPS design is extensively considered in this chapter, chapter 3 includes CPS layer tasks and challenges for actor nodes as well to ensure a complete understanding of WSANs protocol layers.

Next, the design steps and key challenges are discussed.

5.2 KEY CHALLENGES, RESEARCH APPROACH AND DESIGN STEPS

The importance of this research work relates to the viability of the WSANs. WSANs are very popular and will be implemented and used for many applications and might also take over from some WSNs. If WSANs are unreliable, inefficient and resultantly costly, WSANs will not be viable for implementation. It is crucial for the network to be efficient for it to be feasible. The network efficiency is based on the effectiveness of the CPS and how well the protocols address and cater for network constraints.

In section 2.2, the design factors of WSANs are discussed and have to be taken into consideration by the CPS design. The *key challenge* of a WSAN CPS design is to ensure that the protocols together, ideally meet all of the following requirements:

- Real-time communication
- Very low energy consumption
- Low complexity
- Low overhead
- High network reliability
- High data integrity
- High network stability
- Low network congestion

The WSAN protocols must consider and address as much network requirements as possible. However, this has proven difficult, since certain network requirements have a trade-off relationship. Several protocol design studies [2][29] have identified that the most prevailing trade-off between network requirements exists between the latency and the energy efficiency of a network. It is important to consider this trade-off to ensure the desired performance of the network.

The *research approach* and chronological *design steps* that are followed in this chapter, in finally realizing the complete cross-layer enabled CPS design are as follows.

1) A WSANs communication architecture that features a simple and effective cross-layer framework is proposed to enable layer interactions. The cross-layer framework design has to be viable for resource constrained WSAN sensor nodes.

2) Suitable protocols are selected for every layer in the CPS. Where necessary, existing WSAN or WSN layered protocols are modified and adapted to ensure suitability for the defined research focus area. The selected protocols form the basis for protocol development and adoption of the cross-layer paradigm, which promotes efficiency optimization.

3) Specific parameters within each layer that may be beneficial to other layers in the CPS are identified. This information is inserted into the cross-layer framework to enable interaction between layers and to promote network efficiency optimization.

4) Finally, the selected layered protocols (of step 2) are modified and enhanced to include parameters from other layers within their functionality. The enhanced protocols make use of information from other layers to promote optimization of latency and energy efficiency.

As revealed by the literature study in chapter 3, in WSANs and similar types of networks, there exists a trade-off between latency vs. energy. The cross-layer enabled protocols take special care and strive towards providing a solution to the latency vs. energy trade-off. The feasibility of WSANs depends on the optimization of these two main metrics.

5.3 WSAN SIMPLE CROSS-LAYER FRAMEWORK DESIGN

In order to introduce the cross-layer paradigm to the CPS of WSANs, some kind of framework is required. The main requirements of a cross-layer framework design for WSAN nodes were identified to include: *simple and generic design*, *easily accessibility of information*, *low overhead* and *low memory usage*. The design of a Simple Cross-Layer Framework (SCLF) and its operation is discussed in the following subsections.

5.3.1 SCLF Architecture

The cross-layer paradigm is applied to the WSAN CPS by adding the SCLF to the existing architecture. The discussion of existing frameworks in section 4.2 provides an overview of different implementation approaches. Several existing frameworks are not suitable for implementation on resource constraint WSAN sensor nodes since they are often too complex, induce high overhead and require a lot of memory space. With regards to the abovementioned

WSANs cross-layer framework requirements, the proposed SCLF that is added to the communication architecture for WSANs is illustrated in Figure 5.1.

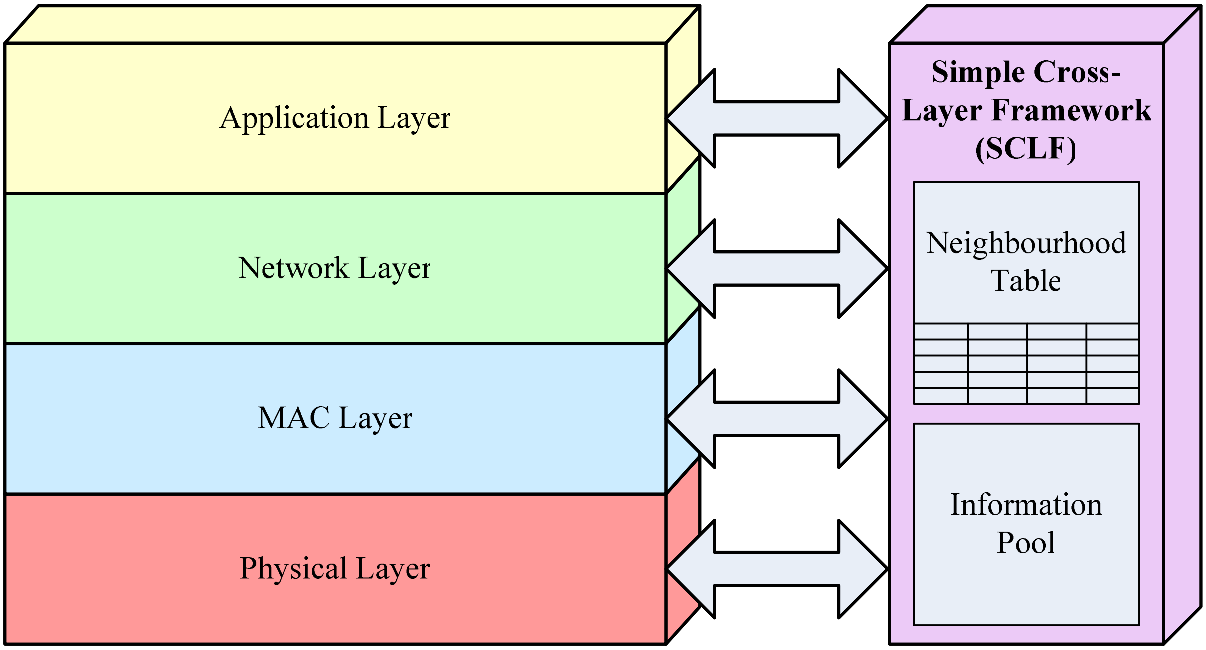

Figure 5.1 The WSAN CPS (left) with Simple Cross-Layer Framework (right).

The SCLF is a combination of the existing MobileMan framework (section 4.2.2) originally designed for ad hoc networks and the Layer Optimization Interface framework (section 4.2.8) designed originally for WSNs. The SCLF is tightly coupled to the WSAN CPS by allowing all layers to access it as in the MobileMan design. The SCLF contains a neighbourhood table and an information pool structure. These structures are similar to the ones used in the Cross-Layer Optimization Interface framework. However, the message pool is replaced by an information pool structure to allow information of a node to be shared among the CPS layers. The information contained in the information pool structure, pertains to the node itself and not the neighbouring nodes. The proposed SCLF does not contain a message pool since storing vast message information in the framework only disrupts the chronological layer progression of the message and introduces a lot of additional overhead and memory space necessary to keep track of every message’s details and state.

The framework design is very simple and allows direct access of its information sharing structures to all layers of the CPS. To keep the framework simple and generic, the design involves no dedicated optimizing entity like the framework discussed in section 4.2.5, which is designed for mobile ad hoc multimedia applications. Consequently, no complex optimizer needs to be developed that involves tedious reprogramming whenever a new layered protocol is introduced. A dedicated optimizer would require a considerable amount of memory space for its complex optimization algorithms, which is not viable for resource constrained WSAN sensor nodes.

The SCLF does not change the architecture of the CPS but merely adds the cross-layer functionality to retain the modularity of the layered protocol stack design. The framework is therefore not dependent on or influenced by the underlying protocols used in the CPS. The details of the operation of the SCLF are discussed next.

5.3.2 SCLF Operation

Information sharing between protocol layers is the main notion of the cross-layer paradigm. Therefore, this framework only provides the means for sharing information and is not directly involved in the actual efficiency optimization of the CPS. The actual efficiency optimization is entrusted to the separate protocol layers which have to accept, interpret and integrate information from other layers into their local layer specific decisions and functionality. This means that the framework does not require any overhead.

As mentioned, the information is presented and shared in two structures namely a neighbourhood table and an information pool. These structures do not require any overhead and are utilized by the separate protocol layers to share relevant information. The neighbourhood table contains information of a particular neighbour node relevant for cross-layer interaction. Any layer can add information pertaining to a specific neighbour to the table. To ensure minimum memory usage and avoidance of multiple copies of the same information in the CPS and SCLF, the information contained within the neighbourood table only exists in the SCLF structure. In this way information updates made by a protocol layer are directly done in the framework structure, which saves a lot of overhead otherwise needed to synchronize information updates. In essence, the table allows a fusion of neighbourhood tables that commonly exist in the NETL and MACL.

The information pool structure contains information used for cross-layer interactions that does not pertain to neighbour nodes. This can be a protocol status variable, for example, where the information would be updated inside the information pool structure whenever the protocol status changes and used by other layers of the CPS.

Information contained in the framework structures may only be changed or updated by the protocol layer that originally submitted it. This means that protocol layers may only read the submitted information of other layers. This ensures that information is not illegally tampered with by other protocol layers, and the integrity of the information is preserved. No precedence is given to any particular protocol layer to access the information contained in the framework structures.

Another cross-layer feature that is proposed is the loose encapsulation of packets. This introduces the ability of a lower layer to read the type field of packets passed to it by its direct, higher CPS layer. The NETL is able to view the APPL packet type field and the MACL protocol is able to view the NETL packet type field. This cross-layer feature avoids the need for a message pool inside the framework as proposed by the Cross-Layer Optimization Interface framework and saves scarce memory resources. A message pool requires message ids and types to be stored. By using loose encapsulation no additional unique message ids are necessary. The complexity is lowered by not having to keep track of all existing packets inside the CPS. Rather than requiring a packet id and packet type variable inside a message pool structure, the already existing packet type field is used for every layer (except the PHYL).

In essence, the WSAN cross-layer framework presents a simple generic facility for information storage and sharing of cross-layer parameters. The framework incurs low overhead and promotes low memory usage. In order to implement the cross-layer paradigm into the CPS, the layered protocols pertaining to every layer have to be modified and adapted to incorporate other layer information into their local functionality.

The layered protocol designs pertaining to every layer in the CPS are given in the next sections. Parameters within each layer that are beneficial to other layers are identified and finally the modified and enhanced layered protocols that implement the cross-layer approach are proposed.

5.4 LAYERED WSAN APPL PROTOCOL DESIGN AND IDENTIFICATION OF CROSS-LAYER PARAMETERS

The WSAN CPS application layer protocol for sensor and actor nodes respectively, has to ideally perform all the primary tasks as stipulated in section 3.2.1. The proposed layered APPL protocol for *sensor* nodes in this section, implements the application functionality as required in the abovementioned basic operational scenario. After considering some applicable existing WSAN APPL literature, suitable basic layered WSAN APPL protocol features are selected. These features are integrated as described in the operational details section of the proposed WSANs APPL protocol design. APPL parameters are then identified that are beneficial to other layers in the CPS, as part of the cross-layer paradigm, towards promoting the optimization of latency and energy efficiency. Finally, some notes concerning the APPL of the WSANs actor nodes are given.

5.4.1 Applicable Existing Literature

The existing literature relating to the WSAN APPL was discussed in section 3.2.2 and, as mentioned before, advanced coordination and collaboration schemes for acting are beyond the scope of this dissertation. A suitable APPL protocol is needed that fulfils the defined basic operational scenario and forms the basis for development and adoption of the cross-layer efficiency optimization paradigm.

The analysis of the existing WSAN literature on APPL specific protocols revealed that WSAN APPL protocols remain a largely unexplored region. However, there exist a few protocols or approaches that are sensible and provide some pointers for the design of a suitable WSANs APPL protocol. When considering the features and problems of existing literature tabulated in Table 3.1 (in section 3.2.2.7), suitable features are presented by the

Network-Centric Actuation Control, the Distributed Coordination Framework and the Adaptive Actuator Allocation, Aggregation and Prioritization Strategy protocols. These protocols will form the design basis for the proposed layered APPL protocol design in this dissertation.

The Network-Centric Actuation Control protocol contains a rule-based system that allows decision taking within the network itself and handles actuation by self-organization methodologies to provide a network-centric actuation control. A network-centric actuation approach is favoured since it enables a fully automated WSAN architecture. It has been revealed that this type of architecture ensures the lowest delays. Sensed data originating from within the network is directly relayed towards actor nodes and not via a central controller. This means that information is evaluated by the network nodes within the network and not by a single controller. Aggregation or decision taking issues are solved without having a global view of the entire network behaviour. This provides an accelerated response by the WSAN, by suggesting localized reactions to events. However, schemes for actor node collaborations and specific rules for action performing are not mentioned by the protocol authors. A suitable approach for these issues are given by the Distributed Coordination Framework, which employs a real-time auction protocol to select the best set of actor nodes to perform action on each overlapping area. This approach may be employed once an actor node cannot handle an event on its own.

The final relevant existing protocol is the Adaptive Actuator Allocation, Aggregation and Prioritization Strategy protocol, which proposes important features like virtual grids, data aggregation and prioritization of data. This is the only protocol that considers prioritization and aggregation of data within the APPL and is very appropriate for the defined basic operational scenario. Furthermore, the notion of virtual grids or clusters is suitable for local decision taking, control and rapid responses within WSANs. However, this protocol features actuator allocation strategies which are less feasible since it incurs frequent re-allocation of actor nodes. Next, the operational details of the proposed APPL protocol for WSANs are given.

5.4.2 Network-Centric Actuation Control with Data Prioritization APPL Protocol Design and Detailed Operation

The operational details of the APPL protocol for WSANs, as implemented on all *sensor* nodes inside the network, are proposed in this section. The proposed new protocol presents a Network-Centric Actuation Control with Data Prioritization (NCAC-DP). The proposed protocol does not present an extremely elaborate APPL protocol, but is rather aimed at satisfying the functional requirements of the defined basic operational scenario of WSAN applications.

The data prioritization is kept simple by specifying sensed data values as either below or above a critical threshold. Sensed data packets are hence categorized as either critical or non-critical. This simple categorization eliminates the need for complex calculations to determine specific packet latency bounds or expiry times. Determining latency bounds is not a trivial task since it is firstly, extremely application specific and secondly, they must be different for every specific hop distance from the destination actor node.

The topic of latency and energy efficiency within the APPL of sensor nodes chiefly depends on the specifics of time synchronization and/or localization and positioning requirements. The proposed APPL protocol primarily needs localization and positioning information for the actor node to more or less know where an actuation response is required. The issue of time synchronization is less important since protocols that are proposed for the remaining CPS layers do not require very strict time synchronization.

Sensor nodes inside WSANs are considered to have two roles. Firstly, sensor nodes are generators of sensed data and send their sensed data towards an actor node. Secondly, sensor nodes are also responsible for relaying other sensor node’s sensed data packets and act as intermediate nodes on the route towards an actor node, i.e. they are also acceptors of data. Conclusively, the APPL protocol situated on WSANs sensor nodes has to include both functionality of data generator and data acceptor.

The proposed NCAC-DP simple layered APPL protocol operates in one of four states that include the INITSETUP, NORMAL, CRITICAL and INCRITICALROUTE. The APPL protocol operation as generator of sensed data is illustrated in Figure 5.2. The protocol states are discussed in the next subsections.

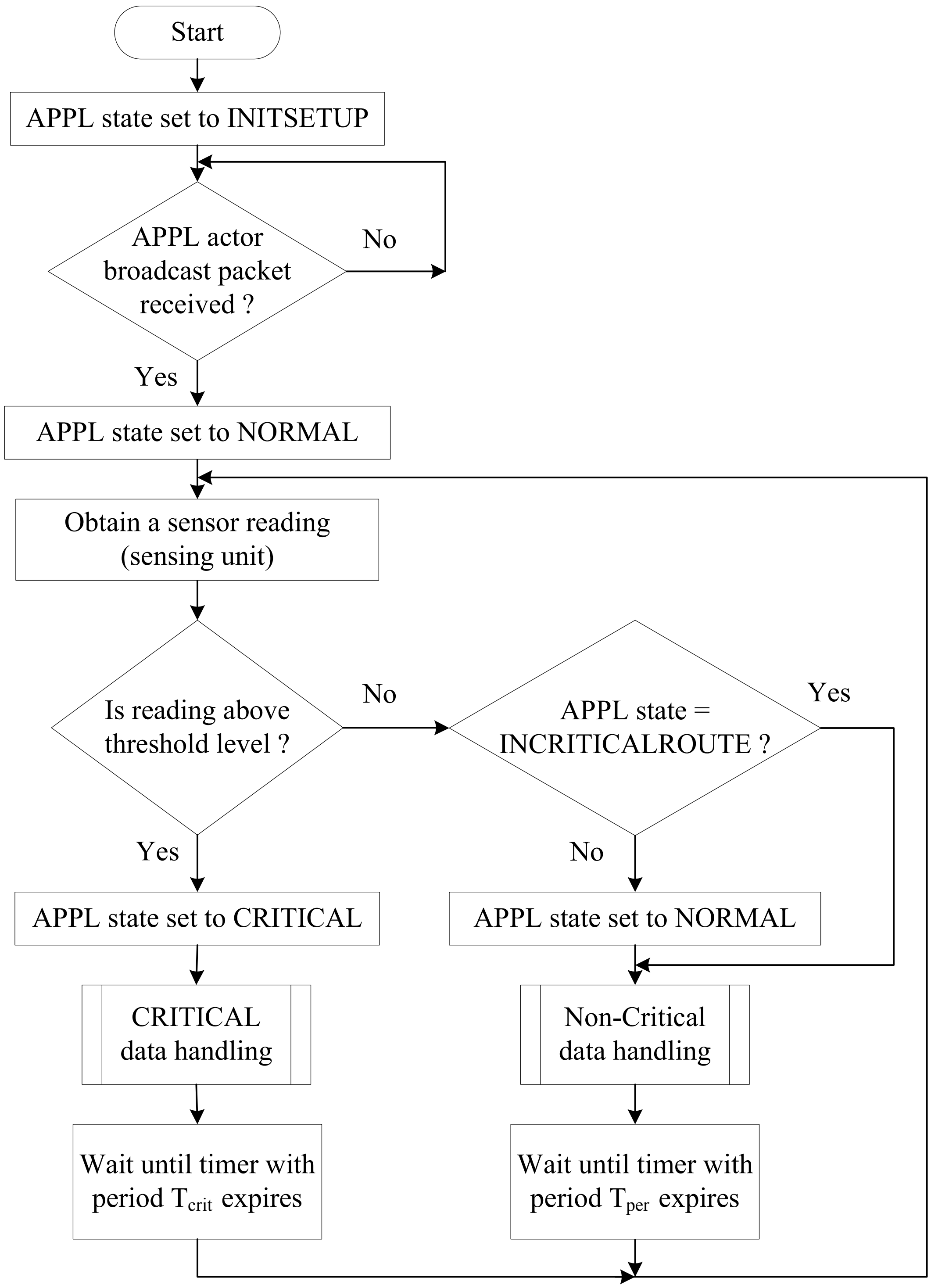

Figure 5.2 Flowchart of the APPL protocol operation as generator of sensed data.

5.4.2.1 INITSETUP state

At WSAN initialization, the APPL protocol enters the INITSETUP state. In this state the APPL does not know if the node is connected and part of the WSAN or not. Adding the node to the network is the task of the NETL protocol. Initially the actor nodes broadcast an initial packet to instigate setup or self-configuration of the network. Within this broadcast packet, APPL information is presented that may be necessary for initial time synchronization and localization or positioning. Elaborate localization or positioning algorithms are beyond the scope of this dissertation and can be obtained from GPS systems or other localization techniques [86] that may involve the RSSI [22][87] for example. It is assumed that all nodes know the coordinates of their location. The issue of time synchronization also needs to be addressed by the APPL since sensor nodes have to include a time stamp in the APPL packet to enable the actor node to determine the time of the initial occurrence and duration of an event. Very small time synchronization packets are sent periodically by actor nodes with GPS systems, to ensure reasonable time accuracy throughout the network. These small packets can also be fitted with APPL specific queries or instructions from the actor nodes regarding threshold level value changes for example. The period of time synchronization packets depends on the inaccuracy of the timing quartz hardware contained on sensor nodes. This period is kept as low as possible to incur only minimal traffic.

In the INITSETUP state, the APPL does nothing but wait for a broadcast packet from an actor node. Only once this packet is received, the time synchronization is obtained, the position coordinates are calculated and subsequently the APPL switches to the NORMAL operation state.

5.4.2.2 NORMAL operational state

The NORMAL operational state is entered either initially once an actor node broadcast packet is received or after a critical event was neutralized or resolved. The primary role of a sensor node is as a generator of periodically sensed data packets. As defined in the basic operational scenario, the physical environment is monitored by obtaining a sensed value periodically. The NCAC-DP APPL protocol sets a sensing timer to instigate the sensing of the physical environment with a period of T_{sense}. Next, in the wide variety of applications it is evident that the data inside the WSANs demonstrates different levels of criticality. Therefore, once the timer expires, a value is sensed and evaluated by comparing it to a critical threshold value. This critical threshold value is specified by the application. In precision agriculture for example, the critical threshold value would be a specific temperature or humidity level. Providing evaluation of sensed values on the sensor nodes enables them to distinguish between different data packet priorities. The sensed value is immediately classified as either non-critical data or as critical data and is assigned the relevant priority. For reasons of simplicity, only two priority levels are proposed. Non-critical sensed data are those sensor readings that fall below the application specified critical threshold level and require no action by the actor nodes. The non-critical data is less delay-sensitive than critical data. On the other hand, sensed values that are above the application specified critical threshold value are immediately rated as critical data and consequently allocated with the highest priority. Once sensed data is classified as being critical, the APPL enters the CRITICAL state.

If the sensed data is rated as non-critical, the APPL constructs the APPL data packet, sets the packet type field to NONCRITICAL_MESSAGE and passes it to the next layer, the NETL. The data packet format is given in section 5.4.2.6. Once the packet is passed to the NETL, the sensing timer is reset with a period of T_{sense}.

5.4.2.3 CRITICAL operational state

Considering the role of the sensed data generator, the CRITICAL state is entered if the sensed value falls above the application specific critical threshold value. This indicates the detection of an event or disturbance in the physical environment and requires rapid action performing by actor nodes. The sensed data is rated as critical, which means that it has the highest priority and is highly delay-sensitive. The timer is reset to a new period of T_{crit}, where $T_{crit} < T_{sense}$. T_{crit} is set to incur sensor readings at least four times more often than in normal non-critical circumstances. This ensures that the corresponding actor node receives regular updates of the extent and severity of a critical event.

The critical packet format is given in section 5.4.2.6. Three types of critical data packets are proposed. The three types include CRITICAL_ATTENTION_MESSAGE, CRITICAL_MESSAGE and CRITICAL_RESOLVED_MESSAGE. The constructed critical data packets are assigned a type according to the current status of the event, before being passed on to the NETL. The CRITICAL_ATTENTION_MESSAGE type is assigned to the first packet sent to the actor node after a critical event is detected. The CRITICAL_MESSAGE type is assigned to data packets that contain critical sensed data that follow the initial attention packet. Finally, the CRITICAL_RESOLVED_MESSAGE type is assigned to the data packet that is sent after the critical event is resolved.

As mentioned, after the detection of a critical event or disturbance, the first critical data packet that is constructed is of type CRITICAL_ATTENTION_MESSAGE. This packet type indicates to a receiving intermediate sensor node's APPL, that a critical data packet was received and consequently enters the INCRITICALROUTE state.

5.4.2.4 INCRITICALROUTE operational state

The INCRITICALROUTE state is entered by an intermediate node if a critical data packet of type CRITICAL_ATTENTION_MESSAGE is received. This is an indication to the APPL that the intermediate node is in the NETL protocol's chosen route towards the actor node. NCAC-DP remains in the INCRITICALROUTE state until it receives a critical packet of type CRITICAL_RESOLVED_MESSAGE, indicating that the event or disturbance is resolved. Once a CRITICAL_RESOLVED_MESSAGE is received, the APPL state is set back to NORMAL and the sensing timer is reset with a period of T_{sense}. In between the initial CRITICAL_ATTENTION_MESSAGE and the final CRITICAL_RESOLVED_MESSAGE type packets, CRITICAL_MESSAGE type packets may be received that signify ongoing event reporting. These critical data packets are simply passed to the NETL without any APPL state change. Very importantly, an intermediate sensor node's periodic data sensing schedule is continued even though NCAC-DP may be in the INCRITICALROUTE state. If at any point, a node senses a value that falls above the critical threshold value, NCAC-DP enters the CRITICAL state.

5.4.2.5 Aggregation

Aggregation within the APPL is not viable since the data packets are passed very quickly to the NETL, upon reception. It is not permissible for the APPL to delay a neighbour data packet until its own next sensor reading is obtained, since this would incur very high delays. When a node is used as an intermediate node, the APPL does nothing with the APPL content of a data packet. Therefore, the best layer for aggregation is considered to be the MACL due to the fact that it employs a queue for data packets. More detail concerning aggregation as employed in the MACL is given in the MACL protocol section 5.6.2.

5.4.2.6 Application Layer Packet Formats

The basic packet format of the NCAC-DP APPL protocol specific packets that are passed on to the NETL is illustrated in Figure 5.3. The data packet format is used for critical and non-critical packets that contain no aggregated data. No source or destination address is contained in the format, since the APPL protocol entails no addressing scheme. Furthermore, the packets are kept as small as possible to ensure latency and energy efficiency.

Type	Source X-coordinate	Source Y-coordinate	Timestamp	Actual Sensed Data
4 bits	16 bits	16 bits	32 bits	Variable

Figure 5.3 The format of a basic NCAC-DP APPL protocol packet.

- The *Type* field holds the packet type which includes NONCRITICAL_MESSAGE, CRITICAL_ATTENTION_MESSAGE, CRITICAL_MESSAGE and CRITICAL_RESOLVED_MESSAGE.
- The *Source X-coordinate* and *Source Y-coordinate* fields entail the coordinate information of the source sensor node of the actual sensed data. This is either in reference to a specific actor node or based on a network wide coordination system.
- The *Timestamp* field contains the time when the actual sensed value was measured.
- The *Actual Sensed Data* field includes the measurement taken by the sensing unit of the source sensor node. May vary with different sensor reading configurations. Some

sensor nodes may feature more than one sensor and hence the packet may contain a larger payload.

An APPL data packet that entails aggregated data of two data packets consists of every field in Figure 5.3 twice, except for the *Type* field. Furthermore, it is important to record the timestamp of both data packets since it is used to determine the time the sensed values were obtained. This time is used by actor nodes to determine the temporal order of incoming data packets.

5.4.2.7 Summary of Protocol Features and Main Parameters

The main protocol features are summarized as follows:

- Sensor readings are periodically obtained.
- Sensor readings are prioritized according to a critical threshold level.
- Contains four operational states.

Table 5.1 shortly summarizes the some important NCAC-DP parameters.

Name/Symbol	**Description**	**Value**
State	State of the APPL	Depends on sensed value.
T_{crit}	Period of sensing when a critical event or disturbance is sensed	Application dependent, should be at least T_{sense} / 3
T_{sense}	Period of sensing when no event or disturbance is sensed	Application dependent

Table 5.1. Summary of the main NCAC-DP protocol parameters.

5.4.3 Identified APPL Parameters for Cross-Layer Interaction

The cross-layer framework as proposed in section 5.3, enables any layer to insert parameters into the information pool and neighbourhood table structures. This information may be read and used by all other layers in the CPS. Specific APPL parameters and features that are useful and beneficial to other layers in the CPS towards promoting overall latency minimization and energy efficiency improvements, are identified as follows.

a. The *APPL state information* parameter is inserted into the information pool structure of the SCLF. In this way the lower layers may distinguish between the non-critical and critical APPL state of a node and adjust their operation or approach accordingly.

b. Loose encapsulation of the APPL data packets is proposed to enable the *NETL* to read particular information contained within an APPL packet header. Specifically, the APPL *packet type* parameter is of primary interest. This parameter may prompt the NETL to handle and react differently to different types of packets with different latency constraints or priorities. It is a cross-layer feature that is an important attribute of the cross-layer paradigm. A loose encapsulation feature primarily enables the NETL to distinguish between critical and non-critical packets in order to instigate a specific reaction by the NETL and ensure appropriate handling of the data packet with regards to routing.

c. The s*ensing period* T_{sense} is inserted into the information pool of the SCLF structure and may be used to configure certain timer periods of lower layers.

No APPL information is inserted into the SCLF's neighbourhood table. The actual incorporation and usage of the identified APPL parameters by other layers in the CPS is discussed in section 5.8.

5.4.4 Notes on the Application Layer protocol for actor nodes

The APPL protocol tasks for actor nodes, as stipulated in section 3.2.1, specifically includes the processing of sensor data, coordination and collaboration towards effective acting and initiation of acting. These are shortly discussed in the following subsections. The remaining primary tasks and challenges relevant to actor nodes have been discussed to a certain extent in the proposed WSAN APPL protocol for sensor nodes.

5.4.4.1 Processing of Sensed Data

The sensed data received from sensor nodes is processed by the actor node APPL protocol. Dependent on the application, this information may be used to gain knowledge and estimate locations of future event occurrences or hot spots. For example, if the temperature of an area is currently steeply rising but still below the critical threshold level, actor nodes may start pre-emptively to collaborate in preventing an event. Therefore, useful non-critical sensed data is stored for analytical and statistical reasons. Only sensed data that is useful to an actor node, i.e. not too old, is stored on the actor node itself. Otherwise, a primary storage entity that may take the shape of a single sink node may be used. This storage entity has greater storage capacity than normal actor nodes and may be accessible by external entities.

5.4.4.2 Coordination and Collaboration towards Effective Acting

The coordination and collaboration of actor nodes is vital in ensuring effective acting in response to the detection of an event occurrence or disturbance in the environment. Intricate algorithms and schemes are beyond the scope of this dissertation. However, it is noted that WSANs should adopt a distributed control approach to ensure rapid localized responses to events. This means that actor nodes that gather sensed values from sensor nodes in their vicinity, decide what action is needed and if any collaboration with other actor nodes is necessary. Only if necessary, other actor nodes should be notified of an event, either when a single actor node cannot resolve the disturbance alone or an event may spread to other areas, unreachable by the concerned actor node. Latency minimization and energy efficiency may be more easily optimized in WSANs that promote distributed control.

An actor node’s actuation range may vary depending on the application and the actual actuation unit. It is assumed that actor nodes within the WSAN are spread in such a way that they may provide relatively good coverage of the applicable physical environment among their actuation radii. This means that the mobility of actor nodes is considered somewhat limited. The issue of handling excessively mobile actor nodes in WSANs is beyond the scope of this dissertation.

5.4.4.3 Initiation of Acting

Once the actor nodes have decided on how an event will be handled, the actuation unit is controlled by the actor node’s APPL protocol to resolve the disturbance or event. The exact location of the event is available since the coordinates of the sensor node that detected the event are included in the data packet. The type of control that is employed by WSANs follows a feedback control approach. Once an event was detected, sensor nodes increase their period of sensing to provide frequent updates of the event status to the actor node. This enables them to perform actions more accurately and effectively.

5.5 LAYERED WSAN NETL PROTOCOL DESIGN AND IDENTIFICATION OF CROSS-LAYER PARAMETERS

The WSAN CPS NETL protocol for sensor and actor nodes respectively, has to ideally perform all the primary tasks as stipulated in section 3.3.1. The proposed layered NETL protocol for *sensor* nodes in this section implements the network structure and routing related functionality as required by the defined basic operational scenario. After considering some applicable existing WSAN NETL literature, the most suitable layered WSAN NETL protocol is selected to serve as a design basis for the adaptation of the cross-layer paradigm. NETL parameters are then identified that are beneficial to other layers in the CPS, as part of the cross-layer paradigm, towards promoting the optimization of latency and energy efficiency. Finally, some notes concerning the NETL of the WSANs actor nodes are given.

5.5.1 Applicable Existing Literature

The existing literature relating to the WSAN NETL was discussed in section 3.3.3. To distinguish between different types of NETL protocols, the much researched field of WSNs NETL protocols reveals that protocols pertaining to the NETL can be classified into three classes, as given in section 3.3.2. The main two protocol classes are the hierarchical-based and the flat-based protocols. Furthermore, these protocols can be subdivided into reactive, proactive and hybrid protocols. A suitable NETL protocol is needed for WSANs that fulfils the defined basic operational scenario and forms the basis for development and adoption of the cross-layer efficiency optimization paradigm.

The trend of current emerging WSAN NETL protocols research shows that a hierarchical cluster-based routing approach is most favourable for WSANs. This approach is the most suitable approach when dealing with the existence of resource richer actor nodes spread among numerous sensor nodes. Very suitably, well-sized clusters of sensor nodes can be formed around actor nodes that take the role of cluster head. The analysis of the existing WSAN literature on NETL specific protocols in section 3.3.3 revealed that the division of the WSAN into clusters consisting of cooperating nodes has several favoured advantages. Protocols that do not form clusters or induce any other kind of network structure are burdened with an added communication load as a consequence of overhead needed firstly for coordination and secondly, to ensure latency and energy efficiency of large indeterminate regions. With the absence of clearly defined, well-sized domains, the task of optimizing the latency and energy efficiency is more complicated and fairly limited. Furthermore, in the non-cluster-based WSAN routing protocol literature, it is often not clear how a sensor node will choose an actor node to forward the sensed data to. This results in an increased route discovery and selection process complexity, since more nodes may be involved in the route towards an actor node. Furthermore, although none of the non cluster-based literature mentions it, some kind of elaborate addressing scheme is requisite. This non-trivial task is necessary since sensor nodes have to include an address in their data packet headers and duplicate addresses may not occur. With cluster-based approaches however, only a simple addressing scheme is required since clusters contain only a limited amount of nodes. Node addressing is solved by assigning a simple addressing scheme to every cluster rather than having to accommodate the entire network. Finally, non cluster-based protocols that divert from using clusters often fail to adhere to the stringent delay requirement primarily posed by WSAN applications. Cluster-based approaches promote distributed control as envisaged by the above proposed NCAC-DP APPL protocol.

When considering the features and problems of existing WSAN NETL literature tabulated in Table 3.3 (in section 3.3.3.4), some protocols entail features that are suitable and applicable for the defined WSAN focus area and basic operational scenario. The most suitable protocols among the existing cluster-based NETL protocols are the HGC, the Event-Driven Clustering protocol and the Hierarchical Energy-Efficient Routing with Ripple-Zones protocols. Among the non cluster-based NETL protocols, the RRDSE and the POWER-SPEED protocols entail some suitable features. These protocols are shortly revisited and suitable features highlighted as follows.

5.5.1.1 Cluster-based Protocols

The Hierarchical, Energy-Efficient Routing with Ripple-Zones protocol (of section 3.3.3.2) features a good network structure setup for excessively large WSANs, by proposing the self-organization of actuator and sensor nodes into separate domains and zones. Aggregation masters that also have the role of MST-root are identified within each domain. The remaining sensor nodes then choose the closest master and claim themselves to be a member of it. A master/root node and its members form a zone and collectively maintain their own id array. This protocol employs a proactive routing approach for intra-zone communication, by routing along transmitter power weighted edges noted in a minimum spanning tree. The problem with proactive routing is that it requires frequent large scale broadcasting for path maintenance, which adds an unwanted traffic load to the network. For the inter-cluster communication, direct or multi-hop transmissions are proposed between aggregation masters towards the actor node. However, the distance between the aggregation masters is comparatively large and requires higher powered sensor node radios with high transmitter ranges, especially for direct transmissions. The use of costly higher powered radios is not feasible for WSANs. Moreover, the protocol makes use of aggregation masters within each zone that aggregate data inside the zone. Rather than forwarding packets directly to the next zone, nodes send their packets to the aggregation node within their zone first. This may lead to an increases path length and consequently increased power consumption and latency. Another less favoured feature is the employment of power control. It renders MACL carrier sense operations ineffective. Power control may lead to a slight decrease in energy consumption but creates serious difficulties for the underlying MACL protocol, which commonly makes use of carrier sense operations to mitigate the hidden terminal problem. Finally, the crucial real-time requirement is not directly addressed by the protocol.

The cluster-based HGC protocol (of section 3.3.3.2) is based on the GAF protocol and proposes a simple geographic anycast routing approach that sends the sensed data to the nearest cluster-head or actor node. The network area is subdivided to form a virtual grid of clusters each containing a sensor node as cluster-head. No cluster-head rotation is specified. In the 2nd coordination level cluster-heads route data towards the closest actor node. This protocol features very low complexity, but the network lifetime may be compromised by the selection of sensor nodes as permanent cluster-heads, since these have to be kept switched on for longer periods of time. Although is it specified that nodes selected as cluster-head may not be further than the radio transmission range R apart, the second level of coordination may require higher powered sensor node radios with high transmitter ranges for medium to low density networks. The use of high powered radio transmitters lowers the cost effectiveness and increases energy consumption of resource constrained WSAN sensor nodes. The second coordination level may be replaced all together, by employing an aggregation algorithm at the APPL for example, to ensure feasibility. This would mean that there is no need for aggregation cluster-head nodes that would require higher powered high energy consuming radios, prolonging the network lifetime. For medium density networks, the actor nodes could then take on the role of cluster-head.

The Event-Driven Clustering protocol (of section 3.3.3.2) features a suitable greedy geographic routing scheme for WSANs. However, clusters are created on-the-fly every time an event occurs. This may require excessive overhead for frequent event occurrences and periodic data gathering applications. Consequently, high energy consumption and shorter network lifetime may be experienced. This clustering approach may be more suitable for WSANs with highly mobile nodes.

5.5.1.2 Non Cluster-based Protocols

Although the existing non cluster-based protocols for WSANs introduce huge disadvantages by not defining a network structure, some manage to effectively address the latency and energy efficiency requirements. The RRDSE protocol and the POWER-SPEED protocol are both reactive routing protocols. A major flaw with these seemingly superior routing protocols, is that they were designed and evaluated in conjunction with the inapt power-inefficient 802.11 MAC protocol. This MAC protocol standard is designed for WLANs. As stated in section 3.3.3.3, this MAC protocol is not at all suitable for WSANs which strive towards high network lifetime. The 802.11 MAC protocol is not designed for low powered sensor nodes but rather for energy resource rich devices such as laptop computers or cellular phones to join a wireless LAN widely used in the home, office and some commercial establishments [43]. This protocol assumes the nodes to be always switched-on. Sensor nodes will suffer from energy depletion very quickly, fragmenting the network and leading to total network failure. Hence, the use of the 802.11 MAC protocol is totally inappropriate for WSANs since it will excessively lower the network lifetime and totally undermine the feasibility of WSANs.

Nevertheless, the RRDSE protocol operates by reactively calculating the next-hop delay and prioritizing the sensed data. However, the RRDSE protocol has little consideration for energy minimization. It requires high overhead with control packets adding to network congestion. Furthermore, it is unclear how the data prioritization is done since no APPL protocol is specifically proposed. The NETL needs to collaborate or interact with the APPL in order to correctly prioritise sensed data and determine latency bounds. Finally, the delay calculations do not consider the underlying timing properties of the MACL scheme. The data rate calculations within the RRDSE routing algorithm are done with network nodes that are always switched on. This means that if the RRDSE routing protocol was to be implemented in conjunction with a MAC protocol that has been designed for low powered sensor nodes (i.e. employing duty cycles), the data rate calculated will be incorrect since no MAC timing properties are considered. With the 802.11 protocol, the actual added delay due to the MACL is minimal when compared to a suitable WSAN MACL protocol. A feasible MACL protocol for WSANs would be contention-based and involve addition delay calculation challenges. The RRDSE calculates the delay at the NETL level and does not consider delays at the MACL.

The POWER-SPEED protocol also employs a reactive routing approach by estimating the forward route delay without the need for overhead. It involves the expiry time of a packet and the total energy consumption in forwarding it towards an actor node. POWER-SPEED sets out to propose a relatively complex solution to the latency vs. energy trade-off. In principle, it is a very suitable approach for WSANs but it is not clear, as with the RRDSE protocol, how the expiry time or latency bound of a packet are obtained and evaluated. This is not a trivial task with latency bounds that will be different and need to be determined by an APPL for every hop distance from the actor node. This task is exacerbated with the absence of a proper virtual network structure which is found in cluster-based NETL protocols. Furthermore, POWER-SPEED is not designed to handle dynamic network conditions since historic upstream delay information is used. Unexpected events will incur an increased traffic load and compromise delay estimations calculated with historic data. Another difficulty with the protocol is that it proposes that highly delay-sensitive packets are to be routed faster by increasing the transmitter power and simply skipping intermediate nodes. For this to be effective, routing tables require information of 2 to 3-hop neighbours. Firstly, this requires very complex and congestion inducing network setup and maintenance procedures and secondly, incurs high memory demands that are not feasible for resource constrained sensor nodes.

Moreover, as with the Hierarchical, Energy-Efficient Routing with Ripple-Zones, POWER-SPEED employs transmitter power adjustments to reach only the intended destination. This may often render MACL protocols that employ carrier sense ineffective. This leads to a worsened hidden terminal problem and increased collision probability for contention-based MAC protocols. POWER-SPEED promotes efficiency at the NETL without considering the negative effects the proposed features incurs on the other protocol stack layers. Furthermore, the transmission power is normally set at the PHYL and not within a strictly layered NETL. Therefore, this protocol cannot function without interaction with the PHYL via some kind of cross-layer framework.

As the RRDSE protocol, POWER-SPEED calculates the delay at the NETL level and does not consider delays at the MACL. Due to MACL duty cycles (commonly used in low powered MACL protocol designs) the delays introduced at the MACL cannot be estimated by the NETL without having more detailed timing information or history. Therefore, the simple delay estimates made by POWER-SPEED for downstream routes based on historic upstream data, does not provide superior accuracy at all. Delays are further strongly influenced by among others, the current channel condition, MACL queue size and network traffic load.

As revealed by the WSAN routing literature review in section 3.3.3, several non cluster-based routing protocols were designed in conjunction with the energy-inefficient MAC 802.11 protocol. This fact makes them unsuitable for deployment in WSANs since the MAC 802.11 protocol assumes the nodes to be always switched on. With permanently switched on nodes, the latency efficiency issue is far less prominent and requires little attention in comparison to

CPS designs that employ feasible MACL protocols specifically designed for low-powered sensor nodes. These MACL protocols employ sleep cycles and contention schemes, which introduce irregular delays and several new challenges.

5.5.1.3 Discussion

In the light of these five protocol evaluations, it is noted that the most suitable NETL scheme for WSANs is a cluster-based protocol that employs reactive routing. A cluster-based approach achieves favoured partitioning and division of the network into clusters. Herewith, latency, energy and reliability efficiency can be optimized locally, within a cluster and ensures a much more focused approach. Furthermore, by partitioning the network, the challenges of providing some time synchronization between nodes and data aggregation within the APPL protocol, are also simplified. A reactive routing approach supports higher adaptability to changing network conditions. In the next section, a layered NETL protocol design is given, which serves as a basis for the adaptation of the cross-layer paradigm.

5.5.2 Cluster-based Hierarchical Energy-Aware Geographic Routing NETL Protocol Design and Detailed Operation

The operational details of a layered NETL protocol for WSANs, as implemented on all *sensor* nodes inside medium to low density networks are proposed in this section. The proposed design presents a Cluster-based Hierarchical Energy-Aware Geographic Routing Protocol (CHEAGR) for WSANs. The protocol is primarily based on the HGC protocol and serves as a basis towards the design of a new cross-layer interacting protocol (section 5.8.2) that extensively incorporates information from other CPS layers to promote latency and energy efficiency. CHEAGR entails minor modifications to the HGC protocol in order to accommodate the defined focus area and ensure feasibility.

First of all, it is proposed that the clusters are not determined by a virtual grid. The virtual grid approach is not viable, since informing the sensor nodes of the dimensions and borders of a virtual grid requires immense overhead and coordination. It is proposed that clusters are simply formed by allowing sensor nodes to assign themselves to the closest actor node. The ratio of sensor nodes to actor nodes is assumed to be around 50:1 for a typical application. Clusters are hence formed around every actor node to reflect an organisation of the 2D plane into a Voronoi diagram, which then takes on the role of cluster-head. The 2nd coordination level of the HGC protocol is removed altogether, by rather employing an aggregation algorithm at the APPL for example, to ensure feasibility. Consequently, there is no need for dedicated aggregation nodes and further subdivision of clusters. In the 2nd coordination level of the HGC protocol, packets are first aggregated towards the local aggregation node and only then sent towards the actor node. This results in a route that involves more nodes and consequently consumes more energy and adds latency. Furthermore, without the 2nd level of coordination, sensor nodes do not require higher powered, high energy consuming radios and are more adaptable to dynamic network conditions. Using low powered shorter range radio transceivers consumes less energy and consequently prolongs the network lifetime.

Secondly, with regards to the routing of data packets, both the HGC and Event-Driven Clustering protocols make use of the favoured geographic routing approach. HGC is based on GAF, which can be considered as a hierarchical protocol [28]. CHEAGR also makes use of a geographical routing approach, in a hierarchical sense, by including the hop distance to the actor node as a dominating part of the packet forwarding criteria of sensor nodes. This paves the way for the usage of a less accurate localisation and positioning algorithm, resulting in a lower protocol complexity and increased efficiency. Furthermore, the original GAF state transitions are not included since the task of putting a node to sleep etc., is commonly performed by the MACL protocol.

The issue of latency minimization within the NETL of sensor nodes is primarily addressed by conducting shortest path geographic routing towards the actor node. Latency minimization is enhanced later on, by the new cross-layer interacting protocol design (section 5.8.2).

The proposed CHEAGR NETL protocol involves an initial network setup phase and a normal operational phase. These two phases are discussed in the following subsections.

5.5.2.1 Network Setup and Neighbourhood Discovery Phase

The issue of setting up a network structure is often omitted by existing WSAN protocols. The proposed setup phase involves the self-organization of actor and sensor nodes into separate domains or clusters. These clusters are clearly defined, well-sized virtual domains that have the task and ability of locally optimizing the latency and energy efficiency, to ensure the feasibility of the overall WSAN.

The formation of clusters around actor nodes begins with the broadcasting of an initial broadcast packet by all actor nodes and once received, by all sensor nodes in the WSAN as illustrated in a simplified WSAN diagram in Figure 5.4.

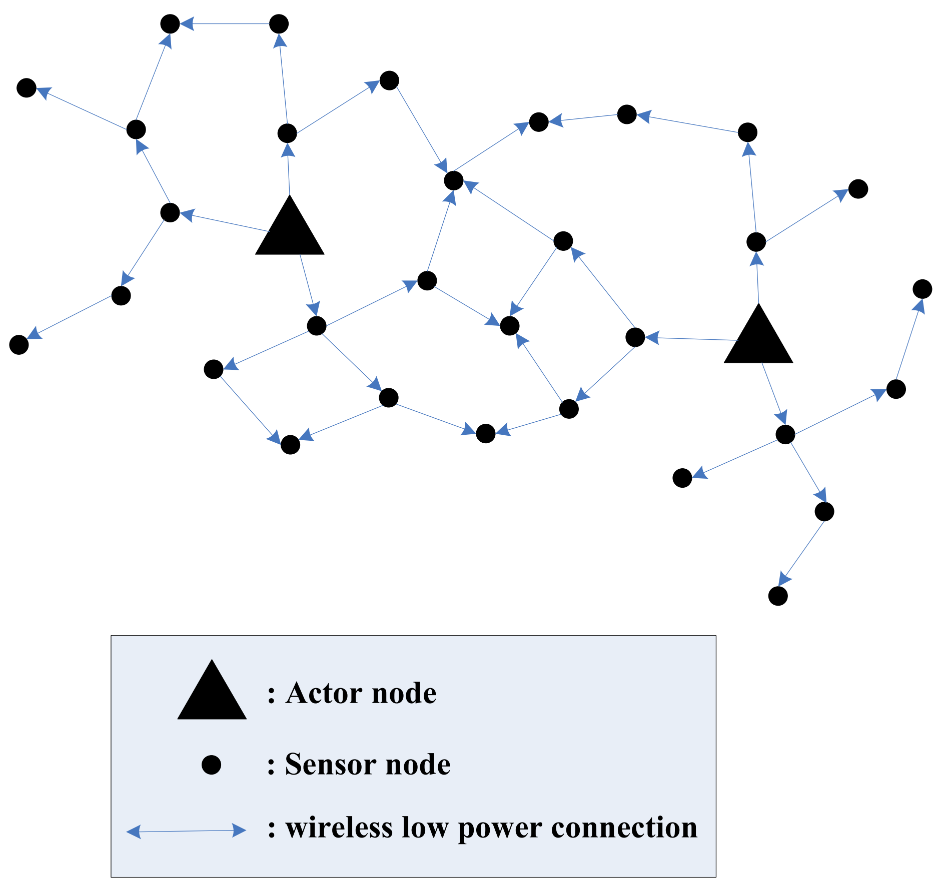

Figure 5.4 Simplified initially non-clustered WSAN with initial setup procedure broadcast packet directions.

The aim of the broadcast packet is twofold. Firstly, the sensor nodes assign themselves to a particular cluster with an actor node as cluster-head and secondly, the sensor nodes discover and gather information of neighbour nodes. Every node features a neighbourhood table that is used to store information of neighbour nodes within transmission range, i.e. single-hop neighbours. The table content is limited to single-hop neighbours since limited resourced sensor nodes in WSANs have a small amount of memory space. The neighbourhood table is populated in the setup and neighbourhood discovery phase.

The initial broadcast packet sent from an actor node contains the packets source address, the actor nodes' unique Id and a hop count. Since the number of actor nodes in the network is low, a short unique address can be assigned to each one.

Meanwhile, sensor nodes within the WSAN wait for the broadcast packet to arrive. On arrival, the sensor nodes handle the setup broadcast packet as illustrated by the flowchart of Figure 5.5. When the sensor nodes, which are direct neighbours of an actor node, receive the setup broadcast packet for the first time, the source actor nodes' unique id is stored in a parameter called MyActorId, and the hop distance to the actor node, in a parameter called Myhops2Actor. The actor node is then added as a neighbour in the neighbourhood table.

To this end, only the direct neighbours of the actor nodes have received the broadcast packet and now belong to a specific cluster, identified by an actor node's unique id. To ensure that a broadcast packet traverses further through the network, the first sensor nodes that received the broadcast packet in turn broadcasts the packet further. Before broadcasting the packet, a sensor node changes the packet's source address to its address and increments the hop count. Finally, to prevent broadcast packet collisions, a sensor node waits for a random time period before transmitting the modified broadcast packet.

In the setup phase, the broadcasting of the initial setup broadcast packet by all sensor nodes enables them to assign themselves to a cluster with a specific actor node as cluster-head and to discover who their neighbours are. Every sensor node broadcasts the setup packet of a particular actor node only once, to prevent an endless broadcast storm. Sensor nodes stay active throughout the duration of the setup and neighbourhood discovery phase, which is limited to a specific time, dependent on the network size.

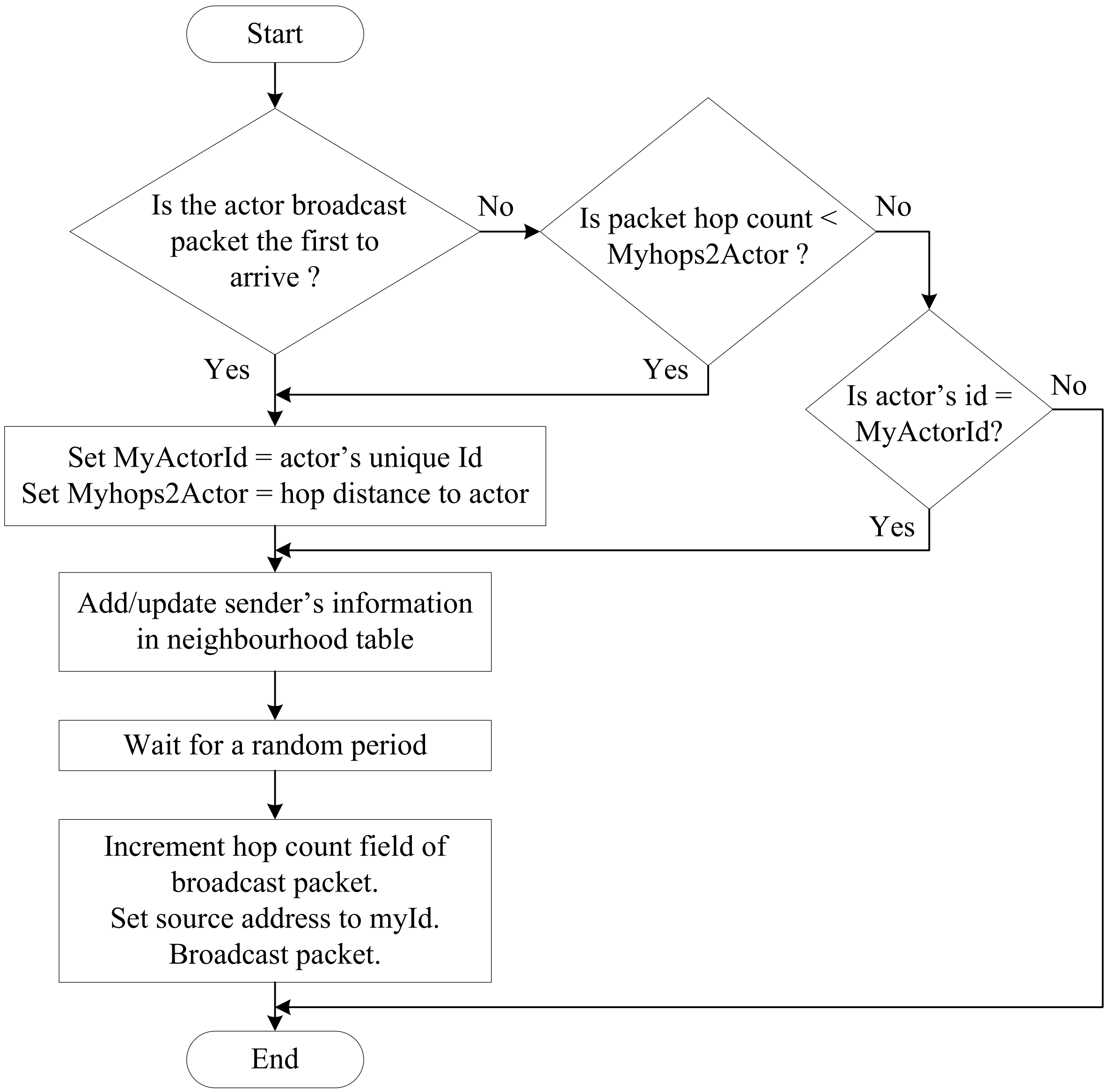

Figure 5.5 Flowchart of the NETL protocol initial broadcast packet handling.

If a sensor node has already received its first broadcast packet and assigned itself to that originating actor node and cluster, all subsequent broadcast packets that contain an equal or larger hop count and the same actor id as the initially stored Myhops2Actor and MyActorId parameters respectively, are used to populate the neighbourhood table. The neighbourhood table initially only contains node addresses and their hop distance to the actor node. If however, a packet is received containing a lower hop count parameter and the same actor id as initially stored Myhops2Actor and MyActorId parameters respectively, the sensor node stores the new shorter hop distance to the actor node as its own and then broadcasts the packet further.

In borderline cases, where sensor nodes are situated on the border of a cluster and have similar hop distances to two actor nodes, are handled as follows. A sensor node always assigns itself to an actor node with the shortest hop distance. If the hop distances to two actor nodes are equal, the sensor node assigns itself to the actor node from which it received a broadcast packet first. Finally, if a sensor node receives a broadcast packet with an equal or larger hop count as the initially stored Myhops2Actor parameter and a different actor id as the initially stored MyActorId parameter, the broadcast packet is discarded.

This setup procedure presents a solution that enables sensor nodes to determine to which virtual cluster they belong to and simultaneously to gather information of neighbour nodes' address and hop distance to the actor node. Once setup, the network needs very little overhead.

The addressing scheme that is employed in the network involves the use of 16 bit addresses which allows a total of 65536 (2^{16}) addresses. Each node is assumed to have a single unique address within the network. This address is used by both the NETL and the MACL protocol to evade complex NETL to MACL address conversions.

5.5.2.2 Normal Operation and Routing phase

The normal operation and routing phase is entered by a network node once it has assigned itself to a cluster. The construction of virtual clusters around actor nodes allows sensor nodes to adopt a single-actor approach and focus on routing and relaying sensed data towards a single actor as illustrated in the simplified WSAN in Figure 5.6. The sensor nodes are not burdened with complex coordination issues, but are primarily concerned with getting the sensed data as fast and as energy efficiently as possible to the actor node that they assigned themselves to. A two level hierarchy exist, with all sensor nodes residing in the first level, and all actor nodes in the second level.

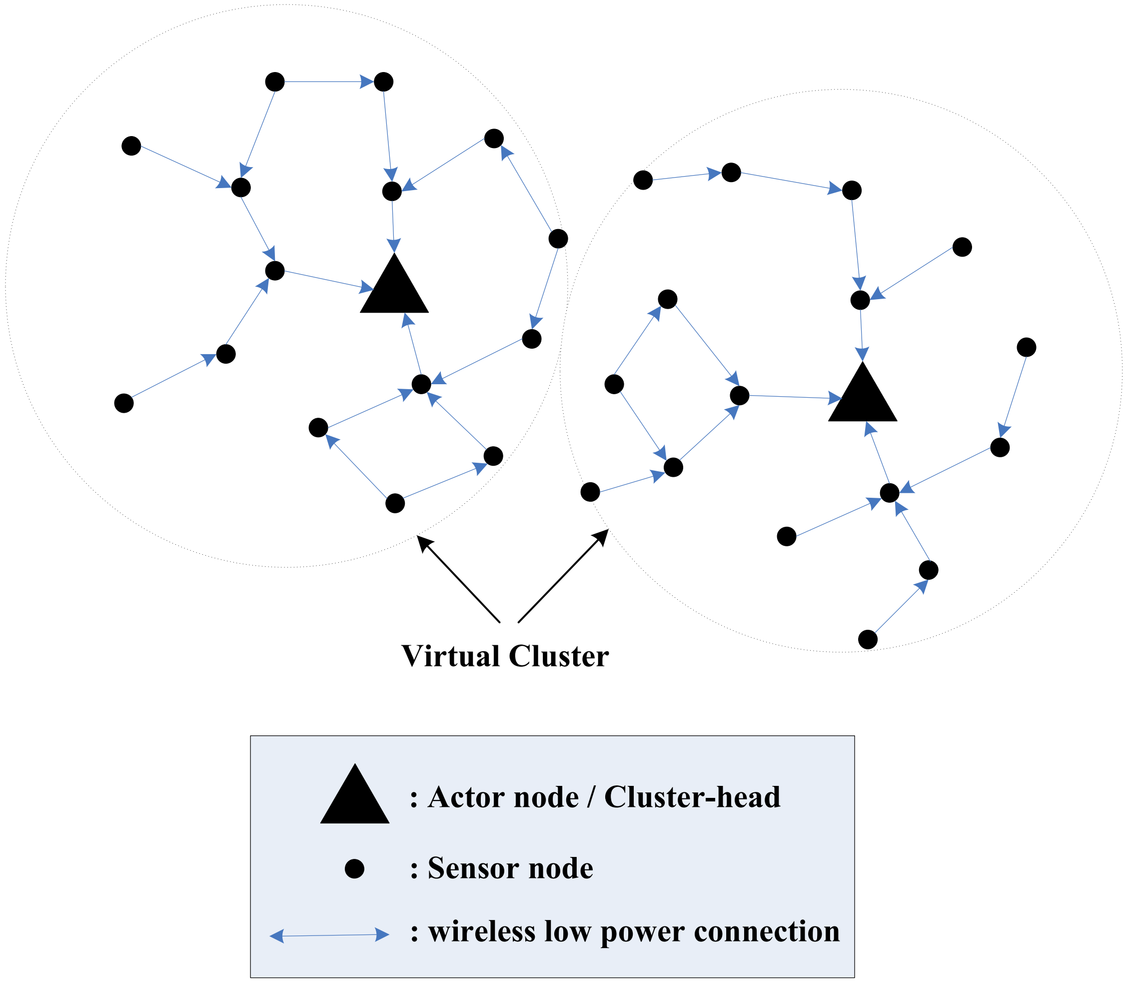

Figure 5.6 Simplified WSAN with resulting virtual clusters and routing directions.

CHEAGR makes use of a reactive routing approach to achieve minimum delay and energy efficient routing. Once the NETL receives a data packet from the APPL, it needs to route it towards the cluster-head actor node. In these types of networks, there exists a general trade-off between latency and energy efficiency. Many of the existing WSANs routing protocols try to solve this trade-off, where latency may be minimized but at the cost of energy efficiency and vice versa. A common approach is hence to find a balance where latency minimization and energy efficiency are acceptably achieved. CHEAGR hence reactively chooses a route that achieves this balance.

5.5.2.2.1 Forwarding of Sensed Data Packets

After receiving a data packet from the APPL protocol, CHEAGR reactively chooses the next-hop node that will ensure minimum latency and energy efficient. Three metrics are used in determining the next-hop node. The first metric is the *hop distance* to the actor node, the second is the *residual energy* E_R of a node and the third is the current *traffic load* experienced by a node. The residual energy and traffic load metrics enable the NETL to react to changing network conditions. These metrics are obtained by keeping record of neighbour nodes' hop distance, traffic load and residual energy inside the neighbourhood table. A table of only the direct neighbour nodes within transmission range is kept, since route forwarding only reactively determines the next single-hop neighbour and not the entire route towards the destination actor node. Information contained within the neighbourhood table is given in Table 5.2.

Neighbour Info	Description
Neighbour address	Unique address of a direct neighbour node within the same cluster.
Hop distance from actor node	The shortest hop distance between a sensor node and its associated actor node.
Residual Energy	Residual energy of neighbour node.
Traffic Load	The traffic load as measured by a neighbour node.

Table 5.2. WSAN CHEAGR NETL protocol neighbourhood table entries.

The first metric that is considered is the hop distance to the actor node. Routing a data packet to the actor node on the shortest path ensures that no unnecessary nodes are involved. In this way minimum energy is consumed, maximising the network lifetime. Furthermore, the less nodes a data packet travels through, the shorter the delay of reaching the intended destination actor node. Therefore, minimizing the number of nodes a data packet travels through provides minimum latency and energy consumption. After being passed a data packet from the APPL, CHEAGR checks the neighbourhood table for prospective nodes that are closer to the actor node than itself in a greedy geographic approach. If there are several nodes that have equal hop distances to the actor node, these will enter a second selection round.

The second selection round involves the residual energy E_R and traffic load metrics. Choosing the sensor node that has the higher residual energy achieves good energy-efficiency by spreading the traffic load to ensure maximum network lifetime and prevent early network fragmentation. If the same sensor nodes are repetitively used, their energy will quickly be depleted and 'holes' would appear in the network. Holes fragment the network and may cut off sensor nodes or groups of sensor nodes from their actor node. Therefore, the residual energy of potential next-hop sensor nodes with equal hop distance is secondly considered to prolong the network lifetime.

The traffic load metric of a sensor node is calculated from the number of data packets the NETL handled, i.e. the sum of packets received from the APPL and the MACL, over a specific period of time $T_{Trafficload}$ and the average sizes of those data packets that have been handled. Specifically, the traffic load is calculated as

$$TrafficLoad = NumberOfPacketsHandled \times AvePacketSize\,. \tag{5.1}$$

Sensor nodes with high traffic loads consume much more energy than nodes with low traffic loads. Furthermore, increased latency is experience by data packets that traverse through very busy nodes. The size of the data packets being handled by the NETL is important since larger files require longer transmission times and hence higher energy consumption.

The residual energy and traffic load metrics are considered jointly in the second selection round of potential next-hop candidates. Between sensor nodes with equal hop distance, the candidate node boasting the highest ratio of

$$\frac{\mathrm{E_R}}{TrafficLoad} \tag{5.2}$$

is chosen as the next hop in the route towards the actor node. This approach enables a node to choose more intelligently between next-hop candidates with very similar residual energy's.

A reactive routing is achieved by determining the most suitable next-hop node among nodes that have a shorter hop distance to the actor node cluster-head. If a sensor node does not find nodes with a hop distance to the actor node lower than its own, nodes that have an equal hop distance are then considered as candidates and also selected according to their residual energy and current traffic load. The discussed next-hop selection process is depicted in Figure 5.7. By reactively considering the three NETL metrics, the protocol achieves min delay and maximizes the network lifetime. The latency vs. energy trade-off is handled by choosing a route that best suites both requirements.

The scalability requirement is satisfied by the proposed reactive routing approach, due to the fact that the routing of sensed data within the self-organizing clusters is not influenced by the cluster size or the number of nodes present.

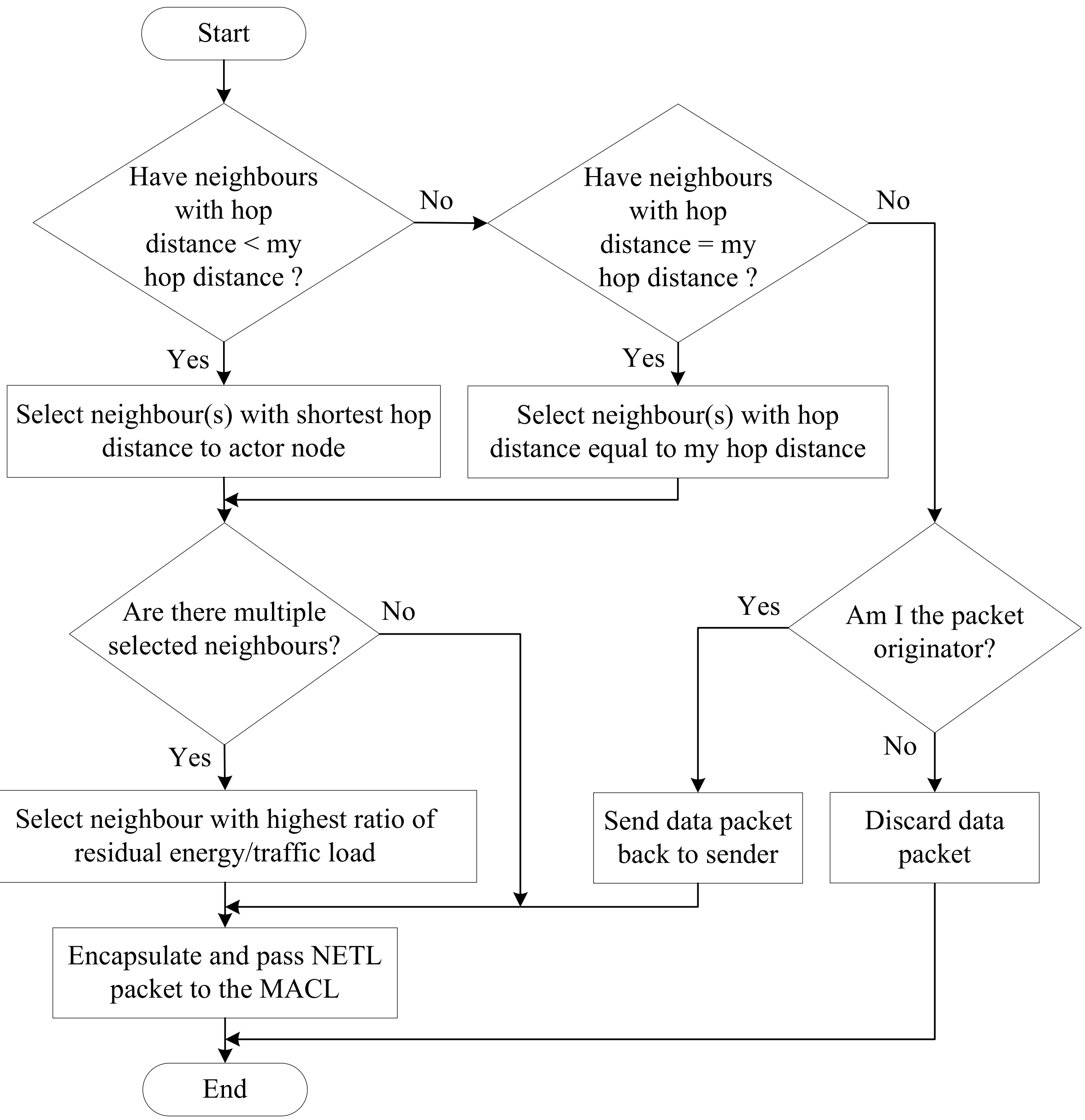

Figure 5.7 The next-hop neighbour selection process.

5.5.2.2.2 Ensuring Reliable Routing

In WSANs data packets should reach the intended actor node with satisfactory reliability. The task of ensuring that a data packet reaches the intended destination node is assigned to the MACL protocols. The MACL protocol only passes the data packets back to the NETL if it reached its re-transmission limit. The MACL protocol may reach its re-transmission limit due to the destination node failing to respond or due to repeated channel collisions. In this case, the data packet is passed back up to the NETL. The NETL will recognize that it is the originator of the data packet by checking the source address and deduce a transmission failure. The NETL protocol then determines an alternative next-hop node, according to the abovementioned forwarding scheme, to which the data packet should be sent. The data packet is then passed to the MACL with a new destination address. If no alternative next-hop node exists, and the packet was received from another node, the packet is sent back to the sender with a notification that it is unable to route the packet further. If the node is the originator of the data packet and no alternative next-hop nodes exist, the packet is discarded.

5.5.2.2.3 Receiving Sensed Data Packet

A sensor node may receive a data packet in one of two instances. Firstly, by being the destination node of a data packet or secondly by overhearing a data packet destined for a nearby node. A node only considers a data packet, if it contains the node's address in its destination field. If a data packet was overheard and contains the incorrect address it is subsequently discarded in the NETL.

5.5.2.2.4 Dispersion of NETL Metric Updates

To ensure effective reactive routing and reaction to changing network conditions, nodes require up-to-date information of forward nodes in the route towards an actor node. The effectiveness of the routing decisions relies on the correctness and accuracy of NETL metrics in the neighbourhood table. Outdated information may result in detrimental routing decisions. However, dispersing NETL information updates is not a trivial task. If done too often, an unnecessary traffic load is incurred and when not done frequent enough, the correctness of the current status of metrics at a node is undermined.

In the neighbour discovery phase, nodes discovered neighbour addresses and hop distances to the actor node. This information is stored in the NETL neighbourhood table. However, in the normal operation phase, the routing decision is, in addition to the hop distance metric, based on the E_R and traffic load metrics. Therefore, the NETL of a node has to instigate the dispersion of NETL metric updates to neighbouring nodes which have hop distances equal to or greater than its own hop distance. For densely populated networks, it may be sufficient to send updates only to nodes with greater hop distances.

The frequency of NETL metric update dispersions is determined according to the consumed energy and the current traffic load at a node. For nodes with a low traffic load, the NETL metric update dispersion is triggered every time a specific amount of energy E_C is consumed. On the other hand, for nodes that experience constant high traffic loads and consequently consume a higher amount of energy during a given period, E_C is doubled to minimize the added NETL metric update dispersion traffic. When the traffic load decreases, E_C is halved again, to maintain satisfactory accuracy. E_C is dynamically changed according to the current traffic load of a node, to prevent an increase in congestion due to NETL update packets at an already busy node.

If the frequency of update packet dispersions is done in a periodic fashion, sensor nodes that experience low traffic are forced to disperse update packets while their specific metrics have not changed much. This may incur a higher traffic load on the network. On the other hand, for sensor nodes that experience higher traffic conditions and consequently consume more energy, a too long period of dispersion may lead to an inaccuracy of current residual energy. This would deem the load balancing routing approach as ineffective. A too short period of dispersion would conversely incur increased congestion.

5.5.2.2.5 Mobility

As mentioned in section 5.4.4, it is assumed that the actor nodes mobility may be somewhat limited. This is ascribed to the fact that the number of actor nodes is sufficiently high and that they are well spread to ensure well-sized clusters. This enables actor nodes to cover a large percentage of their cluster and hence limit their mobility. Therefore, actor nodes will only be required to move if an event or disturbance is detected within its cluster and its actuation unit action range does not cover the problem target area. The actor node then moves closer to the target area such that the actuation unit may resolve the critical event. The limited mobility of actor nodes requires only minimal reconfiguration of a single cluster within the WSAN, since the actor node is never required to leave its cluster. Once the event is resolved, the actor node moves back to its original position and the original network configuration is reclaimed.

5.5.2.3 NETL packet formats

The CHEAGR NETL protocol involves the use of three different packet formats. The first packet format is the initial broadcast packet illustrated in Figure 5.8, initially sent out by the actor nodes in the setup and neighbour discovery phase. The second packet format, depicted in Figure 5.9, is used for normal sensed data packets. Finally, the NETL metric update packet, as depicted in Figure 5.10, is used for sending metric updates to neighbour nodes.

Type	Source Address	Actor unique Id	Hop count
4 bits	16 bits	16 bits	8 bits

Figure 5.8 The format of a CHEAGR initial broadcast packet.

Type	Source Address	Destination Address	APPL packet
4 bits	16 bits	16 bits	Variable

Figure 5.9 The format of a CHEAGR normal data packet.

Type	Source Address	Destination Address	Actor unique Id	Residual Energy	Traffic Load
4 bits	16 bits	16 bits	16 bits	16 bits	16 bits

Figure 5.10 The format of a CHEAGR NETL metric update packet.

- The *Type* field holds the type of packet and includes ACTOR_INIT_BROADCAST_MESSAGE, SENSOR_DATA_PACKET and SENSOR_NETL_INFO_UPDATE_MESSAGE packet types.
- The *Source Address* and *Destination Address* fields contain the addresses of the source node and the address of the intended destination node respectively. The initial broadcast packet contains no destination field since it is not intended for a single destination node.
- The *Actor unique Id* field holds the actor node's unique address as cluster-head of a specific cluster. All sensor nodes within a specific cluster have the same actor node as cluster-head and identify themselves as part of the cluster with the unique actor node's address.
- The *Hop Count* field found in the initial broadcast packet is used by sensor nodes to set their hop distance to the actor node. This value is incremented before broadcasted further through the network.
- The *APPL packet* field contained in the normal NETL data packet is the encapsulated APPL sensed data packet. The size of this field depends on the APPL sensed data packet size.
- The *Residual Energy* field contains the value of E_R of the source node.
- The *Traffic Load* field contains the current traffic load value of the source node.

5.5.2.4 Summary of Protocol Features and Main Parameters

The main protocol features of CHEAGR are summarized as follows.

- Clustering of self-organizing sensor nodes around closest actor nodes.
- Geographic routing involves hop distance to actor node, residual energy and current traffic load.
- Minimum hop distance as primary routing criterion to achieve minimum latency.
- Residual energy included to balance traffic loads among nodes and prolong network lifetime.
- Traffic load included to achieve adaptability to changing network conditions.
- Neighbourhood table contains only direct neighbour nodes within transmission range.

- Once setup needs very little overhead.
- Simplified addressing scheme due to clustering.
- NETL metric status-update packet dispersion frequency determined by consumed energy of a node.

Table 5.3 shortly summarizes the main NETL parameters.

Name/Symbol	Description	Value
MyActorId	The id of the cluster-head actor node to which a node has assigned itself to.	Unique 16 bit actor node id.
Myhops2Actor	A nodes' hop distance to the actor node.	Variable
myNeighboursList	The neighbourhood table that holds NETL information of neighbour nodes within transmission range.	Addresses and node ids
$T_{Trafficload}$	Time period of measured traffic load	500 seconds
myTrafficLoad	The current traffic load of node	Variable
NumberOfPacketHandled	The number of packets handled during the time $T_{Trafficload}$	Variable
AvePacketSize	The average packet size of packets handled during the time $T_{Trafficload}$	Variable
E_C	The amount of energy that is consumed before a NETL metric update is dispersed towards upstream nodes.	Dependent on the application.
E_R	The total residual energy of a node.	Variable

Table 5.3. Summary of the main CHEAGR NETL protocol parameters.

5.5.3 Identified NETL Parameters for Cross-Layer Interaction

The cross-layer framework as proposed in section 5.3 provides a fitting neighbourhood table structure that may be used by a node to store information of direct neighbour nodes, i.e. neighbours within its transmission range. Specific CHEAGR NETL parameters and features that are useful and beneficial to other layers in the CPS towards promoting overall latency minimization and energy efficiency improvements are identified as follows.

a. The *NETL status information* parameter is inserted into the information pool structure of the SCLF. The NETL status may be mainly used as an indication to other layers, to signify that the node has assigned itself to a cluster and has at least one neighbour node in its neighbourhood table.

b. The *neighbourhood table* of the CHEAGR NETL contains information of neighbouring nodes that are within transmission range. As proposed by the SCLF, the table is moved to the neighbourhood table structure inside the SCLF to avoid having duplicates of the same information. The SCLF neighbourhood table structure entails neighbour information from both the NETL and the MACL. The NETL parameters contained in the table include the neighbour nodes' **address**, **hop distance** to its actor node, **traffic load**, **residual energy** E_R and a **timestamp** of when the last update was made. The residual energy is not an inherent NETL parameter, however it is considered here since the NETL uses it extensively in its routing decisions.

c. The *Traffic load* may be useful to other layers of the CPS to enable adaptation to changing traffic conditions. It is stored in the information pool structure within the SCLF.

d. The *hop distance* to the cluster-head actor node is also stored in the information pool structure of the SCLF for other layers to view and may be used by an aggregation algorithm for example.

d. Loose encapsulation of the NETL packets is proposed to enable the MACL to read particular information contained within the NETL packet header. Specifically, the NETL *packet type* parameter is of primary interest. This parameter may prompt the MACL to handle and react differently to different types of packets. It is a cross-layer feature that is an important attribute of the cross-layer paradigm. A loose encapsulation feature enables the MACL to distinguish between NETL metric update packets and different types of data packets.

The actual incorporation and usage of the identified NETL parameters by other layers in the CPS is discussed in section 5.8.

5.5.4 Notes on the NETL protocol for actor nodes

The NETL protocol tasks for actor nodes, as stipulated in section 3.3.1, specifically includes the communication between actor nodes and communication between actor nodes and sensor nodes. These are shortly discussed in the following subsections.

5.5.4.1 Communication between Actor Nodes

The actor-actor communication usually involves two actor nodes communicating directly with each other in a peer-to-peer manor. It is not necessary to devise intricate routing algorithms since the APPL decides which actor nodes are involved in a specific collaboration effort. Therefore, on reception of an APPL packet, the NETL of actor nodes simply forwards it to the destination actor node which is typically only one hop away. A peer-to-peer communication then takes place.

5.5.4.2 Communication to Sensor Nodes

Actor nodes, which act as cluster-head and data gatherer within their clusters, are required to communicate to the sensor nodes if it changes its physical location significantly. As mentioned in section 5.5.2.2, actor node mobility is limited and an actor node only moves if it needs to resolve an event within its cluster that is unreachable by its actuation unit from its current position. If necessary, the CHEAGR NETL instigates reconfiguration of the cluster to temporary accommodate the actor node’s change in position. Once the actor moves back to its original position the original setup is reclaimed.

Another instance where the actor node needs to communicate to sensor nodes within its cluster is if the application stipulates a new threshold level in the APPL protocol. The new threshold level is broadcasted to all the sensor nodes within the cluster by the actor node.

5.6 LAYERED WSAN MACL PROTOCOL DESIGN AND IDENTIFICATION OF CROSS-LAYER PARAMETERS

The WSAN CPS MACL protocol for sensor and actor nodes respectively, has to ideally perform all the primary tasks as stipulated in section 3.4.1. The proposed MACL protocol in this section implements functionality for effectively accessing and sharing of the wireless medium, as suitable for the defined basic operational scenario. After considering some applicable existing WSN MACL literature, the most suitable layered WSAN MACL protocol is selected to serve as a design basis for the adaptation of the cross-layer paradigm. MACL parameters are then identified that are beneficial to other layers in the CPS, as part of the cross-layer paradigm, towards promoting the optimization of latency and energy efficiency. Finally, some notes concerning the MACL of the WSANs actor nodes are given.

5.6.1 Applicable Existing Literature

The existing literature relating to ultra low powered MACL protocols for WSANs and WSNs was discussed in section 3.4.4 and 3.4.3 respectively. To distinguish between different types of MACL protocols, the much researched field of WSNs MACL protocols reveals that these protocols can be classified as either contention-based or schedule-based, as discussed in section 3.4.2. A suitable MACL protocol is needed for WSANs, which fulfils the defined basic operational scenario and forms the basis for development and adoption of the cross-layer efficiency optimization approach.

It is pointed out that WSN MACL protocols can be conditionally adopted by WSANs since the networks as experienced by the MACL protocol are equal in most regards. The significant difference is that WSN MAC protocols do not consider the latency minimization as one of its top priorities. They are predominantly focused on minimizing the energy consumption. Very few WSAN MACL protocols have been proposed and this field of research is still young and unexplored.

The applicability to the defined basic operational scenario and network structure of WSANs need to be considered when choosing between contention-based and schedule-based MACL approaches. The advantages and disadvantages of the two classes of WSN MACL protocols are comprehensively discussed in section 3.4.3.3 and summarized in Table 3.5. It is noted that schedule-based protocols are mostly combined with cluster-based NETL protocols. However, the proposed hierarchical cluster-based NETL protocol design in section 5.5.2 merely uses the cluster structure to assign sensor nodes to a single actor node. By assigning sensor nodes to a single actor node, the routing task is simplified and most of the decisive responsibilities, concerning event control, are shifted to the actor nodes. Hence, the sensor nodes are predominantly occupied with relaying sensed data towards a specific actor node with minimum latency and high energy efficiency. The proposed clustering approach is flexible to changing topologies and scales well to larger networks. Sensor nodes are easily added to an existing WSAN by integrating and self-configuring themselves to the network. These proposed NETL protocol features stand in total conflict with strictly schedule-based MAC protocols. Schedule-based approaches are best suited for completely static networks that maintain small sized clusters. Schedule-based MAC approaches are complex and computational intensive to maintain in a multi-hop network, due to their strict time synchronization and schedule calculation and propagation requirements. Furthermore, they are less suited for event-based operation and are not designed for networks that experience topology changes and mobility of nodes. Total reconfiguration of schedules is necessary every time a node is added to the network and when nodes fail due to energy depletion for example, precious time and energy is wasted by unused idle time slots.

The WSN MACL protocol literature review reveals that there exist a much higher number of contention-based MAC protocols. This is due to the fact that contention-based protocols are energy efficient, scale well, handle topology changes well and only need loose synchronization. The advantages, as summarized in Table 3.5 (section 3.4.3.3), mark the contention-based approach as extremely feasible for WSANs, as long as the issues of collisions and the hidden terminal problem, inherent to the contention-based approach, are handled in a satisfactory manor.

When comparing the existing WSN contention-based protocols, it is revealed that for the defined basic operational scenario involving low to medium traffic loads, the preamble sampling protocols are most relevant. The best-performing and most relevant preamble sampling protocols as discussed in section 3.4.3 includes WiseMAC, CSMA-MPS and B-MAC. The popular S-MAC protocol is outperformed by the preamble sampling protocol B-MAC [56]. Nodes that utilise preamble sampling require less synchronization and subsequently handle topology changes better than the popular S-MAC. The WiseMAC protocol served as the trend setter among the preamble sampling protocols with CSMA-MPS and B-MAC proposed as an enhancement to WiseMAC. The CSMA-MPS enables preamble length and starting point optimization while B-MAC included accurate channel sensing and key parameter adjustment capabilities. A comparison of these three preamble sampling protocols are given in Table 5.4.

Features	WiseMAC	CSMA-MPS	B-MAC
Storage of neighbours' schedules to determine optimum preamble starting point	Supported	Supported	Not Supported
Preamble stops once receiver receives first valid preamble packet	Not Supported	Supported	Not Supported

Table 5.4. WSN MACL preamble sampling protocol comparison.

The features entailed in the first column of Table 5.4 represent very important performance influencing features that promote energy efficiency. Firstly, if a sending node knows its receiving neighbour's wakeup schedule, the optimal preamble starting point can be determined. The starting point can be set to be not too early, but early enough to 'hit' the intended receiver's 'listen' slot. By preventing premature starting of the preamble transmissions, energy is saved and the channel traffic lowered. The second feature presents an energy conserving and time minimising technique, where the preamble is stopped once the intended receiver successfully receives the first valid preamble packet. An optimal preamble length is hence ensured.

It is seen from Table 5.4 that the CSMA-MPS protocol supports both beneficial energy minimization features and presents the best proposed approach to energy efficiency so far. CSMA-MPS shows the most potential for implementation for the defined basic operational

WSAN scenario. Therefore, CSMA-MPS as proposed in [54] will be used as the basis for the WSAN MAC protocol development and adoption of the cross-layer paradigm.

5.6.2 CSMA-MPS Protocol Detailed Operation

The operational detail of the MACL protocol for WSANs as implemented on all *sensor* nodes inside the network is given in this section. As stated in the previous subsection, the CSMA-MPS protocol, as proposed by Mahlknecht and Böck [54], is selected as the most suitable MACL protocol for WSANs. This contention-based MAC protocol is a preamble sampling protocol that makes use of neighbour nodes' wakeup schedules and a low duty cycle to optimize its operation and energy efficiency. CSMA-MPS operates either in an unsynchronized or synchronized mode. These modes and additional detailed features are discussed in the following subsections.

5.6.2.1 MAC Unsynchronized Mode and Neighbourhood Discovery

Upon network initialization, the CSMA-MPS operates in the unsynchronized mode. This mode is explained according to Figure 5.11. Furthermore, Figure 5.12 shows a flowchart of the basic operation. Every node inside the WSAN follows a periodic wake up cycle. As illustrated in the time line of the destination node B, the WSAN nodes wakeup with a period T_W and listen to the channel for a short period (also referred to as *listen interval*). The length of the wakeup period T_W depends on the application but remains constant throughout the network. The starting point of the listen slots are randomly chosen to spread the wakeup times of the nodes in the network.

Upon wake up, the radio is switched on and takes a time denoted by T_{RWU}, to become operational. T_{RWU} depends on the specific implemented radio hardware. Once the radio is operational, the node listens for incoming data for a duration of T_{LS}. If nothing is received, the node is put back to sleep. This cycle is repeated by every node inside the WSAN. The duty cycle can be set very low when the radio has a high bit rate. Consequently, the node spends most of its life in sleep mode and preserves precious energy.

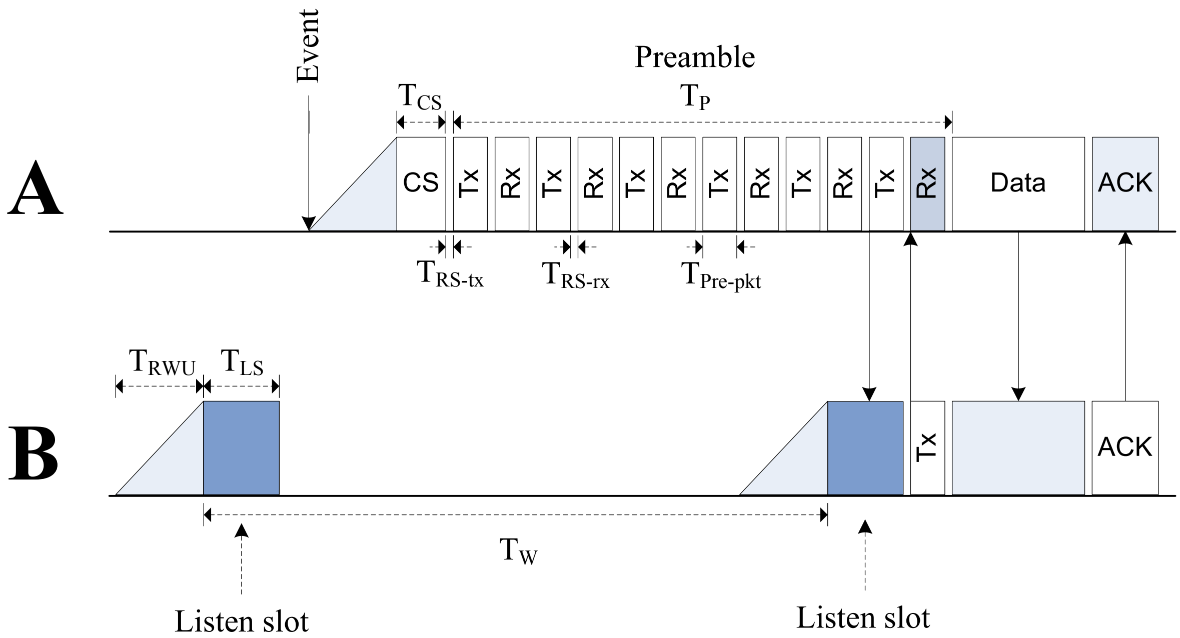

Figure 5.11 CSMA-MPS for WSANs in the unsynchronized mode.

A source node A contends with other neighbour nodes when it wants to communicate with a destination node B. Initially, in the unsynchronized mode, the source node A does not know the schedule or wakeup time of the destination node B. Therefore, A assumes a worst case preamble length of $T_P = T_W$, which ensures that it will 'hit' the destination listen slot. Before commencing with the alternating transmit-receive preamble sequence, the protocol wakes up the radio in receive mode and then senses if the channel is currently occupied. The length of the carries sense is denoted by T_{CS}. If the channel is free, the radio is switched to transmit mode and the preamble sequence started. The radio can either operate in transmit mode or receive mode and takes a time $T_{RS\text{-}tx}$ for it to switch from Rx to Tx and a time $T_{RS\text{-}rx}$ to switch from Tx and Rx mode. During the radio switch, the radio can not be utilized.

The source node A commences with the preamble sequence by transmitting a preamble packet. The preamble packet contains the node A's address in the source field and is kept very short to ensure as short transmission time as possible, to promote energy efficiency. Once the preamble packet was sent, node A switches its radio to Rx and waits for a preamble reply packet. If nothing was received, another preamble Tx packet is sent. This alternating Tx-Rx preamble sequence is continued until the destination node B replies with a preamble reply packet. As illustrated in Figure 5.11, the destination node's listen slot will be 'hit' eventually. Upon reception of a preamble packet, node B replies with a preamble reply packet with the node A's address in the destination field and its own address in the source field of the packet. Node A then receives the preamble reply packet in its next Rx preamble slot and hence knows it can safely transmit the data towards the destination node B, which had switched to Rx mode after transmitting the preamble reply. Node A switches its radio to Tx again before transmitting the concerned data packet. After the data has been sent, node A switches to Rx mode and waits for an acknowledgement packet from node B, signifying the successful reception of the data packet.

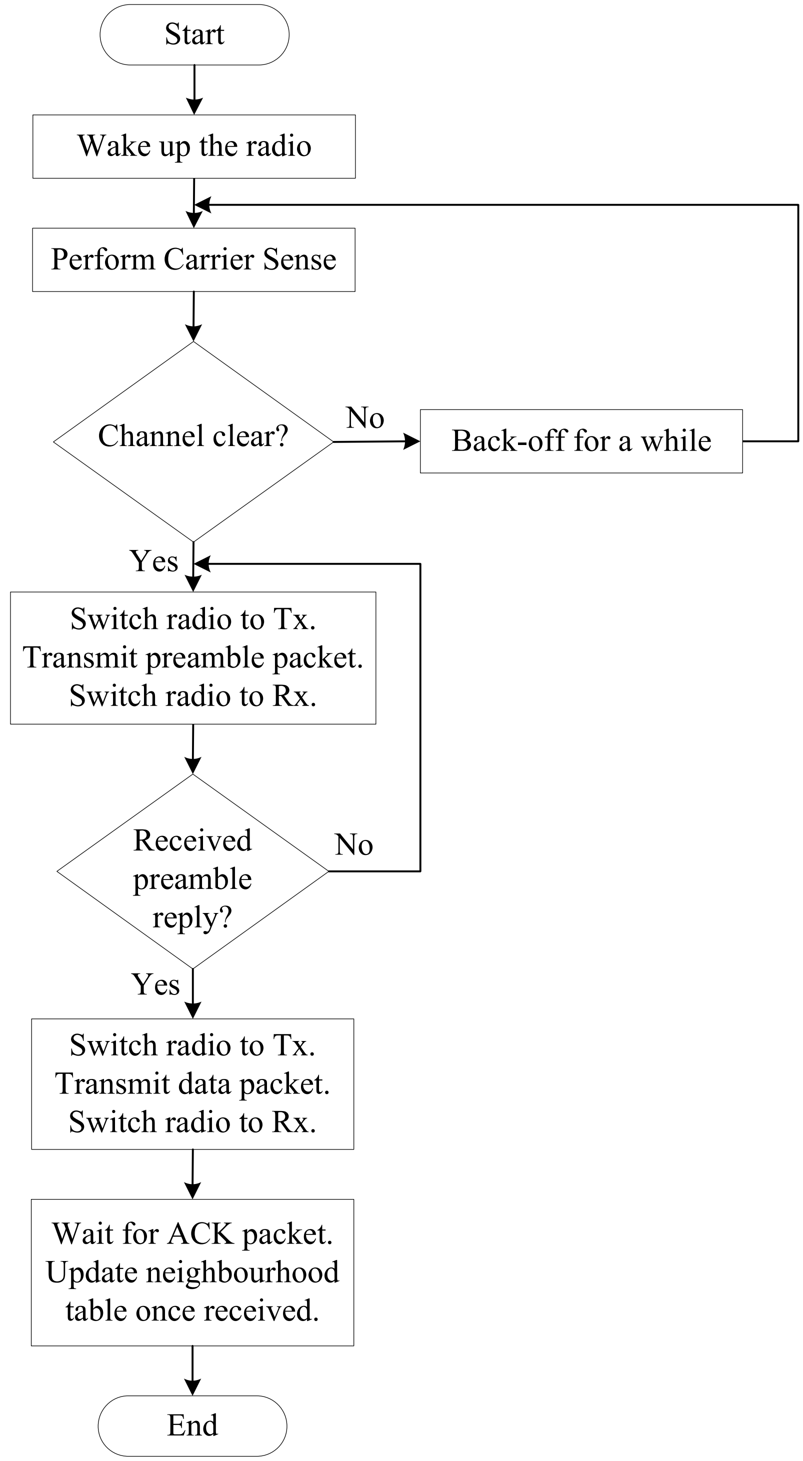

Figure 5.12 MAC protocol operation in the unsynchronized mode.

In the neighbourhood discovery phase, no actual sensed data is sent and node A simply populates its neighbour table with the address and wakeup time of the successfully contacted node B. The neighbourhood table contains entries as shown in Table 5.5 below. This information is gathered in the neighbourhood discovery phase or updated every time an ACK packet is received. Node B includes its wakeup time in the ACK packet it sends after a successful data transfer. This ensures relatively accurate estimations of destination wakeup times by source nodes in the synchronized mode. The nodes learn or refresh their neighbour's sampling schedule during every data exchange by piggybacking onto the acknowledgment packets the remaining time until the next listen slot. Every node keeps an updated table of sampling time offsets for all its known destinations. Since a node has only a few direct destinations, it can manage such a table even with limited memory resources.

Neighbour Info	Description
Neighbour address	The address of the direct neighbour node.
Wakeup Time	The time of the next wakeup of the direct neighbour node.

Table 5.5. WSN MACL Neighbourhood table entries.

The length of periodic listen slots T_{LS} and the carrier sense T_{CS} are kept as small as possible to promote minimum energy consumption. T_{LS} and T_{CS} are calculated as

$$\mathrm{T_{LS}} = T_{CS} = 2 \times T_{\mathrm{Pr}\,e-pkt} + T_{RS-tx} + T_{RS-rx}\,. \tag{5.3}$$

The length ensures that a preamble packet is guaranteed to be received by a destination node.

The nodes operate in the unsynchronized mode during the neighbour discovery phase or if the wakeup schedule of a destination node is unknown. If the wakeup schedule is known, the MAC protocol operates in the synchronized mode, which is explained next.

5.6.2.2 MAC Synchronized Mode

The MACL protocol operates in the synchronized mode if the wakeup schedule of the destination node is known and contained in the MACL neighbourhood table. The wakeup times are used to estimate the next wakeup time of a destination node. With the time estimation of the next listen slot, the optimal length of the preamble sequence T_P can be determined. T_P is minimized to only cater for clock inaccuracy. If the preamble sequence length is kept as short as possible, the number of preamble packets transmitted is greatly reduced, resulting in large energy savings.

The synchronized mode is illustrated in Figure 5.13. Once an event occurs at node A, i.e. the MACL received a packet from the NETL to transmit, the MAC protocol calculates the optimal preamble starting point. The waiting period between the event occurrence and the calculated starting point is denoted by T_{wait}. If during this time, node A has a scheduled listen slot, this listen slot is performed only if there is still enough time to receive a complete packet before the calculated starting point is reached.

The preamble starting point is calculated as 2ΘL before the estimate destination wakeup time to cater for clock inaccuracy of sender and receiver oscillators. As shown in Figure 5.13, if both quartzes have an accuracy of ±Θ, the node has to start sending the first preamble packet in advance to the estimated schedule $t_{estimate}$ for the time of at least 2ΘL. L is the elapsed time from last communication i.e. last received re-synchronisation from the destination node (ACK packet). Furthermore, a random time T_{random} is added to avoid collisions with other nodes that are similarly well synchronized to the destination node B and want to send in the same listen slot. The magnitude of the random time T_{random} is taken as a multiple of $T_{RS\text{-}tx}$ the radio switch time to Tx mode, with $h \times T_{RS\text{-}tx}$, where h is a uniform random integer from interval [0,N]. N represents the number of direct neighbours of a node and is network density specific. In this way, the preamble sequence length T_P is minimized resulting in maximum energy conservation.

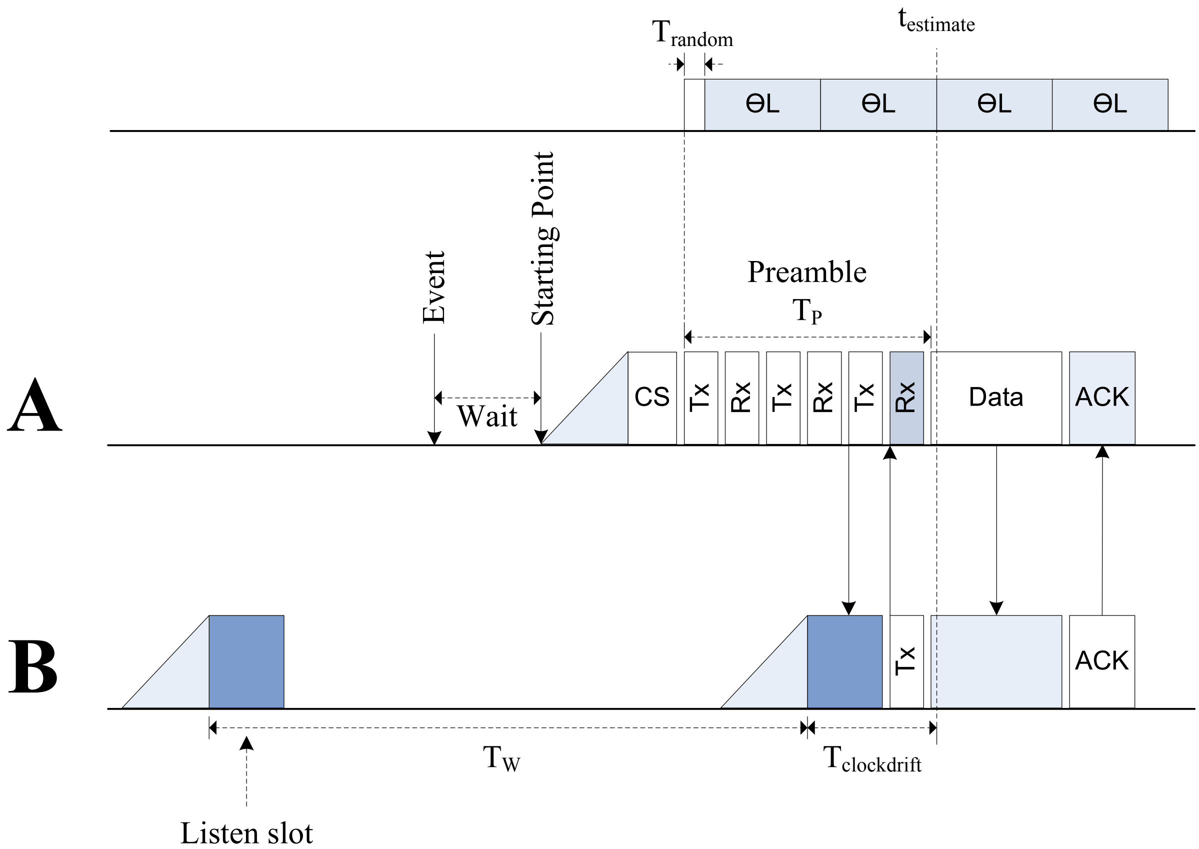

Figure 5.13 CSMA-MPS for WSANs in the synchronized mode.

In addition to the clock inaccuracy calculation and the random time T_{random}, the radio wake up time T_{RWU} and carrier sense time T_{CS} is added to arrive at the final value of the preamble starting point as shown in Figure 5.13. Once the destination node B has received a preamble packet, the same process is followed as in the unsynchronized mode above.

By calculating an optimal starting point, the preamble length is minimized and is considerably shorter than the preamble length of the unsynchronized mode, which equals the wakeup interval T_W. A shorter preamble length implies higher energy efficiency.

5.6.2.3 MACL Queue

The MACL uses a queue to store packets received from the NETL to be transmitted by the node. The only packet queue resides in the MACL since it has to perform contentions and calculate estimate neighbour node's wakeup times. Whenever the MACL protocol is in the process of sending another packet, packets that are received from the NETL are added to the back of the MACL queue. A sending process starts from the instance that it received a packet from the NETL and has calculated the specific preamble starting point and ends once the ACK is received from the destination node. The next packet in the queue is only handled once the MACL is finished with a sending process. Furthermore, when the MACL was unable to transmit a packet during the estimated wakeup time of the destination node, the packet is inserted into the front of the queue.

The maximum queue size is chosen to be relatively small since the resource constraint sensor nodes do not have large memory space. The maximum queue size may be around 10 and stays constant. If the queue is full and another packet is received from the NETL, the oldest packet inside the queue is discarded and the new one inserted at the back. All packets are treated with the same priority.

5.6.2.4 Aggregation

Aggregation is best considered within the MACL of the CPS. As explained above, the MACL involves a queue used for the storage of packets received from the NETL while it waits for the PHYL to finish transmitting the current packet. Since sensor nodes are involved with sending and relaying of multiple packets, aggregation may be highly beneficial. By reducing the amounts of packets in the network, the network traffic load is alleviated. Nodes conserve energy by spending less time transmitting, resulting in higher network lifetime. The proposed data aggregation within the MACL involves the transmission of multiple packets destined for the same neighbour node in one communication session. This means, once the destination node is successfully contacted via the preamble sequence, all packets in the MACL queue are forwarded to it. This avoids the need for a preamble sequence for every packet in the queue that is destined for the same next-hop node, saving scarce energy. However, the number of packets that may be sent in one session is limited to three, to ensure that the channel is not kept occupied for longer period by a communication session between two nodes.

The use of an aggregation function on the actual data is not considered firstly due to the fact that the MACL cannot tap into the APPL frame and secondly the accuracy of sensed data has to be preserved. An actor node has a much more accurate view of the current physical environment if it receives information of every node rather than an average value for a bunch of sensor node. This is important for when preventative measures may be taken by an actor node to prevent a critical event from happening in a specific region. If an actor node only receives an average value for several sensor nodes, it will not be able to determine the exact location of the required action.

5.6.2.5 Channel Occupied and Hidden Terminal Problem

In the event that the node detects that the channel is busy or *occupied* in the CS period, the node backs-off for a while until the channel is idle again. Another CS period is performed only if it is still within the preamble time window in the synchronized mode. If it has passed the time window, a new starting point is calculated to reach the destination node in its next listen slot. In the unsynchronized mode, the node backs-off for a random time and then performs another CS period.

The *hidden terminal problem* is alleviated in the following manor. If the node A wants to transmit, it senses the carrier for a minimum amount of time. The radio is set with a sensing threshold that lies far below the actual receiving sensitivity threshold, for example -100 dBm sensing threshold if receiver has a sensitivity of -87 dBm. In this way the sensing range is much larger than the communication range and collision probability due to problems such as the hidden terminal problem are reduced. However, in densely populated sensor networks the increased sensing range introduces additional back-offs. A counter measure is to use high bit rate transceivers to shorten the time a node is actively transmitting.

5.6.2.6 No Preamble Reply and Collisions

The reason for not receiving a preamble reply packet from a destination node B may be that it is not functional any more, the listen slot was totally missed or that repeated collisions of the transmitted preamble packets occurred and node B was not able to read them.

If the source node A has completed a preamble sequence and no preamble reply packet was received, the data packet is inserted into the front of the MAC queue and the node put to sleep. Once the node is put to sleep, the MAC queue is checked for data packets. The newly inserted packet is then retrieved and a new preamble starting point is calculated to reach node B in its next listen slot. If the source node has tried a total of three times, to reach the intended destination node, the source node utilizes the unsynchronized mode to try and reach the intended destination. If this is unsuccessful, it is evident that the intended destination node is not functional any more. The packet is then discarded and the destination node removed from the neighbourhood table.

Collisions may also occur when sending the data block. Therefore, upon receiving the data block successfully, node B sends an acknowledgement packet back to node A to indicate it has successfully received the data packet. If no ACK packet is received by source node A, it retransmits the data packet if the channel is clear.

5.6.2.7 Synchronization

With the CSMA-MPS protocol a periodic synchronization is not required by the nodes. Nodes synchronize when they first exchange packets and in every data exchange phase by including the clock offset and next wakeup time in the acknowledgment packet. This considerably reduces the overhead at the transmitter due to the much shorter preamble length and the overhead at the receiver due to the reduced listening time. At most each node keeps synchronization information in the neighbourhood table from all its direct neighbours. Network wide synchronization within the MACL is not required.

5.6.2.8 Energy Efficiency

The main energy efficiency MACL issues of collisions, overhearing, overhead and idle listening are well handled by the CSMA-MPS protocol. The MACL protocol maintains high energy efficiency by introducing a periodic listening cycle with very short listen slots. The time spent in idle mode is hence reduced and kept at a minimum. Furthermore, the time spent transmitting contention preamble packets is kept at a minimum by making use of an alternating preamble sampling sequence that is started as late as possible by estimating the destination wakeup time based on known schedules. It is then stopped as soon as the destination received the first preamble packet.

Collisions are reduced by including a carrier sense period to test if the channel is occupied. The channel sensing is enhanced by setting the carrier sensing threshold of the radio to a value higher than the actual sensitivity. This results in a larger range of sensing and reduces the chances of the hidden terminal problem and collisions.

Overhearing is also reduced since nodes wake up for very short periods of time and neighbour nodes have a unique listen slot. In this way, packets are sent at different times to different nodes. Finally, overhead is kept at a minimum by utilising existing data exchanges for MACL specific information exchanges and making control packets as short as possible.

5.6.2.9 Real-Time Communication

The real-time requirement posed by WSANs is not exclusively handled by the CSMA-MPS protocol, which has energy efficiency as its primary priority. The contention-based protocol's overall functionality provides minimized latency to a certain extent when compared to TDMA protocols, but does not have special mechanisms targeted specifically at latency minimization.

5.6.2.10 Scalability and Mobility

The selected protocol scales very well to larger networks and also handles topology changes very well. A newly added node or a node that enters a new region due to mobility immediately adapts and self-configures itself to its new neighbour nodes and start its wakeup cycle at a time that does not to clash with any of its neighbour nodes. The unsynchronized mode is initially used to establish who the neighbour nodes are and what their wakeup times are (neighbourhood discovery).

5.6.2.11 MACL packet formats

The MACL involves the use of three different packet formats. The first packet format is the preamble packet illustrated in Figure 5.14. This format is used for preamble and preamble reply packets in the synchronized mode. The preamble format for the unsynchronized mode does not include the destination address. The packet contains specific fields as follows.

Type	Source Address	Destination Address
4 bits	16 bits	16 bits

Figure 5.14 The format of a MACL preamble packet.

- The *Type* field holds the type of preamble packet, either a MAC_PREAMBLE_MESSAGE or a MAC_PREAMBLE_REPLY_MESSAGE packet.
- The *Source Address* and *Destination Address* fields contain the addresses of the source node and the address of the intended destination node respectively. In the unsynchronized mode, the destination address field is omitted since the address is not yet known.

The second packet format that is used by the MACL is for the data packet. The packet format contains specific fields as show in Figure 5.15, which is explained as follows.

Type	Source Address	Destination Address	NETL packet	More Data Flag
4 bits	16 bits	16 bits	Variable	4 bits

Figure 5.15 The format of a MACL data packet.

- The *Type* field holds the packet type, which would simply be SENSOR_MAC_DATA_MESSAGE.
- The *Source Address* and *Destination Address* fields contain the addresses of the source node and the address of the intended destination node respectively.
- The *NETL packet* field contains the encapsulated NETL packet and may vary in size depending on whether the APPL has employed data aggregation or not.
- The *More Data Flag* field is an optional feature that enables nodes to send all packets contained in its queue, destined for a specific node, to be sent one after the other in a single communication session. A communication session is established once a destination node has replied to a preamble packet. The flag is an indication to the destination, if more data packets are still to come or not.

The third packet format that is used by the MACL protocol is the acknowledgement packet, which is sent after the data packet is been received successfully. The format is shown in Figure 5.16 and explained as follows.

Type	Source Address	Destination Address	Next Wakeup Time
4 bits	16 bits	16 bits	32 bits

Figure 5.16 The format of a MACL ACK packet.

- The *Type* field holds the packet type, which would simply be MAC_ACK_MESSAGE.
- The *Source Address* and *Destination Address* fields contain the addresses of the source node and the address of the intended destination node respectively.
- The *Next Wakeup Time* field contains the time of the next wakeup of the node. This time is noted by the receiving node and used for synchronization purposes and to calculate the listen slot time of a destination node.

5.6.2.12 Summary of Protocol Features

The main protocol features of the MACL protocol are summarized as follows.

- Contention-based protocol employing an alternating tx and rx preamble sequence.
- Periodic listening cycle with very short listen slots. Nodes periodically wakeup during their listen slot, listen for incoming transmissions and go back to sleep after transmission complete or no incoming transmissions detected.
- Makes use of neighbour wakeup schedules to optimize its operation and energy efficiency.
- Operates in unsynchronised mode for unknown destination and synchronised mode for known neighbour nodes.
- May employ aggregation that involves the transmission of multiple packets in the queue destined for the same node in one communication session.

- Optimized for energy efficiency.
- Employs feature that detect channel occupancy and mitigate the hidden terminal problem.

5.6.3 Notes on the MACL protocol for actor nodes

The same MACL protocol is proposed for use by the actor node for the sensor-actor communication. The actor nodes however, have a much richer energy resource than the sensor nodes and it is hence proposed that the wakeup interval T_W is set very low so that the actor nodes appear awake all the time.

5.6.4 Identified MACL Parameters for Cross-Layer Interactions

The MACL is primarily involved in accessing the wireless medium by utilizing a contention-based approach that entails a lot of intricate timing. Providing MACL specific information to other layers in the CPS can have a profound impact on the network efficiency with regards to latency and energy. Specific CSMA-MPS MACL parameters and features that may be useful and beneficial to other layers in the CPS towards promoting overall latency minimization and energy efficiency improvements are identified as follows.

a. The *neighbourhood table* of the MACL contains information of neighbouring nodes that are within transmission range. As proposed by the SCLF, the table is moved to the neighbourhood table structure inside the SCLF to avoid having duplicates of the same information. The MACL neighbourhood table entries are added to the NETL's entries. The additional MACL parameters contained in the table include the neighbours' **wakeup time**, the **wakeup interval,** the **channel condition,** the **time of last successful communication timestamp** and a **reachable flag**. This information may be viewed by all layers in the CPS. The wakeup interval is important for estimating the next wakeup time of a destination node. The *channel condition* is a measure of the number of failed CS performed by the MACL during a specific time period. This value is periodically measured by a node and communicated within the ACK or added to the NETL update packets. A high value indicates a very busy channel and a low value, a less busy channel characteristic of an area with a low node density for example. Compared to the NETL traffic load, the channel condition metric gives a very useful indication of the surrounding node density and the 'busyness' of the channel rather than the 'busyness' of a node. The time of the last successful communication timestamp table entry, holds a timestamp of the last successful communication with a specific neighbour node. Finally, the reachable flag indicates if the destination node was actually successfully reached during the last communication attempts. This information may be viewed by any layer in the CPS.

b. The *Channel condition* of a node is entered into the information pool structure of the SCLF and is a measure of how many CS operations have failed during a specific time period. This gives a useful indication to how busy a channel was during a recent time period.

c. The *Queue information* of the MACL includes the maximum and current queue size. This information is entered into the information pool structure of the SCLF and may be used by higher layers to stall packets to prevent packets from being discarded by the MACL as a result of a queue overflow.

The actual incorporation and usage of these parameters by other layers in the CPS is discussed in section 5.8.

5.7 LAYERED WSAN PHYSICAL LAYER DESIGN

The PHYL is the lowest layer in the CPS as discussed in section 3.5. The most important issue pertaining to the PHYL, that is relevant to this dissertation, is the selection of a suitable radio. Since the radio is the primary energy consumer of a node, it has to be controlled very efficiently to ensure the feasibility of WSANs. This control is the responsibility of the MACL protocol, which instructs the radio to sleep or wakeup at certain times in order to save as much energy as possible and promote latency minimization. To further facilitate the MACL protocol in its operation, certain radio hardware specifications are favoured for use in WSANs. The considerations regarding the radio hardware specifications as revealed in section 3.5.1 are shortly revisited and preferred choices highlighted. Finally, the cross-layer prospects for the PHYL are discussed.

5.7.1 Radio Parameters

The radio parameters as introduced in section 3.5.1 are the transmission power, data rate, turn-on time, tx/rx switch time, sensitivity, carrier sense threshold and carrier frequency.

The *transmission power* is set equal for all sensor nodes inside the WSAN upon network initialization. Although some radios enable different power level settings, the level is kept constant. This is done firstly, to assist the sensor node distribution of specific applications and secondly to simplify the task of determining a suitable carrier sense threshold level. For applications that require specific placements of nodes in order to cover the area of interest, the radio transmission power is used to calculate the maximum communication range of sensor nodes. Therefore, in keeping the power level constant throughout the network, a node placement layout can be easily designed. The second reason for keeping the level constant is to ensure that when setting the carrier threshold level lower than the radio sensitivity, the hidden terminal problem may be avoided or at least alleviated. If the transmission power level is set differently for nodes inside the WSAN, the worst case or highest possible transmission power level of a node must be used to calculate the appropriate carrier sense threshold level. Setting the threshold too high may introduce additional back-offs in the MACL which in turn may increase the latency experienced by a packet.

The *data rate* of communication between two nodes determines the time it takes to send a specific size of data. For latency minimization purposes, the data rate should be as high as possible and is preferred to be at least 500 kbps or 1Mbps. Although sending at the higher data rates uses more energy, the latency reduction gains outweigh the energy consumption. Furthermore, a higher bit rate shortens the time a node is actively transmitting which reducing the probability of collisions within the WSAN.

Two radio parameters that are important for MACL timing issues are the radio *turn-on time* and the radio *tx/rx switch time*. The turn-on time is the time it takes for radio to be operational after being signalled to wakeup. The quicker the radio can switch on, the sooner data transmissions can occur. This parameter only has notable effect on MAC protocols that employ extremely short duty cycles. The radio switch time is the time it takes for the radio to switch between transmit and receive mode. A short switching time enables a short carrier sense period for the preamble sampling protocol and lower latency.

The *sensitivity* and *carrier sense threshold* defines up to which incoming signal power level data can be successfully received and the power level by which the channel is deemed busy, respectively. The radio sensitivity increases with a lower data rate which means that high data rate usage requires a denser node deployment. As for the carrier sense threshold, the PHYL is supplied with this threshold parameter before node deployment to facilitate the MACL protocol's employment of a carrier sense to determine if the channel is free, as mentioned in section 5.6.2. This alleviates the hidden terminal problem and also reduces the collision probability. The carrier sense threshold is set by up to 10dB lower than the radio sensitivity.

Another parameter that influences the communication range of a node is the *carrier frequency* used by the RF communication. Transmitting at higher frequency bands requires more power for the same distance but allows higher bandwidth. The higher bandwidth permits shorter transmitting time and hence lower latency and is thus preferred.

The PHYL is hence also faced by the latency vs. energy trade-off. However, another factor that has to be considered when choosing a radio is the cost. Although the cost of hardware is decreasing, expensive radios with promising capabilities may not be feasible for WSANs. The further a radio is able to transmit, the higher its cost.

5.7.2 PHYL packet format

The PHYL involves the use of a general packet format. The packet format is illustrated in Figure 5.17. The packet contains specific fields as follows.

Preamble (101010101…)	Sync Word	MACL packet	CRC
32 bits	16 bits	Variable	16 bits

Figure 5.17 The format of a MACL Data packet.

- The *preamble* field contains a sequence of 0's and 1's to bit synchronize the transmitting and receiving radios.
- The *Sync Word* field is used to indicate the start of information contained in the frame by means of a specific bit pattern.
- The *MACL packet* field contains the encapsulated MACL packet and may vary in size depending on whether the data aggregation has been employed or not.
- The Cyclic Redundancy Check *(CRC)* field holds the error detection information.

5.7.3 PHYL Cross-Layer Prospect

The cross-layer interaction prospect for the PHYL is very limited. As proposed by some WSN cross-layer implementations, discussed in section 4.3, power control at the PHYL is employed to lower energy consumption. A specific power level is derived to set the communication range to just reach a particular destination node. The minimum range is set to ensure minimum energy consumption when transmitting. This popular approach however, has a fundamental flaw. The hidden terminal problem is reintroduced and exacerbated since the carrier sense operation conducted by nodes who want to reach the same destination node may not detect each others communication attempts. This will result in seriously high collision probabilities and an unacceptable number of retransmissions. The carrier sense operation is hence useless when power control is employed. Furthermore, in WSANs with mobility and topology changes, the use of power control may introduce even more challenges.

The radio used for sensor nodes in the WSAN is selected with the preferred parameters before network initialization or node deployment and hence these parameters stay constant. For the purpose of this dissertation, the PHYL plays a subdued role in the cross-layer paradigm.

5.8 PROPOSED CROSS-LAYER INTERACTIONS AND IMPLEMENTATION DETAILS

While many energy-aware MAC and routing protocols have been proposed for sensor networks, very little research has been done to combine real-time requirements and energy-awareness [29]. The proposed SCLF provides means for communication and information sharing to occur between all layers of the WSAN CPS and assists each layer in addressing this important issue. Parameters that are beneficial to other layers regarding the optimization of the latency and energy efficiency have been identified for every layer. An overview of the identified cross-layer parameters and features is given in Table 5.6. Furthermore, an overview of the cross-layer neighbourhood table structure entries is given in Table 5.7. In the next subsections, it is determined what specific information is useful to which layer and how the proposed layered protocols may incorporate the information from other layers into their own layer's operation to promote the optimization of latency and energy efficiency.

	Parameter/Feature	**Description**
Application Layer	APPL state information	The state information indicates the current state of the APPL protocol.
	T_{sense}	The period a node waits between requesting a value from the sensing unit, i.e. takes a sensor measurement.
	Packet type	This parameter is contained in the APPL packet header and can be read by the next lower layer (NETL). The packet type specifies packets containing a sensed value below the critical threshold as *non-critical* and above the threshold as *critical*.
Network Layer	Status information	The status indicates if the node can be regarded as connected or not connected to the rest of the network.
	Neighbourhood table	The information of neighbour nodes within transmission range. Includes the address, hop distance to its actor node, traffic load, residual energy E_R and a timestamp.
	Traffic load	Calculated as the amount of packets received and sent by the NETL during a specific period, multiplied with the average packet size.
	Hop distance	The hop distance from a node to its assigned actor node.
	Packet type	This parameter is contained in the NETL packet header and can be read by the next lower layer (MACL). The packet type specifies if the packet is a NETL metric update packet or a particular APPL data packet.
MAC Layer	Neighbourhood table	The information of neighbour nodes within transmission range. Includes the neighbours' wakeup time, the wakeup interval, the channel condition, the time of last successful communication and a reachable flag
	Channel condition	The amount of CS failures recorded over a specific time period.
	Queue information	The MAC maximum and current queue size.

Table 5.6. WSAN CPS cross-layer parameter overview.

	Neighbour Info	Description
Network layer info	Neighbour address	The address of the node.
	Hop distance	The hop distance from the node to its actor node.
	Residual energy	The residual battery energy level.
	Traffic load	The latest traffic load value.
MAC layer info	Wakeup Time	The time to the next wakeup.
	Wakeup Interval	The period between wakeups.
	Channel condition	The latest recorded CS failure rate.
	Last successful communication timestamp	The time of the last successful communication.
	Reachable flag	An indication that the last communication attempt was successfully completed.
	Timestamp	The time of last updated entry.

Table 5.7. WSAN SCLF neighbourhood table structure entries.

In this section, the modifications and enhancements to the proposed layered WSAN protocol designs for additional and enhanced functionality aimed at promoting latency and energy efficiency performance gains are proposed. Furthermore, the benefits gained by the incorporation of specific parameters into relevant layers of the CPS are also discussed and finally, a summary of all cross-layer related enhancements and features is given.

5.8.1 The Cross-Layer enabled APPL Protocol Design

The fundamental methodology of the proposed cross-layer CPS involves the immediate reaction by the lower layers to the APPL state and type of data packet. It is stressed that the CPS needs to be highly application aware to ensure overall implementation feasibility to WSANs. As detailed in section 5.4.3, the APPL state information is inserted into the information pool of the cross-layer framework for other layers to read. Furthermore, loose encapsulation of the APPL packets is proposed to enable the next lower layer, the NETL, to be able to specifically read the packet type contained within the APPL packet format. This is important since the NETL can then set its packet type to match the APPL packet type. With loose encapsulation, the MACL can in turn read the NETL packet type. Effectively, the MACL has knowledge of the APPL packet type by having viewed the NETL packet type.

In the existing literature it is revealed that there exist a trade-off between the two main WSAN requirements, namely latency minimization and energy efficiency. Minimizing the latency of a data packet comes at the cost of lower energy efficiency. On the other hand, when providing high energy efficiency the latency increases. This trade-off exists in every layer of the CPS and hence it is proposed that the latency vs. energy trade-off needs to be considered from a CPS-wide perspective, which is made possible by the introduction of cross-layer interactions. The trade-off issue is approached in the following way.

Unlike WSANs, in WSNs there is no need to distinguish between different packets. WSNs are primarily concerned with simple information gathering at the sink. Therefore, with the exception of CoLaNet (limited to very small WSNs), WSN cross-layer literature does not include the APPL. However, in WSANs, considering APPL information provides a suitable solution in handling the latency vs. energy trade-off issue. According to the defined basic operational scenario, observed data is sent from the sensor nodes to the actor nodes. This information is categorised as either critical or non-critical data as proposed by the APPL protocol in section 5.4. Critical data packets report the occurrence of a critical event and are highly delay-sensitive. This means they have to reach the actor node with minimum latency to ensure rapid response in resolving the detected event or disturbance. On the other hand, the non-critical data, which is periodically sent to the actor nodes for statistical, analytical and logging reasons, are less delay-sensitive and hence do not require the highest priority with regards to latency minimization. In summary, the sensed data is categorised as either highly delay-sensitive or lowly delay-sensitive.

This categorisation lends the way for a *cross-layer solution* in handling the latency vs. energy trade-off issue. Whenever critical, highly delay-sensitive information exists, the protocol layers regard latency minimization as the top requirement. Whenever non-critical, delay-insensitive information exists, the protocol layers regard energy efficiency as the top requirement. The requirement inferior to the top requirement is however still of utmost importance and is still extensively considered within the protocol layers. The focus is simply shifted slightly towards one of the two requirements. In order to realize the requirement prioritization, cross-layer interactions are vital in communicating the critical or non-critical APPL state and packet type. In this way, other layers of the CPS are informed to either regard latency minimization or energy efficiency as their top priority.

In the next subsections, a new *Cross-Layer* enabled Network-Centric Actuation Control with Data Prioritization (CL-NCAC-DP) protocol is proposed. The APPL adoption of the cross-layer paradigm is proposed, which involves specific modifications and enhancements to the layered NCAC-DP APPL protocol of section 5.4.2. The additional functionality and benefits gained by the incorporation of specifically identified lower layer cross-layer parameters into the layered NCAC-DP APPL protocol are also discussed.

5.8.1.1 Information used from the NETL

In this section, parameters that are considered by the new CL-NCAC-DP protocol, originating from the NETL, are given. Furthermore, the modifications and enhancements towards efficiency optimization by incorporating these NETL parameters into the layered APPL NCAC-DP protocol (of section 5.4.2) are discussed.

The identified NETL parameter that is useful to the CL-NCAC-DP APPL protocol includes the following.

1. Network Layer Status

5.8.1.1.1 Network Layer Status

At WSAN initialization, or once a new node is inserted into the network, the APPL protocol needs to consider the NETL status parameter, contained in the framework's information pool, before initiating the first sensor reading and passing a sensed data packet to the NETL. This ensures that the APPL does not waste energy by sensing and instigating the sending of data packets while the node is not yet part of the WSAN network or the NETL is not yet finished with its initial setup. Furthermore, it prevents CPS packet queues from overflowing and packets being discarded. Therefore, as illustrated in Figure 5.18, the CL-NCAC-DP protocol remains in the INITSETUP state, and does nothing but sets a timer to periodically poll the NETL status with a period $T_{NETLstatus}$. Only once the NETL status is set to CONNECTED by the NETL, CL-NCAC-DP switches to the NORMAL operation state.

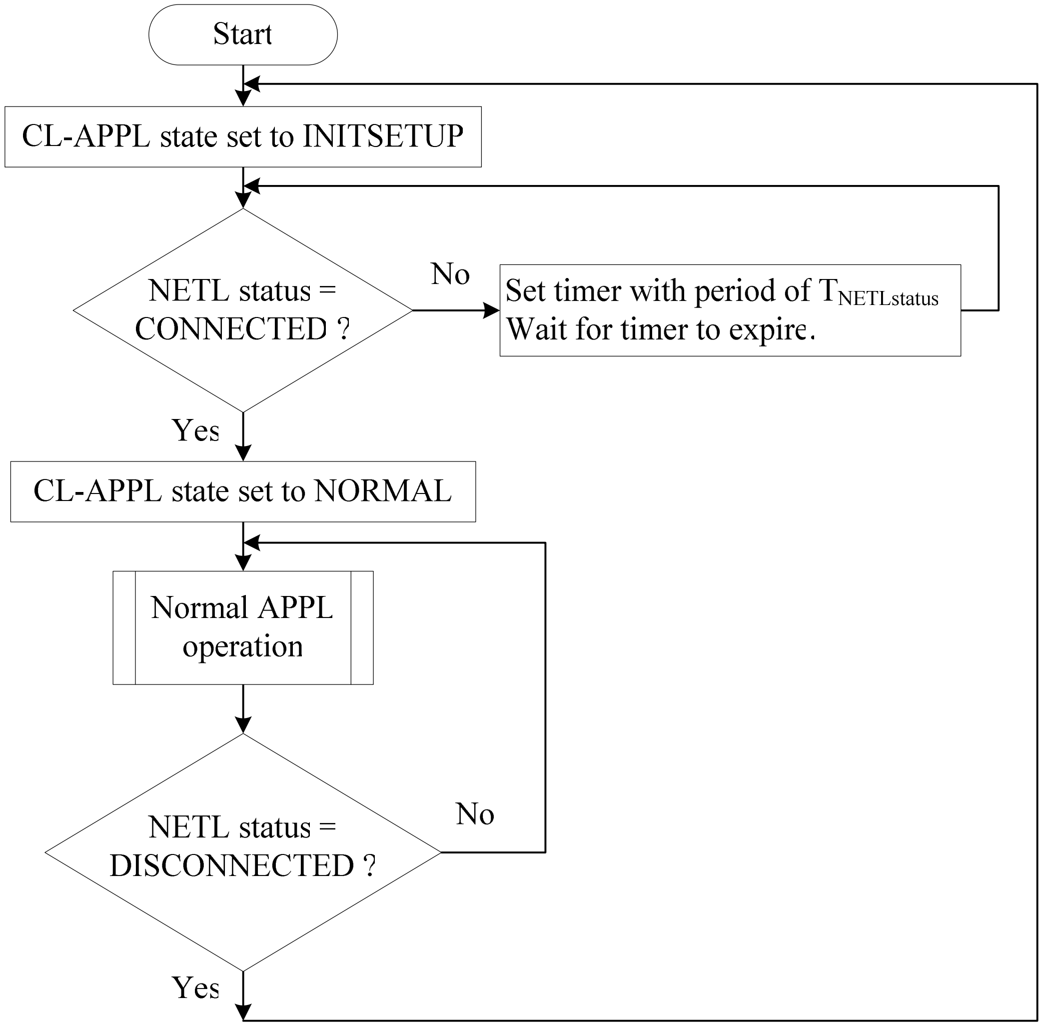

Figure 5.18 Flowchart of CL-NCAC-DP using the NETL status.

In normal operation, the NETL status parameter is still periodically polled, since it may change to DISCONNECTED in the event that the NETL is repeatedly unable to find and connect to a downstream neighbour node, to which it can forward a data packet. The NETL status may change to DISCONNECTED and remain in that state until it re-establishes a connection with a node with equal or lower hop distance to the actor node. During the DISCONNECTED state, CL-NCAC-DP halts its sensing operation and returns to the INITSETUP state. Once the NETL status changes to CONNECTED again, CL-NCAC-DP enters the NORMAL state and continues with normal operation.

5.8.1.2 Information used from the MAC Layer

Identified MACL parameters that are used by CL-NCAC-DP and proposed enhancements to the layered APPL are discussed in this subsection. The identified parameter includes the following.

1. Maximum and current queue size

5.8.1.2.1 Maximum and Current Queue Size

The CL-NCAC-DP protocol uses the MACL queue parameters, contained in the framework's information pool, to determine if sensor reading obtainment from the sensing unit should be delayed. The maximum and current queue size is used to calculate the amount of packets the MACL queue can still accommodate. If the MACL queue is more than 80% full, the APPL obtains a scheduled sensor reading and if it is not critical, discards it. In this way, the MACL is given time to process its packets and empty its queue. This prevents the MACL queue from overflowing and subsequent discarding of data packets.

5.8.1.3 Summary of Additional Parameters for the CL-NCAC-DP Protocol

The proposed CL-NCAC-DP protocol features an additional parameter, given in Table 5.8.

Parameter	Description	Value
$T_{NETLstatus}$	Period of polling the NETL status	Between 5 and 15 seconds

Table 5.8. Summary of the additional parameter for the WSAN CL-NCAC-DP.

5.8.2 The Cross-Layer enabled NETL Protocol Design

The proposed layered NETL protocol, CHEAGR (in section 5.5.2), only bases its routing decisions on metrics pertaining to its own layer. The WSAN requirements have been considered as far as possible and the latency vs. energy trade-off challenge is handled by finding a suitable static balance. However, the optimization of the energy efficiency and latency is restricted to the information contained within the NETL only. This presents a sub-optimal solution, which may be improved by considering and incorporating parameters from other layers.

The cross-layer approach that may be adopted by the NETL provides an improved solution to the latency vs. energy trade-off. As mentioned in the discussion of the CL-NCAC-DP (in section 5.8.1), the fundamental methodology of the proposed cross-layer CPS involves the immediate reaction by protocol layers to specific APPL states and packet types. It is stressed again that the CPS needs to be highly application aware to ensure overall implementation feasibility to WSANs. Therefore, it is proposed that whenever critical, highly delay-sensitive information exists or is received, the protocol layers regard latency minimization as the top requirement. Whenever non-critical, delay-insensitive information exists or is received, the protocol layers regard energy efficiency as the top requirement. It is important to note that the requirement that is not regarded as the primary one is not totally disregarded but simply considered secondarily.

The sensed data is evaluated in the APPL by comparing it to a critical threshold level. An application may require several threshold levels, but for simplistic reasons only two packets types are used. This approach is much simpler and favoured above trying to establish and calculate complex per-packet latency bounds. The per-packet or per-event latency bound approach is used by the RRDSE and POWER-SPEED protocol (in section 3.3.3.3). These NETL protocols simply state that an event possesses a certain latency bound and is routed accordingly, to ensure that the packet reaches the actor node within the latency bound. However, it is never stated how the latency bound is obtained or calculated. It is not a trivial task since the latency bound needs to consider in addition to the type of event, the hop distance from the actor node and the distance between the nodes on a route towards the actor node. This requires excessive overhead and is simply not feasible for energy constrained nodes inside WSANs. Therefore, a more feasible approach is proposed that employs reactive routing by distinguishing between critical and non-critical data rather than trying to calculate a specific per-packet or per-event latency bound. The next-hop node is chosen to satisfy the primary objective of either energy efficiency or latency minimization by comparing between easily obtainable direct neighbour’s timing and residual energy details. This requires much less overhead and is much less complex.

In the next subsections, a new Cross-Layer enabled Cluster-based Hierarchical Energy/Latency-Aware Geographic Routing (CL-CHELAGR) protocol is proposed. The NETL adoption of the cross-layer paradigm is proposed, which involves specific modifications and enhancements to the layered CHEAGR NETL protocol (of section 5.5.2). The additional functionality and benefits gained by the incorporation of specifically identified cross-layer parameters into the layered CHEAGR NETL protocol are also discussed.

5.8.2.1 Information used from the Application Layer

In this section, APPL parameters that are considered by the new CL-CHELAGR protocol are given. Furthermore, the modifications and enhancements for the promotion of latency and energy efficiency optimization by incorporating these identified APPL parameters into the layered CHEAGR NETL (of section 5.5.2) protocol are discussed.

The identified APPL parameters that are useful to the CL-CHELAGR NETL protocol include the following.

1. APPL State
2. Data packet type
3. Sensing period T_{sense}

5.8.2.1.1 APPL State

The APPL state is used to determine if NETL metric update packets are allowed to be sent. If the APPL is not in the NORMAL state, the NETL metric update packets are not created, to prevent a situation of high congestion at a node which is currently used for critical event reporting.

5.8.2.1.2 Sensed data packet type

The data packets received from the APPL have either been classified as critical or non-critical according to a critical threshold level. In the cross-layer design, the CL-CHELAGR is able to view the APPL packet type and handle the packet accordingly. The critical data packets are considered highly delay-sensitive and hence CL-CHELAGR regards latency minimization as the top priority in its routing decisions. This is to ensure that critical packets arrive at the cluster's actor node as soon as possible. The non-critical data packets on the other hand, are considered less delay-sensitive and hence CL-CHELAGR elects energy efficiency as the top priority when routing these packets towards the cluster-head actor node. The routing of packets is hence performed as illustrated in Figure 5.19, which replaces the 'Select neighbour with highest ratio of residual energy/traffic load' process block in the original next-hop neighbour selection flowchart of the layered CHEAGR protocol (in Figure 5.7).

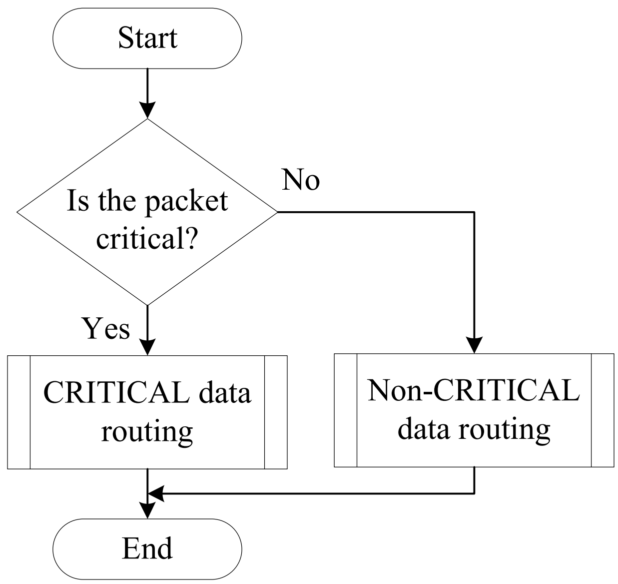

Figure 5.19 The cross-layer routing approach.

The data packets are hence handled and routing criteria applied according to the packet type. The critical data routing process considers the MACL timing properties to promote *latency minimisation*. A detailed explanation follows below in section 5.8.2.2.

The non-critical data routing process considers the residual energy and the traffic load as originally proposed by the CHEAGR protocol (of section 5.5.2.2), i.e. selecting that neighbour node that has the highest ratio of residual energy and traffic load, to promote *energy efficiency* for network lifetime maximization. In addition, CL-CHELAGR considers the MACL channel condition as well, which is also explained below in the section 5.8.2.2.

Once CL-CHELAGR has determined the next-hop node, it constructs the NETL packet by encapsulating the APPL packet and then setting the packet type as classified by the APPL. This is done so that the packet type as initially set by the APPL is traversed through to the MACL. With loose encapsulation, the cross-layer enabled MACL will in turn be able to view the NETL packet type. All packets that are newly generated within the NETL, specifically the metric update packets, are branded as non-critical by CL-CHELAGR.

5.8.2.1.3 *Sensing period T_{sense}*

The APPL T_{sense} parameter is proposed to be used for two different purposes. Firstly, the time period that a new sensor node spends switched on after insertion into an already setup WSAN, can be set by considering the T_{sense} APPL parameter. Since sensor nodes send periodically sensed data packets towards the actor nodes all with similar T_{sense} periods, the new sensor node is considered to have listened to and received sensed data from all its neighbours after a period of $3 \times T_{sense}$. By setting a distinct setup time period, sensor nodes are prevented from spending too much time fully switched on. This reduces the initial energy consumption of the node, although only minimal. After gathering the wakeup times of neighbouring nodes, the new node may contact each node and request further required information.

T_{sense} is secondly used to fine-tune the $T_{TrafficLoad}$ parameter. The time period $T_{TrafficLoad}$, during which the amount of traffic is recorded for determining the traffic load, is set to $3 \times T_{sense}$. This ensures accurate resemblance of the current traffic load experience by a node. If the $T_{TrafficLoad}$ parameter value is set too high, fluctuations in the traffic load can not be traced. If the CL-CHELAGR protocol does not respond to fluctuations, the effectiveness of reactively routing critical event data is compromised. For applications with a long packet generation period T_{sense}, $T_{TrafficLoad}$ will therefore also be longer, incurring fewer calculations while still ensuring ample traffic load monitoring.

5.8.2.2 Information used from the MAC Layer

Information contained within the MACL is identified as very useful and highly beneficial to the NETL in WSANs. The cross-layer information provides crucial assistance for optimization of the latency and energy efficiency. Since the main task of the NETL is to route packets towards the actor node, the criteria the NETL uses to determine a suitable next hop node has a large effect on the resulting latency and energy consumption.

The approach of some existing layered WSAN NETL protocols (RRDSE and POWER-SPEED) that measure delays at the NETL seems very inaccurate and rather erroneous. For correct delay estimations used as forwarding criteria, the NETL has to take into account timing characteristics of the MACL. Since MACL protocol designs that are suitable for WSANs make use of duty cycles, delays that will be experienced by a packet cannot be estimated without considering the MACL timing characteristics of candidate downstream nodes. Delays will be different for every packet that is handled by the NETL at a random point in time. Delays are strongly influenced by the duty cycle timing properties, MACL queue size and channel condition. These specific parameters are different for every node and are not taken into considerations by the existing delay estimating layered NETL protocols. A cross-layer approach is hence needed in order to enable the NETL to incorporate MACL information for accurate delay estimations towards promoting latency minimization.

Identified MACL parameters that are used by the new CL-CHELAGR protocol and proposed enhancements to the layered CHEAGR protocol are discussed in the next subsections. The identified parameters include the following.

1. Neighbourhood table
    - wakeup time
    - wakeup interval
    - channel condition
    - reachable flag
2. Channel condition
3. Queue information

5.8.2.2.1 Neighbourhood table

Information of neighbour nodes contained within the MACL is proposed to be used as additional criteria in the CL-CHELAGR routing decisions. The MACL and NETL information is stored in one central neighbourhood table as proposed by the SCLF. The MACL neighbour node parameters that are useful to the CL-CHELAGR protocol are: the *wakeup time* and *interval*, the current *channel condition* and the *reachable flag* parameter.

First of all, if CL-CHELAGR has received any type of packet from the APPL and has determined potential next-hop candidates based on the hop distance from the actor node, the MACL *flag*, indicating if a node is *reachable*, is checked. The flag will be set to ‘false’ by the MACL, if the MACL protocol was unable to reach the intended destination node on previous attempts. This may be due to energy depletion for example and hence the node may not be functional any more. The CL-CHELAGR is herewith kept up-to-date on which nodes are still functional and reachable and can be considered as a next-hop candidate toward the actor node. This will prevent the MACL from discarding packets received from the CL-CHELAGR protocol that are addressed to ‘dead’ nodes. This in turn increases the reliability of packets reaching the actor node.

As explain in section 0, the APPL data packet type determines the routing approach of critical and non-critical packets. For critical packets, the real-time requirement receives top priority and hence, the following routing approach is taken. The primary concern for critical packets is to get them to an actor node as soon as possible. The quicker the critical event is reported to the actor node the sooner the detected problem can be attended to. It is noted that the most energy efficient route is not always the fastest route. It is proposed that the critical packets are only interested in the fastest route. The layered CHEAGR protocol design (of section 5.5.2) proposed the routing of packets to the next-hop node which has the highest ratio of residual energy and traffic load. This does not guarantee the fastest route and therefore, the MACL protocol wakeup time, wakeup interval and the channel condition parameters are used as additional criteria for critical packet forwarding as follows.

The *wakeup times* of the neighbour nodes are stored by the contention-based MACL for the purpose of estimating the next listen slot and then calculating a starting point for the preamble sampling. It is therefore proposed that CL-CHELAGR bases its decision on the time-to-next-wakeup of the destination node candidates when choosing the next-hop node. In this way the critical packet is always sent to the next-hop node that wakes up the earliest, ensuring the fastest route towards an actor node.

The *wakeup interval* of all sensor nodes in the WSAN, as proposed by the layered CSMA-MPS protocol (of section 5.6.2) is equal and stays constant. However, an enhanced optional MACL feature that is proposed in the next section (section 5.8.3) involves duty cycle doubling for nodes chosen for a critical route. This feature may be employed to speed up the critical packet relaying even more. To ensure that nodes make full use of this feature, the CL-CHELAGR chooses the next-hop node, among the candidates possessing a shorter hop distance, that has enabled the duty cycle doubling feature and hence possesses a shorter wakeup interval. A node with the duty cycle doubling feature enabled is able to receive packet twice as often, reducing the latency of a packet travelling to the actor node. Choosing the node with the shortest wakeup interval ensures that the same route is used towards the actor node as initially selected and notified via a CRITICAL_ATTENTION_MESSAGE packet. Therefore, only the minimum number of nodes in the direct route towards the actor node is involved in enabling duty cycle doubling.

The final MACL neighbour parameter that is proposed to be used by CL-CHELAGR is the *channel condition*. This parameter is an indication of the number of CS failures a neighbour node has recorded over a recent period. A CS failure means that the node detected a carrier during the CS period, implying that the channel is busy and currently occupied. Neighbour nodes with a high number of CS failures will most likely incur a higher delay, since the channel is occupied very often due to a high node density for example. On the other hand, a node with a lower number of CS failures indicates that the node has less difficulty accessing the wireless channel and hence may be able to transmit a packet sooner. Therefore, for the initial CRITICAL_ATTENTION_MESSAGE packet (originating from the critical event detecting sensor node) the node that has the lowest time-to-next-wakeup is chosen as the next-hop node only if that node's number of CS failures is lower than the rest of the candidates'. If the node with the shortest time-to-next-wakeup has a channel condition of 50% more CS failures than the second shortest time-to-next-wakeup candidate, the second shortest time-to-next-wakeup node is chosen as the next-hop node. It is very important to consider the channel conditions, since it influences the latency of packets traversing through the network towards the actor node. If a node experiences a high number of CS failures in trying to reach the next-hop node, it has to repeatedly back-off and may miss the listen slot all together. It then has to try to reach the destination node in its next listen-slot. This adds a considerable amount of latency to the packet.

For the critical delay-sensitive data packets that follow the initial attention packet, the next-hop node with the shortest wakeup interval is chosen. If there are multiple candidates with the same short wakeup interval, the candidate with the shortest time-to-next-wakeup is chosen. Secondarily, if the time-to-next-wakeup times are very similar, the candidate with the best current channel condition is chosen as the next hop.

For non-critical packets, where energy efficiency is considered as the top priority, the route decision is enhanced by considering the channel condition in addition to the traffic load. The next-hop candidate with the highest ratio of

$$\frac{E_R}{TrafficLoad \times CSfailures} \tag{5.4}$$

is chosen as the next hop in the route towards the cluster-head actor node. This approach ensures that packets are not routed towards very busy and high interference prone parts of a cluster. Consequently, the packet latency may decreased since packets are routed to the less ‘busy’ nodes with a less ‘busy’ channel.

During less eventful periods, the majority of packets are non-critical and hence the NETL traffic load parameter is an important one to consider. However, for the routing of critical packets, the traffic load is not considered important since critical packets receive a higher priority than non-critical packets in the CPS. The traffic load parameter is less relevant due to the fact that the critical packets are not delayed by the MACL priority queue and surpass all other non-critical packets with which a node may be busy with, as proposed below in section 5.8.3. Therefore, the initial definition of the NETL traffic load is less relevant for the handling of critical packets, since this is rather a measure of how busy a node is with non-critical packets. Furthermore, due to the fact that the number of critical packets is far less than the number of non-critical packets at any time in the network, the traffic load metric is used in the routing decisions of non-critical packets only.

5.8.2.2.2 Channel condition

The MACL channel condition is an indication of how busy the channel is as experienced by a node’s radio. The only packet that may be delayed by the CL-CHELAGR is its NETL metric update packets it creates and disperses at a frequency dependent on the consumed energy and current traffic load, as discussed in section 5.5.2.2. It is proposed that the channel condition is added to the criteria to determine the frequency of dispersion and in this way regulate the traffic congestion or load on the channel. When a high CS failure rate is recorded (indicating that a lot of activity exists on the wireless channel), the load is alleviated by doubling E_C, to temporarily lessen the amount of NETL packets being generated. Therefore, the CL-CHELAGR considers both the traffic load and channel condition parameters when determining the value of E_C.

5.8.2.2.3 Queue information

The CL-CHELAGR protocol uses the maximum and current MACL queue size parameters, stored in the SCLF’s information pool, to determine if the NETL metric update packet dispersion should be delayed. Since the updates are sent to multiple relevant neighbour nodes, the maximum and current MACL queue size parameters are used to calculate how many NETL update packets can be passed to the MACL at a specific moment. If the MACL queue cannot handle the full amount of NETL update packets to be sent, CL-CHELAGR only passes the amount that ensures the queue does not overflow. CL-CHELAGR waits and periodically checks if it can pass the rest of the update packets to the MACL. In this way, the MACL is given time to empty its queue. This ensures that the MACL queue never overflows and prevents the discarding of packets.

5.8.3 The Cross-Layer enabled MACL Protocol Design

The layered contention-based CSMA-MPS MACL protocol involves the sharing of the wireless medium by periodic wakeup cycles of the nodes. Nodes spend the majority of their lifetime in sleep mode, to conserve energy. This approach primarily strives towards high energy efficiency and maximum network lifetime, by also making use of an alternating preamble sampling mechanism. The protocol simply preambles the destination node that was selected by the NETL and does not have any influence on the next-hop selection process towards an actor node. With the proposed cross-layer paradigm, special focus is set on the WSAN real-time requirement, which has not received specific attention by the CSMA-MPS protocol.

A cross-layer approach is adopted by the layered CSMA-MPS protocol, to provide an improved solution to the latency vs. energy trade-off. As mentioned in the discussion of the cross-layer enabled APPL protocol (section 5.8.1), the fundamental methodology of the proposed cross-layer CPS involves the immediate reaction by protocol layers to the APPL state and APPL packet type. Therefore, whenever critical, highly delay-sensitive information exists, the protocol layers regard latency minimization as the top requirement. Whenever non-critical, delay-insensitive information exists, the protocol layers regard energy efficiency as the top requirement. In this way, the slightly neglected real-time requirement enjoys ample attention when adopting of the cross-layer interaction paradigm. The MACL is hence not solely responsible for satisfying the latency and energy efficiency requirements. It is handled in a more unified manor by the more effective cross-layer interacting CPS.

In this section, a new Cross-Layer enabled CSMA-MPS with Duty Cycle Doubling (CL-CSMA-MPS-DCD) is proposed. The MACL adoption of the cross-layer paradigm is proposed, which involves specific modifications and enhancements to the layered CSMA-MPS MACL protocol of section 5.6.2. The additional functionality and benefits gained by the incorporation of specific cross-layer parameters into the layered MACL protocol are also discussed.

5.8.3.1 Information used from the Application Layer

In this section, APPL parameters that are considered by the new CL-CSMA-MPS-DCD protocol are given. Furthermore, the modifications and enhancements to promote latency and energy efficiency optimization by incoporating the APPL parameter into the layered CSMA-MPS protocol (of section 5.6.2) are discussed.

The identified APPL parameter that is beneficial to the MACL is given as follows.

1. APPL State

5.8.3.1.1 APPL State

The layered NCAC-DP protocol (of section 5.4.2) may operate in one of four states that include INITSETUP, NORMAL, CRITICAL and INCRITICALROUTE. The last three states are considered significant for the CL-CSMA-MPS-DCD. The NORMAL state depicts a node that obtained a sensed value that falls below the critical threshold level. The CRITICAL state indicates that a node has sensed a critical event or disturbance, i.e. a value above the critical threshold level. Finally, the INCRITICALROUTE state is entered if a critical packet is received by a node and acts as an intermediate node in the route towards an actor node.

The proposed cross-layer induced enhancement involves among others, the temporary doubling of the duty cycle for critical event reporting. This feature is adopted from the WSN Dynamic Sensor MAC protocol discussed in section 3.4.3.1 and illustrated in Figure 3.16. However, the primary reason for doubling the duty cycle is to minimize the latency of critical event reporting and not solely for congestion control as proposed by the DSMAC protocol. The major drawback of DSMAC is its unregulated duty cycle changes, which may lead to high energy inefficiencies.

As explained in the CRITICAL operational state of NCAC-DP (in section 5.4.2.3), sensor readings are performed at least three times more often at the onset of a detected critical event. This implies that more than one critical packet will be sent by the node that detected the event, temporarily incurring a higher traffic load. To provide faster packet relaying, to alleviate the incurred traffic load and lower the collision probability, the duty cycle is temporarily doubled by CL-CSMA-MPS-DCD. This is only done when the APPL state changes to INCRITICALROUTE, i.e. the node is used as an intermediate node in the route of a critical event reporting packet. By exactly doubling the duty cycle of the initially synchronized schedule, new schedule exchanges between the DCD (Duty Cycle Doubling) node and all of its direct neighbours are unnecessary. The node is simply able to receive data double as often, while the original schedule is still valid for neighbour nodes. DCD is disabled again if the APPL state changes to NORMAL, after receiving the CRITICAL_RESOLVED_MESSAGE type packet. This ensures that a node returns to a more energy efficient mode once the event has been resolved, by employing the original low duty cycle. If after a time $T_{resolve}$ after receiving the first critical packet, neither more critical packets nor the CRITICAL_RESOLVED_MESSAGE packet is received, DCD is disabled. This is done to prevent a node from unnecessarily remaining in the duty cycle doubling mode and wasting precious energy when the initially detected critical event might already be resolved. This could happen if the CRITICAL_RESOLVED_MESSAGE packet gets lost due to a collision for example and therefore never reaches all the intermediate nodes (all with DCD enabled) in the critical route towards the actor node. However, since ACKs are used to confirm the data transfers, this is highly unlikely and the CRITICAL_RESOLVED_MESSAGE packet should reach all the intended nodes.

A special case to be handled involves a node in the INCRITICALROUTE APPL state that detects a critical event itself and then changes its state to CRITICAL. In this case, DCD is still enabled until the CRITICAL_RESOLVED_MESSAGE packet is received or the resolve timeout (of length $T_{resolve}$) is reached. If the resolve timeout is reached and no more critical packet has to be relayed, CL-CSMA-MPS-DCD will disable DCD.

With DCD, the wakeup interval changes and it is proposed that the destination node communicates the change to the sender by adding another field for the wakeup interval parameter to the MACL ACK packet (sent after successfully receiving the data block). CL-CSMA-MPS-DCD updates the neighbourhood table in the SCLF and this value is in turn used by the CL-CHELAGR protocol in choosing the next-hop candidate with the lowest interval (as explained in section 5.8.2.2). In this way, it is ensured that the same route is taken, once a node has been initially chosen to partake in the critical route towards the actor node. Therefore, only the minimum number of nodes in the direct route towards the actor node is involved in enabling DCD.

The new CL-CSMA-MPS-DCD ACK packet format is shown in Figure 5.20.

Type	Source Address	Destination Address	Next Wakeup Time	Wakeup Interval	Channel condition
4 bits	16 bits	16 bits	32 bits	16 bits	16 bits

Figure 5.20 The new CL-CSMA-MPS-DCD ACK format containing the wakeup interval.

In conclusion, the main objective of temporarily employing DCD is to promote end-to-end latency minimization of delay-sensitive critical packets.

5.8.3.2 Information used from the Network Layer

In this section, NETL parameters that are considered by the new CL-CSMA-MPS-DCD protocol are given. Furthermore, the modifications and enhancements to promote latency and energy efficiency optimization by incorporating these NETL parameters into the layered CSMA-MPS protocol (of section 5.6.2) are discussed. The most important enhancement is the improvement of the ordinary CSMA-MPS MACL queue to a priority queue.

The identified NETL parameters that are beneficial to the MACL are given as follows.

1. Packet type
2. Traffic Load

5.8.3.2.1 Packet type

The NETL packet type is used by the CL-CSMA-MPS-DCD to identify and distinguish between different types of packets. It specifically enables the MACL to distinguish between NETL metric update packets and different types of data packets originating from the APPL. A NETL metric update packet is regarded as a non-critical packet.

The packets received from the NETL have either been classified as critical or non-critical. In the cross-layer paradigm, the CL-CSMA-MPS-DCD is able to view the NETL packet type contained in the header and handle it accordingly. The critical data packets are highly delay-sensitive and hence CL-CSMA-MPS-DCD regards latency minimization as the top priority to ensure that these packets arrive at the cluster's actor node as soon as possible. Dependent on the application however, critical events generally do not occur very frequently and hence the slight negligence of energy efficiency while shifting the focus to minimizing latency is justified. An enhancement to the layered CSMA-MPS MAC protocol is proposed. The basic protocol operation is illustrated in the flowchart of Figure 5.21 and explained as follows.

The first enhancement involves the methodology of handling packets once received from the NETL. It is proposed that the regular CSMA-MPS queue is upgraded to a priority queue. Upon receiving a packet from the NETL, CL-CSMA-MPS-DCD checks if it is a critical or a non-critical type of packet. If it is a *non-critical* packet, the packet is handled normally as stipulated by CSMA-MPS (which can not distinguish between different types of packets). If CL-CSMA-MPS-DCD is currently busy with a sending process, the low priority non-critical packet is simply added to the back of the priority queue. If the received packet is a *critical* packet however, it receives the highest priority. Firstly, if CL-CSMA-MPS-DCD is busy with the sending process of a non-critical packet and is still waiting for the preamble starting point to arrive, i.e. it is still in the waiting period T_{wait}, the sending process is cancelled. The low priority non-critical packet is then inserted after the last critical packet in the priority queue. If the queue does not contain any critical packets, the non-critical packet is inserted at the front.

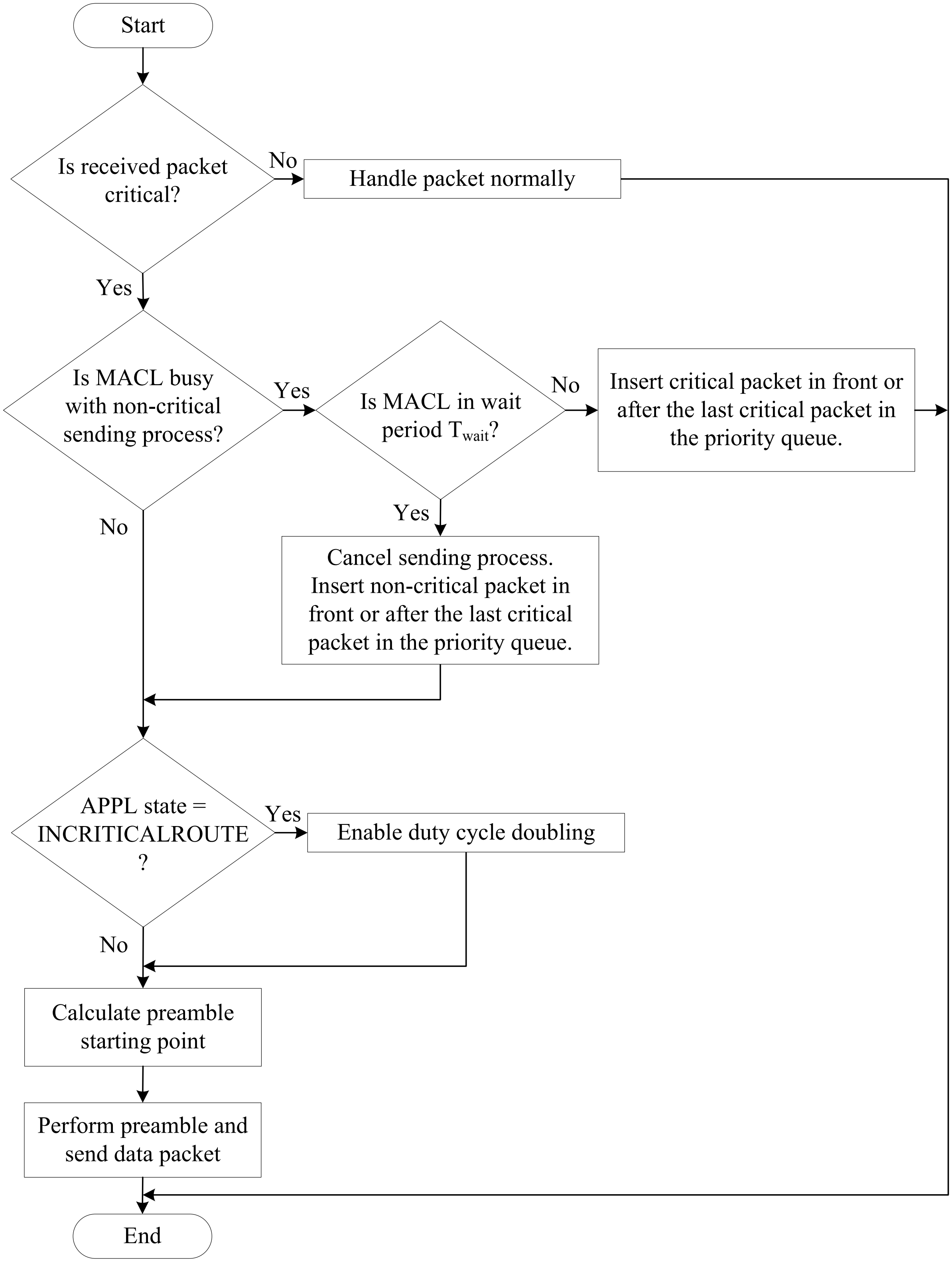

Figure 5.21 A flowchart of the CL-CSMA-MPS-DCD approach for different packet types.

The proposed enhancements of the CL-CSMA-MPS-DCD protocol mainly promote latency minimization for critical packets and energy efficiency for non-critical packets.

As discussed in the normal operation and routing phase of CHEAGR (in section 5.5.2.2), NETL metric update packets are dispersed to the surrounding neighbour nodes. It is proposed that these packets are handled differently to the normal APPL data packets with regards to the MACL queue. Firstly, they are regarded as non-critical packets. Secondly, before a NETL metric update packet is inserted into the MACL priority queue, the queue is first searched to determine if any old NETL metric update packet with the same destination address exist. The old packet is then replaced by the new NETL metric update packet. If no old packet exists, the low priority NETL metric update packet is simply added to the back of the priority queue. This enhancement prevents the occurrence of NETL packet duplicates in the MACL priority queue and reduces the probability of the MACL priority queue overflowing. This in turn decreases the traffic load, promoting the latency minimization and energy efficiency. Furthermore, the MACL data packet format is modified to include the channel condition as illustrated in Figure 5.22. This is only done for NETL metric update packets, since they are sent to upstream nodes. The channel condition may be used for several features of the cross-layer enabled NETL.

Type	Source Address	Destination Address	NETL packet	Channel condition	More Data Flag
4 bits	16 bits	16 bits	Variable	16 bits	4 bits

Figure 5.22 The format of the cross-layer enabled MACL data packet.

A final feature that is proposed for the CL-CSMA-MPS-DCD involves the aggregation of data packets within the MACL. Aggregation involves the sending of multiple non-critical packets that are queued in the priority queue and are destined for the same node. Only non critical packets are involved in the aggregation procedure to prevent latency being added to critical packets. If critical packets were to be sent with one or two other non-critical packets (as part of the aggregation feature), a critical data packet would have to wait for the rest of the non-critical packets to arrive before it can be sent on. Therefore, critical data packets do not partake in the aggregation feature. CL-CSMA-MPS-DCD operates as follows. Once node A established a communication session with a destination node B, i.e. B has replied with a preamble packet, all non critical packets destined for B may be sent one after the other. This approach is viable since the transmitter uses a high bit rate and transmission durations are very short. The first packet in the row of packets sent one after the other contains information in the header, of the number of packets to come. Node B compares this information to the packets received and a final ACK is sent back to the source node A. The source node A can then determine if all packets where received successfully. If any one was not received correctly by node B, node A goes back to SLEEP and re-transmits the failed packets in its next communication session with node B. The number of packets that may be sent is limited to a maximum of three packets, as explained for the layered MACL protocol, to ensure that the channel is not kept occupied for longer period by a communication session between two nodes. This avoids the need for a preamble sequence for every packet in the queue that is destined for the same next-hop node, saving scarce energy.

5.8.3.2.2 Traffic Load

The NETL traffic load is used by the CL-CSMA-MPS-DCD to adjust its priority queue size to prevent packets from being discarded due to the overflowing queue. When a very high traffic load is experienced, the MACL queue size is set to its maximum value and for lower traffic loads decreased accordingly, to free up memory space. The freed memory space may then be used for other purposes. The main concern however, is that the discarding of packets should be prevented at all times. Furthermore, the number of neighbour nodes is also used to determine the MACL queue size, since more dense networks would boast a higher traffic load.

5.8.3.3 Summary of Additional Parameters for the CL-CSMA-MPS-DCD

The proposed CL-CSMA-MPS-DCD protocol features some additional parameters. These parameters are summarized in Table 5.9.

Parameter	Description	Value
$T_{resolve}$	Timeout period during which interval doubling is allowed.	$5 \times T_{crit}$

Wakeup interval	The time between wakeups or listening slots of a node.	Application dependent

Table 5.9. Summary of the additional parameters for the WSAN CL-CSMA-MPS-DCD.

5.8.4 Summary of Cross-Layer Interactions and Protocol Enhancements

The proposed cross-layer interaction and enhancements of the CPS layers are summarized in Table 5.10, Table 5.11 and Table 5.12 for the APPL, NETL and MACL protocols respectively.

APPL CL-NCAC-DP protocol enhancements		
Originating layer	Cross-Layer Parameter	Modification/Enhancement to the layered APPL
NETL	Network Layer Status	▪ The NETL status parameter is polled before initiating the first sensor reading. ▪ Obtainment and sending of sensor readings is stopped once the status changes to DISCONNECTED.
MACL	Maximum and current queue size	Used to delay the sensor reading obtainment from the sensing unit if the MACL queue is too full.

Table 5.10. Summary of the WSANs APPL CL-NCAC-DP enhancements.

NETL CL-CHELAGR protocol enhancements		
Originating layer	Cross-Layer Parameter	Modification/Enhancement to the layered NETL
APPL	State	NETL update metric packets are only sent during the NORMAL APPL state. This prevents congestion during critical event reporting due to these additionally generated packets.
	Sensed packet type	The sensed data packets are classified as either critical or non-critical by the APPL. For critical packets, latency minimization is regarded as the primary objective, and for non-critical packet, energy efficiency maximization is regarded as the primary objective.
	Sensing period T_{sense}	▪ The time period that a new sensor node spends switched on after insertion into an already setup WSAN, is set to three time the sensing period T_{sense}. ▪ The time period $T_{TrafficLoad}$, during which the amount

		of traffic is recorded for calculating the traffic load, is set to 3 × T_{sense} to ensure accurate tracing of fluctuations in the traffic load of a node.
MACL	Neighbourhood table-wakeup time	The routing of critical delay-sensitive packets is based on the time-to-next-wakeup of the destination nodes, where the node with the shortest value is chosen.
	Neighbourhood table-wakeup interval	For the routing of critical packets, the next-hop node with the lowest wakeup interval is chosen over and above the shortest time-to-next-wakeup. If there are multiple candidates with the same low wakeup interval, the candidate with the shortest time-to-next-wakeup is chosen.
	Neighbourhood table-channel condition	▪ For the initial critical attention packet, the channel condition is considered in addition to the time-to-next-wakeup and wakeup interval. If two candidates have the same wakeup interval and very similar time-to-next-wakeup times, the candidate with the best channel condition is chosen. ▪ For non-critical packets, where energy efficiency is considered as the top priority, the route decision is enhanced by considering the channel condition in addition to the residual energy and traffic load.
	Neighbourhood table-reachable flag	The MACL flag is checked to see if the MACL protocol was able to reach the intended destination node on previous communication attempts. If not, the node does not qualify as a next-hop candidate.
	Channel condition	The channel condition (CS failure rate) is added to the criterion for determining the frequency of the NETL metric update dispersion to assist in regulating traffic congestion.
	Maximum and current queue size	The maximum and current MACL queue size is used to determine if NETL metric update packets can be passed to the MACL all at once, piece-wise, or should be delayed all together.

Table 5.11. Summary of the WSANs NETL CL-CHELAGR enhancements.

MACL CL-CSMA-MPS-DCD protocol enhancements		
Originating layer	Cross-Layer Parameter	Modification/Enhancement to the layered MACL
APPL	State	▪ The duty cycle is doubled if the APPL enters the INCRITICALROUTE state and halved again once the APPL switches to the NORMAL state after a CRITICAL_RESOLVED_MESSAGE is received. ▪ If the critical resolved packet is not received and the APPL not set to NORMAL after a time $T_{resolve}$, the duty cycle

		doubling is disabled. This is done to prevent the CL-MACL from keeping the duty cycle doubling enabled in the event of the CRITICAL_RESOLVED_MESSAGE packet got lost.
NETL	Packet type	▪ A priority queue is adopted ▪ Non-critical packets are handled normally, according to the originally proposed layered CSMA-MPS protocol (5.6.2), i.e. the packet is added to the back of the priority queue if the MACL is busy with a sending process. ▪ Critical packets receive the highest priority. ▪ If the MACL is busy with the sending process of a non-critical packet and is waiting for the preamble starting point to arrive, i.e. it is still in the waiting period T_{wait}, the sending process is cancelled and the critical packet handled first. ▪ If a packet of type CRITICAL_MESSAGE, which originated from another node, is received from the NETL after DCD has been disabled, DCD is enabled again and a new timeout timer with time $T_{resolve}$ is set. ▪ Critical packets receive preference in the MACL queue. ▪ The cross-layer feature enables the packets, passed from the NETL, to be identified as either originating from the NETL or APPL. NETL metric update packets are regarded as non-critical. If a new NETL packet is received from the NETL that holds the same destination address as a NETL metric update packet already in the MACL queue, and the newer packet replaces the older one in the queue. ▪ Aggregation: Non-critical packets within the priority queue that have the same destination addresses are sent all in one communication session with the destination node. Limited to a maximum of three packets. ▪ The MACL channel condition parameter is added to the MACL header of NETL metric update packets only.
	Traffic Load	The MAC queue size is adjusted according to the current traffic load, to prevent packets from being discarded due to the overflowing queue. A high traffic load requires a larger queue size, while with low traffic loads, the MAC queue is kept to a minimum size.

Table 5.12. Summary of the WSAN MACL CL-CSMA-MPS-DCD enhancements.

5.9 CHAPTER SUMMARY

In this chapter, a research focus area was initially defined together with key challenges and design steps. A simple cross-layer framework design, containing a neighbourhood table and information pool structures is given for the WSAN CPS. The information contained within the framework can be viewed by all CPS layers.

The initially specified design steps were followed by firstly proposing a complete CPS containing layered protocols and then identifying potential parameters for cross-layer interactions. After that, new cross-layer enabled protocol designs are proposed that involve modifications end enhancement to the layered protocol designs, by incorporating information from other layers. Finally, a fully integrated cross-layer enabled WSAN CPS is achieved. The cross-layer enabled CPS aims at providing performance gains for the WSAN’s defined basic operational scenario with regards to latency and energy efficiency.

Chapter

6 RESULTS AND DISCUSSION

The cross-layer paradigm promises performance gains for network types similar to WSANs. In this chapter, the proposed cross-layer enabled protocols of the previous chapter are evaluated against layered protocol designs. Simulation results for latency and energy efficiency are presented. The results are analysed and discussed with the purpose of establishing if the cross-layer paradigm is beneficial to WSANs under various network conditions.

6.1 SIMULATION AIM

The main simulation aim was to verify that the cross-layer paradigm is indeed advantageous to WSANs with regards to the optimization of latency and energy efficiency. A detailed performance comparison of latency and energy efficiency is conducted between the newly proposed cross-layer enabled CPS, and the layered CPS from which the cross-layer enabled CPS was developed. The layered CPS contains strictly layered protocols that employ no cross-layering at all.

6.2 SIMULATION PLATFORM

Among some popular simulation platforms for simulating computer networks, OMNeT++ was chosen as a suitable simulation environment for WSANs. OMNeT++ is an object-oriented modular discrete event network simulator. It is freely available and boasts several frameworks developed for easy implementation and simulation of networks similar to WSANs. Available frameworks include among others, the INET Framework, SENSIM, EWsnSim and Mobility Framework. Of these frameworks, the Mobility Framework was identified as the most suitable for simulating a WSAN.

It is stated that the Mobility Framework is intended to support wireless and mobile simulations within OMNeT++. The core framework implements the support for node mobility, dynamic connection management and a wireless channel model [88]. Additionally, the core framework provides basic modules that can be derived in order to implement user specific modules.

The Mobility Framework structure provides suitable modules to implement a complete WSAN CPS. A node within the network is represented by a host structure shown in Figure 6.1. A simulated network consists of a number of hosts that each have the CPS layer modules, a Mobility module and a Blackboard module, as depicted.

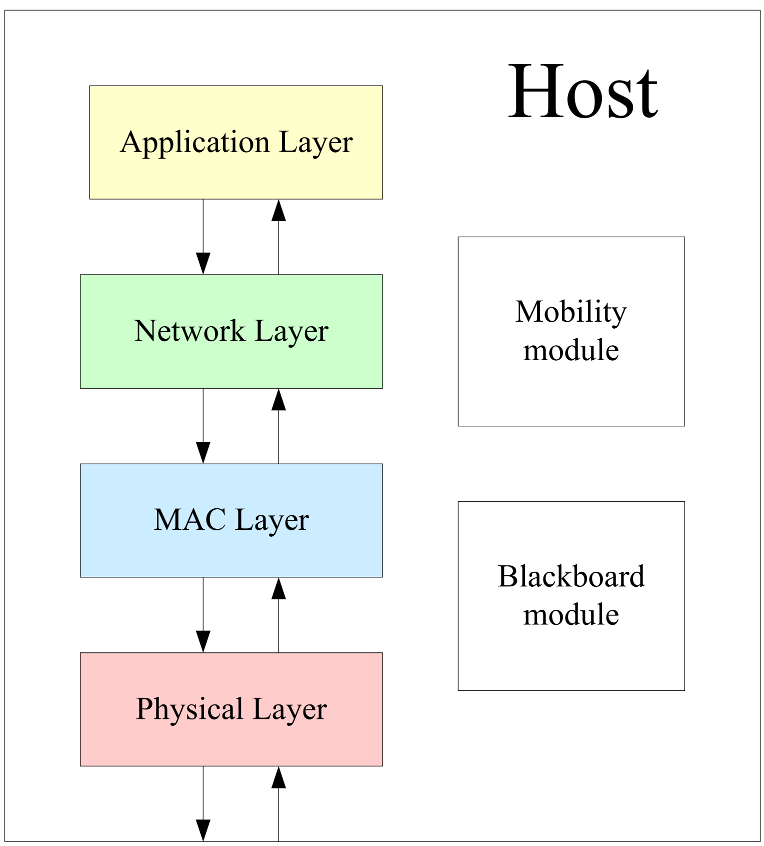

Figure 6.1 The Mobility Framework structure of a host.

The hosts are in turn managed by a ChannelControl module, which controls and maintains all potential connections between the hosts. It is responsible for establishing communication channels between host modules that are within communication distance and tearing down these connections once they loose connectivity. The loss of connectivity may occur due to mobility, a change in transmission power or a failed host as a result of energy depletion.

The CPS layers are implemented in each of the relevant C++ modules in the Mobility framework structure with connections to the direct higher and lower layers. The modules may only pass the packets on to the next direct lower or higher layer. If a packet is created in the APPL for example, it will pass the packet to the next layer, the NETL, which in turn encapsulates the packet, adds its header and passes it to the MACL and so forth. In this way, every layer in the CPS is a separate entity and may pass packets only to the adjacent layers. This constitutes a typically layered CPS.

In order to simulate the desired cross-layer paradigm, all CPS layer modules must be able to communicate with each other. This is achieved by making use of the separate Blackboard module, which may provide an instance for inter-layer communication. Any layer in the CPS can share information by publishing it in the Blackboard module. Any other layer can in turn read the published information by subscribing to it. Only the publisher may change the information. Furthermore, the Blackboard module sends a notification to a subscribed layer if the information is changed by the layer that published the information. In this way, the layer subscribed to a specific piece of information may always keep track of information changes. By using the Blackboard module, CPS layers gain access to any published information and hence the required functionality of the proposed SCLF design is successfully provided by the Mobility Framework. The network host structure with SCLF is illustrated in Figure 6.2.

Finally, the Mobility module is used to determine the position of a node and may also communicate with ChannelControl to facilitate node mobility. New coordinates of a node’s position may be calculated by the Mobility module and sent to ChannelControl, which then determines new or broken connections between hosts.

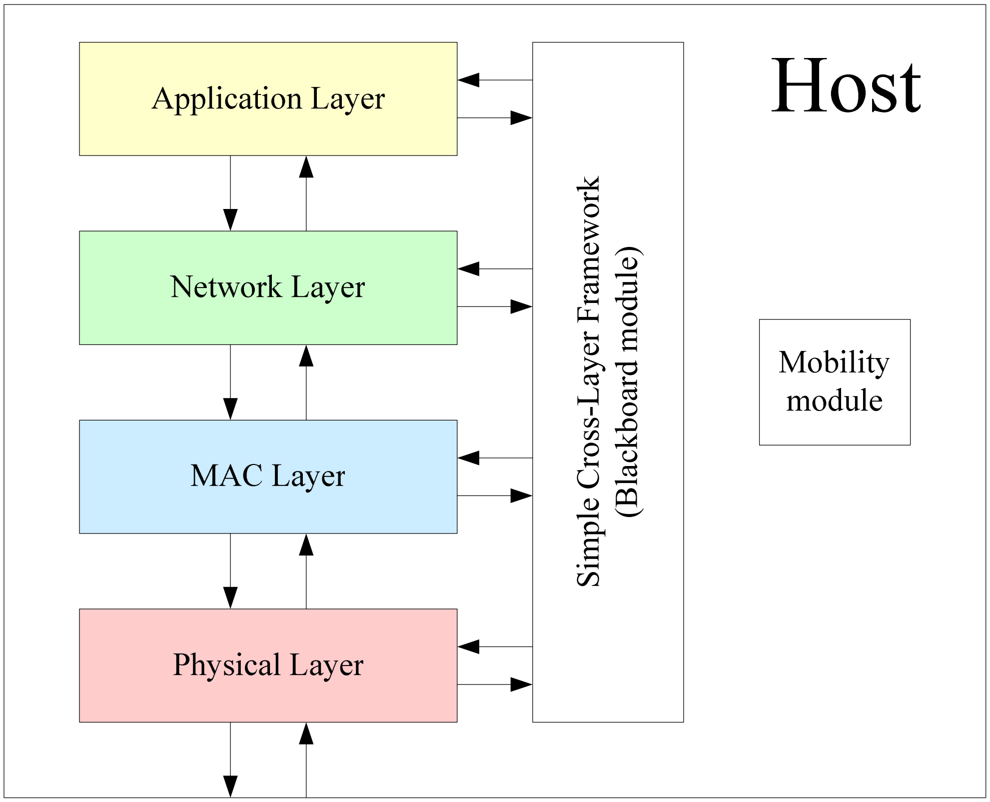

Figure 6.2 The Mobility Framework network host with SCLF.

6.3 SIMULATION DETAILS

In this section, the simulation implementation details are discussed to define how particular characteristics of a real-world WSAN environment are simulated. Furthermore, simulation models for the CPS layers are discussed and present a full implementation of every CPS protocol layer.

With regards to the various simulated protocols, it is to be noted that due to the young nature of the WSAN field of research, it was difficult to find worthy comparison material that is relevant for the defined operational scenario. With regards to existing APPL protocols, only a few exist, but are very application specific. With regards to the NETL, several more WSAN protocols exist, but as revealed in the literature review in chapter 3, the most interesting non cluster-based WSAN NETL protocols have unfortunately been designed and evaluated in conjunction with an inapt WLAN MACL protocol (as discussed in section 5.5.1). The WLAN

MACL protocol is not suitable for use on ultra-low powered WSAN sensor nodes and the NETL protocols can consequently not be considered as viable comparison material. With regards to the MACL, a worthy predecessor to the CSMA-MPS protocol, WiseMAC, was implemented for performance comparison reasons.

The research approach followed in this dissertation involved the selection of an existing favoured layered protocol for each CPS layer, enhancing the layered CPS by applying the cross-layer paradigm and finally verifying the performance improvements. As a result, the newly proposed cross-layer enabled protocols are simulated and performances compared to the layered protocols from which they were developed. This is done to determine if the cross-layer paradigm is beneficial to the WSAN CPS with regards to performance gains.

All simulation relevant models will now be discussed.

6.3.1 The Channel Model

The ChannelControl module requires some parameters to be specified for it to setup connections between the hosts inside the network. It determines the connections by calculating the interference distance according to the free space path loss equation [89].

$$d = \left(\frac{P_{Tx} \times \lambda^2}{16 \times \pi^2 \times P_{Rx}} \right)^{1/\alpha} \tag{6.1}$$

The distance is denoted by d, P_{Tx} is the transmit power, λ is the signal wavelength and can be calculated as the speed of light (2.99792458×10^8 m/s) divided by the signal carrier frequency (2.4 GHz). P_{Rx} is the receiver power and α is the path loss exponent. The path loss exponent replaces the value of 2 in the traditional free space path loss equation, to distinguish between different environments that induce different path losses. The value of 2 is used to calculate the propagation signal loss for open space environments with no intrusive obstacles. A higher value for the path loss exponent indicates a higher loss environment, resulting in a lower interference distance. For instance, in very dense forest areas or office areas with walls, the exponent may typically be closer to 3. For the WSANs environment a value of 2.5 is used. The main parameters relevant for the channel model are given in Table 6.1.

Parameter	Description	Value
λ	signal wavelength	Speed of light / f_c $= 3.0 \times 10^8 / 2.4 \times 10^9$ = 0.125 m
α	path loss exponent	2.5

Table 6.1. Channel model parameters.

6.3.2 The Radio Model

The main component of the PHYL is the radio. The radio model used in all simulations is based on the Chipcon CC2400 radio [13]. The radio model defines the radio states that include {*Sleep*, *Wakeup*, *Listen*, *Transmit*, *Receive*, *Switching}*.

The energy consumption of a radio is determined by multiplying the radio power usage in each state with the time it spends in each of the states. The specific amounts pertaining to the CC2400 radio are shown in Table 6.2 for each state.

Radio State	Power consumption [W]
Sleep	4.95×10^{-6}
Wakeup	3.96×10^{-3}
Listen	3.96×10^{-3}
Switching	3.96×10^{-3}
Transmit	6.27×10^{-2}
Receive	7.92×10^{-2}

Table 6.2. Radio Model state power consumption for the CC2400.

A transmitting radio transmits with a specific transmitter power, bit rate and carrier frequency. The values of these parameters stay constant throughout the simulations.

The radio model incorporates the notion of an interference range and a communication range. The interference range defines up to what distance transmitting nodes may potentially interfere. This range is larger than the communication range and is used by the MACL CS operation to determine if the channel is ‘busy’. If a node A is to transmit while the channel is sensed as ‘busy’, a collision would occur and the received packets would subsequently be invalid. The radio model receiver sensitivity parameter, on the other hand, determines the actual range of successful communication.

Bit errors of a received packet are determined by comparing a list of SNR values recorded during reception of the packet, with a SNR Threshold Level. The SNR value is determined as follows.

$$SNR = \frac{P_{Rx}}{NoiseLevel} \tag{6.2}$$

The radio model contains a NoiseLevel parameter that holds the current noise level. If during the reception of a packet, another packet is received or the received packet’s power is below the radio sensitivity level, the received power level is added to the noise level. Every time the noise level changes during the reception of a packet, a new SNR is calculated and added to the packet’s SNR list. Once the packet reception is over, the SNR values in the list are compared with the SNR Threshold Level. If any of the recorded SNR values fall below the SNR Threshold Level, the packet is regarded as erroneous. Conversely, if a packet is received with a receive power above the sensitivity level and no other interfering transmissions occur, the packet’s list of SNR values will all be above the SNR Threshold level and the packet is regarded as valid.

The main parameters relevant for the radio model are given in Table 6.3.

Parameter	Description	Value
P_t	Transmit power	0 dBm = 1 mW
f_c	RF carrier frequency	2.4 GHz
Bit Rate	Radio bit rate	1 Mbps
Receiver Sensitivity	The minimum receive power for successful communication.	-87 dBm
Carrier Sense Threshold	The power level up to which the CS operation declares the channel as busy or occupied.	-94 dBm
Thermal Noise	The level of thermal noise on the wireless channel.	-110 dBm
SNR Threshold Level	Minimum SNR to successfully recognize a packet.	4 dB

Table 6.3. Radio model parameters.

6.3.3 Network Topology

A realistic network topology of a typical WSAN application example was chosen for the simulations. Each node in a WSAN assigns itself to exactly one actor node and consequently one cluster. Border sensor nodes, that are situated between two actor nodes with an equal hop distance away, will choose only one of the actor nodes (according to the NETL CHEAGR protocol setup and self-configuration scheme) and subsequently belong to only one cluster. The WSAN topology is shown in Figure 6.3, with sensor nodes more or less covering the intended area and the actor nodes located to ensure minimum overlapping of actuation coverage ranges. The WSAN consists of 160 nodes and 3 actor nodes.

Since, for the cluster-based approach, the similar sized clusters that are formed will show similar results, it is feasible to focus on results of one of the clusters within the WSAN. For the conducted simulations, a total number of 50 nodes plus the corresponding actor node form a cluster that covers the intended area more or less, as illustrates in Figure 6.4. The depicted cluster shows nodes with radio links established with a radio sensitivity specification of -87 dBm and transmitter power of 1mW. This enables a maximum communication distance of 75 meters between nodes. As depicted in the figure, some nodes feature more connections than others. The reason for this is that the nodes are not placed at strictly predetermined grid-based locations and also not totally random, but rather in such a way to ensure ample coverage of the indented area.

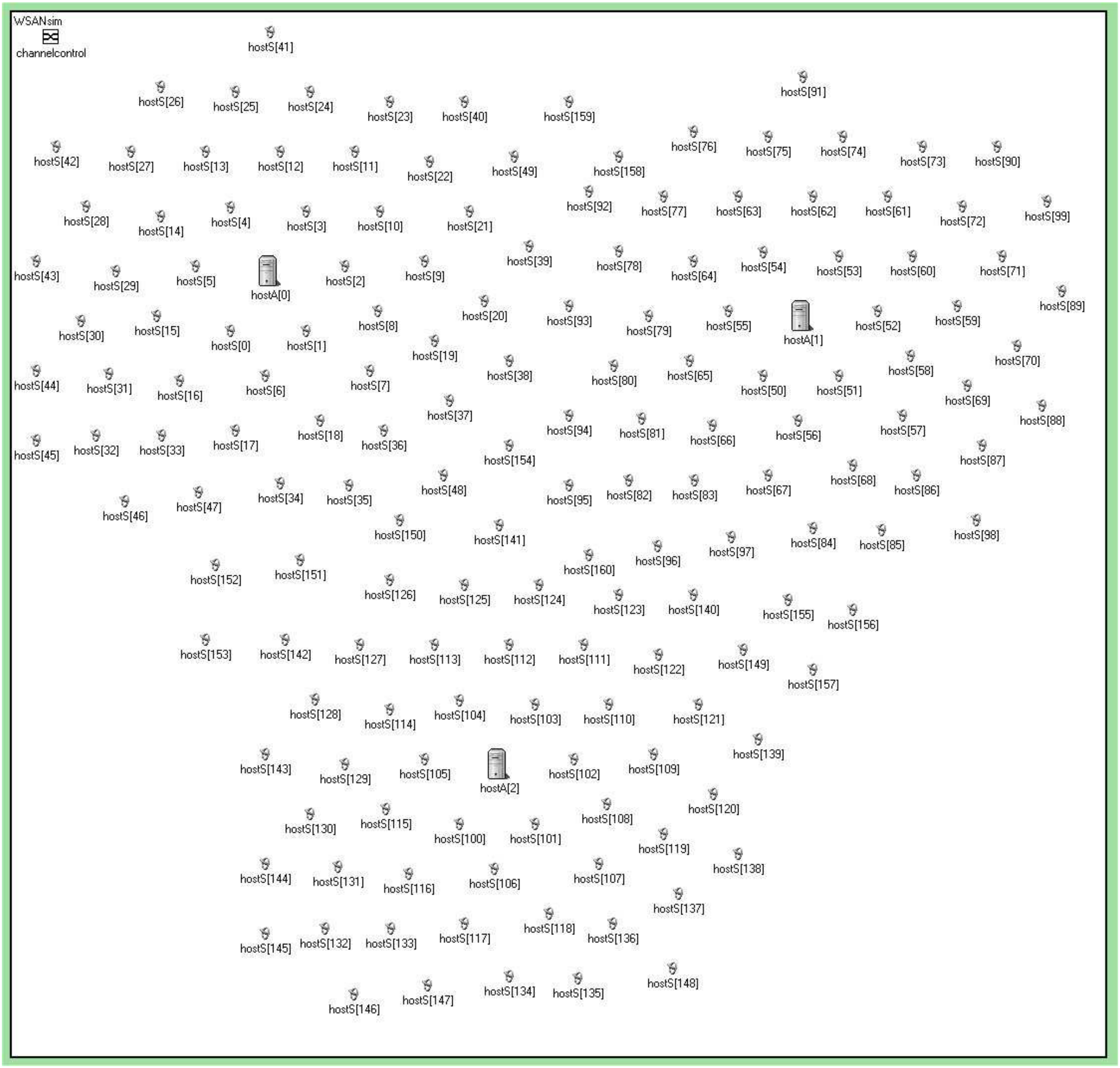

Figure 6.3 The WSAN topology with 160 sensor and 3 actor nodes.

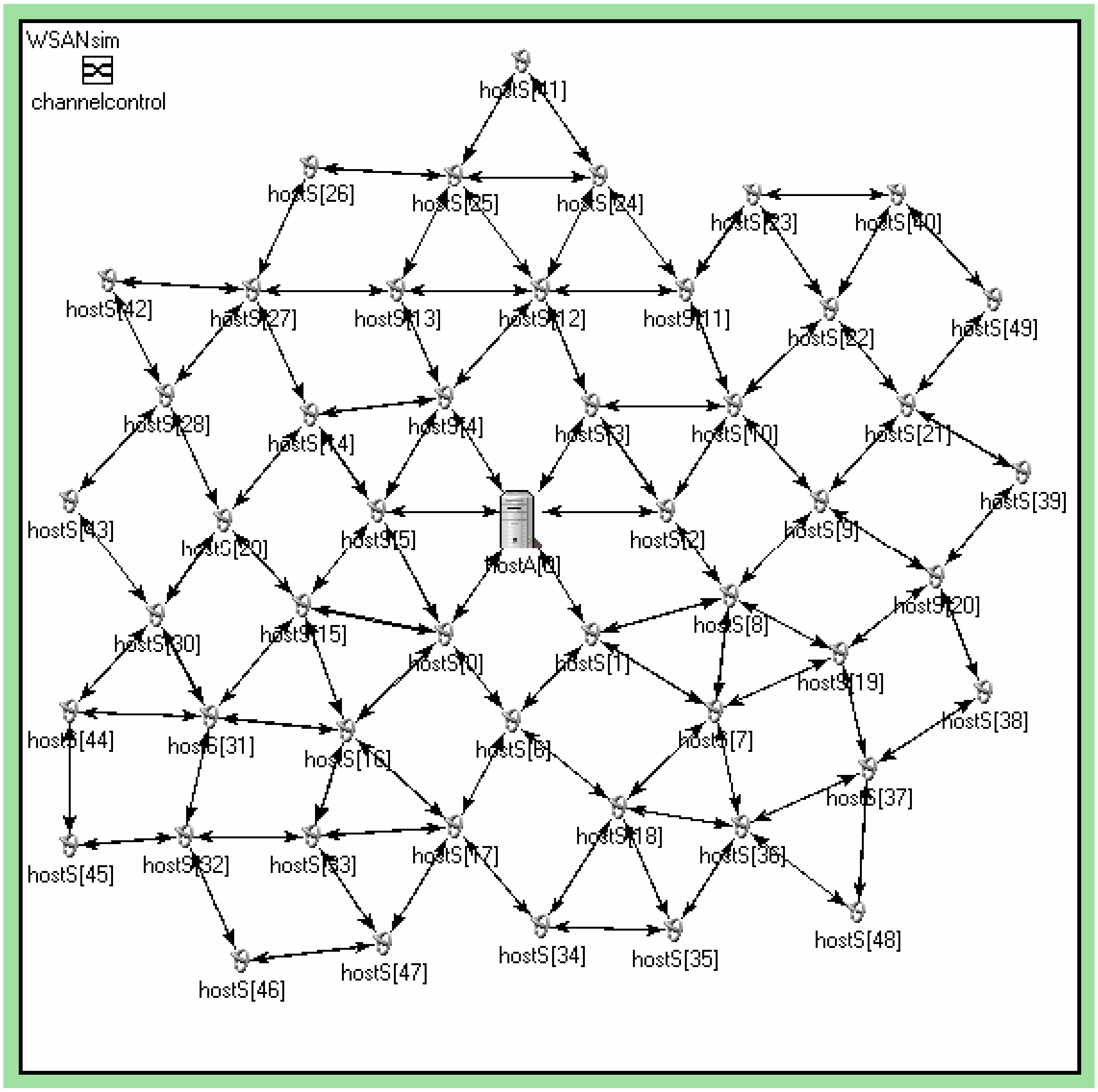

Figure 6.4 A single cluster within the WSAN topology.

The main parameters relevant for the simulation network topology are given in Table 6.4.

Parameter	Value
Number of Sensor Nodes	160
Number of Actor Nodes	3
Playground Size	1100 × 1050

Table 6.4. WSAN topology parameters.

6.3.4 Network Setup

Initially, the NETL protocol will instigate a self-configuration procedure of the network. Every node will detect its direct neighbours in the initial broadcast and network setup phase, which is initiated by a broadcast packet from the actor nodes. The configuration phase is complete once the WSAN has reached a stable state, i.e. every node has assigned itself to an actor node and has identified and gathered all necessary data pertaining to its direct neighbour nodes. Direct neighbour nodes are those, who are within transmission range of the radio.

6.3.5 Statistics Gathering

During the simulations, multiple statistics are continuously gathered to evaluate the overall network performance. The main two include the average *end-to-end latency* of packets and total *energy consumption* of sensor nodes.

The average end-to-end latency experienced by a data packet is calculated as the time from its initial generation by the originating sensor node to its successful reception by the cluster-head actor node. For comparative purposes, the average latency of all data packets received by the cluster-head is calculated.

The packet latency measurements are regarded as the most important among the statistics gathered, since latency minimization is one of the primary objectives of the proposed cross-layer paradigm implementation. Most protocols are already optimized with regards to energy efficiency. In this dissertation, latency minimization of specific important data packets is one of the main objectives.

The energy consumption of a node is calculated as mentioned in section 6.3.2, by recording the times the radio spent in each of its states and multiplying them by the respective radio power ratings. For comparative purposes, the total energy consumption of the cluster is calculated. It should be pointed out that the aim of the protocol simulations is not to derive exact values for energy consumption. Rather, the aim is a relative comparison between the CPSs.

The final metric that is recorded is the number of collisions for the evaluation of the transmitter power control.

6.3.6 APPL model

The functionality of the layered NCAC-DP and proposed cross-layer enhanced CL-NCAC-DP APPL protocols are both implemented as described in section 5.4.2 and 5.8.1 respectively. The sensor node APPL model generates non-critical packets periodically with period T_{sense} and passes them on to the NETL model. Non-critical periodic packet generation commences after a random delay once the network setup phase has reached completion. The random delay is added so that sensor nodes' starting times of periodic packet generation are spread in time and a consistent traffic load is maintained.

The main goal of the APPL model implemented on *actor nodes* is to receive, calculate and log the latency experienced by packets generated by the sensor nodes. No actual simulation of acting or actuator coordination is done since it is beyond the scope of this dissertation and may be different for the wide range of target applications.

The occurrence of critical events inside the network is simulated by randomly selecting *three* sensor nodes that are either 4 or 5 hops away from the actor node beforehand, to present nodes that generate critical events. The selected critical events detecting nodes within a cluster are situated more than three hops away from the actor node. This is done to involve as many nodes towards the actor node as possible. Packets originating from nodes situated only 1 or 2 hops away from the cluster-head actor node, are less influenced by high latency issues. Furthermore, a light-weighted WSAN with few event occurrences seldom suffers from excessive transmission delays. Therefore, the simulations focus on an application scenario that features frequent event occurrences. The APPL model is configured such that the percentage of critical event packets created at the three randomly selected critical sensor nodes is between 40-60% of all their generated packets. As proposed by the APPL protocols, from the onset of a critical event, the node creates event reports at a higher rate than usually (T_{crit}~ T_{sense} /4). T_{crit} is not set too high to avoid totally clogging up the network. After a certain amount of critical packets (15 packets), the event is assumed to be neutralised and the node returns to normal operation. Finally, the critical events are more or less spread evenly over the simulation time.

No latency bounds or expiry times are calculated for APPL packets. All critical data packets should simply arrive at the cluster-head actor node as soon as possible. The model provides every node within a cluster with its unique id and x/y coordinates. Furthermore, every packet is time stamped with the global simulation time once it is created, to ensure that the latency calculated at the actor node presents the correct time between packet generation and successful packet delivery at the actor node and is not influenced by clock drifts.

The main parameters relevant for the APPL model are given in Table 6.5.

Parameter	Value
T_{sense} (Non-critical packet generation interval)	1, 2, 5 or 10min
T_{crit} (Critical packets generation interval once a node has detected a critical event)	1/4 of the non-critical packet interval
Event occurrence	3 randomly selected nodes create critical event packets such that amount equals 40-60% of their total generated packets.
Payload size	13.5 bytes

Table 6.5. APPL model parameters.

6.3.7 NETL model

The functionality of the layered CHEAGR and cross-layer enabled CL-CHELAGR NETL protocols are both fully implemented as described in section 5.5.2 and 5.8.2 respectively. The NETL model initially involves the network setup. All actor nodes broadcast an initial broadcast packet and the sensor nodes assign themselves to a specific cluster around an actor node. After the initial setup, the NETL model's main task involves choosing the next hop node among specific candidates in the route towards the actor node.

CHEAGR is a derivative of a simple existing layered WSAN NETL protocol referred to as Hierarchical Geographic Clustering discussed in section 3.3.3.2. Therefore, CHEAGR represents an exiting layered protocol, which is compared to the new cross-layer enabled WSAN CPS design simulation configurations.

Finally, the use of transmitter power control at the PHYL is proposed by some specific existing WSAN NETL protocols that include the Hierarchical Energy-Efficient Routing with Ripple-Zones, POWER-SPEED and Power Aware Many-to-Many Routing protocols. The major drawback of employing transmitter power control is that the hidden terminal problem is exacerbated. This may lead to a clear increase in the collision probability, requiring more packet retransmissions, resulting in higher energy consumption and increased latency. The non cross-layer interacting CPS simulation configuration is simulated with and without the transmitter power control feature to indicate the difference in collisions probability. This is done to support the claim that power control is unsuitable for use in WSANs.

The main parameters that stay constant and are relevant for the NETL model are given in Table 6.6.

Parameter	Value
Traffic load calculation interval	500 seconds, 3* T_{sense} for cross-layer enabled CPS

Table 6.6. NETL model parameters.

6.3.8 MAC model

The functionality of the layered CSMA-MPS and WiseMAC protocols and the cross-layer enabled CL-CSMA-MPS-DCD protocol are all implemented as described in section 3.4.3.1, 5.6.2 and 5.8.3 respectively. The MACL model provides access into the medium by working closely with the PHYL model. The MACL model controls the radio to achieve the duty cycle with sleep and listen interval times. Furthermore, the MACL model issues commands for the radio when a RX/TX mode switch is necessary, since it can only operate in one of the two modes at any time.

The Mobility Framework's Blackboard module is used to communicate the MACL model commands to the PHYL model and in turn provide the current PHYL state and event information to the MACL model.

Several features contained within the MACL implementation are implemented in such a way as to allow them to be enabled or disabled. This is done to gain deeper insight into what difference a certain feature makes to the network efficiency performance with regards to latency and energy. Specific features that were enabled or disabled within the cross-layer enabled WSAN CPS include the following.

- The duty cycle doubling feature.
- Aggregation: the sending of multiple packets, destined for the same address, in one session.

The main parameters that are relevant for the simulation MACL model are given in Table 6.7.

Parameter	Value
T_w	1,2,5,10, 20 or 30 seconds (value dependent on simulation APPL model parameter T_{sense})
Initial queue size	10
Statistics calculation interval	500 seconds

Table 6.7. MACL model parameters.

6.3.9 PHYL model

The PHYL model used in the simulations basically consists of the radio model as described above in section 6.3.2. Some CC2400 specifications regarding the timing are given in Table 6.8. When employing transmitter power control, the transmitter power can be changed within the radio model. Transmitter power control records incoming signal strength and calculates neighbour node distances. This distance is then used to calculate the minimum transmitter power necessary to reach an intended neighbour node.

Parameter	Value
radioSwitchTime	0.04 milliseconds
radioWakeupTime	1.27 milliseconds

Table 6.8. Physical layer model parameters.

6.3.10 Actor nodes

For the purpose of this dissertation, no elaborate coordination algorithm was implemented for the actor node actuation or collaboration. However, functionality for sensor-actor communication was fully implemented. The implementation enables the actor node to communicate with sensor nodes. Particularly, the MACL and NETL protocols need to correspond to the protocols used by the sensor nodes' CPS.

The mobility of the simulated actor node is set to be very limited. It is assumed that the actor node actuation range achieves sufficient coverage of its cluster nodes.

6.3.11 Overall Simulation parameters

The simulations were conducted in version 3.2 of the OMNeT++ win32 edition and with version fw1_0a6 of the Mobility Framework. Each simulation configuration described in the next section is simulated for a total of 86400 seconds = 1 day.

Parameter	Value
Duration of each simulation	1 day = 86400 seconds

Table 6.9. Overall simulation parameters.

In the next section, the actual obtained simulation results are documented.

6.4 SIMULATION RESULTS

The actual results obtained from the simulation of various relevant WSAN CPSs and features within the CPS are given in this section. For the purpose of evaluating the difference in performance attained by the adoption of the cross-layer paradigm by the WSAN CPS, the new cross-layer enabled WSAN CPS design of section 5.8 is simulated and compared to a non cross-layer, strictly layered WSAN CPS. The strictly layered WSAN CPS includes layered protocol designs as described in sections 5.4.2 (APPL), 5.5.2 (NETL) and 5.6.2 (MACL). Some simulations parameters are changed and features enabled and disabled to ensure a thorough evaluation of the proposed protocol designs and aspect of the designs.

Another layered WASN CPS is simulated that contains the WiseMAC protocol in the MACL. This protocol is only suitable for low density networks that boast a smaller chance of overhearing. When too much overhearing occurs, the energy consumption increases severely.

In the next subsections, the various CPS simulation configurations are discussed and the simulation results are then presented in the following order. First, the average end-to-end latency of critical event data packets of various WSAN CPS configurations is evaluated, followed by the average end-to-end latency evaluation of non-critical data packets. Next, the energy efficiency is evaluated by considering the total energy consumption of sensor nodes in the WSAN. Finally, results for the employment of power control are given.

6.4.1 Simulation Configurations

The fully functional, complete WSAN CPS protocol designs were simulated with different configurations that involve enabling or disabling of several key features. This is done to determine the impact that specific features have on the network performance with regards to packet latency and energy efficiency. Specific CPS simulation configurations are defined with several combinations of three key features that are either enabled or disabled. The features include the following.

1. Cross-layer interactions are either enabled or disabled.
2. Cross-layer with duty cycle doubling at the MACL is either enabled or disabled.
3. Aggregation at the MACL is either enabled or disabled.

The first and most important CPS simulation configuration option concerns the cross-layer interactions. The disabled cross-layer interaction configuration involves the simulation of the non cross-layer, strictly layered WSAN CPS with layered protocols as described in sections 5.4.2 (APPL: NCAC-DP), 5.5.2 (NETL: CHEAGR) and 5.6.2 (MACL: CSMA-MPS). On the other hand, the enabled cross-layer interaction configuration involves the simulation of the new cross-layer enabled WSAN CPS design with cross-layer enabled protocols as described in sections 5.8.1 (APPL: CL-NCAC-DP), 5.8.2 (NETL: CL-CHELAGR) and 5.8.3 (MACL: CL-CSMA-MPS-DCD).

The second CPS simulation configuration option involves a feature pertaining to the cross-layer enabled MACL (CL-CSMA-MPS-DCD) protocol within the cross-layer interactions enabled WSAN CPS design. This feature may only be enabled for the cross-layer interactions enabled CPS. The functional details of the duty cycle doubling feature is described in section 5.8.2.

The final simulation configuration involves the aggregation of data packets at the sensor nodes. This feature may be enabled for the both non cross-layer and cross-layer enabled CPSs. This feature is described in section 5.6.2.4. Only non critical packets are involved in the aggregation procedure to prevent latency being added to critical packets. If critical packets were to be sent with one or two other non-critical packets (as part of the aggregation feature), a critical data packet would have to wait for the rest of the non-critical packets to arrive before it can be sent on. Therefore, critical data packets do not partake in the aggregation feature.

The various WSAN CPS simulation configurations are illustrated in Table 6.10. In the graphs containing the results, the specific CPS simulation configuration is indicated in the legend with the abbreviations as given in the table. For example, a simulation configuration with cross-layer interactions enabled and duty cycle doubling enabled would be indicated as ‘Cross-layer + DCD’ or ‘CL + DCD’ in the legend.

Feature / CPS	Cross-layer Interactions (CL)	Cross-layer duty cycle doubling (DCD)	Aggregation (Aggr)
Non Cross-layer (Non CL)	disabled	disabled	disabled
	disabled	disabled	enabled
Cross-layer (CL)	enabled	disabled	disabled
	enabled	enabled	disabled
	enabled	disabled	enabled
	enabled	enabled	enabled

Table 6.10. Simulated WSAN CPS configurations.

From the simulation network topology, the randomly selected critical nodes are sensor nodes number 28, 39 and 41. Sensor nodes 28 and 41 are sensor nodes with a hop distance of four from the actor node and node 39 with a hop distance of five.

With the assortment of simulation configurations, the fully implemented WSAN CPS variations are simulated with a choice of different parameters, changed to obtain results for a range of different network conditions. Specifically, the period of packet generation parameter T_{sense} (also referred to as sensing interval) within the APPL model, is set to a certain lower value to attain a higher traffic load scenario and to a higher value to attain a lower traffic load scenario. Furthermore, for every specific T_{sense} APPL parameter (i.e. particular traffic load scenario), the MACL wakeup interval, also referred to as listen interval parameter $T_{W,}$ is varied to gain insight into its effects with regards to the latency and energy efficiency performances. Possible APPL sensing period (T_{sense}) and MACL wakeup interval (T_W) parameter values as used in the simulated CPS configurations are given in Table 6.11.

	Higher traffic load		Lower traffic load	
T_{sense}	60 seconds (1min)	120 seconds (2 min)	300 seconds (5 min)	600 seconds (10 min)
T_W	1, 2 and 5 seconds	1, 5 and 10 seconds	1, 5, 10 and 20 seconds	1, 10, 20 and 30 seconds

Table 6.11. Simulation configuration parameters.

6.4.2 Latency Results for Critical Data Packets

The focal point of interest is the end-to-end latency performance of highly delay-sensitive critical packets generated at the three randomly selected critical event detecting sensor nodes. The layered WSAN CPS regards critical and non-critical packets with equal importance i.e. it does not distinguish between them. However, the new cross-layer enabled WSAN CPS presents an approach of explicitly distinguishing between non-critical and critical data packets and specifically focuses on minimizing the end-to-end latency of the critical packets at every layer of the CPS. The effectiveness and timely actuation upon a critical event in a WSAN, depends on the time it takes to get the critical data packets (indicating the current status and severity of a critical event) from event source to actor node.

The critical data packets latency results figures (Figure 6.5 - Figure 6.8) illustrate the latency improvement percentage of critical data packets within the cross-layer interaction enabled CPS WSAN over the cross-layer interaction disabled CPS. The values show the improvement of the average latency of critical data packets when measured in the cross-layer interaction enabled CPS versus the cross-layer disabled CPS. Certain variations are observed for particular configurations. These are evaluated and discussed in the following subsections for higher (T_{sense} of 60s and 120s) and lower (T_{sense} of 300s and 600s) traffic load scenarios.

6.4.2.1 Higher Traffic Load Scenario

Simulated CPS configurations with a sensing interval of 60 seconds, representing a lower traffic load scenario, for corresponding MACL listen intervals of 1, 2 and 5 seconds are illustrated in Figure 6.5. It is observed that all cross-layer enabled CPS configurations result in an increased average latency performance.

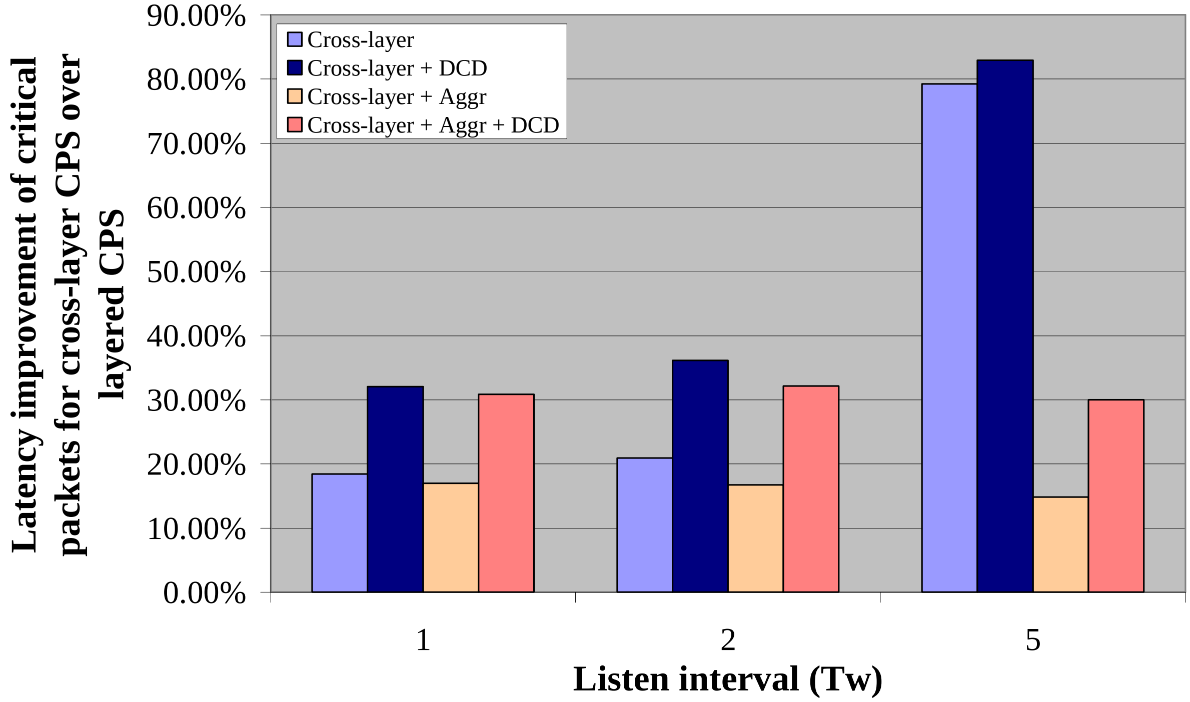

Figure 6.5 Critical packets' average end-to-end latency improvement for simulation configurations with sensing interval of T_{sense} = 60s.

From the figure, it is seen that for a listen interval of 1 second the average latency improvements are between 16.9% and 32%. For a listen interval of 2 seconds, the improvement increases up to 36.1% and for a listen interval of 5 seconds, up to 82.9% for the configuration with aggregation disabled and DCD enabled.

All configurations with *DCD* enabled perform better than their counterparts with DCD disabled. This is an expected result, since the temporary duty cycle doubling of intermediate sensor nodes that are part of the route towards the actor node, wake-up and listen for incoming transmissions twice as often and can hence receive data packets more often. This leads to an improved latency performance due to the fact that the time that packets have to wait for the next listen slot of a specific next-hop node with DCD enabled is shorter. It was observed that the first critical attention packet takes the longest among the critical data packets since nodes only enabled specific latency minimization feature once they receive the initial critical attention packet. Once this packet traverses to the actor node, all sensor nodes along its route go into a latency minimization mode and promote faster relaying of subsequent critical data packets.

As observed in the figure, the latency improvement decreases when the *aggregation* option is enabled. This has to do with the fact that the aggregation already speeds up the relaying of data packets for higher T_W values. Hence, the cross-layer introduced notion of forwarding data packets along a route (NETL) with nodes that wake-up the earliest (MACL) has less of an effect on critical data packet latency. This is especially evident for the simulation configurations with a listen interval of 5 seconds. Here, aggregation has a profound effect on the latency performance. This is attributed to the fact that the low listen interval is less suitable for the high traffic load scenario. Data packets experience high latency due to the fact that sensor nodes wake-up less frequent. Once they wake up, a higher number of neighbour nodes attempt to access the same node, compared to a configuration with a lower listen interval. Consequently, high network congestion is experienced. The congestion is lowered substantially by the employment of aggregation. However, as observed, the DCD feature still introduces an increase latency improvement.

The latency improvements of the simulation configurations simulated with a listen interval of 2 seconds performs similar to the 1 second listen interval simulations. However, an increased performance is seen for configurations with DCD, for both aggregation enabled and disabled configurations. A general tendency that is observed for non-aggregation configurations is that the higher the T_W value, the better the latency performance of the cross-layer enabled CPS. This is attributed to the fact that the critical data packet forwarding of the cross-layer enabled NETL chooses the next-hop node that wakes-up first. The probability that the difference in wake-up times among next-hop candidates is substantially higher for higher values of T_W and hence the cross-layer enabled CPS performs better with regards to the latency minimization.

The best simulated configuration performer for all listen intervals is observed to be the DCD enabled cross-layer configuration. The worst performing configurations are observed to be the cross-layer configurations without DCD for both aggregation and non aggregation enabled configurations respectively.

Simulated CPS configurations with a sensing interval of 120 seconds, still representing a higher traffic load scenario, for corresponding MACL listen intervals of 1, 5 and 10 seconds are illustrated in Figure 6.6. It is observed that all cross-layer enabled CPS configurations result in an increased latency performance.

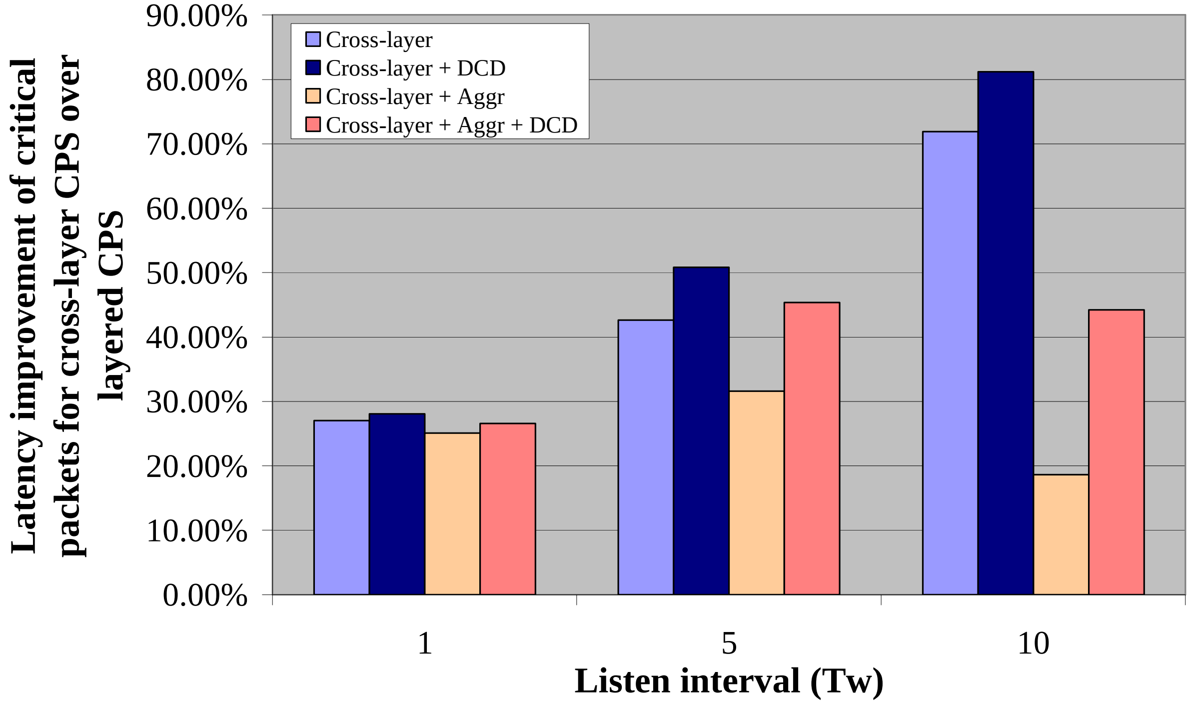

Figure 6.6 Critical packets' average end-to-end latency improvement for simulation configurations with sensing interval of T_{sense} = 120s.

It is observed that for a listen interval of 1 second the average latency improvements are between 25% and 28%. For a listen interval of 5 seconds up to 50.8%. Finally, the latency improvement of simulations with a listen interval of 10 seconds increases up to 81.1% for the configuration with aggregation disabled and with DCD enabled.

The simulated configurations with the marginally lower traffic load scenario of T_{sense} = 120 perform similar to the above given results for T_{sense} = 60 seconds for the listen interval of 1 second. Furthermore, the same performance trend is observed with T_W = 10 seconds as previously explained for T_W = 5 seconds of the T_{sense} = 60 simulation results. Aggregation has a profound effect on the low listen intervals.

The best simulation configuration performers for all listen intervals as expected are the DCD enabled cross-layer configurations. The worst performing configurations are observed to be the cross-layer configurations with aggregation only. Next, results from lower traffic load scenarios are given.

6.4.2.2 Lower Traffic Load Scenario

Simulated CPS configurations with a sensing interval of 300 seconds, representing a lower traffic load scenario, for corresponding MACL listen intervals of 1, 5, 10 and 20 seconds are illustrated in Figure 6.7. It is observed that all cross-layer enabled CPS configurations result in an increased latency performance. Moreover, the cross-layer CPS performs well also in a low traffic load scenario.

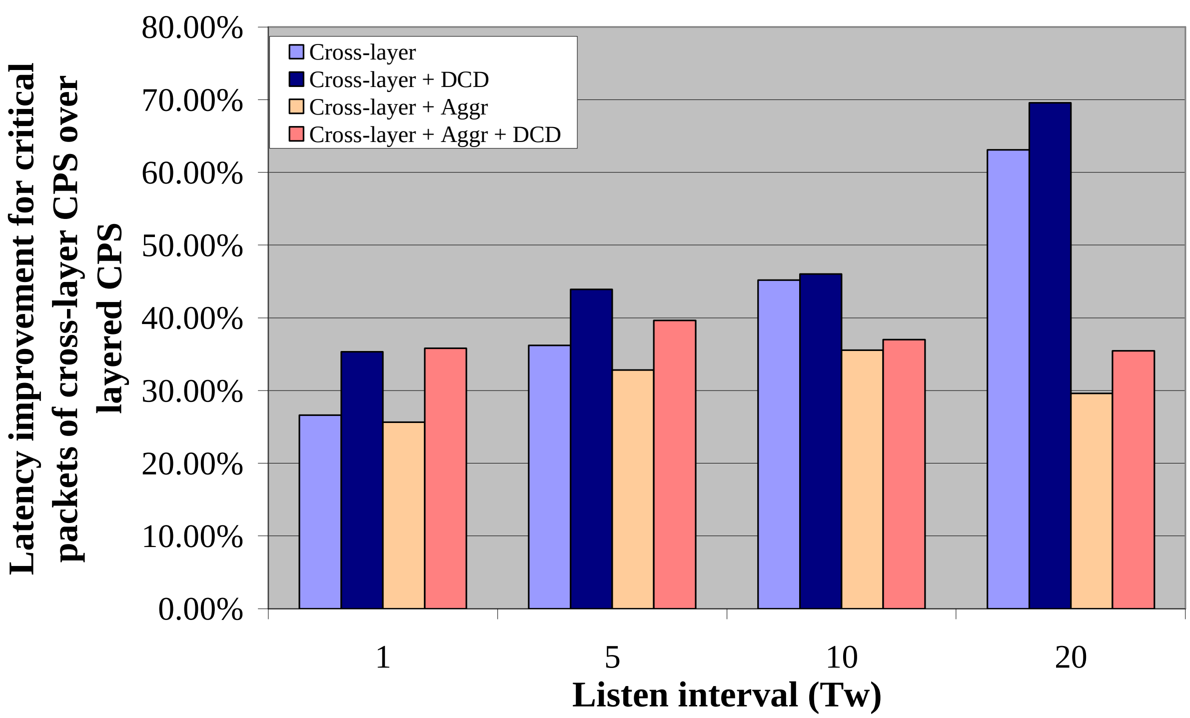

Figure 6.7 Critical packets' average end-to-end latency improvement for simulation configurations with sensing interval of T_{sense} = 300.

It is observed that for a listen interval of 1 second the average latency improvements are between 25.6% and 35.8%. For a listen interval of 5 seconds up to 43.9% and 10 seconds up to 46%. Finally, the latency improvement of simulations with a listen interval of 20 seconds is observed to reach up to 69.6% for the configuration with aggregation disabled and DCD enabled.

As with the higher traffic load scenarios, the latency improvements also increase with the increase in the listen interval T_W. Furthermore, the results for the listen interval of 20 seconds performs as previously explained for T_{sense} = 60 with T_W = 5.

Finally, simulated CPS configurations with a sensing interval of 600 seconds (10 minutes), representing a lower traffic load scenario, for corresponding MACL listen intervals of 1, 10, 20 and 30 seconds are illustrated in Figure 6.8. It is observed that all cross-layer enabled CPS configurations result in an increased latency performance.

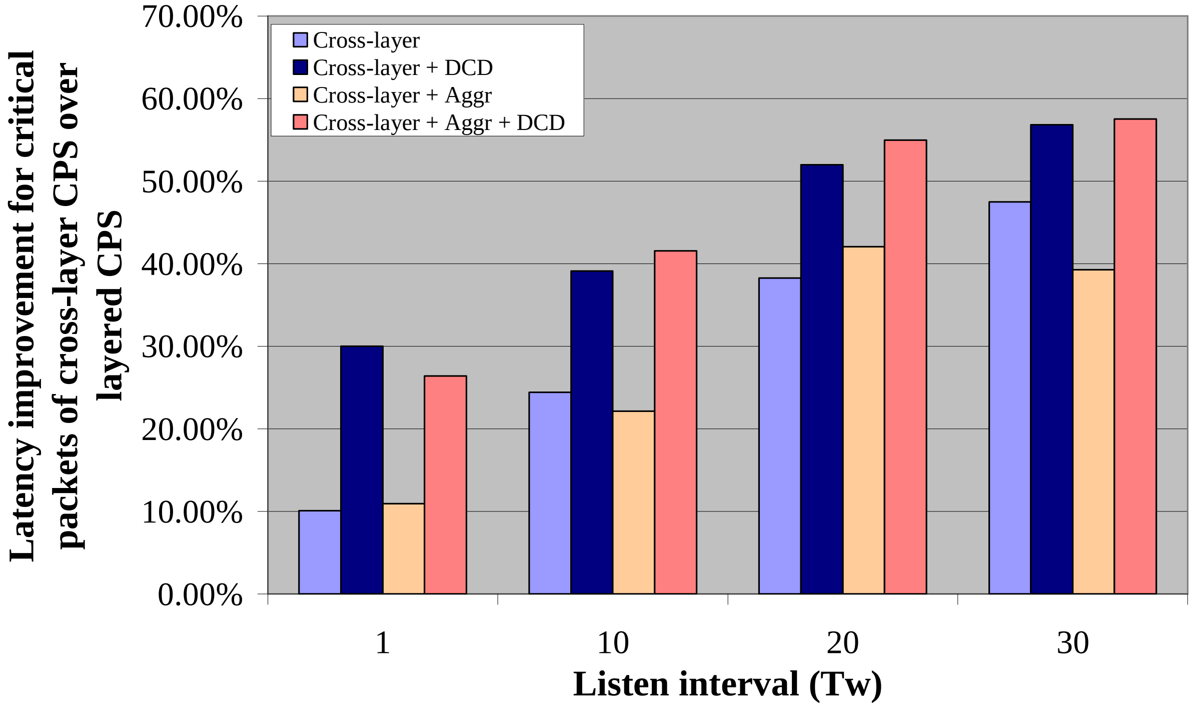

Figure 6.8 Critical packets' average end-to-end latency improvement for simulation configurations with sensing interval of T_{sense} = 600s.

It is observed that for a listen interval of 1 second the average latency improvements are between 10% and 30%. For a listen interval of 10 seconds up to 41.5% and 20 seconds up to 55%. Finally, the average latency improvement of simulations with a listen interval of 30 seconds is observed to reach up to 57.5% for the configuration with DCD and aggregation enabled.

The observed cross-layer performance for the lowest traffic load scenario is similar to the higher traffic load scenarios, where the latency improvements increase with the increase in the listen interval T_W. Although a WSAN that employs a sensing interval of 10 minutes would not use a listen interval of 1 second (for energy preservation reasons), it was simulated for comparative reasons. It is observed that the lower traffic load, incurring a lower level of congestion, results in the cross-layer configurations without DCD performing lower. This is due to the fact that the amount of packets in the network is low and these can hence be relayed very rapidly towards the cluster-head actor node.

Next the average latency of less-delay sensitive non-critical data packets is given.

6.4.3 Latency Results for Non-critical Data Packets

The end-to-end latency results of data packets, regarded as less delay-sensitive and non-critical by the CPS, are given in this section. The non-critical data packets are generated periodically by the sensor nodes within the WSAN. It is important to note that the strictly layered CPS, i.e. with no cross-layer interactions, regards all data packets as non-critical. The APPL logically solely knows of the criticality of specific data packets. However, the other protocol layers of the non cross-layer CPS, regard all data packets as non-critical and handle these accordingly.

The cross-layer enabled CPS however, can distinguish between different data packets (with different priorities) at every layer of the CPS and handles them accordingly. While the latency minimization of delay-sensitive critical packets was of utmost importance, the latency minimization of delay-insensitive non-critical data packets was considered less important in the design of the cross-layer enabled CPS. It was proposed that it is more important to get these data packets with the highest energy efficiency to the actor node, rather than with minimum latency. The main aim of these results is to establish that the latency of non-critical data packets is not totally compromised with the adoption of the cross-layer paradigm.

Firstly, an initial average end-to-end latency comparison is given between a layered CPS containing WiseMAC and the layered WSAN CPS of sections 5.4 to 5.7. Next, the average end-to-end latency results are given for all simulation configurations as stipulated in Table 6.10.

6.4.3.1 Initial Comparison

An initial average end-to-end latency comparison for a sensing interval T_{sense} = 120 seconds, between a layered CPS that implements the existing WiseMAC protocol and the layered WSAN CPS as described in sections 5.4 to 5.7, is given in Figure 6.9. These results are given to establish if the layered WSAN CPS design of this dissertation performs better than an existing layered CPS. WiseMAC is the predecessor of the CSMA-MPS MACL protocol used in the layered WSAN CPS and it is observed that the CPS with WiseMAC performs worse for all listen intervals. According to the simulations performed, for the listen T_W = 1 second, the layered WSAN CPS boasts an improvement of 26%, for T_W = 5 seconds an improvement of 38% and for T_W = 10 seconds an improvement of 24.7%.

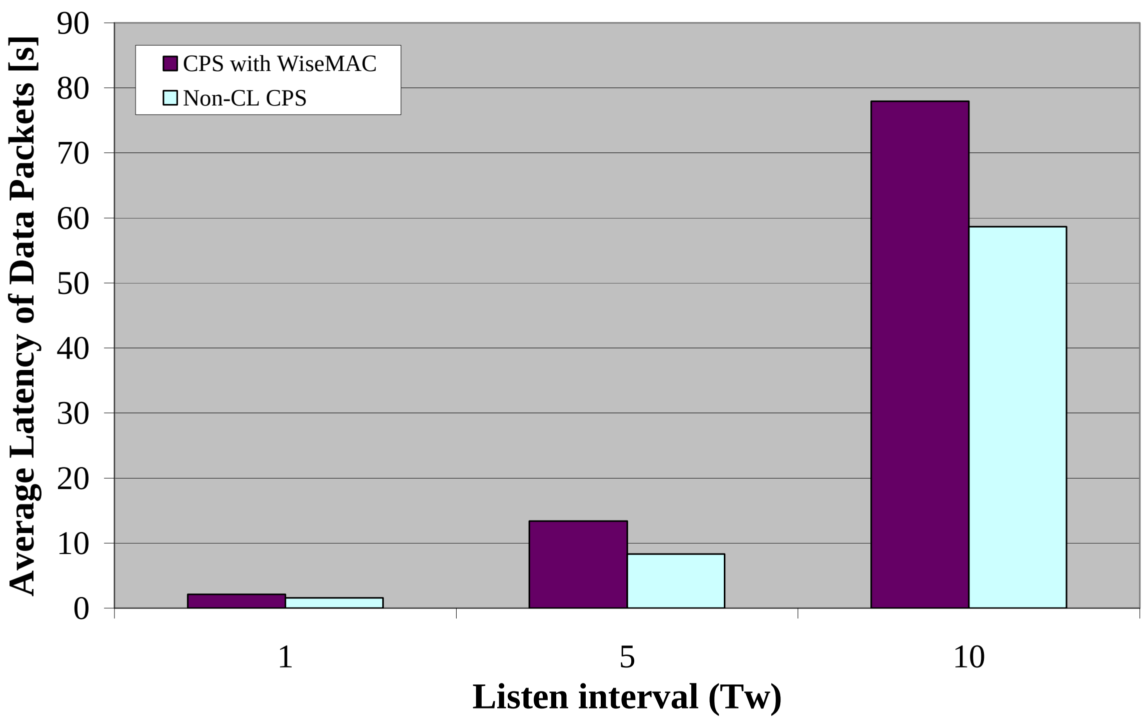

Figure 6.9 Average end-to-end latency of data packets for CPS with WiseMAC and layered WSAN CPS for T_{sense} = 120s.

As mentioned before, the WiseMAC protocol is only suitable for low density networks that boast a smaller chance of overhearing. The difference in latency performance, as illustrated in Figure 6.9, is ascribed to the fact that too much overhearing occurs. Network nodes are forced to receive complete preambles plus data blocks that are not destined for them. This results in hefty delays for other packets that have to wait until the overhearing of foreign packets is over. Consequently, the end-to-end latency experienced by data packets is increased by a notable margin.

It is therefore established that the layered CPS with the existing WiseMAC protocol performs worse than the layered WSAN CPS design that contains the CSMA-MPS protocol with regards to the end-to-end latency of data packets. All subsequent results given in the next section include the layered WSAN CPS containing CSMA-MPS, indicated as ‘Non-CL’ in the graphs, as a reference.

6.4.3.2 Higher Traffic Load Scenario

Simulated CPS configurations with a sensing interval of 60 seconds, representing a higher traffic load scenario, for corresponding MACL listen intervals of 1, 2 and 5 seconds are illustrated in Figure 6.10 and Figure 6.11.

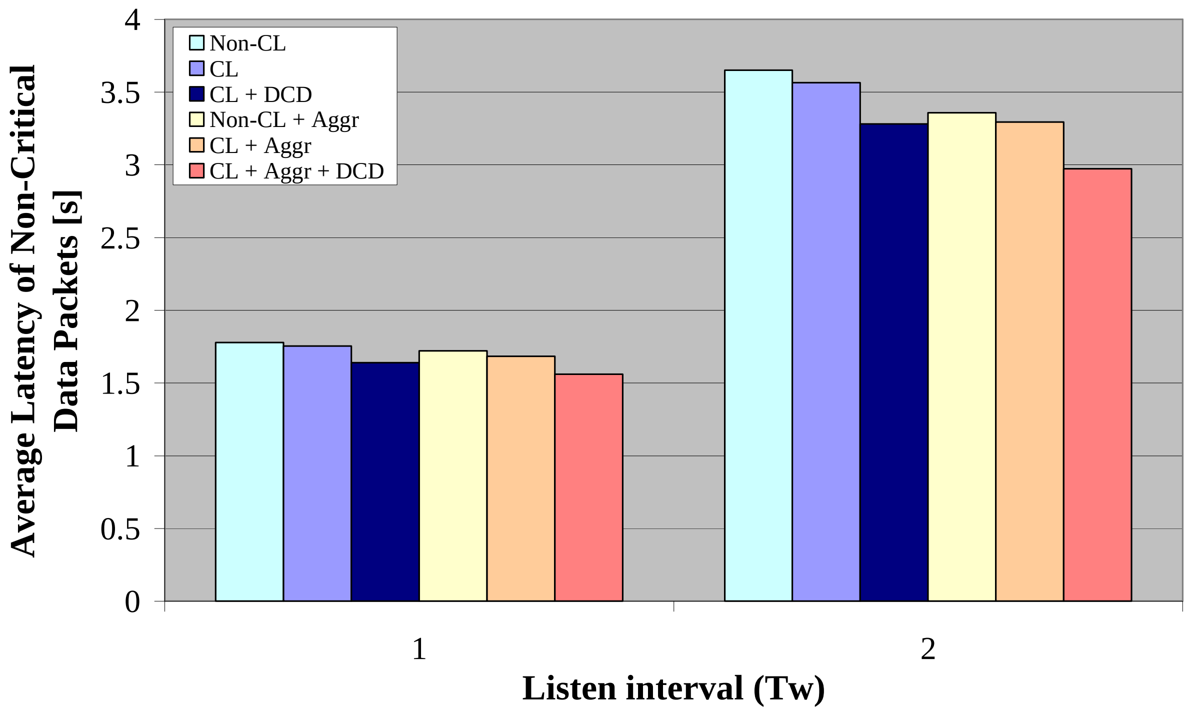

Figure 6.10 Average end-to-end latency of non-critical data packets for simulation configurations with sensing interval of T_{sense} = 60s.

It is observed that the average latency results for listen intervals of 1 and 2 seconds for the cross-layer enabled CPS decreases as compared to the non cross-layer CPS. Although the cross-layer enabled CPS design did not initially specifically aim at decreasing the latency of non-critical data packets, the simulation results does show a decrease in latency. This is attributed to the fact that the cross-layer CPS decreases the traffic load by spreading the load among nodes. Furthermore, the employment of DCD by nodes that receive critical data packets also temporarily influences the latency of non-critical data packets which are relayed during this time. For $T_W = 1$, with DCD enabled, the CPS performs 6% better than without DCD. For $T_W = 2$ it is 9.5% and for $T_W = 5$ it is 16.6%. Similar results are observed for the CPSs with aggregation enabled, with DCD the CPS performs 7.3% better than without DCD for $T_W = 1$, 10% for $T_W = 2$ and 8% for $T_W = 5$.

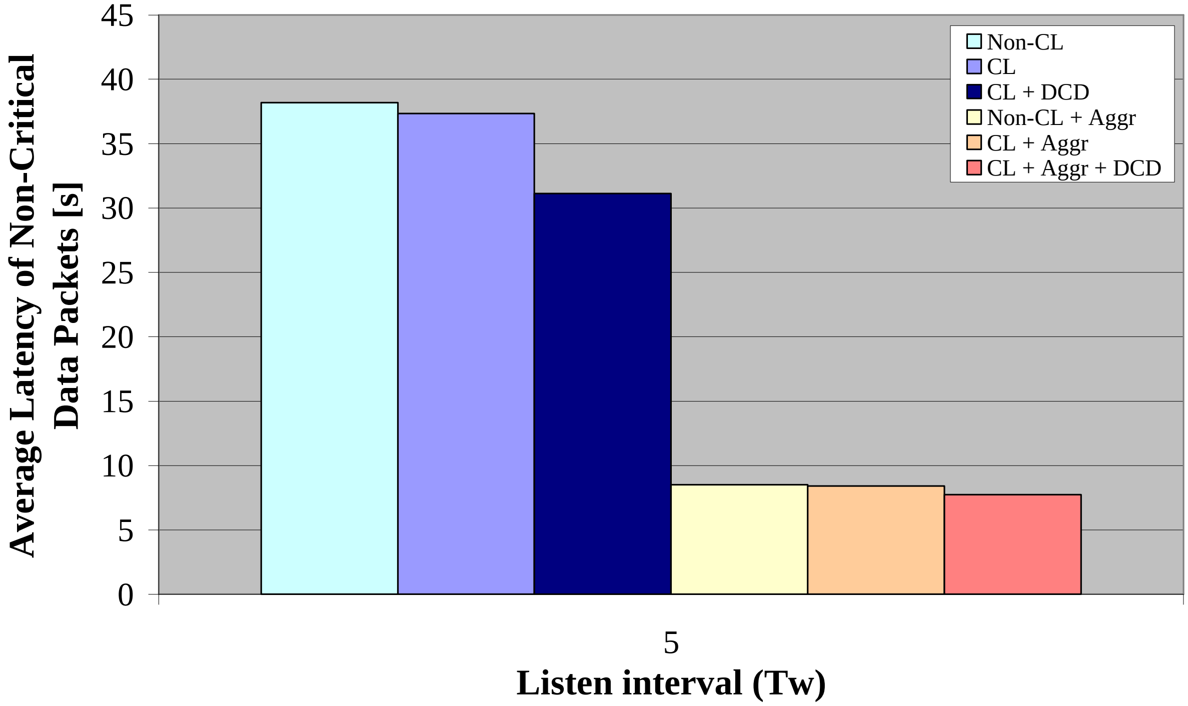

Figure 6.11 Average end-to-end latency of non-critical data packets for simulation configurations with sensing interval of T_{sense} = 60s and T_W = 5.

The results from the simulation configurations for the listen interval of 5 seconds show (as explained in the previous section) that aggregation makes a huge difference to the latency performance of data packets for simulations with lower listen intervals. Furthermore, it is observed that with DCD employed, the average end-to-end latency is lowered further. However, due to these results, the listen interval of 5 seconds would typically not be used without aggregation in a real WSAN application due to the observed response. The CPS configuration without aggregation results in huge delays, which is not viable for WSANs.

Simulated CPS configurations with a sensing interval of 120 seconds, still representing a higher traffic load scenario, for corresponding MACL listen intervals of 1, 5 and 10 seconds are illustrated in Figure 6.12 and Figure 6.13.

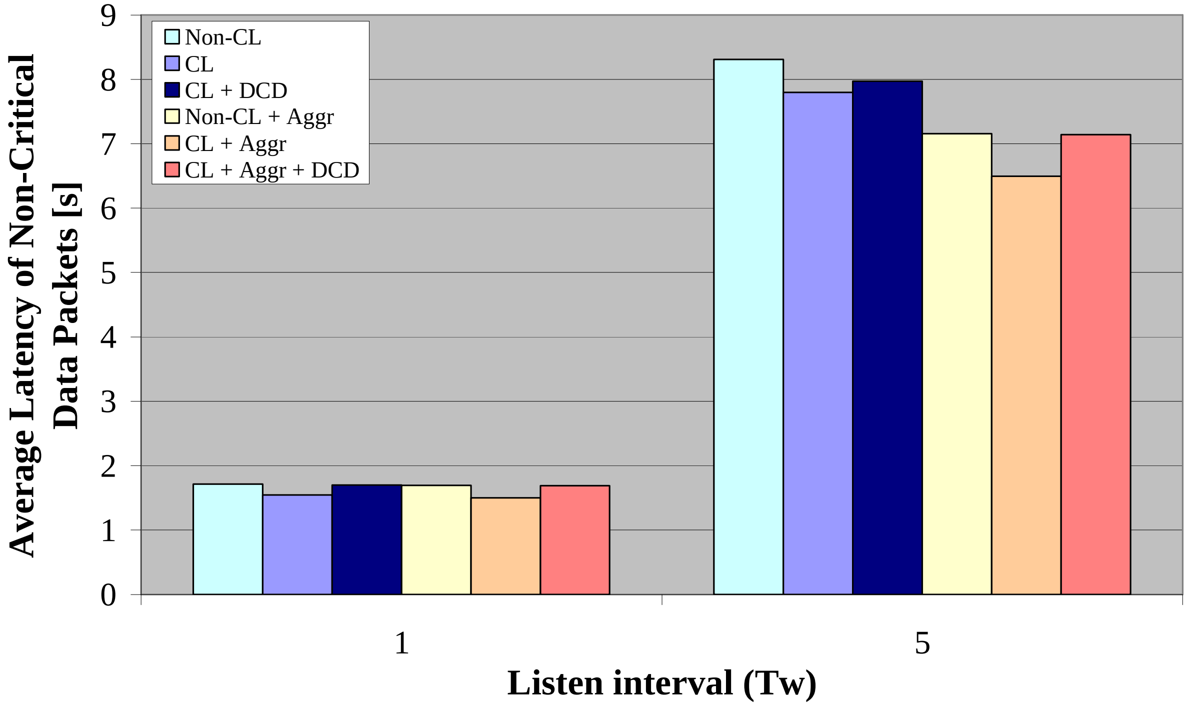

Figure 6.12 Average end-to-end latency of non-critical data packets for simulation configurations with sensing interval of T_{sense} = 120s.

Similar results are observed compared to the sensing interval of 60 seconds. The CL configuration performs better than the Non-CL configuration. However, CL with DCD enabled performs worse than with DCD disabled. This can be attributed to the fact that the handling of the critical packets affects the average latency of non critical slightly negatively. It is observed that with the listen interval of 10 seconds, aggregation has a huge effect on the end-to-end latency and shows the same response as the simulation with T_{sense} = 60 with T_W = 5 seconds, explained above.

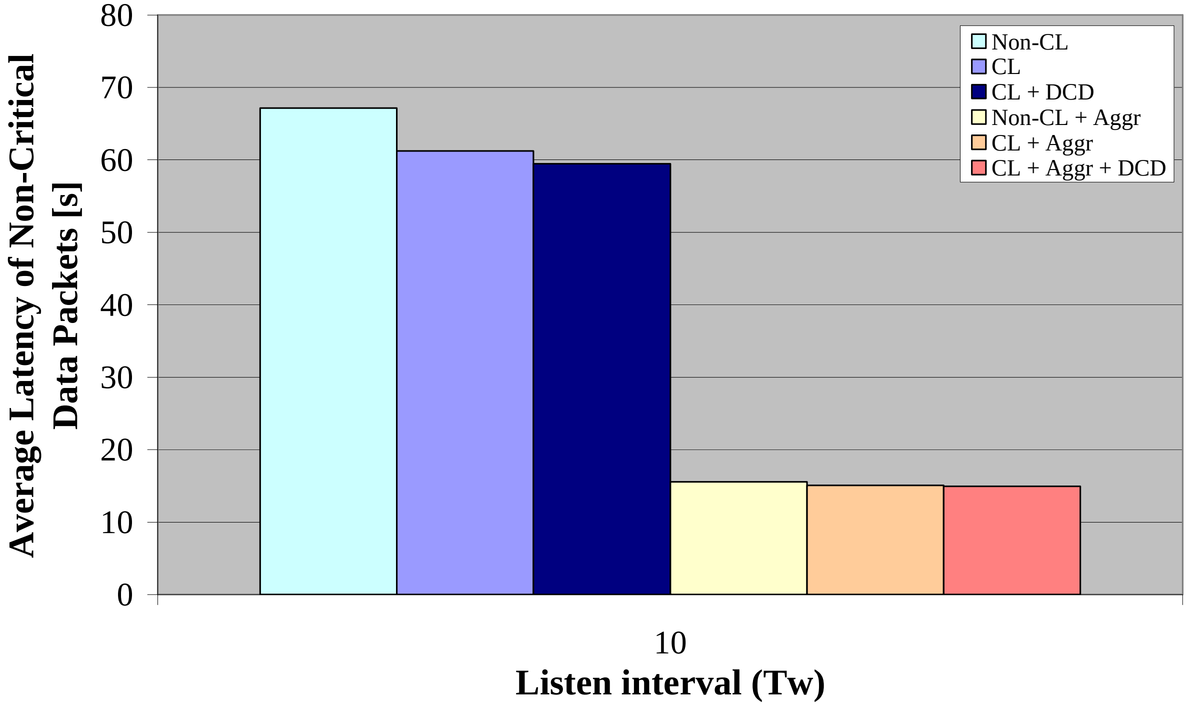

Figure 6.13 Average end-to-end latency of non-critical data packets for simulation configurations with sensing interval of T_{sense} = 120s and T_W = 10.

6.4.3.3 Lower Traffic Load Scenario

Simulated CPS configurations with a sensing interval of 300 seconds, representing a lower traffic load scenario, for corresponding MACL listen intervals of 1, 5, 10 and 20 seconds are illustrated in Figure 6.14 and Figure 6.15. It is observed that for the listen intervals of 10 and 20 seconds, the cross-layer enabled CPS without aggregation provides a reduction in the average end-to-end latency of non-critical data packets of 8.8% and 9.6% respectively. Results for the shortest listen interval T_W = 1 second indicate that there is a minute increase in latency performance. For T_W = 5 it is observed that the latency of non-critical data packets is marginally decreased by the cross-layer configurations.

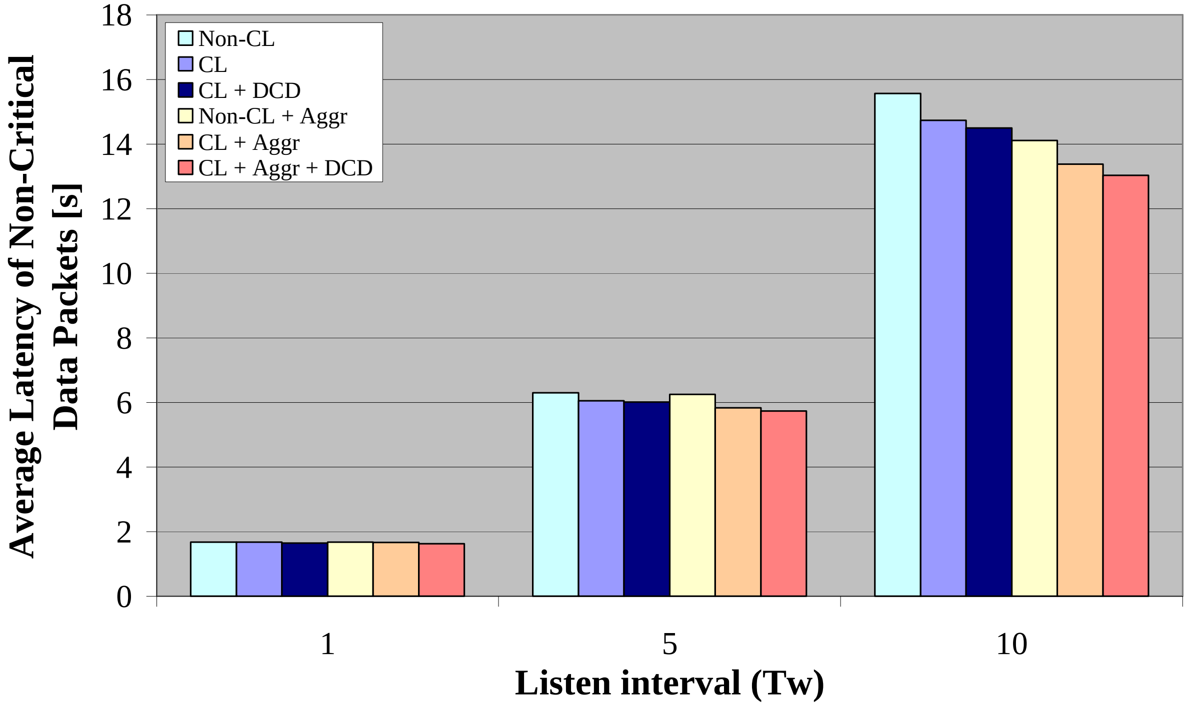

Figure 6.14 Average end-to-end latency of non-critical data packets for simulation configurations with sensing interval of T_{sense} = 300s.

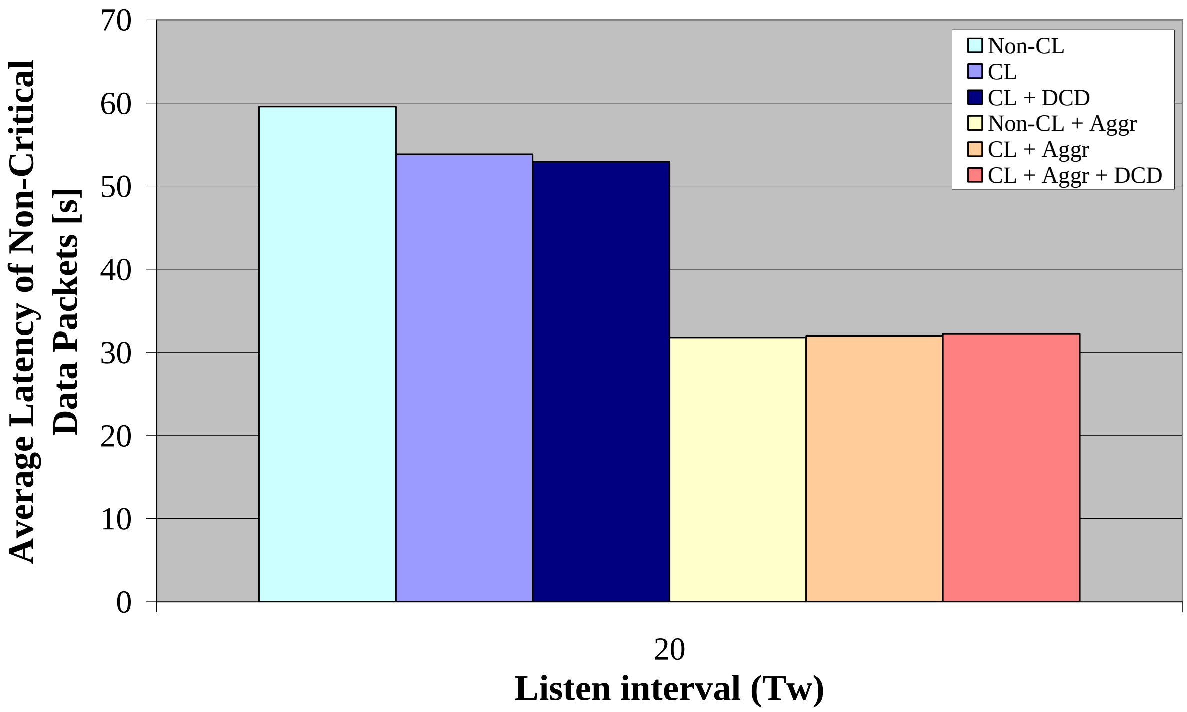

Figure 6.15 Average end-to-end latency of non-critical data packets for simulation configurations with sensing interval of T_{sense} = 300s and T_W = 20.

It is observed that with listen interval of 10 seconds, the latency decreases for the cross-layer enabled configurations of more than 10% latency improvement over the non cross-layer configuration and even further with DCD enabled. The results for the listen interval of 20 seconds again highlight the effect of aggregation on the latency performance for low listen intervals.

Simulated CPS configurations with a sensing interval of 600 seconds, representing a low traffic load scenario, for corresponding MACL listen intervals of 1, 10, 20 and 30 seconds are illustrated in Figure 6.16. It is observed that for the listen interval of 1 and 10 seconds, the latency stays fairly constant throughout the various configurations. For the higher listen interval values, the cross-layer configurations result in lower latencies.

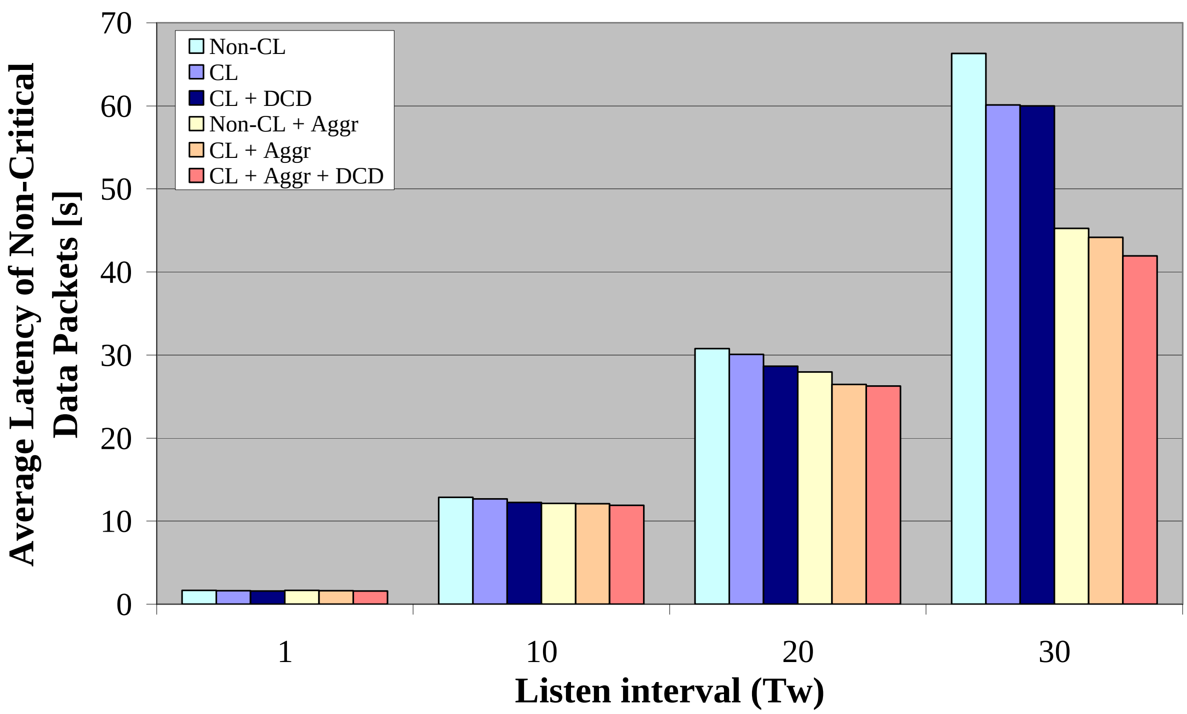

Figure 6.16 Average end-to-end latency of non-critical data packets for simulation configurations with sensing interval of T_{sense} = 600s.

From the above results it is established that the cross-layer CPS does not have a detrimental effect on the end-to-end latency of delay-insensitive data packets. On the contrary, in most cases a noticeable improved latency performance is observed, especially for configurations that have DCD enabled. Furthermore, as expected the latency decreases with the decrease in traffic load.

Next the energy efficiency results are given for simulation configurations as stipulated in Table 6.10.

6.4.4 Energy Efficiency Results

From the above results, it has been established that immense latency improvements are obtained for highly delay-sensitive critical data packets by implementing cross-layer interactions for the WSAN CPS. Furthermore, the latency of delay-insensitive data packets is not compromised by temporarily shifting the focus on latency minimization for critical data packets.

Since the energy efficiency is one of the primary requirements that determine the feasibility of a specific CPS for WSANs, the new cross-layer enabled CPS is simulated against the layered CPS to determine its energy efficiency performance. It was stated in the design that according to the required response, the primary focus is shifted to the requirement mostly needed for a specific scenario. For highly delay-sensitive critical data packets (generated by sensor nodes that detect a critical event), the end-to-end latency minimization is considered as the primary objective while for delay-insensitive data packets, energy efficiency is considered as the primary objective. In both cases the other requirement is not totally disregarded but merely considered secondarily to the primary requirement. It receives a subdued priority but is still considered very important.

In this section, the energy efficiency results of the proposed cross-layer enabled WSAN CPS are given. As previously mentioned, the layered CPS design given in sections 5.4 to 5.7 encompass protocols that primarily focus on energy efficiency. The cross-layer enabled WSAN CPS proposed in section 5.8 on the other hand, introduces additional special latency minimization considerations for delay-sensitive data packets. Energy efficiency results in this section are hence obtained from simulations to establish if the energy efficiency is compromised with the additional cross-layer interactions and latency minimization considerations. Firstly, an initial total energy consumption comparison is given between a layered CPS containing WiseMAC and the layered WSAN CPS sections 5.4 to 5.7. Finally, the results for the total energy consumption of all sensor nodes for the duration of 1 day are given for all simulation configurations as stipulated in Table 6.10.

6.4.4.1 Initial Comparison

An initial total energy consumption comparison for a sensing interval T_{sense} = 120 seconds, between the layered CPS that implements the existing WiseMAC protocol and the layered WSAN CPS as described in sections 5.4 to 5.7, is given in Figure 6.17. These results are given to establish if the layered WSAN CPS design of this dissertation performs better than an existing CPS. WiseMAC is the predecessor of the CSMA-MPS MACL protocol that is used as a basis for the MACL protocol in the layered WSAN CPS and it is observed that the CPS with WiseMAC performs worse for all listen intervals. According to the simulations performed, for the listen T_W = 1 second the layered WSAN CPS boasts a performance improvement of 61%, for T_W = 5 seconds an improvement of 68% and for T_W = 10 seconds an improvement of 66%.

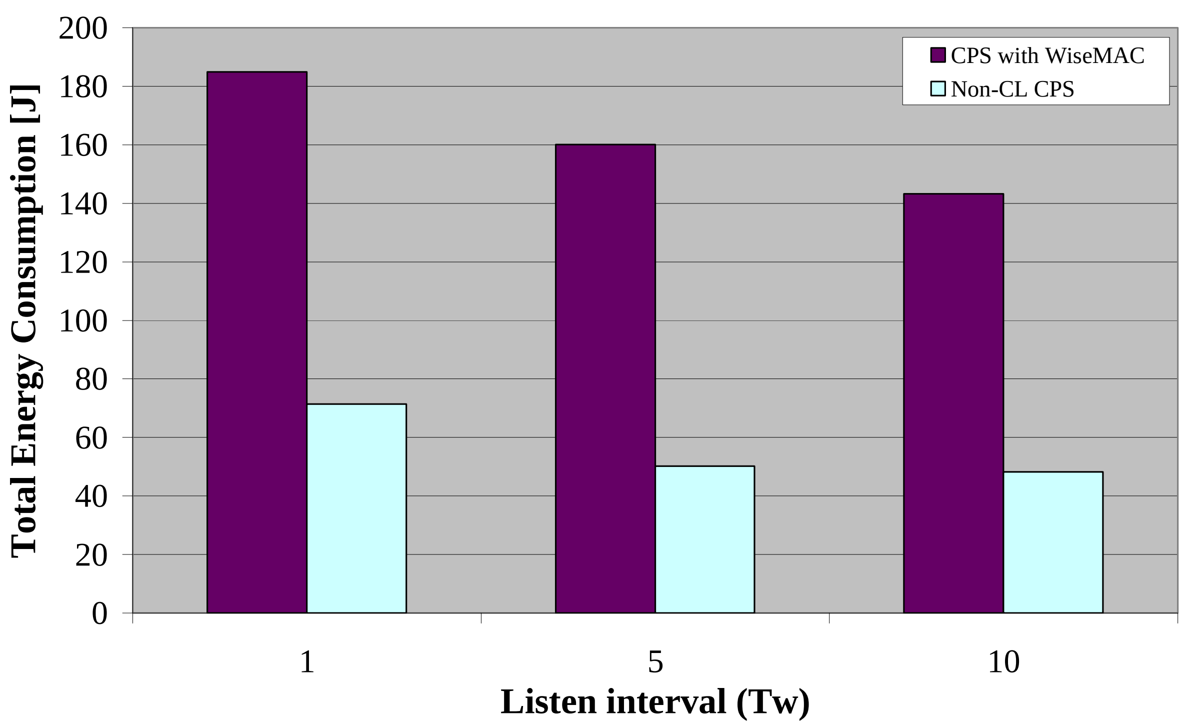

Figure 6.17 Total energy consumption of all nodes for CPS with WiseMAC and layered WSAN CPS for T_{sense} = 120s.

As mentioned before, the WiseMAC protocol is only suitable for low density networks that boast a smaller chance of overhearing. Here, too much overhearing occurs, where network nodes receive complete preambles plus data blocks that are not destined for them. This results in the consumption of an enormous amount of extra energy.

It is therefore established that the layered CPS with the existing WiseMAC protocol performs worse than the layered WSAN CPS design that contains the CSMA-MPS protocol with regards to energy efficiency of network nodes. All subsequent results given in the next sections include the layered WSAN CPS as a reference, indicated as 'Non-CL' in the graphs.

6.4.4.2 Higher Traffic Load Scenario

Simulated CPS configurations with a sensing interval of 60 seconds, representing a higher traffic load scenario, for corresponding MACL listen intervals of 1, 2 and 5 seconds are illustrated in Figure 6.19. It is observed that for all listen intervals, the total energy consumption actually decreases for CL configurations with DCD disabled. This is attributed to the fact that the cross-layer enabled CPS improves on the energy efficiency by always choosing a next-hop node that is less busy and experiences less CS failures, i.e. a less busy medium. Herewith, the less busy nodes spend less energy trying to repeatedly accessing the medium and as result a decrease in the total energy consumption is observed.

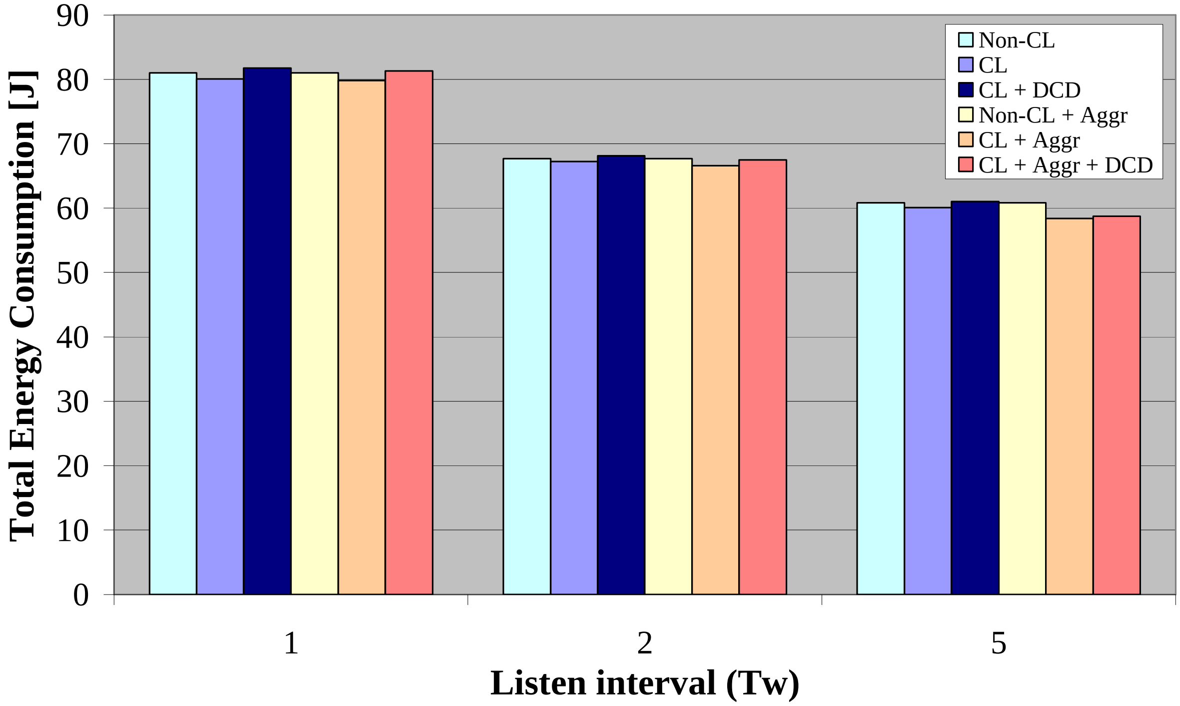

Figure 6.18 Total energy consumption for simulation configurations with sensing interval of T_{sense} = 60s.

For the cross-layer enabled CPS with DCD enabled however, the total energy consumption marginally increases compared to the Non-CL CPS configurations. For T_W = 1 second the increase is a meagre 0.9%, for T_W = 2 it is 0.68% and for T_W = 5 it is 0.4% for aggregation disabled configurations. This negligible minute increase in total energy consumption is an expected result since DCD doubles the duty cycle of nodes in the critical route towards the cluster-head actor node.

Simulated CPS configurations with a sensing interval of 120 seconds, representing a slightly lower traffic load scenario (compared to T_{sense} = 60s), for corresponding MACL listen intervals of 1, 5 and 10 seconds is illustrated in Figure 6.19. Similar results as observed for the sensing interval of 60 seconds is observed. A decrease in the total energy consumption for CL configurations with DCD disabled and a negligible minute increase for configurations with DCD enabled.

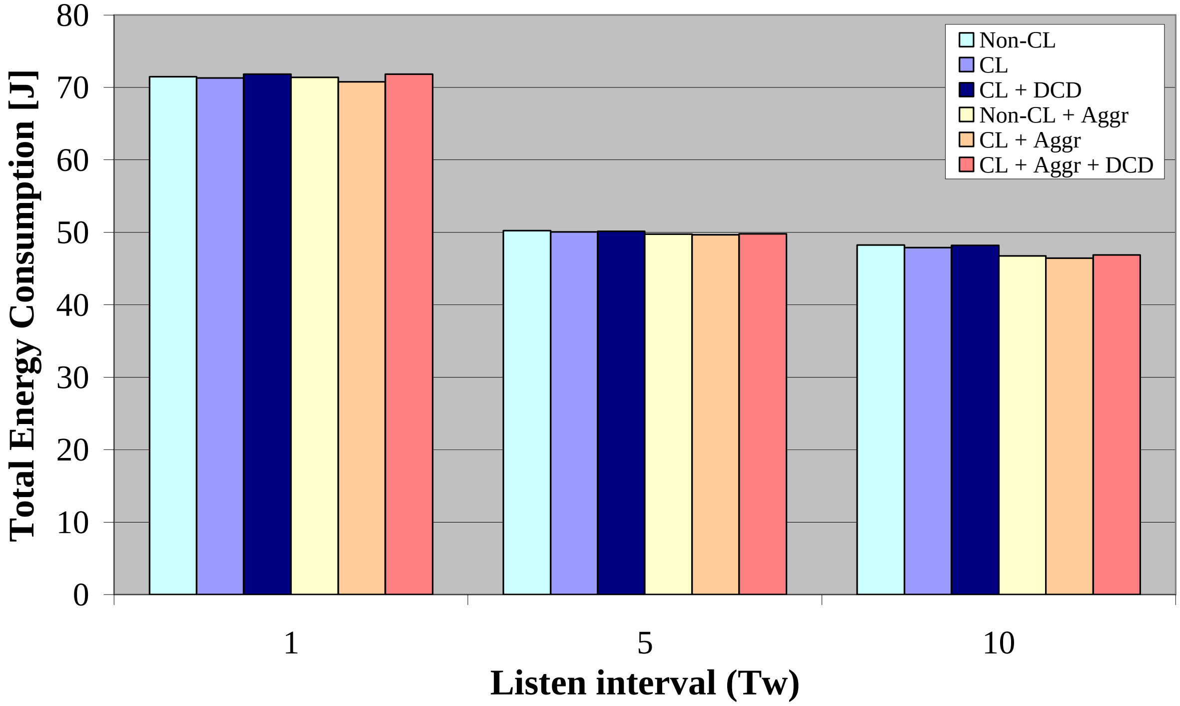

Figure 6.19 Total energy consumption for simulation configurations with sensing interval of T_{sense} = 120s.

6.4.4.3 Lower Traffic Load Scenario

Simulated CPS configurations with a sensing interval of 300 seconds, representing a lower traffic load scenario, for corresponding MACL listen intervals of 1, 5, 10 and 20 seconds are illustrated in Figure 6.20. Again, similar results as observed for the sensing interval of 60 seconds are observed for the listen intervals 1 and 5 seconds. For the listen intervals 10 and 20 seconds, the difference in total energy consumption is observed to be even smaller than for the low listen interval values.

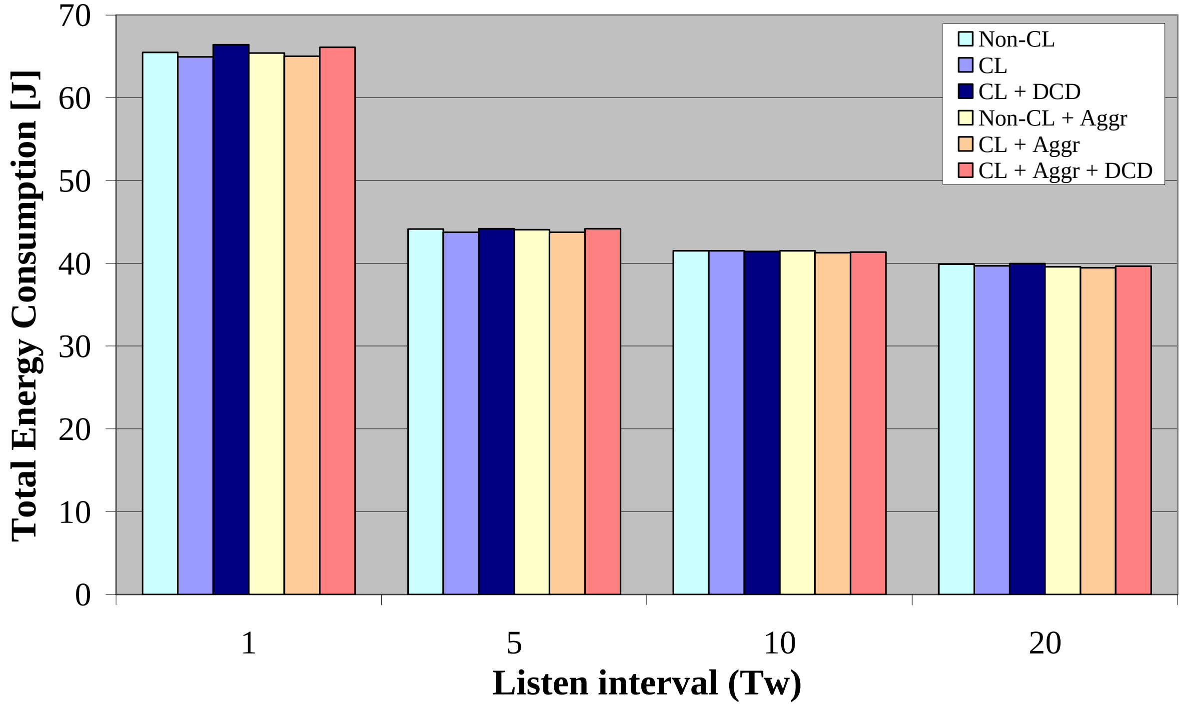

Figure 6.20 Total energy consumption for simulation configurations with sensing interval of T_{sense} = 300s.

Finally, simulated CPS configurations with a sensing interval of 600 seconds, representing a low traffic load scenario, for corresponding MACL listen intervals of 1, 10, 20 and 30 seconds is illustrated in Figure 6.21. It is observed that the energy efficiency difference is negligibly small for all simulated listen intervals. This is attributed to the fact that the nodes are idling most of the time since the traffic load is low.

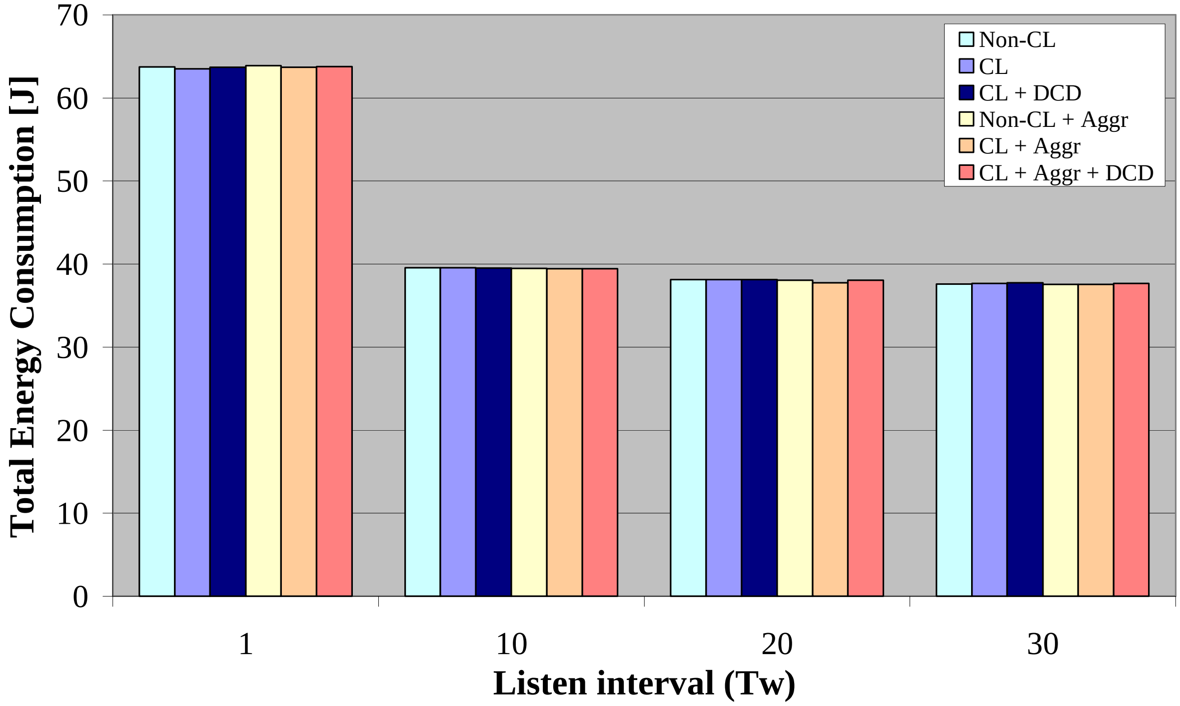

Figure 6.21 Total energy consumption for simulation configurations with sensing interval of T_{sense} = 600s.

In summary, it is therefore established that the energy efficiency is not negatively influenced by the adoption of the proposed cross-layer paradigm implementation. The latency minimization of critical data packets specifically proposed by the cross-layer enabled CPS, does not have a detrimental effect on the energy efficiency. Feasibility of the CPS WSAN is hence maintained.

6.4.5 Power Control Results

The use of transmitter power control at the PHYL is proposed by some specific existing WSAN NETL protocols that include the Hierarchical Energy-Efficient Routing with Ripple-Zones, POWER-SPEED and Power Aware Many-to-Many Routing protocols. Simulated CPS configurations with a sensing interval of 120 seconds for corresponding MACL listen intervals of 1, 5 and 10 seconds are illustrated in Figure 6.22. The non cross-layer CPS is simulated with power control enabled and disabled. The major drawback of employing transmitter power control is that the hidden terminal problem is exacerbated, and as observed leads to a clear increase in the number of collisions. For the listen interval of 1 second an increase of 56% is observed, 128% for 5 seconds and 64% for 10 seconds. A higher number of collisions leads to more retransmissions, resulting in higher energy consumption and increased latency. Therefore, it is established that power control is unsuitable for use in the WSAN.

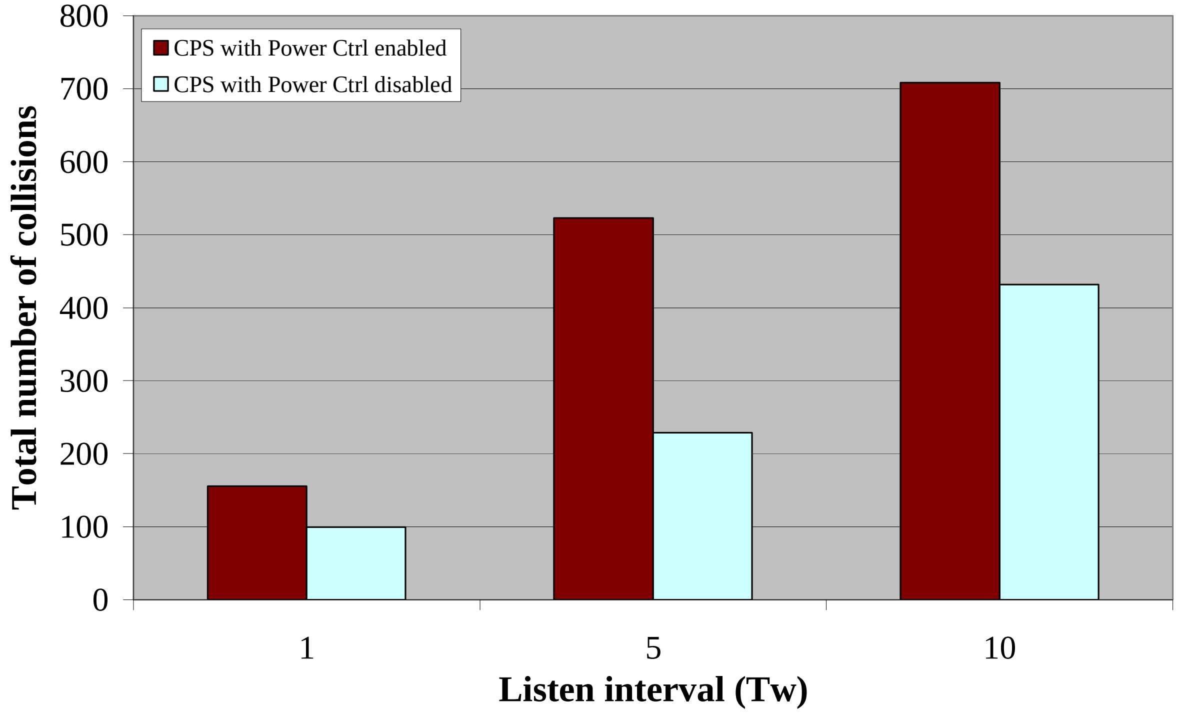

Figure 6.22 Total number of collisions for power enabled and disabled layered CPS for sensing interval of T_{sense} = 120s.

Furthermore, for such a simulated network topology where senor nodes are places as far apart as their transmission range allows (to attain as much coverage area as possible), the power control will not have any positive effect on energy efficiency since the radios will almost always be required to transmit at their maximum power.

6.5 DISCUSSION

The main observations that were made in this chapter are discussed as follows.

The results from the simulated cross-layer enabled WSAN CPS design with cross-layer enabled protocols as described in sections 5.8.1 (APPL: CL-NCAC-DP), 5.8.2 (NETL: CL-CHELAGR) and 5.8.3 (MACL: CL-CSMA-MPS-DCD) indicate that the average end-to-end latency of delay-sensitive critical data packets is improved by a large margin compared to the strictly layered CPS with layered protocols as described in sections 5.4.2 (APPL: NCAC-DP), 5.5.2 (NETL: CHEAGR) and 5.6.2 (MACL: CSMA-MPS). This applies to both higher and lower traffic load scenarios for a complete range of MACL listen intervals. Furthermore, the temporary employment of DCD for nodes in the critical reporting route provides a further increase in the latency performance by another substantial margin for both, aggregation enabled and disabled configurations. In general, the latency performance improvement increases as the listen interval (T_W) increases.

The average end-to-end latency of delay-insensitive non critical data packets, i.e. data packets that do not indicate a critical event, in the cross-layer enabled WSAN CPS is decreased compared to the non cross-layer, strictly layered CPS. This decrease is marginal since the cross-layer CPS focuses on latency minimization for delay-sensitive data packets only, to maintain overall energy efficiency. The average latency decreases more for higher listen interval values. Furthermore, the average latency decreases substantially with high listen interval values for the aggregation enabled configurations. It has therefore been established that by focusing on the minimization of delay-sensitive data packets rather than all data packets, the latency of delay-insensitive data packets is not compromised. In contrast, the latency is decreased especially for cross-layer CPS configurations with DCD enabled.

The total energy consumption results indicate that the focus shift to latency minimization for critical data packets does not have a detrimental effect on the overall network energy efficiency. Extremely low increases are observed. The highest increase is observed for cross-layer CPS configurations that have the DCD enabled. This is expected since DCD temporarily halves the listen slot interval for nodes used for critical event reporting, resulting in higher energy consumption. However, this increase in energy consumption is so minimal that it can be considered as insignificant. It is therefore established that the latency minimization of critical data packets specifically proposed by the cross-layer enabled CPS, does not have an unfavourable effect on the energy efficiency. Feasibility of the CPS WSAN is hence maintained. All in all, the cross-layer CPS design provides an excellent solution to the latency vs. energy trade-off that exists in these types of networks.

Due to the latency vs. energy trade-off, it was important to establish if by shifting the balance more to either one side, the network still remains efficient enough for it to be feasible. It is therefore established that the cross-layer paradigm is indeed beneficial to WSANs. The cross-layer enabled CPS design with DCD enabled within the MACL is the clear winner with regards to network efficiency performance. This applies to networks with higher and lower traffic loads and a full range of listen intervals.

It has been established that the use of power control is unsuitable for WSANs since the hidden terminal problem is exacerbated, leading to a higher collision probability. The higher the number of collisions, the more retransmissions occurs and consequently more energy is consumed and latency is increased.

Finally, ample reliability is achieved by the use of ACKs at the MACL for every packet transmission. In the event of a collision, a packet is simply re-transmitted, ensuring reasonable reliability.

6.6 CHAPTER SUMMARY

In this chapter, the simulation details and results were given of the proposed new cross-layer enabled CPS design in comparison to a layered CPS design that is based on existing layered WSAN protocols. The simulation results were presented and discussed. The three metrics that were presented and discussed are the latency improvements of critical delay-sensitive data packets, the latency of non critical delay-insensitive data packets and the total energy consumption. It was established that the cross-layer enabled CPS consistently outperforms the layered CPS with regards to latency minimization of important critical event reporting data packets and non critical data packets, especially with DCD enabled, while maintaining good energy efficiency.

The aim of this chapter has been to verify the proposed new cross-layer enabled CPS and to establish that the cross-layer paradigm indeed provides performance gains for WSANs.

Chapter

7 CONCLUSION

In this dissertation, a new cross-layer enabled complete WSAN CPS was developed and evaluated. The proposed WSAN CPS features the adoption of the cross-layer paradigm towards promoting optimization of network efficiency with regards to latency and energy. In order to implement the cross-layer paradigm, a new simple cross-layer framework was also proposed for the WSAN CPS.

Recent development has lead to the emergence of distributed Wireless Sensor and Actor Networks, which are capable of observing the physical world, processing the data, making decisions based on the observations and performing appropriate actions. WSANs represent an important extension of Wireless Sensor Networks. In WSANs however, the network comprises heterogeneous sensor nodes and actor nodes. The WSANs ability to make autonomous decisions and perform actions in response to sensor measurements provides a widespread potential for a wide variety of applications. It was shown that WSANs can be implemented in a variety of different application domains that include commercial, industrial, scientific, agricultural and military applications. WSANs can be an integral part of systems such as battlefield surveillance and microclimate control in buildings, nuclear, biological and chemical attack detection, target tracking, home automation, environmental monitoring and sensing and maintenance in large industrial plants.

It was revealed that the specific characteristics of sensor and actor nodes combined with some stringent application constraints impose unique requirements for WSANs. The fundamental challenges for WSANs are energy efficiency, latency minimization and reliability. Moreover, the latency and energy efficiency requirements are in a trade-off relationship. Communication and coordination within the WASN is managed by the CPS, implemented on every node. Careful design of protocols layers in the CPS was crucial in attempting to meet the unique requirements and handle the abovementioned trade-off relationship in WSANs. It was revealed that WSANs have only recently come into existence and little research has been conducted in this field that is showing increasing interest in a lot of different application areas.

A thorough review of existing literature on WSAN and relevant WSN protocols for the APPL, NETL and MACL was performed. Furthermore, existing literature on cross-layering as used in similar types of networks was also reviewed. It was revealed that the popular cross-layer paradigm that showed performance gains for similar types of networks was not yet researched for implementation in WSANs. It was shown that existing protocols focus primarily on energy efficiency. However, most WSAN applications require latency minimization to be considered with equal importance. Since the implementation of the cross-layer paradigm showed performance gains for similar types of networks, the research focused on implementing this favoured paradigm for the WSAN CPS, to facilitate the optimization of latency and energy efficiency. Specifically, a solution to the latency versus energy trade-off was needed.

First, a simple cross-layer framework design was proposed, followed by the actual complete WSAN CPS design that features strictly layered protocols only. The layered protocols includes: for the APPL the NCAC-DP protocol, for the NETL the CHEAGR protocol and for the MACL the CSMA-MPS protocol. These protocols are based on existing simple protocols and were adapted to conform to the defined basic operational scenario. The type of WSAN application that was considered as a design target was one with medium to low network traffic. The proposed layered APPL protocol features a network-centric actuation control approach with data prioritization. The NETL protocols features a cluster-based hierarchical energy-aware geographic routing approach and finally, the MACL a contention-based preamble sampling protocol, best suited for the defined network conditions and characteristics. Very importantly, parameters in every layer of the CPS were then identified that may be beneficial to other layers with regards to the optimization of latency and energy efficiency. Finally, a new cross-layer enabled WSAN CPS was proposed by taking the chosen layered protocols as a development basis and enhancing them to employ cross-layer interactions and incorporate information from other layers into their local decisions. The proposed cross-layer protocols includes: for the APPL the CL-NCAC-DP protocol, for the NETL the CL-CHELAGR protocol and for the MACL the CL-CSMA-MPS-DCD protocol. The cross-layer approach facilitates the cross-layer enabled protocols to distinguish between different data packet priorities and handle them in a specific manor. This is the main notion of the cross-layer enabled CPS. Data prioritization at the APPL enables a more focused response for particular types of data. Every layer in the CPS changes its approach depending on the current data packet type it is handling. Delay-sensitive data packets are handled in order to achieve minimum latency. On the other hand, delay-insensitive non critical data packets are handled in such a way as to achieve the highest energy efficiency. The NETL protocol changes its forwarding approach from along a path containing maximum residual energy nodes to along the fastest path with nodes that wake-up the earliest. The MACL adopts a priority queue together with a duty cycle doubling feature to ensure rapid relaying of delay-sensitive data packets. The CPS design provides a sensible solution to the latency versus energy efficiency trade-off. Either latency minimization or energy efficiency receives an elevated precedence according to the type of data that is to be handled.

Through different simulation configurations and network conditions, the strictly layered and cross-layer enabled CPS designs were compared, with regards to average end-to-end latency and energy consumption performances. The different simulation configurations entail the enabling and disabling of specific protocol features. It was shown that the cross-layer CPS indeed decreases the average latency of critical data packets by quite a margin as compared to the strictly layered CPS design. The latency performance increase is most pronounced when duty cycle doubling at the cross-layer enabled MACL is enabled. Furthermore, the average end-to-end latency of delay-insensitive non critical packets is also decreased, especially when the duty cycle doubling is enabled. Finally, the cross-layer enabled CPS shows consistent energy efficiency performance compared to the layered CPS. The cross-layer enabled CPS affects the total energy consumption of sensor nodes in the network only very marginally. The trivial increase in energy consumption is overshadowed by the high margin of increase in latency performance for delay-sensitive critical data packets. Hence, it can be concluded that the newly proposed cross-layer CPS achieves an immense latency performance increase for WSANs, specifically for critical data packets, while maintaining satisfactory energy efficiency. Furthermore, the end-to-end latency of delay-insensitive non critical packets also enjoys an increase in performance. The newly proposed cross-layer CPS design provides a good solution to the latency versus energy trade-off.

The simple cross-layer framework design incurs minimal overhead and proves to be very suitable for the WSAN CPS. Furthermore, the popular transmission power control feature used by several existing protocols proves to be impractical for use in WSANs.

Future research efforts may focus on more cross-layer features that may facilitate further performance gains. The APPL for instance specifically may incorporate more information from lower layers to increase the network efficiency performance. Furthermore, the actual actuation coordination within the APPL of sensor and actor nodes may benefit from information from other layers in the CPS in providing among others more effective actuation. Moreover, the adoption of the cross-layer paradigm is not limited to the chosen layered protocols. Other layered protocols, chosen for a different operational scenario with for instance no periodic data gathering, may also benefit and enjoy performance gains by incorporating information from other layers of the CPS. Challenges for WSANs with different operational scenarios and network topologies and what role cross-layering can play in other application scenarios should be researched.

Other WSAN issues that include time synchronisation between network nodes, localisation and specifically mobility need to be further researched. Cross-layer interactions between multiple layers may provide means to finding feasible solutions for these issues. Specifically, the mobility management of network nodes poses a mighty challenge. Mobility is a huge problem for the CPS, since nodes inside a network with a dynamic topology may have to update neighbourhood information more often. This in turn incurs a high traffic load on the network, increasing the latency and lowering energy efficiency. Research into how cross-layer interactions may be advantageous in providing a solution to these issues is needed.

REFERENCES

[1] A. Deshpande, C. Guestrin, and S. R. Madden, “Resource-Aware Wireless Sensor Actuator Networks”, *IEEE Data Engineering Bulletin*, vol. 28, no. 1, pp. 40-47, 2005.

[2] I. F. Akyildiz and I. H. Kasimoglu, “Wireless sensor and actor networks: Research challenges,” *Ad Hoc Networks (Elsevier)*, vol. 2, no. 4, pp. 351-367, October 2004.

[3] F. Thouin, R. Thommes, and M. J. Coates, “Optimal Actuation Strategies for Sensor/Actuator Networks,” 2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, July 2006, pp. 1-8.

[4] M. Haeggi, “Mobile sensor-actuator networks: opportunities and challenges,” in *Proceedings of 7th IEEE International Workshop*, Frankfurt, Germany, July 2002, pp. 283-290.

[5] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister, “Smart Dust: Communicating with a Cubic-Millimeter Computer,” *IEEE Computer*, vol. 34, no. 1, pp. 44-51, January 2001.

[6] M. Mauve, J. Widmer, and H. Hartenstein, “A Survey on Position-Based Routing in Mobile Ad Hoc Neworks,” *IEEE Network Magazine*, vol. 15, no. 6, pp. 30-39, November 2001.

[7] F. Hu, X. Cao, S. Kumar, and K. Sankar, “Trustworthiness in Wireless Sensor and Actuator Networks: Towards Low complexity Reliability and Security,” in *Proceedings of IEEE Global Telecommunications Conference GLOBECOM '05*, November 2005, vol. 3, pp. 1696-1700.

[8] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393-422, March 2002.

[9] M. Chow and Y. Tipsuwan, “Network-Based Control Systems: A Tutorial,” in *Proceedings of IEEE IECon 2001 Tutorial*, Denver, CO, November 2001, pp. 1593-1602.

[10] J. Ma, S. Zhang, Y. Zhong, and X. Tong, “SeLoc: Secure Localization for Wireless Sensor and Actor Network,” in *Proceedings of 2006 IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS)*, October 2006, pp. 864-869.

[11] Crossbow Technology Inc, MICA2 datasheet, Document Part Number 6020-0042-08, Rev A, Available: http://www.xbow.com. Last accessed July 2007.

[12] Intel Corporation, Intel Mote Project, http://www.intel.com. Last accessed July 2007.

[13] Chipcon AS, CC2400 datasheet (Rev. 1.5), Available: http://www.chipcon.com. Last accessed July 2007.

[14] E. C. H. Ngai, M. R. Lyu, and J. Liu, "A Real-Time Communication Framework for Wireless Sensor-Actuator Networks," in *Proceedings of 2006 IEEE Aerospace Conference*, Big Sky, Montana, USA, March 2006.

[15] K. Chintalapudi, E. A. Johnson, and R. Govindan, "Structural Damage Detection using Wireless Sensor-Actuator Networks," in *Proceedings of 13th Mediterranean Conference on Control and Automation*, Cyprus, June 2005, pp. 27-29.

[16] W. Zhang, G. Kantor, and S. Singh, "Integrated Wireless Sensor/Actuator Networks in an Agricultural Application," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, Baltimore, USA, 2004, pp. 317-317.

[17] A. Baggio, "Wireless sensor networks in precision agriculture," in Online *Proceedings of the REALWSN 2005*, Available: http://www.sics.se/realwsn05/proceedings.html, Last accessed July 2007.

[18] H. Zimmermann, "OSI Reference Model-The IS0 Model of Architecture for Open Systems Interconnection," *IEEE Communications*, vol. 28, no. 4, pp. 425-432, April 1980.

[19] J. D. Day and H. Zimmermann, "The OS1 Reference Model," in *Proceedings of the IEEE*, vol. 71, no. 12, pp. 1334-1340, December 1983.

[20] J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6-28, December 2004.

[21] T. Melodia, D. Pompili, I. H. Kasimoglu, V. C. Gungor, and I. F. Akyildiz, "A Distributed Coordination Framework for Wireless Sensor and Actor Networks," in *Proceedings of ACM MOBIHOC 2005*, Urbana-Champaign, IL, May 2005.

[22] H. Yuan, H. Ma, and H. Liao, "Coordination Mechanism in Wireless Sensor and Actor Networks," in *Proceedings of First International Multi-Symposiums IMSCCS '06 Computer and Computational Sciences*, 20-24 April 2006, vol. 2, pp. 627-634.

[23] F. Dressler, "Network-centric Actuation Control in Sensor/Actuator Networks based on Bio-inspired Technologies," in *Proceedings of 2006 IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, October 2006, pp. 680-684.

[24] C. J. Rozell and D. H. Johnson, "Evaluating local contributions to global performance in wireless sensor and actuator networks," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS)*, San Francisco, CA, June 2006, *Lecture Notes in Computer Science*, vol. 4026, pp.1-16, 2006.

[25] A. Rezgui and M. Eltoweissy, "A Service-Oriented Query Architecture for Sensor-Actuator Networks," in *Proceedings of 2006 2nd IEEE Workshop on Wireless Mesh Networks*, 2006, pp. 115-117.

[26] E. C. H. Ngai, Y. Zhou, M. R. Lyu, and J. Liu, "Reliable Reporting of Delay-Sensitive Events in Wireless Sensor-Actuator Networks," in *Proceedings of The Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'06)*, Vancouver, Canada, October 9-12, 2006.

[27] A. A. Ahmed, H. Shi, and Y. Shang," A survey on network protocols for wireless sensor networks," in *Proceedings of International Conference Information Technology: Research and Education (ITRE2003)*, 11-13 August, 2003, pp. 301-305.

[28] K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325-349, May 2005.

[29] A. Durresi, V. Paruchuri, and L. Barolli, "Delay-Energy Aware Routing Protocol for Sensor and Actor Networks," in *Proceedings of 11th International Conference on Parallel and Distributed Systems*, 20-22 July 2005, vol. 1, pp. 292-298.

[30] M. Haenggi, "Mobile sensor-actuator networks: opportunities and challenges," in *Proceedings of 7th IEEE International Workshop on Cellular Neural Networks and Their Applications*, Frankfurt, Germany, July 2002, pp. 283-290.

[31] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'01)*, Rome, Italy, July 2001.

[32] F. Hu, Y. Wang, and H. Wu, "Mobile telemedicine sensor networks with low-energy data query and network lifetime considerations," *IEEE Transactions on Mobile Computing*, vol. 5, no. 4, pp. 404-417, April 2006.

[33] Y. Lin and S. Megerian, "Low Cost Distributed Actuation in Large-Scale Ad Hoc Sensor-Actuator Networks," *2005 International Conference on Wireless Networks, Communications and Mobile Computing*, 3-16 June 2005, vol. 2, pp. 975-980.

[34] W. Hu, N. Bulusu, and S. Jha, "A Communication Paradigm for Hybrid Sensor/Actuator Networks," in *Proceedings of 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2004)*, 5-8 September 2004, vol. 1, pp. 201-205.

[35] J. Liu, E. C. H. Ngai, M. R. Lyu, and Y. Zhou, "POWER-SPEED: A Power-Controlled Real-Time Data Transport Protocol for Wireless Sensor-Actuator Networks," in *Proceedings of 2007 IEEE Wireless Communications and Networking Conference (WCNC)*, March 2007, pp. 3736-3740.

[36] T. Fuhrmann, "Scalable Routing in Sensor Actuator Networks with Churn," in *Proceedings of 2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON '06)*, 28 September 2006, vol. 1, pp. 30-39.

[37] E. Cayirci, T. Coplu, and O. Emiroglu, "Power Aware Many to Many Routing in Wireless Sensor and Actuator Networks," in *Proceedings of the Second European Workshop on Sensors, Actuators & Networking Lab.*, Istanbul Tech. Univ., Turkey, 31 January - 2 February 2005, pp. 236- 245.

[38] A. Durresi and V. Paruchuri, "Geometric Broadcast Protocol for Sensor and Actor Networks," in *Proceedings of 2005 19th International Conference on Advanced Information Networking and Applications*, 28-30 March 2005, vol. 1, pp. 343-348.

[39] S. Li, "Wireless Sensor Actuator Network for Light Monitoring and Control Application," in *Proceedings of 2006 3rd IEEE on Consumer Communications and Networking Conference (CCNC 2006)*, 8-10 January 2006, vol. 2, pp. 974-978.

[40] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC Protocols for Wireless Sensor Networks: A Survey," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 115-121, April 2006.

[41] M. Srivastava, "Part IV: Sensor Network Protocols," Center for Embedded Networked Sensing, University of California at Los Angeles, Tutorial, 2005, [Online]. Available: http://nesl.ee.ucla.edu/tutorials/mobicom05/. Last accessed July 2007.

[42] P. P. Czapski, "A Survey: MAC Protocols For Applications of Wireless Sensor Networks," *2006 IEEE Region 10 Conference (TENCON)*, November 2006, pp. 1-4.

[43] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, Chichester: Wiley, 2005, ch. 5.

[44] M. Miladi, T. Ezzedine, and R. Bouallegue, "Latency of Energy Efficient MAC Protocols for Wireless Sensor Networks," *2006 International Conference on Digital Telecommunications*, pp. 60-65.

[45] S. Srivathsan and S. S. Iyengar, "Minimizing Latency in Wireless Sensor Networks - A Survey," in *2007 Proceeding of Advances in Computer Science and Technology*, pp. 559-815.

[46] B. M. Sadler, "Fundamentals of Energy-Constrained Sensor Network Systems," *IEEE Aerospace and Electronic Systems Mag.*, vol. 20, no. 8, pp. 17-35, August 2005.

[47] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *2002 Proceedings of INFOCOM Conference*, pp. 1567-1576.

[48] C. Suh and Y.-B. Ko, "A Traffic Aware, Energy Efficient MAC Protocol for Wireless Sensor Networks," in *2005 Proceedings of ISCAS Symposium*, pp. 2975-2978.

[49] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *2003 Proceedings of SenSys Conference*, pp. 171-180.

[50] H. Pham and S. Jha, "An Adaptive Mobility-Aware MAC Protocol for Sensor Networks (MS-MAC)," in *2004 Proceedings of International Conference on Mobile Ad-hoc and Sensor Systems*, pp. 558-560.

[51] P. Lin, C. Qiao and X. Wang, "Medium Access Control With A Dynamic Duty Cycle For Sensor Networks," in *2004 Proceedings IEEE Wireless Communications and Networking Conference*, vol. 3, pp. 1534-1539.

[52] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks," in *2004 Proceedings of ISCC Symposium*, pp. 244-251.

[53] C. C. Enz, A. El-Hoiydi, J.-D. Decotignie, and V. Peiris, "WiseNET: an ultralow-power wireless sensor network solution," *IEEE Computer* Mag., vol. 37, no. 8, pp. 62-70, Aug 2004.

[54] S. Mahlknecht and M. Böck, "CSMA-MPS: a minimum preamble sampling MAC protocol for low power wireless sensor networks," in *2004 Proceedings of IEEE Workshop on Factory Communication Systems*, pp. 73-80.

[55] C. Schurgers, V. Tsiatsis, and M. B. Srivastava, "STEM: Topology management for energy efficient sensor networks," in *2002 Proceedings of Aerospace Conference*, pp. 1099-1108.

[56] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *2004 Proceedings of ACM SenSys Conference*, pp. 95-107.

[57] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *2000 Proceedings of Ann. Hawaii International Conference on System Sciences*, pp. 908-918.

[58] K. Sohrabi and G. J. Pottie, "Performance of a novel self-organization protocol for wireless ad-hoc sensor networks," in *1999 Proceedings of VTC Conference*, pp. 1222-1226.

[59] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network," *IEEE J. Personal Comm.*, vol. 7, no. 5, pp. 16-27, Oct. 2000.

[60] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," in *2003 Proceedings SenSys Conference*, pp. 181-192.

[61] T. W. Carley, M. A. Ba, R. Barua, and D. B. Stewart, "Contention-free Periodic Message Scheduler Medium Access Control in Wireless Sensor/Actuator Networks," in *Proceedings of 2003 24th IEEE Real-Time Systems Symposium*, 3-5 December 2003, pp. 298-307.

[62] R. Vedantham, Z. Zhuang, and R. Sivakumar, "Hazard Avoidance in Wireless Sensor and Actor Networks," in *Proceedings of 2005 2nd International Conference on Broadband Networks*, 3-7 October 2005, vol. 1, pp. 690-702.

[63] K. Morita, K. Watanabe, N. Hayashibara, T. Enokido, and M. Takizawa, "Efficient Data Transmission in a Lossy and Resource Limited Wireless Sensor-Actuator Network," in *Proceedings of 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 7-9 May 2007, pp. 156-163.

[64] N. A. Vasanthi and S. Annadurai, "Sleep Schedule for Fast and Efficient Control of Parameters in Wireless Sensor-Actor Networks," in *Proceedings of the* First International Conference on Communication System Software and Middleware (Comsware), 8-12 Janyary 2006, pp. 1-6.

[65] K. D. Wong, "Physical Layer Considerations for Wireless Sensor Networks," in *Proceedings 2004 IEEE International Conference on Networking, Sensing and Control*, vol 2, pp. 1201-1206.

[66] R. Winter, J. H. Schiller, N. Nikaein, and C. Bonnet, "CrossTalk: Cross-Layer Decision Support Based on Global Knowledge," *IEEE Communications Magazine*, vol 4, no. 1, pp. 93-99, January 2006.

[67] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *IEEE Wireless Communications*, vol. 12, no.1, pp. 3-11, February 2005.

[68] A. J. Goldsmith and S. Wicker, "Design challenges for energy-constrained ad-hoc wireless networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 8-27, August 2002.

[69] R. Knopp et al., "Overview of the Widens Architecture, A Wireless Ad Hoc Network for Public Safety," in *Proceedings of SECON Poster session*, 2004.

[70] M. Conti, G. Maselli, G. Turi, and S. Giordano, "Cross-layering in mobile ad hoc network design," *Computer Magazine*, vol. 37, no. 2, pp. 48-51, February 2004.

[71] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer, "Application-driven cross-layer optimization for video streaming over wireless networks," *IEEE Communications Magazine*, vol. 44, no. 1, pp. 122-130, January 2006.

[72] I.F. Akyildiz, M. C. Vuran, and O. B. Akan, " A Cross-Layer Protocol for Wireless Sensor Networks," in Proceedings of 2006 40th Annual Conference on Information Sciences and Systems, 22-24 March 2006, pp. 1102 - 1107.

[73] C. J. Merlin and W. B. Heinzelman, "A first look at a cross-layer facilitating architecture for wireless sensor networks," in *Proceedings of the 2006 2nd IEEE Workshop on Wireless Mesh Networks (WiMesh)*, pp. 103-105.

[74] A. M. Safwat, "A novel framework for cross-layer design in wireless ad hoc and sensor networks," in Proceedings of 2004 IEEE Global Telecommunications Conference Workshops, 29 November - 3 December 2004, pp. 130-135.

[75] L. van Hoesel, T. Nieberg, W. Jian, and P.J.M. Havinga, "Prolonging the lifetime of wireless sensor networks by cross-layer interaction," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 78-86, December 2004.

[76] T. Melodia, M. C. Vuran, and D. Pompili, "The State of the Art in Cross-Layer Design for Wireless Sensor Networks," Wireless System./Network Architect. 2005, Springer Lecture Notes in Computer Science (LNCS) 3883, pp. 78-92, 2006.

[77] X. Wang, J. Yin, Q. Zhang, and D. P. Agrawal, "A cross-layer approach for efficient flooding in wireless sensor networks," in *Proceedings of the 2005 IEEE Wireless Communications and Networking Conference*, 13-17 March 2005, vol. 3, pp. 1812-1817.

[78] P. Papadimitratos, A. Mishra, and D. Rosenburgh, "A cross-layer design approach to enhance 802.15.4," in *Proceedings of the 2005 IEEE Military Communications Conference*, 17-20 October 2005, vol. 3, pp. 1719-1726.

[79] R. Jurdak, P. Baldi, and C. V. Lopes, "Adaptive Low Power Listening for Wireless Sensor Networks," *IEEE Mobile Computing*, vol. 6, no. 8, pp. 998-1004, August 2007.

[80] R. W. Ha, P.-H. Ho, and X. S. Shen, "SS-trees: a cross-layer organizational approach for mesh-based wide-area wireless sensor networks," in *Proceedings of 2005 2nd International Conference on Broadband Networks*, 3-7 October 2005, vol. 2, pp. 823-832.

[81] M. Rossi and M. Zorzi, "Integrated Cost-Based MAC and Routing Techniques for Hop Count Forwarding in Wireless Sensor Networks," *IEEE Mobile Computing*, vol. 6, no. 4, pp. 434-448, April 2007.

[82] C.-F. Chou and K.-T. Chuang, "CoLaNet: a cross-layer design of energy-efficient wireless sensor networks," in *Proceedings of 2005 Systems Communications*, 14-17 August 2005, pp. 364-369.

[83] R. Madan, S. Cui, S. Lall, and N. A. Goldsmith, "Cross-Layer Design for Lifetime Maximization in Interference-Limited Wireless Sensor Networks," *IEEE Wireless Communications*, vol. 5, no. 11, pp. 3142-3152, November 2006.

[84] S. Cui, R. Madan, A. Goldsmith, and S. Lall, "Joint routing, MAC, and link layer optimization in sensor networks with energy constraints," in *Proceedings of 2005 IEEE International Conference on Communications(ICC)*, 16-20 May 2005, vol. 2, pp. 725-729.

[85] H. Kwon, T. H. Kim, S. Choi, and B. G. Lee, "Cross-layer lifetime maximization under reliability and stability constraints in wireless sensor networks," in *Proceedings*

of 2005 IEEE International Conference on Communications (ICC), 16-20 May 2005, vol. 5, pp. 3285-3289.

[86] J. Ma, S. Zhang, Y. Zhong, and X. Tong, "SeLoc: Secure Localization for Wireless Sensor and Actor Network," in *Proceedings of 2006 IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, October 2006, pp. 864-869.

[87] N.T. Herscovici, C. Christodoulou, A.I. El-Osery, W. Abd-Almageed, and M. Youssef, "Wireless corner - Calibration-free RF-based localization algorithm for sensor actuator networks using particle filters," *IEEE Antennas and Propagation Magazine*, vol. 48, no. 4, pp. 166-173, 4 August 2006

[88] M. Löbbers, and D. Willkomm, "A Mobility Framework for OMNeT++," User Manual, Version 1.0a4, Available: http://mobility-fw.sourceforge.net/, Last accessed October 2007.

[89] D. B. Green, and A. S. Obaidat, "An accurate line of sight propagation performance model for ad-hoc 802.11 wireless LAN (WLAN) devices," Proceedings of 2002 IEEE International Conference on Communications, 28 April - 2 May 2002, vol. 5, pp.3424-3428.